



Expertise  
and insight  
for the future

Tuukka Rissanen

# Using Open Source E-Commerce Platform for Public Sector Procurements

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

22.4.2020

Author Title	Tuukka Rissanen Using Open Source E-Commerce Platform for Public Sector Procurements – Graphical User Interface Perspective
Number of Pages Date	53 pages + 1 appendices 22 April 2020
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructor	Vesa Ollikainen, Senior Lecturer
<p>E-commerce, in today's digitalized world, is changing the way shopping is done. Especially in the consumer market, suppliers and retailers have been heavily investing in and developing the e-commerce platforms and their user interface design, with the aim of making the shopping experience as convenient as possible. However, in the context of procurement, the systems and their user interfaces have not kept up with such rapid development. Using the traditional procurement applications can be labor intensive, and a more user-centric user interface could potentially improve the processes in procurements.</p> <p>Consequently, the background to this thesis derives from the idea of learning from the graphical user interfaces of B2C e-commerce platforms to improve public sector procurement in a Finnish health care organization. This thesis constitutes of three research objectives. First, the key factors of a graphical user interface of an e-commerce platform were identified based on prior literature. The second objective was to select a valid open source e-commerce application based on the selected evaluation criteria. The final objective was to evaluate, from the graphical user interface perspective, how well the selected application would meet the procurement related requirements of the client. In addition, in the evaluation phase, a set of key modifications were identified in order to improve the feasibility of the open source alternative for procurement use.</p> <p>The user interface of the selected application was evaluated with a usability testing technique: a test user – familiar with public-sector procurements in the health care sector – was given procurement tasks which followed the theoretical framework of the study. Applying qualitative methods, i.e. interviews, each task was then discussed with the test user, aiming to construct an understanding of how well the graphical user interface of the open source e-commerce platform suited the procurement related requirements of the test user.</p> <p>The main findings of the thesis argue that the selected open source e-commerce application has potential, in the graphical user interface perspective, to provide an alternative for the procurement platforms currently used in the organization which the test user works for. Furthermore, the thesis implies that there are clear similarities between the customer journey in e-commerce and in procurement, proving that the user centric elements in graphical user interfaces of e-commerce applications could be further introduced to the public sector procurement processes.</p>	
Keywords	User Interface, Graphical User Interface, E-commerce, Open Source Software

## Contents

### List of Abbreviations

1	Introduction	1
2	Problem Definition and Objectives	2
3	User Interface Design in E-commerce Applications	6
3.1	Breaking Down User Interfaces	6
3.2	Graphical User Interfaces in Web Development	7
3.3	Graphical User Interface Factors in E-Commerce	9
3.3.1	Site Landing	11
3.3.2	Product Discovery	13
3.3.3	Product Presentation	15
3.3.4	Cart Management	17
3.4	Summary	17
4	Open Source E-Commerce Platforms	20
4.1	Open Source Software	20
4.2	Open Source License Types	21
4.3	Comparison of Selected Open Source License Types	24
4.4	Selection of Most Appropriate Platform	25
4.5	Selected Platform and Project Setup	27
4.6	Project Setup	29
5	Methodology and Research Approach	32
6	User Interface Evaluation of Selected Open Source Application	35
6.1	Graphical User Interface Factors of Selected Open Source Platform	35
6.1.1	Site Landing	36
6.1.2	Product Discovery	37
6.1.3	Product Presentation	39
6.1.4	Cart Management	41
6.2	Summary	42
7	Discussion and Conclusions	44

7.1	Theoretical Discussion	45
7.2	Managerial Recommendations	46
7.3	Suggestions for Development and Further Research	46
	References	49

## Appendices

Appendix 1. Set of Tasks and Questions for the Test User

## Table of Figures

Figure 1, Web based e-commerce platform in public sector procurements	3
Figure 2, Visualization of the Research Objectives in a Process Chart	5
Figure 3, Web Application Architecture [11]	8
Figure 4, Conceptual Framework based on the Customer Journey Framework of Mangiaracina and Brugnoli (2009) [42].	10
Figure 5, The F-Shaped Pattern [19].	12
Figure 6, Product search examples in ebay.com	13
Figure 7, Breadcrumb trail in ebay.com	14
Figure 8, Product Card and Product Listing in walmart.com	15
Figure 9, List view vs. Grid view [24]	16
Figure 10, Two types of Open Source Licenses [29]	22
Figure 11, Top Open Source Licenses 2019 [31].	23
Figure 12, MIT, Apache and BSD Compared [34].	25
Figure 13, Cezerin Architecture	28
Figure 14, Cezerin Project Folder Structure	31
Figure 15, Summary of User Interface Evaluation Techniques by Jeffries et al. (1991).	33
Figure 16, Home Page View of the Application	36
Figure 17, Breadcrumb Trail of the Application	38
Figure 18, Product Listing of the Application	40
Figure 19, Cart Preview in the Application	41
Figure 20, Identified Needs for Development of the Cezerin Platform	43

## List of Abbreviations

B2C	Business to Customer – A setting where the business activities, for instance selling, occurs from a company, or business, to an individual consumer, or customer.
GUI	Graphical User Interface – A user interface that contains graphical elements, such as colors, icons, images, buttons, etc. The interaction of giving inputs and receiving outputs happens, in a graphical user interface, through the combination of these element
JSON	JavaScript Object Notation – A simple format for transmitting data. It is a human-readable format due to its attribute-value structure. Despite the name, JSON is not JavaScript dependable but rather used in wide variety of systems.
NoSQL	Not Only SQL / Non SQL – A Database model where the architecture does not follow relational tables, which make NoSQL databases horizontally scalable.
OSS	Open Source Software – A type of software where the source code, which conducts the software, is accessible by anyone.
UI	User Interface – A Point of human-computer interaction through which the user gives controls to the system and receives information from the system

## 1 Introduction

Online shopping, commonly referred to as e-commerce, is an example of an industry in which the global trade of goods has been facilitated by these leaps in technology. As the traditional retail business is continuously moving online, suppliers are developing more and more innovative ways to attract customers and make the shopping experience increasingly customer friendly. This is clearly an important objective since the potential number of consumers that shop online is estimated to be around 2.05 billion in 2020 [1].

The key to the process of making shopping customer friendly lies in the layout and the overall web-design of the e-commerce application. In other words, one of the most important aspects in e-commerce is the user interface (UI) of the application, which entails how the customer interacts with the service provider [2]. To illustrate this with a few examples, a good user interface most likely includes aspects such as simple navigation and fast checkouts. The importance of the user interface lies in its effect on business since, based on statistics, approximately 76 customers out of 100 abandon their cart during the purchasing process [3]. This proves that an illogical and confusing website with bad user experience through poor user interface design can be detrimental for sales. On the other hand, a well-designed site delivering smooth experience for customers is most likely to increase sales for the supplier.

As illustrated in the examples above, a user interface is a crucial part of e-commerce in a business to customer (B2C) setting. The importance can be seen when visiting any of the most well-known e-commerce platforms. Amazon and eBay, for instance, are prime examples of how the graphical user interface (GUI) provides a trouble-free and customer friendly user experience. Most e-commerce customers are likely to agree that such platforms designed with the ease of use in mind, such as navigation, proper pictures, layout and overall clarity, offer a smooth alternative to traditional brick and mortar shopping.

## 2 Problem Definition and Objectives

Even though the platforms in the B2C market have seen vast investments and developed rapidly over the past few decades, it could be argued that procurement functions have not kept up with the technological developments in terms of the graphical user interface. Procurements in companies are often done using by using heavy and stiff software. The user centricity that has been an essential focus in B2C e-commerce industry has not yet fully transferred to procurement processes in public and private sector companies.

The potential inefficiency can be illustrated through an example scenario where a Finnish health care center has to order a product that is running out. The usual case in such situations is that a staff member, such as a doctor or a nurse orders the necessary product using an existing platform. These platforms used especially in the public sector include desktop applications often based on SAP. In some scenarios, the employees might even use a credit card and make purchases in a nearby wholesale.

Since the purchases are predominantly made by employees whose main area of expertise does not lie within IT, the traditional desktop applications may be hard to use. For instance, in many cases the orders have to be placed by using a specific product number or product identification series, rather than selecting the option from a catalogue of products represented with pictures. In addition, the lack of graphical and user-guiding elements makes the software slow to use which, inevitably, increases costs in terms of labor. The second option of using a corporate credit card has its downsides as well since it is, at least in many cases, inefficient to spend time driving to a wholesale and back for a product that could be ordered through an online platform. Compared to the ease of use of the e-commerce platforms, it would seem that both options described above could be inefficient and eventually expensive. These case examples, which have been identified through a variety of conversations with the Finnish health care staff, suggest that an alternative solution could be beneficial.

The background to this thesis derives from the idea of learning from graphical user interfaces of B2C e-commerce platforms to improve public sector procurement. The current purchasing processes could potentially be made smoother and more pleasant to the staff by introducing an alternative procurement application. The long-term implication

deriving from the study would be, in short, to be able to provide public sector organizations a platform that utilizes the user centric elements already present in today's e-commerce platforms. This, in turn, could potentially make the procurement faster, cheaper and easier for the people responsible for it. The concept of the platform and its transformational effect on the process is visualized in Figure 1.

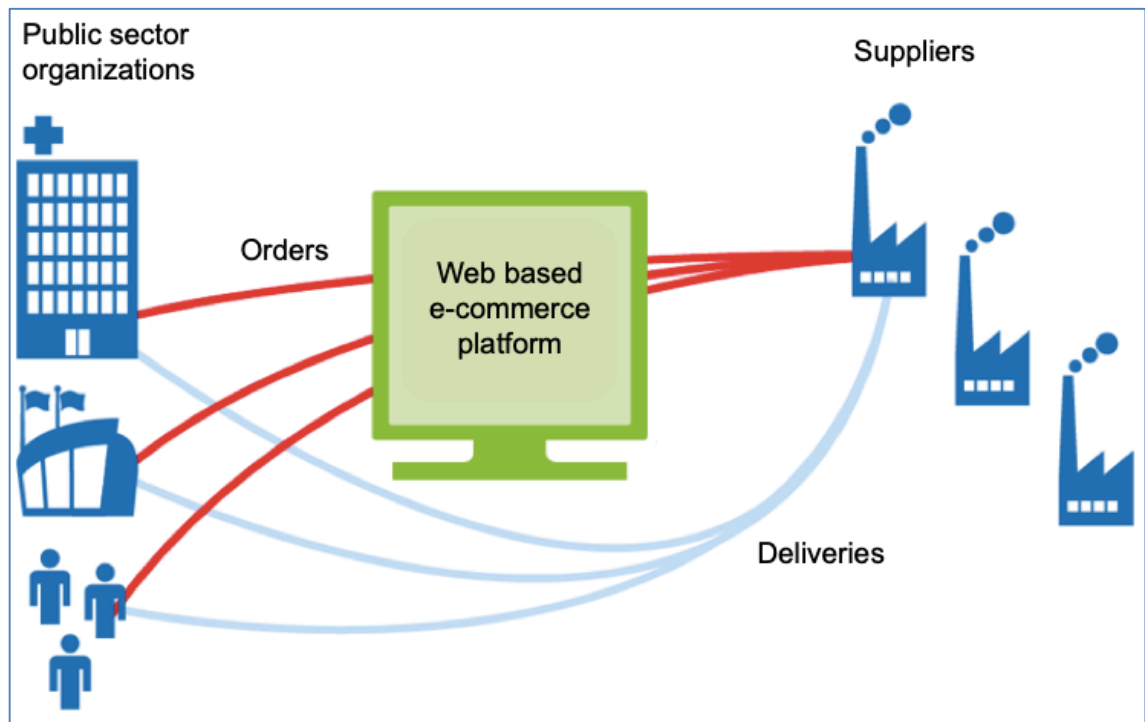


Figure 1, Web based e-commerce platform in public sector procurements

As represented through Figure 1, this study focuses on the fulfillment phase of the procurement where the buyer has already selected the suppliers. In other words, the focus is put on the methods of how and where the buyer places the actual orders. To further elaborate this, the idea is as follows: the public-sector organization first selects the suppliers based on tender processes, after which the suppliers are selected. The products selected by the tender process are then added to the web-based e-commerce application. At this point, the public-sector staff can now place orders through the platform based on their current needs, and the orders are transmitted to the supplier.

The author is a shareholder in a company called MidLog Oy and the results of the study are to be applied in product development of the company. In addition to providing useful theoretical perspectives, the thesis also serves as a means to present the potential client

with an alternative solution for procurements which could be developed further in the future.

The first objective of this study was to investigate the elements that build a user centric web-based e-commerce platform. Through this, the idea was to gather a well-rounded understanding of the factors that constitute a user-friendly e-commerce platform. Understandably, there are numerous aspects related to e-commerce applications and their user interfaces and it should be pointed out that the present study does not aim to discuss every possible factor. The objective was rather to identify the key factors that are obligatory parts of a graphical user interface of an e-commerce application. The graphical user interface perspective was chosen based on the personal interest as well as the future career plans of the author.

The second objective was to evaluate different options for the base of the e-commerce platform. For this purpose, a set of open source e-commerce platforms were evaluated. From the open source projects, the most suitable one was then selected for the base of the present study and project.

Finally, the user centric elements and the feasibility of the selected open source platform were evaluated. In other words, the graphical user interface of the selected open source platform was evaluated with selected research methods. The evaluation was mainly done by presenting the open source alternative to a test user, able to evaluate if the presented platform, from the graphical user interface perspective, would suit the business requirements of the organization in which the test user is employed in. Through the evaluation the aim was to identify if the open source alternative would be a viable option and if not, could it become one with modifications, and finally what should those modifications be. In addition to evaluating the feasibility from the test user viewpoint, the theoretical aspects were derived to the empirical part of the study to see if the factors in consumer e-commerce platforms are transferrable to the public-sector procurement context. This helps in understanding whether there are differences in the requirements for user interfaces between the consumer-centric e-commerce market and public-sector procurement.

