

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Bakalářská práce**

**User experience webové aplikace**

**Tomáš Kubát**

**© 2023 ČZU v Praze**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tomáš Kubát

Informatika

Název práce

**User experience webové aplikace**

Název anglicky

**Web application user experience**

---

### Cíle práce

Hlavní cíl práce je zhodnocení současného stavu user experience vytvořené webové aplikace na základě uživatelského testování a návrh vhodných zlepšení.

Dílní cíle práce jsou:

- Analýza dostupných odborných informačních zdrojů v oblasti user experience
- Vytvoření prototypu aplikace pro otestování uživateli
- Návrh testovacího scénáře, zpracování výsledků uživatelského testování

### Metodika

Práce je rozdělena na dvě části, teoretickou a praktickou.

Teoretická část práce se zabývá analýzou odborné literatury, která se vztahuje k vývoji webových aplikací a user experience.

V praktické části bude popsán vývoj prototypu webové aplikace. Dále bude vytvořen testovací scénář pro uživatelské testování. Na základě zpracování výstupních dotazníku budou navrženy změny s cílem zvýšení celkové user experience webové aplikace.

## Doporučený rozsah práce

40-50

## Klíčová slova

user experience, webová aplikace, uživatelské testování, testování použitelnosti

---

## Doporučené zdroje informací

ECCHER, C. Profesionální webdesign : techniky a vzorová řešení. Brno: CP Books, 2005. ISBN 80-251-0547-4.

HARTSON, Rex a Pardha RYLE. UX book. United States: Morgan Kaufmann, 2012. ISBN 0123852412.

KRUG, S. *Nenuťte uživatele přemýšlet! : praktický průvodce testováním a opravou chyb použitelnosti webu*. Brno: Computer Press, 2010. ISBN 978-80-251-2923-4.

TIELENS THOMAS, Mark. React in Action. Shelter Island: Manning Publications, 2018. ISBN 978-1617293856.

---

## Předběžný termín obhajoby

2022/23 LS – PEF

## Vedoucí práce

Ing. Jan Pavlík

## Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 14. 7. 2022

**doc. Ing. Jiří Vaněk, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 27. 10. 2022

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 12. 03. 2023

## **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "User experience webové aplikace" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2022

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Janu Pavlíkovi, Ph.D. za ochotu a cenné rady, které byly poskytnuty při vedení této bakalářské práce. Zároveň bych chtěl poděkovat své rodině za podporu při studiu.

# User experience webové aplikace

## Abstrakt

Cílem bakalářské práce, je vytvořit webovou aplikaci a tuto aplikaci otestovat uživateli. Testování bude probíhat, podle předem stanoveného scénáře z testování budou učiněny závěry navrhnuty úpravy aplikace. Práce je rozdělena na dvě části, teoretickou a praktickou.

Teoretická část práce se zabývá analýzou odborné literatury, která se vztahuje k vývoji webových aplikací a user experience.

V praktické části bude popsán vývoj prototypu webové aplikace. Dále bude vytvořen testovací scénář pro uživatelské testování. Na základě zpracování výstupních dotazníku budou navrženy změny s cílem zvýšení celkové user experience webové aplikace.

**Klíčová slova:** user experience, webová aplikace, uživatelské testování, testování použitelnosti

# **Web application user experience**

## **Abstract**

The aim of the bachelor thesis is to create a web application and test this application with users. Testing will take place, according to a predetermined scenario, conclusions will be drawn from the testing and proposed adjustments to the application will be made. The work is divided into two parts, theoretical and practical.

The theoretical part of the work deals with the analysis of professional literature related to the development of web applications and user experience.

The practical part will describe the development of a web application prototype. Next, a test scenario will be created for user testing. Based on the processing of the output questionnaire, changes will be proposed with the aim of increasing the overall user experience of the web application.

**Keywords:** user experience, web application, user testing, usability testing

# Obsah

<b>1 Úvod.....</b>	<b>10</b>
<b>2 Cíl práce a metodika .....</b>	<b>11</b>
2.1 Cíl práce .....	11
2.2 Metodika .....	11
<b>3 Teoretická část.....</b>	<b>12</b>
3.1 Webové aplikace .....	12
3.1.1 Webové aplikace oproti Webové stránce.....	12
3.1.2 Přístupy k tvorbě webových aplikací.....	12
3.2 Návrh aplikace .....	14
3.2.1 Typografie.....	14
3.2.2 Barvy.....	14
3.2.3 Přístupnost .....	16
3.3 Webové technologie.....	19
3.3.1 HTML .....	19
3.3.2 CSS .....	20
3.3.3 JavaScript.....	22
3.3.4 ECMAScript .....	22
3.3.5 Organizace kódu do komponentů .....	22
3.3.6 Imperativní a deklarativní způsoby UI .....	23
3.3.7 React .....	23
3.3.8 Next.js .....	24
3.4 Přístupy vývoje softwaru.....	24
3.4.1 Vodopádový model.....	25
3.4.2 Agilní vývoj .....	25
3.4.3 Vývoj řízený testy .....	25
3.4.4 Lean vývoj .....	25
3.5 Testování aplikací .....	25
3.5.1 Definice UX.....	25
3.5.2 Definice UI .....	26
3.5.3 Definice pojmu Usability .....	26
3.5.4 Rozdíl mezi pojmy UX a UI.....	26
3.5.5 Proč testovat UX.....	26
3.5.6 Testování.....	27
3.5.7 Typy testování.....	27
<b>4 Praktická část .....</b>	<b>29</b>
4.1 Popis prototypu aplikace .....	29
4.1.1 Požadavky aplikace.....	29



4.1.2	Struktura aplikace a její náležitosti .....	30
4.1.3	Požadavky jednotlivých zobrazení .....	30
4.1.4	Cviky a jejich kategorie v aplikaci .....	31
4.1.5	Architektura aplikace .....	32
4.1.6	Barvy v aplikaci .....	32
4.2	Tvorba prototypu.....	32
4.2.1	Customizace MUI.....	32
4.2.2	Nastavení typografie .....	33
4.2.3	Úprava vzhledu komponentů z MUI .....	33
4.2.4	Vývoj základních komponent .....	34
4.2.5	Práce s knihovnou react-hook-form-mui .....	34
4.2.6	Práce s načítáním v aplikaci.....	34
4.2.7	Vývoj jednotlivých zobrazení.....	35
4.2.7.1	Přihlášení .....	35
4.2.7.2	Registrace .....	36
4.2.7.3	Zapomenuté heslo.....	38
4.2.7.4	Nástěnka tréninků.....	39
4.2.7.5	Tvorba tréninku .....	41
4.3	Testování.....	44
4.3.1	Funkce podrobeny testování .....	45
4.3.2	Testovací scénář.....	46
4.3.3	Výsledná zpráva.....	46
4.4	Zhodnocení testování .....	47
4.4.1	Analýza chyb v UX.....	48
4.4.2	Řešení nalezených chyb.....	48
<b>5</b>	<b>Zhodnocení a doporučení .....</b>	<b>53</b>
<b>6</b>	<b>Závěr.....</b>	<b>54</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>55</b>
<b>8</b>	<b>Seznam obrázků, tabulek, grafů a zkratek.....</b>	<b>59</b>
8.1	Seznam obrázků .....	59
8.2	Seznam tabulek .....	59
8.3	Seznam grafů.....	60
8.4	Seznam použitých kódů .....	60
8.5	Seznam použitých zkratek.....	60
<b>Přílohy.....</b>		<b>61</b>
	Příloha 1 Testovací scénář klient .....	61
	Příloha 1 Testovací scénář trenér .....	62

# 1 Úvod

Webové aplikace se staly nedílnou součástí našeho života. Využíváme je k řadě činností, od správy osobních financí až po plánování firemních akcí nebo komunikaci s rodinou. Vývoj webových aplikací je složitý proces, ale samotná tvorba aplikace není dostačující pro její úspěch. Je nutné, aby uživatelé byli spokojeni s používáním aplikace a měli pozitivní zkušenost s celým procesem. Proto je user experience dnes považován za klíčový faktor pro úspěšnost webových aplikací.

User experience zahrnuje mnohem více než jen pouhé testování. Do něj patří také kategorie jako je navigace, grafický design, interakce, obsah, funkčnost a přístupnost. Všechny tyto kategorie by měly být brány v potaz při návrhu webové aplikace a při jejím následném testování, aby byla aplikace pro uživatele co nejprívětivější.

Věnovat se user experience není pouze otázkou zlepšení uživatelského rozhraní, ale také otázkou přemýšlení o uživateli a jeho potřebách. Proto bude v této práci kladen důraz na uživatelskou perspektivu a bude se snaženo co nejlépe pochopit potřeby uživatelů a vytvořit aplikaci, která bude co nejvíce odpovídat jejich očekáváním a potřebám.

V této bakalářské práci budou všechny tyto body propojeny za účelem návrhu webové aplikace s cílem co nejlépe vyhovět uživatelským potřebám. Následné testování aplikace umožní identifikovat případné chyby, tyto chyby budou analyzovány a u každé chyby bude uvedeno řešení, které by mohlo chybu eliminovat.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Hlavní cíl práce je zhodnocení současného stavu user experience vytvořené webové aplikace na základě uživatelského testování a návrh vhodných zlepšení.

Dílní cíle práce jsou:

- Analýza dostupných odborných informačních zdrojů v oblasti user experience
- Vytvoření prototypu aplikace pro otestování uživateli
- Návrh testovacího scénáře, zpracování výsledků uživatelského testování

### **2.2 Metodika**

Práce je rozdělena na dvě části, teoretickou a praktickou.

Teoretická část práce se zabývá analýzou odborné literatury, která se vztahuje k vývoji webových aplikací a user experience.

V praktické části bude popsán vývoj prototypu webové aplikace. Dále bude vytvořen testovací scénář pro uživatelské testování. Na základě zpracování výstupních dotazníku budou navrženy změny s cílem zvýšení celkové user experience webové aplikace.

## **3 Teoretická část**

### **3.1 Webové aplikace**

Webová aplikace představuje software, který je zprostředkovaný internetovým prohlížečem. Na rozdíl od desktopových či mobilních aplikací, nemusí být webové aplikace fyzicky instalovány, a proto nezabírají místo na disku. K otevření webové aplikace potřebuje koncový uživatel pouze webový prohlížeč. Webové aplikace, díky tomu, že nejsou vázány na konkrétní operační systém, zaznamenávají v poslední době velký úspěch. A jsou častěji preferovány vývojáři, protože při vývoji mobilních či desktopových aplikací jsme vázáni na platformu, na které bude aplikace spouštěna (Kod'ousková 2021).

#### **3.1.1 Webové aplikace oproti Webové stránce**

Pod pojmem webová stránka si můžeme představit jednoduchou webovou prezentaci, kteréhokoliv řemeslníka. Tedy webová stránka je primárně určena k předání informací či propagaci firmy. Obecně tedy webová stránka zobrazuje statický obsah. Webové aplikace se již ze slova aplikace více zaměřují na interaktivitu. Poskytují nám nějakou funkcionalitu, sdílení dokumentů kontaktní formuláře, rezervační systém kina a další (Kod'ousková 2021).

#### **3.1.2 Přístupy k tvorbě webových aplikací**

V současné době se používají dva přístupy tvorby webových aplikací Single-Page Application (dále jen SPA) a Multi-Page Application (dále jen MPA). Buď se využívá jeden nebo druhý přístup, nebo jsou tyto dva přístupy kombinovány.

##### **MPA**

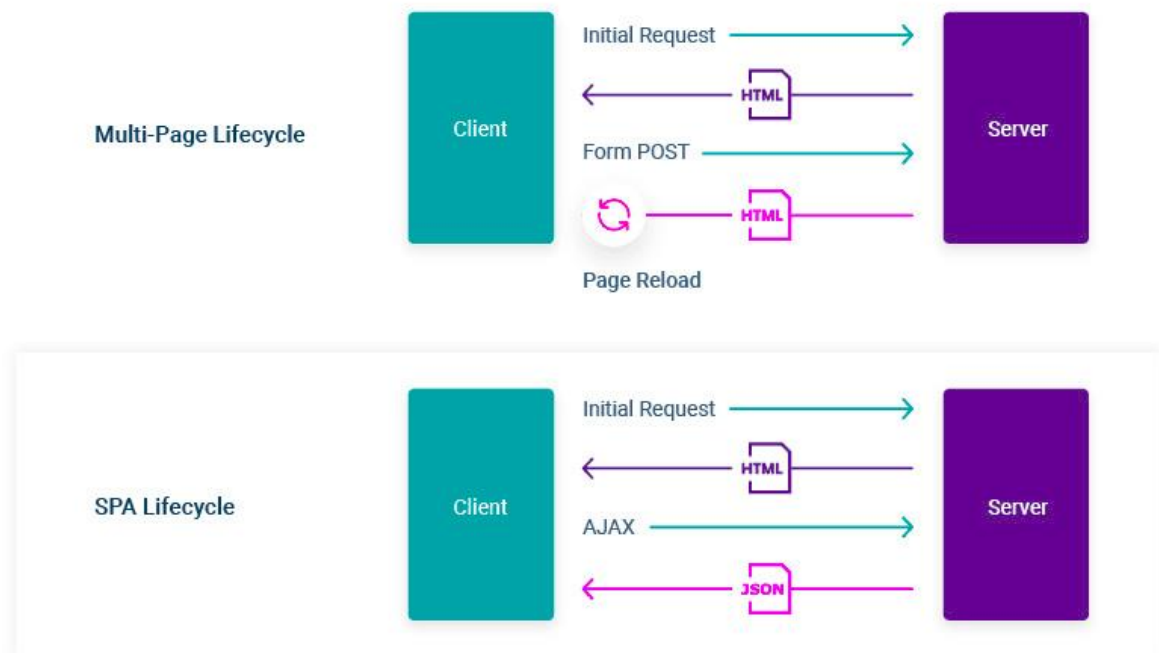
MPA funguje na principu, kdy se při, kterékoliv akci (kliknutí na hypertextový odkaz, odeslání formuláře atd.) celá stránka v prohlížeči načte znovu. To znamená, že při každém dotazu je klientovi načtena kompletní stránka i s obsahem. Také je obsah stránky tvořen již na serverové části aplikace. Díky, tomu je tento přístup vhodný pro SEO.

Obecně je přístup MPA používán při komplexnějších aplikacích (Neoteric 2018).

##### **SPA**

SPA funguje na principu, že zdroje pro zobrazení stránky (HTML, CSS, JS) jsou načteny v počátečním požadavku pouze jednou. Dále je obsah stránky měněn pomocí Javascriptu. Kdy se data získávají pomocí AJAX dotazů a Web socketům. Stránka se

v prohlížeči znova nenačítá, a data jsou získávána na pozadí. Díky neopětovanému načítání vytváří přístup SPA lepší user experience. Z důvodu, že obsah stránky je tvořen Javascriptem, a ne na úrovni serveru. Není vhodné pro SEO (Jahoda 2015).



Obrázek 1 - Porovnání MPA a SPA, zdroj: <https://vivivity.com/wp-content/uploads/2020/08/spa-mpa-lifecycle.jpg>

## SSR

SSR neboli Server side rendering, vychází z přístupu SPA, ale je rozdílný v tom, že html je vygenerováno na straně serveru, předáno prohlížeči, při prvním načtení webové aplikace probíhá hydrace, která předá data klientu, dále se již o všechno stará Javascript. SSR má oproti klasickému SPA, výhodu v tom, že díky generování na serveru je zachováno plnohodnotné SEO. SSR vyžaduje, aby běžel server, který zpracovává HTTP požadavky. Frameworky umožňující SSR, jsou např. Next.js Nuxt.js (Vercel 2022).

## SSG

SSG neboli Server side generating, vychází z přístupu SSR, ale nepotřebuje běžící server. Narozdíl od přístupu SSR, není html generováno při každém HTTP požadavku, ale je pouze generováno při tzv. Buildu webové aplikace. Při tomto procesu, je pro každou adresu webové aplikace vygenerován obsah. Tento obsah si při provozu aplikace pouze promítá (Vercel 2022).

## **3.2 Návrh aplikace**

### **3.2.1 Typografie**

Typografie je obor, který se zabývá uspořádáním písma a jazyka tak, aby byl výsledek čitelný a vizuálně přitažlivý pro finálního čtenáře. Typografie se zabývá stylem, vzhledem a strukturou písma, za cílem vyvolat ve čtenáři konkrétní emoce a předat sdělení (Hannah 2022b).

#### **Písmo**

Písmo představuje nesčetné množství znaků, různých velikostí a různých šířek. Písmo je nadřazená skupina, která obsahuje více fontů. Příkladem písma může být písmo Roboto (Hannah 2022b).

