



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH A IMPLEMENTACE INFORMAČNÍHO SYSTÉMU

IMPLEMENTATION OF THE INFORMATION SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Přemysl Chovaneček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lukáš Novák, Ph.D.

BRNO 2021

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	Bc. Přemysl Chovaneček
Studijní program:	Systemové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	Ing. Lukáš Novák, 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 diplomovou práci s názvem:

Návrh a implementace informačního systému

Charakteristika problematiky úkolu:

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

Cíle, kterých má být dosaženo:

Cílem je analýza a specifikace požadavků informačního systému vybrané organizace, posouzení jeho efektivnosti na základě firemní strategie a následná implementace informačního systému na základě vytvořeného návrhu.

Základní literární prameny:

BASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: podnik v informační společnosti. 3. aktualiz. a dopl. vyd. Praha: Grada, 2012. 323 s. ISBN 978-80-247-4307-3.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2. přeprac. a aktualiz. vyd. Praha: Grada, 2009. 496 s. ISBN 978-80-247-2615-1.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. rozš. vyd. Praha: Ikar, 2000. 178 s. ISBN 80-247-0087-5.

SCHWALBE, Kathy. Information Technology Project Management. Boston, MA: Cengage Learning, 2016. 672 s. ISBN 978-1-285-45234-0.

SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání diplomové 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

Diplomová práce se věnuje návrhu a implementaci informačního systému pro malý podnik. V práci jsou zahrnuty strategické analýzy podniku a hodnocení současné situace spojené s pandemií Covid-19. Dále se práce věnuje specifikaci a analýze požadavků zadavatele, sestavení návrhu a samotné implementaci systému odpovídající uvedeným návrhům. Závěrem se práce věnuje zhodnocení a předpokládaným přínosům informačního systému.

Klíčová slova

Informační systém, Webová aplikace, SPA, Laravel, Vue.js, GraphQL, MySQL databáze

Abstract

The thesis describes the design and implementation of an information system for a small business company. The thesis includes strategic analyzes of the company and an assessment of the current situation associated with the Covid-19 pandemic. Subsequently, the thesis describes the process of specification and analysis of the requirements of the client, the design and implementation of the system corresponding to the proposals. Finally, the thesis provides the evaluation and expected benefits of the information system.

Keywords

Information system, Web application, SPA, Laravel, Vue.js, GraphQL, MySQL database

CHOVANEČEK, Přemysl. *Návrh a implementace informačního systému* [online]. Brno, 2021 [cit. 2021-05-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/133694>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav Informatiky. Vedoucí práce: Ing. Lukáš Novák, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená diplomová 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

.....

podpis autora

Poděkování

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Lukáši Novákovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych rád poděkoval svému kolegovi a příteli Janu Hanusovi, který se podílel na tvorbě grafických návrhů systému.

OBSAH

ÚVOD	9
VYMEZENÍ PROBLÉMU A CÍLE PRÁCE	10
1 TEORETICKÁ VÝCHODISKA PRÁCE	13
1.1 Data a informace	13
1.2 Informační společnost	14
1.3 Systém	15
1.4 Informační systém	16
1.5 Životní cyklus systému v informatice	17
1.6 Varianty životního cyklu	20
1.7 Základní klasifikace informačních systémů	22
1.8 Náklady na informační systém	23
1.9 Webové aplikace	23
1.10 Rozhraní API	24
1.11 Single-page application	27
1.12 Kompozice aplikace	28
1.13 Relační databáze	29
1.14 Návrhové diagramy	30
1.15 Hypertext Markup Language	31
1.16 Přístup Mobile First Design	32
1.17 Kaskádové styly	32
1.18 JavaScript, dostupné knihovny	34
1.19 PHP a knihovna Laravel	37
1.20 Databáze MySQL	38
2 ANALÝZA SOUČASNÉ SITUACE	39
2.1 Charakteristika společnosti	39
2.2 Analýza současné situace	40
2.3 McKinseyho model 7S	41

2.4	Analýza PESTLE	43
2.5	Porterova analýza pěti sil	46
2.6	Analýza SWOT	48
2.7	Shrnutí analýz	50
3	VLASTNÍ NÁVRHY ŘEŠENÍ	52
3.1	Výběr systému	52
3.2	Lewinův model	52
3.3	Identifikace a specifikace požadavků	54
3.4	Diagram případů užití	57
3.5	Datový model	58
3.6	Návrh komponent systému	59
3.7	Návrh uživatelského rozhraní	61
3.8	Implementace uživatelského rozhraní	64
3.9	Datová implementace	66
3.10	Bezpečnost	68
3.11	Automatické posílání informací emailem	70
3.12	Návrh testování systému	71
3.13	Produkce a hostování webové aplikace	71
3.14	Použité nástroje	72
3.15	Časové plány a milníky	72
3.16	Zhodnocení nákladů	73
3.17	Zaškolení	74
3.18	Potenciální přínosy informačního systému	75
	ZÁVĚR	76
	SEZNAM POUŽITÝCH ZDROJŮ	77
	SEZNAM OBRÁZKŮ	84

ÚVOD

Diplomová práce se věnuje oblasti informačních systémů. V dnešní době jsou informační systémy nedílnou součástí našeho světa – podporují podnikání v mnoha oblastech jako například finance, logistika, prodej, nákup a podobně. Podnik, který chce zůstat konkurenceschopný, musí držet krok s technologickými trendy. Současná situace spojená s pandemií Covid-19 navíc poukázala na možné existenční problémy podniku, s čímž jsou spojeny rostoucí nároky na alternativní – zejména distanční formy podnikání.

Oblast pohostinství patří mezi oblasti nejvíce pandemií postižené. Podnikům tohoto charakteru hrozí bankrot – jsou odkázány pouze na výdej, případný rozvoz, který často souvisí s vysokými provizemi inkasovanými rozvozovými firmami.

První část práce se zabývá především teoretickým východiskům spjatým s problematikou informačních systémů – jako je jeho struktura či životní cyklus. Nastíněny jsou také současně moderní principy vývoje informačních systémů v podobě webových aplikací.

Část druhá je zaměřena na charakteristiku firmy, zkoumání vnějšího a vnitřního prostředí na základě analýz. Dále je zde věnována pozornost současné situaci ve firmě.

Poslední část tvoří vlastní návrhy řešení informačního systému. Zpracování vychází z analýzy specifikace požadavků klienta. Obsažen je návrh uživatelského rozhraní, návrh rolí v systému se vyskytujících, návrh modelu dat a samotná implementace pomocí nejmodernějších technologií.

VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

Úvodem se kapitola věnuje vymezení problému práce. Druhá část zahrnuje popis cílů diplomové práce, kterých je třeba dosáhnout pro vyřešení uvedené problematiky.

Vymezení problému

Na jaře roku 2020 propukla v Evropě pandemie nemoci Covid-19. Česká republika nebyla výjimkou – koncem téhož roku dokonce patřila k zemím dominujícím evropským statistikám skrze rostoucí reprodukční číslo R. [1]

Úřad vlády se s podobnou situací dříve nesešel. V reakci na politiku ostatních, silně zasazených států, česká vláda vyhlásila řadu mimořádných opatření, došlo také k zavedení a následovnému prodlužování nouzového stavu. [2]

Zavedení vládních nařízení měl negativní dopad zejména na malé živnostníky a podnikatele. Mnozí z nich byli ohroženi, ba dokonce zničeni bankrotem, jiní se museli adaptovat na měnící se podmínky. Konkrétně v oblasti gastronomie byli podnikatelé nuceni nejprve přejít na provoz v rámci venkovních prostorů, následně se museli potýkat s úplným uzavřením podniků. Později je situace přinutila přejít na nový režim provozu, tj. režim formou výdejních okének. [3] Zmíněná nařízení přiměla mnoho restaurací rozšířit podnik o možnost rozvozu jídel, či řízení elektronických objednávek. Bohužel zde vzniká tlak na využití zpoplatněných, konkurenčních služeb, což finanční situaci opět neprospívá.

Zkoumaný podnik se rozhodl nepříznivou situaci řešit vlastním informačním systémem, jež by měl disponovat nejen řešením elektronických objednávek, ale také rezervačním systémem, systémem správy zaměstnanců a jejich směn, případně řízení článků a událostí.

Cíle diplomové práce

Na základě výše popsaného problému byly stanoveny cíle práce, mezi které patří analýza a specifikace požadavků informačního systému vybrané organizace, posouzení jeho efektivnosti na základě firemní strategie a následná implementace informačního systému na základě vytvořeného návrhu.

Elementární terminologie

Podkapitola obsahuje výčet termínů a definic, které se v práci častokrát opakují. Typicky se jedná o slova převzatá z anglického jazyka se špatně vyjádřitelným překladem v jazyce českém. Většina z nich je v průběhu práce detailně vysvětlena, nicméně pro lepší orientaci v textu je vhodné pojmy vysvětlit na začátku práce.

Framework

Pojmem framework rozumíme nastavbu architektury daného programovacího jazyka. Framework může obsahovat podpůrné knihovny, utility, koncepty apod., které usnadňují a urychlují vývoj softwaru.

Frontend

Frontend neboli část uživatelského rozhraní aplikace je část softwaru, se kterou pracuje a komunikuje uživatel. Vizuální pocity a interakce se softwarem se uživateli jeví mnohdy důležitější než samotný proces manipulace s daty.

Backend

V rámci práce a implementace související s daty hovoříme o Backendu. Backend je část, kterou naopak uživatel nevidí. Ačkoli se jedná o skrytou část, probíhá v ní řídicí logika aplikace nebo softwaru.

API

Definice pojmu je detailně vysvětlen v teoretické části práce, nicméně si jej lze představit jako číšníka, který zprostředkovává objednávky mezi hostem a kuchyní.

Request

Pojmem request rozumíme dotaz, který je spojen se zasíláním požadavku na čtení dat, či zápis dat na server a databáze. Dotazy úzce souvisejí s předchozím pojmem API.

Instance, instanciace

Instancí rozumíme objekt konkrétní třídy v programovacích jazycích s objektivně orientovaným přístupem. Jestliže máme třídu zvíře, objekt odvozený z této třídy může být například chameleon. Pojmem instanciace rozumíme vytvoření nové instance.

Invokace metody

Význam spojení je v programovacích jazycích odvozený z volání funkce. Metody jsou v objektivně orientovaných jazycích konstrukce podobné funkcím.

1 TEORETICKÁ VÝCHODISKA PRÁCE

V této kapitole jsou rozebrána teoretická východiska práce. Podkapitoly se chronologicky věnují fundamentálním tématům z oblasti informačních systémů, jakožto jeho struktury, životního cyklu, až po konkrétní technologie používané v dnešní době.

1.1 Data a informace

Pro hlubší zkoumání oblasti informačních systémů nesmíme opomenout esenciální pojmy, které s touto oblastí souvisí.

Data

S pojmem data se v dnešní době setkal pomalu každý. Mluvíme o faktech, která interpretují jevy a objekty, jejich chování a vlastnosti. Nejčastější forma uchování dat je numerická. [4]

Data lze získat například měřením či pozorováním, lze je dále zpracovávat a transformovat. Příkladem může být konverze analogových dat – dat získaných na základě pozorování různých fyzikálních jevů, na data digitální – data popsána pomocí numerických hodnot. [5]

Podniková data

Ve spojitosti s informačními systémy rozumíme pod pojmem podniková data právě data taková, která jsou nositeli skutečností souvisejících s aktivitami podniku. Zároveň je možné tyto data interpretovat, přenášet a dále zpracovávat. [6]

Rozdělení podnikových dat do následujících tří skupin prezentuje kniha Podniková informatika. [7]

Data o společenských podmínkách podnikání – obsahují fakta z oblasti mikro a makrookolí organizace. Patří zde například údaje o ekonomických trendech, pracovní síle, dostupnosti materiálu, kapitálu apod.

Data o trhu – zahrnuje skutečnosti o aktuálním vývoji situace na trhu, nabídce, poptávce, či konkurenci.

Interní data – data, která vypovídají o konkrétním podniku. Patří zde například data o predikci vývoje, alokaci a omezení zdrojů, či obchodní a finanční plány. Analýza těchto dat pomáhá managementu reagovat efektivně vůči okolnímu dění.

Informace

Pojem informace lze definovat prostřednictvím dat, kterým jejich uživatel přisuzuje určitý význam a která uspokojují konkrétní objektivní informační potřebu svého příjemce. Nejčastěji bývá nositelem informací text, zvuk, obraz či numerická data. Na rozdíl od dat, informace nelze skladovat. [8]

Vnímání definice pojmu informace je velice subjektivní záležitost, proto lze informaci dekomponovat na tři elementární pohledy. Prvním z nich je pohled **syntaktický**, který se zabývá vnitřní strukturou informace, vazbami mezi jednotlivými znaky, které informaci tvoří. Jiným pohledem je pohled **sémantický**, jenž je orientován na stránku obsahovou, bez ohledu na vztah k jejímu příjemci. Poslední možností je pohlížení na informaci z pohledu **pragmatického** – pohledu, který se zabývá významem informace pro příjemce. Pragmatický pohled je nejbližší lidem z oblasti managementu, kteří jej chápou jako nezbytnou část rozhodovacího procesu. [6]

Mluvíme-li o procesu zpracování informací v souvislosti s informačními systémy, máme na mysli manipulaci (přenos, transformaci) s konkrétními daty přidruženého významu, které jsou součástí informačního systému.

1.2 Informační společnost

Informační společnost lze chápat jako komunitu aplikující informační a komunikační technologie do každodenního života.

V informační společnosti prudce narůstá, ba dokonce dominuje podíl HDP (hrubého domácího produktu) vázaného na zmíněné technologie. Společnost se postupem mění a přechází od použití zemědělských a průmyslových zdrojů, ke zdrojům informačním. Vlivem

takového růstu se mění nejen paradigma podnikání, ale také paradigma vzdělávání, či výzkumu. Změny, které vývoj informační společnosti přináší se dotýkají mnoha aspektů podniků – zákazníků, dodavatelů, konkurence i zaměstnanců. [9]

1.3 Systém

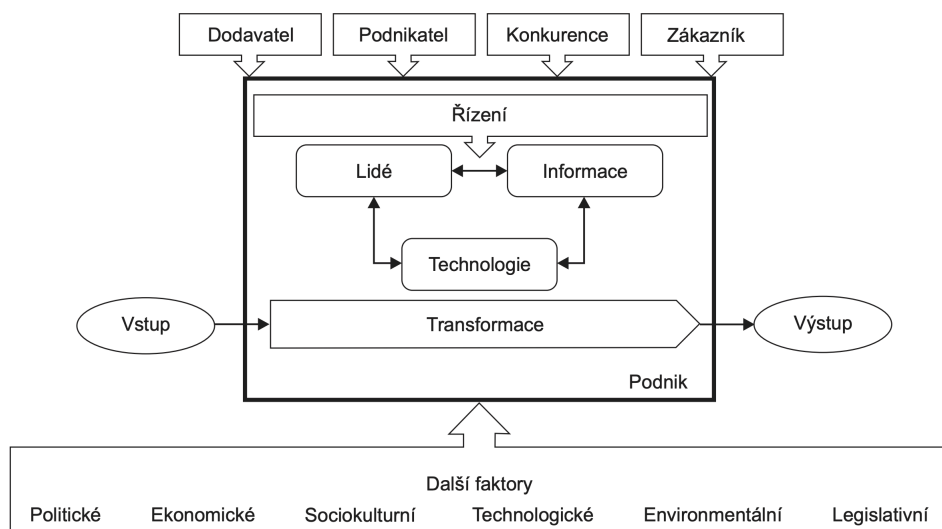
Systém lze definovat jako uspořádanou množinu elementárních prvků mezi nimiž účinkují vzájemné vazby. Mnohdy je systém hierarchicky složen z podsystémů a dále.

Prvek systému

„Prvky systému považujeme obecně za elementární, dále nedělitelné části systému představující jeho rozkladové (dekompoziční) složky. Prostřednictvím prvků popisujeme strukturální vlastnosti systému; chování prvků ve vzájemných interakcích, které vyplývají ze struktury systému, určuje chování celého systému.“ [10]

Podnik jako systém

Příkladem systému může být podnik. Následující interpretace je inspirována literaturou [7].



Obr. 1.1: Schéma systému podniku. Obrázek převzat z literatury [7].

Množina prvků systému (obrázek 1.1) může obsahovat podnikatele, dodavatele, konkurenci, zákazníka, ale také faktory analýzy PESTLE. Vnitřní část systému tvoří prvky jako jsou informace, lidé, technologie. Lze říci, že tyto prvky reprezentují podsystém systému. Kupříkladu prvek informace je možné v podniku substituovat **informačním systémem**.

Podnik může vykazovat vlastnosti otevřeného systému a to skrze vstupy a výstupy závislé na prvcích jako je konkurence, či zákazník, nebo dodavatel.

Systémy, které se zabývají ukládáním, přenosem nebo také transformací informací můžeme klasifikovat jako systémy informační, viz podkapitola 1.4.

1.4 Informační systém

Potřeba sbírat, vytvářet a distribuovat užitečná data napříč organizací vedla člověka k myšlence, jež zapříčinila vznik informačních systémů. Informační systém lze popsat kombinací hardware, software a telekomunikačních sítí, která vyhovuje těmto potřebám. [11]

Struktura informačního systému

Mezi konkrétní prvky struktury informačního systému patří následující komponenty. Stanovení struktury vychází z literatury [12].

Software neboli programové vybavení, poskytuje uživateli schopnost pracovat se systémem skrze fakta reálného světa. Informační systém má svou „frontendovou část“, se kterou komunikuje uživatel, dále část „backendovou“, která komunikuje se serverem a databází.

Hardware informačního systému můžeme interpretovat jako fyzické výpočetní vybavení, které umožňuje chod systému. Současně poskytuje také fyzické úložiště dat.

Databáze je nedílná komponenta systému. Uchovává veškerá data, která do systému vstupují a ze systému vystupují.

Lidská složka prezentuje úkony a uplatnění člověka v rámci informačního systému.

Organizační struktura se věnuje zejména legislativním záležitostem spojených s provozem informačního systému.

Kontext informačního systému bývá zasazen do prostředí reálného světa, cílem je zvýšení efektivity organizace.

1.5 Životní cyklus systému v informatice

Softwarové inženýrství je oblast, která s implementací informačních systémů velice úzce souvisí. Musíme si uvědomit, že software prochází tzv. vlastním životním cyklem.

Nehledě na to, zda se jedná o aplikaci webovou, mobilní či jiný softwarový produkt, nevyhnutelnou součástí procesu vývoje je dekompozice projektu do několika etap či fází. Hovoříme-li o životním cyklu systému, máme tím na mysli sled fází spojených s jeho vývojem.

Ačkoliv se principiálně fáze různých metodik podobají, jejich sled není striktně definován. Následující metodika a popis jsou čerpány z odborné literatury [13]. Metodika rozděluje průběh životního cyklu na 4 stupně.

Stupeň I. se váže pouze k podnikatelskému prostředí. Zahrnuje následující dvě analýzy.

