

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

**Informační systém, který funguje jako virtuální IT
support**

Zdeněk Kohout

© 2021 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Zdeněk Kohout

Systémové inženýrství a informatika
Informatika

Název práce

Informační systém, který funguje jako virtuální IT support

Název anglicky

Web application, that functions as a virtual IT support

Cíle práce

Navrhnete HI-FI prototyp aplikace, která umožní efektivní sdílení znalostí a postupů v rámci IT podpory.

Metodika

Teoretická část bakalářské práce se bude skládat ze studia odborných informačních zdrojů. Na základě tohoto zjištění budou popsány technologie, které budou použity při vývoji technologie a popisu, jak byla tato technologie využita. Postupujte dle následující metodiky:

- Provedte logický návrh aplikace
- Návrh databáze
- Grafický návrh aplikace
- Prototyp
- Naprogramování Hi-Fi prototypu
- Definování závěrů

Doporučený rozsah práce

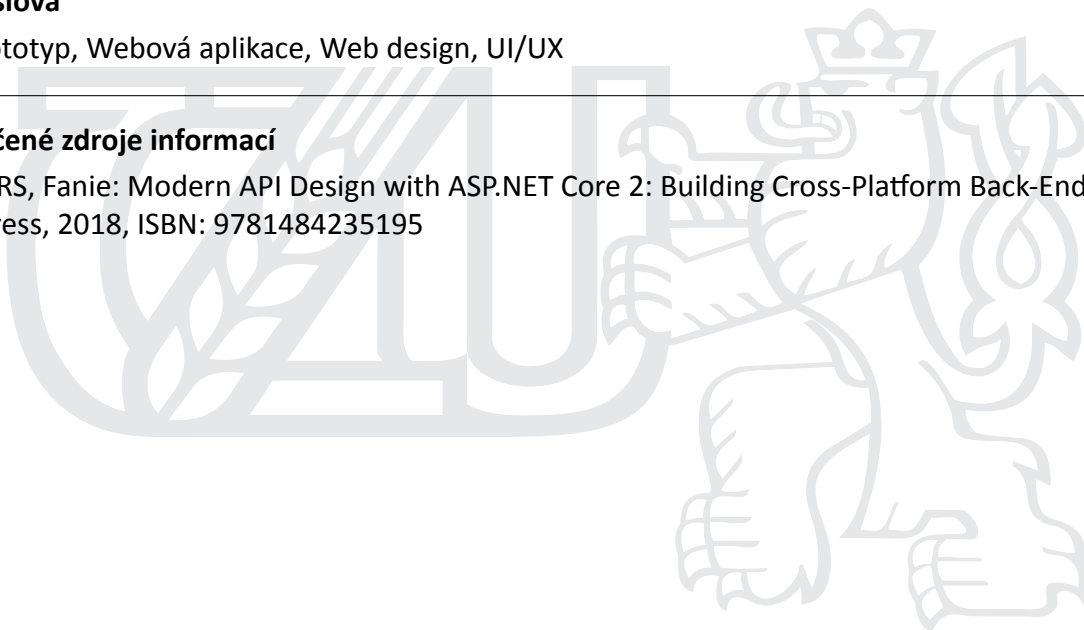
30-40

Klíčová slova

Hi-Fi prototyp, Webová aplikace, Web design, UI/UX

Doporučené zdroje informací

REYNDERS, Fanie: Modern API Design with ASP.NET Core 2: Building Cross-Platform Back-End Systems, Apress, 2018, ISBN: 9781484235195



Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Josef Pavlíček, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 13. 03. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Informační systém, který funguje jako virtuální IT support“ jsem vypracoval 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. 2021

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce Ing. Josefu Pavlíčkovi, Ph.D. za ochotu, vstřícnost a cenné rady při psaní této práce.

Informační systém, který funguje jako virtuální IT support

Abstrakt

Bakalářská práce je zaměřena na vývoj Hi-Fi prototypu webové aplikace, která funguje jako virtuální IT support. Práce se skládá z teoretické a praktické části a zhodnocení výsledků. Teoretická část přináší statistická data, díky kterým je možné specifikovat cílovou skupinu a navrhnout požadované persony. Dále se teoretická část soustředí na představení a zhodnocení základních technologií, které se používají při vývoji Hi-Fi prototypu.

Praktická část práce se skládá z potřebných návrhů, které byly následně použity při vývoji Hi-Fi prototypu. Nejprve byl navrhnout logický návrh, poté podoba dat, a nakonec byl sestaven prvotní návrh uživatelského rozhraní. Následně byly vypsány Use Casy, které se budou v aplikaci nacházet. Dále byla dle návrhu vytvořena struktura aplikace. Byly definovány jednotlivé komponenty struktury a byla popsána jejich funkčnost. Poté byl naprogramován Hi-Fi prototyp a nastýlován pomocí jazyka SCSS. Výsledný produkt byl poté v simulovaném prostředí podroben uživatelskému testování. Z výsledků a osobních interview s testovanými subjekty byly vyvozeny závěry a navrhnutá vylepšení aplikace.

Klíčová slova: Hi-Fi prototyp, informační systémy, IT support, prototyp, web design, webová aplikace, UI/UX

Web application, that works as a virtual IT support

Abstract

The bachelor thesis is focused on the development of a Hi-Fi prototype of a web application that works as a virtual IT support. The work consists of theoretical and practical part and evaluation of results. The theoretical part provides statistical data, thanks to which it is possible to specify the target group and design the required persons. Furthermore, the theoretical part focuses on the introduction and evaluation of basic technologies used in the development of Hi-Fi prototypes.

The practical part of the work consists of the necessary designs, which were then used in the development of a Hi-Fi prototype. First, a logical design was designed, then the form of the data, and finally the initial design of the user interface was constructed. Subsequently, the Use Cases, which will be in the application, were listed. Furthermore, the structure of the application was created according to the design. The individual components of the structure were defined and their functionality was described. Then the Hi-Fi prototype was programmed and styled using the SCSS language. The resulting product was then subjected to user testing in a simulated environment. Conclusions were drawn from the results and personal interviews with the tested subjects and improvements to the application were suggested.

Keywords: Hi-Fi prototype, information systems, IT support, prototype, web design, web application, UI/UX

Obsah

1 Úvod.....	9
2 Cíl práce a metodika	11
2.1 Cíl práce	11
2.2 Metodika	11
3 Teoretická východiska	12
3.1 Statistická analýza	12
3.1.1 Digitální dovednosti a práce se softwarem	12
3.1.2 Neformální vzdělávání v České republice	15
3.2 Vývoj webové aplikace	15
3.2.1 HTML	15
3.2.2 CSS	16
3.2.3 JS.....	16
3.2.4 Vue.js	16
3.2.5 CRUD	16
3.2.6 Uživatelské rozhraní (UI)	17
3.2.7 Uživatelský prožitek (UX).....	17
3.2.8 Prototyp.....	17
3.2.9 Technologie pro vývoj Hi-Fi prototypu	18
3.2.9.1 Adobe XD.....	18
3.2.9.2 Figma.....	18
3.2.9.3 Axure RP	19
3.2.9.4 Grafický návrh + frontend	19
4 Vlastní práce	21
4.1 Specifikace požadavků	21
4.1.1 Použité technologie.....	22
4.1.1.1 Figma.....	22
4.1.1.2 NodeJS.....	22
4.1.1.3 NuxtJS	22
4.1.1.4 SCSS.....	22
4.1.2 Vývoj Hi-Fi prototypu	23
4.1.2.1 Logický design uživatelského rozhraní	23
4.1.2.2 Logický návrh ukládaných dat	25
4.1.2.3 Pohled uživatele.....	28

4.1.2.4	Pohled administrátora	30
4.1.2.5	Prvotní návrh uživatelského rozhraní	32
4.2	Naprogramování Hi-Fi prototypu	36
4.2.1	Vytvoření struktury projektu	36
4.2.2	Dodatečné stylování uživatelského rozhraní	38
4.3	Cílová skupina.....	40
4.4	Persony	40
4.4.1	Persona A	40
4.4.2	Persona B	41
4.4.3	Persona C	42
4.4.4	Antipersona	42
4.5	Testování	43
4.5.1	Testovací formulář	44
5	Výsledky a diskuse	46
5.1	Výsledky	46
5.1.1	Navrhovaná vylepšení aplikace	47
6	Závěr.....	48
7	Seznam použitých zdrojů	50
8	Přílohy	52

Seznam obrázků

Obrázek 1 - Ilustrace vývoje Hi-Fi prototypu. Zdroj: Modular Design Frameworks: A Projects-based Guide for UI/UX Designers.....	20
Obrázek 2 – návrh přihlašovací obrazovky. Zdroj: vlastní zpracování	33
Obrázek 3 - návrh domovské stránky. Zdroj: vlastní zpracování	34
Obrázek 4 – návrh formuláře pro vytvoření návodu. Zdroj: vlastní zpracování	34
Obrázek 5 – návrh pracovního návodu. Zdroj: vlastní zpracování.....	35
Obrázek 6 – návrh administrátorského panelu. Zdroj: vlastní zpracování	36
Obrázek 7 – globální styly v NuxtJS. Zdroj: vlastní zpracování	38
Obrázek 8 - příklad vytvořených SCSS mixinů. Zdroj: vlastní zpracování	39
Obrázek 9 – globální SCSS proměnné v NuxtJS. Zdroj: vlastní zpracování	40

Seznam tabulek

Tabulka 1 – Statistické údaje o digitálních dovednostech. Zdroj: ČSÚ	13
Tabulka 2 – Statistické údaje o digitálních dovednostech. Zdroj: ČSÚ	14

1 Úvod

Požadavky na IT odborníky stále rostou, v mnoha firmách však musí trávit čas děláním triviálních úkonů, při kterých není využita jejich odbornost. Z vlastní zkušenosti znám neustále se opakující problémy s tiskárnami, MS Office, nebo připojením k internetu. Aplikace, do které by se dala všechna tato data nahrát, by tak mohla pomoci a celý proces zjednodušit. Koncept této aplikace je především ten, že nestojí pouze na IT odbornících, ale data do ní mohou vkládat i ostatní zaměstnanci, například nadřízení a manažeři. Zkrátí se tak doba potřebná pro zaučování nebo zdlouhavé žádosti o opravu něčeho, co vlastně rozbité není.

