

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

# NÁVRH, TVORBA A IMPLEMENTACE SOFTWAREVÉ APLIKACE VE FIREMNÍM PROSTŘEDÍ

DESIGN, CREATION AND IMPLEMENTATION OF SOFTWARE APPLICATIONS IN THE CORPORATE ENVIRONMENT

## DIPLOMOVÁ PRÁCE

MASTER'S THESIS

## AUTOR PRÁCE

AUTHOR

Bc. Lukáš Kulda

## VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lukáš Novák, Ph.D.

BRNO 2022

# Zadání diplomové práce

Ústav: Ústav informatiky  
Student: **Bc. Lukáš Kulda**  
Vedoucí práce: **Ing. Lukáš Novák, Ph.D.**  
Akademický rok: 2021/22  
Studijní program: Informační management

Garant studijního programu 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, tvorba a implementace softwarové aplikace ve firemním prostředí**

### **Charakteristika problematiky úkolu:**

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

### **Cíle, kterých má být dosaženo:**

Cílem práce je analyzovat, navrhnout a implementovat webovou aplikaci do firemního prostředí.

**Základní literární prameny:**

GÁLA, L., J. POUR a Z. ŠEDIVÁ. Podniková informatika. 2. přeprac. a aktualiz. vyd. Praha: Grada Publishing, 2009. ISBN 978-80-247-2615-1.

HARDCASTLE, E. Business Information Systems. Ventus Publishing ApS, 2008. ISBN 978-87-7681-463-2.

PRETTYMAN, S. Learn PHP 7: object oriented modular programming using HTML5, CSS3, Javascript, XML, JSON, and MYSQL. Apress, 2015. ISBN 978-1-4842-1730-6.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. 2. vyd. Brno: Computer Press, 2000. ISBN 978-80-251-2878-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně dne 28.2.2022

L. S.

---

doc. Ing. Miloš Koch, CSc.  
garant

---

doc. Ing. Vojtěch Bartoš, Ph.D.  
děkan

## Abstrakt

Tato diplomová práce se zabývá návrhem a implementací webové aplikace pro řízení zásob v zemědělské společnosti. Výsledkem této práce je aplikace, která má za úkol zlepšit nedostatky společnosti právě v oblasti řízení zásob. První důležitou kapitolou jsou teoretická východiska, o které se práce opírá. Další hlavní kapitolou je analýza současné situace společnosti, na jejímž základě je odůvodněna potřeba změny. Hned poté následuje kapitola popisující zavedenou změnu, implementaci aplikace a její přínos pro společnost.

## Abstract

This diploma thesis deals with the design and implementation of a web application for inventory management in an agricultural company. This work is an application that aims to improve the shortcomings of the company in the field of inventory management. The first important chapter is the theoretical basis on which the work is based. The next main chapter is the analysis of the current situation of the company which justifies the needed change. This is immediately followed by a chapter describing the introduced change, the implementation of the application and its benefits for the company.

## Klíčová slova

informační systém, skladový systém, webová aplikace, databáze, ASP.NET Core, C#, React, PostgreSQL, vývoj, SLEPTE, Porter, 7S, SWOT, Lewinův model, analýza rizik

## Keywords

information system, storage system, web application, database, ASP.NET Core, C#, React, PostgreSQL, development, SLEPTE, Porter, 7S, SWOT, Lewin's model, risk analysis

## Citace

KULDA, Lukáš. *Návrh, tvorba, a implementace softwarové aplikace ve firemním prostředí* [online]. Brno, 2022 [cit. 2022-05-07]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/141012>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská. Vedoucí práce Ing. Lukáš Novák, Ph.D.

# Návrh, tvorba, a implementace softwarové aplikace ve firemním prostředí

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Lukáše Nováka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Lukáš Kulda  
8. května 2022

## Poděkování

Tímto bych chtěl poděkovat rodině a všem blízkým, kteří mě podporovali při studiu. Především bych chtěl poděkovat vedoucímu mé diplomové práce Ing. Lukáši Novákovi Ph.D. za odbornou pomoc a cenné rady.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Vymezení problému a cíle práce</b>	<b>4</b>
2.1	Vymezení problému . . . . .	4
2.2	Cíle práce . . . . .	4
<b>3</b>	<b>Teoretická východiska práce</b>	<b>5</b>
3.1	Základní pojmy . . . . .	5
3.1.1	Data, informace, znalosti . . . . .	5
3.1.2	Informační systém . . . . .	6
3.1.3	Framework . . . . .	7
3.1.4	Databáze . . . . .	8
3.2	Principy a technologie webových aplikací . . . . .	10
3.2.1	Architektury webových aplikací . . . . .	10
3.2.2	Hypertext Transfer Protocol . . . . .	12
3.2.3	PostgreSQL . . . . .	13
3.2.4	JSON Web Token . . . . .	13
3.2.5	Back-end technologie . . . . .	16
3.2.6	Front-end technologie . . . . .	19
3.2.7	Další nástroje . . . . .	22
3.3	Analytické nástroje . . . . .	24
3.3.1	SLEPTE analýza . . . . .	24
3.3.2	Porterova analýza . . . . .	25
3.3.3	7S analýza . . . . .	27
3.3.4	SWOT analýza . . . . .	28
3.3.5	ZEFIS . . . . .	29
3.4	Řízení projektu . . . . .	30
3.4.1	Lewinův model . . . . .	30
3.4.2	Skórovací metoda . . . . .	31
3.4.3	Časová analýza PERT . . . . .	32
<b>4</b>	<b>Analýza problému a současné situace</b>	<b>33</b>
4.1	Představení společnosti . . . . .	33
4.1.1	Organizační struktura společnosti . . . . .	33
4.2	Současná situace a řešení . . . . .	34
4.2.1	Nedostatky současného řešení . . . . .	35
4.3	Strategické analýzy . . . . .	36
4.3.1	SLEPTE . . . . .	36

4.3.2	Porterova analýza . . . . .	37
4.3.3	7S analýza . . . . .	39
4.3.4	SWOT analýza . . . . .	40
4.3.5	Souhrn analýz . . . . .	41
<b>5</b>	<b>Vlastní návrhy řešení, přínosy návrhů řešení</b>	<b>42</b>
5.1	Popis navrhované změny . . . . .	42
5.1.1	Požadavky na implementaci . . . . .	42
5.2	Lewinův model změny . . . . .	43
5.2.1	Fáze rozmrazení . . . . .	43
5.2.2	Fáze přechodu a aplikace změny . . . . .	45
5.2.3	Fáze zmrazení . . . . .	46
5.3	Analýza rizik . . . . .	47
5.3.1	Identifikace rizik . . . . .	48
5.3.2	Ohodnocení rizik . . . . .	49
5.3.3	Návrh opatření . . . . .	50
5.3.4	Mapy rizik . . . . .	51
5.4	Časová analýza . . . . .	52
5.5	Finanční zhodnocení změny . . . . .	56
5.6	Vlastní řešení pomocí webové aplikace . . . . .	57
5.6.1	Databáze . . . . .	57
5.6.2	Back-end . . . . .	61
5.6.3	Front-end . . . . .	63
5.7	Audit výsledného řešení pomocí portálu ZEFIS . . . . .	69
5.7.1	Nedostatky, doporučení a řešení vybraného systému . . . . .	69
5.8	Budoucí vylepšení . . . . .	70
5.8.1	Mobilní aplikace . . . . .	70
5.8.2	Integrace s dalšími systémy . . . . .	70
5.9	Přínosy výsledného řešení . . . . .	70
<b>6</b>	<b>Závěr</b>	<b>72</b>
	<b>Literatura</b>	<b>73</b>

# Kapitola 1

## Úvod

Informační technologie se dostávají více a více do všech oblastí současného života. V posledních letech došlo k obrovskému nárůstu využití všemožných technologií a celkově je digitalizováno každé dnešní odvětví. Některá odvětví a jednotlivé společnosti již informační technologie využívají delší dobu, jiné s nimi přichází teprve do styku. Poptávka je tedy vysoká a všichni hledají způsoby, jak tyto technologie využít, aby byli schopni řešit různé problémy rychleji a efektivněji.

V současné době jsou požadovány variabilní řešení pomocí informačních technologií, jelikož někteří lidé operují v terénu, tak vyžadují flexibilnější řešení například pomocí mobilních aplikací. Často jsou také vyžadována řešení, která jsou dostupná na více platformách. Jde primárně o stanovení vhodného řešení ke konkrétnímu problému.

Jádrem této práce je zemědělská společnost, která potřebuje provést změnu v řízení skladových zásob. Cílem této práce je analýza aktuálního problému, současné situace společnosti, návrh a implementace aplikace, která zlepší nedostatky v oblasti řízení skladových zásob.

Kapitola 2 nejprve čtenáře seznámí s problémem, který bude předmětem řešení této práce. Zároveň jsou zde definovány cíle, kterých má být v rámci této práce dosaženo. Následující kapitola 3 obsahuje teoretická východiska práce, kde jsou popsány nástroje, možné postupy a technologie, na kterých je založena praktická část práce. V kapitole 4 je analyzováno prostředí společnosti pomocí postupů a nástrojů popsaných v kapitole 3. Samotné řešení, včetně jeho zhodnocení a přínosu, je rozebíráno v předposlední kapitole 5. Poslední kapitola 6 shrnuje dosažené výsledky této práce.

## Kapitola 2

# Vymezení problému a cíle práce

Tato kapitola v následující sekci stručně a výstižně rozebírá problém, se kterým se zvolená zemědělská společnost potýká. V sekci 2.2 jsou stanoveny cíle práce, kterých má být dosaženo.

### 2.1 Vymezení problému

V zemědělské oblasti existuje mnoho různorodých procesů. Je nutné se starat o pozemky, pole, louky, také o zvířata, ale i o stroje nebo různé zásoby. Zvolená zemědělská společnost potřebuje využívat mnoho různých skladů. Slouží ke skladování velkých množství rostlinných plodin, krmiv pro zvířata a další materiál. Sklady se nachází na různých místech v rámci různých středisek, kdy dochází k manipulaci s materiálem na denní bázi.

Vedení společnosti aktuálně není spokojeno s aktuálním řešením v řízení skladových zásob. Z toho důvodu bude v rámci této práce vytvořena responzivní webová aplikace. Mělo by se jednat o jednoduchou a intuitivní aplikaci, která by měla umožňovat základní operace související s manipulací skladových zásob primárně prostřednictvím mobilního telefonu. Každá operace bude zaznamenávána, takže bude dostupná kompletní historie operací konkrétního materiálu. Měla by poskytovat aktuální stav jednotlivých skladů. Historii transakcí by mělo být možné filtrovat, aby bylo možné vytvářet měsíční výkazy.

### 2.2 Cíle práce

Cílem práce je analýza prostředí společnosti, ze které by měl vzejít následný návrh a implementace aplikace. Přičemž by aplikace měla splňovat veškeré požadavky společnosti jako zadavatele. Také je cílem použít co nejadekvátnější nástroje pro návrh a implementaci aplikace. Výsledkem by měla být webová aplikace s responzivním uživatelským rozhraním, jednoduchým použitím a především by měla zefektivnit procesy spojené se skladováním materiálu.

## Kapitola 3

# Teoretická východiska práce

V této kapitole jsou popsány a vysvětleny veškeré pojmy, postupy, metody, nástroje a technologie, které jsou využity v rámci této práce pro dosažení stanovených cílů.

### 3.1 Základní pojmy

Tato sekce popisuje jednotlivé základní pojmy, které jsou používány napříč touto diplomovou prací.

#### 3.1.1 Data, informace, znalosti

Data reprezentují specifické vlastnosti objektů kolem nás, tedy v reálném světě. Jsou množinou popisující objekt bez kontextu [51]. Tedy bez širšího popisu sama o sobě nedávají smysl. Například následující věta „Průměrná teplota v Praze je v prosinci 5°C“ může být uložena takto [41]:

- Praha – hodnota atributu „město“
- Prosinec – hodnota atributu „měsíc“
- 5 – hodnota atributu „teplota“

Z hlediska práce s daty je možno rozlišovat následující druhy dat [41], [51]:

- **Strukturovaná data** – U daného objektu je možné přesně popsat jeho strukturu. Pro tento objekt existuje schéma, kterým jej lze popsat. Typickým příkladem, kde se strukturovaná data využívají, jsou relační databázové systémy. V relačních databázích se data běžně uchovávají v tabulkách. Příkladem strukturovaných dat může být například adresa trvalého bydliště.
- **Nestrukturovaná data** – U těchto dat nelze jednoduše popsat jejich strukturu. Může se jednat například o audiozáznamy, videozáznamy nebo toky bitů.
- **Semistrukturovaná data** – Jedná se o nestrukturovaná data, která obsahují alespoň částečně nějakou volnou strukturu.

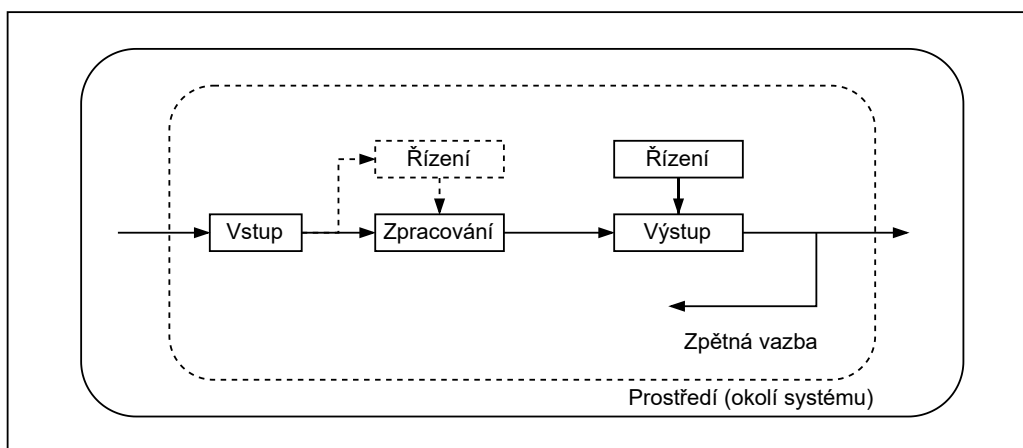
Pojem *informace* se používá v mnoha disciplínách a oborech. Z toho důvodu se vyskytuje i více různých definic tohoto pojmu. Informacemi se tedy myslí, že to jsou data zasazená do kontextu, tudíž jsou použitelná a srozumitelná [41]. Pokud se pro data vytvoří pojem (koncept) a zasadí se do kontextu, tak získáme informaci [51].

I přestože informace jsou užitečným zdrojem pro jedince i organizace, nelze však veškeré informace považovat za užitečné a použitelné. Dobré či zlé informace lze rozlišovat na základě toho, zda obsahují atributy, které jsou v daném kontextu relevantní a kvalitní. Informace musí být využívány ve správný čas, jelikož se jejich hodnota v čase mění. Tedy informace, která je dostupná a užitečná dnes, zítra již užitečná být nemusí [17]. Využíváním informací získáváme zkušenosti, a jsou tedy transformovány do znalosti [51].

### 3.1.2 Informační systém

Systém je množina prvků a vazeb mezi nimi. V oblasti informačních technologií je takový systém nazýván jako informační systém. Účelem informačního systému je vhodná interpretace informací, jejich zpracování a přenášení. Tento systém je tvořen lidmi, nástroji a metodami, které jsou seskupeny do následujících komponent, viz obrázek 3.1 [51]:

- **Vstup** – Všechny prvky, které zachycují informační a další vstupy, které jsou určeny pro další zpracování.
- **Zpracování** – Všechny prvky, které transformují vstupy do požadovaného výstupu.
- **Výstup** – Všechny prvky, které přenáší informační a další výstupy uživateli.



Obrázek 3.1: Komponenty informačního systému

Převzato [51]

### Podnikové procesy

„Proces je definován jako soubor vzájemně souvisejících nebo vzájemně působících činností, který přeměňuje vstupy na výstupy. Činnost využívají zdroje (lidí, nástrojů, materiálů, apod.). Proces může mít více vstupů a také více výstupů.“ [51].

Konkrétní proces je pak vždy spuštěn definovanou událostí. Taková událost může být například vstup informace, člověka, materiálu do podniku. Také časová událost, interní potřeba změny nebo výjimečný stav jako je například porucha [51].

### Podnikový informační systém

Podnikové informační systémy jsou systémy, které se používají v podnicích. Každý podnik využívá různé informační systémy dle potřeby. Tyto systémy podporují nebo přímo slouží k plánování, kontrole, předpovědi, koordinaci, rozhodování a k operačním aktivitám v rámci organizace [17].

Každý podnik má několik organizačních úrovní, přičemž každá tato úroveň potřebuje specifický druh informací, včetně jejich specifického způsobu zpracování [16].

Každý informační systém je ve výsledku jedinečný projekt, který se v rámci podniku zavádí. S tím se pojí i životní cyklus informačního systému. Ten se neustále zkracuje, a často se během implementace projektu mění nebo rozšiřuje i jeho zadání. To se většinou děje ve větších podnicích ve chvíli, kdy implementace trvá déle než rok. S tím mohou růst i vynaložené náklady na implementaci informačního systému [16].