#### **Font**

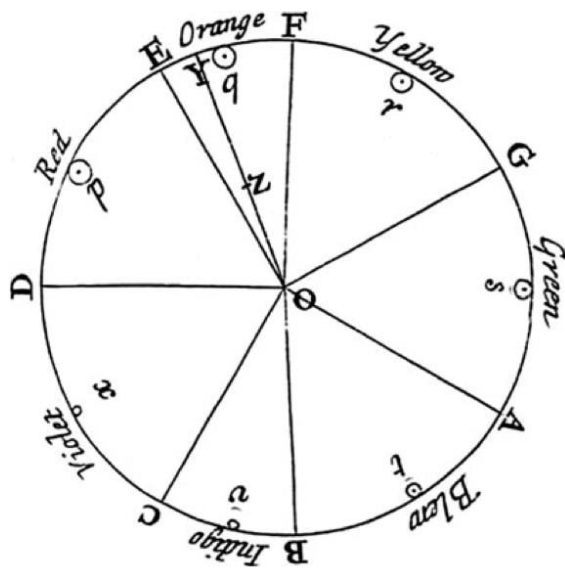
Font představuje již specifickou variaci z písma. Má tedy určenou velikost a šířku a rozsah znaků (Hannah 2022b).

### **3.2.2 Barvy**

Barvy ovlivňují uživatelský pocit z webu, i přestože si to uživatel ani neuvědomuje. Dle studie provedené Institute for Color Research, si lidé stanoví názor na produkt do 90 sekund poté co ho uvidí. Z 62 % do 90 % je tento názor stanoven právě na barvě produktu (Hannah 2022a).

#### **Teorie barev**

Teorie barev pomáhá ke stanovení vhodných barev, které by produkt měl obsahovat. Informuje o tom, jak správně spojit barvy v estetické a psychologické úrovni. Převážně je založena na Isaacu Newtonově barevném kole, které bylo vytvořeno v roce 1666. V tomto kole jsou barvy rozděleny na 3 kategorie, hlavní sekundární a terciární (Hannah 2022a).



Obrázek 2- Newtonovo barevné kolo, zdroj: <https://dpbnri2zg3lc2.cloudfront.net/en/wp-content/uploads/old-blog-uploads/color-wheel-issac-ne-qdxan.jpg>

Jeden z nejdůležitějších aspektů teorie barev je barevná harmonie. Barevná harmonie klade důraz na to, aby kombinace barev, byly na oko příjemné.

### **Barevné palety**

Barevná paleta je kombinací barev, které jsou používány při návrh produktu ze strany UI návrháře. Barevné palety nám umožňují rozklíčovat prvky na různé kategorie, to napomáhá tomu, že konečný uživatel si tyto kategorie spojuje (Hannah 2022a).

### **Tvorba barevné palety**

Existuje velká řada metod, které umožňují zvolit barevnou paletu pro produkt. Každá z metod má své výhody i nevýhody.

### **Monochromatická metoda výběru**

Monochromatická metoda výběru, funguje tak, že je zvolena jedná barva a k této barvě jsou na zvoleny podřadné odstíny a nadřaděné odstíny (Vallaure 2023).



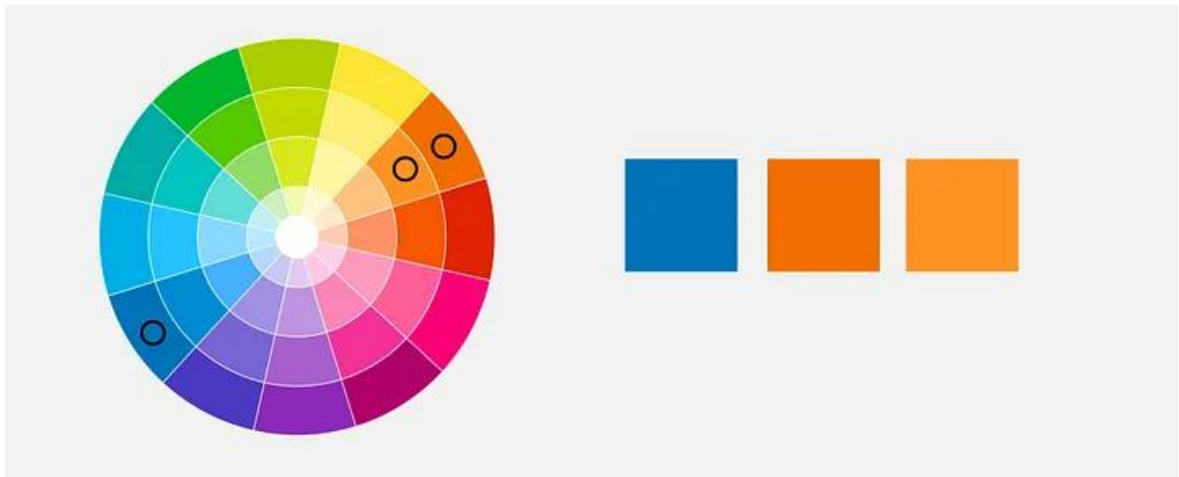
Obrázek 3 - Monochromatická metoda výběru, zdroj: [https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*cx3Wq94OBG5lprxQWPQDA.jpeg](https://miro.medium.com/v2/resize:fit:720/format:webp/1*cx3Wq94OBG5lprxQWPQDA.jpeg)

### **Analogická metoda výběru**

Analogická metoda výběru, postupuje tak, že od vybrané barvy postupujeme po kruhu na jednu nebo druhou stranu. Maximální rozpětí úhlu vybraných barev nesmí přesáhnout 90 stupňů (Vallaure 2023).

### **Komplementární metoda výběru**

Komplementární metoda výběru, funguje tak, že jsou z barevného schématu vybrány barvy, které jsou umístěny naproti sobě (Vallaure 2023).



Obrázek 4 - Komplementární metoda výběru, zdroj: [https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*9EhaxHHMGmqOc8yz9\\_c5w.jpeg](https://miro.medium.com/v2/resize:fit:720/format:webp/1*9EhaxHHMGmqOc8yz9_c5w.jpeg)

### **3.2.3 Přístupnost**

Přístupnost představuje v tématice UI/UX to, že je aplikace uzpůsobena lidem s nějakou formou postižení (Initiative (WAI) 2020).



Dle článku: „AccessiText: Automated Detection of Text Accessibility Issues in Android Apps“ trpí 15% populace na světě nějakou formou postižení, to ze přístupnosti dělá, jeden z klíčových prvků při návrhu softwaru. Největší faktor, který uvádí přístupnost do popředí, je to, že se aplikací se stali klíčové pro každodenní život (Alshayban a Malek 2022).

Standardy přístupnosti ve webovém rozhraní se zabývá WAI, která spadá pod W3C. Standardy se nazývají WCAG a v tuto chvíli je nejnovější specifikace WCAG 2.1 publikován v červnu roku 2018 (Initiative (WAI) 2022).

### **Přístupnost z pohledu barev**

V přístupnosti mají barvy své místo, hlavní parametr, který je z pohledu přístupnosti u barev hlídán, je kontrast barev, které jsou používány a poměr mezi barvou na pozadí a barvou v popředí (Lyonnais 2022).

### **Poměr barevného kontrastu**

Barevný kontrast představuje rozdíl světelnosti mezi prvkem v popředí a prvkem v pozadí. V kontextu přístupnosti, čím větší rozdíl, tím je to vhodnější pro uživatele s nějakou formou postižení. Poměr kontrastu je dvoučíselná hodnota, která tento rozdíl představuje. Poměr kontrastu nabývá hodnot od 1:1 do 21:1 (BoIA 2019).

První z hodnot představuje hodnotu jasu světlejší barvy a druhá hodnota představuje hodnotu jasu tmavší barvy (BoIA 2021).

### **Doporučené techniky barevné přístupnosti dle WCAG**

Dle WCAG by poměr barevného kontrastu pro běžný text měl být minimálně 4,5:1. U textů s velkou velikostí písma, je minimální poměr 3:1. Také je v WCAG zmíněno, že by barva neměl být primární prvek, které předává informaci (BoIA 2021).

### **Zlatý řez**

Zlatý řez je míra, které je široce užívána matematikou, designem i přírodou. Příklady užití zlatého řezu, jsou k vidění všude ve světě. Rozdělení semínek slunečnice, galaxie, moderní způsoby fotografování až po velkolepé malby od Da Vinciho nebo Michelangela (Van Essendelft 2020).

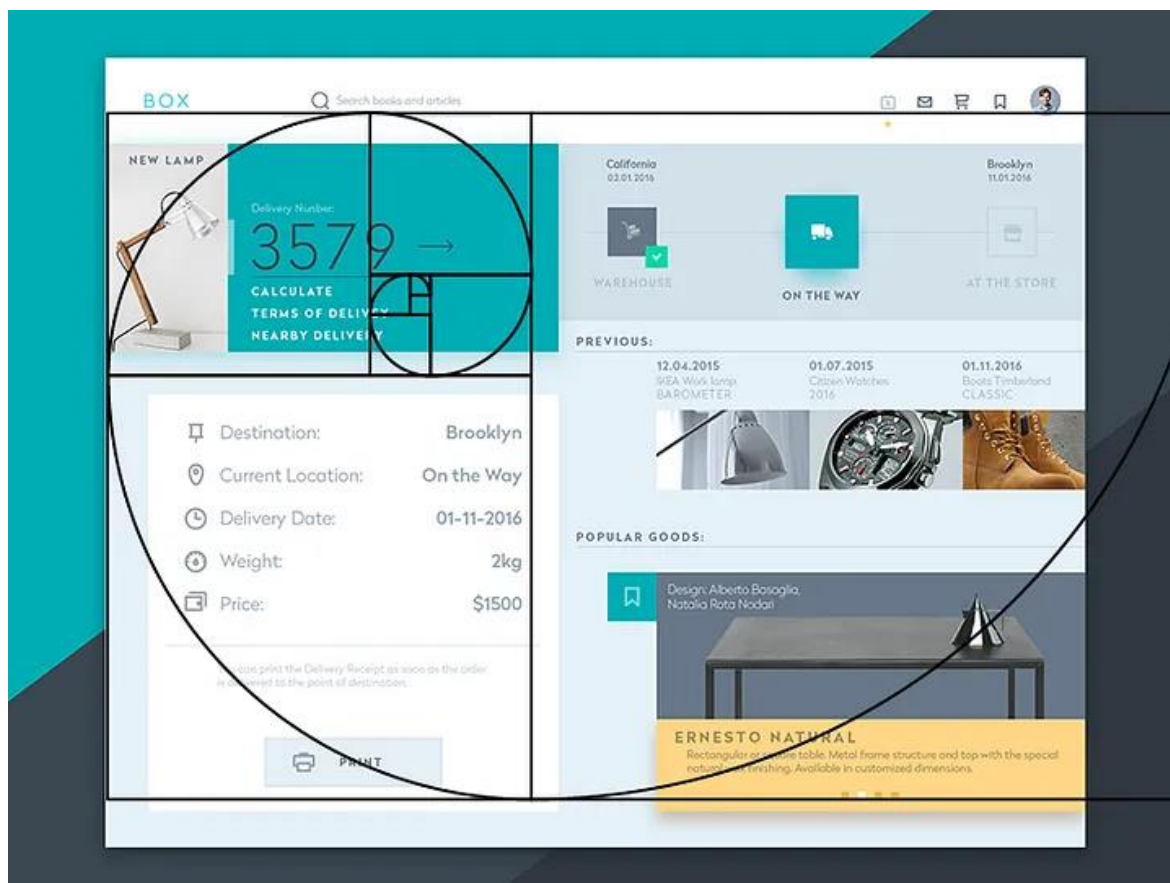
### **Prostorová kompozice**

Prostorová kompozice v UI je pojem, který do UI zavádí to, jaké elementy by měly být pospolu, v jakém uskupení a pořadí by tyto elementy měly být. A také jaké vnější a vnitřní odsazení by tyto elementy měly obsahovat (tubik 2018).

K dodržení správné prostorové kompozice v UI, existuje celá řada způsobů a metod.

### Zlatý řez v UI

V UI designu je zlatý řez, využíván tak, že jsou odstupy elementů postupně umísťovány pomocí poměru zlatého řezu. To má za důsledek, že design působí postupnou vizuální hierarchií a první dojem pro koncového uživatele je pozitivní. Zlatý řez také můžeme využít v typografii. Odstupňováním nadpisů dle poměru zlatého řezu. Při spojení všech těchto výhod, působí design navrhnutý pomocí zlatého řezu, uceleně a poutavě. (tubik 2018)

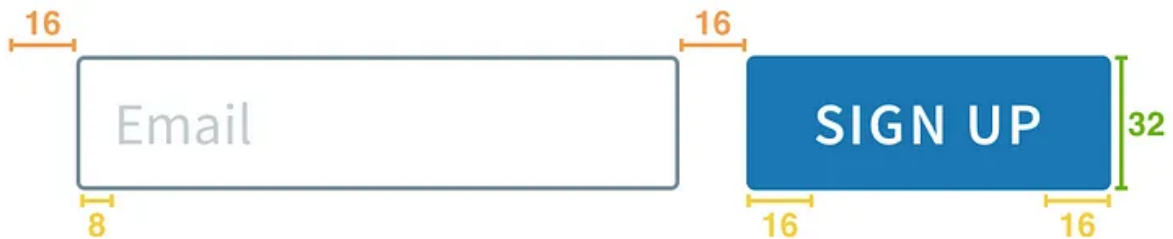


Obrázek 5 - Použití zlatého řezu v UI, zdroj:  
[https://miro.medium.com/v2/resize:fit:720/format:webp/0\\*ZE70D\\_z3RBzwjWz.png](https://miro.medium.com/v2/resize:fit:720/format:webp/0*ZE70D_z3RBzwjWz.png)

### 8bodový mřížkový systém

8bodový mřížkový systém, je systém odsazení, který odsazení postupně stupňuje hodnotou 8. Tím, že je hodnota stejná, umožňuje optimalizace pro širokou škálu šířky zobrazení a také finální design působí konzistentně. Konstantní hodnota 8, má také za cíl umožnit

jednoduchou spolupráci mezi UI návrhářem a frontend vývojářem. To může vést k rychlosti vývoje a předejití chyb (Rumman 2022).



Obrázek 6 - Ukázka použití 8bodové mřížky, zdroj: [https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*qR-NIVfU5XtHocXdpKThPw.png](https://miro.medium.com/v2/resize:fit:720/format:webp/1*qR-NIVfU5XtHocXdpKThPw.png)

### 3.3 Webové technologie

#### 3.3.1 HTML

HTML, je značkový jazyk, který je využíván k tomu, aby obsah na webu měl strukturu. Struktura obsahu je přiřazena tím, že je určeno, čím obsah je (Howe 2014).

HTML, má více verzí, poslední změnou je HTML 5. HTML 5, klade důraz na sémantiku, kdy k tomuto účele přidává další sémantické prvky. Jako jsou article, section, nav atd. Také HTML přidává zcela nové elementy jako jsou canvas a video (Pilgrim 2010).

#### Elementy

Elementy jsou v HTML struktury, které určují obsah webové stránky. Elementy mohou být párové či nepárové. Párový element znamená, že k `<>` náleží i končící značka `</>` (Howe 2014).

```
1. <a>Odkaz</a>  
2. 
```

Kód 1 - Ukázka párových a nepárových elementů v HTML, zdroj: autor

#### Atributy

Atributy slouží k doplnění doplňujících informací k elementu. Základní atributy, které může obsahovat každý element jsou třída (class), identifikátor (id). Třída představuje klasifikaci elementu, jedna třída může být užitá u více elementů. To u identifikátoru neplatí, ten může být použit se stejným jménem vždy jen u jednoho prvku (Howe 2014).

#### Struktura HTML dokumentu

Každý HTML dokument obsahuje hlavní (kořenový) element `<html>`. Tento element obsahuje dva podřízené elementy. První z nich je `<head>`, tento element obsahuje např. element

<title>, který určuje titulek stránky. Dále obsahuje element <body>, v tomto elementu již jsou elementy jako jsou p, div, article, section, atd (Howe 2014).

### 3.3.2 CSS

Je jazyk, který umožňuje popsání toho, jak bude Html vypadat vizuálně. Primárně je určeno k tomu, aby byl vzhled webové stránky oddělen od struktury (Html). CSS je vyvíjeno a standardizováno společností W3C. CSS prošlo za dobu své existence velkou řadou změn a rozšíření (Mozilla 2023c).

#### Princip CSS

CSS jazyk postavený na pravidlech, to znamená, že vývojář specifikuje pravidla, která se mají konkrétním prvkům přiřadit. K určení prvků, kterým mají být přiřazeny pravidla, se využívají selektory. Každé pravidlo obsahuje selektor a k němu vlastnosti (Mozilla 2023c).