Derived from the elaboration above, the objective of this study can be partitioned into

three research questions:

- 1) Which factors constitute a user-centric user interface (UI) in a web-based e-commerce platform?
- 2) Which open source e-commerce platform should be selected to be evaluated?
- 3) Does the user interface of the selected open source platform meet the requirements of the test user and if not, is there a need for modifications and which these modifications would be?

Another way to illustrate the objectives is to think about the research questions as a part of a process. There is a described business purpose behind the idea of the procurement platform, hence the objective can be understood as a process with individual tasks. This process chart is visualized in Figure 2.



Figure 2, Visualization of the Research Objectives in a Process Chart

The thesis proceeds with an introduction to user interface design in e-commerce applications, presented in Chapter 3. This is followed by an overview of open source e-commerce platforms and a selection of the most applicable one in Chapter 4. Thereafter, the methodology and research approach are discussed in Chapter 5, followed by the user interface evaluation of the selected open source platform in Chapter 6. Finally, the

findings, managerial implications and suggestions for further research are presented in Chapter 7 which concludes the thesis.

### **3 User Interface Design in E-commerce Applications**

As stated in the objectives, a part of the study was to go investigate elements in e-commerce platform design that are considered to improve user experience. The purpose of this chapter is to find out what kinds of factors in e-commerce platforms constitute a viable graphical user interface and, more specifically, what features make these factors user-centric.

#### **3.1 Breaking Down User Interfaces**

Before digging deeper into the user interfaces in the e-commerce context, it is beneficial to break down the term user interface and the meaning behind that term. To start with a dictionary definition, the Oxford Learner's Dictionary defines user interface as "the way a computer gives information to a user or receives instructions from a user" [4]. This definition can be further elaborated with the notion that a user interface is an interface in which the user of the application gives controls and receives information from the application [5]. In simple terms, a user interface, or a UI for short, is the part between human and technology where the interaction happens [6]. It should be emphasized, that as the dictionary definition states, the interface both receives inputs and provides outputs and hence, the information flow in a user interface is two-way.

Presumably for the younger generations the first association of the term user interface relates to the fast, colorful and graphical interfaces that can be found in every phone and computer. However, user interfaces have not always been this sophisticated; when the concept of a computer was first invented, the commands were text-based and inserted into a command line. Depicted in the modern perspective, the first user interfaces were similar to the current programming environments. The most basic interaction the user had with the computer was similar to actual programming. It was not until the year 1979 that a company named Xerox developed a prototype that has been considered to be the first version of a graphical user interface; typically referred as graphical user interface.

The graphical user interface idea of Xerox was then developed further by Steve Jobs, whose company Apple later introduced Lisa, a computer that had a fully working graphical user interface. [7,8].

As all modern-day computer and application users know, the user interfaces in practically all occasions are graphical user interfaces. This development has been one of the reasons that made personal computers and software available to the public: due to the graphical user interfaces, the use of computers and software was no more reliant on programming and rather difficult text based commands. Graphical user interfaces introduced a mouse, buttons, icons, pictures and colors that were understandable to the wide public who did not necessarily have competence or prior knowledge in computer sciences or programming. [7,9].

In conclusion, a user of an application or computer operates the target with commands that are inserted through a user interface. At the same time the user can observe the outcomes of the commands through the same interface by interpreting the outputs. In a graphical user interface, the commands are given and system outputs observed through different graphical elements including text, graphics, buttons, colors, and icons. As the user interfaces in web development are graphical user interfaces, from now on the term graphical user interface is used.

### 3.2 Graphical User Interfaces in Web Development

Considering the importance of the graphical user interfaces in providing easy access to all applications, they are a de facto in today's web development. It can be assumed that every web application is interacted with through a graphical user interface. Architecture-wise, the graphical user interfaces

in web-development are displayed to the client in a web browser. The code that compiles the graphical user interface in browser for the client is referred as the client side code [10]. In some instances, the client side of the software is defined as the front-end. The positioning of the frontend in web application architecture is presented in Figure 3.

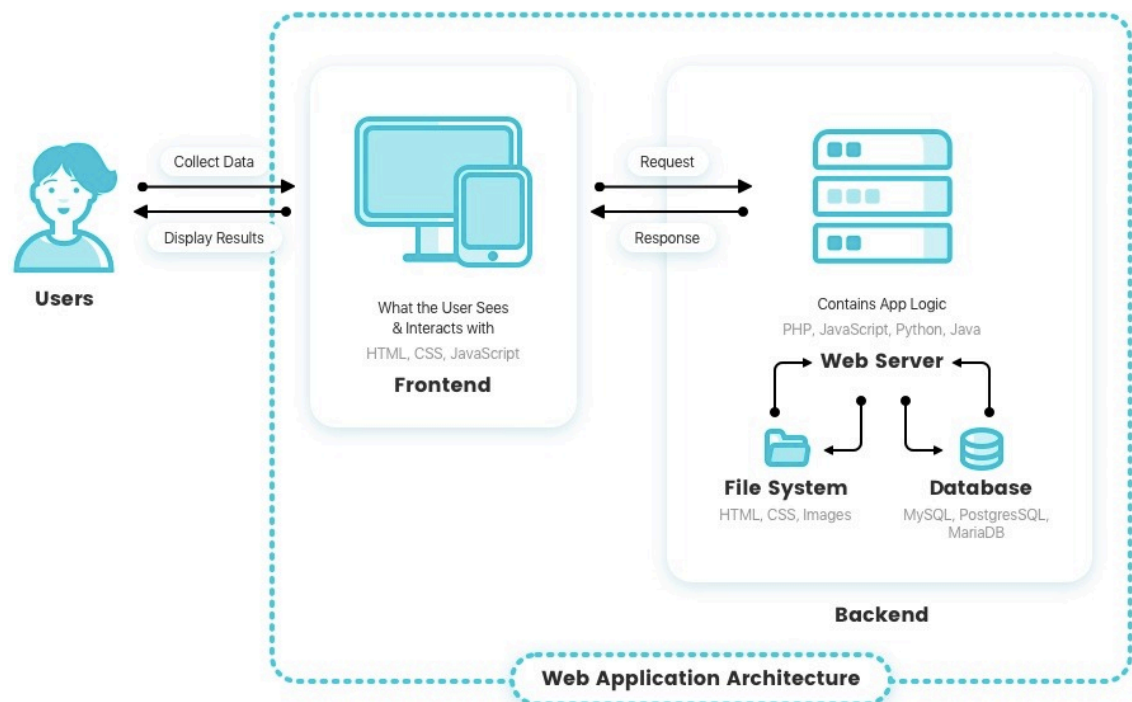


Figure 3, Web Application Architecture [11]

In a web application, the frontend is defined as the part of an application that the user interacts with. This part of the code is rendered on the browser; constructing everything that the user observes once using an application. In other words, the user interface of a web application is constructed into the frontend code, which is then rendered and displayed to the user through a web browser. As seen in Figure 3, the frontend, or the user interface of the application, collects the user inputs and also displays the results to the user. The arrows indicate the two-way information flow defined earlier in Subchapter 2.1. [11].

In addition to the frontend and client side code, a web application consists of the server side code or the backend code. The backend is responsible for the logic of the application, meaning that it takes in the inputs of the user and handles them as intended. The results, or the outputs, are then sent back to the frontend and displayed to the user in the browser. Furthermore, in most web applications the backend is connected to a database where all the information viable to the application is stored, fetched and edited [11]. As the topic of the thesis is user interface and its design, this study focuses only on the frontend part of a web application. Backend code and databases are only discussed if it is necessary given the current context.

A web-based e-commerce application follows the same basic architecture presented in Figure 1. Hence, it can be stated that the graphical user interfaces in web-based e-commerce applications are constructed in the frontend code. At this point, it should be pointed out that the frontend code can, in addition to a browser, run as a mobile application in a mobile device. The topic of the study being related to a traditional web application and specifically to the frontend of the application, the mobile applications are not discussed further.

The following chapter aims to discuss further the concept of user centricity in graphical user interface design specifically in the e-commerce context. The objective is to conduct a set of guidelines to user centric graphical user interface design that can be utilized in the user interface evaluation phase later in the study.

### 3.3 Graphical User Interface Factors in E-Commerce

In the graphical user interface of an e-commerce application<sup>1</sup>, the key is to make the shopping experience smooth and pleasant for the customer. Graphical user interface in this context is defined as all that is shown on a webpage for the customer. Since that definition is quite broad to begin with, a more detailed approach requires breaking the components into different parts; including the tools of an application, such as navigation, search bar and user guidance [12]. Hence, before discussing further the user centricity, the key components that construct a graphical user interface in e-commerce platforms should be defined. As the objective in this paper is to discuss and evaluate the main features of the graphical user interface of an application, the aim of this chapter is to define the most viable graphical user interface factors of an e-commerce application.

Before discussing the user centricity, the factors should first be identified. To identify the key factors of the graphical user interface, a framework of customer journey is utilized. Customer journey maps, over a period of time, the relationship that a customer, or in this context the user, has with a business or an application. The customer journey describes, as a process, what the customer goes through when interacting with the service provider, and what are the needs of the customer during this interaction. [43].

---

<sup>1</sup> The study focuses completely on graphical user interfaces in desktop environment

The customer journey as a concept is used across industries in a variety of contexts. Hence, the concept of customer journey should be investigated in the e-commerce environment. Customer journey in the e-commerce should, as stated above, identify the aspects that the user interacts with once using a web e-commerce application. Derived from these aspects a set of the most viable graphical user interface factors can then be identified.

A study by Mangiaracina and Brugnoli (2009) identified the customer journey in the e-commerce environment. The study argues that the customer journey in e-commerce sites can be split in to five main phases, which can be visualized as a process chart. The five phases are site landing, product discovery, product presentation, cart management, and checkout. [42].

To better serve the purposes of the thesis, only the first four phases of the framework of Mangiaracina and Brugnoli are applied: the final phase, checkout, was left out since according to Mangiaracina and Brugnoli it deals with shipping, shipment tracking and payment options, which are out of the scope of the present study. Hence, in the study, the customer journey in the graphical user interface context of an e-commerce application is defined as a four-staged process from site landing to product discovery, product presentation and finally to cart management. The conceptual framework for the study was constructed based on the customer journey framework by Mangiaracina and Brugnoli and it is presented in Figure 4.

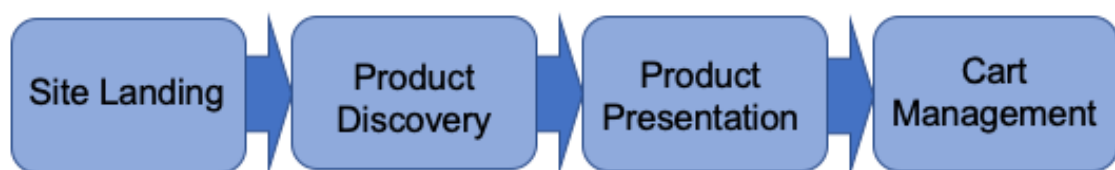


Figure 4, Conceptual Framework based on the Customer Journey Framework of Mangiaracina and Brugnoli (2009) [42].

In the following subchapter, a more detailed discussion of the graphical user interface factors, based on the customer journey, is presented. Each part of the journey is defined through a graphical user interface perspective in order to develop an understanding of how this customer journey is seen in the graphical user interface factors of an e-commerce application.

### 3.3.1 Site Landing

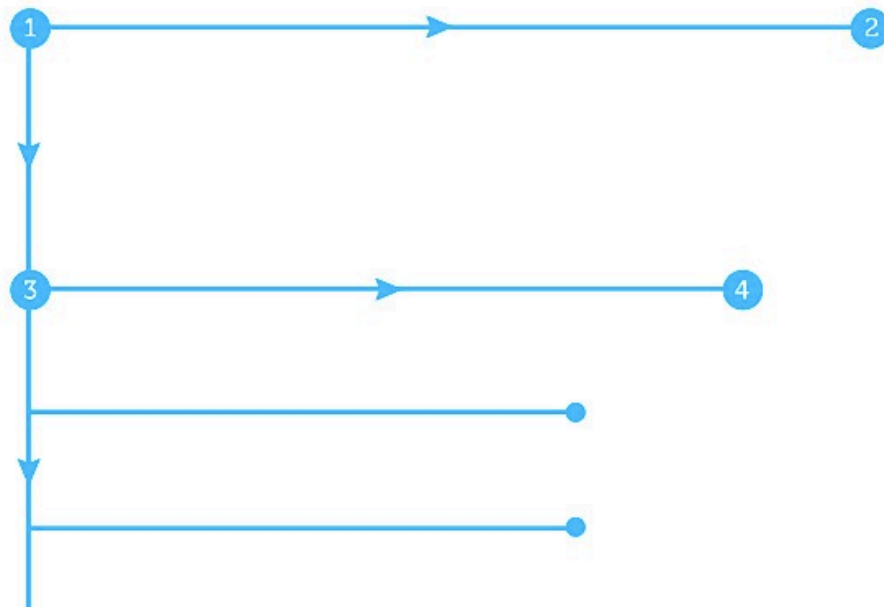
Derived from Figure 4, the first factor of an e-commerce graphical user interface is the site landing. In practice, the site landing represents the “home page” of an application. It is a particularly significant factor since it is the first thing that the user sees once opening an application, or in other words, when the user lands on the site. Hence, the site landing gives the first impression and engages the user to the site. It is crucial that all the other factors of an e-commerce graphical user interface are easily accessible from the home page and that the home page is interpretable: it should contain only the relevant content in a readable and understandable form. [13].

Assuming that the user is using the application for the first time, the first impression provided by the home page is an aspect that cannot be over-emphasized. Hence, in order to make the home page user centric, there should be an objective to aim for a combination of readability and learnability. Readability describes how the structure of the home page is executed and how the content and the other components are laid out in a manner that the end result is understandable. The home page should not be too crowded of different elements and it should contain only the necessities in a simple layout. Learnability means that the intended processes should be clear for the user. Elaborated as a process, when the user lands on the home page, it should be clear how the desired action can be executed. For instance, if the user is willing to search for a specific product, it should be clear how this should be done i.e. where to begin with. A viable option to provide learnability function is to give guidance in a form of graphic or verbal elements, such as arrows, text containers and/or coloring. [17].