- **Strategická analýza**
- **Úvodní analýza**

Přechod z podnikatelského prostředí na prostředí zohledňující požadavky na vývoj odpovídá **Stupni II.** Zahrnuje také dvě etapy

- **Podrobná analýza**
- **Úvodní návrh (design)**

Etapy **Stupně III.** zahrnují samotnou tvorbu softwaru, ladění a akceptační testování ze strany uživatelů. Patří zde etapy

- **Podrobného návrhu**
- **Implementace**

Poslední, **Stupeň IV.** je spojen s **fází údržby a provozu.**

Strategická analýza

Při fázi strategické analýzy dochází k vymezení problému, také k identifikaci esenciálních procesů, jež bychom měli v systému zohlednit.

Úvodní analýza

Zde dochází k rozpracování vymezeného problému a k identifikaci objektů, jejich atributů a metod, které se vztahují ke stanoveným procesům. Fázi lze chápat jako přechod od nynějšího stavu ke stavu očekávanému. Výstupy této fáze mohou ovlivnit prostředí organizace, což může vést k nezbytným změnám tohoto prostředí a definovaných procesů.

Podrobná analýza

Analýza rozebírá detailní model – objekty, jejich vazby a závislosti. Zahrnuje popis funkčního rozhraní vůči uživatelům.

Úvodní návrh

Prvním a stěžejním krokem softwarové implementace je návrh. Předcházející fáze analýz lze považovat za dokončené. Návrh rozebírá také koncepty uživatelského rozhraní.

Podrobný návrh

Jedná se o fázi, ve které dochází k transformaci elementů vytvořeného modelu do podoby vhodné k cílovému implementačnímu prostředí. Dochází k výběru vhodných programovacích jazyků a typu databáze. Stanovují se znovupoužitelné komponenty, pro usnadnění a optimalizaci.

Implementace

Hlavní cíle fáze implementační je samotný vývoj, sestavení, či programování konkrétního softwaru. Dochází například k propojení jednotlivých funkčních komponent, nasazení do produkčního prostředí.

Fáze údržby a provozu

Je poslední fáze životního cyklu. Fáze probíhá od začátku nasazení systému, přes užívání, k dokončení provozu. Podíl práce, finanční a časové náročnosti na provoz může být až 4x větší než sečtená pracnost a náročnost všech předchozích fází.

Požadavky na informační systém

Při řešení požadavků se zadavatelem je vhodné sestavit kostru požadavků. Ta zlepší komunikaci a vzájemné porozumění obou stran, tedy strany zadavatele a strany zpracovatele.

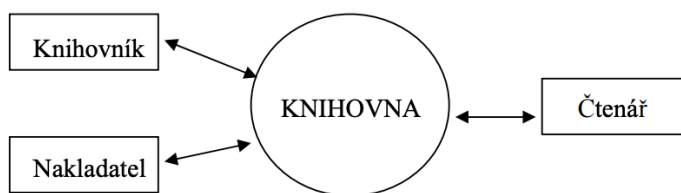
Požadavky uživatelů se z pravidla dělí do dvou kategorií. První z nich jsou funkční požadavky popisující funkčnost systému. Druhou kategorií požadavků nazýváme požadavky nefunkčními. Nefunkční požadavky řeší otázky, spolehlivosti, dostupnosti, ceny, doby řešení apod. Struktura požadavků a její popis vychází ze skript univerzity VŠB-TU Ostravy [14].

Funkční požadavky

Pro sestavení základní struktury funkčních požadavků bychom měli zodpovědět následující otázky.

- **K čemu a proč je informační systém zapotřebí** – je třeba si uvědomit, co informační systém přinese, popsat současný stav evidence, zdůvodnit, proč systém nevyhovuje, charakterizovat představy o funkčnosti.
- **Kdo s ním bude pracovat** – primární funkce by měly připadat uživatelům s nejčastějším přístupem, sekundární funkce uživatelům s přístupem občasným. Příklad lze vidět na obrázku 1.2.
- **Jaké budou vstupy** – entity a jejich atributy, které jsou základem pro datovou analýzu.
- **Jaké budou výstupy** – výstupy očekávané uživatelem.
- **Jaké funkce má plnit** – informace jsou nezbytným podkladem pro další analýzy. Máme na mysli operace, které budou s daty prováděny. Často dochází k rozšíření definic vstupů.

- **Jaké je okolí informačního systému** – aktéři, kteří jsou zdrojem, nebo cílem informací vstupujících, či vycházejících ze systému. Pojem aktér není totožný s pojmem uživatel.



Obr. 1.2: Schéma zachycuje aktéry a jejich vazby na informační systém. Například aktér nakladatel zasílá do systému nabídku, či informace o dodaných knihách, naopak odebírá informace o knihách objednaných. Obrázek převzat z literatury [14].

Nefunkční požadavky

Nefunkční požadavky jsou dodatečné informace o systému vedoucí například k požadovanému způsobu řešení.

- **Priority programu** – stanovení priorit v rámci efektivity, spolehlivosti, paměťové náročnosti, přenositelnosti.
- **Způsob řešení** – předepsané standardy, metody, jazyk implementace, uživatelské rozhraní apod.
- **Vnější požadavky** – například legislativní omezení, omezení cenové, apod.

1.6 Varianty životního cyklu

Průběhu životního cyklu odpovídá několik osvědčených modelů. Následuje charakteristika nejpoužívanějších z nich. Aplikace různých alternativ životního cyklu je závislá na kontextu vyvíjeného softwaru.

Projekty v oblasti IT jsou na rozdíl od oblastí jiných velice různorodé. Existují společnosti, které si vystačí s malým počtem zaměstnanců a běžným hardware. Naopak existují organizace analyzující ohromné množství dat obchodních procesů, implementují software pro širokou škálu uživatelů, mají robustní síťovou infrastrukturu a podobně. [15]

Vodopádový model

Jedná se o sekvenční model (princip vodopádu), který obsahuje 7 fází. Model nese podmínku dokončení konkrétní fáze pro začátek fáze následující. Přístup sebou nese určitá úskalí. Základní myšlenkou bývá předpoklad, že navržené řešení fáze je kompletní, fáze musí být zpracována perfektně. Řešení výskytu chyb ve fázi požadavků, odhalených až ve fázi implementační může být finančně náročné, někdy až destruktivní. Na druhou stranu, dojde-li k odhalení chyb v příslušné fázi, může to vést k ušetření času i financí. Nicméně u robustních, složitějších projektů je v současnosti již nepravděpodobné před začátkem fáze nové, dovést fázi předcházející do perfektního stavu. [16]

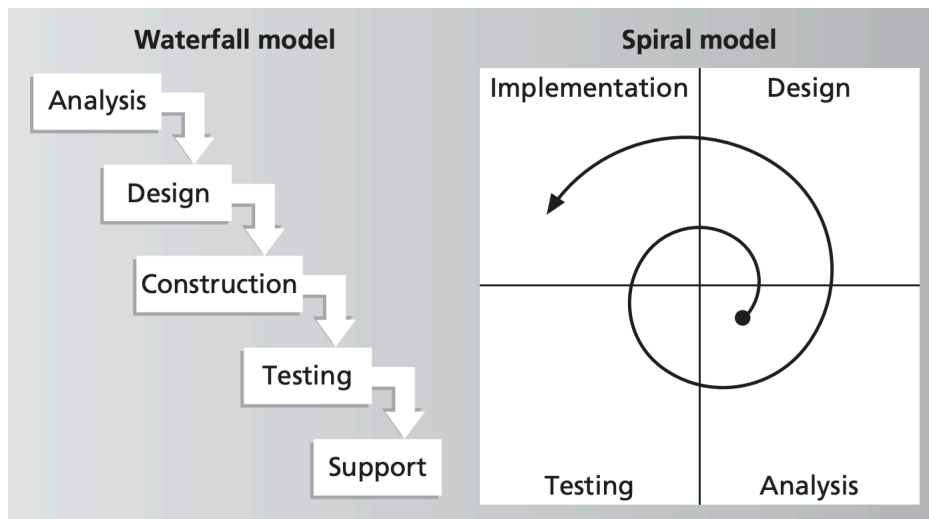
Iterativní model

Jak již bylo zmíněno, sekvenční přístup je dnes téměř nepoužitelný, proto vznikl model iterativní. Iterativní model rozšiřuje model vodopádový. Pro model již není nezbytné znát kompletní zadání. Vyvíjí se neúplné, ale funkční prototypy, v tom spočívá jeho výhoda. Problémy mezi týmem a požadavky klientů lze odhalovat daleko dřív a to díky revizi vyvinutých prototypů. Zadavatelé pak mají přehled o stavu projektu během každé iterace celého životního cyklu. [17]

Spirálový model

Model opět staví na funkčních prototypech. Každý prototyp můžeme chápat jako výslednou verzi, každá nová verze může být potenciálně verze finální. Zároveň je verze možným vstupem do dalšího cyklu. Model pracuje v 6 fázích, kde typicky počet opakování je 2-3. [13] Síla modelu spočívá ve vývoji nezávislých komponent, ze kterých dochází k sestavení výsledného softwaru. Přístup však klade větší nároky na projektové manažery a to kvůli neustálým možnostem produkt vylepšovat a optimalizovat. Srovnání s vodopádovým modelem lze vidět na obrázku 1.3.

Spirálový model životního cyklu opět rozšiřuje a zdokonaluje model vodopádový. Původně byl aplikován na velké softwarové projekty, například projekty vládní. V současné době je většina softwaru vyvíjena pomocí iterativního nebo spirálového zpracování částí, než pomocí přístupu lineárního. [15]



Obr. 1.3: Srovnání životního cyklu vodopádového a spirálového modelu. [15]

1.7 Základní klasifikace informačních systémů

Klasifikace informačních systémů vyplývá z požadavků podniku na druhu či způsobu zpracování informací. Nejčastěji se setkáváme s klasifikací čtyř úrovní. Následující charakteristika úrovní vychází z odborné literatury. [6]

Provozní úroveň Systémy zpracovávající rutinní požadavky. Může se jednat o zpracování objednávek, přijímání plateb. Takové systémy naplňují každodenní chod podniku. Často se těmto systémům říká provozní, či transakční.

Znalostní úroveň Systémy jako jsou ERP, CRM, ale také kancelářské aplikace pro podporu řízení či toku dokumentů.

Řídicí úroveň Systémy prosperující oblasti středního a vrcholového managementu. Zahrnují podporu rozhodování. Systémy disponují například funkcí generování reportů, které napomáhají sestavovat analýzy, či řešit složitější rozhodovací úlohy.

Strategická úroveň Informační systémy pokrývající strategickou oblast. Využívány bývají hojně ve vrcholovém managementu. Slouží například k identifikaci dlouhodobých trendů, nebo k predikci změn. Často jsou schopny na tyto změny reagovat.

1.8 Náklady na informační systém

Při rozhodování podniku o výběru vhodného informačním systému jsou klíčové zejména aspekty spojené s kvalitou a s náklady na nasazení a provoz systému. Následující metodika čerpá z metodiky Ministerstva vnitra ČR. [18]

Z hlediska nákladu lze využít model TCO (Total Cost of Ownership) v oblasti ICT. Model člení základní pohledy na ICT řešení a náklady do čtyř kategorií dle průběhu životního cyklu. Většina z těchto kategorií je pak dále detailněji členěna. Kategorie jsou následující

- *náklady na pořízení, implementaci*
- *náklady na provoz a údržbu*
- *náklady na dodatečný rozvoj či změny*
- *náklady na zrušení aplikace*

Model popisuje k čemu náklady sloužily, nikoli jakého byly druhu. Současně také rozděluje náklady na jednorázové (projektově orientované) a na náklady průběžné, nejčastěji náklady spojené s údržbou či provozem.

1.9 Webové aplikace

Mluvíme-li o webových informačních systémech, máme na mysli jejich implementaci formou jakési webové aplikace. Webové aplikace v dnešní době prorážejí kupředu a snižují podíl aplikací nativních a to hned z několika důvodů. Zejména se samy aktualizují, jsou flexibilní vůči platformám a zařízením.

Webová aplikace není nic jiného než program, který je však proti standardním aplikacím dostupný prostřednictvím internetu. Aplikace běží na vzdáleném serveru. Uživatel k aplikaci přistupuje skrze její uživatelské rozhraní neboli „frontend“, které komunikuje se serverem pomocí zaslání dotazů takzvaného API rozhraní, viz podkapitola 1.10.

Vývoj webových aplikací má podobnou historii jako vývoj v oblasti softwarového inženýrství. Dřívější webové aplikace se skládaly z mnoha skriptů, avšak scházela jim určitá systematická struktura. Rozšířil se koncept individuálních komponent, ze kterých může být aplikace sestavena mnohem snadněji. [19]

Protokol HTTP

Jedná se o klient-server komunikační protokol aplikační vrstvy modelu ISO/OSI. Protokol HTTP je schopen iniciovat některé požadavky ze strany klienta prostřednictvím webového prohlížeče. Obvykle se váže k portu 80.

Mezi nejdůležitější metody protokolu HTTP patří metody popsané v článku [20]. K metodám patří

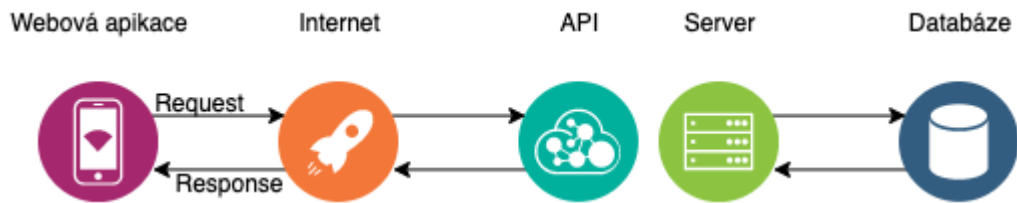
- GET – metoda vyšle request (požadavek) na server pro získání dat. V praxi se může jednat o výpis článku daných rubrik, seznam uživatelů a podobně.
- POST – metoda vyšle request s přenosem dat na server. Typická činnost je odesílání vyplněného formuláře při registraci uživatele.
- PUT – idempotentní metoda velice podobnou metodě POST. Rozdíl je v tom, že metoda PUT vytváří objekt se jménem ekvivalentním objektu přenášeného. Jde v podstatě o reverzní postup metody GET.
- DELETE – metoda žádá o smazání dat či konkrétního objektu. Často je zapotřebí mít dostatečná oprávnění.

1.10 Rozhraní API

API není nic víc, než soubor kódů programovacího jazyka, který umožňuje transfer dat od jednoho softwarového produktu k jinému. Zároveň zahrnuje podmínky pro výměnu dat. [21]

Například se může jednat o komunikační bod mezi uživatelským rozhraním softwaru produktu a řídicí logikou tohoto produktu. Pro lepší pochopení lze koncept API vysvětlit na používání aplikace počasí.

Telefonní aplikace počasí funguje na principu získání dat o vyhledávané lokalitě. API služba počasí obsahuje výčet implementovaných funkcí – získání dat teploty, transformaci do jiné stupnice, data spojená s polohou, zeměpisnou šířkou a délkou. API také zahrnuje definice těchto funkcí, informace o tom, jakým způsobem má služba požadavky získat



Obr. 1.4: *Komunikace informačního systému skrze API. (Zdroj: vlastní zpracování, 2021)*

a vrátit. Pokud chceme, stačí napsat město, které nás zajímá. Po potvrzení vyšle aplikace na vzdálenou API službu dotaz s předaným parametrem, tedy jménem města. V aplikacích tedy vysíláme konkrétní žádosti za pomoci metod aplikačního rozhraní, na základě kterých jsme schopni získávat data. Konkrétní implementace požadovaných funkcí je uživateli skryta, je zakódována v API.

Při vývoji **informačního systému** lze koncept API aplikovat v rámci komunikace mezi uživatelským rozhraním a funkcemi systému. Příkladem je proces autentizace uživatele systému, kde se uživatel identifikuje zadanými parametry. Na obrázku **1.4** lze vidět schéma transferu dat. Standardně dochází k poslání tzv. requestu na server společně se zadanými parametry. Část našeho serveru přijímá žádosti a zároveň na ně zasílá odpověď, a to ve formě dat, která jsou uložena v databázi. Současně se pošle také status kód žádosti, který má konvenčně stanovené hodnoty dle typu odpovědi např. (200 OK).

CRUD

Vytváření API pro webové aplikace popisuje akronym **CRUD**. Jedná se o koncept, který popisuje čtyři základní metody `create`, `read`, `update`, `delete`. [22] Chceme-li aby byly operace s našimi modely dat korektní a kompletní, musí rozhraní podporovat všechny uvedené metody.

Příkladem může být manipulace s uživatelem informačního systému.

- Při inicializaci jeho profilu (registraci) je třeba provést počáteční transfer do databáze – aplikace metody `create`.
- V případě zobrazení svého uživatelského profilu v systému je aplikována metoda `read`.

- V případě aktualizace údajů profilu, kupříkladu příjmení, dochází k použití přístupu `update`.
- Posledním případem je smazání uživatelského profilu pomocí metody operace `delete`.

REST

REST, akronym pro REpresentational State Transfer, je koncept architektury rozhraní, původně navržený pro distribuované hypermediální systémy. Architekturu navrhnul Roy Fielding v roce 2000 v rámci jeho disertační práce. Fielding je mimo jiné jeden z tvůrců protokolu HTTP. [23]

Architektura REST má 6 základních principů, jež by měly být dodrženy. Jestliže je tak učiněno, rozhraní lze nazývat RESTful. Pojem RESTful se váže k webovým službám, které zmíněnou REST architekturu používají. [24]

Nejčastěji je architektura spojena s protokolem HTTP, který má v sobě implementovány metody CRUD viz 1.9. To je výhodné pro vývojáře, protože se nemusí zabývat dodatečným řešením této problematiky.

Konstrukce a invokace zaslání dotazu se provádí přes *URL* adresu obsahující požadovanou metodu, takzvaný endpoint – tedy cílový bod, tělo požadavku a autentizační údaje.

GraphQL

GraphQL je obvykle popisováno jako „frontend“ orientována API technologie, která umožňuje vývojářům snazší přístup k požadavku na získání dat. Na rozdíl od REST se nejedná o architekturu, nýbrž o dotazovací jazyk vyvinut společností Facebook. Jazyk není vázán na specifickou databázi, nebo úložiště dat, ale na data a kód vývojáře.[25] Výhodou GraphQL oproti REST je způsob získávání dat. Například REST potřebuje pro získání dat uživatele a jeho příspěvků použít dva dotazy a vrátit data ze dvou endpointů. GraphQL dokáže získat tyto data jedním dotazem, v oblasti vývoje moderních aplikací se jedná o výrazný krok kupředu.

Mutace jsou fundamentálním prvkem GraphQL. Mutace slouží ke vkládání, aktualizaci či mazání dat. Oproti tomu dotaz slouží pouze ke čtení dat.

```
type Mutation
{
  setEmail(email: String): String
}
```

```
type Query
{
  getEmail: String
}
```

Kód ukazuje rozdíl mezi mutací, konkrétně zapsání emailu (např. při změně adresy uživatele v systému), v porovnání s dotazem, který vrátí adresu.

Lighthouse & Apollo

Lighthouse je framework, který umožňuje efektivně sestavit a implementovat GraphQL server komunikující s „backendovou“ částí aplikace.

Opačným směrem operuje framework Apollo Vue, který poskytuje integraci deklarativního dotázování vůči GraphQL serveru, a to přímo uvnitř Vue komponent (podkapitola **1.18**).