#### Selektory v CSS

K určení prvků, kterým mají být přiřazeny pravidla, se využívají selektory. Selektory jsou děleny na různé typy:

- Selektory dle typu, třídy nebo ID
- Atributové selektory
- Pseudo-selektory

Selektory dle typu, třídy nebo ID, jsou základními typy selektorů. Použití jich je jednoduché a jsou základním stavebním prvkem CSS stylů. (Mozilla 2023b)

```
1. img {
2.     max-width: 100%;
3.     height: auto;
4.     padding: 0;
5.     margin: 0;
6. }
7.
8. .heading {
9.     color: black;
10. }
11.
12. #main-item {
13.     background-color: white;
14. }
15.
```

*Kód 2 - Ukázka použití Selektorů dle typu, třídy nebo ID, zdroj: autor*

Atributové selektory navazují na Selektory dle typu, třídy nebo ID a přidávají jim možnost vybrat si takové elementy, u kterých je specifický atribut, nebo specifický atribut obsahuje řetězec (Mozilla 2023a).

a[title]	Element, který obsahuje atribut title.
a[href="https://example.com"]	Element, který obsahuje atribut href s hodnotou <a href="https://example.com">https://example.com</a> .
p[class~="special"]	Element, který obsahuje řetězec „special“.
li[class^="box-"]	Element, u kterého atribut class začíná „box-“.
li[class\$="-box"]	Element, u kterého atribut class končí „box-“.

Tabulka 1 - Ukázka atributových selektorů, zdroj: autor

Pseudo-selektory jsou selektory, které reflektují speciální stav elementu. Syntax využití pseudo-selektorů je takový, že první obsahuje element, u kterého má být speciální stav zjištěn, a poté následuje pseudo-selektor (Andybell 2023).

:hover	Když uživatel na prvek najede myší.
:active	Když uživatel klikne na prvek a ještě nepustil tlačítko myši.
:focus	Pseudo-selektor :focus představuje situaci, kdy uživatel klikne do pole formuláře.
:checked	Představuje stav zaškrtačovacího tlačítka, které je vyplněno.
:first-child	Vybere prvek, který je první ze všech prvků nadřazeného elementu.
:last-child	Vybere prvek, který je poslední ze všech prvků nadřazeného elementu.
:first-of-type	Vybere prvek, který je první ze všech prvků nadřazeného elementu, ale nebere všechny prvky, ale pouze prvky, které jsou konkrétní element.
:last-of-type	Vybere prvek, který je poslední ze všech prvků nadřazeného elementu, ale nebere všechny prvky, ale pouze prvky, které jsou konkrétní element.

## Jednotky v CSS

Některé vlastnosti v CSS (width, margin, padding, font-size, border-radius), mohou mít jako hodnotu jednotku. Jednotky jsou děleny na relativní a absolutní (freeCodeCamp.org 2020).

Relativní jednotky už podle svého názvu představují jednotky, které jsou k něčemu vázány (šířka displeje koncového zařízení, šířka nadřazeného element). U jednotek absolutních je tato hodnota dána pevně a na ničem nezávisí (freeCodeCamp.org 2020).

Do relativních jednotek řadíme em, rem, %, vw, vh. Em představuje konkrétní font-size, který je pro element nastavený. Jednotka rem představuje relativní jednotku k font-size kořenovému prvku html. Vw představuje šířku zařízení v pixelech. Vh představuje výšku zařízení v pixelech (freeCodeCamp.org 2020).

Do absolutních jednotek řadíme px, cm, mm, pt (freeCodeCamp.org 2020).

### **3.3.3 JavaScript**

JavaScript je skriptovací jazyk, představený Brendanem Eichem v roce 1995. Od té doby byl tento jazyk převzat webovými grafickými prohlížeči. Díky JavaScriptu se procházení webu posunulo velmi dopředu. Pohaní všechny webové aplikace a umožnil, funkcionalitu webových aplikací, kdy není potřeba znovu načítat stránku. Také je javascript používán na serverové části webových aplikaci, což je umožněno díky softwaru Node.js (Haverbeke 2019).

### **3.3.4 ECMAScript**

Je standardizace jazyka Javascript, o kterou se stará organizace ECMA, pod specifikací, která se nazývá ECMAScript specification. Poslední verze byla vydána pod jménem ECMAScript 2017 v červnu 2017. Z důvodu toho, že ne prohlížeče neimplementují změny z novějších verzí ECMAScriptu hned, využívají se transpilátory, které převedou kód napsaný podle nové specifikace do kódu, který podporuje většina prohlížečů (Mohan a Prusty 2018).

### **3.3.5 Organizace kódu do komponentů**

V poslední době, se ve vývoji webových aplikací přišlo k vývoji, kdy jsou části kódu organizovány do komponentů. Což umožňuje přehlednost z pohledu organizace kódu, stejně tak přehlednost z pohledu UI. Tím, že je výsledná aplikace rozdělaná na mnoho malých částí, je škálovatelnost aplikace značně jednodušší. U většiny knihoven, které umožňují organizaci kódu do komponentů, je struktura komponentů taková, že komponent má svůj state a zároveň komponent přijme a vrací data. Frameworky, které využívají komponentový přístup jsou např. Vue.js, React.js, Svelte (Harsh 2018).

### 3.3.6 Imperativní a deklarativní způsoby UI

Imperativní a deklarativní způsoby, definují to, jak bude vývojáři umožněno měnit UI. Imperativní způsob představuje to, kdy vývojář musí při uživatelské interakci s UI nadefinovat každou akci, která bude této interakci následovat. V deklarativní způsobu se postupuje tak, že vývojář nadefinuje stavy, které mohou nastat a zbytek se postará knihovna (Amaury 2022).

### 3.3.7 React

React je knihovna pro JavaScript, která umožňuje organizaci kódu do komponent. React využívá JSX, pro strukturu vykreslování jednotlivých komponent. React využívá deklarativní způsob změny UI. React je primárně využíván, pro psání frontendu webových aplikací, avšak s příchodem React native, pronikl React i na mobilní zařízení. React byl vydán a je udržován Facebookem (React 2018).

#### Základní principy Reactu

V Reactu existují dva způsoby, jak psát kód jeden je funkcionální a druhý založený na třídách. Oba způsoby jsou podporovány, ale do popředí se dostal způsob funkcionální s použitím hooků.

#### Hooky

Hooky jsou v Reactu funkce, které umožňují znovupoužití logiky na více místech také nám hooky umožňují definovat state komponentu a napojit vedlejší efekty do kódu. React má řadu vestavěných hooků a také poskytuje možnost si definovat svoje vlastní hooky. Nejdůležitější vestavěné hooky představují: useState, useEffect, useMemo (Facebook 2023).

#### Hook useState

Hook useState umožňuje mít v komponentu state, který je uložen do unmountování komponentu. Hook useState má jako parametr iniciální hodnotu pro state a návratová hodnota hooku je pole, kde první prvek je hodnota uložena ve státu a druhý prvek je funkce umožňující měnit hodnotu (Facebook 2023).

#### Hook useEffect

Hook useEffect umožňuje zavolat definovanou funkci, když se nějaký prvek ze závislostí změní. Hook useEffect má dva parametry první je funkce, která bude zavolána

a druhý parametr, je pole obsahující všechny závislosti, u kterých má změna spustit funkci. Návratová hodnota není (Facebook 2023).

### Hook useMemo

Hook useMemo s umožňuje k uchování hodnoty, která se bude přepočítávat jen, když se některá ze závislostí změní. Parametry funkce jsou funkce, která vrací hodnotu, která má být uchována a pole závislostí. Návratová hodnota hooku je hodnota, spočítaná funkcí (Facebook 2023).

### React context a hook useContext

React context je určen ke sdílení proměnné napříč komponenty. K vytvoření contextu má React funkci createContext. Tato funkce bere jako parametr výchozí hodnotu pro context. Aby bylo možné context využít v komponentu, je potřeba využít pomocný hook useContext s proměnou kterou je návratovou hodnotou createContext (Facebook 2023).

```
1. export type UserContextValue = {
2.   user: FirestoreUser
3.   userData: User
4. }
5.
6. const UserContext = React.createContext<UserContextValue>({
7.   user: null,
8.   userData: null,
9. })
```

*Kód 3 - Ukázka využití contextu v Reactu, zdroj: autor*

### 3.3.8 Next.js

Je Javascriptová knihovna, a zároveň nadstavba pro React, která umožňuje tvorbu Reactových aplikací, obsahující SSR. Next.js obsahuje vlastní routovací systém, umožňuje načítat data na serveru a mnoho dalšího (Vercel 2023).

## 3.4 Přístupy vývoje softwaru

Vývoj softwaru je komplikovaná disciplína, která vyžaduje, aby byl v průběhu vývoje vývojář o vývoji informován a vývoj byl efektivní. Kvůli tomu existuje řada způsobů, podle kterých týmy postupují, aby dosáhly efektivitu a přehledu vývoje (Kumar a Bhatia 2014).



### **3.4.1 Vodopádový model**

Vodopádový model je druh vývoje softwaru, který je jeden z nejstarších a nejpoužívanějších. Tento přístup se skládá z 4 postupně jdoucích kroků (Kumar a Bhatia 2014).

- Analýza požadavků – Jsou zjištěny potřeby, které zákazník má a co by software měl obsahovat (Kumar a Bhatia 2014).
- Návrh – Na základě požadavků, je stanoveno, jak by měl software vypadat (Kumar a Bhatia 2014).
- Implementace – Na základě návrhu, vypracují programátoři software (Kumar a Bhatia 2014).
- Testování – Software je otestován, zda splňuje požadavky, které byly stanoveny (Kumar a Bhatia 2014).

### **3.4.2 Agilní vývoj**

Agilní vývoj softwaru, představuje vývoj, kdy je software vyvíjen v iteracích. Tedy software je rozdělen na menší části a ty jsou rozděleny do iterací a zpracovávány. Agilní vývoj software má různé typy, jako jsou například Scrum nebo Kanban (Kumar a Bhatia 2014).

### **3.4.3 Vývoj řízený testy**

Vývoj řízený testy, probíhá tak, že první jsou pro software napsány testy a až poté je napsán software. Výhody vývoje řízeného testy může být to, že software je psaný s testy, tedy není potřeba složité hledání chyb (Kumar a Bhatia 2014).

### **3.4.4 Lean vývoj**

Lean vývoj softwaru, je založený na minimalizaci plýtvání a maximalizaci hodnoty pro zákazníka. Tento přístup snaží na minimalizaci zbytečných kroků, jako jsou například časté schůzky.

## **3.5 Testování aplikací**

### **3.5.1 Definice UX**

Ve zkratce je pojem UX proces vytváření produktů (aplikací), které jsou pro uživatele praktické a intuitivní. Obecně k tomu, abychom tvořili správné UX, musíme znát naše

uživatelé, to znamená znát jejich potřeby chování a smysl v kterém budou naši aplikaci používat. Prvky, které by produkt se správným UX měl obsahovat zahrnují (Babich 2020).

- Použitelnost – Produkt musí být lehce použitelný a intuitivní.
- Užitečný – Produkt musí řešit problém některého z uživatelů produktu. Nemůže být bez smyslu.
- Atraktivní – Produkt musí vizuálně působit atraktivně a uživateli předávat pozitivní energii.
- Nápomocný – V případě, že uživatel narazí v aplikaci na problém, měl by mít možnost tento problém jednoduše vyřešit.
- Dostupný – Produkt by měl být optimalizovaný i pro lidi s postižením (Babich 2020).

### **3.5.2 Definice UI**

Pojem UI představuje to, jak je produkt navrhnut po designové stránce. UI představuje to co uživatel používá, aby s daným produktem komunikoval.

### **3.5.3 Definice pojmu Usability**

Pojem usability, nám v kontextu UX představuje to, zda je uživatel schopen plnit úkoly, kvůli kterým je náš produkt vytvořen rychle a efektivně (International Organization for Standardization 2019).

### **3.5.4 Rozdíl mezi pojmy UX a UI**

Pojem UX je často milně zaměňován s pojmem UI, i když jsou tyto pojmy podobné, v realitě je v nich rozdíl. Pojem UI nám představuje to, jaký má produkt vzhled oproti UX, které se zabývá čistě chováním uživatele v našem produktu a tím jaký z toho má uživatel pocit (Babich 2020).

### **3.5.5 Proč testovat UX**

Důvod, proč testujeme UX je, že to je jediný způsob, jak porozumět potřebám uživatelů našeho produktu. Když nevíme, jaké potřeby naši uživatelé mají, nemůžeme určit, zda je pro ně náš produkt vhodný, tím pádem tedy tvoříme něco, co nevíme, k čemu je. Dále nám testování našeho produktu pomůže vytvořit produkt, který je jednoduchý a zábavný k použití. Bez

testování bychom na trh přišli s produktem, u kterého by nám nebylo jasné, jestli ho uživatelé budou používat (Mortensen 2022).

### 3.5.6 Testování

Před každým testováním, je důležité si vůbec určit, pro koho je produkt určen, tomu uzpůsobit, jaké publikum budeme k našemu testování volit. Tedy např. Pokud tvoříme aplikaci pro děti, musíme tuto aplikaci testovat na dětském publiku. Pokud bychom zvolili publikum, které neodpovídá naší cílové skupině, stalo by se testování irelevantní. Zároveň musíme do testování zhodnotit i to, jaké prostředky k tomu máme, v případě malých produktů, nemůžeme zvolit obsáhlejší testování, protože by nám to finance neumožnili (Babich 2019b).

### 3.5.7 Typy testování

#### Guerrilla testování

Guerrilla testování nám v rovině testování představuje jeden z nejméně nákladných způsobů testování. Hlavní výhodou tohoto testování je, že může být prováděno kdekoli (kavárna, různé veřejné akce). Jako značnou nevýhodu můžeme brát, že to není kompletní testování, a ne vždy ve všech případech je závěr věcný (Ligertwood 2022).

Příklad scénáře Guerrilla testování:

Úkol	Scénář
Najděte pizzu šunkovou a bacon a dejte si ji do košíku.	Dostal jste chuť na pizzu. Proto si do košíku zvolíte dvě svoje oblíbené pizzy.
Objednejte si pizzu a zadejte slevový kód z minulého nákupu.	Pizzu už máte v košíku, proto si ji přejete objednat zároveň máte z minulé objednávky ještě slevový kód.

Tabulka 2 - Příklad scénáře při Guerrilla testování, zdroj: autor

Příklad poznámek:

Úkol	Poznámka
Najděte pizzu šunkovou a sýrovou a dejte si ji do košíku.	Uživateli trvalo dlouho, než zjistil, kde je tlačítko na přidání do košíku.
Objednejte si pizzu a zadejte slevový kód z minulého nákupu.	Uživatel nenašel pole na vložení slevového kódu.

Tabulka 3 - Příklad poznámek při Guerrilla testování, zdroj: autor

#### Testování usability v laboratoři

Lab usability testování je testování, které probíhá ve speciálních laboratořích, které jsou určeny k testování. Každé Lab usability testování obsahuje moderátora, který má pro danou funkci vzdělání a uživatele, kteří podle instrukcí od moderátora, interagují s produktem, který je testován. Lab usability testování je vhodné použít v případě, kdy potřebujeme kompletní zhodnocení a máme na to i příslušný rozpočet. Z důvodu faktoru, že potřebujeme laboratoř, v které bude testování probíhat a zkušeného moderátora je toto testování finančně náročné (Box UK 2022).

### **Vzdálené usability testování**

Moderované vzdálené testování, umožňuje testovat produkt vzdáleně s přítomností moderátora. Oproti, testování v laboratoři, kde jsou testovací subjekty přítomny osobně. Vyžaduje vzdálené testování větší přípravu od moderátora, protože vše probíhá online (Expérience 2022).

### **Nemoderované vzdálené usability testování**

Nemoderované vzdálené testování, nevyžaduje přítomnost moderátora. Také není sjednaná jednotná schůzka, proto toto testování má výsledky rychleji. Z důvodu nepřítomnosti moderátora, je obtížné testovat produkt, který není ještě plně hotov. Protože není jak testovacím subjektům, vysvětlit případné nejasnosti v produktu (Whitenton 2019).

### **Kontextové dotazy**

Kontextové dotazy nám umožňují zjistit, zda produkt, který plánujeme vytvořit. By si na trhu našel uživatele, kteří by ho používali. Využívány jsou převážně na začátku tvorby produktu. Kontextové dotazy jsou prováděny, tak, že testovací subjekt je pozorován pozorovatelem při specifické činnosti, kterou chce pozorovatel analyzovat. Kontextové dotazy jsou vhodné, když potřebujeme porozumět problematice do hloubky. V případě, softwarových produktů, proto nejsou tolik vhodné (Salazar 2020). Např. Může kontextový dotaz představovat to, když chceme analyzovat vzhled dálkového ovládání k televizi.

