



Expertise
and insight
for the future

Nguyen Hong Loan

Shopping Center Web Application enhancement and redesign through React and NodeJS.

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

10 April 2020

Author Title Number of Pages Date	Nguyen Hong Loan Shopping Center Web Application enhancement and redesign through React and NodeJS. 39 pages 10 April 2020
Degree	Bachelor of Engineering
Degree Program	Information Technology
Professional Major	Mobile Solutions
Instructors	Kari Salo, Principal Lecturer
<p>The main objectives of the thesis were to enhance and redesign the shopping center web application by using modern technology. Additionally, this thesis is used to demonstrate the outcome of the author's learning of React, Redux, and NodeJS. The result of this project is a redesigned offers page, which is fit for the requirements of UX and meet the user's needs.</p> <p>This project consists of investigating the main principles of user experience and user interface design by conducting interviews aimed at collecting background information about the user's behaviors, needs, and suggestions. The next step is creating the wireframes, which helped to make choices about how to build a webpage structure and decide the suitable place for different visual elements. After that, the project continued with webpage implementation. A REST web service was built to provide data to the application. Visual design creation was based on choosing suitable fonts, color palette, and interaction details. ReactJS with Redux, Bootstrap was chosen as the framework for frontend due to author's eagerness to learn new technology.</p>	
Keywords	Redesign, UX, UI, React, Redux, NodeJS, Web Development

Contents

List of Abbreviations

1	Introduction	1
2	User experience and user interface in theory	2
2.1	User experience design	2
2.1.1	Information architecture	4
2.1.2	Interaction design	6
2.1.3	Visual design	7
2.1.4	Usability	9
2.2	User interface	11
3	Technology in theory	12
3.1	React and Redux	13
3.1.1	Virtual DOM	13
3.1.2	React component	14
3.1.3	Component lifecycle	15
3.1.4	Redux	15
3.2	NodeJS	18
3.3	Bootstrap	19
4	Website development	19
4.1	Collecting background information	20
4.2	Wireframing	22
4.3	Website implementation	25
4.3.1	Data structure	25
4.3.2	Color palette and font	27
4.3.3	Webpage implementation	28
4.3.4	Outcome and screenshots	32
4.4	Testing and feedback	35
5	Conclusions	36
	References	38

List of Abbreviations

UX	User Experience
UI	User Interface
IA	Information Architecture
Ixd	Interaction Design
API	Application Programming Interface
REST	Representational State Transfer
ES6	ECMA Script 2015. One version of ECMAScript standard
DOM	Document Object Model
JSON	JavaScript Object Notation
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets

1 Introduction

The aim of this project was to enhance and redesign the shopping center web application using React, Redux framework, and NodeJS. Additionally, this thesis is used to demonstrate the outcome of the author's learning of React, Redux, and NodeJS.

In Finland, especially in the capital area, there are so many shopping centers that attract a huge amount of people visit every day. Furthermore, the practice of technological solutions for daily life is increasing fast in these times. That means shopping centers tend to use their websites to provide information to their customers as well as to promote their brands, campaigns, or offers. Besides, customers nowadays rely on their laptops and mobile phones to cope with their day-to-day routine (for example, checking shopping center campaigns before they go there). Therefore, it creates a high requirement for website development in taking user's attention and attracting them to visit the websites.

In the scope of this thesis, none of shopping centers were involved, but their websites were utilized as a part of the interview. The interview was processed with some users at Sello, Iso Omena, and Itis to learn more about users' behavior and the issues they faced when using the existing websites. After that, I analyzed the findings and defined which parts of the existing websites could be enhanced or what new features could be implemented. The outcome of this thesis can help the shopping centers have a general view of their customers' thoughts, what issues they face, what they want while using the websites, and a good example to improve their websites.

2 User experience and user interface in theory

In our time, the modern knowledge of design is extensive. The actual definition of design contains the range of thoughts and changes from printing to technologies and industry. The theories of user experience and user interface design are not simple to explain in a few sentences. Since these are somewhat up-to-date disciplines, the question of their meaning might be confusing to many people. Subsequently, this chapter determines the disciplines of UX and UI as well as identify their essential roles in web design. Besides, the example of UX and UI allows explaining the interdependence between these two concepts.

2.1 User experience design

The definition of UX is described in many different ways and used to indicate different things in different studies. However, no agreement has been met on what correctly UX means. Following one of the most popular UX journal, Hassenzahl and Tractinsky define UX as a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (organizational/social setting, meaningfulness of the activity, voluntariness of use, etc.) [18].

UX design stands for user experience design. In other words, user experience is any interaction someone can have with a service or product. UX design considers every component that forms this experience, how it presents the user feel, and how simple it is for the user to achieve their wanted errands. This might be something from how a product feels in their hands, how easy user can process the checkout when viewing and shopping something online. The main purpose of UX design is to create simple, proficient, significant, and helpful, friendly experiences for users [1].

UX could be a board umbrella term, but when it comes to web application, it can be divided up into four main disciplines as figure 1 illustrates: Information Architecture, Interaction Design, Visual Design, and Usability.

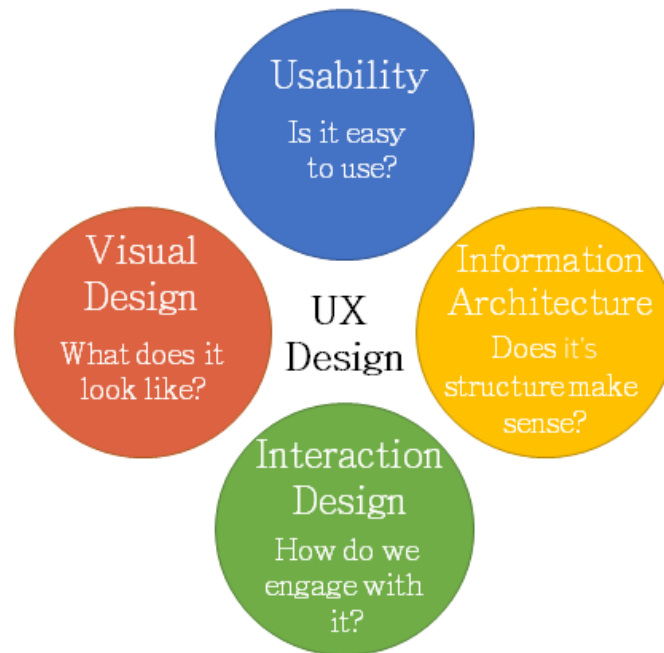


Figure 1. Four main disciplines of UX design.

Why is UX design important?

Nowadays, everyone would probably be satisfied with describing and justifying the relevance of designing and enhancing the user experience looks almost unnecessary, with so much importance on user-centered design [2]. However, true things are not always going this way. Before clients knew truly the value of user-centered design, they made these choices centered on two points: what they thought was great, outstanding and what the client required to see (Figure 2).

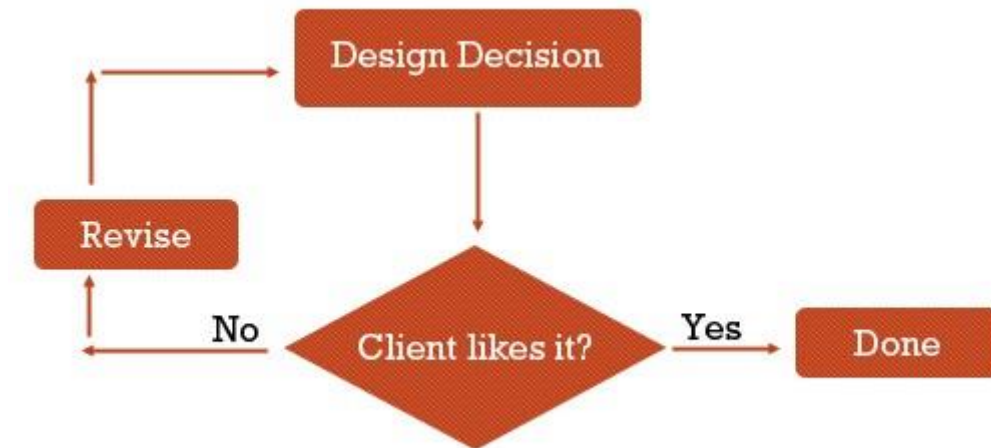


Figure 2. Outmoded decision process of UX design. Copied from [2]

Over the last few years, however, has witnessed several changes in the Website application. Not only has it become global, but website applications have grown more complex and fully featured that, to be productive and efficient, they need to have great UX designs.

Additionally, users have been accessing websites through numerous diverse ways: mobile phones, laptops, tablets, various browsers, different kinds of Internet connections. Moreover, they have become conscious of the value of accessibility — not only for those with particular needs but also for those who have older mobile devices or don't have broadband connections and so forth [2].

With all of these tremendous changes, the website applications that have reliably stood out from the crowd were the ones that were pleasurable to practice and approach the user's demands efficiently [2].

2.1.1 Information architecture

Information architecture is a document that gives an operational map of how product or service performances plus features run for users. It's associated with an outline for digital products; furthermore, it represents pages, content, interactions, and behaviors for the whole product [3].

People often utilize the definition of information architecture (IA) to mean the menus on websites, but this is only a part of it. Designers create information architecture when building products like a regular part of the UX process. It could be much more than just a sitemap to indicate what page links where since it defines every road and path that users can go through applications or websites. In other words, information architecture is meant to assist users in knowing their virtual environment and finding what they are searching for. The last aiming of information architecture is to form a structure/design that meets the balance between a business' and users' needs [3].

A great idea to summarize IA is a picture that showed in the book - "Information Architecture for the World Wide Web" by Louis Rosenfeld and Peter Morville [3]. As mentioned by them, IA includes content, context, and users (figure 3).