Aplikace bude navrhována tak, aby byla dostupná všem. Grafická podoba aplikace má cílit na všechny věkové skupiny zaměstnanců. Návrh představí graficky čistou aplikaci, která neobsahuje přebytečná a matoucí tlačítka, nebo funkce, použité písmo bude poměrně velké a navigace bude jednoduchá a srozumitelná. U popisů navržených problémů bude vyvinuta snaha o co největší jasnost a přehlednost popsanou v pár bodech.

Pokud se zamyslíme i nad pandemickou dobou, kterou kolem sebe již druhým rokem máme, aplikace by byla i dobrým nástrojem online přenosu informací. V žádné své části nepotřebuje fyzickou přítomnost obou stran, data se dají přidávat a přijímat odkudkoli.

Momentálně se na trhu ale nikdo této problematice nevěnuje, existuje velmi mnoho aplikací, které slouží k nahrávání různých pracovních postupů a kdokoli je může využívat. Nejznámější takovou aplikací je YouTube, který je podobných návodů plný. Bohužel ale ani zde nelze nalézt odborné, a hlavně spolehlivé informace, jelikož přidávání videí není nikým kontrolováno, a tak může na YouTube přidat kdokoli cokoli. Další aplikací, která se zabývá problematikou nahrávání návodů a různých postupů je Pluralsight, ta se však zabývá pouze technickou stránkou věci. Naleznete zde přednášky ověřených odborníků, které se věnují především technologiím. Pokud se ovšem v dané problematice vyskytne nějaký problém, uživatel musí pomoc hledat jinde. Pluralsight na rozdíl od YouTube není zadarmo a pro firmy nemá smysl platit jeho licenci, jelikož nenabízí kromě odborných přednášek nic navíc. Samozřejmě je ještě možnost vyhledat si problém na jednom z mnoha internetových prohlížečů, jako například Google Chrome. Tento postup je poměrně běžný

v odvětví informačních technologií, avšak někdy je těžké rozlišit, který výsledek vyhledávání je pro daný problém nejlepší. Opětovné hledání stejného problému je také časově náročné. Vedle toho je tu problematika vnitřních originálních programů, které si firma nechá vyrobit na míru. Pro takové programy nelze nalézt návody na internetu. I zaměstnanci většinou pracují pouze s interními programy a setkávají se s opakovanými problémy poměrně často. Kdyby firma měla program, který umožňuje snadno a rychle přidávat návody a odborné postupy, ušetřily by se dané firmě nejen náklady na IT odborníky, ale zároveň by se pracovní výkon zaměstnanců zefektivnil a zaměstnanci by nepřicházeli o čas, který by jinak využili k odborné práci.

Myslím si, že aplikace by byla velmi vhodná a využitelná nejen díky zmíněným kvalitám, ale také pro svou jednoduchost. Pro zaměstnance by nešlo o naprosto novou technologii, jelikož svou grafickou strukturou bude připomínat jiné známé formáty a prostředí.

2 Cíl práce a metodika

2.1 Cíl práce

Návrh HI-FI prototypu aplikace, která umožní efektivní sdílení znalostí a postupů v rámci IT podpory.

2.2 Metodika

Teoretická část bakalářské práce se bude skládat ze studia odborných informačních zdrojů. Na základě těchto zjištění budou popsány technologie, které budou použity při vývoji technologie a popisu, jak byla tato technologie využita. Postupujeme podle následující metodiky:

- Bude proveden logický návrh aplikace
- Návrh databáze
- Grafický návrh aplikace
- Prototyp
- Naprogramování Hi-Fi prototypu
- Definování závěrů

3 Teoretická východiska

Vlastní práce

3.1 Statistická analýza

3.1.1 Digitální dovednosti a práce se softwarem

Začněme s daty z Českého statistického úřadu z roku 2019, která nám lépe přiblíží uživatele naší aplikace. Navrhovaná aplikace má sloužit jako pomůcka firmám a v jejich struktuře pak IT oddělení, které většinu technických problémů musí řešit osobně. Problémy se často opakují, a tím se tak často zbytečně plýtvá časem odborníků. Statistický úřad uvádí, že přestože od roku 2006 do roku 2018 vzrostl počet IT odborníků, více než dvojnásobně (z 1,8 % na 3,9 %), firmy „stále hlásí nedostatek IT expertů, které by potřebovaly pro rozvoj své činnosti v oblasti digitalizace.“¹

Další důležitá data, která přímo vypovídají o naší cílové skupině, říkají, že běžné činnosti na počítači, kam řadíme přenos souborů mezi počítačem a jiným zařízením nebo kopírování souborů či složek v rámci počítače, provádělo v posledním roce 51 % osob starších 16 let. Činnosti spojené s kopírováním mezi počítači pak provádělo ještě o několik procent méně lidí (47 %). U složitějších úkonů, jako je instalace softwaru, čísla značně klesají, za sledovaný rok to bylo 29 % lidí. U činností s programy, které jsou ve většině zaměstnání nezbytné pro výkon povolání, jako je MS Word, MS Excel, prezentační softwaru, programy na úpravu fotografií, se už čísla pohybují od 54 % k 18 %. Z dat také vyplývá, že gramotnost se zvyšuje s kvalifikovaností dané profese. „Například mezi manažery nalezneme 92 % osob, které v posledním roce kopírovaly soubory či složky, tabulkový procesor využilo 90 % z nich a v programovacím jazyku psalo 14 %.“² Naproti tomu jsou samozřejmě nekvalifikovaní zaměstnanci, u nichž se čísla pohybují mnohem níže (kopírování souborů či složek 34 %, využití tabulkového procesoru 19 % a u programování méně než půl procenta). Rozdíly jsou pak uváděny i ve spojitosti s pohlavím zaměstnanců. Podrobná data naleznete v příložených tabulkách.

¹ 2. Digitální dovednosti a práce se softwarem. ČSÚ. Český statistický úřad. ČSÚ [online]. Praha: © 2019. [cit. 28. 2. 2020]. Dostupné z: <https://www.czso.cz/documents/10180/90577057/06200419k02.pdf/c25ec46b-0d58-46c7-b7c6-8ed836b79866?version=1.1>.

² Tamtéž.

Tabulka 1 – Statistické údaje o digitálních dovednostech. Zdroj: ČSÚ

	Textový procesor (např. MS Word)			Prezentační software (např. MS PowerPoint)			Software či aplikace na úpravu fotek		
	v tis.	% 1)	% 2)	v tis.	% 1)	% 2)	v tis.	% 1)	% 2)
Celkem 16+	4 708,9	53,7	65,9	1 583,3	18,0	22,2	2 796,6	31,9	39,2
Pohlaví									
Muži 16+	2 309,6	54,6	64,9	876,7	20,7	24,6	1 435,2	33,9	40,3
Ženy 16+	2 399,2	52,8	67,0	706,7	15,5	19,7	1 361,4	29,9	38,0
Věková skupina									
16–24 let	759,6	88,0	89,4	487,4	56,5	57,4	598,1	69,3	70,4
25–34 let	993,7	72,3	74,2	371,8	27,1	27,8	781,0	56,9	58,4
35–44 let	1 138,3	67,1	68,8	333,4	19,6	20,2	698,3	41,1	42,2
45–54 let	900,1	60,5	64,0	240,7	16,2	17,1	369,2	24,8	26,2
55–64 let	592,9	45,5	55,3	125,1	9,6	11,7	235,8	18,1	22,0
65–74 let	269,8	21,4	40,4	23,7	1,9	3,5	95,2	7,6	14,3
75+	54,5	6,9	35,4	1,3	0,2	0,8	19,0	2,4	12,3
Vzdělání (25–64 let)									
Základní	48,5	14,0	20,3	5,6	1,6	2,4	54,1	15,6	22,7
Střední bez maturity	738,9	36,6	41,4	74,5	3,7	4,2	404,2	20,0	22,7
Střední s maturitou + VOŠ	1 675,1	75,6	77,1	418,5	18,9	19,3	888,1	40,1	40,9
Vysokoškolské	1 162,5	90,9	91,2	572,4	44,8	44,9	737,9	57,7	57,9
Ekonomická aktivita (16+)									
Zaměstnaní	3 362,8	66,5	68,8	1 008,9	20,0	20,7	1 909,7	37,8	39,1
Ženy v domácnosti*	256,9	62,4	65,0	52,7	12,8	13,3	223,8	54,3	56,7
Studenti	635,2	94,2	95,0	463,0	68,6	69,3	480,4	71,2	71,9
Starobní důchodci	361,1	15,8	37,8	32,0	1,4	3,3	122,8	5,4	12,9
Invalidní důchodci	49,5	21,7	34,1	9,3	4,1	6,4	28,5	12,5	19,6

Tabulka 2 – Statistické údaje o digitálních dovednostech. Zdroj: ČSÚ

	Instalace programu do počítače			Stažení aplikace do telefonu nebo tabletu			Programování		
	v tis.	% 1)	% 2)	v tis.	% 1)	% 2)	v tis.	% 1)	% 2)
Celkem 16+	2 509,6	28,6	35,1	3 870,2	44,1	54,2	497,6	5,7	7,0
Pohlaví									
Muži 16+	1 592,0	37,6	44,7	2 097,6	49,6	58,9	373,8	8,8	10,5
Ženy 16+	917,6	20,2	25,6	1 772,6	39,0	49,5	123,8	2,7	3,5
Věková skupina									
16–24 let	511,3	59,3	60,2	754,2	87,4	88,8	117,0	13,6	13,8
25–34 let	673,7	49,1	50,3	1 031,8	75,1	77,1	156,2	11,4	11,7
35–44 let	642,1	37,8	38,8	1 023,6	60,3	61,9	112,3	6,6	6,8
45–54 let	397,8	26,7	28,3	626,5	42,1	44,5	61,6	4,1	4,4
55–64 let	207,8	16,0	19,4	320,9	24,6	30,0	40,7	3,1	3,8
65–74 let	65,9	5,2	9,9	102,8	8,2	15,4	8,4	0,7	1,3
75+	11,0	1,4	7,1	10,5	1,3	6,9	1,4	0,2	0,9
Vzdělání (25–64 let)									
Základní	31,4	9,0	13,1	88,2	25,4	36,9	2,5	0,7	1,1
Střední bez maturity	372,6	18,4	20,9	766,9	38,0	43,0	28,0	1,4	1,6
Střední s maturitou + VOŠ	833,8	37,6	38,4	1 255,9	56,7	57,8	148,6	6,7	6,8
Vysokoškolské	683,7	53,5	53,6	891,6	69,7	69,9	191,7	15,0	15,0
Ekonomická aktivita (16+)									
Zaměstnaní	1 858,3	36,8	38,0	2 792,4	55,2	57,2	355,7	7,0	7,3
Ženy v domácnosti*	102,8	24,9	26,0	251,9	61,2	63,8	17,5	4,2	4,4
Studenti	425,6	63,1	63,7	603,9	89,5	90,3	110,9	16,4	16,6
Starobní důchodci	76,6	3,4	8,0	132,4	5,8	13,9	13,6	0,6	1,4
Invalidní důchodci	19,4	8,5	13,4	46,3	20,3	31,9	-	-	-

