



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Modulární webový informační systém pro volnočasový areál
Student:	Marian-Daniel Rolník
Vedoucí:	Ing. Marek Suchánek
Studijní program:	Informatika
Studijní obor:	Informační systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Cílem práce je vyvinout modulární webový informační systém podporující vybrané klíčové oblasti činnosti volnočasového areálu.

1. Zmapujte a popište klíčové činnosti volnočasového areálu RY:BAR Lipno. Při popisu použijte formální konceptuální modelování.
2. Proveďte stručnou rešerši existujících řešení podporujících tyto činnosti.
3. Navrhněte vlastní řešení ve formě modulární webové aplikace, kde moduly budou podporovat jednotlivé vybrané dílčí oblasti činností volnočasového areálu. Během návrhu zohledněte základní principy a koncepty Normalized Systems Theory.
4. Implementujte a následně otestujte navržené řešení. Zdůvodněte výběr programovacího jazyka i případných dalších technologií a nástrojů (například webového frameworku).
5. Zhodnoťte ekonomicko-manažerské náklady a přínosy vlastního řešení a porovnejte jej s již existujícími. Ve zhodnocení se mimo jiné zaměřte na evolvabilitu řešení ve vztahu k návrhu i k možnostem zvolených technologií.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 26. prosince 2017



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Modulární webový informační systém pro volnočasový areál

Marian-Daniel Rolník

Katedra softwarového inženýrství
Vedoucí práce: Ing. Marek Suchánek

14. května 2018

Poděkování

Tímto bych chtěl poděkovat Ing. Marku Suchánkovi za jeho dohled, opravdu skvělý přístup k vedení práce a za jeho čas. Dále bych chtěl poděkovat své rodině za dlouholetou podporu ve studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Marian-Daniel Rolník. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Rolník, Marian-Daniel. *Modulární webový informační systém pro volnočasový areál*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato bakalářská práce se zabývá vývojem informačního systému, který bude podporovat klíčové činnosti volnočasového areálu.

Nejprve se věnuje analýze volnočasového areálu RY:BAR. Na základě této analýzy jsou vytvořeny modely za použití konceptuálního modelování. Následně je podle těchto modelů vytvořena webová aplikace pro volnočasové areály se základními principy Normalized Systems Theory.

Tato webová aplikace je napsána v PHP frameworku Symfony, který podporuje modularitu ve webových aplikacích. Vytvořené řešení poskytuje evolvabilitu při dalším vývoji, který bude následovat.

Přínosem této práce je obecný webový systém, využitelný pro jakýkoliv volnočasový areál s možností zvolení potřebných modulů pro klíčové činnosti areálu. Výhodou systému jsou jeho nízké náklady, plynoucí z ekonomického zhodnocení této aplikace v areálu RY:BAR.

Klíčová slova normalizovaný informační systém, modulární webová aplikace, analýza volnočasového areálu, ekonomicko-manažerské zhodnocení, evolvabilita, Normalized Systems Theory, PHP, ORM, Symfony

Abstract

This bachelor thesis deals with the development of an information system that will support key activities of a leisure complex.

First part of the thesis analyzes the leisure area RY:BAR, which have provided basis for creating models using the conceptual modeling. Secondly, these models served to construct a web application for lesiure time complexes, using basic principles of Normalized Systems Theory.

This web application is written in the Symfony PHP framework that supports modularity in web applications. The created solution is scalable for the future development.

The goal of this work is to create a general web-based system, usable for any leisure complex with the possibility of selecting the necessary modules for the key activities of the complex. The unique selling point of the system is scalability for future expansion and addition of new modules. Moreover, the solution itself is low cost which has resulted from the economic evaluation of this application in the RY:BAR complex.

Keywords normalized information system, modular web application, analysis of a leisure complex, economic evaluation, evolvability, Normalized Systems Theory, PHP, ORM, Symfony

Obsah

Úvod	1
1 Cíl práce	3
2 Popis a analýza volnočasového areálu RY:BAR	5
2.1 Popis volnočasového areálu RY:BAR	5
2.2 Klíčové činnosti areálu RY:BAR	7
2.3 Požadavky na systém	11
3 Existující řešení	13
3.1 Zkoumané parametry	13
3.2 Řešení	13
3.3 Zhodnocení výsledků	17
4 Návrh vlastního řešení	19
4.1 Architektura	19
4.2 Moduly	22
4.3 Normalized Systems Theory	24
4.4 Grafické uživatelské rozhraní	26
5 Implementace	29
5.1 Použité technologie	29
5.2 Průběh implementace	32
5.3 Aktivity a jejich přidávání	37
5.4 Přidávání nových modulů a zprovoznění aplikace	38
5.5 Nové moduly	38
5.6 Zprovoznění aplikace	38
6 Testování	41
6.1 Průběh testování	41

6.2	Scénáře	41
6.3	Shrnutí testování	47
7	Ekonomicko-manažerské zhodnocení a evolvabilita řešení	49
7.1	Přínosy systému	49
7.2	Náklady na vývoj a zavedení systému	49
7.3	Návratnost investice	51
7.4	Porovnání se stávajícími řešeními	53
7.5	Rozšiřitelnost modulů	53
7.6	Nové moduly a rozšíření v budoucím vývoji	54
	Závěr	55
	Bibliografie	57
	A Seznam použitých zkratk	61
	B Obsah příloženého CD	63

Seznam obrázků

2.1	Orientační mapa areálu	7
2.2	Strukturální model baru areálu RY:BAR	8
2.3	Strukturální model aktivit areálu RY:BAR	9
2.4	Flowchart aktivit	10
3.1	Screenshot z webové stránky www.gutovka.cz [4]	14
3.2	Screenshot z webové stránky www.parkfrymburk.cz [5]	15
3.3	Screenshot z webové stránky www.actionpark.cz [6]	16
4.1	Třívrstvá architektura	19
4.2	Hybrid 3vrstvé architektury a MVC [11]	21
4.3	Role uživatelů v systému	22
4.4	Grafické rozhraní aplikace	28
5.1	Adresář pro přidání nového modulu	36
5.2	Vzhled přidávání aktivity	37
5.3	Vhled aktivity po předchozím přidání	38
7.1	NPV podle roků využívání systému	52

Seznam tabulek

6.1	Scénář č. 1	42
6.2	Scénář č. 2	43
6.3	Scénář č. 3	44
6.4	Scénář č. 4	45
6.5	Scénář č. 5	46
7.1	Náklady a výnosy v 10 letech	51

Úvod

V současné době se často zabýváme tím, kde strávit volný čas. Je to často kladená otázka lidí, hlavně v létě, kdy musí vymyslet program pro své děti. K trávení volného času slouží volnočasové areály, kterých je v České republice čím dál tím více, ale lidé o nich moc neví a narazí na ně většinou jen náhodou, když kolem tohoto místa tráví svou dovolenou.

V této bakalářské práci je provedena analýza jednoho volnočasového areálu jménem RY:BAR, který se nachází na jihu Čech. Tento areál byl vybudován v roce 2015 a nadcházející rok byl uveden do provozu. V tomto areálu je možnost provozovat mnoho aktivit, ale povědomí o těchto aktivitách schází.

Cílem práce je vytvořit informační webový systém, který bude sloužit pro návštěvníky volnočasového areálu jako zdroj informací, ve kterém si budou mít možnost rezervovat nějakou aktivitu, kterou areál nabízí. Tento informační systém může sloužit jakémukoliv volnočasovému areálu, který bude poskytovat nějaké aktivity. Bude vytvořen na základě Normalized Systems Theory, na bázi modulů, které bude možno přidat nebo odebrat, na základě požadavků provozovatelů areálu.

Nejprve bude provedena analýza, poté výběr technologií, které budou podporovat Normalized Systems Theory, pomocí kterých se systém vytvoří. Budou popsány systémy, které jsou již vytvořené, pokud nějaké existují, z těchto zjištění bude vycházet návrh a implementace nového systému, který bude řešit nedostatky již vytvořených systémů. Na závěr proběhne testování a ekonomicko-manažerská analýza tohoto systému.

Cíl práce

Cílem této bakalářské práce je vytvoření informačního systému pro volnočasové areály, na základě analýzy jednoho areálu jménem RY:BAR, který se nachází na jihu Čech. Proběhne jeho analýza, porovnání jiných informačních systémů pro tyto areály a následně návrh a implementace nového informačního systému. Tento informační systém bude sloužit pro různé potřeby provozovatele areálu a cílem je vytvořit ho na základě Normalized Systems Theory, která podporuje modularizaci systémů. Důležitým požadavkem je též jednoduchost a přehlednost systému tak, aby s ním mohl pracovat běžný uživatel.

Hlavním cílem je tedy vytvoření funkčního, přehledného a evolvabilního systému, který bude podporovat oblasti vytyčené zadavatelem, pro jakýkoliv volnočasový areál. Cílem bude též ekonomicko-manažerské zhodnocení a vyhodnocení evolvability tohoto systému.

Popis a analýza volnočasového areálu RY:BAR

V této kapitole je analyzován volnočasový areál RY:BAR. Z této analýzy plynou problémy tohoto areálu, ze kterých budou některé řešené pomocí nového informačního systému.

2.1 Popis volnočasového areálu RY:BAR

V jižních Čechách, nedaleko největší vodní nádrže v České republice podle rozlohy – Lipno I [1], se nachází volnočasový areál RY:BAR, který slouží v letních měsících návštěvníkům jako zdroj zábavy, odpočinku, občerstvení a adrenalinu. Provoz tohoto areálu je pod záštitou společnosti Amenity Resorts, působící ve třech městech České republiky.

2.1.1 Vznik a vývoj

Areál byl vybudován v roce 2015 a následující rok byl uveden do provozu. V areálu se nacházejí dva rybníky, ve kterých jsou vysázeny ryby, proto neslouží ke koupání. Na jednom rybníku se provozuje wakeboarding od prvního roku působení, na druhém jsou mechanické a elektrické loďky pro malé děti.

Kolem rybníků jsou volně k dispozici lehátka, která může využívat široká veřejnost. Areál disponuje také bistroem, ve kterém se připravují pokrmy výhradně z ryb, v nabídce jsou také nápoje. V neposlední řadě jsou zde také dětské prolézačky. Je možnost vypůjčit si zde tenisové rakety nebo náčiní na petánque a hrát lazy tennis a petánque na vyznačených místech. Poslední uvedené aktivity se přidaly na začátku letního období roku 2017. Nachází se zde také menší letní kino, které promítá filmy dvakrát do týdne. Kino bylo též novinkou loňského roku.

2.1.2 Problémy areálu

V areálu je široká možnost trávení volného času, ale lidé o některých aktivitách nevědí. Jediná zmínka o programu letního kina se nachází na vitríně bistra a další aktivity jsou uvedené jen na ceníku, který se nachází na stěně bistra. Veřejnost se tedy dozvídá, jaké jsou možnosti, až když tento areál navštíví. Tímto přichází o možný zisk, který by plynul z prodeje občerstvení těmto zákazníkům nebo z využití placených aktivit areálu.

Dalším problémem, kromě nevyužívání všech možností areálu, je absence rezervačního systému pro wakeboard, což má za následek, že areál přichází o některé zákazníky, kteří by si zajezdit chtěli, ale nemají čas čekat v pořadí.

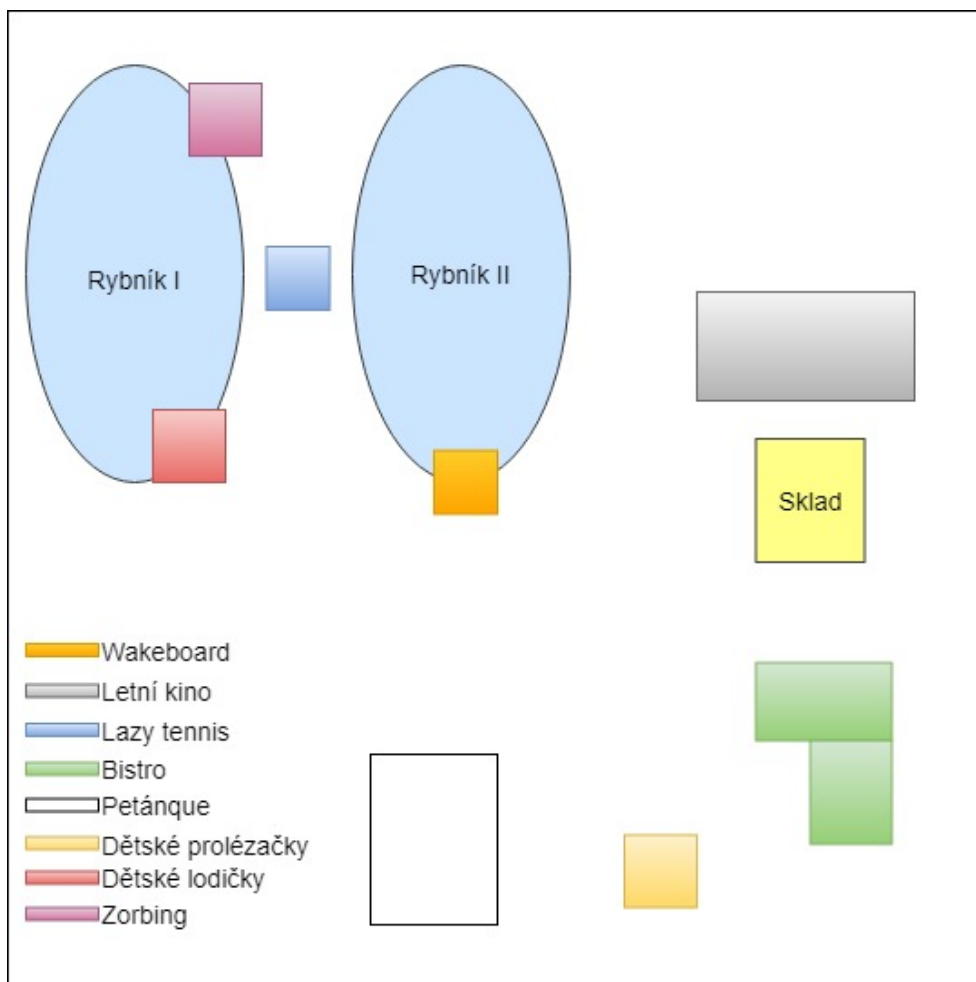
Zaměstnanci areálu, kteří jsou zaměstnáváni většinou jen sezonně, tedy brigádníci, jsou zaměstnaní většinou na dohodu o provedení práce a jsou placeni hodinově. Každému zaměstnanci je přidělen arch, kam si píše odpracované hodiny, což má za následek to, že se arch může ztratit a zaměstnanci si zpětně hodiny dopisují. [2]

2.1.3 Shrnutí

Areál je určen hlavně pro jednorázové návštěvníky Lipna nad Vltavou v letním období, hledající aktivní nebo pasivní využití volného času. Z rozhovoru s provozovatelem tohoto areálu plyne přání účasti více lidí na akcích a aktivitách pořádaných tímto areálem [2], v čemž by mu měl pomoci výstup této bakalářské práce.