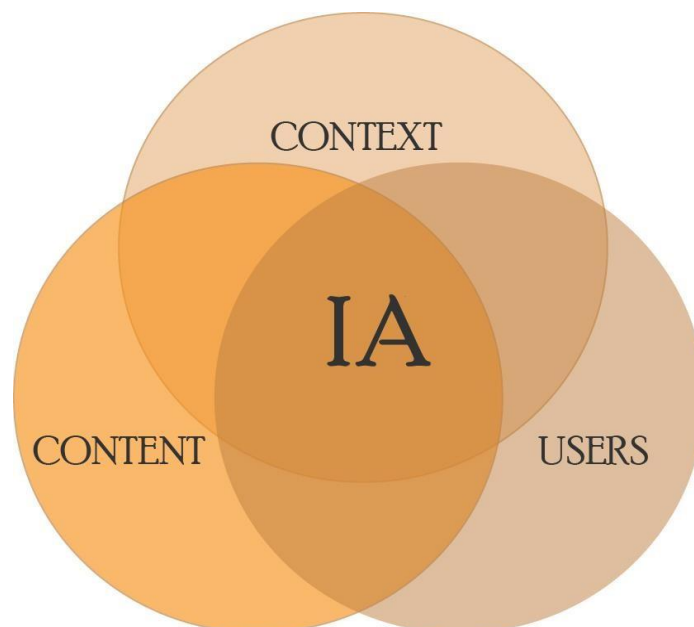


Figure 3. The 'three circles of IA' as content, users and context.

Each section indicates:

- Context is significantly linked to business goals, financial scope, culture, politics, technology [3].
- Content describes the structure of information. In other words, this part is responsible for the proper document, data types, and volume [3].

- Users section of IA can be defined as the way users observing information. Furthermore, it combines the evaluation of IA offered by user's behavior and experience [3].

2.1.2 Interaction design

As reported by the Interaction Design Association, interaction design (Ixd) explains the structure and behavior of interactive systems [4]. Moreover, interaction design regarding design to the entire interconnected system, which includes the device, the interface, the context, the environment, and the people [4]. Figure 4 shows its complex and wide-ranging field, containing five primary principles, which are consistency, learnability, visibility, predictability, and feedback.

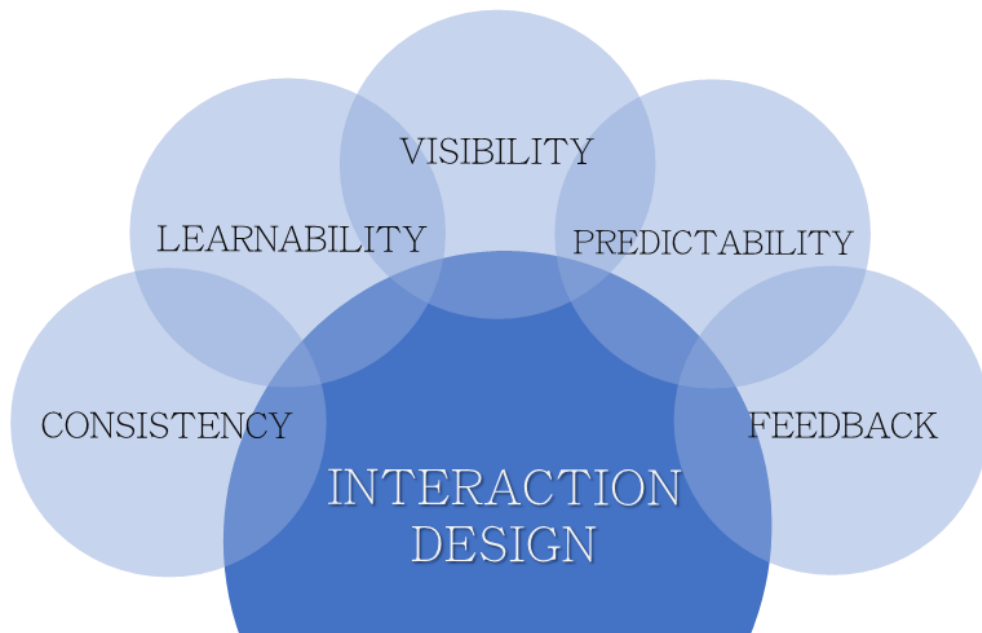


Figure 4. The five main principles of Ixd (Interaction Design).

The first component of interaction design is consistency. Due to the standard human behavior theory, people are delicate, sensitive to all types of changes [4]. Therefore, they instantly notice when anything unusual or any new appearance for their experience. This consistency aims to set a rule of design that enables users to concentrate on their tasks, feel comfortable while discovering something new.

The second principle is learnability. Interactions should be easy to remember and learn. The more natural interaction is, the better ways of actions are, due to the ability to remember and learn an interface. Ideally, users should be able to experience interface once, get it, and keep in mind for a long time [4]. Mostly, users recently want to practice an interface at least several times before they get it, then after that, it's a success; if users can remember what they experienced.

Another principle, called visibility, is one of the essential principles of interaction design. In general, it explains the existence of an interaction with a user. Users shouldn't have to seek for chances to interact. They shouldn't guess while interacting due to desperation or confusion. Additionally, interaction shouldn't depend on luck or random discovery. Hidden interactions decrease usability and effectiveness [4]. For example, all buttons and hyperlinks are assumed to be "clickable". It means some specific style between the content and the component, like color, font-weight, or font-size.

Predictability in interaction design indicates that great design should establish accurate expectations regarding what has been planned to happen before any actual interaction comes up. Designers can use video instructions, labels, icons, etc. as the best guides for users [4].

Feedback provides recognition of the user's interactions and information about their outcomes. It can be applied to understand what is happening as well as what might appear ahead [4].

2.1.3 Visual Design

Visual design principles is a combination of graphic and user experience. It helps to improve the usability and interaction by the usage of elements such as images, shapes, typography, grid, and colors [5]. Figure 5 presents five principles of visual design.

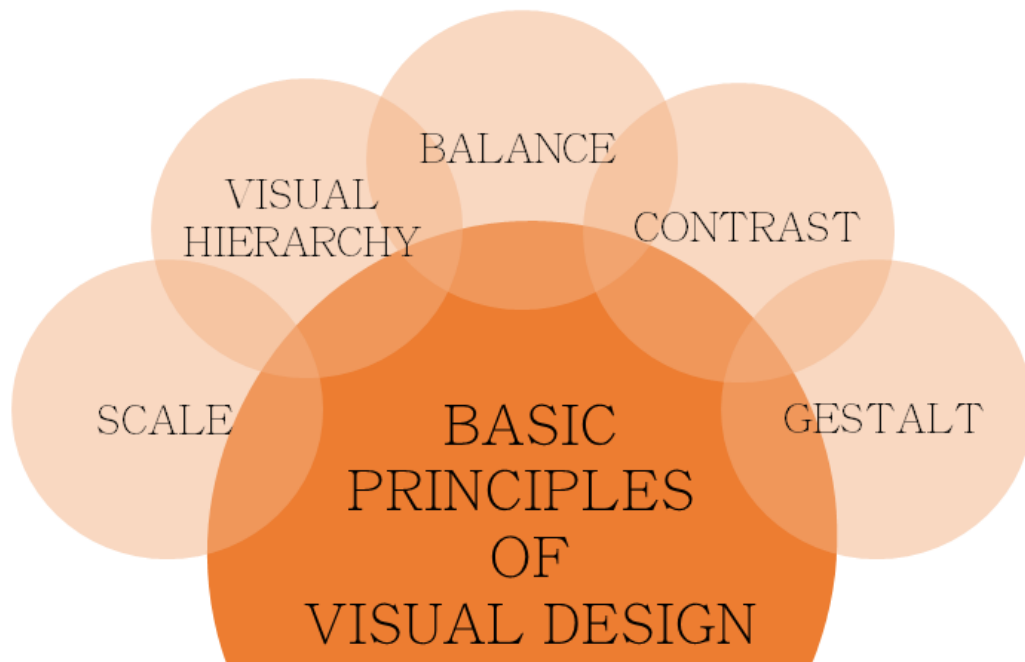


Figure 5. The five basic principles of Visual Design.

The first one is **scale** that mentions applying relative size to indicate importance and rank in a layout. To be specific, when following this scale principle correctly, the more important elements in a layout are definitely bigger than the less important ones. It's reasonable and simple to comprehend: the bigger something is, the more notice it is [5].

The next principle is **visual hierarchy**. It mentions managing the eye on the layout so that it goes to different elements following their importance level [5].

It is usually actualized by variations in scale, value, color, spacing, placement, and other signals [5]. In case users have got a difficult time finding out where to see on the layout, it can be assumed that a clear visual hierarchy of the layout is missing.

The **balance** principle in visual design introduces a pleasing order and proportion of design elements. Balance happens when the equally divided amount of elements on both sides of an imaginary axis (often vertical but can also be horizontal) is set, going through

the middle of the screen [5]. Designers only meet the needs of a great balanced design when no area is taken so much attention that users can't notice the others.

The **contrast** points out the juxtaposition of visually different elements to bring the truth that these elements are dissimilar [5]. In detail, contrast gives user's view with a noticeable distinction between two elements aiming to indicate that they are different (belong to separate sections, run different functions, work separately). Color is usually used to create contrast in the layout.

Gestalt principles demonstrate the way people simplify and coordinate complicated images that be made up of numerous elements, by subconsciously managing all elements into an ordered system that builds a whole, instead of understanding them as a set of different parts [5].

2.1.4 Usability

Usability is a quality attribute that assesses how simple and straightforward it is for users to perform functions on websites. Additionally, the word "usability" points out easy to use methods for developing during the design process [6].

Why is Usability important?

If users find it difficult to use on a website, they tend to leave and find another one. Moreover, in the case of the homepage doesn't clearly show what a website offers, or what they can experience on the site, they will go away. Another case of users gets lost when trying to find the information they need; they seem to leave also. If information from a website is difficult to read or doesn't have the answers they are looking for, they leave. Because users can visit lots of alternative websites available, leaving is the primary line of defense when they face difficulties. Therefore, it is worthless and inefficient in developing a website if no one visits long enough to get the information on it.