1.11 Single-page application

Není tomu tak dávno, kdy webový prohlížeč stahoval jednotlivě každou stránku domény z jejího ftp serveru. Tento princip je stále velice rozšířený a dominuje. S vynalézavostí vývojářů a zlepšováním paradigmat programovacího jazyka javascript však vzniklo spojení single-page application (SPA), které získává stále více na popularitě. Trh podnikání a firem je plný konkurence a kompetence. Řešení v podobě SPA může být jeden z klíčových faktorů při rozhodování zákazníků.

Single page aplikace (česky jednostránková aplikace) funguje prostřednictvím webového prohlížeče bez nutnosti dodatečného načítání a výskytu odezvy způsobenou čekáním na stažení relevantních požadovaných stránek. Při prvním načítání SPA se totiž stáhne celý balíček aplikace. Aplikace může pracovat plynule, nebo v offline módu. Příkladem SPA může být facebook, jenž je napsán v javascriptové knihovně React, která mu náleží. [26]

V případě méně navštěvovaných komponentů aplikace (např. modulu pro správu podniku, ke které zákazníci nemají přístup) lze využít dodatečné stažení těchto komponent při jejich užití. Tyto méně často užívané komponenty tak nemusí být součástí prvního stažení a inicializace aplikace. To přináší dvě výhody. V první řadě nedochází ke zbytečnému zpomalování při prvotním načtení, druhou výhodou je fakt, že zákazník nemusí mít v cache paměti svého zařízení zbytečná data, které jsou pro něj irelevantní. S ohledem na robustnější aplikace jsou tyto optimalizační kroky žádoucí.

1.12 Kompozice aplikace

Pro naplnění představ zadavatele je klíčová etapa návrhu. Etapa specifikuje operace, aktéry a datový model systému. Nelze opomenout také další úhly pohledu. První dojem může rozhodnout o tom, zda se zákazník rozhodne pro setrvání na webu podniku či nikoli.



Obr. 1.5: Obrázek zachycuje možnou variantu kompozice a implementace webových aplikací dle programovacích jazyků. (Zdroj: vlastní zpracování, 2021)

Základem návrhu a implementace webové aplikace je dekompozice tohoto procesu do několika menších částí. Návrh a implementace konkrétních částí softwaru obvykle vyžaduje individuální a adekvátní prostředky.

Implementační část webových aplikací zpravidla zahrnuje použití značkovacího jazyka HTML, stylovacího jazyka CSS. V dnešní době je nevyhnutelnou součástí práce s daty a s vizuální stránkou aplikace jazyk javascript, potenciálně zmíněné GraphQL pro komunikaci se serverem. Nedílnou součástí je „backend“ softwaru – serverová část, databáze a logika. V takovém případě se často skloňují například jazyky PHP, Java, C# a podobně, které odkazují data na databázi skrze dotazování. Typicky se jedná o relační databáze, či databáze multidimenzionální. V případě dotazování bývá obvykle použit deklarativní jazyk SQL, či jeho alternativy. Shrnutí implementačních jazyků a technologií lze vidět na obrázku 1.5.

1.13 Relační databáze

Před vysvětlením pojmů spojených s relačními databázemi je vhodné definovat pojem relace, který s uvedeným konceptem úzce souvisí.

Relace

„Relace na doménách D_1, D_2, \dots, D_n je dvojice $R = (R, R^)$, kde $R = R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$ je schéma relace a $R^* D_1 x D_2 x \dots x D_n$ je tělo relace. Schéma relace zapisujeme často zjednodušeně ve tvaru $R(A_1, A_2, \dots, A_n)$. Počet atributů n relace se označuje stupeň (řád) relace, kardinalita těla relace $m = |R^*|$ se označuje kardinalita relace.“*

[27]

Relační databáze

Relační databáze fungují na principu relačního modelu dat. Relaci si můžeme představit jako tabulku, kde každý řádek je právě jeden její záznam. Záznam je identifikovatelný pomocí hodnoty zvoleného atributu, ta je v databázi unikátní, atribut je tzv. primární klíč.

Může se jednat například o jednoho studenta univerzity. Ten má svá data uložená v databázové tabulce student. Tabulka obsahuje atributy – například email, adresa, ročník, obor a také identifikační číslo, které je v systému unikátní. V takovém případě bude identifikační číslo primárním klíčem tabulky.

Relační model je znám jako silný model, zejména kvůli způsobu reprezentace a možnosti dotazování dat různých typů. Zaručení dostupnosti a správnosti dat je dáno pravidly o zachování integrity. Co se týče způsobu dotazování, vznikl deklarativní jazyk SQL, který je v různých alternativách používán dodnes. Relační model je v současnosti stále nejrozšířenější databázový model. [28]

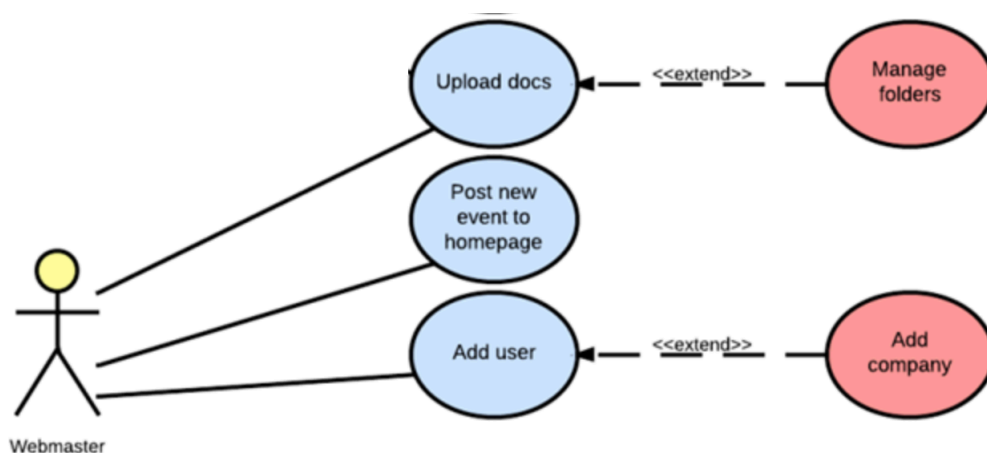
1.14 Návrhové diagramy

Podkapitola popisuje využití formálních prostředků vhodných pro specifikaci požadavků, také konceptu návrhu základního schématu datové struktury aplikace.

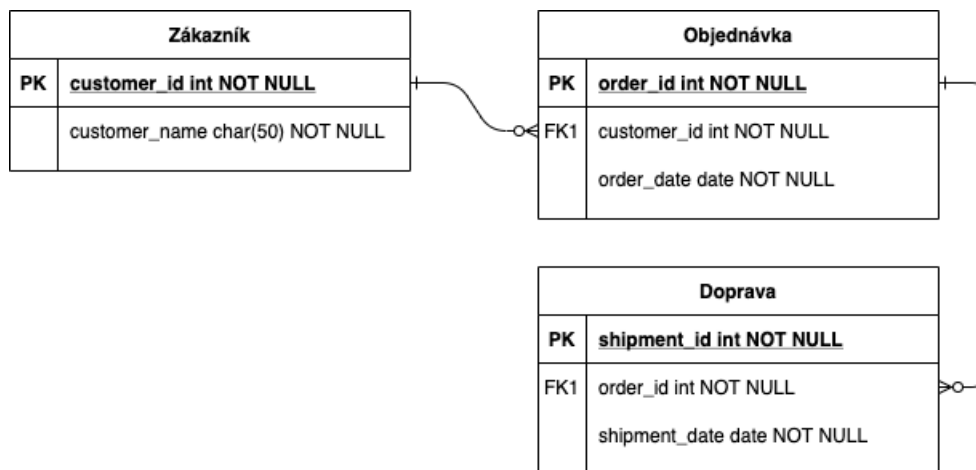
Unified Modeling Language

Unified Modeling Language je vizuální modelovací jazyk, který pomáhá vzájemnému porozumění zadavatele a vykonavatele projektu. Příklad diagramu jazyka je znázorněn na obrázku 1.6.

Jazyk využívá pro popis behaviorální a strukturální diagramy, které se dále dělí na konkrétní diagramy. Jedním z nejpoužívanějších diagramů z kategorie behaviorálních je tzv. Use case diagram (česky diagram případů užití), který popisuje interakci uživatele se systémem. [29]



Obr. 1.6: Diagram případů užití. [29]



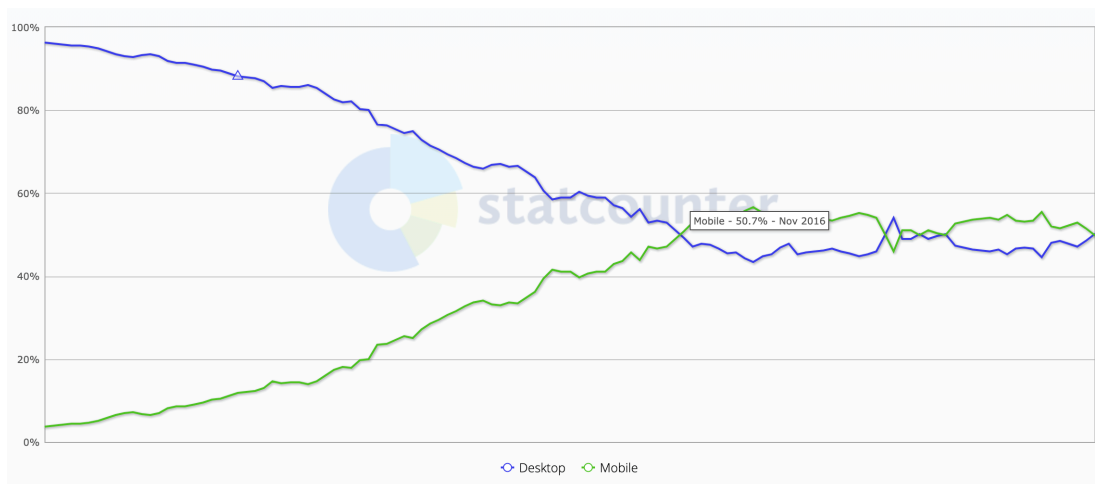
Obr. 1.7: Entity-Relationship Diagram (Zdroj: vlastní zpracování, 2021)

Entity-Relationship Diagram

Vhodným diagramem pro analýzu dat vyskytujících se v systému je ER diagram. Nejčastěji se pomocí něj modelují zmíněné relační databáze (podkapitola 1.13). Diagram je tvořen entitami, které reprezentují datové objekty v systému (obrázek 1.7). Může se jednat o zákazníka, či správce. Jednotlivé entity zahrnují vlastní atributy. Příkladovým atributem entity zákazník je jméno, telefonní číslo či pohlaví. Zpravidla spojují entity vazby, které reprezentují jejich vztah v systému. Vazby mohou být obohaceny o kardinalitu vůči jednotlivým entitám. Součástí diagramu bývají někdy integritní omezení, záleží však na konvenci značení.

1.15 Hypertext Markup Language

HTML vychází z pojmu Hypertext Markup Language. Jazyk je značkovací, je fundamentálním v oblasti vývoje webu. Umožňuje organizovat strukturu webu. Jazyk se skládá ze značek, atributů a elementů, konkrétně řádkových a blokových, které mají unikátní a specifický význam. V současnosti se používá verze HTML5. Ta poskytuje lepší celkový management webových aplikací než verze předchozí.



Obr. 1.8: Srovnání využití internetu na mobilních a stolních zařízeních (říjen 2010 – říjen 2020). [30]

1.16 Přístup Mobile First Design

Naprostou samozřejmostí je dnes možnost užívání webových aplikací na chytrých zařízeních. Dvě miliardy lidí – 51% populace přistupuje k internetu prostřednictvím chytrých telefonů. V roce 2025 se očekává růst této hodnoty na 72,5%. Mobile First Design představuje filozofii o zlepšení tzv. user experience (spokojenosti uživatelů) na základě preference tvorby návrhů. Mobilní zařízení v takovém případě dostávají přednost před zařízeními většími. [31]

Graf 1.8 interpretuje užívání internetu na mobilních a stolních zařízeních za období říjen 2010 – říjen 2020. Je zřejmé, že využití mobilního přístupu zaznamenalo obrovský procentuální růst a nejen že vyrovnalo, ale pomalu předhání podíl užívání stolních zařízení.

1.17 Kaskádové styly

Uživatelé často přehlížejí chyby spojené s funkční logikou, či strukturou aplikace zejména proto, že je většinou vůbec nedokážou detekovat. To ovšem neplatí o vzhledu. Vzhled je velice důležitá část webových aplikací.

Kaskádové styly (nejčastěji CSS neboli Cascading Style Sheets) patří k dalším klíčovým jazykům v oblasti vývoje webu. Jazyk umožňuje stylizování zmíněných HTML elementů,

animace či dokonce vizuální rozvržení (layout) bloků webu pomocí moderních konceptů jako je flexbox, či grid.

Syntactically Awesome Style Sheets

Význam akronyma SASS je Syntactically Awesome Style Sheets. SASS je rozšířením klasického jazyka CSS. Přesněji jde o CSS preprocessor, který má svou vlastní CSS rozšířenou a unikátní syntaxi. Výhod má SASS oproti CSS hned několik. SASS obecně zbavuje CSS kód redundance. Je schopen používat proměnné, umožňuje zanořování, nebo disponuje například tzv. mixins, což jsou v podstatě funkce, které lze aplikovat na bloky kódu bez nutnosti kopírování dat. S použitím SASS syntaxe odpadá nutnost členit kód do závorek – jako členění používá odsazování kódu. Alternativou pro vývojáře, kterým tento způsob nevyhovuje je preprocessor SCSS.



```
@include media-breakpoint-up(sm)
#app
  background-color: green
  h3
    color: yellow

@include media-breakpoint-up(md)
#app
  font-size: 20rem
  background-color: blue

@include media-breakpoint-up(lg)
#app
  background-color: pink

// Extra small devices (portrait phones, less than 576px)
// No media query since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
```

(a) Variabilita elementů dle šířky zařízení. (Zdroj: vlastní zpracování, 2021)

(b) Vymezení různých velikostí dle Bootstrap. [32]

Obr. 1.9: Implementace responsibility aplikace vůči variabilní šířce zařízení. (Zdroj: vlastní zpracování, 2021)

Media Query, Tailwindcss, Bootstrap & Vuetify

Media Queries přináší do CSS možnost selekce napříč různými zařízeními. Podmínky umožňují vývojářům přiřadit variabilní vlastnosti elementům na základě specifikovaného

argumentu, např. šířky zařízení. Na základě svých proporcí, zařízení automaticky detekuje vhodnou větev stylů a aplikuje daná pravidla na zahrnuté elementy. Media Queries lze implementovat jak explicitně v CSS, tak implicitně prostřednictvím knihoven jako je například Tailwindcss, Bootstrap, nebo Vuetify – knihovna pro javascriptový framework Vue (podkapitola **1.18**).

Knihovny poskytují spoustu vlastních CSS tříd aplikovatelných na HTML elementy. Tyto dostupné prostředky šetří enormní množství času a práce „frontendovým“ vývojářům. Tailwindcss, Bootstrap i Vuetify mají stanovené tzv. media breakpointy ve svých mixins, které nesou hodnoty šířky zařízení. Pomocí nich můžeme definovat například barvu pozadí pro displeje malých smartphonů, tabletů, či běžných monitorů. Implementaci lze vidět na obrázku **1.9a**, kde *xs*, *sm*, *md*, *lg* jsou variabilní velikosti šířky v pixelech. Konkrétní detaily zachycuje obrázek **1.9b**. Implicitně by mělo být aplikováno pravidlo stylizování na rozměry chytrých telefonů. Ostatní rozměry se uvádí explicitně dle výše zmíněných podmínek.

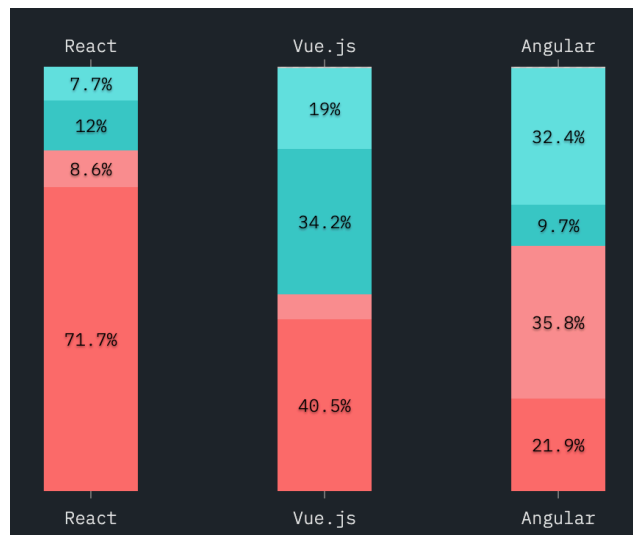
1.18 JavaScript, dostupné knihovny

JavaScript je dynamicky typovaný skriptovací jazyk, který přináší flexibilitu a dynamičnost do oblasti vývoje webu. Patří mezi nejpoužívanější programovací jazyky světa. Používá se jak na straně klienta, tak na straně serveru. Jazyk slouží k vývoji webových i mobilních aplikací. Zjednodušeně lze říct, že JavaScript řídí chování webu.

Existuje řada knihoven neboli frameworků, pro zjednodušení vývoje aplikací. Mezi nejpoužívanější patří React, Angular, Vue a Node.js. Poslední z nich se zaměřuje na rozdíl od předešlých jmenovaných na „backendovou“ část. Jejich popularitu mezi dotázanými lze sledovat na obrázku **1.10**.

Vue

Vue je poměrně mladý, nicméně velice progresivní „frontendový“ framework. První verze vyšla v roce 2014, verze třetí pak v říjnu 2020. Vue 3 se těší stále větší popularitě, nabízí obrovskou míru flexibility z pohledu vývoje, kombinuje principy Reactu a Angularu.



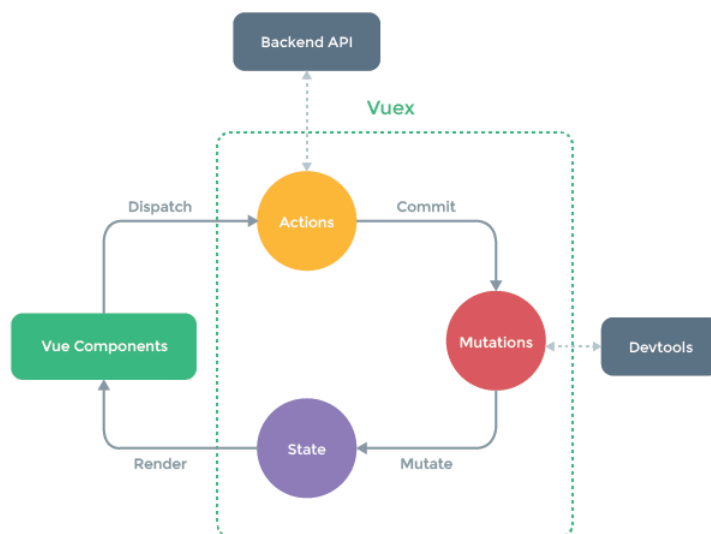
Obr. 1.10: Obrázek je převzat z článku. [33] Srovnání popularity JavaScriptových frameworků. Světlé modrá barva interpretuje procentuální část tázaných lidí, kteří o frameworku slyšeli, avšak jim nepřipadá atraktivní. Tmavší odstín modré zachycuje část lidí, kteří o frameworku slyšeli a mají zájem se ho naučit. Růžová barva poukazuje na část lidí, kteří framework již používali, ale nechtěli by s ním pracovat znovu. Červená barva pak znázorňuje část tázaných, kteří framework používali a vrátili by se k němu opakovaně.

