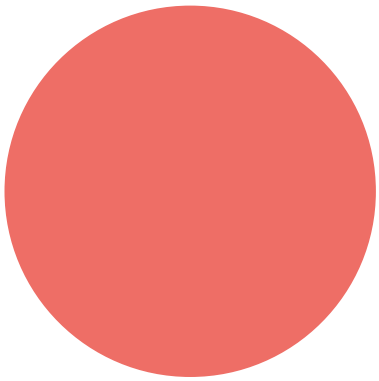
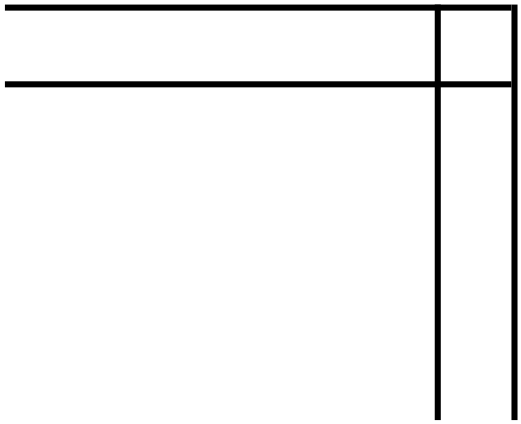
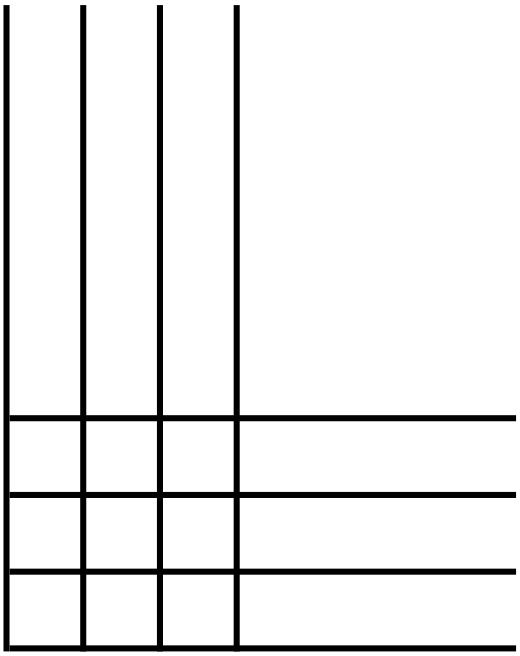


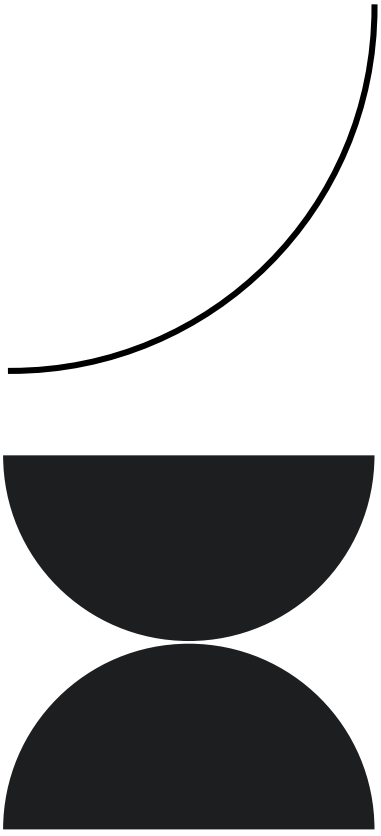


The Building Blocks Of a UI Sandwich

Examining the Efficiency and the Customization of Design Systems

Janne Koivisto





The Building Blocks Of a UI Sandwich

Examining the Efficiency and the Customization of Design Systems

Janne Ilkka Sakari Koivisto

School of Arts, Design and Architecture

Master of Arts Thesis

Master's Programme in New Media Design & Production

Aalto University, School of Arts, Design and Architecture

Supervisor: Teemu Leinonen

Advisor: Ville Sirén

Aalto Media Lab - 1st of September 2019

Aalto University
School of Arts, Design
and Architecture

Aalto University, P.O. BOX 31000, 00076 AALTO
www.aalto.fi
Master of Arts thesis abstract

Author Janne Ilkka Sakari Koivisto

Title of thesis The Building Blocks Of a UI Sandwich: Examining the Efficiency and the Customization of Design Systems

Department Department of Media

Degree programme New Media Design & Production

Year 2019 **Number of pages** 109 **Language** English

Abstract

Design systems are everywhere. Designers working in the digital user interface industry in the 2010s cannot avoid bumping into them. Much of the discussion surrounding design systems arise from the efficiency gain to the team and customer.

From a research standpoint, this thesis sets out to draw clear lines for design systems. As little to no academic research into design systems as a concept has been done, this thesis creates categorization method for design systems with examples. Also, this thesis shows what are seen as the foundations for design systems in the industry literature. This literature is studied in the historical context of other systematic design methods.

Using design systems makes a common look and feel to multiple services and enables designers to quick-start their work with an existing asset bank. However, for designers working with multiple interfaces and visual styles, design systems can become a demerit and more of an obstacle than an enabler. Customization of design systems is still problematic as they are built with a single source of truth.

This need for customization is tested out with a concept for designers – a sandbox in which designers can play around with concepts, customize them and quickly find them. The same concept is also tested with developers to gain insight into team work with systematic way of working in design.

This thesis ends with a conclusion about how design systems serve designers and developers and outlines the systems' benefits and limitations.

Keywords design system, UX, UI, designer, usability, mobile device, iPhone, teamwork

Aalto-yliopisto
Taiteiden ja suunnittelun korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Taiteen maisterin opinnäytteen tiivistelmä

Tekijä Janne Ilkka Sakari Koivisto

Työn nimi The Building Blocks Of a UI Sandwich: Examining the Efficiency and the Customization of Design Systems

Laitos Median laitos

Koulutusohjelma New Media Design & Production

Vuosi 2019 **Sivumäärä** 109 **Kieli** englanti

Tiivistelmä

Design systemit ovat kaikkialla. Suunnittelijat käyttöliittymäalalla eivät voi välttyä termiltä 2010-luvun ammatillisessa keskustelussa. Suurin osa siitä keskittyy design systemien työtehokkuuden lisäämiseen.

Akateemisesta näkökulmasta, tämä työ pyrkii rajaamaan design system-käsitteen rajoja, sillä aiheesta ei ole ammatillisen kirjallisuuden lisäksi yhtään akateemista tutkimustyötä. Tässä työssä pyritään esimerkiksi muovaamaan kategorisointimetodi erilaisille design systemeille eri esimerkein. Tässä työssä pyritään määrittämään myös design systemin tärkeimmät elementit mitä ammattikirjallisuudessa mainitaan. Ammattikirjallisuus on myös reflektoitu historian kontekstissa muista systemaattisista muotoilukonventioista.

Design systemien käyttö yhtenäistää usean palvelun ulkoasuja sekä auttaa suunnittelijoita aloittamaan työnsä nopeasti valmiilla design systemin kirjastolla. Suunnittelijat, jotka tekevät töitä usean eri visuaalisen tyylin ja käyttöliittymän kanssa, voivat kuitenkin kokea design systemin käytön monimutkaistavaksi. Design systemien mukauttaminen on edelleen erittäin ongelmallista kun ne perustuvat yhteen totuuteen.

Tämä mukauttamistarve testataan työssä suunnittelijoille tarkoitetussa konseptissa – hiekkalaatikkomallissa, jossa suunnittelijat voivat testata eri palasia ja muokata niitä vapaasti ja löytää ne nopeasti. Sama konsepti testataan myös kehittäjien kanssa jotta ymmärretään myös paremmin design systemien vaikutusta tiimityöskentelyyn.

Tämän pro gradu-työn päätännässä keskustellaan design system-konseptin roolia suunnittelijoille ja kehittäjille sekä määritellään rajat sille, mikä design systemin muodostaa.

Avainsanat design system, käyttöliittymäsuunnittelu, käytettävyys, mobiilikäyttöliittymä, iPhone, tiimityöskentely

Acknowledgements

This paper has been guided by my supervisor Teemu Leinonen and advisor Ville Sirén whom I wish to thank for the guidance throughout the process. Your input and help have brought this together. A big thank you!

Thanks to IBM Finland and to my team at IBM iX for the possibility to dedicate this project to you and providing me with insight, inspiration and endless motivation to make this happen. Moreover, thanks to the folks in Austin for your endless push to redevelop Carbon and supporting me in this research.

I wish to thank Aalto University's Media Lab and USchool for breaking me out of my shell as a designer. Philip, Jonna and Pipsa and the rest of the staff at DoM, it was also a tremendous pleasure working with you in Media Lab.

Ira Laitakari-Svärd, you have been a superb mentor over the time of my studies. Thank you for all your insight, motivation and help, both professionally, and academically.

Many people have been very supportive throughout my studies. Thanks to my family for supporting me. Misael, Tomi, Julius, Marianna, and to all my friends in Finland, and abroad, thank you.

Thank you Laura for being the most important part of my life. And Tyyne for being such a silly and awesome dog, coming to bug me at regular intervals when I was sitting still and writing too long.

Terminology

Carbon	IBM's open-source design system for digital products and experiences.
Client	A person or company using the commercial services of an other company.
Designer	A profession in which a person creates and oversees the development of an outlook or functionality of a service.
Design System	A set of pre-existing tools and assets for creating digital user interfaces.
Kit	A visual set of user interface components, patterns, guidelines and templates usable by anyone.
Mobile device	A portable electronic device with a user interface, such as a phone or tablet.
OS	An operating system running on a device, such as Android or iOS on mobile devices or Windows or MacOS on desktop computers.
Repository	An electronic location where files of a single project are stored.
React	A library in JavaScript code language for building mostly single-page user interfaces, maintained by Facebook and other individuals.
Sketch	A computer program for creating vector graphics and storing them in several formats.
Stakeholder	A person, client or company with interest or input in a particular issue.
Usability	Ease of access and use of a product or a digital service.
UI	User Interface, sometimes referred as Graphical User Interface, a way of presenting (interactive) content on a display in graphical format.
UX	User Experience, the overall experience of a user interacting with a product or a service.

Contents

Introduction

A person designing digital user interfaces has to deal with many issues. The building blocks that are used to create an interface from an initial sketch on a piece of paper to the finalized product on an end-user's device can vary vastly. Brad Frost, the author of *Atomic Design* (2013), says the following:

> *"What is an interface made of? What are our Lego bricks? What are our Subway sandwich pieces that we combine into millions of delicious combinations?" (Frost, 2013)*

A designer has to take into account not only one combination of elements, but multiple. Moreover, a hypothetical UI sandwich needs to be customized to the combination required over and over. An essential ingredient - that is an element of an interface, such as a button - would require great effort to make it to the end customer if it were redesigned multiple times for each combination. We need to think of these elements as ready-made, out-of-the-box ingredients, such as baked bread, which require smaller effort from the sandwich creator than starting to bake the bread for each sandwich upon the order.

This is where design with a systematic approach comes

into place. With evolving tools, user interface designers turn to ready-made assets since this shortens the time-span of designing assets for different digital solutions.

Systematic design methods are not a new concept in user-interface construction or design. An example of a human-created user interface is the printed book, for which systematic printing machine was invented by Johannes Gutenberg in 1430s with interchangeable lettering and capacity to print books in larger quantities and faster (“Printing Press” n.d.). As the digital tools used by modern day designers have the capacity to change the building blocks in the same way as the printing machine operator would change the lettering, this same methodology changes the way designers work.

Context

As a UX or UI designer in the late 2010s, it is inevitable to come across design systems. For example, there is an active design systems *Slack*-workspace, design system conferences, design system meetups, publications and websites. Design systems, be it a single file sitting on a designer’s laptop or a widely synced library, are a fundamental part of UX or UI designers’ daily workflow.

This thesis looks at design systems from a designer’s point of view and the role design systems take in their daily work. Moreover, the personal objective is to try to understand how to benefit from design systems in a consulting design work in the field of multiple different clients. The paper does not try to answer the role of design systems to developers, clients or other stakeholders, but merely to a designer as part of a larger team.

Purpose of Research

This study aims to clarify the discussion surrounding design systems. It plots the path from first design patterns to current status of digital libraries in user interface design area. As the terminology in the field is overlapping, misused and often mixed causing confusion, in the following chapters the concepts in current discussion are studied in a historical and the industry context, since many of these concepts are still new to academia. Through defining a framework surrounding design systems, this thesis aims to underline the need for clarification and further studies regarding the field - a unified conception on what defines a design system and what are the entities that form these systems. After gaining an understanding the theoretical concept of the topic, the knowledge is used to develop a concept with design systems for a specific team developing mobile apps.

This developed concept aims to increase efficiency and understanding at a workplace between different stakeholders; developers, designers, clients and project managers. Using an existing design system is considered as a starting point which is assessed through user testing. The concept developed is mirrored against the system to see if efficiency has been improved and if the team has found the concept beneficial. The user testing is used to test the claims of using design systems in the industry.

From Lists to Modules: the Path to Systematic Design

This chapter aims to define the concept of design systems using the theory and knowledge in the industry and academia. It sets out boundaries for methodology as well as defines more closely what a design system is and is not. Beginning with the earliest patterns in design, this chapter places design systems into the context of the ever-evolving ways of working and of design practice. Then, this chapter aims to provide tools for tackling the afore-mentioned research questions in detail and, moreover, to handle them with theoretical data against practical meth-odological data as discussed later.

Patterns and Design Problems

Patterns are referred to as recurring solutions in user interface design. They offer a solution to a common problem that can be replicated multiple times (Curtis, 2009). The first mentions of patterns in the user-centered design field are from the year 1977 when Christopher Alexander, Sara Ishikawa and Murray Silverstein published their book *A Pattern Language* (Malone, 2017). It outlined the way of designing buildings, towns and construction via a systematic approach to achieve a timeless way of

building for more than 250 different types of buildings (1977). Alexander et al. say that “[...]towns and buildings will not be able to become alive, unless they are made by all the people in society, and unless these people share a common pattern language[.]” (p. 10) This language, therefore, encompasses more people into design process in a co-design-like manner, meaning a fundamental change in the client-designer relationship and resulting in more user-centered design results (Chisholm, 2015). One could also argue that this was the birthpoint of user centered design for user interfaces, a full ten years before Don Norman’s *User Centered System Design* in 1987 and the birth of interaction design as a medium.

Nathan Curtis (2017) summarizes Alexander et al.’s method as one following this pattern:

Name [of the object]

Problem Statement

Sensitizing Picture (always at the top)

Use When (and often Don’t Use When)

Guidelines rich with examples and scenarios

From Alexander et al.’s pattern language in 1977 to current day, at the center of every design process, there is still a design problem. Nowadays, a designer’s goal overall is to try to reach a solution to a problem, which is formed in design thinking exercises into a problem statement, whereas Alexander et al.'s language describes the ready-made problem (Lee, 2017). A current-day problem statement, according to the Interaction Design Foundation’s Rikke Dam and Teo Siang (Dam & Siang, 2019), might be

formulated into the following sentence:

> *[User . . . (descriptive)] needs [need . . . (verb)] because [insight. . . (compelling)]*

The three elements - user, needs and insight - are three of the main elements that create a *Point of View* (POV) statement, which helps designers to iterate on a solution-first and goal-oriented manner. Another common design problem statement in the current day is the *How Might We-statement* (HMW), which is a broader statement igniter. HMW-statements are meant to ask for more concrete solutions (Dam & Siang, 2019). In comparison, Alexander et al.'s problem statements are told from the designer's own point of view and they are not formulated sentences, but echo the writers' work experience. Consider the following example about designing such stairs in public spaces where people could also sit (p. 604): "The trouble is that this [elevated vantage point] will usually have the effect of removing a person from the action. Yet most people want to be able to take the action in and to be part of it at the same time." This problem statement makes the assumption that most people want to be able to take and be a part of an action at a vantage point - similar to the current formulation as to a user group needing something to gain insight. As to how to proceed from this design problem, Alexander et al. propose the solution as presented in figure 1. (p. 605)

Since the publication of *A Pattern Language*, the approach to creating guidelines and systems for design have started with a user-centered focus: creating solutions and designs that serve "people in society" (p. 10) and solve their prob-

In any public place where people loiter, add a few steps at the edge where stairs come down or where there is a change of level. Make these raised areas immediately accessible from below, so that people may congregate and sit to watch the goings-on.

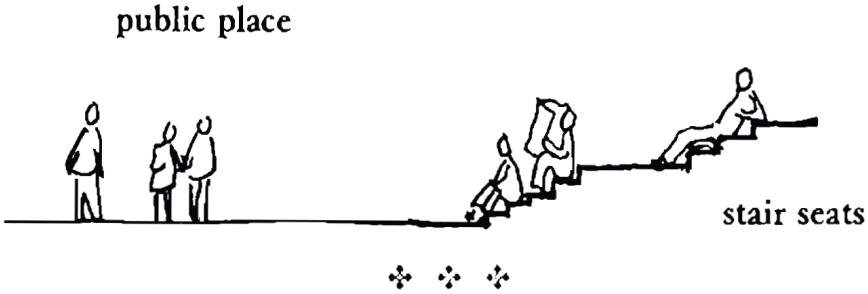


Figure 1. Alexander et al.'s example of a design pattern for public space planning. Often accompanied with imagery, the pattern is described alongside the image.

lems. This was before the first usability studies conducted in the 1980s and 1990s (Gould & Lewis, 1985). However, these patterns only solve one problem at a time, where other systematic solutions might offer concrete tools to replicate into the design. In the digital context, these were introduced as pattern libraries.