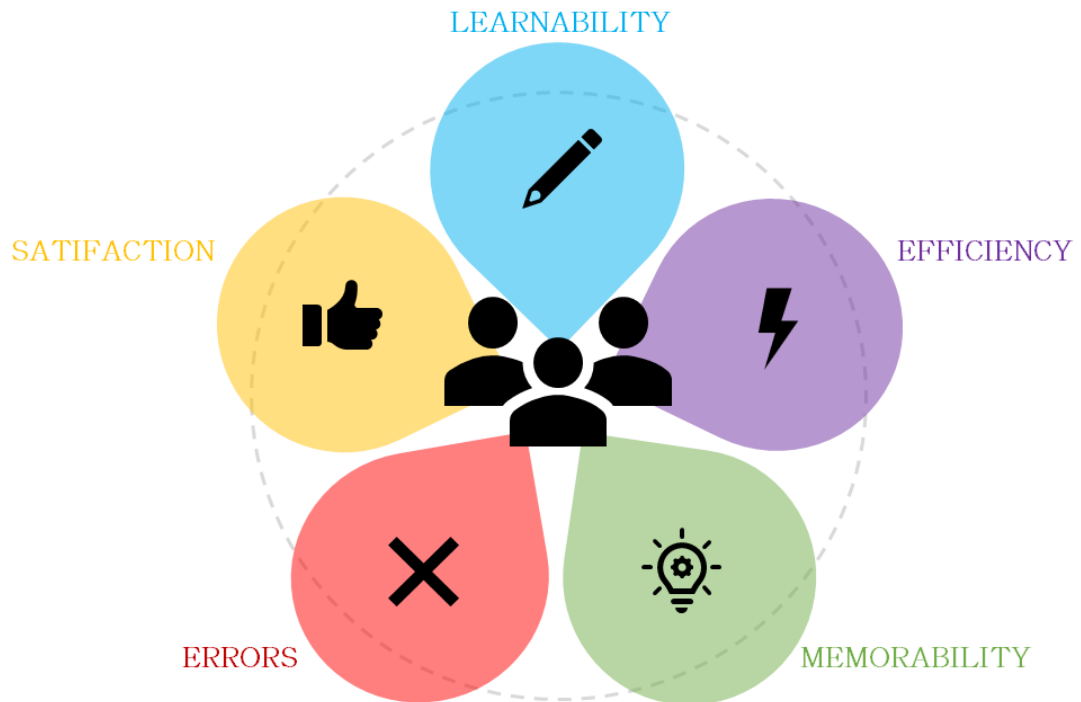


Figure 6. The five components of Usability.

Jakob Nielsen defines usability by five quality components, as figure 6 illustrates:

- Learnability measures how much easiness a website can provide to users to fulfill primary tasks for the initial time they meet the design [6].
- Efficiency is about how fast users can complete tasks after they have known the design [6].
- Memorability is about how simple users can restore their knowledge when they go back to the design after an amount of time not practicing it [6].
- Errors are about how many faults users get, how severe these faults are, and how simple they can fix their faults [6].
- Satisfaction — How comfortable it is to use the design [6].

Other essence attributes of usability involve utility and usefulness. Utility refers to the design's capacity to function; does it provide the valuable features that the user needs? Usefulness refers to the ability to accomplish tasks efficiently and proficiently.

2.2 User Interface

UI design, known as user interface design, concentrates on predicting, forecasting what users might want to do and guaranteeing that all visual elements of a digital product's interface are easy to access, read, and interact with [7].

User interface design is usually mixed with user experience design. Basically, UI design, which plays an important role in the design process, is a part of UX design [7]. Because UI and UX often go hand-in-hand closely, they need to be separated into different parts of the design process.

In general, UX design is the process of improving, fulfilling user's satisfaction by developing the usability and accessibility of a product, website, or application. Alternatively, UI design is more concerned with the look and overall feel of design—in other words, what users see when they interact with the product. UI design makes the website maximize efficiency, responsiveness, and accessibility, as well as concentrate on aesthetics [7].

UI elements usually divided into four following categories shown below:

- **Input controls:** they support users to add information into the system. This includes buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field [7].
- **Navigation components:** navigation components assist users in moving around a website or application. General navigation components contain some elements such as search field, slider, tags, icons, image carousel [7].
- **Informational components:** their mission is to give information to users. This contains tooltips, icons, progress bars, notifications, modal windows (popup) [7].

- Containers: they keep linked content together, for instance, accordions. The accordion is a vertical list of items which utilize show or hide functionality. Accordion allows the display of only one collapsed item at a time. Figure 7 represents an example of accordion [7].

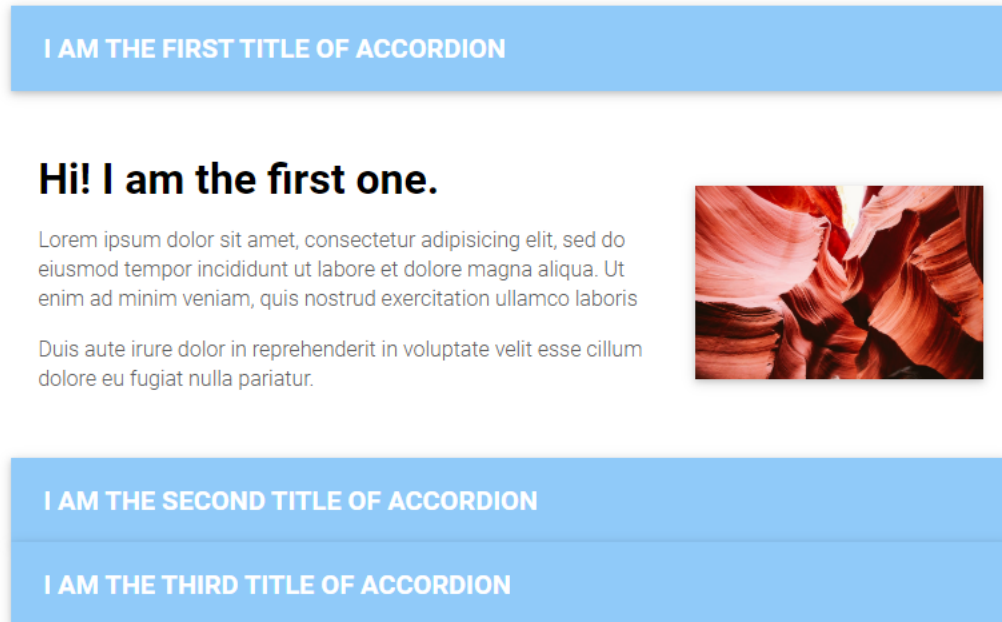


Figure 7. Accordion with a picture.

3 Technology in theory

This chapter is divided into three sub-chapters to show and concentrate on explaining the theoretical technology used in this project. In 3.1, the summary introduction of React and Redux - the core front-end framework that fully applied to generate the user interface. Coming after is in 3.2, the general understanding of NodeJS, which used to build a simple REST web service to provide data to the application. Finally, the presentation of Bootstrap 4 - is the latest version of Bootstrap, which is one of the most popular HTML, CSS, and JavaScript framework for developing responsive websites [17].

3.1 React and Redux

ReactJS (React or React.js) is an open-source JavaScript library that is used for building user interfaces, especially for single-page applications. It's managed the view layer for websites or mobile apps [8]. React also supports developers in creating reusable UI components. React was first founded by Jordan Walke, who is working as a software engineer at Facebook. The initial time React was implemented on Facebook's newsfeed was in 2011 and on Instagram was in 2012. Later then, React speedily become one of the most popular chosen libraries by developers, among many other UI libraries.

React is high-grade in performance, easy to learn, and fast to set up. To improve the capability in reusable code, React has encapsulated components, that can be shared and reused between application codebase. Accordingly, React supports to enhance the codebase consistency, straightforward in testing, decrease duplicated work, saving time given on the working process. Websites or mobile apps can be split into small components, even if how big it is.

To understand React intensely, developers should know the main points (virtual DOM, React component, component lifecycle) that make React success toward others.

3.1.1 Virtual DOM

To get things direct - DOM stands for *Document Object Model* and is an abstraction of a structured text that is an HTML code for web developers. The DOM is called HTML DOM, and elements of HTML become nodes in the DOM [9].

Nowadays, the DOM trees are huge. Since developers are increasingly worked towards dynamic web application that generates the pages/data in real-time, they need to put more effort to work with the DOM tree [9]. Consequently, this is a real performance and development issue.

React comes with a helping hand. Developers simply declare how a component should look like instead of traversing the DOM tree manually that might look a bit low level of

techniques. React supports the low-level job by calling the HTML DOM API methods under the hood [9].

Virtual DOM is a simplified and lightweight version of the HTML DOM. In React web application, whenever a change of data occurs, the entire application UI is re-rendered and stored in its virtual DOM representation. Virtual DOM is a practical method to update the application UI, and for this reason, it improves the performance of the applications [9].

3.1.2 React Component

React components are the structure blocks of any React application, and a common React application will have several of these. Basically, a component can be a JavaScript class or function. Functional component or class component receives inputs or properties (props) as an option and returns a React element that defines how a part of the whole UI should appear [10].

There are two different ways to write components: Functional (Stateless) and Class (Stateful) components.

Functional components are completely presentational and are simply rendered by a function that accepts props as an argument and returns a React element to be presented to the page. Typically, it is preferred to use functional components whenever possible thanks to their predictability, conciseness, as well as easy to understand and test [10].

```
const ExampleReact = () => <h1>Moi, I'm a functional component!</h1>;
```

Listing 1. A simple React functional component.

Class components are built using ES6's class syntax. To use this, developers need you to extend from `React.Component` as well as implement a render function (in listing 2 below for example) that returns a React element. They have some further features such as the ability to contain logic (for instance, methods that handle `onClick` events), include numerous other functional components, and update changes of internal state or props [10].

```

class ExampleReact extends React.Component {
  render() {
    return <h1>Moi, I'm a class component!</h1>;
  }
}

```

Listing 2. A simple React class component.

3.1.3 Component Lifecycle

There are several lifecycle methods that React provides at different phases of a component's life. React automatically calls the responsible method according to the stage in which the component is. These methods give developers better control over components, and they can handle them using these methods [11].

React component lifecycle can be separated into three phases that are shown in the Figure below: mounting, updating, and unmounting.