#### 3.1.3 Framework

V současné době existuje velké množství programovacích jazyků, které se aktivně využívají pro řešení složitých a komplexních problémů. Každý jazyk má určitou míru složitosti a některé ponechávají programátorovi větší zodpovědnost, pokud programátor programuje například v jazyce C, musí se starat o správu paměti. Pokud by programátor zanedbal při psaní programu správu paměti, mohlo by docházet k častým chybám při běhu programu. Jiné jazyky jsou zase v tomto ohledu jednodušší, programátor tak nemusí psát konstrukce jazyka, kterými by se o paměť staral. Správa paměti je zde jen jako příklad, kterým se mnohé jazyky mohou lišit.

Nicméně i dnes pouze za pomoci programovacích jazyků a dostupných knihoven v rámci jejich balíčkových ekosystémů, by vývoj některých běžných standardních řešení trval poměrně dlouho.

Proto je dnes populární tzv. „Framework“ ve vývoji aplikací. Jedná se o sadu nástrojů v programování, pomocí kterých lze vytvářet dobře strukturovaný a spolehlivý software. V podstatě poskytuje hotové komponenty nebo řešení, které jsou přizpůsobeny za účelem urychlení vývoje softwaru [14].

Framework je definován principem „Inversion of Control“ (IoC)<sup>1</sup>, ale může obsahovat právě i knihovny. Při tradičním programování uživatelský kód volá do knihovny pro přístup k opakovaně použitelnému kódu. S IoC frameworkem volá vlastní části kódu v případě potřeby. Framework může zahrnovat podpůrné programy, kompilátory, knihovny kódů, sady nástrojů a rozhraní API pro vývoj softwaru a vytváření systémů. Open-source frameworky se neustále aktualizují a vylepšují. Účelem frameworku je usnadnit vývoj a poskytovat standardní funkce na nízké úrovni, aby vývojáři mohli zaměřit úsilí na prvky, které dělají projekt jedi-

<sup>1</sup><https://www.netinbag.com/cs/internet/what-is-inversion-of-control.html>

nečným. Použití vysoce kvalitní, předem prověřené funkčnosti zvyšuje spolehlivost softwaru, zrychluje dobu programování a zjednodušuje testování [14].

## Druhy Frameworků

Frameworky se používají už celkem dlouho, a tak jich existuje mnoho. Každý je vhodný pro řešení různých problémů, některé jsou více populární, některé méně. Kvalitní frameworky mají dobrou a bohatou dokumentaci, aktivní komunitu a mají stabilní verze. Mezi některé druhy frameworků patří následující [14]:

- **Webové frameworky** – Front-end frameworky většinou poskytují komponenty obsahující HTML, CSS a JavaScript.
  - Front-end
    1. Angular
    2. React JS
    3. Vue JS
  - Back-end
    1. Ruby on Rails
    2. PHP Frameworky (Laravel, CakePHP, ...)
    3. Django
    4. ASP.NET (Core)
- **Frameworky pro vývoj mobilních aplikací**
  1. React Native
  2. Flutter
  3. Xamarin

### 3.1.4 Databáze

Databáze je kolekce vzájemně souvisejících souborů dat [51]. Databáze může být sice něco co lze napsat na papír, například telefonní seznam by mohl být označen za databázi, avšak v dnešním pojetí se tento pojem pojí spíše s informačními technologiemi.

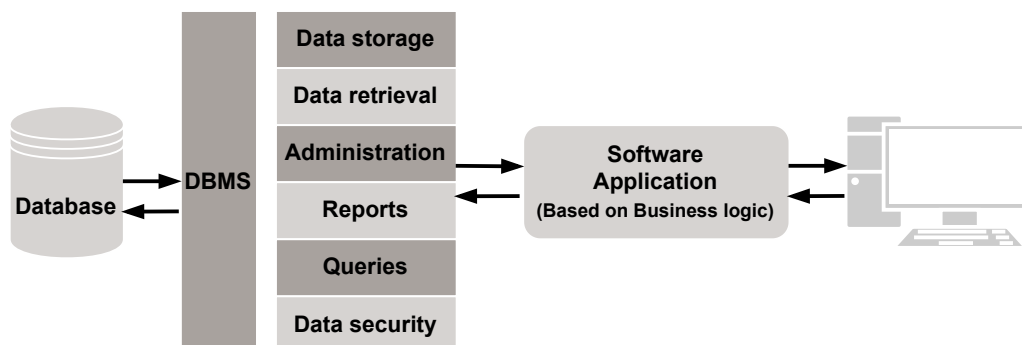
S databázemi se setkáváme prakticky na každém kroku. Jsou nedílnou součástí mnoha webových, mobilních aplikací nebo informačních systémů. Umožňují persistenci dat, tedy uložení dat i přes dobu životnosti aplikací [6].

#### Systém řízení báze dat

Systém řízení báze dat (SŘBD), anglicky Database Management System (DBMS) je aplikace, která interaguje s programovým jazykem a databází za účelem aktualizace, vkládání a mazání dat. K tomu používá jazyk SQL, který bude popsán později v této sekci (3.1.4) [38].

SŘBD jednoduše představuje logické rozhraní pro uživatele a aplikace. Existuje mnoho typů databází, přičemž každý tento typ má své výhody či omezení. Relační databáze se považuje za výchozí typ databáze v tomto paradigmatu. SŘBD poskytují přístup k databázím, které

spravují prostřednictvím různých prostředků včetně příkazového řádku, rozhraní API, programovacích knihoven a administrativních rozhraní. Prostřednictvím těchto kanálů lze data přijímat do systému, organizovat je podle potřeby a vracet podle požadavků [6].



Obrázek 3.2: Systém řízení báze dat  
Vlastní zpracování na základě [9].

## Typy databází

V průběhu let se vyvinulo mnoho typů databází vzhledem k tomu, že se v průběhu času mění požadavky na způsoby použití databází. Výběr typu databáze může mít zásadní vliv na to, jaké operace může aplikace využívající danou databázi provádět, jak budou konceptualizována data a funkce, které SŘBD nabízí. Různé typy databází a jejich příklady ukazuje následující seznam [8]:

- **Souborové** – CSV soubory
- **Hierarchické** – Souborové systémy, DNS, LDAP
- **Relační** – MySQL, MariaDB, PostgreSQL
- **NoSQL** – MongoDB, Redis
- **Dokumentové** – MongoDB

## Relační databáze

Relační databáze jsou v současnosti velmi používané. Databáze organizuje data prostřednictvím tabulek. Tabulka představuje schéma pro ukládání jednotlivých záznamů. Každý sloupec představuje atribut schématu a má konkrétní datový typ. Záznam je pak reprezentován řádkem v tabulce [8].

Relační databáze jsou nejstarším obecným typem databáze, který se dodnes široce používá. Ve skutečnosti relační databáze tvoří většinu databází, které se v současnosti používají. K propojování tabulek se používají cizí klíče. Relační databáze jsou vysoce výkonné databáze [8].

## SQL

SQL (Structured Query Language) je dotazovací jazyk, který umožňuje přístup a manipulaci s daty pomocí příkazů. Umožňuje filtrovat, agregovat, sumarizovat data, propojovat tabulky. Ukázka SQL query je ve výpisu 3.1 [8].

```
SELECT firstName
FROM users
WHERE age BETWEEN 18 AND 60;
```

Výpis 3.1: Ukázka SQL kódu

V rámci této query se vybere sloupec `firstName` z tabulky `users` a aplikuje se filtr, kde věk uživatele je mezi 18 až 60 lety.

## Migrace databáze

Migrace databáze znamená migraci schémat databáze. Jedná se o změnu, která upraví strukturu objektů v relační databázi. Tato změna může zahrnovat například přidání tabulek, sloupců, odebrání prvků, změnu relací, typů nebo omezení. K migracím existují i nástroje, které pomáhají migrace provést snáze a bezpečně [7].

## 3.2 Principy a technologie webových aplikací

V této sekci jsou popsány principy a technologie, které primárně souvisí s vývojem, provozem, údržbou a bezpečností webových aplikací.

### 3.2.1 Architektury webových aplikací

Architektur webových aplikací je několik a hraje zásadní roli při vytváření webových aplikací. Tato sekce pojednává o některých možných architekturách webových aplikací, jejich vrstev, komponent, modelů, typů a osvědčených postupů [49].

Webové aplikace fungují na principu klient-server. Kde klientská část představuje uživatele, prostřednictvím webového prohlížeče. Jedná se o tu část, kterou uživatel vidí na výstupu. Pro tuto část se používají front-end technologie, které budou popsány později v této sekci (3.2.6). Serverová část je nějaký výkonný počítač, který odpovídá na HTTP požadavky klienta. Oba komunikují přes počítačovou síť. Pro serverovou část se používají back-end technologie, viz sekce 3.2.5 [49].

Webové aplikace se běžně skládají z následujících vrstev [31]:

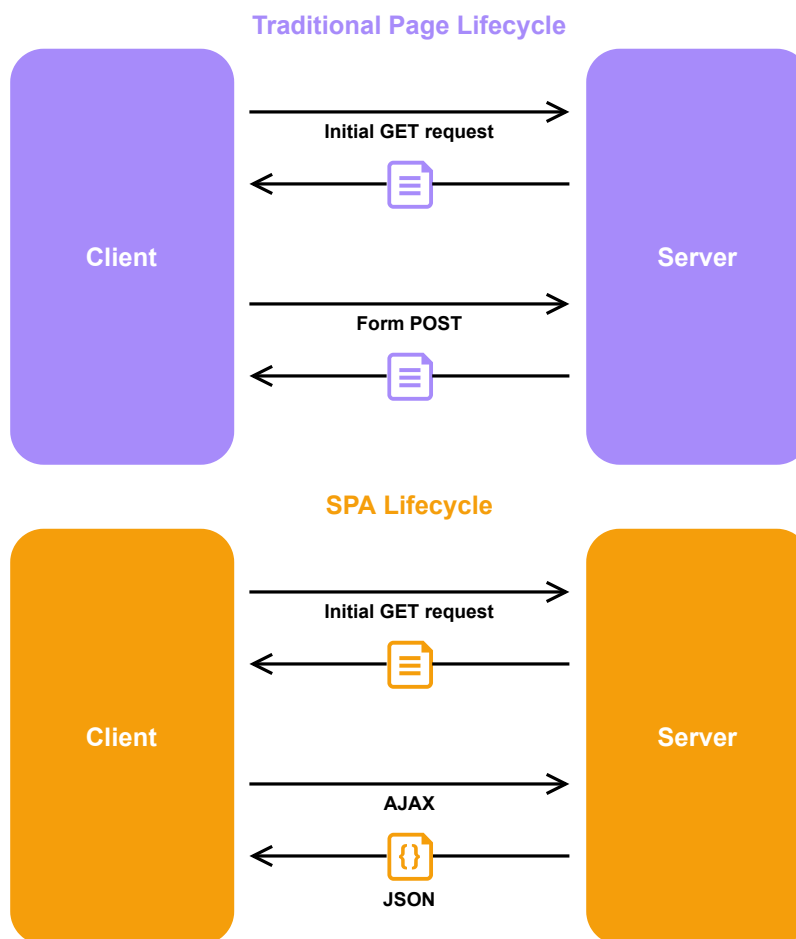
- **Prezentační vrstva** – Také nazývána jako vrstva uživatelského rozhraní. Uživatel tvoří požadavky skrze tuto vrstvu, jenž interaguje pouze s logickou vrstvou. S žádnou jinou vrstvou by neměla přijít do styku.
- **Logická vrstva** – Tato vrstva obsahuje samotné chování aplikace, které se odvíjí od toho, jak je samotná aplikace implementovaná. Zpracovává požadavky, které přichází z vyšší prezentační vrstvy. Komunikuje s datovou vrstvou, na které mnohdy závisí.

- **Datová vrstva** – Poskytuje přístup k perzistentním datům. Typicky se jedná o databázi.

Existuje několik typů architektur webových aplikací, jako například architektura postavená na tzv. „Microservices“, Serverless architektura, Progressive Web Applications nebo Single Page Applications (SPA). Jelikož pro tuto práci bude použita architektura SPA, tak ostatní zde nebudou podrobněji rozebrány [49].

### Single Page Application

Jak už název napovídá, jedná se o aplikaci, která má v podstatě všechny informace zabalené do jedné stránky. Oproti tradičním vícestránkovým webovým aplikacím dochází k načtení stránky pouze jednou, klient pak posílá požadavky na server, kdy požadovaná data dojde zpět a stránka se dynamicky aktualizuje. Nedochozí tak k přesunu například na jinou stránku, která by se musela celá načíst. Rozdíl v životních cyklech komunikace lze vidět na obrázku 3.3 [44].



Obrázek 3.3: Rozdíl mezi životními cykly komunikace tradičních webových aplikací a jednostránkových aplikací

Vlastní zpracování na základě [44].

Front-end obsahuje více logiky než tradiční vícestránkové aplikace. Z toho důvodu jsou webové stránky na straně klienta vyvíjeny pomocí frameworků JavaScriptu, jako je například React – ty jsou popsány v sekci 3.2.6.

Výhody a nevýhody SPA jsou následující [44]:

- **Výhody**

- **Rychlé načítání** – Jelikož server neposílá žádné další HTML nebo CSS soubory, tak aplikace bývají mnohem rychlejší než tradiční, kde dochází k načítání celé stránky při dalším požadavku.
- **Schopnost Cachování dat** – Umožňuje cachovat data lokálně, takže si je při prvním požadavku ze serveru může uložit, aplikace s nimi může tak nadále pracovat, v některých případech to umožňuje uživateli do určité míry pracovat offline.
- **Lepší User Experience** – Díky dostupným frameworkům a možnostem, které SPA nabízí, uživatelé nemusí dlouze čekat na načítání stránek. Dynamické načítání komponent, které mohou být i animovány, poskytují uživateli příjemnou práci s aplikací.
- **Rychlý vývoj** – Tím, že SPA jsou jakožto front-end část webové aplikace odděleny od back-end části, tak tato architektura umožňuje vyvíjet front-end nezávisle na back-end části.

- **Nevýhody**

- **Špatná optimalizace v rámci vyhledávačů** – Jelikož se obsah načítá dynamicky na základě požadavků klienta, tak vyhledávače, boti, crawlers nedokážou skenovat jednotlivé stránky.
- **Komplexnost** – Ve chvíli, kdy je aplikace velmi komplexní, její výkonnost může klesat, což by mohlo zhoršit bezproblémovou práci s aplikací.
- **Analytika** – Analytické nástroje mohou sledovat zobrazení stránky pouhým přidáním jediného kódu na stránku. Stránky v SPA jsou ale generovány dynamicky, logika proto musí být přidána do kódu, což může být problematické.

### 3.2.2 Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) je aplikační protokol, používá se pro přenos dat, a to jakýchkoliv, tedy dokumentů nebo i binárních. Funguje na princip klient-server. Kdy spojení je iniciováno ze strany klienta, běžně přes webový prohlížeč. Pro přenos dat využívá TCP na transportní vrstvě. Je to jednoduchý, bezstavový, pro člověka snadno čitelný protokol [19].

Ve chvíli, kdy klient chce navázat komunikaci se serverem, tak postupuje dle následujících kroků [19]:

1. **Otevření TCP spojení:** Pomocí TCP se pošle jeden nebo více požadavků, na které následně dostane odpověď. Klient může zahájit nové spojení, znovu použít stávající spojení nebo otevřít více spojení najednou.
2. **Poslání HTTP zprávy:** Použití protokolu HTTP, od verze HTTP/2 jsou zapouzdřeny do rámců a nejsou jednoduše člověkem čitelné.

### 3. Přechtení odpovědi ze serveru

### 4. Ukončení spojení nebo znovupoužití pro další požadavky

Existují dva typy HTTP zpráv a to pro požadavek (výpis 3.2) a pro odpověď (výpis 3.3), přičemž každá má vlastní formu [19].

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

Výpis 3.2: HTTP požadavek

Požadavek se skládá z HTTP metody, která definuje operaci, jež chce klient provést. Například GET vyžádá data, POST umožňuje předat data. Dále URL serveru, verzi HTTP protokolu, hlavičky pro další informace pro server a tělo zprávy – to se používá například při metodě POST, kdy data jsou poslány v těle požadavku [19].

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
```

```
<!DOCTYPE html... (here come the 29769 bytes of the requested web page)
```

Výpis 3.3: HTTP odpověď

Odpověď obsahuje také verzi protokolu HTTP, stavový kód, který indikuje úspěšnost operace včetně jejího odůvodnění. Dále stavovou zprávu, hlavičky a případně opět tělo zprávy [19].

### 3.2.3 PostgreSQL

PostgreSQL je open-source relační databáze, která je výkonná, rozšiřuje SQL a poskytuje mnoho funkcí, které bezpečně ukládají a škálují komplikované databáze. Je dostupná pro platformy Windows, Linux a MacOS [37].

Je součástí mnoha knihoven populárních programovacích jazyků, umožňuje definování vlastních datových typů a funkcí [37].

### 3.2.4 JSON Web Token

Než bude vysvětlen pojem JSON Web Token, tak bude prvně vysvětlen pojem JSON. JavaScript Object Notation (JSON) formát pro výměnu dat. Je jednoduše čitelný pro běžného člověka, jednoduše se parsuje a generuje. Je to textový formát, který není závislý na jazyce [24].

JSON má podobu objektu, je to neuspořádaná množina párů název – hodnota. Objekt začíná levou složenou závorkou { a končí pravou složenou závorkou }. Za každým jménem následuje dvojtečka : a dvojice název – hodnota jsou odděleny čárkou ,, příklad objektu lze vidět ve výpise 3.4 [24].

