

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH A IMPLEMENTACE WEBOVÉ APLIKACE PRO SPRÁVU UDÁLOSTÍ

DESIGN AND IMPLEMENTATION OF A WEB APPLICATION FOR ENTERPRISE EVENT MANAGEMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Josef Sponar

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dydowicz, Ph.D.

BRNO 2021

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Josef Sponar**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **Ing. Petr Dydowicz, Ph.D.**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh a implementace webové aplikace pro správu událostí

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrh řešení, přínos práce
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Cílem je vytvořit informační systém, který bude sloužit k lepší správě událostí v rámci týmu do sta osob a zefektivní tak komunikaci mezi jednotlivci.

Základní literární prameny:

BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy. Podnik v informační společnosti. Praha: Grada, 2008. 283 s. ISBN 978-80-247-2279-5.

MOLNÁR, Z. Automatizované informační systémy. Praha: Strojní fakulta ČVUT, 2000. 126 s. ISBN 80-01-02269-2.

MOLNÁR, Z. Efektivnost informačních systémů. Praha: Grada Publishing, 2000. 142 s. ISBN 80-716-410-X.

PECINOVSKÝ, R. Myslíme objektivně v jazyku Java: kompletní učebnice pro začátečníky. Praha: Grada, 2009. 570 s. ISBN 978-80-247-2653-3.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. Brno: Computer Press, 2010.
501 s. ISBN 978-80-251-2878-7.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

Mgr. Veronika Novotná, Ph.D.
ředitel

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Bakalářská práce se zaměřuje na návrh, tvorbu a implementaci webové aplikace pro správu událostí v zapsaném spolku Křesťané Kuřim. Zpracovává teoretická východiska, analýzu společnosti a poznatky z těchto analýz a teoretické části později shrnuje v praktické části práce.

Klíčová slova

Webová aplikace, C#, .NET, správa událostí, implementace, 7S analýza, Datový model

Abstract

Bachelors' thesis focuses on design and implementation of web application for enterprise event management in Křesťané Kuřim z. s. Thesis covers theoretical basis, analysis of the organization. Practical part of thesis is then outcome of the analysis and the theoretical basis.

Keywords

Web application, C#, .NET, management of events, implementation, 7S analysis, data models

Bibliografická citace dle ČSN

SPONAR, Josef. *Návrh a implementace webové aplikace pro správu událostí*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/135309>.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Petr Dydowicz.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 16. května 2021

.....

Podpis autora

Poděkování

Děkuji především trojjedinému Bohu, který mě posiloval při práci. Dále bych chtěl poděkovat také svému vedoucímu práce Ing. Petru Dydowiczovi, PhD. za ochotu vést moji práci a za motivaci, kterou jsem měl díky termínům, které mi dával. Také bych chtěl moc poděkovat Joepovi Heinovi a Andrejovi Henry za cenné připomínky z jejich praxe a rady, které mi v průběhu mé práce dávali. V neposlední řadě bych chtěl poděkovat své rodině a přátelům, kteří mě podporovali v průběhu celého mého studia.

Obsah

Úvod	11
Vymezení problému a cíle práce	12
1 Teoretická východiska práce	13
1.1 Organizace	13
1.1.1 Organizační struktura	13
1.2 Management	14
1.2.1 Základní pojmy	15
1.2.2 Procesní a projektový management.....	16
1.3 Nástroje projektového managementu	16
1.3.1 Identifikační listina.....	17
1.3.2 Work Breakdown Structure.....	17
1.3.3 Ganttovy diagramy	17
1.3.4 RIPRAN analýza	18
1.3.5 Analýza zainteresovaných stran	20
1.4 Analýzy použité pro popis současného stavu	21
1.4.1 McKinseyův rámec „7S“	21
1.4.2 Moscow analýza.....	22
1.5 Informační systém	23
1.5.1 Data, informace, znalosti, moudrost.....	23
1.5.2 Systém, informační systém	24
1.5.3 Objektové datové modelování.....	24
1.6 Objektově orientované programování	26
1.6.1 Zapouzdření.....	26
1.6.2 Dědičnost.....	26
1.6.3 Polymorfismus	26

1.7	Webová aplikace	27
1.8	Programovací jazyk C#	27
2	Analýza problému a současné situace	28
2.1	Popis zapsaného spolku.....	28
2.2	Organizační struktura	29
2.3	7S Analýza	30
2.3.1	STRUCTURE: vhodně nastavená hierarchie, kontrolní mechanismy, sdílení informací	30
2.3.2	SYSTEMS: všechny prvky ICT řešení.....	31
2.3.3	STYLE: styl řízení společnosti, přístup k riziku.....	32
2.3.4	STAFF: hodnocení motivovanosti a schopnosti kooperace pracovníků ..	32
2.3.5	SKILLS: hodnocení dovedností pracovníků a jejich aplikace v praxi	33
2.3.6	SHARED VALUES: identifikace hodnot sdílených pracovníky	33
2.3.7	STRATEGY: jednání podle předchozích bodů	34
2.3.8	Závěr	34
2.4	Analýza stávajícího řešení.....	34
2.4.1	SWOT analýza stávající řešení	34
2.5	Popis hlavních podnikových procesů a jejich vývojové diagramy	35
2.5.1	Moscow analýza – požadavky zadavatele	39
2.6	Existující současná řešení	39
2.6.1	Rozhodovací tabulka ohledně informačních systémů	40
2.7	Shrnutí analýz.....	43
3	Vlastní návrh řešení, přínos práce.....	44
3.1	Technické řešení.....	44
3.1.1	Úvod.....	44
3.1.2	Frontend	45

3.1.3	API – end points (koncové body).....	47
3.1.4	Service layer.....	48
3.1.5	Entity Framework.....	48
3.1.6	Database.....	49
3.1.7	Datový model.....	50
3.2	Implementace.....	55
3.2.1	Testování a quality management.....	56
3.2.2	Zpracování strategie a zaškolení uživatelů.....	57
3.3	Plán na dokončení informačního systému.....	57
3.3.1	Identifikační listina.....	57
3.3.2	RIPRAN analýza rizik.....	59
3.3.3	Analýza zainteresovaných stran.....	61
3.3.4	Analýza vybavení.....	61
3.3.5	Work Breakdown Structure.....	62
3.3.6	Plán realizace projektu.....	63
3.3.7	Ekonomické zhodnocení.....	65
3.4	Přínosy.....	65
	Závěr.....	67
	Seznam použité literatury.....	68
	Seznam tabulek.....	72
	Seznam obrázků.....	73

Úvod

Každá společnost ke své funkci využívá nějakou formu informačního systému. Ať jde o slovně definované nakládání s papíry a zvyklosti ohledně emailů, nebo rozsáhlý ERP systém. Nový informační systém může firmě ve spoustě ohledech pomoci, pokud bude dobře navržen a bude splňovat jejich očekávání. Na druhé straně, pokud systém nebude přesně odpovídat požadavkům a skutečným potřebám organizace, nebude pro organizaci nápomocný a může dojít k velkým újmám. Nezáleží tedy až tak na kvalitě naprogramovaného nástroje pro řízení a správu firmy, jako spíše o porozumění společnosti a jejím potřebám, na které vhodně aplikujeme některou ze současných možností řešení, které společnost posunou v její efektivitě a využití zdrojů. Jak již název práce napovídá, rozpracuji možnost vlastní tvorby webové aplikace, která bude svou podstatou naplňovat funkci informačního systému.

Vymezení problému a cíle práce

Cílem práce je navrhnout a implementovat systém, který bude spravovat události, jednotlivé činnosti v rámci události a bude přiřazovat zodpovědnosti za jednotlivé činnosti způsobilým pracovníkům. Pro tvorbu informačního systému jsem si vybral zapsaný spolek Křesťané Kuřim, pro který budu systém navrhovat.

Dále navážu popisem teoretické části, ve které shrnu základní teoretická východiska pro celou práci, a to jak z technického hlediska, tak i z pohledu manažerského.

V kapitole věnované analýze současného stavu přiblížím současný stav společnosti. Osvětlím, jak vypadá současné řešení, které přestává být vyhovující, a uvedu hlavní procesy, které budou později zahrnuty do informačního systému. Ukážu na možná alternativní řešení, která již dnes pro křesťanské organizace existují. Porovnáám jejich vhodnost pro danou organizaci s ohledem na požadavky a cenu.

Poslední a hlavní částí je samotný návrh aplikace, ve kterém popíšu jeden z moderních modelů architektury aplikací a jejich hlavní výhody oproti dřívějším architekturám aplikací. Zpracuji plán a potřebnou projektovou dokumentaci pro tvorbu tohoto systému, připravím rozpočet a zhodnotím přínosy práce.

1 Teoretická východiska práce

V této části shrnu teoretická východiska práce, která jsou nezbytně nutná pro zahrnutí do bakalářské práce, včetně stručného zařazení do kontextu problematiky jednotlivých částí.

1.1 Organizace

Definice organizace podle anglického slovníku Merriam-Webster je buď proces systematizace a zapojení částí do logické jednoty nebo funkčního celku, nebo sdružení či společnost s administrativní a funkční strukturou a je charakterizována naprostou shodou s požadavky a pravidly takovéto společnosti.¹ Pro zavedení a návrh informačního systému je nutné pochopit význam organizace a její strukturu. Mnoho procesů a komunikace se odehrává prostřednictvím informačního systému, který slouží k podpoře a komunikaci v rámci tohoto funkčního celku.

1.1.1 Organizační struktura

Organizační struktura je systém, který popisuje způsob řízení činností a aktivit, za účelem dosažení cílů organizace. Tyto aktivity zahrnují pravidla, role a odpovědnosti. Organizační struktura dále zahrnuje, jakým způsobem bude nakládáno s informacemi mezi jednotlivými úrovněmi v rámci společnosti.²

Podle způsobu, jakým jsou v organizaci uplatňována pravidla, role a zodpovědnosti, pak dělíme organizační struktury na liniové, štábní a kombinované.

Liniová struktura se uplatňuje v menších podnicích, existuje zde jediný řídicí stupeň, který reprezentuje ředitel, pravomoci jsou přidělovány přímo. Vedoucí pracovníci pak vykonávají svoji pravomoc na vertikální úrovni. Při snaze aplikovat tuto strukturu na větší podnik dochází k vysokým požadavkům na kvalifikovanost vedoucích zaměstnanců a jejich rozhodování.

Štábní struktura vyplňuje nedostatky lineární struktury přidělením každému důležitému rozhodovacímu prvku (vedoucímu) v hierarchii štábní středisko. Toto středisko dohlíží

¹ Organization: Definition of Organization by Merriam-Webster. Merriam-Webster [online]. Springfield: Merriam-Webster, ©2021, 2021 [cit. 2021-02-17]. Dostupné z: <https://www.merriam-webster.com/dictionary/organization>

² Organizational Structure. Investopedia [online]. New York: Investopedia, 2020, 2020 [cit. 2021-02-10]. Dostupné z: <https://www.investopedia.com/terms/o/organizational-structure.asp>

na rozhodování daného bodu, přináší odborné poznatky sloužící jako podpora rozhodování pro daný uzel. Štábní skupina je složena z osobních poradců, sekretářek a odborných či funkčních poradců. Znalosti těchto poradců jsou často hluboké, ale chybí jim dobrý přehled o celkovém chodu společnosti nebo odvětví.³

Další typy organizačních struktur jsou struktury kombinované, řadíme zde liniově štábní, maticovou a funkční strukturu. Tyto struktury vznikají z důvodu diferenciací jednotlivých organizací podle prostředí a rozsahu, ve kterém působí. K mojí bakalářské práci se nejvíce vztahuje funkční struktura.

Funkční struktura je jednou ze základních forem organizace, kde jsou zaměstnanci/dobrovolníci řazeni do skupin podle specializace jednotlivých útvarů. Každá část organizace musí odvést svůj pracovní podíl na vysoké úrovni, což se poměrně dobře daří díky hlubokým znalostem svého oboru a specializaci jednotlivých útvarů.

Výhodou funkční organizační struktury jsou jasné cíle v rámci jednotlivých úseků a dokonalejší řízení, které má za následek i lepší motivaci pracovníků k týmové spolupráci. Ucelená strategie funkční struktury je zapříčiněna způsobem rozhodování – je centralizováno od shora dolů.

Nevýhodami funkční struktury může být nezdravé soupeření na úrovni úseků, dělení na úseky, které může vést k velké oddělenosti pracovníků. Toto oddělení může vést ke snížení inovativnosti celého podniku.⁴

1.2 Management

Již z anglického slova manage – řídit můžeme odvodit podstatu samotné disciplíny, která se zabývá řízením. V jednoduchosti bychom mohli říct, že jde o usměrňování lidské činnosti v efektivní a účelnou. K dosažení efektivity a účelnosti v mnoha organizacích je třeba se orientovat ve velkém množství dalších disciplín, které souvisí se specifiky řízení dané činnosti a kontextu, ve kterém je řízena. Dochází tak k průniku mnoha dalších disciplín, mezi které patří: psychologie, sociologie, ekonomie, účetnictví, právo, marketing, matematika, informatika a statistika. Efektivní řízení se bude opírat

³ CEJTHAMR, Václav a Jiří DĚDINA. Management a organizační chování. 2., aktualiz. a rozš. vyd. Praha: Grada, c2010. Expert (Grada). ISBN isbn978-80-247-3348-7. s.203-205.

⁴ CEJTHAMR, Václav a Jiří DĚDINA. Management a organizační chování. 2., aktualiz. a rozš. vyd. Praha: Grada, c2010. Expert (Grada). ISBN isbn978-80-247-3348-7. s.214-215

o poznatky z těchto a mnoha dalších disciplín proto, aby dosáhnul vytyčeného cíle v rámci této problematiky.⁵

Myslívová definuje management takto: „*Nejobecněji je možné management charakterizovat jako souhrn všech činností, které je třeba udělat, aby byla zabezpečená funkce organizace. Méně obecná definice udává, že management je proces plánování, organizování, vedení, kontroly a řízení pracovníků organizace a jimi využívaných zdrojů, jehož působením organizace směřuje k vytyčeným podnikatelským cílům s vynaložením co nejmenších nákladů.*“⁶ Dále poukazuje na to, že pojem management označuje buď vedoucí pracovníky, činnost, kterou vedoucí pracovníci vykonávají pomocí delegování pravomocí a zodpovědností na své podřízené, nebo dále poukazuje na využití tohoto termínu jako označení vědní činnosti a předmětu odborného vyučování.⁷

1.2.1 Základní pojmy

Proces „*je souhrnem činností transformujících souhrn vstupů do souhrnů výstupů (zboží nebo služeb) pro jiné lidi nebo procesy, používající k tomu lidi a nástroje.*“⁸ Anglický slovník Merriam-Webster definuje slovo proces jako série po sobě jdoucích činností nebo operací vedoucí k cíli.⁹ Jedním z běžných procesů, se kterým se setkáváme každý den, je nakupování v obchodě.

Pro větší podniky je dnes nezbytností tyto procesy systematizovat a zaznamenávat. Důvody pro takové činnosti mohou být zvýšení efektivity, snížení nákladů a zlepšení kvality procesů. Pokud procesy nejsou dostatečně funkční, můžeme tyto procesy kompletně nahradit jinými lépe funkčními. Této činnosti se říká Business process reengineering.

Slovo projekt pochází z latiny, kde „pro-jicio“ znamená plánovat vpřed – a obecně se tento termín používá pro popis nestandardních řešení, které se po ukončení projektu

⁵ KOTRBA, Tomáš a Miroslav FORET. Základy managementu. Znojmo: Soukromá vysoká škola ekonomická, 2009. ISBN isbn978-80-87314-00-5. s. 5-6.

⁶ MYSLIVCOVÁ, Kateřina. Základy managementu. Praha: Soukromá vysoká škola ekonomických studií, [2013]. ISBN isbn978-80-7523-000-3.

⁷ MYSLIVCOVÁ, Kateřina. Základy managementu. Praha: Soukromá vysoká škola ekonomických studií, [2013]. ISBN isbn978-80-7523-000-3.

⁸ ŘEPA, Václav. Podnikové procesy: procesní řízení a modelování. Praha: Grada, 2006. Management v informační společnosti. ISBN isbn80-247-1281-4. s. 13.

⁹ Přeloženo z Process. Merriam-Webster [online]. Springfield: Merriam-Webster, ©2021 [cit. 2021-02-15]. Dostupné z: <https://www.merriam-webster.com/dictionary/process>

neopakují.¹⁰ Projekty se vyznačují těmito atributy: jedinečný cíl, nejistota (riziko), potřeba projektového týmu, komplexnost (složitost) a přesné vymezení.¹¹

1.2.2 Procesní a projektový management

Již z předchozích základních definic může být jasné, jaký je rozdíl mezi projektovým a procesním managementem. Procesní management se zabývá řízením lidí a dalších zdrojů v opakujících se činnostech, které se dají předvídat, a není zde díky předvídatelnosti tak vysoká míra nejistoty jako v případě projektů. Využívá se tak v případě, kde je potřeba dosáhnout efektivně a opakovaně stejné kvality. Díky procesnímu řízení jsou pracovníci jednodušeji zastupitelní. Procesní management nemá za úkol procesy pouze stanovit a monitorovat, ale i vylepšovat, a tak dosahovat vyšší výkonnosti a kvality.

Projektové řízení se využívá tam, kde je k jednotlivým problémům potřeba přistupovat individuálně. Nelze zvolit stejný postup vícekrát po sobě z důvodu neopakovatelnosti této události v závislosti na kontextu v dané organizaci. Takovým příkladem může být například instalace nového zařízení do výrobní linky nebo implementace nového docházkového systému. Samotné každodenní používání systému či zařízení bude již spojeno s řízením procesů, nikoliv projektů. Tuto práci již nebude vykonávat různorodý projektový tým vytvořený z více specialistů s různou profesní orientací, ale skupina dobře zaškolených pracovníků, kteří pracují s nově zavedeným prvkem každý den.¹²

1.3 Nástroje projektového managementu

V této podkapitole popíšu hlavní součásti projektových nástrojů, které se zpracovávají při plánování projektu, ještě před jakoukoliv realizací projektu.

