



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**INFORMAČNÍ SYSTÉM PRO SPRÁVU
PARTICIPATIVNÍHO ROZPOČTU**

INFORMATION SYSTEM FOR MANAGEMENT OF PARTICIPATORY BUDGETING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ALEŠ SEDLÁČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ HYNEK, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Sedláček Aleš**
Program: Informační technologie
Název: **Informační systém pro správu participativního rozpočtu**
Information System for Management of Participatory Budgeting
Kategorie: Informační systémy

Zadání:

1. Prostudujte problematiku participativního rozpočtování a jeho aplikaci pro tvorbu rozpočtů obcí. Prozkoumejte existující systémy určené pro tento účel.
2. Prostudujte principy tvorby informačních systému a použitelných uživatelských rozhraní.
3. Analyzujte požadavky na tvorbu informačního systému pro správu participativního rozpočtu vybrané obce. Detekujte jednotlivé typy uživatelů a jejich vlastnosti. Porovnejte požadavky s existujícími systémy určenými pro tento účel.
4. Navrhněte informační systém dle detekovaných požadavků z bodu 3.
5. Navržený informační systém implementujte.
6. Výsledné řešení otestujte ve spolupráci s vybranou obcí.

Literatura:

- Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Morgan Kaufmann Publishers/Elsevier, 2010, ISBN: 9780123750303.
- Preece, J., Sharp, H. a Rogers, Y.: *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2015, ISBN: 9781119020752.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hynek Jiří, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 22. října 2020

Abstrakt

Cílem této práce je vytvořit informační systém pro participativní rozpočet, kde lidé mohou ovlivnit, jakým způsobem je zajištěn rozvoj kvality života občanů. Mohou navrhnout své nápady, jak vylepšit své okolí nebo uskutečnit akci z peněz poskytnutých z rozpočtu obce. Zvolený problém jsem vyřešil způsobem, jenž má za úkol spojit občany se zastupitelstvem a zlepšit tak vzájemnou komunikaci. Občan může v systému zastávat roli navrhovatele nebo hlasujícího. Navrhovatel předloží detailní návrh, který obsahuje název, místo provedení, popis, fotky a další informace. Pokud návrh získá dostatečnou podporu hlasujících a výsledek analýzy proveditelnosti je kladný, lze projekt realizovat. V praxi jsem vytvořil systém, který jsem konzultoval s členy zastupitelstva městského obvodu Ostrava-Svinov. Bylo mi umožněno provést průzkum mínění občanů obce. Webová aplikace avšak nebyla vyvinuta speciálně pro jednu obec, využít ji mohou i jiné subjekty, které by měly zájem.

Abstract

The aim of this work is to create an information system for the participatory budgeting that the citizens could use to influence the development of their surroundings. The system allows them to suggest their own ideas how to improve their surroundings or to carry out an event with the money provided from the municipality's budget. I solved the chosen problem in a way that aims to connect citizens with the council and thus, improve the communication with the public. A citizen can play the role of a proposer or a voter in the system. The proposer submits a detailed proposal which includes a name, place of execution, description, photos and more. Furthermore, if it receives sufficient support from the voters and the result of the feasibility analysis is positive, the project can be performed. In practice, I created a system that I consulted with members of the Ostrava-Svinov municipal district council. I was allowed to conduct a survey of the citizens opinions. However, the web application was not developed specifically for one municipality, it can also be used by other entities that would be interested.

Klíčová slova

Informační systém, Webová aplikace, Participativní rozpočet, Angular, Node.js, PostgreSQL

Keywords

Information system, Web application, Participatory budgeting, Angular, Node.js, PostgreSQL

Citace

SEDLÁČEK, Aleš. *Informační systém pro správu participativního rozpočtu*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

Informační systém pro správu participativního rozpočtu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka, Ph.D.. Další informace mi poskytlo zastupitelstvo obce Ostrava-Svinov. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Aleš Sedláček
5. května 2021

Poděkování

Tímto bych rád poděkoval Ing. Jiřímu Hynkovi, Ph.D., za odborné vedení bakalářské práce, četné konzultace, hodnotné rady a vždy velmi podrobnou zpětnou vazbu. Dále děkuji zastupitelstvu obce Ostrava-Svinov za poskytnuté informace. V neposlední řadě děkuji Kateřině Kotarové a dalším dobrovolníkům při testování aplikace.

Obsah

1	Úvod	3
2	Participativní rozpočet	4
2.1	Definice participativního rozpočtu	4
2.2	Zásady participativního rozpočtu	4
2.3	Participativní rozpočet ve světě	5
2.4	Participativní rozpočet v České republice	9
2.5	Zhodnocení participativního rozpočtování	10
3	Veřejné informační systémy	11
3.1	IS pro širokou veřejnost	11
3.2	Techniky pro analýzu informačních systémů	13
3.3	Návrh řešení	16
3.4	Implementace řešení	19
3.5	Testování	20
4	Analýza problému	23
4.1	Cílová skupina	23
4.2	Existující řešení	27
4.3	Finální požadavky systému	29
5	Návrh aplikace	30
5.1	Typ aplikace	30
5.2	Technická stránka systému	30
5.3	Použití informačního systému	31
5.4	Uchovávání a zpracování dat	32
5.5	Uživatelské rozhraní	34
6	Implementace aplikace	37
6.1	Použité technologie	37
6.2	Frontend	38
6.3	Backend	40
6.4	Databáze	42
7	Testování	44
7.1	Testování při vývoji	44
7.2	Uživatelské testování	44
8	Závěr	46

Literatura	47
A Designové návrhy	50

Kapitola 1

Úvod

Participativní rozpočet je inovativní nástroj podporující přímou demokracii. Zapojuje širokou veřejnost do procesu rozhodování. Umožňuje přímou formou ovlivnit veřejný rozpočet. Vzniká tím možnost rozhodovat a diskutovat, jak bude s penězi naloženo. Samotná myšlenka participativního rozpočtu vznikla ve městě Porto Alegre v Brazílii v roce 1989 [22]. Dále se participování rozšiřovalo po světě a do České republiky se dostalo poprvé do povědomí až v roce 2012, kdy tuto metodu použilo několik malých měst a vesnic [15].

Informační systém pro participativní rozpočtování má za úkol lidem tuto formu rozpočtování přiblížit. Aktuálně v Česku existuje několik organizací poskytující webový systém samosprávám. Jedná se o organizaci Agora¹, nabízející své služby výhradně obcím. Momentálně zajišťuje správu systému pro 24 měst. Největší zastoupení zde mají jednotlivé městské části hlavního města Prahy. Pod druhou z těchto organizací, konkrétně pod organizací Insitute H21², funguje projekt Participace 21. Služby nabízí nejen obcím, ale i školám a neziskovým organizacím.

Cílem mé práce je vytvořit informační systém kombinující prvky předem zmíněných existujících systémů, obohacených o vlastní nápady, a který by byl založen na výsledcích dotazníku tak, aby co nejvíce dbal na potřeby potencionálního uživatele. Vybral jsem implementaci ve webovém prostředí. Důvodem výběru je snadný přístup ze všech platform jako například desktopový počítač, mobil či tablet.

Práce je rozdělena do několika kapitol. Kapitola po úvodu 2 se zabývá samotnou teorií participativního rozpočtování. Zahrnuje definici, nutné zásady definující participativní proces, rozebírá jednotlivé modely fungující po světě a nahlíží do historie jak celosvětově, tak v České republice. Třetí kapitola 3 popisuje principy tvorby informačních systémů a použitelných uživatelských rozhraní obecně. Čtvrtá kapitola 4 je zaměřena na analýzu problému, výsledky dotazníku a diskuzi s členy rady městské části statutárního města Ostrava-Svinov. Dále kapitola pět 5 pojednává o návrhu aplikace. Následuje kapitola o samotné implementaci kódu (kapitola 6). Po vytvoření reálné aplikace přichází kapitola testování 7, kdy je systém rozdělen do několika časových úseků v závislosti na tom, zda se testuje část návrhová, část hlasující nebo finální část, tedy část výsledková. Reálný provoz systému bohužel vyzkoušet kompletně nejde, jelikož se jedná o zdlouhavý proces trvající mnohdy celý rok. Závěrečná kapitola 8 shrne myšlenku práce a zhodnotí dosažené cíle.

¹Organizace Agora: <http://agorace.cz/>.

²Projekt Participace 21: <https://www.participace21.cz/>.

Kapitola 2

Participativní rozpočet

Pro demokracii je typický proces diskuze, interakce a svobodného rozhodování. Participativní rozpočtování je nástroj spadající pod tuto ideologii. Umožňuje občanům ovlivnit rozpočet obce vlastními nápady. Jestliže je navržený projekt proveditelný, má podporu obce a získal dostatečný počet hlasů od hlasující komunity, nebrání nic v cestě realizace.

2.1 Definice participativního rozpočtu

Pojem participativní rozpočet označuje proces aktivního zapojení občanů do fungování obce. Samotný rozpočet, určený k realizaci návrhů, je každoročně vyčleněná částka z celkového rozpočtu obce, která má za úkol financovat veřejně prospěšné projekty.

Přesná definice popisující participativní rozpočtování neexistuje, avšak kupříkladu program OSN pro lidská sídla [31] definici popisuje jako „*proces (mechanismus) ve kterém společnost rozhoduje anebo přispívá k rozhodování o využití části anebo celého objemu veřejných zdrojů*“. Další definice je pak od autora De Souza [11], který byl jedním z prvních iniciátorů participativního rozpočtu v Porto Alegre: „*participativní rozpočet je mechanismus přímé, dobrovolné a univerzální demokracie, kde lidé mohou diskutovat a rozhodovat o veřejných zdrojích a veřejných politikách. Občanská participace není limitovaná jen na volení zástupců, ale může být použita také na rozhodování o výdajových prioritách a ke kontrole vlády. Spojuje v sobě prvky přímé a reprezentativní demokracie*“.

Obecně lze rozdělit nápady do dvou kategorií [21]. Jedná-li se o projekt dlouhodobějšího rázu, například nové lavičky či sportoviště, označujeme tyto projekty jako projekty *tvrdé*. Naopak chce-li občan uskutečnit jednorázovou akci, workshop či setkání komunity, jedná se o projekt *měkký*.

2.2 Zásady participativního rozpočtu

Každá obec si může upravit pravidla speciálně pro své účely a navrhnout je tak, že tyto pravidla budou zcela unikátní, přitom se však musí držet několika bodů, které obecně definují participaci [37]:

- Musí být stanovena jasná pravidla.
- Obec má volenou samosprávu a úředníky.
- Jedná se o opakovaný proces (nestačí jedno kolo volby/diskuze).

- Transparentnost je základem celého procesu.
- Veřejná kontrola výsledků.

Kreativitě při navrhování se meze nekladou, navrhovatel však musí dbát několika pravidel, zaručující pozdější kladné vyhodnocení v analýze proveditelnosti. Jedná se o výčet těchto pravidel:

- Projekt musí být veřejně prospěšný.
- Musí se nacházet na území celku, který participativní rozpočet organizuje.
- Jeho náklady nesmí převyšovat maximální náklady stanovené obcí.

2.3 Participativní rozpočet ve světě

První zmínku o participativním rozpočtu datujeme k roku 1989. Jedná se o město Porto Alegre v Brazílii, které se stalo pionýrem inovativních reformních programů. Po úspěšném zavedení se metoda rozrostla do mnoha dalších obcí a států Jižní Ameriky jako Bolívie, Guatemala, Nicaragua a Peru [2]. Dále pak do Evropy a celého zbytku světa.

2.3.1 Poučení z Latinské Ameriky

Porto Alegre lze klasifikovat jako poměrně bohaté město s počtem obyvatel přesahující 1,4 milionu obyvatel k roku 2010.¹ Historie participativního rozpočtu se začala psát roku 1988, kdy vyhrála obecní volby progresivní strana s názvem Dělnická strana, jejíž kořeny spadají do období vojenské diktatury (1964-1985) [24]. Kampaň byla založena na demokratické účasti a „inverzi výdajových priorit“. Po desetiletí bylo pravidlem utrácení veřejných financí ve střední a vyšší třídě, strana však přišla se záměrem pomoci odloučené sociální skupině zapojit se více do politického dění.

Nově zvolený starosta přebíral zkrachovalou obec a neorganizovanou byrokracii. V prvních dvou letech probíhalo experimentování s různými technikami pro řešení finanční krize poskytující přímou účast v politických aktivitách a dříve zmiňovanou metodou otočení priority výdajů. Samotná metoda participativního rozpočtování vzešla z experimentování.

Účast v úvodním a následujícím ročníku participativního rozpočtu v letech 1989 až 1990 byla méně než 1000 občanů. Mírné zlepšení přišlo roku 1992, kdy se do procesu zapojilo okolo 8000 účastníků. Velkým průlomem lze považovat znovuzvolení Dělnické strany a další ročník, kdy se do participování zapojilo více než 20 tisíc lidí (tabulka 2.1). Hlavním důvodem bylo postupné zjištění obyvatelstva, že lze touto formou velmi výrazně ovlivnit okolí [35].

Rozšíření do světa

Metoda participování pomalu dosahovala větší popularity. Rozšíření lze popsat v pěti fázích datovaných v průběhu 25 let [7].

První fázi lze nazvat fází *zkušební*, jedná se o výše zmíněné město Porto Alegre a první i druhé období, trvající celkově 8 let. V tomto časovém úseku metodu uplatnilo celkově více než 30 obcí.

¹Demografický vývoj města Porto Alegre: <http://population.city/brazil/porto-alegre/>.

Rok	Belo Horizonte	Ipatinga	Recife	Porto Alegre	São Paulo
1990	n.a.	630	n.a.	976	n.a.
1991	n.a.	470	n.a.	3,694	n.a.
1992	n.a.	483	n.a.	7,610	n.a.
1993	n.a.	563	n.a.	10,735	n.a.
1994	15,216	572	n.a.	9,638	n.a.
1995	26,823	681	n.a.	11,821	n.a.
1996	36,508	604	30,000	10,148	n.a.
1997	31,795	683	n.a.	11,908	n.a.
1998	19,418	1,533	30,000	13,687	n.a.
1999	21,175	2,136	n.a.	14,776	n.a.
2000	31,369	2,018	30,000	14,408	n.a.
2001	n.a.	5,015	42,800	16,612	34,000
2002	28,124	981	67,100	28,549	55,000
2003	n.a.	2,374	69,500	26,807	80,000

Tabulka 2.1: Počet účastníků participativního rozpočtu ve městech v Brazílii, 1990–2003 [36] (* n.a. znamená nepoužitelné)

Ke druhé fázi, taktéž označovanou velkým brazilským rozšířením, došlo v dalším volebním období obcí, tudíž mezi lety 1997 až 2000, během nichž tento přístup přijalo více než 140 obcí v zemi. Vždy se značnými obměnami.

Třetí fáze se objevuje hlavně po roce 2000. Jedná se o rozšíření experimentů mimo Brazílii a se širokou diverzifikací. Děje se tak během období, kdy se v latinskoamerických a evropských městech objevilo mnoho iniciativ participativního rozpočtování, inspirovaných stávajícími modely. Zejména pak odvozené od města Porto Alegre, což v některých případech znamenalo zahrnutí podstatných změn od originálního designu.

Čtvrtá fáze ukazuje trend začínající v letech 2007 a 2008. Vznikla národní i mezinárodní síť participativního rozpočtování, kde silné slovo zaujímaly státy jako Brazílie, Kolumbie, Argentina, Španělsko ale i Německo. Zmiňuje se i nově utvořené Chilské fórum, portugalská iniciativa a projekty ve Velké Británii či Spojených státech amerických. Hlavním cílem této fáze je zmapování aktuálního rozšíření ve světě.

Pátá a zároveň poslední fáze referuje o snaze integrovat participativní rozpočet do velkých, více komplexních systémů. Stále můžeme proces nazvat procesem ve výstavbě, ale indikátory poukazují na konsolidaci v nejbližších letech.