```
{
  "user":
  {
    "firstName": "John",
    "lastName": "Doe",
    "role": "admin",
    "transactions":
    {
      "id": "1"
      "timestamp": "1650376959"
    }
  }
}
```

Výpis 3.4: Ukázka JSON objektu

JSON Web Token (JWT) je standard definovaný RFC 7519<sup>2</sup>. Definuje kompaktní a samostatný způsob pro bezpečný přenos informací mezi stranami jako JSON objekt. Tyto informace jsou důvěryhodné, jelikož jsou digitálně podepsány pomocí tzv. „secret“ (s **HMAC algoritmem**) nebo páru veřejného a soukromého klíče za použití **RSA** nebo **ECDSA** kryptografie [25].

Podepsané tokeny mohou ověřit integritu nároků v nich obsažených, zatímco zašifrované tokeny tyto nároky skryjí před ostatními stranami. Když jsou tokeny podepsány pomocí páru veřejného/soukromého klíče, podpis také potvrzuje, že pouze strana, která drží soukromý klíč, je tou, která jej podepsala [25].

Struktura tokenu se skládá ze tří částí [25]:

- **Hlavička** – Typicky obsahuje dvě části, nejprve typ tokenu a následně algoritmus použitý pro podpis.
- **Tělo** – Obsahuje nároky. Jsou 3 typy nároků. Tato část je kódována do Base64Url formy.
  - **Registrované nároky** – Předdefinované nároky, nejsou povinné, ale doporučené. Mohou to být například tyto nároky: **iss**(emitent), **exp**(expirace tokenu) a další.
  - **Věřejné nároky** – Definované těmi, kteří používají JWT. Jsou definované v registru<sup>3</sup> nebo musí být definované jako URI, aby nedošlo ke kolizím jmenného prostoru.
  - **Soukromé nároky** – Obsahuje vlastní nároky, které lze sdílet mezi stranami.

<sup>2</sup><https://datatracker.ietf.org/doc/html/rfc7519>

<sup>3</sup><https://www.iana.org/assignments/jwt/jwt.xhtml>

- **Podpis** – Podpis se používá k ověření toho, zda zpráva nebyla během cesty změněna. V případě užití tokenů podepsaných soukromým klíčem lze ověřit, že odesílatel je ten, za koho se vydává. K vytvoření podpisu se vezme zakódovaná hlavička, tělo, „secret“, algoritmus specifikovaný v hlavičce a následně dojde k podpisu.

Výsledkem je řetězec zakódovaný v `Base64Url`, který lze předávat například prostřednictvím HTTP hlavičky. Příklad tokenu lze vidět na obrázku 3.4 a jeho dekódovanou formu lze vidět na obrázku 3.5.

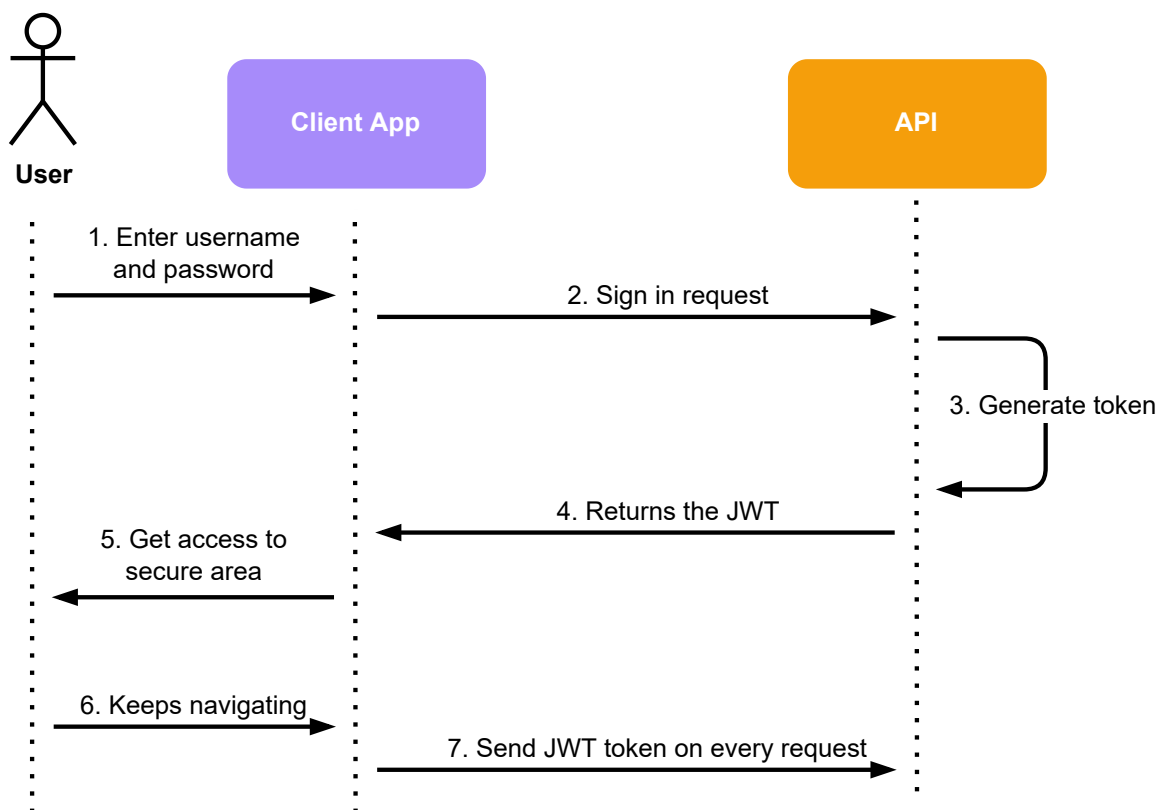
```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzkwMjQ.fKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Obrázek 3.4: JSON Web token

```
{
  "alg": "HS256",
  "typ": "JWT"
}.{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}.[Signature]
```

Obrázek 3.5: Dekódovaný JSON Web token

Autentizace uživatele funguje tak, že prvně zadá uživatelské jméno a heslo, odešle se požadavek na autorizační server. Ve chvíli kdy přihlášení proběhne úspěšně, tak autorizační server (emitent tokenu) pošle token zpět klientovi. Klientská aplikace pak musí posílat token při každém požadavku, pokud uživatel chce přistupovat k částem aplikace, které jsou chráněné [25]. Diagram tohoto procesu lze vidět na obrázku 3.6.



Obrázek 3.6: Diagram autentizace pomocí JWT  
Vlastní zpracování na základě [26].

### 3.2.5 Back-end technologie

Back-end představuje tu část aplikace, kterou uživatel nevidí, obsahuje veškerou logiku aplikace. V sekci 3.1.3 byly již zmíněny některé back-end technologie. V rámci této práce budou popsány použité technologie při implementaci což bude primárně jazyk C# a framework ASP.NET Core.

#### Jazyk C#

C# (vyslovováno jako „See Sharp“) je moderní, objektově orientovaný a typově bezpečný programovací jazyk. Umožňuje vývoj mnoha druhů bezpečných a robustních aplikací, které běží v .NETu. Má své kořeny v rodině jazyků C [30].

Jazyk C# verze 1.0 byl vyvinut společně s Visual Studio .NET 2002, a připomínal Javu. Byl původně vyvinut jako alternativa Javy pro platformu Windows [29]. Od té doby se jazyk vyvinul a aktuálně je dostupná verze C# 10.0, která je součástí .NET 6, jenž byla vydána koncem letošního roku.

#### ASP.NET Core

ASP.NET Core je vysoce výkonný multiplatformní open-source framework pro vývoj moderních aplikací. Microsoft v rámci tohoto frameworku nabízí spoustu možností, jakým lze

vyvíjet aplikace. Pomocí tohoto frameworku lze vyvíjet webová API, služby, Internet of Things (IoT), mobilní aplikace a další [2].

Vývojářům umožňuje v celku rychle a jednoduše vyvíjet webové API a také webové aplikace na základě MVC patternu. Obsahuje šablony projektů, které jsou již předpřipravené, vývojář tím má ušetřenou práci. Může se tak soustředit přímo na psaní kódu, který aplikaci tvoří jedinečnou.

Výhodou tohoto frameworku je, že umožňuje velice snadno vytvářet vlastní kontrolery (anglicky *Controllers*), což jsou části aplikace, které reagují na HTTP požadavky přicházející na server.

Controller je vždy nějaká třída dědicí z `ControllerBase` třídy (lze dědit i z jiných, například `Controller`) a tomuto kontroleru je třeba přiřadit cestu, která bude představovat koncový bod, na který se klientské aplikace mohou dotazovat. Součástí tohoto kontroleru jsou metody, které mohou být označeny atributy, jež říkají jakou HTTP metodu přijímají. Mohou mít i vlastní cestu koncového bodu, pokud není uvedena pomocí atributu, tak se použije cesta kontroleru. Dále lze v parametrech metody předat například další informace, jako id uživatele, query nebo další data – ty lze extrahovat z cesty nebo z těla HTTP požadavku. Jako návratový typ metody lze použít prakticky cokoliv, framework jej totiž zapouzdří do HTTP odpovědi. Nicméně běžně se používá interface `IActionResult` nebo přímo třída `ActionResult`, které mohou specifikovat stavový kód odpovědi. Vnitřní implementace kontroleru a jeho metod závisí na vývojáři, běžně se jedná o operace s databází. Příklad metody kontroleru lze vidět ve výpisu 3.5, kód byl převzat [4].

```
1 [HttpGet("{id}")]
2 public async Task<ActionResult<TodoItem>> GetTodoItem(long id)
3 {
4     var todoItem = await _context.TODOItems.FindAsync(id);
5
6     if (todoItem == null)
7     {
8         return NotFound();
9     }
10
11     return todoItem;
12 }
```

Výpis 3.5: Příklad metody kontroleru

Tato metoda přijímá aktuální cestu kontroleru a reaguje na GET požadavek, který je poslán na koncový bod, kde je v cestě koncového bodu specifikováno id položky. Metoda následně vyhledá položku v databázi a vrátí ji formou HTTP odpovědi. Pokud položka nebyla nalezena, vrátí se HTTP odpověď se stavovým kódem 404 Not Found.

## Entity Framework Core

Entity Framework (EF) Core je další framework vyvinutý společností Microsoft. Je open-source, multiplatformní a používá se pro přístup k datům. Může sloužit jako Object-Relation

mapper, to znamená, že umožňuje pracovat s databází pomocí objektů .NET. Poskytuje rozhraní, skrze které lze přistupovat k databázím [13].

Tento framework podporuje mnoho běžně používaných poskytovatelů databází jako jsou například SQL Server, SQLite, Cosmos, PostgreSQL, OracleDB a další viz<sup>4</sup> [13].

EF Core využívá dva přístupy pro užití databází v projektu [13]:

- **Code First** – V tomto přístupu se předpokládá scénář, kdy výsledná databáze ještě neexistuje. Umožňuje ji tak vygenerovat skrze napsaný kód. Nejprve jsou nejprve vývojářem napsány modely (třídy) v jazyce C#. Modely mohou běžně obsahovat své vlastnosti, a pak také navigační vlastnosti, které ve výsledku tvoří relaci mezi modely (tabulkami). Dodatečně lze konfigurovat pomocí atributů nebo FluentAPI další věci jako klíče, omezení a mnohé jiné. Poté se vytvoří třída, která představuje databázový kontext, kterému je potřeba nakonfigurovat modely, aby představovaly entity databáze. Microsoft poskytuje i nástroje pro vytvoření Code First modelů již z existující databáze.
- **Database First** – V tomto případě je třeba použít EF Designer, kterému se nastaví databáze, která se má použít pro generování modelů. Pokud jsou v rámci databáze nastavená již nějaká další nastavení nebo omezení, tak se v rámci modelů také vygenerují.

Při **Code First** přístupu je nakonec také potřeba provést migraci databáze, aby se reálně databáze vygenerovala. To lze udělat pomocí několik příkazů v příkazové řádce nebo pomocí nějakého nástroje s grafickým rozhraním. Pojem migrace byl vysvětlen v sekci 3.1.4.

Následující příklad kódu (výpis 3.6) obsahuje modely napsané v jazyce C#, které jsou pak použity pro vygenerování databáze, jejíž strukturu lze vidět na obrázku 3.7. Kód byl převzat [3].

---

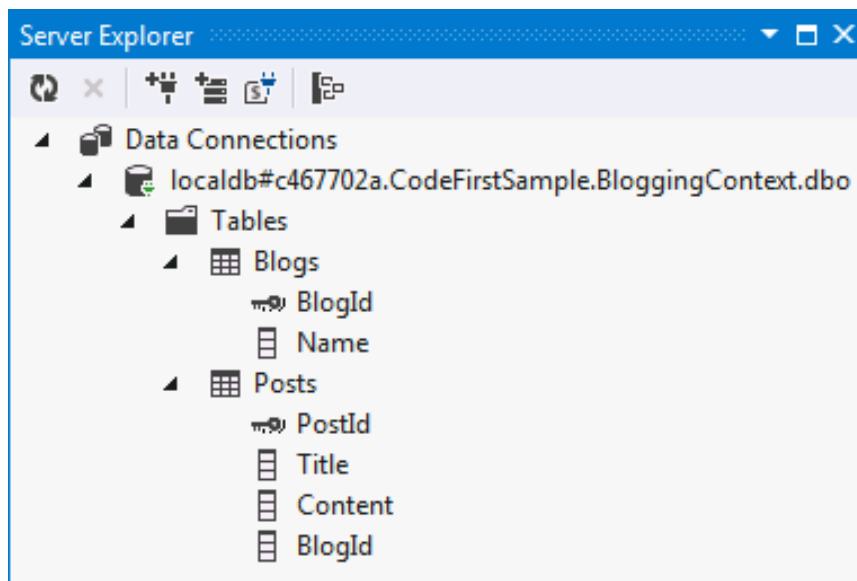
<sup>4</sup><https://docs.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli>

```

1 public class Blog
2 {
3     public int BlogId { get; set; }
4     public string Name { get; set; }
5
6     public virtual List<Post> Posts { get; set; }
7 }
8
9 public class Post
10 {
11     public int PostId { get; set; }
12     public string Title { get; set; }
13     public string Content { get; set; }
14
15     public int BlogId { get; set; }
16     public virtual Blog Blog { get; set; }
17 }

```

Výpis 3.6: Modely napsané v jazyce C#



Obrázek 3.7: Vygenerované modely

Převzato [3]

### 3.2.6 Front-end technologie

Front-end představuje část aplikace, se kterou interaguje uživatel. Jedná se o veškeré grafické elementy, komponenty jako jsou tabulky, grafy, tlačítka, formuláře a jiné. V sekci 3.1.3 byly zmíněny některé možné front-end technologie. Pro tuto práci jsou popsány webové technologie a frameworky.

## HTML

HyperText Markup Language (HTML) je značkovací jazyk, který definuje strukturu webových stránek [38]. „Hypertext“ označuje odkazy, které vzájemně propojují webové stránky, a to buď v rámci jednoho webu, nebo mezi weby. Odkazy jsou základním aspektem webu, ve chvíli kdy stránka je dostupná z internetu se stane účastníkem World Wide Webu [18].

K vytvoření struktury dokumentu používá značky. Dokument se tedy skládá z řady prvků, které se používají k zabalení různých částí obsahu. Lze vytvářet nadpisy, odstavce, obrázky, tabulky a mnohé další prvky. Element obsahuje otevírací tag, obsah a uzavírací tag. Existují i elementy, které jsou samo-uzavírající se (pouze jeden tag). Otevírací tagy mohou mít další atributy, například třídu nebo identifikátor. Příklad struktury HTML dokumentu lze vidět ve výpise 3.7 [18].

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Title</title>
6   </head>
7   <body>
8     <div>
9       <p>Pararaph text</p>
10    </div>
11  </body>
12 </html>
```

Výpis 3.7: Ukázka struktury HTML kódu

- `<!DOCTYPE html>` – Označuje začátek dokumentu, jeho typu.
- `<html></html>` – Tento element zapouzdřuje obsah celé stránky, označuje se jako kořenový element.
- `<head></head>` – Hlavička není součástí výsledné zobrazené stránky, ale vkládají se sem dodatečné informace, odkazy na styly, odkazy pro analytiku a další.
- `<title></title>` – Titulek dokumentu.
- `<body></body>` – Tělo dokumentu, obsahuje veškeré elementy, které budou zobrazené na výstupu.

## CSS

Cascading Style Sheets (CSS) je jazyk, pomocí kterého lze HTML elementům přiřadit styly. Umožňuje vybírat HTML elementy a přiřazovat konkrétní styly, ty mohou být definovány pro skupiny elementů, pro všechny, pro konkrétní nebo jen pro vybrané. Často se k tomu využívají právě identifikátory nebo třídy, které definují styl pro určité elementy. Styly lze psát přímo do atributu HTML tagu (inline styl) nebo je lze mít v samostatném souboru

s příponou `*.css`. Lze tak nastavovat barvy textu, pozadí, šířku/výšku elementu a další. Ukázku stylů lze vidět ve výpise 3.8 [5].