¹⁰ Procesní a projektové řízení. Management.cz [online]. Management.cz, ©2015, 2014 [cit. 2021-02-17]. Dostupné z: <http://www.management.cz/procesni-a-projektove-rizeni/>

¹¹ Projektová kritéria (atributy). PM Consulting [online]. Choceň: PM Consulting, c[2005-2021], [2018] [cit. 2021-02-17]. Dostupné z: <https://www.pmconsulting.cz/slovníkový-pojem/projektova-kriteriata-atributy/>

¹² Procesní a projektové řízení. Management.cz [online]. Management.cz, ©2015, 2014 [cit. 2021-02-17]. Dostupné z: <http://www.management.cz/procesni-a-projektove-rizeni/>

1.3.1 Identifikační listina

Identifikační listina je základním dokumentem vytvářeným při zahajování projektu. Ustanovuje manažera projektu a opravňuje ho k nakládání se zdroji organizace. V identifikační listině jsou popsány:

- „*Definice cíle, výstupů, obsahu a rozsahu projektu, rámce rozpočtů, milníků*
- *Přidělení zdrojů (lidských, finančních) minimálně na fázi plánování*
- *Stanovení kompetencí projektového manažera*
- *Identifikace hlavních rizik, předpokladů a kritérií úspěšnosti*
- *Kvalita projektu*
- *Zahájení, ukončení projektu*
- *Zadavatel, sponzor, manažer projektu, tým*¹³

1.3.2 Work Breakdown Structure

Work breakdown structure(WBS) je hierarchickou dekompozicí práce, kterou má projektový tým vykonat, a jednotlivé části práce, které mají být součástí výstupů celého projektu. Toto rozložení práce slouží k plánování, kontrole, monitoringu i nacenění práce. Work breakdown structure je vyvinuta, aby určila všeobecné pochopení rozsahu projektu. Musí být udělána tak, aby byly jasně definované výstupy, které se postupně rozpadají na menší celky. Tyto drobnější celky přidávají detailní popis hlavního cíle a určují v jaké míře a rozsahu se bude projekt řešit. Všechny úkoly, které musí být udělány, jsou zahrnuty právě ve WBS, pokud ne, nebude s nimi počítáno při plánování a v důsledku nebudou hotovy.¹⁴

1.3.3 Ganttovy diagramy

Ganttův diagram je jedním z nástrojů manažerského řízení, který pomáhá vizualizovat plánování projektů. Pomáhá zjednodušit představení složitých projektů. Jednotlivé úkoly jsou seřazeny v jednom sloupci pod sebou. Je jim přiřazeno datum začátku úkolu a jeho

¹³ SMOLÍKOVÁ, Lenka. 5. Přednáška, Řízení projektů v ICT: Zainteresané strany. [Brno], 2020.

¹⁴ What is a Work Breakdown Structure? [online]. Workbreakdownstructure.com, ©2021 [cit. 2021-4-27]. Dostupné z: <https://www.workbreakdownstructure.com/>

konec, zodpovědná osoba nebo zodpovědný tým a „předcházející úkoly“ (úkoly, které mají být udělány před začátkem tohoto úkolu), pak následuje vizualizace na časové ose, kde jsou zobrazeny úkoly jejich návaznosti, data dokončení úkolů a milníky – důležitá kontrolní data projektu, kterých je potřeba dosáhnout.¹⁵

1.3.4 RIPRAN analýza

Metoda RiSk PRoject ANalysis představuje empirickou metodu pro analýzu rizik projektu.¹⁶ „Metoda RIPRAN™ vznikla původně pro analýzu rizik automatizačních projektů v rámci výzkumného záměru na VUT v Brně v roce 2000, kdy se autor B. Lacko (pracovník Ústavu automatizace a informatiky) snažil analyzovat rizika při vývoji automatizačních systémů v rámci výzkumných projektů pracoviště.“¹⁷

Tuto metodu lze vypracovávat pomocí slovního ohodnocení nebo případně pomocí číselného vyjádření. Protože neznáme přesné statistické hodnoty jednotlivých pravděpodobností hrozeb, využijeme tabulky 3*3 pro slovní hodnocení rizik, dostupné na stránkách webu ripran.cz. Autoři zde uvádí ještě možnost použití tabulek s vyšším členěním hrozeb (do pěti kategorií – 5*5). Toto členění je však doporučeno pro projekty s dobře zajištěnými statistickými údaji.¹⁸

V současném projektu budeme postupovat v několika krocích, které budeme postupně zaznamenávat do tabulky. Jeden krok symbolizuje jeden sloupec. V první řadě určíme **hrozby**, které mohou na náš projekt působit. Hrozba je označení příčiny/zdroje problému, který může negativně ovlivnit výsledek projektu a jeho výstupy, datum dokončení, atd.¹⁹ Následně připíšeme **scénář**, při kterém může tato hrozba působit na náš projekt.

Následně určujeme **pravděpodobnost** podle metodiky RIPRAN. Pravděpodobnost můžeme vyhodnocovat buď slovně, nebo číselně. Slovně ohodnotíme pravděpodobnost

¹⁵ What is a Gantt chart? Association for Project Management [online]. Association for Project Management, ©2020, 2020 [cit. 2021-4-27]. Dostupné z: <https://www.apm.org.uk/resources/find-a-resource/gantt-chart/>

¹⁶ RIPRAN: Metoda pro analýzu projektových rizik [online]. Lacko, [2000] [cit. 2021-4-27]. Dostupné z: <https://ripran.cz/>

¹⁷ Historie vzniku metody. RIPRAN: Metoda pro analýzu rizik [online]. Lacko, [2000] [cit. 2021-4-27]. Dostupné z: <https://ripran.cz/historie.html>

¹⁸ Tabulky pro verbální hodnocení rizik: Soustava 3 x 3 x 3. [Brno], [2000]. Dostupné také z: <https://ripran.cz/tab3.pdf>

¹⁹ Hrozba (Threat). ManagementMania.com [online]. ManagementMania.com, © 2011-2016, 17. 2. 2016 [cit. 2021-4-27]. Dostupné z: <https://managementmania.com/cs/hrozba-threat>

hrozby a pravděpodobnost scénáře. K tomuto hodnocení využíváme tabulku pravděpodobnosti tříd.

Vysoká pravděpodobnost	VP	Nad 66 %
Střední pravděpodobnost	SP	33 až 66 %
Nízká pravděpodobnost	NP	Pod 33 %

Tabulka 1: Třídy pravděpodobnosti analýzy RIPRAN²⁰

Po zařazení jednotlivých hrozeb a scénářů do tříd pravděpodobnosti doplníme výslednou pravděpodobnost na základě této tabulky:

Pravděpodobnost hrozby	Pravděpodobnost scénáře	Výsledná pravděpodobnost
VP	VP	VP
VP	SP	SP
VP	NP	NP
SP	VP	SP
SP	SP	SP
SP	NP	NP
NP	VP	NP
NP	SP	NP
NP	NP	NP

Tabulka 2: Tabulka pro vyhodnocení výsledné pravděpodobnosti hrozby²¹

Kromě pravděpodobnosti doplníme i **dopad**, který hrozba na projekt bude mít. K přiřazení správné kategorie dopadu využijeme tabulku pro ohodnocení dopadu s doporučenými hodnotami.

²⁰ Vlastní zpracování dle: Tabulky pro verbální hodnocení rizik: Soustava 3 x 3 x 3. [Brno], [2000]. Dostupné také z: <https://ripran.cz/tab3.pdf>

²¹ Tabulky pro verbální hodnocení rizik: Soustava 3 x 3 x 3. [Brno], [2000]. Dostupné také z: <https://ripran.cz/tab3.pdf>

Velký nepříznivý dopad na projekt VD	Ohrožení cíle projektu nebo ohrožení koncového termínu projektu nebo možnost překročení celkového rozpočtu projektu nebo škoda přes 20 procent z hodnoty projektu.
Střední nepříznivý dopad na projekt SD	Škoda od 0,51 do 19,5 % z hodnoty projektu nebo ohrožení termínu, nákladů. Respektive zdrojů některé dílčí činnosti, což bude vyžadovat mimořádné akční zásahy do plánu projektu.
Malý nepříznivý dopad na projekt MD	Škody do 0,5 % z celkové hodnoty projektu nebo dopady vyžadující určité zásahy do plánu projektu

Tabulka 3: Tabulka pro hodnocení dopadu na projekt²²

Na základě doplněného dopadu hrozby a celkové pravděpodobnosti doplňujeme třídu hodnoty rizika jako výslednou hodnotu. Tato hodnota nám pomáhá určit, které hrozby můžeme akceptovat a které je nutno řešit. Pro hrozby s vysokou a střední hodnotou rizika je potřebné připravit opatření, kterým minimalizujeme nebo předejdeme hrozbě.

	Velký nepříznivý dopad na projekt	Střední nepříznivý dopad na projekt	Malý nepříznivý dopad na projekt
Vysoká pravděpodobnost	Vysoká hodnota rizika VHR	Vysoká hodnota rizika VHR	Střední hodnota rizika SHR
Střední pravděpodobnost	Vysoká hodnota rizika	Střední hodnota rizika SHR	Nízká hodnota rizika NHR
Nízká pravděpodobnost	Střední hodnota rizika SHR	Nízká hodnota rizika NHR	Nízká hodnota rizika NHR

Tabulka 4: Přiřazení třídy hodnoty rizika²³

Podle výsledků analýzy doplníme opatření, která zamezí působení hrozby, sníží pravděpodobnost jejího výskytu nebo povede ke snížení dopadu hrozby.

1.3.5 Analýza zainteresovaných stran

Analýza zainteresovaných stran pomáhá projektovému týmu přemýšlet o jednotlivých osobách a společnostech ve vztahu k projektu. Je dobré, aby projektový manažer upravil projekt takovým způsobem, aby projekt vyhovoval všem zainteresovaným stranám. Zainteresované strany – stakeholdery – rozdělíme podle vztahu k organizaci na vlastní

²² Vlastní zpracování dle: Tabulky pro verbální hodnocení rizik: Soustava 3 x 3 x 3. [Brno], [2000]. Dostupné také z: <https://ripran.cz/tab3.pdf>

²³ Tabulky pro verbální hodnocení rizik: Soustava 3 x 3 x 3. [Brno], [2000]. Dostupné také z: <https://ripran.cz/tab3.pdf>

a externí. Následně vypíšeme jejich očekávání a cíle, kterých chtějí dosáhnout. Určíme sílu příčiny – tedy to, jakým způsobem mohou projekt ovlivnit, a proč jsou pro nás tyto strany důležité. Po vypsání těchto jednotlivých atributů každé strany určíme subjektivně hodnocení na stupnici od jedné do desíti, kde číslo deset má nejvyšší váhu. Toto výsledné hodnocení udává, jaký vliv mají jednotlivé strany na náš projekt.²⁴

1.4 Analýzy použité pro popis současného stavu

V této podkapitole popíšu, jaké jednotlivé analýzy jsem použil pro zhodnocení současného stavu společnosti.

1.4.1 McKinseyův rámec „7S“

Tento rámec vznikl v americké konzultační firmě, podle které má také svůj název. V současnosti definovali sedm prvků, které jsou kritickými faktory úspěchu:²⁵

- *„STRUCTURE: vhodně nastavená hierarchie, kontrolní mechanismy, sdílení informací*
- *SYSTEMS: všechny prvky ICT řešení*
- *STYLE: styl řízení společnosti, přístup k riziku*
- *STAFF: hodnocení motivovanosti a schopnosti kooperace pracovníků*
- *SKILLS: hodnocení dovedností pracovníků a jejich aplikace v praxi*
- *SHARED VALUES: identifikace hodnot sdílených pracovníky a stakeholdery*
- *STRATEGY: vychází z vize společnosti, je v souladu s výše uvedenými aspekty.*²⁶

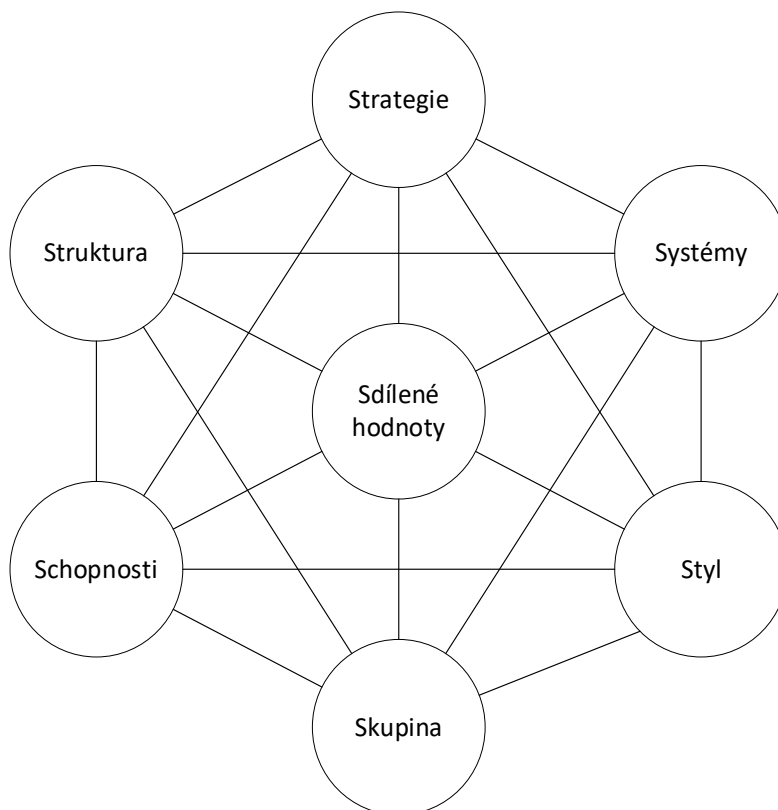
Jak je patrné z obrázku číslo 1. jednotlivé body jsou vzájemně provázány a pro úspěch celé společnosti je důležité, aby jednotlivé body byly ve vzájemné harmonii. Strategie, struktura a systémy se nazývají tvrdé dovednosti, protože jsou lehčeji definovatelné a hmatatelné – dají se dobře ovlivnit a sdílet.

²⁴ SMOLÍKOVÁ, Lenka. 5. Přednáška, Řízení projektů v ICT: Zainteresané strany. [Brno], 2020.

²⁵ KAŠÍK, Josef a Jiří FRANEK. Základy podnikové diagnostiky. Ostrava: VŠB-TU Ostrava, 2015. ISBN 978-80-248-3888-5. s.38

²⁶ ŠPATENKA, Jan. Prezentace, Podnikové informační systémy: Analýza vnitřního prostředí 7S. Brno, [2021]. Dostupné také z: https://moodle.vutbr.cz/pluginfile.php/340649/mod_resource/content/1/03%20-%20Podnikov%C3%A9%20informa%C4%8Dn%C3%AD%20syst%C3%A9my.pdf

Ostatní body této analýzy se nazývají měkké dovednosti. Obsahují v sobě prvek kultury a nejsou jednoduše hmatatelné. Bude náročnější je měnit i sdílet.²⁷



Obrázek 1: 7S analýza²⁸

1.4.2 Moscow analýza

Je analýza používaná v případě vysokého množství požadavků od zadavatele nebo v případě, kdy je potřeba určit jaké priority v projektu potřebujeme naplnit v rámci možností současných limitů (času, kapacit a rozsahu). Současné požadavky vytvořené zadavatelem nebo tvůrcem projektu se pak řadí do těchto čtyřech skupin:

- MUST – popisuje požadavky, které projekt musí obsahovat proto, aby byl výsledek uspokojivý.

²⁷ KAŠÍK, Josef a Jiří FRANEK. Základy podnikové diagnostiky. Ostrava: VŠB-TU Ostrava, 2015. ISBN 978-80-248-3888-5. s.39

²⁸ KAŠÍK, Josef a Jiří FRANEK. Základy podnikové diagnostiky. Ostrava: VŠB-TU Ostrava, 2015. ISBN 978-80-248-3888-5. s.39

- SHOULD – ukazuje na prvky, které mají vysokou prioritu a měli by být zahrnuty do projektu. Těchto požadavků se dá dosáhnout i jiným způsobem mimo řešení projektu, a proto se v kritických případech dá vynechat.
- COULD – tato oblast požadavků obsahuje vytoužené prvky řešení projektu, které často přidávají na výsledné přívětivosti celého řešení. Nejsou však nepostradatelné, a proto se do vývoje zařazují až k závěru, kdy to čas a kapacity dovolí.
- WON'T – prvky, které v nejbližší verzi projektu nebudou uskutečněny, ale do budoucna je dobré, je mít definované.²⁹

1.5 Informační systém

Pro definování informačního systému je důležité si na začátek určit některé další termíny, jako jsou data, informace, znalosti a moudrost. Dále se pak zaměříme na samotnou definici systému.

1.5.1 Data, informace, znalosti, moudrost

Data jsou běžně chápána jako objektivní hodnoty, které mohou pro člověka nabývat na významnosti. V takovém případě se z dat stávají informace. Data jsou interpretovatelná – schopna vypovídací hodnoty. Data jsou vázána na určitou formu nosiče.

Informace jsou data, která člověk interpretuje a považuje za důležitá, snižují rozhodovací nejistotu. Mají vypovídající hodnotu. Jedinec musí umět informaci zachytit, zpracovat její obsah a samotný obsah pro něj musí být přínosný.

Znalost je dovednost porozumět sdělovaným informacím v kontextu informací již předem získaných. Kromě samotných jednotlivých informací znalost zahrnuje i vztahy mezi nimi. Znalost určuje, jak nakládat s dalšími informacemi.