Hlavní faktory vedoucí k úspěchu

Skrze rozšíření metody v Latinské Americe, Benjamin Goldfrank [2] sepsal na základě zkušeností několik hlavních faktorů ovlivňující úspěšné zapojení obce do formátu participování:

- Zvolený starosta původně pochází z obce, ve které figuruje. Zároveň jej lze považovat za levicově smýšlejícího.
- Opozice složená z malých lokálních stran je slabá nebo zcela neexistující.
- Financování nebo technickou podporu projektu poskytuje národní nebo mezinárodní organizace.

- Obec má dostatečné příjmy, aby mohla významně investovat do veřejných prací nebo programů.
- Existuje tradice účasti a spolupráce v rámci místních občanských sdružení nebo místních zvykových organizací.

Model Porto Alegre

Ve světě se postupně formovaly zásady, které vyústily v přesné modely. Používají je různé obce v rámci státu. Níže je popsán první model [16].

Model Porto Alegre je prvním modelem. Předpoklady jsou, aby plánovací oddělení přímo podléhalo starostovi, který je řídicím a koordinačním orgánem. Dále je potřeba založení oddělení komunitních vztahů, jehož pracovníci jsou vyčleněni pro jednotlivé regiony a mají za úkol mobilizaci místních obyvatel a sdružení. Poslední povinností je projektový systém managementu, který zajistí přehledné informace o zacházení s rozpočtovými zdroji města a o dílčích projektech.

Harmonogram modelu se skládá ze dvou částí. První probíhá od března do června, kdy se koná první kolo sousedských schůzí, regionálních a tématických shromáždění. Finální část probíhá od července do září. Jedná se o druhé kolo sousedských schůzí, regionálních a tématických shromáždění zakončené regionálním a tématickým fórem.

Rozhodovací proces je komplexní a skládá se z více na sebe navazujících částí:

1. **Sousedská shromáždění** slouží jako nejnižší úroveň. Zde jsou formulovány základní požadavky, účast je neomezená, jedná se hlavně o aktivní diskuzi.
2. **Regionální shromáždění**, kterých je v Porto Alegre přesně 16, mají funkci hlídat administrativu a stanovit pořadí priorit regionů a sousedství, volí se zástupce za region a dva radní do městské rozpočtové rady.
3. **Regionální fóra zástupců** slouží k prezentaci prioritně vybraných projektů a dále vyjednávání a kontrolu implementace administrativou města.
4. **Městská rozpočtová rada** obvykle přitahuje méně pozornosti veřejnosti, jde hlavně o řešení problematiky v oblastech jako je doprava, územní plánování, zdravotnictví nebo sociální služby a školství. Volení zástupci do Městské rady obecně komunikují intenzivněji s administrativou města, než je tomu v případě sousedských a regionálních shromáždění.

Model Saint-Denis

Dalším z příkladů je model francouzského města Saint-Denis [16]. Iniciátorem procesu je radnice a podstatou je, aby se otázek týkajících se současnosti a budoucnosti města zúčastnilo co nejvíce obyvatel. Harmonogram modelu je následující:

1. **Duben:** zvolí se z řad dobrovolníků delegáti jednotlivých městských částí.
2. **Květen:** probíhá sepisování požadavků veřejnosti.
3. **Červen:** městské zastupitelstvo zkoumá proveditelnost návrhů.
4. **Září-prosinec:** městské zastupitelstvo a delegáti rozhodují o participativním rozpočtu.

5. **Březen:** představení stanoviska delegátů před městskou radou a hlasování.

Rozhodovací proces je o něco jednodušší než tomu bylo v modelu Porto Alegre. V polovině roku se uskuteční setkání obyvatel jednotlivých městských čtvrtí, kde se formulují požadavky a delegují reprezentanti do rozpočtového výboru. Následně se vytvářejí tématické pracovní skupiny a později rozpočtové skupiny. V lednu se sestaví globální městský rozpočet zahrnující potřeby občanů vycházející z participativního rozpočtu.

Model Slovensko

Předpoklady pro model naší sousedské země jsou složeny z několika bodů [16]. Iniciátorem jsou aktivní občané a neziskové organizace. Mají na starost vedení procesu participativního rozpočtu. Obec platí práci. Dle velikosti města se rozhoduje o dostatečném počtu alokovaných zaměstnanců. Je nutná spolupráce odborů města. Jako první krok schválí zastupitelstvo částku na rozdělení v participativním rozpočtu a peníze na jeho propagaci. Příkladem fungujícího procesu jsou města Bratislava nebo Banská Bystrica.

Harmonogram je hustě členěný, skoro každý měsíc se děje něco jiného:

1. **Leden-únor:** uveřejnění informací a pravidel o participativním rozpočtu přijatém v minulém roce.
2. **Březen:** předložení zpráv o participativním rozpočtu za minulý rok do městského zastupitelstva.
3. **Duben-listopad:** pravidelné setkávání participativních komunit.
4. **Duben:** uskutečnění veřejného fóra jednotlivých komunit, prezentujíc práce a sbírání návrhů.
5. **Květen:** konzultace s odbory města.
6. **Září-říjen:** odstartování internetového hlasování.
7. **Listopad-prosinec:** prezentace vybraných projektů a schválení rozpočtu pro následující kalendářní rok v zastupitelstvu.

Hlasovací proces modelu je blízký dnešním metodám fungující i v České republice. Celoročně fungují tématická uskupení, ve kterých se scházejí lidé, kteří mají zájem o dlouhodobější práci na určitém tématu nebo místě. Členové komunit volí své členy koordinační rady. S nápady na projekty však může přijít kdokoli, stačí sepsat nápad do formuláře, který je k dispozici jak v elektronické, tak papírové formě. Hlasování je rozděleno do tří metod, kdy každá má svou váhu:

- Hlasování formou vhození obálky do urny (váha 30%).
- Internetové hlasování (váha 20%).
- Veřejné hlasování (váha 50%).

2.4 Participativní rozpočet v České republice

V České republice byl tento inovativní systém poprvé využit v letech 2012 až 2014. Jmenovitě se jednalo o menší města a vesnice Nelahozeves, Příbor, Pržno a Třanovice.

Jednalo se o velmi zjednodušenou formu procesu, kdy autority v obcích nechaly občany rozhodnout, do jakých předem vybraných projektů (hlavně kulturní nebo sportovní akce) budou vyčleněné finance vloženy. Rezidentům nebylo umožněno navrhování vlastního nápadu, mohli jedině ovlivnit distribuci výdajů. Stále se však jedná o aktivní zapojení široké veřejnosti do politiky obce a můžeme tak mluvit o participativním rozpočtování.

První opravdové participování, které se drželo základních pravidel, se uskutečnilo v městské části Praha 7 v roce 2014. Obec tehdy vyčlenila z veřejného rozpočtu jeden milion korun českých [33]. Celý proces však nezískal velký ohlas. Hlasovalo totiž jen 0,15% obyvatelstva. Problém nastal kvůli akumulaci těchto faktorů:

- Načasování do předvolební kampaně.
- Nedostatečná informační kampaň.
- Hlasování pouze fyzicky a na málo místech.

V průběhu dalších let se povědomí o participativním rozpočtování rozšířilo. Po neúspěšném sólovém pokusu Prahy 7 v roce 2014 se počet obcí, které využívaly participativní rozpočet, zvyšoval. Po komunálních volbách na konci roku 2018 se zapojilo do procesu přes 50 obcí [3]. Poslední úplný sběr dat z roku 2019 přinesl příjemné překvapení v podobě celkového zapojení 63 obcí.²

2.4.1 Model Praha 7

V českém prostředí vznikl první model v městské části Praha 7 [16]. Předpokladem je radnice jako organizátor. Radnice spolupracuje s Fórem zdravých měst³ a o proces se stará na část úvazku jeden úředník. Harmonogram města je:

1. **Červen:** zveřejnění výzvy k podání návrhů.
2. **Srpen:** konečný měsíc pro podávání návrhů.
3. **Září:** posouzení proveditelnosti projektů a veřejné setkání spojené s hlasováním.
4. **Listopad:** zveřejnění ankety neboli výsledek hlasování.

Rozhodovací proces v českém prostředí funguje na principu, kdy je základem podávání návrhů. Návrhy může podávat kdykoliv a to tak, že si sjedná konzultační schůzku na příslušném odboru. Konzultace však není podmínkou akceptování návrhu, je totiž nutné posouzení proveditelnosti komisí. Druhou fází je zapojení občanů do výběru pěti až deseti projektů do finální části. Vše je zakončeno hlasováním v anketě a realizaci vítězných návrhů.

²Participativní rozpočty v České republice v roce 2019: <https://www.participativni-rozpocet.cz/>.

³Jedná se o setkání s veřejností, kterého se účastní nejen obyvatelé, ale také neziskové organizace: <https://www.dataplan.info/cz/home/fora-zm>

2.5 Zhodnocení participativního rozpočtování

Participativní rozpočtování může přinést mnoho výhod a může pomoci v rozvoji obce. Studie naznačují, že proces může být ku prospěchu jak obyvatelstvu, tak obecnímu zastupitelstvu. Níže je shrnuto několik kladných a záporných aspektů.

2.5.1 Klady

- Pomáhá propojení občanů a zastupitelstva.
- Motivuje občany k větší spolupráci.
- Dochází k výměně cenných informací.
- Výdaje rozpočtu jsou přesněji zaměřeny na potřeby obyvatelstva.
- Snižuje možnost zneužití politického postavení.

2.5.2 Zápory

- Obtížnější komunikace s vícero subjekty najednou.
- Je těžké zapojit velké procento obyvatelstva.
- Participativní rozpočtování je nákladný a čas zabírající proces.
- Vzájemná důvěra je nezbytná pro efektivní fungování.

Kapitola 3

Veřejné informační systémy

Kapitola se věnuje tvorbě veřejných informačních systémů (IS) obecně, nahlíží na správný návrh řešení systému pro širokou veřejnost a nutnost jeho znovupoužitelnosti. Je žádáno, aby bylo možné aplikaci použít pro jakýkoliv subjekt, který má o řešení zájem. Taktéž je nahlíženo na různé techniky analýzy problému, srovnání výhod a nevýhod a nastiňují se možnosti implementace. V závěru jsou popsány formy testování vhodné k systému pro participativní rozpočet.

3.1 IS pro širokou veřejnost

Jelikož je systém určen obcím, musí být aplikace velmi uživatelsky přívětivá. Systém není navrhován do soukromého sektoru, kde aplikaci ve většině případů využívají zaměstnanci odborně proškolení a mnohdy s praxí v informačních technologiích. Musí se tedy dbát na různorodost uživatelů a navrhnout řešení tak, aby bylo co nejjednodušší, stručné, přímočaré a intuitivní.

3.1.1 Data v IS pro veřejný sektor

Důležitým aspektem IS pro veřejnost je uchovávání a následná prezentace veřejných dat, dostupných komukoliv a kdykoliv. Ukládají se však i citlivá data jako osobní údaje uživatelů. Do IS pro veřejný sektor spadají například systémy pro knihovny, muzea, ale také pro státní správu jako například finanční úřad, matriční úřad nebo úřad práce, kdy se snažíme obyvatelům i úředníkům ulehčit byrokratický proces.

Za veřejné data lze považovat kupříkladu statistiky, mapy, encyklopedie, katalogy zboží, služby nebo události. V případě systému pro participativní rozpočet to jsou projekty navržené obyvateli obce, počet hlasů, jež projekt získal, výsledky participace pro různé kalendářní roky nebo kontaktní informace organizátora.

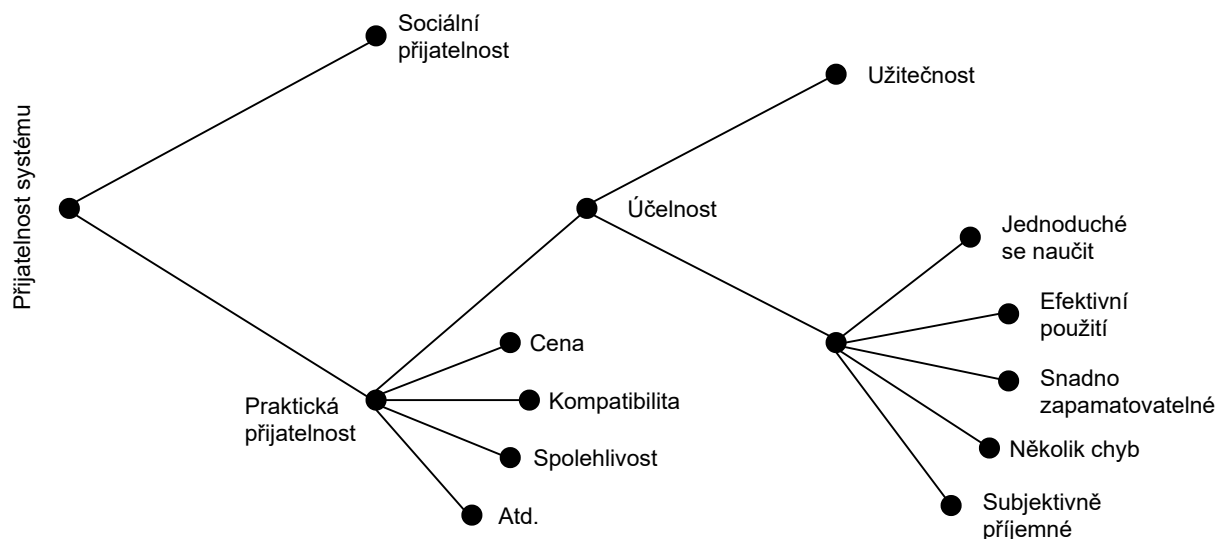
Na druhou stranu soukromá data, které ve většině případů uživatelé IS poskytují, jsou jejich jména, adresy, kontakty, platební údaje či rodná čísla a další osobní údaje. Tato data musí být velmi dobře zajištěna z hlediska bezpečnosti, jakýkoliv únik dat uživatelů by mohl znamenat velké problémy, a proto je třeba dle metodické příručky zajistit [12]:

- **Důvěrnost:** k údajům mají přístup pouze autorizované subjekty.
- **Integrita a autenticita:** data smí modifikovat jen autorizované subjekty a původ informací je ověřitelný.

- **Dostupnost:** data jsou autorizovaným subjektům do určité doby dostupná, nedojde tedy k odmítnutí služby, kdy subjekt nedostane, na co má právo.

3.1.2 Přijatelnost systému

Model přijatelnosti systému 3.1 definoval Jakob Nielsen ve své knize [26] jako seznam atributů. Model je kombinací sociální a praktické přijatelnosti, kterou dále rozděluje na kategorie jako *cena*, *kompatibilita*, *spolehlivost* a *účelnost*. Zaměřuje se hlavně na podkategorii účelnosti s názvem použitelnost.



Obrázek 3.1: Nielsenův model přijatelnosti systému, překresleno z knihy Jakoba Nielsen [26]

Použitelnost IS

Je to jeden z nejdůležitějších faktorů přijatelnosti systému. Přesněji lze pojem definovat jako vlastnost určující, jak jednoduché je uživatelské prostředí používat. Jakob Nielsen stanovil deset heuristických pravidel použitelnosti, jedná se o sadu doporučení vypsanou níže [26]:

- **Viditelnost stavu systému:** systém musí vždy dát uživateli najevo, kde se právě nachází a podávat uživateli zpětnou vazbu o tom, co právě řeší.
- **Shoda mezi systémem a skutečným světem:** design by měl být pro uživatele srozumitelný. Neměl by používat interní názvy, fráze a pojmy neznámé uživateli. Je třeba postupovat podle konvencí v reálném světě.
- **Uživatelská kontrola a svoboda:** uživatelé často udělají něco omylem. Potřebují mít tedy vždy možnost nouzového východu pro přerušování probíhajícího procesu.
- **Standardizace a konzistence:** systém by měl používat prvky, které jsou pro uživatele známé a snadno rozpoznatelné (např. ikony). Zároveň by měl být v zobrazování prvků konzistentní, tedy mít stejné funkční prvky na všech stránkách.