Pattern Libraries in Digital User Interface Design

The first signs of pattern libraries emerged to the digital context through *Design Patterns: Elements of Reusable Object-Oriented Software*, which was largely based on Alexander et al.'s methodology (Gamma, Helm, Johnson, & Vlissides, 1994). These pattern libraries proposed ready-made solutions to issues that had been solved by designers beforehand and that knowledge, therefore,

been shared (p. 11): “One thing expert designers know not to do is solve every problem from first principles. Rather, they reuse solutions that have worked for them in the past. When they find a good solution, they use it again and again. [...] These patterns solve specific design problems and make object-oriented designs more flexible, elegant, and ultimately reusable. They help designers reuse successful designs by basing new designs on prior experience.” Generating a pattern or using a pre-existing framework to solve design problems could therefore be a powerful tool for a designer. Overall, pattern libraries are collections of design elements that appear repeatedly on a specific user interface (Fanguy, 2018).

This, however, might lead to the critique of a design looking alike. *Bootstrap*, a front-end toolkit for rapidly developing web applications created by Twitter, for example has become popular for designing and building websites among developers and designers (Hajdarbegovic, 2018; “Bootstrap from Twitter”, 2011). With a ready-made pattern library, results might start to look a lot alike and therefore lack uniqueness, as Mary Collins (2016) points out with a visual comparison of *Bootstrap* web page themes in *WordPress* in figure 2.

Nevertheless, using popular pre-existing libraries and solutions have their benefits as well. *Bootstrap* has been proven to be an effective layout for responsive design, meaning scaling and functioning well from desktop environments to mobile screens (Caliman, 2015).

Gamma et al. proposed this solution well ahead of *Bootstrap* in their introduction of their design patterns. They work well and solve problems, which in turn makes designers worry less about solving the same problems over and over again.

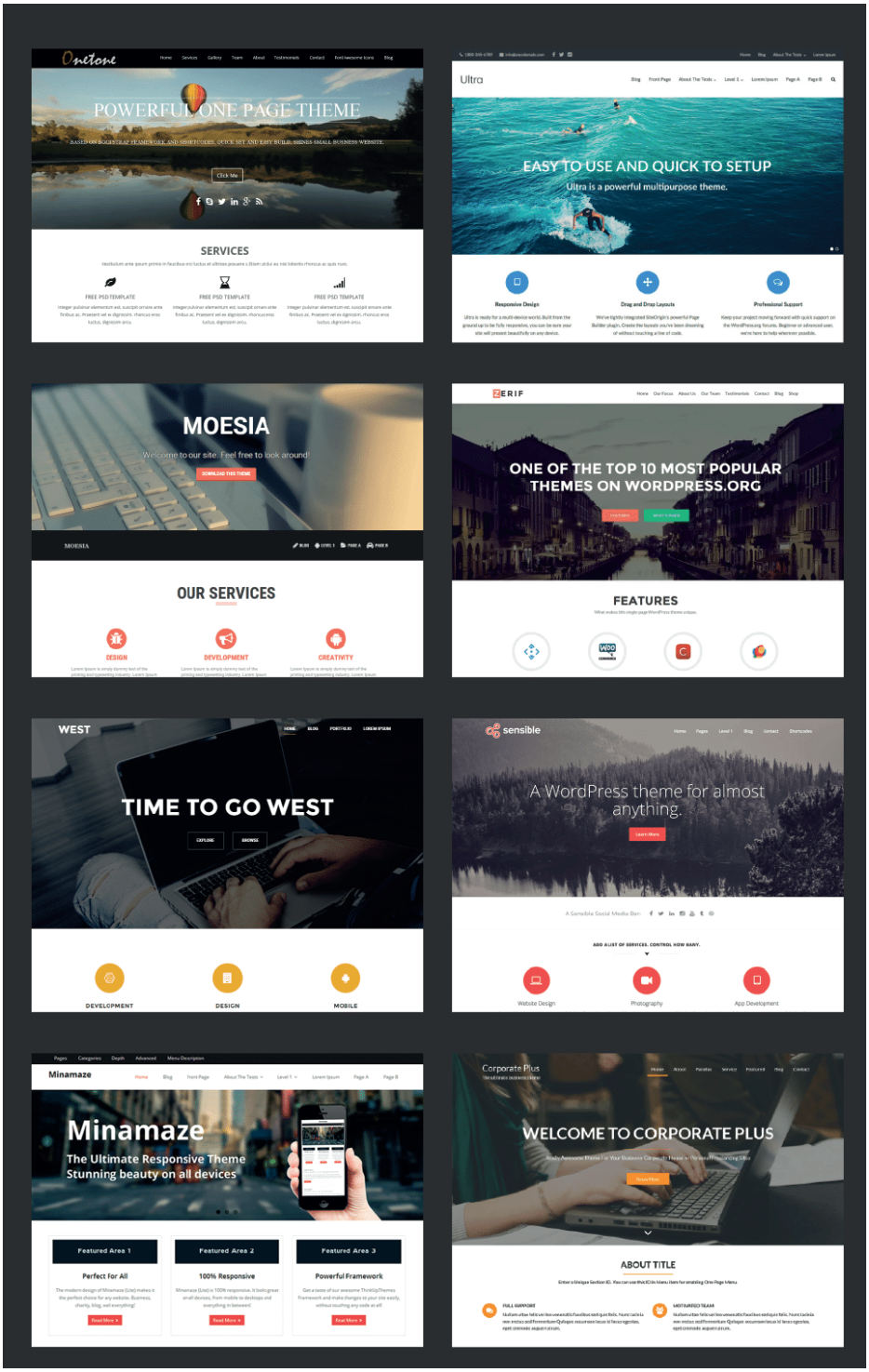


Figure 2. Mary Collins' comparison of Bootstrap websites in Wordpress.

This problem solving was supported also after *Design Patterns* by Jennifer Tidwell (2005) who proposed patterns for effective interaction design in more than one platform. Using patterns for design is proven effective in her piece since “[e]ven though individuals are unique, people behave predictably. [...] So when you observe people using your software, or performing whatever activity you want to support with new software, you can expect them to do certain things. [...A]n interface that supports these patterns well will help users achieve their goals far more effectively than interfaces that don’t support them.” (p. 10) Solutions such as *Bootstrap*, in their lack of uniqueness, might prove effective. What is needed from the designer, is customizability to meet the end-users’ needs and client specifications. Sometimes only one aspect might be changed, where modular structure plays a big role.

Modular Design

The object-oriented programming as presented by Gamma et al. is an example of designing a system with using modular design principles, which became the next systematic approach after pattern libraries (Frost, 2013). Modular design relies on the same principle of repeatability as pattern libraries but with a more systematic approach. If the design elements of a product are treated as modules, they can be easily clustered into larger entities, leading to more independence to the designer (Berners-Lee, 2013). In difference to Alexander et al.'s and Gamma et al.’s design patterns, modular design introduced the idea of an agile approach to design; more rapid and shorter iterations of a solution, building on top of the previous builds (Gallagher, Dunleavy, & Reeves, 2019). Continuing with the idea of rapid iteration through more repeatability, with a modular approach maintenance

of such a pattern library would also be a key issue to modular design, as Berners-Lee (2013) describes: “[W]hen you want to change the system, you can with luck in the future change only one part, which will only require you to understand (and test) that part. This will allow other people to independently change other parts at the same time.”

This modular thinking is largely how platforms such as the code hosting platform *GitHub* work currently. Repositories - meaning a storage space containing all the data one specific project needs such as folders, data, images and code - can be changed by all project members in the repository and therefore synced and updated to all the owners of the project. The parts are divided typically into separate files containing different information and functionalities, meaning that only parts of the project could be edited without updating the whole project (“Hello World”, 2016). However, these repository structurings can differ largely based on which code language is used and how. For some languages, such as JavaScript’s *React*, a single file can signify just one element whereas for some the whole styling of a website can be included in a single file, such as in CSS.

The aforementioned Bootstrap in fact is hosted and still found in *GitHub* and remains one of the most popular repositories in the service. Regarding the user interface design, *Bootstrap* is referred to as a toolkit for building web-based solutions. These tools in the toolkit are a more developer-friendly set of frameworks for starting design. Moreover “[f]ront-end frameworks are tools that provide a specific solution and a particular look and feel” (Frost, 2013, p. 22). Frameworks are modular by nature and they focus on a single project or design problem (Frost, 2013). For multiple design problems, frameworks, pattern libraries or modular design by themselves are not enough

to be used systematically. These are bonded together into design systems.

Design Systems

A design system is a collection of reusable components, such as a visual sketch of an element accompanied by a code snippet, guided by clear standards, that can be assembled together to build any applications (Fanguy, 2019). Systems in design are living organisms that are constantly updated and maintaided (Frost, 2013). Previ-ously discussed pattern libraries might focus on a related issue to solve with components, guidelines and use cases

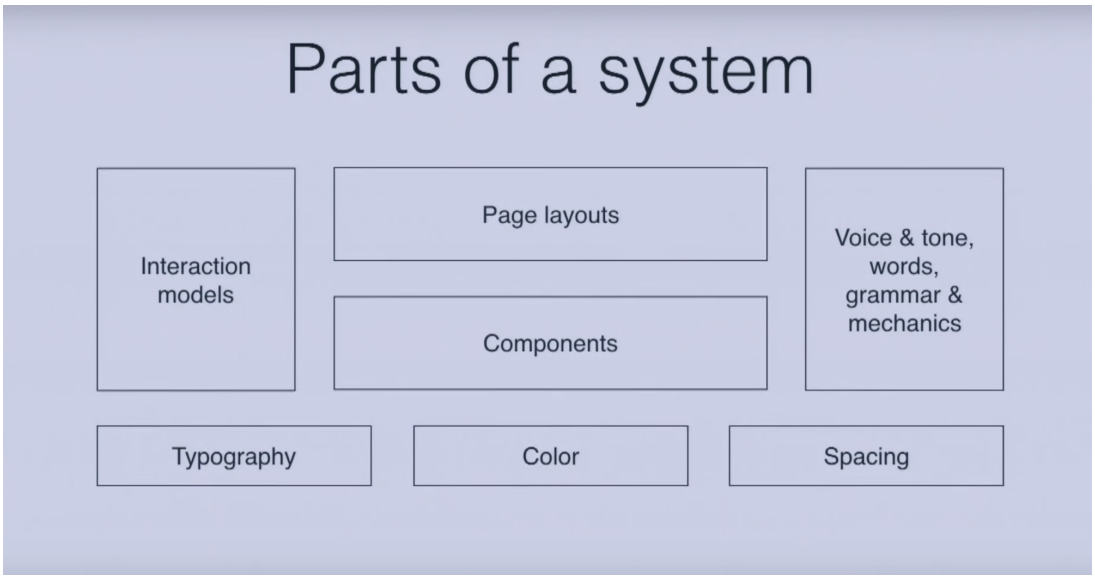


Figure 3. Diana Mounter's view on the structuring of a design system in her presentation at Jamstack 2017

(such as Bootstrap) or style guides on typography, color and spacing. A design system works as an umbrella term for these, since it encompasses also the values, seman-

tics, syntax and context which form the whole (Grainer, 2017). Diana Mounter, Design Systems Manager at *Github* (2017) illustrates the structure of the contents of a design system in the following manner in figure 3.

These can then be divided into the industry-wide terminology that is often confusing. The base, typography, color and spacing, are typical definitions of a style guide. Center, layouts and components, refer more to the pattern library whereas interaction models, voice and tone refer more to design language. These concepts are further defined in the following sub-chapters.

Brand and Front-End Style Guide

Style guides are static elements that are not updated on a regular basis, hence referred to as an "artifact". In comparison, "[a]n artifact is something you'd find on an archaeological dig or in a museum, whereas a system is a living, breathing entity" (Frost, 2013, p. 140). In a brand setting, a style guide is something that a brand or a client might have in the form of a visual identity and brand style guide. In the user interface context, it might include also samples of best practices, style guide for logos, client names and colors that should be displayed, iconography, imagery and templates (Grainer, 2017). A brand style guide is often made public for other stakeholders, such as *Spotify's* branding guidelines (figure 4) for other developers who are integrating *Spotify's* streaming services into their own platform. They list numerous use cases for how to use and not to use their logo and other branding materials, such as coloring, album art and widgets ("Branding Guidelines", n.d.).

Brad Frost (2014), suggests that style guides are thought to encompass brand identity, design language, voice and tone, writing, patterns and code style guides. Moreover,

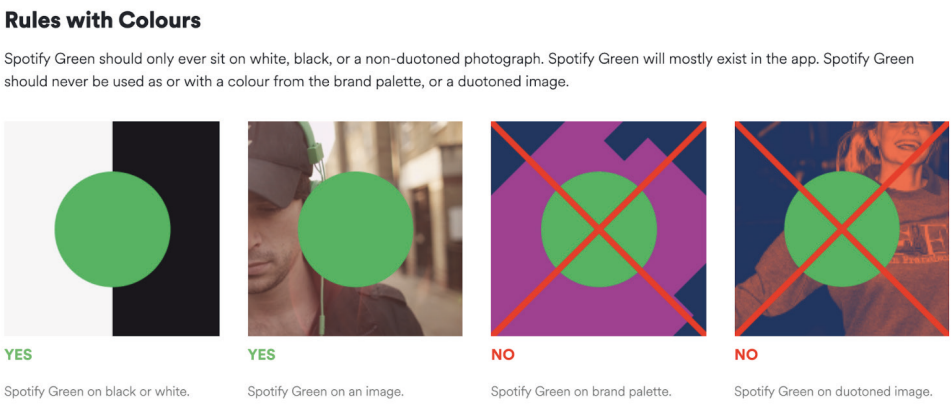


Figure 4. An example of a brand guideline practice on how to use colors by Spotify.

style guides were not considered to have a clear definition in 2014 and they are still mixed with other terms. Therefore, it is important to make a clear distinction to style guides and front-end style guides.

Separating a front-end oriented style guide from the traditional static brand-related use is largely the nature of the guide. In user interface context, it can be seen as a deliverable as well as a tool. Front-end style guides are living organisms of elements and their information, such as code snippets, guides and vector graphics (Laubheimer, 2016). In broader terms, a definition might say that style guides in the user interface context are a modular collection of all elements in a single user interface (Gothelf and Seiden, 2013). Therefore, one might argue that a style guide creates a shared vocabulary for all stakeholders - a button in a style guide is a button to the end user as well as a designer, developer or project manager.

Getting lost with all the terminology and similarity of terms is evident. Frost (2013) suggests calling brand identity, design language, voice and tone, writing, patterns and code style guides as separate entity called style guide and

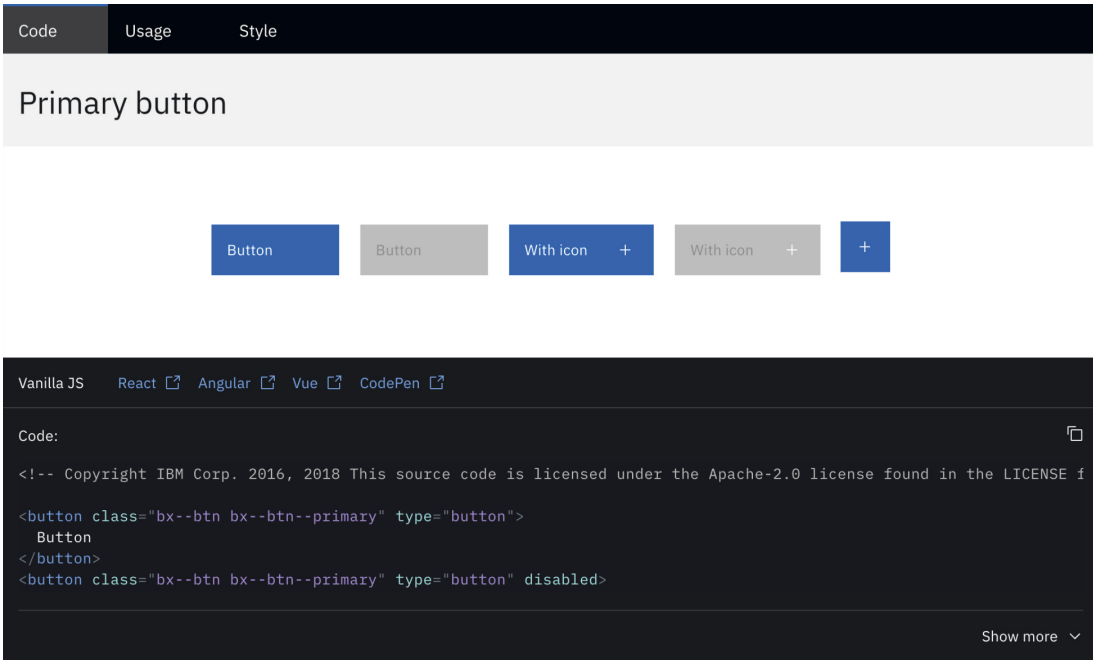


Figure 5. Pattern structuring in Carbon. An element is accompanied with tabs for different instructions such as use cases, stylings and ready-made code.

front-end style guide a pattern library.

Pattern Library

Pattern libraries have likewise many names: UI, component or asset libraries. Where style guides offer guidance on how to create, a pattern library is “a centralized hub of all the UI components that comprise [a] user interface” (Frost 2013, p. 65). However, this library consists of components, which are a list of reusable elements. It weaves together style guides and patterns (instructions on how to and how not to create in a particular setting) with a variety of other guidelines for design, such as behavior, flow and actions (Curtis, 2017).