Komponenty

Framework Vue pracuje na principu komponent – to je jedna z jeho největších předností. Umožňuje členit aplikaci do menších částí, což zajišťuje znovupoužitelnost kódu, zvyšuje přehlednost a strukturu projektu. Komponenty lze vnořovat a importovat do jiných komponent a podobně. Komponenty jsou používány prostřednictvím dvou forem – definice globální a lokální cestou.

Router

Dalším efektivním nástrojem, který Vue poskytuje je Router. Ten umožňuje linkovat *URL* adresy v prohlížeči na cesty, které komponenty definují. Router disponuje také možností definice komponentů jako lazy loading route, což je princip, jemuž se podrobněji věnuje podkapitola 1.11. Dochází jen k přepínání jednotlivých komponent v rámci SPA bez komunikace se serverem.



Obr. 1.11: *Koncept nástroje Vuex.* [34]

Vuex

Klíčovým pro tzv. state management – řízení stavu je nástroj Vuex. Vuex poskytuje centralizovaný sklad pro všechny komponenty aplikace. Složen je ze čtyř segmentů – obrázek 1.11. Jedná se o stav, mutace, akce a vykreslování změn do komponent.

Stav uchovává proměnné centralizovaného skladu. Mutace jsou operace, jež se aplikují na stav (spouštěčem může být tlačítko apod.), akce mohou provádět asynchronní operace. Je doporučeno veškeré mutace potvrzovat akcemi, právě kvůli možnosti výskytu asynchronních operací. Akce zároveň komunikují pomocí API s „backendem“ webové aplikace. Pomocí tzv. getterů jsou pak data ze skladu přenášeny do komponent, kde následně dochází k jejich vykreslování.

Z výše zmíněných prostředků lze vyvodit obrovské plus, které Vue přináší. Kód je za jejich použití snadno pochopitelný, znovupoužitelný a snižuje riziko výskytu chyb. Vue má v repertoáru také další aspekty, například animace, direktiva a podobně.

```

public function definition()
{
    return [
        'name' => $this->faker->name,
        'email' => $this->faker->unique()->safeEmail,
        'email_verified_at' => now(),
        'password' => '$2y$30$92dIXUNpkj00r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password
        'remember_token' => Str::random( length: 10),
    ];
}

```

Obr. 1.12: Továrna a nástroj Faker (Zdroj: vlastní zpracování, 2021)

1.19 PHP a knihovna Laravel

Dalším jazykem je jazyk PHP. Podobně jako u JavaScriptu, opět se jedná o skriptovací jazyk, ovšem u jazyka PHP skripty pracují na straně serveru a komunikují s databází.

Laravel

Laravel označuje framework pro jazyk PHP, nejen že velice dobře zvládá kombinaci se zmíněným JavaScriptovým frameworkem Vue, ba dokonce Laravel v období pandemie deklarovalo Vue svým „frontendovým“ standardem. [35]

Podobně jako Vue, Laravel je velmi mladý a pravidelně aktualizovaný framework. Mezi hlavní přínosy patří například Eloquent ORM – česky „Objektivně relační mapování“, které umožňuje mapovat databázové tabulky na instance PHP (instance třídy reprezentuje řádek tabulky). Dalším užitečným prvkem je zabudovaný nástroj příkazové řádky *Artisan*, který umožňuje generovat databázovou strukturu, kostru kódu, základní strukturu architektury MVC s klíčovými metodami nebo usnadňuje migraci napříč jiným vývojovým prostředím. [36]

Dalším užitečným nástrojem je nástroj **Faker**, který umožňuje generování náhodných id, emailů, hesel, nebo i vět a paragrafů. Databázový **Seeder** umožňuje naplnit databázi testovacími daty. Prostřednictvím modelu, následné invokace jeho metody továrny (například továrny na uživatele viz obrázek (1.12) a nástroje Fakeru dokáže seeder vytvořit například 20 uživatelů s náhodnými údaji.

1.20 Databáze MySQL

System řízení báze dat MySQL představuje velmi rychlý, vícevláknový, multi-uživatelský, sql prostředek. MySQL je volně dostupný prostředek pro jakékoli účely, komerční i soukromé. [37] Patří mezi nejrozšířenější a nejpopulárnější databáze po celém světě. Databáze pracuje s relačním databázovým modelem. Kompletní databáze SQL s tabulkami, indexy, triggerami apod. pracuje typicky (na rozdíl například od SQLite) na straně serveru. Také v rámci vývoje webových aplikací se jedná o jedno z nejpoužívanějších databázových řešení.

Laravel a Vue společně tvoří silnou dvojici, jež umožňuje tvořit špičkové webové aplikace. Veškerá důležitá část se odehrává na „frontendu“, který pomocí frameworku Apollo, GraphQL a Lighthouse komunikuje s Laravelem bez potřeby obnovovat webový prohlížeč.

2 ANALÝZA SOUČASNÉ SITUACE

V první části kapitoly je věnována pozornost představení společnosti a popisu její struktury. Dále kapitola rozebírá současný stav společnosti a zkoumá vliv vybraných faktorů na její fungování. V poslední řadě odhaluje silné a slabé stránky, hrozby a příležitosti společnosti pomocí analýzy SWOT.

2.1 Charakteristika společnosti

Následující informace o společnosti jsou čerpány z obchodního rejstříku. [38]

Název společnosti: HORMAK Group s.r.o.

Sídlo: Jurkovičova 244/1, Lesná, 638 00 Brno

Identifikační číslo: 06857302

Den zápisu: 13. února 2018

Restaurace & Pub Les s názvem společnosti HORMAK Group s.r.o. je česká společnost působící v oblasti gastronomie. Relativně mladým podnikem prochází rodinná atmosféra. Restaurace vznikla v roce 2018. Podnik se nachází v klidné části města Brna.

Majitel je současně také spoluvlastník dalšího gastronomického podniku, kde by se po případné úspěšné implementaci, nasazení a překonání krize dal navržený informační systém využít také – prostřednictvím migrace či rozšíření.



Obr. 2.1: Logo Restaurace & Pub Les. [39]

Podnik se specializuje jak na moderní zahraniční pokrmy, tak na nabídku jídel z české kuchyně. V oblasti severního Brna, konkrétně v oblasti Lesné patří k podnikům velice

populárním, s moderním a novým interiérem. Každý všední den je k dispozici obědové menu. Restaurace si zakládá na kvalitní nabídce konzumního lihu a piva, funguje také v rámci večerního provozu. Obědové menu se vydává od 11-15 hodin, otevírací doba restaurace je od 11-23 hodin.

V létních měsících restaurace pořádá venkovní akce, jež jsou často doprovázeny živou hudbou. Zázemí restaurace se neustále rozrůstá. Přibyla nová letní zahrádka, v procesu návrhu a výstavby je i zahrádka zimní. Celkem čítá kapacita podniku na 37 stolů. V letní sezóně prodej obědového menu pravidelně dosahuje průměru 130 kusů jídel za den. Díky možnému venkovnímu grilování a dostatečné kapacitě venkovního posezení má restaurace vybudovanou silnou základnu zákazníků v oblasti její lokality.

2.2 Analýza současné situace

Důsledkem vládních opatření spojených s řešením pandemie Covid-19 podnik provází nepříjemné problémy nejen v oblasti financí. Na základě stavu, ve kterém se podnik nachází již od března roku 2019, je restaurace nucena se střídavě adaptovat na otevírací dobu, možnosti výdeje jídel, či úplné uzavření hospody. Situace donutila některé zaměstnance restaurace dočasně změnit povolání, a to kvůli nestabilním příjmům podniku.

Majitel restaurace věří, že by podniku nejen v současné, ale i v době budoucí pomohla optimalizace procesů rozvozu jídel a řešení elektronických objednávek. Tato optimalizace by byla schopna překonat, či alespoň zmírnit dopad zmíněných problémů. Restaurace nevyužívá žádné z poskytovaných populárních služeb (Wolt, Dáme Jídlo), protože má obavy, že by nebyla schopna pokrýt vyšší zájem zákazníků (v rámci rostoucí poptávky Brna). Nevole je rovněž spojena s poplatky službám, které rozvoz zprostředkovávají, nehledě na současně nízké výnosy.

Se zavedením procesu elektronických objednávek je ideální spojit proces řízení online rezervací. Integrace obou procesů by přinesla ihned několik optimalizačních výhod v kontextu posilování efektivity podniku, například redukci vytížení zaměstnanců, zavedení elektronická evidence.

Co se týče správy směn zaměstnanců, současné řízení bývá často neefektivní. S ohledem na fakt, že se jedná o rodinný podnik (zaměstnanci také pracují v druhé restauraci spoluvlastněné majitelem), dochází někdy ke komplikacím a komunikačním problémům způsobených nedostatkem organizačních praktik. Zavedený systém by mohl současně přinést modul pro správu směn s přístupem zaměstnanců i majitele.

Majitel má v úmyslu systém do firmy co nejdříve zavést. Změna v podobě nového systému by měla přinést zjednodušení průběhu zmíněných firemních procesů, a také podnik rozšířit o procesy nové. Chce-li podnik zůstat konkurenceschopný, automatizace procesů je v dnešní době nevyhnutelná, navíc sebou nese mnoho výhod.

2.3 McKinseyho model 7S

Analýza vnitřního prostředí modelem 7S má za úkol odhalit nesrovnalosti a slabé stránky podniku. [40] Analýza je tvořena sedmi složkami, které jsou vstupem do souhrnné analýzy SWOT.

Informace, jež jsou vstupem do analýzy 7S jsou získány prostřednictvím ústního sdělení vybraného podniku. [41]

Strategie

Podnik stále doufá a věří v překonání současné situace a opětovný příznivý vývoj. Jak již bylo zmíněno, podnik je mladý, rozzrůstal se a velice prosperoval. Plánuje se výstavba zimního venkovního posezení. Další z cílů, je nasazení informačního systému, který by podniku pomohl optimalizovat jeho řízení. V současnosti však nelze stanovit a vytyčit konkrétní data z důvodu nepredikovatelné budoucnosti.

Struktura

Na vrcholu organizační struktury stojí většinový majitel a jednatel. Oba zastupují také manažerské funkce, starají se o zásobování (logistiku) i provozní činnosti restaurace. Na starost mají rovněž správu směn, organizují akce a podobně.

Standartně podnik zaměstnává čtyři kuchaře. Aktuálně byl jejich počet snížen ze čtyř na dva, jelikož jich současná situace více nevyžaduje, a to z důvodu nestabilního provozu. Za běžného provozu podnik zaměstnává také 8 číšníků, z nichž někteří pracují na částečný úvazek. Právě variabilita dostupnosti zaměstnanců, občasné křížení a takřka náhodné zastupování směn ve druhém majitelově podniku nekoresponduje s efektivním řízením. Typicky jsou na směně dva číšníci, při organizovaných večerech pak tři. Nedílnou součástí jsou také služby spojené s úklidem, které bývají vykonávány stabilně dvěma zaměstnanci. Ti však na chod podniku nemají příliš vliv.

Systémy

Veškeré objednávky a rezervace stolů probíhají telefonicky, podnik současně nedisponuje žádným systémem, který by podobné funkce umožňoval. Restaurace využívá pouze pokladní systém KASAmx. Restaurace má statické webové stránky, jejichž hlavním problémem je právě statický obsah. Bez zkušenějšího zásahu není možná ani změna jídelního lístku, natož vykonávání pokročilých funkcí. Mezi využívané marketingové platformy podniku patří facebook a instagram, kde zaměstnanci propagují různé jídelní nabídky či plánované události.

Styl řízení

V podniku převažuje demokratický styl řízení. Podnik vyniká komfortní a příjemnou atmosférou, zakládá si na měkkých dovednostech zaměstnanců. Prioritou je kvalitní vystupování a komunikace. Zaměstnanci jsou často součástí rozhodování, přicházejí s různými nápady a podílí se na úpravách prostor. Vzhledem ke kvalitě vzájemných vztahů si může manažer dovolit delegovat mnoho povinností na některé zaměstnance (tvorba a udržování propagace nabídek, akcí, apod).

Skupina, spolupracovníci

Jak již bylo řečeno, mezi zaměstnanci převládá přátelský kontakt. Zaměstnanci jsou ochotni a schopni vykonávat příslušné funkce a navzájem se doplňovat. V podniku převládá mladý

kolektiv. Podnik umožňuje svým zaměstnancům spolupráci formou brigády, mezi zaměstnanci patří i studenti. Kolektiv je velice pozitivně naladěný a aktivně se podílí na plánování atraktivních nabídek, akcí a podobně.

Schopnosti

Mezi nejlepší a nejsilnější schopnosti podniku patří jeho celkový energy management. Vitální přístup, zájem majitele i zaměstnanců udržuje podnik po celou dobu krize v optimistickém postavení. Zmíněný kooperativní přístup je silnou stránkou, tedy i schopností podniku.

Sdílené hodnoty

Základní sdílenou vizí je růst podniku. Nejen majitel, ale také zaměstnanci usilují o zlepšování a budování kvalit podniku. Mezi tyto kvality patří například dobré jméno restaurace, či budování pozitivních a dobrých vztahů mezi zaměstnanci navzájem, tak i vůči svým zákazníkům. Důležité sdílené hodnoty doplňuje poctivý, tvořivý přístup a profesionalita. Negativní postoj je shledáván nežádoucím a destruktivním. Podnik umí velice dobře pracovat jako tým.

2.4 Analýza PESTLE

Druhou prováděnou analýzou podniku je analýza PESTLE. Analýza se zabývá politickým, ekonomickým, sociálním, technologickým, legislativním a v poslední řadě ekologickým prostředím. [42] Ve vybraném podniku stojí za zmínění analýza politického a legislativního prostředí, dále také prostředí sociálního a ekonomického. Nutno dodat, že v současné situaci se jedná o velice nestabilní a proměnlivé aspekty. Zjištěné informace lze opět považovat za vstupy do analýzy SWOT.

Politické a legislativní prostředí

Největší vliv má na podnik aktuálně prostředí politické a legislativní, které silně ovlivňuje prostředí ekonomické. V důsledku řady opatření přijatých orgány státní správy k ochraně obyvatelstva a prevenci nebezpečí vzniku a rozšíření onemocnění Covid-19 byla činnost

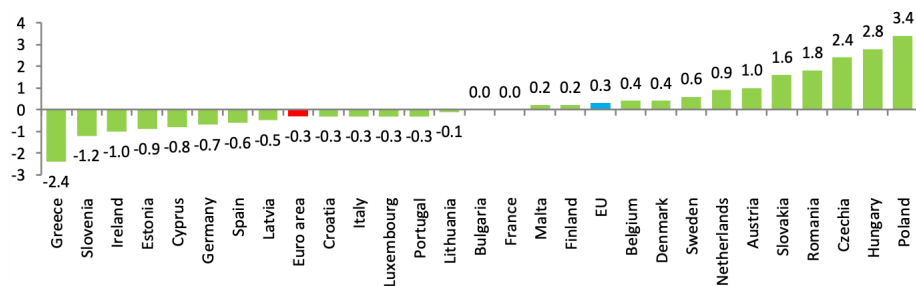
řady společností včetně restaurací a pohostinství zakázána či omezena, a tedy došlo k zásadnímu nebo výraznému negativnímu zásahu do jejich podnikání bezprostředně v důsledku uvedených opatření. Zavedená opatření vedou nemalé množství podniků až k problémům souvisejícím s jejich existencí.

Ohlédnutím do nedávné minulosti lze usoudit, že podnik neměl a nemá situaci jednoduchou. Na základě úprav a změn legislativy a zákonů, vyhlášení nouzového stavu dle Ústavního zákona o bezpečnosti České republiky 110/1998 Sb.[43], musel podnik střídavě otevírat, zavírat, užívat výdejní okénko, měnit provozní dobu. Majitelé přitom netušili a stále netuší, jak se bude dále situace vyvíjet, tudíž platili stálý nájem i v případě nutnosti uzavření podniku. Fakt, že vývoj vládních opatření byl obtížně predikovatelný, velice dynamický a měnil se v podstatě denně, dělá celou situaci z pohledu podniku ještě mnohem horší.

Současně se na podnik také vztahují legislativní předpisy o hygieně, provozu či výstavbě a rekonstrukci. V případě informačního systému, by uživatelé měli být v rámci dodržení obecného nařízení o ochraně osobních údajů (GDPR) informováni o uložení osobních údajů ve formě cookies prohlížeče (viz následující kapitola).

Ekonomické prostředí

Globální ekonomická krize je patrná již od prvního čtvrtletí roku 2020. Nejen gastronomickou oblast, ale celý terciální hospodářský sektor se dá hodnotit jako sektor nejvíce postižený krizí. Mezi hlavní faktory ovlivňující ekonomické prostředí lze považovat zejména inflaci a vývoj cen, který je nejvyšší za posledních 8 let. [45] Na obrázku 2.2 lze vidět, že míra inflace v České republice byla v prosinci roku 2020 třetí nejvyšší v Evropské unii. Růst cen je třeba očekávat i v roce současném (tj. 2021), na základě omezené nabídky způsobené pandemií Covid-19, růstem ceny komodit, zvyšováním cen energií či rychlostí a postupu vakcinace v porovnání s ostatními státy.



Obr. 2.2: Míra inflace EU za rok 2020. [44]

Ceny nemovitostí mají neustálou tendenci růstu, avšak ceny krátkodobých pronájmů mají aktuálně spíše efekt opačný, a to klesající. [45] Nutno dodat, že právě platba astronomických cen pronájmů přiměla mnoho podniků k insolvenční, což zapříčinilo růst nabídky prostor dostupných na tomto trhu.

Sociální prostředí

Jak již bylo zmíněno, restaurace se nachází v severní části Brna, konkrétně na území městské části Lesná. Na severu Brna žije více než čtyřicet dva tisíc obyvatel, městská část Lesná pak čítá přes šestnáct tisíc obyvatel. [46]

Restaurace je mladý a moderní podnik. V uvedené lokalitě je jednou z nejnavštěvovanějších. Mnoho zákazníků navštěvuje podnik pravidelně a rádi se vracejí. Další skupinu tvoří množství zákazníků nových, v letním období také mnoho cyklistů. Třetí část zákazníků tvoří ti, kteří provádí objednávky telefonicky a využívají možnost výdeje.

Ve výsledku se zákazníci snažili podnik podpořit kdykoliv to šlo, nicméně v odpoledních hodinách restaurace zela prázdnotou z důvodů vládních nařízení. Kvalifikace kuchařů a číšníků je samozřejmostí. Podnik je energický, což zaměstnancům přidává na snaze a podniku na celkové kvalitě.

Technologické prostředí

Ačkoli se to nezdá, nepřímo oblast gastronomie zasahují i inovace a nové technologie. Návrhovaný informační systém, jež je předmětem této práce, by měl optimalizovat provoz podniku a zvýšit tak jeho celkovou efektivitu. Co se prodeje a zákazníků týče, online

objednávky a rozvozy jsou dnes již naprostou samozřejmostí. Nicméně nutno dodat, že při realizaci vlastního podnikání, není systém rozvozu jídla pro podnik primární ani klíčový.