The home page should be seen as a container that provides immediate access to the main features of the application. Any e-commerce application is dependent on its search and navigation functions; hence the navigation and search components should be placed clearly visible and easily accessible in the home page view. In addition to the navigation and search components, the home page should present the company name and logo in addition to other relevant information about the application such as versions, copyrights, etc. [18].

Considering the layout, it is important to utilize the knowledge of how people read and observe content. One of such approaches is the F-shaped pattern that is a commonly

accepted framework to illustrate how people and web users observe content. The model suggests that the user first observes the top left corner, where the eye moves right on the other side of the page. If nothing that catches the attention of the user is found, then the same pattern is repeated one level lower, generating a pattern resembling the letter F. This pattern is described in Figure 5. [19].



Our eyes are trained to start at the top-left corner, scan horizontally, then drop down to the next line and do the same until we find something of interest.

Figure 5, The F-Shaped Pattern [19].

This is an important implication for the home page design of an application. It provides a clear layout design and implies where the most important components should be placed; and in which order.

In conclusion, the home page should aim for minimalistic and not content-heavy design that guides the user to do the desired action. The necessary content should be laid out in an understandable way that follows our natural habit of observing content in web pages. This can be done by laying the components in F-Shaped pattern based on the importance of each component.

### 3.3.2 Product Discovery

Followed by the site landing, the very essence of an e-commerce application is product discovery, or the action of searching for products. Naturally, this key requirement of searching is not possible without an element that enables the action of searching. In e-commerce graphical user interfaces, the search is executed by using a search bar where the user fills in the search terms. In addition to the search bar, the search should enable the usage of fixed selections. For instance, if the user is looking for products that belong to a specific category, there should be a pre-made link to the products of that category: a user looking for disposable gloves should be given the opportunity to click an element that directs the user to catalogue containing all the disposable gloves. These two different search options are presented in Figure 6. [12,13,14,15,16].

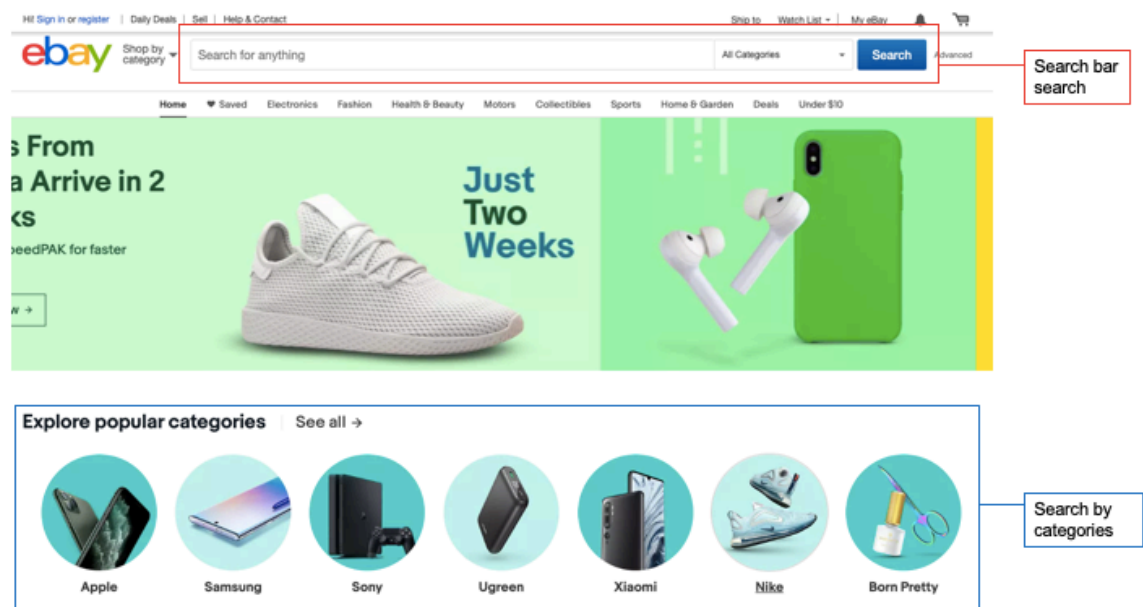


Figure 6, Product search examples in ebay.com

Searching for products is one of the key functions of any given e-commerce application. Search tools are an important part of the application and especially the search bar is usually a critical component.

Digging deeper into the phase of product discovery, the placement of the search bar should be easily visible and eye-catching. In addition, an encouraging element such as a “search here” label text inside the box should be added to add a call for action. The

search box should also be big enough to hold longer text queries. Adding a magnifying glass to the box provides a universal symbol indicating that the box is used for searching. Another important search feature is the categories and pre-made product listings that should be provided to the user. In order to construct the category selection in a user-centric manner, the naming should be clear, the links to the categories easily available and the number of categories and subcategories should be as low as possible. [21,22].

As the usage of an e-commerce application includes searching and browsing, there should be a component that helps the user to navigate between different elements. To begin with, one of the main features of product discovery is to offer the user a possibility to return to the home page at any point during the using session. This can be implemented by introducing a home page button or a navigation bar, which includes, in addition to the access to home page, access to the product categories. The important part is that the user knows how and where the returning to the home page can be triggered. In addition, one key feature of the product discovery is the general navigability of the application: the path to products should be simple and accessible with as few actions as possible. [17].

One specific feature to increase navigability in the product discovery phase in an e-commerce site is a visible path that is shown to the user throughout the search process. This provides the user information concerning the current position on the path as well as easy access back to the previous view; given that the names in the paths act as hyperlinks. The function provided by this is that if the product currently being viewed is not desired, there is an easy, visible, path to the other products in the same category. This element, called the breadcrumb trail, is illustrated in Figure 7. [20].

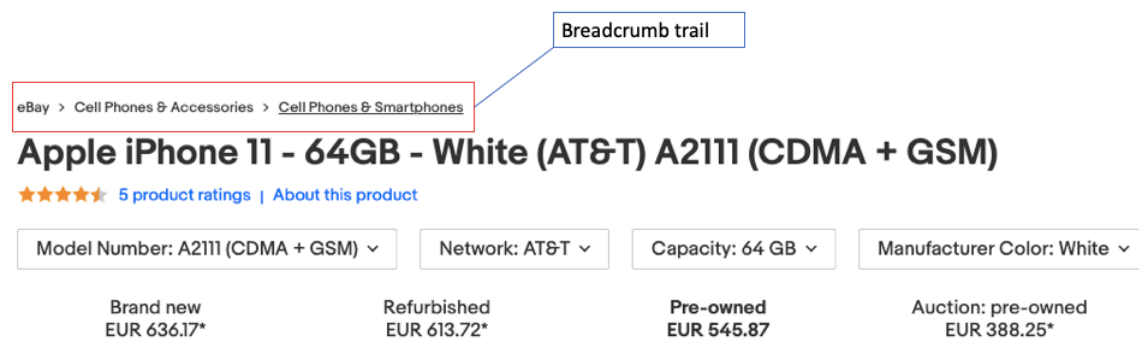


Figure 7, Breadcrumb trail in ebay.com

As seen in Figure 7, the positioning in the example is done by laying out the trail on top of the product view. Here, a following pattern can be seen: home page > category > subcategory. This can be called an attribution based trail, where the user can access the previous view or, in other words, deselect an attribute that filters the results. For instance, in the chosen example, the user can access the wider range of cell phones and accessories by clicking the text “Cell Phones & Accessories”. [21].

### 3.3.3 Product Presentation

As the search process completes, it is followed by the product presentation. In e-commerce, this element is referred to as product cards. Described as a process, after the searching, whether done through the search bar or category selection, a product or a set of products should be presented to the user. Hence, a product card in this context refers to the format in which an individual product is displayed to the user. The set of product cards, called the product listing, is a crucial part since that defines the layout of the product cards in the container reserved for the search results. In Figure 8, these two components are illustrated using an example from Walmart.com. [12,13,14,15,16].

The image shows a screenshot of the Walmart.com website's product listing for kitchen mixers. The page features a navigation bar at the top with a search bar and various icons. Below the navigation bar, there are several filter options on the left side, including 'Category', 'Delivery & Pickup', 'Special Offers', and 'Brand'. The main content area displays a grid of product cards. Each card includes an image of the product, a title, a price, and a star rating. A red box highlights the entire product listing area, and a blue box highlights a single product card. Labels 'Product listing' and 'Product card' point to these respective areas.

Figure 8, Product Card and Product Listing in walmart.com

The search results in an e-commerce application being individual products, the presentation of products, i.e. the search results, is crucial. In product card layout and design, the guidelines regarding readability described by Megharani and Madhuri (2016) should be applied. The amount of potential information to be put on the product card is vast, thus there should be a critical evaluation on which are the necessary components and details to be included in the product card. Since the product card acts as the preview and there are, in many cases, many product cards in the search results, the user should be able to receive the necessary data by glancing at the product card. According to Whitmore (2016), the product card should include a product name that fits inside the card, a short description of the product, an easily visible price, a good quality product photo and customer reviews. In addition, it is a viable option to add interactive functions to the product cards, such as a hovering effect which highlights the card when the user lays a cursor over it. [17,13].

Concerning how the product cards are laid out, there are essentially two main options to choose from: grid view and the list view. These options are illustrated in Figure 9 below; list view on the left and grid view on the right.

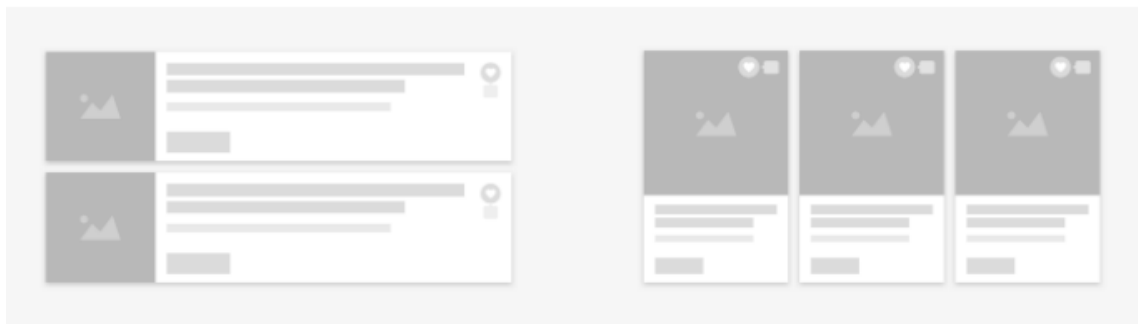


Figure 9, List view vs. Grid view [24]

There are clearly some differences between these two listing options. As seen in Figure 9, the grid view emphasizes the picture and the graphical elements, whereas the list view requires reading and content evaluation. There is no simple answer to which one is more user centric, and hence the selection on which view should be used is done based on the context: is the product selection done by evaluating the picture or is there other, more detailed, aspects that the user evaluates before the product selection. [24,25].

### 3.3.4 Cart Management

Upon arriving to the desired product, the user must confirm the selection that is going to be ordered; a brick and mortar shopping equivalent of this being the shopping cart and adding products into it. In the e-commerce context, this product selection is also referred to as cart management. Derived from the tangible shopping carts, the first important note related to e-commerce sites is that the cart, in addition to its content, should be continuously accessible throughout the shopping journey. At the least, a preview of the cart should be accessible without opening a new window; the user should be given the possibility to check the content without intervening the shopping process by opening a new window or a tab. Usually this can be done by having a navigational element labeled as “cart” or with a shopping cart icon, which can be universally recognized. [48].

In addition to the preview, there should be a separate window for the cart, which is accessed before the checkout and order confirmation. The purpose at this point is to provide the user the option to revise the content and check that the products and their attributes match the requirements of the user. To avoid errors, there should be a convenient method for the user to change the quantities and possibly remove the products from the cart. Furthermore, the product details should be visible in the cart and an easy access back to the product page should be present; for instance, the user should be able to navigate back to the product page by clicking the product in the cart. [49,50].

A valid guideline for cart management is to first provide a preview option for the cart as well as a detailed cart page which includes the product details before confirming the order. For modifications, the user should be able to change the quantities or remove the product from the cart, in addition to the possibility to return to the product page by clicking the product icon in the cart.

## 3.4 Summary

To summarize, based on the references used and the constructed conceptual framework, it is possible to define the minimum viable graphical user interface factors related to an e-commerce application. These can be described as follows:

- 1) Site Landing
- 2) Product Discovery
- 3) Product Presentation
- 4) Cart Management

When it comes to the user centric aspects of the presented components, it should first be noted that the general understandability and readability concerns all of the components. The home page design should provide access to the necessary elements, which are the category based explorative search and the search bar. The layout should follow the F-Shaped pattern and highlight the necessary components. Moreover, there should be some feature or element that guides the user on how and where different actions can be executed, such as text or graphics emphasized with coloring.

Concerning product discovery, the search bar should be easily visible and it should catch the attention of the user on the home page. In order to communicate the function of the search bar, a magnifying glass and text label, such as “Search here”, should be added. For customers wishing to explore through categories, there should be an easy access to categories and potential subcategories which show the complete product lists. However, the number of categories should be carefully selected since too many choices can end up being confusing for the user.

The product discovery should be facilitated with different navigational factors. The navigational bar, where the product categories are placed, needs to be easily accessible and readable and it must not contain too many elements. A key feature of product discovery, especially in situations of user errors, is to provide an easy access to the home page regardless of the current position of the user on the application. Since the nature of an e-commerce application focuses on search activities, there should be an additional navigational element, defined as breadcrumb trail, that visualizes the path between search and product pages. If possible, the breadcrumb trail should provide hyperlinks guiding back to the previous view and/or home page.