### **Třídění karet**

Dle studie provedené Amy Schade Yunnuo Cheng, Samyukta Sherugar v roce 2016 na 215 testovacích subjektech ve Spojených státech amerických a ve Velké Británii, je největším problém UX na webech špatně seřazený obsah stránky (Schade et al. 2016).

Tento problém řeší testovací technika třídění karet. Třídění karet spočívá v tom, že je zvoleno 40–80 hlavních prvků, které se budou zobrazovat na webu, který tvoříme. Tyto prvky jsou napsány na karty. Poté necháme testovací subjekty roztrždit tyto do skupin a každou ze zvolených skupin vhodně pojmenovat. Poté se uživatele moderátor ptá, proč skupiny karet roztrždil takto (Sherwin 2018).

### **Nahrávání interakce**

Nahrávání interakce spočívá v tom, že je produkt anonymně nahráván. To se děje na pozadí aplikace a uživatel o tom neví. To je oproti ostatním typům testování výhoda, protože výstupy z tohoto testování jsou reálné a uživatel se nechová jinak kvůli tomu, že ví, že je testován. Nahrávání interakce je nejúčinnější spolu s další metodou testování. Hojně bývá využíváno ve spolupráci s A/B testování. Na nahrávání interakce se dají používat nástroje, jako jsou třeba Hotjar, FullSession nebo Inspectlet (Babich 2019b; FullSession 2021).

### **A/B testování**

A/B testování je druh testování, který probíhá, tak, že jsou testovány dvě podoby produktu, kdy testujeme, které podoba si vede lépe. Tedy, u které podoby testovací subjekt čekal kratší dobu, kliknul správně atd. A/B testování je vždy využíváno s dalšími způsoby testování. Aby výsledky byly zaznamenány (Babich 2019a).

## **4 Praktická část**

### **4.1 Popis prototypu aplikace**

Před samotným vývojem prototypu aplikace, je důležité stanovit to, co aplikace bude umět a jaké funkce bude obsahovat. Aplikace bude určena jako fitness aplikace, která umožňuje trenérům plánovat tréninky a klientům tyto tréninky zobrazit.

Cílová skupina aplikace, jsou lidé, kteří se zajímají o fitness. Může se jednat o trenéry, kteří by chtěli stavět tréninkové plány online anebo klienty, kteří chtějí cvičit z domu.

Aplikace bude obsahovat následující zobrazení Přihlášení, Registrace, Zapomenuté heslo, Nástěnka tréninků.

#### **4.1.1 Požadavky aplikace**

Jako hlavní požadavky, které by aplikace měla mít byly stanoveny tyto:

- Rychlost

- Validace formulářů

Rychlost, byla stanovena proto, že přímo souvisí s uživatelskou spokojeností. formulářů byla zvolena proto, že v aplikaci bude více komplexnějších formulářů, proto je správná validace podstatná.

#### **4.1.2 Struktura aplikace a její náležitosti**

Průběh aplikace bude takový, že budou definovány dvě skupiny uživatelů, jedna skupina trenérů a druhá skupina klientů. Trenérům bude umožněno přiřazovat klientům tréninky a klienti v aplikaci budou mít možnost tyto tréninky plnit.

#### **Struktura tréninku**

Trénink bude v aplikaci představovat hlavní datovou strukturu. Trénink se bude skládat z atributů, datumu na kdy je trénink naplánován, klienta, kterému je trénink naplánován, názvu tréninku, který bude pro klienta důležitý k identifikování, stručnému textu, který bude sloužit jako informace pro klienta a cviků. Cviky budou děleny do tří kategorií, warmUp, main, a coolDown. Každá z těchto kategorií může mít libovolný počet cviků.

#### **4.1.3 Požadavky jednotlivých zobrazení**

##### **Přihlášení**

- Vstupy pro e-mail a heslo.
- Tlačítka pro přihlášení a registraci.

##### **Registrace**

- Vstupy pro Jméno, Příjmení, Datum narození, Telefonní číslo, e-mail a heslo s opakováním hesla.
- Kontrola kódu trenéra.

##### **Zapomenuté heslo**

- Vstup pro e-mail.
- Tlačítka na odeslání.

##### **Nástěnka tréninků**

- Vyobrazení nadcházejících a uplynulých tréninků.
- Možnost otevřít si detail tréninku.

## Tvorba tréninku

- Možnost přidat cviky.
- Vstupy pro název tréninku, základní informace, klient a datum tréninku.
- Možnost nastavit čas / počet opakování u jednotlivých cviků a pauzu jednotlivých sérii.

## Detail tréninku

- Seznam cviků, které trénink obsahuje.
- Tlačítka na splnění tréninku.
- Informace o tom, který trenér trénink vytvořil.

### 4.1.4 Cviky a jejich kategorie v aplikaci

Cviky budou v základní verze prototypu aplikace uloženy v JSON souboru. Tento soubor bude obsahovat pole všech cviků, kdy každý cvik bude obsahovat tyto atributy:

- Název cviku – name.
- Popis cviku – description.
- Kategorie cviku, bude nabývat hodnot „Warm up“, „Hlavní část“ a „Cool down“ – category.
- Zda může být u cviku nastaven čas jednotlivé série – editTime.
- Zda může být u cviku nastaven počet opakování – editQuantity.

```
1. [
2.   {
3.     "name": "Protážení nohou",
4.     "description": "Protáhněte si nohy od spodu až dolů",
5.     "category": "Warm up",
6.     "editQuantity": "ANO",
7.     "editTime": "ANO"
8.   },
9.   {
10.    "name": "10 kliků",
11.    "description": "Provedte 10 kliků",
12.    "category": "Hlavní část",
13.    "editQuantity": "ANO",
14.    "editTime": "NE"
15.  },
16.  {
17.    "name": "Vyklusání 10 minut",
18.    "description": "Provedte vyklusání",
19.    "category": "Cool down",
20.    "editQuantity": "NE",
21.    "editTime": "ANO"
22.  }
23. ]
24.
```

Kód 4 - Ukázka konfigurace cviků, zdroj: autor

#### 4.1.5 Architektura aplikace

Pro tvorbu aplikace, byla zvolena knihovna Next.js spolu se serverlesss architekturou firebase od společnosti Google. Next.js umožňuje vystavit vlastní koncové url, které by při rozšíření aplikace o další funkce mohli odstranit potřebu mít další server. Na frontend bude využita knihovna MUI spolu s knihovnou emotion.sh. K správné validaci formulářů bude využita knihovna react-hook-form, konkrétně její nadstavba react-hook-form-mui, ta propojuje komponenty pro formulář z MUI na react-hook-form.

#### 4.1.6 Barvy v aplikaci

Dle teoretické části, budou zvoleny barvy, které se v aplikaci budou používat. Primární barva budou sloužit ke primární akci na každém zobrazení. Sekundární barva budou využívána k navázání na sekundární akci v zobrazení. Barvy elementů budou následující:



Obrázek 7 - Základní barvy v aplikaci, zdroj: autor

V aplikaci budou využívány tři barvy na texty. Primární barva pro texty, bude využívána k obecnému textu, který bude v aplikaci. Sekundární barva, bude používána v kontrastu s primární, aby bylo docíleno, že uživatel pochopí, že hlavní text je ten, který je určen primární barvou.

## 4.2 Tvorba prototypu

### 4.2.1 Customizace MUI

Knihovna MUI, je obsáhlou knihovnou, která dává důraz na to, aby každý komponent, který knihovna obsahuje, byl konfigurovatelný. To platí ať už z pohledu prvků komponentů tak i z pohledu vzhledu komponentu. Oba tyto faktory, je možné nastavovat buď globálně, to vede k tomu, že každý překonfigurovaný komponent, bude v aplikaci obsahovat tyto změny. Nebo se nabízí možnost měnit faktory pouze pro jedno využití komponentu, např. v zobrazení.



Obecně je vhodné, mít většinu konfigurací nastavených globálně, aby bylo předejito duplikátům v kódu.

#### 4.2.2 Nastavení typografie

Typografie je prvek, který bude v celé aplikaci stejný, proto je vhodné nastavit pravidla globálně. V aplikaci bude text využíván na více místech, bude se týkat o nadpisy, tlačítka a odstavce. K vykreslení nadpisů a odstavců je v knihovně MUI určen komponent Typography.

#### 4.2.3 Úprava vzhledu komponentů z MUI

##### MuiCssBaseline

MuiCssBaseline je komponenta z knihovny MUI, která je určena k resetování vestavěných stylů prohlížečů. Je vhodné do ní umístit globálně přepisované CSS vlastnosti pro kořenový element. K docílení přehlednosti stylů, zde nastavíme font-size na 62.5 %. To způsobí, že hodnota 1 relativní jednotky rem bude 10px.

```
1. MuiCssBaseline: {  
2.     styleOverrides: {  
3.         html: {  
4.             fontSize: '62.5%',  
5.         },  
6.         body: {  
7.             fontSize: '1.4rem',  
8.             position: 'relative',  
9.         },  
10.     },  
11. },
```

*Kód 5 - Nastavení atributů u MuiCssBaseLine, zdroj: autor*

##### MuiButton

MuiButton je komponenta z knihovny MUI, která umožňuje pracovat s tlačítky. Zde bude nutné tlačítka nastavit, aby reflektovali barvy, které byly v předchozí části zvoleny.

MuiButton obsahuje prop „variant“, tato prop je typu string a může obsahovat hodnoty „contained“ a „outlined“. Hodnota „contained“, nastaví vzhled tlačítka na vyplněno barvou oproti tomu hodnota „outlined“ nastaví vzhled na nevyplněn pouze s rámečkem. V aplikaci bude vzhled tlačítek vždy vyplněn, proto je tato prop nastavena globálně na „contained“.

Tlačítko používá jako výchozí barvu textu barvu nastavenou v paletě, pro primární tlačítko bude muset být barva nastavena na bílou.



Obrázek 8 - Ukázka vzhledu tlačítka, vlevo sekundární, vpravo primární, zdroj: autor

#### 4.2.4 Vývoj základních komponent

Základní komponenty představují v aplikaci komponenty, které budou využívány napříč zobrazeními. Proto je vhodné, aby byly vytvořeny zvlášť a byly konfigurovatelné pro ostatní zobrazení. Základní komponenty, které budou využívány jsou tyto:

##### **Dialog**

Tento komponent bude základní Dialog z MUI rozšířený o ikonu zavření, kterou je možné skrýt, pokud je to v zobrazení nutné.

##### **Link**

Komponent Link bude představovat hypertextový odkaz. Tento komponent propojí Link komponent z knihovny MUI s Link komponentem z knihovny Next.js.

##### **PageTitle**

Komponent PageTitle bude využíván k zobrazení nadpisů u zobrazení. Zároveň bude umožňovat zobrazit načítací indikátor.

#### 4.2.5 Práce s knihovnou react-hook-form-mui

Knihovna react-hook-form-mui je nadstavbou knihovny react-hook-form. Práce se knihovnou je jednoduchá. Knihovna obsahuje komponent FormContainer. Tento komponent obaluje všechny pole formuláře, které se v něm mají objevovat. Zároveň komponent obsahuje dvě props, formContext a onSubmit. Prop formContext, je context, který je možné vytvořit pomocí pomocného hooku useForm z knihovny.

#### 4.2.6 Práce s načítáním v aplikaci

Dle Jakoba Nielsena je důležité, aby uživatel vždy věděl, co se v aplikaci děje a na co zrovna čeká. To může zvyšovat celkovou použitelnost aplikace (Nielsen 2020).

Proto bude v aplikaci dbát důraz na informování uživatele o tom, že se na pozadí něco načítá. V knihovně MUI, je k tomu komponent `LoadingButton`, který používá komponent `Button` a pouze v případě, že prop `loading` je pravdivá, přidá načítací ikonu.

Aby se s načítáním pracovalo přehledně, byl proto vytvořen speciální hook `useLoading`. Tento hook jednoduše volá hook `useState`.

```
1. export const useLoading = (initialValue = false) => {
2.   const [loading, setLoading] = useState<boolean>(initialValue)
3.
4.   return { loading, onLoadingChange: setLoading }
5. }
```

*Kód 6 - Pomocný hook useLoading, zdroj: autor*

## 4.2.7 Vývoj jednotlivých zobrazení

Aby bylo docíleno přehlednosti aplikace a kód byl věcný, bude každé zobrazení umístěno ve vlastní složce. Každé zobrazení v aplikaci bude jeden React komponent. Aby bylo docílena přehlednosti kódu a aplikace byla udržitelná bude logika aplikace umístěna v separátních funkcích (actions), které budou mít jako návratovou hodnotu vždy funkce, které jsou v konkrétním zobrazení potřebné.

### 4.2.7.1 Přihlášení

Jak bylo zmíněno v požadavcích, zobrazení Přihlášení bude obsahovat dva vstupy. Jeden pro e-mail a druhý pro heslo. Pro toto bude využita knihovna `react-hook-form-mui`. Všechna logika bude pro přihlášení uložena v hooku `useLoginPageActions`.

Nejprve si v tomto hooku, vytvoříme `formContext` s poli `e-mail` a `password`.

```
1. const formContext = useForm<{
2.   email: string
3.   password: string
4.   all: string
5. }>()
6.
```

*Kód 7- Ukázka vytvoření form kontextu pro přihlášení, zdroj: autor*

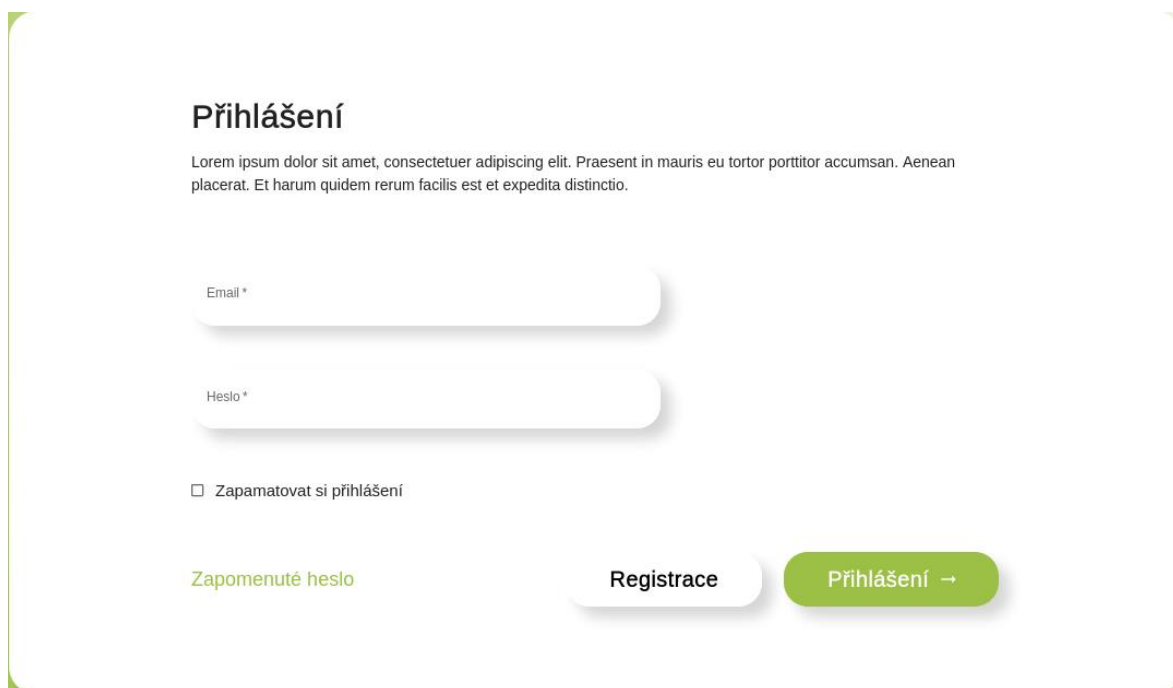
Při kliknutí na tlačítko přihlásit na zobrazení přihlášení, bude potřeba uživatele informovat o načítání, zda údaje jsou správné či nikoliv. Proto v hooku `useLoginPageActions` bude použit hook `useLoading`.

Také bude v hooku nadefinována a exportována funkce `onSubmit`, která bude volána v případě odeslání formuláře. V této funkci bude nejprve nastaveno načítání na `true`, pomocí funkce `onLoadingChange` z hooku `useLoading`. Poté bude pokus o přihlášení uživatele funkcí z `firebase signInWithEmailAndPassword` a na konci bude načítání resetováno zavoláním `onLoadingChange` s hodnotou `false`.