- **Rozeznání je lepší než vzpomnutí:** relevantní informace jsou vždy viditelné a získatelné. Uživatel si nemusí pamatovat informace při přechodu na jinou část stránky nebo dialogu.
- **Flexibilita a efektivita použití:** aplikace umožňuje používat systém co nejefektivněji a to kupříkladu pomocí klávesových zkratk nebo přizpůsobení stránky podle vlastních představ.
- **Estetický a minimalistický:** uživatelské rozhraní by mělo obsahovat pouze relevantní informace, mělo by se vyhýbat přebytečnému textu, ve kterém by mohla hlavní zpráva zaniknout.
- **Prevence chyb:** ještě lepší, než chybu dobře oznámit, je této chybě předejít. Každé kvalitní by mělo projít rozsáhlým testovacím procesem (a to průběžně, nejen v jeho zavedení), ať je co nejméně pravděpodobné, že uživatel skončí na nějaké chybě.
- **Rozpoznat, diagnostikovat a zotavit se z chyb:** chybové zprávy obsahují srozumitelný text bez chybových kódů. Přesně definují původ problému a navrhnou řešení.
- **Nápověda a návody:** přestože je vždy lepší, aby mohl být systém používaný bez nápověd či návodů, i tak je to často nutné. Lze využít sekce FAQ (*Frequently Asked Questions*), kde se uživatel může dozvědět více.

3.2 Techniky pro analýzu informačních systémů

Pro lepší pochopení cílů a požadavků systému, je třeba analyzovat problém. Existuje mnoho metod zabývajících se získáváním relevantních informací ke správnému návrhu řešení. Blíže se seznámíme s tzv. přístupem zaměřeným na člověka. Mezinárodní norma ISO 16982 [13] definuje nejčastěji používané metody a silně doporučuje kombinování vícero možností k dosažení co nejlepšího výsledku (tabulka 3.1).

V samotné kapitole o analýze problému bude blíže rozepsána použitá dotazníková metoda, rozhovory vedené se zástupci městské části Ostrava-Svinov, konkrétně se starostou obce a 1. místostarostou, jehož kompetence je podpora a rozvoj webových stránek, Facebooku a Svinovského hlasatele.

Jméno metody	Zapojení uživatelů	Krátký popis metod
Sledování uživatelů	✓	Shromažďování přesným a systematickým způsobem informací o chování a výkonu uživatelů v kontextu konkrétních úkolů během činnosti uživatele.
Měření související s výkonem	✓	Shromáždění kvantifikovatelných měření výkonu za účelem pochopení dopadů problémů s použitelností.
Analýza kritických incidentů	✓	Systematické shromažďování konkrétních událostí (pozitivních nebo negativních).
Dotazníky	✓	Metody nepřímého hodnocení, které shromažďují názory uživatelů na uživatelské rozhraní v předdefinovaných dotaznících.
Rozhovory	✓	Podobně jako dotazníky s větší flexibilitou a zahrnující osobní interakci s dotazovaným.
Přemýšlení nahlas	✓	Zahrnuje to, aby uživatelé během používání testovaného systému neustále verbalizovali své myšlenky, víry, očekávání, pochybnosti, objevy a další.
Společný design a vyhodnocení	✓	Metody, které umožňují různým typům účastníků (uživatelé, vývojáři produktů a specialisté na lidské faktory atd.) Spolupracovat při hodnocení nebo návrhu systémů.
Kreativní metody	✓/✗	Metody, které zahrnují vyvolání nových produktů a vlastností systémů, obvykle vyjmutých ze skupinových interakcí. V kontextu přístupů zaměřených na člověka jsou členy těchto skupin často uživatelé.
Metody založené na dokumentech	✗	Zkoumání stávajících dokumentů odborníkem na použitelnost za účelem odborného posouzení systému.
Přístupy založené na modelech	✗	Použití modelů, které jsou abstraktními reprezentacemi hodnoceného produktu, umožňující predikci výkonu uživatelů.
Vyhodnocení expertem	✗	Hodnocení založené na znalostech, odborných znalostech nebo praktických zkušenostech v ergonomii specialisty použitelnosti.
Automatizované vyhodnocení	✗	Algoritmy se zaměřením na kritéria použitelnosti nebo na používání ergonomických systémů založených na znalostech, které diagnostikují nedostatky produktu ve srovnání s předdefinovanými pravidly.

Tabulka 3.1: Nejpoužívanější metody analýzy [13] (* ✓ znamená zapojení uživatelů, ✗ znamená nezapojení)

3.2.1 Sumarizace analýzy požadavků

Na základě detailní analýzy je potřeba zkompletovat požadavky na informační systém. Dělí se na dva druhy a to uživatelské a systémové [20].

Uživatelské požadavky jsou popsány v přirozeném jazyce a odráží očekávání uživatelů, kteří budou systém používat. Není potřeba vědět, jak bude systém implementován, občas se však objeví v uživatelských požadavcích omezení na práci v systému či menší diagramy.

Naproti tomu požadavky systémové jsou více specifické, popisují systémové funkce, služby a omezení. Požadavky obsahují, co navrhovaný systém musí splňovat, jak se má chovat a co vše může uživateli nabídnout. Systémové požadavky se dále dělí na požadavky funkční a nefunkční:

- **Funkční požadavky:** popisují funkce systému, jež jsou poskytovány, jak by měl systém reagovat v určitých situacích, včetně situací krizových (chyby), a taktéž se zde zahrnuje, co by systém dělat neměl.
- **Nefunkční požadavky:** řeší mimo jiné výkon systému. Dále zde spadá škálovatelnost, popisující schopnost zvýšení dočasné či permanentní kapacity výpočtu bez potřeby zasahování do samotného softwaru, dále udržitelnost, spolehlivost, dostupnost a rozšiřitelnost či modifikovatelnost. Je dobré v rámci nefunkčních požadavků zvážit vytvoření architektury vhodné k dalšímu rozšiřování.

3.2.2 Funkční požadavky

Funkční požadavky se nejčastěji modelují pomocí diagramů případů užití (*use case diagram*). Use case diagram je dohoda o chování budoucího systému, diagram samotný popisuje jednotlivé aktéry případy užití. Aktér je role komunikující se systémem, kdežto jednotlivý případ užití je sada úkonů vedoucí k dosažení určitého cíle.

Příklady funkčních požadavků jsou:

- Systém umožní vytvořit požadavek.
- Systém umožní smazání vytvořeného požadavku.
- Systém umožní změnu údajů požadavku.

3.2.3 Nefunkční požadavky

Nefunkční požadavky je třeba neustále monitorovat a neopomíjet v návrhu řešení. Z výpisu požadavků výše lze uvést některé konkrétní příklady:

- Odezva požadavku nesmí přesáhnout dobu 5 sekund.
- Systém musí být dostupný na všech desktopových i mobilních platformách.
- Architektura celého systému bude spadat pod tzv. třívrstvou architekturu.

3.3 Návrh řešení

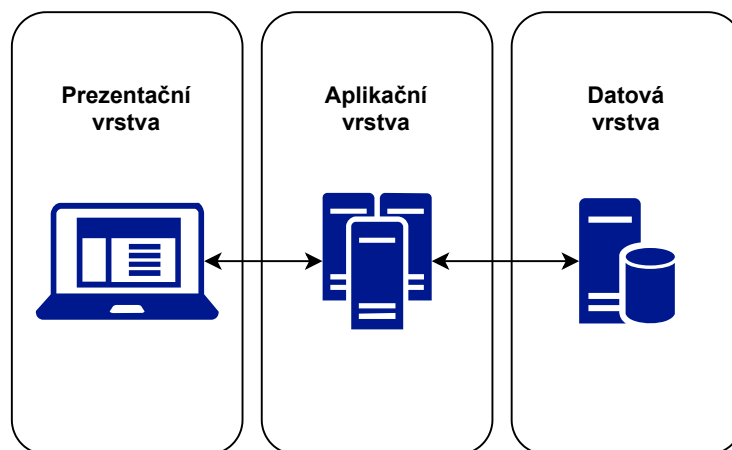
S jasným zadáním a zmapovanými požadavky všech zúčastněných stran lze přistoupit k návrhu řešení. Následuje obecný popis návrhu architektury a z jakých částí se skládá. Dále pokračuje popis dat, začínajíc textovým výkladem a následnou iterací se dostaneme ke grafickým prvkům vyobrazení dat, respektive diagramům. Na závěr budou představeny jednotlivé modely dat s exemplárními příklady a možnosti návrhu designu aplikace.

Architektura systému

Většina moderních architektur webových aplikací používá vícevrstvou architekturu, kdy logicky rozděluje celky kódu do vrstev.

Jednou z nejpopulárnějších architektur ve webovém prostředí je třívrstvá architektura [8]. Vyznačuje se rozdělením do 3 logických vrstev (obrázek 3.2):

- **Prezentační vrstva:** první vrstva, vyznačuje se interakcí uživatele s aplikací. Dostupná přímo z prohlížeče, operačního systému nebo GUI (Grafické uživatelské rozhraní) aplikace.
- **Aplikační vrstva:** taktéž nazývána logickou vrstvou nebo prostřední vrstvou, je jádrem aplikace. V této úrovni se sbírají informace z prezentační vrstvy a provádějí se změny na datové vrstvě pomocí vytvořeného rozhraní.
- **Datová vrstva:** někdy nazývána databázová vrstva. Místo, kde ukládáme a pracujeme s informacemi zpracovanými aplikační vrstvou. Data mohou být uložena v relačních, objektově-relačních nebo čistě objektových databázích.

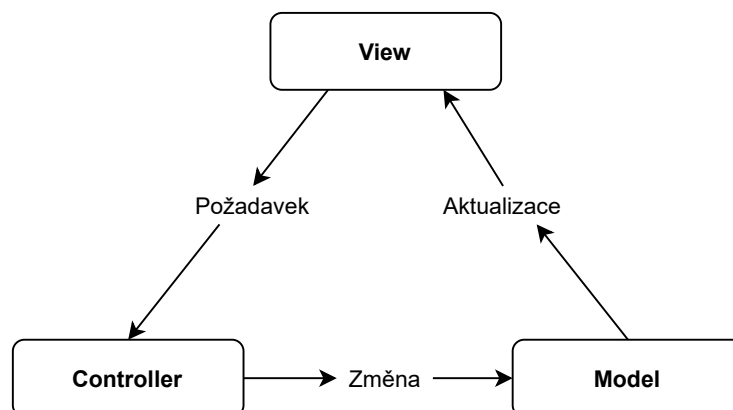


Obrázek 3.2: Jednotlivé prvky třívrstvé architektury

Další architekturou je MVC neboli model-view-controller, skládající se rovněž ze tří vzájemně propojených částí. Nápadně se architektura může podobat dříve popsané třívrstvé architektuře (obrázek 3.3), jenže zde platí pár rozdílných specifik [17]:

- **Model:** obsahuje data a stav hodnot. Představuje data, která se přenášejí mezi komponenty nebo jakoukoli jinou související logikou. Příkladem mohou být uložené informace o zákazníkovi z databáze.

- **View:** stará se o prezentaci výstupu uživateli. Nejčastěji se jedná o HTML šablonu obsahující tagy značkovacího jazyka. Stejně jako model neví, odkud data přišla.
- **Controller:** prostředník pro funkčnost celého vzoru. Komunikuje jak s modelem, tak s view prvkem. Existuje mnoho přístupů implementace, avšak nejčastěji má každá entita právě jeden controller.



Obrázek 3.3: Jednotlivé prvky MVC architektury

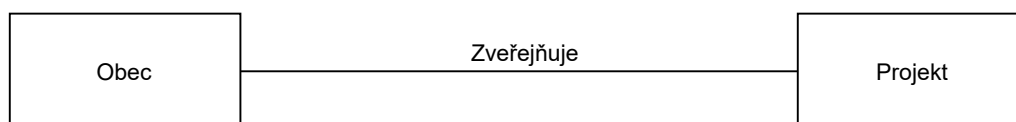
Třívrstvá architektura i architektura MVC mohou společně tvořit jeden celek informačního systému. Třívrstvá architektura definuje obecnější rozdělení vrstev prezentačních, aplikačních nebo datových, kdežto architektura MVC tvoří jednoduchou strukturu kódu, ve které se lze rychle a snadno orientovat. Architekturu MVC najdeme nejčastěji v různých knihovnách a nastavbách programovacích jazyků.

3.3.1 Metody návrhu jednotlivých součástí IS

Požadavky zákazníka jsou většinou v textové formě a zohledňují, čeho všeho by měl být IS schopen a čeho by se měl vyvarovat. Jednotlivé role uživatelů systému poté znázorníme graficky pomocí diagramu případu užití dříve zmíněného ve funkčních požadavcích 3.2.2. Další metody návrhu vyvozujíc z požadavků zákazníka mohou zohledňovat entitně vztahový model (ERD) pro konceptuální znázornění dat nebo designový návrh aplikace.

Konceptuální modelování dat

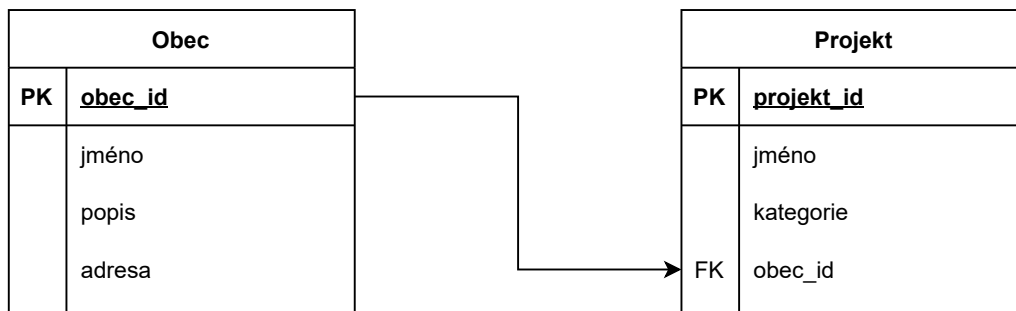
Konceptuální model se zabývá modelováním reality. Vzniká za účelem popisu dat v databázi bez ohledu na jejich uložení. Jednou z možností je tzv. ER (z angličtiny Entity-relationship) model, poskytující formální návrh databáze a jejího logického schématu pomocí kolekce entit a vztahů (obrázek 3.4).



Obrázek 3.4: Ukázka jednoduchého ER diagramu mezi dvěma entitama

Logické modelování dat

Logický model se na druhou stranu vztahuje ke konkrétnímu datovému modelu a používá jeho konstrukční dotazovací a manipulační prostředky. Možným nástrojem je relační model dat (obrázek 3.5), založený na matematické teorii množin a predikátové logice a definuje způsob reprezentace dat, způsob jejich ochrany (integritní omezení) a možné operace nad daty.



Obrázek 3.5: Grafický návrh drátěného mobilu aplikace pro mobilní a desktopové zařízení

Designové návrhy

Návrh uživatelského rozhraní aplikace začíná drátěným modelem (*wireframe*), na kterém je odladěna logická struktura obsahu, ovládání a navigace. Dle požadovaného typu aplikace a použití (desktopové zařízení, chytré telefony nebo tablety) je následně vytvořen grafický návrh, který je v iteracích konzultován a laděn s klientem.

Cílem grafického návrhu není jen vytvoření kvalitního designu, který bude splňovat představu klienta a plně vyhovovat uživatelům, ale také ukázat ovládání a funkční možnosti aplikace, aby bylo zřejmé, jak výsledná aplikace funguje. Grafické návrhy procházejí iteračním procesem a existuje několik typů návrhů [32].

Základní drátové modely

Hlavní výhodou je rychlost tvorby. Náčrt lze vytvořit za pár minut, s pomocí tužky a papíru, na tabuli nebo pomocí speciálního softwaru. Modely nevyžadují žádné znalosti programování. Další výhodou jsou velmi nízké náklady. Navíc lze modely snadno revidovat, buďto gumou nebo uspořádáním obrázků.

Proklikávací modely

Propojují různé obrazovky prostřednictvím klikáním na akční body. Tyto prototypy jsou pokročilejší verze základních drátových modelů a také nejjednodušší verze interaktivních prototypů. Některé aplikace pomohou vytvořit prototypy od samého začátku, zatímco jiné umožní nahrávat skici nebo papírové prototypy a vytvořit tak digitální podobu, aby bylo možno návrhy otestovat a iterovat.