Ekologické prostředí

Podnik z oblasti pohostinství musí dbát platných předpisů. Může jít například o předpisy hygienické, odpadovou politiku a tzv. „*food waste*“, který se stává populárním tématem. Podnik by měl nejen z etického hlediska, ale také z hlediska vlastních úspor implementovat metody řízení množství surovin, potravin a zpracování odpadu.

2.5 Porterova analýza pěti sil

Následující podkapitola se zabývá analýzou konkurence, a to pomocí metody Michaela Eugena Portera. Cílem je analyzovat sílu konkurence v konkrétním odvětví na základě pěti faktorů konkurence. [47]

Informace obsaženy v rámci analýzy konkurence jsou získány prostřednictvím ústního sdělení vybraného podniku. [41]

Stávající konkurence

Stávající konkurenci je třeba hodnotit za normálních podmínek (za normální podmínky označujeme stav bez opatření a omezení daných vládou apod.). V případě Restaurace & Pub LES je důležitá konkurence lokální, přesněji v městské části Brna – Lesné a blízkého okolí v oblasti gastronomie. Mezi největší konkurenty patří restaurace Dukát a restaurace Charlies 4est.

Restaurace Dukát představuje silnou konkurenci, a to kvůli dlouholetému provozu, nově provedené rekonstrukci a nižším cenám, čímž podnik míří na nepatrně jinou kategorii klientely. Restaurace však na úkor nižších cen může být na jiné úrovni v oblasti kvality produktů a služeb. Dukát je od podniku LES vzdálen takřka přes silnici, nicméně disponuje výhodnějším umístěním, a to u hlavní silnice. Tento fakt může být pro Dukát konkurenční výhodou, jde-li o výběr náhodných zákazníků.

Druhá restaurace, jež je v oblasti na dosah se nazývá Charlies 4est. Nachází se zhruba 10 minut chůze od restaurace LES. Podobně jde také o relativně mladou restauraci, jejíž prostory prošly modernizací. Podnik nabízí podobně jako restaurace LES tankové pivo, což může být pro zákazníky jedno z lákadel.

Během opatření spojeného s uzavřením restaurací měly oba konkurenční podniky taktéž omezený provoz, a to skrze výdejní okénko.

Nová konkurence

Pandemie přinesla mnohým podnikům v oblasti gastronomie nepříznivé výsledky. Některé podniky byly nuceny ukončit svou činnost.[48] Postupem času vzniká nabídka prostor k pronájmu. V situaci, kdy Česká republika bude proočkována vakcínou a dojde k rozvolnění všech opatření (v obou případech by se mohlo jednat o letní sezónu roku 2021), lze očekávat opět příznivý vývoj a rostoucí zájem ze strany zákazníků. Fakta, příležitosti a podmínky pro vstup do podnikání v dané živnosti dle mého názoru mohou otevřít dveře potenciálním zájemcům o nové podniky.

Vliv odběratelů

Vliv odběratelů je pro podnik klíčový. V době výdejového okénka byl odběr obědových menu někdy až třikrát menší než za normálního provozu. Zájem odběratelů souvisí s výše uvedenou konkurencí. Pro udržení či budování nových vztahů je nutné uspokojovat potřeby odběratelů. Ti preferují hlavně kvalitu, v poměru s cenou produktů.

Vliv dodavatelů

Podnik se snaží poskytovat zajímavou nabídkou jídel, která jsou zároveň kvalitní. Pravidelně využívá například domácí pečivo – například pro burgery a produkty podobné. Nicméně podnik nelze kategorizovat do skupiny luxusních. Lihoviny a jiné produkty či suroviny jsou srovnatelné s kvalitou produktů velkoobchodů. Tudíž vliv dodavatelů je na podnik minimální.

Substituční produkty

Výše zmíněné restaurace nabízí vzájemně nahraditelné substituční produkty. Ačkoli se LES snaží působit ve zmíněném lokálním prostředí, substituční produkty lze obstarat skrze online služby, či kamenné obchody s lihovinami a potravinami.

V současné situaci jsou lidem přístupné substituční produkty skrze služby jako je Dáme jídlo či Wolt.[49] Služby obsahují rozsáhlý repertoár gastronomických podniků nabízejících produkty různých světových kuchyní. Uvedené služby v době pandemie byly ty, které na jednu stranu gastronomické podniky držely při životě, na stranu druhou jim hypoteticky uškodily, a to skrze spektrum nabídek široké konkurence v rámci celého města.

2.6 Analýza SWOT

Jedná se o přehlednou a souhrnnou metodu, která pracuje s identifikací silných a slabých stránek, dále také identifikuje příležitosti a hrozby. Analýza hodnotí jak vnější, tak vnitřní prostředí firmy. [50] Výsledkem analýzy by měl být jakýsi přehled či případný souhrn opatření, na základě kterých může podnik realizovat konkrétní kroky.

Silné stránky

Mezi silné stránky společnosti nepochybně patří lokalita a prostory obecně. Restaurace disponuje rozsáhlým vnitřním i venkovním prostorem, nechybí zde ani dětské hřiště, které využívá spoustu rodin s dětmi. Další velkou výhodou je také parkoviště, které je součástí areálu.

Důležité jsou výstupy z McKinseyho modelu 7S (podkapitola 2.3). Schopnosti, sdílené hodnoty, skupina a samotný styl řízení, kterým podnik disponuje může být klíčem k dobremu fungování podniku. Prostor je velice živé a čisté, srší z něj pozitivní atmosféra. Týmovost, poctivost a vzájemný respekt jsou dle mého názoru klíčové faktory, díky kterým je restaurace v této oblasti tak úspěšná.

Slabé stránky

Slabou stránkou může být nedostatek vlastních finančních zdrojů pro samotný provoz podniku. Jednat se může o finance spojené s výstavbou exteriéru, ale v současnosti bohužel o finance provozní.

Za další stránku oslabující společnost lze považovat komunikaci napříč vnitřním prostředím. Podnik využívá mnoho komunikačních kanálů, což často vede k nedorozuměním v plánování směn – vliv na tento fakt má správa druhého podniku majitele. Benevolentní a přátelský rodinný přístup na jednu stranu umožňuje příjemnou atmosféru na pracovišti, na druhou stranu by nebylo na škodu implementovat organizační prvek, či systém, který umožní centralizované řízení a přehled.

Restaurace nevyužívá žádný systém, který by ji pomohl podnik lépe řídit, případně zlepšit jeho služby. Ačkoli se jedná o faktor vnitřního prostředí podniku, v posledním roce bylo znát, že tento nedostatek může mít nemalý podíl na podporu fungování podniku. Provázanost vnějšího a vnitřního prostředí je zřejmá. Krize zasáhla většinu podniků. Restaurace nebyla téměř schopna na situaci reagovat, nebyla na nic takového připravena. Podnik byl v podstatě odkázán na naději v brzké zlepšení situace a rozvolnění probíhajících opatření. Právě tato neznalost situace, čekání a nemožnost fungování se podniku prodražila. Situaci lehce zlepšilo zavedení a možnost využití prodeje formou výdejního okénka. Vedení podniku se shodlo, že informační systém s možností objednávek a rozvozu by v takové situaci mohl být velice přínosný.

Příležitosti

Restaurace je pořád relativně nová, snaží se rozvíjet a růst i přes aktuální situaci. V dnešní době je mnohem větší trend chodit do podniků na oběd, večeři, či posezení než dříve. Před pandemií se restauraci podařilo oslovit dostatečně velké množství zákazníků z přilehlých oblastí, kteří zůstávají podniku věrní a snaží se jej podpořit. Dle analýzy sociálního prostředí analýzy PESTLE 2.4 a Porterovy analýzy pěti sil 2.5 lze usoudit, že podnik má v dané lokalitě potenciál.

Pomineme-li na chvíli vliv pandemie, ke hlavním příležitostem, které sebou nasazení moderního informačního systému nese, patří vývoj značky podniku na poli konkurence, dále možnost oslovit a rozšířit spektrum zákazníků. Současně systém přinese možnost řízení směn, rezervací a možnost objednávat jídlo elektronicky. Současně by měl informační systém přinést řešení pro případ, že by se podobné uzavření podniku opakovalo. V takovém případě bude podnik schopen se na situaci adaptovat a provádět objednávky.

Hrozby

Největší hrozbou se jeví krach společnosti. Ačkoliv to vypadá, že dochází ke zlepšení situace spojené s pandemií, časté změny legislativy jsou teď naprosto běžné. Je třeba tak brát v potaz scénář, kdy se podnik může ocitnout na hranici insolvence a bude nucen ukončit svou činnost.

Na základě analýzy ekonomického prostředí viz 2.4 lze pozorovat vysokou míru inflace. Zdražování surovin, energií a zejména nájmu může mít na podnik velice nepříznivý vliv.

Ačkoli existují v blízkém okolí dva konkurenční podniky, nemají takový vliv, který by podniku způsobil vážnější problémy spojené s jeho provozem či existencí. Po pandemii však může dojít k otevření nových podniků.

2.7 Shrnutí analýz

Analýza vnitřního prostředí podniku poukázala na ambice a chuť majitele i zaměstnanců dále budovat kvality podniku. Společně dále kooperují a snaží se mu pomoci v růstu. Podnik se rychle v dané lokalitě stal konkurenceschopným. Možnost nasazení informačního systému vítá a podporuje. Pokud nebereme v potaz dle mého názoru nepřímou konkurenci (skrze online služby), jsou v dané lokalitě pouze dva další konkurenční podniky.

Analýza vnějšího prostředí se věnovala zejména aktuálním, zdlouhavým restrikcím a omezením na podnik dopadajících. Podnik patří do oblasti pandemií nejvíce zasažených, i přesto se snaží situaci překonat.

Na základě výše analyzovaných segmentů SWOT analýzy lze vyvodit některé současné problémy, zejména však pozitiva spojená s fungováním společnosti. Zavedení informač-

ního systému by mohlo některé ze zmíněných, nepříznivě vyvíjejících se problémů vyřešit, či alespoň částečně zmírnit. Největší hrozbou se jeví aktuální vývoj situace spojené s pandemií, který by mohl vést nejen k zadlužení, ale také k úplnému bankrotu.

Ačkoli se aktuální situace nemusí zdát příznivá – existují jisté překážky a problémy, například kvůli opatřením danými neustálou změnou legislativy apod., postupnou vakcinací populace lze vyhlížet příznivou prognózu. Podnik se dívá na situaci s nadhledem a věří, že je schopen období přečkat, podpořit růst a řízení podniku nasazením vlastního informačního systému.

3 VLASTNÍ NÁVRHY ŘEŠENÍ

Z předchozí, analytické části diplomové práce vyplývá zjištění, zda a za jakých podmínek by bylo vhodné systém implementovat. Kapitola vlastní návrhy řešení se věnuje aplikaci Lewinova modelu, identifikaci požadavků, sestavení návrhu a následné implementaci systému. Zavedení systému by mělo vést ke zkvalitnění průběhu firemních procesů, odstranění, či částečné eliminaci odhalených slabých míst podniku relevantních ke změně, respektive k naplnění požadavků zadavatele souvisejících s návrhem a nasazením vlastního informačního systému.

3.1 Výběr systému

Výběr informačního systému přináší několik možností a alternativ. S ohledem na velikost podniku, jeho potřeby, či existující řešení konkurence připadá v úvahu projekt na zakázku. V takovém případě lze vycházet z teorie, prostředků a metod popsanych v první kapitole této práce.

Majitel má v úmyslu systém co nejdříve do firmy zavést. Změna v podobě nového systému přinese zjednodušení průběhu firemních procesů, také podnik doplní o procesy nové.

3.2 Lewinův model

Lewinův model lze rozčlenit do tří fází – fáze rozmrazení, fáze změny a fáze zamrazení. [51] Úvodem je třeba určit síly, které změnu podporují, současně síly, které jsou proti. Provedena bude analýza **silového pole**, a to pomocí kladného a záporného ohodnocení působících sil na škále hodnot **1 - 10**. Výsledná převaha konkrétních sil rozhodne o realizaci změny. V případě podniku je interpretace následující.

Síly působící pro plánovanou změnu

- Požadavek na objednávkový systém, správa rezervací a směn
- Růst tržeb, růst efektivity a pole působnosti
- Aktualizace akčního obsahu a událostí
- Podpora v kritických situacích

Síly působící proti plánované změně

- Nepříznivý vývoj pandemie
- Potenciální neochota zaměstnanců se systémem pracovat
- Pořizovací cena

I přes aktuální situaci majitel počítá s progresivním vývojem a věří v brzké zlepšení. Převládají **síly působící pro nasazení systému**, tudíž bude změna realizována.

Účastníci změny

Agentem změny je externí jednatel, respektive já, jakožto autor práce a tvůrce systému. Na základě konzultací, z nich plynoucích požadavků zadavatele bude postupně vytvářen návrh a prováděna implementace. Investorem změny je majitel podniku neboli zadavatel projektu. Advokátem změny jsou samotní zaměstnanci podniku.

Intervence

Co se týče intervence, změna zasáhne tři intervenční oblasti. Systém pomůže eliminovat kolize ve směnách, redukovat vytížení zaměstnanců skrze telefonické objednávky a rezervace, zároveň pomůže podniku inovovat a jít s dobou po technologické stránce.

Zasažené intervenční oblasti

- **Lidské zdroje a jejich řízení** – systém umožní zlepšení administrativy a řízení změn zaměstnanců.
- **Technologie firmy** – v současné době podnik žádný systém nemá, podnik bude mít nový technologický prostředek, který pomůže k jeho zefektivnění a zviditelnění.
- **Komunikační a organizační toky a procesy firmy** – systém poskytne lepší komunikaci napříč podnikem; poskytne centralizované místo pro s podnikem související záležitosti.

Organizační struktura firmy zůstává jedinou nezasaženou intervenční oblastí. Systém bude mít vliv na majitele i personál, avšak nebude mít vliv na celkovou organizační strukturu podniku.

Fáze rozmrazení

Fáze rozmrazení sebou nese analýzu současného stavu a snahu o získání dostatečného množství informací spojených se změnou. Na základě informací, specifikací a analýzy požadavků pak vzniká návrh podoby a struktury systému, kterou agent realizuje.

Fáze změny

Fáze spočívá v realizaci změny, ve vytvoření návrhu případů užití a datového modelu. Dále ve vytvoření návrhu uživatelského rozhraní, samotné implementaci systému a v poslední řadě nasazení systému do produkčního prostředí. Detailnímu popisu se věnuje kapitola později.

Fáze zamrazení

Poslední, konečná fáze modelu je fáze zamrazení. Realizovaný systém je představen a nasazen do produkce. Probíhá finální testování a hodnocení úspěšnosti systému. Bude sledováno množství objednávek, čtvrtletí po nasazení bude také hodnocena spokojenost zaměstnanců a zákazníků, za účelem zjistit přínosnost systému pro podnik.

3.3 Identifikace a specifikace požadavků

Specifikace požadavků je stanovena na základě konzultace se zadavatelem. Požadavky lze rozdělit na požadavky funkční a nefunkční, viz podkapitola 1.5.

Funkční požadavky

Informační systém přinese řešení v organizaci a správě směn zaměstnanců, kteří se střídají na směnách dle potřeb napříč více objekty. Současný stav zadavateli nevyhovuje, organizace a řízení směn probíhá chaoticky. Zadavatel by rád přijímal elektronické objednávky, případně řešil online rezervace.

Požadavky lze zachytit návrhem případů užití systému popsanych v sekci 3.4. Se systémem budou pracovat zaměstnanci, zákazníci a správce (případně majitel) podniku. Právě uvedení aktéři budou v systému rozdělení dle oprávnění svých rolí.

Vstupy a entity systému řeší entity-relationship diagram, jenž je znázorněn na obrázku **3.2**. Výstupy systému jsou informace dostupné aktérům. Jedná se o přehled a možnost zápisu směn, přehled objednávek a rezervací v případě zaměstnanců a správce. Správce by měl mít také přehled o zákaznících a zaměstnancích, spravovat jejich účty a profily.

Registrace a přihlašování

Přístup do systému bude prováděn výhradně prostřednictvím ověření identity dle emailu a hesla uživatele.

Ostatní funkce a operace prováděné nad daty, jenž bude systém umožňovat rozebere podkapitola **3.6**.

Nefunkční požadavky

Způsob řešení a implementace nechává zadavatel na zpracovateli neboli agentu změny. Použité technologie budou v případě UI a vizualizace Vue. V případě aplikační logiky se jedná o PHP framework Laravel za použití MySQL databáze.

Responzivita webové aplikace

Je naprosto nezbytné, aby byl systém použitelný na zařízení všech rozměrů a platform. Proto bude systém vyvíjen formou webové aplikace. Kvalitní, responzivní weby mají obrovský vliv a tvoří dobrý dojem při hodnocení zákazníkem, současně však také při hodnocení a plnění podmínek responzivního návrhu v případě webových vyhledávačů. Analýza poměru využití přístupu k internetu prostřednictvím mobilních zařízení vůči přístupu stolních zařízení toto jen potvrdila, viz podkapitola **1.16**.

Moderní přístup

V době, kdy na internetu lidé tráví opravdu veliké množství času je snadné narazit na hezké, plynulé weby, či webové aplikace. Právě odtud plyne inspirace a požadavek na moderní, rychlý a efektivní systém odpovídající současnosti.

Uživatelské rozhraní a uživatelská zkušenost

Uživatelské rozhraní a zkušenost, či použitelnost jsou dalšími klíčovými požadavky. Systém a manipulace s ním musí být snadno pochopitelná a dostupná ideálně pro všechny kategorie a třídy společnosti. Rozhraní bylo, je a bude postupně konzultováno a upravováno vůči požadavkům zadavatele.

Rozšiřitelnost a modifikovatelnost

Dalším důležitým požadavkem je případná rozšiřitelnost systému o nové komponenty (v případě nových nápadů). Mohlo by se v prvé řadě jednat o napojení systému na online platební bránu, případně rozšíření systému o nové komponenty, jako je například oceňování pravidelných online zákazníků. Architektura aplikace je řešena právě tak, aby byl požadavek splněn. V případě zájmu o rozšíření je možné všechny entity upravit bez nutnosti zásahu. Další možností je vývoj a propojení systému s nativní aplikací.

Dostupnost a spolehlivost

Systém musí být za každou cenu dostupný a řízení provozu efektivní, plynulé. Cílem není podniku přidělovat starosti. To souvisí s výběrem kvalitní služby pro hostování aplikace. Integrita, konzistence a optimalizace aplikace je základ k vývoji dobrého produktu.