Posledním termínem, který stojí nad znalostmi, je moudrost – jedná se o dovednost jednat na základě daných znalostí a informací.³⁰

²⁹ MoSCoW Analysis – Prioritising Your Requirements for Delivery. Business analysis excellence [online]. Melbourne: Business Analysis Excellence, ©[2021], ©2014 [cit. 2021-03-24]. Dostupné z: <http://business-analysis-excellence.com/moscow-analysis/>

³⁰ KOCH, Miloš. Datové a funkční modelování. Vyd. 2. Brno: Akademické nakladatelství CERM, 2006. ISBN 80-214-3252-7. s.4–6

1.5.2 Systém, informační systém

„Systém je účelově definovaná neprázdná množina prvků a množina vazeb mezi nimi, přičemž vlastnosti prvků a vazeb mezi nimi určují vlastnosti (chování) celku.“³¹

Na základě této definice systému můžeme definovat informační systém jako množinu prvků podílejících se na sdílení a ukládání informací. V některých předchozích publikacích se poměrně dosti zdůrazňují prvky hardware a software. Toto jsou však jedny z mnoha prvků, které zdaleka nejsou těmi nejdůležitějšími. V přemýšlení o správném pochopení informačního systému nám pomůže definice Sodomky a Klčové:

„Podnikový informační systém vytvářejí lidé, kteří prostřednictvím dostupných technologických prostředků a stanovené metodiky zpracovávají podniková data a vytvářejí z nich formalizační a znalostní bázi organizace sloužící k řízení podnikových procesů, manažerskému rozhodování a správně podnikové agendy.“³²

Jak si můžeme všimnout, v této definici dochází ke změně hlavního zaměření z hardware a software prvku na uživatele – člověka. Tato změna nám pomáhá chápat to, že nejdůležitějším prvkem je skutečně člověk, který s programem pracuje. Autoři předchozí definice ukazují na to, že právě přílišné zaměření na hardware/software je jedním z prvků neúspěchu mnoha podniků.³³

1.5.3 Objektové datové modelování

Datové modelování je disciplína, která nám pomáhá při tvorbě informačního systému. Díky této disciplíně určujeme, jaká data budeme ukládat a jaké vztahy mezi jednotlivými množinami dat budeme definovat pro dobrou funkčnost systému. Pro popis vzájemných vztahů se dá použít více názvosloví.

Objekt je základní jednotkou a odpovídá nějaké reálně existující entitě. Objekt má určité atributy – data, která chceme o objektu uchovávat, a vlastnosti – metody, které určují, jak se daný objekt chová.

³¹ GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009. Expert (Grada). ISBN 978-80-247-2615-1. s.23

³² SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7. s. 61

³³ [Tamtéž.]

Atribut se popisuje pomocí typu a délky atributu. Může nabývat pouze množině definovaných dat – reálná čísla, celá čísla, text...³⁴

Primární klíč integritní omezení používané pro zajištění jedinečnosti objektu v rámci svojí třídy. Může se jednat o uměle vytvořený čítač formou ID – identifikačního čísla, složeniny čísel a písmen nebo se může jednat o některý reálný atribut daného prvku již zapsaného v databázi. Banka má tedy například možnost použít – číslo účtu, ID zákazníka, rodné číslo nebo číslo občanky. Z bezpečnostních důvodů se však nejčastěji používá uměle vytvořený klíč. Těmto alternativním klíčům se říká kandidátní klíče.

Složený primární klíč, vzniká spojením více atributů dohromady, u kterých máme jistotu, že se daná kombinace nebude opakovat v rámci celé třídy. Kromě toho je požadavkem na složený primární klíč i minimálnost takového klíče. Nechceme jako složený primární klíč používat všechny atributy daného objektu. Tím by vznikalo nesmyslné uvádění redundantních informací napříč databází formou cizího klíče.

Cizí klíč je primární klíč objektu z jiné třídy. Díky cizím klíčům můžeme popisovat vztahy mezi objekty z různých tříd.³⁵

Mezi jednotlivými třídami určujeme několik základních vztahů nebo vazeb.

- 1:0..1 – jeden objekt třídy A může mít pouze vazbu vůči jednomu nebo žádnému objektu třídy B.
- 1:0..N – jeden objekt třídy A může mít vazbu na žádný nebo více objektů z třídy B.
- 0..N:0..M – jeden objekt z třídy A může mít žádnou nebo více vazeb na objekty z třídy B, a zároveň objekt z třídy B může mít žádnou nebo více vazeb na objekty z třídy A. Tento vztah se řeší dekompozicí pomocí dekompoziční třídy, ve které jeden objekt uchovává atributy primárních klíčů z objektu třídy A i B.
- Každý z výše zmíněných vztahů se vyskytuje i v omezení, kdy musí existovat alespoň jedna vazba na objekt z druhé třídy. Hlavní koncept však zůstává stejný.

³⁴ Objektově orientované programování a evoluce vývoje softwaru. ITnetwork.cz [online]. Praha: David Čápka, ©2021, ©2021 [cit. 2021-03-26]. Dostupné z: <https://www.itnetwork.cz/navrh/objektove-orientovane-programovani-a-evoluce-vyvoje-softwaru/>

³⁵ KOCH, Miloš. Datové a funkční modelování. Vyd. 2. Brno: Akademické nakladatelství CERM, 2006. ISBN 80-214-3252-7. s. 29

1.6 Objektově orientované programování³⁶

Hlavní myšlenkou objektově orientovaného programování je znovupoužitelnost jednotlivých částí programu ať již vlastního, nebo s pomocí využití již předprogramovaných knihoven, obsahující různé části kódu, které se pravidelně využívají. V rámci objektově orientovaného programování se vyskytují tři základní principy, které vysvětlují funkčnost objektového programování a jeho specifika.

1.6.1 Zapouzdření

Princip zapouzdření umožňuje skrýt určité metody a funkce používané uvnitř třídy pro interní záležitosti. Samotný programátor dané třídy definuje, jaké metody mohou ostatní zvenku používat, a jak v důsledku nakládat s daty uloženými v jednotlivých atributech. Jako ostatní programátoři tedy víme, jaké metody můžeme používat, nevíme, jak fungují, ale předpokládáme, že fungují spolehlivě, a že nám dovolí nakládat s takovými daty, která jsou pro nás bezpečná.

1.6.2 Dědičnost

Dědičnost vyjadřuje schopnost vytvářet podtřídy, které si přebírají vlastnost původní třídy. Krom těchto vlastností a metod můžeme použít i metody další, které se již budou vyskytovat pouze pro danou podtřídu. Pokud zároveň dopíšeme některý atribut mateřské třídy, projeví se tato změna i v ostatních třídách, které dědí vlastnosti mateřské třídy. Můžeme vytvořit i třídy abstraktní, které nebudou mít žádný vlastní objekt a jsou vytvořeny pouze pro dědění jednotlivých atributů.

1.6.3 Polymorfismus

„Polymorfismus umožňuje používat jednotné rozhraní pro práci s různými typy objektů... Polymorfismus nám umožňuje přepsat si libovolnou metodu u každé podtřídy tak, aby

³⁶ Objektově orientované programování a evoluce vývoje softwaru. ITnetwork.cz [online]. Praha: David Čápka, ©2021, ©2021 [cit. 2021-03-26]. Dostupné z: <https://www.itnetwork.cz/navrh/objektove-orientovane-programovani-a-evoluce-vyvoje-softwaru/>

*dělala, co chceme. Rozhraní tak zůstane zachováno a my nebudeme muset přemýšlet, jak se to u onoho objektu volá.*³⁷

1.7 Webová aplikace

Při vývoji jednotlivých informačních systémů se dnes setkáváme se dvěma přístupy, které popisují náročnost informačního systému na zařízení uživatele – klienta. Těmito přístupy nebo metody jsou:

- Metoda tlustého klienta: většina výkonů s daty je přenechána na zařízení uživatele. Server slouží pouze k ukládání výsledných operací provedených na počítači většinou v rámci robustní aplikace s vysokými požadavky na výkon.
- Metoda tenkého klienta: většinu procesů práce s daty přesouvá na server a na stranu uživatele již posílá pouze výsledné hodnoty. Zařízení klienta je tak méně zatěžované. Ke komunikaci se serverem slouží buď nainstalovaná aplikace, webová stránka. Tato metoda se v poslední době využívá čím dál častěji, díky zlepšování stability a dostupnosti internetového připojení.³⁸

Webová aplikace je tedy dostupná z většiny funkčních zařízení a pokud je optimalizovaná pro mobilní zařízení, je dostupná takřka odkudkoliv.

1.8 Programovací jazyk C#

C# je moderní programovací jazyk, který je objektově orientovaný a má pevnou definici datových typů. Patří do rodiny jazyků C – C#, C, C++, Java, JavaScript a má s těmito jazyky mnoho společného. Je to strong-type language – to znamená, že záleží na použití velkých a malých písmen. Již od začátku je připravený pro využívání dalších softwarových částí. Jedná se o open source program, jehož hlavní dokumentaci vytváří společnost Microsoft. C# je jedním ze tří jazyků, které umožňují využití knihoven platformy .NET je platforma s širokou základnou mající uplatnění napříč mnohými odvětvími.³⁹

³⁷ Objektově orientované programování a evoluce vývoje softwaru. ITnetwork.cz [online]. Praha: David Čáпка, ©2021, ©2021 [cit. 2021-03-26]. Dostupné z: <https://www.itnetwork.cz/navrh/objektove-orientovane-programovani-a-evoluce-vyvoje-softwaru/>

³⁸ GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009. Expert (Grada). ISBN 978-80-247-2615-1. s. 57

³⁹ A tour of the C# language. Docs.microsoft.com [online]. Washington: Microsoft, ©2021, 1. 2021 [cit. 2021-4-27]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

2 Analýza problému a současné situace

V této části se chci zaměřit na popis současného stavu řešení a analyzovat možné nápravy případných nedostatků současného řešení.

2.1 Popis zapsaného spolku

Název společnosti	Křesťané Kuřim
Adresa	Smetanova 572/4, 664 34 Kuřim
Právní forma	Zapsaný spolek
IČO:	07706791

Tabulka 5: Popis společnosti⁴⁰

Křesťané Kuřim je zapsaný spolek založený za účelem vzdělávání a vyučování biblických pravd. Hlavní činností je pořádání bohoslužeb, vyučování související s hlavními cíli spolku. Dále se zapsaný spolek podílí na organizaci společenských akcí, jako například anglické tábory, rodinné pobyty a biblické konference.

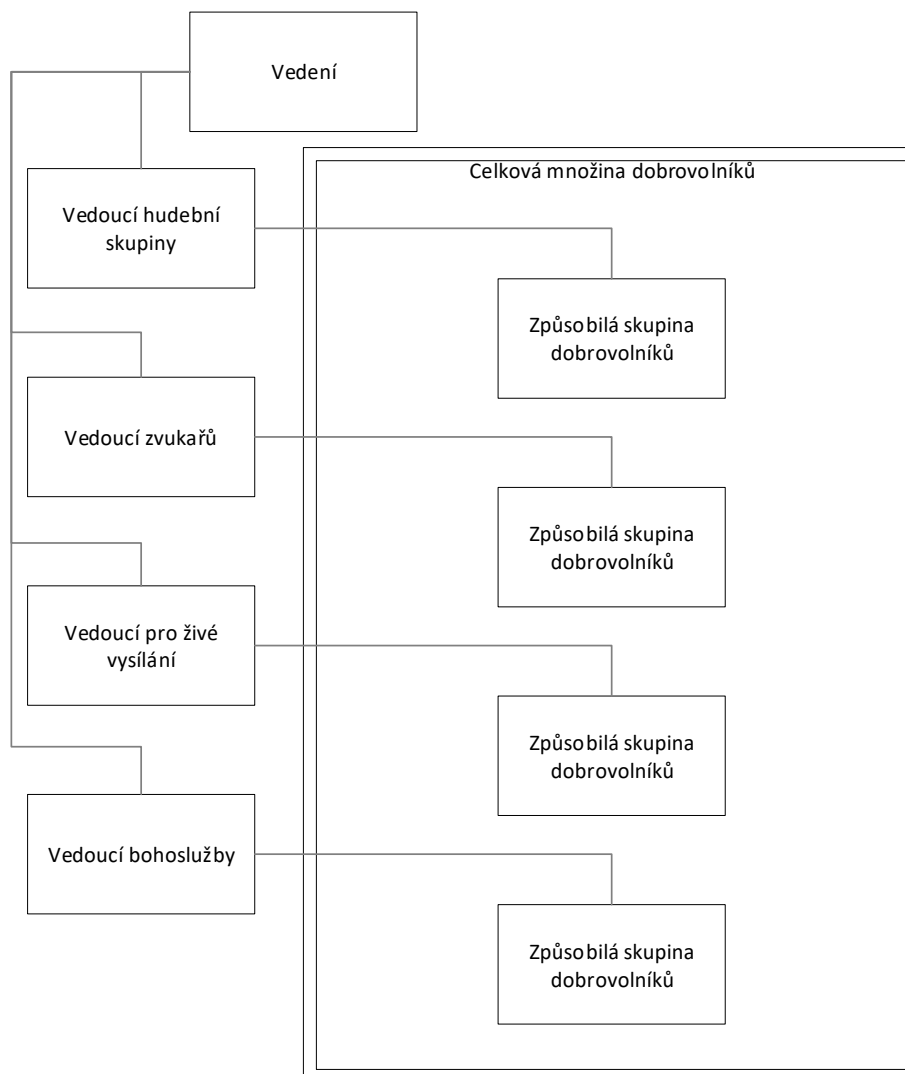
Zapsaný spolek využívá externích služeb účetnictví a přednáškové činnosti. Celkový chod je zajišťován seskupením dobrovolníků, kteří jsou aktivně zapojeni do chodu společnosti.⁴¹

⁴⁰ Vlastní zpracování dle: Kontakt: Křesťané Kuřim. Křesťané Kuřim [online]. Kuřim: Křesťané Kuřim, ©2021, 2021 [cit. 2021-03-17]. Dostupné z: <https://krestanekurim.cz/kontakt/>

⁴¹ Stanovy Spolku. Kuřim: Křesťané Kuřim, 2019, 7 s. L 25552/SL1/KSBR. Dostupné také z: <https://or.justice.cz/ias/ui/vypis-sl-detail?dokument=56204859&subjektId=1038458&spis=1153055>

2.2 Organizační struktura

Organizační strukturu bychom mohli popsat podle následujícího nákresu a nejlíže by se dala popsat jako struktura funkční:



Obrázek 2: Organizační schéma společnosti⁴²

V tomto projektu se budu zabývat pouze funkčním pohledem na strukturu organizace. Dále bych mohl aplikovat i pohled přes strukturu maticovou, ve které jsou zahrnuty možnosti průniků mezi jednotlivými funkčními celky v případě krátkodobých projektů. Tyto projekty bývají ve většině případů silně neformálního charakteru a v současné době

⁴² Vlastní zpracování

není požadavek ani důraz na jejich formalizaci. Proto tento pohled vynechám a dále budu na podnik vztahovat strukturu funkční.

Jedinou výjimkou neziskové organizace oproti podniku je, že dobrovolníci nejsou striktně přiděleni k jednomu funkčnímu celku, ale podle svých možností se zapojují do více funkčních celků. Je tedy možné, aby jeden dobrovolník měl dva vedoucí, pro rozdílné oblasti působnosti. V tomto případě je potřeba zajistit, aby nedocházelo k přidělení nezvládnutelného množství zodpovědností (činností) v rámci jedné události.

2.3 7S Analýza

7S analýza má ukázat na současnou vnitřní analýzu organizace. Jako podklad využiji své vlastní zkušenosti, které s tímto církevním spolkem mám již od roku 2017, kdy jsem se začal zapojovat jako dobrovolník.

2.3.1 STRUCTURE: vhodně nastavená hierarchie, kontrolní mechanismy, sdílení informací

Organizační strukturu jsem popsal výše v podkapitole 1.3 Organizační struktura. Vhodnost této struktury se mi zdá na současnou situaci poměrně efektivní a dostačující potřebám organizace. Problém nastává v případě, kdy vedoucí zapomene přidělit k dané události osobu zodpovědnou za určitou činnost. Kontrolní mechanismus v současné chvíli probíhá tak, že osoba, která je organizátorem celé akce nebo události, se přibližně jeden nebo dva dny před začátkem podívá do rozpisu a kontaktuje vedoucího činnosti s prosbou o doplnění zodpovědné osoby.

Neexistence kontrolních procesů částečně zapříčiňuje i špatné sdílení informací. Informace nejsou sdíleny včas a v dostatečné míře. Dobrovolníci i v případě, že dlouhodobě dopředu ví, že nebudou na určité události přítomni, provádí změny přibližně dva týdny před začátkem události. Pokud by se zavedl jednotný komunikační kanál s vedoucím a oni by dali vědět ještě předtím, než se určí rozpis, výrazně by se snížilo množství úprav zodpovědností na poslední chvíli. Protože v současné chvíli neexistuje ustálený komunikační portál/informační systém, bylo by velice náročné pro jednotlivé vedoucí udržovat záznam o nedostupnosti jednotlivých dobrovolníků.

Samotná hierarchie je dobře nastavená, ale kontrolní mechanismus a sdílení informací nikoliv. Ke zlepšení kontrolních mechanismů by měl sloužit nový informační systém,

který bude vedoucí dopředu upozorňovat na přidělení zodpovědností. Jednotliví dobrovolníci budou také schopni do systému zadat, kdy se nemohou z jakéhokoliv důvodu události účastnit. V případě nutnosti změny bude možné zobrazit náhled volných dobrovolníků s danou specializací pro danou událost.