Kódované prototypy

Vytvoření kódovaného prototypu je složitější, protože vyžaduje, aby návrhář nebo grafik uměl programovat. Tyto prototypy jsou nejvíce podobné konečnému designu produktu a jsou nejlepší volbou pro generování zpětné vazby od uživatelů.

3.4 Implementace řešení

Po kompletním návrhu aplikace následuje implementace. Implementace zahrnuje výběr technologií, přesnou specifikaci jednotlivých programových modulů, tvorbu testovacích prototypů a následně finálního řešení. Ve vývoji webových aplikací se používají důležité pojmy *frontend* a *backend*. Úzce navazují na předchozí třívrstvou architekturu. Frontend lze považovat za prezentační úroveň a backend přirovnáváme k aplikační části nebo pokud zahrneme do této části i databázi, pak můžeme backend nazývat jak aplikační, tak datovou vrstvou.

3.4.1 Frontend

Klientská část popisující interakci uživatele se systémem. Obsah frontend části tvoří veškeré uživatelské rozhraní a jejich komponenty typu grafy, navigace, tabulky, tlačítka a další. Strukturu, design, chování a obsah všeho, co se zobrazuje na obrazovce prohlížeče při otevření webů, webových aplikací nebo mobilních aplikací, implementuje vývojář frontend části.

Nejčastěji využitý je značkovací jazyk HTML zprostředkovávající šablonu webu obohacenou o styly v jazyce CSS. O lehkou logiku zahrnující validaci se stará jazyk JavaScript na klientské vrstvě. Pro implementaci existuje mnoho frameworků usnadňující programátorům práci. Jedná se o balíčky obsahující předepsaný standardizovaný kód v souborech a složkách. Nabízejí základ, na kterém se staví finální design systému.

Prvním z abecedního seznamu je Angular. Jedná se o jediný framework založený na jazyce TypeScript, což je nástavba jazyku JavaScript. Největším kladem je tzv. *two-way binding*. Jedná se o real-time synchronizaci dat s prezentační vrstvou. Druhým jmenovcem je React. Jeden z jednodušších frameworků na naučení. React vystupuje z řady tím, že používá virtuální DOM (*Document Object Model*), který má mnoho nadstandardních funkcí. Posledním vyjmenovaným frameworkem je Vue.js. Oproti oběma zmiňovaným frameworkům je Vue.js o mnoho menší ve velikosti a kombinuje obě výhody Angularu a Reactu – virtuální DOM a two-way binding, na druhou stranu nemá takovou komunitní podporu a teprve získává na popularitě [6].

3.4.2 Backend

Důležitou součástí je server, aplikace a databáze. Samotný backend je neviditelným jádrem klientské aplikace, která zajišťuje chod pomocí vystaveného API (Application programming interface) [19].

Mezi příklady backend procesů patří [4]:

- zpracování příchozího požadavku z webové stránky,
- přístup k datům z databáze pomocí dotazů SQL,
- ukládání nebo aktualizace záznamů v databázi,

- šifrování a dešifrování dat.

Backend si lze představit jako aplikační server zpracovávající HTTP požadavky [28]. Konkrétní příklady aplikačních serverů se liší od programovacího jazyka. V jazyku Java existuje technologie Tomcat¹ nebo JBoss². Pod jazyk JavaScript spadá technologie Node.js³ určená k vytváření škálovatelných síťových aplikací.

Spojujícím rozhraním komunikace na aplikačním serveru je velmi často tzv. rozhraní REST [5]. Je to architektura, navržená pro distribuované prostředí. Rozhraní implementuje čtyři základní metody pod označením CRUD, blíže to jsou metody:

- **GET (Retrieve):** pro přístup ke zdrojům.
- **POST (Create):** pro vytvoření dat.
- **PUT (Update):** podobná metoda jako POST, avšak používající se převážně pro změnu dat.
- **DELETE (Delete):** pro smazání zdroje.

Poslední součástí backendu je databáze neboli datová vrstva. Její hlavní činností je ukládání, modifikace a mazání dat. Rozdělit databázi lze minimálně do dvou tříd: relační a nerelační. Relační databáze je založená na tabulkách, které uchovávají jednotlivé hodnoty záznamů a vztahy mezi jednotlivými entitami. Příkladem relační databáze je MySQL, Oracle nebo PostgreSQL. Nerelační databáze je na druhou stranu založena na nestrukturovaných datech. Místo tabulek jsou zde přímo objekty a místo řádků se ukládají samotné instance objektů. Nerelační databázi lze nazývat kupříkladu MongoDB.

3.5 Testování

Po implementaci prvního prototypu nebo při iteračním procesu zlepšování je třeba provádět testování. Stejně jako u analýzy požadavků, testování se dělí na dva celky. Prvním celkem jsou *black box* testy. Neuplatňují se přímo v kódu, nýbrž kontrolují reakci systému na požadavky uživatele. Typicky se jedná o funkční nebo akceptační testování. Druhým celkem je *white box* testování. Hledí na kontrolu správnosti kódu a mezi příklady klasického *white box* testování patří *unit testing* nebo *E2E (End-to-End)* testování.

3.5.1 Iterativní proces zlepšování IS

Správným způsobem, jak postupně zlepšovat použitelnost systému je iterativní proces. Čím více verzí a rozhraní vytvoříme, tím lepší bude produkt. Jednou z možností iterativního procesu zlepšování je uživatelské testování. Není třeba vynaložit velké finanční nebo časové náklady na mohutné testování. Stačí se držet několika jednoduchých pravidel [25]:

1. Vybrat reprezentativního uživatele, například žadatele cestovního pasu skrze informační systém nebo zákazníka e-shopu.
2. Požádat uživatele, aby vykonal sadu úkolů.

¹Apache Tomcat: <http://tomcat.apache.org/>

²JBoss: <https://www.redhat.com/en/technologies/jboss-middleware/application-platform>

³Node.js <https://nodejs.org/>

3. Sledovat, co uživatelé dělají. Zjistěte, kde chybují a kde naopak vše pochopí správně a rychle. Nechte uživatele mluvit.

Důležité je uživatele nechat pracovat jednotlivě a bez pomoci. Jakákoliv nápověda může způsobit znehodnocení výsledku. Závěrem je nutno ještě jednou zdůraznit, aby bylo uživateli při testování nejen nasloucháno, ale hlavně aby byl testovací subjekt nepřetržitě sledován.

3.5.2 Funkční testy

Testuje se část kompletně funkčního systému. Existují například:

- **Smoke testy:** po kompilaci kódu a následné spustitelné aplikaci se zkouší základní nepostradatelné funkcionality. Jedná se o rychlé ověření, zda hlavní funkce programu, které nebývají často upravovány, fungují.
- **Sanity testy:** podobný jako smoke test, ověřuje se racionalita malých změn a funkcionality. Exemplárním příkladem může být test správného výsledku $2+2$, nezkouší se pokročilá logika typu $\sin(30)+\cos(20)$.
- **Regresní testy:** potvrzení, zda nedávná změna kódu nepozměnila již implementovanou funkcionalitu existujícího kódu.
- **Testy použitelnosti:** vytvoření úkolů a předání kontroly nad systémem testovacímu subjektu. Kontrolovat lze real-time, sleduje se orientace, schopnost rychlé reakce v systému a další aspekty. Závěrečným posudkem uživatele je buďto splnění/nesplnění úkolu.

3.5.3 Unit testy

Přichází na řadu nejdříve. Píší je programátoři a jedná se o ověření samotného kódu. Testují se tzv. jednotky [29]. Nehledí se na správnou funkcionalitu propojení jednotlivých komponent. Unit testování může sloužit i jako dobrá dokumentace. Ke kódu se může dostat nový programátor nebo programátor po velmi dlouhé době a díky unit testům rychleji pochopí, jaký je význam kódu.

3.5.4 E2E testy

Ověřuje kompletní aplikační tok. Zkratka E2E neboli *End-to-end* popisuje proces systému nebo služby od začátku do konce a jeho kompletní funkční řešení [27]. Při velké komplexnosti systému se stává nereálným manuální ověření všech nových změn a odchyčení všech chyb. Proto existují E2E řešení fungující na automatizaci testů různých scénářů. Může se jednat například o kontrolu prvků uživatelského rozhraní, nalezení HTML elementu, následné kliknutí nebo vyplnění formuláře a kontroly hraničních hodnot. Například framework Angular poskytuje ve svých projektech knihovnu Protractor. Programátor vytvoří scénář a očekává konstantní chování/výstup systému.

3.5.5 Akceptační testy

Realizace testů prováděna na straně zákazníka. Předává se funkční software a ověřuje se, zda produkt splňuje očekávané funkce a kvalitu. Zjišťují se chyby, které se při vývoji nepodařilo najít. Zákazník nemá zažité praktiky využívání aplikace a chová se v novém prostředí nepředvídatelně.

Frontend framework Angular pracuje s knihovnami Jasmine a Karma, které generují unit testy. Výchozí nastavení Angular projektu způsobí, že každý nově vytvořený komponent obsahuje i vlastní soubor s vytvořenou šablonou pro testování. Na backendu funguje například knihovna Mocha nebo Jest poskytující snadné mockování dat a pokrytí kódu testy.

Kapitola 4

Analýza problému

Prvním bodem správného informačního systému je zjištění potřeby uživatelů. Následující kapitola popisuje konkrétní požadavky cílové skupiny, která byla rozdělena na dvě části, veřejnost a zastupitelstvo obce. Posléze jsou rozebrána existující řešení a závěrem shrnuty kompletní požadavky systému.

4.1 Cílová skupina

Pojem cílová skupina označuje v marketingu demografickou skupinu lidí, kteří se s největší pravděpodobností budou zajímat o službu společnosti nebo produkt (reklamní kampaň). V případě informačního systému jsou to potenciální obyvatelé obce nebo zastupitelé. Cílové publikum lze rozdělit dle několika demografických kritérií [18]:

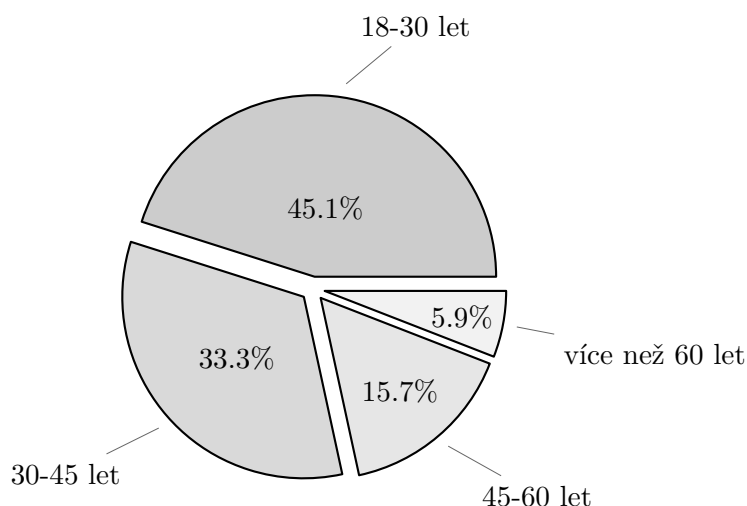
- **Geografické:** službu informačního systému nabízíme skupině lidí se zájmem o rozšiřování veřejného blaha jedné obce. Přesah je vítán, uživatelé nemusí být z jedné obce, pro podporu veřejně prospěšného místa může projevit zájem kdokoliv z celého regionu nebo kraje.
- **Věkové:** řada výrobků a služeb je určena pro určitou generaci. V rámci aplikace pro veřejnost je však věkové rozložení různorodé a je třeba zohledňovat potřeby všech.
- **Pohlaví:** v dnešní době se mezi pohlavní rozdíly stírají. Podobně jako u věkového kritéria je potřeba zohlednit všechna pohlaví.
- **Vzdělání a společenské postavení:** pomocné síly se základním vzděláním jsou zvyklé na zcela jinou komunikaci než vrcholní manažeři s vysokoškolským vzděláním, což je potřeba zohlednit třeba ve složitosti jazyku, jakým systém s uživatelem komunikuje. Spousta odborných, přejatých nebo cizích slov může vést k neochotě uživatele podat návrh nebo se účastnit hlasování.
- **Životní styl:** lidé jsou různí. Nehledě na lokalitu, věk, pohlaví nebo vzdělání mohou být zastánci různého životního stylu. Někteří jsou sportovně založení, jiní tráví svůj čas na pohovce u televize. Jedni vyhledávají společnost, druzí se straní okolnímu světu. Životních stylů je nepřehledné množství a mohou se vzájemně proplétat. Obecně je v systému potřeba myslet na různorodost lidí a jejich potřeby.

Následně jsou rozebrány dvě cílové skupiny, které se budou se systémem stýkat. Jedna z nich je veřejnost, velmi široká skupina z hlediska demografických kritérií a druhá skupina je zastupitelstvo obce, jejichž potřeby jsou konkrétnější a demografické rysy předvídatelnější.

4.1.1 Veřejnost

Pro zjištění demografických rysů, potřeb a zájmech byla použita dotazníková metoda analýzy zmiňována v tabulce 3.1. Pomocí nástroje pro vytváření formulářů od společnosti Google¹ byl vytvořen dotazník dotazující se na několik kritérií zohledňující funkční požadavky.

Z hlediska geografického demografického kritéria byl dotazník uveřejněn na sociálních sítích městské části Ostrava-Svinov². Věk ve vyplněných dotaznících se lišil, největší část zabírá věková kategorie 18-30 let a 30-45 let viz graf 4.1, v nemalém počtu se ale objevují všechny věkové kategorie. Systém je tudíž třeba navrhnout tak, aby byl co nejjednodušší a přístupný všem lidem bez ohledu na technické schopnosti a schopnosti používat počítač, mobil či tablet.

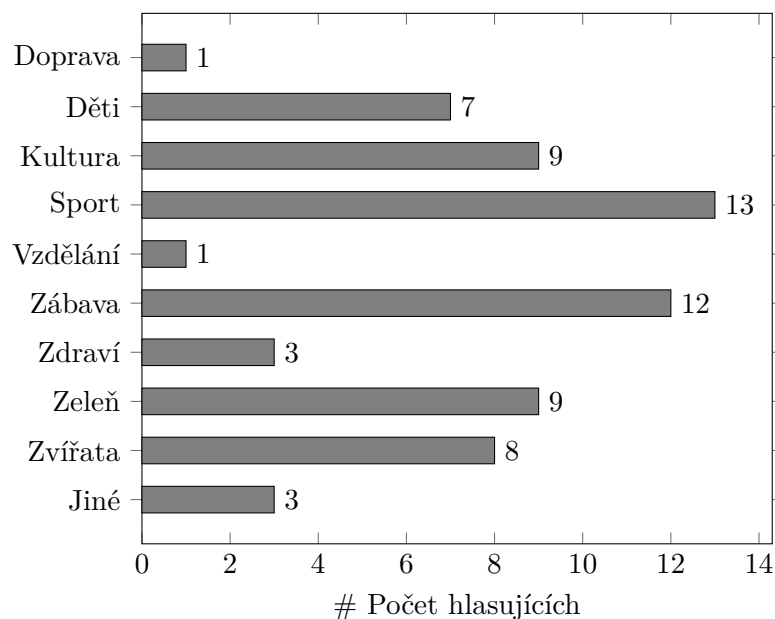


Obrázek 4.1: Koláčový graf znázorňující věkovou kategorii veřejnosti

Poslední otázkou korelujících s demografickým kritériem životního stylu je otázka z dotazníku: „Do jaké kategorie toto místo veřejně prospěšného rázu spadá?“. Položená otázka je podmíněna kladnou odpovědí na dotaz: „Postrádáte ve svém okolí místo veřejně prospěšného rázu? (dětské hřiště, výběh pro psy, lavičky, ...)“. Zjišťujeme, jaké kategorie veřejně prospěšného místa obyvatelům obce chybí. Výsledkem analýzy je komplementace kategorií, které se budou objevovat v informačním systému. Zároveň bylo umožněno dotazníkem zvolit možnost *Jiné* a definovat kategorii vlastní. Dotazovaní avšak při výběru této kategorie nebyli dostatečně angažováni a novou kategorií nevyplnili nebo nechtěli vyplnit. Z toho důvodu v systému bude figurovat seznam kategorií korelujících s grafem 4.2 níže.