UI bude u zobrazení přihlášení vypadat tak, že bude nadpis s textem a následovat bude formulář, pod formulářem budou dvě tlačítka. Primární tlačítko bude to, které bude odesílat formulář a vedle něj bude sekundární tlačítko, které po kliknutí odkáže na stránku registrace. Tlačítka budou vnořena do pomocného divu, u kterého bude nastaven css atribut display:flex aby byla tlačítka umístěna horizontálně vedle sebe.

```
1. <Styled.ActionsWrapper>
2.   <Styled.SecondaryButton
3.     onClick={() => router.push('register')}
4.   >
5.     Registrace
6.   </Styled.SecondaryButton>
7.   <LoadingButton
8.     loading={loading}
9.     type="submit"
10.    variant="contained"
11.    endIcon={<ArrowRightAltIcon />}
12.    onClick={() => onClearErrors('all')}
13.  >
14.    Přihlášení
15.  </LoadingButton>
16. </Styled.ActionsWrapper>
```

Kód 8 - Ukázka tlačítek na zobrazení přihlášení, zdroj: autor



**Přihlášení**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent in mauris eu tortor porttitor accumsan. Aenean placerat. Et harum quidem rerum facilis est et expedita distinctio.

Email \*

Heslo \*

Zapamatovat si přihlášení

[Zapomenuté heslo](#)

[Registrace](#) [Přihlášení →](#)

Obrázek 9 - Ukázka výsledného vzhledu zobrazení přihlášení, zdroj: autor

#### 4.2.7.2 Registrace

Zobrazení registrace, je určeno k registraci nových klientů do aplikace. Registrace bude obsahovat formulář s dvěma kroky, první krok budou zadání základních údajů a na druhém kroku bude uživatel zadávat, kterého trenéra bude mít přiřazeno. Je důležité, aby u tohoto dvou krokového formuláře byla správně implementována validace všech polí.

Validace polí na zobrazení registrace, bude následovná. Všechny pole jsou povinná a zároveň při odeslání formuláře musí aplikace kontrolovat, zda se email neshoduje s emaily uživatelů ve firebase. Také musí být validováno pole na vložení unikátního kódu trenéra, kde v případě, že uživatel zadá neznámý kód aplikace mu to dá najevo.

Logika aplikace bude rozdělená do třech pomocných hooku. UseFirstTabActions, který bude obstarávat logiku pro první krok formuláře. Každý z kroků formuláře bude obsahovat vlastní react-hook-form kontext. UseSecondTabActions, ten bude obstarávat pro druhý krok formuláře a useRegistrationPageActions, kde bude definována logika pro celé zobrazení jako je uložení celého formuláře.

### První krok formuláře

První krok formuláře bude obsahovat pole jméno, příjmení, datum narození, telefonní číslo, e-mail, heslo, heslo znovu, všechny tyto pole budou povinná.

V logice bude definována pomocná funkce onEmailAlreadyInUse, která bude volána v useRegistrationPageActions, pokud již uživatel s emailem existuje.

```
1. const onEmailAlreadyInUse = () => {  
2.     firstTabFormContext.setError('email', {  
3.         type: 'custom',  
4.         message: 'Email, je již používán.',  
5.     })  
6. }
```

Kód 9 - Ukázka pomocné funkce onEmailAlreadyInUse, zdroj: autor

**Registrace**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent in mauris eu tortor porttitor accumsan. Aenean placerat. Et harum quidem rerum facilis est et expedita distinctio.

Jméno\*  
Tes

Příjmení\*  
Test

Datum narození\*  
10.02.2023

Telefonní číslo  
+420 774 446 542

Email\*  
testovaciemail@test.cz

Heslo\*  
.....

Heslo znovu\*  
.....

[Další →](#)

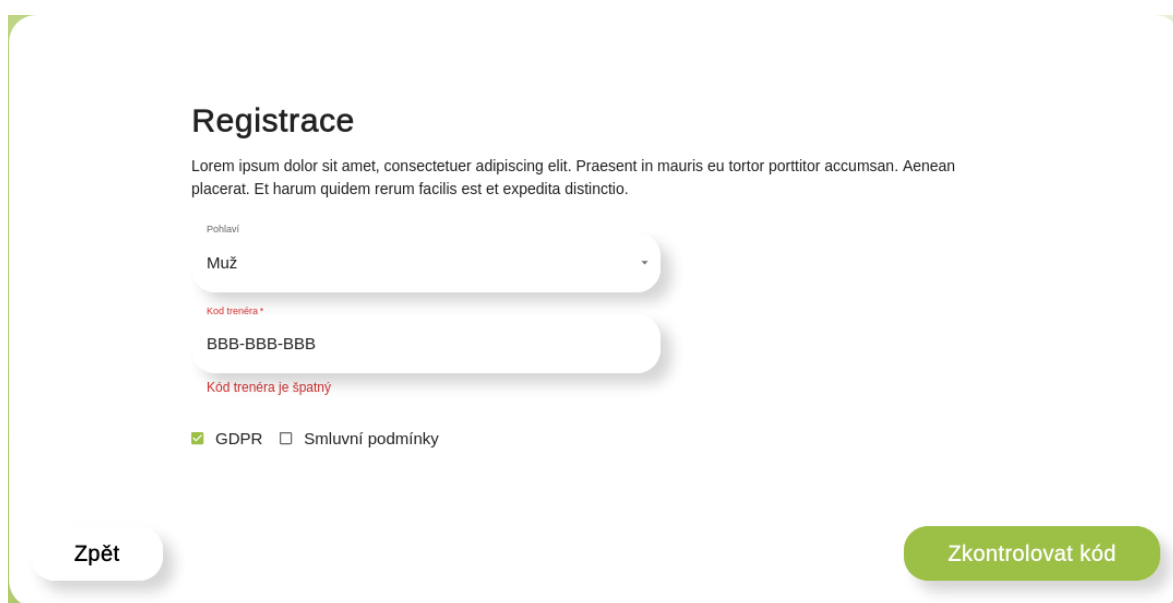
Obrázek 10 - Ukázka prvního kroku formuláře s vyplněnými daty, zdroj: autor

## Druhý krok formuláře

Druhý krok formuláře, bude obsahovat dvě pole. Pole pohlaví a pole kód trenéra. Logika bude taková, že při odeslání formuláře, bude zkontrolována z firebase firestore zda trenér s kódem existuje. Pokud existuje bude zobrazeno v uživatelském rozhraní jeho jméno. V opačném případě bude u pole pro kód trenéra (coachCode) nastavena chyba s textem, že kód trenéra je špatný.

```
1. if (_invalidCode) {  
2.   secondTabFormContext.setError('coachCode', {  
3.     type: 'custom',  
4.     message: 'Kód trenéra je špatný.',  
5.   })  
6. }
```

Kód 10 - Ukázka nastavení chyby v případě nevalidního kódu, zdroj: autor



Obrázek 11 - Ukázka uživatelského rozhraní při špatně zadaném kódu, zdroj: autor

## Odeslání formuláře

Při odeslání formuláře bude nejprve zkontrolováno, zda email existuje ve firebase, pokud existuje bude zavolána pomocná funkce `onEmailAlreadyInUse`. V opačném případě bude ve firebase vytvořen nový uživatel.

### 4.2.7.3 Zapomenuté heslo

Dle požadavků aplikace, obsahuje zobrazení Zapomenuté heslo pouze vstup pro email a tlačítka zpět a odeslat.

Logika u zobrazení Zapomenuté heslo, bude taková, že při odeslání formuláře, se pokusíme zavolat funkci `sendPasswordResetEmail` z knihovny firebase a pokud tato funkce selže zobrazíme uživateli informaci o tom, že email není v aplikaci evidován. Pokud funkce neseleže zobrazíme uživateli hlášku, že email s resetováním byl úspěšný.

## Zapomenuté heslo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent in mauris eu tortor porttitor accumsan. Aenean placerat. Et harum quidem rerum facilis est et expedita distinctio.

Email: \*

ahojda456@seznam.cz

Tento email nevidujeme.

Reset hesla

Obrázek 12 - Ukázka chybové hlášky u zobrazení Zapomenuté heslo, zdroj: autor

### 4.2.7.4 Nástěnka tréninků

Nástěnka tréninků je určena k informování klientů o tom, které tréninky je čekají. Nástěnka bude rozdělena do dvou bloků, první blok bude obsahovat nadcházející tréninky a další blok již uplynulé tréninky.

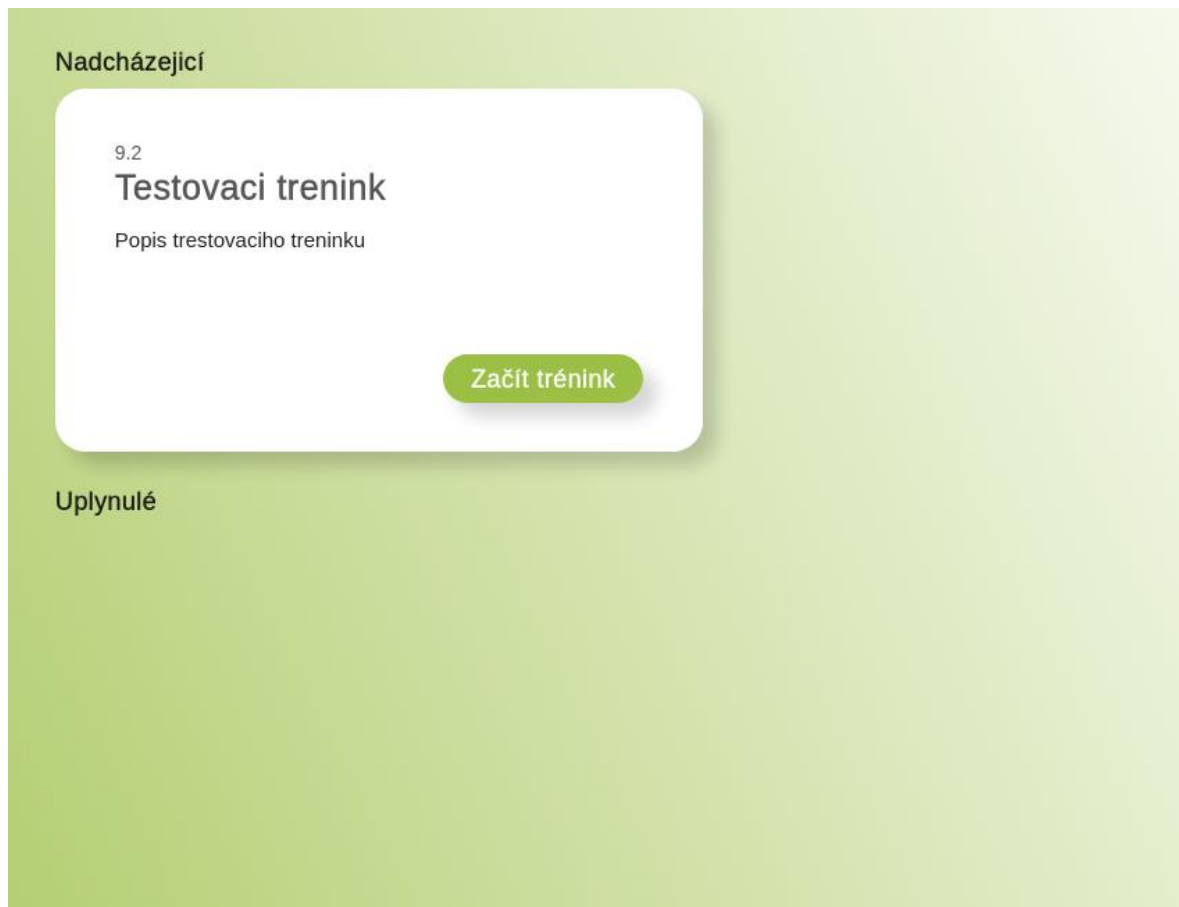
V logice zobrazení se bude nacházet proměna uložena ve statuu a zároveň funkce `onOpenTraining`, která bude mít jako parametr trénink, který má být otevřen.

Z firebase získáme dokončené a nedokončené tréninky a uložíme je do proměnných `finishedTrainings` a `unFinishedTrainings`. Obě tyto proměnné obsahují pole tréninku, v uživatelském rozhraní bude potřebovat je pro iterovat a vepsat je do Mui Grid komponentu.

```
1. <Typography variant="h3">Nadcházející</Typography>
2. <Grid
3.   container
4.   columnSpacing={12}
5.   rowSpacing={unFinishedTrainings?.length ? 3 : 0}>
6.   {unFinishedTrainings?.map((training) => (
7.     <Grid item key={training.id} xs={12} smx={6}>
8.       <TrainingBox
9.         training={training}
10.        onOpenTraining={onOpenTraining}
11.      />
12.    </Grid>
13.  )>
14. </Grid>
15. </Grid>
16.
17. <Typography variant="h3">Uplynulé</Typography>
18. <Grid
19.   container
20.   columnSpacing={12}
21.   rowSpacing={finishedOrderedTrainings?.length ? 3 : 0}>
22.   {finishedOrderedTrainings?.map((training) => (
23.     <Grid item key={training.id} xs={12} smx={6}>
24.       <TrainingBox
25.         training={training}
26.         onOpenTraining={onOpenTraining}
27.       />
28.     </Grid>
29.   )>
30. </Grid>
```

```
29.     </Grid>
30.   )})
31. </Grid>
```

Kód 11 - Ukázka výpisu tréninků na zobrazení Nástěnka tréninků, zdroj: autor



Obrázek 13 - Ukázka vzhledu nástěnky tréninků, zdroj: autor

Ve specifikacích je určeno, že uživatel má mít možnost si trénink otevřít. To bude umožňovat komponent Dialog, který byl definován v sekci Vývoj základních komponentů. Dialog bude uživateli poskytovat o tréninku informace jako jsou název tréninku, trenér, který vytvořil trénink a také základní informace o tréninku. Aby bylo možné využít tento komponent na více místech bude vytvořen univerzální komponent TrainingDetailDialog,

Tento komponent bude obsahovat props openedTraining a onOpenTraining. Prop openTraining bude uchovávat aktuální otevřený trénink, pokud bude tato proměna nedefinována, nebude Dialog zobrazován. Prop onOpenTraining, bude funkce sloužící k otevírání a zavírání Dialogu.

```
1. type Props = {
2.   openedTraining: WithId<UserTraining> | null
3.   onOpenTraining: (training: WithId<UserTraining>) => void
4. }
```

Kód 12 - Ukázka props u komponentu TrainingDetailDialog, zdroj: autor

V komponentu budou dále už jen vypsány atributy tréninku, jak bylo uvedeno.



```

1. export default function TrainingDetailDialog(props: Props) {
2.   const onClose = () => {
3.     props.onOpenTraining(undefined)
4.   }
5.
6.   return (
7.     <Dialog
8.       open={Boolean(props.openedTraining)}
9.       onClose={onClose}
10.      dialogTitle={props.openedTraining?.training.title}
11.    >
12.      <Typography variant="h3">Info</Typography>
13.      <div>{props.openedTraining?.training.info}</div>
14.      {props.openedTraining.coachName}
15.    </Dialog>
16.  )
17. }
18.

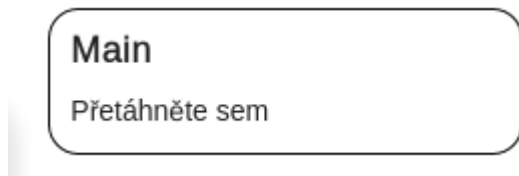
```

Kód 13 - Finální podoba komponentu TrainingDetailDialog, zdroj: autor

#### 4.2.7.5 Tvorba tréninku

Tvorba tréninku je zobrazení, které je z pohledu uživatelského rozhraní nejkompexnější. Skládá se z postranního prvku a hlavního prvku, kdy postranní prvek vypisuje cviky a hlavní prvek obsahuje pole kam může uživatel cviky přetahovat a zároveň jsou zde i vstupy pro název tréninku, základní informace, klient a datum tréninku.

Z pohledu UX je na zobrazení Tvorba tréninku důležité, aby uživatel při přetahování cviků viděl, ke které konkrétní kategorii může být aktuálně přetahovány cvik přetažen.



Obrázek 14 - Ukázka zvýraznění kategorii, do které může být prvek přetažen, zdroj: autor

Aplikace bude v budoucnu zobrazovat velké množství cviků, proto je důležité zajistit, aby projíždění cviků a jejich vyobrazení bylo rychlé. Dle Lentona je rychlost základní prvek k dosažení správného UX (Lenton 2018).

Pro vykreslení mnoha prvků je potřeba využít virtuální element, který nevykresluje všechny element, ale pouze elementy, které v danou chvíli uživatel vidí. To umožňuje knihovna react-window. Tato knihovna poskytuje dva pomocné komponenty Grid a List. Komponent Grid je určen pro případy, kdy potřebujeme zobrazit více sloupců. List je vhodný pro případy s pouze jedním sloupcem.

V případě popisované aplikace bude využit pomocný komponent List. Props u pomocného komponentu List jsou následovné:

- Šířka finálního prvku – width.
- Výška finálního prvku – height.