To help understand pattern libraries better in the midst of constructing user interfaces, Frost (2013) created atomic design methodology, which categorizes and defines the

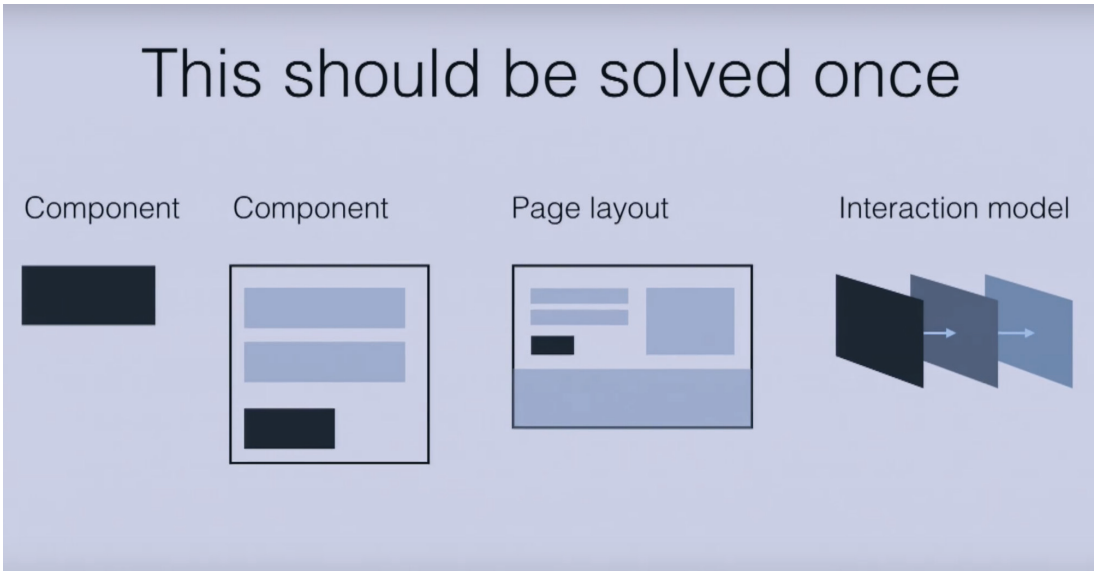


Figure 6. GitHub's Diana Mounter's explanation of atomic approach to design systems.

elements of a design into a systematic hierarchy. The hierarchical levels are interlinked and therefore create the interface design system as a whole: atoms, molecules, organisms, templates, pages. On the simpler end, atoms are foundational building blocks like basic HTML elements such as a typing field. Molecules tie atoms together into simple groups of elements functioning as a unit. Organisms are already more complex components, comprised of molecules and/or atoms. For example, a menu bar on a website would be an organism. Templates are the basic skeleton of a layout and content structure with little or no ready content from previous stages. Pages are near-ready prototypes which take templates and fill them with representative content (Frost, 2013).

Patterns consist of all of these aforementioned building blocks, as seen in figure 6. However, a most typical element in a pattern would be a molecule or an organism - a simple element of a few objects or a small entity of

Figure 7. Location card in Rizzo and accompanying code.

smaller objects. A pattern from a *Lonely Planet's Rizzo* design system, for example, could be a location card, which encompasses a header style, body style, an icon and box background as well as formatting, as seen in figure 7. This is presented with a code snippet, appropriate tagging as well as the use case ("Lonely Planet Style Guide", 2019).

Overall, these terms used for style guides and pattern libraries and the contents they withhold are often misused, causing confusion and difficulties understanding the difference (Toman, 2017). Therefore for this thesis, a style guide will refer to the underlying elements, foundation of a visual designer: colors, typography, iconography, and proper use cases and rules. In other words, all the static content of a digital product. A pattern library, in this thesis, refers to the content building over this base (templates, grids, outlines, sketches, modules and components).

Design and Visual Language

A fundamental part of a design system is the design language (Fanguy, 2019). Often, this item is left out of early illustrations or explanations of design systems, since design language has (similar to previous concepts) multiple names, most notably visual language, design principles or design rules (Mandelbaum, 2014; The Interaction Design Foundation, n.d.).

One definition by *InVision's* Fanguy (2019) describes a design language as having four entities; color, typography, sizing, spacing, and imagery. *Airbnb's* Karri Saarinen (2018), on the other hand, describes a design language to be like any other language: "[m]isunderstandings arise if the language is not shared and understood by everyone using it." Design language for a client, project, designer or any other stakeholder, is in that sense an essential foundation for communication. An example of one of the first design systems that outlines the use of visual language is *Material Design*, Google's design system used for *Android* devices, launched June 25th 2014 (The Interaction Design Foundation, 2019). Android user interface look and feel is recognizable and has a clear mission statement behind it; "Material Design is inspired by the physical world and its textures, including how they reflect light and cast shadows. Material surfaces reimagine the mediums of paper and ink." This mission works as the foundation for Google's *Android* user interface (Google, n.d.).

An example of an underlying design language behind a design system is IBM's design language which works as the foundation to the open-source *Carbon* design system (IBM, n.d.). IBM's design language is guided by four principles: 1. Design is an exercise of decision-making: experience, judgement, responsibility and timing. 2. In order to guide, continuity and creativity must co-exist in design. 3.

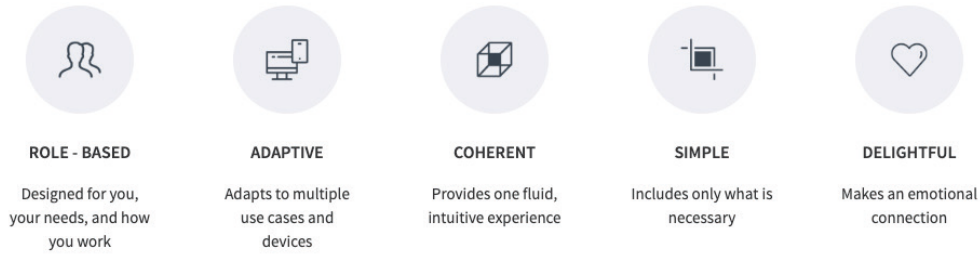


Figure 8. A sample of a design language in SAP Fiori design system. More than elements, it shows also the tone of voice and what emotions are aimed at from users.

Everything communicates, both the things we do and the things we do not do. 4. To guide is to lead (IBM, n.d.).

The IBM design language as the foundation for a design system underlines the difference between a design system and a design language in the following way: Design language includes the underlying philosophy (e.g. brand guidelines), gallery of use cases, elements (style guide) and disciplines. *Carbon* has guidelines (such as accessibility, spacing and motion), components, patterns (use cases), and resources.

A design language is, to some extent, the way a design system is built to be its own unique entity - the recognizability and underlying philosophy behind it. A design language is more than a style guide - it offers reasoning to the decisions made and guides a stakeholder to understand why an element looks, feels and works the way it does. Elements of user interface design, such as transition animations and paddings, might not be presented in pattern libraries to style guides, which is why these assets might find place in design language.

System Categories

As discussed, the underlying design language defines how a design system is recognized. However, many of the current listings of different design systems lack categorization based on the language recognizability and tagging (Limcaco, 2019; “Design Systems - Evernote.Design”, 2019). Since documentation of such systems is vital (Frost, 2013), in the spirit of *A Pattern Language*, some distinctions can be made to approach these systems. These distinctions are discussed below.

Public Sector

An obvious distinction between systems is the public and private sector. The distinction stems from not only a difference in the type of business they are conducting, but also the design language behind it. The public sector might not have a concrete story and philosophy such as a company like Google with *Material Design*, but rather, a diverse target group, for which the system is aimed at. Some distinctions can be made within the public sector, too.

Governmental Services

A popular example of a public sector design system is the *United States Web Design System*, that was introduced to build trust among the citizens using their governmental services. Consistency across governmental services and platforms was a key factor in building trust, therefore the United States governmental websites were transformed to a design system by United States Digital Services and 18F, part of the Technology Transformation Services (“A Design System for the Federal Government”, 2019). The guiding principles of the design system follow usability and accessibility closely: putting users first, making it easy to create,

easy to access, consistency and sharing of the solutions as well as starting from existing solutions (“Design Principles | United States Web Design System”, n.d.). There is an increasing amount of governmental design systems such as the Italian public administration design system (https://designers.italia.it/), British gov.uk (https://design-system.service.gov.uk/) or the Australian government (https://designsystem.gov.au/), which shows the interest in design systems also in the public sector.

Country Brand

A brand or visual tool for public sector can easily be seen as a brand guideline just like any other company brand guidelines. Many country brands, such as the Swedish visual identity (“Identitytool for Sweden”, n.d.) or the Estonian brand guidelines (“Brand Estonia”, n.d.) are referred to as design systems. As closely as these are related to current design systems thinking, these lack the idea of shared content and vast pattern libraries. The design principles and design language is present in both of these examples as well as the style guide. However, both of these examples are used in multiple governmental websites such as the Estonian e-society website (https://e-estonia.com/) or the Swedish education website (https://studyinsweden.se/) which underlines the scaling of design systems to multiple devices, use cases and contexts, just as much as the previously discussed U.S. governmental design system.

Private Sector

The private sector could be divided into following categories: software, human interface guidelines, brand, and independent. Software refers to the design system enabling developers to scale their designs to solutions to match the software in question on other platforms.

User interface is a more theoretical category merely by device manufacturers and it provides guidelines for how to design for specific types of devices. Brand is a category that attempts to scale a common visual look of a private entity on multiple services, offered by themselves or others. Lastly, independent is a system developed by individuals, not for a specific stakeholder but holding more value in other principles.

Software

Software design systems are developed by software developers and designers with existing materials to unify the output of their designs with other stakeholders. Examples, which can be found online, include *Workday* (https://design.workday.com/), which provides a platform for cloud-based financial management and human resources management ("About Workday", n.d.). Their designs are used on multiple devices, services and companies, and by using an aligned design system, they are able to unify and scale the design, hence creating a visually strong brand. Likewise, *Shopify* offers a platform for online shopping, or e-commerce, for which they offer a design system. Using *Shopify's Polaris* design system, clients of the platform can create a recognizable and scalable shop experience ("Shopify Polaris", n.d.).

Human Interface Guidelines

Device manufacturers have an incentive to create design systems to create unified experiences across their devices. Without guidelines on margins, bezels, device outlook and unified style guiding, device experiences could be confusing. Moreover, designers and developers would have to solve all device-specific technical and design aspects by themselves. *Apple Human Interface Guidelines* (HIG) was the first introduced guideline in 1987 alongside

the *Apple 2* computer. Shortly after that, Windows 2 was released also with an accompanying HIG (Fogel, 2016). Since HIGs often provide design materials, patterns and libraries, these are often considered design systems as well. The *Apple Human Interface Guideline* design assets are also built-in into some user interface designer software.

User Interfaces

Opposing the device-specific approach, Google's *Material Design* focuses more on the user interface side. *Material Design* is a language for Andorid-mobile user interfaces (Fogel, 2016; The Interaction Design Foundation, 2019). Since Google does not manufacture all Android phones, nor does it limit the amount of custom operating systems based on *Android*, their design system is essential for other software developers, device manufacturers as well as operating system developers ("Android Open Source Project", n.d.). A unified system gives tools and guidance to developers regardless of device - a style to follow to avoid user confusion.

Brand

A brand-based design system, while a fully-functioning design system, is usually for one service provider rather than many. As an example, *Airbnb's* design system is built only for their own use. Brand design systems are rarely public since it might include intellectual property of the company in question ("5 Tips from an Airbnb Designer on Maintaining a Design System", n.d.).

Independent

An independent design system is not built by large teams but rather individuals working in teams or solo, publishing

it to the public for anyone to use. It might have an artistic or professional motive behind it. The design system *React95* replicates *Windows 95* operating system's user interface into a coherent design system with elements of the operating system that can be used by a designer or developer to design a *Windows 95* inspired user interface products ("Storybook", n.d.).

System Management Roles

As with any type of systems, such as air conditioning, electrical grid or an operating system, there are different management roles for how that system is being kept operational and maintained. The following sub-chapters of design system maintenance types are defined by Nathan Curtis (2015) from *EightShapes*.

Solitary

This model is seen as the overlord - one person, or a small team, operates and keeps all the rights to themselves. This system only serves the needs of a specific team, structuring it accordingly, not taking into account other views. It is seen as an inclusive model, that lacks diversity. The team builds the design from the outside of the use environment and it is then shared with the team.

Centralized

A centralized system creates a system from the inside to outside, rather than the aforementioned overlord-view of management. The central team takes into account requests but does not take any contributions from the outside world. A centralized design system is easily spread among the base.

Federated

This model takes into account several participants who contribute to the system in question. Using Google's *Committee by Design* -approach, committees federate a system which is then directed at the representative (Bohn, 2013). The representatives are empowered individuals working together to form the basis of a system. An elected team then makes the decisions based on collective say. In a federated team, expertise is used as guidance; the representatives contribute based on their role such as visual designer, user experience designer, developer or project manager, for example.

Causes for Design Systems: Common Arguments in Favor and Against

There are multiple reasons why a design system would or would not exist in a company. As there is still little research and evidence on how design systems benefit or limit work, resources from publications such as *Medium* and public sources from a company's own websites have been used to gather a background into design systems.

In Favor

One of the most repeated arguments for the need of a design system is speed and efficiency. By solving smaller problems up front and giving ready made tools, a design takes less time and therefore costs less, notwithstanding the costs of maintaining and creating a design system in the first place (Fanguy, 2017; Araújo, 2018).

Another argument is the ability to scale and to be consistent. A design system enables the designs to be applied on multiple platforms more quickly and to maintain consistency between them (Araújo, 2018; Serrault, 2019).

Lastly, design systems are seen to improve communication and team work. With a unified library staying consistent, all stakeholders have a common language to talk about. Designers show a visual asset to which a developer has a code ready for example (Linders, 2019).

Against

Much of the debate surrounding design systems is about the design work and about how creating a common system is limiting creativity. This creativity limitation might also be an asset, as some might argue, that then designers can get more creative with other issues that design systems do not solve (Skjoldbroder, 2018; Hacq, 2017).

Design systems are considered to limit the ability to explore solutions in the building phase. In other words: while using a design system, one might become biased to their own decisions as those decisions are repetitive (Skjoldbroder, 2018).

Lastly, systems are seen as inflexible and complex and tend to lose value over time if not maintained. The rules set in a design system limit the solutions one is able to make in a custom setting, where design systems do not necessarily meet everyone's needs (Skjoldbroder, 2018; Vendrik, 2019).

In addition to the arguments discussed above, design system pattern libraries are still focusing on the static elements of a design. Design languages and style guides might give tools on how animations or gestures are designed (e.g. *Material Design* where sets of animation guidelines are given as well). However, design systems do not yet provide patterns for animation design nor do the most used applications have capabilities for them

(Keshtcher, 2017). This lack of animation, in turn, might lead to inconsistency, since all animations have to be custom-made by the designer and developer separately.

This chapter has discussed the context of design systems. In the next chapter, research scheme for this thesis will be discussed in detail.

Research Scheme: Methods and Approach

As systematic design has been around for hundreds of years and yet its terminology has remained unclear, this thesis tries to answer how a design system is different from other similar concepts (such as pattern libraries, style guides, and design language) and, furthermore, what constitutes a design system (Grainer, 2017). Therefore, the first research question is:

To what extent does a design system differ from other industry-used framework standards?

After having understood the historical background to the industry-wide term and its boundaries, we can dive deeper into the building blocks of design systems. Reflecting against the current literature on the topic, a definition of the pros and cons of a design system should be visible in design work; moreover in work which has to scale for multiple different clients with different visual identities. Therefore:

What are the benefits and limitations of using a single design

system on several projects with different clients?

Lastly, as one of the arguing points for a systematic way of working has been efficiency, we need to question how design systems change the way we work. (Frost, 2016)

To what extent does a design system help designers work more fluently with and between customers, front-end developers and co-workers?

These methods outlined below were carried out within a unit in a large multinational information technology company operating in Helsinki, Finland, which offers vast amount of digital solutions. The focus of the unit in which these methods are conducted in focuses on enterprise customers. The methods in this thesis are twofold: Empirical research is carried out to gain insight into systematic thinking, design systems and design work in the user interface and software development field. In turn, the insight gathered from empirical data leads to own creative output.

The author of this thesis works at the company in question and is part of product development teams in a UX and UI designer role. Prior to this research, the author had limited information about design systems and did not utilize a design system in his daily routine as a designer. For the purpose of this thesis (more specifically empirical data gathered with colleagues) concepts of design systems were also studied and introduced to colleagues

before any research activities.

Methods

Since the nature of design systems for a designer is highly visual, the natural approach to research was using qualitative research methods. A popular research method for designers is applied research, which defined the boundaries surrounding this thesis. This paper follows a looping process of research, in which the outcome of the study will improve the next iteration through monitoring, evaluating and further redeveloping the creative output as seen in figure 9 (Muratovski, 2015).

Practice-led Research

As design systems have little to none theoretical background and research, this thesis aimed at generating new insights via practice-led research methods. Practice-led research aims at increasing knowledge about the research field and not the artifact in question. Using design systems while examining and researching the current state of the field surrounding them is essential to the research topic since no coherent theoretical structure exists (Muratovski, 2015). Additionally, for theory regarding design systems and patterns in design work, secondary research is used through industry-wide literature. Furthermore, as little to no academic study is available and terms are found to bounce back and forth - a term might mean different things to different professionals in the industry. For clarifying this, this thesis tries to underline some of the concept definitions using notable industry literature in this subject to emphasize the meaning behind a term.