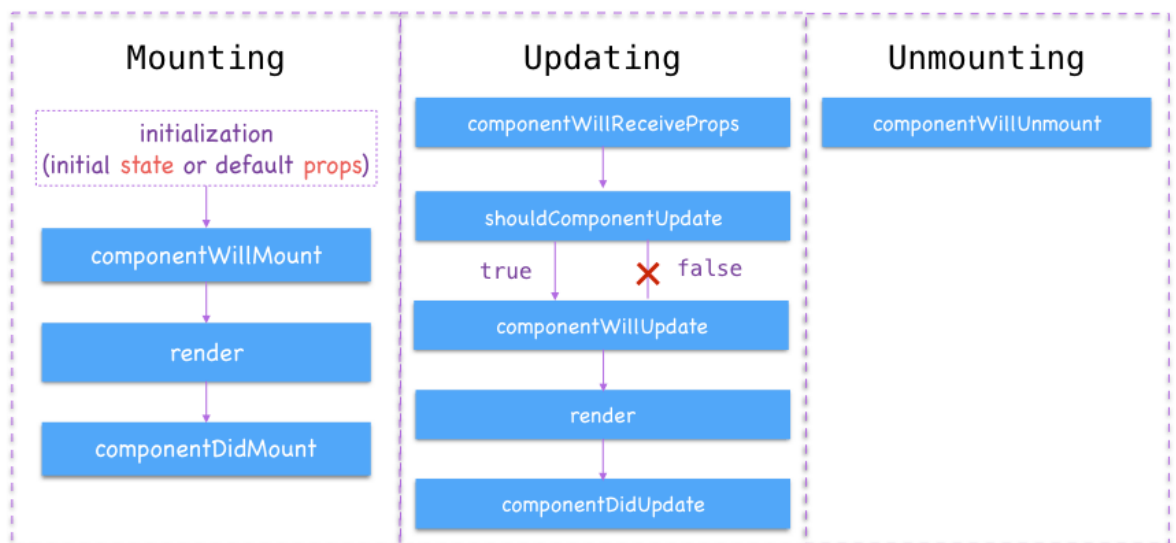


Figure 8. React component lifecycle. Copied from [11]

3.1.4 Redux

Following the documentation states, Redux is a predictable state container for JavaScript web applications. Redux was created by Dan Abramov in June 2015. It was motivated by Facebook's Flux and functional programming language Elm. Redux became popular

pretty fast thanks to its simplicity, small size (only 2 KB), and comprehensive documentation [12].

Most of the developers incorporate Redux with React, besides it can be implemented with any other client frameworks. For example, in several projects, they use Redux with AngularJS, Vue.js, or Meteor, although Redux going with React is still the most common combination [12].

Redux includes three main parts: actions, store, and reducers. Figure 9 below summarizes the Redux operation flow.

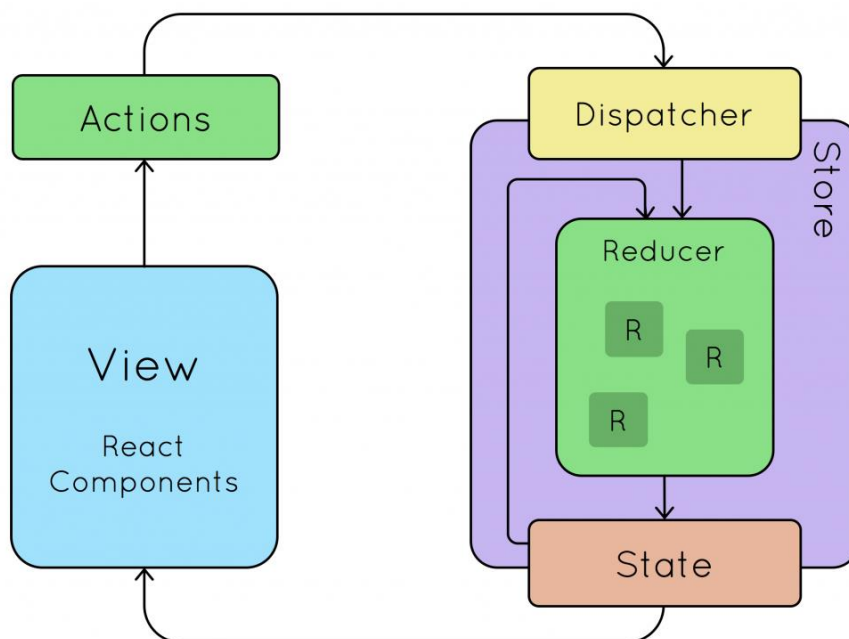


Figure 9. Redux operation flow. Copied from [13]

Actions are events that send data from the application (user interactions, internal events such as API calls) to the store [12]. Figure 10 below shows two actions that were implemented in the project to fetch data of offers list and shopping center. In this case, internal actions are JS objects that have constant (FETCH_OFFERS or FETCH_CENTER_DATA) as a type of property, defining what kind of action and payload of information being transferred to the store.

```

export function fetchOffers() {
  const request = axios.get(`${BASE_URL}/offers`);
  return {
    type: FETCH_OFFERS,
    payload: request
  };
}

export function fetchCenterData() {
  const request = axios.get(`${BASE_URL}/shoppingcenter`);
  return {
    type: FETCH_CENTER_DATA,
    payload: request
  };
}

```

Figure 10. Redux fetch action in the project.

Reducers are pure functions that define how the app state changes. The reducer function takes two arguments: the previous app state, the action being dispatched, and returns the new app state [12]. In this project (figure 11), reducers are functions that take the current state (with initial is a constant object named initialState) of the application and an action (imported from Redux action file) and then return a new state (one or all values in the initialState could be updated).

```

import { FETCH_OFFERS, FETCH_CENTER_DATA, FILTER_BY_BUSINESS } from '../actions';

const initialState = {
  offers: [],
  filteredOffers: [],
  selectedBusinessID: '',
  businessList: []
}

export default function (state = initialState, action) {
  switch (action.type) {
    case FETCH_CENTER_DATA:
      return { ...state, businessList: action.payload.data.business };
    case FETCH_OFFERS:
      return { ...state, offers: action.payload.data, filteredOffers: action.payload.data };
    case FILTER_BY_BUSINESS:
      return {
        ...state, filteredOffers: action.value !== '' ? state.offers.filter(offer => offer.businessId === action.value.id) : state.offers,
        selectedBusinessID: action.value.id
      };
    default:
      return state;
  }
}

```

Figure 11. Redux reducer in the project.

Store holds the state of the application. It is the object, not a class that provides some helper methods to enter the state, dispatch actions, and register listeners. A new state is returned via reducers when any action occurs. That makes Redux very simple and predictable [12].

3.2 NodeJS

NodeJS is a server-side platform that was found by Ryan Dahl in 2009. It is built on top of the V8 JavaScript runtime and used a non-blocking I/O (Asynchronous) model that makes it the best for data-intensive, real-time applications [15]. Being built on Google Chrome's V8 JavaScript Engine, NodeJS library is very quick in code performance [14].

NodeJS is regularly used to build back end services that interact with client-side applications. These applications get and send data through a back-end service called an API. The API works as an interface between many programs; therefore, they are able to connect to each other [14]. Presently, thanks to the perfection of NodeJS, JavaScript can be run on the server-side too [15]. The following Figure 12 shows a NodeJS server lifecycle.

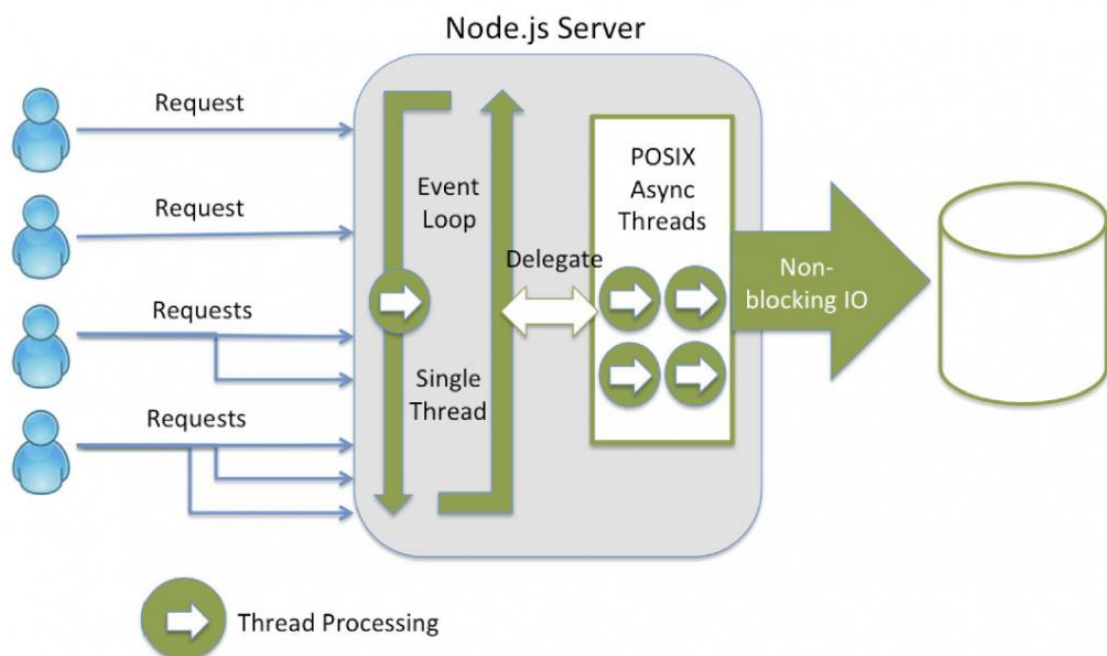


Figure 12. NodeJS server lifecycle. Copied from [16]

3.3 Bootstrap

As mentioned by the official website, Bootstrap is one of the most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web [17].

Since bootstrap is chosen by many developers and has many advantages, it's a great framework to learn and to work with. In this project, Bootstrap is quite easy to install and implement in React components. Below are some benefits that the framework can provide:

- Quickly and easily prototype new designs.
- Prevent duplication between projects.
- Make practical and effective use of responsive design to allow the webpages to work well on different screen sizes – mobile phones, desktop, tablets and other devices.
- Guarantee cross-browser compatibility.

4 Website development

This section includes a summary of the whole design process, which resulted in defining a set of goals related to the enhancement of the shopping center UI, implementing a single webpage that represents users' opinions from the research.

The primary job of the redesign process was defining issues. At this stage, I processed face-to-face interviews with some users at Sello, Iso Omena, and Itis. As a result, I learned more about users' behavior and the issues they faced when using the existing websites. The next step was to analyze the findings from the interview, then draw the wireframe.

Afterward, it was already possible to start developing and implementing a concept of changes together by thinking through every element to refresh in detail.