3.1.2 Neformální vzdělávání v České republice

Další statistické údaje, které pro naši aplikaci považujeme za důležité, jsou data o neformálním vzdělávání v České republice. Naše cílová skupina již nespadá do povinné školní docházky, musí se vzdělávat v programech neformálního vzdělávání. Bohužel v době, kdy starší zaměstnanci navštěvovali školní prostředí, výuka informačních technologií nebyla, nebo se zabývala věcmi, které jsou dnes již nevyužitelné. V ideálním případě by nová generace mohla mít lepší základy, ale to není předmětem našeho zkoumání. Neformální výuka probíhá formou kurzů, školení, soukromých lekcí, seminářů a workshopů. Tato forma výuky je více méně dobrovolná, často jednorázová a od formálního vzdělávání se velmi liší. Podle statistických údajů se v roce 2011 neformálního vzdělávání zúčastnila téměř třetina osob od 18 do 69 let. Tento vzorek osob však nebyl reprezentativní a mnoho účastníků mělo vyšší odborné nebo vysokoškolské vzdělání. Motivace těchto osob souvisela především s výkonem povolání (78 %). Z dat vyplývá, že z nabízených kurzů je nejčastěji volena výuka jazyků a rozvíjení právního povědomí. O teoretickou či praktickou přípravu na příchozí a zaváděné výrobní technologie, či ovládání nových přístrojů a náradí, byl zájem 11 % dotazovaných, což není zanedbatelné.

3.2 Vývoj webové aplikace

Webová aplikace je interaktivní počítačový program. Je vytvořena pomocí webových technologií (HTML, CSS, JS). Jejím úkolem je ukládat data do databáze a pracovat s nimi pomocí operací CRUD. Webová aplikace je přístupná skrze internetový prohlížeč, jako je například Google Chrome, Mozilla Firefox a další.

3.2.1 HTML

HTML umožňuje webovým vývojářům vytvářet strukturu webové stránky pomocí speciálních značek (
; ; <div>). HTML není programovací jazyk, tudíž není možné při jeho využívání vytvářet dynamické funkce.

3.2.2 CSS

CSS určuje vzhled, a tím tak dodává webové stránce na přívětivosti. S CSS lze kontrolovat téměř všechny vzhledové aspekty webové stránky, jako například barvu písma, fonty, animace nebo různá odsazení elementů. Pomocí CSS lze upravovat i responzivní design, to znamená, že je možné definovat speciální stylování pro různé velikosti obrazovek.

3.2.3 JS

JavaScript je textový programovací jazyk používaný jak na straně klienta, tak na straně serveru. Zatímco HTML a CSS jsou jazyky, které dodávají webovým stránkám strukturu a styl, JS poskytuje webovým stránkám interaktivní prvky, které uživatele zaujmou.

Začlenění JavaScriptu vylepšuje uživatelské prostředí webové stránky tím, že ji převádí ze statické stránky na interaktivní.

3.2.4 Vue.js

Vue.js je volně dostupný, výkonný a progresivní javascriptový framework, který umožňuje vytvářet Single-page aplikace. První zdrojový kód frameworku Vue.js byl vytvořen v roce 2013 Evanem You, který v tu dobu pracoval pro Google. Vue.js umožňuje vytváření komponent, ze kterých se postupně skládá celá aplikace. Komponenta je Vue instancí, která se skládá ze tří částí: HTML, script a style. Komponentu lze použít na jedné ze stránek nebo v jiné komponentě. V posledních letech se z Vue.js stal silný konkurent pro Angular a React, a to zejména s vydáním Vue 3.0. Vue.js je oblíbený i u mnoha větších firem, které si ho zvolily za svůj primární frontendový framework.

3.2.5 CRUD

CRUD je akronym, který odkazuje na čtyři základní funkce, které jsou považovány za nezbytné pro práci s databází. První z nich je Create, tato funkce umožňuje v databázi vytvářet záznamy a ukládat je do tabulek. Funkce Read slouží pro čtení záznamů, které jsou v databázi uloženy. Update aktualizuje záznamy v databázi. A poslední funkce – Delete umožňuje záznamy z databáze mazat.

3.2.6 Uživatelské rozhraní (UI)

Uživatelské rozhraní je grafické rozvržení aplikace. Skládá se ze všeho, co uživatel vidí a s čím interaguje. To vše musí být co nejfunkčněji navrženo tak, aby byl uživatel spokojený. Příkladem jsou tlačítka, texty, obrázky, posuvníky, pole pro zadávání textů a ostatní položky. Uživatelské rozhraní pak zahrnuje rozložení obrazovky, přechody, animace rozhraní a každou jednotlivou mikrointerakci.

3.2.7 Uživatelský prožitek (UX)

Uživatelský prožitek podává informace o interakci uživatele s aplikací. Hodnotí se plynulost zážitku, intuitivnost, ale i těžkopádnost a zmatečnost. Uživatelský prožitek hledá odpovědi na otázky ohledně logiky, jednoduchosti, vstřícnosti a funkčnosti. Uživatelský prožitek je určen tím, jak snadné nebo obtížné je pracovat s prvky uživatelského rozhraní, které návrháři vytvořili.

3.2.8 Prototyp

Prototyp je hrubá verze konečného produktu, která umožňuje vzniku nových nápadů a ukazuje záměr designových rozhodnutí. Prototyp se běžně užívá pro prezentaci investorům. Ti si mohou předem prohlédnout, jak by vypadala výsledná verze, a tím šetří čas i peníze při vývoji. Prototyp může být cokoli, od kresby na papír (Low-fidelity), po propracovaný návrh, který obsahuje možné interakce, jako například možnost proklikat si stránku (High fidelity).

Low-fidelity prototypy jsou většinou vypracovány v papírové podobě a nepodporují uživatelské interakce. Takovýto prototyp je často zhotoven formou ručně načrtnutých návrhů, nebo je vykreslen softwarem. Low-Fidelity prototyp slouží k rychlé vizualizaci všech designových alternativ, a tím vyvolává inspiraci pro nové inovace a zajímavá designová řešení. Další výhodou tohoto přístupu je, že se designéři cítí pohodlněji při navrhování nových změn, protože se jedná pouze o hrubý náčrtek.

High-fidelity prototypy jsou návrhy vytvořené pomocí profesionálního počítačového softwaru. Většinou dovolují realistické uživatelské interakce. Díky tomu prototypy dokážou prezentovat konečný uživatelský zážitek z uživatelského rozhraní. Pro Hi-fi prototyp je charakteristické účinné zachycení přesného prožitku uživatelů při testování (čas

na vykonání úkolu atd.), a zároveň schopnost demonstrovat konečný produkt klientům, managementu a dalším.

3.2.9 Technologie pro vývoj Hi-Fi prototypu

Hi-Fi prototyp lze vyvíjet pomocí několika technologií. Nižší popsané technologie umožňují vytvářet jak uživatelské rozhraní, tak interaktivní prototyp. Každá z technologií má své klady i zápory.

3.2.9.1 Adobe XD

Aplikace Adobe XD, vytvořená společností Adobe, má velmi jednoduché a přívětivé uživatelské rozhraní a je vhodná i pro začátečníky. Aplikace je, na rozdíl od ostatních produktů Adobe, vytvořena speciálně pro UI/UX designéry. Jednou z výhod Adobe Suit je stále aktivní komunita uživatelů a také neustále přibývajících nápomocných návody. Adobe XD obsahuje pro mnoho uživatelů revoluční způsob tvoření mřížek díky funkci „repeat grid“. Stačí pouze kliknout na element, zvolit „repeat grid“ a potom už jen horizontálně či vertikálně potáhnout pousuvník. Další silnou stránkou Adobe XD jsou nástroje pro tvoření prototypu, které nevyžadují znalost programování. Díky tomu mohou designéři přímo vidět, jak bude element reagovat na různé eventy (klik, scrollování). Ovšem Adobe XD nemá pouze klady. V aplikaci například nelze tvořit komplikované objekty, které jsou v UI designu zásadní. Také není v nativní verzi možné exportovat design do CSS. I přes své nedostatky je Adobe XD jedním z nejpoužívanějších softwarů pro vývoj UI/UX designu.

3.2.9.2 Figma

Figma je webová aplikace, která funguje v prohlížeči, tím pádem nezáleží, na jakém operačním systému je spuštěna. To ovšem znamená, že bez připojení k internetu nelze Figma používat. Velkou výhodou Figmy je její cena. Nabízí totiž pro malý dvoučlenný tým designérů licenci a neomezené cloudové úložiště zdarma. Oproti Adobe XD je možné na jednom projektu pracovat současně s jinými uživateli. Další výhodou aplikace je spousta pluginů, které pro Figma vytvořila její komunita. Figma je však také náročná na hardware, takže na starším notebooku bude vytváření designu velmi zdlouhavé a obtížné. Prototypování ve Figmě je velmi podobné Adobe XD, u elementu se vybere akce (např.

kliknutí), na kterou reaguje, a reakce na danou akci. Figma také nabízí exportování do CSS, což může zásadně usnadnit následné stylování aplikace. Figma je pravděpodobně nejpoužívanější aplikací pro vývoj UX designu.