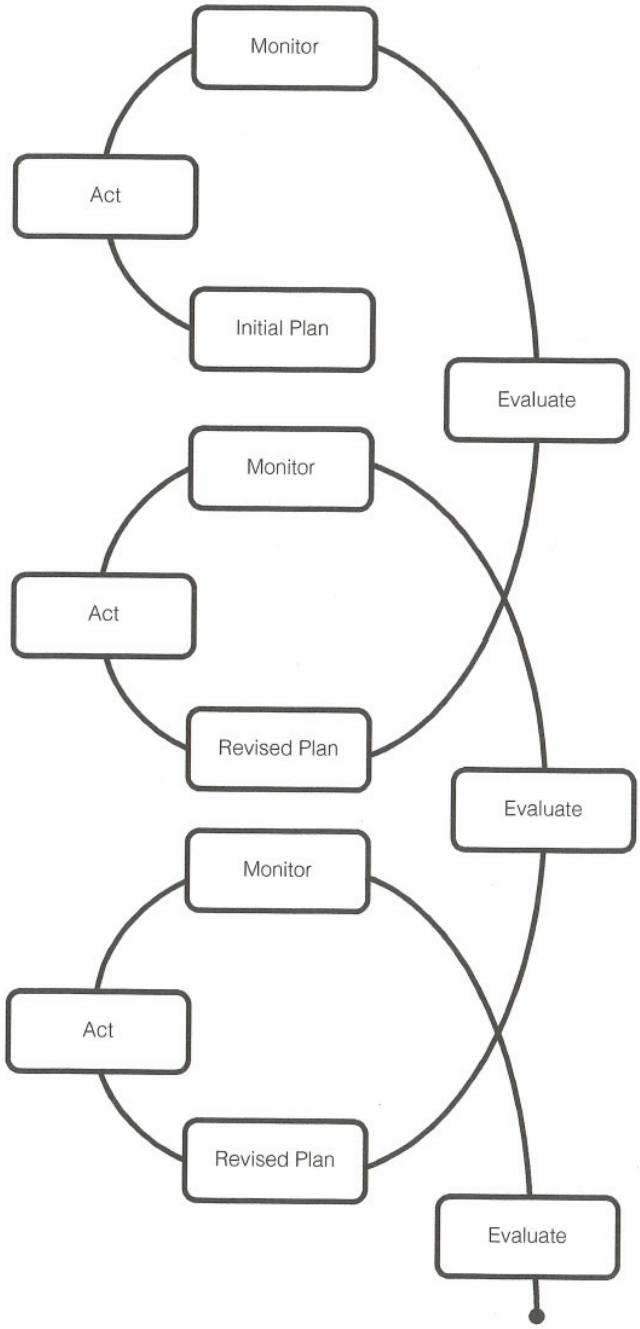


Figure 9. Gjoko Muratovski's applied research method is a method in which reiteration happens in an inductive manner.

Decontextual Testing

Initially, a workshop was hosted for designers with a design thinking exercise as a theoretical framework method, to gain deeper understanding and context about the end users (Esposito, n.d.; “IBM Enterprise Design Thinking”, n.d.).

Tests followed a decontextualized structure, meaning user tests where end product or related objects are not involved in the tests. This was aimed at finding out users’ initial attitudes, expectations and needs, since a design system was not in use, nor was the initial product of the study still in production. Decontextual tests can be conducted in various ways, for which a hybrid test method was created. The users were presented with tasks involving hypothetical scenarios. After scenario tasks, users were interviewed in a focus group with an unstructured interview method (Bank & Cao, 2015; Muratovski, 2015).

Scripted Testing On-site

Two focus groups, developers and designers, were subject to scripted testing after initial concept design of a solution. Scripted testing can be moderated or unmoderated. Moderated testing, in which an observer or a moderator is observing the test subject in the same space, is supported in early stages of development phase. As solutions for this thesis were still in development, moderated tests were suitable for the occasion. The tasks that were given to users were direct tasks, which are instructional in a statement manner, such as “Design a home page of a news website”. Since these tasks are technical in nature, the purpose was to test if a solution works in a technical and systematic manner as intended. The participants were given identical tools to work with and as a task, a

wireframe of the outcome that was hoped to be achieved. The tests were conducted in a hallway usability testing manner, in which random users from target audience are asked to use a product (Bank & Cao, 2015).

After the use of the product, the users were interviewed. These interviews were conducted as semi-structured interviews, which allow for more room for discussions, improvements and ideas from the participants. This interview data was recorded on tape and transcribed later for data collection (Muratovski, 2015).

Guiding the preparation of the scripted testing, Jeff Rubin and Dana Chisnell's work *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests* (2011) was used. Particularly, chapter 9 "The Process for Conducting a Test: Conduct the Test" (p. 201-228) gives a code of conduct for a usability test, in order to have as little external effect as possible to the outcomes of the test. As a moderator for the tests, this chapter caused the following decisions: participants would have to see the moderator but the moderator shouldn't be on the sightline of the participant, nor too close in order to avoid intimidation. Hence, the moderator would be sitting behind, slightly on the right from the participant within a reasonable distance. (p. 221-222)

The moderator would not comment on the outcomes or decisions made during the test. If necessary to intervene, it would happen as a last resort for reasons such as time or participant being lost. The moderator would pay extra attention to body language, tone of voice and other gestures (p. 211-212)

The 'Thinking Aloud' method would be used to collect data from the participants. The method helps participants focus, helps data collection and shows the pain points of

the participants just as they happen. (p. 204-205)

Debriefing would include a post-test questionnaire and a semi-structured interview based on the reflection in the questionnaire. (p. 231-235)

Interviews would be audio recorded with the consent of the test subjects. Tests would be observed by the moderator taking notes at the same time.

As a modification to a typical usability test setup, the test assignments were different for the two focus groups. Designers were given a more conceptual task to design an interface to include certain aspects in the list with given tools.[1] Developers, on the other hand, were given a ready-made mockup of an interface screen that they had to code.[2]

Approach

The design system in place at the company, *Carbon*, was used as a personal starting point for design work. Without creating a whole new system, *Carbon* gave a platform to look for the advantages and disadvantages of an ever growing system. Moreover, this system was not widely used in the unit study was conducted in and therefore provided a starting point for assessing the reasons for using a design system.

From a designer point of view, customization of a design system might mean a *Sketch*-based library, which is used in the design of an interface. For example, a designer might add a button to a canvas, try to change the coloring of an element and be faced with a warning: this element belongs to a library; it needs to be detached in order to change coloring. What might arise from this is that a

confusing amount of linked and unlinked assets are found in a design and complicates developer work - rather than using the assets from a design system, the developer is back at designing everything from scratch, based on a drawing from designer. Solutions for this, however, such as building your own theme, have been provided for systems such as *Shopify's Polaris. Polaris* attempts to solve customization with theming and guidance on how to build one for their design system (Vendrik, 2019).

Upon coming into terms with the fact that no new system was needed, but a custom pattern library that could be used more rapidly without scaling, a mobile UI kit was determined to be the outcome of this creative work. Most design systems are built web-first, whereas the custom pattern library was built mobile-first for two *iOS* devices, based on results from generative and secondary research done prior to creative work.

Moreover, scaling a design system to mobile takes time. Sometimes prototyping has to happen already inside a design workshop, meaning, that a designer would have to design a sketch of a potential UI while meeting a client to decide on next steps. For these, design systems can begin to be a complicating asset, not an enabler.

Enter, a custom UI kit.

Mobile UI Kit for Custom Design Work

This chapter covers both the design work conducted as well as the empirical data gathered. These two entities provide insight about the nature of design systems as a part of a designer's work. Empirical data was gathered prior to setting up a framework for creating the UI kit and after the first iteration. This chapter is a demonstration of the applied research method in practice.

Workshop

The design process started with a workshop organized for the team members in the workplace on the 14th of May 2019. The purpose of this workshop was to gather information on the mindset of the team members as well as to gain further understanding of how design systems were understood and approached by the participants.

The workshop started with a background on the thesis, introduction to design systems and to *Carbon*. After the introduction, the rest of the time was dedicated to tasks.

Tasks were introduced with a hypothetical scene, so that the participant could relate more to a setting in which they would be completing these tasks.

Task 1: Home Screen Layout

First, participants (3 designers) were given a task to draw a home screen of their mobile devices in 1 minute without looking at their phones as a warm-up activity. From the 3 participants, 1 had an *Android* and 2 an *iPhone*. After the exercise, participants were asked to evaluate their home screens based upon the real user interface. All of them noted that their most used apps were the ones they remembered where they were located in and how they looked like. Moreover, on the general operating system level, all of them noted the existence of the top bar; relevant information on system operations such as battery level or connectivity.

The reason for this exercise was to find out what elements are seen from a designer point of view as essential in a user interface and having most impact in their daily lives. These would then be most likely the objects that the designer would start plotting out in the beginning of a design project; outlines of a system, notifications, screen layout and margins. An example of this is found in figure 10.

Task 2: Application Elements

The second task was to create a mobile application with one functionality for a specific industry and a specific user group. These user groups as well as the applications were invented to give participants more space for imagination. Participants were asked to map out all the elements they would draw for the application in question, sketching on a board first and then mapping the elements one by one. Elements would be mapped out on post-its, one post-it responding to one element drawn on a display.

After completing the post-its as seen in figure 11, partic-

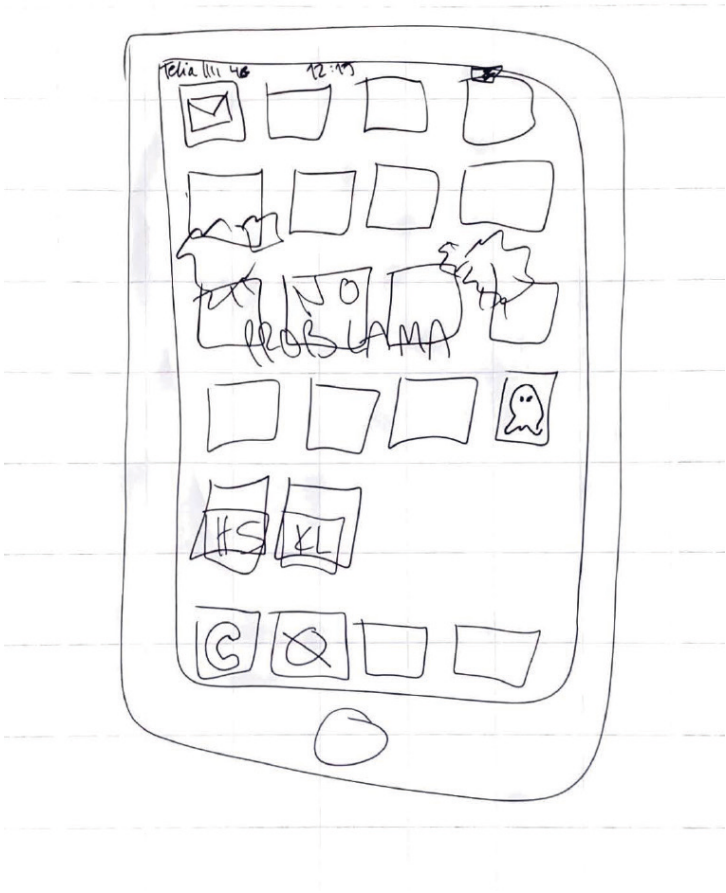


Figure 10. In task 1, one of the designers drew the following sketch. She noted that the most used applications were easiest to remember visually as well as her background image, notification bar and basic grid.

ipants were asked to think of these post-its as steps in a linear manner; each step leads closer to a prototype of an interface. At this stage, they were asked to mark an approximate time needed for completion of each step. One of the three participants failed to estimate a time for completion. After time estimate, participants were asked to place these steps into the linear map starting from left to right, left being first step, right being last.

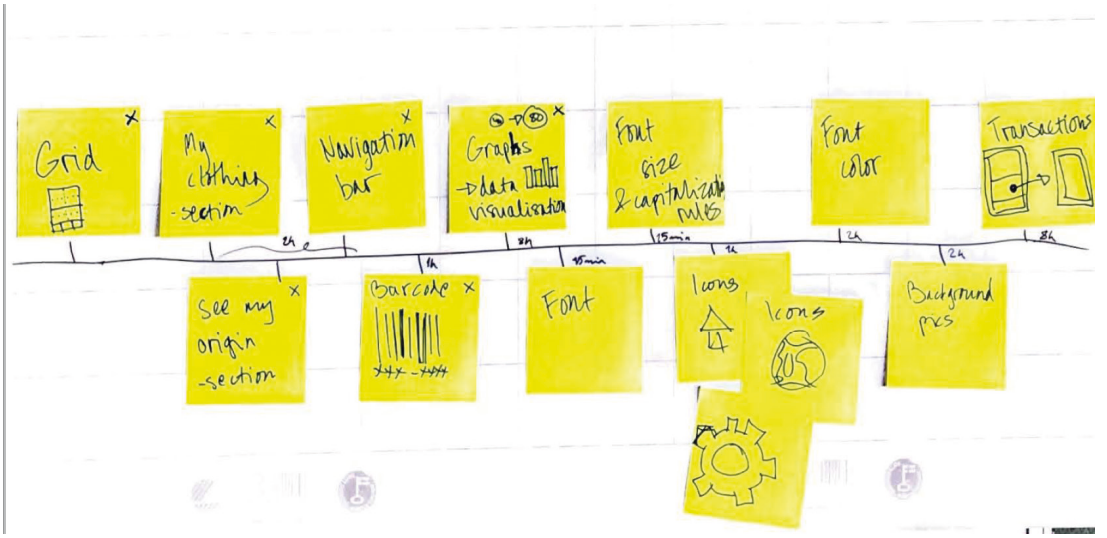


Figure 11. Task 2 outcome from one of the participants showing the timeline starting from grid and ending in animating transitions.

As the last task, participants were asked to present the outcomes to others, clustering the elements at the same time.

Outcomes from the exercise varied. Repetitive aspects that all the participants pointed out were typical aspects of a style guide: icons, texts and other objects. Color, however, was mentioned only by one of the participants. Patterns such as grids and layouting of the device were also repeated as well as menu structuring. The rest was largely industry-specific such as barcodes, ratings or communications, which were not universal in all the tasks given to participants.

One designer said the time needed for their single application to be produced into a clickable prototype would take roughly one working day, whereas the other one said from 30 minutes to 2 hours. One of the participants was unable to complete a time estimate. Given that these time estimates were based on a designer doing all the work themselves, rather than using a design system, it was an

introduction to showing how a design system would help them with aspects of their design work; giving ready made patterns, guides, icons and more for the designer as a helping hand. After this workshop, creative work for implementing a design system started.

Custom UI Kit for Mobile Devices

Given the outcomes from the initial workshop, the design work focused on giving tools to support a design system and the designer in a quick iteration phase. These tools would need to be quickly modifiable and easily addable to a canvas and not to be parts of the design system itself, but a developing iteration to support it. As tools such as the *Apple Human Interface Guidelines* provide all elements as separate with hard customization possibilities, the mobile UI kit was considered to be a solution to reduce the amount needed for getting pieces from several places. Simple but necessary things that were seen important in the workshop, such as the status bar, grids or icons, are time consuming to find in a quick iteration phase, whereas the UI kit would provide these for the designer in the beginning.

Why: Design Systems Are Boring - So That Designers Are Not

The process of taking a design system into use in a team requires a great mindset change. Initial thoughts can be that ready-made elements limit their creativity or that the system does not serve their purposes (Skjoldbroder, 2018). To some extent, both of these statements are found to be true as well; in that case, a creative designer has free hands to work as they see most fit and develop their own solution. A specific type of a button might just not work.

Figure 12. A hierarchy of a Carbon design system for a dropdown menu. To place it on canvas, user would have to go through 5 sub-menus.

Or using the same elements over and over can become boring from a designer's point of view. These problems sometimes create innovative and better solutions for a client, however; "[t]he design system carries the burden of the boring, so that designers and developers don't have to" (Clark, 2017). The reasoning behind the kit which supports a design system, can be underlined from Chapter 2: to increase speed and efficiency of quick mobile prototyping, improve communication with clients and other stake-holders. As an addition to tackle the creativity aspect of feeling that design system limits designers' work, the kit aimed to increase creativity with larger creativity options.

This led to the conclusion of design systems being enablers rather than disablers; they enable a designer to work faster and leave more space for ideation, discussion and research. However, as mentioned before, the nature of design systems being a new concept and lacking hierarchy structuring and common language, the naming of objects can get confusing. Therefore, for each design system, a learning curve exists and sub-menu after sub-menu, the right element is not found as easily as one would hope (McKenna, 2018). This marked the beginning for the design work.

Version 0.1 - Structure and Contents

As many repeatable elements - icons, buttons, menu bars, navigation - are something that in a quick prototyping phase a designer does over and over, the kit would be done in *Sketch*, a UI prototyping and design platform. Opposing an asset hierarchy as seen in figure 12 of a design system based in *Sketch*, this kit would emphasize visual aspects first. Hence, the assets would follow a 'drag & drop' logic: a template of a mobile device would be provided alongside a list of UI elements that could be added and modified if needed (see figure 14).