2.3.2 SYSTEMS: všechny prvky ICT řešení

V rámci zapsaného spolku Křesťané Kuřim, je ke správě jednotlivých setkání a událostí využíván systém tabulek Google Spreadsheets. V řádcích jsou zapsány jednotlivé zodpovědnosti, v každém sloupci je zapsané datum a jsou doplněna jména jednotlivých osob zodpovědných za danou činnost. V tabulce číslo 3 uvedené níže, je nastíněn útržek z této tabulky.

[Název činnosti]	[Datum události]	12.2.2021
Zvučení	[zodpovědná osoba za činnost popsanou v prvním sloupci daného řádku]	Marcel Tron
Livestream	[zodpovědná osoba]	Josef Sponar
Promítání	[zodpovědná osoba]	Izák Morhed

Tabulka 6: Ukázka ze současného systému⁴³

Současné řešení přestává být vhodné, protože se začíná hromadit větší množství činností, které je potřeba přidělit jednotlivým dobrovolníkům. Toto přidělení mají na starosti vedoucí jednotlivých činností, kteří zastřešují celý chod dané činnosti. Jsou v plném rozsahu zodpovědní za všechny související části mající vliv na činnost a její hladký průběh včetně pověření delegovat další osoby v částečném nebo celkovém rozsahu dané činnosti.

V aktuální situaci dochází k problémům na několika úrovních.

1. Vedoucí zapomene doplnit seznam na další měsíc.
2. Lidé přiřazení na jednotlivé pozice nejsou mnohokrát upozorněni na jejich zodpovědnost v dostatečném časovém předstihu.
3. Dochází ke kolizím – více souběžných činností je přiděleno jedné osobě.

⁴³ Vlastní zpracování

4. V případě, že zodpovědná osoba nemůže určitou činnost v rámci události vykonat, tato osoba má povinnost oslovit a najít náhradu za sebe samotného. Problémem je, že nikde není seznam osob, které jsou způsobilé k výkonu dané činnosti ve formalizovaném zavedení.

Druhým prvkem je emailová skupina, která přeposílá emailové zprávy všem dobrovolníkům.

2.3.3 STYLE: styl řízení společnosti, přístup k riziku

Styl řízení společnosti je neformální a vysoce přátelský. Neklade se zde veliký nebo žádný důraz na postavení vedoucího oproti ostatním. Ze strany vedení i ostatních není vidět odpor k formalizaci potřebných částí. Celkové motto, které se uplatňuje, vychází z reformační glosy Martina Luthera: „*Ecclesia semper reformanda est.*“, které volně přeloženo znamená „*církev neustále se reformující*“ a v tomto duchu se přistupuje i k řízení společnosti. Celkové řízení je založeno na biblických principech, které jsou patrné z Nového zákona. A dále je podpořeno a přeneseno i do všech dalších oblastí. Toto je důležité především z důvodů sdílených hodnot v rámci tohoto církevního spolku.

K rizikům je přistupováno opatrně, větší změny jsou před zavedením do provozu diskutovány s lidmi, kteří v dané oblasti pracují nebo mají hluboké zkušenosti. Nelze však říct, že by postoj vůči změnám byl negativistický.

2.3.4 STAFF: hodnocení motivovanosti a schopnosti kooperace pracovníků

Motivace pro působení v dobrovolné organizaci není dosahováno vnější formou pomocí peněz nebo benefitů, samotný chod organizace je z velké části hrazen právě samotnými dobrovolníky díky dobrovolným darům. Zde je nutno podotknout, že nedochází k jakékoliv formě manipulace, jak tomu bývá v případě nejrůznějších hnutí nebo sekt. Je zde důraz na dobrovolnost a schopnost dělat zvolenou činnost radostně, s motivací oslavit Boha. V případě, že se jednotlivec rozhodne nezapojovat vůbec, nepřichází vyhrožování nebo odstříhnutí od komunity vytvořené okolo Křesťané Kuřim.

Díky tomu, že se do organizace zapojují lidé s vnitřní motivací založenou na vlastním přesvědčení a světonázoru, je dosahováno vysoké kvality výstupu i bez dodatečné motivace. Mezi jednotlivci je však vidět veliká vděčnost jeden druhému za činnost nebo

službu, na které se podílí. A tak je tato vnitřní motivace doplněna oceněním a uznáním ze strany ostatních.

Kooperace mezi jednotlivci je podpořena přátelským prostředím a vzájemnými vztahy mezi pracovníky a v této oblasti nevidím nejmenší problém. Tato oblast je také založena na sdílených hodnotách a principech vycházejících z Bible.

2.3.5 SKILLS: hodnocení dovedností pracovníků a jejich aplikace v praxi

K hodnocení dovedností pracovníků není zavedena žádná formální metoda. Jednotliví dobrovolníci se zapojují do jednotlivých činností podle svojí vlastní touhy. Lidem je dána svoboda zapojit se do jednotlivých služeb či činností.

V případě, že někdo má touhu se dobrovolně zapojit, je pak podle jeho znalostí a zkušeností zvolen adekvátní individuální přístup a zaškolení potřebné pro zodpovědný výkon činnosti. Mezi jednotlivými dobrovolníky probíhá dobrý způsob neustálého předávání nových znalostí a zkušeností v jednotlivých oblastech jejich práce.

Je zcela běžné, že někteří dobrovolníci si vybírají činnosti, ve kterých nedosahují nejvyšší míry profesionality, ale jedná se o činnost, která je baví a poskytuje jim radost a odpočinek od práce. U složitějších činností se většinou najdou výborně kvalifikovaní dobrovolníci. Na účetnictví tak například dohlíží skupina lidí s ekonomickým vzděláním a zkušenostmi v podnikání, kde se jejich zkušenosti velice snadno aplikují do praxe a chodu spolku.

2.3.6 SHARED VALUES: identifikace hodnot sdílených pracovníky

Všichni dobrovolníci zapojující se do chodu a pořádání bohoslužeb jsou věřící křesťané. Společnými hodnotami tedy jsou hluboké křesťanské pravdy zakotvené v Bibli.

Šedesát šest knih Bible v originálních jazycích jsou neomylným Božím slovem. Na základě této zprávy vidíme, že člověk je hříšný, nedokáže sám ze sebe činit dobro a za své skutky si zaslouží odloučení od Boha v pekle. Bůh však ve svojí milosti poskytl způsob, jak přivést hříšníky k sobě. Dal svého jediného syna Ježíše Krista, kterého nechal zemřít za hříchy každého, kdo v něho věří. Třetího dne po jeho smrti byl vzkříšen z mrtvých, a tím poskytl důkaz o pravosti všeho, co říkal a učil. Zároveň ten, kdo v něj věří, je zadarmo a bez potřeby svých skutků ospravedlněn a díky Boží práci v něm dokáže se správnými motivy dělat to, co se Bohu líbí a má zaslíbený věčný život v nebi. Tato

zpráva se vztahuje na všechny lidi bez rozdílů. Hlavním posláním celé církve je jít a činit učedníky ze všech národů. Tato práce však začíná v místním společenství s touhou zakládat podobné spolky v dalších městech a zemích.⁴⁴

2.3.7 STRATEGY: jednání podle předchozích bodů

Strategií je připravit další lidi s touhou jít zakládat další podobná společenství v místech, kde jsou podobná společenství potřeba a kde je o ně zájem. Tento cíl se však nedá časově ohraničit, jelikož je dbáno na integritu životů jednotlivých učitelů a misionářů. Trvá dlouhou dobu, než se osvědčí.

Jedním z klíčových prvků v naplnění této strategie jsou společná setkání a bohoslužby. K organizaci těchto setkání by měl pomoci systém pro řízení událostí.

2.3.8 Závěr

Závěrem analýzy můžeme vidět, že organizace je lépe připravena v měkkých oblastech (bod 4. až 7.), než v těch tvrdých (bod 1 až 3). Je potřeba nastavit kontrolní mechanismy a mechanismy pro lepší sdílení informací. Tomu bude nápomocný informační systém, který sjednotí komunikaci mezi uživateli a v důsledku sníží i délku přípravy jedné události, sníží nejistotu v průběhu organizace událostí mezi dobrovolníky.

2.4 Analýza stávajícího řešení

Z předchozí analýzy vyšlo najevo, že jednou z nedostatečných prvků organizace je současná správa jednotlivých událostí. V rámci této podkapitoly vyhodnotím SWOT analýzu pro současné řešení.

2.4.1 SWOT analýza stávající řešení

Silné stránky

- Současné řešení umožňuje správu prostřednictvím iOS a Android aplikace.
 - Mnoho lidí je navyklých tuto platformu používat.
- Současné řešení nevyžaduje složitou správu.
- Minimální náklady na údržby a úpravy tohoto řešení
 - údržbu zvládne velké množství uživatelů.

⁴⁴ Více na <https://krestanekurim.cz/kdo-jsme/> a <https://krestanekurim.cz/co-ucime/>

- Prostředí Google Sheets je vytvářeno i v češtině.

Slabé stránky

- Informace ve větším množství nejsou přehledné.
- Chybí kontrola zdvojení zodpovědností.
- Řešení neumožňuje emailové připomenutí zodpovědných osob.
- Data může upravovat kdokoliv s odkazem ke sdílení.

Příležitosti

- Možnost zamknout tabulku pro editaci anonymním uživatelem.
- Možnost doprogramování maker v JavaScriptu.
- Možnost přidat další rozšíření funkcionality přes platformu Chrome store.
- Možnost tvorby výběrových listů pro jednotlivé tabulky (dropdown menu).

Hrozby

- Podle současných změn politiky Google je diskutabilní, jak dlouho bude Google sheets jako služba zdarma bez omezení – hrozí nárůst nákladů spojených s tímto řešením.
- Přístup přes nezabezpečený odkaz může způsobit narušení integrity nebo dostupnost tabulky.
- Vzhledem k nemožnosti integrovat vlastní přístupová práva pro každý řádek zvlášť, může docházet k nechtěným úpravám.

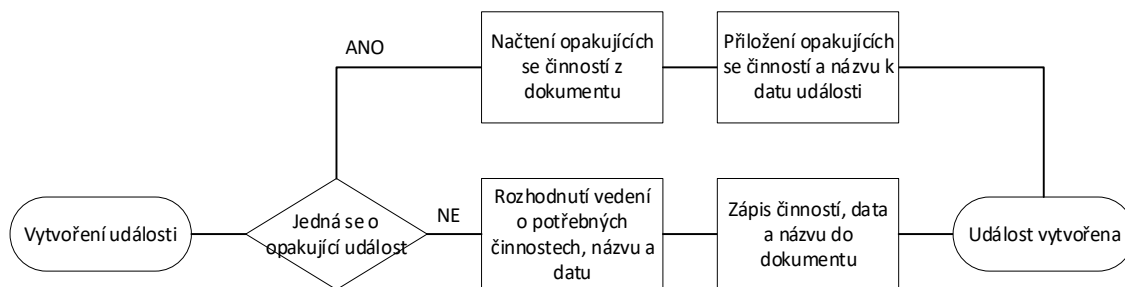
Závěr SWOT analýzy

Po zvážení silných stránek a příležitostí se současné řešení nezdá jako vyhovující. Jednotlivé vylepšení, kterých je možno dosáhnout pomocí rozšíření z Chrome store nebo doprogramování v JavaScriptu, by vyžadovalo mnoho práce související s pravidelnou údržbou a řešení by se tak stalo neefektivním.

2.5 Popis hlavních podnikových procesů a jejich vývojové diagramy

Vytvoření události s jednotlivými činnostmi. – Vedení vytvoří událost a zadá, které činnosti budou uskutečněny v rámci události. Tyto informace předá do rukou vedoucích jednotlivých skupin. Je zde rozdíl, zda se činnost opakuje každý týden a jednotlivé

činnosti jsou v rámci události stanoveny a opakují se, nebo se může jednat o událost, která bude obsahovat různé skupiny činností.



Obrázek 3: Diagram vytvoření události⁴⁵

Zrušení události. – Vedení informuje všechny účastníky včetně dobrovolníků o zrušení události.

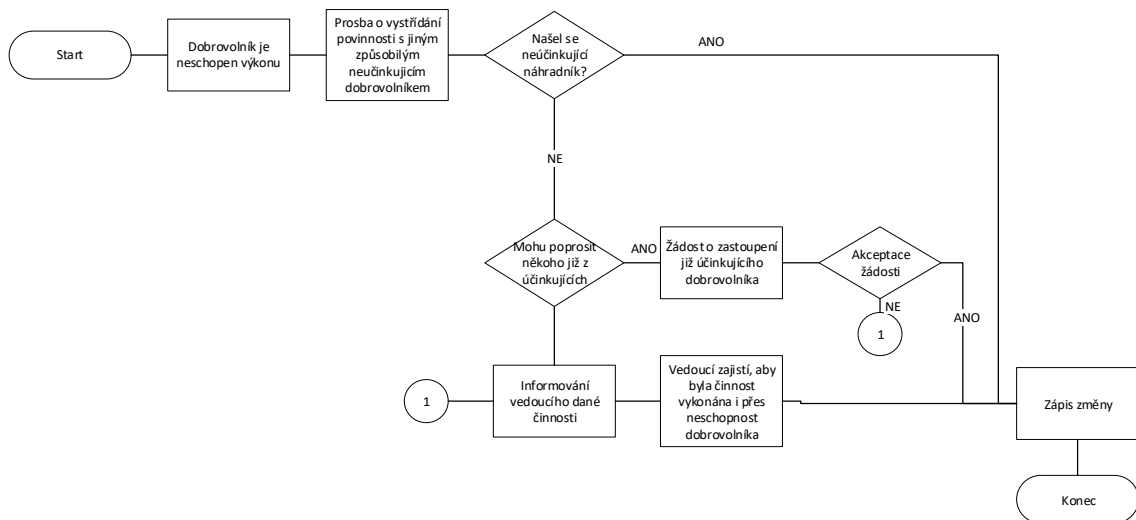


Obrázek 4: Diagram zrušení události⁴⁶

Změna dobrovolníka za jiného. – Zapsaný dobrovolník zjistí, že není schopen z jakýchkoliv důvodů vykonat danou činnost, ke které byl zapsán. Osloví všechny dobrovolníky z jeho skupiny, kteří nejsou již do dané události zapsáni s prosbou o vystřídání povinností. První náhradník, který potvrdí svoji účast je vyměněn za chybějícího a ostatním se dá vědět, že daný problém je již vyřešen.

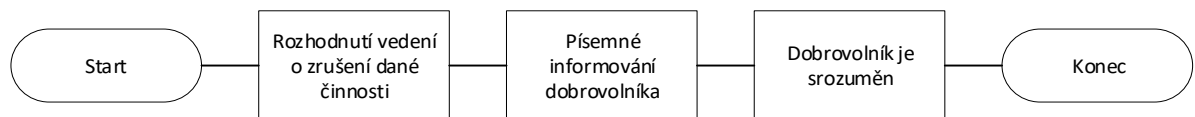
⁴⁵ Vlastní zpracování

⁴⁶ Vlastní zpracování



Obrázek 5: Diagram změny dobrovolníka za jiného⁴⁷

Zrušení činnosti v rámci události. – Vedení dá informaci dobrovolníkovi vykonávající činnost o nepotřebnosti dané činnosti v průběhu události.



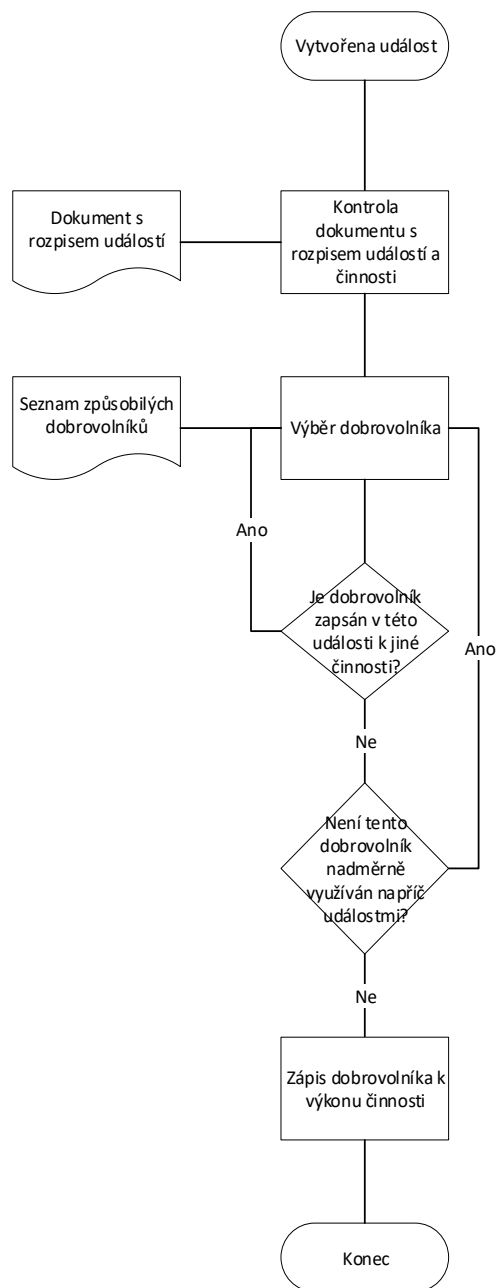
Obrázek 6: Diagram zrušení činnosti v rámci události⁴⁸

Doplnění dobrovolníků pro danou činnost v rámci události. – Ve chvíli, kdy jsou definovány jednotlivé činnosti pro danou událost, vedoucí dané skupiny doplní dobrovolníka pro konkrétní činnost a datum. Musí však ohlídat několik pravidel.

- Dobrovolníci nejsou zapsaní na více činností v rámci jedné události.
- Střídání dobrovolníků, pokud je to možné – kvůli zamezení přetížení jednoho člena skupiny.

⁴⁷ Vlastní zpracování

⁴⁸ Vlastní zpracování



Obrázek 7: Diagram doplnění dobrovolníků pro danou činnost v rámci události⁴⁹

⁴⁹ Vlastní zpracování

2.5.1 Moscow analýza – požadavky zadavatele

Na základě požadavků shrnutých v předchozí kapitole, můžeme naše požadavky na informační systém shrnout pomocí Moscow analýzy.