3.2.9.3 Axure RP

Axure RP je dalším velmi oblíbeným nástrojem pro vytváření UI/UX designu. Mezi výhody Axure RP patří možnost vytvářet responzivní design, ve kterém lze uchovávat a dále pracovat s daty, která uživatel zadává v průběhu používání aplikace. Díky tomu lze v Axure RP vytvořit plně funkční prototyp, který simuluje jak frontend, tak i některé funkce backendu a práci s databází. To má však svou cenu, protože pokud chce uživatel využívat tyto pokročilé funkce, neobejde se bez zkušeností s programováním. Axure RP nabízí trial licenci zdarma na 30 dní, nebo placenou individuální licenci, která vyjde na 25 dolarů měsíčně. Axure RP je jedním z nejpokročilejších nástrojů pro vytváření UI/UX designu, a proto je velmi oblíbený mezi mnoha designéry.

3.2.9.4 Grafický návrh + frontend

Tento způsob se skládá ze dvou kroků. Nejdříve je potřeba vytvořit grafický návrh pomocí některého z grafických editorů a poté naprogramovat prototyp pomocí libovolné frontend technologie. Tento postup je časově, a i technicky nejnáročnější, jelikož není možné jednoduše předělat prototyp a některé změny dokonce vyžadují přeprogramování celého projektu. Postup má však i pozitivní stránku, autor totiž získá zkušenosti jak s návrhem UI designu, tak s programováním a děláním návrhu celého projektu. Takovýto způsob je ideální pro studenty či frontendové samouky. Způsob také nabízí velkou svobodu, protože nezávisí na konkrétním softwaru, autor může zvolit nekomplikovanou cestu, která vyžaduje pouze znalost HTML a CSS, ale i komplikovanější přístup, který zahrnuje JavaScript či TypeScript.

Způsob současné práce na designu a developmentu doporučuje také publikace Modular Design Frameworks, ve které autor píše:

„Modernější procesy, jako je Agile, naznačují, že designéři i vývojáři mají během celého procesu od začátku do spuštění konstantní zpětnou vazbu. Designéři se zaměřují na design, jak ho vývojář vyvíjí, a pracují bok po boku.“³(Vlastní překlad autora práce.)



Obrázek 1 - Ilustrace vývoje Hi-Fi prototypu. Zdroj: Modular Design Frameworks: A Projects-based Guide for UI/UX Designers

³ „More modern processes, like Agile, suggest that both designers and developers have a constant feedback loop throughout the entire process, from inception to launch. Designers focus on the design as the developer is developing it, and they work side by side.“

CABRERA, James: Modular Design Frameworks: A Projects-based Guide for UI/UX Designers, Apress: 2017, p. 5.

4 Vlastní práce

4.1 Specifikace požadavků

Výsledný Hi-fi prototyp musí splňovat následující soubor požadavků. Musí mít jak prostředí pro uživatele, tak pro administrativní správu uživatelů.

V uživatelském prostředí musí být možné:

- Přihlášení
- Přidání okruhu problémů
- Zobrazení problémového okruhu
- Možnost filtrovat problémové okruhy
- Zobrazení vypracovaných návodů
- Přidání návodu k danému okruhu

V administrátorském prostředí musí být možné:

- Přihlášení
- Zobrazení všech uživatelů
- Přidání a smazání uživatele
- Zobrazení pracovní pozice, která se přiděluje uživatelům
- Přidání pracovní pozice
- Možnost filtrovat pracovní pozice

Výsledný Hi-Fi prototyp bude realizován pomocí logických návrhů a vytvořením frontend verze, která bude pracovat s testovacími daty ve formátu JSON. Tato data simulují výstup z webové API, na které by se po otestování a odsouhlasení zákazníkem začalo pracovat. Jelikož by se pro práci s API musela funkčnost zcela předělat, neobsahuje Hi-Fi prototyp manipulaci s testovými daty.

Jako frontend technologie bude zvolena Vue.js, jelikož nabízí dobrý výkon a osobně mám s vývojem Vue aplikací předchozí zkušenosti. Pro Vue aplikaci se zvolí rozšíření NuxtJS, které upravuje strukturu nativního Vue.js projektu. Díky NuxtJS také není nutné přidávat cesty k jednotlivým stránkám. NuxtJS dále nabízí možnost server side renderingu, který zajistí, že se stránka předrenderuje na straně serveru a klient stránku už

pouze načte. To zařídí, že „roboti“ mohou projít obsah stránky, což u renderování na straně klienta není možné.

4.1.1 Použité technologie

4.1.1.1 Figma

Pro grafický návrh uživatelského rozhraní byla zvolena webová aplikace Figma, protože nabízí licenci zdarma pro menší týmy a zároveň poskytuje jednoduché a účinné ovládání. Figma také nabízí možnost pro extrahování CSS stylu elementu, což značně urychlí proces finálního stylování.

4.1.1.2 NodeJS

NodeJS je open source serverové prostředí, které lze spustit na jakémkoli operačním systému. Pro serverové operace používá jazyk Javascript. Velkou výhodou NodeJS je, že funguje asynchronně, tudíž se nemusí čekat na odpověď serveru. Instalace NodeJS zahrnuje také NPM (Node Package Manager), což je největší knihovna softwarů na světě a obsahuje více než 800 tisíc balíčků kódů (packages). NPM také obsahuje příkazovou řádku, do které lze zapsat příkazy, například instalace kódových balíčků či spuštění serveru.

4.1.1.3 NuxtJS

NuxtJS vylepšuje a optimalizuje nativní Vue.js aplikace. Nabízí například renderování na straně serveru, automaticky generované routy a lepší předpoklady pro SEO. Nuxt také defaultně obsahuje Vuex Store, který slouží k ukládání dat dostupných z celé aplikace. S vývojem NuxtJS aplikací mám předešlé zkušenosti, proto byl pro mne první volbou.

4.1.1.4 SCSS

Preprocesor SCSS je nadstavbou jazyka CSS. Uživateli umožní dělat složité operace, které následně přepíše do jednoduchého CSS. Hlavní výhodou SCSS je tvoření proměnných, do kterých uživatel může uložit hodnotu, nebo možnost napsat si vlastní

funkci, která pracuje s payloadem, který může specifikovat, jak se daná funkce bude chovat. Díky těmto kvalitám byl preprocessor SCSS vybrán pro tento projekt.

4.1.2 Vývoj Hi-Fi prototypu

Po specifikování a vypsání požadavků, které musí prototyp splňovat, a zvolení technologií, pomocí kterých jich dosáhneme, můžeme pokračovat k návrhu konkrétní aplikace. Začneme logickým návrhem, následně provedeme návrh datové struktury, a poté navrhne prvotní podobu uživatelského rozhraní. Nakonec postoupíme k naprogramování prototypu podle provedených návrhů.

4.1.2.1 Logický design uživatelského rozhraní

Počáteční obrazovkou aplikace bude přihlašovací overlay, který obsahuje logo aplikace a dvě textová pole. První pro uživatelské jméno a druhé pro heslo. Přihlašovací overlay dále obsahuje dvě tlačítka, která umožňují přihlásit se buď jako uživatel, nebo jako administrátor.

Při stisknutí tlačítka „Přihlásit se jako uživatel“ overlay zmizí a zobrazí se domovská stránka. Ta v horní části obsahuje lištu s názvem firmy, pod kterou uživatel pracuje a jméno a příjmení zaměstnance. Horní lišta také zahrnuje tlačítko pro odhlášení.

Spodní část domovské stránky je rozdělená na levou a pravou. Levá část slouží jako list všech problémových okruhů. V listu je možné vyhledávat pomocí pole pro vyhledávání, které je umístěno na vrchu listu. Jako první položka seznamu je problémový okruh, který místo názvu a popisu obsahuje textová pole a slouží pro přidání nového okruhu.

Pravá část domovské stránky obsahuje mřížku o čtyřech sloupcích, do které jsou dosazovány problémy, které spadají pod zvolený problémový okruh. Každý problém je popsán názvem a stručným popisem. Problémy v mřížce také umožňují daný problém smazat či modifikovat pomocí tlačítek, která jsou umístěna v pravém horním rohu.

Při kliknutí na konkrétní problém se přes mřížku zobrazí overlay, který obsahuje název a popis problému a postup rozdělený v několika krocích. Overlay lze zavřít tlačítkem, které je umístěno v jeho pravém horním rohu.

Při stisknutí tlačítka „Přihlásit se jako administrátor“ se zobrazí stránka, která je rozvržena podobně jako domovská stránka. Horní lišta, stejně jako domovská stránka, zobrazuje název firmy, pod kterou se administrátor přihlásil a možnost odhlásit se.

Dolní strana je také rozvržena na levou a pravou stranu. Levá slouží jako seznam pracovních pozic, které si firma pro své účely vytvořila. V seznamu lze opět filtrovat obdobným způsobem. V pravé části se nachází tabulka všech uživatelů, kteří jsou pod firmou registrovaní. Tabulka obsahuje sloupec pro ID uživatele, jméno, příjmení, roli, počet okruhů, které si uživatel vytvořil, a počet problémů, které uživatel vypracoval.

V pravé části se pod tabulkou nachází tlačítka pro manipulaci uživatelů, které administrátor označí. Administrátor má možnost přidat nového uživatele, modifikovat údaje stávajícího uživatele, anebo označené uživatele odstranit. Pro přidávání nového uživatele stránka zobrazí pod tlačítkem „Přidat“ formulář, kam administrátor zadá všechny potřebné údaje.

4.1.2.2 Logický návrh ukládaných dat

Data jsou uložena do tří souborů, které mají svou distinktivní funkcionalitu. Soubory jsou ve formátu JSON.