2.2 Klíčové činnosti areálu RY:BAR

Pro přehlednost je uvedena na Obrázku 2.1 jednoduchá mapa, kde se jaká aktivita nachází. Ve skladu se uchovává vybavení na jednotlivé aktivity.



Obrázek 2.1: Orientační mapa areálu

V předchozí části bylo zmíněno několik aktivit a činností areálu RY:BAR, které jsou pro jeho chod důležité. Dále tyto činnosti můžeme dělit na zpoplatněné a nezpoplatněné.

Mezi zpoplatněné činnosti areálu patří:

- wakeboarding,
- dětské mechanické a elektrické lodičky,
- petánque,

2. POPIS A ANALÝZA VOLNOČASOVÉHO AREÁLU RY:BAR

- lazy tennis,
- občerstvení v bistro,
- zorbing.

Mezi nezaplatněné činnosti areálu patří:

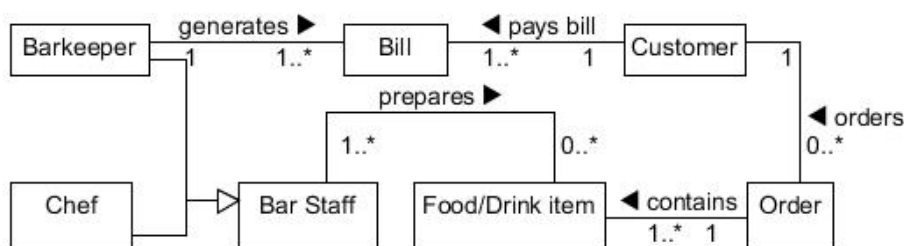
- lehátka a posezení v areálu,
- dětské prolézačky,
- letní kino.

Informační systém bude podporovat jen některé z nich, jelikož některé z těchto činností tolik nelákají veřejnost jako jiné. Největším lákadlem je wakeboard, adrenalinový sport díky kterému je areál aktraktivní. Lidi též zajímá bistro, ve kterém jsou prodávány jen rybí speciality. [2] Návštěvnost letního kina je menší, jak už bylo popsáno v předešlé kapitole, ale informační systém by tuto činnost měl podporovat, protože pokud se návštěvník přijde podívat na film, využije i nabídky bistra nebo by mohl přijít vyzkoušet wakeboarding.

2.2.1 Bistro

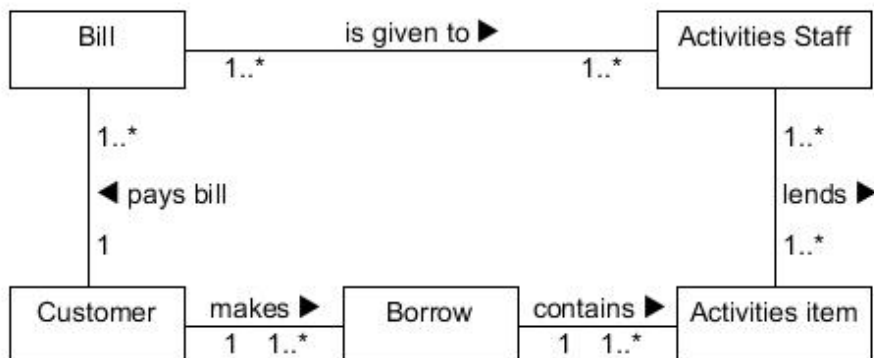
V areálu se nachází bistro, které vydává účtenky za veškeré koupené zboží, tzn. všechny pokrmy a nápoje a taktéž přístup k wakeboardu, dětským lodičkám a zorbingu, zapůjčení si náčiní pro hraní petánque nebo lazy tennisu.

Obrázek 2.2 popisuje strukturu kupování občerstvení a aktivit v RY:BARu. Barman bistra vydává účtenky, na kterých se nachází položky z bistra i položky aktivit v areálu. Pokud se na účtence nachází nějaké aktivity, jde zákazník s touto účtenkou za obsluhou wakeboardu, kteří mají na starost všechny aktivity, které se v RY:BARu provozují, jak je ukázáno na Obrázku 2.3.



Obrázek 2.2: Strukturální model baru areálu RY:BAR

Systém, který podporuje prodej občerstvení a aktivit již existuje, a nebude předmětem této práce.



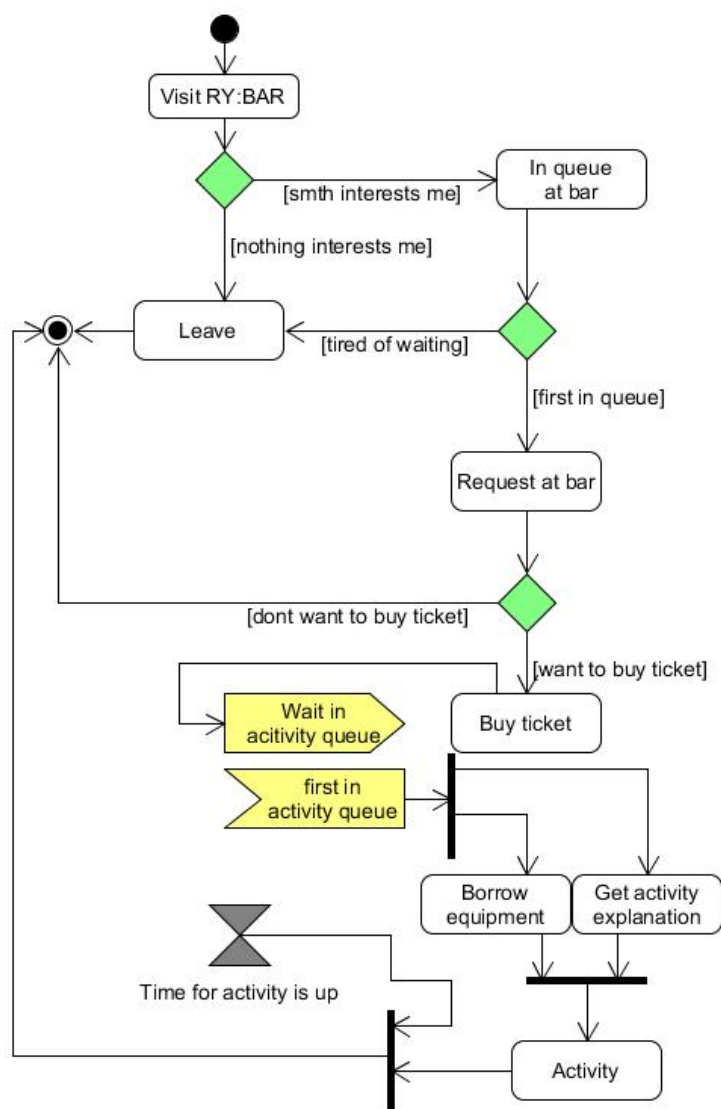
Obrázek 2.3: Strukturální model aktivit areálu RY:BAR

2.2.2 Wakeboarding, dětské lodičky a zorbing

V areálu je možnost wakeboardingu a půjčování dětských lodiček, jak je uvedeno v předešlých tématech. Na Obrázku 2.4 je možnost zkoumat celý proces, jak může návštěvník využít tyto tři aktivity. Jedná se o ten samý workflow pro všechny aktivity. Jediný rozdíl je v tom, že lodiček pro děti je k dispozici cca 15, kdežto na wakeboardu může jezdit současně pouze jeden člověk.

Z tohoto plyne problém, že se zákazníci řadí do pořadníku, o stavu pořadníku se je snaží informovat obsluha wakeboardu. Pokud se ale pořadník stále plní, zaměstnanci přicházejí o přehled [3], kdo je v pořadníku a zákazníci jsou nespokojení.

Už více zákazníků komunikovalo se zaměstnanci RY:BARu a dožadovali se nějakého rezervačního systému, aby nevznikaly dohady a problémy, kdo je zrovna v pořadníku na prvním místě.



Obrázek 2.4: Flowchart aktivit

2.2.3 Ostatní zpoplatněné aktivity

Z workflow pro ostatní placené aktivity nevyplývají žádné problémy. Je to dáno i tím, že o tyto aktivity není takový zájem a možné problémy by se mohly objevit, kdyby bylo více zákazníků, ale podle zaměstnanců areálu více zákazníků žádné problémy nepřinese. [3]

2.2.4 Letní kino

Letní kino se nachází za skladem podle Obrázku 2.1. Problémem je, že kino není dobře viditelné, protože z jedné strany je ohraničené stromy a z druhé strany je wakeboard, který strhne nejvíce pozornosti. Do budoucna by bylo dobré, kdyby účast na filmech byla větší, tím pádem by i tržby za ostatní aktivity stouply. Kino promítá filmy dvakrát týdně. Jeden film je rodinný pro rodiny s dětmi, druhý film je pro účastníky, kteří mají rádi starší české nebo zahraniční filmy. Pokud by byla účast větší, bylo by to dobrou reklamou pro ostatní činnosti areálu.

2.3 Požadavky na systém

Podle vytyčených hlavních činností areálu budou některé z nich podporované informačním systémem.

Jedny z nejdůležitějších funkcí je informační, pro uživatele, kteří by areál navštívili. Další důležitou funkcí je možnost rezervace pro wakeboard, s tím souvisí též správa uživatelů a zaměstnanců. Dalším důležitým bodem je program letního kina a možnost jeho správy. Jednotlivé funkční požadavky jsou děleny podle modulů, které se budou vytvářet.

2.3.1 Funkční požadavky

F 1. aktivity areálu

F 1.1. zobrazení všech aktivit v areálu

F 2. uživatelské prostředí

F 2.1. registrace uživatele

F 2.2. přihlášení a odhlášení uživatele

F 2.3. zobrazení profilu

F 2.4. editace profilu

F 2.5. smazání uživatele

F 3. správa systému

F 3.1. zobrazení všech uživatelů

F 3.2. nastavení role uživatele

F 3.3. smazání uživatelů

F 3.4. přidání stránky s novou aktivitou

F 3.5. odebrání stránky s aktivitou

F 3.6. možnost úpravy stránky s aktivitou

2. POPIS A ANALÝZA VOLNOČASOVÉHO AREÁLU RY:BAR

F 4. rezervační systém

F 4.1. vytvoření nové rezervace

F 4.2. smazání rezervace

F 4.3. přehled rezervací podle role uživatele

F 5. letní kino

F 5.1. přehled programu kina

F 5.2. vytvoření nové události

F 5.3. editace události

F 5.4. smazání události

2.3.2 Nefunkční požadavky

N 1. **Webová aplikace**, pro kterou není potřebná instalace žádných programů. Bude funkční ve webovém prohlížeči.

N 2. **Česká lokalizace**, jelikož většina návštěvníků jsou Češi. [2, 3]

N 3. **Možnost rozšiřitelnosti** pro budoucí možný rozvoj aplikace nebo jejích funkcí.

N 4. **Uživatelská přívětivost**, aby se uživatelé v systému neztráceli.

N 5. **Nízké náklady** na správu systému, hosting aj.

Existující řešení

V této kapitole se věnuji existujícím řešením a jejich srovnání. Nejprve jsou vytyčeny parametry, které budou pro vývoj aplikace důležité, následně jsou zkoumána jednotlivá řešení a splnění jednotlivých parametrů.

3.1 Zkoumané parametry

Je důležité, aby vyvíjená aplikace splňovala některé parametry, zejména se jedná o nefunkční požadavky, které jsou uvedeny v předešlé kapitole.

Parametry, které budou zkoumány:

- vzhled a uživatelská přívětivost,
- dostatečná informační hodnota,
- potřeba rezervací a jejich řešení,
- odezva webových stránek,
- celkový dojem z webové stránky.

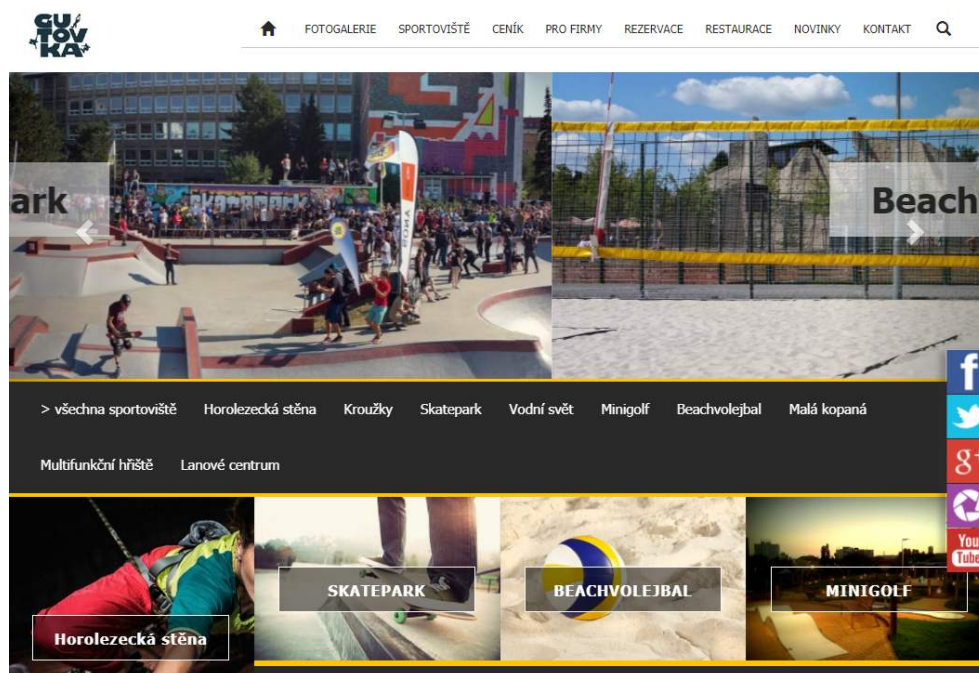
3.2 Řešení

V následujících kapitolách jsou uvedena řešení pro jednotlivé areály. Některé areály postrádají svou webovou stránku, někdy jsou jen uvedena na webových stránkách obce, ve které se nachází. Taková řešení zde budou také popsána a diskutována.