Informační systém	Požadavky
Musí mít	<ul style="list-style-type: none">• Uživatelské prostředí v češtině.• Upozornění na změny pro vedoucí jednotlivých částí.• Sdílení informací o nedostupnosti uživatele ve vztahu k jednotlivým událostem.• Emailové notifikace posílané z informačního systému jednotlivým uživatelům.• Plánování události a možnost přiřadit odpovědné osoby/uživatele ke konkrétní části události.• Mechanismus kontroly, zda odpovědná osoba může vykonávat danou činnost.
Měla by mít	<ul style="list-style-type: none">• Přehledný školicí portál a videa.• Možnost připnout k jednotlivým událostem soubory docx a pdf.• Dobře definovanou podporu na další dva roky provozu.
Bylo by dobré, kdyby měl	<ul style="list-style-type: none">• Synchronizaci s Google kalendářem a dalšími aplikacemi třetích stran.• Multijazyčnost (možnost přepnout do angličtiny).
Nemusí mít	<ul style="list-style-type: none">• Modul pro sledování zapojení jednotlivců – pro předcházení přetížení.• Kontrolu příjmů a výdajů organizace.

Tabulka 7: Požadavky zadavatele⁵⁰

2.6 Existující současná řešení

Jako alternativy uvedu další existující řešení z několika spřátelených církevních organizacích. Popíšu jakým způsobem se v jejich infrastruktuře řídí jednotlivé činnosti v rámci opakovaných setkání a bohoslužeb.

Ze zkušenosti a také od dotázaných dobrovolníků z dalších spolků vím, že v rámci České republiky řešení organizace vypadá velice podobně jako to, které jsem popsal výše –

⁵⁰ Vlastní zpracování

tabulka, předem určující zodpovědnosti jednotlivých dobrovolníků nebo jednoduché kalendáře.

Měl jsem však možnost dotázat se i na řešení v Grace Community Church – Huntsville,⁵¹ Alabama. Zde již dlouhou dobu funguje hierarchická organizační struktura, ve které hlavní vedoucí kontaktuje pouze vedoucí jednotlivých činností. Tito vedoucí pak využívají vlastních řešení pro organizaci jednotlivých rozpisů služeb. Toto řešení však postupně po jednotlivých činnostech přenášejí do informačního systému Realm.

Krom Realm Church jsem našel ještě mnoho dalších služeb pro církevní řízení. Rád bych tedy zhodnotil současné alternativy: Realm,⁵² Churchtrac⁵³ a Breeze.⁵⁴ Jedná se o softwary pro církevní správu, které jsou podle zprávy webu Capterra.com⁵⁵ hodnoceny jako jedny z nejlepších. Společnost Capterra zpracovává každý rok hodnocení softwaru v různých softwarových kategoriích. Jednotlivé produkty pak řadí podle počtu hodnocení na jejich stránce a počtu vyhledání příslušných klíčových slov ve vyhledávačích, patřících ke každému produktu.⁵⁶

2.6.1 Rozhodovací tabulka ohledně informačních systémů

Níže uvedená tabulka porovnává jednotlivá hotová řešení v kontrastu s požadavky na funkčnost daného systému.

Jedním z hlavních požadavků je, aby byl program v češtině, a to v řešení programu nelze. Nedostatkem dalších řešení je absence češtiny v uživatelském rozhraní. Dále jsou systémy Breeze a ChurchTrack primárně zaměřeny pro americký trh. Obě ceny jsou placeny v dolarech, cena se tudíž bude každý měsíc měnit podle převodního kurzu české koruny na americký dolar. Další nevýhodou je, že produkty jsou nabízeny jako služby

⁵¹ MOORHEAD., Jonathan, *FWD: Ministry Responsibilities* [E-mailová komunikace]. 26.01. 2021 8:31

⁵² Church Management Software Platform | Realm by ACS Technologies. Church Management Software Solutions | ACS Technologies [online]. Florence: ACS Technologies, ©2020, 2020 [cit. 2021-01-28]. Dostupné z: <https://www.acstechnologies.com/realm/>

⁵³ ChurchTrac: Church Management Software for Small and Large Churches [online]. Jacksonville: Churchtrac Software, [2020] [cit. 2021-01-28]. Dostupné z: <https://www.churchtrac.com/>

⁵⁴ Breeze Church Management Software: The World's Easiest ChMS [online]. Breeze, ©2021 [cit. 2021-01-28]. Dostupné z: <https://www.breezechms.com/>

⁵⁵ TOP 20 CHURCH MANAGEMENT SOFTWARE. Capterra.com [online]. Arlington: Capterra, 2020 [cit. 2021-01-27]. Dostupné z: <https://www.capterra.com/church-management-software/#top-20>

⁵⁶ Capterra research methodologies: Capterra's Top 20 Report Methodology. Capterra.com [online]. Arlington: Capterra, [2020], [2020] [cit. 2021-01-27]. Dostupné z: <https://blog.capterra.com/research-methodologies/>

a jsou placeny formou pravidelných poplatků, u kterých není veliká záruka, že se cena nebude měnit.

Nejvyšší prioritou pro organizaci je právě prostředí dostupné v českém jazyce, a to v žádné z již hotových řešení není možné. V žádném z řešení není možné přidat vlastní překlad.

Informační systém	Požadavky	Váha požadavku	Realm	Breze	ChurchTrack	Vlastní vývoj
Musí mít	uživatelské prostředí v češtině	10	Ne	Ne	Ne	Ano
	upozornění na změny pro vedoucí jednotlivých částí	9	Ano	Ano	Ano	Ano
	sdílení informací o nedostupnosti uživatele ve vztahu k jednotlivým událostem	9	Ano	Ano	Ano	Ano
	emailové notifikace posílané z informačního systému jednotlivým uživatelům	9	Ano	Ano	Ano	Ano
	plánování události a možnost přiřadit odpovědné osoby/uživatele ke konkrétní části události	10	Ano	Ano	Ano	Ano
	mechanismus kontroly, zda odpovědná osoba může vykonávat danou činnost	10	Ano	Ano	Ano	Ano
Měla by mít	přehledný školicí portál a videa	6	Ano	Ano	Ano	??
	možnost připnout k jednotlivým událostem a částem událostí soubory docx a pdf	4	Ano	Ano	Ano	Ano
	dobře definovanou podporu na další dva roky provozu	6	Ano	Ano	Ano	Ne
Bylo by dobré, kdyby měl	synchronizaci s Google kalendářem a dalšími aplikacemi třetích stran	3	Ne	Ano	Ne	Může být
	multijazyčnost (možnost přepnout do angličtiny)	4	Ne	ne	Ne	Může být
Cena	implementaci a licenci/vývoj	10	Nedostupné	0	0	4500000
	roční provoz/ práce	5	Nedostupné	16000	8000	0

Tabulka 8: Rozhodovací tabulka⁵⁷

⁵⁷ Vlastní zpracování

2.7 Shrnutí analýz

V 7S analýze jsme komplexně ohodnotili vnitřní stav společnosti. Zjistili jsme hlavní tvrdé i měkké stránky společnosti. Dochází zde k jasnému nedostatku v oblasti technologických řešení. Toto nedostatečné technologické řešení má sloužit k organizaci událostí, setkání a bohoslužeb, což je jedním z hlavních cílů společnosti.

Mezi hlavní nevýhody technologického řešení patří nepřehlednost, nedostatečné zabezpečení, absence notifikací uživatele, absence uživatelských práv a jejich ověření, absence automatické kontroly uživatele.

Tyto nedostatky nelze řešit jednoduchou úpravou současného systému, práce by se pak stala těžkopádná. Uživatelé by mohli narušit některou z důležitých částí programu a tím způsobit jeho nefunkčnost. Na základě těchto zjištění byl proveden průzkum trhu a byla vybrána kvalitní levná řešení. Žádné z těchto řešení však nesplňovalo všechna naše očekávání, a proto bude nejlepší navrhnout tvorbu vlastní webové aplikace, která bude plnit funkci informačního systému dané společnosti.

Cena tohoto řešení je sice větší než jakékoliv další dostupné řešení, ale vzhledem k tomu, že se jedná o opěrný bod v oblasti technologií, je tato investice dobrou volbou.

3 Vlastní návrh řešení, přínos práce

3.1 Technické řešení

3.1.1 Úvod

Aplikace se bude zakládat na principech pětivrstvé architektury, rozdělené podle logiky jednotlivých částí. Již dlouhou dobu je zvykem⁵⁸ dělit aplikace na tři vrstvy, podle hardwaru, na který jsou vázané – nejčastěji tak dochází k třívrstvému modelu, databáze (databázový server), serverové části (aplikační server) a uživatelského rozhraní (zařízení uživatele). Uživatel komunikuje skrze uživatelské prostředí se serverovou částí a ta následně provádí operace nad databází informačního systému a zasílá požadovaná data zpět.

Z třívrstvé verze se postupně vyvíjejí další vrstvené architektury, jednou pro C#.NET typickou architekturou je architektura nazývaná jako čistá nebo jako cibulová,⁵⁹ kde jednotlivé vrstvy nemají znatelnou návaznost na hardware jako v případě třívrstvého modelu. Jednotlivé vrstvy jsou logického charakteru. Tyto vrstvy vytváří vlastní abstrakci databázových vztahů mezi třídami. Tyto třídy jsou již navíc zakomponovány uvnitř jedné z vrstev, a proto je řešení ve velké míře nezávislé na napojených externích databázích, které je možné měnit, jak po stránce technologické (změna z MS SQL na Postgre databázový server), tak i fyzické (výměna směrování z jednoho serveru na jiný.)⁶⁰

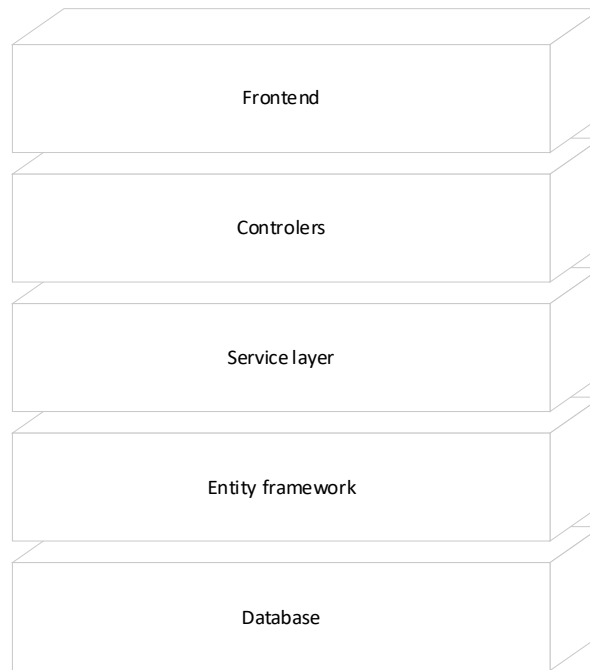
Kromě toho je možné měnit i formu zobrazení jednotlivých výstupů z controler vrstvy. S naší serverovou aplikací může komunikovat jak webové rozhraní, tak i desktop aplikace nainstalovaná na zařízení nebo další externí software. Díky této funkcionalitě je možné přidávat libovolné množství rozhraní pracující se stejnou aplikací nad stejnou databází.

V dalším popisu budu pokračovat podle níže uvedeného nákresu (Obrázek 8).

⁵⁸ Multitier architecture: Three-tier architecture. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2021, 2021 [cit. 2021-04-20]. Dostupné z: https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture

⁵⁹ Common web application architectures: Clean architecture. Microsoft Docs [online]. Washington: Microsoft, ©2021, 12.1.2020 [cit. 2021-04-20]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures#clean-architecture>

⁶⁰ The Clean Architecture. The Clean Code Blog [online]. Washington: Robert C. Martin (Uncle Bob), [2012], 2012 [cit. 2021-04-20]. Dostupné z: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>



Obrázek 8: Pětivrstvá architektura⁶¹

3.1.2 Frontend

Frontend v češtině znamená uživatelské rozhraní. Je to část aplikace, která je spouštěná na uživatelské straně v internetovém prohlížeči. Pro následný projekt zvolíme HTML a AngularJS framework a Bootstrap knihovnu. Dalšími možnostmi, které bychom mohli využít, jsou samostatně stojící aplikace. Na každou platformu by bylo potřeba vytvořit vlastní aplikaci, tato cesta by celou práci několikrát prodražila a vyžadovala by podporu a tvorbu aktualizací pro každou platformu zvlášť. Webové rozhraní je tak jedním z nejlevnějších a nejjednodušších řešení, které je hodně rozšířené. Bootstrap i AngularJS nabízí možnost tvorby responzivního uživatelského prostředí – se změnou velikosti obrazovky například z počítače na telefon nebo tablet se změní velikost a uspořádání jednotlivých položek na prohlížené webové stránce.⁶²

Uživatelské rozhraní poté komunikuje s jednotlivými ovladači (controllery) pomocí REST API. REST je zkratkou pro „Representational State Transfer“ popisující styl architektury pro tvorbu informačních systémů založených na hyper médiích a je plněn

⁶¹ Vlastní zpracování

⁶² Common client-side web technologies. Docs.microsoft.com [online]. Washington: Microsoft, [2021], 2020 [cit. 2021-04-21]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-client-side-web-technologies>

nezávislý na jakémkoliv protokolu přenášejícím tyto informace. V základu se však nejčastěji jako nosič používá HTTP protokol. Výhodou REST je, že vyhovuje otevřeným standardům a neupravuje API. Díky tomu lze využít jakékoliv webové aplikace, která dokáže posílat HTTP dotazy a přijímat HTTP odpovědi.

API koncových bodů (application program interface) pracuje tím způsobem, že uživatelské prostředí posílá požadavky na informace serverové aplikaci a ta odesílá odpověď na jejich požadavky. Základními požadavky na server jsou GET, POST, PUT, PATCH, a DELETE.⁶³

Důležitou vlastností REST API je jeho bezstavovost. To znamená, že neuchovává žádné informace nebo stav ohledně stavu spojení klienta se serverem. Veškeré požadavky od klienta serveru musí obsahovat všechny informace, které jsou pro výkon požadavku serverem potřebné. Nelze tak využít žádných načtených dat uložených na serveru. V důsledku je tedy klient plně zodpovědný za ukládání a práci se všemi informacemi týkající se stavu aplikace. Díky tomu je serverová strana zbavena nutnosti ukládat informace o jednotlivých spojeních s klienty. Bezstavovost je výhodná v první řadě proto, že není potřeba, aby dotazy jednoho klienta směřovali vždy na jeden konkrétní server s uloženými informacemi o spojení, ale mohou se dotazovat na jakýkoliv server dané aplikace. Z druhé strany serveru nemusí obsahovat rozsáhlou logiku pro nakládání s dotazy. Zatřetí bezstavovost je výhodná, protože je jednoduché rozhodovat, které odpovědi si má klient pamatovat a které ne. Tím se zvyšuje efektivita a rychlost aplikací, protože nezáleží na kontextu předchozí komunikace. Začtvrté server nikdy neztratí přehled o tom, v jaké části aplikace se klient nachází, protože tyto informace se vždy posílají dohromady v novém dotazu.⁶⁴

V současné chvíli je řešení testováno pomocí testovacího prostředí Swagger, ve kterém je možné testovat jednotlivé controlery a jejich funkcionalitu, která se projeví i v externí databázi. Je tak možné přes testovací prostředí otestovat přihlášení a poté se přes Microsoft SQL management studio tool podívat na tabulky a výsledný formát uložených dat.

⁶³ Web API design. Docs.microsoft.com [online]. Washington: Microsoft, ©2021, 2018 [cit. 2021-04-21]. Dostupné z: <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design>

⁶⁴ Statelessness in REST APIs. REST API Tutorial [online]. restfulapi.net, ©2020, 2020 [cit. 2021-4-29]. Dostupné z: restfulapi.net/statelessness/

Ve výsledném informačním systému bude potřeba několik modulů stránek, které budou nabízet práci vždy pouze určité informace. V následující tabulce popíšu, jaké jednotlivé zobrazení a práce budou jednotlivé moduly používat.

Název Modulu	Popis
Login	Přihlašování, žádost o obnovení hesla.
NewLogin/PasswordSetup	Nastavení nového hesla nebo prvotní přihlášení do systému.
UserSetup	Přihlašovací formulář pro úvodní vyplnění nezbytných dat o uživateli.
Dashboard	Úvodní obrazovka po přihlášení do systému, zmenšený náhled na kalendář a výpis všech budoucích činností v rámci jednotlivých událostí.
Calendar	Kalendář, ve kterém bude zobrazený měsíční kalendář s nadcházejícími událostmi, možnost přidat data ve, kterých je jednotlivec nedostupný, možnost přejít na náhled na jednu událost.
EventList	Seznam jednotlivých vytvořených událostí, u kterých je vidět výčet všech činností a zodpovědných osob za jednotlivé činnosti. Možnost přidávat a odebírat jednotlivé činnosti z událostí a měnit zodpovědné osoby.
EventEdit	Náhled pouze na jednu událost, lze prohlížet nebo editovat událost v plném rozsahu.
EventPartList/edit	Seznam všech činností, jejich popis a zobrazení vedoucích jednotlivých služeb. Určití uživatelé mohou přidávat, měnit a odebírat jednotlivé činnosti.
UserClaimManagement	Modul otevřený pouze pro uživatele s příslušnými právy. Hlavní možností je přidávat a odebírat práva uživatelů pro práci v databázi.
UserSettings	Nastavení uživatele a editace emailové adresy, změna hesla, a všech informací ohledně uživatele.
UserManagement	Další z nástrojů pro administrátora umožňuje posílat pozvánky uživatelům a seskupovat uživatele do rodin. Přidávat uživatelům možnost nést určité zodpovědnosti.