Company.json

Soubor vytvořený primárně pro administrátorskou správu. Jsou zde uloženy údaje o firmě, jako například jméno firmy a speciální heslo, které bude vyžadováno zejména při mazání administrátorského účtu. Dále soubor obsahuje seznam administrátorů a jejich údaje. Nakonec jsou zde uloženy pracovní pozice v poli Roles, které si firma může sama vytvářet.

```
{
  "companyName": "",
  "masterPassword": "",
  "Admins":[
    {
      "id": ,
      "firstname": "",
      "lastname": "",
      "username": "",
      "password": ""
    }
  ],
  "Roles":[
    {
      "name":""
    }
  ]
}
```

User.json

User.json slouží jako adresář všech uživatelů. V relační databázi byla tato data vázána k zaměstnavateli, ale pro účely HI-FI prototypu stačí tento soubor vytvořit staticky. Jsou zde zapsané údaje o uživateli, ID, uživatelské jméno, heslo, jméno a příjmení zaměstnance a pracovní pozice.

```
{
  "id": ,
  "username": "",
  "password": "",
  "firstname": "",
  "lastname": "",
  "role":""
}
```

Topics.json

V tomto souboru jsou vypsané všechny problémové okruhy a problémy, které do daného okruhu spadají. V případě relační databáze by byl tento soubor vázán na konkrétního uživatele, ale pro demonstrační účely zobrazíme tato data všem uživatelům. Problémové okruhy jsou označeny identifikačním číslem, jménem a krátkým popisem. Návody jsou uloženy v poli Tutorials. Jednotlivé kroky jsou obsaženy v poli Texts, kde jsou také uloženy cesty k obrázkům, které se váží k danému kroku.

```
{
  "id": ,
  "name": "",
  "description": "",
  "Tutorials": [
    {
      "name": "",
      "description": "",
      "Texts":[
        {
          "image": "",
          "text":""
        },
      ]
    }
  ]
}
```

4.1.2.3 Pohled uživatele

Přihlášení

Use Case

Uživatel očekává možnost vložení svých přihlašovacích údajů, za účelem přihlášení.

Jmenovitě jde o:

- Přihlašovací jméno
- Heslo

Uživatel očekává možnost přihlásit se pomocí těchto údajů.

Scénář

System uživateli zobrazí přihlašovací obrazovku, která obsahuje dvě textová pole pro zadání přihlašovacího jména a hesla. Přihlašovací obrazovka také obsahuje tlačítko „Přihlásit“. Po jeho stisknutí systém zkontroluje, jestli uživatel zadal správné jméno a heslo, a pokud ano, systém uživatele přihlásí.

Seznam problémových okruhů

Use Case

Uživatel očekává, že na domovské stránce uvidí jednotlivé popsané problémové okruhy zařazené do seznamu. Očekává možnost filtrování podle názvu okruhu.

Scénář

System uživateli zobrazí domovskou stránku, na které je v pravé dolní části zobrazen vertikální seznam všech problémových témat, která se vztahují k pozici uživatele. Položky v seznamu jsou popsány názvem problémového okruhu a krátkým popisem. Nad seznamem systém zobrazí pole pro vyhledávání. Při zadání každého písmena se seznam znovu vyfiltruje.

Přidání problémového okruhu

Use Case

Uživatel požaduje možnost přidat nový problémový okruh a definovat jeho vlastnosti konkrétně:

- Název problémového okruhu
- Popis problémového okruhu

Scénář

System na domovské stránce, vždy v levé části obrazovky, vykreslí list problémových okruhů. První položka listu nabídne dvě textová pole pod sebou. Horní textové pole je popsáno textem „Název nového okruhu“ a slouží na nadepsání okruhu. Spodní textové pole je popsáno „Krátký popis okruhu“, do kterého uživatel запиše popis problémového okruhu. Napravo od textových polí se nachází tlačítko „+“, při jeho stisknutí se do seznamu přidá nová položka s definovaným názvem a popisem.

Zobrazení vypracovaných návodů k problémovému okruhu

Use Case

Uživatel očekává seznam vypracovaných návodů pro zvolený problémový okruh.

Scénář

System při kliknutí na libovolný okruh vykreslí seznam zapsaných problémů do čtverců umístěných v mřížce o čtyřech sloupcích. Problémy jsou popsány uvnitř čtverců, obsahují název problému a pod ním bližší popis problému.

Přidání návodu

Use Case

Uživatel očekává, že bude moci přidat nový návod k zvolenému problémovému okruhu a definovat jeho vlastnosti:

- Název problému
- Popis problému
- Pro koho bude návod viditelný
- Postup návodu

Scénář

System vykreslí domovskou stránku se seznamem problémových okruhů. Uživatel klikne na libovolný problémový okruh a system mu zobrazí seznam návodů pro daný okruh. Na prvním místě seznamu návodu je položka s nápisem „Přidat návod“. Po kliknutí na toto tlačítko system v pravé části obrazovky zobrazí formulář, který má v levé horní části

textová pole pro zapsání názvu problému a pro krátký popis. V pravé horní části je seznam rolí, pro které chce uživatel řešení zveřejnit.

Spodní část je rozdělena na kroky, ve kterých uživatel popíše postup. Pod kroky jsou v levé části vedle sebe umístěna tlačítka pro přidávání a odebrání kroků. V pravé dolní části formuláře je umístěno tlačítko „Přidat“, kterým uživatel přidá zadaný postup.

Odhlášení uživatele

Use Case

Uživatel očekává možnost odhlášení a návratu na přihlašovací obrazovku.

Scénář

System v horní části obrazovky zobrazí lištu, která v pravé části vypíše nápis „Odhlásit se“. Při kliknutí na toto tlačítko, systém uživatele odhlásí.

4.1.2.4 Pohled administrátora

Přihlášení

Use Case

Uživatel očekává možnost přihlásit se do účtu administrátora, pomocí svých přihlašovacích údajů:

- Přihlašovací jméno
- Heslo

Scénář

System zobrazí úvodní obrazovku, která obsahuje dvě textová pole pro zadání přihlašovacího jména a hesla. Přihlašovací obrazovka také obsahuje tlačítko „Přihlásit se jako administrátor“. Pokud uživatel stiskne tlačítko „Přihlásit se jako administrátor“, systém zkontroluje, zda uživatel zadal správné přihlašovací údaje a jestli má administrátorský přístup. Pokud vše odpovídá, systém uživatele přihlásí do administrátorského panelu.

Seznam uživatelů

Use Case

Uživatel očekává, že uvidí v administrátorském panelu seznam všech uživatelů, kteří v jeho společnosti aplikaci používají. V seznamu uživatel očekává tyto informace:

- ID
- Jméno
- Příjmení
- Pracovní pozice
- Počet problémových okruhů, které uživatel vytvořil
- Počet návodů, které uživatel vytvořil

Uživatel očekává možnost přidat uživatele do seznamu a definovat jeho vlastnosti:

- Jméno
- Příjmení
- Přihlašovací jméno
- Heslo
- Pracovní pozice

Uživatel očekává možnost ze seznamu odebrat uživatele.

Scénář

System zobrazí v pravé části administrátorského panelu tabulku, do které načte uživatele, kteří mají v aplikaci pod danou společností účet. Tabulka o uživateli zobrazuje informace v následujícím pořadí: ID, Jméno, Příjmení, Počet okruhů, Počet návodů. System také u každého uživatele zobrazí napravo v tabulce zaškrtačací políčko. System v administrátorském panelu pod tabulkou vykreslí tlačítka „Přidat“ a „Odebrat“. Po stisku tlačítka „Přidat“, system pod tabulkou zobrazí formulář pro přidání uživatele. Formulář obsahuje textová pole pro zadání přihlašovacího jména, hesla, jména a příjmení. Pod textovými poli vybere z rozbalovacího seznamu, jakou pracovní pozici bude mít uživatel přidělen. Vespod formuláře poté uživatel potvrdí přidání uživatele tlačítkem „Vytvořit“. Pro odebrání záznamu uživatel zaškrtně políčko u požadovaného uživatele poté stiskne tlačítko „odebrat“, a tím uživatele odstraní.

Zobrazení všech pracovních pozic

Use Case

Uživatel očekává v administrátorském panelu seznam pracovních pozic. Uživatel také očekává možnost v seznamu filtrovat pomocí klíčových slov.

Scénář

System zobrazí administrátorský panel, který v levé dolní části vykreslí vertikální seznam všech pracovních pozic, které byly pro danou společnost vytvořeny. Nad seznamem systém vykreslí vyhledávací pole, do kterého uživatel zadá klíčová slova. Při zadání nového písmene se seznam znovu překreslí.

Přidání pracovní pozice

Use Case

Zaměstnavatel očekává možnost přidat novou pracovní pozici a specifikovat její název.

Scénář

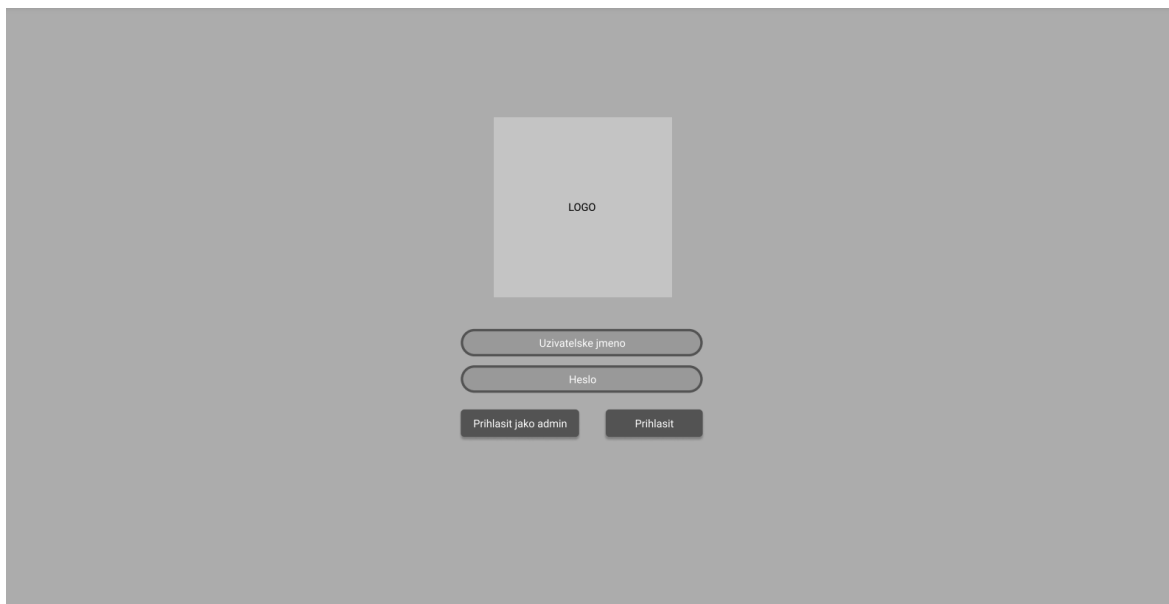
System přidá na vrch seznamu pracovních pozic položku pro přidání pracovní pozice. Tato položka se skládá z textového pole, ve kterém je vepsáno „Název nové pozice“, do tohoto textového pole uživatel specifikuje název pozice, kterou chce vytvořit. Napravo od textového pole se nachází tlačítko „+“, které přidá novou pracovní pozici s názvem, který je v textovém poli.