Usability testing is a very important step in any website redesigning process. As a result, it guaranteed users that the new webpage was improved rather than the current ones. It was necessary to evaluate the outcome with different devices (mobile, laptop) and to check if it performs well on different browsers.

4.1 Collecting background information

The purpose of the interview was to define the common users' views regarding the existing user interface and functionality, observe the ways they behave in the online environment and what their possible expectations on upcoming features. Additionally, the interview was processed to aim at collecting their suggestions and interests in some visual elements.

All the interviewees were customers who visited in shopping mall center. The interview took place in Iso Omena, Itis, and Sello, starting from 14 to 15 in December of 2019. Interviews were completed in the coffee shop in the shopping malls during the weekend at around 2-6 pm. In total, there were six customers (two in each center) that were four women and two men participated in the interview. Their age is around 20-40. All of them were chosen randomly by the author and were willing to spend 10-20 minutes interview while they were in the coffee shops. To keeping the condition casual and comfortable, the interview data was collected by taking notes and not be recorded any audio or video.

The overall topic of questions was related to user opinions towards the existing user interface of isoomena.fi, itis.fi, sello.fi, and their personal goals of usage. Three shopping center websites were used as a part of the interview. Furthermore, the purpose of the user questionnaire was to analyze the issues of the current website's functionality, usability and visualization, as well as to evaluate the project scope.

The first part of the interview was to focus on users' backgrounds and goals when using the website. This part stimulated me to learn more about the existing group of users from my perspective.

The second part of the interview focused on defining all the issues and shortcomings of the current user interface following users' views. Every interviewee shared their opinions

about the current content that could be structured in a more useful way or some existing features that need to be replaced or updated. This valuable information was applied in exploring innovative solutions to help redesign the user interface, feature, and structure of the website.

Below is the list of general questions used in the interview:

- Have you ever visited the website (Iso Omena, Itis, Sello)?
- Which devices (laptop, phone, tablet) do you often use to surfing web?
- What information do you search for on the website?
- Is it easy to find what you are looking for?
- Which pages makes you feel the most interested in and the most useful when surfing the website?
- Can you show me how to get these landing pages?
- What do you think about these landing pages? (font size, color, content...)
- Do you have any ideas to make the pages look better or to make you feel more convenient and comfortable to use them?

Background information found from the interview:

- All of them have visited the website for more than one.
- One person uses only mobile to surfing web (or use shopping center application). All others use both laptop and mobile, but mobile is more often.
- Opening hours of shopping malls, opening hours of stores in malls, and especially exceptional opening hours are the most common information that they search for. All of them found it quite easy to get the necessary info. In Itis website, this info is placed

right on the home page. In Iso Omena and Sello websites, there are buttons on the homepage link to the opening hour page.

- The other common information that they are interested in is sales, discounts, offers, or campaigns. Offers pages in three websites are pretty straightforward to navigate to thanks to the links on the main menus.
- In Sello offers page, offer cards show quite adequately information such as offer prices, store names, offer images, offer validation. The good point is to offer prices that are highlighted by red color. One of the comments to make the page looks better is should have the filters by categories or search field. Another one is maybe smaller offer cards on mobile (two offers in a row) to reduce scrolling so much.
- In Itis offer page, almost everything looks good (offer prices, store names, offer images, offer valid date). However, the offer card is quite boring because of only black and white. Additionally, the text description of the offer is too long that should be implemented as collapsed elements.
- In Iso Omena offer page, it has good font size, color theme, and clear main content. A great feature is about pop-up when clicking on the offer card that shows full of offer information. However, missing the valid offer date make users a bit unpleasant. Having a search box or filters to view only their attractive offers is one of the suggestions during the interview. The other idea is to show offers list into small size cards because there is the pop-up to present all the information.
- One unique concern from one interviewee in Iso Omena is about the way to parking and parking benefits.

4.2 Wireframing

Once collecting background information via the interview was completed, I started to begin building the wireframe. Since offer pages are the most common information that users concerned about, I decided to set my thesis project for enhancing UI and features of the offer page. Wireframing has initially been indicated for creating first-class webpage prototypes, using generally realistic scales of visual elements. See figures 13-15 below:

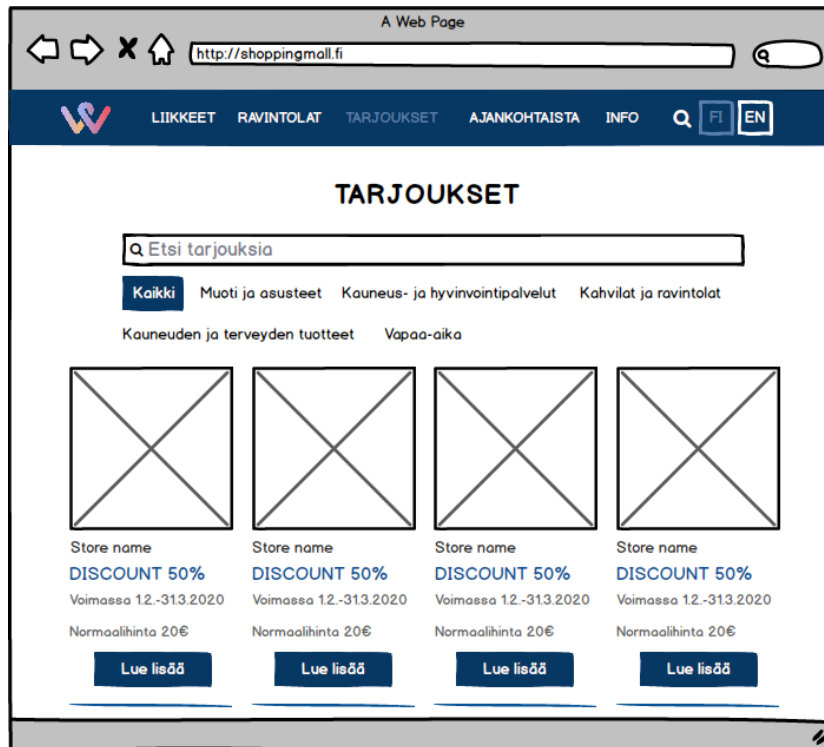


Figure 13. Offers page wireframe on desktop.

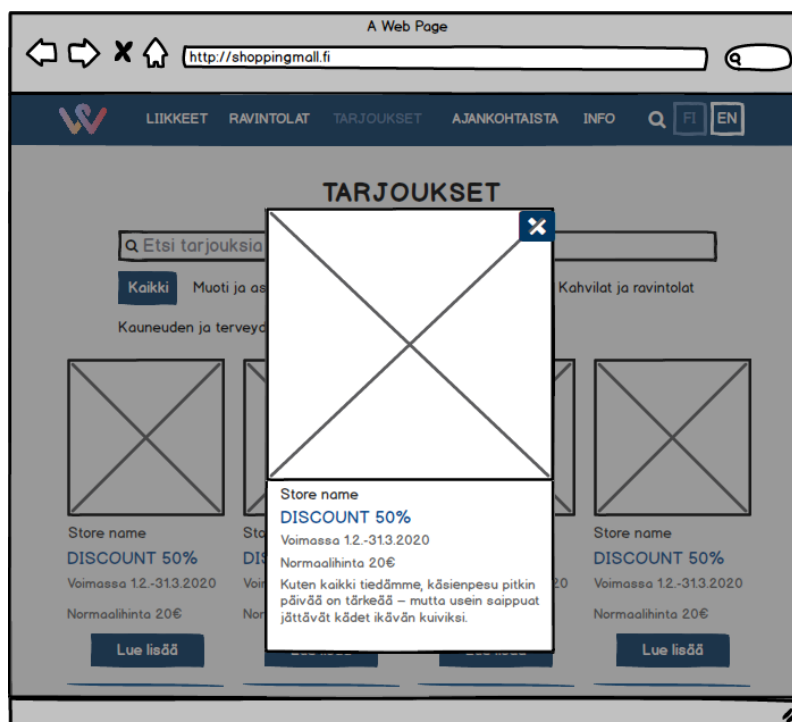


Figure 14. Pop-up offer card wireframe on desktop.

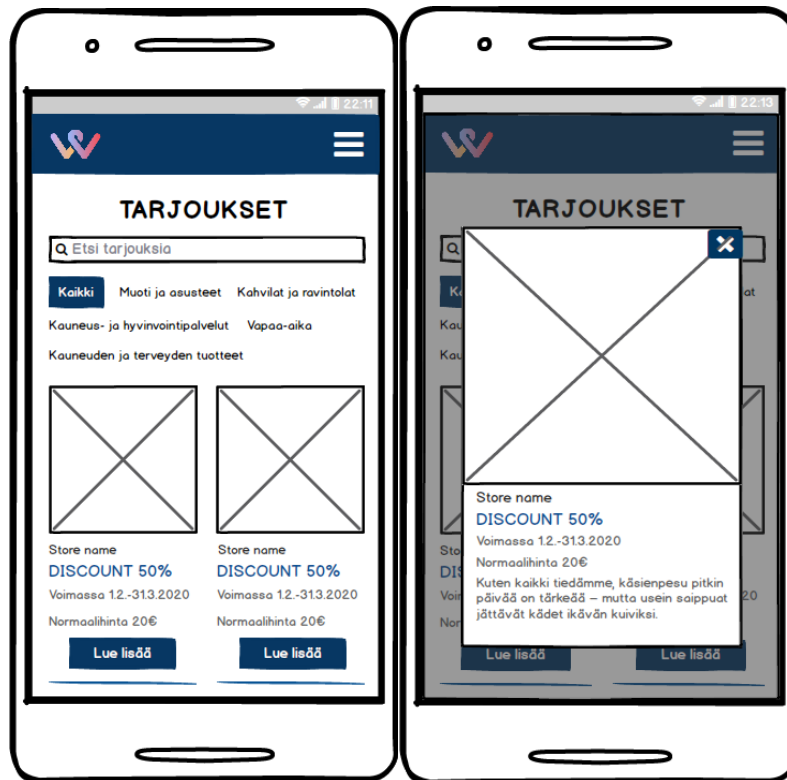


Figure 15. Offers page and pop-up offer card wireframe on mobile.

More in detail of the wireframe process, the header includes the logo, list of navigation items, search icon, language selectors. The header background color in dark color creates high contrast with all the included content in white. The landing page and selected language are applied in a different color to help users clearly know where they are on the website. On the mobile view, list of navigation items is designed to be collapsed in the hamburger icon (indicating to open/close menu button).