Bezpečnost

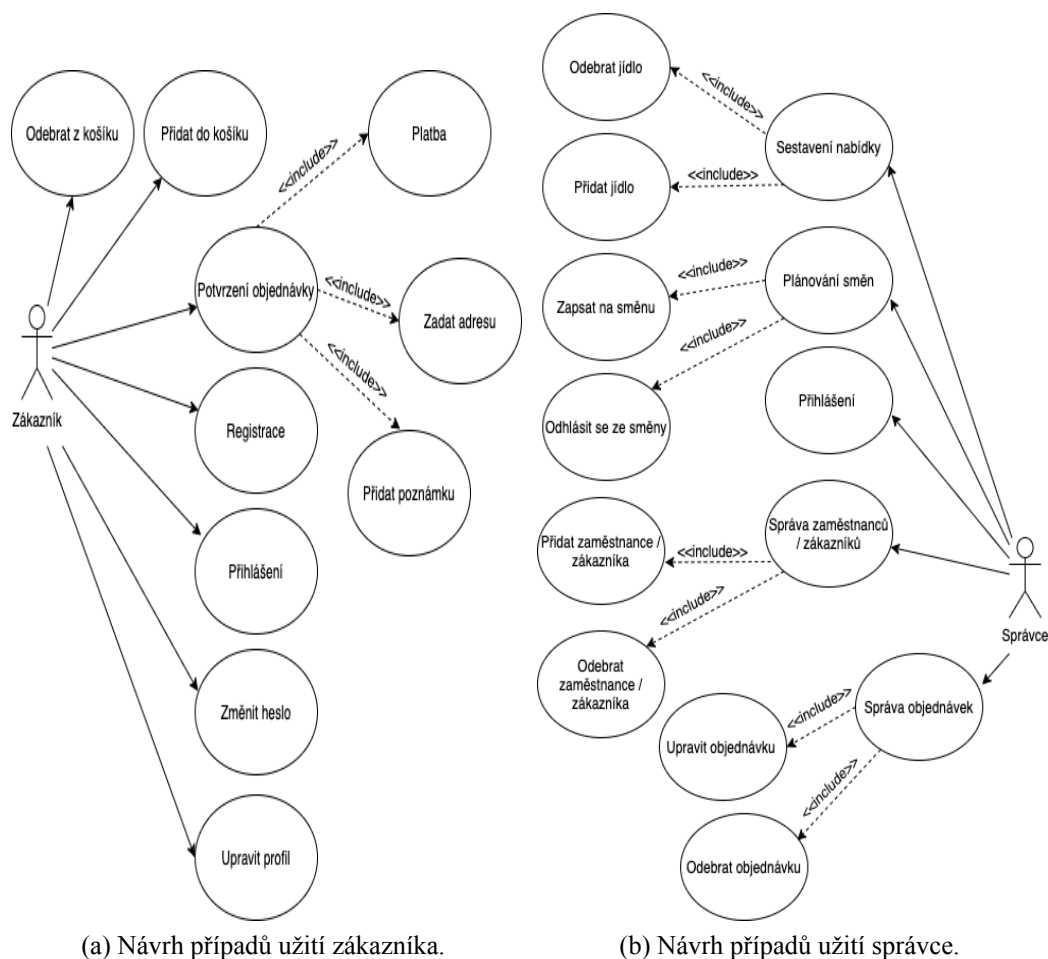
Často podceňovaným faktorem je bezpečnost. Eliminace zdrojů zranitelnosti systému proti útokům, neautorizovanému přístupu, hashování hesel je samozřejmostí. Procesy autentizace, autorizace a šifrování jsou popsány ve fázi implementace systému.

Cena

Poslední, nefunkčním požadavkem, je cena, která má být uměrná práci odvedené při návrhu a zavedení systému. Kalkulace byla stanovena na cca 60 000 Kč bez daně, včetně všech úprav (náklady kapitola zohledňuje později).

3.4 Diagram případů užití

Pro pochopení specifikace požadavků je dobré sestavit návrh případů užití pomocí diagramu jazyka UML. Diagram pomáhá předběžnému vymezení funkcí a komponent systému.

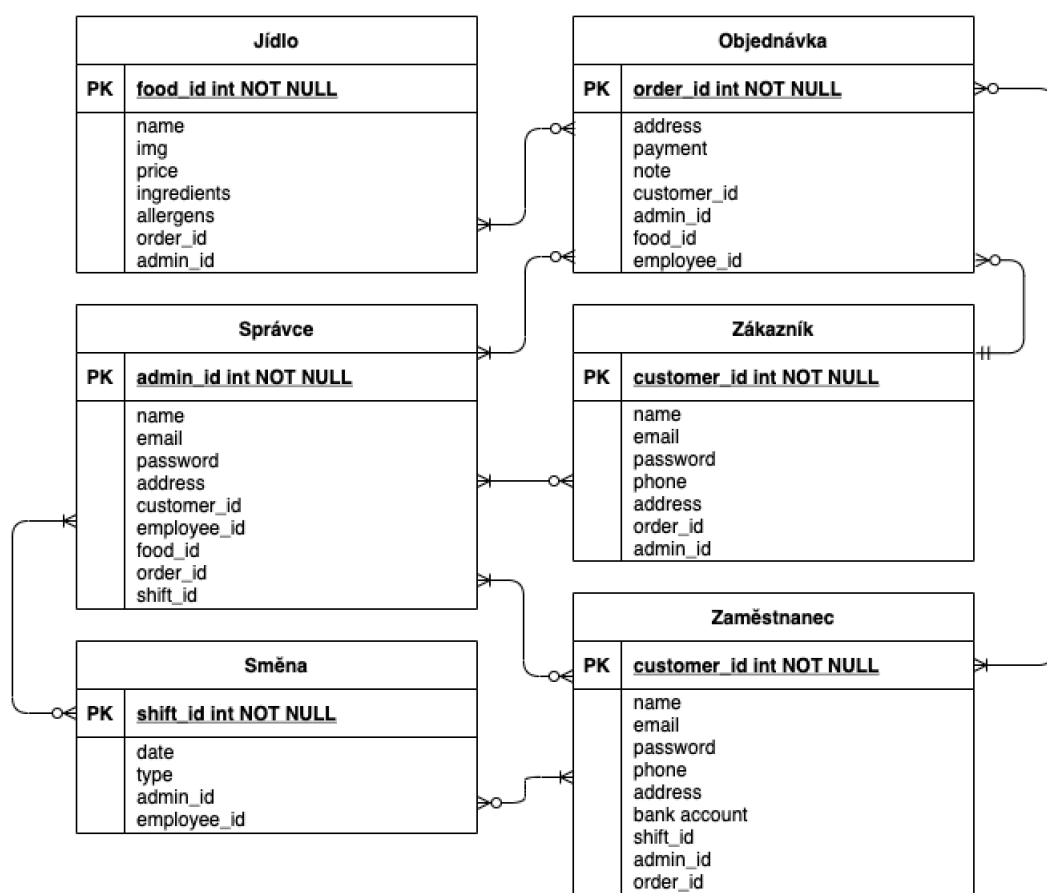


Obr. 3.1: Návrh případů užití pomocí jazyka UML. (Zdroj: vlastní zpracování, 2021)

V oblasti pravomocí a rolí našeho systému jsou přítomni opět tři aktéři – správce, zákazník a zaměstnanec. Prvním aktérem je zákazník. Schéma jeho funkcí lze pozorovat na obrázku 3.1a. Obrázek 3.1b znázorňuje klíčové funkce, kterými by měl disponovat aktér správce. Patří zde správa nabídky a objednávek, zaměstnanců a jejich směn. Třetí aktér, zaměstnanec, disponuje funkcemi kombinující funkce zákazníka a část funkcí správce. Navíc má možnost spravování svých směn a spravování objednávek.

3.5 Datový model

Datový model systému se dá znázornit pomocí entity-relationship diagramu. Mezi hlavní entity datového modelu patří entita správce, zákazník, objednávka, zaměstnanec, směna a jídlo (obdobně entita menu pro daný týden). Uvedený datový model popisuje schéma na obrázku 3.2. Základní koncept, nezachycuje ovšem tabulky vazeb kardinality $M:N$. Tyto vazby jsou znázorněny specifickou grafickou vazbou a cizími klíči, které jsou atributy v příslušných tabulkách. Pro grafickou jednoduchost a přehlednost práce obrázek neobsahuje entity typu článek, rezervace a menu, nicméně jsou entity součástí datového modelu.



Obr. 3.2: Zjednodušený návrh datového modelu pomocí ER diagramu. Schéma neobsahuje vazební tabulky. (Zdroj: vlastní zpracování, 2021)

3.6 Návrh komponent systému

Podkapitola rozebírá návrh struktury systému z hlediska přístupných komponent dle role aktéra. Role aktérů jsou pro řešení oprávnění klíčové.

Co se týče funkcí a operací nad daty systému, je vhodná interpretace také verbálním popisem. Popis lze brát jako jakýsi návrh dle prvotních požadavků. Současně popis funkcí do jisté míry interpretovaly diagramy případů užití v předchozí podkapitole 3.4.

Systém bude mít implementováno konkrétní adresování (komponenty vykreslované při dané URL cestě). Jedná se o adresy, na které budou přistupovat samotní uživatelé. Ve skutečnosti používáme architekturu single page application. Vykreslujeme tedy komponenty do jedné stránky.

Aktér správce

Prvním aktérem je aktér správce. Správce má k dispozici téměř všechny dostupné funkce a je oprávněn přístupem k následujícím komponentám.

- *employees* – komponenta obsahující správu zaměstnanců, jedná se editaci profilu a procházení seznamem zaměstnanců
- *add-employee* – komponenta umožňující přidání zaměstnance do systému
- *employee* – komponenta konkrétního profilu zaměstnance
- *customers* – obdobná komponenta jako *employees*, avšak jedná se o správu zákazníků
- *customer* – komponenta umožňující přístup k profilu konkrétního zákazníka
- *set-menu* – komponenta zahrnující správu jídelního lístku, kde by měl být seznam s možností rozšíření
- *shifts* – komponenta se správou směn. Obsahuje možnost vytvořit směnu se zaměstnanci, která se následně přidá ke směnám existujícím. Směnu by mělo být možné editovat (změnit zaměstnance), či celou směnu odebrat.
- *orders* – komponenta s přehledem objednávek, s možností úpravy či smazání
- *events* – komponenta pro správu událostí
- *order* – komponenta vykreslující konkrétní objednávku

Předpokládáme nízký počet správců, tudíž všechny příslušné routy budou implementovány jako *lazy loading routes*, podkapitola **1.18**.

Aktér zákazník

Druhým aktérem je zákazník, ten by měl mít přístup primárně k objednávkám.

- *customer* – komponenta umožňující přístup k profilu konkrétního zákazníka
- *registration* – komponenta umožňující registraci
- *confirm* – komponenta umožňující potvrzení objednávky, případně přidání poznámky či zvolení způsobu platby
- *order* – komponenta orientovaná na přehled objednávky, případně jejího aktuálního stavu
- *cart* – komponenta zahrnující nákupní košík s jeho položkami
- *reservation* – komponenta umožňující rezervaci stolu

Aktér zaměstnanec

Komponenty přístupné aktéru zaměstnanec by měly disponovat podobnými funkcemi jako komponenty zákazníka. Kromě toho má možnost správy objednávek a plánování vlastních směn.

- *shifts* – společná komponenta se správcem, podobně jako komponenta *orders* a komponenta *order*

Sdílené komponenty

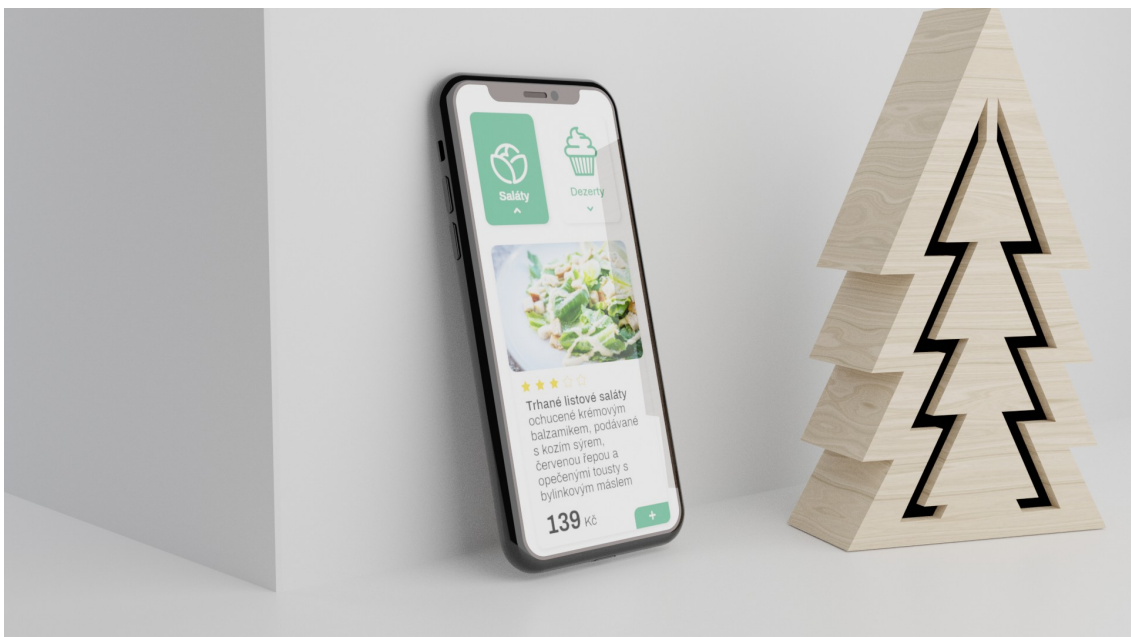
Komponenty sdílené mezi všemi rolemi jsou následující.

- *menu* – komponenta s přehledem aktuální nabídky
- *login* – komponenta pro přihlášení do systému
- *change-password* – komponenta pro změnu hesla
- *orders* – komponenta s přehledem objednávek, funkčnost bude ovšem redukována na základě rolí

V poslední řadě je třeba nastavit speciální případ přesměrování, pro veškeré ostatní, invalidní, či neoprávněné adresování. V takovém případě bude uživatel přesměrován na chybovou stránku.

3.7 Návrh uživatelského rozhraní

Na základě specifikace požadavků jsme vytvořili (společně s kolegou Janem Hanusem) návrh uživatelského rozhraní. Vytvořené návrhy jsou dvojího typu, návrh uživatelského rozhraní pro mobilní zařízení a také návrh pro zařízení větších rozměrů – osobní a stolní počítače.



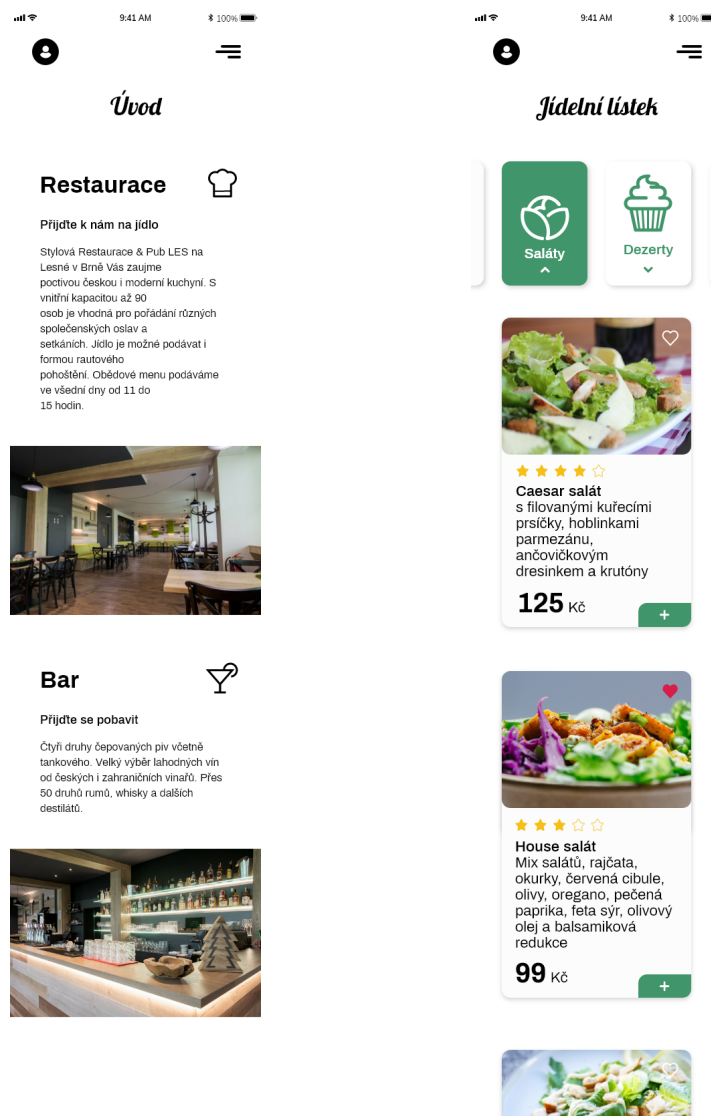
Obr. 3.3: Vizualizace aplikace pro mobilní zařízení. (Zdroj: vlastní zpracování, Jan Hanus, 2021)

Ikony jsou použity pod licencí služeb dostupných na webu www.iconfinder.com. Fotografie a obrázky jsou čerpány pod volnou licencí.

Mobilní zařízení

V případě návrhu pro mobilní zařízení je klíčová responzivita. Následující obrázky ukazují možnou, pravděpodobnou podobu systému. Prvním je vizualizace aplikace (obr. 3.3), dále

obrazovka s úvodní stránkou (obr. 3.4a) a obrazovka s jídelním lístkem (obr. 3.4b). Návrhy pomáhají jak při domluvě se zadavatelem, tak při samotné implementaci. Důraz je kladen na minimalistické a intuitivní zpracování. Ikona uživatele vlevo nahoře bude odkazovat na komponenty s přihlašovací obrazovkou. Uživatel systému zde má také možnost registrace. Požadavek na strukturu registračního formuláře je obsáhlejší, formulář bude obsahovat více textových polí, dle potřeby zadavatele. Principiálně se jedná o jméno a příjmení, adresu, telefonní kontakt, u zaměstnanců také o číslo bankovního účtu.



(a) Návrh úvodní obrazovky.

(b) Návrh jídelního lístku po roztažení.

Obr. 3.4: Návrh úvodní stránky a menu jídel (mobilní zařízení). (Zdroj: vlastní zpracování, Jan Hamus, 2021)

Obr. 3.5: Návrh profilu (stolní zařízení). (Zdroj: vlastní zpracování, Jan Hanus, 2021)

Datum	Objednáno	Čas přijetí	Adresa	Číslo objednávky	Cena celkem
11.1.2021	House salát 1x	13:11	Vedlejší ulice 17 Brno	25888713	99 Kč
02.1.2021	Caesar salát 2x	10:54	Vedlejší ulice 17 Brno	12988897	260 Kč
28.12.2020	Caesar salát 1x	11:32	Vedlejší ulice 17 Brno	19898714	1815 Kč
	Třihany listový salát 5x				
	House salát 1x				
7.11.2020	Třihany listový salát 1x	10:44	Vedlejší ulice 17 Brno	11147896	99 Kč

Obr. 3.6: Návrh přehledu objednávek (stolní zařízení). (Zdroj: vlastní zpracování, Jan Hanus, 2021)

Stolní zařízení

Ačkoliv lze očekávat větší přístup z mobilních zařízení, návrh v rámci stolního zařízení je samozřejmě nutný, a to nejen pro přehlednou správu podniku. Výše přiložené obrázky

demonstrují dva návrhy – přehled účtu zákazníka (obr. 3.5) a přehled objednávek (obr. 3.6).

3.8 Implementace uživatelského rozhraní

Implementace vychází z již vytvořeného návrhu. Některé detaily se však liší na základě respecifikace požadavků.

Předmět implementace uživatelského rozhraní spočívá v inicializaci a nastavení adresování jednotlivých komponent, předávání a manipulace s daty ze strany uživatele i systému. Samozřejmostí je také formátování a stylizace webu pomocí kaskádových stylů.