4.1.2.5 Prvotní návrh uživatelského rozhraní

V této kapitole je proveden návrh uživatelského rozhraní v programu Figma. Hi-fi prototyp byl designován tak, aby vyhověl požadavkům uživatele i zaměstnavatele. Jednotlivé designy jsou poté blíže popsány a je definován jejich účel.

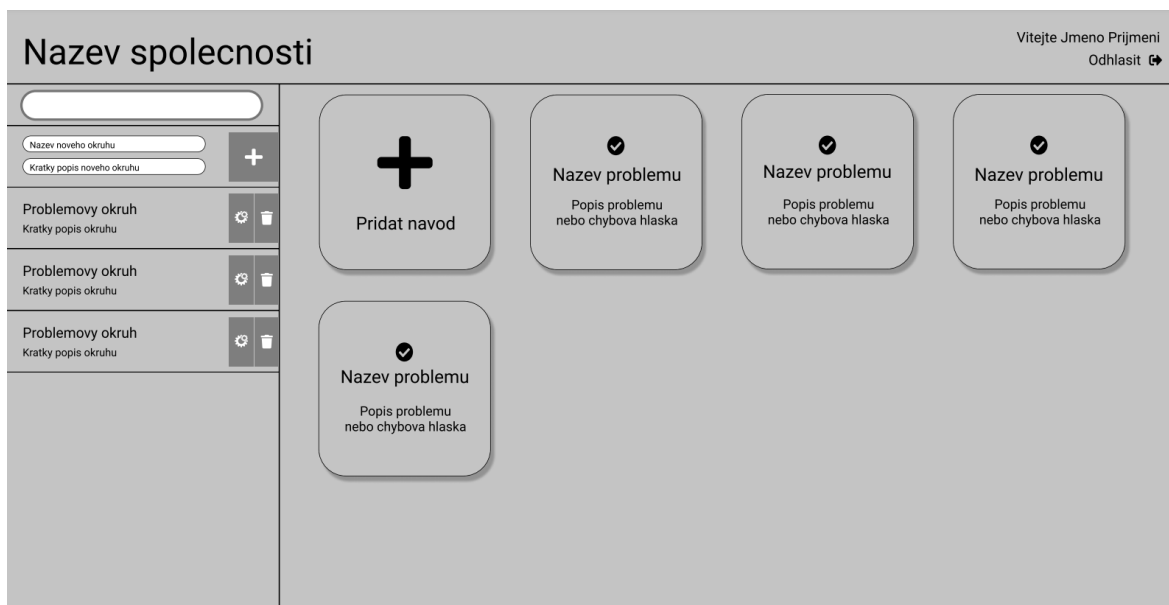
Přihlašovací obrazovka je designově velmi simplicistní, zobrazuje logo aplikace a dvě textová pole, která obsahují placeholder, který označuje jejich funkci. První textové pole slouží pro zadávání uživatelského jména, v druhém uživatel zadává své heslo. Pod

textovými poli jsou dvě horizontálně seřazená tlačítka, která označují způsob, kterým se chce uživatel přihlásit. Buď jako administrátor, nebo jako uživatel.



Obrázek 2 – návrh přihlašovací obrazovky. Zdroj: vlastní zpracování

Domovská stránka je navržena tak, aby byla jak funkční, tak i moderní. Obrazovka je rozdělená na tři části, v horní části obsahuje název společnosti, pro kterou uživatel pracuje, a také je zde možnost se odhlásit. Spodní, takzvaná „hlavní“ část, je složena ze dvou seznamů, levý seznam ukazuje soupis všech problémových okruhů a umožňuje je pozměnit či smazat. V levé části se také nachází pole pro filtrování seznamu a pod ním je formulář, který slouží k přidání nového okruhu. V pravé části je seznam návodů, které byly vypracovány ke zvolenému okruhu, seznam má na prvním místě možnost přidání nového návodu. Návody jsou umístěny do čtyř sloupců.



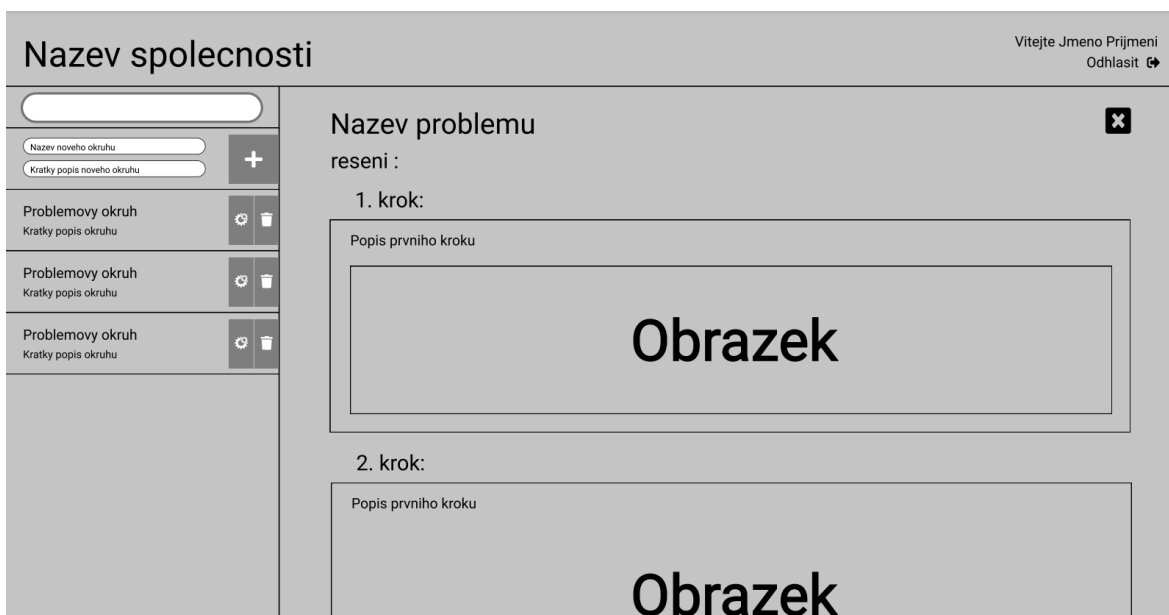
Obrázek 3 - návrh domovské stránky. Zdroj: vlastní zpracování

Stránka pro přidání nového návodu, se zobrazuje na domovské stránce přes pravou dolní část. Tento overlay obsahuje nejprve textová pole pro název návodu a pro krátký popis. Vpravo zobrazuje seznam pod sebou vypsáných pracovních pozic, ze kterých uživatel vybere, pro koho je tento návod určen. Pod těmito poli jsou dále umístěna textová pole, do kterých uživatel v jednotlivých krocích sepíše postup návodu.



Obrázek 4 – návrh formuláře pro vytvoření návodu. Zdroj: vlastní zpracování

Stránka pracovního návodu obsahuje pod sebou vypsání kroky, které uživatel vytvořil. Zobrazí se přes pravou část domovské stránky. Každý pracovní návod je popsán v horní části obrazovky nadpisem, pod nímž následují pole pro vypsání kroků. Tento návrh je proto velmi jednoduchý a efektivní. Stránku pracovního návodu lze zavřít tlačítkem, které je umístěné v pravé horní části.



Obrázek 5 – návrh pracovního návodu. Zdroj: vlastní zpracování

Administrátorská stránka slouží jako přehled pro zaměstnavatele. Pro zachování integrity aplikace, je stránka rozvržena stejně jako domovská stránka. Na levé straně se nachází seznam všech pracovních pozic, které byly pro společnost vytvořeny. Na pravé straně bude umístěn seznam uživatelů. Pod tabulkou uživatelů se nachází tlačítka, pomocí kterých lze v tabulce provádět změny. Po stisknutí tlačítka „Přidat“ se pod tlačítkem zobrazí formulář pro přidání uživatele.



Obrázek 6 – návrh administrátorského panelu. Zdroj: vlastní zpracování

4.2 Naprogramování Hi-Fi prototypu

Za využití předchozích kroků bude možné přejít k realizaci Hi-Fi prototypu. Počáteční instalace nového Nuxt projektu bude provedena v konzoli NPM příkazem „npm init nuxt-app <project-name>“. Následně bude vytvořena struktura aplikace, díky které bude prototyp demonstrovat všechny Use Casy a požadavky logického návrhu. Poté budou vytvořeny datové soubory, do kterých budou vepsána testovací data. Na konec budou jednotlivé elementy nastýlovány tak, aby odpovídaly grafickým návrhům.

4.2.1 Vytvoření struktury projektu

LoginComponent.vue

Tato komponenta funguje jako přihlašovací overlay, který se pomocí Vue.js funkce „v-if“ renderuje pouze, pokud není nikdo přihlášený. Obsahuje logo aplikace a dvě textová pole pro přihlašovací jméno a heslo. Pod textovými poli jsou dvě tlačítka, která odlišují, jestli se uživatel hlásí jako administrátor, nebo jako uživatel. Při stisku jednoho z tlačítek je provedena jednoduchá funkce na ověření uživatele a při úspěšném přihlášení je ID uživatele posláno do Vuex Storu, ze kterého si data přebírá Homepage.vue a podle něj upravuje svůj obsah.

Adminpage.vue

Stránka, do které se lze přihlásit pouze administrátorskými přihlašovacími údaji, které se zadávají do LoginComponent.vue. Administrátor má díky této stránce přístup k tabulce uživatelů a seznamu pracovních pozic. Pod tabulkou uživatelů jsou i tlačítka pro modifikování a přidávání uživatelů. Administrátor zde může také přidávat a odebírat pracovní pozice.

Homepage.vue

Homepage.vue slouží jako domovská stránka. Obsahuje HeaderComponent.vue, TutorialLayout.vue a seznam témat.

Od LoginComponent.vue dostane Homepage.vue informaci o konkrétním uživateli, který se právě přihlásil. Tomu poté ukáže pro něj relevantní témata a návody.

Homepage.vue pracuje s daty uloženými company.json, topics.json a users.json.

HeaderComponent.vue, umístěný v levém horním rohu, obsahuje název konkrétní společnosti, která webovou aplikaci pro své zaměstnance zřídila, v pravém rohu, naopak podává informaci o konkrétním uživateli, jeho jméno a příjmení.