In reference to product presentation and product card design, readability should be considered. The product card should contain necessary elements in a manner that is easily interpreted by the user. In most cases at the least a picture, product name, short description and potentially customer reviews should be added to the product card. To make the browsing experience interactive, a hovering effect or similar element could be added to provoke the product selection. Concerning the listing of the product cards, there should be a separate evaluation based on the products included in the application: if the products are selected by using more detailed information, a list view should be used and, if the products are selected based more on the appearance in the pictures, a grid view could be a more viable option.

For the cart management, the user should be given a possibility to check the cart content throughout the shopping without opening a separate window. In addition, the actual cart view prior to checkout and order confirmation should provide enough details for the user to evaluate whether the content is as wanted. Fixing any errors in the product selection, the cart should provide the chance for modifications in quantities, removal of unwanted products and easy access back to the product page for instance by clicking the product in the cart.

It should be acknowledged that these components do not take into account the graphical user interface of an application in its entirety and within the scope of the thesis not all possible elements can be discussed. Based on literature and multiple visits to e-commerce sites, there are numerous other elements included in an e-commerce application on top of the four components described above. Furthermore, in all of the four phases the optimal layout and functionalities depend on the context and the target audience; there is likely to be a vast difference in the product presentation requirements of a fast fashion consumer ordering a new dress compared to a health care employee ordering critical surgical supplies. A more detailed discussion on the future development, alongside with other key components of graphical user interface, is presented in Chapter 7.

## 4 Open Source E-Commerce Platforms

In the introduction, a set of research objectives were defined. In the previous chapter, a framework on graphical user interface factors was identified and, in addition, a discussion was done on how to construct the factors in a user centric manner. The next step is to identify the basis for the application. Hence, the objective of this chapter is to answer the second research question and evaluate different open source e-commerce platforms. As the future objectives of the author includes the commercialization of the application, there should be a proper analysis on which open source platform should be used as the basis for the application. In other words, this chapter presents an in-depth evaluation process on which open source e-commerce platform should be chosen as the basis of this project. Based on the evaluation, the selection of the most suitable platform is presented.

### 4.1 Open Source Software

Everyday users of software rarely, or never, see the actual lines of code, otherwise called the source code. Traditionally, the source code is only accessible for the company or a development team that is responsible for the programming work. This type of software is defined as proprietary software which cannot be edited or distributed without a permission of the owner of the software. For instance, the Microsoft Office software family are proprietary software, owned by Microsoft. [26].

On the other hand, open source software, as the name suggests, is open and accessible for the public. Not only is the open source accessible by the public, but it can be edited, improved and re-distributed by anyone. A case in point, LibreOffice is an open source software: the application as well as the source code are public. [26].

From a point of view of a developer, there are a variety of reasons why open source software is used for a project. Perhaps one of the most fundamental reasons to choose an open source project is the cost. Given the context of the present study, the alternative to using an open source e-commerce platform is to develop an e-commerce application from start to finish, which would be labor intensive, inefficient and costly compared to utilizing an open source software. Other crucial aspects to using an open source software from the developer viewpoint are the control and customization: the developer has the

control over the software and can make customizations and alternations to the application. [26,27].

Considering the objective of the thesis, the above reasons were the basis of the evaluation when deciding whether to use an open source as the basis for the project. As mentioned, the application under development utilizes a web-based e-commerce application architecture. Hence, it would be cost and labor intensive to develop the entire application from start to finish. In other words, by utilizing an open source platform one avoids having to re-invent the wheel. On the other hand, since the application is intended to be used in public sector procurement, it is evident that some alterations have to be made to the selected open software platform. This is the second reason why an open software was chosen to be used: an open source can be freely customized to match the specific requirements.

In conclusion, the use of an open source software is efficient in this specific context. Time and labor resources are saved when the fundamental architecture and application logic are already completed, free of charge. Furthermore, an open source software does not limit changes to the source code. In other words, an open source software in this context includes the basic e-commerce application logic but alterations are implementable without restrictions. This is an important aspect when considering the future development of the platform.

## 4.2 Open Source License Types

Despite the fact that open source software is generally available for developers, there is, however, a variety of licenses that affect how open source software should be treated. In particular the context and business purpose of the application should be considered when evaluating the licenses. The purpose of this subchapter is to inspect the different aspects behind the most common open source software licenses. Finally, a decision is presented on which license would suit the objective of the project, i.e. under which license type the selected open source platform should be.

There are two main categories under which the individual and more specific license types fall and these categories should be carefully evaluated before taking advantage of any

open source software. The two license types are defined as copyleft and permissive licenses. They share similarities when it comes to the copying, altering and distribution of the software and hence, according to Byfield (2015) “to this extent, both are considered free licenses”. The difference is that these two licenses define the license that the altered software is released as: a permissive license type allows the developer to add their own copyright statement, whereas the copyleft license restrict the license type to the same one that the original open source software had. The relation of the permissive and copyleft licenses to the individual open source licenses is illustrated in Figure 10. [28].

### Two types of OSS Licenses

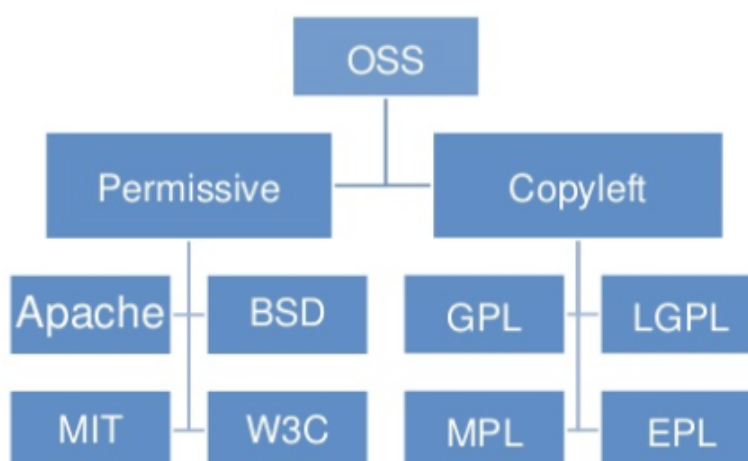


Figure 10, Two types of Open Source Licenses [29]

As Figure 10 indicates, the permissive and copyleft types act as categories for the individual licenses, such as Apache and GPL. With the help of the chart above, the difference between copyleft and permissive types can be elaborated further. For instance, the GPL, or General Public License, falls under the copyleft license. The practical demonstration of this is that if an open source software under that license is modified, the end result should be distributed under the same GPL license. On the contrary, when a permissive open source software, such as Apache, is modified, the developer has an option to alter the license as well. In other words, the developer can add copyright statements that the original license did not had. [28].

Derived from this, it can be deduced that the open source platform chosen should fall under the permissive license type; as the aim is to develop a platform that could be commercialized in the future, a possibility to add copyright statements should be kept open. As this option is not possible among the copyleft open source software, the next phase is to evaluate permissive open source software in depth.

As already seen in Figure 10, there are numerous different license types that are permissive. Hence, there should be an in-depth analysis of the permissive open source licenses. Due to the large number of licenses, a short list is constructed by evaluating the most popular licenses among all open source software. They are highlighted in Figure 11.

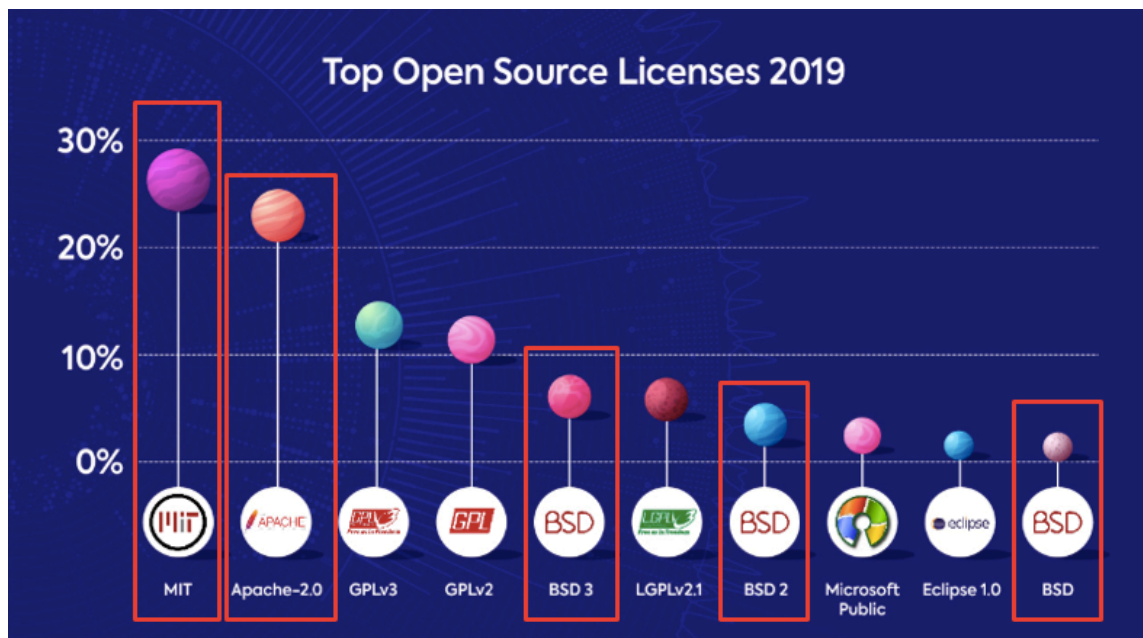


Figure 11, Top Open Source Licenses 2019 [31]

In Figure 11, the permissive license types are indicated with the red frames. As seen, a total of five permissive licenses appeared in the top ten most used open source licenses in 2019. However, the BSD license appears three times with its different clause types. Since clause 3 is the newest and most used clause, the final short list of license types consists of the top three permissive license types, including MIT, Apache-2.0 and BSD 3 licenses. [31,32].

### 4.3 Comparison of Selected Open Source License Types

Derived from the discussion above, a short comparison between the three license types is introduced in this subchapter. Each type is discussed individually, after which a short conclusion on which type of license the used open source platform should have is introduced considering the context of the study and the purpose of the application.

**MIT:** The MIT license, originating from the Massachusetts Institute of Technology, is a permissive license notable for its permissive nature. The license is distributed under a disclaimer which explains that the original developers are not to be held responsible for any claims. On the other hand, the MIT is one of the most permissive licenses and it does not restrict the commercial use of the secondary product. Provided that the original source is appropriately acknowledged in the secondary software, the MIT license is extremely simple to understand and it does not restrict the use of the software; turning it into a proprietary software is permitted. [32,33,34].

**Apache 2.0:** For most parts, the Apache license resembles the MIT license, but it discusses the legal aspect more in-depth. Technically, the license does not prohibit commercial use of the software, or making it proprietary, as long as the original patent, copyright and trademark notices are not breached. [32,33,34].

**BSD 3:** The BSD license, originating from the Berkeley Software Distribution, again allows free usage of the code, similar to both MIT and Apache licenses. However, the BSD clause 3 has a specific non-endorsement clause that restricts the usage of the name of the original creator for product promotion. Despite this, as stated, the BSD 3 license is an extremely permissible license allowing practically any re-usage of the code as long as the original disclaimer and copyrights are stated. [32,33,34].

In conclusion, according to Anar (2017), there is no great difference between the top three permissive open source software licenses. Especially when considering these licenses for the purposes of commercializing the software, technically all three licenses

allow this procedure. A comparison table between these license types is provided in Figure 12.

License	Code Use, Modification & Distribution	User Obligations	Patent License Grant	Linking code to other licenses	Trademark Grants
Apache 2.0	Permissive. Must notify users of code modifications.	Can't remove copyright, patent, trademark and attribution notices	Yes: Explicitly defined	Permissive	Not allowed
MIT	Permissive	Future licensed software must contains a copy of license and copyright notice	Not explicitly defined	Permissive	Not mentioned. Advance permission required
BSD	Permissive	BSD 3-clause: avoid appearance that product is endorsed by the original developers	Not explicitly defined*	Permissive	Not mentioned

(\*) Facebook's modified BSD+Patents license contains a specific, restrictive patent section.

Figure 12, MIT, Apache and BSD Compared [34].

Figure 12 supports the analysis of the three license types. The main aspects of each license type are relatively similar to each other and for a developer, any of the three licenses provide a non-restrictive usage of the code. However, there is an aspect of the Apache 2.0 that should be taken into consideration; there is an obligation in Apache 2.0 to add prominent notifications to changes made to the code which, according to Anar (2017), can act as a limitation for developers.

Due to the obligation in the Apache 2.0 license, a decision was made to evaluate open source software which possess a BSD-3 or MIT licenses. As the limitation in the Apache 2.0 might lead to an increased workload in the future development, the open source software that is to be chosen as the basis of the application should have a MIT or BSD-3 license. In the following subchapter, the final selection on the open source platform is presented.

#### 4.4 Selection of Most Appropriate Platform

As discussed, given the context of the study and the future development, an MIT or a BSD-3 license would seem the most appropriate. However, the license type itself should not be the only attribute under evaluation. According to an evaluation by Wheeler (2011),

the evaluation process should take into account the business needs of the desired end-product as well as how these functionalities can be added. In practical terms, first it should be identified whether the open source platform can meet the business requirements and secondly, how the desired functionalities that the developer or development team has, can be implemented. [35]

In addition to the two aspects of adding functionality and matching the requirements, the star count on the projects GitHub repository was used as an evaluation criterion. The star count was used since, according to study by Borges & Valente (2018), the stars given to an open source project reflects the appreciation of the community as well as the popularity of the project. In other words, the star count is an indicator of the approval of the developer community towards a project, hence it is useful when constructing a shortlist of the open source platforms [36].

A number of listings of open source e-commerce platforms were evaluated based on the attributes described in the previous paragraphs. The most popular e-commerce platforms utilized PHP as the programming language, which conflicts with the adding of functionalities. This is due to the fact that the author does not have competence in PHP and hence these platforms are not legitimate candidates. As the author has experience mainly in React and JavaScript, the evaluation was focused on open source platforms that utilize these technologies. Another impracticality of the evaluated open source platforms, in addition to the used programming languages, was the fact that many of the platforms were actually only the frontend executions. This means that in order to develop a full presentable application, a backend integration should be done.