Implementace začíná vytvořením instance Vue, která je navázána na html element s id app. Element je vnořen do souboru typu blade.php (souboru šablonovacího systému Laravelu). Tímto postupem začleníme a inicializujeme instanci frameworku Vue do kostry projektu frameworku Laravel.

```
import Vue from "vue";

new Vue({
  el: '#app',
  router,
  apolloProvider,
});
```

Na Vue instanci je možné navázat další pluginy, např. (router, apolloProvider, ..). Současně je na instanci vhodné navázat globální Vue komponenty systému. Mezi globální komponenty, jež jsou součástí našeho systému patří například navigační menu a záhlaví webu. Komponenta je uchovávána v jiném souboru než instance Vue aplikace.

```
Vue.component('intro', require('./components/Intro.vue').default);
```

Jedním z dalších použitých pluginů je například plugin V-Calendar. Kalendář umožňuje interaktivní výběr data či plánování událostí. V našem systému slouží kalendář pro zřízení rezervací a také pro plánování akcí podniku. Online objednávky budou fungovat pouze v rámci aktuálního dne. Kalendář lze pozorovat na obrázku 3.7.

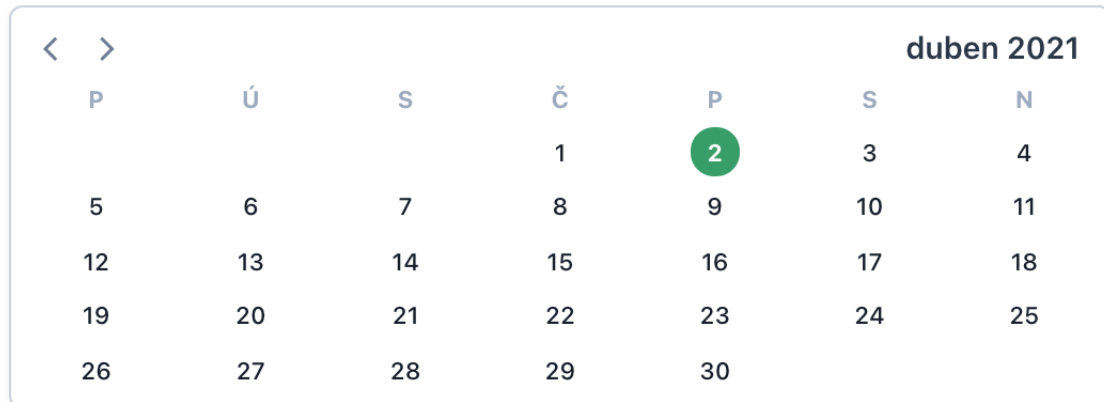
Při načtení komponenty rezervace dojde v životním cyklu instance Vue k invokaci metody `mounted`. V metodě vrátíme instanci Vue proměnou s datem instancí a přiřazením objektu `Date`.

```
mounted()
{
    return this.selectedDate = new Date()
}}
```

Na základě Vue direktiva `v-model` (použitého na komponentu pluginu `V-Calendar`) jsme schopni navázat tzv. *two way bindings* komunikaci. Přínos zmíněného direktiva spočívá ve schopnosti předávat komponentě data jak formou pasivní (v případě inicializace a přiřazení dnešního data), tak formou aktivní – uživatel zaklikne v komponentě kalendáře konkrétní den. Proměnná `selectedDate` obsahuje datum dle daného případu, díky čemuž jsme schopni s touto informací dále pracovat (například odeslat požadavek na rezervaci).

Rezervujte si u nás místo!

Bude-li to možné, rádi vám rezervujeme místo dle vašeho výběru.



<	>	duben 2021						
P	Ú	S	Č	P	S	N		
			1	2	3	4		
5	6	7	8	9	10	11		
12	13	14	15	16	17	18		
19	20	21	22	23	24	25		
26	27	28	29	30				

Obr. 3.7: Ukázka kalendáře. (Zdroj: vlastní zpracování, 2021)

Direktivum `v-model` je obdobou direktiva `v-bind`. Rozdíl spočívá v tom, že `v-bind` je určeno pro *one way bindings* komunikaci. Obě zmíněná direktiva budou v aplikaci spojena s předáváním dat na základě vyplňování vstupů, formulářů a interakce uživatelů se systémem.

Princip a příklad implementovaných odesílání dotazů skrze uživatelské rozhraní je popsán v sekci **3.10**, konkrétně v části autentifikace a autorizaci na straně uživatelského rozhraní.

Klíčovou, oprávněnou obavou zadavatele je bezpečnost systému. Proto je vhodné navrhnout proces autentifikace, autorizace a validace. Jmenované procesy jsou úzce spojeny v rámci uživatelského rozhraní aplikace s datovou částí aplikace. Zmíněným procesům je věnována pozornost v následující sekci.

3.9 Datová implementace

Datová implementace vychází z datového návrhu, tedy z Entity-relationship diagramu viz obrázek **3.2**. Je potřeba brát v úvahu entity článek, rezervace a menu.

Prvním krokem datové implementace je konfigurace MySQL databáze. V lokálním prostředí webového serveru je nejprve nutné databázi vytvořit. V laravel projektu se konfiguruje příslušná IP adresa, port, název a přihlašovací údaje.

Počáteční potřebné modely budou tedy správce, zaměstnanec, zákazník, objednávka, směna, rezervace, jídlo, menu a článek. Modely vytvoříme v Laravelu prostřednictvím PHP nástroje artisan. Předáním parametrů `make:model NAME -m`, kde NAME je název modelu, postupně vytvoříme všechny modely. V případě názvu modelů se typicky jedná o anglické názvy. Díky přepínači `-m` vzniknou s vytvořenými modely také kostry migrací.

Samotné tabulky je třeba implementovat pomocí tzv. migrací. Migrace jsou metody, které vytvoří tabulky s požadovanými atributy a sloupečky. Migrace je důležité nahrávat v pořadí tvorby tabulek a databáze z prostého důvodu – nelze aplikovat vazby na neexistující tabulky. Tabulky budeme migrovat do databáze skrze nástroj artisan, konkrétně parametrem `migrate`.

Příkladem tvorby tabulky je tvorba tabulky zákazníka. Obdobně dochází k vytvoření ostatních modelů a entit systému. Nezbytná je také deklarace kardinalit vazeb mezi jednotlivými modely.

Začněme kupříkladu modelem zákazník a modelem objednávka. Zákazník může mít více objednávek. Ve třídě zákazníka deklarujeme kardinalitu vazeb skrze invokaci metody

```

Schema::create('customer', function (Blueprint $table)
{
    $table->id();
    $table->email();
    $table->foreignId('order_id')->constrained('order');
})

```

orders(), jejíž návratová hodnota bude objekt typu HasMany. Každý záznam tabulky objednávka bude obsahovat customer_id, které nese identifikační číslo vlastníka objednávky.

```

class Customer extends Model
{
    public function orders(): hasMany
    {
        return $this->hasMany(Order::class, 'customer_id');
    }
}

```

Opačným případem je vazba z modelu objednávky na model zákazníka. V takovém případě deklarujeme kardinalitu ve třídě objednávky pomocí invokace metody customer(), která vrací objekt typu BelongsTo. Tělo metody popisuje závislost objednávky, která patří právě jednomu konkrétnímu zákazníkovi třídy Customer, jehož identifikátor je roven hodnotě cizího klíče customer_id.

```

class Order extends Model
{
    public function customer(): BelongsTo
    {
        return $this->belongsTo(Customer::class, 'customer_id');
    }
}

```

Konfigurace GraphQL schématu

Pro vytvoření modelů je nutné vytvořit a nastavit také Lighthouse schéma GraphQL. Schéma bude obsahovat GraphQL typy, dotazy a mutace.

Mezi typy patří výše uvedené modely. Každý typ bude obsahovat příslušné atributy společně s jejich datovými typy, současně typ obsahuje navázané relace s jejich typy.

Provádění mutací

Každou změnu v databázi je třeba provádět skrze mutace. Mutace budou korespondovat typem s navrženými entitami. Pro práci s přístupovými daty zaměstnance implementujeme následující tři mutace. Naopak pro čtení dat zaměstnance jsou využity dotazy, viz teoretická část, sekce **1.10**.

```
type Mutation {  
  newEmployee(name: String!, email: String!, password: String!): User  
  updateEmployee(id: ID, email: String, password: String): User  
  deleteEmployee(id: ID): User  
}
```

3.10 Bezpečnost

Sekce popisuje návrh procesů autentifikace, autorizace a validace ze strany klienta a uživatelského rozhraní a strany serveru.

Autentifikace

Kvůli schopnosti systému rozpoznat, zda jedná s anonymním, nebo naopak autorizovaným uživatelem je třeba pracovat s informacemi o autorizaci formou cookies.

V případě našeho systému se bude jednat o autentifikaci založenou na principu sezení resp. `session`, z důvodu užití vlastního GraphQL API přímo uvnitř single page aplikace. Alternativní možností je autentifikace za pomoci posílání tokenů, avšak ta je vhodnější pro větší, komplexnější aplikace. Session budou uchovávané v souborech, doba expirace je standartně 2 hodiny. Autentifikace pracuje formou stavů, stav je udržován na straně serveru.

Autentifikace je proces prostupující od vrstvy uživatelského rozhraní až k databázi. Je nutné zavést CSRF (Cross-site request forgery) ochranu. Jedná se o prostředek ochrany proti neautorizovaným příkazům neautentifikovaných uživatelů. Kvůli zvolenému přístupu posílání dotazů prostřednictvím frameworku Apollo a javascriptu, prohlížeč neposílá cookies automaticky. Proto bude třeba CSRF token, (který je přiřazen každé uživatelské session), přidat manuálně. Hodnota CSRF tokenu musí být přidána do hlavičky

každého zaslání dotazu GraphQL. Současně je třeba v každém dotazu povolit zaslání pověření (credentials jako je cookie, apod.).

V dalším kroku bude provedena implementace Lighthouse schématu GraphQL mutací a jejich pravidel. Na „backendové“ části se o komunikaci mezi GraphQL mutacemi a databází postará Laravel a jeho nástroje. Prostřednictvím zaslání dotazů skrze framework Apollo a javascriptový framework Vue jsou data předávána a validována na „frontendové“ části aplikace.

Po úspěšné autentifikaci je uživateli udělen přístup do systému. Následný proces autorizace se postará o přidělení oprávnění ke konkrétním komponentám.

Autorizace

Aby nedocházelo k neoprávněnému zásahu do části systému a jeho dat, je nutné implementovat ochranu, která tento scénář pokryje.

Při provádění GraphQL mutací umožňuje Laravel aplikovat systém ochrany (policy) na konkrétní datové modely. Implementujeme jakési pojistky, které budou vyhovovat schématu systému. Je třeba implementovat rozdílné pojistky pro zaměstnance, zákazníka a správce. Příkladem je pojistka pro mutaci, která se stará o smazání zaměstnance, či zákazníka. Pojistek bude implementováno několik, respektive je zapotřebí ošetřit každou mutaci, která vyžaduje konkrétní autorizaci, obvykle identifikačním číslem uživatele.

Autentifikace a autorizace na straně uživatelského rozhraní

Se zabezpečením možného, neoprávněného zásahu do dat je také nezbytné provést autentifikaci a autorizaci v rámci vizuální části aplikace. Předáním proměnné s informací o stavu autentifikace z datové části aplikace (skrze Apollo a Lighthouse GraphQL) do části vizuální budeme schopni autorizaci zařídit. Na základě podmíněného vykreslování (prostřednictvím stavu) lze rozhodnout o zobrazovaném obsahu.

Příkladem je implementace a invokace metody `authenticate()`, která slouží k ověření a identity uživatele při přihlašování. Kliknutím na tlačítko přihlásit metoda invokes mutaci `Login`. Pro ověření výsledku mutací je využíván javascriptový objekt typu

promise, na který lze navázat metodu catch. Metoda odchyťává a reaguje na chyby neboli případy nenaplněného slibu. Postup se pojí s odchyťáváním chyb v rámci mutací.

Validace

Validace bude prováděna jak na straně „frontendu“, tak na straně „backendu“. Uživatelé systému vidí upozornění formou validačních hlášek v případě nedodržení konstrukce vyplňování formulářů. Jedná se o formát emailové adresy, požadavku na heslo. Obdobně validovány budou také požadavky na objednávky a rezervace. U požadavku na čas rezervace je například prováděna validace času prostřednictvím následujícího regulárního výrazu, kde možnost rezervace stolu je pouze v časovém rozmezí 11:00-21:59 hod.

$$\wedge ([1]?[1-9]|2[0-1]):[0-5][0-9]\$$$

Velmi důležitá je také validace na straně serveru. Pokročilejší uživatelé jsou schopní javascriptové pravidla obejít a vnutit systému nepravé, škodlivé vstupy. Takovému chování je třeba zabránit. V případě útoků typu SQL Injection zasáhne opět Laravel, který většinu dotazů chrání automaticky. V rámci validace vstupů na straně serveru používáme pravidla GraphQL. V případě neplatných vstupů server GraphQL vrátí zprávu, kterou lze opět odchyťit na „frontendu“ aplikace.

```
register(  
  email: String! @rules(apply: ["required", "email", "unique:users"])  
  password: String! @rules(apply: ["required", "min:8"]) @hash  
  name: String! @rules(apply: ["required", "min:2"])  
): User @create
```

Kód zachycuje popis pravidel mutace pro registraci. Konkrétně se jedná o atribut emailu, hesla a jména. Spolu s pravidly je zde také direktivum @hash, které hashuje heslo před zapsáním do databáze.

3.11 Automatické posílání informací emailem

Dá se říct, že téměř všechny zmíněné procesy vyžadují podporu rozesílání emailových zpráv. Implementace spádá do oblasti Laravelu, kde je k dispozici knihovna SwiftMailer, disponující podporou posílání emailu pomocí protokolu SMTP apod. [52]

3.12 Návrh testování systému

Před samotným nasazením softwaru do produkce je třeba provést akceptační testování zákazníkem. Na základě reportů a dotazníků bude potřeba beta verzi softwaru upravit, či odladit dle zjištěných podrobností a kritérií.

Skupina testovacích uživatelů bude ve věkovém rozmezí od 15 do 60 let. Spadá zde majitel a zaměstnanci restaurace, kteří se zaměří také na přístup aktéra zákazníka. Dalšími testery budou konkrétní zákazníci. Ve skupině bude probíhat diskuze na témata související, například co je nutné změnit, případně vylepšit. Testovaná skupina nebude mít žádné předešlé znalosti o webové aplikaci.

Mezi klíčové segmenty prováděného testování patří

- *Prívětivost uživatelského rozhraní*
- *Schopnost porozumění systému a jeho funkcím*
- *Pocity z aplikace, očekávání a obavy*
- *Otázky, zda se vyskytly nesrovnalosti, či došlo k neočekávanému chování*

Zpětná vazba má za úkol odhalit pozitivní, ale také negativní připomínky a nápady testované skupiny směrem k webové aplikaci. Uživatele by neměli být zmateni či vystresováni, cílem je objevit podněty k případnému vylepšení. Díky těmto poznatkům je webová aplikace laděna k pocitům testerů, avšak zároveň musí splňovat požadavky zadavatele. Primárně důležitá je zpětná vazba ze strany správce a zákazníků, kterým je software také určen.

3.13 Produkce a hostování webové aplikace

Nasazení do produkce je plánováno koncem léta roku 2021, a to z důvodu náročnosti letní sezóny, tedy v případě, že budou gastronomické podniky otevřeny. Pro hostování webové aplikace byla vybrána hostingová služba společnosti Wedos. Společnost patří k jedné z nej kvalifikovanějších na českém trhu [53]. Na jejich serverech je možné nasazení webové aplikace ve formě PHP frameworku, a to Laravelu. Cenová nabídka je zahrnuta v podkapitole 3.16. Alternativní možností jsou také Amazon Web Services, které nabízejí širokou

škálu atraktivních provázaných služeb, nicméně Wedos je kvalitním a populárním řešením na českém trhu.

Hosting od Wedosu disponuje parametry jako je podpora aktuálních verzí PHP 8.0, 7.x v kombinaci s databázemi MySQL/MariaDB o velikosti 1GB. Současně je k dispozici mailing, HTTPS, ochrana proti útokům DDOS a IPS a v poslední řadě nepřetržitá podpora ze strany poskytovatele. Odesílání emailu systémem hostovaným na wedosu sebou nese určitá kapacitní pravidla proti zasílání nevyžádané pošty, nicméně podnik dokáže s přehledem podmínky dodržet [53]. Společnost nabízí také flexibilní možnosti navýšení či dodatečné dokoupení dalších služeb.

3.14 Použité nástroje

Grafické návrhy byly zpracovány pomocí software společnosti Adobe a softwaru Sketch. Webový HTTP server pro vývoj aplikace byl obstarán open-source softwarem XAMPP. Samotný vývoj systému je spjat s programovacími jazyky a nástroji popsány v první, teoretické kapitole.

Pro verzování aplikace je používán nástroj pro správu verzí GIT. Vzdálený repozitář byl nastaven na webové službě www.github.com. Nástroj a vytvořený repozitář umožňuje efektivní přecházení mezi verzemi a platformami při samotném vývoji aplikace.

3.15 Časové plány a milníky

Časový plán projektu lze dekomponovat do pěti milníků. Esenciální je analýza a specifikace požadavků. Po ujasnění podoby systému budou probíhat pravidelné schůzky se zadavatelem, a to formou metodiky SCRUM. Jakmile bude systém dokončen, bude spuštěno testování.

Milníky projektu

- *Specifikace, analýza požadavků a procesů*
- *Návrh a implementace*
- *Testování a zkušební provoz*

- *Zaškolení*
- *Spuštění produkčního prostředí*

Datum počátku projektu je stanoveno na květen 2021, s tímto datem jsou tedy spojeny první milníky. Datum fáze testování a zkušebního provozu je odhadováno na konec třetího kvartálu roku 2021.

Důležitý je očekávaný samotný časový harmonogram projektu. Pro zadavatele není datum spuštění systému striktně důležité, nicméně je nutné stanovit jisté milníky. Klíčová je tedy časová náročnost jednotlivých činností a to kvůli ceně projektu.

Činnost	Doba trvání
Vypracování podkladů	1 den
Zpracování specifikace	2 dny
Návrh procesů a toků dat, UML, ER diagramy	4 dny
Návrh uživatelského rozhraní	7 dní
Definice a inicializace databáze, serveru GraphQL	4 dny
Implementace uživatelského rozhraní a datová implementace	17 dní
Tvorba dokumentace	3 dny
Školení zaměstnanců	1 den
Testování a zkušební provoz	2+5 dní
Nasazení do produkce	1 den

Tab. 3.1: *Odhad doby trvání jednotlivých činností. (Zdroj: vlastní zpracování, 2021)*

Z tabulky 3.1 vyplývá, že odhadovaná doba projektu je cca 47 dní (vyjmu zkušebního provozu 42), přičemž časově a nákladově nejnáročnější část je fáze implementační. Na základě domluvy se zadavatelem je začátek samotné implementace datován v průběhu května tohoto roku, tzn. roku 2021.

3.16 Zhodnocení nákladů

Zhodnocení nákladů vychází z modelu Total Cost of Ownership, popsáno v sekci 1.8. Pro podnik není důležité ani nutné vlastnit žádný nový hardware, systém bude nasazen na serveru vybraného hostingu.