Navigation within the file would also tackle an issue faced by a designer when beginning from an empty canvas (see figure 13). It would provide entities for a sandbox for iteration, references for website and app layouts and collage of safe areas and margins defined by the device manufacturer. Also, lastly a page with possible ready-made links for clicks would exist for export to InVision or other prototyping platform.

Sandbox would be the actual building space, from which a user could jump back and forth only if necessary to references or layouts. In the sandbox, the designer would find most of the tools necessary, cutting back on time consuming browsing of menus or going to different files to search for the right asset. All the pages indexing with a 1 would be starting points from which a user could ideate and copy the necessary starting elements onto the sandbox and go forward with the toolkit in the sandbox.

In the sandbox, the user would find two artboards with an outlined screen for an *iPhone 8* and *X*. Next to this, the user has another artboard with a listing of the 21 most needed UI elements.[3] As a guide for planning what elements to include, the initial workshop alongside with

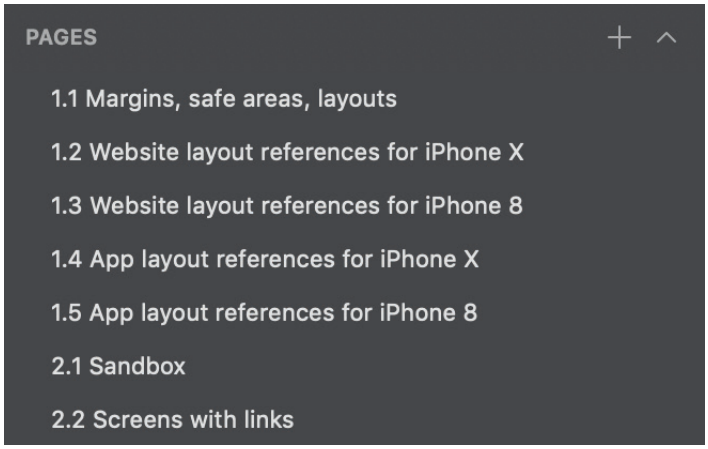


Figure 13. Page structuring for the mobile UI kit.

industry literature was used to decide upon included elements in the first iteration. (De La Riva, 2019; Assistant Secretary for Public Affairs, 2013) After testing and using, the iteration would then continue on adding elements that would be missing.

The initial idea for this structuring of the kit was to remove anxiety of having one large file with large amounts of data to be copy-pasted into different places. A sandbox would increase playfulness into design, within the designer could play around with the elements at hand and go to other parts of the kit if necessary. Since the hierarchy in *Sketch* works in pages, the decision was made to separate the starting point, references, the sandbox and links to further decrease anxiety with large amounts of data.

The sandbox asset artboard would include the UI elements which are multipliable by the designer or modifiable in the same place if needed. Moreover, this decision to have all elements in alphabetical order as a visual list rather than a traditional menu structure was based on the initial observation that browsing the menus takes a lot of time

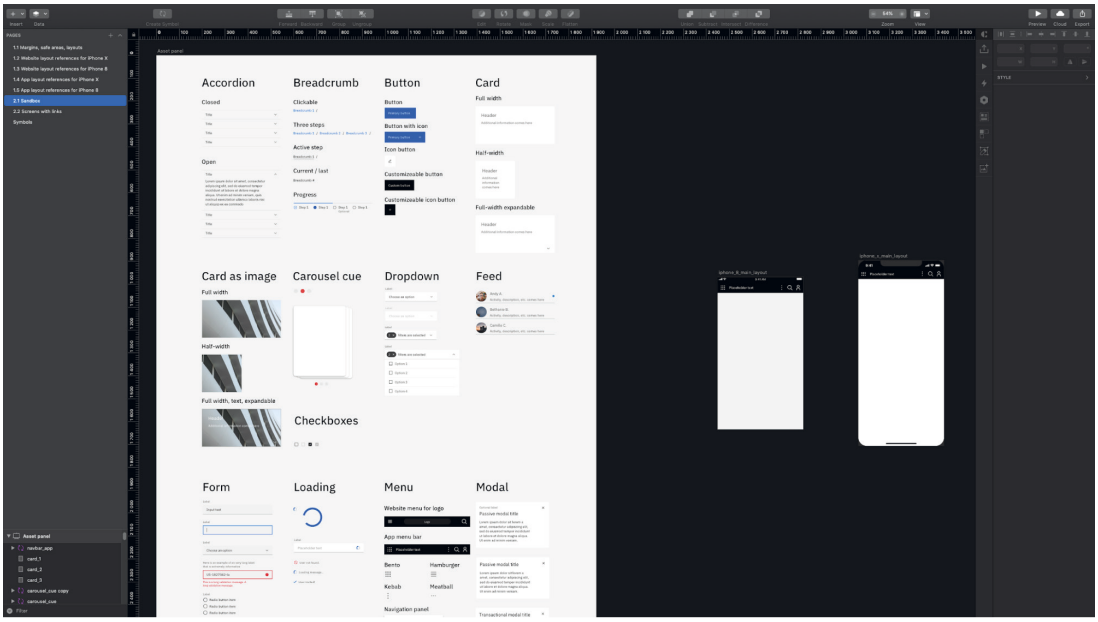


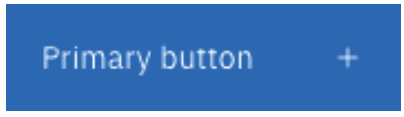
Figure 14. A view of the sandbox with two iPhone screens on the right and asset panel on the left.

out of the hands of the designer.

On the right-hand side of the asset artboard, the two iPhone screens are presented for selection with a ready-made status bar, an app bar and a visual home button for the *iPhone X* along with margins and outlines for design. By applying layout settings, official *iPhone* grids are also present on these displays.

Naming of the elements in the system's element panel would follow *Carbon's* initial naming system, such as the following for a clickable button with both text and icon.

button / 01 primary / default / 02 text and icon / 01 enabled



In this naming method, first the element is described, after which the conditions are explained. In the afore-mentioned example from *Carbon*, the button is primary in nature, meaning that the button is helpful to the goals desired by the user and therefore has a higher contrast than a secondary button which would not be helpful towards the task in question (Babich, 2016). Default refers to the sizing of the button after which the type of the button is stated (icon, text and icon or text). Lastly, the condition in which the button exists is stated, such as inactive, when it is clicked or the enabled state. Using this methodology, the elements are then categorized automatically by *Sketch* to submenus, which can get confusing. Reading this item in text, however, seems much simpler. Following *Carbon's* system of naming elements, the naming of elements is summed to be the following:

type of element / condition / size / contents / action

After this naming formula, items were made on canvas and the iPhone frames created. Using subjective testing, the kit was then taken to designers and developers to see how they would use this concept of sandbox and to see how it would help their work.

Usability Tests With Developers

The first evaluation for the needs of a UI kit were done with developers to further understand how they perceive design systems as part of their work and how a custom kit could be used by the team would to solve some customization issues and to further improve collaboration.

Developer tests were on the 16th of July at the premises of

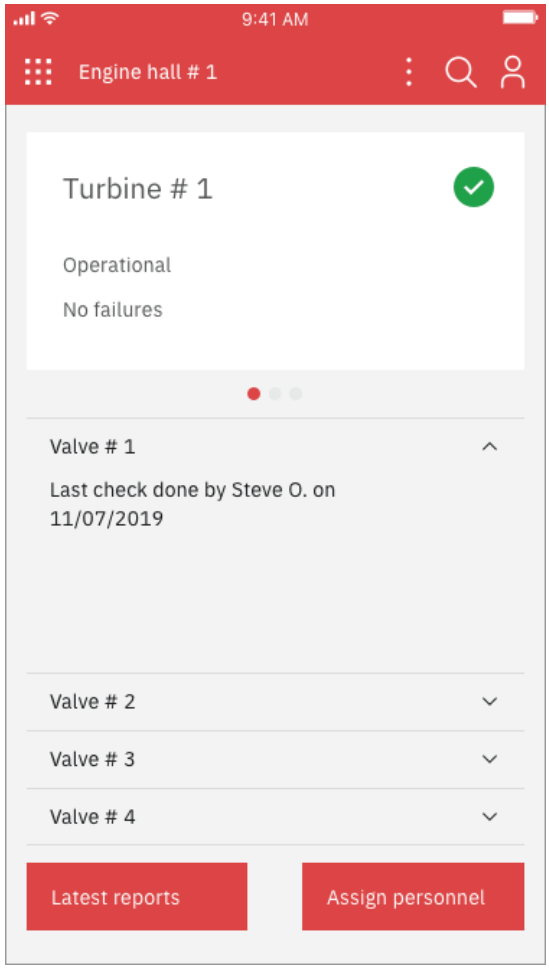


Figure 15. Task given to developers. Also the intended outcome from designer tests.

the company. A setting was created in a team room, where typical projects happen. The developer was given free choice to sit where they wanted and the moderator would follow and sit behind, on the left side of the developer, within a speaking and viewing distance to monitor the activities, according to Rubin and Chisnell's (2011) guidelines for moderating. After the completion of the test, the moderator would sit next to the developer and hand out the questionnaire, exit the space while the test subject

filled it in and, after completion, interview the user in question.

First, the developers were introduced to *Carbon*, after which the first task was given to them: an image of a monitoring web app for a maintenance worker in the industry field on an *iPhone 8* display as seen on figure 15. Their task was to begin replicating this design into code with their own choice of tools. The developers were asked to use their own *MacBook Pros*. Both developers chose *Visual Studio Code* as their code editing platform, *Terminal* for file and system management and a web browser for viewing their work.

Both developers started with outlining the file structure and chose to use *React* as the basis for their project. Both users noticed the use of *Carbon* in the design and started to use *Carbon* as their source code and guide.

> *"When I received the png, first I recognized a bunch of Carbon elements in the design, which were used. I started with setting up the project file structure and finding out the elements used in the design from Carbon website." (Developer 2)*

Before entering the second task, both developers were able to create a basic outline for the design. Some elements that were in the picture were already almost exactly in the same form as the components in the design. However, both developers struggled with errors from the software; namely, the *iOS simulator* not functioning correctly and the *Carbon* project structure not aligning properly with *React's* own project structuring. After these errors, the 30 minutes scheduled for the first assignment were done and the second task given.

In the second task, the *Sketch* file was given to the developers and they were asked to continue on the work based on the design file for another 30 minutes. Both pointed to this being a major help for their work.

> *"If I wouldn't have been stuck on designing the buttons, the Sketch file would have been even more useful. Particularly with the header, which had multiple icons and styles that I could have replicated more easily with the file." (Developer 1)*

> *"If I would have had the Sketch file from the beginning, I would have known a hierarchy of the design; which element goes inside which and which are the main components I would start to work with." (Developer 2)*

One of the developers also noted that even if the hierarchy is there, he does not pay attention the naming of the items, which state the object being a custom element or a linked external element to the designer. That said, both developers spent a major part of the rest of the time with the design file, one with exporting icons into the design and the other with button alignment. These main aspects of the design are the first things that a developer has to focus on upon starting a project.

> *"First the main hierarchy and the components have to be created, after which we can focus on more customization and tweaking that have been thought on the design." (Developer 1)*

Customization is a major setback in developers' experience. This test included 3 custom elements and 3 ready-made *Carbon* elements.[4] Both designers were able

to recreate at least 2 of the 3 design system elements, whereas both noted custom items being harder to start from nothing.

> *"Creating the accordion took me very little time in comparison to what it would have taken if I had to do it myself. Hence, with the animation of an expandable tile, the implementation of it would be very time consuming and I would have to design the animation from beginning." (Developer 2)*

> *"Making custom elements is very time consuming. Sometimes googling ways to make an object takes so much time and after, you might end up with a solution that does not make sense. The final outcome has to be reasonable also, not just any type of solution." (Developer 1)*

The custom elements the designers were provided were not present in code snippets but just as a visual design on the design file provided. Within the timeframe, developers were not able to start to recreate the custom elements, but rather, focused on the ready-made solutions. If these custom elements had been made with a unified design system, one of the developers noted that it could have improved the process.

> *"If there is a need for a web app for example, the actual building of such an app is very quick using a design system if the elements needed are there, plus all the team members know about it and speak the same language." (Developer 2)*

On the questionnaire, both developers felt that they had not completed all the tasks given. However, both of them

were able to replicate 66 % of the elements, 4 out of 6, in 60 minutes. Nevertheless, they felt that the task and tools they were provided were easy to use. Regarding using design systems and custom mobile kits, they felt that they were well organized but were left hoping a bit more for functions in the menus of the design systems as well as the custom mobile kit. The hardest parts in this design were exporting *SVG* code from the design file into code and recreating the header.

A notable outcome of this test from a designer's point of view was realizing how the design elements' naming conventions was not as important as one might think (Palkó, 2019). Both developers recognized the elements from the photo and *Sketch* file and did not need to look for the correct name. For a designer, typically the naming of an element tells a lot about the structuring of where it belongs and for which element it is intended. The developers, on the contrary, went solution-first.

> *"A Navbar_app is just a name for me. I started doing this task reflecting on the previous project I did that had a navigation bar similar to this which is why I decided on my way of developing it." (Developer 2)*

> *"Jumping from one page to another when copying a code snippet for a specific solution takes time and makes it rather inefficient. I wish they were in the same place in pure code without implementation to Storybook[5]." (Developer 1)*

After the developer test, a conclusion can be made that a custom elements with design systems are a major drawback in efficiency. It consumes time from to create an item that does not exist or to customize a ready-made asset into the desired outcome. Hence, both developers

spent almost 15 minutes on aligning ready-made buttons and changing their color. Moreover, without code library for these custom elements in the UI kit, they will not benefit a developer later on.

Usability Tests With Designers

After the developer tests, an evaluation for the needs of a UI kit were done with designers to further understand any differences between these two groups and further, possible strengths or weaknesses in the custom UI kit or the design system.

Designer tests were on the 24th of July and 22nd of August at the premises of the company. A similar setting to the developer workspace was created in the same team room. Each designer was given free choice to sit as in developer tests, following Rubin and Chisnell's (2011) guidelines for usability tests. After the completion of the test, moderator would sit next to the designer and hand out the questionnaire, exit the space while the test subject filled it in and, after completion, interview the user in question.

Before the test, the designers were asked about their background and their familiarity level with *Sketch* and design systems in general. Both had a relatively on-off relationship with *Sketch*; not using it on a daily basis since both of their work based on business design and development. After ensuring functionality of *Sketch* and *Carbon*, the first task (figure 16) was given to them: an outlined list of elements in a UI in a descending order from top to bottom. Their task was to begin replicating this UI in a free visual style in *Sketch* with *Carbon*. The designers were asked to use their own *MacBook Pros*. They were given a tight timeframe of 15 minutes to design as much of the user interface as possible after which the second assessment would be given.

Your task is to design an application for a hypothetical maintenance manager supervising the daily operations at engine halls in a hypothetical factory space.

Create a sketch of a web application, appearing to be open in Safari on an iOS device with the following aspects in an descending order from top to bottom:

- it has a header/navigation bar with:
    - a menu icon
    - a kebab icon
    - a search icon
    - a profile icon
    - a header text "Engine hall # 1"
- a carousel with a card open stating the following
    - a header text: Turbine # 1
    - additional information: Operational / No failures
    - an icon stating the status "OK"
- an accordion with 4 bars, opening icons on the right side;
    - accordion headers stating "Valve # 1" to "Valve # 4" in ascending order.
    - first bar open with the following information below the header:
        - "Last check done by Steve O. on 11/07/2019"
- two buttons below the accordion:
    - left: Latest reports
    - right: Assign personnel

- main accent colour of the hypothetical company: #DC4343
- main base colour of the hypothetical company: #F4F4F4
- font: IBM Plex Sans, Regular

The sketch is meant to be the beginning screen for an interaction; touching a button or icon. You do not need to plan the next steps but only this initial screen outlined above.

Figure 16. Task given to designers based on the sketch from figure 15.

After the task, in the same manner as with the developers, a questionnaire was handed out to the subjects. Both designers felt that *Carbon* was more difficult to use than a visual mobile kit. However, one of the test subjects felt they completed the tasks and the other one felt they did not.

A common problem with the test subjects was understanding the logic of a design system: the name of an element might not mean the same thing universally. Browsing the system visually or in menus takes time. Moreover, without the context of a client, for whom the designer is working, some elements were harder to understand.

> *"Using Carbon I started wandering around looking for basic wireframes just to get the simplest task of setting up my canvas and placing first elements done. I spent a long time looking for icons under the icons-menu which I could not find in the design system. It would be great to seem them visually." (Designer 1)*

> *"It took forever to get started with basic elements on the canvas. Most time consuming was browsing submenu after submenu for an element and trying to decide whether I should design it myself rather than try to find it from Carbon." (Designer 2)*

Getting started with a design was based on two things: starting with prior knowledge on the topic or design and copying some of prior designs done to avoid reapplying the same solutions multiple times. This underlined the design system logic of having to solve each design problem over and over upon a new design. Moreover, as with developers, designers start from the largest and most unique solutions to get a wireframe of the design done first and then they proceed to smaller, UI wide details such as clock or battery status.

> *"Most of my time went into starting the task; setting up canvas and templates with basic icons and elements. The custom elements were even more difficult since they were not found in the design system. Finding bits and pieces from older proj-*

ects could take longer than trying to design something from bottom up. I would leave the smallest details to last, starting with largest objects." (Designer 1)

Main difference to the tasks with developers was the speed in which designers started to work. After 15 minutes of using *Carbon*, the users were introduced to the mobile UI kit. For getting to know the kit, the users were given 5 minutes to browse and to decide if that would prove beneficial for the initial task, on which they were then asked to continue working on.