- Počet prvků, které bude List obsahovat – itemCount.
- Výška jednoho řádku v pixelech – itemSize.
- Funkci, která vykreslí každý jednotlivý prvek – children.

### Funkce getRenderRow

Funkce getRenderRow je pomocná funkce, jako parametr funkce je string určující, o kterou kategorii půjde, který může nabývat hodnot „warmUp“, „main“, „cooldown“ a návratovou hodnotou je funkce, která vykreslí dva prvky vedle sebe podle indexu, obdrženého z parametru funkce.

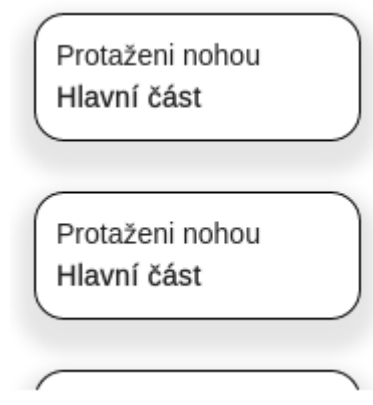
```

1. const getRenderRow = (type: CategoryType) => {
2.   return ({ index, style }) => (
3.     <div style={style}>
4.       <Grid container>
5.         <GridItem
6.           index={index * 2}
7.           category={type}
8.           categories={categories}
9.         />
10.        <GridItem
11.          index={index * 2 + 1}
12.          category={type}
13.          categories={categories}
14.        />
15.      </Grid>
16.    </div>
17.  )
18. }

```

Kód 14 - Ukázka funkce getRenderRow, zdroj: autor

Aby uživatel aplikace viděl, že cviků je více a může mezi nimi posouvat zvolíme výšku Listu na 200 pixelů a šířku jednotlivého řádku na 89 pixelů. To bude mít za výsledek to, že při prvním načtení zobrazení budou vidět dva prvky plus malá část dalšího, tím uživateli dojde, že prvků je více.



Obrázek 15 - Ukázka vykreslení virtuálního listu cviků, zdroj: autor

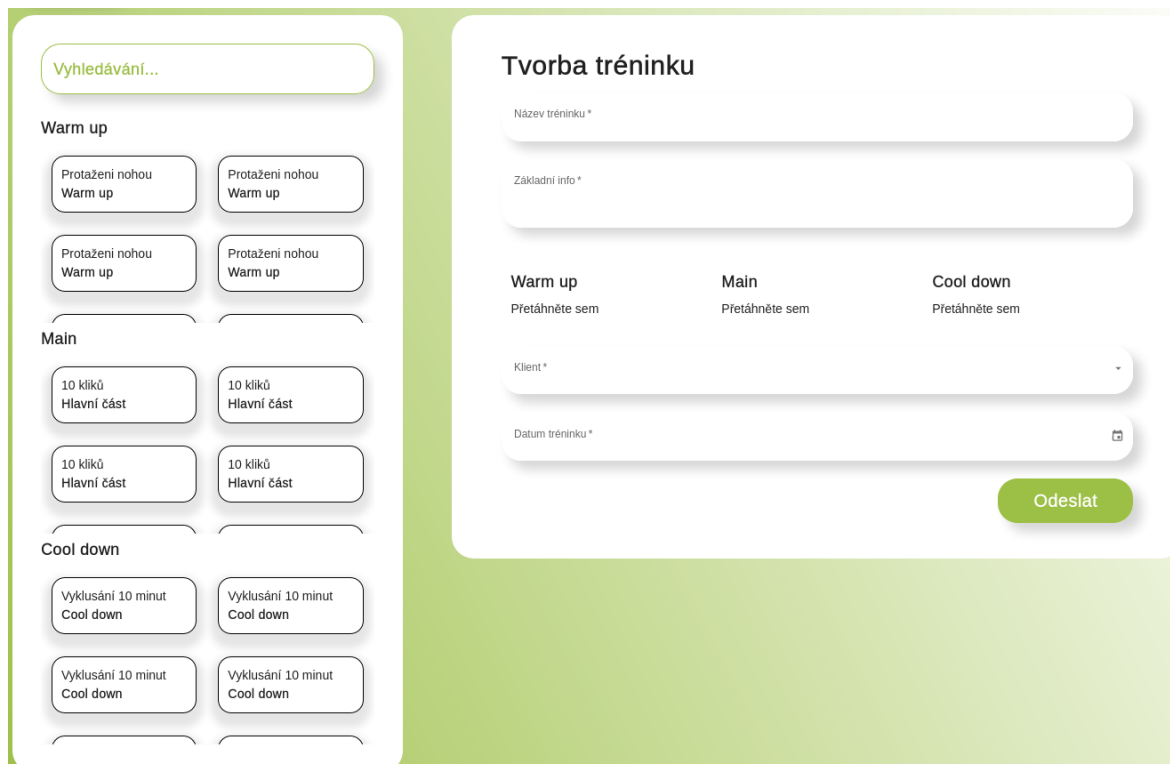
Počet prvků, které bude List obsahovat zjistíme jednoduše. Z objektu, kde je jako klíč název kategorii a hodnota, pole všech cviků stačí přechíst atribut length, který představuje počet prvků tohoto pole.

```

1. <List
2.     width={640}
3.     height={200}
4.     itemCount={Math.ceil(
5.         categories[ExerciseCategoryTypes.COOLDOWN].length / 2
6.     )}
7.     itemSize={89}
8.     >
9.     {getRenderRow(ExerciseCategoryTypes.COOLDOWN)}
10. </List>

```

Kód 15 - Ukázka použití komponentu List z knihovny react-window, pro kategorii Cool down, zdroj: autor



Obrázek 16 - Ukázka zobrazení Tvorba tréninku, zdroj: autor

V požadavcích zobrazení Tvorba tréninku je uvedeno, že zobrazení má poskytovat ke každému cviku možnost nastavit si pro tento cvik čas nebo počet opakování a pauzu pro každou sérii. To bude docíleno pomocí modálního okna, v kterém bude možnost tyto atributu k cviku přiřadit.

V modálním oknu bude muset být navázáno na to, zda cvik umožňuje nastavit počet opakování a počet sérii, pokud to cvik umožňuje budou zobrazeny pole pro vstup. V případě, kdy bude cvik umožňovat nastavení jak počtu opakování, tak počtu sérii bude zobrazeno přepínací tlačítko, která bude mezi pole pro vstupy přepínat.

```

1. {openedExercise?.editQuantity === 'ANO' &&
2.   openedExercise?.editTime === 'ANO' && (
3.     <FormControllLabel
4.       value="top"
5.       control={
6.         <Switch
7.           color="primary"
8.           checked={checked}
9.           onChange={handleChange}
10.          />
11.        }
12.       label={'Čas / opakování'}
13.       labelPlacement="top"
14.     />
15.   )}

```

Kód 16 - Ukázka přepínacího tlačítka v modálním okně konfigurace cviku, zdroj: autor

The screenshot shows a modal window with the following elements:

- Title:** Protažení nohou
- Info:** A white rounded rectangle containing the word 'Info'.
- Čas / opakování:** A toggle switch that is currently turned off.
- Počet serií \*:** A text input field with a dropdown arrow on the right.
- Čas serie \*:** A text input field with a clock icon on the right.
- Pauza \*:** A text input field with a clock icon on the right.
- Uložit:** A green rounded button at the bottom.

Obrázek 17 - Ukázka modálního okna sloužícího k konfiguraci cviků, zdroj: autor

### 4.3 Testování

Prototyp aplikace, který byl vytvořen, bude otestován, tak, že testovacímu subjektu bude zaslána adresa, kde je prototyp aplikace zpřístupněn. Testování bude probíhat nemoderovaně.

Spolu s adresou mu bude zaslán i testovací scénář s výslednou zprávou z testování. Každý testovací subjekt projde aplikaci podle scénáře a u každého bodu zaznamená do výsledné zprávy, zda se mu krok povedl popřípadě, jestli k němu má nějaké připomínky. Pro testování bylo vybráno 6 testovacích subjektů, kdy tyto subjekty bylo rozděleny do dvou skupin po 3 subjektech. Skupiny jsou trenéři a klienti, kdy každá skupina bude testovat specifické funkce aplikace.

Testovací subjekty byly vybrány tak, aby splňovali cílovou skupinu aplikace, tedy lidé, kteří se zajímají o fitness. Každý testovací subjekt vidí prototyp aplikace poprvé, aby byly zajištěny korektní výsledky testování.

#### **4.3.1 Funkce podrobeny testování**

Aby testování bylo přínosné a závěry z něj byly věcné, je nutné, abychom si stanovili, které funkce prototypu aplikace budou testovány. Aplikace zobrazuje většinu podstatnějších funkcí až potom co uživatel projde přihlášením nebo se zaregistruje. Proto budou jako první testovány zobrazení Registrace a Přihlášení, dále bude testování rozděleno na zobrazení, která má přístupné trenér a zobrazení pro klienta.

##### **Přihlášení**

Zobrazení přihlášení je hlavním zobrazením aplikace. Toto zobrazení je shodné jak pro trenéry, tak klienty. Proto bude přihlášení testováno v obou případech.

##### **Zapomenuté heslo**

Zobrazení Zapomenuté heslo jsou spolu se zobrazením Přihlášení jediná zobrazení, která nejsou vázány na to, zda je uživatel klient nebo trenér. Proto bude také testováno v obou případech.

##### **Registrace**

Zobrazení Registrace představuje hlavní vstup, jak se uživatelé mohou do aplikace připojit. Je tedy důležité že je průchod tímto zobrazením svižný. U testování registrace, bude kladen důraz na vybírání kódu trenéra, protože tato funkcionalita, by pro uživatele nemusela být jasná.

##### **Trenér – Nástěnka tréninků**

Sekce nástěnka tréninků, představuje pro uživatele, který je trenér hlavní sekci, kde vidí přehled o tom. Kdy mají jeho klienti naplánované tréninky. Zde bude kladen důraz na to, zda uživateli dojde, jak trénink otevřít.

### **Trenér – Tvorba tréninku**

Tvorba tréninku je pro uživatele, který je trenér podstatnou částí. Umožňuje mu vytvořit a naplánovat svým svěřencům, trénink s různými cviky. Zde je důležité otestovat, zda uživatel ví, jak jednotlivé cviky přiřadit ke tréninku. Také je důležité otestovat nastavení konfigurací pro jednotlivé cviky.

### **Klient – Nástěnka tréninků**

Sekce nástěnka tréninků, představuje pro klienta sekci, kde vidí, které tréninky má naplánované. Zde bude kladen důraz na to, zda uživatel ví, jak si dány trénink zobrazí a zjistí informace o něm.

#### **4.3.2 Testovací scénář**

Na základě funkcí, které mají být otestovány, bude stanoven testovací scénář, který bude odevzdán uživatelům, kteří budou součástí testování. Testování probíhá tak, že uživateli je zadán úkol spolu s pod úkoly a uživatel vyplní do testovací zprávy, zda byl schopen úkol splnit, popřípadě zda má k úkoly nějaké poznatky.

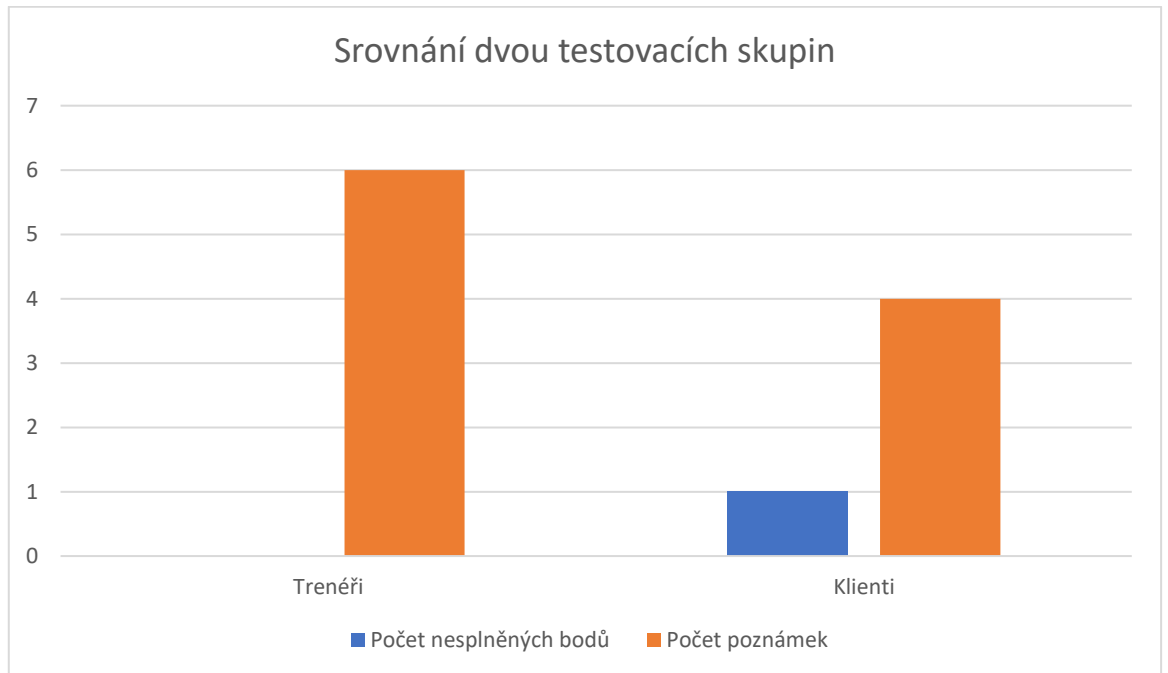
#### **4.3.3 Výsledná zpráva**

Výsledná zpráva bude předmětem k analýze pro následné zlepšení aplikace. Je proto důležité, aby zpráva obsahovala všechny výstupná hodnocení, které potřebujeme. U každého úkolu u jednotlivých bodů testování bude kolonka, zda se úkol testovacímu subjektu povedlo splnit, poznámky k úkolu a pokud je úkol otázka, odpověď k této otázce.

## 4.4 Zhodnocení testování

Po uskutečnění testování byly všechny výsledky autorem zpracovány a následně analyzovány. Dle analýzy bylo zjištěno, že krom jednoho kroku, který nebyl jeden testovací subjekt schopný splnit, byly ostatní kroky úspěšně.

Dle analýzy bylo také zjištěno, že u skupiny trenérů bylo dohromady 6 poznámek k bodům u skupiny klientů byly poznámky 4. To může značit, že sekce pro klienta je lépe navržena.



Graf 1 - Ukázka výsledků testování, zdroj: autor

#### 4.4.1 Analýza chyb v UX

Všechny poznámky od respondentů byly shromážděny do tabulky za účelem provést analýzu, objevit poznámky, které jsou duplicitní a z toho složit funkcionality aplikace, které potřebují změny, aby byly z pohledu UX optimální.

Text poznámky	Testovací skupina
Nebyl jsem schopný zadat datum, zkoušel jsem ho napsat, ale to nešlo.	Klienti
U datumu nelze napsat datum ručně i když je po najetí na něj ikonka, „psát“, ikonka země u bodu s telefonem je lehce posunutá dolů, není proto vidět tak zřetelně.	Klienti
Při zadání datum jsem tápal, protože ikona na zadání byl a malá v rohu.	Klienti
Ze začátku jsem neviděl tlačítko odhlásit.	Klienti
Informační hláška mi přijde nepřehledná.	Trenéři
Byl jsem na stránce Zapomenuté heslo a nevěděl jsem, jak se vrátit zpět až jsem to provedl tlačítkem zpět v prohlížeči.	Trenéři
Tápal jsem nad tím, jak nastavit ke cviku dodatečné údaje.	Trenéři
Vyhledávání jednotlivého cviku je pro všechny tři sekce v jednom hledáčku.	Trenéři
Když chci datum napsat, hodí se čárka nesmyslně před číslo.	Trenéři
Chvilí jsem nemohl najít tlačítko na změnu na opakování raději bych to dal blíže oknu kterého se změna týká.	Trenéři

Tabulka 4 - Analýza chyb v UX, zdroj: autor

Dle analýzy byly stanoveny typy nevhodných komponent aplikace na následující:

- Zadávání datumů v aplikaci
- Tlačítko odhlásit v aplikaci
- Vzhled informační hlášky u zobrazení Přihlášení
- Zobrazení Zapomenuté heslo neobsahuje tlačítko zpět
- Konfigurace dodatečných údajů ke cviku
- Vyhledávání na zobrazení Tvorba tréninku
- Přepínání Čas série a počet sérii v konfiguraci dodatečných údajů na zobrazení Tvorba tréninku

#### 4.4.2 Řešení nalezených chyb

##### Vhodné řešení chyby Zadávání datumů v aplikaci



V aplikaci je jako vstup pro datum využíván komponent `DatePickerElement` z knihovny `react-hook-form-mui`. Při otestování autorem bylo zjištěno chybné chování komponentu stejně jako popisovali respondenti. Není možné zadat datumy pomocí klávesnice.