3. EXISTUJÍCÍ ŘEŠENÍ

3.2.1 Volnočasový areál Gutovka

Volnočasový areál Gutovka se nachází v Praze u zastávky metra Strašnická. Tento areál je rozlohou o mnoho větší než areál RY:BAR. Na Gutovce se dá provozovat větší množství aktivit a sportů.



Obrázek 3.1: Screenshot z webové stránky www.gutovka.cz [4]

Na první pohled webové stránky areálu www.gutovka.cz [4] vypadají dobře, bohužel některé odkazy v zápatí hlavní stránky odkazují na stránky bez obsahu. Vzhled je příjemný, barevná kombinace vhodná. Některé části webu jsou nepřesně umístěny, jak je vidět i na Obrázku 3.1 u položky horolezecká stěna.

Informační hodnota stránky je i vzhledem k velké nabídce dostatečná. U každého ze sportovišť je napsáno, zda je v provozu a jak se využívá. Jen chybí přímé odkazy z jednotlivých stránek na rezervační systém.

Rezervační systém je realizován systémem, který poskytuje společnost Bizzy. Tento systém je zcela nepřehledný, nehodí se vizuálně do prostředí stránky a není pro uživatele intuitivní. Na druhou stranu je možné rezervovat více sportovišť a je určen pro větší sportovní areály, kterým Gutovka je. Řešení rezervací by mohlo být určitě zlepšeno.

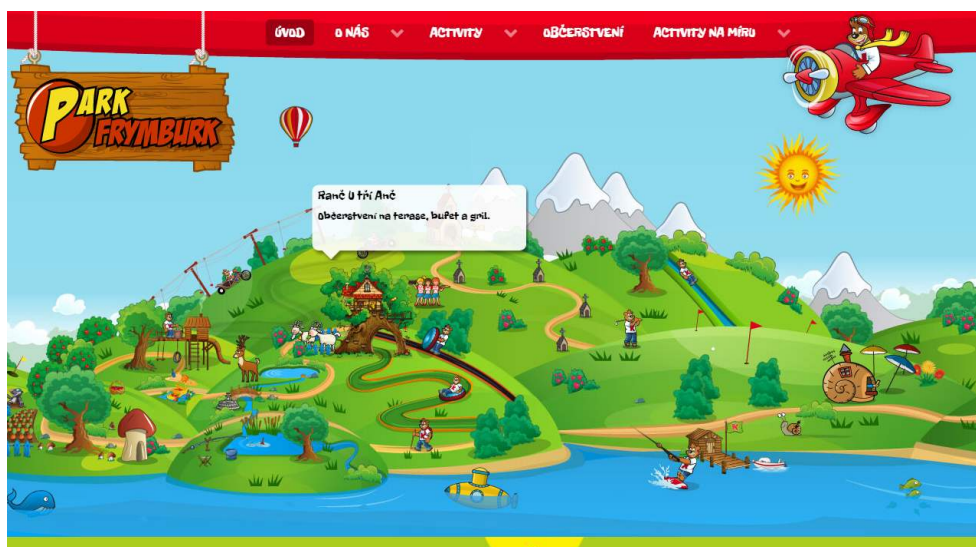
Webové stránky reagují rychle, jen u rezervačního systému je problém s delší odezvou. Ve webovém prohlížeči Google Chrome se stránka rezervací vůbec nenačte.

Areál má dobré webové stránky, je vidět, že se o stránky stále někdo stará, ale nejsou dotažené do konce. Objevují se někde malé chyby, ale pro běžného

uživatelé nemají vliv na používání.

3.2.2 Activity park médi Kubíka ve Frymburku

Frymburk se nachází na jihu Čech a tento areál může být RY:BARu konkurencí. Obsahuje více aktivit, je na větším pozemku a také se zde nachází wakeboard.



Obrázek 3.2: Screenshot z webové stránky www.parkfrymburk.cz [5]

Webové stránky www.parkfrymburk.cz [5] jsou vzhledově příjemné, až na font, který je zde použit. Nedá se dobře přečíst, ale areál je určen hlavně rodičům s dětmi, které mají přístup k internetu, font jim tedy tolik nevadí. Webová stránka má animovaný vzhled a vypadá spíše určená pro děti.

Informace na těchto stránkách jsou dostatečné, návštěvník stránek se zde dozví všechny potřebné informace, navigace stránek je intuitivní.

V areálu ve Frymburku se nachází také wakeboard, stejně jako v RY:BARu. Stránka této aktivity odkazuje na externí stránku wakeareálů Wakemaster, kde jsou zobrazeny všechny pobočky, kde je možné jezdit na wakeboardu. Je zde i možnost rezervace, ale po kliknutí se zákazník dostane na další externí stránku, kde je už možnost si wakeboard rezervovat. Tato navigace není intuitivní, je složité se proklikat až k samotné rezervaci.

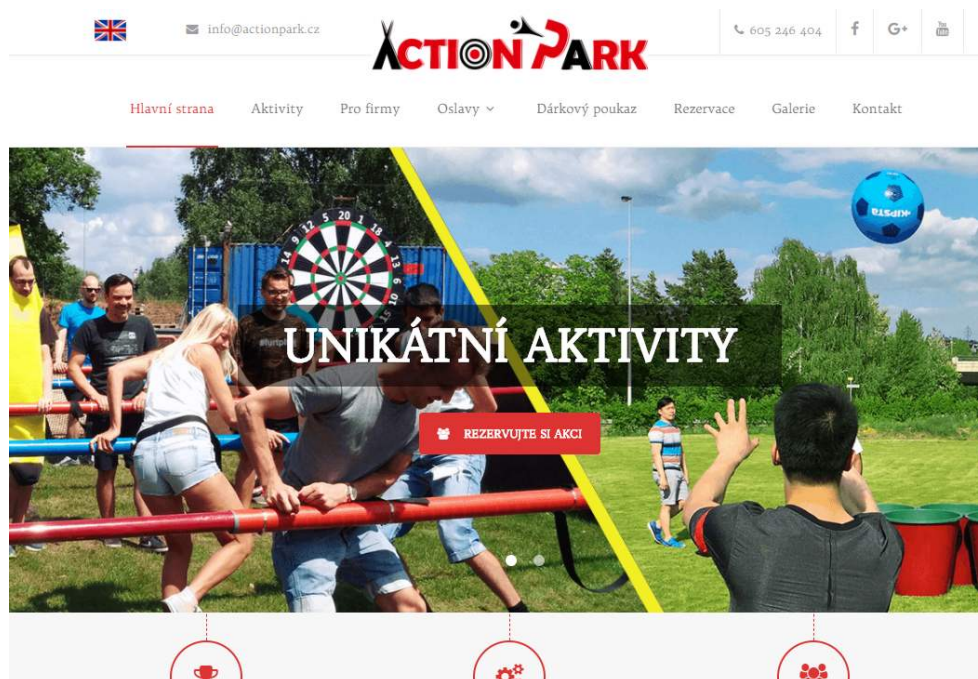
Odezva webových stránek je horší, vzhledem k velkému obsahu scriptů, které stránku zpomalují a nejsou dobře umístěné ve zdrojovém kódu stránky.

Tyto webové stránky jsou v pořádku, obsahují hodně informací, největší slabinou je zde vzhled stránek, který není elegantní, ani jednoduchý. Font je nepříjemný, ale na druhou stranu se zde člověk dozví vše potřebné.

3. EXISTUJÍCÍ ŘEŠENÍ

3.2.3 Action Park Praha

Action Park Praha se nachází v Praze 12 v Lahovičkách. Podle webových stránek www.actionpark.cz [6] nabízí areál netradiční sportovní aktivity, dále možnost teambuildingů a občerstvení pro skupiny.



Obrázek 3.3: Screenshot z webové stránky www.actionpark.cz [6]

Vzhledově jsou tyto webové stránky asi nejlepší ze všech porovnávaných řešení. Webové stránky jsou velice příjemné, i přes větší množství odkazů není problém se zde vyznat. Návštěvník má přehled, na jaké stránce se nachází, je jednoduché najít to, co zákazník potřebuje.

Na těchto stránkách se nachází mnoho informací. Zákazník se dozví, kde areál je, jak se k němu dostat nebo kdy je otevřen. Všechny aktivity jsou rozepsány a je zde i napsána cena a možnost rezervace. Jsou zde i odkazy na stránky Facebooku nebo YouTube s dalšími informacemi o aktivitách, hodnoceníh apod.

Rezervace jsou na těchto stránkách též obsažené, ale jsou řešené jen pomocí jednoduchého formuláře, který odešle email správcům areálu, kteří musí dále tuto rezervaci dále řešit a potvrdit zpět zákazníkovi. Tento systém rezervací není vhodný. Při větším počtu formulářů odeslaných od zákazníků může vzniknout zmatek v rezervacích a dále vznikají chyby a návštěvníci mohou být nespokojeni.

Odezvě stránek se nedá nic vytknout, web je rychlý a návštěvníci nebudou mít s odezvou problém.

Webové stránky www.actionpark.cz jsou nejlepší ze všech uvedených řešení. Kromě rezervací je web téměř bezchybný. Zákazník se zde dozví všechny potřebné informace jednoduše a rychle, není potřeba zbytečně dlouhého hledání.

3.2.4 Volnočasový areál (VČAS) - Město Bakov nad Jizerou

Zde není potřeba rozdělovat parametry na jednotlivé odstavce, jako u ostatních, jelikož se jedná jen o několik statických stránek na webových stránkách obce [7].

Toto řešení není vhodné, postrádá některé důležité informace. Není zde například uvedena mapa, jak se do areálu dostat. Webové stránky obsahují informace o historii, které pro zákazníka nemusí být důležité.

O možnosti rezervací aktivit zde není zmínka, není to zřejmě potřeba. Podle webových stránek je možná rezervace chatků, kalendář ale není zveřejněn, stránky byly aktualizovány naposled v roce 2016. Rezervace je možná jen telefonicky.

3.2.5 Volnočasový areál obce Radim u Kolína

Webové stránky tohoto areálu jsou stejně jako u předchozího pouze statické. Tyto navíc mají jen jednu stranu, na které jsou umístěny všechny informace [8].

Tento volnočasový areál je určen pro míčové sporty a je v něm několik hřišť. Je možná i rezervace, jen není napsáno, kde se sportoviště můžou rezervovat a jakým způsobem.

Toto řešení není vhodné pro areál RY:BAR. Postrádá většinu informací, není uživatelsky přívětivé, neobsahuje možnost online rezervací a vzhled není atraktivní. Tento areál bude zřejmě určen hlavně pro místní obyvatele obce Radim u Kolína.

3.3 Zhodnocení výsledků

U všech řešení se objevují funkce, které mají areály společné, některé je mají pokryté více, některé méně. Hlavní funkcí porovnávaných řešení je podání informací.

Výčet funkcí které podporuje alespoň jeden z areálů:

- přehled jednotlivých aktivit,
- možnosti rezervací,
- přehled občerstvení,
- přehled novinek.

3.3.1 Uživatelské prostředí a správa uživatelů

Ani jeden z areálů nepodporuje přihlášení do systému areálu. Vyskytuje se zde problém změn na webu. Pokud chce majitel nebo zaměstnanec areálu změnit nějaké věci, které už neplatí nebo se změnilly, nemá takovou možnost. Musí oslovit tvůrce webu a jednat o změnách s ním. Toto může být problémem, pokud bude areál z nějakých důvodů potřeba například na jeden den uzavřít.

Další související problém s absencí uživatelského prostředí je možnost rezervací přímo ve webové aplikaci areálu. Jen jeden z areálů má svůj vlastní rezervační systém, který je řešený jen formulářem, který informuje emailem správce systému, který dále musí řešit rezervace a odpovídat zákazníkovi. Ostatní areály buď možnost rezervací nemají nebo je zde jen informace o možnosti rezervace po telefonu.

3.3.2 Cena

Jednotlivé areály mají webovou aplikaci vytvořenou firmami, které se zabývají jejich tvorbou. Webová aplikace areálu ve Frymburku je vytvořena firmou Agionet s.r.o., která na svém webu uvádí, že jednoduché webové stránky podle šablony jsou schopni vytvořit za částku 20 000,- [9]. Cena se bude samozřejmě navyšovat s dalšími požadovanými funkcími, se kterými roste i počet hodin, které na projektu budou odpracované. Tato firma na svých webových stránkách uvádí, že hodinová sazba se pohybuje kolem 600,-.

Všechny uvedené webové stránky se jeví jako vytvořené podle nějaké šablony, která byla upravena. Nejsou dokonalé, což plyne z použití a upravování šablony. Neobsahují pokročilejší funkce jako správu uživatelů, možnost správců přidávat nebo odebírat potřebné věci.

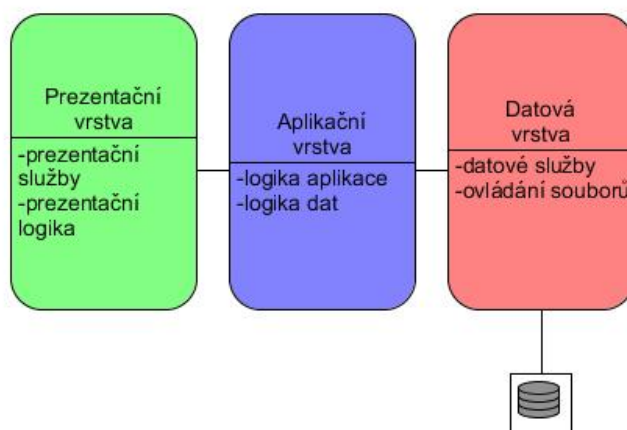
Pokud chce společnost pro svůj areál vytvořit systém, který bude přizpůsobený potřebám areálu, cena bude vysoká. Reklamy na levné weby zaujmou majitele areálu, pokud ale bude potřeba vytvořit webovou aplikaci na míru, cena těchto firem bude značně vyšší, než je uvedena.

Návrh vlastního řešení

V této kapitole je popsána architektura, Normalized Systems Theory a jak modulární rozdělení podporuje tuto teorii. Z tohoto návrhu nadále vychází samotná implementace.

4.1 Architektura

Celý systém bude postaven na třívrstvé architektuře tak, jak je ukázáno na Obrázku 4.1. Výhodou třívrstvé architektury oproti dvouvrstvé a jednovrstvé je možnost oddělení jednotlivých složek aplikace. V třívrstvé architektuře se odděluje prezenční, aplikační a databázová vrstva. Jednotlivé vrstvy se starají o přidělené části aplikace, nejsou samy sebou ovlivňovány a pokud je potřeba změny v nějaké vrstvě aplikace, neovlivňuje to vrstvy ostatní, což by v méněvrstvé aplikaci neplatilo.