However, further investigation of the open source listings unveiled a platform called Cezerin, that has a React based frontend and a Node.js backend. In addition, the platform has a ready MongoDB database integration. The Cezerin platform has a complete infrastructure which means that in order to demo the application, no further database or backend integrations would be necessary.

When the open source software was investigated, Cezerin was practically the only viable option considering the evaluation criteria presented above. A short comparison to each criterion is presented below:

- 1) **License type:** Cezerin is released under the MIT licence, which means that it can be copied, modified and commercialized free of charge as long as the original copyright statements are visible. In addition, the license type allows the end product to be released as a proprietary application with no obligations to re-distribute the source code.
- 2) **Business needs:** As stated, Cezerin is released with backend and database integration, hence there is already an infrastructure for adding products and categories. This will enable the client demos later in the future, so the platform meets the business needs.
- 3) **Adding functionalities:** The technologies, throughout the frontend, backend, and database, are familiar to the author. The familiarity of the technologies is the reason why Cezerin is considered to be a platform where the planned functionalities are possible to be constructed. In addition, the technologies are applicable to the modern-day programming world, hence the functionality adding should be conceivable in a situation where the development process is possibly outsourced in the future.
- 4) **GitHub stars:** Cezerin project is awarded a total of 1,889 stars on GitHub. Based on this, there seems to be approval for the Cezerin open source software provided by the developer community.

Based on the above evaluation, Cezerin met the requirements of each criterion for open source platform selection, hence Cezerin was selected as the basis for the project.

#### 4.5 Selected Platform and Project Setup

Prior to the user interface evaluation, in this subchapter the used platform and its technologies are further introduced. As mentioned above, Cezerin is distributed as a complete platform that includes both frontend and backend as well as an integration to a database. In most parts, the architecture of Cezerin follows the basic pattern of a web application presented in Figure 3. However, the Cezerin frontend is divided into two

parts: the customer facing frontend and the admin facing frontend. The complete architecture with the technologies is presented below in Figure 13. The technologies are written in italics under each element.

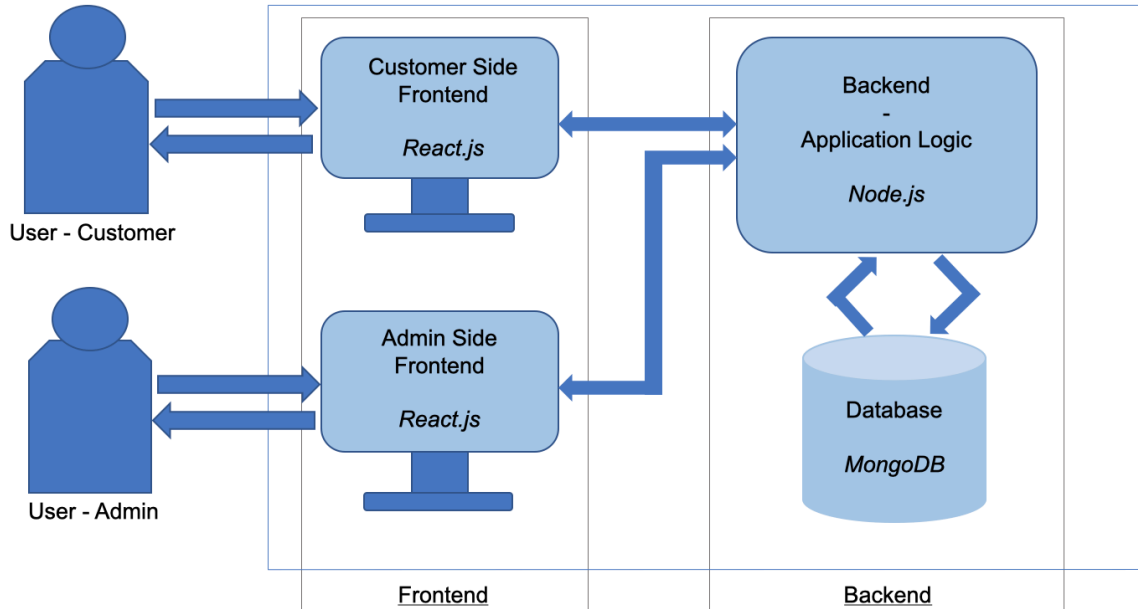


Figure 13, Cezerin Architecture

As seen in Figure 13, two frontend types are identified. First, there is the admin side frontend. Through this separate frontend, the administrator, or in this case the service provider, can add, delete and modify the products and product categories. These inputs are then transferred to backend and into the database. The customer side frontend is where the products are then searched, viewed and ordered. Customer inputs are transferred then to the backend which is integrated to a database. Based on Figure 13 above, there are a total of three main technologies that Cezerin utilizes and the following paragraphs gives an introduction to each technology.

**MongoDB:** Majority of software applications require a database where data is stored. In an e-commerce environment, viable data exists related to products, stocks, and user credentials. In the case of Cezerin, the database is MongoDB which is the leading NoSQL database. NoSQL mean that the data is not stored in relational tables, but rather as documents. As there is no relational structure in MongoDB, the data is “nested within a single data structure.” In addition to the structure of the database, the strengths of MongoDB are the JSON format that the data is stored in and its scalability for large amount of data. [37].

**Node.js:** Node.js, as the latter part of the name indicates, is a JavaScript based runtime environment. In practice, Node.js allows the backend, or the server side, code to be written in JavaScript. As Node.js utilizes the V8 engine, which is used outside the browser, the environment is considered to be very performant. [38].

**React.js:** React.js is a JavaScript based library developed for user interface development. React is based on the idea that complex interfaces are constructed by using smaller elements which are, in React, declared as components. React nature is declarative, where the programmer declares the different components and the actual implementation occurs in React. [39].

#### 4.6 Project Setup

In this subchapter, the process of the project setup is elaborated by describing the necessary actions that had to be taken prior to the actual modifications to the code. The objective was to ensure that the open source platform runs locally without any errors. In addition, a few mock-products were inserted in to the database. The project setup and all possible modification source code were executed in a Macintosh operating system environment with macOS Mojave version 10.14.6.

Prior to the actual Cezerin installation, the necessary infrastructure has to be put in place. As the project is based on Node, React, and MongoDB, a Node Package Manager, or NPM, PM2 daemon process manager, Node, and MongoDB were required to run the project.

They are conveniently installed in a macOS environment by utilizing Homebrew, which is an open source package management system for macOS [40]. First, the Homebrew had to be installed, which could be done through the terminal by following the instructions provided at the Homebrew website at <https://brew.sh>.

Once homebrew was installed, the Homebrew commands were accessible. Node was installed simply by running “brew install node” command on the terminal. As Node installation includes the NPM as well, there was no need for a separate NPM installation. Securing the installation, “npm -v” and “node -v” commands were inserted, which

resulted showing the version numbers 6.13.7 and v13.10.1, indicating that both were installed successfully.

The same logic follows with MongoDB. Terminal command “brew install mongodb-community” starts the MongoDB installation. After the installation, two separate terminal windows were opened. To these windows “mongod” and “mongo” commands were inserted; exactly in that order. “Mongod” started the mongo daemon, and “mongo” the MongoDB control panel. This proved that both installations were successful.

As the installations were done, the necessary infrastructure for Cezerin was installed. Subsequently, the actual Cezerin project was built.

Reflecting back to Figure 10, Cezerin is constructed of three different elements. Before the installation and project setup, it should be understood that there exist, in fact, three separate applications that run simultaneously. In other words, in order to evaluate and modify the customer side frontend, the complete Cezerin project has to be installed properly.

The setup phase was initiated by accessing cezerin.org, where the developer manual is available. The setup followed the installation manual provided by the Cezerin development team. [41].

Logically, the first step was to create a local desktop folder for the actual files. This was named simply Cezerin. After the folder was created, it was accessed in the terminal with command “cd Desktop/Cezerin”. As the Cezerin is hosted in three separate repositories at Github, they were imported by “git clone <repository url>” commands while in the Desktop/Cezerin folder. This resulted three separate folders containing the files for the customer side frontend, admin side frontend and backend. The file structure of the project after cloning is visualized in Figure 14.

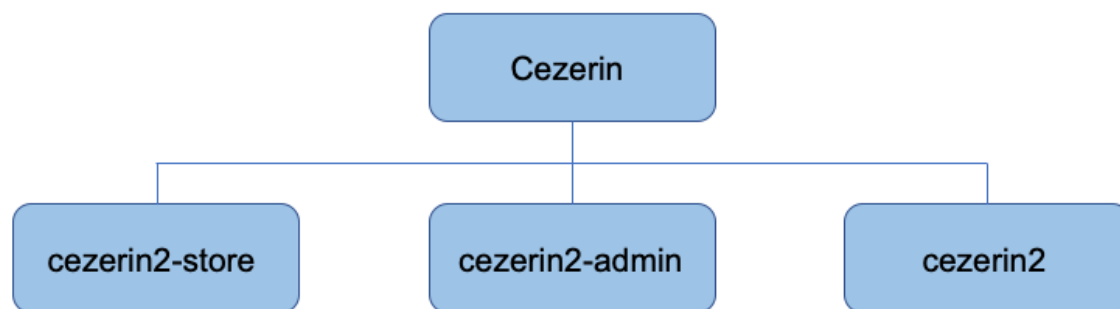


Figure 14, Cezerin Project Folder Structure

To elaborate the file contents, the cezerin2-store contains the customer facing frontend and the cezerin2-admin contains the admin side frontend. The cezerin2 file contains the backend code.

Since the software is technically ready-to-deploy, no modifications were needed to the backend side besides the database connection. This was accessed by opening the cezerin2 config folder, where the server.js file is located. At the top of the file, the database variables are defined with const-declarations. The default settings establish a connection to a MongoDB localhost at address 127.0.0.1 and port 27017. These are the default settings for MongoDB localhost, hence no modifications were needed and with the default settings a database connection could be established. The database name is defined as “shop”. As discussed, MongoDB does not require any database initialization and Cezerin backend connects to the MongoDB localhost when the application is started.

Following the cloning, all three files were accessed in the terminal in three separate terminal windows. Before the application can be started, node modules are required to each folder. This was done by using the Node Package Manager commands. Once inside the folder a “npm install” command was given, which installed the node modules to the project folders. This was done separately in each three folders.

All necessary actions were hereby executed and the project setup was complete. The application was then launched with the following commands in the specific order, each in their own terminal window:

- 1) In a terminal window, MongoDB daemon was started with “mongod” command

- 2) In a terminal window, MongoDB was started with “mongod” command
- 3) In a terminal window, while accessing the folder at Desktop/Cezerin/cezerin2, the backend was started with “npm start” command
- 4) In a terminal window, while accessing the folder at Desktop/Cezerin/cezerin2-store, the customer side frontend was started with “npm start” command
- 5) In a terminal window, while accessing the folder at Desktop/Cezerin/cezerin2-admin, the admin side frontend was started with “pm2 start process.json” command

After these steps, the application was running and the customer side-frontend were accessed at address <http://localhost:3000> and the admin-side frontend at address <http://localhost:3002>. No errors were shown in the terminal nor in the browser console, indicating that the application is running successfully.

As part of the setup, a set of mock-up products were added to the store stock. The products were picked to match the products in the existing procurement platform of the health care district and they are as follows: latex glove, nitrile glove, hand sanitizer, face mask. This was done in order to help the evaluation and implementation of the graphical user interface factors. These products were added in the admin side frontend and they were allocated to three different product categories. In addition, the default logo was changed to the logo of the company which the author works for.

## 5 Methodology and Research Approach

The empirical focus in the study is on the evaluation of the graphical user interface of the selected open source platform. Hence, in order to meet the research objectives, data had to be gathered in the evaluation phase of the study. The data gathering, or research methods, can be divided into two different types which are, according to Koskinen et al.

(2005), qualitative and quantitative methods. To elaborate the methods, Eriksson & Kovalainen (2008) argue that a quantitative approach has a focus on numeral data, testing hypotheses, and statistics, whereas the qualitative method aims to create a holistic understanding of a specific, or perhaps an abstract, phenomenon.

Since the final objective of this study aimed to evaluate the usability of the graphical user interface factors, the emphasis was naturally on the word evaluate. In this context, a qualitative approach enabled a more in-depth understanding on the usability of the graphical user interface. In addition, the qualitative approach helped to construct a perception on the possible points of improvement that would increase the usability. In other words, through a qualitative research approach, a conclusion could be made on how an e-commerce platform would, from a user interface perspective, suit public sector procurements; and if not, which modifications would improve the suitability.

In practice, the qualitative methods are, in most cases, interviews. However, there are methods in existence specifically tailored for user interface evaluation. Jeffries et al. (1991) have identified four different methods that are used to evaluate user interfaces: heuristic evaluation, usability testing, guidelines and cognitive walkthrough. These are presented in Figure 15.

	Advantages	Disadvantages
Heuristic evaluation	Identifies many more problems Identifies more serious problems Low cost	Requires UI expertise Requires several evaluators
Usability testing	Identifies serious and recurring problems Avoids low-priority problems	Requires UI expertise High cost Misses consistency problems
Guidelines	Identifies recurring and general problems Can be used by software developers	Misses some severe problems
Cognitive Walk-through	Helps define users' goals and assumptions Can be used by software developers	Needs task definition methodology Tedious Misses general and recurring problems

Figure 15, Summary of User Interface Evaluation Techniques by Jeffries et al. (1991)

To identify the most suitable user interface evaluation technique for the study, each method was reflected to the objectives of the study. According to Jeffries et al., the three techniques of cognitive walkthrough, guidelines and heuristic evaluation are usually conducted by developers and/or user interface specialists. As the objective of the study was to find how the user interface meets the requirements of an end-user in the procurement use, the usability testing method appeared to be the most suitable option. To further elaborate, Jeffries et al. (1991) describe usability testing technique as a method in which “the interface is studied under real-world or controlled conditions, with evaluators gathering data on problems that arise during its use” (p.119). In this technique, a selected test user is given different tasks that help to understand, according to Jeffries et al., “how well the situated interface supports the users’ work environment” (p.119). There is, in addition to assigning the tasks, a possibility to ask the selected test user questions related to the tasks and the user interface.