Seznam, který je umístěný v levém sloupci, obsahuje názvy problémových okruhů, ze kterých uživatel vybírá. Po rozkliknutí nadřazeného tématu se v pravé části obrazovky objeví TutorialLayout.vue, ve kterém jsou rozmístěny konkrétní problémy vztahující se k uživatelem zvolenému tématu.

HeaderComponent.vue

Rodičem HeaderComponent.vue je buď Homepage.vue, nebo Adminpage.vue, od kterých si tato komponenta bere data, která promítne do svých proměnných. Adminpage.vue obsahuje název firmy, přivítání uživatele s jeho jménem a heslem a také nabízí funkci pro odhlášení. Funkce pro odhlášení emituje událost, kterou poslouchá Homepage.vue. ta při jejím spuštění proměnnou s přihlášeným uživatelem resetuje z Vuex Storu, a tím ho odhlásí.

TutorialLayout.vue

Tato komponenta je umístěna na pravé straně domovské stránky. Zobrazuje položky do specifikovaného čtyř sloupcového gridu. Jako první položka je pevně přidělená položka „Přidat návod“, která přes celou tuto komponentu zobrazí TutorialCreateComponent.vue. Při kliknutí na jinou položku s konkrétním postupem se

zobrazí komponenta `TutorialComponent.vue`, do které se pošlou data o tom, jaký postup chce uživatel vidět.

TutorialComponent.vue

Slouží k zobrazení návodu, přijímá data od `TutorialLayout.vue`. Přijatá data jsou ve formě tutoriálu, který uživatel zadal. Tato data poté komponenta rozmístí do jednotlivých kroků návodu. Obrázky jsou v jednotlivých krocích zatím pouze pevně přiděleny a slouží pouze k demonstraci. Komponentu lze zavřít kliknutím na červené tlačítko s křížkem v pravém horním rohu.

TutorialCreateComponent.vue

Komponenta, která slouží jako formulář, do kterého uživatel zadá název problému, krátký popis a určí, zda tento postup uvidí i ostatní lidé skrze udělené pracovní pozice, anebo pouze on. Poté do jednotlivých kroků vypíše svůj postup. Kroky může uživatel přidávat a ubírat pomocí tlačítek umístěných pod posledním krokem. Formulář se odešle tlačítkem „Vytvořit“.

4.2.2 Dodatečné stylování uživatelského rozhraní

Main.scss

Soubor obsahující styly, které jsou používány skrze celý projekt. Aby byl tento soubor vidět globálně, je zapotřebí přidat cestu k souboru do konfiguračního souboru `nuxt.config.json` do sekce CSS (viz obrázek 8).

```
/** Global CSS
 */
css: [
  '~/assets/style/main.scss'
],

```

Obrázek 7 – globální styly v NuxtJS. Zdroj: vlastní zpracování

Main.scss bylo použito na určení stylů pro html elementy, například element body, ve kterém je určen rozměr stránky, jak se mají elementy uvnitř zarovnávat, nebo například defaultní barva textu. Main.scss dále obsahuje SCSS mixiny, které byly vytvořeny pro urychlení procesu stylování.

```
@mixin flex($direction,$align){
  display: flex;
  flex-direction: $direction;
  align-items: $align;
}
@mixin wh($x:100% ,$y: 100%){
  width: $x;
  height: $y;
}
```

Obrázek 8 - příklad vytvořených SCSS mixinů. Zdroj: vlastní zpracování

Variables.scss

Tento soubor obsahuje všechny proměnné, které jsou používány v celém projektu, primárně se jedná o hodnoty RGB pro barvy, ale také v něm byly určeny některé velikosti odsazení.

Aby bylo možné tento soubor sdílet v celém projektu, bylo zapotřebí ho přidat do konfiguračního souboru `nuxt.config.js` následujícím způsobem:

```
styleResources: {  
  scss: [  
    '~/assets/style/variables.scss'  
  ]  
},  
/*
```

Obrázek 9 – globální SCSS proměnné v NuxtJS. Zdroj: vlastní zpracování

Styl komponent

Vue instance se skládá z „template“ části pro zápis HTML, „scriptu“ pro javascript a z části „style“, kam se zapisují styly pro danou vue instanci. K sekci style je možné připsat, jaký jazyk bude používán. Vepsáním `lang="scss"` se definuje jazyk SCSS.

4.3 Cílová skupina

Cílovou skupinu tvoří všichni dospělí jedinci, kteří se ve svém zaměstnání setkávají s mnoha rozdílnými softwary a rozličnými postupy, které často zapomínají. Aplikace má za cíl šetřit práci nadřízeným a IT support oddělení, jelikož umožňuje zaměstnancům vytvořit si postupy, na které by se jinak museli ptát.

4.4 Persony

4.4.1 Persona A

Jméno	Kateřina Petrážová
Věk	35
Pohlaví	Žena
Zájmy	Rodinné výlety, čtení, vaření

Typický den	Kateřina se probouzí kolem šesté hodiny ráno, aby si stihla připravit svou oblíbenou snídani a kávu. Po snídani vyjede do práce, do které musí jezdit autem, protože bydlí na vesnici a její zaměstnání je v hodinu vzdáleném městě. Kateřina pracuje jako státní úřednice, její náplní práce je připravovat veřejné vyhlášky a interní smlouvy. Kateřina používá ke své práci mnoho interních softwarů, které jí občas dělají vrásky, protože každý program má spoustu svých specifických postupů. Kateřina je často zapomene a musí se pak ptát svých kolegů nebo nadřízených. Po práci přijede domů za svým manželem a dětmi se kterými jde na dlouhou procházku přírodou. Večer pošle děti spát a kouká s manželem na jejich oblíbený pořad.
Historie	Vystudovala magisterský obor na ČVUT. Poté nastoupila na pozici referentky veřejné správy na Úřad vlády.

4.4.2 Persona B

Jméno	Eduard Králík
Věk	26
Pohlaví	Muž
Zájmy	Informační technologie, muzika, sledování televize
Typický den	Eduard ráno brzy vstává, protože jeho práce vyžaduje přítomnost už od časných ranních hodin. Eduard pracuje jako IT support a už od začátku směny za ním lidé chodí s problémy, které jsou často stejné nebo podobné. Eduardovi přijde, že řeší stejné problémy téměř každý den a přemýšlí, jestli neexistuje nějaký způsob, jak problémy jednoduše vypracovat pouze jednou, přičemž lidé by si potom daný problém sami našli a ušetřil by se tak čas všech zúčastněných. V práci často zůstává přesčas, protože jeho firma má nedostatek zaměstnanců na pozici IT support, a zaměstnanci jsou proto přepracováni. Po práci jede domů za přítelkyní, se kterou hraje stolní hry nebo společně hrají počítačové hry.
Historie	Vystudoval střední technickou školu, po složení maturitní zkoušky nastoupil na pozici IT support do menší firmy.

4.4.3 Persona C

Jméno	Milada Horáková
Věk	40
Pohlaví	Žena
Zájmy	Nakupování, cestování, vaření
Typický den	Managerka malého týmu v účetnické firmě, kam nově přijali posilu na pozici junior marketéra. Náplní její práce je řízení práce teamu, na což ale Milada momentálně nemá čas, protože se musí starat o zaučování nového zaměstnance.
Historie	Vystudovala střední školu ekonomickou, po škole začala pracovat v soukromé firmě, kde se postupně vypracovala až na pozici managerky.

4.4.4 Antipersona

Jméno	Amanda Primová
Věk	70
Pohlaví	Žena
Zájmy	Zahradničení, dovolené u more, wellness
Typický den	Amanda vstává velmi brzy ráno kvůli insomnii, snídá pouze müsli. Jelikož je Amanda už v důchodu, dopoledne vaří a odpoledne se věnuje svému muži a své zahrádce. O prázdninách svým vnoučatům. Večery tráví popíjením svého oblíbeného nealkoholického piva a sleduje svou oblíbenou reality show. Chodí spát většinou kolem 10. hodiny.
Historie	Amanda vystudovala střední odbornou školu. Celý život pracovala jako pošťačka a v důchodu už si nijak nepřivydělává. Vystačí si se svým důchodem, jelikož bydlení má již několik let zařízené.

4.5 Testování

Při testování bude zvolena forma usability engineeringu v kontrolovaném simulovaném prostředí, které bude modelovat kancelářské prostředí. Zároveň bude vytvořen testovací formulář, který obsahuje tabulku, kam testovaný subjekt o sobě vyplní základní informace. Formulář představí uživateli scénář, který mu pomůže vcítit se do simulované situace. Nakonec testované subjekty obdrží tabulku s předdefinovanými, realistickými úkoly. Tabulka také obsahuje prostor na postřehy či připomínky k jednotlivým úkolům. Participanti zůstanou v anonymitě, abychom zachytili autentické používání aplikace. Pozorování proto bude pouze formou záznamu obrazovky. Na tomto záznamu bude poté provedena analýza uživatelského chování v prostředí aplikace a změřen čas všech úkolů. Nakonec bude provedeno osobní interview s jednotlivými participanty, při kterém bude zachycen jejich celkový pocit z používání aplikace a probrány jednotlivé úkoly.

Cíl testování:

- Zjistit designové nedostatky v aplikaci
- Definovat případná budoucí vylepšení aplikace
- Identifikovat úkoly, které participantům dělají problémy
- Změřit, jak dlouho trvalo participantům plnit jednotlivé úkoly

Na konci testování bude sestavena tabulka, ve které budou sepsány vlastnosti participantů a také to, jak si v testování vedli. Z této tabulky a z osobních interview budou poté vyvozeny závěry, které budou sloužit jako podklad pro návrhy na vylepšení aplikace.

4.5.1 Testovací formulář

Pohlaví	
Věk	

Jak byste popsal(a) svou znalost práce s počítačem:

- Začátečník
- Pokročilý
- Expert

Simulovaný scénář:

Jste nový zaměstnanec velké mezinárodní firmy. IT administrátor vám řekl, že v této firmě používají aplikaci, do které si mohou zaměstnanci ukládat své pracovní postupy a že pro vaši pozici je v ní nasdíleno spoustu takových postupů. Dal vám za úkol tuto aplikaci projít a seznámit se s jejími funkcemi.