Obrázek 4.1: Třívrstvá architektura

Zásadní výhody třívrstvé architektury:

- **nezávislost jednotlivých vrstev,**
- **přizpůsobování velikosti jednotlivých vrstev,**
- **správa bezpečnosti,**
- **oddělení znalostí a jednotlivých kompetencí.**

Tyto všechny uvedené vlastnosti jsou nezávislé na jednotlivých vrstvách.

Systém bude obsahovat více modulů a jelikož je požadavkem, aby aplikace byla modulární, tato architektura nejlépe podporuje tento požadavek. Jednotlivé moduly budou obsahovat svoje řadiče a pohledy, které budou spolupracovat s databázovou vrstvou.

4.1.1 MVC architektura

MVC architektura je rozdělena na Model-View-Controller, jak již napovídá název. Základní myšlenkou MVC architektury je modularita, kde je kód možno použít znova beze změny [10]. Jednotlivé vrstvy jsou popsány v následujících kapitolách a na Obrázku 4.2 je model, jak MVC v třívrstvé architektuře funguje.

4.1.1.1 Model

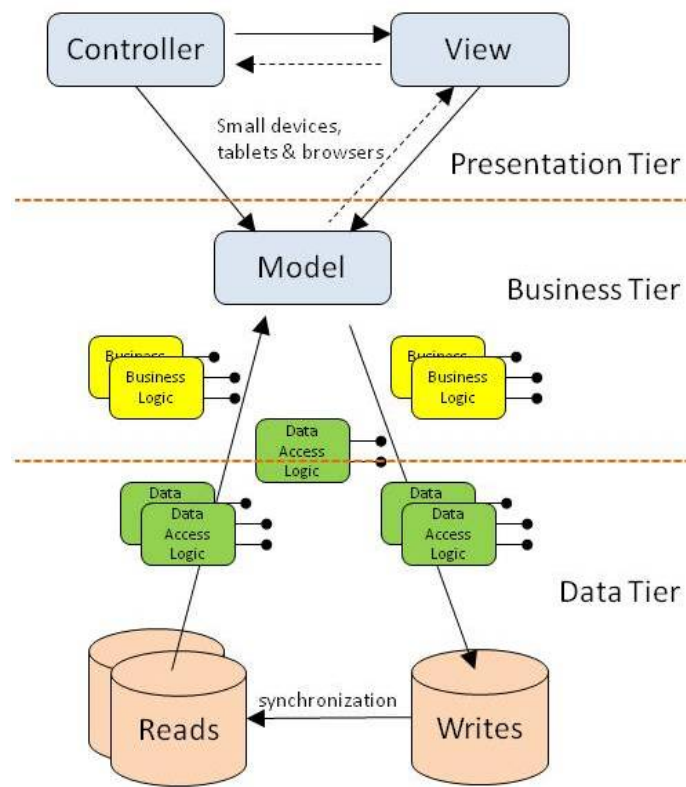
V této vrstvě je model reprezentován datovou vrstvou a aplikační vrstvou, která komunikuje s datovou vrstvou. V datové vrstvě je umístěna databázová logika a samotné tabulky databáze, se kterou budou další 2 vrstvy spolupracovat. Databáze nebude obsahovat žádnou výpočetní logiku, pouze data.

4.1.1.2 View

Tato vrstva je určena pro zobrazování uživateli. Uživatel vidí to, co mu View vrstva zobrazí, jako jsou např. tabulky, výpisy uživatelů, výpis profilu, rezervace apod.

4.1.1.3 Controller

Controller nebo také aplikační logika je část, která vše řídí, komunikuje přímo jen s vrstvou Model, aby mohla zpracovávat data. Při uživatelském kliknutí na odkaz Controller nejdříve zpracuje požadavek a nechá zobrazit uživateli výsledek. [12]



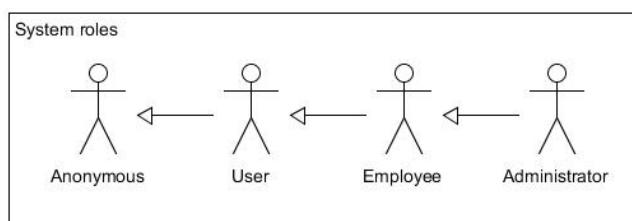
Obrázek 4.2: Hybrid 3vrstvé architektury a MVC [11]

4.1.2 Zabezpečení a správa uživatelů

Aplikace bude podporovat i správu uživatelů a rozdělení přístupu k modulům a částem aplikace, aby byly splněny všechny požadavky. Správa uživatelů bude probíhat v jednom modulu, který na tuto problematiku bude určen. Jednotlivé přístupy uživatelů potom budou řešeny omezením cest v konfiguraci samotné aplikace. Uživatelé budou rozděleni podle Obrázku 4.3 na 3 skupiny, kde administrátor bude mít svá práva a navíc práva zaměstnance, uživatele a anonymního uživatele a takto kaskádovitě.

4.1.3 Databáze

Databáze je důležitou součástí aplikace a bude potřeba pro uchování všech dat, potřebných pro fungování aplikace. Pro aplikaci bude využita relační databáze za použití ORM (Object-Relational Mapping). ORM bude využito kvůli nevelké potřebě datové logiky a nepotřebnému složitému dotazování nad databází. ORM je schopné převádět objekty vytvořené v aplikační logice na jednotlivé řádky tabulky za definovaných podmínek.



Obrázek 4.3: Role uživatelů v systému

4.2 Moduly

Moduly jsou základním prvkem této aplikace, každá část aplikace bude tvořena modulem. Některé moduly budou úplně nezávislé, jiné moduly budou funkční jen za přítomnosti modulů, na kterých jsou závislé, toto však nepopírá teorii normalizovaných systémů, jak bude popsáno v další kapitole.

4.2.1 Základní modul

Tento modul je základním stavebním kamenem celého systému. Bude zde zajištěno, jak má aplikace fungovat, bude zde všechno nastavení, od nastavení napojení databáze po nastavení emailu apod. V základním modulu bude nastaven i základní administrátorský profil, ke kterému bude přístup po přidání modulu uživatelského rozhraní. Pokud bude chtít zákazník někde tento systém zavádět, bez základního modulu se neobejde. Tento modul bude pokrývat funkčnosti F 1., kde budou vloženy základní aktivity areálu, které zákazník provozující tento systém bude chtít.

Bude obsahovat menu, které bude generováno podle přítomných modulů. Pokud bude existovat jen tento základní modul, webová aplikace bude obsahovat jen základní domovskou stranu s menu a aktivitami areálu.

4.2.2 Modul uživatelského rozhraní

Modul uživatelského rozhraní bude podporovat funkční požadavky F 2. Uživatel bude mít možnost vytvořit účet, přihlásit se, editovat profil nebo ho celý smazat. Pokud uživatel zapomene heslo, bude mít možnost nechat si systémem vygenerovat nové heslo, které mu bude odesláno na email. Tento modul bude důležitý pro všechny ostatní moduly plánované pro tuto aplikaci, bez tohoto modulu se nebude moci do systému nikdo přihlásit ani nijak manipulovat se systémem.

Uživatelé anonymní role nebudou mít v systému přístup k žádným dynamickým prvkům aplikace. Jediné, co budou moci v systému dělat je proklikat jednotlivé odkazy, které jim budou zpřístupněné.

Uživatelé s rolí uživatele budou mít přístup ke svému profilu, k jeho editaci nebo mazání a dalším funkcí modulu, které budou v systému přítomny a budou mít oprávnění (např. vytváření rezervací pokud bude přítomen modul rezervačního systému).

Uživatelé role zaměstnance budou mít přístup, stejně jako role uživatele, ke svému profilu a dalším funkcí modulu, pokud budou moduly přítomné a budou mít oprávnění (např. zobrazení všech rezervací nebo smazání rezervace uživatele).

Administrátorská role bude mít přístup ke všem funkcí modulu předchozích rolí a navíc bude mít oprávnění pro další funkcí modulu (např. přidání nové aktivity pokud bude přítomen modul správy systému).

4.2.3 Modul správy systému

Tento modul bude sloužit hlavně pro uživatele s administrátorským přístupem do aplikace. Pokrývá funkční požadavky F 3. V tomto modulu bude mít uživatel s oprávněním možnost zobrazení všech uživatelů zaregistrovaných do systému. Bude jim moci měnit role, aby měli přístup do jiných částí systému nebo bude moci uživatele mazat. V tomto modulu bude navíc možnost zobrazení aktivit areálu a přidání nových aktivit, jejich mazání nebo editace.

Tento modul není povinný pro chod systému. Pokud bude chtít provozovatel tohoto systému další funkcí modulu a dynamičnost systému, bude tento modul vyžadovat.

4.2.4 Modul rezervačního systému

Tento modul bude sloužit k vytváření rezervací v systému. Modul bude pokrývat funkční požadavky F 4. Pokud bude chtít provozovatel používat tento modul, musí mít i modul uživatelského rozhraní, na kterém je tento modul založen.

Do tohoto modulu bude mít přístup každý uživatel. Pro anonymní uživatele bude modul nepoužitelný. Přihlášení uživatelé budou mít přístup k vytváření a mazání rezervací. Bude zde i přehled jejich vytvořených rezervací. Uživatelé role zaměstnance budou mít přístup k zobrazení všech rezervací a k jejich mazání. Administrátoři budou mít práva stejná jako zaměstnanci.

4.2.5 Modul letního kina

Tento modul bude podporovat funkcí modulu F 5. V modulu bude možnost pro uživatele s rolí zaměstnance a administrátora přidávat nové příspěvky, kdy se jaký film bude promítat. Budou mít možnost editace těchto příspěvků a jejich mazání. Ostatní uživatelé budou mít možnost si tyto příspěvky zobrazit.

4.3 Normalized Systems Theory

Tato kapitola se zabývá teorií normalizovaných systému, jejími teorémy a jak je modulární návrh podporovaný touto teorií.

4.3.1 NST

Normalized Systems Theory nebo v českém překladu teorie normalizovaných systému je teorie, která byla vytvořena jako teorie pro vytváření softwaru, který bude více agilní. Software, který je agilní a může se snadno měnit se nazývá evolvabilní software. Této evolvability může být dosaženo po aplikování specifických teorému při vývoji aplikace, která bude mít nakonec evolvabilní a agilní architekturu, protože dnešní obchodní požadavky a nároky se mění každým dnem. [13]

Teorie byla vytvořena na základě studování změn a evolvability na úrovni architektury softwaru aplikováním konceptů jako jsou stabilita a entropie na studii modulární struktury architektury softwaru. Pro aplikování teoretické stability v softwarové architektuře stabilita implikuje, že pro každý konečný vstup existuje i konečný výstup, i pro $T \rightarrow \infty$. V systémové architektuře to znamená, že sada změn by měla mít za následek omezené množství změn nebo dopadů na systém, i pro $T \rightarrow \infty$. Z tohoto vyplývá, že počet dopadů na systém vytvořených nějakou změnou nesmí být spojen s velikostí systému a měl by zůstat konstantní i když bude systém růst. Jinými slovy toto znamená, že stabilita systému požaduje, aby důsledky změn neměly vliv na celý systém, ale jen na část, pro kterou jsou změny určeny. [14]

Tato teorie normalizovaných systému se skládá ze čtyř teorému:

- Separation of Concerns (SoC),
 - **Teorém 1:** Akce nebo entita může obsahovat jen jediný úkol.
- Data Version Transparency (DVT),
 - **Teorém 2:** Entita může být modifikována, jen pokud nebude mít dopad na ostatní entity, které ji používají jako vstup nebo výstup.
- Action Version Transparency (AVT),
 - **Teorém 3:** Entita akcí může být rozšířena, pokud to nebude mít dopad na volající komponenty.
- Separation of States (SoS).
 - **Teorém 4:** Akce nebo kroky ve workflow jsou od sebe odděleny udržováním stavu po každém kroku nebo akci.

4.3.1.1 Separation of Concerns

Teorém 1 (SoC) říká, že akce nebo entita může obsahovat jen jediný úkol. Pokud se podíváme na jednotlivé moduly v kapitole 4.2, jsou rozděleny podle jednotlivých úkolů. Podle funkčností, které budou podporovat bude každá funkčnost, akce nebo entita rozdělena samostatně podle jednotlivých řadičů a pohledů. Tyto řadiče a pohledy budou odděleny pro jednotlivé úkoly. Žádný z nich nebude podporovat více funkcí.

4.3.1.2 Data Version Transparency

Podle teorému 2 (DVT) je povinnost entit, které budou modifikovány (přidávání, úprava, mazání dat), aby neovlivňovaly tímto chováním ostatní entity nebo akce, které se v systému dějí. Požadavky z tohoto teorému budou zajištěny ORM, který bude přesně tuto funkčnost zajišťovat. Při použití ORM je zajištěna persistence dat. Pokud některá z entit komunikuje s aplikací, ORM je schopné zajistit pro tento objekt nebo entitu dočasnou persistenci, než se data uloží.

4.3.1.3 Action Version Transparency

Teorém 3 (AVT) požaduje, aby při rozšíření některé entity, která pracuje se systémem, nedošlo k nepříznivému dopadu na komponenty, které tyto entity volají. Systém musí být stále funkční. Toto bude zajištěno modulárním rozdělením, kde jednotlivé moduly budou mít na starost funkčnosti, které budou dále dělené na podfunkčnosti. Jelikož je použita databáze a systém bude schopen načítat objekty z databáze a dále s nimi pracovat, při rozšíření některé entity bude u volajících komponent zajištěno, aby pracovaly s přijatým objektem způsobem, který systém nerozbije. S tímto teorémem souvisí dynamičnost aplikace, která bude existovat.

4.3.1.4 Separation of States

Požadavky teorému 4 (SoS) budou zajištěny opět rozdělením do modulů a rozdělením na jednotlivé řadiče, které se budou starat o zajištění stavu po každém kroku nebo akci. Modulární rozdělení a framework, ve kterém se aplikace bude vyvíjet tento teorém budou podporovat.

4.3.2 Shrnutí vývoje podle NST

Ve světě je stále více a více softwarů, od roku 1981, kdy společnost IBM uvedením svého IBM PC na trh spustila boom ve světě počítačů, se dramaticky rozvíjel software a jeho dopad na současnou společnost [15]. Se stále přibývajícím softwarem je žádoucí, aby se vyvíjel za použití NST, kdy bude software evolvabilní, tudíž náklady na rozšíření nebo změnu funkčnosti nebudou tak vysoké, jako náklady na vývoj softwaru úplně nového. Při správném dodržování

teorémů NST je kód pro některé komponenty nebo entity znovu použitelný, bez většího nebo jen s minimálním zásahem.