Based on the above, the usability testing appeared to be the best option for the evaluation technique. In order to legitimately utilize this method, the test user to be chosen must possess knowledge of the goals and objectives that are aimed to be accomplished by using the interface. In this context, the test user should therefore know how the procurements in public sector are done and which user interface factors are necessary for doing the procurements. This way it is possible to understand, as being said, that how well the user interface would support the day-to-day tasks and procurement work environment of the test user.

The test user selected for the usability evaluation is an employee of one of the Finnish Health Care Districts. The test user has vast experience with varying procurement applications used in the public sector, specifically in the health care context. The test user was therefore able to provide valuable reflections and comparisons between the e-commerce platform and the existing procurement applications in the health care sector.

After the test user was selected, the test user was, following the usability testing technique, given tasks based on the e-commerce customer journey framework. After the tasks, the user provided feedback on how well the user interface enabled these tasks, reflecting to the requirements that the procurement processes have in their organization. The tasks were constructed based on the conceptual framework of the study, where the

customer journey through the application was divided into five factors (of which four were chosen for this particular thesis). For each factor, a set of tasks was done and in addition, a set of questions were asked throughout the testing session. The questions related to the user interface factors described in Chapter 3, and the aim was to identify whether the user interface meets the requirements of the test user and if not, what should be improved. The tasks and the questions are found in Appendix 1. It should be acknowledged that in this context the tasks were quite detailed and not on an abstract level, which is usually the case in usability evaluation method. The tasks were structured based on the customer journey process and as the goal was to construct a detailed understanding based on the customer journey, the tasks were more detailed as well.

In conclusion, in the study the evaluation of the user interface was done by using a usability testing method including elements of an interview. The user was given tasks based on the conceptual framework and while the user completed these tasks, questions were asked on how well the user interface under evaluation met the requirements of the user. Through this, an analysis on how well an e-commerce platform, in the user interface perspective, could be used for public sector procurement purposes and if not, what would be the necessary modifications to the existing user interface.

## **6 User Interface Evaluation of Selected Open Source Application**

As the methodological approach was presented in the previous chapter, the factors described in Chapter 3 are, in this chapter, evaluated through the research approach described above. The objective of this chapter is to derive from the crucial factors of an e-commerce application and evaluate how well these transmit to the procurement needs of the potential client, or in this context the test user, and are they present in the graphical user interface of the selected open source platform. In other words, in this chapter, the results of the usability testing are discussed through a usability evaluation, reflecting to the user centric design in the user interfaces in e-commerce platforms.

### **6.1 Graphical User Interface Factors of Selected Open Source Platform**

The following subchapters cover each user interface factor based on the customer

journey framework by Mangiaracina and Brugnoli. By using the usability testing method, the test user was guided through each step of the customer journey. In each phase of the customer journey the test user was given tasks and asked questions related to the user interface and how well it met the requirements of the test user. The empirical results of the usability testing are discussed in the subchapters below, following the four phases of the customer journey, i.e. the theoretical framework of the study.

### 6.1.1 Site Landing

The site landing was evaluated by observing the home page of the application once it was running at localhost. The home page view that was rendered on the browser demonstrates what the client would see once opening the application. The home page view is illustrated in Figure 16.

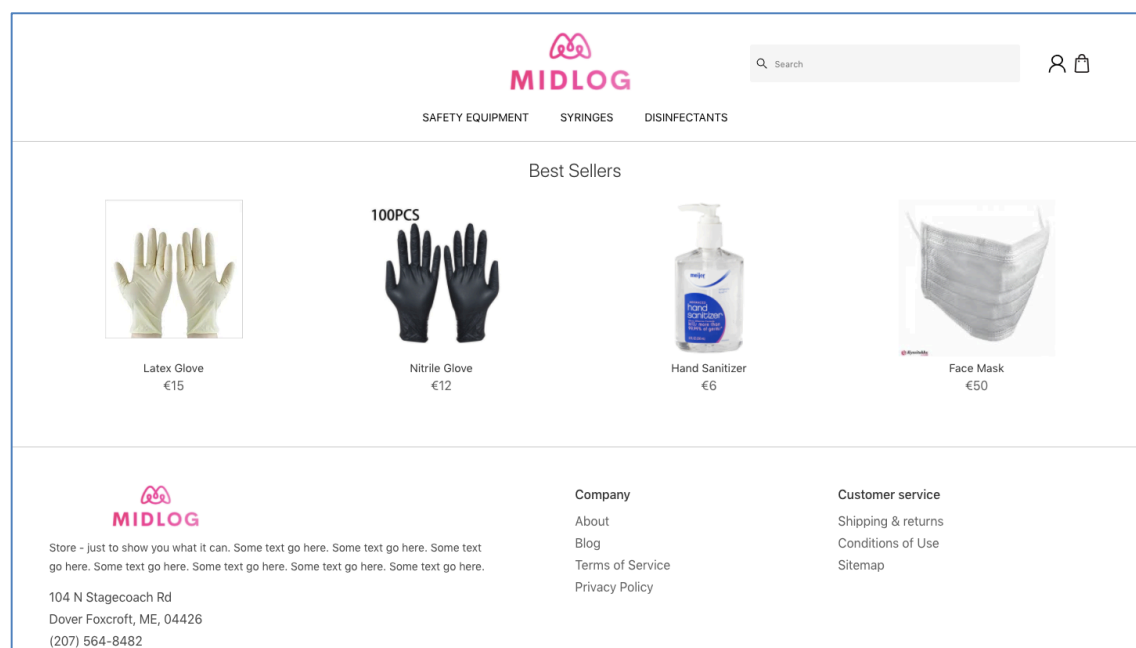


Figure 16, Home Page View of the Application

Once the test user opened the home page, a task was given to locate the product categories and the search box. The user found the search box immediately, which supports the finding that the F-shaped layout is a powerful way of catching the attention of the user. As seen in Figure 16, the search box is located on the top container element, on the right after the company logo.

In addition, the category selection was spotted by the user after the search box, which further supports the F-shaped layout. The navigation bar including the category selection is located under the search box representing the second horizontal line of the F-shape model. Laying out the search elements by utilizing this model clearly provided the user with immediate access to the tools and based on the tasks in this testing, no further guidance was needed in order for the test user to find the search tools.

However, it became apparent in the interview that from the test user viewpoint, in the procurement context, the site landing should still, despite the convenient placement of the search tools, aim for a simpler design. In other words, the user emphasized that in the procurement setting, all the unnecessary marketing elements should be removed. As seen in Figure 16, there is a component in the first page that guides the user to the best-selling products. This element was, based on the test setting, unnecessary and even somewhat confusing. When inquiring about a possible alternative, the test user suggested that the part in the layout which now contains the best sellers could be replaced with category boxes, similar to the example provided in Figure 6 in Chapter 3. This approach also supports the theoretical findings that the content in site landing should be kept in a minimum and instead, the marketing content should be replaced by components that guide the user: as in this setting the best-sellers section should be replaced by a category selection with images.

The site landing, or the home page, evaluation proved that the layout in an e-commerce application follows the generally approved design guidelines of web-development. In addition, these features seem transmittable to the procurement setting and practically the only necessary identified modification need was to replace the best-selling section with category selection. For this modification, the example presented in Figure 6 could be followed.

### 6.1.2 Product Discovery

After the site landing, the user was given tasks related to the product discovery. First, the user was asked to search for a specific product through the text box search. The product was found with the first try and the product page was opened successfully by clicking the product card. Then the user was asked about the most convenient way to

locate the other products in the same category. At this point, the test user chose the previous page button of the browser rather than any component in the user interface. This was chosen due to “learned habits” in web browsing.

After the user was asked about the breadcrumb trail, it became evident that this was left unnoticed due to its faded coloring and small size. This is illustrated in Figure 17.

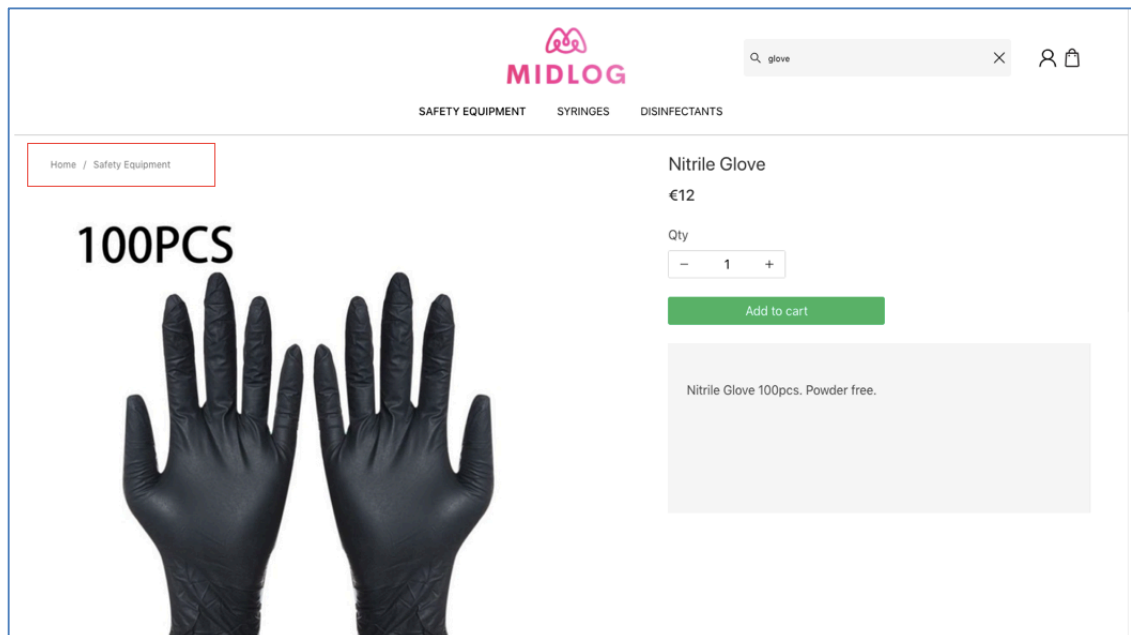


Figure 17, Breadcrumb Trail of the Application

When discussed further, it turned out that the breadcrumb trail would be a particularly convenient component for procurement purposes in the health care sector. This is because the products that are ordered, take gloves for instance, have a vast number of attributes and it is crucial to be able to scan between these attributes. A case in point, when an order has to be made for powdered and non-powdered gloves, the user could easily switch between these categories through the breadcrumb trail. The breadcrumb was also a component that does not have an equivalent in the current systems that the test user uses for procurements.

The user was then asked to search other products through the category navigation. This task was carried out successfully and there were no errors or moments of doubt in the interaction with the user interface. On the other hand, in the category section the test user pointed out that the filtering tools should not contain the price information. This is

due to the fact that the prices are already agreed upon with the supplier once added to the application. Hence, the price information is not, in this context of procurement in the health care sector, necessary for the platform user. This proved the notion presented in Subchapter 3.4. that the factors of graphical user interface are always context-specific and the relative importance of the product information varies between different industries and users.

As the theoretical framework suggested, the product discovery phase should be facilitated with navigational elements, hence the final task for the test user was to navigate back to the home page of the application. The user clicked the company logo on top of the page, which opened the home page. This proved that considering navigation, a home page button is not necessarily mandatory since it seems that clicking the company logo is considered as a path to the home page.

Considering the product discovery, the search tools of an e-commerce application again seems to be well-suited for the procurement needs in the health care context. The text-based search is convenient but especially the category-based search and the breadcrumb trail visualization were valued features by the test user. When the differences to the current procurement systems were discussed, it became evident that they do not have a same type of visualization and agile search tools: the search is rather text based and there is no efficient way for moving between different product categories. In this sense, the e-commerce factors in product discovery, could, in fact, could make the procurement processes more efficient.

### 6.1.3 Product Presentation

The product presentation phase was evaluated with a task to first utilize the search components and then figure out how many gloves are in one package; testing the ability of the user to investigate the product page information. The product page proved to have a F-shaped layout which can be seen in Figure 17. Under the title on the right side of the product picture, the information box is quickly available and the test user spots it immediately. The product presentation in the product page hence provides an easily understandable aspect.

The product listing was also considered for the most part appropriate for procurement purposes. However, the test user suggested that in addition to the images, a short description should be included already in the product listing phase. The current product listing is presented in Figure 18.