¹Formuláře Google: <https://docs.google.com/forms/>

²Facebook příspěvek na stránce městské části Ostrava-Svinov: <https://www.facebook.com/svinov/posts/4115767928435018/>



Obrázek 4.2: Pruhový graf znázorňující typ nového místa veřejně prospěšného rázu

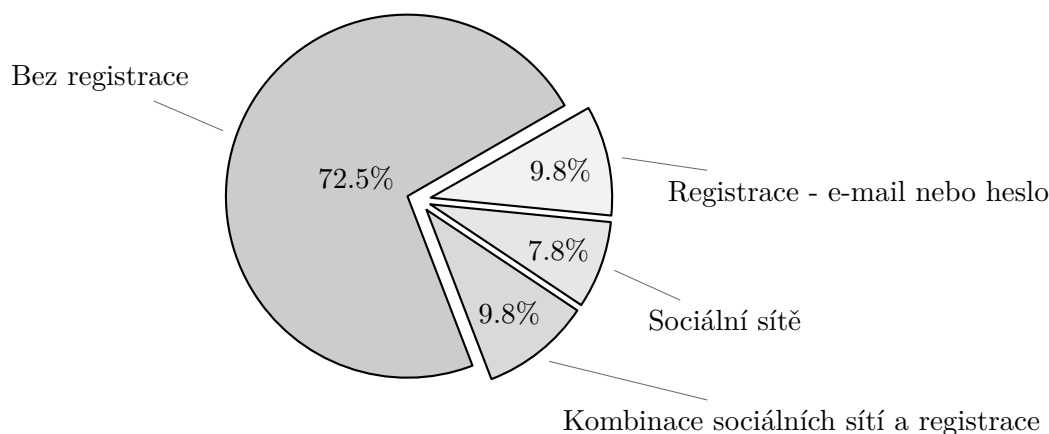
Požadavky na funkčnost systému

Dotazník přiložený obyvatelstvu Svinova obsahuje i otázky, jejichž odpovědi formují funkčnost samotného systému. Prvním celkem je navrhování projektu. Ochota detailně popsat navrhovaný projekt je důležitá informace, kterou bylo potřeba zanalyzovat. Veřejnost dostala možnost vybrat, které informace je ochotna sdělit. Název, popis a adresu projektu je ochotna vypsat drtivá většina dotazovaných. Propad ochoty byl viditelný u vyfocení prostoru uskutečnění nebo soupisu celkových finančních nákladů.

Další otázka směřovala k osobním údajům navrhovatele. Anonymní projekty není možné vytvářet, vždy je nutné mít alespoň jeden validní identifikátor, který zamezí nadměrnému zneužívání. Jméno autora a e-mail jsou ochotni sdělit bez mála výjimek všichni, na druhou stranu telefonní číslo a adresu bydliště spousta dotazovaných uvádět nechce, z toho důvodu nebudou v informačním systému viditelné.

Druhým celkem je forma hlasování. Systém hlasování nesmí být zneužit a musí zabránit opakovanému hlasování jedné osoby. Pro tento účel je třeba identifikovat každého hlasujícího. Nejpříjemnější možností plynoucí z výsledků (graf 4.3) je pro dotazované hlasování bez potřeby nutnosti registrovat. Autentizace bude navržena s ohledem na výsledek a nebude požadovat složitý proces přihlašování nebo registrace. Jediný vstup podmíněný heslem bude pro administrátory, respektive zástupce obce.

Poslední otázkou formující hlasování byla: „V hlasovacím seznamu se nachází nespočet projektů rozdělených podle kategorií. Kolikrát byste chtěli mít možnost hlasovat? (jedná se jen o kladné hlasy)“. Jasným výběrem, s většinou hlasů byly 3 hlasy. Informační systém bude reflektovat výsledek otázky dotazníku a nebudou hlasování nijak dále modifikovat.



Obrázek 4.3: Koláčový graf znázorňující způsob hlasování

4.1.2 Zastupitelstvo obce

Následuje analýza druhé strany. Zastupitelstvo se zabývá správou informačního systému, vyvěšováním pravidel participativního rozpočtu, časového harmonogramu. Kontakt byl navázán se samotným starostou městské části Ostrava-Svinov a poté i s 1. místopředsedou obce se specializací na investiční akce, ale i podporu pro rozvoj webových stránek, sociální sítě Facebook a Svinovského hlasatele³, za účelem rozeslání dotazníku.

Schůze se starostou obce nastínila již existující systém participativního rozpočtování, do kterého se Svinov zapojil. Jedná se o participování na úrovni celého statutárního města Ostravy. Velké kontroly se ale obci nedostává a městská část funguje hlavně jako oblast možné realizace. Při rozhovoru byl kladen důraz na možnost úprav projektů. Dle slov 1. místopředsedy Svinova občan sice může mít originální a potřebný nápad, ale detaily realizace mohou být nad jeho síly. Laik bez odborné znalosti nedokáže navrhnout přesné a detailní náklady nebo náklady úplně vynechá a obec následně může nedostatky doplnit.

Dalšími požadavky zastupitelů na systém jsou:

- Koordinátor má mít možnost provádět posouzení o proveditelnosti projektů.
- Aplikace má dovolovat úpravu harmonogramu jednotlivých částí participace.
- Lze jednoduše nahlížet do předešlých ročníků participativního rozpočtu.
- Možnost definovat oblast působnosti a následně kdykoliv mít možnost vymezenou oblast modifikovat.
- Přidávat zástupce obce pro případ, kdy by jeden koordinátor systému pro participativní rozpočet nedokázal zpracovávat všechny požadavky.

³Odkaz na Svinovský hlasatel: <https://svinov.ostrava.cz/cs/o-svinove/zpravodaj>

4.2 Existující řešení

Informační systémy pro správu participativního rozpočtu v nynější době nabízí několik organizací. V České republice se jedná o zejména 2 organizace. Jedná se o Participaci 21 pod záštitou Institutu H21⁴ a participativní rozpočet od neziskové organizace Agora⁵.

4.2.1 Participace 21

Platforma nejen pro obce. Fungující software institutu podporuje i participaci ve školách a organizacích. Od ostatních platform se odlišuje především způsobem hlasování. Použita je metoda D21 [14], kdy každý volič má vždy k dispozici více hlasů, než kolik je vítězných možností. Všechny hlasy mají stejnou váhu a volič je může, ale nemusí využít všechny. Pro každou z možností přitom lze hlasovat pouze jedním hlasem. Metoda umožňuje využít také minusový hlas, pro jeho udělení je však potřeba udělit alespoň dva plusové hlasy. Pro odevzdání hlasů je potřeba vyplnit dotazník s demografickými údaji pro sestavení finálních statistik hlasování, samotná autentizace uživatele ale chybí nebo není jasně viditelná, z pohledu bezpečnosti může být systém náchylný na zneužití hlasovací části a znehodnocení výsledků. Na obrázku 4.4 je názorná ukázka systému.



Obrázek 4.4: Testovací řešení pro systém Participace 21

4.2.2 Participativní rozpočet Agora

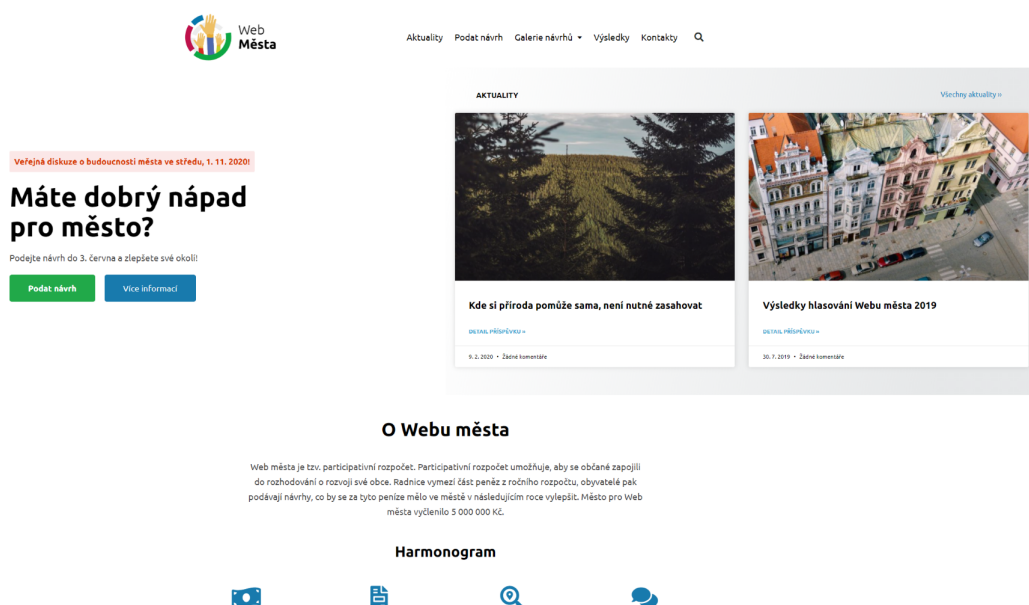
Jedná se o neziskovou organizaci, zabývající se podporou obcím ve formě realizace participativního webu. Provází koordinátora participativním rozpočtováním (zastupitele obce) pomocí své vlastní vytvořené metodiky popisující přípravu, vyhodnocení a příklady participativního rozpočtu (obrázek 4.5). V nynější době organizace podporuje více než 32 měst⁶.

⁴Institut H21: <https://www.ih21.org/>

⁵Agora: <http://agorace.cz/>

⁶Počet zapojených měst pro participativní rozpočet organizace Agora: <https://www.participativni-rozpocet.cz/zapojena-mesta/>

Samotná obec Ostrava-Svinov by se do projektu zapojit mohla, musela by ale dodržovat striktně zavedenou metodiku systému. Definovaný harmonogram je neměnný a obec se podle něj musí řídit.



Obrázek 4.5: Testovací řešení pro systém participačního rozpočtu Agora

4.2.3 Brněnské řešení na míru

Participativní rozpočet s názvem „Dáme na vás“ pro město Brno⁷ je projekt vytvořený na míru. První zmínka se datuje k roku 2016, kdy vznikla koaliční smlouva pro zastupitelstvo města Brna v období 2016-2018 [1]. Vyskytuje se zde koncept města Smart city, pojednávající o možnosti participování: „Využijeme nové modely participace občanů na chodu města. Vytvoříme metodiku participace, která nastaví pravidla komunikace a zapojování veřejnosti do rozhodování. Pilotně otestujeme participativní rozpočet“. První ročník proběhl v roce 2017 a do samotného navrhování projektu i hlasování mohl zasáhnout kdokoli bez podmínky trvalého bydliště.

4.2.4 Shrnutí existujících řešení

Veškeré zmíněné znovupoužitelné systémy nebo systém na míru pro město Brno jsou kvalitní a použitelné. Důkladná analýza však naznačila možné mezery v nabízené volnosti zastupitelů při tvorbě pravidel, harmonogramu a nastavení celého systému. Zároveň se navrhovaný informační systém zaměřuje na bezpečnost hlasování, která v ostatních systémech pokulhává nebo nebyla dostatečně jasně nastíněna. Svou metodou autentizace uživatele skrze telefonní číslo minimalizuje možnost zneužití.

⁷Brněnský informační systém pro správu participačního rozpočtu: <https://damenavas.brno.cz/>

4.3 Finální požadavky systému

Po analýze jednotlivých cílových skupin a existujících řešení vznikly konkrétní požadavky na systém. Požadavky jsou shrnuty do bodů níže:

1. Systém rozlišuje dva typy uživatelů: občany a zastupitele (koordinátory).
2. Hlasující občan musí být řádně ověřen.
3. Jakýkoliv uživatel systému má možnost nahlížet do hlasovací ankety a filtrovat projekty podle kategorie, názvu nebo počtu hlasů.
4. Uživatel může nahlížet do historie výsledků participativního rozpočtu předešlých let.
5. Uživatel se může informovat o časovém harmonogramu jednotlivých částí participativního rozpočtu.
6. Občan může navrhnout projekt, pakliže se participativní proces nachází v navrhovací části harmonogramu.
7. Navrhovací formulář obsahuje sekce: název projektu, jméno autora, kontaktní údaje autora, kategorie projektu, možnost nahrávání fotky, popis projektu, soupis nákladů a možnost zanesení souřadnic projektu na mapě.
8. Uživatel může osobně kontaktovat obec.
9. Zastupitel má možnost vstoupit do interního systému obce.
10. Interní systém obce obsahuje možnost rozhodovat o proveditelnosti projektech, nastavení harmonogramu, vymezení oblasti působnosti obce na mapě, nastavení loga, kontaktních informací a správu uživatelů.

Kapitola 5

Návrh aplikace

Tato kapitola se věnuje popisu postupu návrhu webové aplikace pro správu participativních rozpočtů. Výsledný návrh vznikl po kompletní analýze informací a sepsání všech požadavků. V první části se řeší typ aplikace a její architektura. Následně jsou pomocí diagramu použité rozebrány jednotliví aktéři aplikace. Poslední částí kapitoly je popis modelu dat a prototypu uživatelského rozhraní.

5.1 Typ aplikace

Informační systém navrhnut pro participativní rozpočet je určen pro většinu prohlížečů jak na desktopových, tak mobilních zařízeních. Záruku optimalizace zobrazení pro všechny druhy zařízení (mobily, notebooky, tablety atd.) tvoří responzivní webový design. Ať už se uživatel připojí z jakéhokoliv zařízení, je mu nabídnuta plná funkcionality bez omezení ve formě prohlížení, navrhování, hlasování či správě interního systému.

5.2 Technická stránka systému

Aplikace je rozdělena do tří logických celků tvořících jeden komunikující ekosystém. Jedná se o třívrstvou architekturu vytvořenou prezentační (frontend), aplikační (backend) a datovou vrstvou.

5.2.1 Třívrstvá architektura

Prezentační vrstva

Webové prostředí poskytující přístup k obsahu participativního rozpočtu. Uživatel vyplňuje vstupy formulářů a po odeslání očekává reakci systému. Zároveň je zpřístupněno uživateli hlasovat pro jednotlivé projekty. Prostředí následně vyžaduje pro akceptování hlasu zadat SMS (z anglického Short message service) kód obdrženy po zadání telefonního čísla. Vše je komunikováno s aplikační vrstvou pomocí API rozhraní.

Aplikační vrstva

Služba složena z API rozhraní pro komunikaci prezentační vrstvou a databázovou vrstvou. Přijímá vstupy prezentační části a následně probíhá zpracování dat a vyhodnocení. Aplikační část se rozděluje na dvě části:

1. **Služba zpracovávající požadavky uživatelů a autorizace:** poskytující komunikaci s databází obsahující informace o uživateli a interním systému.
2. **Služba zpracovávající požadavky systému:** pro přístup k veřejným datům participativního rozpočtu a logiku prezentační části.

Databázová vrstva

Vrstva je rozdělená do dvou celků. První databáze je relační databáze založena na dvourozměrné struktuře tvořené záhlavím a tělem. Jednotlivé tabulky a jejich propojení jsou dále popsány v části 5.4 této kapitoly.

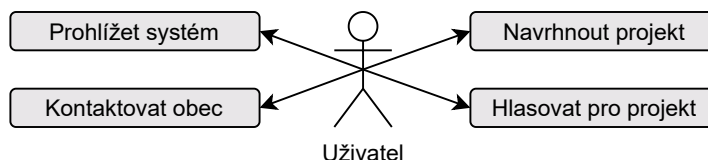
Druhá databáze služby Firebase má formu NoSQL databáze¹. Databáze je dokumentově orientovaná a každý dokument tvoří jednoduchý zápis tvořen z jednotlivých klíčů namapovaných na hodnotu.