Rozvoj softwaru podle NST podporuje agilní metody vývoje softwaru, jelikož teoremy vyžadují modularitu aplikace a při použití agilních metod se návrh systému při vývoji mění nebo je vytvořen zcela nový [16]. NST podporuje vývoj aplikací, kde se v pozdějších fázích vývoje vyskytují změny. Tyto změny potom nemají při dodržování NST dopad na již vytvořené komponenty systému.

4.4 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní (GUI) je důležitou součástí aplikace, jelikož uživatelé nebudou tento systém navštěvovat pravidelně, jako ostatní systémy, ale zavítají do systému jen v době dovolené v létě. Pokud bude grafická strana špatná, bude to mít negativní důsledky na používání a výnosy společnosti, která tuto aplikaci používá. Grafické uživatelské rozhraní musí pokrývat ne-funkční požadavky N 4. Byl vytvořen jednoduchý návrh, podle kterého bude vytvořen vzhled aplikace. Následně bude v testovací části po implementaci na vzhled a přehlednost kladen důraz, tak aby aplikace požadavky splňovala.

4.4.1 Persony

Persona je profil člověka, pro kterého se tvoří obsah, v tomto případě se bude jednat spíše o vizuální než obsahovou stránku. Vytvořené persony budou se systémem pracovat. Jsou vytvořeny 3 persony, které přiblíží při vývoji, pro koho bude systém tvořen. Hlavní informace o tom, kdo volnočasové areály navštěvuje a jaká je cílová skupina, jsou od zaměstnanců RY:BARu [3].

Cílovou skupinou jsou lidé od 15 do 60 let, kteří navštěvují areál na dovolené. Záměrem je vyzkoušet nabídku v bistro nebo různé aktivity, případně se informovat, co se v areálu děje.

4.4.1.1 John Doe

- Jméno: John Doe;
- Věk: 36;
- Povolání: podnikatel;
- Vzdělání: vysokoškolské;
- Rodinný stav: ženatý, 3 děti;
- Bydliště: Praha.

John se rozhodl vyrazit na krátkou dovolenou s přáteli z práce a se svými dvěma dětmi, manželka zůstává doma s novorozencem. Johnovi je jedno, kolik na dovolené nechá peněz, je bohatý dost a chce si svou dovolenou užít. Nabízí se možnost navštívit volnočasový areál, nemá o něm ale žádné informace, hledá je tedy na internetu, jelikož mu je 36 let, nemá s technologiemi problém. Doufá, že najde vše potřebné k tomu, aby si volný čas užil.

John nedbá na peníze, požaduje vysokou kvalitu a široké spektrum nabízených produktů nebo aktivit.

4.4.1.2 Jana Dvořáková

- Jméno: Jana Dvořáková;
- Věk: 19;
- Povolání: zatím studuje;
- Vzdělání: střední;
- Rodinný stav: svobodná, bez dětí;
- Bydliště: Český Krumlov.

Jana Dvořáková je fanynkou extrémních sportů. Jezdí každoročně na různé srazy sportů jako jsou Red Bull Big Air, Big Jump apod. Jednou by chtěla jezdit po celém světě a věnovat se profesionálně snowboardingu. Doteď se zúčastnila jen pár soutěží v České republice, ale všechny soutěže vyhrála. Má hodně přátel a ti ji doporučili podívat se do volnočasového areálu, kde se provozuje další z adrenalinových sportů jménem wakeboard. Nikdy to nezkoušela, ale chce se dozvědět více. Jejím záměrem je zkusit podobné sporty snowboardingu, které může provozovat v létě.

Jana je závislá na peněžní podpoře rodičů, stačí jí malý výběr kvalitních aktivit.

4.4.1.3 Pavel Beneš

- Jméno: Pavel Beneš;
- Věk: 28;
- Povolání: zaměstnanec volnočasového areálu;
- Vzdělání: základní;
- Rodinný stav: rozvedený, 1 dítě;
- Bydliště: Písek.

4. NÁVRH VLASTNÍHO ŘEŠENÍ

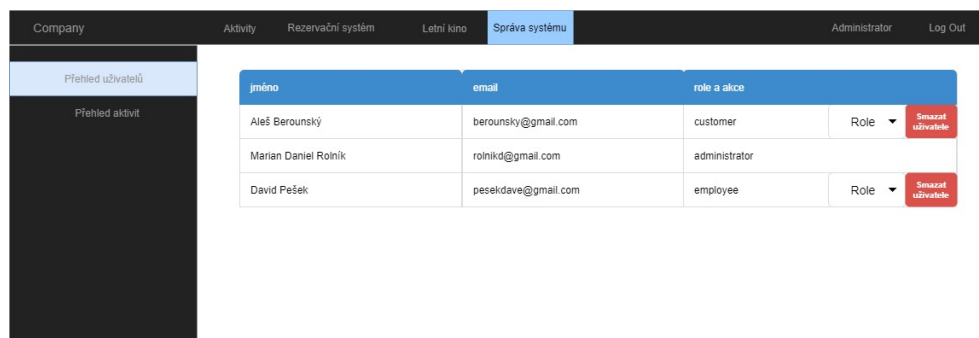
Pavel je zaměstnancem volnočasového areálu, má v péči jedno dítě, a chodí do práce každý den. Má i svoje koníčky, ale práce ho připravuje o čas se jim věnovat. Pavlovi dělají problém nové technologie, nemá o ně zájem a radši si vše píše na papír. Nejradši by si našel jinou práci, zatím ale nemá možnost, jelikož je v práci každý den a nestíhá si najít nic jiného.

Pro Pavla nejsou důležité peníze, nemá je kde utratit. Pokud by si měl vybrat nějaké aktivity, byly by to takové, které by se líbily jeho 6leté dceři.

4.4.2 Vzhled aplikace

Podle person byl vytvořen vzhled, který by těmto personám měl vyhovovat, v sekci Testování bude na základě těchto person vybráno několik testerů, kteří budou testovat funkčnost a přehlednost aplikace.

Obrázek 4.4 je příkladem, jak bude aplikace vypadat.



Obrázek 4.4: Grafické rozhraní aplikace

Implementace

V této kapitole je popsána implementace celé aplikace, jsou zde uvedeny použité technologie, průběh vývoje, zajímavé části kódu a ukázka vzhledu aplikace.

5.1 Použité technologie

V této sekci jsou rozepsány jednotlivé technologie, způsob jejich využití a důvod jejich využití.

Existuje nespočetné množství technologií, které mohou být použity. Nefunkční požadavek N 1. udává, že systém bude webovou aplikací. Tento požadavek zužuje rozsah technologií, které se mohou použít. Webové aplikace jsou populární a podle průzkumů na www.stackoverflow.com se zde nachází největší počet webových developerů. Mezi nejpopulárnější programovací jazyky pro webové aplikace u korespondentů na [stackoverflow](http://stackoverflow.com) patří javascript, java, nebo PHP [17]. Pro vývoj aplikace byly vybrány tyto technologie:

- HTML,
- PHP,
- CSS,
- MySQL,
- JavaScript.

5.1.1 HTML

HTML neboli HyperText Markup Language je programovací jazyk, který je využíván pro tvorbu webových stránek. Nejnovější verze HTML5 je z roku 2014, tento jazyk se ale stále vyvíjí a je základem téměř pro každou webovou

aplikaci. Jazyk HTML používá, jak napovídá název, hypertextové odkazy, které uživatele odkazují na další webové stránky. Obsahuje také tzv. html tagy, kterými se dá manipulovat s vloženými řádky. Např. kód ` HTML ` zajistí, aby byl text **HTML** vypsán tučně.

5.1.2 CSS

Jazyk CSS je určen pro stylování. Zjednodušeně to znamená, že pokud je v HTML vytvořen nějaký tag, je mu přidělena class a v hlavičkovém souboru HTML je přidán soubor s .css příponou, můžeme touto technologií ovlivňovat, jak se budou jednotlivé části kódu chovat. V současné době je hodně rozšířena technologie Bootstrap framework.

5.1.2.1 Bootstrap

Bootstrap je frontendový framework, který velmi šetří práci developerům. Jeho výhodami jsou jednotný vzhled, jednoduché použití a hlavně je zde možnost upravování podle potřeby díky preprocesoru LESS [18]. Díky tomuto frameworku je ulehčena práce na projektech, jelikož se vývojář může plně věnovat jiným věcem a nastavit si stylování podle frameworku Bootstrap. Odpadá zde nutnost pro vývojáře učit se a znát jazyk CSS, pokud nemá potřebu měnit již vytvořené stylování v tomto frameworku. Tento framework obsahuje rozsáhlou dokumentaci na webových stránkách www.getbootstrap.com [19].

5.1.3 PHP

PHP je programovací jazyk využívaný pro webové aplikace, pro jejichž vývoj byl i vytvořen. Tento jazyk je velmi rozšířený, není tak populární jako javascript, který v současnosti využívá nejvíce vývojářů [17], ale v PHP byly vytvořeny webové aplikace jako je Facebook, Yahoo, Wikipedie [20]. Je zde možnost psaní ve frameworku nebo v tzv. plain PHP. Obě možnosti mají své výhody a nevýhody. Jednou z výhod používání frameworku je, že je předem určená struktura psaní, existuje vytvořená dokumentace k většině frameworků a pokud má framework silnou bázi uživatelů, je možnost dohledat mnoho řešení k problémům, které už někdo měl a úspěšně vyřešil. Další výhodou používání frameworků je jejich zabezpečení, k většině frameworkům existují napsané konfigurace, které se starají o bezpečnost uživatelů na stránkách, pokud by někdo používal plain PHP, musel by tyto funkčnosti zajistit a napsat sám.

Byly porovnány 3 frameworky:

- **Laravel**,
- **Symfony**,
- **Codeigniter**.

5.1.3.1 PHP Framework Codeigniter

Tento framework byl uveden do provozu v roce 2016 a byl zde od začátku fungování PHP aplikací. V Codeigniter je napsáno mnoho aplikací, které už v něm i napsané zůstanou, pokud ale někdo bude vytvářet nové aplikace, použije spíše jiné, modernější frameworky.

Struktura tohoto frameworku je MVC, což by bylo výhodné pro vyvíjenou aplikaci. Jeho struktura byla vytvořena na základě plně objektově orientovaného programování, ale uživatelé tento framework používali podle sebe. Bohužel zde chybí podpora pro Composer, který je užitečný v generování kódu u jiných frameworků a chybí zde i podpora Middleware, což značně ztěžuje komunikaci mezi objekty.

Jelikož Codeigniter nepodporuje Composer, u databázových aplikací musí vývojář přistoupit k využití knihoven třetích stran, pro komunikaci s databází. Codeigniter ovšem nabízí vlastní ORM.

Codeigniter neobsahuje templatovací systém, který by ulehčil práci, vývojář musí tedy vše psát v plain PHP, nemůže si vytvořit žádné vzory. [21]

5.1.3.2 PHP Framework Laravel

PHP framework Laravel je dnes nejpoužívanějším frameworkem na světě. Je jedním z open-source frameworků na trhu a byl vytvořen v roce 2011. Zajímavé je, že je postaven na frameworku Symfony. Je to opět framework, který podporuje vývoj MVC aplikací.

Tento framework poskytuje jednoduchou práci s databází, nabízí pokročilé funkce s dotazováním nad databází. Kromě toho poskytuje vlastní VCS (Version Control System), který pomáhá s managementem databáze.

Obsahuje vlastní systém autentizace, u kterého stačí nastavit konfiguraci, databázi, řadiče a pohledy, aby tento systém začal fungovat. [22]

5.1.3.3 PHP Framework Symfony

PHP framework Symfony je další z open source frameworků, který byl spuštěn v roce 2011. Podporuje vývoj aplikací s MVC architekturou.

Tento framework je v aktuální verzi Symfony4, která byla vydána na začátku roku 2018. Tento framework se stále vyvíjí a má velkou uživatelskou základnu. Dokumentace je dobře sepsaná a obsáhlá. Na stackoverflow.com mají jak Laravel, tak Symfony téměř stejný počet přispěvatelů.

Hlavní výhodou frameworku Symfony je jeho možnost modularity. V předchozí verzi – Symfony3 – byla modularita řešena pomocí Bundles. Rozdělení do modulů je obsaženo v návrhu, proto je tento framework v tomto ohledu výhodný. V novějším Symfony4 není vytvořeno dělení do Bundles, ale pokud je záměr vytvářet modulární aplikaci, dá se projekt rozdělit do složek, které reprezentují Bundles a používají svůj vlastní namespace.

Pokud je aplikace v Symfony implementovaná dobře, její rychlost se rovná sumě všech individuálních funkcí, které obsahuje, jinými slovy je její rychlost závislá na tom, jak vývojář zpřístupní funkcím přistupovat k jádru frameworku. Je tedy rychlejší než mnoho jiných frameworků, pokud se použije správně.

Symfony poskytuje možnost použití komponent, které již byly vytvořeny, znovu. Je zde vysoká míra možnosti opakovatelnosti kódu.

Komponenty, které Symfony poskytuje mohou být použity i samostatně, dokonce i v jiných frameworkách. [22, 23]

5.1.3.4 Použitý framework

Z důvodu požadavků aplikace na modularitu, vysokou rychlost a jednoduchost používání byl z těchto tří frameworků vybrán framework **Symfony**. Poskytuje pro potřeby aplikace nejlepší podmínky a jeho obsáhlá dokumentace a vysoká míra přispěvatelů napomůže k lepším výsledkům implementace.

5.1.4 MySQL

Databáze MySQL je jednou z nejpoužívanějších databází pro webové aplikace, kterou hostingsy v České republice podporují. Tato databáze je relační, je možnost ji i využít s frameworkem Symfony. Framework Symfony pro komunikaci s databází používá integrovaný ORM systém Doctrine.

5.1.4.1 ORM Doctrine

ORM Doctrine je používán frameworkem Symfony. ORM je programovací technika pro konvertování dat mezi relačními a objektově orientovanými datovými modely.

5.2 Průběh implementace

V této kapitole bude popsán průběh implementace a některé problémy při implementaci.

5.2.1 Generování kódu

V použitém frameworku Symfony je možnost generování velkého množství kódu, které je velkou výhodou pro větší projekty.

5.2.1.1 Nastavení frameworku

Pro nastavení a zprovoznění frameworku je potřeba mít nainstalovaný Composer, pomocí kterého je generován kód a importovány součásti systému, které

jsou nutné pro vývoj. Pro užívání frameworku Symfony4 je nutné mít PHP 7.1 nebo vyšší.