Tabulka 9: Tabulka plánovaných modulů⁶⁵

3.1.3 API – end points (koncové body)

API koncové body jsou přístupové body serverových aplikací, ve kterých definujeme, jakým způsobem je možno s jednotlivými příkazy nakládat, popřípadě které servisní nebo kontrolní panely můžeme používat. Každý koncový bod API controller definuje, jaké

⁶⁵ Vlastní zpracování

operace budou na serveru provedeny. Využívá přitom pouze ty funkcionality definované v service interface

3.1.4 Service layer

Servisní vrstva definuje jednotlivá pravidla, podle kterých se bude s daty nakládat, a jak bude vypadat například průběh ověřování hesla nebo vytvoření nové události. Definujeme takzvanou business logiku samotné aplikace. Servisní vrstvu je možné rozdělit na dvě podčásti, Service interface (rozhraní služeb) a implementaci service.

Interface deklaruje jaké všechny veřejné metody určité třídy lze využít a zda jsou dostupné pro API controller. Podle této deklarace potom můžeme určit jednotlivé metody a začít implementovat (definovat) jejich průběh.⁶⁶

Implementaci jednotlivých metod – jejich definice, tedy to, jak budeme nakládat s daty, budeme zadávat do oddělených souborů. Platí pravidlo, že by neměly existovat v implementaci veřejné metody, které nebudou adresovány v části jejich rozhraní (interfaces).

Toto rozdělení implementace od rozhraní vyplývá z jednoho základního principu nazývaného „Dependency Injection“, který popisuje možnost oddělení jednotlivých částí do menších nezávislých celků, na kterých mohou pracovat různé skupiny programátorů. Někteří definují jednotlivé metody a další je používají při vytváření REST API ovladačů, přičemž nemusí rozumět přesnému průběhu jednotlivých metod. Stačí, že ví, jak se daná metoda adresuje a jaké jsou její vstupní a výstupní hodnoty.⁶⁷

3.1.5 Entity Framework

Oficiální definicí Entity Framework je: „Entity Framework je objektově-relační mapovač, který uschopňuje vývojáře k práci s databází prostřednictvím .NET objektů. Tím se minimalizuje psaní většiny příkazů potřebných k získání dat z databáze, který by bylo jinak nutné zapsat.“⁶⁸

⁶⁶ .NET Interface-based Programming. Code Magazine [online]. Houston: EPS Software, 1993, 2019 [cit. 2021-04-21]. Dostupné z: <https://www.codemag.com/article/0205091/.NET-Interface-based-Programming>

⁶⁷ Dependency Injection for dummies? Medium [online]. Medium, [2021], 8. 5. 2020 [cit. 2021-4-29]. Dostupné z: medium.com/flawless-app-stories/dependency-injection-for-dummies-168dad181a3d

⁶⁸ What is Entity Framework? Entity Framework Tutorial [online]. EntityFrameworkTutorial.net, [2020] [cit. 2021-04-21]. Dostupné z: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

Entity Framework vrstva vytváří abstrakci datového modelu. Díky tomu je datový model jakoby implementován přímo v samotné serverové aplikaci. Tuto část můžeme vytvořit buď vygenerováním modelu z již hotové databáze, nebo vlastnoručním přepisem jednotlivých tříd z databáze do entitní vrstvy, anebo budeme definovat jednotlivé třídy v serverové aplikaci a poté budeme pomocí nástrojů pro migraci převádět náš modul do externí databáze.⁶⁹

Při tvorbě této práce využijeme metodu code-first. Právě proto, že můžeme nejvíce využít potenciálu předprogramovaných knihoven .NET a principů objektově orientovaného programování. Nemusíme již vytvářet veškeré třídy a všechny atributy tříd. Vytvoříme pouze třídu, která bude dědit vlastnosti po některé z předprogramovaných tříd, která obsahuje nejlepší praktiky využívané v dnešních informačních systémech a odpadá tak práce s mnoha atributy. K těmto podděděným atributům již poté přiřadíme vlastní atributy, které jsou pro naše řešení výjimečné.⁷⁰

Přes Entity Framework je možné provádět transakce nad databází. Zadávat příkazy SQL INSERT, UPDATE, DELETE. Zahrnuje v sobě již vestavěná pravidla, která jsou připravena ke konfiguraci modelu.⁷¹

3.1.6 Database

Tato vrstva je v současném řešení chápána jako externí. Slouží pouze pro ukládání dat. Budeme využívat Microsoft SQL server 2019⁷² a k jeho správě a pro kontrolu a ověření propojení serveru s databází Microsoft server management studio 2018⁷³.

⁶⁹ Entity Framework Core. Docs.microsoft.com [online]. Washington: Microsoft, ©2021, 20.9.2020 [cit. 2021-04-21]. Dostupné z: <https://docs.microsoft.com/en-us/ef/core/>

⁷⁰ What is Code-First? Entity Framework Tutorial [online]. EntityFrameworkTutorial.net, ©2020, [2020] [cit. 2021-04-21]. Dostupné z: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>

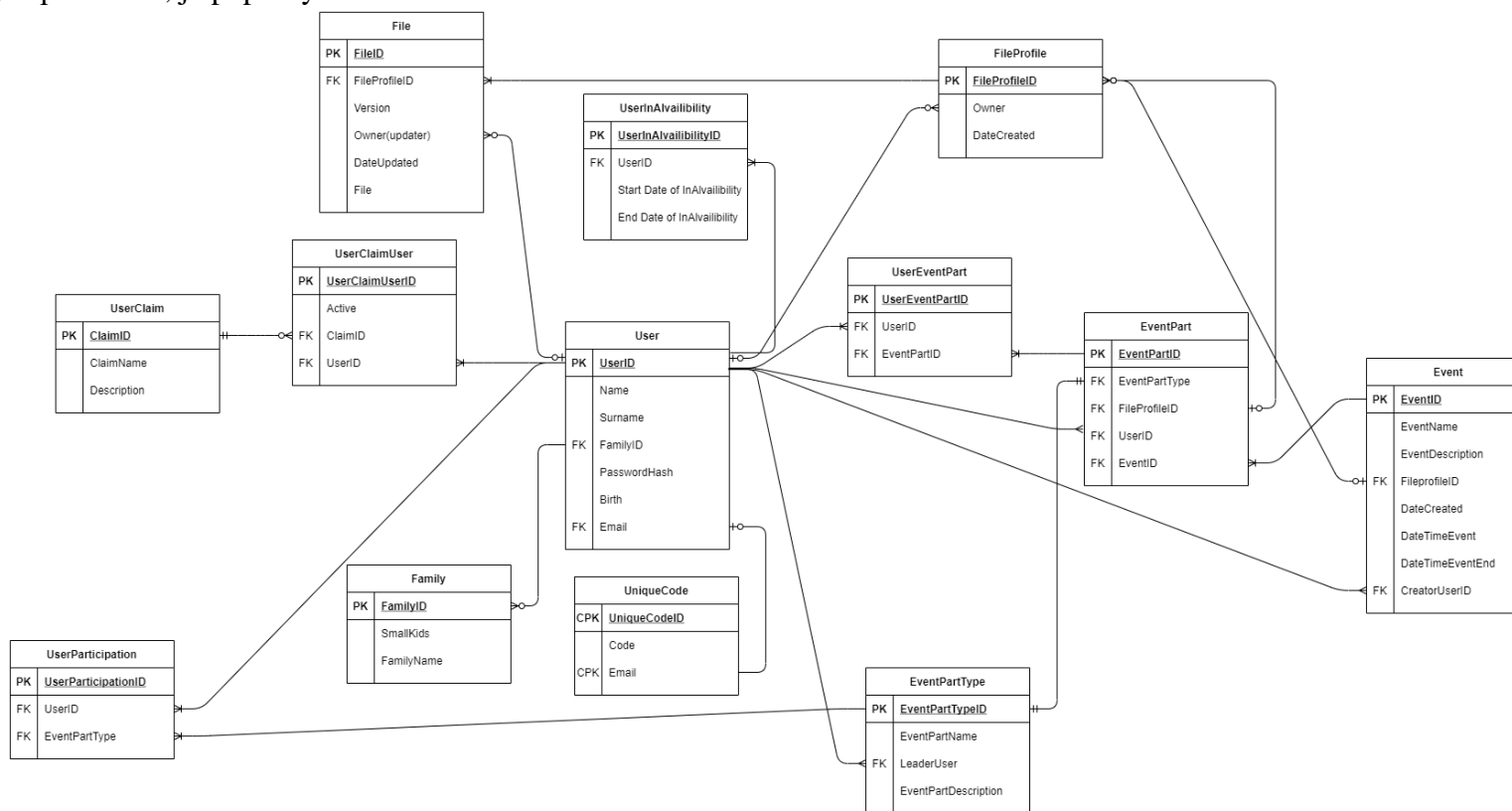
⁷¹ What is Entity Framework? Entity Framework Tutorial [online]. EntityFrameworkTutorial.net, [2020] [cit. 2021-04-21]. Dostupné z: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

⁷² Dostupné na: <https://www.microsoft.com/en-us/windows-server>

⁷³ Zdarma ke stažení zde: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

3.1.7 Datový model

Výsledný datový model, který obsahuje všechny potřebné třídy, aby byly objekty schopné vztahů mezi sebou a funkčně odpovídaly výše uvedeným procesům, je popsán zde:



Obrázek 9: Relační schéma databáze⁷⁴

⁷⁴ Vlastní zpracování

K datovému modelu následuje dokumentace, ve které jsou popsány jednotlivě zvolené datové typy navržené tak, aby byla udržena datová čistota. Datové typy jsou popsány podle standardních datových typů používaných databází Microsoft SQL. Tyto datové typy však nemají vždy přímý ekvivalent v rámci technologie C#. Samotnou dokumentaci popíšeme v databázových typech, protože pro migraci a překlad ze C# do databázových typů není náročné nadefinovat konverzi mezi jednotlivými požadavky.

V prvním sloupci tabulky každé třídy je na prvním místě název třídy. Zbytek popisuje, které atributy jsou primární klíče (PK), cizí klíče (FK) nebo složený primární klíč (PCK).

User	Attributes	Data Type	NotNull Condition
PK	UserID	int	NotNull
	Name	varchar(20)	NotNull
	Surname	varchar(30)	NotNull
FK	FamilyID	int	
	PasswordHash	varchar(256)	NotNull
	Birth	date	
FK	Email	varchar(50)	NotNull

Tabulka 10: Třída User⁷⁵

Třída User je jednou z hlavních tabulek. Většina ostatních tabulek přejímá z této tabulky primární klíč *UserID* datového typu integer.

Další specifickou částí této tabulky je *PasswordHash*, ve které je zašifrované heslo uživatele. Po zašifrování již nelze zobrazit heslo původní, ale ASP.NET aplikace heslo zašifruje a porovná se zašifrovaným heslem v databázi. Současný algoritmus, který se používá pro šifrování v rámci třídy identity user, se jmenuje „ASP.NET Core Identity Version 3“⁷⁶.

Email je složenou součástí složeného primárního klíče v tabulce unique token. Tuto tabulku používáme k autentizaci uživatele při prvním přihlášení nebo obměně hesla. K emailu se vytvoří a přiřadí 16bitový kód, tento kód je zároveň zaslán na uživatelský email. Uživatel se pomocí kliknutí na odkaz s příponou tohoto kódu dostane na stránku umožňující změnu hesla a následné přihlášení do informačního systému.

⁷⁵ Vlastní zpracování

⁷⁶ Exploring the ASP.NET Core Identity PasswordHasher. Andrew Lock | .NET Escapades [online]. Andrew Lock | .NET Escapades, ©2021, 24.10.2017 [cit. 2021-04-21]. Dostupné z: <https://andrewlock.net/exploring-the-asp-net-core-identity-passwordhasher/>

Poslední atribut, který potřebuje vysvětlení, je *familyID*. Účelnost tohoto atributu spočívá především v možnosti seskupovat uživatele do rodin, u kterých budeme udržovat informaci, zda mají menší děti. V případě, že by ve **třídě family** byl atribut *SmallKids* roven 1 a oba manželé měli zodpovědnost na některé z události. Je následně potřeba, aby potvrdili, že následná kombinace bude fungovat. Upozornění na tuto skutečnost dojde již ve chvíli, kdy vedoucí dané části události přiřadí osobu ke kolizní události.

UserInAvailability	Attributes	DataType	NotNull Condition
PK	UserInAvailabilityID	int	NotNull
FK	UserID	int	NotNull
	StartDateOfInAlvailability	date	NotNull
	StartDateOfInAlvailability	Date	NotNull

Tabulka 11: Třída UserAlvailability⁷⁷

Třída UserAvailability ukládá informace o uživateli, kdy není možné využít jeho služby. Dochází tak ke snížení toku informací mezi vedoucím a osobou způsobilou k výkonu práce.

UserParticipation	Attributes	DataType	NotNull Condition
PK	UserParticipationID	Int	NotNull
FK	UserID	Int	NotNull
FK	EventPartTypeID	Int	NotNull

Tabulka 12: Třída UserParticipation⁷⁸

Třída UserParticipation udržuje vztahy o způsobilosti jednotlivých uživatelů k výkonu jednotlivých činností a popisují, zda může uživatel nést zodpovědnost spojenou s výkonem určité činnosti.

UniqueCode	Attributes	DataType	NotNull Condition
CPK	UniqueCodeID	Int	NotNull
CPK	Email	varchar(50)	NotNull
	Code	varchar(256)	NotNull

Tabulka 13: Třída UniqueCode⁷⁹

Třída UniqueCode je třída vytvořena pro obnovu hesel a prvotní přihlášení uživatele do systému. Uživateli na zadaný email dojde zpráva s prosbou o přihlášení do systému skrze hyper-textový odkaz obsahující kód ověřující uživatele, ke kterému bude probíhat buď vytvoření profilu nebo změna hesla. Obsahuje složený primární klíč složený z umělého

⁷⁷ Vlastní zpracování

⁷⁸ Vlastní zpracování

⁷⁹ Vlastní zpracování

identifikátoru *UniqueCode* generovaného pro každý nový kód a z emailové adresy – *Email* – pro kterou je daný kód vytvořený.

Family	Attributes	DataType	NotNull Condition
PK	FamilyID	Int	NotNull
	SmallKids	bit(boolean)	NotNull
	FamilyName	varchar(30)	NotNull

Tabulka 14: Třída Family⁸⁰

Třída Family sdružuje dohromady manželské páry, které mají malé děti, ale chtějí se zapojovat do chodu jednotlivých událostí. Tato třída je více popsána výše u popisku **třídy User**.

UserClaimUser	Attributes	DataType	NotNull Condition
PK	UserClaimUserID	int	NotNull
FK	ClaimID	int	NotNull
FK	UserID	int	NotNull
	Active	bit(boolean)	NotNull

Tabulka 15: Třída UserClaimUser⁸¹

Třída UserClaimUser bude upravovat jednotlivé oprávnění uživatele pro operace v rámci informačního systému a budou přidělovány systémovým administrátorem buď před prvním přihlášením uživatele do systému – při vytvoření unikátního kódu pro přihlášení zasláný na email, nebo kdykoliv později podle potřeby v samostatném modulu.

UserClaim	Attributes	DataType	NotNull Condition
PK	ClaimID	int	NotNull
	ClaimName	varchar	NotNull
	Description	text	

Tabulka 16: Třída UserClaim⁸²

Třída UserClaim tvoří seznam všech oprávnění v rámci informačního systému, popisuje význam a uplatnění jednotlivých oprávnění práce v databázi.

⁸⁰ Vlastní zpracování

⁸¹ Vlastní zpracování

⁸² Vlastní zpracování

File	Attributes	Data Type	NotNull Condition
PK	FileID	int	NotNull
FK	FileProfileID	int	NotNull
	Version	smallint	NotNull
FK	Owner	int	NotNull
	DateUpdated	datetime	NotNull
	File	varbinary(max)	NotNull

Tabulka 17: Třída File⁸³

V rámci informačního systému, bude možné připnout k události – **Event** – nebo k samotné činnosti (**EventPart**) dokument. Krom informací ohledně uživatele, který objekt vytvořil a náležitostí souvisejících s vytvořením, bude možné sledovat i informace ohledně postupných změn v daném dokumentu. Pro dokument se vytvoří samostatný profil, ke kterému budou později přiřazovány změny, včetně změn v názvu, protože primárním klíčem mezi profilem a samotnou verzí souborů je pouze *FileId*, které jsou nezměnitelné.

FileProfile	Attributes	Data Type	NotNull Condition
PK	FileProfileID	int	NotNull
FK	Owner	int	NotNull
	DateCreated	datetime	NotNull

Tabulka 18: Třída FileProfile⁸⁴

Třída **FileProfile** a její funkce je popsána výše u popisu **třídy File**.

Event	Attributes	Data Type	NotNull Condition
PK	EventID	int	NotNull
	EventName	varchar(50)	NotNull
	EventDescription	varchar(max)	
FK	FileProfileID	int	
	DateCreated	datetime	NotNull
	DateTimeEvent	datetime	NotNull
	DateTimeEventEnd	datetime	NotNull
FK	CreatorUserID	int	NotNull

Tabulka 19: Třída Event⁸⁵

Třída Event je jedna z hlavních tříd. Ukládá informace o názvu a datu samostatné události. Dále je možné k události přiřadit soubor s dodatečnými informacemi, vyplnit popis události a kdy bude událost začínat a končit.

⁸³ Vlastní zpracování

⁸⁴ Vlastní zpracování

⁸⁵ Vlastní zpracování

EventPart	Attributes	DataType	NotNull Condition
PK	EventPartID	int	NotNull
FK	EventPartType	int	NotNull
FK	FileProfileID	int	
FK	UserID	int	NotNull
FK	EventID	int	NotNull

Tabulka 20: Třída EventPart⁸⁶

Třída **EventPart** popisuje jednu činnost v rámci celé události. Propojuje uživatele s událostí a určuje jakou činnost bude mít uživatel na zodpovědnost. Zároveň je také možné k této samotné činnosti přiřadit dokument, ve kterém jsou přidány bližší informace pro zodpovědného jednotlivce, například pro zvukaře a livestream je možno doplnit zvláštní požadavky pro danou událost.