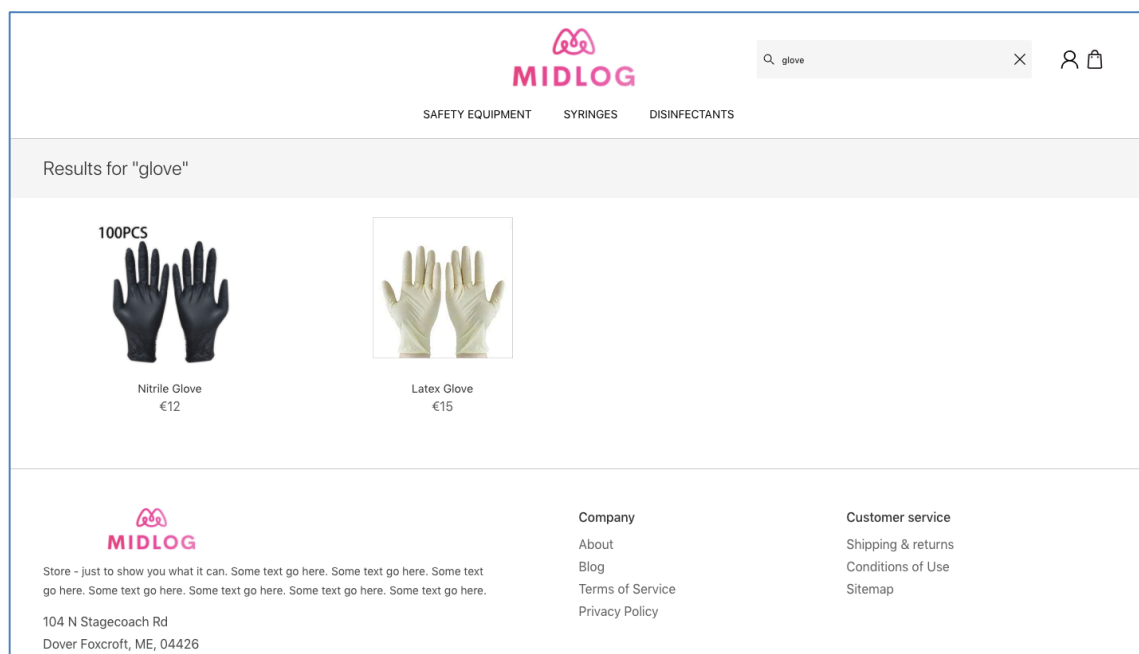


Figure 18, Product Listing of the Application

As seen, the price information, which was defined unnecessary in the previous subchapter, is, in addition to the product name, the only textual element of the product card. Derived from the theoretical perspective, the grid view which this user interface utilizes does not necessarily allow for detailed information in the product listing view. As the test user elaborated, in order to prevent mistakes in the product selection, a brief introductory text or attribute listing should be made available in the product cards. Otherwise the product listing was considered appropriate and the interactivity of the hovering effect on the cards was appreciated; further supporting the theoretical findings.

In further discussions regarding the product presentation in general, similar to the product discovery, there were vast differences to the current procurement applications. The current systems do not utilize any graphical aspects in the forms of product images, which, from the user viewpoint in the test setting, were found useful. Hence, the graphical aspect in the form of images in the product presentation could make the procurements more user-friendly. On the other hand, the product presentation in the open source

platform under development should be improved by removing the price information and adding detailed description to the product cards in the product listing phase. One option for doing this could be, derived from the theoretical discussion, switching the product listing view from grid to list, allowing for more room in the product card for detailed product information.

#### 6.1.4 Cart Management

In the cart management phase the test user was instructed to add two different products to the cart. After the first product was added, the user was asked to check the current content. This naturally required the user to find the cart in the page. This was an easy task as the “add to cart button” triggered the preview of the cart position in the page. The test user now clearly saw how to access the cart. This cart preview trigger is presented in Figure 19, the moment after the “add to cart button” was clicked.

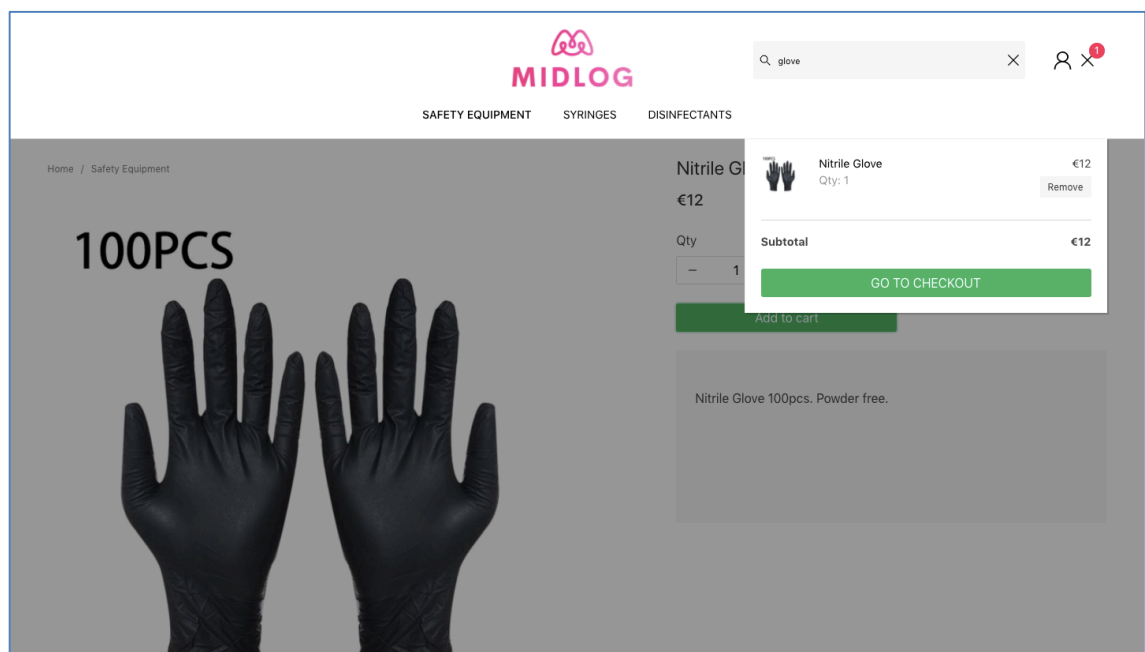


Figure 19, Cart Preview in the Application

After adding the second product, the test user was asked to proceed to the checkout phase. At this point a set of modifications were asked to be performed; first, modifying the quantities and removing a product. Thereafter, the user was asked to navigate back to the product page of a product that was in the cart, and the first action of the user was to click the product name in the cart, which then opened the product page. This supported

the findings that the product page should be accessible by clicking the product in the cart.

The test user perceived the cart as exceptionally logical to use and it had numerous advantages to the existing platforms. First of all, the current procurement applications do not have an equivalent to the cart management feature. In other words, the process of placing an order is based on individual searches where the products are individually searched and placed on an order sheet, which is not accessible throughout the product discovery phase. To compare, an e-commerce alternative includes dynamic cart management where the content of an order is continuously accessible and modifiable. In addition, the easy access from the cart to the product page and vice versa was considered efficient by the user.

To conclude the user evaluation on the cart management factor, the visualization of the content of the cart is an aspect which benefits stem from the basics of how human memory works. For instance, in a situation where the order has a lot of different products in it, the memorization of the content should be as easy as possible. The test user pointed out that as the e-commerce alternative presents the order contents in the cart conveniently with images, this helps the user to scan through the order and secure that everything is correct. As the current applications, according to the test user, do not have the product images on the order view, the e-commerce alternative could provide an improved version of the cart management in the public-sector procurements.

## 6.2 Summary

Based on the empirical study, all the tasks in the usability testing were proven to be convenient for the test user. This supported the theoretical perspective of the customer journey in e-commerce platforms and the user-centricity in the factors constituting the phases of the customer journey. As discussed, in many parts from the layout to individual components, the graphical user interface of the selected open source e-commerce application followed the general design guidelines of user-centricity of e-commerce.

The user interface evaluation quite indisputably supported the notion that the customer journey in e-commerce as well as the factors in e-commerce graphical user interfaces

are somewhat effortlessly transferrable to the procurement context in this case. Overall, based on the perceptions of the test user, there were virtually no pitfalls related to using an e-commerce platform for the procurement purposes. It could be deduced that the customer journey in procurements is similar to that in e-commerce applications designed for consumers.

Despite the general applicability of the e-commerce customer journey and the graphical user interface factors to the procurement context, a need for certain improvements was identified. To begin with, all the marketing-like content should be removed and category selection with images should be added to the site landing view. In addition, the price-related information was deemed unnecessary for this context, whereas the product cards would benefit from more detailed information than merely the product name. Hence, the option of replacing the grid view with list view should be considered since, based on the theory, the list view better enables the usage of more detailed information in the product cards. The identified needs for development are visualized in Figure 20 below.

<i>Customer journey phase</i>	<i>Identified need for development</i>
<b>Site Landing</b>	<b>Replace best seller section with product categories with images</b>
<b>Product Discovery</b>	<b>Enlarge the breadcrumb trail and emphasize it with color</b>
<b>Product Presentation</b>	<b>Remove price information and consider replacing grid view with list view, allowing product description in the product preview</b>

Figure 20, Identified Needs for Development of the Cezerin Platform

It should be emphasized that in addition to the fact that the open source alternative, in the graphical user interface perspective, could be applicable for the procurement context, it seemed in some parts to even improve the procurement processes when compared to the existing applications. The graphical user interface of an e-commerce platform offered an easy-to-understand system which was both readable and navigable. The use of images, navigational elements and dynamic cart management were aspects that were considered as useful and at the same time something that is lacking in the current procurement applications.

## 7 Discussion and Conclusions

The aim of this thesis was to explore the following research questions:

- 1) Which factors constitute a user-centric user interface (UI) in a web-based e-commerce platform?
- 2) Which open source e-commerce platform should be selected to be evaluated?
- 4) Does the user interface of the selected open source platform meet the requirements of the test user and if not, is there a need for modifications and which these modifications would be?

The discussion and conclusions chapter aims to draw together the main findings of the thesis process of and to shed light on its implications. The chapter comprises of three parts. First, the theory presented in Chapter 3 is reflected upon in light of the empirical findings. Thereafter, the managerial recommendations discuss the value that this thesis could provide in practice. The last part of the conclusion identifies potential limitations of this thesis, aspects to investigate further as well as ideas for future development of the platform.

## 7.1 Theoretical Discussion

Following the introduction to graphical user interfaces, the framework chosen to provide structure for this thesis was the customer journey framework by Mangiaracina and Brugnoli (2009). The customer journey framework proved successful for identifying the most crucial graphical user interface factors throughout the customer journey in the e-commerce context.

The empirical findings from the user interface evaluation proved that the elements composing user centrality, presented in the review of literature, such as F-shaped layout, navigational elements, and product presentation were in fact the elements that made the open source platform user-centric for the test user. Hence, the empirical part of this thesis was in line with the theoretical perspectives of user-centrality. The usability testing method as well as the interview with the test user proved that the four key factors of site landing, product discovery, product presentation and cart management in the selected platform were considered easy to use and convenient. In other words, the smooth completion of the tasks implied that the graphical user interface of the selected open source e-commerce platform can be argued to be user-centric.

Finally, an intriguing finding that this study implies is the similarity of the customer journey in the business to customer e-commerce and public-sector procurement contexts. The four phases of the customer journey presented in the theoretical framework followed remarkably well the steps that the test user goes through in the procurement in the health care context. Due to this similarity in the customer journeys it can be argued based on the empirical findings, that the user centrality as well as the key factors of graphical user interfaces in the e-commerce context can be transferrable to the procurement environment. Only a few key differences were found between the B2C e-commerce graphical user interface factors found in the theory and the requirements in health care sector procurement; these related to the product card information, price information and marketing content.

Based on the above, the first research objective was successfully met by constructing a holistic understanding on how a user-centric graphical user interface is constructed in the e-commerce environment.

## 7.2 Managerial Recommendations

This thesis provides a variety of valuable insights for practical implications. First of all, the customer journey of e-commerce and public-sector procurement in the chosen sector of health care were found to be similar in most parts. Consequently, it can be deduced that the e-commerce features could be further applied to software and applications in the procurement context. As the test user perceived the platform to be both user centric as well as sufficient for procurement purposes, further development is a feasible option to be seriously considered. As the objective of this study was to provide a user centric potential alternative to procurement, clearly an applicable option has been found: the selected open source platform not only met the requirements of the test user but was able to provide improvements compared to the existing platforms. In this sense, the third research objective was fulfilled by proving that an e-commerce graphical user interface could be used in public sector procurements and that the identified customer journey in e-commerce seems to be applicable in the public sector procurement context.

In addition, the evaluation process considering the chosen open source platform Cezerin was excessive as well as in-depth. This analysis, combined with the empirical findings, suggests that the presented open source platform provides a solid basis for future development. In other words, the study suggests that first of all, an open source e-commerce platform could have potential in procurements and, secondly, the presented Cezerin open source application would, based on its license type and used technologies, suit the procurement purposes. Hence, Cezerin, in the graphical user interface perspective, has potential to provide an alternative to the current procurement platforms which are used in the organization of the test user. Through the evaluation in Chapter 4, the second research question was answered when a legitimate open source platform was selected based on an in-depth analysis and evaluation of different selection criteria.

## 7.3 Suggestions for Development and Further Research

As stated, the study implies that the graphical user interface of a chosen open-source e-commerce platform has potential to improve the procurements compared to the existing applications. However, there were concrete points of improvement on how the e-

commerce platform could meet the needs of the client organization even better. It became evident through the empirical study that the price information and marketing content, such as the best-selling products in the home page should be removed. Instead of the best-sellers, in the home page, category boxes images should be introduced; as presented in Figure 6.

Furthermore, the price information in the product details and filtering tools was unnecessary, hence that should be deleted in the following development phases of the application. Considering the product listing, the application currently utilizes the grid view, and that should be replaced to a list. This is due to the procurement needs of the client where, already in the product preview, there should be room for detailed product description. As learned from the theoretical discussion, the grid view may not be suitable for this purpose. Another point of further development was the breadcrumb trail which should be visually more eye-catching. These identified key needs for development further answered the third research question by identifying how the user interface of the open source platform can be concretely improved and with which modifications.

When it comes to the empirical study, it should be emphasized that the usability testing as well as the overall evaluation only had one test user. As this is the case, it is a given fact that the biases and attributes of the test user may be dominant in the evaluation. A case in point, the test user navigated to the home page quite naturally by clicking the company logo. This is of course a well acknowledged route for experienced web-application users, but the question remains that how would have an inexperienced user navigated back to home page.

The lack of a wider test user base is clearly the most crucial limitation of this study. Hence, when the application is further developed, the following tests should account a larger number of test users. One potential approach would be, after the changes above are implemented, to have multiple rounds of usability testing in order to understand smaller details and how the background of the test user may alter the results. In addition, other evaluation techniques should be utilized to construct a holistic understanding on how the user interface could be developed further.

The user interface evaluation was carried out within the context of the Finnish health care industry, whereby the test user could only make comparisons to the current procurement applications used in this specific industry. In order to make the findings of this thesis more generally applicable to procurement, other industries should be considered. There are likely to be vast differences in both procurement processes as well as applications across the multitude of public and private sector industries and companies. Therefore, another limitation of this thesis is the context and industry specificity. In the scope of this thesis the focus was only on the application and its graphical user interface, but gathering a thorough understanding of the procurement processes in any given industry is crucial for the development of a truly successful procurement application.