Composer umožňuje vytvoření struktury pro vývoj aplikace jednoduchým konzolovým příkazem:

```
composer create-project symfony/website-skeleton
název-projektu
```

Díky Composeru je možno vkládat do frameworku různé balíčky a funkce, které se budou později používat příkazem:

```
composer require název-balíčku
```

V aplikaci jsou použity tyto balíčky:

- doctrine,
- swiftmailer,
- server,
- make,
- router,
- security,
- debug.

5.2.1.2 Generování řadičů

Po vytvoření kostry a zahrnutí všech potřebných funkčností se mohou pomocí příkazů

```
bin/console make:controller
>controllerName
```

generovat kódy pro řadiče. Po použití tohoto příkazu se vytvoří nový řadič

```
/src/Controller/ControllerNameController.php
```

se jménem, které je předáno při použití příkazu make. Dále je vytvořen pohled

```
/src/templates/controller_name.html.twig
```

rozšiřující základní pohled

```
/src/templates/base.html.twig
```

Base.html.twig je základní jednotkou celého projektu a je v ní uloženo, jak má celý projekt vypadat. Jelikož Symfony využívá šablony `.twig`, existuje zde velké množství přizpůsobitelných částí, které se můžou upravit v rozšiřujících šablonách.

```
1 {% block názevbloku %}  
2 {% endblock %}
```

Kód 5.1: base.html.twig blok

Anotace v Kódu 5.1 umožňuje v šablonách rozšiřujících základní šablonu vytvářet dodatečný kód, který bude platit jen v určitém pohledu. Například se vyplatí vytvořit v základní šabloně blok `javascripts`, kde se pro jednotlivé pohledy mohou vkládat skripty psané v jazyce JavaScript.

Bloková anotace je v implementaci často využívána, dá se zanořovat a jednotlivé šablony mohou rozšiřovat jiné šablony. Většina modulů v implementaci obsahuje šablony, které rozšiřují šablonu základní a samy jsou rozšiřovány pohledy jednotlivých stran v modulu.

5.2.1.3 Generování entit a formulářů

Stejným principem jsou vytvářeny i entity, se kterými pracuje Doctrine2 a formuláře, které jsou používané řadiči. Řadič vygeneruje pohled a odešle ho do šablony, ve které se formulář zobrazí.

```
bin/console make:entity  
bin/console make:form
```

Použitím příkazu pro entitu se vygeneruje Kód 5.2.

V Kódu 5.2 jsou vidět některé věci, které framework Symfony používá. Anotace

```
*@var int
```

říká, jaký typ atributu je popsán. Anotace

```
@ORM\Column
```

s parametry udává jméno, typ sloupce. Dají se přidávat

```
"length"  
"nullable"
```

atributy, které dále upravují nastavení entity. Entita v Symfony je defaultně nenulová.

```
1 <?php
2
3 namespace App\Entity;
4
5 use Doctrine\ORM\Mapping as ORM;
6
7 /**
8  * @ORM\Entity(repositoryClass="App\Repository\ActivityRepository")
9  */
10 class Activity
11 {
12     /**
13      * @var int
14      *
15      * @ORM\Column(name = "id", type = "integer")
16      * @ORM\Id
17      * @ORM\GeneratedValue(strategy = "AUTO")
18      */
19     private $id;
```

Kód 5.2: Příklad entity

5.2.1.4 Shrnutí generování kódu

Tato funkce je velmi užitečná, bohužel s sebou nese některé nedostatky. Pokud je kód vygenerovaný a není napsaný vlastními silami, může člověk, který s frameworkem tolik nepracuje ztratit přehled, jaký kód se generuje a co znamená. Toto je problém na začátku práce s frameworkem, kdy zkušenosti s technologií nejsou velké a je vygenerováno plno nepotřebného kódu. Postupem času se ale většina generovaného kódu smaže nebo přepíše, aby vyhovoval potřebám aplikace.

Hlavní výhodou je samostatné generování namespaceů a vkládání potřebných závislostí, bez kterých by aplikace nefungovala.

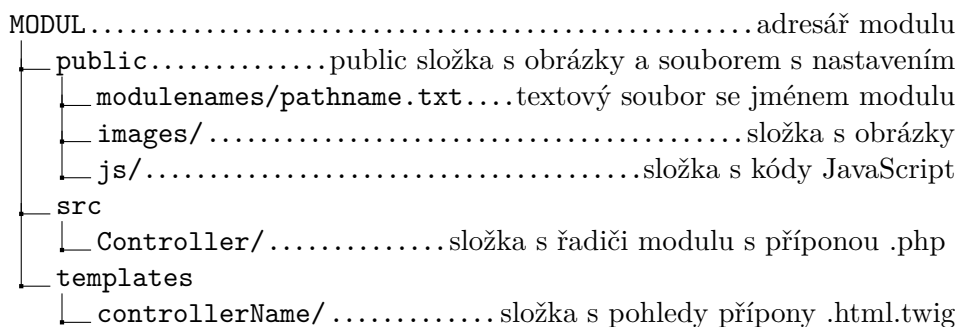
5.2.2 Implementace dynamického menu

V aplikaci je implementováno dynamické menu, které se mění podle toho, jaké moduly jsou přidány. Toto je důležitá funkčnost, pokud je cílem vyvíjet aplikaci modulárně.

Tato funkčnost je vytvořena tzv. embedded controllerem neboli zanořeným řadičem, kde kód pro menu, které je obvykle v každém pohledu aplikace (kód by tedy měl být obsažen v základní šabloně `base.html.twig`), je umístěn v samostatné šabloně a má samostatný řadič, který generuje pohled s polem, ve kterém jsou umístěny názvy pro jednotlivé odkazy a jejich cesty. Tento řadič je volán ze základní šablony `base.html.twig`.

Pro tuto funkčnost je nutné zavést určité konvence. Pokud se přidá nový modul, jeho struktura musí být podle Obrázku 5.1.

5. IMPLEMENTACE



Obrázek 5.1: Adresář pro přidání nového modulu

Pro správnou funkčnost je potřeba, aby jednotlivé cesty u modulů byly nastaveny správně. Jedná se o hlavní cestu, která se nastavuje v anotaci podle Kódu 5.3.

```
1 class ActivitiesController extends Controller
2 {
3     /**
4      * @Route("/activities", name="tl_activities_left")
5      */
6     public function index()
7     {
```

Kód 5.3: Příklad entity

Tato cesta je napsaná v proměnné

```
name="tl_activities_left"
```

a jako oddělovače má podtržítka, která oddělují jednotlivé části cesty. První položkou je tag, který říká, která cesta má být hlavní položkou menu, druhý tag říká, jak se jmenuje soubor s jednotlivými nastaveními a třetí tag říká, na jaké straně menu se má modul zobrazovat. Pokud se přidá nový modul a jsou dodržena tato pravidla, aplikace sama vygeneruje potřebné odkazy do menu.

```
1 foreach ($availableApiRoutes as $routename){
2     $parsearray = explode("_",$routename);
3     if($parsearray[0] == "tl" && $parsearray[2] == "left" ){
4         $routeactuellname = 'name_' . $routename;
5         $filepath = "./modulenames/$parsearray[1].txt";
6         $myfile = fopen("$filepath","r");
7         $leftroutes[] = ['routename'=>$routename,'routeactuellname'
=>fread($myfile,filesize("$filepath"))];
8         fclose($myfile);
9     }
10 return $this->render('menu/menu.html.twig', ['leftroutes'=>$
leftroutes, 'rightroutes'=>$rightroutes]);
```

Kód 5.4: Ukázka z menuController.php

V šabloně je potom jednoduchý cyklus, kterým se vygenerují odkazy podle předaných hodnot.

```

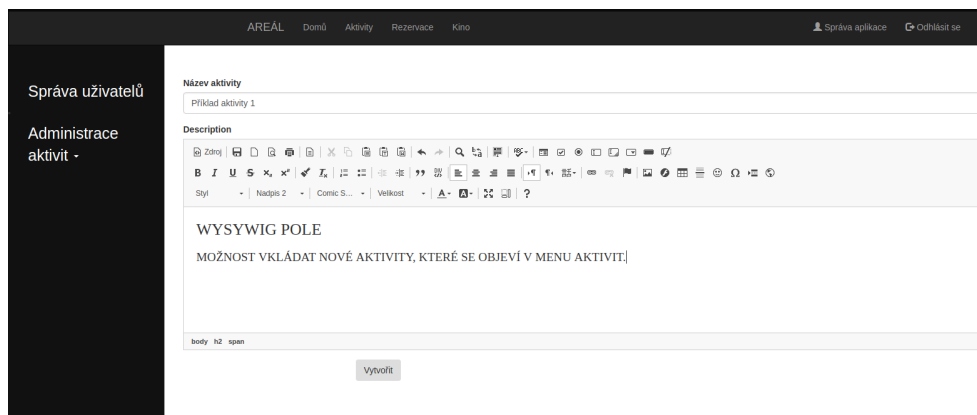
1
2 {% for routename in leftroutes %}
3     {% set name = "" ~routename.routeactuallname %}
4     {% set path = "" ~routename.routename%}
5     <li class="{% if app.request.attributes.get('_route') == name %}
6     active{% endif %}" ><a href="{% path(path) %}">{{ name }}</a></
    li>
7 {% endfor %}

```

Kód 5.5: Kód z menu.html.twig

5.3 Aktivity a jejich přidávání

Pokud je přítomen modul administrátora, je zde naimplementovaná možnost přidávání nových aktivit, které se po vytvoření objeví jako nový odkaz v menu modulu aktivit. Nejprve se aktivita vytvoří podle obrázku 5.2.



Obrázek 5.2: Vzhled přidávání aktivity

Tato aktivita se uloží do databáze a po vstoupení do modulu aktivit, řadič načte aktivity z databáze a vytvoří pro ně odkazy, jak je v kódu 5.6

```

1 class ActivityController extends Controller{
2     /**
3     * @Route("/activities/activity/{id}", name="activity")
4     */
5     public function index($id){
6         $em= $this->getDoctrine()->getRepository('App:Activity');
7         $activities = $em->findAll();
8         $activity = $em->find($id);
9         return $this->render('activities/activity/activity/index.
    html.twig', ['activities'=>$activities, 'activity'=>$activity]);

```

Kód 5.6: Ukázka z ActivityController.php

Následně je možnost v menu Aktivity nalézt tento odkaz s vyplněnými poli, jak je vidět na Obrázku 5.3.



Obrázek 5.3: Vhled aktivity po předchozím přidání

5.4 Přidávání nových modulů a zprovoznění aplikace

5.5 Nové moduly

Pro přidání nových modulů je potřeba, aby byla do kořenového souboru aplikace vložena struktura, která je popsána na Obrázku 5.1. Správce souborů automaticky spojí jednotlivé soubory správně do složky a modul je připraven k použití. Důležité je také dodržovat konvence, které jsou v Obrázku 5.1 také popsány.

5.6 Zprovoznění aplikace

Pro zprovoznění aplikace se musí provést řada kroků [24]:

1. adresář s aplikací nahrát na server nebo do lokálního úložiště, podle toho, kde bude aplikace spuštěna;
2. nainstalovat potřebné závislosti balíčků, ty ale mohou být nainstalovány před nahráním na server nebo na lokální disk;
3. nastavit databázi a provést migraci dat nebo entit do databáze, aby aplikace mohla databázi používat;
4. vyčistit cache složku.

Některé servery, na které se aplikace bude nahrávat se od sebe liší. Je důležité dbát na proměnné prostředí, které jsou v lokálním vývoji umístěné v souboru `.env`. Podle toho, kam je potřeba nahrát aplikaci se tyto proměnné můžou lišit a můžou být umístěné jinde. Pokud je potřeba nahrát aplikaci do produkce, minimálně se musí změnit proměnná `APP_ENV="prod"` v souboru `.env`.

Testování

Tato kapitola pojednává o testování, které bylo provedeno na bázi uživatelských testů podle vytvořených scénářů. Testování proběhlo na 4 různých lidech, ve věkové kategorii 15-60 let, kteří spadají do cílové skupiny areálu RY:BAR [2].

Testování probíhalo uživatelsky, jelikož hlavním cílem je základní funkčnost a orientace uživatele v systému, automatizovanými testy by nedošlo k výsledkům, které by poskytly potřebnou uživatelskou odezvu.

6.1 Průběh testování

Cílová skupina byla pozvána do místnosti, kde se nacházel připravený počítač. Před samotným testováním proběhl s testery krátký rozhovor, pro uvolnění nálady. Nebylo jim vysvětleno, k čemu aplikace slouží. Bylo jim pouze sděleno, že existuje volnočasový areál, který má webové stránky.

Následně se jednotlivci postupně střídali v provádění scénářů. Aby testeři neviděli rozhraní aplikace, střídali se v testování odděleně.

6.2 Scénáře

Zde budou uvedeny možné scénáře, kterými by běžný uživatel mohl projít, když bude aplikaci používat. V jednotlivých scénářích je:

- cíl,
- předpokládaný průběh,
- předpokládaný výsledek,
- průběh testování,
- změny provedené v systému po testování.

6.2.1 Vytvoření profilu

Název testu	Vytvoření profilu
Účel testu	Otestovat, zda uživatel nalezne registrační tlačítko a zda se přihlásí do systému, který uloží data vložené uživatelem
Čas	2 minuty
Předpokládaný průběh	Uživatel klikne na tlačítko s registrací, zobrazí se formulář, který vyplní. Registruje se tlačítkem registrovat. Systém data uloží a uživatele přihlásí.
Předpokládaný výsledek	Data jsou uložena v databázi a uživatel je přihlášen na svém účtu. Lze ověřit nahlédnutím do databáze nebo kliknutím na tlačítko Profil.
Průběh testu	Ve všech případech test proběhl bez problémů. Naplnil předpokládaný průběh testu.
Poznámky	Jeden tester měl menší problém s nalezením registračního tlačítka.

Tabulka 6.1: Scénář č. 1

Tento test proběhl velmi dobře, jelikož to byl první prováděný test, zmínili se 3 ze 4 testerů o vzhledu aplikace. Líbilo se jim, že je aplikace vzhledově jednoduchá ale postrádali více obrázků a barevnost.

Tato aplikace je teprve první verzí, proto bude téma vzhledu řešeno v následujícím vývoji. Jedná se o Bootstrap stylování, které se dá jednoduše měnit (změnou kódu v souboru `mystyles.css`) a bude zasahovat do celého systému, jelikož je aplikace navržena podle šablon, které rozšiřují základní šablonu.