EventPartType	Attributes	DataType	NotNull Condition
PK	EventPartTypeID	int	NotNull
	EventPartName	varchar(30)	NotNull
FK	LeaderUser	int	NotNull
	EventPartDescription	text	

Tabulka 21: Třída EventPartType⁸⁷

Třída **EventPartType** popisuje název a popis jednotlivých činností využitelných napříč událostmi.

UserEventPart	Attributes	DataType	NotNull Condition
PK	UserEventPartID	int	NotNull
	UserID	varchar(30)	NotNull
FK	EventPartID	int	NotNull

Tabulka 22: Třída UserEventPart⁸⁸

Třída **UserEventPart** řeší dekompozici vazby M:N mezi EventPart a User. Díky této vazbě může být k jedné části/zodpovědnosti přiřazeno více osob.

3.2 Implementace

V této podkapitole popíšu, jakým způsobem dojde k implementaci informačního systému a jeho následné vylepšování. Na tuto kapitolu navážu plánem tvorby informačního systému.

⁸⁶ Vlastní zpracování

⁸⁷ Vlastní zpracování

⁸⁸ Vlastní zpracování

3.2.1 Testování a quality management

Dnes existují dvě hlavní metodologie přístupu k vývoji softwaru. Tradičním přístupem je metoda waterfall, kdy se postupně přechází z jedné části tvorby na další – od zjištění požadavků společnosti přes vývoj, testování, úpravu zjištěných chyb až k finální implementaci a údržbě. Jednotlivé fáze trvají delší dobu a jednotlivé kroky se po jejich dokončení nedají příliš změnit. Oproti tomu moderním přístupem je Agile metodologie, která rozděluje jednotlivé úkoly na menší celky, které jsou průběžně testované a implementované. Tím je zajištěno, že se při vývoji vytváří to, co je pro zákazníka nejlépe využitelné. Waterfall metoda je levnější a časově méně náročnější. Agilní metoda a její iterační postup sice zaručuje kvalitnější práci, ale je nákladnější.⁸⁹

Aplikaci tedy v našem případě vytvoříme nejprve celou podle zadaných požadavků a poté ve dvou cyklech. Během implementace otestujeme a případně upravíme její funkčnost.

Testování bude probíhat poskytnutím přístupu do systému dobrovolníkům na technických pozicích, kteří jsou s podobnými systémy zvyklí pracovat a mají alespoň základní znalosti programování. Po jednom měsíci testovacího provozu budou vyladěny technické nedokonalosti a začne probíhat zaškolování celé skupiny klíčových uživatelů.

V průběhu celého tohoto procesu budou moci uživatelé psát na emailovou adresu podpory nebo budou moci využít pop-up formulář. V obou případech bude potřeba, aby uživatelé vždy zanechali svůj email pro zpětný kontakt a dořešení případných nejasností.

Pro případné zlepšení kvality bude pop-up formulář přístupný i po skončení spouštěcí fáze. Kromě toho bude jednou ročně rozeslán formulář, ve kterém se znovu proaktivně zeptáme, jaké možnosti změn by uživatelé uvítali. Zároveň bude rozeslán se zprávou o tom, jaké uživatelské návrhy byly použity, jaké ne a proč.

Cílem této metody je, aby se nestala podpora otravnou a uživatelé byli motivováni k interakci a návrhům na změny.

⁸⁹ Agile Vs Waterfall: Know the Difference Between Methodologies. Meet Guru99: Free Training Tutorials and Video for IT Courses [online]. Guru99, ©2021, [2021] [cit. 2021-04-21]. Dostupné z: <https://www.guru99.com/waterfall-vs-agile.html#:~:text=Waterfall%20vs%20Agile%20Key%20Difference&text=Agile%20performs%20testing%20concurrently%20with,once%20the%20project%20development%20starts.>

3.2.2 Zpracování strategie a zaškolení uživatelů

V první řadě proběhne skupinové zaškolení klíčových uživatelů pomocí společného setkání a rychlý přehled celého systému. Následně budou mít uživatelé možnost si vyzkoušet práci v systému nanečisto a ptát se otázky. Další doškolení bude probíhat formou individuálních dotazů.

Jak pro klíčové uživatele, tak i pro uživatele koncové bude vytvořen společný video tutoriál, který bude dobře strukturovaný a pro lepší orientaci mezi jednotlivými tutoriály bude sloužit samostatná stránka.

Interface systém jako takový bude dobře dokumentovaný a u všech důležitých oblastí budou info-hover button prvky, které zobrazí nápovědu a případný odkaz na video, které více podrobně vysvětlí aktuální problematiku.

Koncovým uživatelům bude systém představen formou rychlé video prezentace, která bude součástí registračního formuláře a nepůjde přeskočit. Po přehrání budou moci vyzkoušet demo verzi se základními úkoly týkajícími se běžného uživatele. Jde o to, aby se uživatelům již při prvním setkání se systémem dostalo hands-on experience a neztrácel se tak čas na společných ukázkách systému.

3.3 Plán na dokončení informačního systému

Tato podkapitola je souhrnem jednotlivých potřebných dokumentů vytvářených ve fázi zahájení a plánování projektu. Jedná se o důležité manažerské nástroje pomáhající manažerovi lépe kontrolovat průběh tvorby ve fázi tvorby.

3.3.1 Identifikační listina

Název projektu:	Tvorba a implementace informačního systému
Záměr	1. Zvýšení efektivity komunikace 2. Snížení zmatku nejistoty předcházející událostem 3. Vyšší spokojenost dobrovolníků
Cíl projektu	Tvorba informačního systému ve webovém rozhraní.
Výstupy projektu	1. Nový informační systém

Plánované interní náklady	40 000 Kč včetně rezerv
Plánované externí náklady	350 000 Kč (externí programátoři včetně rezerv)
Plánovaný termín zahájení:	5. 7. 2021
Plánovaný termín dokončení:	3.12. 2021
Hlavní milníky	Dokončení vývoje backendové aplikace 2.8.2021 Dokončení uživatelského prostředí 13.8.2021 Testování aplikace 17.8.2021 Dokončení přípravy školicích videí – rozeslání emailů 8.10.2021 Implementace a testování aplikace v ostrém provozu 22.10.2021
Lokalizace projektu	Tým řízení projektu ve fázi implementace projektu i dobrovolníci zapsaného spolku.
Kritéria úspěšnosti	Nepřekročíme plánované náklady Dodržení termínů Celkové kvalitní zpracování
Schválené výjimky	Je možné překročit plánovaný čas dokončení vývoje maximálně do 30.8.2021
Zadavatel projektu:	Vedení organizace
Sponzor projektu:	Vlastní zdroje
Manažer projektu	Josef Sponar
Tým řízení projektu:	Manažer projektu Externí frontend programátor Externí backend programátor

Tabulka 23: Identifikační listina⁹⁰

⁹⁰ Vlastní zpracování

3.3.2 RIPRAN analýza rizik

Poř. číslo rizika	Hrozba	Scénář	P-st	Dopad	Velikost rizika	Návrh na opatření
1	Nedodržení termínu dokončení	Některá část vývoje aplikace se opozdí z důvodů nečekaných technických problémů souvisejících s nestandardními požadavky.	NP	MD	NHR	Akceptace rizika.
2	Překročení nákladů	Při vytváření systému dojde k zjištění, že je potřeba přidat další části, se kterými se původně nepočítalo, ale jsou nezbytně důležité pro další fungování.	SP	VD	VHR	Do plánování zahrnout finanční rezervy
3	Překročení nákladů	Programátor dostane na jeden úkol příliš mnoho času na úkor dalších složitějších úkolů, které poté nebude stíhat. Důsledkem se posune termín dokončení projektu.	SP	SD	SHR	Dostatečná časová rezerva v rámci jednotlivých úkolů a dobré vyměření času kvalifikovanými pracovníky.
4	Nedostupnost pracovníků	Pracovníci v jednotlivých časech nebudou plně dostupní a naše plánování a využití jednotlivých pracovníků nebude odpovídat realitě a projekt se v důsledku opozdí.	VP	SD	VHR	Zavedení standardizované dokumentace kódu pro plynulou náhradu za jiného pracovníka.

5	Nepřehledné uživatelské prostředí	Uživatelé nebudou schopni vyznat se v informačním systému, v důsledku dojde ke snížení efektivity a nárůstu nespokojenosti uživatelů.	NP	VD	SHR	Dobrá komunikace požadavků front-end programátorovi. Výběr kvalifikovaného programátora na základě jeho zkušeností a portfolia
6	Nefunkční systém	Informační systém se začne chovat nepředvídatelně. V případě nahrání vysokého množství událostí za týden nebo velkého množství zodpovědností za činnosti v rámci události.	NP	VD	SHR	V první fázi testování je potřeba vyzkoušet i nadměrné zatížení a zvláštní požadavky.
7	Nekvalitní zpracování IS a nespokojenost uživatelů	Běžní uživatelé nebudou dostatečným způsobem odesílat zpětnou vazbu ve fázi ostrého testování. Což zapříčiní dlouhodobou nekvalitu systému a uživatelskou nespokojenost.	VP	VD	VHR	Vytvořit jednoduchý dotazník pro zpětnou vazbu.

Tabulka 24: RIPRAN analýza⁹¹

Na základě definovaných hrozeb jsme připravili opatření, která zamezí jednotlivým hrozbám. Akceptována byla pouze jediná hrozba a to hrozba číslo jedna – nedodržení termínu dokončení, protože výsledná velikost rizika je hodnocena jako nízká.

⁹¹ Vlastní zpracování

3.3.3 Analýza zainteresovaných stran

Kategorie	Stakeholders	Očekávání	Cíle	Síla příčiny	Ohodnocení
Vlastní	Vedení	Vyšší zapojení dobrovolníků	Oslava Boha Plynulý chod jednotlivých událostí Zjednodušení organizace událostí	Znalost vlastní společnosti Finanční zdroje	9
	Projektový manažer	Přiměřená mzda Další vzdělávání	Oslava Boha Větší spokojenost dobrovolníků	Pracovní nasazení Přehled o projektu	8
	Dobrovolníci (uživatelé informačního systému)	Přehledný IS, který bude pomáhat při službě a nebude rozptylovat	Oslava Boha Zajištění chodu určité oblasti služby	Ochota spolupracovat	3
Externí	Programátoři	Finanční ohodnocení	Maximální finanční ohodnocení	•Kvalita poskytovaných služeb	5

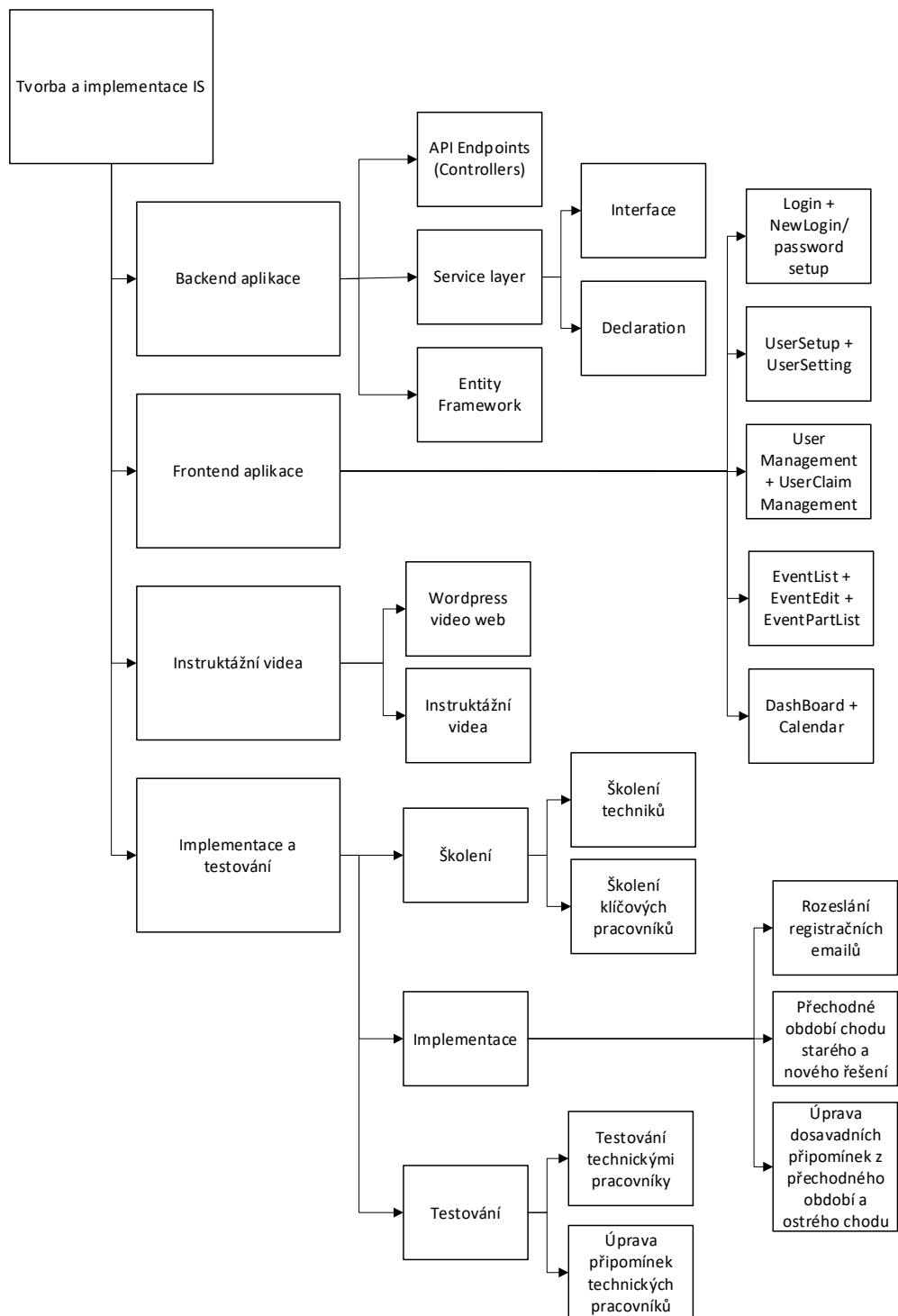
Tabulka 25: Analýza zainteresovaných stran⁹²

3.3.4 Analýza vybavení

Tato analýza je v teoretické části opomenuta, ačkoliv patří do rámce jednotlivých analýz potřebných pro výkon projektu. Analýza vybavení je součástí projektové přípravy především proto, aby se projektový tým donutil zamyslet nad jednotlivými potřebami projektu, které ovlivní výsledný rozpočet. Tuto část nechávám nevyplněnou, protože tento projekt nezahrnuje vybavení, které by bylo potřeba zakoupit, ani vybavení, s jehož využitím by rostla cena projektu.

⁹² Vlastní zpracování, dle: SMOLÍKOVÁ, Lenka. 5. Přednáška, Řízení projektů v ICT: Zainteresované strany. [Brno], 2020.

3.3.5 Work Breakdown Structure



Obrázek 10: Work breakdown structure⁹³

⁹³ Vlastní zpracování

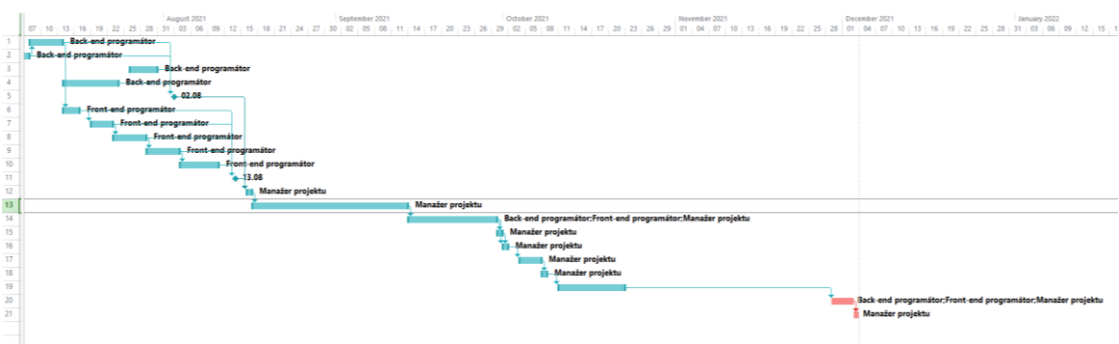
Na obrázku 10 je zpracované hierarchické rozdělení řešení do jednotlivých hlavních částí projektu, které mají být zpracovány. Tyto části jsou podkladem pro tvorbu plánu realizace projektu a jsou dále rozepsány v plánu realizace projektu.