```
1 .button {  
2   background-color: #FF9500;  
3   padding: 0 20px 20px 20px;  
4   border: 5px solid black;  
5 }
```

Výpis 3.8: Ukázka struktury CSS stylů

Text `.button` je selektor elementu, v tomto případě je styl aplikován na všechny elementy, které mají v atributu tuto třídu. Přiřadí se jim vlastnosti, kdy budou mít specifikovanou barvu pozadí `#FF9500`, padding a černý 5px rámeček.

Styly se však chovají v každém prohlížeči trochu jinak, o to je složitější je psát. Vývojář musí dbát na podporu všech prohlížečů, kde chce, aby se webová stránka zobrazovala rozumně bez chyb. To může prodloužit vývoj aplikací, nicméně existuje spousta frameworků, které tyto problémy umí vyřešit.

## JavaScript

JavaScript (JS) je populární programovací jazyk, který přidává dodatečnou interaktivitu webovým aplikacím. Jedná se v celku o jednoduchý jazyk s dlouhou historií. Jedná se o skript, který je spouštěn u uživatele přímo ve webovém prohlížeči [23].

Skript lze vložit přímo do dokumentu HTML pomocí tagu `<script></script>` nebo jej lze vložit do samostatného dokumentu, který HTML dokument musí referencovat, aby se spustil. Tento skript se vykoná při přenesení na stranu klienta. Mimo jiné umí právě vybírat elementy, které jsou přístupné skrze Document Object Model (DOM). Může tak průběžně vykonávat nějaký kód na pozadí a měnit elementy, které jsou zobrazené uživateli [23].

Na JavaScriptu je založených několik populárních frameworků, které se aktivně využívají při vývoji webových aplikací, patří mezi ně například Vue.js, Angular nebo React.

## Tailwind CSS

Tailwind je CSS framework, který usnadňuje vývojáři se stylizací HTML elementů. Byl vytvořen právě z důvodu, že CSS dokumenty jsou těžko udržitelné. Tailwind poskytuje právě již připravené třídy, které obsahují veškeré styly. Vývojář tak nemusí psát žádný CSS kód, ale pouze jednotlivým elementům přiřazuje třídy [47].

Obsahuje ve výsledku nejlepší praktiky, které by měl vývojář při psaní CSS dodržovat. Obsahuje prvky, které pomáhají udržovat design konzistentní. Jeho součástí jsou i breakpoints, které umožňují snadno vytvářet responzivní design stránek. Jeho výhodou také je, že automaticky odstraňuje veškeré nepoužité styly, když se sestavuje aplikace na produkci. Takže finální verze aplikace obsahuje co nejmenší balíček CSS stylů. Také je dostupný v rámci populárních frameworků jako je například React, Vue.js, Angular. Tailwind sice poskytuje

již připravené třídy, palety, ale lze jej i přesto rozšiřovat podle potřeby. Ukázku třídy lze vidět ve výpise 3.9 [47].

```
1 bg-gray-50 background-color: rgb(249 250 251);
```

Výpis 3.9: Tailwind třída

Název třídy je `bg-gray-50` a k ní je odpovídající CSS `background-color: rgb(249 250 251);`.

## React

React je open-source JavaScript knihovna pro vytváření webových uživatelských rozhraní. Používá se pro vytváření Single Page Aplikací (sekce 3.2.1). Základním stavebním blokem Reactu jsou komponenty, což může být buď třída nebo běžná JavaScriptová funkce. Tyto komponenty jsou překreslovány dynamicky během uživatelovi interakce s aplikací, tedy při změně různých vstupních parametrů [40].

Součástí Reactu je taky jazyk JSX. Jedná se o rozšíření syntaxe JavaScriptu, běžně se používá při implementaci komponent. Je to vlastně JavaScript rozšířený o HTML tagy. Příklad komponenty lze vidět ve výpise 3.10.