The title of the page is in the biggest font-size. Below that is the input with the placeholder "Etsi tarjouksia" (search for offers) on the left and the search icon on the right. Thanks to this, it's easy to understand that input field is used for searching offers function. Another feature in the wireframe that comes from the user's suggestions in the interview is the filter. The function of the filter assists users to view their interesting field of business. Moreover, the selected category is highlighted by white text with dark background-color that looks similar to a button, supporting users recognize the filter is clickable.

The main content of the page is a list of offers. An offer card includes the offer image, store name, offer title, valid date, offer sub-title, show more button, and the line break. Thanks to some feedback from the interview, besides the offer image, the most important information is the offer title that is set in bigger font-size, uppercase text, unique color when compared to other information in a card. As a result of two comments from the interview, which are about complaining too long description of the offer and great pop-up offer feature when clicking on the offer card, the show-more button is created to make the offer card shorter. The other findings from the interview that need to be considered in the wireframing process is about inconvenient scrolling page on mobile. To reducing this uncomfortable, the offers are displayed four in a row on desktop and two in a row on mobile.

Thanks to the wireframing tool, it was simple and straightforward to adjust the scales of visual elements, such as pop-up, buttons, and icons. Figure 13, 14 indicates the offers page and pop-up offer card wireframing on desktop. Figure 15 shows the wireframing mobile version of the offers page and pop-up offer card. More visual elements were implemented during the rest of the project process. As a result, wireframing helped me to save plenty of time.

4.3 Website implementation

4.3.1 Data structure

Figure 16 below illustrates tables of data structure that gives a general view of storing, connecting, and organizing information data in the project.

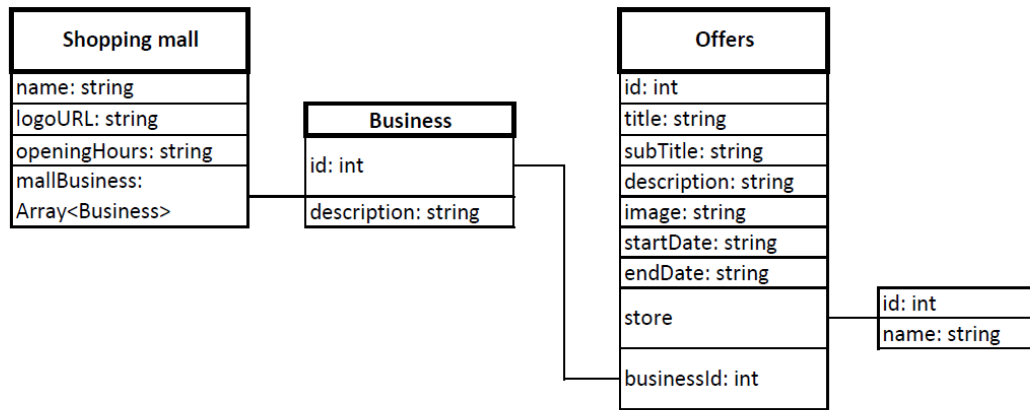


Figure 16. Data structure.

```

{"shoppingCenter": {
  "name": "Shopping center W",
  "logoURL": "https://someDomain.com/logo.png",
  "openingHours": "Maanantai-Perjantai 09-21, Lauantai 09-19, Sunnuntai 10-19",
  "business": [
    {"id": 1, "description": "Muoti ja asusteet"},
    {"id": 2, "description": "Kahvilat ja ravintolat"},
    {"id": 3, "description": "Kauneuden ja terveyden tuotteet"},
    {"id": 4, "description": "Sisustaminen ja kodin tarvikkeet"},
    {"id": 5, "description": "Vapaa-aika"}
  ]
}
}

```

Listing 3. JSON data of shopping mall center.

```

{"offers": [
  {
    "id": 1
    "title": "Discount 50%",
    "subTitle": "Clearance sale",
    "description": "this is description",
    "image": "https://somedomain.com/17164/IMG_8387.JPG",
    "startDate": "2020-02-01",
    "endDate": "2020-04-30",
    "store": {"id": 16, "name": "Storename 16"},
    "businessId": 5,
  }
]}

```

Listing 4. JSON data of offers.

4.3.2 Color palette and font



Figure 17. Color palette with codes: primary #062A62 and secondary #3D71C3.

Since each shopping mall centers have their color schemes, the color palette in this project was chosen by the author's favorite. As the figure 17 represents above, the primary color is dark blue with code #062A62, and the secondary one is blue with code #3D71C3.

The primary color goes to the “hot spots” on the web page that is used to attract users' consciousness and assist them in taking action. There are many elements that are applied to the primary color to highlight important information, such as buttons, headlines, icons.

To highlight less important information on the website, such as secondary buttons, secondary icons, subheadings, active navigation items, or supporting content, the secondary color is used.

Other colors are black, white, and gray that known as the most common neutral colors. These type of shades are mostly used for text, paragraph to create contrast for essential elements such as a button with white color and blue background color. The use of neutral color in this project is much similar to white space that helps to provide users a break and allows them time to better digest and prioritize information on the page. For example, the main information of an offer card is the offer title that is highlighted by blue. The other content of the offer, like sub-title or description, is in black.

When searching for the ideal web safe font - means that users don't have to have any particular font installed on their devices to see it correctly; it is often the Google Fonts library with over 800 free licensed fonts. Google Fonts library is one of the best places for finding web safe fonts. Additionally, it ensures that the web page appearance stays uniform across all browsers and devices.

Extra-Light
Light
Regular
Medium
Semi-Bold
Bold
Extra-Bold

Figure 18. Different styles of Dosis font.

Figure 18 shows different styles of Dosis font. It was selected for the project by author's idea. Dosis font style looks good on screen, easy to use, and suitable for a majority of browsers.

4.3.3 Webpage implementation

In the header, as can be seen in figure 19 for the desktop version and figure 20 for the tablet and mobile view, the primary color is applied as background color; the logo is located at the left corner. In the middle area of the header part (desktop view), there is a list of navigation items that are in white text color and in secondary blue color when it is selected or hovering. On the mobile view, navigation items are collapsed and only shown when clicking on the hamburger icon on the right corner. The same color discipline is applied to the language selectors in the right corner of the header.



Figure 19. Header and the first part of offers page on desktop view.



Figure 20. Header and the first part of offers page on mobile view.

The other components are shown in figure 19, 20, the main title of the page "TARJOUKSET" and the search box. The main title element is a `<h1>` with black color, font-weight 500, font-size 45px. The search box has the primary color border and search

icon on the right with the same color. The feature of the search hasn't been done in this project and be mentioned for further development.

Below the search box is the list of offers categories. The filter by the line of business is the main feature that completes the existing general offers pages. Users can filter their interesting offers by choosing one of the lines of business category, which is highlighted by the primary blue background-color. The same style of white text and blue background is presented when hovering. The result of the offers categories created as a React component can be viewed in figure 21.

```

class OffersList extends Component {
  componentDidMount() {
    this.props.fetchOffers();
    this.props.fetchCenterData();
  }

  renderBusinessList(business, index) {
    const { selectedBusinessID } = this.props.offersPage;
    const className = `offers-category ${business.id === selectedBusinessID ? 'active' : ''}`;
    return (
      <div className={className} key={index}>
        <p className="offers-category-name" onClick={() => this.props.selectBusiness(business)}>
          {business.description}
        </p>
      </div>
    );
  }
}

```

Figure 21. Offers categories component.

The main content of the page is a list of offers; each of them is called an offer card. An offer card includes the offer image, store name, offer title, valid date, offer sub-title, show more button, and the line break. The advanced feature in the offer card when compared to the one in Iso Omena, Sello, Itis is supporting all sizes of the image that was implemented by using the blur background image to full fill the original image into a square shape (see in figure 22). Store name, sub-title, valid date are in black color, font-size 16px, font-weight normal. Offer title is highlighted by bigger font-size (18px), secondary blue color, text uppercase. Show more button is in white uppercase text, primary blue background, and secondary blue background when hovering. The line break indicates the ending of an offer card, which is 2px height and in secondary blue color.

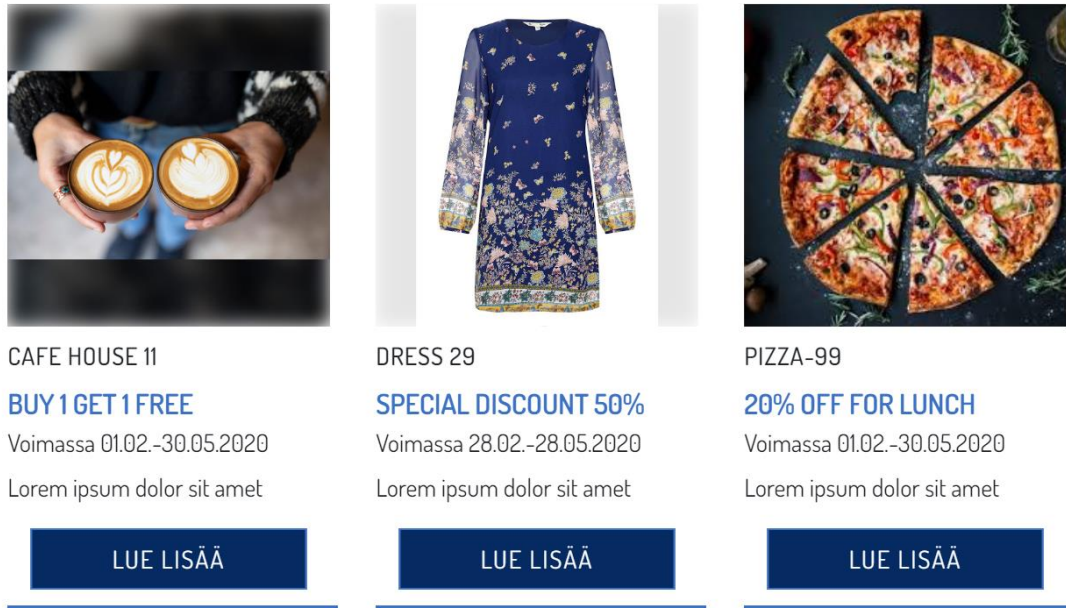


Figure 22. Offer card.