"I could imagine that with pretty low experience level with Sketch or design systems, anyone could use this kit to drag and drop features or plan a website into a mobile UI. Best is that you do not have to start from nothing but have already some layouts ready or the sandbox-view from which to browse and drag and drop features onto the layout." (Designer 1)

"The drag and drop logic that the kit has made me feel less intimidated about being able to customize items and find them on the right place. However it seems difficult to know which place I should pick an item such as accordion from and how much I can customize them. Instructions for this would be nice." (Designer 2)

As seen, designers feel at ease seeing the items visually but get somewhat confused as to how much can be customized and how. The lack of instructions in the mobile UI kit created more confusion to one of the designers whereas the other did not mention that.

However, using a custom library might prove difficult in

working with developers. Even if developers and designers agree that they can speak the same language, custom elements will still have to be made uniquely.

> *"Using a design system can definitely impact the delivery of the final product to the customer. We can build quicker iterations into code also as a team more quickly. It's not just how it looks like but how it works." (Designer 1)*

Visual design was also an issue one of the designers brought up. Using a pre-existing library might present lack of creativity in designing an interface.

> *"With so much time spent on trying to understand naming of elements and if a usable element can be provided by the system, much less time is spent on creative aspects of the design; the look and feel as well as the visual design. With a visual view of elements, I can see instantly what I need and then design the layout better." (Designer 2)*

After the designer tests, the questionnaires were handed out to the test subjects. As with the developers, they both felt unable to complete all the tasks in the given time-frame. In difference to the developers, designers found *Carbon* much more difficult to use and assets more disorganized than developers. For copy pasting elements into canvas and getting started, the mobile UI kit was found by designers to serve their purpose better and to be easier to use. Hence, the file structuring, naming and design system not being visually as present as the mobile UI kit, could be the reason for such vast difference for visually thinking

designers, whereas developers might think of assets in a different manner. Moreover, for someone not using design systems daily, keeping up to date on the updates and learning the how-tos, a visual kit can be more beneficial, not having to go through the steep learning curve of getting to know the design system.

Discussion: the Future of Design Systems

At the beginning of the research process, a design system was not highly utilized by the team setup where it was introduced. During, design systems became more familiar to the team, who were eager to understand, evaluate and assess design systems from a designer point of view and to start using them in their daily routine. Also, during this process design systems became a strong interest of the author. From a subjective standpoint, design systems improved methodology of quick iteration, communications with developers and gave more understanding of the current software used in the industry.

A clear understanding of what a design system is was not sure in the beginning. After the tests with the focus groups, the team members understood more how and what design systems are and how they could benefit from them.

It remains unclear as to whether design systems are still a buzzword[6] after 5 years in the making or a term that is permanent to the industry, since terminology is very confusing and there is no clear authority defining and setting boundaries for the concepts. Many refer to themselves as design systems: pattern libraries, style guides or brand guidelines, code or asset libraries or even obsolete

Sketch files. A clear clarification for what a design system is and what it is not is needed which will help the discussion and research surrounding the topic. As pointed out in Chapter 2, clarification as to what types of systems there are is still not present. These different types have different agendas which should be visible from the beginning to the potential user. It is sometimes made clear in the system's design language. In that chapter, the reader is presented with a concept for classifying systems.

As the boundaries around design systems are unclear, the lack of animations, gestures and moving objects or images is a potential challenge for design systems in the future. If in the user tests the developers had been given a static design without accordion opening animation, the test would have failed. Designing transitions, the moving of objects and triggering them is a vital part of a mobile design and static design systems do not serve these purposes. Inconsistencies appear if there are no guidelines to the timing of transitions or the way they are portrayed. Animation is often seen as the last part of a design, which is why Val Head (2019), the Senior Design Advocate at *Adobe*, proposes including *Motion Guidelines* as part of design systems as a unique entity (such as animation durations, easing guidelines and naming the effects). This might solve initial confusion and at least provide a starting point for designing animations. Some design systems have already implemented their own animation and motion guidelines, such as IBM's *Carbon* and Google's *Material Design*. Mostly, the aforementioned guidelines focus on transitions within interactions with user - how a push of a button happens and how the user gets from place A to place B. However, designing animation for something other than transitions - icons, videos, graphics - is left without consideration in design systems overall. To some extent, animations can be done,

but it still requires multiple software; exporting a vector graphic from *Sketch* to an animation software such as *Adobe After Effects*, from which then the designer hands out the design to a developer, who redesigns the video from the beginning again. Making an animation might require several steps. Without guidelines, these iterations can end up changing the final outcome. Without native animation possibilities in vector graphics editors such as *Sketch*, design systems fail to set boundaries outside of them. Other vector graphics tools such as *InVision Studio* are beginning to tackle this issue by introducing animation tools built-in.

The current stage of design system structure is visualized in various ways. Before the concepts have found a permanent definition, a design system can be seen as a color palette: Some of the colors might mix and overlap, whereas others are kept in form and separate. A designer draws the outcome with these colors from the palette and mixes them if necessary. Base colors define the principles for the design, highlights are elements which in turn emphasize on a particular aspect. As the lines between these colors are still undefined, the boundaries also remain translucent, as seen in figure 17.

In the aforementioned figure, the journey is upwards. We start from design language and end in the pattern library. As a basis for any design system, the principles and guides have to be defined - to whom is one designing this content, for which platforms, for what purpose, what tools should be used and how. After this, the style guide provides assets for typical issues (icons, logos, colors and photos). These might overlap, since a design language might provide some guides, such as layouts or typical icons. Lastly, the pattern library ties this knowledge together. A pattern library is the most visible part of a design system, on top of the below layers; how each of

these principles and elements should be used and where, with tools for both a designer and a developer - code snippets, design files and grids for layout. All elements might not be present in code in the pattern library, as things such as coloring might exist in universal code in style guides.

The user tests provided insight into team communication and particularly the different natures of work between designers and developers. As stated in Chapter 4, the developers noted that customization takes much longer than any other ready-made component in the design system and hence is left last. A bridge between designers and developers is essential and therefore full-stack developers or designers who code are an important asset to a design system team. If these team members do not exist, appropriate tools should be acquired or developed.

An example of such a tool is *Airbnb Design's* (Airbnb Design, n.d.) *React-Sketch.app*, originally launched to the public on the 29th of November 2016. (Airbnb Design, 2016) It enables developers to render their code instantly to design, develop iterations straight from code to Sketch. This tool could be seen as the way of the future for the UI field, where all the designed elements are already code that could be published to a particular coding language. However, the *React-Sketch.app* is only designed to work through code into *Sketch* and not vice versa, which means that designers who do not code are unable to use it. Then again, tools like these underline how design and development go hand in hand and how methodologies are getting closer to each other.

However, some argue that this bridge between developers and designers is everlasting and designing with code will not be the way of the future. (Van Dalen, 2016) Then again, this mindset of designers becoming hybrid designers

or unicorns[7] is seen as something that designers must get used to and learn the skills necessary in order to keep up to date in the industry, since hiring a unicorn is cheaper than hiring a developer and a designer. (Lipman, 2019) These type of designers are, however, seen as over-worked and not able to cover the grounds they work on entirely. As an employer, seeking for a unicorn in their team might reflect the lack of ability to build teams to employees. (Neeman, 2013) Rather, it would be beneficial to build upon creating better tools for mutual understanding between a designer and a developer, to enhance their workflow with tools such as *React-Sketch.app*. In a similar manner, a sandbox with existing drag-and-drop items such as the mobile UI kit, would be beneficial for all stakeholders discussed in this thesis: designers could ideate, test and prototype solutions together with clients and project managers and developers would be able to take these solutions into practice.

In a larger context, as design systems enable designers to disregard some aspects of the design process, one might wonder what role automation and machine learning might take in the future of designing user interfaces. Companies such as Adobe started developing artificial intelligence to make web design in 2017 and the largely referenced Airbnb have invested in automation for their designs too (Schwanley, 2018; Ungerleider, 2017). The next possible steps for design systems are efficient layouts with inputs for designer to drag and drop elements into place or to select a template from a ready-made library. Or, it could be something beyond imagination at the time of writing. However, it remains interesting to find out how the profession of a designer will change with the introduction of these automation systems in collaboration with design systems as well.

Overall, design systems are efficient user interface

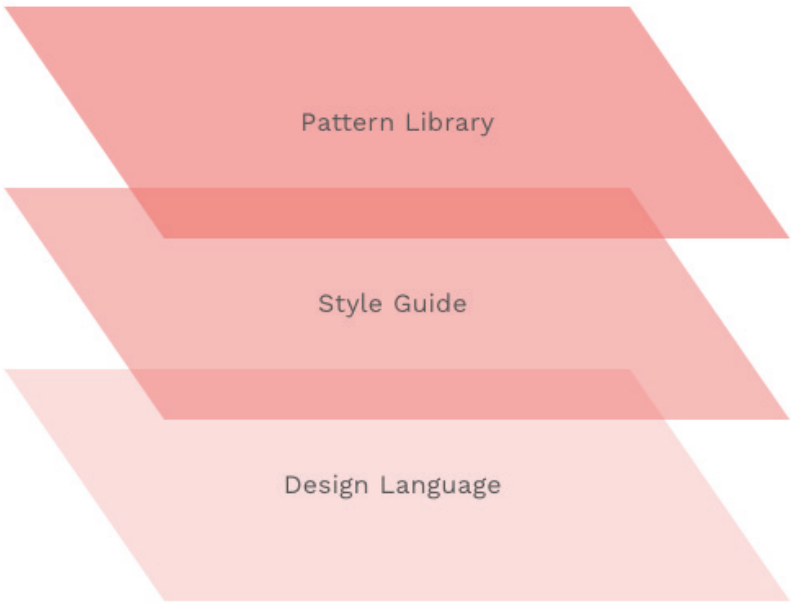


Figure 17. A look at how the structure of design system is seen. Adopted from Anna Molly's UI Design System Structure.

construction toolboxes that benefit all stakeholders to some extent. Without any research or theory into them and moreover to the motives behind such systems, the design systems used can get confusing with constantly changing software, methodologies and UI trends. The mobile UI kit underlines some of these issues with current design systems. Learning to use a system as well as browsing for a specific item in a tight timeframe can feel difficult and overwhelming. In such situations, a custom kit with visual aids can prove beneficial. However, designers and developers see the kit differently, and hence, the kit serves mostly designers. Without a

well-functioning code base, custom elements are still difficult to make as a developer. In the development of these code bases, the file structuring has to be addressed and, moreover, well-structured in the design system for the developers to quick start their work in tight timeframes.

Conclusions

Reflecting on the research presented in this thesis, the following conclusions can be made.

To what extent does a design system differ from other industry-used framework standards?

A design system is a maintained, unified framework for a team to design digital products. It enables users from different teams and areas to speak the same language and to design consistent products with a kit of reusable components, patterns on how to assemble them and principles on their uses. Other standards such as brand guidelines, design languages and patterns are considered to be particles and building blocks of design systems. Toolkits such as Bootstrap, are not considered design systems as they lack some of the vital elements that are seen to define a design system: a pattern library, a style guide and a design language. A design system is categorizable into a public and a private sector, under which sub-categories exist. This categorization is vital to understand the nature of the system and potential needs and uses for them. Where a company brand style guideline is evident in nature of its purpose, a design system might not have as

clear of an objective.

What are the benefits and limitations of using a single design system on several projects with different clients?

Design systems both limit customization as well as increase efficiency, as minimal problems that require time are removed from the workflow. Ready-made elements provide quick iteration, but might create a bias of solutions starting to look alike. Therefore customization of the system has to be taken into account by the designing party of the design system so that the design system works for large numbers of different stakeholders. Customization of the design system can be taken into account by generating an add-on to the system, such as a kit for a specific purpose, that adds particles to the system for a specific need.

To what extent does a design system help designers work more fluently with and between customers, front-end developers and co-workers?

Both designers and developers mentioned in their usability tests that the kit helped their work to some extent as well as a general design system. Moreover, a design system makes their work easier, since simple design problems do not have to be solved for each project, but they can start with a ready canvas and tools to build the product needed, as underlined by the developer test's accordion example on Chapter 4. However, for designers there exists

a with the tools they use: there is a steep learning curve exists in order to learn the specific naming of assets in a design system to quickly iterate on a project. Even if a design system makes their work more efficient and helps with communication with developers and other designers, the naming and software used can make it more difficult than it seems, if an override to an asset has to be made. Moreover, designers emphasized that the outcomes may start to seem alike when working with a single source of truth on every project. Hence, a set of customized elements is needed for designers to be able to change them quickly for different stakeholders. Customization is particularly important for clients to see their visual look or product in their own unique style. In this sense, a design system needs to be modular in nature: a designer should be able to change a style guide or design language to change the look to respond to clients' needs the best. In this way, the outcome of this research, the mobile UI kit with customizability, is essential - it does not limit designers to the ready-made design systems but builds on top of them by being able to change the parts the client requires.

As for developers, a design system repository is important, but likewise with designers, the software and structuring of these can make the usability more challenging for them on a project. For new projects, it is easier to replicate a file structure or a framework from a previous project and start building on top of it. However, the developers might end up creating the custom elements themselves rather than using ready-made code if the naming of the items is not underlined. Naming standards are essential to designers when creating a design and they need to be addressed when sharing works with developers. Moreover, as seen in the developer tests, developers struggle with the filing structure if naming conventions are not correct

and laid out in the design system. As with developers, a steep learning curve exists for using these repositories correctly.

Research Validity

As design systems are still a nonexistent research topic in the academic field, the drawn conclusions are hypothetical and only applied to the shared industry-specific knowledge. Many private companies and entities have conducted internal studies and research into their own design systems, which does not add to the knowledge in the academia, but rather echo prior findings and theories proposed by others in the field (Toman, 2018).

When first academic research evolves from this topic, further assessment can be made regarding the theory and practice of the uses of design systems. For this, testing multiple design systems on multiple clients would be a suitable framework to assess the effectiveness, cost, creative value and use cases of the design systems. Moreover, this would help draw boundaries for design systems as a research topic.

For user testing 4 users were tested, 2 from each focus group. Jacob Nielsen's (2000) standard has become testing typically 5 users for one user group, which leads to answering more than 85 % of the usability problems. For multiple test groups Nielsen recommends 3-4 users per group, which was not met in this study. However, this research should point out approximately 70 % of the usability problems with two users in each focus group and therefore be enough for the next iteration. Moreover, the system was created in a solitary management model, presented in Chapter 2, which means only one entity contributed to maintaining the system. Using a more open model would undoubtedly have had different results.

The outcome of the mobile UI kit was done in *Sketch*, which is one of the most used vector graphics editors in the UI and UX design industry (Sketch, 2019). Hence, this solution is problematic with other software; structuring of files in other software such as *Adobe XD* is not done in pages but rather everything in a single canvas. Therefore, this solution does not prove to be universal for every platform but rather for a single one. Design systems are mostly cloud based as a set of symbols, which will be the outcome of this project at a later iteration as well.

As the developer tests also pointed out, naming items is problematic between designers and developers. Where designers would pay attention to the careful naming of items, developers might neglect these entirely and rather focus on the hierarchy. This might suggest also that design systems are thought to serve designers first and developers second. To unify these systems, it would be beneficial to provide research into design system structuring to set out clear standards for what the contents of design systems are. With design systems in general, the naming and structuring might be a potential issue in the future alongside with motion and animation design.

This thesis has not touched the issue of versioning and changelogs of design systems, which is a large theme that goes hand in hand with managing and operating a design system. Since at the initial state of this concept, the system is managed from a solitary standpoint, versions are kept offline. *Carbon* for example updates constantly and syncs widely to all parties which have installed it via Sketch. The UI kit, once introduced, plans to work in a similar manner through *GitHub* as an updating file rather than as a library.

Future Development

The outcome of this thesis, the mobile UI kit, will be further developed in cooperation with team members from the received feedback and to cater to the needs of the users in real-life client situations. By using the kit as a sandbox, a starting point, the kit will start finding areas which are needed in addition to the current assets. As of now, this kit can be seen as an extra tool in the designer's toolbox when using *Carbon* as it underlines how to improve efficiency of a design system and create your own palette to accompany a design system to ease the learning curve of such a system and moreover the software it is based on.

For custom elements in the kit which are proven effective through client workshops, the next step is to design a code repository for *React* and *Swift* to implement quickly also into functional mobile user interfaces. Moreover, a drag and drop browser-based mockup of a user interface would be effective in a client setting to change elements to and from the display and show what could be done quickly and interactively.

As with the kit itself, the items which are found to be unnecessary can be removed from the sandbox; there could be an additional extra tools-page to which the infrequent tools can be added. Moreover, the layout references could include existing references from client workshops to support user interview claim of starting with the same mindset of a previous project to support a new one. More tests are to be done in order to achieve more understanding of the kit and its requirements in the workplace. Moreover, as with both the design system and the kit, more quantitative research proves to be necessary to find out how much both of these help both developers and designers' work in comparison to prior work methods.

The next iteration will also be taken into *GitHub* and shared with the team. This will also be a testing point for naming to establish a common understanding between designers and developers and furthermore try to build a functioning file structuring. With the user tests showing that naming of items does not prove effective for all stakeholders, there needs to be clear guidelines for use in general for the designers and developers. Moreover, for designers, the sandbox will be converted into a set of symbols to enable the use of the kit in a design system-like manner to not to rely on a single file with their work. Also other design software such as *Adobe XD* will be taken into consideration.