```
1 export const Component = (props) => {
2   return (
3     <div>
4       <h1 className="font-bold">{props.title}</h1>
5       <div>
6         {props.children}
7       </div>
8     </div>
9   );
10 };
```

Výpis 3.10: Ukázka komponenty v Reactu

Komponenta `Component` vykreslí `<div>` obsahující nadpis úrovně jedna a další `<div>` uvnitř, který obsahuje další potomky. Potomkem mohou být například další komponenty. Tato komponenta se vykreslí znovu při změně vstupních parametrů `props`.

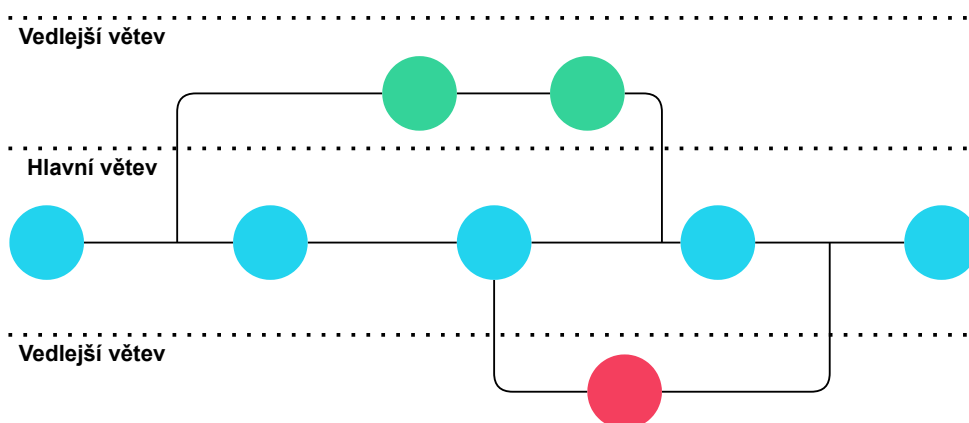
### 3.2.7 Další nástroje

#### Verzovací systém GIT

Při vývoji aplikací se běžně využívají verzovací systémy. Nejpopulárnějším a nejvyužívanějším je verzovací systém `git`. `git` je volně dostupný open-source verzovací systém, který je schopný pojmout verzování i větších projektů rychle a efektivně.

Verzování v `gitu` funguje tak, že projekt je uložen v repozitáři někde na vzdáleném serveru. Odtud si vývojář(i) mohou stáhnout kopii lokálně k sobě. Vše funguje na základě příkazů,

kteře tento systém poskytuje. Základním stavebním kamenem gitu jsou větve. Hlavní větev se nazývá `master` nebo nově `main`, vývojáři si pak mohou vytvářet vlastní větve, kde implementují vlastní změny v kódu. Tyto změny je poté třeba zaznamenat a uložit pomocí příkazu `commit`, ke kterému autor přidá zprávu (většinou obsahuje stručný popis změny). Změny lze provádět přímo v hlavní větvi, což není běžné při kolaboraci více vývojářů nebo je lze provést ve větvi vytvořené z jiné a poté změny sloučit do hlavní větve. Tento systém pak má kompletní historii o změnách, jaké byly provedeny, kým a kdy. V případě, že změna způsobí chyby v aplikaci, lze se vrátit ke starším verzím. Ukázkou větvení v gitu lze vidět na obrázku 3.8.



Obrázek 3.8: Ukázkou větvení v gitu  
Vlastní zpracování dle [15].

Každý kruh představuje jeden `commit`. Každé spojení představuje sloučení změn do hlavní větve.

## Docker

Docker je open-source projekt pro automatizaci nasazování aplikací, ty jsou snadno přenositelné a mohou běžet prakticky kdekoli, jsou tvořeny pomocí „kontejnerů“.

Kontejner je vlastně jednotka softwaru, která zabaluje kód se všemi jeho závislostmi do jednoho balíčku. Je rychlý, spolehlivý, obsahuje jen to co potřebuje a lze jej přenášet snadno mezi různými prostředími. Obraz kontejneru je samostatný spustitelný balíček softwaru, obsahuje kód, runtime prostředí, systémové nástroje, knihovny, nastavení. Jsou dostupné na Windows, Linux i MacOS [10].

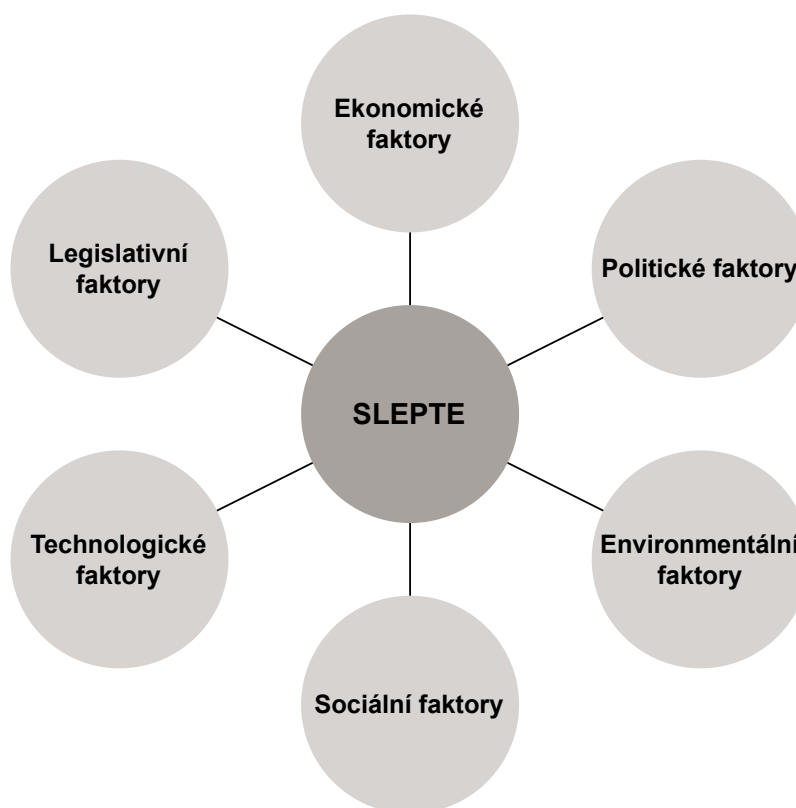
K vytvoření obrazu kontejneru z konkrétní aplikace se používá soubor `Dockerfile`, který specifikuje, jakým způsobem se má kontejner vytvořit, co potřebuje, jaké závislosti má mít atd [11].

### 3.3 Analytické nástroje

Tato sekce pojednává o analytických nástrojích a modelech, které jsou v rámci této diplomové práce použity pro analýzu současného stavu společnosti XYZ a.s.

#### 3.3.1 SLEPTE analýza

Tato metoda se používá při analýze a hodnocení vnějšího podnikatelského prostředí organizace. Využívá se k udržení úspěchu organizace a konkurenceschopnosti neustálým sledováním vnějšího prostředí [42]. Jedná se o rozbor společenských, ekonomických, legislativních, politických, technických a ekologických oblastí. Všechny tyto oblasti nějakým způsobem do určité míry působí na podnik. Někdy lze tuto analýzu vidět i pod názvy PEST, STEP, PESTLE [28]. Každé písmeno této metody představuje jednu oblast. Následně jsou přiblíženy jednotlivé oblasti respektive faktory [42].



Obrázek 3.9: Model SLEPTE  
Vlastní zpracování na základě [33].

#### Politické faktory

Politické faktory souvisejí spíše s aktuální politickou situací zemí, ve kterých organizace podniká. Závisí například na obchodních omezeních, daňových reformách, politické stabilitě, míry korupce, na regulacích a dalších. Pokud subjekt působí například v rámci EU, tak bude působit i v mezích legislativy EU, a následně v mezích zákonů jednotlivých zemí.

### **Ekonomické faktory**

Mezi ekonomické faktory lze zařadit například míru nezaměstnanosti, inflaci, směnné kurzy a další. Jedná se také o faktory, které jsou závislé na fiskální a monetární politice státu. Může se taky jednat o vývoj cen na trhu v různých odvětvích nebo například mzdu vyplacenou za práci.

### **Sociální faktory**

Jedná se o společenské faktory, které se týkají kupříkladu spotřebitelských návyků, postavení široké veřejnosti. Mezi tyto faktory patří i vzdělanost, demografický vývoj populace, náboženství, sociální skupiny nebo zástupnost pohlaví.

### **Technologické faktory**

Pro moderní podniky jsou technologické faktory zásadní v každodenním provozu. Aspekty zde zahrnují technologické pobídky, úroveň infrastruktury, výzkumnou a vývojovou činnost, rychlost technologických změn a přístup k technologiím.

### **Legislativní faktory**

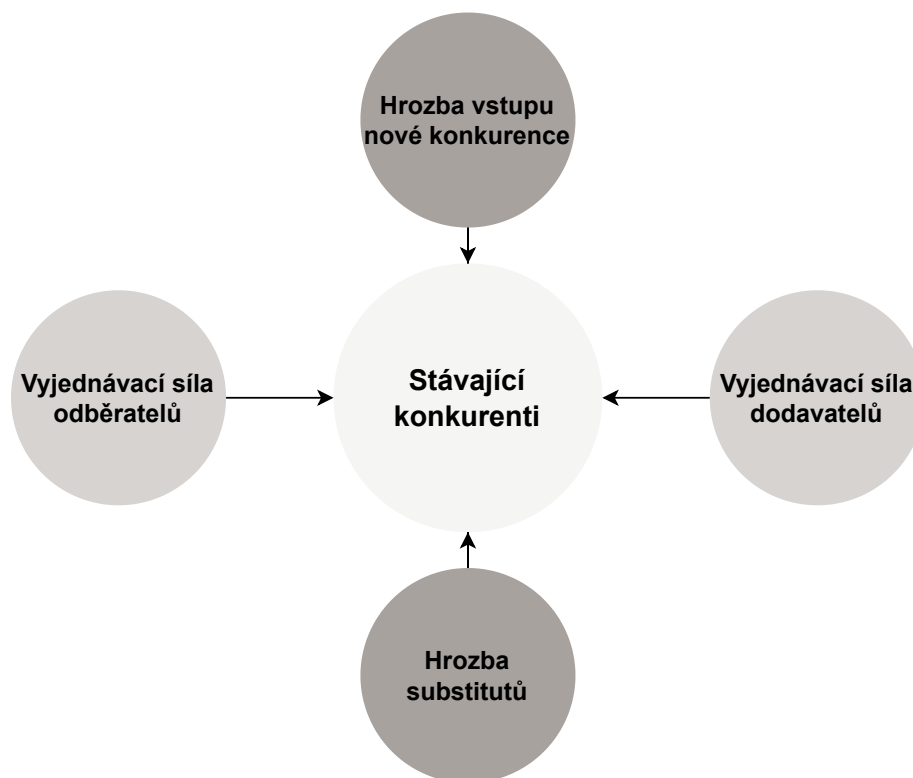
Právní faktory odkazují na vládní legislativu, která ukládá organizaci povinnosti nebo omezení. Zákony a předpisy jsou neustále aktualizovány v celé řadě oblastí. Právní faktory zahrnují zákony pro diskriminaci, antimonopolní zákony, normy, standardy, duševní vlastnictví a zaměstnávání.

### **Environmentální faktory**

Mezi Environmentální faktory patří přírodní katastrofy, prostředí podniku, klimatické změny nebo ochrana životního prostředí.

### **3.3.2 Porterova analýza**

Porterův model pěti sil se používá pro analýzu konkurenčního prostředí společnosti. Lze jej aplikovat na jakýkoli ekonomický segment, pro porozumění konkurenci. Analýza pěti sil může být použita k vedení obchodní strategie ke zvýšení konkurenční výhody. Model lze vidět na obrázku 3.10, následně jsou popsány jednotlivé síly, které na společnost působí [35].



Obrázek 3.10: Porterův model  
Vlastní zpracování na základě [36].

### Stávající konkurence

První z pěti sil se týká počtu konkurentů a jejich schopnosti podbízet se společnosti. Čím větší je počet konkurentů spolu s počtem ekvivalentních produktů a služeb, které nabízejí, tím menší je síla společnosti. Dodavatelé a kupující vyhledávají konkurenci společnosti, pokud jsou schopni nabídnout lepší nabídku nebo nižší ceny. Naopak, když je konkurenční rivalita nízká, má společnost větší sílu účtovat vyšší ceny a nastavovat podmínky obchodů tak, aby dosahovala vyšších tržeb a zisků.

### Hrozba vstupu nové konkurence

Síla společnosti je také ovlivněna silou nových subjektů vstupujících na její trh. Čím méně času a peněz stojí konkurenta vstoupit na trh společnosti a být efektivním konkurentem, tím více by mohla být pozice zavedené společnosti výrazně oslabena. Odvětví se silnými překážkami vstupu je ideální pro stávající společnosti v tomto odvětví, protože společnost by mohla účtovat vyšší ceny a vyjednávat lepší podmínky.

### Vyjednávací síla odběratelů

Jedná se o schopnost odběratelů, kteří mohou ovlivnit cenu a poptávané množství daného produktu. Tato síla je ovlivněna počtem odběratelů společnosti, a toho, jak je každý jednotlivý odběratel důležitý pro společnost. Tedy i to jaké jsou náklady na nalezení nových odběratelů. Pokud má společnost pouze několik silných odběratelů, tak mají větší vyjedná-

vací sílu. Při větším množství menších odběratelů má naopak společnost větší vyjednávací sílu.

### **Vyjednávací síla dodavatelů**

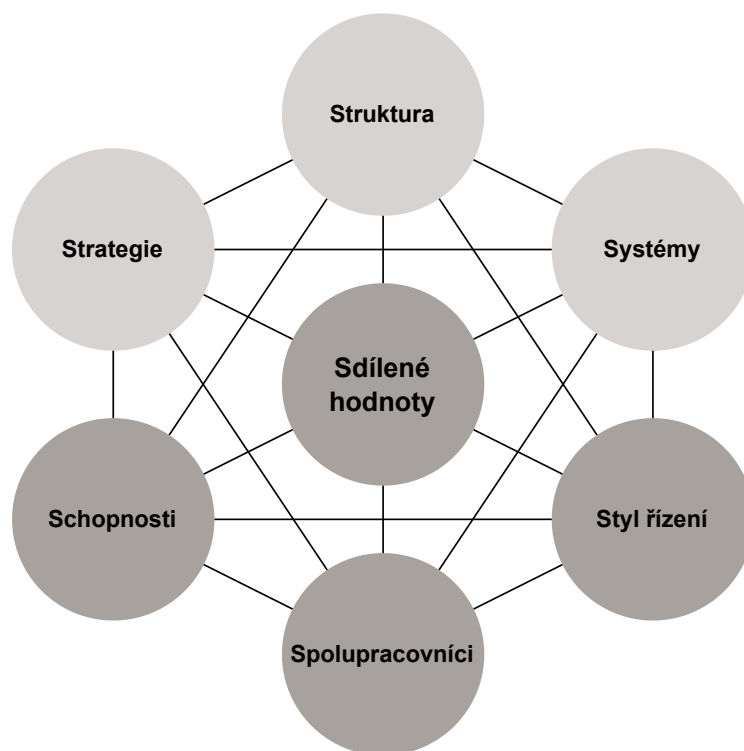
Jedná se o schopnost dodavatelů, kteří mohou ovlivnit cenu a nabízené množství daných vstupů. Tato síla je ovlivněna počtem dodavatelů klíčových vstupů, unikátností vstupu, velikostí a silou dodavatele a náklady na přechod k jinému dodavateli.

### **Hrozba substitutů**

Poslední z pěti sil se zaměřuje na náhradní zboží nebo služby, které lze použít místo produktů nebo služeb společnosti. Společnosti, které produkují zboží nebo služby, pro něž neexistují žádné blízké substituty, budou mít větší sílu zvýšit ceny a zajistit výhodné podmínky. Když jsou k dispozici blízké substituty, zákazníci budou mít možnost vzdát se nákupu produktu společnosti a síla společnosti může být oslabena.

### **3.3.3 7S analýza**

Model McKinsey 7S je nástroj, který analyzuje interní prostředí společnosti pohledem na 7 klíčových vnitřních prvků, aby se zjistilo, zda jsou efektivně sladěny a umožňují organizaci dosáhnout svých cílů. Model lze vidět na obrázku 3.11, následně jsou popsány jednotlivé klíčové oblasti [1].



Obrázek 3.11: Model 7S McKinsey  
Vlastní zpracování dle [1].

## **Strategie**

Jedná se o definici cílů a stanovení způsobu, jak se cílů dosáhne. Dobrá strategie je ta, která je jasně formulovaná, je dlouhodobá, pomáhá dosáhnout konkurenční výhody a je posílena silnou vizí, posláním a hodnotami. Strategie by měla být v souladu s dalšími šesti prvky.

## **Struktura**

Struktura představuje způsob, jakým je společnost uspořádána. Jinými slovy jde o organizační schéma společnosti a zahrnuje to, kdo je komu zodpovědný [1].

## **Styl řízení**

Styl řízení představuje způsob, jakým je společnost řízena vedením, tedy to jakým způsobem jednají, jak se chovají, jak konají.

## **Systémy**

V tomto pojetí jsou systémy procesy a postupy společnosti, které společnost využívá při každodenních aktivitách. Zahrnuje to i technické a informační systémy.

## **Sdílené hodnoty**

Sdílené hodnoty jsou jádrem modelu McKinsey 7s. Jedná se o vizi společnosti a její firemní kulturu. S tím souvisí i to co společnost dělá, jakým způsobem a proč.

## **Spolupracovníci**

Tato oblast se zabývá tím, jaké zaměstnance bude společnosti potřebovat, a jak budou přijímáni, školeni, motivováni a odměňováni.

## **Schopnosti**

Schopnosti zahrnují dovednosti, znalosti, zkušenosti společnosti jako celku. Tedy to jakými schopnostmi disponuje nebo jí naopak chybí. Schopnosti, které je potřeba v rámci společnosti rozvíjet.

### **3.3.4 SWOT analýza**

SWOT analýza je nástroj, pomocí kterého se posuzují slabé a silné stránky společnosti. Stejně tak se posuzují možné příležitosti pro společnost, nebo hrozby, kterým může čelit. Model lze vidět na obrázku 3.12, následně jsou popsány jednotlivé oblasti [46].



Obrázek 3.12: SWOT matice  
Vlastní zpracování na základě [46].

### **Silné stránky**

Zde se analyzují silné stránky společnosti, tedy to v čem vyniká, v čem se odlišuje od ostatních, jaké má zázemí, jedinečné technologie, produkty a další.

### **Slabé stránky**

Důležitá je identifikace slabých stránek společnosti, které ji mohou bránit v lepším výkonu. Jedná se o oblasti, které společnost potřebuje zlepšit pro udržení nebo zvýšení konkurenceschopnosti.

### **Příležitosti**

Příležitosti se týkají příznivých vnějších faktorů, které by mohly společnosti poskytnout konkurenční výhodu.

### **Hrozby**

Hrozby se týkají faktorů, které mají potenciál uškodit společnosti. Například sucho je hrozbou pro společnost produkující pšenici, protože může snížit výnos plodin. Dalšími hrozbami mohou být například rostoucí náklady na materiály, rostoucí konkurence, omezená nabídka pracovních sil.

### **3.3.5 ZEFIS**

ZEFIS je webový portál, kde si lze provést audit informačního systému pomocí několika dotazníků. Pomáhá zlepšit efektivnost fungování společnosti. Jedná se o prověření oblastí v fungování firmy, jejich procesů a informačních systémů. Ověřuje úroveň bezpečnosti, včetně GDPR [50].

Po vyplnění základních informací a následném projití dotazníků se zobrazí výsledek auditu. Tento výsledek obsahuje oblasti, kde jsou nedostatky, hrozby, míru rizika a další. Umožňuje vyhledávat slabiny v systémech a procesech.

## 3.4 Řízení projektu

V této sekci jsou vysvětleny nástroje a postupy, které jsou v rámci práce využité k řízení projektu.

### 3.4.1 Lewinův model

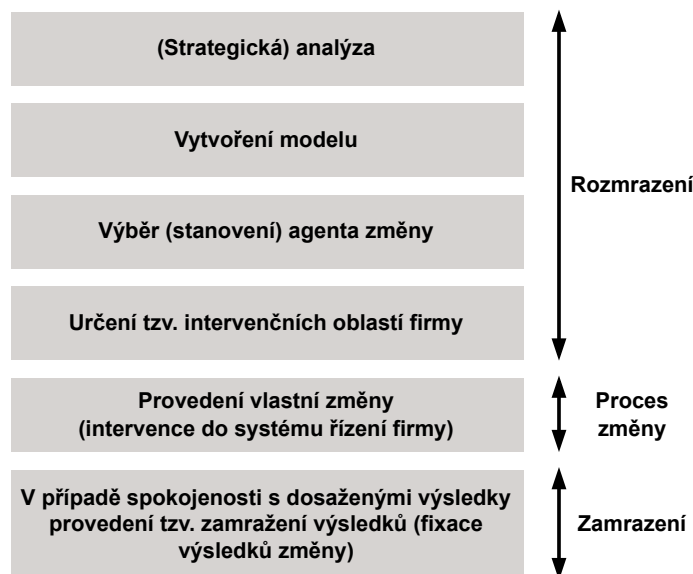
Lewinův model se používá při řízení změny v organizaci. Změny jsou iniciovány vnějšími nebo vnitřními faktory. Vnější příčinou může být například vývoj nových materiálů, nová legislativa, změna globální politiky a jiné. Vnitřní příčiny mohou být například nízká efektivita společnosti, nový produkt, nové cíle.

Proces řízené změny v organizaci lze rozdělit na několik etap [43]:

- **Analytická etapa**
- **Návrhová etapa** – vytvoření modelu změny, stanovení agenta změny (podporován sponzorem), určení ovlivněných procesů
- **Realizační etapa** – vlastní změna
- **Zpětnovazební etapa** – úpravy nebo akceptování výsledků

Podle Lewina se tedy každá úspěšná změna skládá ze tří kroků, přičemž dílčí etapy těchto kroků lze vidět na obrázku 3.13 [43]:

- **Fáze rozmrazení** – Příprava změny
- **Fáze změny** – Přechod na novou úroveň, intervence ve měněném systému
- **Fáze zamrazení** – Fixace dosažených výsledků



Obrázek 3.13: Základní kroky procesu modelování řízené změny ve firmě  
Vlastní zpracování na základě [43].

Nejprve je potřeba provést analýzu silového pole, které obsahuje podpůrné síly a brzdné síly. Důležité je, aby podpůrné síly převažovaly, pro úspěšné zavedení změny. Při převažujících podpůrných sil lze přistoupit k identifikaci rolí změn. Je potřeba stanovit agenta změny, sponzora a případně i advokáty. Následně je potřeba analyzovat jak se změna dotkne intervenčních oblastí, ty jsou [43]:

- Technologie
- Toky a procesy
- Organizační struktura
- Lidské zdroje

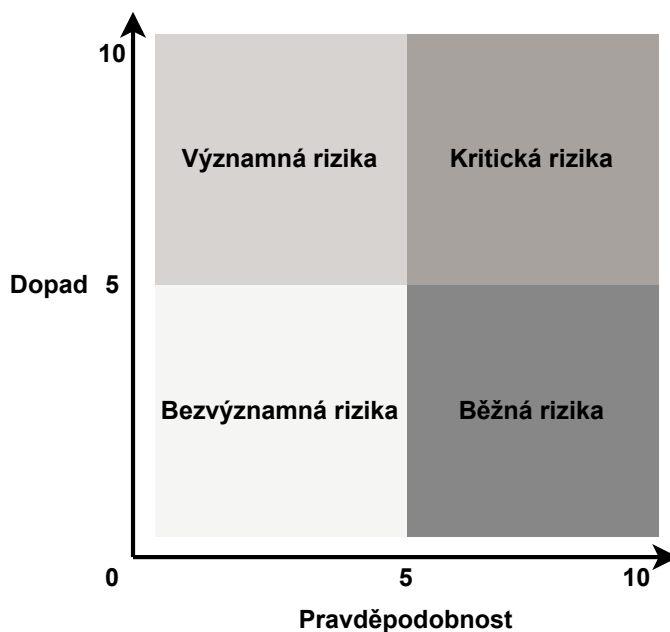
Pak následuje tedy proces vlastní změny, která je jedinečná pro každou organizaci. Po zavedení změny se provede zmrazení, kdy se změna stabilizuje.

### 3.4.2 Skórovací metoda

Skórovací metoda se používá pro analýzu rizik. Používá se s mapou rizik, kdy se vytvoří tabulka s hodnotami a pravděpodobnostmi výskytu hrozeb. S tím je vytvořena i tabulka s hodnotami a ohodnocením dopadu rizika. Má následující tři kroky [22]:

1. **Identifikace rizik** – Prvním krokem je identifikace možných rizik, která jsou relevantní pro řešený problém. Rizika lze identifikovat například pomocí brainstormingu, myšlenkových map, různých registrů, zkušeností a dalších. K hrozbě se pojí vždy nějaký scénář, při identifikaci lze postupovat dvěma směry – od hrozby ke scénáři nebo naopak.

2. **Ohodnocení rizik** – Dalším krokem je kvantitativní ohodnocení veškerých rizik. Vynásobí se pravděpodobnost výskytu s dopadem, což vytvoří celkovou hodnotu rizika. Čím vyšší hodnota, tím větší riziko.
3. **Návrh opatření rizik** – Posledním krokem je návrh na opatření všech rizik. V tomto kroku by mělo dojít ke snížení rizik na akceptovatelnou úroveň. Po tomto kroku by se rizika měly vyskytovat ideálně v kvadrantu bezvýznamných rizik, který lze vidět na mapě rizik na obrázku 3.14.



Obrázek 3.14: Mapa rizik skórovací metody  
Vlastní zpracování na základě [22].

### 3.4.3 Časová analýza PERT

Project Evaluation and Review Technique (PERT) je metoda hodnocení a kontroly projektu. Jedná se o postup tvorby a hodnocení síťových diagramů tvořených úkoly a událostmi a související kontroly postupu projektu vzhledem k plánovanému diagramu. Odhady vychází z kombinace optimistických, běžných a pesimistických variant trvání jednotlivých úseků projektu a dalších statistických výpočtů a predikcí [45].

Nevýhodou této metody je její složitost. Je podobná metodě CPM, která ale používá pouze jeden odhad délky trvání aktivity. Obě tyto metody mají definovanou kritickou cestu, což je cesta, kde je potřeba dodržovat přesně harmonogram, kdyby došlo ke zpoždění jediné aktivity na cestě, došlo by ke zpoždění celkového projektu [45].

Pro sestavení časové analýzy lze vytvořit tabulku nebo síťový graf. Síťový graf je konečný, souvislý, orientovaný, acyklický, hranově nebo uzlově definovaný graf, který vyjadřuje závislosti na jednotlivých činnostech [21].

## Kapitola 4

# Analýza problému a současné situace

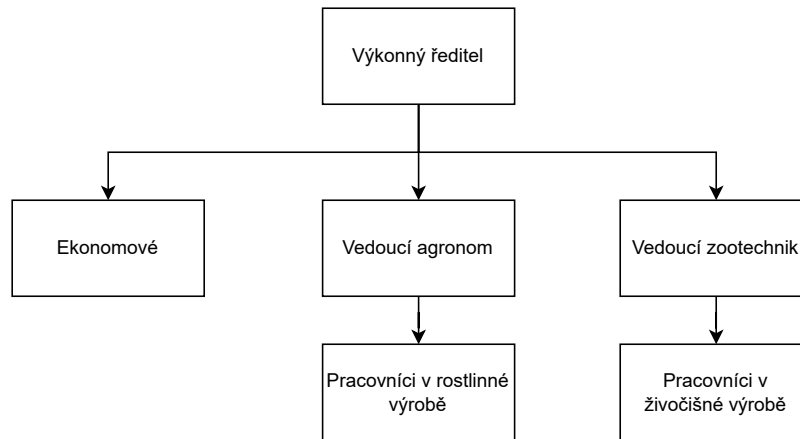
V této kapitole je nejprve blíže představena společnost a čtenář je zároveň uveden do současné situace společnosti. Následně jsou provedeny jednotlivé strategické analýzy společnosti v obecném a oborovém okolí a následně interním prostředí. Na konci této kapitoly je vytvořen souhrn všech těchto analýz.

### 4.1 Představení společnosti

Společnost, která je řešena v rámci této práce, je zprivatizovaný zemědělský podnik po pádu socialistického režimu v České republice. Ze zemědělského hlediska se jedná o střední podnik, z hlediska ekonomického o malý, jelikož má obrat okolo 70 miliónů korun. Jedná se pouze o dílčí část, neboť je součástí koncernu několika podniků. Společnost hospodaří v Královéhradeckém kraji na více než 1600 ha zemědělské půdy. Zabývá se chovem krav francouzského plemene bez tržní produkce mléka, údržbou krajiny a s ní související rostlinnou výrobou, na kterou se primárně zaměřuje. Aktuálně zaměstnává okolo 20 pracovníků v několika střediscích výroby.

#### 4.1.1 Organizační struktura společnosti

Společnost sice patří pod koncern firem, nicméně organizační struktura samotné společnosti je spíše plochá a jednoduchá. Společnost řídí výkonný ředitel, který je členem představenstva koncernu. Pod jeho vedením je vedoucí agronom, vedoucí zootechnik, kteří mají pod sebou pracovníky v prvovýrobě. Také pod ním figurují dva ekonomové. Organizační strukturu lze vidět na obrázku [4.1](#).



Obrázek 4.1: Organizační struktura společnosti

## 4.2 Současná situace a řešení

Společnost začala více dbát na zavedení různých a nových technologií v posledních letech. Obměnila některé zastaralé systémy a zavádí systémy tam, kde chybí. Aktuálně využívá několik informačních systémů a nástrojů, které jí pomáhají efektivněji vést podnik. Na trhu pro toto odvětví aktuálně neexistují nástroje pouze od jednoho dodavatele, a tak společnost využívá různé balíčky, licencované systémy nebo má poskytnut software formou SaaS.

Společnost v současné situaci nedisponuje žádným informačním systémem pro řízení skladových zásob. Proto aktuálně hledá a vyžaduje systém, který by zlepšil jejich nedostatek v řízení skladových zásob.

Aktuálně to společnost řeší následovně: Existuje jeden člověk, který má na starosti evidenci skladů a je za ně zodpovědný, tím je ekonom, který záznamy zapisuje přímo do účetního systému. Tyto záznamy zapisuje jednou za měsíc z tištěných přejímek, které dostává od osoby, která se stará o inventuru skladů. Osoba, která se stará o inventuru skladů, je rovněž jediná osoba, jenž zapisuje záznamy do přejímky. Inventura probíhá periodicky každý týden až dva. Tištěné přejímky jsou po zavedení do účetního systému každý měsíc archivovány. Tištěnou přejímku lze vidět na obrázku 4.2.



Není vlastně poskytnut ani žádný aktuální stav skladů. Aby se zjistilo jaké skladové položky jsou dostupné na skladě, kolik a o jaký sklad se jedná, tak někdo fyzicky musí sklad zkontrolovat.

## 4.3 Strategické analýzy

V této sekci jsou postupně provedeny strategické analýzy nejprve obecného okolí, následně oborového okolí, interního prostředí společnosti a nakonec jsou výsledky analýz shrnuty.

### 4.3.1 SLEPTE

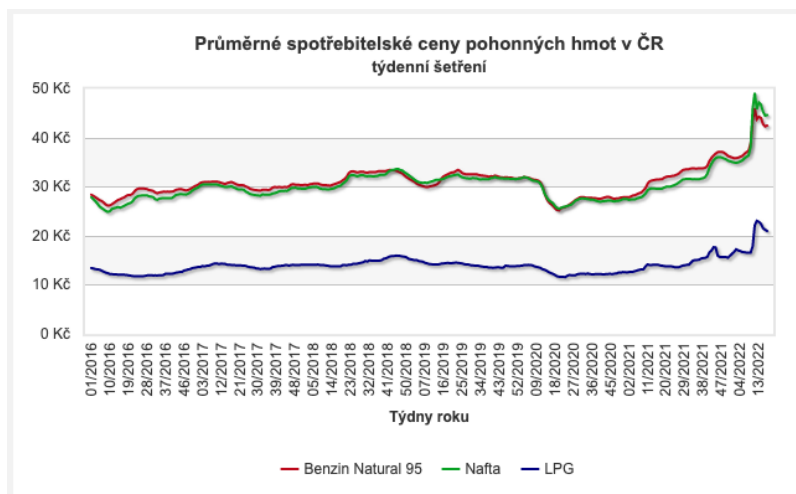
#### Politické faktory

Společnost působí pouze v Královéhradeckém kraji v České republice, kde je politická situace stabilní. Nicméně je jen velmi malou součástí většího koncernu několika společností, které působí v ČR a zemích EU. Sama obchoduje s tuzemskými nebo partnery EU, kteří mají své pobočky v ČR.

Válka na Ukrajině, která začala v únoru tohoto roku (2022) má na společnost a nejen na ni obrovský vliv. A to vzhledem k různým sankcím vůči Rusku, omezení dovozu a vývozu, kdy Ukrajina tvořila v loňském roce přes 30% světového vývozu obilí [48]. Tento fakt bude mít na společnost jak negativní dopad v některých oblastech, tak bohužel nutno říci, že i pozitivní dopad by to mohlo mít pro společnost.

#### Ekonomické faktory

Společnost obchoduje s komoditami, jejichž cena se odvíjí podle aktuálních cen na tuzemských i zahraničních burz. Výše zmíněná válka zapříčinila růst cen pšenice [48], [39]. Sledování cen na trhu je důležitou součástí jejího působení a velmi jí to ovlivňuje. Například rostoucí ceny pohonných hmot musí reflektovat ve své strategii [34].



Obrázek 4.3: Graf cen pohonných hmot

Převzato [34]

V zemědělství v ČR jsou dlouhodobě jedny z nejnižších platů, toto odvětví je pro lidi čím dál méně zajímavé a především málo lukrativní s porovnáním vůči jiným odvětvím [32]. Skutečnost, že je nízký zájem o tuto práci podporuje i rostoucí inflace, která v únoru tohoto roku (2022) stoupla na 12,7% [20].

### **Sociální faktory**

V kraji a především v okolí města Trutnov, od kterého se výrobní středisko společnosti nachází pouze několik kilometrů, má dlouholetou tradici v zemědělském vzdělání. Jedná se o celkem velké množství zaměření, nicméně zájem o toto vzdělání pomalu klesá a ti, kteří toto vzdělání chtějí, pak většinou pokračují na univerzity do velkoměst. Pracovníci v tomto odvětví chybí primárně v prvovýrobě, což může být do budoucna značný problém, jelikož dochází ke stárnutí populace, která v tomto odvětví odchází do důchodu.

Dalším důležitým faktorem je fyzická i psychická náročnost práce. V zimě sice tolik práce nebývá, avšak je potřeba pracovat i venku téměř za každého počasí. V období žní je naopak mnohem delší pracovní doba a pracuje se často až do večera.

### **Technologické faktory**

I pro zemědělskou společnost je velmi důležitý vývoj aktuálních technologií. Klíčové technologie jsou zde primárně navigační systémy, které jsou na dobré úrovni a stále se zlepšují. Ty umožňují například optimální výpočet cest, což může značně šetřit náklady. Důležité jsou i skladové systémy, komunikační, automatizační systémy pro ideální krmení zvířat a další.

Co se týče automatizačních systémů, dochází k automatizaci a autonomii různých strojů. To pak umožňuje vznik nových pozic v zemědělství, kdy by se zaměstnanci museli starat spíše o údržbu a monitorování strojů. Nicméně by to vyžadovalo vzdělanější lidi pro tuto činnost.

### **Legislativní faktory**

Odvětví zemědělství je v rámci Evropské unie z velké části závislé na dotacích. Pro tuto společnost to tvoří asi 50% příjmů. Po vstupu ČR do EU jsou zemědělcům nabízeny evropské dotační programy. Tyto programy bývají vhodně doplněny národními dotačními programy. Dotovaná částka závisí na typu balíku a jeho naplnění [12]. Odvětví je závislé na standardech a regulacích chovu zvířat nebo rostlinné výroby, které stanovuje EU.

### **Environmentální faktory**

Společnost se musí řídit zákony o likvidaci biologických odpadů. Z důvodu globálního oteplování také musí regulovat spotřeby energie, pohonných hmot. Také musí přecházet na udržitelné zdroje, alespoň z nějaké části. Zde tvoří tlak EU formou programů pro ekologičtější Evropu.

## **4.3.2 Porterova analýza**

### **Stávající konkurence**

Při vstupu ČR do EU většina konkurenčních českých společností nedokázala konkurovat zahraničním subjektům. Vstupem do EU se celý zemědělský sektor vystavil přímé konkurenci.

renci dalších 27 států. Kvalita výrobků se u jednotlivých konkurencí liší. Společnosti musí dodržovat určité standardy stanovené českými zákony a směrnicemi EU. Podmínky pro hospodaření, včetně podpor z rozpočtu EU a z národních rozpočtů se však v jednotlivých členských státech liší. Podmínky vyjednané před vstupem deseti nových členských států v roce 2004 znevýhodňují jejich zemědělce oproti farmářům v tzv. „starých členských státech“. Zemědělci „v nových členských státech“ obdrželi v prvním roce svého členství pouze 25% přímých plateb oproti „starým členským státům“. Výše dotací je postupně navyšována a byla vyrovnána roku 2013 [27]. Podmínky se ale v průběhu let mění a ministerstvo zemědělství se neustále snaží zlepšovat podmínky českým zemědělcům v EU. V rámci chovu skotu jsou jak společnost, tak celkově ČR soběstační.

Společnost vnímá jako konkurenty primárně další zemědělské podniky, které hospodaří v zeměpisné blízkosti podniku. Také jako konkurenci vnímá ostatní členské státy EU, a to především ty, které jsou technologicky vyspělejší v oblasti zemědělství. Společnost si musí hlídat ceny komodit, aby nezvedli cenu nad úroveň běžné konkurence, jinak by nebyli schopni prodat. Je tedy možné tvrdit, že konkurence je v tomto odvětví silná.

### **Hrozba vstupu nové konkurence**

Zemědělský trh má spoustu schopných hráčů na trhu, ke všemu je tento trh velmi složitý, zrádný a velmi rizikový na to, aby se zde objevila nová schopná konkurence. Společnosti v 3/4 hospodářského roku vykazují ztrátu, až po období žní dochází k ziskům daného roku. Mnohem pravděpodobnější je vstup možných investorů do tuzemských zemědělských společností, které by se pak mohli stát značnou konkurencí. Existence norem, směrnic, důraz na ekologii opět svědčí spíše v opak vstupu nové konkurence. Hrozba vstupu nové konkurence je vysoce nepravděpodobná.

### **Vyjednávací síla odběratelů**

V průběhu let došlo k vyčištění trhu a úbytku středně velkých odběratelů v tomto odvětví. Trh je tvořen primárně velkými společnostmi nebo menšími soukromými podniky. Ti, kteří výkyvy trhu ustáli, tak odebírají výrobky do dnes. Společnost má sice více dlouhodobějších spoluprací, ale eviduje zájem nových odběratelů. U odběratelů jde spíše o schopnost platit včas a dlouhodobou spolupráci. Ztráta partnera není kritická, avšak nepříjemná z důvodu nejistoty délky spolupráce. Odběratelé vyžadují dodržování standardů, při nedodržení dochází ke srážkám z celkové ceny. Ceny se řídí podle aktuálních cen na burze komodit. Zemědělský sektor je vysoce závislý na odběratelích. Společnost totiž nevytváří jakýkoliv finální produkt, ale jedná se spíše o polotovary, které jsou určeny pro další zpracování. V případě skotu se jedná o pře-prodej telat pro další chovné účely, nebo u staršího skotu na porážku.

### **Vyjednávací síla dodavatelů**

Na trhu existuje mnoho dodavatelů osiv pro půdu a krmných směsí. V EU se nesmí pěstovat nepovolené geneticky modifikované plodiny, zemědělci mají tedy zkušenosti s nákupem kvalitních osiv od certifikovaných dodavatelů. Co se týče osiv, tak dodavatelů je několik a jsou většinou schopni pojmout kapacity pro více konkurentů. Další komoditou, se kterou musí společnost obchodovat jsou krmné směsi, kde je soběstačná asi z 45%. U pohonných

hmot je důležité mít spolehlivého dodavatele. Ceny se opět řídí podle aktuálních cen na burze komodit. Zemědělský sektor je vysoce závislý na dodavatelích.

### **Hrozba substitutů**

České hovězí maso je považováno za velmi kvalitní surovinu, které se těžko nahrazuje. Existují už výzkumy, které umožňují výrobu syntetického hovězího masa, ale to ještě několik desítek let pravděpodobně potrvá, než se stane běžným produktem. Hrozbou jsou výrobu hovězího masa stravovací trendy jako vegetariánství nebo veganství. Ekologické nebo etické aspekty by tuto výrobu taky mohlo ohrozit.

Rostlinná výroba je aktuálně díky cenám a trendům výhodná. Celková hrozba substitutů je střední, záleží, jestli se jedná o živočišnou nebo rostlinnou výrobu.

### **4.3.3 7S analýza**

#### **Strategie**

Strategie společnosti je založena na přísné dodržování standardů, aby produkovala kvalitní plodiny a měla kvalitní zdravá zvířata. Sleduje ceny na trhu a snaží se provádět dobře načasované nákupy nebo prodeje. Dokazují to meziroční výnosy o 10%. V zemědělském sektoru je totiž poměrně těžké utržit vyšší zisky. Snaží se být šetrnější k přírodě. Lišit se nějak výrazně od konkurence nelze, spíš se dbá na zaseté postupy a konzistenci. Nejen, že společnost provozuje primárně rostlinnou a živočišnou výrobu, ale stará se o okolní louky a trvalé travní porosty. Také spolupracuje s lesnickým svazem a myslivci. Má v okolí tak dobrou reputaci.