5.3 Použití informačního systému

Diagramy případů použití ukazují možnosti práce s informačním systémem z různých pohledů. Existují uživatelé bez přístupu do interního systému, koordinátor obce a koordinátor se speciálním přístupovým oprávněním „majitel“.

5.3.1 Uživatel

Každý nový uživatel, který se dostane do informačního systému obce má automaticky přiřazené základní pravomoci dle diagramu případu užití (obrázek 5.1).

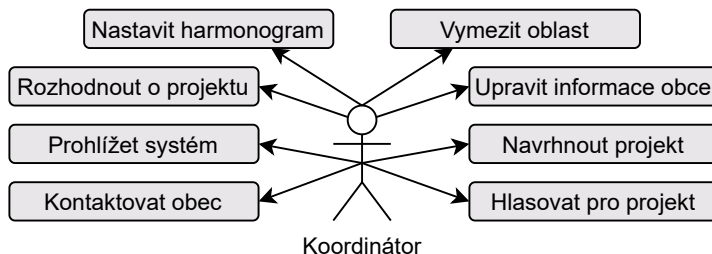


Obrázek 5.1: Diagram případu užití uživatele

¹Firestore datový model: <https://firebase.google.com/docs/firestore/data-model>

5.3.2 Koordinátor

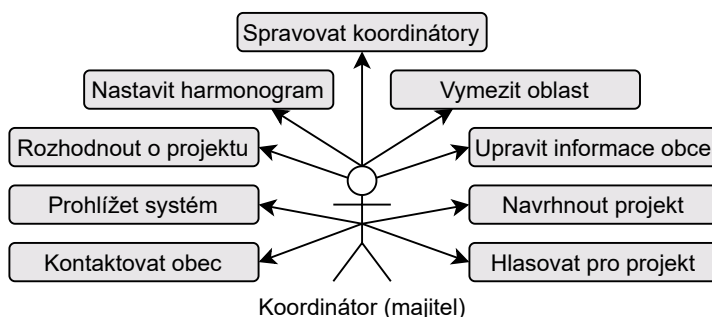
Uživatel systému s právy pro pohyb v interním systému participativního rozpočtu se jmenuje Koordinátor. Po přihlášení má navíc oproti klasickému uživateli možnosti správy systému viz obrázek 5.2.



Obrázek 5.2: Diagram případu užití koordinátora

5.3.3 Koordinátor s oprávněním majitele

Posledním konkrétním subjektem je koordinátor s oprávněním majitele. Mimo oprávnění klasického uživatele a koordinátora může spravovat i jednotlivé koordinátory (obrázek 5.3). Správa koordinátorů obnáší vytváření nových účtů a blokace účtů existujících.



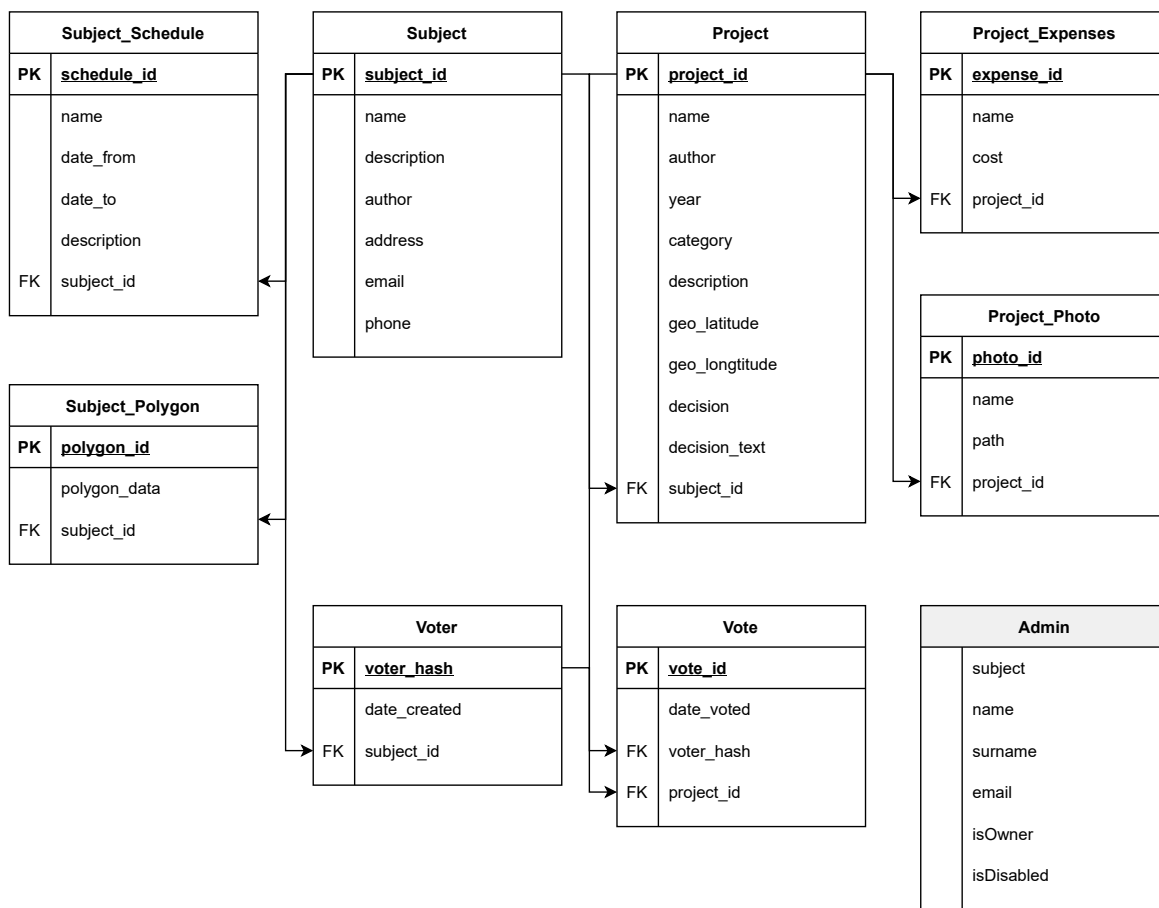
Obrázek 5.3: Diagram případu užití koordinátora (majitele)

5.4 Uchovávání a zpracování dat

U webových aplikací, běžící na serveru, lze využít několik způsobů ukládání informací. Systém pro participativní rozpočet využívá kombinaci databáze a ukládání do souborů. Databáze udržuje informace o obcích, projektech, hlasujících a přihlašovacích údajích zastupitelů, kdežto systém pro ukládání souborů udržuje grafický obsah důležitý pro logo obce či jednotlivé obrázky projektů.

5.4.1 Modely dat

Pro hlavní funkcionalitu aplikace byla vytvořena schéma relační databáze (obrázek 5.4), popisující jednotlivé tabulky. Tabulka *Admin* je specifická osamostatněním. Nemá jediný vztah a tudíž ani primární či cizí klíč. Důvodem je tvorba tabulky v jiné databázi, sloužící pro autorizaci koordinátorů či zastupitelů obce.



Obrázek 5.4: Schéma relační databáze obsahující jednotlivé tabulky

Tabulka Subject

Informace důležité pro jednotlivé obce. Zobecnění a použití jména *Subject* je kvůli budoucímu možnému rozšíření mimo obce. Možní budoucí zájemci participativního rozpočtu jsou školy či organizace. Tabulka obsahuje unikátní ID subjektu, název, krátký popis, jméno koordinátora (author) a kontaktní údaje jako jsou:

- address (adresa),
- email (e-mail),
- phone (telefonní číslo).

Koordinátor subjektu může navrhnout několik různých časových úseků participace. Tvoří harmonogram („Subject_Schedule“), obsahující datum začátku a konce a název časového úseku. Pro vymezení prostoru působnosti obce je nutno vytvořit polygon na mapě („Subject_Polygon“), ve kterém lze zaznamenat souřadnice projektu.

Tabulka Project

Projekty jsou vytvářeny pro jednotlivé subjekty. Ve formuláři pro návrh lze vyplnit náklady, proto vztah s tabulkou „Project_Expenses“. Zároveň lze přidat jednu fotku („Pro-

ject_Photo“). Každý projekt při vytvoření očekává rozhodnutí obce (doména „decision“ a „decision_text“). Jedná se o rozhodnutí proveditelnosti, tedy zda je vše v rámci pravidel. Rozhodnutí o proveditelnosti se provádí před samotným hlasováním.

Tabulka Vote

Každý zaznamenaný hlas je uložen v tabulce „Vote“. Jedná se o tabulku propojenou pomocí cizího klíče s tabulkou hlasujícího („Voter“) a projektu („Project“). Pro účely nahlížení do historie participativních rozpočtů a ochraně násobného hlasování, mimo jiné, každý hlas získá časovou značku zaznamenání hlasu. Hlasující může v jednom roce uplatnit maximálně 3 hlasy.

Tabulka Admin

Osamocená tabulka vytvořena pro uchovávání přihlašovacích údajů koordinátora/zastupitelů do interního systému participativního rozpočtu. Uchovává se:

- **subject:** přihlášení koordinátora lze provést jen pro obec, ve které působí.
- **name, surname:** identifikace koordinátora, po přihlášení je v systému viditelná kombinace subjektu a celého jména.
- **email:** důležitý údaj propojující uživatele s ověřováním přihlašování třetí stranou (Firebase Authentication²).
- **isOwner:** dvoustavová hodnota určující, zda koordinátor obce má plné práva ke změnám v systému a správě uživatelů.
- **isDisabled:** dvoustavová hodnota určující, zda je účet koordinátora aktivní. V momentu, kdy nabývá hodnoty `true` uživatel nemá možnost přihlásit se do interního systému. Zároveň není možné zablokovat koordinátora s aktivní vlajkou „isOwner“.

Důvod izolace tabulky je uložení informací na jiném databázovém prostředí. Pro autentizaci a ukládání informací o koordinátorech do interního systému je použita služba třetí strany Firebase. Disponuje nástroji jako je autentizace včetně knihoven pro propojení s aplikací, databázového či cloudového uložení, hostingu a dalších nepřeberných množství nástrojů³, nevyužitých v tomto projektu.

5.5 Uživatelské rozhraní

V informačním systému je uživatelské rozhraní rozděleno na šest veřejných částí a několik součástí interního systému. Návrhy obrazovek, neuvedené ve veřejné nebo interní části níže, jsou k nalezení v příloze A.

²Dokumentace služby Firebase: <https://firebase.google.com/docs>

³Produkty služby Firebase: <https://firebase.google.com/products-build>

5.5.1 Veřejná část

Viditelná část pro občany a všechny uživatele, kteří chtějí navrhovat nebo hlasovat v informačním systému, obsahuje tyto obrazovky:

- **Domovská část:** část rozhraní, do kterého se dostane kdokoliiv přistupující k obecnímu participativnímu rozpočtu. Uživatel ihned získá stručný popis procesu obce a díky ukazateli průběhu je obeznámen o aktivní části harmonogramu. Dynamické tlačítko přeměruje uživatele do sekce dle harmonogramu. V momentě, kdy lze projekty navrhovat, je kliknutím na vytvářecí tlačítko uživatel přeměrován do navrhovacího formuláře.
- **Navrhovací část:** uživatelské rozhraní je rozděleno do dvou bloků (obrázek 5.5). Prvním blokem je formulář obsahující jednotlivé pole pro vstupy uživatele. Druhým blokem je mapa, do které lze zanést bod a určit tak souřadnice navrhovaného projektu.
- **Hlasovací část:** matice projektů s náhledovým obrázkem a možností nahlédnutí na detail jednotlivých projektů. V horní části stránky lze projekty filtrovat dle kategorie, názvu nebo stavu rozhodnutí zastupitelstvem.
- **Výsledková část:** seznam projektů v čistě textové podobě, kdy u každého projektu lze vyčíst počet hlasů. Stejně jako u hlasovací části občan může zobrazit detail projektu nebo filtrovat vyobrazené projekty.
- **Část s harmonogramem:** stránka čistě informativní, bez možnosti interakce. Rozdělení harmonogramu do jednotlivých bodů, u každého bodu uživatel najde datum začátku a konce bode, název a detailní popis.
- **Část věnována historii:** archiv projektů z minulých let. Identická stránka jako část výsledková.
- **Kontaktní část:** obsahuje kontaktní informace obce včetně adresy a sociálních sítí. Zároveň disponuje formulářem pro odeslání zprávy obci.

5.5.2 Interní část

Mimo možností viditelných v horním menu lze vstoupit i do interního systému. Pro větší bezpečnost není možné „prokliknout se“ do části nepřístupné prostým uživatelům. Jedinou možností, jak se do interního systému dostat, je upravit URL a dostat se tak k přihlášení. Seznam dostupných obrazovek pro přihlášené zastupitele obsahuje:

- **Rozhodovací část:** seznam projektů s možností zobrazení detailu. Zastupitel může navrhovaný projekt akceptovat nebo zamítnout, s možností odůvodnit svoje rozhodnutí.
- **Část pro nastavení harmonogramu:** seznam harmonogramů obce. Harmonogram lze upravit nebo smazat, pokud nepatří do povinné trojice (navrhování, hlasování nebo výsledky).
- **Část pro vymezení oblasti působnosti:** mapa s možností vymezení polygonu neboli vymezení oblasti. Zastupitel má možnost polygon bodů vymazat a zaznačit seznam bodů nový.

Název

Autor Email Unikátní ID

Kategorie Fotky

Nadpis

Popis

MAPA

ODESLAT ZAHODIT

Obrázek 5.5: Návrh navrhovací stránky informačního systému

- **Část pro nastavení informačního systému:** stránka rozdělená na dva celky. V prvním celku je možné nahrát nové logo obce, viditelné v navigačním menu. V druhém celku lze změnit kontaktní informace nebo přidat odkaz na sociální síť.
- **Uživatelská část:** koordinátor obce nemusí pracovat v informačním systému sám. Pomocí této části lze přidat další zastupitele nebo neaktivní zastupitele zablokovat.

Kapitola 6

Implementace aplikace

V této kapitole jsou postupně popsány procesy tvorby jednotlivých částí informačního systému. Počáteční sekce se věnuje výběru technologií. Poté je dle třívrstvé architektury popsána prezentační vrstva, dále pak nazývána *frontend*. První vrstva navazuje na aplikační vrstvu, později označována termínem *backend*. V backend části je popsána správa požadavků včetně komunikace s databází. Nakonec je rozepsána datová vrstva, typy databází využitých v informačním systému a jejich rozdíly.

6.1 Použité technologie

Pro efektivní tvorbu webových aplikací se často využívá frameworků (zmíněných v kapitole 3.4.1). Vybráním frameworku zaručí vývojáři dodržení předem definovaných konvencí, předcházením opakujícím se problémům, a tím v závěru ušetřením spousty času. Zároveň je zaručena vyšší kvalita projektu a jednodušší rozšířitelnost do budoucna. Pro frontend i backend část byl vybrán programovací jazyk JavaScript [34] s jeho standardem ECMAScript vydávaný neziskovou organizací Ecma international¹. V aplikaci byl využit standard ECMA-262². Jazyk zároveň nabízí spoustu frameworků pro použití na frontend části.

Dle vlastních preferencí a rozšířenosti předdefinovaných modulů a komponent³ jsem se rozhodl pro Angular⁴. Zároveň je použit jazyk TypeScript⁵ (viz obrázek 6.1). Jedná se o nástavbu nad jazykem JavaScript, která jej rozšiřuje o statické typování a další atributy, které známe z objektově orientovaného programování. Samotný kód psaný v TypeScriptu se kompiluje do JavaScript kódu.