Dle sekce `Issues` u repositáře `react-hook-form-mui` hlásilo tuto chybu více vývojářů, který tuto knihovnu používali. Autor knihovny píše, že ve verzi 5.14.0 už by tato chyba měla být opravena (Dohomi 2022).

V aplikaci je používána verze knihovny `react-hook-form-mui` 5.12.4, proto by k vyřešení této chyby mělo stačit tuto knihovny aktualizovat na nejnovější verzi 5.15.1.

Po aktualizaci knihovny se již tato chyba neprojevuje.

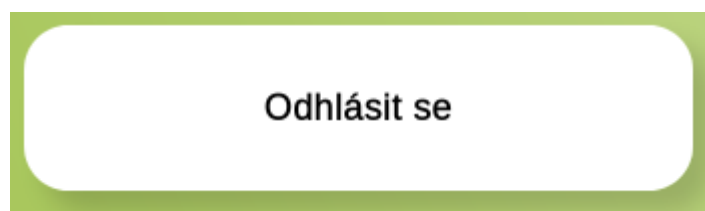
### **Vhodné řešení chyby Tlačítko odhlásit v aplikaci**

Tlačítko odhlásit je v aplikaci zobrazeno jako odkaz stejně jako například odkaz `Nástěnka tréninku`. To může být klamné, protože odhlášení v aplikaci je akce, proto by mělo být vykresleno jako tlačítko.



*Obrázek 18 - Odhlášení v aplikaci, zdroj: autor*

Vhodné řešení je tedy předělat tlačítko na vzhled tlačítek, které jsou v aplikaci. Zároveň by toto tlačítko mělo být zobrazeno jako sekundární protože, to není hlavní akce na stránce.



*Obrázek 19 - Vhodná úprava tlačítka Odhlásit se, zdroj: autor*

### **Vhodné řešení chyby Vzhled informační hlášky u zobrazení Přihlášení**

V aplikaci je chybová hláška, že přihlášení nebylo úspěšné vykreslena stejnou barvou jako je v aplikaci vykreslen text.

Špatné jméno/heslo  
Email \*

asd@asd.cz

Heslo \*

●●●●●●●●

Obrázek 20 - Ukázka chybové hlášky při přihlášení, zdroj: autor

Dle Vivka Kumara by chybové hlášky měly mít barvu, která je zvýrazní (Kumar 2018). Proto je vhodné pro vyřešení této chyby změnit barvu chybové hlášky na červenou.

Špatné jméno/heslo  
Email \*

asd@asd.cz

Heslo \*

●●●●●●●●

Obrázek 21 - Ukázka zvýraznění hlášky, zdroj: autor

### Vhodné řešení chyby Zobrazení Zapomenuté heslo neobsahuje tlačítko zpět

Zobrazení zapomenuté heslo neobsahuje tlačítko na vrácení zpět na zobrazení Přihlášení.

Zapomenuté heslo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent in mauris eu tortor porttitor accumsan. Aenean placerat. Et harum quidem rerum facilis est et expedita distinctio.

Email: \*

Reset hesla

Obrázek 22 - Chybný vzhled zobrazení Zapomenuté heslo, zdroj: autor

To je chyba, která může být vyřešena přidáním tlačítka zpět.

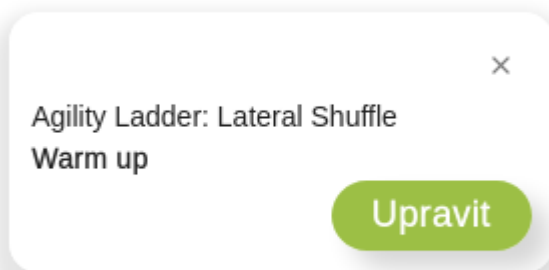


Obrázek 23 - Opravený vzhled zobrazení zapomenuté heslo, zdroj: autor

### Vhodné řešení chyby Konfigurace dodatečných údajů ke cviku

Když uživatel přetáhne cvik na zobrazení Tvorba tréninku musí poté ještě kliknout u cviku na tlačítko Upravit. To je nevhodné chování, protože dodatečné údaje u cviku by správně uživatel měl vyplnit hned.

#### Warm up



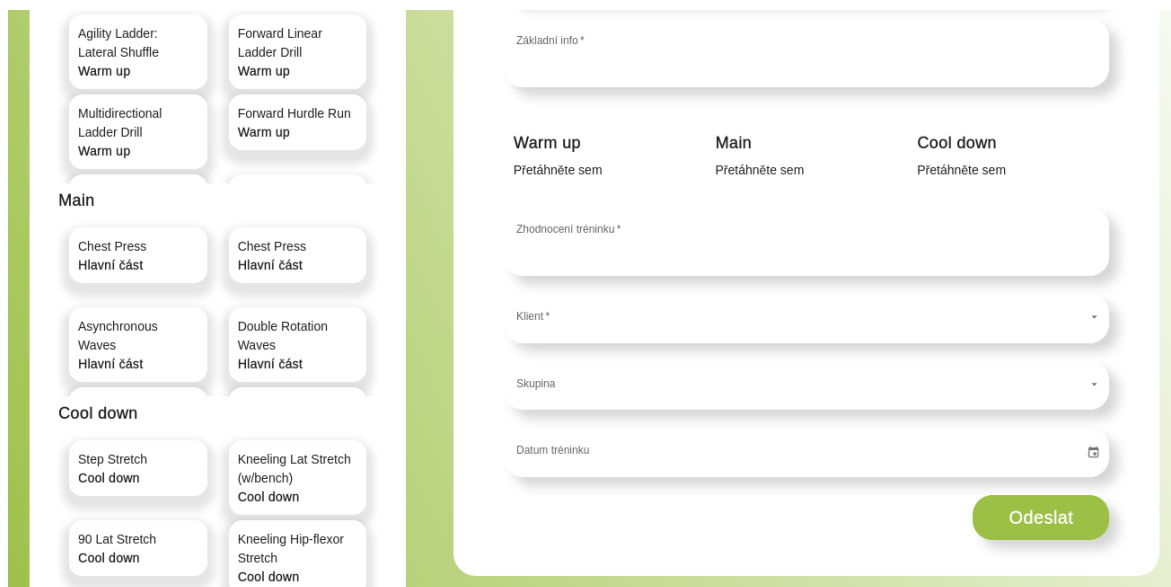
Přetáhněte sem

Obrázek 24 - Vzhled po přetažení cviku, zdroj: autor

Proto je správné řešení takové, že hned potom co uživatel cvik přetáhne, se mu otevře modální okno na nastavení konfigurovatelných údajů u cviku.

### Vhodné řešení chyby Vyhledávání na zobrazení Tvorba tréninku

Vyhledávání cviků u zobrazení Tvorba tréninku je v současné chvíli nahoře nad všemi cviky, to může způsobovat, že pokud uživatel sjede na stránce dolů toto vyhledávání nevidí.

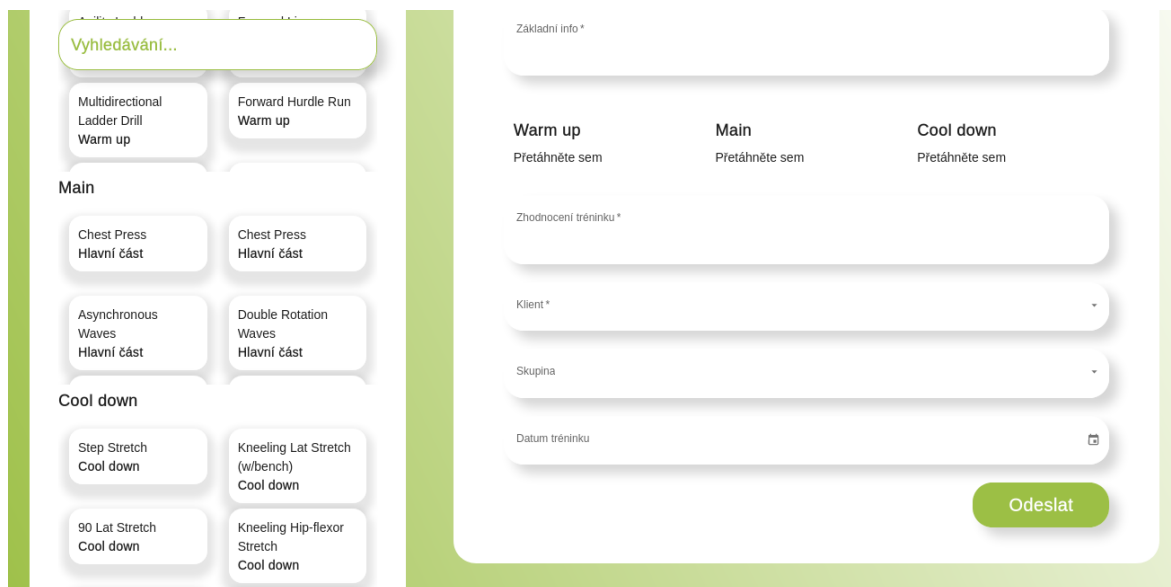


Obrázek 25 - Skryté vyhledávání. zdroj: autor

Řešení tohoto problému, by mohlo být přidat k vyhledávacímu elementu css atribut `position: sticky`. Tím docílíme, že při scrolování bude vždy vyhledávací prvek nahoře.

1. `position: sticky;`
2. `top: 1rem;`
3. `z-index: 1000;`

Kód 17 - Ukázka pozměněných css atributů, zdroj: autor



Obrázek 26 - Opravené vyhledávání, zdroj: autor

## 5 Zhodnocení a doporučení

Prototyp, který byl vytvořen, obsahuje všechny předem stanovené funkcionality. Vývoj probíhal postupně, začínaje nastavením základních parametrů a pokračováním definováním barev v prototypu. Cílem bylo vytvořit přehledný prototyp, který bude uživatelům snadno použitelný a srozumitelný. Po dokončení vývoje byla aplikace testována uživateli, kteří poskytli zpětnou vazbu na prototyp. Tato zpětná vazba byla kladná a uživatelé uvedli, že by rádi používali prototyp aplikace.

Prototyp aplikace byl otestován testovacími subjekty, kdy se z testování podařilo zjistit nedostatky. Zároveň, žádný z testovacích subjektů nevykázal žádné větší problémy s plněním úkolů. To dokazuje, že testovací scénář byl sestaven, tak, aby objevil nedostatky v aplikaci a zároveň, aby umožnil testování uskutečnit hladce.

Testování probíhalo nemonerovaně, na 6 testovacích subjektech, kdy každý z testovacích subjektů byl vybrán, aby odpovídal cílové skupině aplikace. Testování probíhalo podle testovacího scénáře a výsledky byly každým subjektem vloženy do testovací zprávy, která byla v další části analyzována.

Z chyb, které byly objeveny během testování testovacími subjekty, byla každá chyba autorem prověřena a identifikována. Po identifikaci chyb bylo pro každou chybu stanoveno řešení, které by tuto chybu mohlo eliminovat. Řešení chyb obsahuje i řešení, které doporučuje aktualizaci knihovny, která je používána. To ukazuje, že hledání řešení bylo provedeno do hloubky a bylo ukázáno více způsobů, jak chyby v řešit.

Aby se z prototypu stala plnohodnotná aplikace, měli by být doplněny sekce jako jsou Můj účet a Průběh tréninku pro klienta. Přidání těchto sekcí, by vzhledem k dobré struktuře aplikace mělo být uskutečnitelné. Dále by mělo proběhnout další uživatelské testování za účelem, aby každé vylepšení, které bylo stanoveno bylo otestováno a zjištěno, zda opravdu opravuje chybu, ke které se vztahuje.

## 6 Závěr

Na základě provedené analýzy odborné literatury se podařilo navrhnout a vytvořit prototyp webové aplikace. Při tvorbě prototypu byl dbán důraz na to, aby byl prototyp správně navržen z pohledu UI. Tento prototyp používá v danou chvíli nejnovější knihovny, ať už se jedná o knihovnu pro obecnou funkcionalitu nebo o knihovnu umožňující správnou validaci formulářů. Postup, který byl použit pro tvorbu prototypu může být využit i v jiných případech, protože si zakládá na knihovnách, které jsou mezi vývojáři rozšířené a do budoucna budou rozšiřovány.

U prototypu, který byl vytvořen, byly stanoveny části, u kterých je testování podstatné. Na základě částí, které se mají otestovat byl sestaven testovací scénář. Tento testovací scénář byl využit k uživatelskému testování. Z tohoto uživatelského testování byly testovacími subjekty vyplněny výsledné zprávy.

Z analýzy výsledných zpráv byly objeveny chyby, které aplikace obsahuje a mohli by být opraveny. Tyto chyby se převážně týkali třech kategorií, interakce uživatele s aplikací, grafického designu aplikace a funkčnosti aplikace. Chyby byly autorem objeveny v aplikaci a u každé chyby byly stanoveny vylepšení, které by mohli každou chybu opravit.

Postup, který byl v práci stanoven k zjištění nedostatků webové aplikace je obecný a je možné ho využít k testování, kterékoliv webové aplikace.

## 7 Seznam použitých zdrojů

ALSHAYBAN, A. a S. MALEK, 2022. AccessiText: automated detection of text accessibility issues in Android apps. In: *ESEC/FSE 2022 - Proceedings of the 30th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering* [online]. s. 984–995. ISBN 978-1-4503-9413-0. Dostupné z: [doi:10.1145/3540250.3549118](https://doi.org/10.1145/3540250.3549118)

AMAURY, Ricardo, 2022. *Imperative v Declarative UI Design - Is Declarative Programming the future?* [online] [vid. 2023-02-17]. Dostupné z: <http://www.rootstrap.com/blog/imperative-v-declarative-ui-design-is-declarative-programming-the-future>

ANDYBELL, 2023. Pseudo-classes. *web.dev* [online] [vid. 2023-02-21]. Dostupné z: <https://web.dev/learn/css/pseudo-classes/>

BABICH, Nick, 2019a. A/B Testing Best Practices & Examples | Adobe XD Ideas. *Ideas* [online]. [vid. 2022-09-11]. Dostupné z: <https://xd.adobe.com/ideas/process/user-testing/the-role-of-ab-testing-in-product-design/>

BABICH, Nick, 2019b. The Top 7 Usability Testing Methods | Adobe XD Ideas. *Ideas* [online]. [vid. 2022-09-11]. Dostupné z: <https://xd.adobe.com/ideas/process/user-testing/top-7-usability-testing-methods/>

BABICH, Nick, 2020. What Is UX Design? User Experience Definition | Adobe XD Ideas. *Ideas* [online]. [vid. 2022-09-10]. Dostupné z: <https://xd.adobe.com/ideas/career-tips/what-is-ux-design/>

BOIA, 2019. *What is color contrast?* [online] [vid. 2023-02-14]. Dostupné z: <https://www.boia.org/blog/what-is-color-contrast>

BOIA, 2021. *Designing for Color Contrast: Guidelines for Accessibility* [online] [vid. 2023-02-14]. Dostupné z: <https://www.boia.org/blog/designing-for-color-contrast-guidelines-for-accessibility>

BOX UK, 2022. Unboxing: laboratory usability testing | Insight | Box UK. *www.boxuk.com* [online]. [vid. 2022-09-11]. Dostupné z: <https://www.boxuk.com/insight/unboxing-laboratory-usability-testing/>

DOHOMI, 2022. DatePickerElement bugged when entering the date by keyboard · Issue #91 · dohomi/react-hook-form-mui. *GitHub* [online] [vid. 2023-03-05]. Dostupné z: <https://github.com/dohomi/react-hook-form-mui/issues/91>

EXPIRIENCE, World Leaders in Research-Based User, 2022. Remote Moderated Usability Tests: How to Do Them. *Nielsen Norman Group* [online] [vid. 2022-09-11]. Dostupné z: <https://www.nngroup.com/articles/moderated-remote-usability-test/>

FACEBOOK, 2023. *Built-in React Hooks* [online] [vid. 2023-03-05]. Dostupné z: <https://beta.reactjs.org/reference/react>

- FREECODECAMP.ORG, 2020. CSS Unit Guide: CSS em, rem, vh, vw, and more, Explained. *freeCodeCamp.org* [online] [vid. 2023-02-18]. Dostupné z: <https://www.freecodecamp.org/news/css-unit-guide/>
- FULLSESSION, 2021. Top 8 Session Recording and Replay Tools [Our 2021 Review]. *FullSession* [online] [vid. 2022-09-11]. Dostupné z: <https://www.fullsession.io/blog/session-recording-replay-tools/>
- HANNAH, Jaye, 2022a. *Color Theory And Color Palettes — A Complete Guide [2023]* [online] [vid. 2023-02-12]. Dostupné z: <https://careerfoundry.com/en/blog/ui-design/introduction-to-color-theory-and-color-palettes/>
- HANNAH, Jaye, 2022b. *What Is Typography, and Why Is It Important? [2023 Guide]* [online] [vid. 2023-02-12]. Dostupné z: <https://careerfoundry.com/en/blog/ui-design/beginners-guide-to-typography/>
- HARSH, Agarwal, 2018. ReactJS Components. *ReactJS Components* [online]. [vid. 2023-02-17]. Dostupné z: <https://www.geeksforgeeks.org/reactjs-components/>
- HAYERBEKE, Marijn, 2019. *Eloquent JavaScript: a modern introduction to programming*. Third edition. San Francisco: No Starch Press. ISBN 978-1-59327-950-9.
- HOWE, Shay, 2014. *Learn to Code HTML and CSS: Develop and Style Websites*. B.m.: New Riders. ISBN 0-13-347757-6.
- INITIATIVE (WAI), W3C Web Accessibility, 2020. Introduction to Web Accessibility. *Web Accessibility Initiative (WAI)* [online] [vid. 2023-02-14]. Dostupné z: <https://www.w3.org/WAI/fundamentals/accessibility-intro/>
- INITIATIVE (WAI), W3C Web Accessibility, 2022. WCAG 2 Overview. *Web Accessibility Initiative (WAI)* [online] [vid. 2023-02-14]. Dostupné z: <https://www.w3.org/WAI/standards-guidelines/wcag/>
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2019. *ISO 9241-210:2019(en), Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems* [online] [vid. 2022-09-11]. Dostupné z: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>
- JAHODA, Bohumil, 2015. *Single page aplikace* [online] [vid. 2022-09-10]. Dostupné z: <https://jecas.cz/spa>
- KOŽDOUSKOVÁ, Barbora, 2021. Web, webová stránka a webová aplikace, v čem je rozdíl? *Rascasone* [online] [vid. 2022-09-06]. Dostupné z: <https://www.rascasone.com/cs/blog/web-webova-aplikace-rozdil>
- KUMAR, Gaurav a Pradeep Kumar BHATIA, 2014. Comparative analysis of software engineering models from traditional to modern methodologies. In: *2014 Fourth International Conference on Advanced Computing & Communication Technologies*. B.m.: IEEE, s. 189–196.