#### **Struktura**

Jak již bylo zmíněno, společnost funguje pod velkým koncernem, jehož organizační struktura je divizionální. Jinak je organizační struktura poměrně plochá a liniová. Společnost řídí výkonný ředitel, který zde působil prvně jako traktorista a následně posledních 20 let jako vedoucí agronom. Ten pod sebou má další dva ekonomy. Dále pod sebou má vedoucího agronoma a vedoucího zootechnika. Agronom má pod sebou 12 pracovníků v terénu. Zootechnik má pod sebou další 4 zootechniky. Agronom se stará o rostlinnou výrobu a práci deleguje, průběh konzultuje a prezentuje řediteli. Zootechnik se stará o blahobyt a zdravotní stav zvířat, zahrnuje to krmení, veterinářské prohlídky, rození telat a další. Problém občas spočívá v malých zkušenostech některého z pracovníků, kdy vedoucí není vždy k dispozici. Naštěstí při takovém počtu pracovníci fungují skoro jako rodina a jsou si schopni vzájemně pomoci a obohatit své zkušenosti.

#### **Styl řízení**

Převažuje demokratický styl řízení. Pracovníci mají přidělený primárně svůj traktor, o který mají povinnost se starat. Občas dochází k poruchám a z části i vinou pracovníků, což někteří nesnáší nejlépe. Tato práce je náročná na psychiku. Nicméně vedení firmy tyto skutečnosti chápe a snaží se o tom otevřeně komunikovat a netrestat zaměstnance. I přestože opravy nejsou levná záležitost. Vedení firmy zde působí už několik let, a tak je k nim chován respekt, který stejně tak chovají ke svým zaměstnancům. Nicméně občas zde funguje i autoritářský

styl řízení. Názory pracovníků jsou brány v potaz, jelikož sami většinou dobře znají stav pole nebo zvířat, proto jejich názory jsou zohledněny při rozhodování a dalším plánování.

## **Systemy**

Zaměstnanci vykonávají práci každý v danou dobu někde jinde, často samostatně. S traktory jezdí na pole a louky, nebo jsou na středisku a starají se o zvířata nebo údržbu strojů. Důležité jsou proto pro jejich efektivitu práce komunikační media, aby se byli schopni domluvit podle potřeby. Dalšími klíčovými systémy jsou navigační systémy, které sdílí polohu techniky, kolegů a umožňují optimalizaci tras za účelem šetření pohonných hmot. Dále se využívají různé účetní systémy. Bohužel společnost nedisponuje žádným skladovým systémem. Zaměstnanci se řídí zajetými dlouhodobými postupy.

## **Sdílené hodnoty**

Společnosti velice záleží na věrnosti zaměstnanců a jejich spokojenosti. Každý zaměstnanec má možnost zapojit se do nových věcí a nové nápady jsou vítány. Každý pracovník dodržuje standardy v prvovýrobě, jakoby mu byla společnost vlastní. Zaměstnanci jsou často z okolních vesnic nebo měst, často se znají již dříve, fungují podobně jako rodina. Společnost má taky dlouhodobé odběratele a dodavatele, se kterými udržuje kvalitní dlouhodobé vztahy. Tyto vztahy zajišťují jistá privilegia a výhodné ceny.

## **Spolupracovníci**

Pro společnost jsou důležití zaměstnanci v terénu (traktoristé), na kterých úspěch společnosti stojí. Naneštěstí se jedná o hůř placenou pozici v poměru s ostatními v tomto odvětví. Společnost se snaží motivovat různými bonusy a zvýšenou sazbou v období žní. Také jim poskytuje a financuje rozšíření profesních možností, jako jsou kvalifikace na operaci s různými stroji. Spolupracovníci v tomto okolí jsou často lidé, kteří už se znají z prostředí a okolních vesnic, škol a podobně. Všichni si mezi sebou tykají a účastní se běžných zemědělských slavností, jakou například dožínky po úspěšné sklizni.

## **Schopnosti**

V toto odvětví se často něco porouchává a nevychází různé věci podle plánu. Důležitá je psychická odolnost a vnímání, že to tak je. Většina nových mladistvých zaměstnanců do roka vyhoří, kvůli delší pracovní době a pracovním nasazení. Důležitá je se schopnost učit se, přizpůsobovat a být otevřen novým věcem. Technologie se vyvíjí a traktor už není jen traktor, ale těžký stroj s řadou dalších počítačů, senzorů a podobně. Důležité je umět s těmito technologiemi zacházet, aby mohla být práce vykonávána efektivněji. To se týká nejen traktorů. Někteří starší zaměstnanci se nechťejí v různých směrech rozvíjet, protože jsou na něco „zvyklí“. Několik let zkušeností v tomto oboru jsou velmi cenné.

### **4.3.4 SWOT analýza**

#### **Silné stránky**

Silných stránek má společnost hned několik. Jedná se především o dlouhodobé vztahy se dodavateli i odběrateli, kteří jsou velké společnosti, což poskytuje společnosti značný komfort a jistotu. Společnost je soběstačná ve výrobě hovězího masa. Zaměstnanci jsou úzce

spjatí a velmi dobře přijímají novou pracovní sílu a jsou samostatní, což je v tomto oboru velmi dobrá hodnota. Podnik a jeho střediska se nachází v kraji, kde panuje dlouholetá tradice v zemědělství a fungují zde lesnické odborné školy, které mohou potenciálně přivést nové zaměstnance.

### **Slabé stránky**

Slabou stránkou je do jisté míry technologická zaostalost v dílčích oblastech podniku. Společnost například nemá informační systém pro sledování pohybu skladových zásob a informace tak eviduje pomocí papírových přejímek, které následně vykazuje do účetního systému. Jde celkově o zastaralý proces, který má několik slabin. Jsou jimi, potenciální ztráta nebo zpoždění informace, nízká integrita dat, malá časová efektivita. Slabou stránkou jsou taky nízké platy. Také je složité a aktuálně nemá jak přesně kontrolovat činnosti svých zaměstnanců efektivně. Musí být prováděny tak kontroly vedoucím tím, že objíždí a kontroluje aktivitu na pozemcích.

### **Příležitosti**

Příležitostmi pro společnost spočívají primárně v náboru nové mladé pracovní síly. To podporuje demografický vývoj a lokace podniku. Vhodné by bylo navázat spolupráci formou praxí s lesními školami, exkurzemi a podobnými věcmi, které by mohly zvýšit povědomí o společnosti. Je nehezke říci, že válka na Ukrajině společnosti poskytuje různé příležitosti a to formou vývoje cen komodit na trhu, kdy by při rostoucích cenách mohla utržit značný zisk. Příležitosti spočívají i v integraci různých systémů a automatizačních technologií, při dnešním technologickém pokroku. Příležitosti vidí i v dotačních balíčcích.

### **Hrozby**

Rychlost technologických pokroků v kombinaci s nízkou ziskovostí, kterou primárně musí investovat do obnovy a údržby provozu, ohrožuje společnost, kdy by mohla zaostat vůči zahraniční konkurenci. Vyšší míra inflace způsobuje celkový nárůst cen, společnost tak musí své ceny taky reflektovat, hrozí jí potenciální snížení poptávky po některých produktech. Vysoké ceny pohonných hmot zvyšují náklady společnosti na provoz výroby, stejně tak cena energií. Stravovací trendy, ohrožují poptávku po hovězím mase.

#### **4.3.5 Souhrn analýz**

Z analýz vyplývá, že zde existuje několik potenciálních problémů, se kterými společnost není spokojená. V dlouhodobém hledisku tyto jednotlivé problémy ohrožují její pozici na trhu. Aktuální stav je ale primárně nevyhovující v absenci skladového systému. Pracovníci v terénu, kteří skladové operace provádí, postupují pouze podle doporučení. V případě dodržení doporučení předají informace osobě zaznamenávající přejímky. Tyto přejímky jsou periodicky předávány ekonomovi, který je pracně musí přepisovat do účetního systému. Nejsou jasně dané zodpovědnosti, jasně dané postupy, které se musí dodržovat, ale fungují spíše jako doporučení. Proces je zastaralý a obsahuje mnoho chyb a zbytečných kroků. Neexistuje snadný přehled aktuálního stavu skladů.

## Kapitola 5

# Vlastní návrhy řešení, přínosy návrhů řešení

V této kapitole je nejprve popsána navrhovaná změna, která zahrnuje i požadavky na výsledné řešení. Následně pomocí Lewinova modelu je vytvořen návrh na zavedení změny ve společnosti, na kterou poté navazuje analýza rizik pomocí skórovací metody. Současně je provedena časová analýza zavádění změny. Sekce 5.6 obsahuje vlastní řešení. V sekci 5.7 je proveden audit pomocí portálu ZEFIS, jenž je popsán v kapitole 3 v sekci 3.3.5. Nakonec je zhodnocena finanční stránka projektu a přínosy výsledného řešení.

### 5.1 Popis navrhované změny

Na základě souhrnu z předešlých analýz je nutná změna v oblasti řízení skladových zásob společnosti. Je třeba zavést informační systém pro řízení skladových zásob, jelikož společnost žádným aktuálně nedisponuje. Tento informační systém by měl optimalizovat aktuální procesy spojené s řízením skladových zásob, zvýšit bezpečnost a integritu dat.

#### 5.1.1 Požadavky na implementaci

Společnost vyžaduje ve výsledku jednoduchý systém, nevyžaduje komplexní větší systém se spoustou funkcí, jelikož je pravděpodobně nevyužije, navíc zaškolení zaměstnanců by bylo také náročnější. Společnost má aktuální požadavky na systém:

- **Střediska** – Mají různé sklady
- **Sklady** – V rámci různých středisek
- **Materiál** – Konkrétní typ materiálu, který bude možné zvolit při provedení operace na skladě (nejedná se o položku, která je na skladě).
  - Přidání
  - Odebrání
  - Změna
- **Položka skladu** – Položka obsahující typ materiálu, množství.
  - Přidání na sklad

- Odebrání ze skladu
- Předání na jiný sklad
- **Transakce** – Transakce provedená na skladě v konkrétní čas obsahující položku, se kterou bylo manipulováno.
  - Zobrazení v tabulce
  - Filtrování podle data (pro měsíční výkazy)
  - Export
- **Role informačního systému**
  - Pracovník provádějící operace se skladem
  - Vedoucí, ekonom

## 5.2 Lewinův model změny

Na základě analýz a požadavků na informační systém bude pro zavedení tohoto systému použit Lewinův model změny.

### 5.2.1 Fáze rozmrazení

Nejprve jsou informováni zaměstnanci o nadcházející změně a proč je nutná. Proč je nutná, vyplývá z provedených analýz. Budou seznámeni do určité míry s postupem nasazení systému ve společnost a budou se účastnit uživatelského testování a zpětné vazby na systém, pro případná vylepšení. Budou jim osvětleny i přínosy a náležitosti využití informačního systému. Analýzu silového pole lze vidět v tabulce 5.1.

#### Analýza silového pole

Hnací síly		Brzdné síly	
Popis	Síla	Popis	Síla
Eliminace ztráty informací	10	Zaměstnanci nepřijímající nové technologie	-3
Zrychlení procesu operace na skladě	2	Nížší rozpočet na implementaci systému	-8
Zvýšení integrity dat	8	Potřeba zaškolení zaměstnanců	-2
Jasnější zodpovědnost rolí	4	Nízká technologická gramotnost zaměstnanců	-4
Snazší management plánování zásob	6	Problematické testování	-5
Budoucí integrace s dalšími systémy	6	Schvalovací proces nasazení systému	-6
<b>Součet</b>	<b>36</b>	<b>Součet</b>	<b>-28</b>

Tabulka 5.1: Analýza silového pole

Podle tabulky silového pole je zřejmé, že převažují hnací síly, změna je tedy možná. Síly jednotlivých sil byly diskutovány a stanoveny vývojářem systému ve spolupráci s vedením firmy.

## **Role změn**

### **Agent**

Primární agentem této změny je výkonný ředitel, který o zavedení změny inicioval. Koordinuje změnu a je mostem mezi externím vývojářem, který systém vyvíjí a zaměstnanci, kteří jej mají využívat. Je zodpovědný za nasazení systému do produkce, včetně jednotlivých kroků nasazení systému. Zastřešuje spolupráci s vývojářem. Informuje o změně zaměstnance.

Dalším agentem změn je externí vývojář, který má zodpovědnost za splnění požadavků na systém, které vyžaduje výkonný ředitel. Jeho zodpovědnost také tkví ve vytvoření podpůrných materiálů pro užívání systému a školení zaměstnanců. Zaměstnance proškoluje v systému a získává zpětnou vazbu, pro vylepšení systému.

### **Sponzor**

Sponzorem je výkonný ředitel, který je tedy i ze značné části agentem změn. Spolupracuje s dalším agentem změn (vývojářem) a má finanční a lidské zdroje.

Posledním sponzorem změn je IT oddělení koncernu pod kterým společnost funguje. Politika informačních technologií společnosti totiž spadá pod tu, která platí pro celý koncern. Tento sponzor má pravomoc ke schválení systému z hlediska kybernetické bezpečnosti systému.

### **Advokát**

Advokátem změny jsou zaměstnanci, a to jak v terénu, tak ekonomové nebo vedoucí pracovníci. Ti o zavedení změny stojí, kvůli značnému usnadnění práce a jasnějšímu rozdělení zodpovědností v procesu.

## **Intervenční oblasti**

### **Technologie**

Nový systém bude přístupný skrze webové rozhraní. Pracovníci, kteří jej budou používat, využijí služební mobilní telefony, které mají k dispozici. Ekonom bude IS využívat na firemních počítačích nebo notebooku. Osoba vyplňující přejímky bude ušetřena o shromažďování informací o pohybech na skladu, stejně tak pracovníci provádějící operace na skladě nebudou povinni informaci předávat dál. Ekonom v rámci měsíčních výkazů na konci měsíce transakce pouze exportuje a zadá do účetního systému. U některých skladů bude nutné nasadit access pointy pro přístup k internetu.

### **Toky a procesy**

Změna usnadní několik procesů ve firmě. Pracovníci si nebudou muset pamatovat nebo zaznamenávat podle svého uvážení informaci o pohybu na skladě, ale provedenou operaci

na skladě zadají do ISu, čímž jejich zodpovědnost končí. Ekonomovi se velmi ulehčí proces, kdy si v systému vyfiltruje transakce podle data, které pouze exportuje a vloží do účetního systému. Nebude muset přepisovat záznamy z tištěných přejímek jednou měsíčně v rámci výkazů do účetního systému. Nasazením systému by mělo dojít ke zvýšení celkové efektivity a snížení časových nároků těchto procesů, ale i integrity dat.

### **Organizační struktura**

Organizační struktura se nijak nezmění implementací tohoto systému. Pro systém bude dělat údržbu externí vývojář v případě chyb systému, stejně tak o jeho dostupnost.