Pro backend část kódu byla využita čistá forma jazyka JavaScript. Důvod je jednoduchost samotné mezivrstvy mezi klientskou a datovou vrstvou. Pro škálovatelnou internetovou komunikaci s možností obsloužit mnoho uživatelů najednou bylo využito rozhraní Node.js⁶. Primární účel prostředí je tvorba serverové části webových aplikací. Rozhraní samotné obsahuje vestavěný modul *http*⁷, s nímž lze spouštět webový server. Pro jednodušší

¹Odkaz na Ecma international: <https://www.ecma-international.org/>

²ECMA-262 specifikace: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

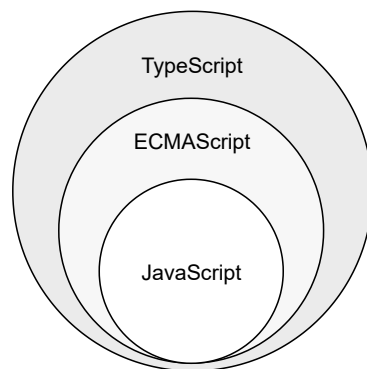
³Dostupné komponenty pro framework Angular: <https://material.angular.io/>

⁴Dokumentace Angular: <https://angular.io/docs>

⁵Dokumentace TypeScript: <https://www.typescriptlang.org/docs/>

⁶Runtime prostředí Node.js: <https://nodejs.org/>

⁷Node.js modul http: <https://nodejs.org/dist/latest-v8.x/docs/api/http.html>



Obrázek 6.1: Nástavby jazyka JavaScript

práci s kódem ale byly využity frameworky rozhraní Node.js, usnadňující práci s požadavky. Výpis důležitých frameworků použitých v systému:

- **express**⁸: rychlý a minimalistický webový framework,
- **node-postgres**⁹: kolekce modulů pro propojení s databází PostgreSQL,
- **crypto**¹⁰: Vestavěný modul Node.js obsahující kolekci kryptografických funkcí.

Databázová vrstva obsahuje dva typy databází. Pro uchovávání dat o obcích, projektech a hlasech byla použita relační databáze PostgreSQL¹¹. Pro výběr relační databáze jsem se rozhodl kvůli silné provázanosti jednotlivých entit mezi sebou. Druhý typ databáze pro uchovávání uživatelů interního systému je využita databáze služby Firebase, tedy Firestore databáze¹².

6.1.1 Další použité nástroje

Mimo výběr jazyků byly použity i jiné nástroje. Pro správu verzí kódu byl použit git, respektive webová služba GitHub¹³ podporující vývoj softwaru. Jako vývojové prostředí vhodné pro vývoj webových aplikací byl zvolen editor Visual Studio Code¹⁴. V poslední řadě je třeba zmínit multiplatformní aplikaci Postman¹⁵. Aplikace zvládá velkou škálu věcí od testování (dále rozepsáno v kapitole 7), přes interakci s API.

6.2 Frontend

Frontend část aplikace tvoří komponenty, v nichž se definuje vzhled a logika na straně klienta. Tvorba jednotlivých komponent je prováděna pomocí Angular CLI (*Command Line Interface*) [30]. CLI framework disponuje velkým počtem příkazů pro spouštění, sestavení, testování nebo tvorbu aplikace. Názorný příklad vytvoření nové komponenty:

⁸Framework express: <https://expressjs.com/>

⁹Modul node-postgres: <https://node-postgres.com/>

¹⁰Modul crypto: <https://nodejs.org/api/crypto.html>

¹¹Dokumentace PostgreSQL: <https://www.postgresql.org/docs/>

¹²Dokumentace Firestore: <https://firebase.google.com/docs/firestore>

¹³GitHub verzování: <https://github.com/>

¹⁴Editor Visual Studio Code: <https://code.visualstudio.com/>

¹⁵Aplikace Postman: <https://www.postman.com/>

```
$ ng generate component <název-komponenty>
```

Vytvořená komponenta s názvem *dialog* obsahuje čtyři soubory ve složce se jménem identickým jako je název komponenty. Struktura složky vypadá následovně:

```
dialog
├── dialog.component.html
├── dialog.component.ts
├── dialog.component.css
└── dialog.component.spec.ts
```

Dosud nezmiňovaný soubor s koncovkou *.spec.ts* slouží pro testování kódu. Využit je nejčastěji k unit testům (viz kapitola 3.5). Vytvářet komponenty lze i bez pomoci, je však třeba dbát na správnou konvenci. Níže je vyobrazena anotace Angular komponenty, která napomáhá k propojení *html*, *ts* a *css* souborů.

```
1 @Component({
2   selector: 'app-dialog',
3   templateUrl: './dialog.component.html',
4   styleUrls: ['./dialog.component.css']
5 })
6 export class DialogComponent {
7   ...
8 }
```

Výpis 6.1: Ukázka anotace nové komponenty

Anotace *@Component* obsahuje objekt s několika vlastnostmi:

- **selector**: přidává nový HTML tag, který je možný použít skrze projekt.
- **templateUrl**: relativní cesta k souboru HTML komponenty.
- **styleUrls**: seznam relativních cest ke stylujícím souborům.

6.2.1 Komunikace s Firebase službou

Komunikace s Firebase službou probíhá napřímo z frontend části a je navázaná pomocí importování modulu *AngularFireModule*¹⁶ s následnou inicializací aplikace, jak lze vidět na exemplárním kódu 6.2 níže, přesněji řádek 14.

```
1 export const environment = {
2   firebaseConfig: {
3     apiKey: 'secret-key',
4     authDomain: 'participatory-budgeting-295014.firebaseio.com',
5     databaseURL: 'https://participatory-budgeting-295014.firebaseio.com',
6     projectId: 'participatory-budgeting-295014',
7     storageBucket: 'participatory-budgeting-295014.appspot.com',
8     messagingSenderId: '216903213229',
9     appId: '1:216903213229:web:834d95392cc79f5aba6624',
10    measurementId: 'G-63SELFMJLM',
11  },
12 };
13 @NgModule({
14   imports: [ AngularFireModule.initializeApp(environment.firebaseConfig) ]
15 })
```

Výpis 6.2: Ukázka nastavení Firebase služby pro framework Angular

¹⁶ Firebase modul: <https://www.npmjs.com/package/@angular/fire>

Následně postačí v komponentě využívající Firebase knihovnu injektovat jednu z mnoha služeb, které knihovna nabízí. V projektu je využíván *AngularFireAuth* pro registraci a přihlašování uživatelů nebo *AngularFirestore* pro práci s Firebase databází.

Proces hlasování

Hlasování pro projekt probíhá na frontend části ve dvou fázích. První fáze spočívá v otevření dialogu (viz obrázek 6.2) pro zadání telefonního čísla. Správné zadání telefonního čísla hlídají validátory speciálně upraveny pro evropskou normu čísel. Telefonní číslo lze zadat ve formátu bez předvolby, v takovém případě bude číslu přidána automaticky předvolba česká.

Druhou fází je potvrzení telefonního čísla. Zde se může objevit upozornění hlasujícího na vyčerpaný počet hlasů. V našem případě se jedná o 3 hlasy na jedno telefonní číslo ročně. Pokud číslo dosud neprovedlo tři hlasování, změní se rozložení dialogu a očekává se zaslání kódu službou Firebase. Odeslání správného kódu spustí procesy pro uložení hlasu na serverové a databázové vrstvě a upozorní hlasujícího na úspěšné nebo neúspěšné dokončení.

Je potřeba zadat telefonní číslo

Telefonní číslo

Ověřit

* Pokud nezadáte předvolbu, automaticky bude přidána česká předvolba +420

Odesláním hlasu souhlasíte se [zpracováním osobních údajů](#)

Je potřeba zadat ověřovací kód z telefonu

Ověrovací kód

Odeslat

Obrázek 6.2: Ukázka hlasovacího dialogu

6.3 Backend

Backend část aplikace tvoří naslouchající webový server na předem definovaném portu uložném v souboru *.env*, ve kterém je definována hodnota proměnné *DATABASE_PORT*. Jelikož je informační systém soubor aplikací běžící na různých portech, server musí zajistit sdílení zdrojů webové stránky na jiné doméně. Pro tyto účely slouží mechanismus CORS (Cross-origin resource sharing) [23]. Framework Express disponuje podporou přidávání tzv. middleware služeb (pojem detailněji popsán v kapitole 6.3.1). Jedním z middleware modulů je i *cors*¹⁷ používaný v projektu.

¹⁷Dokumentace modulu cors: <https://expressjs.com/en/resources/middleware/cors.html>

6.3.1 Komunikace frontend-backend

Směrování určuje způsob, jakým aplikace reaguje na požadavek klienta na konkrétní koncový bod. Každá trasa má obslužnou rutinu, která se provede při shodě trasy.

```
1 app.post(  
2   "/api/vote-projects",  
3   voteProjectController.getAllVoteProjects  
4 );
```

Výpis 6.3: Ukázka metody očekávající požadavek na dané trase

Odpověď obsluhy zpátky odesílateli obsahuje HTTP status kód [28], uvádějící, zda byl konkrétní požadavek HTTP úspěšně dokončen. Nejčastějšími skupinami kódů v projektu jsou:

- **2xx úspěch:** třída stavových kódů, zasílána v každém případě, kdy je požadavek úspěšně zpracován, většina kódů je zasílána s hodnotou úspěchu nebo daty z databáze,
- **4xx neúspěch na straně klienta:** tato třída stavového kódu je určena pro autorizační problémy v aplikaci, příkladem je nevalidní JWT token zaslaný klientem,
- **5xx neúspěch na straně serveru:** kódy stavu odpovědi označují nemožnost zpracovat požadavek, v aplikaci se objevuje při zachycení jakékoliv chyby, která je automaticky zasílána s kódem.

Mimo statusových kódů může obsluha vracet odesílateli data. Pokud se jedná o požadavek získání informací z databáze, struktura vrácených dat kopíruje jednotlivé záznamy databázových entit. Více o vytváření SQL dotazů v kapitole 6.3.2.

Autorizace chráněných požadavku

Autorizace zabezpečených tras funguje na základě přidání middlewaru. Jsou to funkce, které mají přístup k objektu požadavku (*req*), objektu odpovědi (*res*) a *next* funkci v cyklu žádosti a odpovědi aplikace [10]. Cílem middlewaru autorizace je ověřit JWT token zaslaný v autorizační hlavičce HTTP požadavku a povolit či zakázat vykonání samotné obslužné rutiny.

```
1 verifyJWTToken: async (req, res, next) => {  
2   if (!req.headers.authorization) {  
3     res.status(401).send("Unauthorized");  
4   }  
5  
6   const token = req.headers.authorization.split(" ")[1];  
7  
8   try {  
9     await admin  
10      .auth()  
11      .verifyIdToken(token)  
12      .then(() => {  
13        next();  
14      });  
15   } catch (err) {  
16     return res.status(401).send("Unauthorized");  
17   }  
18 }
```

Výpis 6.4: Ukázka zabudování middleware metody pro verifikaci JWT tokenu

6.3.2 Komunikace backend-databáze

Většina požadavků klienta očekává od serveru informace z databáze. Pokud se jedná o relační databázi uchovávající informace o subjektech, projektech nebo hlasech, aplikace využívá modul *node-postgres* (dříve zmiňovaný v kapitole 6.1). Navazování komunikace s databází probíhá formou vytvoření nové instance *Pool* obsahující všechny potřebné údaje k navázání spojení.

```
1 const Pool = require("pg").Pool;
2
3 pool = new Pool({
4   user: process.env.DATABASE_USER,
5   password: process.env.DATABASE_PASSWORD,
6   host: process.env.DATABASE_URL,
7   port: process.env.DATABASE_PORT,
8   database: process.env.DATABASE_NAME,
9   ssl: false,
10  });
```

Výpis 6.5: Ukázka navázání komunikace s databází

Po úspěšném navázání připojení jsou prováděny jednotlivé dotazy nad databází. Výsledek dotazu je chybová hláška v případě neúspěchu nebo porušení interních pravidel databáze. V opačném případě dotaz disponuje výsledkem obsahující mimo jiné počet záznamů a jejich výčet.

```
1 const userExists = await pool.query(
2   "SELECT subject_id FROM Voter WHERE voter_hash = $1",
3   [phoneHash]
4 );
```

Výpis 6.6: Ukázka provedení databázového dotazu

6.4 Databáze

Databázová vrstva informačního systému je rozdělena na dvě části. První část je vytvořena v relační databázi. Uchovává veškerá data mimo uživatele. Instance uživatelů jsou ukládány v dokumentové databázi služby Firebase. Důvodem ukládání uživatelů zvláště je lepší provázanost s knihovnou Firebase použitou na frontendové části (zmiňováno v kapitole 6.2.1).

6.4.1 Relační databáze

K vytvoření struktury databáze v testovacím či reálném prostředí byl sepsán SQL skript (*schema.sql*), který spuštěním promaže a znovu vytvoří všechny existující tabulky informačního systému. Zároveň inicializuje nové generátory identifikátorů jednotlivých entit a vytvoří vazby mezi tabulkami. Pro testovací účely byl následně vytvořen druhý skript se jménem *seed.sql*. Skript přidá umělé data do předem vytvořené struktury databáze. Data se vážou na subjekt se jménem Svinov, se kterým byl celý informační systém konzultován a testován.

V průběhu implementace systému zároveň proběhlo ke změně schématu databáze. Bylo potřeba definovat unikátní název subjektu (tabulka *Subject*), tudíž v databázi přibylo nové integritní omezení určující jednoznačnost jména a unikátního čísla subjektu (viz výpis 6.7, přesněji řádek 6).

```

1 CREATE TABLE Subject(
2     subject_id SERIAL PRIMARY KEY,
3     subject_name VARCHAR(50) NOT NULL,
4     ...
5     UNIQUE (schedule_name, subject_id)
6 );

```

Výpis 6.7: Ukázka navázání komunikace s databází

Jméno subjektu, respektive obce, je důležitý prvek. Použitý je v URL informačního systému a dvojice identických jmen by zapříčinila kolizi a nejasnost reference na data subjektu. Zároveň si lze povšimnout názvu *Subject*. Účelně nebyla použita anglická verze slova obec (municipality). Informační systém tímto krokem podporuje možné rozšíření aplikace nejen pro obce. Příklad subjektů, které by mohly mít zájem o participativní rozpočet:

- neziskové organizace,
- školy,
- společnosti s přebytkovými finančními prostředky.

6.4.2 Dokumentová databáze

Firebase databáze je založena na tzv. dokumentech. Struktura zároveň velmi připomíná JSON typ zápisu dat. Rozdíly jsou v extra datových typech, které JSON neobsahuje a omezení velikosti jedné hodnoty na maximum 1MB. Data jsou organizována jednoduchým systémem složek/souborů. Každý soubor je přístupný prostřednictvím odkazu. Informační systém využívá databázi pro ukládání jediné entity. Zastupitelé mají možnost se přihlásit do informačního systému a mimo uloženého přihlašovacího jména a hesla je nutno ukládat dodatečné informace jako je reference na subjekt, ke kterému se účet váže, informaci, zda je účet blokováno, uživatelskou roli, jméno uživatele a e-mail.

Kompletní strukturu jednotlivých dokumentů lze reprezentovat pomocí objektů. Níže je ukázka příkladu uživatele subjektu **Svinov**:

```

{
  "email": "testovaci@uzivatel.cz",
  "isDisabled": true,
  "isOwner": false,
  "name": "Uživatel",
  "subject": "svinov",
  "surname": "Testovací"
}

```

Databáze zároveň definuje vlastní pravidla pro manipulaci s daty, respektive zápis a čtení. V návaznosti na bezpečnost v kapitole 6.3.1 byla implementována pravidla, dovolující přístup k datům pouze přihlášeným uživatelům. Frontend část volá metody pro manipulaci s daty (viz kapitola 6.2.1) v dokumentové databázi. Jakýkoliv dotaz musí nést informace o přihlášeném uživateli, v opačném případě pravidla databáze zamezí přístupu.

Kapitola 7

Testování

Poslední kapitola pojednává o testování informačního systému. Na základě popsaných technik v kapitole 3.5. Již před dokončením implementační fáze byl systém průběžně testován pomocí unit testů pro ověření validity všech částí. Po vytvoření prvního funkčního prototypu byla navázána komunikace s vybranou obcí Ostrava-Svinov, jednotlivými občany obce a provedeny uživatelské testy. V závěru kapitoly je shrnut proces testování.