Úvodní konzultace, zpracování prvotních návrhů a sestavení kalkulace projektu je zdarma. Uvedené náklady se řadí do kategorie nákladů jednorázových.

Výčet jednorázových nákladů

- *Náklady spojené s detailní analýzou*
- *Návrh uživatelského rozhraní*
- *Implementace systému*
- *Náklady na spuštění*

Náklady spojené s detailní analýzou požadavků, návrhy uživatelského rozhraní a náklady na pořízení a implementaci systému byly kalkulovány na 60 000 Kč bez daně, což plní požadavek pořizovací ceny ze strany zadavatele.

Co se týče nákladů průběžných, nákladů spojených s provozem, umístěním a hostováním aplikace, ročně je nutné hradit poplatek za webhosting. Podnik již vlastní svou doménu. Pořizovací cena hostingu pro první rok vychází na 516 Kč bez daně. V případě potřeby změnit poskytovatele hostovací služby jsou náklady obdobné.

V případě nákladů dodatečného rozvoje (platební brána, propojení s nativní aplikací, systém odměn) má samozřejmě vliv množství a složitost požadovaných úprav. V takovém případě je nutno provést kalkulaci.

3.17 Zaškolení

S nasazením nového systému je nutné provést zaškolení personálu, který si musí osvojit způsoby fungování nejen procesů, ale celého systému. Během uvedení do provozu jsou plánovány pravidelné konzultace.

Samotný proces školení se uskuteční v rámci jednoho dne. Nejprve proběhne školení administrace s majitelem. Posléze se budou účastnit zaměstnanci, se kterými bude demonstrováno plánování a řízení směn, dále bude demonstrováno řízení elektronických objednávek a rezervací.

Ná závěr školení vyplní majitel společně se zaměstnanci dotazník náročnosti, spokojenosti a výhod implementovaného systému.

3.18 Potenciální přínosy informačního systému

Dle mého názoru je zřejmé, že návrh a nasazení nového informačního systému v podniku má přínosů hned několik. Ačkoli byla podporující fakta a přínosy několikrát zmíněna, závěrem diplomové práce je vhodné jejich následující shrnutí.

- *Objednávkový systém*
- *Systém správy směn*
- *Systém online rezervací*
- *Centralizovanost informací a pravidelná aktualizace*
- *Podpora v kritických situacích*
- *Zvýšení celkové kvality podniku (růst tržeb, efektivita)*

S možnými rozšířeními a modifikacemi systému lze dosáhnout dalších přínosů, funkcí a celkově lepších možností spojených se zlepšováním úrovně a kvality podniku.

ZÁVĚR

Diplomová práce nastínila řešení problému výběru a implementace informačního systému. Práce zahrnuje návrh a postup implementace požadovaných procesů podniku Restaurace & Pub LES. Jedná se o proces řízení elektronických objednávek, správu událostí, rezervací a směn. Systém vznikl také na podporu podniku v boji proti nepříznivým situacím, obdobným jako je například situace současná, již více než rok trvající pandemie Covid-19.

V rámci teoretické kapitoly byly popsány typické metodiky a postupy v oblasti vývoje informačních systémů. Současně práce prezentovala kategorie informačních systémů a dále se také věnovala principům vývoje konkrétního systému, a to prostřednictvím tvorby webové aplikace.

Analýza podniku a jeho současného stavu poukázala zejména na zvládnutí situace spojené s pandemií, také na potenciální faktory ovlivňující rozhodnutí zavedení vlastního informačního systému. Jednalo se o analýzu 7S, o Porterovu analýzu pěti sil, analýzu PESTLE a analýzu SWOT.

Na základě představení zmíněných analýz současné situace a stavu podniku byl navržen a zhodnocen proces změny pomocí Lewinova třífázového modelu. Práce popsala řešení výběru a způsobu implementace informačního systému. Koncept a návrhy se odrazily od specifikace konkrétních požadavků zadavatele, k jejichž interpretaci posloužily také grafické diagramy případů užití a datového modelu aplikace.

Na základě specifikace požadavků prostřednictvím zmíněných diagramů byla realizována grafická, responzivní interpretace návrhu systému, podle které byla započata implementace uživatelského rozhraní aplikace. Implementace datové části vychází rovněž ze specifikace potřeb klienta, návrhu datového modelu a požadavků realizovanými nad daty.

Během fáze návrhu a vývoje docházelo k pravidelnými konzultacím a specifikaci požadavků dle potřeby. Zpětná vazba formou testování prototypu přinášela pravidelné návrhy na změny. Prozatímní prototyp systému je testován prostřednictvím webových služeb Amazonu. Finální verze a nasazení systému do produkce je datováno na druhou polovinu letních prázdnin roku 2021.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] Česká televize *V Česku je přes sto tisíc nakažených, nemocnice od čtvrtka nesmí provádět plánované zákroky* [online]. 2020 [cit. 1. 1. 2021]. Dostupné z URL: <https://ct24.ceskatelevize.cz/domaci/3247608-rizikovy-index-se-vratil-na-paty-stupen-prisna-omezeni-v-cesku-plati-ctvrty-den>.
- [2] VLÁDA ČESKÉ REPUBLIKY *Vláda rozhodla o prodloužení nouzového stavu do 30. dubna, podpoří kulturní instituce více než miliardou korun* [online]. 2020 [cit. 1. 1. 2021]. Dostupné z URL: <https://www.vlada.cz/cz/media-centrum/aktualne/vlada-rozhodla-o-prodlouzeni-nouzoveho-stavu-do-30--dubna--podpori-kulturni-institute-vice-nez-miliardou-korun-180949/>.
- [3] OČENÁŠKOVÁ, Adéla. *U okének máme malé marže, lamentují restaurace. Přeživší jsou na pouhé třetině tržeb* [online]. 2020 [cit. 1. 1. 2021]. Dostupné z URL: <https://zpravy.aktualne.cz/ekonomika/gastropodniky-jsou-na-30-procentech-trzeb-z-unora-u-okenek-n/r~df2a31bc24dd11eb9c800cc47ab5f122/>.
- [4] Australian Bureau of Statistics. *Statistical Language - What are Data?* [online]. 2013 [cit. 19. 11. 2020]. Dostupné z URL: <https://www.abs.gov.au/websitedbs/D3310114.nsf/Home/Statistical+Language+-+what+are+data>.
- [5] MathsIsFun. *Analog and digital* [online]. 2020 [cit. 19. 11. 2020]. Dostupné z URL: <https://www.mathsisfun.com/data/analog-digital.html>.
- [6] SODOMKA, Petr, KLČOVÁ Hana. *Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd.*, Brno: Computer Press, 2010. s. 501. ISBN 978-80-251-2878-7.
- [7] GÁLA, Libor, POUR, Jan, ŠEDIVÁ Zuzana. *Podniková informatika. 2. přeprac. a aktualiz. vyd.*, Praha: Grada, 2009. s. 496. ISBN 978-80-247-2615-1.
- [8] MOLNÁR, Zdeněk. *Efektivnost informačních systémů. 2. rozš. vyd.*, Praha: Ikar, 2000. s. 178. ISBN 80-247-0087-5.

- [9] BASL, Josef, BLAŽÍČEK, Roman. *Podnikové informační systémy: podnik v informační společnosti. 3. aktualiz. a dopl. vyd.*, Praha: Grada, 2012. s. 323. ISBN 978-80-247-4307-3.
- [10] RÁBOVÁ, Zdeňka, JANOUŠEK, Vladimír, PERINGER, Petr, ZENDULKA, Jaroslav a ČEŠKA Milan. *Modelování a simulace. 3. přepr.vyd.*, Brno: VUT, 1992. s. 227. ISBN 80-214-0480-9.
- [11] VALACICH, Joseph a SCHNEIDER Christopher. *Information Systems Today - Managing in the Digital World, fourth edition.*, Prentice-Hall, 2010. s. 624. ISBN 978-0136078401.
- [12] DANEL, Roman. *Informační systémy - elektronická skripta* [online]. 2011 [cit. 20. 11. 2020]. Dostupné z URL:
https://homel.vsb.cz/~dan11/rd_is_skripta.htm.
- [13] POLÁK, Jiří, MERUNKA, Votěch a CARDA, Antonín. *Umění systémového návrhu.*, Praha: Grada, 2003. s. 195. ISBN 80-247-0424-2.
- [14] ŠARMANOVÁ, Jana. *Databázové a informační systémy* [online]. 2007 [cit. 21. 11. 2020]. Dostupné z URL:
<http://www.elearn.vsb.cz/archivcd/FEI/DAIS/DAIS.pdf>.
- [15] SCHWALBE, Kathy. *Information Technology Project Management.*, Boston, MA: Cengage Learning, 2016. s. 672. ISBN 978-1-285-45234-0.
- [16] GALLAGHER, Aiden, DUNLEAVY, Jack a REEVES, Peter. *The Waterfall Model: Advantages, disadvantages, and when you should use it* [online]. 2019 [cit. 29. 11. 2020]. Dostupné z URL:
<https://developer.ibm.com/technologies/devops/articles/waterfall-model-advantages-disadvantages/>.
- [17] MARTINŮ, Jiří a ČERMÁK, Petr. *METODIKY VÝVOJE SOFTWARE* [online]. 2018 [cit. 29. 11. 2020]. Dostupné z URL:
https://dl1.cuni.cz/pluginfile.php/886974/mod_resource/content/1/Metodiky-v%C3%BDvoje-software-studijn%C3%AD-text.pdf.

- [18] Ministerstvo vnitra ČR. *Metodika výpočtu TCO ICT služeb veřejné správy* [online]. 2016 [cit. 21. 1. 2021]. Dostupné z URL:
<https://www.mvcr.cz/soubor/metodika-tco-ict-sluzeb-vs-pdf.aspx>.
- [19] JAZAYERI, Mehdi. *Some Trends in Web Application Development* [online]. 2007 [cit. 21. 11. 2020]. Dostupné z URL:
https://www.researchgate.net/publication/4250861_Some_Trends_in_Web_Application_Development.
- [20] Nordic APIs AB. *Ultimate Guide To 9 Common HTTP Methods, Documentation* [online]. 2020 [cit. 29. 11. 2020]. Dostupné z URL:
<https://nordicapis.com/ultimate-guide-to-all-9-standard-http-methods/>.
- [21] AltexSoft. *What is API: Definition, Types, Specifications, Documentation* [online]. 2019 [cit. 22. 11. 2020]. Dostupné z URL:
<https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>.
- [22] ALTVATER, Alexandra. *What are CRUD Operations: How CRUD Operations Work, Examples, Tutorials & More* [online]. 2017 [cit. 18. 12. 2020]. Dostupné z URL:
<https://stackify.com/what-are-crud-operations/>.
- [23] FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures* [online]. 2000 [cit. 23. 11. 2020]. Dostupné z URL:
<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
- [24] Restfulapi.net. *What is REST* [online]. 2018 [cit. 23. 11. 2020]. Dostupné z URL:
<https://restfulapi.net/>.
- [25] GraphQL. *A query language for your API* [online]. 2020 [cit. 25. 11. 2020]. Dostupné z URL:
<https://graphql.org/>.
- [26] JAMAN, Shifat. *Everything you need to know about “Single-page-application”* [online]. 2019 [cit. 19. 11. 2020]. Dostupné z URL:

- <https://medium.com/@shifat.jaman/single-page-application-everything-you-need-to-know-6f00d87e5130>.
- [27] ZENDULKA, Jaroslav a RUDOLFOVÁ, Ivana. *Databázové systémy. IDS. Studijní opora.*, FIT VUT v Brně. 2008. s. 217.
- [28] ORACLE. *Co je relační databáze?* [online]. 2020 [cit. 16. 12. 2020]. Dostupné z URL:
<https://www.oracle.com/cz/database/what-is-a-relational-database>.
- [29] JAMAL, Madiha. *UML Diagrams: What you need to know?"* [online]. 2019 [cit. 27. 11. 2020]. Dostupné z URL:
<https://medium.com/datadriveninvestor/uml-diagrams-what-you-need-to-know-d07d4101e7c4>.
- [30] StatCounter. *Desktop vs Mobile Market Share Worldwide* [online]. 2020 [cit. 27. 11. 2020]. Dostupné z URL:
<https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#monthly-201010-202010>.
- [31] HO TRAN, Tony. *What does mobile-first design mean for digital designers?* [online]. 2019 [cit. 20. 11. 2020]. Dostupné z URL:
<https://www.invisionapp.com/inside-design/mobile-first-design/>.
- [32] Bootstrap. *Responsive breakpoints* [online]. 2020 [cit. 27. 11. 2020]. Dostupné z URL:
<https://getbootstrap.com/docs/4.0/layout/overview/>.
- [33] State of JavaScript. *Front End Frameworks* [online]. 2019 [cit. 30. 11. 2020]. Dostupné z URL:
<https://2019.stateofjs.com/front-end-frameworks/>.
- [34] Vuejs. *What is Vuex?* [online]. 2020 [cit. 30. 11. 2020]. Dostupné z URL:
<https://vuex.vuejs.org/#what-is-a-state-management-pattern>.
- [35] BHATT, Kiran. *Vue As Frontend And Laravel As Backend: A Deadly Combination To Build A Perfect Web App* [online]. 2020 [cit. 30. 11. 2020]. Dostupné z URL:

- <https://medium.com/quick-code/vue-as-frontend-and-laravel-as-backend-a-deadly-combination-to-build-a-perfect-web-app-be2a47c473cb>.
- [36] ROSE, Scarlett. *Laravel and Vue.js: Why Is This Couple Getting Popular?* [online]. 2019 [cit. 30. 11. 2020]. Dostupné z URL:
<https://towardsdatascience.com/laravel-and-vue-c30770f1e88>.
- [37] SQLite. *MySQL 8.0 Reference Manual* [online]. 2021 [cit. 28. 1. 2021]. Dostupné z URL:
<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>.
- [38] Ministerstvo spravedlnosti České republiky - Portál justice. *Veřejný rejstřík a sbírka listin*. [online]. 2020 [cit. 18. 12. 2020]. Dostupné z URL:
<https://or.justice.cz/ias/ui/rejstrik-firma.vysledky?subjektId=1004355&typ=UPLNY>.
- [39] LESPUB. *Restaurace & Pub LES* [online]. 2020 [cit. 27. 12. 2020]. Dostupné z URL:
<https://www.lespub.cz/>.
- [40] JUREVICIUS, Ovidijus. *McKinsey 7s Model* [online]. 2013 [cit. 2. 2. 2021]. Dostupné z URL:
<https://strategicmanagementinsight.com/tools/mckinsey-7s-model-framework.html>
- [41] HORÁKOVÁ, B. Situace podniku. *Informace o firmě* [ústní sdělení]. Restaurace & Pub LES. Jurkovičova 244/1, Brno, 2021.
- [42] ManagementMania.com. *PESTLE analýza* [online]. 2015 [cit. 2. 2. 2021]. Dostupné z URL:
<https://managementmania.com/cs/pestle-analyza>.
- [43] Ministerstvo zahraničních věcí České republiky *Nouzový stav v ČR* [online]. 2021 [cit. 25. 3. 2021]. Dostupné z URL:
https://www.mzv.cz/moscow/cz/viza_a_konzularni_informace/covid_19/nouzovy_stav_v_cr.html.
- [44] Eurostat. *Annual inflation stable at -0.3% in the euro area* [online]. 2021 [cit. 26. 1. 2021]. Dostupné z URL:

<https://ec.europa.eu/eurostat/documents/2995521/11562903/2-20012021-AP-EN.pdf/af0bd15f-2231-0a76-bc38-683e9b2faded>.

- [45] GOLKA, Petr. *Česko má nejvyšší inflaci za 8 let a třetí nejvyšší v EU* [online]. 2021 [cit. 26. 1. 2021]. Dostupné z URL:
<https://www.kurzy.cz/zpravy/575535-cesko-ma-nejvyssi-inflaci-za-8-let-a-treti-nejvyssi-v-eu/>.
- [46] Wikipedia. *Lesná (Brno)* [online]. 2011 [cit. 2. 2. 2021]. Dostupné z URL:
[https://cs.wikipedia.org/wiki/Lesn%C3%A1_\(Brno\)](https://cs.wikipedia.org/wiki/Lesn%C3%A1_(Brno)).
- [47] STŘELEČEK, Jiří. *PORTERŮV MODEL KONKURENČNÍCH SIL* [online]. 2012 [cit. 2. 2. 2021]. Dostupné z URL:
<https://www.vlastnicesta.cz/metody/porteruv-model-konkurencnich-sil-1/>.
- [48] Aktuálně.cz. *Konec v gastru nevyloučil ani Pohlreich. Krachují i jiná známá jména, ukazuje přehled* [online]. 2020 [cit. 2. 2. 2021]. Dostupné z URL:
<https://zpravy.aktualne.cz/finance/nakupovani/restaurace-koronavirus/r~fe815b4c0d2911eb842f0cc47ab5f122/>.
- [49] Aktuálně.cz. *Byznys s rozvozem jídla má zlaté časy. Třetinu jim dáme na provizi, nařikají podniky* [online]. 2020 [cit. 2. 2. 2021]. Dostupné z URL:
<https://zpravy.aktualne.cz/ekonomika/rozvozy-restaurace-koronavirus/r~4ddb5f5c891e11eab115ac1f6b220ee8/>.
- [50] OTÁHAL, Martin. *Co to je SWOT analýza? A k čemu slouží?* [online]. 2012 [cit. 18. 12. 2020]. Dostupné z URL:
<https://mladypodnikatel.cz/co-to-je-swot-analyza-t2797>.
- [51] VACULÍK, Josef. *Řízení změn I. Díl, Výbrané kapitoly – základy a postupy. 1. vyd.*, Pardubice: Univerzita Pardubice, 2006. s. 141. ISBN 80-7194-833-0.
- [52] Laravel. *Mail - Laravel* [online]. 2019 [cit. 20. 2. 2021]. Dostupné z URL:
<https://laravel.com/docs/8.x/mail>.

[53] WEDOS *WEDOS.cz - WEBHOSTING - DOMÉNY - SERVERY* [online]. 2021
[cit. 18. 12. 2020]. Dostupné z URL:
<https://www.wedos.cz/>.

SEZNAM OBRÁZKŮ

1.1	Schéma systému podniku	15
1.2	Aktéři informačního systému	20
1.3	Životní cyklus vodopádového a spirálového modelu	22
1.4	Komunikace informačního systému	25
1.5	Kompozice webové aplikace dle programovacích jazyků	28
1.6	Diagram případů užití	30
1.7	Entity-Relationship Diagram	31
1.8	Srovnání využití internetu na mobilních a stolních zařízeních	32
1.9	Implementace responsibility aplikace vůči variabilní šířce zařízení	33
1.10	Srovnání popularity JavaScriptových frameworků	35
1.11	Koncept nástroje Vuex	36
1.12	Továrna a nástroj Faker	37
2.1	Logo Restaurace & Pub Les	39
2.2	Míra inflace EU za rok 2020	45
3.1	Návrh případů užití pomocí jazyka UML	57
3.2	Návrh datového modelu pomocí ER diagramu	58
3.3	Vizualizace aplikace pro mobilní zařízení	61
3.4	Návrh úvodní stránky a jídelního lístku (mobilní zařízení)	62
3.5	Návrh profilu (stolní zařízení)	63
3.6	Návrh přehledu objednávek (stolní zařízení)	63
3.7	Ukázka kalendáře	65