The pop-up offer is shown when users click on the show more button, looks similar to the offer card. Some differences are the offer description represent instead of the show more button, and a close pop-up button on the right corner instead of the line break. The React component of the offer card can be viewed in the figure 23.

```
renderOffers(offer, index) {
  const background = { backgroundImage: `url(${offer.image})` },
    modalTarget = '#offer-' + index, popupID = 'offer-' + index;
  return (
    <div className="col-6 col-md-4 col-lg-3 mb-5" key={index}>
      <div className="offer-image">
        <div className="offer-image-blur" style={background}></div>
        <img src={offer.image}></img>
      </div>
      <div className="offer-info">
        <h5 className="offer-store-name">
          {offer.store.name}
        </h5>
        <p className="offer-title">{offer.title}</p>
        <p className="offer-valid-date">Voimassa {this.formatDate(offer.startDate, offer.endDate)}</p>
        <p className="offer-subtitle">{offer.subTitle}</p>
        <div className="show-more-btn" data-toggle="modal" data-target={modalTarget}>Lue lisää</div>
      </div>
      <hr className="line-break"></hr>
    </div>
  );
}
```

Figure 23. Offer card component.

4.3.4 Outcome and screenshots

Figure 24 below shows the screenshot of the offers page version on desktop, which includes a static header, page title, search box, filter by the line of business, and a list of offers (four offers in a row).

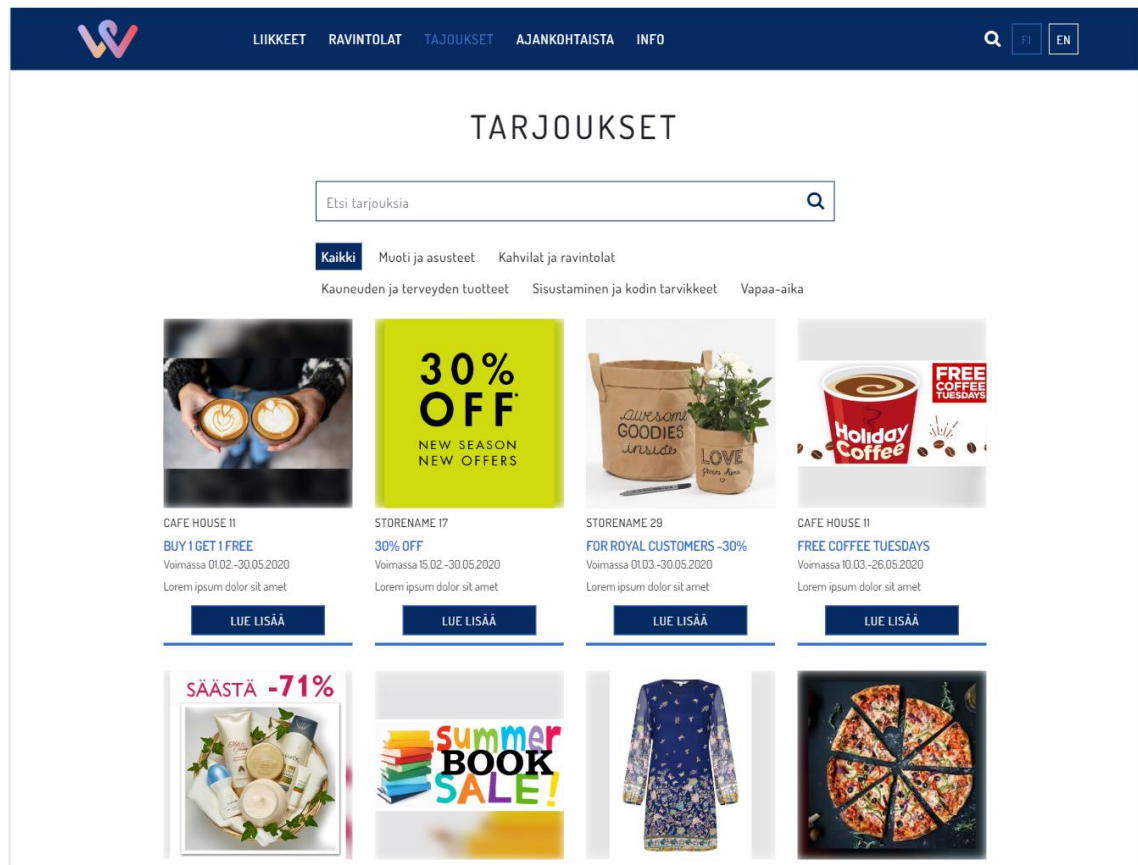


Figure 24. Offers page on desktop.

Figure 25 displays the offers page with an offer popup when users click show-more button version on desktop.

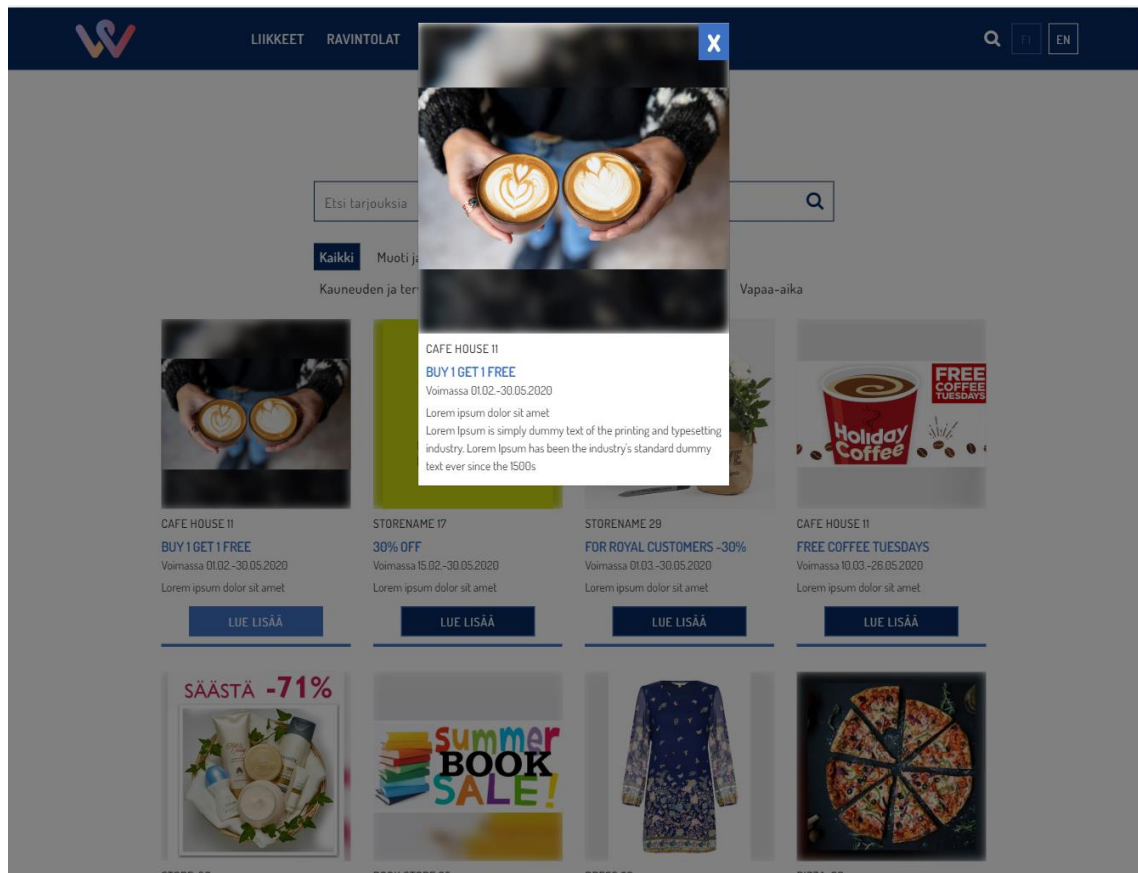


Figure 25. Offer pop-up on desktop.

Figure 26 below presents the screenshots of the offers page and an offer popup version on tablet. This page includes a static header (navigation items are collapsed by hamburger icon), page title, search box, filter by the line of business, and a list of offers (three offers in a row).

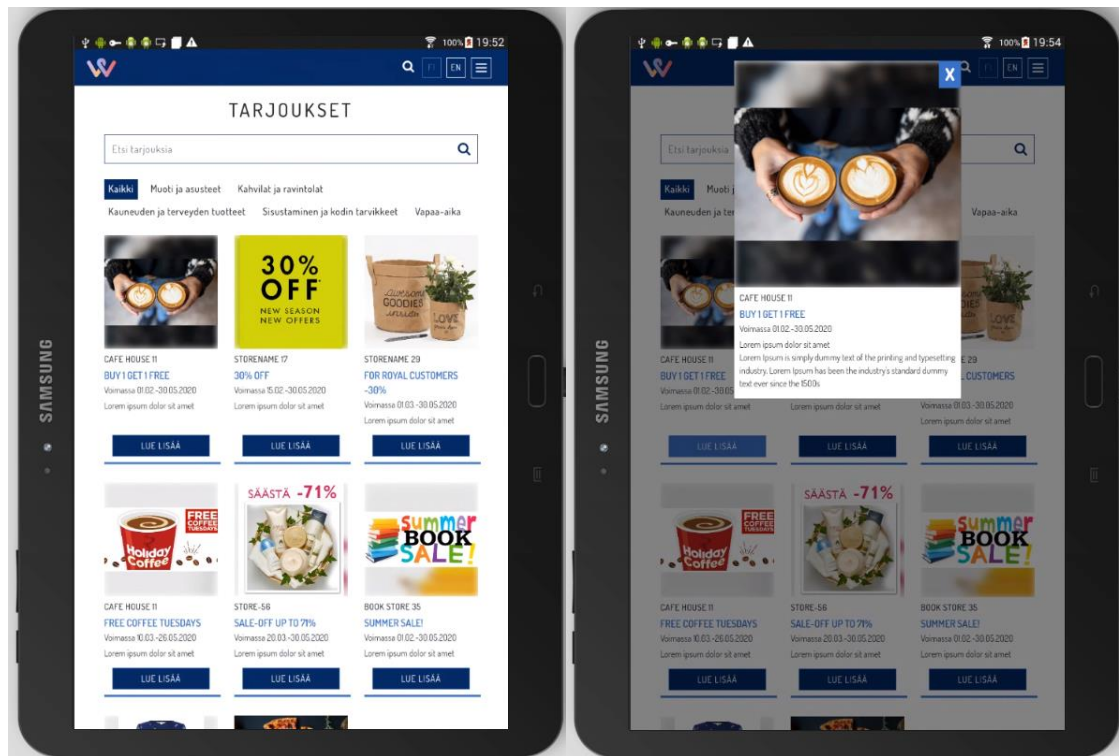


Figure 26. Offers list and offer pop-up on tablet.