3.3.6 Plán realizace projektu

Číslo	Jméno úkolu	Trvání	Začátek	Konec	Předcházející činnost	Resource Names
1	Vývoj API Endpoints	4 days	Thu 08.07.21	Tue 13.07.21	2	Back-end programátor
2	Vývoj Service Interface	3 days	Mon 05.07.21	Wed 07.07.21		Back-end programátor
3	Vývoj Service declaration	5 days	Mon 26.07.21	Fri 30.07.21		Back-end programátor
4	Entity Framework	8 days	Wed 14.07.21	Fri 23.07.21		Back-end programátor
5	Dokončení Back-end aplikace	0 days	Mon 02.08.21	Mon 02.08.21	1;2;3;4	Manažer projektu
6	Login + NewLogin/password setup	3 days	Wed 14.07.21	Fri 16.07.21	1	Front-end programátor
7	UserSetup + UserSetting	4 days	Mon 19.07.21	Thu 22.07.21	6	Front-end programátor
8	User Management + UserClaim	4 days	Fri 23.07.21	Wed 28.07.21	7	Front-end programátor
9	EventList + EventEdit + EventPartList	4 days	Thu 29.07.21	Tue 03.08.21	8	Front-end programátor
10	DashBoard + Calendar	5 days	Wed 04.08.21	Tue 10.08.21	9	Front-end programátor
11	Dokončení Front-End aplikace	0 days	Fri 13.08.21	Fri 13.08.21	6;7;8;9;10	Manažer projektu
12	Školení techniků	1 day	Mon 16.08.21	Mon 16.08.21	5;11	Manažer projektu
13	Testování technickými pracovníky	20 days	Tue 17.08.21	Mon 13.09.21	12	Manažer projektu
14	Úprava připomínek technických pracovníků	12 days	Tue 14.09.21	Wed 29.09.21	13	Back-end programátor;Front-end programátor;Manažer projektu
15	Školení klíčových pracovníků	1 day	Thu 30.09.21	Thu 30.09.21	14	Manažer projektu
16	Wordpress Video Website	1 day	Fri 01.10.21	Fri 01.10.21	14;15	Manažer projektu

17	Instruktažní videa	4 days	Mon 04.10.21	Thu 07.10.21	16	Manažer projektu
18	Rozeslání registračních emailů	1 day	Fri 08.10.21	Fri 08.10.21	17	Manažer projektu
19	Přechodné období současného chodu starého a nového řešení	10 days	Mon 11.10.21	Fri 22.10.21	18	
20	Úprava dosavadních připomínek z přechodného období a ostrého chodu	4 days	Mon 29.11.21	Thu 02.12.21	19	Back-end programátor; Front-end programátor; Manažer projektu
21	Dokončení projektu	1 day	Fri 03.12.21	Fri 03.12.21	20	Manažer projektu

Tabulka 26: Časový plán⁹⁴



Obrázek 11: Gantův diagram časového plánu⁹⁵

V následném projektu je nechána delší doba na testování, protože program se nevyužívá každý den stejně intenzivně. Během týdne bude program spravovat maximálně 5 až 6 událostí, průměrně však asi jen 2 až 4. A k jejich využití se pokaždé nedostanou všichni uživatelé během jednoho týdne. V důsledku toho je fáze úprav dosavadních připomínek z přechodného období a ostrého chodu odsazená o více než jeden měsíc.

⁹⁴ Vlastní zpracování v nástroji MS Project

⁹⁵ Vlastní zpracování v nástroji MS Project

3.3.7 Ekonomické zhodnocení

	Pozice	Denní sazba [Kč]	Odpracovaný počet dnů	Náklady projektu [Kč]	Vyplaceno pracovníkům [Kč]
Interní Pracovníci	Manažer	1500	25	37500	24750
externí zdroje	Front-end programátor	4500	36	162000	162000
	Back-end programátor	5000	36	180000	180000
Celkem [Kč]				379500	366750

Tabulka 27: Ekonomické zhodnocení projektu⁹⁶

Zde je ekonomické zhodnocení celkové práce v korunách českých. Ve sloupečku náklady projektu je zahrnuto kolik korun bude muset investor pokrýt. Celkový součet interních a externích nákladů je 379 500 Kč. Externisté pracují jako OSVČ a nemají pevně stanovenou pracovní dobu a místo výkonu práce, proto zaměstnavatel vyplácí fakturu a neodvádí sociální a zdravotní pojištění.

3.4 Přínosy

Mezi hlavní přínosy práce patří naplánovaný vývoj aplikace jak po stránce technické, tak i projektové. Je vytýčený cíl, kterého je potřeba dosáhnout a podmínky jeho dosažení. Byl stanovený rozpočet pro celou práci. Realizace tohoto projektu je nyní v jednání. Co se týče finančního ohodnocení mého výkonu na této práci. Svůj výkon jsem ohodnotil cenou 50 000 korun českých.

Největším přínosem však pro mě je znovu zopakované znalosti navrhování informačních systémů prohloubené o pochopení objektově orientovaného programování, Entity Frameworku, práce s více vrstvou architekturou a práce s git vývojovým prostředím. Díky těmto znalostem mám přehled o současných trendech programování.

V projektové části věci jsem mohl hlubší pochopení technologií zapracovat do plánování projektu, tvorby jednotlivých analýz a manažerských nástrojů, které jsem předtím využíval pouze teoreticky a na smyšlených projektech, které byly vzdálené realitě z důvodu nedostatečného zvládnutí dané technologie a přehledu o časové náročnosti jednotlivých úkolů.

⁹⁶ Vlastní zpracování

Nejvíce jsem mohl těžit ze spolupráce uvnitř zapsaného spolku Křesťané Kuřim, kde jsem pro svoji práci získal cenné názory od programátorů, business analytiků a podnikatelů, které budu moci uplatnit dále při studiu a práci.

Závěr

V této práci jsem popsal teoretická východiska související s návrhem, tvorbou a implementací informačního systému. Popsal jsem základní technologické náležitosti využití při technologickém návrhu a teoretický popis jednotlivých nástrojů projektového managementu využitých v kapitole návrhu vlastního řešení.

Analýzou vnitřního stavu společnosti a současného řešení vyšlo najevo, že je potřeba vytvořit nový informační systém. Byla provedena analýza hlavních procesů ve společnosti. Pomocí agilní analýzy MOSCOW jsme určili požadavky na informační systém a poté rozhodovací tabulkou porovnali vlastní řešení s již existujícími programy, které by mohly nahradit stávající řešení.

V části vlastního návrhu jsem navrhl webovou aplikaci na technologii .NET framework a popsal jsem moderní přístup k tvorbě počítačových programů. Objasnil jsem základy pětivrstvé architektury využívající principy objektového programování a poukázal jsem na její výhody a nekomplexnost – jednotlivé části mohou být programovány s velice nízkou závislostí na dalších logických částech.

Popis technologie jsem doplnil o projektovou dokumentaci potřebnou pro zahájení projektu. Navrhl jsem plán tvorby projektu, jeho rozložení v čase a výsledné ekonomické zhodnocení práce. V projektu byla zahrnuta i implementace a její nezbytné detaily.

Těmito jednotlivými částmi byly naplněny cíle práce a zhodnoceny její přínosy, které z práce vyplývají.

Seznam použité literatury

Agile Vs Waterfall: Know the Difference Between Methodologies. *Meet Guru99: Free Training Tutorials and Video for IT Courses* [online]. Guru99, ©2021, [2021] [cit. 2021-4-21]. Dostupné z:

<https://www.guru99.com/waterfall-vs-agile.html#:~:text=Waterfall%20vs%20Agile%20Key%20Difference&text=Agile%20performs%20testing%20concurrently%20with,once%20the%20project%20development%20starts>.

A tour of the C# language. *Docs.microsoft.com* [online]. Washington: Microsoft, ©2021, 01/28/2021 [cit. 2021-4-27]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

Breeze Church Management Software: The World's Easiest ChMS [online]. Breeze, ©2021 [cit. 2021-1-28]. Dostupné z: <https://www.breezechms.com/>

Capterra research methodologies: Capterra's Top 20 Report Methodology. *Capterra.com* [online]. Arlington: Capterra, [2020], [2020] [cit. 2021-1-27]. Dostupné z: <https://blog.capterra.com/research-methodologies/>

Common client-side web technologies. *Docs.microsoft.com* [online]. Washington: Microsoft, [2021], 2020 [cit. 2021-4-21]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-client-side-web-technologies>

Common web application architectures: Clean architecture. *Microsoft Docs* [online]. Washington: Microsoft, ©2021, 12.1.2020 [cit. 2021-4-20]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures#clean-architecture>

KOCH, Miloš. *Datové a funkční modelování*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2006. ISBN 80-214-3252-7.

Dependency Injection for dummies? *Medium* [online]. Medium, [2021], 8.5.2020 [cit. 2021-4-29]. Dostupné z: medium.com/flawless-app-stories/dependency-injection-for-dummies-168dad181a3d

Entity Framework Core. *Docs.microsoft.com* [online]. Washington: Microsoft, ©2021, 20.9.2020 [cit. 2021-4-21]. Dostupné z: <https://docs.microsoft.com/en-us/ef/core/>

Exploring the ASP.NET Core Identity PasswordHasher. *Andrew Lock | .NET Escapades* [online]. Andrew Lock | .NET Escapades, ©2021, 24.10.2017 [cit. 2021-4-21]. Dostupné z: <https://andrewlock.net/exploring-the-asp-net-core-identity-passwordhasher/>

Historie vzniku metody. *RIPRAN: Metoda pro analýzu rizik* [online]. Lacko, [2000] [cit. 2021-4-27]. Dostupné z: <https://ripran.cz/historie.html>

Hrozba (Threat). *ManagementMania.com* [online]. ManagementMania.com, © 2011-2016, 17.02.2016 [cit. 2021-4-27]. Dostupné z: <https://managementmania.com/cs/hrozba-threat>

Church Management Software Platform | Realm by ACS Technologies. *Church Management Software Solutions / ACS Technologies* [online]. Florence: ACS Technologies, ©2020, 2020 [cit. 2021-1-28]. Dostupné z: <https://www.acstechnologies.com/realm/>

ChurchTrac: Church Management Software for Small and Large Churches. *ChurchTrac* [online]. Jacksonville: Churchtrac Software, [2020], [2020] [cit. 2021-1-28]. Dostupné z: <https://www.churchtrac.com/index>

SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.

Kontakt: Křesťané Kuřim. *Křesťané Kuřim* [online]. Kuřim: Křesťané Kuřim, ©2021, 2021 [cit. 2021-3-17]. Dostupné z: <https://krestanekurim.cz/kontakt/>

CEJTHAMR, Václav a Jiří DĚDINA. *Management a organizační chování*. 2., aktualiz. a rozš. vyd. Praha: Grada, c2010. Expert (Grada). ISBN isbn978-80-247-3348-7.

MoSCoW Analysis – Prioritising Your Requirements for Delivery. *Business analysis excellence* [online]. Melbourne: Business Analysis Excellence Pty, ©[2021], ©2014 [cit. 2021-3-24]. Dostupné z: <http://business-analysis-excellence.com/moscow-analysis/>

Multitier architecture: Three-tier architecture. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021, 2021 [cit. 2021-4-20]. Dostupné z: https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture

.NET Interface-based Programming. *Code Magazine* [online]. Houston: EPS Software, 1993, 2019 [cit. 2021-4-21]. Dostupné z: <https://www.codemag.com/article/0205091/.NET-Interface-based-Programming>

Objektově orientované programování a evoluce vývoje softwaru. *ITnetwork.cz* [online]. Praha: David Čápka, ©2021, ©2021 [cit. 2021-3-26]. Dostupné z: <https://www.itnetwork.cz/navrh/objektove-orientovane-programovani-a-evoluce-vyvoje-softwaru/>

Organizace. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2019, 2019 [cit. 2021-2-17].

Organization: Definition of Organization by Merriam-Webster. *Merriam-Webster* [online]. Springfield: Merriam-Webster, ©2021, 2021 [cit. 2021-2-17]. Dostupné z: <https://www.merriam-webster.com/dictionary/organization>

Organizational Structure. *Investopedia* [online]. New York: Investopedia, 2020, 2020 [cit. 2021-2-10]. Dostupné z: <https://www.investopedia.com/terms/o/organizational-structure.asp>

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika. 2.*, přeprac. a aktualiz. vyd. Praha: Grada, 2009. Expert (Grada). ISBN 978-80-247-2615-1.

ŘEPA, Václav. *Podnikové procesy: procesní řízení a modelování*. Praha: Grada, 2006. Management v informační společnosti. ISBN isbn80-247-1281-4.

ŘEPA, Václav. *Podnikové procesy: procesní řízení a modelování*. Praha: Grada, 2006. Management v informační společnosti. ISBN isbn80-247-1281-4.

ŠPATENKA, Jan. *Prezentace, Podnikové informační systémy: Analýza vnitřního prostředí 7S*. Brno, [2021]. Dostupné také z: https://moodle.vutbr.cz/pluginfile.php/340649/mod_resource/content/1/03%20-%20Podnikov%C3%A9%20informa%C4%8Dn%C3%AD%20syst%C3%A9my.pdf

Procesní a projektové řízení. *Management.cz* [online]. Management.cz, ©2015, 2014 [cit. 2021-2-17]. Dostupné z: <http://www.management.cz/procesni-a-projektove-rizeni/>

Process. *Merriam-Webster* [online]. Springfield: Merriam-Webster, ©2021 [cit. 2021-2-15]. Dostupné z: <https://www.merriam-webster.com/dictionary/process>

Projektová kritéria (atributy). *PM Consulting* [online]. Choceň: PM Consulting, c[2005-2021], [2018] [cit. 2021-2-17]. Dostupné z: <https://www.pmconsulting.cz/slovníkovy-pojem/projektova-kriteriata-tributy/>

RIPRAN: *Metoda pro analýzu projektových rizik* [online]. Lacko, [2000] [cit. 2021-4-27]. Dostupné z: <https://ripran.cz/>

Stanovy Spolku. Kuřim: Křesťané Kuřim, 2019, 7 s. L 25552/SL1/KSBR. Dostupné také z: <https://or.justice.cz/ias/ui/vypis-sl-detail?dokument=56204859&subjektId=1038458&spis=1153055>

Statelessness in REST APIs. *REST API Tutorial* [online]. restfulapi.net, ©2020, 2020 [cit. 2021-4-29]. Dostupné z: restfulapi.net/statelessness/

Tabulky pro verbální hodnocení rizik: Soustava 3 x 3 x 3. [Brno], [2000]. Dostupné také z: <https://ripran.cz/tab3.pdf>

The Clean Architecture. *The Clean Code Blog* [online]. Washington: Robert C. Martin (Uncle Bob), [2012], 2012 [cit. 2021-4-20]. Dostupné z: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>

TOP 20 CHURCH MANAGEMENT SOFTWARE. *Capterra.com* [online]. Arlington: Capterra, 2020 [cit. 2021-1-27]. Dostupné z: <https://www.capterra.com/church-management-software/#top-20>

Web API design. *Docs.microsoft.com* [online]. Washington: Microsoft, ©2021, 2018 [cit. 2021-4-21]. Dostupné z: <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design>

What is a Gantt chart? *Association for Project Management* [online]. Association for Project Management, ©2020, 2020 [cit. 2021-4-27]. Dostupné z: <https://www.apm.org.uk/resources/find-a-resource/gantt-chart/>

What is a Work Breakdown Structure? [online]. Workbreakdownstructure.com, ©2021 [cit. 2021-4-27]. Dostupné z: <https://www.workbreakdownstructure.com/>

What is Code-First? *Entity Framework Tutorial* [online]. EntityFrameworkTutorial.net, ©2020, [2020] [cit. 2021-4-21]. Dostupné z: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>

What is Entity Framework? *Entity Framework Tutorial* [online]. EntityFrameworkTutorial.net, [2020] [cit. 2021-4-21]. Dostupné z: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

KOTRBA, Tomáš a Miroslav FORET. *Základy managementu*. Znojmo: Soukromá vysoká škola ekonomická, 2009. ISBN isbn978-80-87314-00-5.

MYSLIVCOVÁ, Kateřina. *Základy managementu*. Praha: Soukromá vysoká škola ekonomických studií, [2013]. ISBN isbn978-80-7523-000-3.

KAŠÍK, Josef a Jiří FRANEK. *Základy podnikové diagnostiky*. Ostrava: VŠB-TU Ostrava, 2015. ISBN 978-80-248-3888-5.

SMOLÍKOVÁ, Lenka. *5. Přednáška, Řízení projektů v ICT: Zainteresané strany*. [Brno], 2020.

Seznam tabulek

Tabulka 1: Třídy pravděpodobnosti analýzy RIPRAN	19
Tabulka 2: Tabulka pro vyhodnocení výsledné pravděpodobnosti hrozby.....	19
Tabulka 3: Tabulka pro hodnocení dopadu na projekt.....	20
Tabulka 4: Přiřazení třídy hodnoty rizika	20
Tabulka 5: Popis společnosti.....	28
Tabulka 6: Ukázka ze současného systému	31
Tabulka 7: Požadavky zadavatele	39
Tabulka 8: Rozhodovací tabulka.....	42
Tabulka 9: Tabulka plánovaných modulů.....	47
Tabulka 10: Třída User	51
Tabulka 11: Třída UserAvailability	52
Tabulka 12: Třída UserParticipation	52
Tabulka 13: Třída UniqueCode.....	52
Tabulka 14: Třída Family.....	53
Tabulka 15: Třída UserClaimUser	53
Tabulka 16: Třída UserClaim.....	53
Tabulka 17: Třída File.....	54
Tabulka 18: Třída FileProfile.....	54
Tabulka 19: Třída Event.....	54
Tabulka 20: Třída EventPart	55
Tabulka 21: Třída EventPartType	55
Tabulka 22: Třída UserEventPart.....	55
Tabulka 23: Identifikační listina	58
Tabulka 24: RIPRAN analýza.....	60
Tabulka 25: Analýza zainteresovaných stran.....	61
Tabulka 26: Časový plán.....	64
Tabulka 27: Ekonomické zhodnocení projektu.....	65

Seznam obrázků

Obrázek 1: 7S analýza	22
Obrázek 2: Organizační schéma společnosti	29
Obrázek 3: Diagram vytvoření události.....	36
Obrázek 4: Diagram zrušení události.....	36
Obrázek 5: Diagram změny dobrovolníka za jiného	37
Obrázek 6: Diagram zrušení činnosti v rámci události.....	37
Obrázek 7: Diagram doplnění dobrovolníků pro danou činnost v rámci události.....	38
Obrázek 8: Pětivrstvá architektura.....	45
Obrázek 9: Relační schéma databáze.....	50
Obrázek 10: Work breakdown structure	62
Obrázek 11: Gantův diagram časového plánu	64