Při řešení úkolů, kde musíte vyplnit textová pole vyplňte „test“.

Přihlašovací údaje pro uživatele

Uživatelské jméno : test

Uživatelské heslo : test

Přihlašovací údaje pro administrátora

Uživatelské jméno : test

Uživatelské heslo : test

Úkol	Byl uživatel schopen úkol splnit?	Poznámky k úkolu
Přihlaste se pomocí zadaných přihlašovacích údajů.		
Pokuste se přidat jeden problémový okruh.		
Pokuste se přidat jeden problémový postup k libovolnému okruhu.		
Zobrazte si postup pro zaseklý papír v tiskárně.		
Zobrazte si postup pro vycentrování textu v Microsoft Word.		
Odhlase se z aplikace.		
Přihlaste se jako administrátor.		
Pokuste se přidat uživatele do tabulky uživatelů.		
Odhlase se z aplikace.		

5 Výsledky a diskuse

Testováno bylo 8 subjektů, které spadají do námi stanovené cílové skupiny. Zároveň byl kladen důraz na rozmanitost pohlaví a věku, aby byly pokryty všechny demografické skupiny.

Výsledky testování jsou zapsány v tabulce, kde jsou popsány vlastnosti jednotlivých subjektů, a také informace, jak si respondenti vedli v testování. Celkový dojem a připomínky, které participanti napsali do formulářů, byly poté probrány při osobním interview.

5.1 Výsledky

Věk	Pohlaví	Znalost práce s počítačem	Celkový čas	Splněných úkolů
22	Žena	pokročilý	2 minuty 56 sekund	9/9
21	Muž	expert	2 minuty 35 sekund	9/9
28	Žena	pokročilý	4 minuty 5 sekund	9/9
60	Žena	začátečník	6 minuty 46 sekund	6/9
61	Muž	začátečník	6 minuty 52 sekund	6/9
46	Žena	pokročilý	4 minuty 36 sekund	9/9
39	Muž	pokročilý	3 minuty 25 sekund	9/9
28	Zena	pokročilý	2 minuty 52 sekund	9/9

Testování ukázalo, že většina testovaných subjektů byla schopna úkoly bez problému splnit. Jediný případ, ve kterém se nepodařilo splnit všechny úkoly, je u věkové skupiny 60+, která se zasekla u přihlášení se jako administrátor. To se dá ale vzhledem k technickým znalostem subjektů omluvit, protože při osobním interview uvedli, že nevěděli, co znamená administrátorský přístup.

Na základě osobních interview poté subjekty uvedly, že se jim vzhled aplikace líbil. Valná většina subjektů měla dotaz na mobilní verzi a uvedly, že by ocenily možnost mít tuto aplikaci stále po ruce. Mladším subjektům přišla některá tlačítka příliš velká, ale

naopak právě to se líbilo věkově starším subjektům. Jelikož je tato aplikace cílena na všechny věkové skupiny, neplánuje se design tlačítek měnit. Některé subjekty také zmínily, že si nevšimly možnosti resetovat heslo, údajně se jim často stává, že svá hesla zapomínají a takovou možnost by ocenily i zde. Barevnou paletu chválily často mladší subjekty, a naopak starší říkaly, že je pro ně aplikace příliš temná.

5.1.1 Navrhovaná vylepšení aplikace

Navrhovaná vylepšení vycházejí z předchozího uživatelského a autorského testování. To ukázalo, že největší zájem měly subjekty o rozšíření aplikace na mobilní zařízení. Jelikož nejsou vždy v situaci, kdy mají přístup k počítači a informace by měly být dostupné rychle a snadno nehledě na tom, jestli se nachází v kanceláři či kdekoli jinde. Další návrh na rozšíření je spíše kosmetického charakteru, a to přidání světlejší palety barev s větším kontrastem. Testované subjekty také projevily zájem o možnost resetovat si heslo, jelikož se jim poměrně často stává, že si na něj nemohou vzpomenout. Poslední návrh na vylepšení se váže pouze na administrátorskou stránku, do které by se mohlo přidat více funkcionality, například grafy aktivit atd.

6 Závěr

V teoretické části byla prozkoumána statistická data, která byla následně využita pro zvolení cílové skupiny a definování vybraných person a jedné antipersony. Specificky se jedná o data zabývající se počítačovou gramotností v České republice a o data popisující neformálním vzdělávání v České republice. Dále se v teoretické části bakalářská práce zabývá technologiemi, které jsou běžně používány při vývoji webových aplikací. Nejprve rozebírá základní technologie, které jsou nezbytné pro vývoj funkční webové stránky. Dále se zaobírá základy UI designu, a to konkrétně definováním uživatelského rozhraní a prototypu. Tím se dostáváme k poslednímu tématu zařazenému v teoretické části, a to k technologiím využívaným při vývoji Hi-Fi prototypu, o kterém se psalo dříve. Tyto technologie byly stručně popsány a byly zhodnoceny jejich silné a slabé stránky.

Praktická část je zaměřena na návrh a naprogramování zvoleného Hi-Fi prototypu. Nejprve byly sepsány a analyzovány požadavky, které musí Hi-Fi prototyp splňovat. Poté se přešlo k vývoji. Vývoj začal zvolením technologií, které byly použity při následné práci. Výběr technologií vycházel z analýzy, která byla provedena v teoretické části. Vývoj pokračoval logickým návrhem uživatelského rozhraní, jenž se zabýval návrhem logické funkcionality, která bude v aplikaci probíhat. Následně byl vypracován logický návrh ukládaných dat, ve kterém byla určena datová struktura použitých JSON dat. Tato data byla navrhovaná tak, aby simulovala budoucí výstup od backend serveru. Protože se Hi-Fi prototyp dělí na část uživatelskou a administrátorskou, v další části byly sepsány požadované Use Casy a k nim odpovídající scénáře, a to jak z pohledu uživatele, tak z pohledu administrátora. Prvotní návrh uživatelského rozhraní byl proveden jako další a spočíval v konkrétním stylování již přednavržených struktur. Poté byla naprogramována struktura a funkčnost navrženého prototypu. Tato struktura byla rozdělena do vytvořených komponent a stránek, k těm byly vypsány jejich jednotlivé funkce a celková role, kterou plní v projektu. Po naprogramování struktury a jednoduché funkcionality bylo ještě potřeba elementy dodatečně nastylovat.

Hotový Hi-Fi prototyp byl následně otestován subjekty, které spadají do cílové skupiny. Testování proběhlo v simulovaném prostředí, kam byly testované subjekty pozvány. Subjekty poté obdržely testovací formulář, který obsahoval úkoly, které během

testování plnily. Obrazovky subjektů byly nahrávány a poté detailně prozkoumány. Výsledné dojmy a připomínky byly zdokumentovány a použity při stanovení výsledků a případných návrhů na vylepšení.

7 Seznam použitých zdrojů

Literární zdroje:

CABRERA, James, 2017. *Modular Design Frameworks: A Projects-based Guide for UI/UX Designers*, Apress. ISBN: 9781484216873

PAVLÍČEK, Josef, Karolína SOLANSKÁ a KOL., b.r. *Učebnice Interakce člověk počítač* [Elektronický dokument]. Praha.

REYNDERS, Fanie, 2018. *Modern API Design with ASP.NET Core 2: Building Cross-Platform Back-End Systems*, Apress. ISBN: 9781484235195.

Internetové zdroje:

BABICH, Nick: *How To Communicate The Value Of User Research* [online]. Dostupné z: <https://uxplanet.org/how-to-communicate-the-value-of-user-research-df60c0f25a31>

BABICH, Nick: *10 Simple Tips to Improve User Testing* [online], Dostupné z: <https://uxplanet.org/10-simple-tips-to-improve-user-testing-6a86c84e2794>

DAITYARI, Shaumik: *Angular vs React vs Vue: Which Framework to Choose in 2021* [online]. Dostupné z: <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>

ČESKÝ STATISTICKÝ ÚŘAD, *V ČR se vzdělává polovina zaměstnanců* [online]. Dostupné z: <https://www.czso.cz/csu/czso/870031a611>

ČESKÝ STATISTICKÝ ÚŘAD, *Digitální dovednosti* [online]. Dostupné z: <https://www.czso.cz/csu/czso/digitalni-dovednosti-2018>

ESPOSITO, Emily: *Low-fidelity vs. high-fidelity prototyping* [online], Dostupné z: <https://www.invisionapp.com/inside-design/low-fi-vs-hi-fi-prototyping/>

IBRAGIMOVA, Eleonora: *High-fidelity prototyping: What, When, Why and How?* [online]. Dostupné z: <https://blog.prototypr.io/high-fidelity-prototyping-what-when-why-and-how-f5bbde6a7fd4>

KOK, Lau Tiam, 2020. *Hands-on Nuxt.js Web Development*, Packt: 2020, ISBN: 9781789952698

LEŠKA, Šimon: *Co je to UX a UI design?* [online]. Dostupné z: <https://www.blueghost.cz/clanek/co-je-to-ux-a-ui-design/>

MIFSUD, Justin, 2012. *Paper Prototyping As A Usability Testing Technique*. *Usability Geek* [online]. Dostupné z: <https://usabilitygeek.com/paperprototyping-as-a-usability-testing-technique/>

NELSON, Brett, 2017. *Getting to Know Vue.js*, Apress. ISBN: 9781484237816.

NUXTJS, *NuxtJS Documentation* [online]. Dostupné z: <https://vuejs.org/v2/guide/installation.html>

OMOLE, Olayinka: *Nuxt.js: a Minimalist Framework for Creating Universal Vue.js Apps* [online]. Dostupné z: <https://www.sitepoint.com/nuxt-js-universal-vue-js/>

PAVLÍČEK, Josef. *Interakční design*. Prezentace prezentována na: [Česká zemědělská univerzita; Praha]

PAVLÍČEK, Josef. *Interakce člověk počítač*. Prezentace prezentována na: [Česká zemědělská univerzita; Praha]

SKÓLSKI, Paweł: *Single-page application vs. multiple-page application* [online]. Dostupné z: <https://goo.gl/eqwg7t>

VUE.JS, *Vue Documentation* [online]. Dostupné z: <https://vuejs.org/v2/guide/installation.html>

8 Přílohy

Aplikaci a její kód lze stáhnout na tomto odkazu:
<https://github.com/ZdenCo/supper-frontend>