Figure 27 below presents the screenshots of the offers page and an offer popup version on mobile. This version is quite similar to the one on tablet; there is only one difference in displaying a list of offers (two offers in a row).

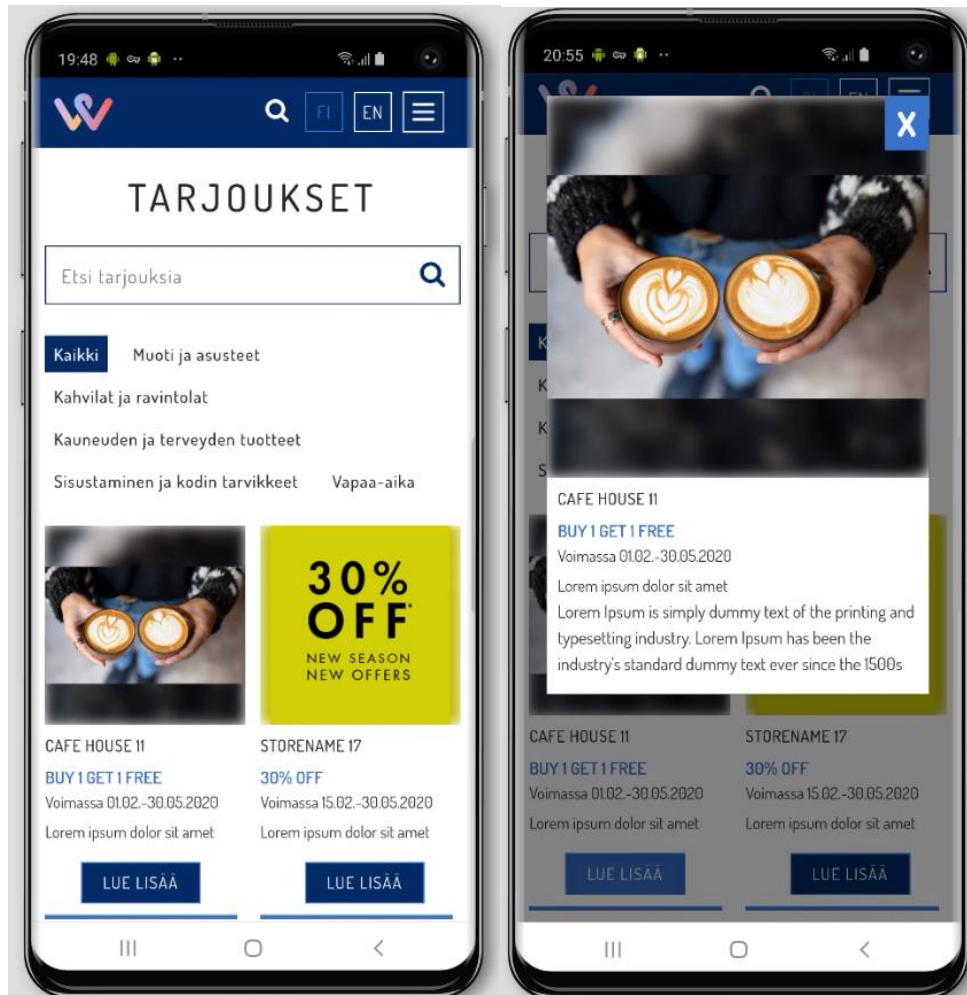


Figure 27. Offers list and offer pop-up on mobile.

4.4 Testing and feedback

Before processing user testing, I have tested the page on different browsers and devices to see if the page was working well. It looks good without any styling bugs as I expected on Chrome, IE11, Firefox, Safari, Android phones, and tablets.

To understand how others see the webpage and how others see the usability of the site, I found a few friends to test the re-designed webpage. Due to the outbreak situation, I couldn't visit shopping centers and process the user testing there. Therefore, three friends living in the same building, who are all students (three men), age 25-30, were invited to my house for the user testing. Since the testing was conducted at home, it was comfortable for them to test and give feedback. It took around 30 minutes for each user testing. The testing data was collected by taking notes. The method of testing was to open the three offers pages of Iso Omena, Itis, Sello, and asked them to give comments while going through the web pages. After that, the new offers page was opened. I asked them to use and give comments while observing their behavior and actions made during the testing. Suggestions of what was missing were called as well as if there was something the user wanted to find but could not, or if there were some tasks they found difficult to perform.

The overall response was quite positive. They were all actually like the new offers page with the categories. The other good feedback was offer cards showing adequate information such as offer images, store names, valid dates, and so forth. Additionally, the offer popup looked really nice in case some users want to see the full description of offers. Besides, some great comments were given to the color theme and the responsive on different devices. However, the search feature and change page to English were missing. Since they are all foreigners, they were trying to click the language selector in the header to change to English, but that did not happen. Another feedback was about the missing function of the search box that also hasn't been implemented. The feedback suggested that store names or offer titles should be added into the search feature.

5 Conclusions

The purpose of this thesis was to enhance and redesign the shopping center web application using React, Redux framework, and NodeJS. Moreover, this study was conducted to demonstrate the outcome of the author's learning of React, Redux, and NodeJS.

At the first stage, the interview was conducted to understand usability problems in the existing shopping center websites (Iso Omena, Itis, Sello). It provided valuable

information about the user's behaviors, needs, and suggestions. The information from the interview was taken into account and was helpful when defining which page of the website can be improved. According to the results of usability evaluation, wireframes were created; they help to make choices about how to build a webpage structure and decide the suitable place for different visual elements.

The next step was the webpage implementation. A REST web service was built to provide data to the application. Visual design creation was based on choosing suitable fonts, color palette, and interaction details. ReactJS with Redux, Bootstrap was chosen as the framework for frontend due to author's eagerness to learn new technology.

The final result was a redesigned offers page, which is fit for the requirements of UX. According to user testing conducted at the end of the project, it can be concluded that the new offers page usability was improved and meet the user's needs. Finally, missing the search function in the offers page will be something for further development to make the experience more enjoyable for the users.

References

1. What Is User Experience (UX) Design? Everything You Need To Know To Get Started [Internet]. Careerfoundry.com. 2020 [cited 22 February 2020]. Available from: <https://careerfoundry.com/en/blog/ux-design/what-is-user-experience-ux-design-everything-you-need-to-know-to-get-started>
2. [Internet]. Fatrabbitcreative.com. 2020 [cited 22 February 2020]. Available from: <https://www.fatrabbitcreative.com/blog/what-is-user-experience-design>, <https://www.smashingmagazine.com/2010/10/what-is-user-experience-design-overview-tools-and-resources/>
3. The Comprehensive Guide to Information Architecture [Internet]. Toptal Design Blog. 2020 [cited 1 March 2020]. Available from: <https://www.toptal.com/designers/ia/guide-to-information-architecture>
4. The Five Principles of Interaction Design [Internet]. Info.obsglobal.com. 2020 [cited 2 March 2020]. Available from: <https://info.obsglobal.com/blog/2014/08/the-five-principles-of-interaction-design>
5. 5 Principles of Visual Design in UX [Internet]. Nielsen Norman Group. 2020 [cited 2 March 2020]. Available from: <https://www.nngroup.com/articles/principles-visual-design/>
6. Usability 101: Introduction to Usability [Internet]. Nielsen Norman Group. 2020 [cited 7 March 2020]. Available from: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
7. What Is A User Interface, And What Are The Elements That Comprise one? [Internet]. Careerfoundry.com. 2020 [cited 8 March 2020]. Available from: <https://careerfoundry.com/en/blog/ui-design/what-is-a-user-interface/>
8. What and Why React.js [Internet]. C-sharpcorner.com. 2020 [cited 13 March 2020]. Available from: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>

9. The difference between Virtual DOM and DOM [Internet]. React Kung Fu. 2020 [cited 14 March 2020]. Available from: <https://reactkungfu.com/2015/10/the-difference-between-virtual-dom-and-dom/>
10. Understanding React Components [Internet]. Medium. 2020 [cited 14 March 2020]. Available from: <https://medium.com/the-andela-way/understanding-react-components-37f841c1f3bb>
11. Posts B. React.js Tutorial: React Component Lifecycle [Internet]. Codevoila.com. 2020 [cited 14 March 2020]. Available from: <https://www.codevoila.com/post/57/reactjs-tutorial-react-component-lifecycle/>
12. Redux - An Introduction — Smashing Magazine [Internet]. Smashing Magazine. 2020 [cited 15 March 2020]. Available from: <https://www.smashingmagazine.com/2016/06/an-introduction-to-redux/>
13. React Redux: Building Modern Web Apps with the ArcGIS JS API [Internet]. ArcGIS Blog. 2020 [cited 15 March 2020]. Available from: <https://www.esri.com/arcgis-blog/products/3d-gis/3d-gis/react-redux-building-modern-web-apps-with-the-arcgis-js-api/>
14. Node.js - Introduction - Tutorialspoint [Internet]. Tutorialspoint.com. 2020 [cited 18 March 2020]. Available from: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm
15. What is NodeJS and Why You need to learn it [Internet]. Medium. 2020 [cited 18 March 2020]. Available from: <https://medium.com/@michaelhenderson/what-is-nodejs-and-why-you-need-to-learn-it-f0760ba9a76a>
16. node.js: An Introduction for Decision Makers | Blog - Number8 [Internet]. number8. 2020 [cited 18 March 2020]. Available from: <https://number8.com/node-js-introduction-decision-makers/>
17. Mark Otto a. Bootstrap [Internet]. Getbootstrap.com. 2020 [cited 19 March 2020]. Available from: <https://getbootstrap.com>

18. Hassenzahl M, Tractinsky N. User experience - a research agenda. Behaviour & Information Technology. 2006;25(2):91-97.