Open questions as to the future of this project concern the nature of the system itself. *Sketch* is at the time of writing the largest software in the industry as others are gaining popularity. Therefore, the project cannot be solely based on *Sketch* file structuring but needs to expand to more platforms. With development and further iterations of the kit, the elements have to be widely synced with the users. Version control in *GitHub* will account to some of these changes but requires users to also keep in sync. Moreover, as some of the elements are from *Carbon* and some custom-made, users have to be synced with two systems which requires more effort. With the next step of sharing the project to the team, feedback will prove to be essential to define the future of the concept presented in this thesis.

References

5 tips from an Airbnb designer on maintaining a design system. (n.d.). Retrieved July 10, 2019, from DesignSystems.com website: https://www.designsystems.com/5-tips-from-an-airbnb-designer-on-maintaining-a-design-system/

About Workday. (n.d.). Retrieved July 10, 2019, from Workday website: https://www.workday.com/en-us/company/about-workday.html

A design system for the federal government. (2019, June 20). Retrieved July 10, 2019, from 18F: Digital service delivery website: https://18f.gsa.gov/what-we-deliver/us-web-design-system/

Airbnb Design. (2016, November 29). airbnb/react-sketchapp. Retrieved July 17, 2019, from GitHub website: https://github.com/airbnb/react-sketchapp

Airbnb Design. (n.d.). Introduction · react-sketchapp. Retrieved July 17, 2019, from react-sketchapp website: http://airbnb.io/react-sketchapp/

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Shlomo, A. (1977). A Pattern Language: Towns, Buildings, Construction. Oxford University Press.

Android Open Source Project. (n.d.). Retrieved July 10, 2019, from Android Open Source Project website: https://source.android.com/

Araújo, J. (2018, September 10). Design Systems: benefits, challenges & solutions - UX Collective. Retrieved July 15, 2019, from Medium website: https://uxdesign.cc/design-systems-62f648c6dccf

Assistant Secretary for Public Affairs. (2013). User Interface Elements. Retrieved from https://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html

Babich, N. (2016, May 7). Primary & Secondary Action Buttons. Retrieved August 5, 2019, from Medium website: https://uxplanet.org/primary-secondary-action-buttons-c16df9b36150

Bank, C., & Cao, J. (2015). The Guide to Usability Testing - Free e-book by UXPin. Retrieved from https://www.uxpin.com/studio/ebooks/guide-to-usability-testing/

Berners-Lee, I. (2013, March 4). Principles of Design. Retrieved July 9, 2019, from Principles of Design website: https://www.w3.org/DesignIssues/Principles.html#Modular

Bohn, D. (2013, January 24). Redesigning Google: how Larry Page engineered a beautiful revolution. Retrieved July 10, 2019, from The Verge website: https://www.theverge.com/2013/1/24/3904134/google-redesign-how-larry-page-engineered-beautiful-revolution

Bootstrap from Twitter. (2011, August 11). Retrieved August 2, 2019, from https://blog.twitter.com/developer/en_us/a/2011/bootstrap-twitter.html

Brand Estonia. (n.d.). Retrieved July 10, 2019, from estonia.

eu website: https://brand.estonia.ee

Branding Guidelines. (n.d.). Retrieved July 9, 2019, from Spotify for Developers website: https://developer.spotify.com/branding-guidelines/

buzz, n.1 : Oxford English Dictionary. (2019, June). Retrieved July 15, 2019, from Oxford English Dictionary website: www.oed.com/view/Entry/25493

Caliman, D. (2015, August 21). To use or not to use Bootstrap Framework? - Creative Tim's Blog. Retrieved July 8, 2019, from Creative Tim's Blog website: https://blog.creative-tim.com/web-design/use-not-use-bootstrap-framework/

Chisholm, J. (2015, October 25). What is co-design? Retrieved July 8, 2019, from Design for Europe website: http://designforeurope.eu/what-co-design

Clark, J. (2017, April 3). The Most Exciting Design Systems Are Boring | Big Medium. Retrieved July 10, 2019, from https://bigmedium.com/ideas/boring-design-systems.html

Collins, M. (2016, April 15). Web Design Trends: Why Do All Websites Look The Same? | Friday Blog. Retrieved July 8, 2019, from Friday Digital Agency | A Digital Marketing & UX Agency based in Dublin website: https://www.friday.ie/blog/why-do-all-websites-look-the-same/

Curtis, N. (2009, January 9). Components Versus Patterns. Retrieved July 22, 2019, from UX Articles by UIE website: https://articles.uie.com/components_vs_patterns/

Curtis, N. (2015, September 17). Team Models for Scaling a Design System - EightShapes - Medium. Retrieved July 10, 2019, from Medium website: https://medium.com/eightshapes-llc/team-models-for-scaling-a-design-sys-

tem-2cf9d03be6a0

Curtis, N. (2017, January 27). Patterns ≠ Components - EightShapes - Medium. Retrieved July 9, 2019, from Medium website: https://medium.com/eightshapes-llc/patterns-components-2ce778cbe4e8

Dam, R., & Siang, T. (2019, May 5). Stage 2 in the Design Thinking Process: Define the Problem and Interpret the Results. Retrieved July 8, 2019, from The Interaction Design Foundation website: https://www.interaction-design.org/literature/article/stage-2-in-the-design-thinking-process-define-the-problem-and-interpret-the-results

De La Riva, M. (2019, June 13). 32 User Interface Elements For UI Designers: Your Ultimate Glossary. Retrieved July 14, 2019, from https://careerfoundry.com/en/blog/ui-design/ui-element-glossary/

Design principles | United States Web Design System. (n.d.-a). Retrieved July 10, 2019, from https://designsystem.digital.gov/about/design-principles/

Design principles | United States Web Design System. (n.d.-b). Retrieved July 11, 2019, from https://designsystem.digital.gov/about/design-principles/

Design Systems, when and how much? — Diana Mounter. (n.d.). Retrieved from https://www.youtube.com/watch?v=Hx02SaL_IH0

Design Systems - Evernote.Design. (2019). Retrieved July 10, 2019, from Evernote.Design website: https://www.evernote.design/categories/design-systems/

Esposito, E. (n.d.). 4 types of research methods all designers should know | Inside Design Blog. Retrieved July 10, 2019, from Inside Design Blog website: https://www.

invisionapp.com/inside-design/research-methods-designers/

Fanguy, W. (2017, December 19). The benefits of shared design systems. Retrieved July 15, 2019, from https://www.invisionapp.com/inside-design/shared-design-systems/

Fanguy, W. (2018, March 9). 3 lessons learned from the history of design systems. Retrieved July 9, 2019, from https://www.invisionapp.com/inside-design/3-lessons-learned-history-design-systems/

Fanguy, W. (2019, June 24). A comprehensive guide to design systems | Inside Design Blog. Retrieved July 9, 2019, from Inside Design by InVision website: https://www.invisionapp.com/inside-design/guide-to-design-systems/

Fogel, A. (2016, December 21). Platform guidelines matter and you should follow them. Retrieved July 10, 2019, from Medium website: https://medium.com/dashlane-insights/platform-guidelines-matter-and-you-should-follow-them-30c85bf59aa1

Frost, B. (2013). Atomic Design. Brad Frost Web.

Frost, B. (2014, November 18). Style Guides. Retrieved July 9, 2019, from Brad Frost website: http://bradfrost.com/blog/post/style-guides/

Frost, B. (2016). Maintaining Design Systems | Atomic Design by Brad Frost. Retrieved July 8, 2019, from http://atomicdesign.bradfrost.com/chapter-5/

Gallagher Aiden Dunleavy Jack. (2019, April 23). The Agile Method: Everything you need to know. Retrieved July 9, 2019, from IBM Developer website: https://developer.ibm.com/articles/agile-method-everything-you-need-to-know/

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software (Adobe Reader). Pearson Education.

Google. (n.d.). Material Design | Introduction. Retrieved July 9, 2019, from Material Design website: https://material.io/design/introduction/#principles

Gothelf, J., & Seiden, J. (2013). Lean UX: Applying Lean Principles to Improve User Experience. “O’Reilly Media, Inc.”

Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. Communications of the ACM, 28(3), 300–311.

Grainer, S. (2017, July 17). Design Systems, Style Guides, and Pattern Libraries: Oh My! - UXcellence. Retrieved July 8, 2019, from UXcellence website: https://uxcellence.com/2017/design-systems-style-guides-pattern-libraries

Hacq, A. (2017, June 28). Atomic Design & creativity - UX Collective. Retrieved July 15, 2019, from Medium website: https://uxdesign.cc/atomic-design-creativity-28ef74d71bc6

Hajdarbegovic, N. (2018, February 13). Learn About Bootstrap: Get Started Developing Responsive Websites. Retrieved July 8, 2019, from WhoIsHostingThis.com website: https://www.whoishostingthis.com/resources/bootstrap/

Head, V. (2019, February 21). Including Animation In Your Design System — Smashing Magazine. Retrieved July 15, 2019, from Smashing Magazine website: https://www.smashingmagazine.com/2019/02/animation-design-system/

Hello World. (2016, April 7). Retrieved July 9, 2019, from GitHub Guides website: https://guides.github.com/activities/hello-world/

IBM. (n.d.-a). Carbon Design System. Retrieved July 9, 2019, from Carbon Design System website: https://www.carbondesignsystem.com/getting-started/about-carbon

IBM. (n.d.-b). Philosophy. Retrieved July 9, 2019, from IBM Design Language website: https://www.ibm.com/design/language/philosophy/principles/

IBM Enterprise Design Thinking. (n.d.). Retrieved July 10, 2019, from IBM Enterprise Design Thinking website: https://www.ibm.com/design/thinking/

Identitytool for Sweden. (n.d.). Retrieved July 10, 2019, from Identitytool for Sweden website: https://identity.sweden.se

Keshtcher, Y. (2017, August 30). Top 22 Prototyping Tools For UI And UX Designers 2019. Retrieved July 15, 2019, from Medium website: https://blog.prototypr.io/top-20-prototyping-tools-for-ui-and-ux-designers-2017-46d59be0b3a9

Laubheimer, P. (2016, March 27). Front-End Style-Guides: Definition, Requirements, Component Checklist. Retrieved July 9, 2019, from Nielsen Norman Group website: https://www.nngroup.com/articles/front-end-style-guides/

Lee, G. “gb.” (2017, June 27). Designer’s indispensable skill: the ability to write and present a solid problem statement. Retrieved July 8, 2019, from UX Planet website: https://uxplanet.org/designers-indispensable-skill-the-ability-to-write-and-present-a-solid-problem-statement-56a8b4b8060

Limcaco, J. (2019). Design Systems Gallery. Retrieved July 10, 2019, from Design Systems Repo website: https://designsystemsrepo.com/

Linders, B. (2019, May 24). How Design Systems Support Team Communication and Collaboration. Retrieved July 15, 2019, from InfoQ website: https://www.infoq.com/news/2019/05/design-systems-teams/

Lipman, A. (2019, March 4). Unicorn Rare Hybrid: On Being A Designer And A Developer | Women Who Code. Retrieved July 17, 2019, from Women Who Code website: https://www.womenwhocode.com/blog/unicorn-rare-hybrid-on-being-a-designer-and-a-developer

Lonely Planet Style Guide. (2019). Retrieved July 9, 2019, from Lonely Planet Travel Guides and Travel Information website: https://rizzo.lonelyplanet.com/styleguide/ui-components/cards

Malone, E. (2017, March 31). A History of Patterns in User Experience Design - erin malone - Medium. Retrieved July 8, 2019, from Medium website: https://medium.com/@emalone/a-history-of-patterns-in-user-experience-design-f21f7eaabb83

Mandelbaum, M. (2014). Design Principles. Retrieved July 15, 2019, from http://learndesignprinciples.com/index.html

McKenna, B. (2018, November 8). The learning curve design problem - UX Collective. Retrieved July 14, 2019, from Medium website: https://uxdesign.cc/the-learning-curve-design-problem-4d4dc2965098

Molly, A. (2017, February 6). Basic about UI design system. Retrieved September 1, 2019, from https://medium.com/@notannamolly/building-ui-design-system-fb5d-

c5b58dc5

Muratovski, G. (2015). Research for Designers: A Guide to Methods and Practice. SAGE.

Neeman, P. (2013, July 25). The Unicorn Designer Dilemma: How To Avoid It. Retrieved July 17, 2019, from Usability Counts website: http://www.usabilitycounts.com/2013/07/24/unicorn-designer-dilemma-avoid/

Nielsen, J. (2000, March 19). Why You Only Need to Test with 5 Users. Retrieved July 15, 2019, from Nielsen Norman Group website: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

Palkó, B. (2019, January 9). Good Design Practices: Naming Conventions - UX Planet. Retrieved July 16, 2019, from Medium website: https://uxplanet.org/good-design-practices-tips-file-naming-conventions-af32eba681f2

Printing Press. (n.d.). Retrieved July 8, 2019, from Encyclopedia Britannica website: https://www.britannica.com/technology/printing-press

Rubin, J., & Chisnell, D. (2011). Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons.

Saarinen, K. (2018, June). Building a Visual Language. Retrieved July 9, 2019, from Airbnb.Design website: https://airbnb.design/building-a-visual-language/

SAP Fiori Design Guidelines. Retrieved July 9, 2019, from https://experience.sap.com/fiori-design/

Serrault, D. (2019, May 7). Design systems at scale - UX Collective. Retrieved July 15, 2019, from Medium website: https://uxdesign.cc/design-systems-at-scale-

5b3beb1b50c9

Shanley, A. (2018, April 9). The Future of UX Design Is Automation. Retrieved July 10, 2019, from https://www.uxmatters.com/mt/archives/2018/04/the-future-of-ux-design-is-automation.php

Shopify Polaris. (n.d.). Retrieved July 10, 2019, from Shopify Polaris website: https://polaris.shopify.com/

Sketch. (2019, March 13). Sketch raises $20m in Series A funding from Benchmark. Retrieved August 5, 2019, from Medium website: https://blog.sketchapp.com/sketch-raises-20m-in-series-a-funding-from-benchmark-ea298764d7d1

Skjoldbroder. (2018, June 18). Design system weaknesses. Retrieved July 8, 2019, from Prototypr website: https://blog.prototypr.io/design-system-weaknesses-81a562232d98

Storybook. (n.d.). Retrieved July 10, 2019, from https://react95.github.io/React95/

Storybook: UI component explorer for frontend developers. (n.d.). Retrieved July 16, 2019, from https://storybook.js.org

The Interaction Design Foundation. (n.d.). What are Design Principles? Retrieved July 10, 2019, from The Interaction Design Foundation website: https://www.interaction-design.org/literature/topics/design-principles

The Interaction Design Foundation. (2019, June 28). Google's Material Design - Android Design Language. Retrieved July 9, 2019, from The Interaction Design Foundation website: https://www.interaction-design.org/literature/article/google-s-material-design-android-design-language

Tidwell, J. (2005). Designing Interfaces: Patterns for Effective Interaction Design. “O’Reilly Media, Inc.”

Toman, J. (2017, May 20). Design systems, style guides, pattern libraries. What the hell is the difference? Retrieved July 9, 2019, from Product Unicorn website: https://product-unicorn.com/design-systems-style-guides-all-those-libraries-what-the-hell-is-the-difference-4c2741193fdc

Toman, J. (2018, March 9). Our internal research for a new design system & tips on how to do it. Retrieved July 15, 2019, from Medium website: https://design.kiwi.com/our-internal-research-for-a-new-design-system-tips-on-how-to-do-it-9e703404e30f

Ungerleider, N. (2018, July 9). Adobe Is Building An AI To Automate Web Design. Should You Worry? Retrieved from https://www.fastcompany.com/3068884/adobe-is-building-an-ai-to-automate-web-design-should-you-worry

uxplanet.org. (2018, March 8). Are UX Developers a Thing? Why Hybrid Designers Are So In-Demand Right Now. Retrieved July 17, 2019, from Medium website: https://uxplanet.org/are-ux-developers-a-thing-why-hybrid-designers-are-so-in-demand-right-now-9956716598cf

Van Dalen, J. (2016, September 20). Why You Shouldn’t Hire A Designer/Developer Hybrid - Modus Create. Retrieved July 17, 2019, from Modus Create website: https://moduscreate.com/blog/why-you-shouldnt-hire-a-designer-developer-hybrid/

Vendrik, K. (2019, July 10). Getting design system customization just right - Shopify UX. Retrieved July 11, 2019, from Medium website: https://ux.shopify.com/getting-design-system-customiza-

tion-just-right-3012151ef5ea

Endnotes

1 Designers were to use Sketch in two phases; first phase was to design a certain type of an interface with using Carbon Design System. On the second phase, the mobile UI kit was given and the task was remade. For both stages, designers were given 15 minutes to complete, 30 minutes in total. The tasks can be found in the appendix.

2 Developers were to code a functioning web application based on the sketch provided in two phases; first, the developers were given an image of the UI and second phase access to the mobile UI kit. Both phases had 30 minutes for completion, 1 hour in total. After these tasks the questionnaire and interview were conducted. The tasks can be found in the appendix.

3 List of elements in order of appearance: accordion, breadcrumb, button, card, card as image, carousel cue, checkboxes, dropdown, feed, form, loading, menu, modals, notifications, pagination, picker, search bar, switchers and sliders, tabs, toggles and tooltips. Carbon Design System is open source in nature and some elements are implemented and customized from there.