6.2.2 Přihlášení do systému po zapomenutí hesla

Název testu	Přihlášení do systému po zapomenutí hesla
Účel testu	Otestovat, zda systém odešle správně nové heslo na email uživatele. Otestovat, zda je uživatel schopen se přihlásit do systému pod svým emailem, pokud zapomněl heslo.
Čas	5 minut
Předpokládaný průběh	Uživatel se bude snažit přihlásit do systému. Po stisknutí tlačítka přihlášení je zde odkaz na formulář, pokud uživatel heslo zapomněl. Po odeslání formuláře se správným emailem přijde uživateli na jeho email nové vygenerované heslo, se kterým se přihlásí do systému.
Předpokládaný výsledek	Uživatel našel formulář pro zapomenuté heslo. Uživatel si nechal poslat heslo a přihlásil se do systému. Systém vygeneroval nové heslo, poslal ho uživateli na email. Následně ho zakódoval a uložil do databáze.
Průběh testu	Test proběhl ve 3 ze 4 případů v pořádku. Čtvrtému testerovi nepřišel email s heslem, po odeslání nového formuláře email po pár minutách dorazil. Uživatelé se podle předpokladu s novým heslem do systému dokázali přihlásit.
Poznámky	Email, ze kterého se odesílají hesla uživatelům je nastaven na osobní email pro účely vývoje. V produkčním prostředí bude použit emailový hosting na serveru, se kterým by neměl být problém.

Tabulka 6.2: Scénář č. 2

Výsledky tohoto testu byly v pořádku, tudíž není potřeba nijak měnit tuto funkčnost. V aplikaci je nastavený osobní email (Gmail) s heslem, který se do aplikace Gmail přihlásí a až poté odešle mail. Toto může mít vliv na rychlost. Testerům nepřišly maily ihned, ale až po chvíli, tudíž se může stát, že některý z uživatelů odešle formulář vícekrát, v domněnání, že email odeslán nebyl a pak uživateli přijde více emailů a vytvoří spam.

Do aplikace byla přidána zpráva, která oznámí uživateli, zda byl email úspěšně odeslán. Pokud nebyla nalezena emailová adresa v databázi, zpráva oznámí uživateli, že email neexistuje.

6.2.3 Vytvoření a zrušení rezervace

Název testu	Vytvoření a zrušení rezervace uživatelem
Účel testu	Otestovat, zda systém správně uloží rezervaci uživatele a zda je možné rezervaci zrušit. Otestuje také, zda je uživatel schopen najít rezervační systém a vyznat se v něm.
Čas	10 minut
Předpokládaný průběh	Uživatel si zobrazí rezervační modul ve webové aplikaci, dozví se, že je potřeba přihlášení pro vytvoření rezervace. Budou mu předložena přihlašovací data. Uživatel se přihlásí, vybere si datum a hodinu rezervace a rezervaci potvrdí.
Předpokládaný výsledek	Uživatel našel rezervační systém v aplikaci, úspěšně se přihlásil s fiktivními daty, které mu byly poskytnuty, vytvořil rezervaci. Systém úspěšně uloží rezervaci k uživatelskému účtu a odešle email s potvrzením rezervace.
Průběh testu	Uživatelé neměli problém s hledáním rezervace, testy probíhaly tak, jak bylo předpokládáno v průběhu.
Poznámky	Výhrady k rezervačnímu systému

Tabulka 6.3: Scénář č. 3

Tento test probíhal velice dobře. Testerů po předchozích testech už věděli, jak se v aplikaci pohybovat. Byly zde ale výhrady k rezervačnímu systému. Hlavní výhradou bylo, že je provedený v tabulce a ne graficky, což není velice uživatelsky přívětivé.

V implementaci je rezervační systém připraven na budoucí rozvoj grafického rozhraní, které bude pro uživatele příjemnější. Jsou implementovány tabulky a propojení v databázi a rezervační modul se v budoucnu bude rozvíjet.

6.2.4 Změna role uživatele administrátorem

Název testu	Změna role uživatele administrátorem
Účel testu	Otestovat přehlednost modulu správy systému administrátorem a otestovat správné fungování databáze s uživateli.
Čas	10 minut
Předpokládaný průběh	Tester se přihlásí s přidělenými daty do systému, nalezne modul správy systému, zobrazí si všechny uživatele a změní roli uživatele.
Předpokládaný výsledek	Uživatel je schopen se přihlásit do systému, zpřístupní se mu administrátorská záložka, na tuto klikne, dále se mu zobrazí další menu s uživatelskými profily, kde najde rozbalovací kolonku s rolemi, vybere roli a uloží. Systém si do databáze uloží novou roli uživatele a oznámí testerovi, že změna role proběhla.
Průběh testu	Ze 4 uživatelů pouze 2 se chtěli přihlásit do systému jako administrátoři, zbylí 2 hledali na úvodní stránce systému. Po kliknutí na přihlášení jim byl předán lísteček s přihlašovacími daty administrátora. Tyto bez problému vyplnili, ale všichni hledali, kde se nachází položka úpravy rolí uživatelů. S pomocí ji našli a změnili role uživatele bez problému.
Poznámky	Při testování nebyli testeři nijak naváděni k provádění určitých úkolů, nebylo jim explicitně sděleno, kde můžou potřebnou funkčnost aplikace nalézt. Byly jim pokládány otázky, kde by čekali tuto funkčnost.

Tabulka 6.4: Scénář č. 4

Tento test neproběhl tak, jak bylo očekáváno. Souvisí to i s tím, že tento test je určen pro administrátory systému, kterým musí být sděleno a vysvětleno vývojářem nebo někým, kdo již systém zná, jak pracovat v administrátorském prostředí. Po nalezení správného odkazu se neobjevil problém.

V dalším vývoji aplikace bude barevně zvýrazněno, kde se jaké položky nachází, aby bylo administrátorům jasné, jak postupovat.

6.2.5 Vytvoření nové aktivity administrátorem

Název testu	Vytvoření aktivity administrátorem
Účel testu	Opět otestovat přihlášení administrátora do systému a správu aktivit
Čas	15 minut
Předpokládaný průběh	Tester se přihlásí do systému pod administrátorským účtem, nalezne modul správy systému, nalezne správu aktivit a zvolí přidání nové aktivity. Zobrazí se mu wysiwyg okénko, kam zadá data podle přidělené předlohy. Tlačítkem uloží aktivitu.
Předpokládaný výsledek	Tester je schopen se přihlásit do systému, zpřístupní se mu administrátorská položka systému. Vytvoří se nová aktivita v databázi. V základním menu se přidá nový odkaz na aktivitu.
Průběh testu	Jelikož v předchozím testu byli testeři seznámeni s administrátorským prostředím aplikace, nebyl pro ně problém najít odkaz, který je navede na správnou stránku aplikace. Zde se objevil wysiwyg editor, ve kterém měli testeři možnost použít různé funkce tohoto editoru.
Poznámky	Nejvíce času zabralo vytváření stránek ve wysiwyg editoru.

Tabulka 6.5: Scénář č. 5

Tento test proběhl u všech testerů v pořádku. Někteří jen měli výhradu k tomu, jak se vytváří nová aktivita v editoru. Po vytvoření vypadala stránka přesně jak byla vytvořena v editoru.

Pro budoucí vývoj aplikace bude důležité oddělit jednotlivé celky stránky aktivity, kde bude možnost upravovat vzhled podle jednotlivých polí s možností vložení obrázků. Pro informační účely aplikace je zatím wysiwyg editor dostačující.

6.3 Shrnutí testování

Testování probíhalo v pořádku, nevyskytly se skoro žádné problémy. Po testování aplikace byly testerům položeny otázky, jak se jim aplikace líbila a jak jim připadala práce s aplikací. Odpovědi byly podobné a bylo v nich obsaženo:

- Aplikace je příliš formální.
- Aplikace neobsahuje mnoho obrázků.
- Hlavní stránka vypadá dobře, zbytek není moc přívětivý.
- Menu je přehledné, pro jednotlivé moduly je umístěno nalevo, což usnadňuje orientaci v systému.

Tyto ohlasy byly očekávané a pro další rozvoj systému bude rozšířeno stylování a budou přidány skripty, které v aplikaci budou podporovat animace. Animovanost webové aplikace je důležitým faktorem pro uživatelskou spokojenost.

Ekonomicko-manažerské zhodnocení a evolvabilita řešení

V této kapitole jsou popsány náklady na systém, jeho přínosy a následně je hodnocena i evolvabilita systému a přidávání nových modulů do aplikace.

7.1 Přínosy systému

Systém by byl velkým přínosem pro areály, které by ho využívaly. Po zavedení systému by měly tyto areály možnost prezentovat se na webových stránkách.

Dynamickým řešením webové aplikace je přispěno k tomu, aby zde byly menší zásahy vývojářů a vybraní administrátoři systému ho mohli upravovat podle svých představ.

Velkou výhodou systému je také rezervační systém, který může pomoci tomu, aby nevznikaly zbytečné dohady o tom, kdo si aktivitu v areálu zaplatil první nebo kdo má zrovna přednost.

Celkově je pro areály, které systém nemají, výhodné si tento systém pořídit, ulehčí jim práci a poskytne podporu v některých vybraných oblastech.

7.2 Náklady na vývoj a zavedení systému

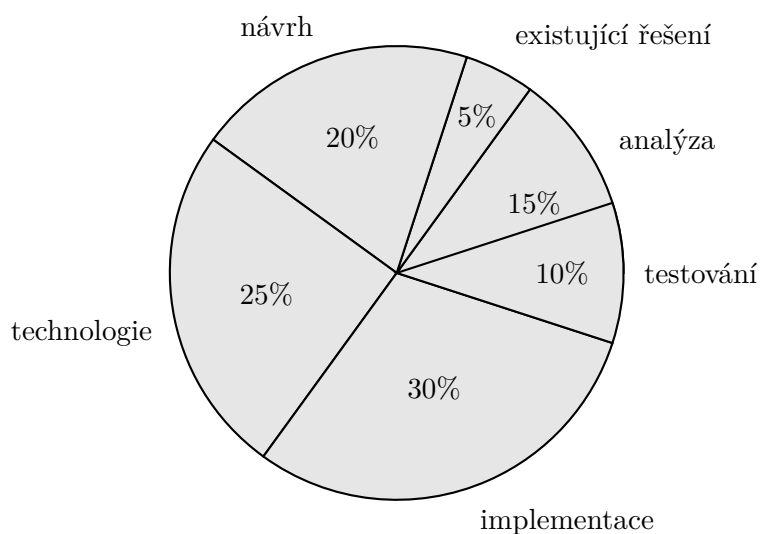
7.2.1 Náklady na vývoj systému

Pro výpočet nákladů na vývoj systému je použita sazba 350,- na jednu člověkohodinu. Celková náročnost práce je odhadována na 350 hodin, která zahrnuje:

- analýzu areálu (15 % z celkové práce),
- zkoumání existujících řešení (5 % z celkové práce),

7. EKONOMICKO-MANAŽERSKÉ ZHODNOCENÍ A EVOLVABILITA ŘEŠENÍ

- návrh aplikace (20 % z celkové práce),
- hledání potřebných technologií a učení se jejich používání (25 % z celkové práce),
- implementace (30 % z celkové práce),
- testování (10 % z celkové práce).



Náklady na vývoj aplikace činí tedy 122 500,-.

7.2.2 Náklady na zavedení systému

Systém je vytvořen jako webová aplikace. Společnost, která bude tento systém využívat musí zajistit potřebný webhosting, který se neplatí jednorázově, ale paušálně, většinou každý rok. Systém byl vytvořen tak, aby ho podporovaly běžné webhostingy v České republice.

Mezi tyto patří například:

- Wedos,
- Český hosting,
- ONEbit,
- Savana,
- Ebola.

Z hodnocení na webové stránce www.5nej.cz vychází pro rok 2018 nejlépe webhosting Wedos [25]. Cena za registraci domény .cz činí 151,-. Roční platba za službu je 1452,- s DPH.

7.2.3 Celkové náklady

Prvotní náklady na systém jsou ve výsledku 122 651,-. Po zavedení systému je nutné počítat také s náklady spojenými se školením zaměstnanců. Jelikož je práce se systémem přímočará a jednoduchá a náklady na jednu člověkohodinu zaměstnanec areálu činí 150,-, cena za školení pro tři zaměstnance po dvou hodinách bude

$$n = 3 \times 150 \times 2 = 900$$

Do roční sazby bude počítána i správa systému vývojářem, za kterou bude účtováno 10 000,- ročně.

Dohromady náklady na zavedení systému a školení činí **123 551,-** a roční sazba činí **11 752,-** pokud za rok přibude jeden nový zaměstnanec, který se musí se systémem naučit pracovat. Zaměstnanci nebudou dostávat větší plat za používání systému, na používání nebudou náklady existovat.

7.3 Návratnost investice

Dle předpokladu se existencí informační webové stránky a rezervačního systému do areálu přiláká 30 nových zákazníků. Zákazníci většinou v okolí areálu nějaký čas pobývají a za dobu svého pobytu v areálu utratí průměrně okolo 1800,- [3].

Je počítáno s tím, že každým rokem, po dobu 5 let přibudou 3 noví zákazníci. Od dalších let zákazníci nebudou přibývat a počet zákazníků, využívajících tento systém se ustálí.

rok	náklady v Kč	výnosy v Kč
0	123 551	0
1	11 752	54 000
2	11 752	59 400
3	11 752	64 800
4	11 752	70 200
5	11 752	75 600
6	11 752	75 600
7	11 752	75 600
8	11 752	75 600
9	11 752	75 600
10	11 752	75 600

Tabulka 7.1: Náklady a výnosy v 10 letech

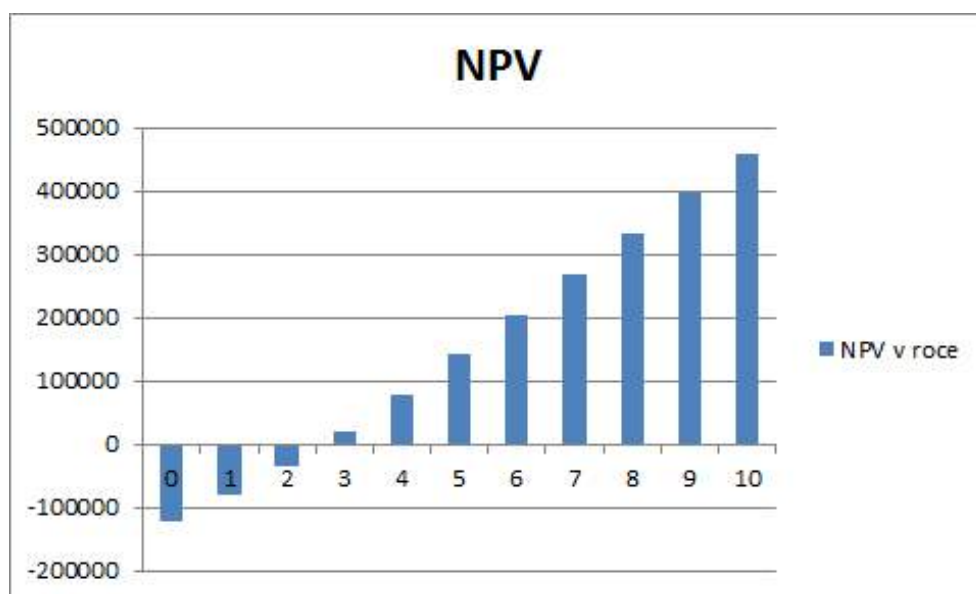
Z Tabulky 7.1 je vytvořen tok peněz (CF - cash flow) za 10 let používání systému a následně je vytvořena veličina NPV (Net Present Value - současná čistá hodnota), která zohledňuje hodnotu peněz v budoucím období, jelikož investiční rozhodování má dlouhodobé účinky. Náklad kapitálu je zde zohledněn tzv. diskontní sazbou, která charakterizuje časovou hodnotu peněz [26]. Pokud jsou nějaké peníze vloženy do banky, banka tyto peníze za rok zhodnotí podle úrokové sazby. Vzorec s tímto zhodnocením počítá a v dalších letech znehodnocuje tyto zainvestované peníze.