- KUMAR, Vivek, 2018. Creating Error Messages | Best practice in UX Design. *Medium* [online] [vid. 2023-03-05]. Dostupné z: <https://uxdesign.cc/creating-error-messages-best-practice-in-ux-design-cda3be0f5e16>
- LENTON, Mitch, 2018. Good Performance is Good Design: The 5 principles of Performant Design. *Medium* [online] [vid. 2023-03-02]. Dostupné z: <https://uxplanet.org/good-performance-is-good-design-the-5-principles-of-performant-design-54ca44c7c4e0>
- LIGERTWOOD, Guy, 2022. Guerrilla Testing: Hallway Usability Tests for UX | Adobe XD Ideas. *Ideas* [online]. [vid. 2022-09-11]. Dostupné z: <https://xd.adobe.com/ideas/process/user-testing/hallway-usability-test-guerrilla-testing/>
- LYONNAIS, Sheena, 2022. Color Contrast Considerations for Accessibility Design & UX | Adobe XD Ideas. *Ideas* [online]. [vid. 2023-02-14]. Dostupné z: <https://xd.adobe.com/ideas/principles/web-design/color-contrast-considerations-accessibility-design/>
- MOHAN, Mehul a Narayan PRUSTY, 2018. *Learn ECMAScript: discover the latest ECMAScript features in order to write cleaner code and learn the fundamentals of JavaScript*. Second edition. Birmingham, UK: Packt Publishing. ISBN 978-1-78862-962-1.
- MORTENSEN, Ditte Hvas, 2022. User Research: What It Is and Why You Should Do It. *The Interaction Design Foundation* [online] [vid. 2022-09-11]. Dostupné z: <https://www.interaction-design.org/literature/article/user-research-what-it-is-and-why-you-should-do-it>
- MOZILLA, 2023a. *Attribute selectors - Learn web development | MDN* [online] [vid. 2023-02-18]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors/Attribute\\_selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Attribute_selectors)
- MOZILLA, 2023b. *Type, class, and ID selectors - Learn web development | MDN* [online] [vid. 2023-02-18]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors/Type\\_Class\\_and\\_ID\\_Selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Type_Class_and_ID_Selectors)
- MOZILLA, 2023c. *What is CSS? - Learn web development | MDN* [online] [vid. 2023-02-18]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/What\\_is\\_CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS)
- NEOTERIC, 2018. Single-page application vs. multiple-page application. *Medium* [online]. [vid. 2022-09-10]. Dostupné z: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>
- NIELSEN, Jakob, 2020. 10 Usability Heuristics for User Interface Design. *Nielsen Norman Group* [online] [vid. 2023-02-19]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- PILGRIM, Mark, 2010. Dive into HTML5. URL: <http://diveintohtml5.info/index.html> (дата обращения: 11.10. 2013).
- REACT, 2018. *React – A JavaScript library for building user interfaces* [online] [vid. 2023-02-17]. Dostupné z: <https://reactjs.org/>

- RUMMAN, 2022. Everything you should know about 8 point grid system in UX design. *Medium* [online] [vid. 2023-02-17]. Dostupné z: <https://uxplanet.org/everything-you-should-know-about-8-point-grid-system-in-ux-design-b69cb945b18d>
- SALAZAR, Kim, 2020. Contextual Inquiry: Inspire Design by Observing and Interviewing Users in Their Context. *Nielsen Norman Group* [online] [vid. 2022-09-11]. Dostupné z: <https://www.nngroup.com/articles/contextual-inquiry/>
- SHERWIN, Katie, 2018. Card Sorting: Uncover Users' Mental Models for Better Information Architecture. *Nielsen Norman Group* [online] [vid. 2022-09-11]. Dostupné z: <https://www.nngroup.com/articles/card-sorting-definition/>
- SCHADE, Amy, Yunnuo CHENG a Samyukta SHERUGAR, 2016. Top 10 Enduring Web-Design Mistakes. *Nielsen Norman Group* [online] [vid. 2022-09-11]. Dostupné z: <https://www.nngroup.com/articles/top-10-enduring/>
- TUBIK, 2018. Golden Ratio. Bring Balance in UI Design. *Medium* [online] [vid. 2023-02-15]. Dostupné z: <https://uxplanet.org/golden-ratio-bring-balance-in-ui-design-765c954f0ff9>
- VALLAURE, Christine, 2023. Ultimate Guide to Color in UX/UI Design. *Medium* [online] [vid. 2023-02-13]. Dostupné z: <https://uxplanet.org/ultimate-guide-to-color-in-ux-ui-design-de8eb104b5d3>
- VAN ESSENDELFT, Abigail, 2020. The Golden Ratio: Mathematics in Nature and Artl.
- VERCEL, 2022. *Basic Features: Pages / Next.js* [online] [vid. 2023-03-12]. Dostupné z: <https://nextjs.org/docs/basic-features/pages#server-side-rendering>
- VERCEL, 2023. *Next.js by Vercel - The React Framework* [online] [vid. 2023-03-05]. Dostupné z: <https://nextjs.org/>
- WHITENTON, Kathryn, 2019. Unmoderated User Tests: How and Why to Do Them. *Nielsen Norman Group* [online] [vid. 2022-09-11]. Dostupné z: <https://www.nngroup.com/articles/unmoderated-usability-testing/>

## 8 Seznam obrázků, tabulek, grafů a zkratek

### 8.1 Seznam obrázků

Obrázek 1 - Porovnání MPA a SPA, zdroj: <a href="https://lvivity.com/wp-content/uploads/2020/08/spa-mpa-lifecycle.jpg">https://lvivity.com/wp-content/uploads/2020/08/spa-mpa-lifecycle.jpg</a> .....	13
Obrázek 2- Newtonovo barevné kolo, zdroj: <a href="https://dpbnri2zg3lc2.cloudfront.net/en/wp-content/uploads/old-blog-uploads/color-wheel-issac-ne-qdxan.jpg">https://dpbnri2zg3lc2.cloudfront.net/en/wp-content/uploads/old-blog-uploads/color-wheel-issac-ne-qdxan.jpg</a> .....	15
Obrázek 3 - Monochromatická metoda výběru, zdroj: <a href="https://miro.medium.com/v2/resize:fit:720/format:webp/1*cx3Wq94OBG5lprxQWPQDA.jpeg">https://miro.medium.com/v2/resize:fit:720/format:webp/1*cx3Wq94OBG5lprxQWPQDA.jpeg</a> .....	16
Obrázek 4 - Komplementární metoda výběru, zdroj: <a href="https://miro.medium.com/v2/resize:fit:720/format:webp/1*9EhaxHHMGmqOc8yJz9_c5w.jpg">https://miro.medium.com/v2/resize:fit:720/format:webp/1*9EhaxHHMGmqOc8yJz9_c5w.jpg</a> .....	16
Obrázek 5 - Použití zlatého řezu v UI, zdroj: <a href="https://miro.medium.com/v2/resize:fit:720/format:webp/0*ZE70D_z3RBzwjmwz.png">https://miro.medium.com/v2/resize:fit:720/format:webp/0*ZE70D_z3RBzwjmwz.png</a> .....	18
Obrázek 6 - Ukázka použití 8bodové mřížky, zdroj: <a href="https://miro.medium.com/v2/resize:fit:720/format:webp/1*qR-NIVfU5XtHocXdpKThPw.png">https://miro.medium.com/v2/resize:fit:720/format:webp/1*qR-NIVfU5XtHocXdpKThPw.png</a> .....	19
Obrázek 7 - Základní barvy v aplikaci, zdroj: autor .....	32
Obrázek 8 - Ukázka vzhledu tlačítka, vlevo sekundární, vpravo primární, zdroj: autor .....	34
Obrázek 9 - Ukázka výsledného vzhledu zobrazení přihlášení, zdroj: autor .....	36
Obrázek 10 - Ukázka prvního kroku formuláře s vyplněnými daty, zdroj: autor .....	37
Obrázek 11 - Ukázka uživatelského rozhraní při špatně zadaném kódu, zdroj: autor .....	38
Obrázek 12 - Ukázka chybové hlášky u zobrazení Zapomenuté heslo, zdroj: autor .....	39
Obrázek 13 - Ukázka vzhledu nástěnky tréninků, zdroj: autor .....	40
Obrázek 14 - Ukázka zvýraznění kategorií, do které může být prvek přetažen, zdroj: autor .....	41
Obrázek 15 - Ukázka vykreslení virtuálního listu cviků, zdroj: autor .....	42
Obrázek 16 - Ukázka zobrazení Tvorbě tréninku, zdroj: autor .....	43
Obrázek 17 - Ukázka modálního okna sloužícího k konfiguraci cviků, zdroj: autor .....	44
Obrázek 18 - Odhlášení v aplikaci, zdroj: autor .....	49
Obrázek 19 - Vhodná úprava tlačítka Odhlásit se, zdroj: autor .....	49
Obrázek 20 - Ukázka chybové hlášky při přihlášení, zdroj: autor .....	50
Obrázek 21 - Ukázka zvýraznění hlášky, zdroj: autor .....	50
Obrázek 22 - Chybný vzhled zobrazení Zapomenuté heslo, zdroj: autor .....	50
Obrázek 23 - Opravený vzhled zobrazení zapomenuté heslo, zdroj: autor .....	51
Obrázek 24 - Vzhled po přetažení cviku, zdroj: autor .....	51
Obrázek 25 - Skryté vyhledávání. zdroj: autor .....	52
Obrázek 26 - Opravené vyhledávání, zdroj: autor .....	52

### 8.2 Seznam tabulek

Tabulka 1 - Ukázka atributových selektorů, zdroj: autor .....	21
Tabulka 2 - Příklad scénáře při Guerrilla testování, zdroj: autor .....	27
Tabulka 3 - Příklad poznámek při Guerrilla testování, zdroj: autor .....	27
Tabulka 4 - Analýza chyb v UX, zdroj: autor .....	48

### 8.3 Seznam grafů

Graf 1 - Ukázka výsledků testování, zdroj: autor .....	47
--	----

### 8.4 Seznam použitých kódů

Kód 1 - Ukázka párových a nepárových elementů v HTML, zdroj: autor .....	19
Kód 2 - Ukázka použití Selektorů dle typu, třídy nebo ID, zdroj: autor .....	20
Kód 3 - Ukázka využití kontextu v Reactu, zdroj: autor .....	24
Kód 4 - Ukázka konfigurace cviků, zdroj: autor .....	31
Kód 5 - Nastavení atributů u MuiCssBaseLine, zdroj: autor .....	33
Kód 6 - Pomocný hook useLoading, zdroj: autor .....	35
Kód 7 - Ukázka vytvoření form kontextu pro přihlášení, zdroj: autor .....	35
Kód 8 - Ukázka tlačítek na zobrazení přihlášení, zdroj: autor .....	36
Kód 9 - Ukázka pomocné funkce onEmailAlreadyInUse, zdroj: autor .....	37
Kód 10 - Ukázka nastavení chyby v případě nevalidního kódu, zdroj: autor .....	38
Kód 11 - Ukázka výpisu tréninků na zobrazení Nástěnka tréninků, zdroj: autor .....	40
Kód 12 - Ukázka props u komponentu TrainingDetailDialog, zdroj: autor .....	40
Kód 13 - Finální podoba komponentu TrainingDetailDialog, zdroj: autor .....	41
Kód 14 - Ukázka funkce getRenderRow, zdroj: autor .....	42
Kód 15 - Ukázka použití komponentu List z knihovny react-window, pro kategorii Cool down, zdroj: autor .....	43
Kód 16 - Ukázka přepínacího tlačítka v modálním okně konfigurace cviku, zdroj: autor .....	44
Kód 17 - Ukázka pozměněných css atributů, zdroj: autor .....	52

### 8.5 Seznam použitých zkratk

**AJAX** – Technologie webových aplikací umožňující asynchronní přenos dat mezi webovým serverem a klientem.

**HTTP** – Je protokol, který se používá pro přenos informací na webu.

**JSON** – Formát pro výměnu dat, který je čitelný pro lidi i počítače.

**URL** – Jedná se o adresu webové stránky, nebo jiného zdroje na internetu.

**FRONTEND** – Představuje klientskou část webové aplikace.

## **Přílohy**

### **Příloha 1 Testovací scénář klient**

#### **Klient – Registrace a přihlášení**

- a) Otevřete si webovou aplikaci na adrese app.traineee.cz.
- b) Přejděte na stránku registrace.
- c) Vyplňte svoje osobní údaje a kód trenéra, který je X3W-OI0-XXA.
- d) Zaregistrujte se.
- e) Co se stalo po odeslání registrace?

#### **Klient – zapomenuté heslo**

- a) Odhlaste se z aplikace.
- b) Přejděte na stránku Přihlášení.
- c) Zapomněli jste svoje heslo a chcete si vygenerovat nové.
- d) Resetujte svoje heslo pomocí emailů, který jste zadali při registraci.

#### **Klient – Detail tréninku**

- a) Znovu se přihlaste do aplikace.
- b) Přejděte na sekci Nástěnka tréninků.
- c) Otevřete si první trénink.
- d) Jak se jmenuje trenér, který vystavil trénink?

## **Příloha 1 Testovací scénář trenér**

### **Trenér – Přihlášení**

- a) Přihlaste se pomocí údajů trener1@test.cz a 123456789.
- b) Bylo přihlášení úspěšné?
- c) Přihlaste se pomocí údajů trener1@test.cz a 12345678910.
- d) Co se stalo po přihlášení?

### **Trenér – Zapomenuté heslo**

- a) Odhlaste se.
- b) Přejděte na stránku Zapomenuté heslo.
- c) Resetujte si heslo pro email trener1@test.cz.
- d) Co se zobrazilo po odeslání požadavku pro resetování hesla?

### **Trenér – Tvorba tréninku**

- a) Znovu se přihlaste údaji trener1@test.cz a 12345678910.
- b) Přejděte na sekci Vytvořit trénink.
- c) Přidejte si “Lateral Hurdle Run” do sekce Wam up a nastav počet sérii na 3, počet opakování na 12 a pauzu na 00:30.
- d) Přidejte si cvik “Asynchronous Waves” do sekce Main a nastav počet sérii na 3, čas série na 00:45 a pauzu na 00:30.
- e) Přidejte si cvik “Step Stretch” do sekce Cool down a nastav počet sérii na 3, čas série na 00:40 a pauzu na 00:15.
- f) Trénink se bude jmenovat Testovací trénink bude určen pro Testovací klient 1 a bude vázán na datum 20.12.2022.
- g) Odešlete trénink.