4 Custom elements were a carousel, carousel cue and the header. Design system elements were buttons, accordion and tile.

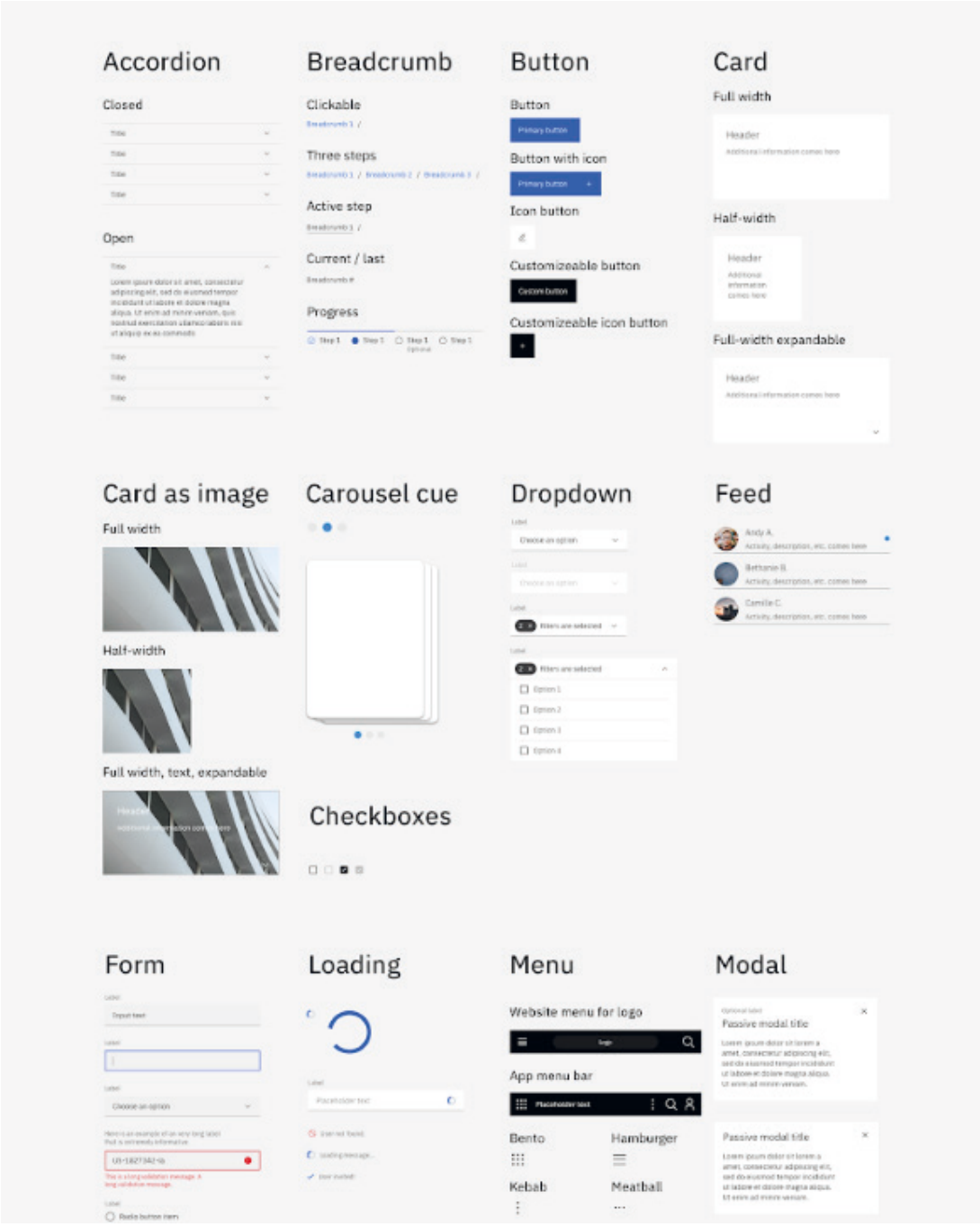
5 “Storybook is an open source tool for developing UI components in isolation for React, Vue, and Angular. It makes building stunning UIs organized and efficient.” (“Storybook: UI Component Explorer for Frontend Developers” n.d.)

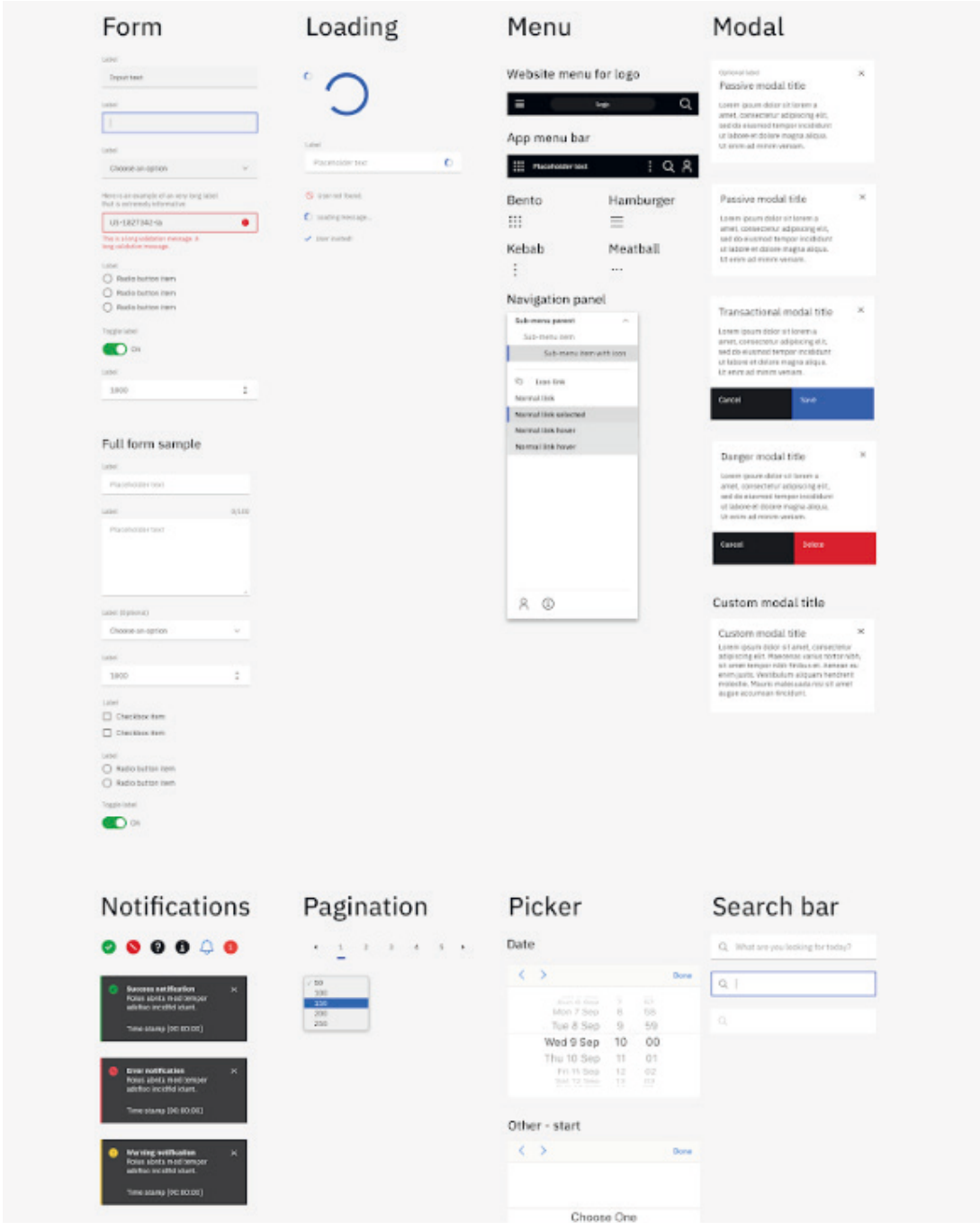
6 Buzzword is “a catchword or expression currently fashionable; a term used more to impress than to inform, esp. a technical or jargon term.” (“Buzz, n.1 : Oxford English Dictionary” 2019)

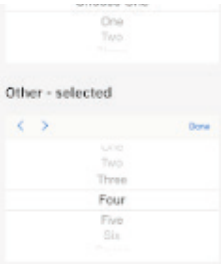
7 “Hybrid designers, otherwise known as designer-developer unicorns, are not only masters of wireframing, user testing and visual design; they can also pack a punch in the programming department.” (uxplanet.org 2018)

Appendix

Asset panel from Sandbox (in three parts)




Form
Full form sample
Loading
Menu
Website menu for logo
App menu bar
Bento
Hamburger
Kebab
Meatball
Navigation panel
Modal
Passive modal title
Passive modal title
Transactional modal title
Cancel
Save
Danger modal title
Cancel
Delete
Custom modal title
Custom modal title
Notifications
Pagination
Picker
Date
Mon 7 Sep 8 56
Tue 8 Sep 9 59
Wed 9 Sep 10 00
Thu 10 Sep 11 01
Done
Other - start
Choose One
Search bar



Switchers and sliders

Tabs

Tooltips

Word tooltip

Icon tooltip

Arrow below

Arrow aside

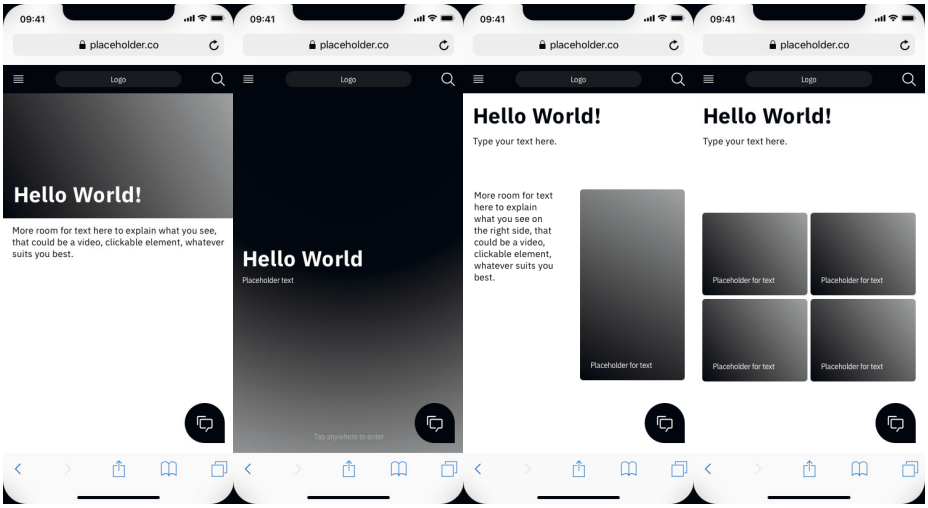
Arrow top

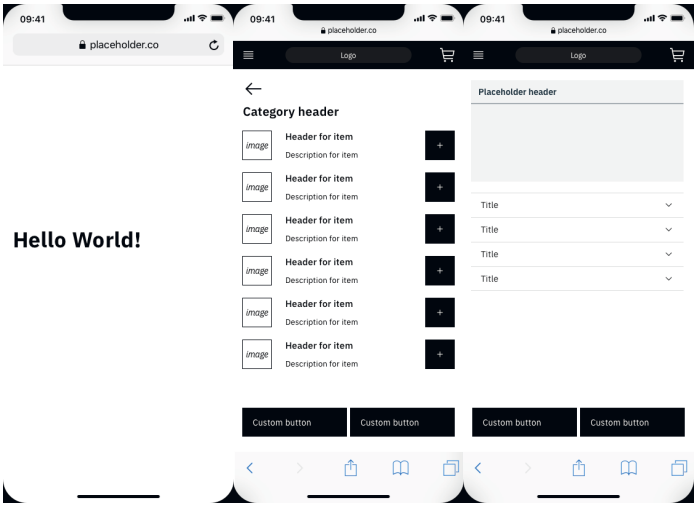
Toggle

Interactive

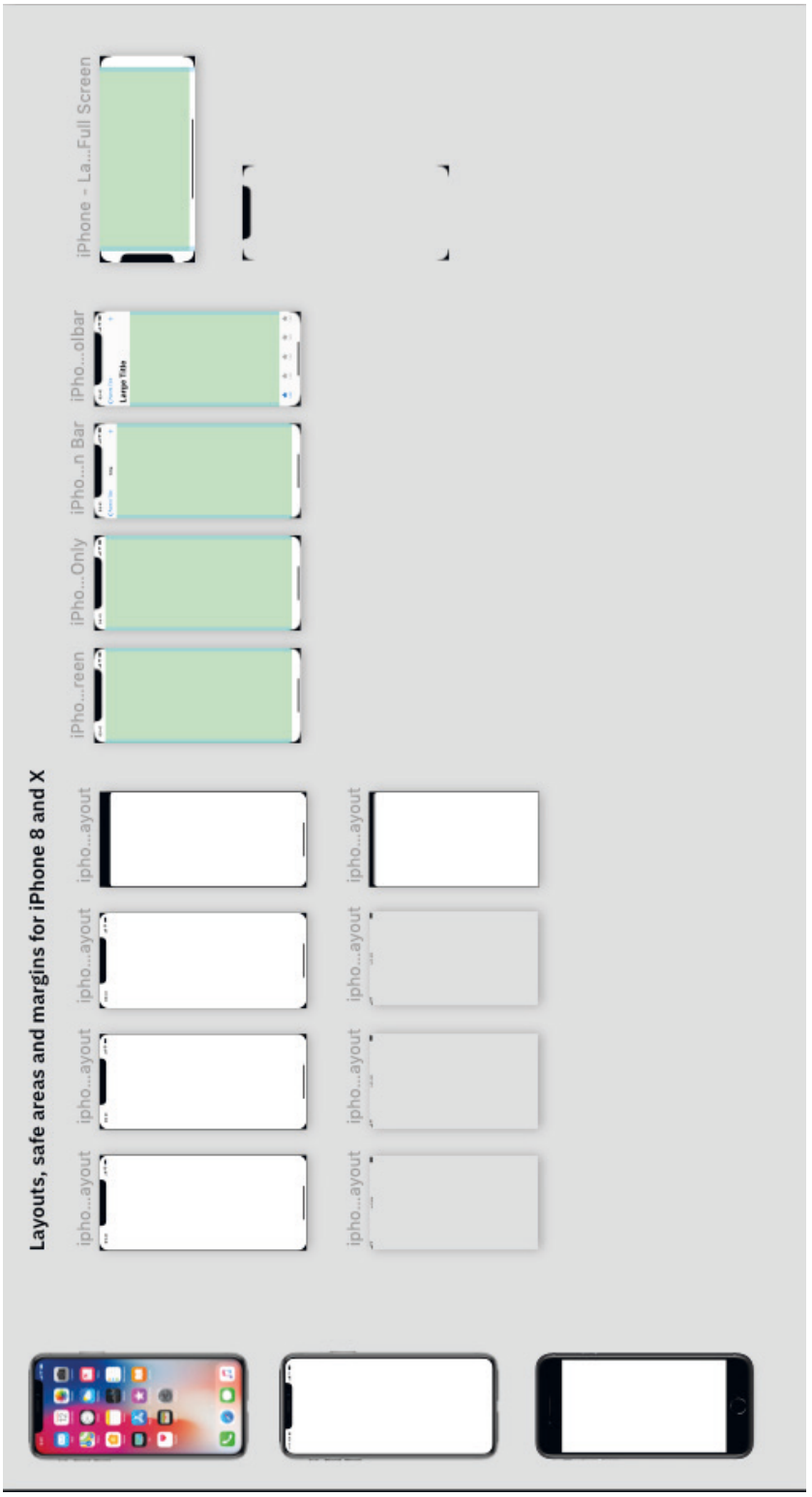
Empty boxes

Reference layouts





Margins


Layouts, safe areas and margins for iPhone 8 and X
iPho...reen
iPho...Only
iPho...n Bar
iPho...olbar
iPhone - La...Full Screen
Large Title
ipho...ayout

Questionnaire

Usability test of Carbon Design System and the mobile UI kit
Developer focus group
16th July 2019

Question 1 Do you feel that you successfully completed all the tasks?

Yes **No**

Question 2.1 I found Carbon Design System...

Very difficult to use **1 ... 2 ... 3 ... 4 ... 5 ... 6** **Very easy to use**

Question 2.2 The assets were well organized and they were easy to find.

Strongly disagree **1 ... 2 ... 3 ... 4 ... 5 ... 6** **Strongly agree**

Question 2.3 All of the functions I expected to find in the menus were present.

Strongly disagree 1 ... 2 ... 3 ... 4 ... 5 ... 6 Strongly agree

Question 2.4 What assets did you find most difficult to replicate?

Question 3.1 I found mobile UI kit to be...

Very difficult to use 1 ... 2 ... 3 ... 4 ... 5 ... 6 Very easy to use

Question 3.2 The assets were well organized and they were easy to find.

Strongly disagree 1 ... 2 ... 3 ... 4 ... 5 ... 6 Strongly agree

Question 3.3 All of the functions I expected to find in the menus were present.

Strongly disagree 1 ... 2 ... 3 ... 4 ... 5 ... 6 Strongly agree

Question 3.4 What assets did you find most difficult to replicate?

Master of Art Thesis
Janne Koivisto

Aalto University
School of Arts, Design and Architecture
Department of Media
Aalto Media Lab

Supervisor
Teemu Leinonen

Advisor
Ville Sirén

Print
Picaset