NPV je počítáno tímto vzorcem [26]:

$$NPV(i, N) = \sum_{t=0}^N \frac{CF_t}{(1+i)^t}$$

N je doba životnosti projektu, CF_t je cash flow v daném roce a $i = 0,5\%$ je diskontní sazba v České republice [27].

Pro jednotlivé roky byla vypočítána podle tohoto vzorce hodnota NPV a jak je vidět na Obrázku 7.1, z grafu plyne, že by se měla investice vrátit do tří let od zavedení systému.



Obrázek 7.1: NPV podle roků využívání systému

7.4 Porovnání se stávajícími řešeními

Problémy, které tento systém řeší, jsou specifické a komplexní. Aplikace, která by podporovala všechny funkčnosti a byla evolvabilní, v závislosti na požadavcích zákazníka, neexistuje. Existují jen dílčí aplikace, které podporují jednotlivé části systému. Pokud by se systém skládal z více částí, které by spolu nesouvisely, ale byly by podporovány aplikacemi různých společností, nebyla by zde možnost zajistit kompaktnost aplikace a jednotný vzhled.

Vložení nové funkčnosti by se tedy mělo nejlépe řešit implementací nového modulu a jeho vložení, jak je popsáno v kapitole 5.4.

7.5 Rozšiřitelnost modulů

V této kapitole je popsáno, jak by mohly být jednotlivé moduly rozšířeny, jaké funkčnosti by mohly přibýt.

7.5.1 Moduly

7.5.1.1 Základní modul

Možnosti rozšíření základního modulu:

- možnost dynamičnosti úvodní strany, přidání polohy administrátorem, bez dopadu na systém;
- podpora pro různé stylování aplikace.

7.5.1.2 Modul uživatelského rozhraní

Možnosti rozšíření modulu uživatelského rozhraní:

- přidání dalších dodatečných informací např. fotografie v profilu.

7.5.1.3 Modul rezervačního systému

Možnosti rozšíření rezervačního modulu:

- lepší propojení databáze;
- kalendář rezervací;
- notifikace rezervace emailem;
- kalendář, ve kterém budou spravovány rezervace;
- možnost výběru více aktivit pro rezervaci.

7.5.1.4 Modul letního kina

Možnosti rozšíření modulu letního kina:

- přidání kalendáře s filmy;
- lepší vzhled, možnost rezervace sedadel;
- odeslání novinek emailem, pokud se objeví nový film.

7.5.2 Shrnutí

Aplikace je implementovaná tak, aby pokryla základní funkčnosti, které se budou v dalším rozvoji aplikace vyvíjet. Další vývoj aplikace bude z hlediska rozšiřitelnosti modulů jednoduchý, jelikož je aplikace vyvíjena modulárně a na základě NST.

7.6 Nové moduly a rozšíření v budoucím vývoji

Do aplikace bude možné jednoduše přidávat moduly tak, jak je popsáno v kapitole 5.4. V budoucím rozvoji je zde prostor pro mnoho nových modulů a rozšíření. Přidávání nových modulů je jednoduché. V aplikaci by se v budoucnosti mohly objevit tyto moduly a rozšíření:

- **Modul recenzí**, ve kterém budou moci uživatelé hodnotit areál pomocí stupnice a napsat svůj komentář.
- **Modul skladu**, který bude sloužit zaměstnancům pro správu skladových položek v areálu.
- **Modul ceníku**, který bude interaktivní a po upravení položek zákazníkovi spočítá celkovou sumu, kterou v areálu utratí.
- **Modul zaměstnaneckých archů**, který by ušetřil spotřebu papíru, a kterým by se předešlo ztrátě archu papírového.
- **Notifikace emailem**, které uživatelům budou chodit, pokud se objeví nová aktivita nebo když se změní zásadní komponenta aplikace.

Nové moduly, které budou do nového systému vloženy musí být napsány v PHP frameworku Symfony, za použití stejných postupů, které byly použity v této práci.

Závěr

Cílem práce bylo navrhnout, implementovat a otestovat webovou aplikaci pro volnočasové areály. Návrh a implementace aplikace měly být v souladu s Normalized Systems Theory. Poté mělo být provedeno ekonomicko-manažerské zhodnocení vytvořené aplikace.

Nejprve byla provedena analýza volnočasového areálu RY:BAR za použití formálního konceptuálního modelování. Bylo provedeno porovnání stávajících řešení a došlo se k závěru, že žádné z těchto řešení není zcela použitelné pro specifické areály, ve kterých se mění skladba aktivit.

Bylo navrženo vlastní řešení, které podporuje principy Normalized Systems Theory a je založeno na modulární webové aplikaci. Tento návrh je vytvořen tak, aby podporoval evolvabilitu systému, rozšiřování aplikace a modifikaci pro specifické areály.

Tento návrh byl implementován jako webová aplikace za použití PHP frameworku Symfony. Byly popsány veškeré technologie použité v implementaci a byla zohledněna modularita aplikace podle návrhu. Aplikace podporuje informační funkci pro uživatele, uživatelské prostředí, dále poskytuje možnost rezervací a správu systému administrátorem. Hlavní částí, kterou aplikace podporuje, je možnost evolvability a základ pro přidávání nových modulů v budoucím rozvoji.

Aplikace byla otestována uživatelským testováním podle vytvořených scénářů, ze kterých vyplynuly nové nápady pro rozvoj systému.

Na závěr bylo provedeno ekonomicko-manažerské zhodnocení aplikace, kterým se vytyčily přínosy systému. Bylo také provedeno zhodnocení nákladů a návratnost investice do této aplikace a v neposlední řadě také zhodnocení dalších možností vývoje a přidávání nových částí aplikace.

Byly tedy naplněny všechny cíle práce a aplikace je připravená na další rozvoj, vzhledem k použití modulárního řešení.

Bibliografie

1. ČTK, redakce. *Největší přehradý a vodní elektrárny v České republice* [online]. 2014 [cit. 2018-04-21]. Dostupné z: <https://voda.tzb-info.cz/11949-nejvetsi-prehrady-a-vodni-elektrarny-v-ceske-republice>.
2. MARTYKÁN, Tomáš. *Informační schůzka o areálu RY:BAR s provozovatelem* [rozhovor]. Lipno nad Vltavou 999, 2018.
3. MARTÍNEK, Jakub. *Informační schůzka o areálu RY:BAR se zaměstnancem* [rozhovor]. Lipno nad Vltavou 199, 2018.
4. DORLAND, spol. s r.o. *Sportovní centrum Gutovka* [online]. 2018 [cit. 2018-04-30]. Dostupné z: <https://gutovka.cz>.
5. PARK FRYMBURK. *Park frymburk* [online]. 2018 [cit. 2018-04-30]. Dostupné z: <http://www.parkfrymburk.cz>.
6. ACTIONPARK.CZ. *Action Park* [online]. 2018 [cit. 2018-04-30]. Dostupné z: <https://www.actionpark.cz/>.
7. MĚSTO BAKOV NAD JIZEROU. *VOLNOČASOVÝ AREÁL (VČAS)* [online]. 2018 [cit. 2018-05-05]. Dostupné z: <https://www.bakovnj.cz/cs/turista/volnocasovy-areal-vcas/>.
8. OBEC RADIM. *Volnočasový areál* [online]. 2018 [cit. 2018-05-05]. Dostupné z: <https://www.obecradim.cz/informace-o-obci/volnocasovy-areal/>.
9. AGIONET S.R.O. *Orientační ceník - tvorba webových stránek* [online]. 2018 [cit. 2018-05-05]. Dostupné z: <http://www.agionet.cz/cenik/orientacni-cenik.html>.
10. MIESSLER, Daniel. *An MVC Primer* [online]. 2015 [cit. 2018-05-07]. Dostupné z: <https://danielmiessler.com/study/mvc/>.

11. RAWHSTONE, Peter. *MVC in a three-tier architecture* [online]. 2011 [cit. 2018-05-08]. Dostupné z: <http://criticaltechnology.blogspot.cz/2011/09/mvc-in-three-tier-architecture.html>.
12. BOREK, Bernard. *Prezentační vzory z rodiny MVC* [online]. 2009 [cit. 2018-05-07]. Dostupné z: <https://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>.
13. OORTS, G.; HUYSMANS, P.; BRUYN, P. D.; MANNAERT, H.; VERELST, J.; OOST, A. Building Evolvable Software Using Normalized Systems Theory: A Case Study. In: *2014 47th Hawaii International Conference on System Sciences*. 2014, s. 4760–4769. ISSN 1530-1605. Dostupné z DOI: 10.1109/HICSS.2014.585.
14. VERELST, Jan; SILVA, Alberto Rodrigues; MANNAERT, Herwig; FERREIRA, David Almeida; HUYSMANS, Philip. Identifying Combinatorial Effects in Requirements Engineering. In: PROPER, Henderik A.; AVEIRO, David; GAALLOUL, Khaled (ed.). *Advances in Enterprise Engineering VII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, s. 88–102. ISBN 978-3-642-38117-1.
15. DIXON, Chris. *Software Is Eating the World* [online]. 2016 [cit. 2018-05-07]. Dostupné z: <https://medium.com/software-is-eating-the-world/what-s-next-in-computing-e54b870b80cc>.
16. KNESL, Jiří. *Agilní vývoj: Úvod* [online]. 2009 [cit. 2018-05-07]. Dostupné z: <https://www.zdrojak.cz/clanky/agilni-vyvoj-uvod/>.
17. STACKOVERFLOW.COM. *Most Popular Languages by Occupation* [online]. 2017 [cit. 2018-05-07]. Dostupné z: <https://insights.stackoverflow.com/survey/2017>.
18. MICHÁLEK, Martin. *K čemu je dobrý Bootstrap a frontend frameworky?* [online]. 2013 [cit. 2018-05-07]. Dostupné z: <https://www.zdrojak.cz/clanky/k-cemu-je-dobry-bootstrap-frontend-frameworky/>.
19. OTTO, Mark; THORNTON, Jacob. *Bootstrap* [online]. 2018 [cit. 2018-05-08]. Dostupné z: <https://www.getbootstrap.com>.
20. GOYAL, Arun. *UPDATED – Top 10 Successful Websites Developed in PHP* [online]. 2017 [cit. 2018-05-07]. Dostupné z: <https://www.octalsoftware.com/blog/top-10-successful-websites-developed-php>.
21. THE RIGHT SOFTWARE PVT LTD. *PHP Framework Comparison 2018: CodeIgniter vs Laravel vs Yii vs CakePHP* [online]. 2017 [cit. 2018-05-08]. Dostupné z: <https://therightsw.com/codeigniter-vs-laravel-vs-yii-vs-cakephp/>.

22. VALUECODERS.COM. *Symfony vs Laravel : Which PHP framework to choose?* [online]. 2017 [cit. 2018-05-08]. Dostupné z: <https://www.valuecoders.com/blog/technology-and-apps/symfony-vs-laravel-php-framework-choose/>.
23. HANSEN, Steven. *7 Good Reasons to Use Symfony Framework for Your Project* [online]. 2017 [cit. 2018-05-08]. Dostupné z: <https://hackernoon.com/7-good-reasons-to-use-symfony-framework-for-your-project-265f96dcf759>.
24. SYMFONY. *How to Deploy a Symfony Application* [online]. 2018 [cit. 2018-05-14]. Dostupné z: https://symfony.com/doc/current/deployment.html#content_wrapper.
25. 5NEJ.CZ. *Srovnání 5 nejlepších Hostingů/Webhostingů 2018* [online]. 2018 [cit. 2018-05-09]. Dostupné z: <https://www.5nej.cz/srovnani-webhostingu/>.
26. KUČEROVÁ, Blanka. *Ekonomika podniku* [online]. 2011 [cit. 2018-05-12]. Dostupné z: <https://edux.fit.cvut.cz/oppa/BI-EKP/cviceni/Krit%5C%C3%5C%A9ria%5C%20ekonomick%5C%C3%5C%A9%5C%20efektivnosti.pdf>.
27. ČNB. *Diskontní sazba (%) (Oficiální úrokové sazby ČNB (měsíční průměr) (%)) - ekonomika ČNB* [online]. 2018 [cit. 2018-05-09]. Dostupné z: <https://www.kurzy.cz/cnb/ekonomika/oficialni-urokove-sazby-cnb-mesicni-prumer/diskontni-sazba/>.

Seznam použitých zkratek

- IS** Informační systém
- NST** Normalized Systems Theory
- PHP** Hypertext Preprocessor
- ORM** Object-relational-mapping
- VČAS** Volnočasový areál
- MVC** Model-View-Controller
- SoC** Separation of Concerns
- DVT** Data Version Transparency
- AVT** Action Version Transparency
- SoS** Separation of States
- PC** Personal computer
- IBM** International Business Machines
- GUI** Graphical User Interface
- HTML** HyperText Markup Language
- CSS** Cascading Style Sheets
- SQL** Structured Query Language
- VCS** Version Control System
- ENV** Environment

A. SEZNAM POUŽITÝCH ZKRATEK

APP Application

CF Cash Flow

NPV Net Present Value

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	
├ webapp	zdrojový kód implementace
└ thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
└ thesis.pdf	text práce ve formátu PDF