7.1 Testování při vývoji

Přirozeným procesem při vytváření systému je testování jednotlivých částí kódu. Každá změna v implementaci může znamenat potenciální riziko stability systému. Pomocí jednotkových testů zaměřených na vytvoření komponent (zmiňovaných v kapitole 6.2) byly chyby snadno detekovány a iterativně opraveny.

Druhým důležitým krokem bylo testování bezpečnosti. V rámci kapitoly 6.3.1 byly důkladně prověřeny všechny interní požadavky, nepřístupné bez autorizačních údajů, respektive přihlášení do interního systému. Nástroj Postman vypsáný v dalších použitých nástrojích 6.1.1 dovoluje jednoduše simulovat HTTP požadavky. Případy bez autorizační hlavičky v momentě, kdy jsou nutné, byly zamítnuty a testování považováno za úspěšné.

7.2 Uživatelské testování

Po vytvoření prvního fungujícího prototypu přichází na řadu testování uživatelské. Informační systém používají dva typy uživatelů. Zástupci obce dostali za úkol splnit navržené scénáře a zhodnotit. Následně byl systém testován občany.

7.2.1 Zastupitelstvo obce

Prvním cílem testování bylo zastupitelstvo obce v zastoupení 1. místopředsedy městské části Ostrava-Svinov. Vytvořený scénář s typickými úkony v interním systému byl předán k vyzkoušení a zhodnocení. Bohužel aktuální situace vyvolala ve spoustě lidí opatrnost a v případě testování informačního systému nedovolovala osobní kontakt s testující osobou. Situace byla vyřešena formou e-mailové a telefonní komunikace. Zasláný scénář obsahoval tyto body:

1. rozhodnout o proveditelnosti projektu,
2. nastavit časový harmonogram systému (změnit údaje/přidat nový harmonogram),
3. zkusit zadat novou oblast působnosti obce,
4. v nastavení systému přepsat jednu položku nebo změnit logo,
5. přidat nového uživatele nebo existujícího uživatele zablokovat.

Po následné komunikaci s místopředsedou obce nebyly nalezeny větší nedostatky, týkající se bodů scénáře. Několik připomínek však vzneseno bylo. Seznam je sepsán níže:

1. nedostatečně řešené obecné nařízení o ochraně osobních údajů, zkráceně ONOOU (*anglicky GDPR, General Data Protection Regulation*) [9],
2. nemožnost editace projektu zastupitelstvem,
3. absence definice rolí při přidání nového zastupitele,
4. absence generátoru hesel při vytváření uživatele.

Většina výše definovaných připomínek představovala nové funkce systému. Veškeré připomínky byly v krátkém časovém úseku úspěšně implementovány a systém lze z pohledu zastupitelstva brát za finální.

7.2.2 Vybraní občané obce Svinov

Druhým typem uživatele informačního systému jsou občané. Jejich úkolem v systému jsou hlavně body předložené formou testovacího scénáře níže:

1. vytvořit nový projekt,
2. hlasovat pro existující projekt.

Zpětná vazba byla z pohledu funkcionality ve všech případech kladná. Připomínky na systém vznikly pouze k uživatelskému prožitku. Umístění tlačítek v prvním prototypu nebyla intuitivní a většina testujících žádala o přemístění. Jednalo se o obrazovku pro vytvoření projektu, potažmo dialog potvrzující odeslání projektu. Iterativním procesem byla provedena změna rozmístění tlačítek a systém otestován znovu. Druhým připomínaným problémem bylo časté zapomenutí přidání nákladu v procesu vytváření projektu. Jednodušší projekty se mohou skládat z jediného nákladu, systém však náklad „nepropíše“ do databáze, pokud nebyl přidán do seznamu pomocí kliknutí na tlačítko se symbolem „+“. Po diskuzi nad lepším řešením s občany, kteří testovali systém, bylo nalezeno řešení. Tlačítko se symbolem „+“ bylo zaměněno za jasnější tlačítko s textem **přidat**.

7.2.3 Závěr testování

Všechny předložené scénáře splnily svůj účel, systém byl odladěn a nedostatky připomenuty ze strany zastupitelstva i občanů doladěny. Zároveň se podařilo systém rozšířit o několik funkcionalit (viz kapitola 7.2.1) díky aktivnímu přístupu testujících. Výsledný informační systém lze vyzkoušet na adrese <https://participatory.herokuapp.com/>.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo vytvořit informační systém pro participativní rozpočet. Práce byla zaměřena na analyzování požadavků občanů i zastupitelů městské části Ostrava-Svinov a následným vytvořením aplikace. Zároveň byl kladen důraz na bezpečnost a široké spektrum možností pro individuální nastavení jednotlivých obcí.

Práci jsem zahájil teoretickou přípravou. Důkladně jsem si nastudoval obecnou problematiku participativních rozpočtů a historii jeho využití ve světě i v České republice. Následně probíhaly konzultace se zastupitelstvem obce a byl vytvořen dotazník, jehož cílem bylo zjištění požadavků občanů pro obsah informačního systému. Na základě získaných podnětů byl zpracován návrh a následná implementace prvního prototypu informačního systému. Společně s implementací jsem prováděl funkční testování kódu a veškeré nedostatky opravil. Poté byl první prototyp aplikace důkladně uživatelsky otestován všemi zainteresovanými stranami. Po funkční stránce prototyp neměl větších nedostatků, zastupitelstvo si však přálo rozšíření interního systému o editaci projektů, splnění povinnosti o ochraně osobních údajů správcem informačního systému a přidání generátoru náhodných hesel při vytváření nového účtu zastupitele. Požadavkům jsem vyhověl a v další verzi prototypu přidal. Zároveň jsem tímto otestoval jednoduchou rozšiřitelnost celé aplikace.

V souvislosti se zpracováním bakalářské práce jsem nebyl nové znalosti týkající se problematiky participativního rozpočtování. Věřím, že jsem díky svému projektu rozšířil možnost obcí v České republice zapojit své občany do programu rozpočtování. Finální vývoj aplikace není u konce. V budoucnu bych rád rozšířil kompetence občana o možnost jeho přihlášení. V účtu občana by se nacházel přehled jeho navržených projektů a historie jeho hlasování. Zároveň by občan mohl nabýt možnosti editovat vlastní projekt nebo diskutovat nad projektem cizím. Mým cílem je oslovit obce na území České republiky a dostat aplikaci do ostrého provozu.

Literatura

- [1] ANO 2011, ŽÍT BRNO S PODPOROU PIRÁTŮ, STRANA ZELENÝCH, TOP 09. *Koaliční smlouva pro zastupitelstvo města Brna v období 2016-2018* [online]. 2016 [cit. 2021-02-16]. Dostupné z: https://www.brno.cz/fileadmin/user_upload/sprava_mesta/magistrat_mesta_brna/KPMB/Koalicni_smlouva_20.6.2016.pdf.
- [2] BENJAMIN, G. Lessons from Latin American Experience in Participatory Budgeting. Leden 2007, [cit. 2020-12-10].
- [3] BRABEC, D. Participatory Budgeting in the Czech Republic between 2014 and 2018. *Annales UMCS, Politologia*. Prosinec 2019, sv. 26, s. 91. DOI: 10.17951/k.2019.26.2.91-101.
- [4] CHRISTENSSON, P. *Backend Definition* [online]. [cit. 2021-04-20]. Dostupné z: <https://techterms.com/definition/backend>.
- [5] COSTA, B., PIRES, P. F., DELICATO, F. C. a MERSON, P. Evaluating a Representational State Transfer (REST) Architecture: What is the Impact of REST in My Architecture? In: *2014 IEEE/IFIP Conference on Software Architecture*. 2014, s. 105–114. DOI: 10.1109/WICSA.2014.29.
- [6] DHADUK, H. *Best Frontend Frameworks of 2021 for Web Development* [online]. [cit. 2021-01-23]. Dostupné z: <https://www.simform.com/best-frontend-frameworks/>.
- [7] DIAS, N. *Hope for democracy, 25 years of participatory budgeting worldwide* [online]. 2014 [cit. 2020-12-11]. Dostupné z: http://www.in-loco.pt/upload_folder/edicoes/1279dd27-d1b1-40c9-ac77-c75f31f82ba2.pdf.
- [8] EDUCATION, I. C. *Three-Tier Architecture* [online]. [cit. 2021-01-23]. Dostupné z: <https://www.ibm.com/cloud/learn/three-tier-architecture#toc-other-mult-CmW8tx9X>.
- [9] EVROPSKÝ PARLAMENT A RADA. NAŘÍZENÍ EVROPSKÉHO PARLAMENTU A RADY (EU) 2016/679. [online]. [cit. 2021-04-26]. Dostupné z: <https://eur-lex.europa.eu/legal-content/CS/TXT/PDF/?uri=CELEX:32016R0679&from=EN>.
- [10] EXPRESSJS. *Using middleware* [online]. [cit. 2021-04-25]. Dostupné z: <https://expressjs.com/en/guide/using-middleware.html>.
- [11] GENRO, T. a DE SOUZA, U. *Presupuesto participativo* [online]. 1998 [cit. 2020-12-05]. Dostupné z: https://books.google.cz/books?hl=cs&lr=&id=w64_joMXSuAC&oi=fnd&pg=PA15&dq=

Propuesto+Participativo:+la+experiencia+de+Porto+Alegre&ots=-Yj7J-T9po&sig=C0y76z088njhUN4jucSvLdZ81Wg&redir_esc=y#v=onepage&q=Propuesto%20Participativo%3A%20la%20experiencia%20de%20Porto%20Alegre&f=false.

- [12] HANÁČEK, P. a STAUDEK, J. *Bezpečnost informačních systémů* [online]. 2000 [cit. 2021-01-14]. Dostupné z: http://media0.vesele.info/files/media0:50f8645ae2040.pdf.upl/uvis_bezpecnost_20000701.pdf.
- [13] ISO. *Technical report ISO/TR 16982* [online]. 2002 [cit. 2021-01-23]. Dostupné z: <https://www.sis.se/api/document/preview/901955/>.
- [14] JANEČEK, K. *D21 – Janeček method* [online]. 2021 [cit. 2021-02-14]. Dostupné z: https://uploads-ssl.webflow.com/5b046d892298cf84d9a87d56/5ff43cf8b01c725e9df3c415_5.1.2021.pdf.
- [15] KOHOVOLIT.EU sdružení a FÓRUM 50% S PODPOROU NADAČNÍHO FONDU HYUNDAI. *RozpočetProVsechny.cz* [online]. [cit. 2020-12-04]. Dostupné z: <https://rozpocetprovsechny.cz/>.
- [16] KOL., V. T. a. *Průručka participativního rozpočtu* [online]. 2015 [cit. 2020-12-05]. Dostupné z: <http://alternativazdola.cz/wp-content/uploads/2012/04/prirucka-participativniho-rozpocetu.pdf>.
- [17] KRASNER, G. E. a POPE, S. T. *A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System* [online]. 1988 [cit. 2021-01-23]. Dostupné z: https://www.researchgate.net/profile/Stephen_Pope/publication/248825145_A_cookbook_for_using_the_model_-_view_controller_user_interface_paradigm_in_Smalltalk_-_80/links/5436c5f30cf2643ab9888926/A-cookbook-for-using-the-model-view-controller-user-interface-paradigm-in-Smalltalk-80.pdf.
- [18] LAKE, L. *What Is a Target Audience?* [online]. [cit. 2021-02-14]. Dostupné z: <https://www.thebalancesmb.com/what-is-a-target-audience-2295567>.
- [19] LI, L. a CHOU, W. Design and Describe REST API without Violating REST: A Petri Net Based Approach. In: *2011 IEEE International Conference on Web Services*. 2011, s. 515. DOI: 10.1109/ICWS.2011.54.
- [20] MAIDEN, N. User Requirements and System Requirements. *IEEE Software*. 2008, sv. 25, č. 2, s. 90–91. DOI: 10.1109/MS.2008.54.
- [21] MAREK, D. a KANTOR, T. *Příprava a řízení projektů strukturálních fondů Evropské unie*. Brno: Společnost pro odbornou literaturu - Barrister & Principal, 2009. ISBN 9788087029565.
- [22] MARQUETTI, A., SILVA, C. a CAMPBELL, A. Participatory Economic Democracy in Action: Participatory Budgeting in Porto Alegre, 1989-2004. *Review of Radical Political Economics*. Leden 2011, sv. 44, s. 62–81. DOI: 10.1177/0486613411418055.
- [23] MDN WEB DOCS. *Cross-Origin Resource Sharing (CORS)* [online]. [cit. 2021-04-24]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>.

- [24] NAPOLITANO, M. *The Brazilian Military Regime, 1964–1985*. Oxford University Press, duben 2018. DOI: 10.1093/acrefore/9780199366439.013.413. Dostupné z: <https://oxfordre.com/latinamericanhistory/view/10.1093/acrefore/9780199366439.001.0001/acrefore-9780199366439-e-413>.
- [25] NIELSEN, J. *Usability 101: Introduction to Usability* [online]. [cit. 2021-02-11]. Dostupné z: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [26] NIELSEN, J. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. ISBN 9780080520292.
- [27] PAUL, R. End-to-end integration testing. In: *Proceedings Second Asia-Pacific Conference on Quality Software*. 2001, s. 211–220. DOI: 10.1109/APAQS.2001.990022.
- [28] R. FIELDING, E. a J. RESCHKE, E. *RFC 7231 - Hypertext Transfer Protocol (HTTP) Status Code Registry* [online]. [cit. 2021-04-24]. Dostupné z: <https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>.
- [29] RUNESON, P. A survey of unit testing practices. *IEEE Software*. 2006, sv. 23, č. 4, s. 22–29. DOI: 10.1109/MS.2006.91.
- [30] TECHOPEDIA. *Command Line Interface (CLI)* [online]. [cit. 2021-04-23]. Dostupné z: <https://www.techopedia.com/definition/3337/command-line-interface-cli>.
- [31] UN HABITAT. *72 Frequently Asked Questions about Participatory Budgeting* [online]. UN-HABITAT, P.O. Box 30030, 00100 Nairobi, Kenya. UMP-LAC, García Moreno 751 entre Sucre y Bolívar, P.O. 17-01- 2505, Quito - Ecuador: UN-HABITAT, 2004 [cit. 2020-12-05]. Dostupné z: <https://unhabitat.org/sites/default/files/download-manager-files/72%20Frequently%20Asked%20Questions%20about%20Participatory%20Budgeting%20%28English%29.pdf>.
- [32] UXMATTERS. *Prototyping User Experience* [online]. [cit. 2021-04-19]. Dostupné z: <https://www.uxmatters.com/mt/archives/2019/01/prototyping-user-experience.php>.
- [33] VOJTIŠKOVÁ, K. *Když o rozpočtu rozhodují obyčejní občané* [online]. [cit. 2020-12-11]. Dostupné z: <http://www.socialwatch.cz/?p=899>.
- [34] W3TECHS.COM. *Usage statistics of JavaScript for websites* [online]. [cit. 2021-04-23]. Dostupné z: <https://w3techs.com/technologies/details/pl-js>.
- [35] WAMPLER, B. A Guide to Participatory Budgeting. *Brian Wampler*. Leden 2000.
- [36] WAMPLER, B. a AVRITZER, L. The spread of participatory democracy in Brazil: From radical democracy to good government. *Journal of Latin American Urban Studies*. Leden 2005, sv. 7, s. 37–52.
- [37] ČERNÝ, V. *Metodika tvorby participativního rozpočtu pro města v ČR* [online]. Petra Rezka 12, Praha 4, 140 00: Agora CE, o.p.s, 2016 [cit. 2020-12-05]. Dostupné z: https://www.participativni-rozpocet.cz/wp-content/uploads/2017/05/Methodika_Participativni_rozpocet_AGORA_CE.pdf.

Příloha A

Designové návrhy

Kategorie	Řazení	Název	

Obrázek A.1: Návrh rozložení hlasovací stránky informačního systému

Kategorie	Název

Obrázek A.2: Návrh rozložení výsledkové stránky informačního systému

Obrázek A.3: Návrh rozložení kontaktní stránky informačního systému

Obrázek A.4: Detailní návrh rozložení domovské stránky informačního systému