Finally, it should be acknowledged that from a business perspective the application needs further evaluation and development in the backend side. As this study only evaluated the frontend code, it is still unclear that how the backend would suit the business requirements of the client organization. Hence, in addition to the mentioned modifications to Cezerin platform and broadening of the test user base, the further development and research should begin to analyze the suitability of the backend code and database structure of Cezerin platform.

## References

- 1 Law, Thomas. 2019. 19 Powerful Ecommerce Statistics That Will Guide Your Strategy in 2020. Oberlo. Online. <<https://www.oberlo.com/blog/ecommerce-statistics-guide-your-strategy>> Accessed 16.03.2020
- 2 Norman, Don & Nielsen, Jakob. The Definition of User Experience (UX). Nielsen Norman Group. Online. <<https://www.nngroup.com/articles/definition-user-experience/>> Accessed 16.03.2020
- 3 Richard, Charles. Ecommerce UX: What It Takes To Create The Best User Experience For Your Online Store. BigCommerce. Online. <<https://www.bigcommerce.com/blog/ecommerce-ux/#what-is-ecommerce-user-experience>> Accessed 16.03.2020
- 4 Oxford Learner's Dictionary. Online. <<https://www.oxfordlearnersdictionaries.com/definition/english/user-interface>> Accessed 16.03.2020
- 5 Christensson, Per. 2009. User Interface Definition. TechTerms. Online. <[https://techterms.com/definition/user\\_interface](https://techterms.com/definition/user_interface)> Accessed 16.03.2020
- 6 Smashing Magazine. 2011. User Interface Design in Modern Web Applications. Online. <<https://www.smashingmagazine.com/user-interface-design-in-modern-web-applications>> Accessed 16.03.2020
- 7 Amy, Ko. A history of user interfaces. Online. <<https://faculty.washington.edu/ajko/books/uist/history.html>> Accessed 17.03.2020
- 8 Powell, Adam. Web 101: A History of the GUI. Online. <<https://www.wired.com/1997/12/web-101-a-history-of-the-gui/>> Accessed 17.03.2020
- 9 Computer Hope. 2019. GUI. Online. <<https://www.computerhope.com/jargon/g/gui.htm>> Accessed 17.03.2020
- 10 Banga, Swapnil. 2020. Web Application Architecture: Definition, Models, Types, and More. Hackr.io. Online. <<https://hackr.io/blog/web-application-architecture-definition-models-types-and-more>> Accessed 18.03.2020
- 11 Dabbs, Mark. 2019. The Fundamentals of Web Application Architecture. Reinvently. Online. <<https://reinvently.com/blog/fundamentals-web-application-architecture>> Accessed 18.03.2020

- 12 BigCommerce Essentials. What is ecommerce UI?. Online. <<https://www.bigcommerce.com/ecommerce-answers/what-is-ecommerce-ui/>> Accessed 19.03.2020
- 13 Tubik Studio. UX for Web: 5 Must-Have Elements for Any E-Commerce Website. UX Planet. Online. <<https://uxplanet.org/ux-for-web-5-must-have-elements-for-any-e-commerce-website-ef613d1d3f4e>> Accessed 19.03.2020
- 14 The App Solutions. Ecommerce Ui/Ux Design: Principles and Elements. Online. <<https://theappsolutions.com/blog/design/ecommerce-ux-principles/>> Accessed 20.03.2020
- 15 Tubik Studio. UX for Web: 11 Profitable Strategies for E-Commerce UI Design. UX Planet. Online. <<https://uxplanet.org/11-profitable-strategies-for-e-commerce-ui-design-8f71b2b98a9f>> Accessed 20.03.2020
- 16 Carlson, Billy. UI Design Basics for Ecommerce Pages. Balsamiq. Online. <<https://balsamiq.com/learn/resources/articles/ecommerce-ui-best-practices/>> Accessed 20.03.2020
- 17 Megharani P & Madhuri R. 2016. Investigating Procedures for Intuitive GUI design of Indian ECommerce shopping sites. IOSR Journal of Computer Engineering (IOSR-JCE) 18(4):159-169. Online. <<https://pdfs.semanticscholar.org/4f6e/9c888fb6844b518e6e3bc249a8b46e04916c.pdf>> Accessed 21.03.2020
- 18 Kristensen, Emil. 2019. 13 E-Commerce Homepage Best Practices for Conversions in 2019. Sleeknote. Online. <<https://sleeknote.com/blog/e-commerce-homepage-best-practices>> Accessed 22.03.2020
- 19 Babich, Nick. 2017. F-Shaped Pattern for Reading Content. UX Planet. Online. <<https://uxplanet.org/f-shaped-pattern-for-reading-content-80af79cd3394>> Accessed 22.03.2020
- 20 Baymard Institute. E-Commerce Sites Need 2 Types of Breadcrumbs (68% Get it Wrong). UX Research Articles. Online. <<https://baymard.com/blog/ecommerce-breadcrumbs>> Accessed 23.03.2020
- 21 Tyrrell, Paige. 2018. Top 25 Best Practices for Ecommerce Site Search. PrefixBox. Online. <<https://www.prefixbox.com/blog/ecommerce-site-search/>> Accessed 23.03.2020
- 22 Expert Commentator. 2016. 12 e-commerce tips for your category pages. SmartInsights. Online. <<https://www.smartinsights.com/ecommerce/ecommerce-strategy/12-e-commerce-tips-category-pages/>> Accessed 23.03.2020

- 23 Whitmore, Stuart. 2016. 7 Secrets to Successful E-Commerce Product Listings. The Self Employed. Online. <<https://www.theseemployed.com/ecommerce/7-secrets-to-successful-product-listings/>> Accessed 23.03.2020
- 24 Mei. 2018. List or Grid, It Is Not Important. UXPlanet. Online. <<https://uxplanet.org/listview-or-gridview-it-is-not-important-b767d20ec05e>> Accessed 24.03.2020
- 25 Expert Commentator. 2016. 12 e-commerce tips for your category pages. SmartInsights. Online. <<https://www.smartinsights.com/ecommerce/ecommerce-strategy/12-e-commerce-tips-category-pages/>> Accessed 23.03.2020
- 26 Opensource.com. What is Open Source? Online. <<https://opensource.com/resources/what-open-source/>> Accessed 26.03.2020
- 27 Noyes, Katherine. 2010. 10 Reasons Open Source Is Good for Business. PCWorld. Online. <[https://www.pcworld.com/article/209891/10\\_reasons\\_open\\_source\\_is\\_good\\_for\\_business.html%201](https://www.pcworld.com/article/209891/10_reasons_open_source_is_good_for_business.html%201)> Accessed 26.03.2020
- 28 Byfield, Bruce. 2015. Open Source Debate: Copyleft vs. Permissive Licenses. Datamation. Online. <<https://www.datamation.com/open-source/open-source-debate-copyleft-vs.-permissive-licenses.html>> Accessed 26.03.2020
- 29 Mukundan, Chakrapani., Prashant, Phillips & Raghav, Sarda. 2015. Software licensing & open source software. NASSCOM. Online. <<https://www.slideshare.net/nasscom-emerge/software-licensing-open-source-software>> Accessed 26.03.2020
- 30 Goldstein, Ayla. 2020. Open Source Licenses in 2020: Trends and Predictions. WhiteSource. Online. <<https://resources.whitesourcesoftware.com/blog-whitesource/top-open-source-licenses-trends-and-predictions>> Accessed 26.03.2020
- 31 tldrLegal. #Permissive Licenses. Online. <<https://tldrlegal.com/licenses/tags/Permissive>> Accessed 26.03.2020
- 32 The Kiuwan Team. 2019. Open Source Licenses: A Comparison Of The Most Popular Types. Online. <<https://www.kiuwan.com/blog/comparison-popular-open-source-licenses/>> Accessed 2.4.2020
- 33 Exygy. 2016. Which License Should I Use? MIT vs. Apache vs. GPL. Online. <<https://exygy.com/blog/which-license-should-i-use-mit-vs-apache-vs-gpl/>> Accessed 2.4.2020

- 34 Mazur, Anar. 2017. Apache license 2.0, MIT license or BSD license : Who is the fairest of them all? Snyk. Online. <<https://snyk.io/blog/mit-apache-bsd-fairest-of-them-all/>> Accessed 2.4.2020
- 35 Wheeler, David. 2011. How to Evaluate Open Source Software / Free Software (OSS/FS) Programs. Online. <[https://dwheeler.com/oss\\_fs\\_eval.html#detailed\\_addfunctionality](https://dwheeler.com/oss_fs_eval.html#detailed_addfunctionality) /> Accessed 2.4.2020
- 36 Borges, Hudson & Valente, Marco Tulio. 2018. What's in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform. Online. <<https://arxiv.org/pdf/1811.07643.pdf>> Accessed 2.4.2020
- 37 MongoDB. NoSQL Databases Explained. Online. <<https://www.mongodb.com/nosql-explained>> Accessed 2.4.2020
- 38 NodeJS. Introduction to Node.js. Online. <<https://nodejs.dev/>> Accessed 2.4.2020
- 39 React. Tutorial: Intro to React. Online. <<https://reactjs.org/tutorial/tutorial.html>> Accessed 2.4.2020
- 40 Homebrew. Online. <<https://brew.sh>> Accessed 2.4.2020
- 41 Cezerin. Using Source Code. Online. <<https://cezerin.org/docs/#/using-source-code?id=a-cezerin-backend-api-installation>> Accessed 2.4.2020
- 42 Mangiaracina, Riccardo & Brugnoli Gianluca.. 2009. The ecommerce customer journey: A model to assess and compare the user experience of the ecommerce websites. The Journal of Internet Banking and Commerce, 14(3), 1-11. Online. <<http://www.icommercentral.com/open-access/the-ecommerce-customer-journey-a-model-to-assess-and-compare-the-user-experience-of-the-ecommerce-websites.php?aid=38368>> Accessed 2.4.2020
- 43 Visual Paradigm. What is Customer Journey Map? Online. <<http://visual-paradigm.com/guide/customer-experience/what-is-customer-journey-mapping/>> Accessed 2.4.2020
- 44 Cezerin. Using Source Code. Online. <<https://cezerin.org/docs/#/using-source-code?id=a-cezerin-backend-api-installation>> Accessed 2.4.2020

- 45 Koskinen, I., Alasuutari, P. & Peltonen, T. 2005. Laadulliset menetelmät kauppatieteissä (Qualitative methods in business research). Vastapaino, Tampere
- 46 Eriksson, P. & Kovalainen, A. 2008. Qualitative methods in business research. SAGE Publications Ltd., London
- 47 Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. 1991. User interface evaluation in the real world: a comparison of four techniques. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 119-124). Online. <<https://dl.acm.org/doi/pdf/10.1145/108844.108862>> Accessed 6.4.2020
- 48 Lam, Jeremiah. 7 shopping cart design guidelines. UXCollective. Online. <<https://uxdesign.cc/7-shopping-cart-design-guidelines-4f20bdd9f4e9>> Accessed 6.4.2020
- 49 Najjar, Lawrence J. 2005. "28 Designing E-commerce User Interfaces.". Online. <[http://lawrence-najjar.com/papers/Designing\\_e-commerce\\_user\\_interfaces\\_2011.pdf](http://lawrence-najjar.com/papers/Designing_e-commerce_user_interfaces_2011.pdf)> Accessed 6.4.2020
- 50 Fang, Xiaowen, and Gavriel Salvendy. 2003. "Customer-centered rules for design of e-commerce web sites." *Communications of the ACM* 46.12: 332-336. Online. <<https://dl.acm.org/doi/pdf/10.1145/953460.953518>> Accessed 6.4.2020

## Appendix 1

### Task Set and Interview Guide

#### Site Landing

**1. Locate the search box**

**2. Locate the product categories**

*What specially caught your attention in the site? Why?*

*Is there some information missing in the home page? What?*

*Are the core elements, considering your knowledge on procurments, found on the home page?*

*Are there elements that are unnecessary and and/or confusing? Is so, Which ones and why?*

*Is the layout, considering the procurement needs, appropriate and understandable?*

.....

#### Product discovery

**1. Find a nitrile glove using text search**

**2. At the nitrile glove product card, what should be the easiest way to locate other gloves?**

**Which tool was used and why? Why not other tools?**

**3. Navigate back to home page**

**4. Find a hand sanitizer using category search**

*Does the site guide you properly? Was it easy to find the search tools?*

*Are the core search elements, considering your knowledge on procurements, present in the user interface?*

*Considering how you find products in your day-to-day procurements, what are the search tools that you used? What product attributes*

*Are there elements that are unnecessary and and/or confusing? Is so, Which ones and why?*

*Compared to the existing methods, is finding the products more/less/equally convenient? Why is that?*

.....

Product presentation

**1. Search for gloves**

**2. Access the product page**

**3. How many pieces are in a package?**

In the product listing, is all the information available? What is missing?

Is the listing and the products in the listing easy to read, what caught your attention in the product listing? Was this a good thing?

Is it convenient to access the product page?

At the product page, is the layout and information missing? What is the most important information considering your needs?

How could the product page be improved?

.....

Cart Management

- 1. Add a nitrile glove product to the cart**
- 2. Search for hand sanitizer**
- 3. At the hand sanitizer product page, check the cart content, what is in the cart?**
- 4. Add sanitizer in the cart**
- 5. Proceed to checkout**
- 6. Remove Sanitizer**
- 7. Change the number of gloves in the cart**
- 8. Navigate back to the nitrile glove product page**

**Which was chosen why?**

*Is it convenient to manage and check the cart content?*

*Considering your needs, what functions are the necessary and were they present? What should be improved?*

.....

*In addition to the tasks and task related questions, do you have any other observations related to the user interface?*

*In general, would it meet the requirements?*

*Should there be any points of improvement?*

*Compared to existing system, what are the pros and cons of the tested user interface?*

.....