### **Lidské zdroje**

V průběhu nasazení se zaměstnanci zúčastní školení a uživatelského testování systému. Změní se zde náplně práce, respektive dojde k ujasnění zodpovědností zaměstnanců. Osoba zaznamenávající pohyby do tištěné přejímky bude ušetřena o velké množství práce, jelikož tištěné přejímky budou eliminovány novým systémem. Pracovníci provádějící operace na skladě budou muset tento systém využívat, ale měl by jim spíše ušetřit práci, a to v tom ohledu, že nebudou muset předávat dále informace o pohybu na skladě.

#### **5.2.2 Fáze přechodu a aplikace změny**

V této sekci je zkonstruována tabulka akcí, které je třeba provést pro zavedení změny. Tabulka 5.2 obsahuje jednotlivé akce, které je třeba provést. Tyto akce jsou pak analyzovány v časové analýze v sekci 5.4.

#	Název úkolu
A	Rozhodnutí vedení o implementaci nového skladového systému
B	Vytvoření požadavků na skladový systém
C	Předání požadavků na systém vývojáři
D	Development skladového systému
E	Nákup Access Point zařízení
F	Instalace Access Point zařízení
G	Ukázka zkušební verze systému
H	Úprava specifikace požadavků
I	Zavedení úprav do systému
J	Testování zkušební verze systému
K	Zajištění a konfigurace webového serveru
L	Deployment na webový server
M	Inventura s migrací dat
N	Korektura migrovaných dat
O	Příprava školení zaměstnanců
P	Udělení přístupu zaměstnancům a nastavení rolí
Q	Školení zaměstnanců
R	Spuštění systému pro vybrané zaměstnance
S	Zpětná vazba od uživatelů
T	Implementace změn na základě zpětné vazby uživatelů
U	Finální nasazení systému na produkční server

Tabulka 5.2: Seznam úkolů změny

### 5.2.3 Fáze zmrazení

Cílový stav by měl být takový, že pracovníci v terénu by měli systém běžně využívat bez nějakých zásadních problémů. Systém by měl být dostupný téměř 24/7 a měl by mít minimální střední hodnotu mezi poruchami. Agent změn bude celou situaci průběžně monitorovat, aby zjistil, zda je změna úspěšná.

Metrikou úspěšnosti bude dostupnost systému na produkci – sledují se hodnoty MTBF (Mean Time Between Failure) a MTTR (Mean Time To Repair). Další metrikou bude počet bugů reportovaných v rámci systému jeho uživateli.

Změna bude viditelná spíše interně, dodavatelé ani odběratelé výrazně navenek tuto změnu nepocítí. Jedná se spíše o interní změnu v organizaci.

### 5.3 Analýza rizik

Pro analýzu rizik změny je použita skórovací metoda. Pro pravděpodobnost výskytu rizik je použita tabulka 5.3 a pro měření míry dopadu tabulka 5.4. Nejprve budou identifikovány rizika, tedy potenciální hrozby a jejich scénáře, sekce 5.3.1. Následně jejich ohodnocení dle tabulek 5.3 a 5.4, sekce 5.3.2. Poté bude v sekci 5.3.3 budou provedeny návrhy na opatření. Mapy rizik před a po opatřeních jsou v sekci 5.3.4.

Hodnota	Procentuálně	Slovní hodnocení
1-2	0% - 19%	velmi nepravděpodobná
3-4	20% - 39%	nepravděpodobná
5-6	40% - 59%	pravděpodobná
7-8	60% - 79%	více pravděpodobná
9-10	80% - 100%	velmi pravděpodobná

Tabulka 5.3: Pravděpodobnost výskytu

Hodnota	Slovní hodnocení
1 - 2	Velmi malé
3 - 4	Malé
5 - 6	Střední
7 - 8	Vysoké
9 - 10	Kritické

Tabulka 5.4: Metrika dopadu

### 5.3.1 Identifikace rizik

Rizika byly identifikovány brainstormingem. Vzešlo z toho 10 hrozeb, ve kterých nastane k nim adekvátní scénář, viz tabulka 5.5. Hrozby se týkají zaváděné změny, informačního systému, procesu a kolaborace zaměstnanců.

#	Hrozba	Scénář
1	Systém není dostatečně otestován	Systém bude obsahovat vysokou míru bugů, což zpomaluje nasazení systému.
2	Slabé zabezpečení systému	Při slabém zabezpečení systému by mohlo dojít k neoprávněnému přístupu do systému nebo jeho částem, což by narušilo integritu dat.
3	Špatné přiřazení rolí a pravomocí v systému	Při špatně nastavených pravomocích uživatelů by si pracovník například mohl zobrazovat a exportovat transakce, k čemuž nemá pravomoc.
4	V místě skladů není dostatečná konektivita k internetu	Pracovníci nebudou moci systém užívat u některých skladů
5	Zaměstnanci neumí používat systém	Zaměstnanci mohou zadat chybná data do systému.
6	Systém neobsahuje požadovanou funkcionalitu	Systém nepokryje některé požadavky a nepřinese přidanou hodnotu pro zaměstnance.
7	Úprava a migrace dat nebyla provedena důkladně	Historická data nebudou konzistentní nebo nekompletní, což zbrzdí nasazení systému.
8	Neochota zaměstnanců používat nový systém	Zaměstnanci, zejména starších generací budou bojkotovat práci s novým systémem.
9	Hackerské útoky na systém	Mohou způsobit výpadek systému nebo odcizení dat.
10	Výpadky serveru	Při nedostatečném hardwarové výkonu může systém zaznamenávat výpadky a zaměstnanci jej nebudou moci použít pro běžné úkony.

Tabulka 5.5: Identifikace hrozeb a scénářů

### 5.3.2 Ohodnocení rizik

Na základě předchozí tabulky, kdy byly identifikovány rizika, byla vytvořena následující tabulka 5.6, která ohodnocuje identifikovaná rizika, hodnota rizika nabírá hodnotu od 0 do 100.

#	Hrozba	Scénář	Pst	Dopad	Hodnota rizika
1	Systém není dostatečně otestován	Systém bude obsahovat vysokou míru bugů, což zpomaluje nasazení systému.	3	7	21
2	Slabé zabezpečení systému	Při slabém zabezpečení systému by mohlo dojít k neoprávněnému přístupu do systému nebo jeho částem, což by narušilo integritu dat.	3	8	24
3	Špatné přiřazení rolí a pravomocí v systému	Při špatně nastavených pravomocích uživatelů by si pracovník například mohl zobrazovat a exportovat transakce, k čemuž nemá pravomoc.	4	6	24
4	V místě skladů není dostatečná konektivita k internetu	Pracovníci nebudou moci systém užívat u některých skladů	7	8	56
5	Zaměstnanci neumí používat systém	Zaměstnanci mohou zadat chybná data do systému	8	4	32
6	Systém neobsahuje požadovanou funkcionalitu	Systém nepokryje některé požadavky a nepřinese přidanou hodnotu pro zaměstnance.	5	8	40
7	Úprava a migrace dat nebyla provedena důkladně	Historická data nebudou konzistentní nebo nekompletní, což zbrzdí nasazení systému.	6	8	48
8	Neochota zaměstnanců používat nový systém	Zaměstnanci, zejména starších generací budou bojkotovat práci s novým systémem.	5	3	15
9	Hackerské útoky na systém	Mohou způsobit výpadek systému nebo odcizení dat.	2	9	18
10	Výpadky serveru	Při nedostatečném hardwarové výkonu může systém zaznamenávat výpadky a zaměstnanci jej nebudou moci použít pro běžné úkony.	5	4	20

Tabulka 5.6: Ohodnocení rizik

### 5.3.3 Návrh opatření

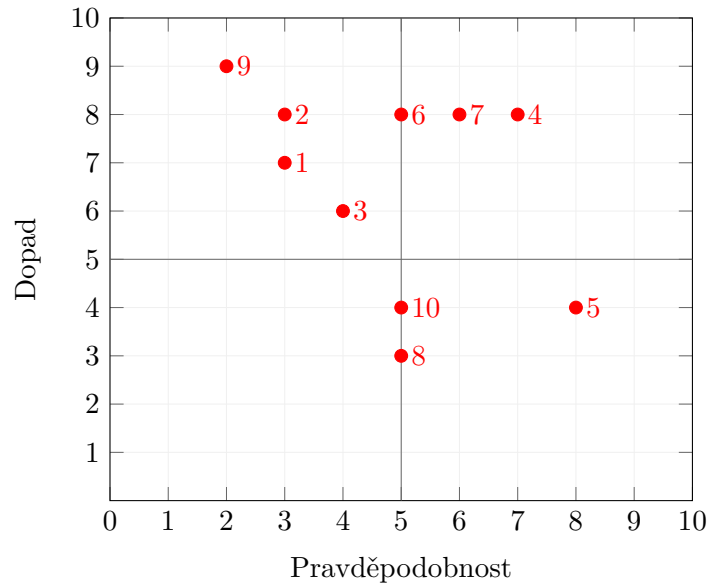
Výsledné hodnoty rizik jsou v celku vysoké, některé zasahují dokonce i do oblasti kritických rizik. Z toho důvodu jsou navrženy opatření pro jednotlivá rizika. Tyto opatření by měly snížit hodnotu rizika tak, že se budou vyskytovat v kvadrantu bezvýznamných rizik.

#	Hrozba	Opatření	Pst	Dopad	Hodnota rizika
1	Systém není dostatečně otestován	Dostatek jednotkových a integračních testů společně s uživatelským testováním a jejich vyhodnocením.	1	3	3
2	Slabé zabezpečení systému	Využívat spolehlivé technologie a ověřené metody zabezpečení IS.	1	4	4
3	Špatné přiřazení rolí a pravomocí v systému	Správně definovat role a jejich pravomoce v požadavcích.	2	3	6
4	V místě skladů není dostatečná konektivita k internetu	Zajistit dostupnost a kvalitu internetového připojení u skladových prostor pomocí access pointů	3	3	9
5	Zaměstnanci neumí používat systém	Dostatečné školení zaměstnanců a vytvoření návodů v ISu.	4	1	4
6	Systém neobsahuje požadovanou funkcionalitu	Řádně specifikovat požadavky, ověřovat při zkušební verzi a testování.	2	2	4
7	Úprava a migrace dat nebyla provedena důkladně	Kontrola dat po migraci, manuální úprava po migraci.	3	2	6
8	Neochota zaměstnanců používat nový systém	Kontrola a motivace zaměstnanců k využívání ISu.	2	1	2
9	Hackerské útoky na systém	Dostatečné zabezpečení na úrovni síťové komunikace, firewall, silná hesla.	2	4	8
10	Výpadky serveru	Zajistit výkonný hardware a dostupnost serveru, automatické restartování.	3	1	3

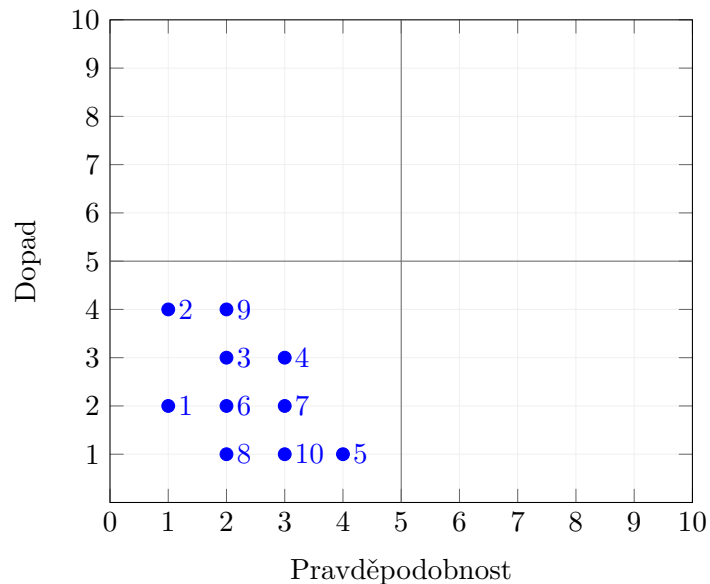
Tabulka 5.7: Návrh opatření

### 5.3.4 Mapy rizik

Na grafu 5.1 lze vidět v jakých kvadrantech se nachází jednotlivá rizika před návrhy na opatření. Jak je vidět, některé se nachází i v kvadrantu kritických rizik, takže by změna mohla být pravděpodobně neúspěšná. Po návrhu opatření rizik jsou všechna rizika v kvadrantu bezvýznamných rizik, zde je třeba dodržovat opatření, aby tomu tak zůstalo, viz graf 5.2.



Obrázek 5.1: Mapa rizik před opatřeními



Obrázek 5.2: Mapa rizik po opatřeních

## 5.4 Časová analýza

Časová analýza je provedena pomocí metody PERT. Seznam činností, jejich návaznost a časové ohodnocení lze vidět v tabulce 5.8. Odhady doby trvání je uveden ve dnech, hodnoty byly stanoveny na základě předchozích zkušeností s vývojem systémů a nasazením do společnosti. Byla zkonstruována tabulka 5.9 a síťový graf, viz obrázky 5.3 a 5.4.

Hodnota  $y_{ij}$  byla stanovena na základě rovnice střední doby trvání činnosti:

$$y_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}$$

- Optimistický odhad:  $a_{ij}$
- Nejpravděpodobnější odhad:  $m_{ij}$
- Pesimistický odhad:  $b_{ij}$
- Střední doba trvání činnosti:  $y_{ij}$

Činnost	Předcházející činnost	$a_{ij}$	$m_{ij}$	$b_{ij}$	$y_{ij}$
<b>A</b>	-	1	2	3	<b>2</b>
<b>B</b>	A	2	4	6	<b>4</b>
<b>C</b>	B	1	2	3	<b>2</b>
<b>D</b>	C	20	35	50	<b>35</b>
<b>E</b>	C	3	4	5	<b>4</b>
<b>F</b>	E	4	7	10	<b>7</b>
<b>G</b>	D	1	2	3	<b>2</b>
<b>H</b>	G	2	3	4	<b>3</b>
<b>I</b>	H	4	8	12	<b>8</b>
<b>J</b>	F, L	6	11	16	<b>11</b>
<b>K</b>	G	2	3	4	<b>3</b>
<b>L</b>	I, K	2	3	4	<b>3</b>
<b>M</b>	J	3	5	7	<b>5</b>
<b>N</b>	M	1	2	3	<b>2</b>
<b>O</b>	J	2	3	4	<b>3</b>
<b>P</b>	J	2	3	4	<b>3</b>
<b>Q</b>	O	2	3	4	<b>3</b>
<b>R</b>	N, P, Q	1	2	3	<b>2</b>
<b>S</b>	R	3	5	7	<b>5</b>
<b>T</b>	S	4	8	12	<b>8</b>
<b>U</b>	T	2	3	4	<b>3</b>

Tabulka 5.8: Seznam činností, jejich návaznost a časové odhady

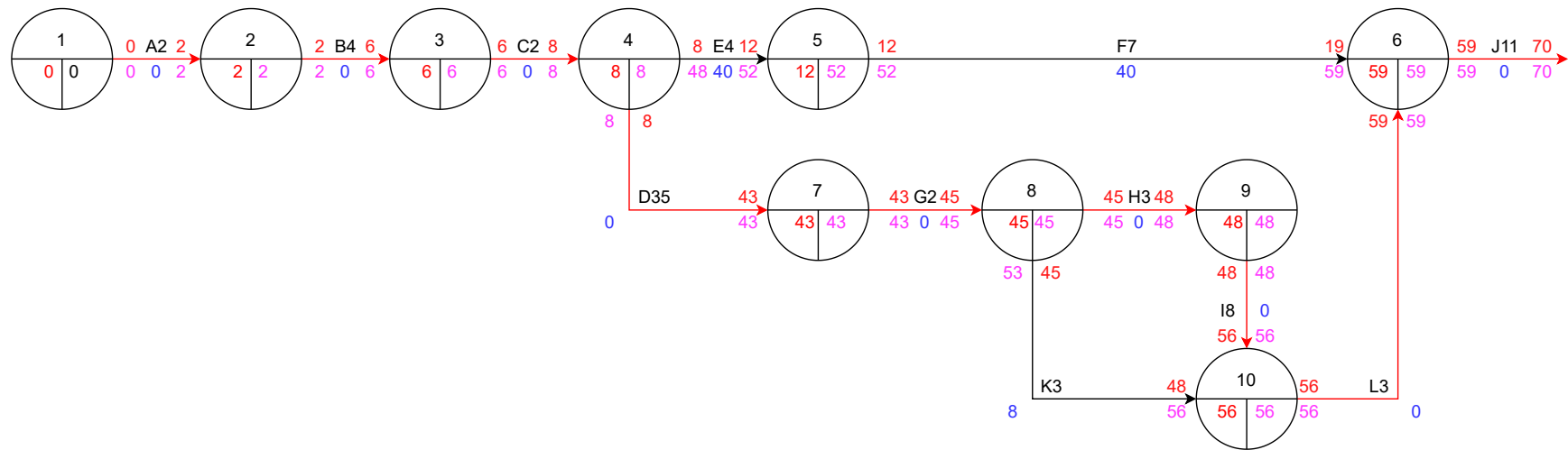
Tabulka PERT obsahuje jednotlivé činnosti, každá činnost obsahuje následující informace mimo ty, které byly zmíněny výše:

- **Výchozí uzel - Konečný uzel:**  $i - j$
- **Začátek možný:**  $ZM$
- **Konec možný:**  $KM$
- **Začátek přístupný:**  $ZP$
- **Konec přístupný:**  $KP$
- **Celková rezerva činnosti:**  $RC$

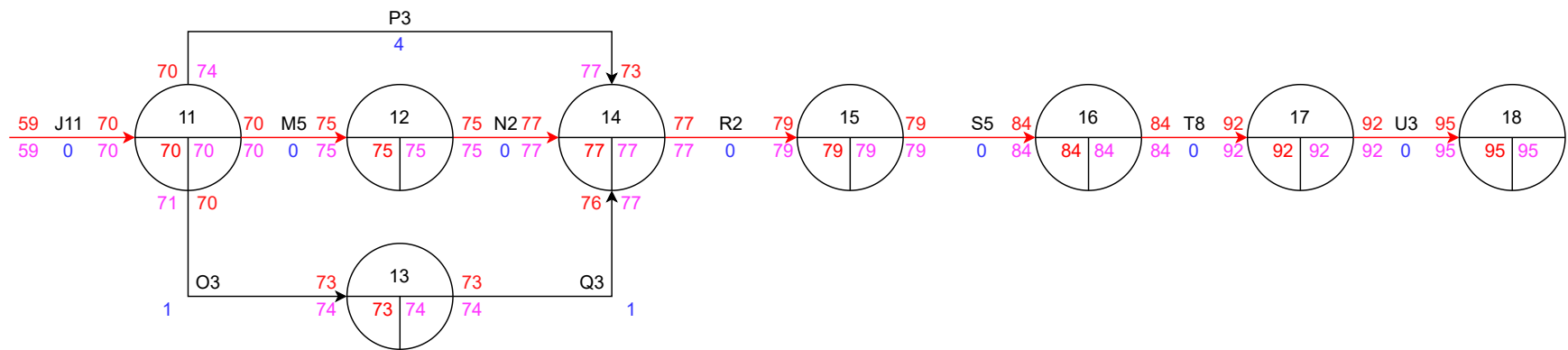
Kritická cesta je vyznačena červeně. Cesta: **A-B-C-D-G-H-I-L-J-M-N-R-S-T-U**. Zavedení změny by mělo trvat 95 dní.

Činnost	i	j	$a_{ij}$	$m_{ij}$	$b_{ij}$	$y_{ij}$	ZM	KM	ZP	KP	RC
A	1	2	1	2	3	2	0	2	0	2	0
B	2	3	2	4	6	4	2	6	2	6	0
C	3	4	1	2	3	2	6	8	6	8	0
D	4	7	20	35	50	35	8	43	8	43	0
E	4	5	3	4	5	4	8	12	48	52	40
F	5	6	4	7	10	7	12	19	52	59	40
G	7	8	1	2	3	2	43	45	43	45	0
H	8	9	2	3	4	3	45	48	45	48	0
I	9	10	4	8	12	8	48	56	48	56	0
J	6	11	6	11	16	11	59	70	59	70	0
K	8	10	2	3	4	3	45	48	53	56	8
L	10	6	2	3	4	3	56	59	56	59	0
M	11	12	3	5	7	5	70	75	70	75	0
N	12	14	1	2	3	2	75	77	75	77	0
O	11	13	2	3	4	3	70	73	71	74	1
P	11	14	2	3	4	3	70	73	74	77	4
Q	13	14	2	3	4	3	73	76	74	77	1
R	14	15	1	2	3	2	77	79	77	79	0
S	15	16	3	5	7	5	79	84	79	84	0
T	16	17	4	8	12	8	84	92	84	92	0
U	17	18	2	3	4	3	92	95	92	95	0

Tabulka 5.9: PERT tabulka



Obrázek 5.3: Síťový graf PERT, část první



Obrázek 5.4: Síťový graf PERT, část druhá

## 5.5 Finanční zhodnocení změny

Tato část shrnuje finanční vyčíslení zaváděné změny. Finanční náklady jsou rozděleny na jednorázové a opakující se.

V rámci jednorázových nákladů hlavní položkou jsou náklady na vytvoření systému, které zahrnují implementaci systému, testování, nasazení do produkce. Další náklady jsou vynaložené na zajištění internetového spojení u některých skladů. To znamená náklady za hardware potřebný ke konektivitě a náklady za instalaci. U některých skladů již připojení je dostatečné. Jako poslední jednorázové náklady budou vynaloženy na školení a dokumentaci.

Do periodických nákladů spadá primárně provoz produkčního serveru. Jak již bylo řečeno, použije se Cloud VPS od společnost Contabo<sup>1</sup>. Tento poskytovatel byl zvolen kvůli dobré dostupnosti, serverům v Německu a nízké ceně. Jako plán je zvolen Cloud VPS S (4 vCPU Cores, 8GB RAM, 200GB SSD). Tento plán vychází na 4.99€ měsíčně, při potřebě výkonnějšího serveru se plán navýší na Cloud VPS M za 8.99€ měsíčně. Další periodické náklady jsou vynaložené na údržbu systému a případné změny – lépe řečeno se jedná o opakující se jednorázové poplatky, firma je zaplatí jen v případě potřeby údržby systému nebo jeho změny.

Náklady	Cena v Kč
<b>Access Pointy</b>	<b>30 000</b>
Hardware	20 000
Instalace	10 000
<b>Vývoj aplikace</b>	<b>103 200</b>
Implementace aplikace	84 000
Testování	19 200
<b>Uvedení do provozu</b>	<b>9 600</b>
Nasazení na produkční server	3 600
Dokumentace	1 800
Školení	4 200
<b>Celkem</b>	<b>142 800</b>

Tabulka 5.10: Jednorázové náklady

Náklady	Perioda opakování	Cena v Kč
Produkční server	měsíčně	~125
Údržba systému	v případě potřeby	4000

Tabulka 5.11: Opakující se náklady

<sup>1</sup><https://contabo.com/en/>

## 5.6 Vlastní řešení pomocí webové aplikace

Na základě popisu navrhované změny v sekci 5.1 a požadavků na informační systém bude vhodné tento systém implementovat jako webovou aplikaci. Webová aplikace byla zvolena z toho důvodu, že byly identifikovány role uživatelů, kteří aplikaci budou používat. Jedna role vyžaduje aplikaci takovou, kterou lze jednoduše použít z mobilního telefonu. Druhý typ role vyžaduje použití běžného uživatelského počítače nebo notebooku. Oba tyto typy zařízení mají v dnešní době zabudované webové prohlížeče v rámci operačních systémů, proto bude vhodné řešení pomocí webové aplikace, která bude mít název „**Skladovac**“.

K vytvoření webové aplikace lze použít mnoho dostupných nástrojů, programovacích jazyků a frameworků. Implementovaný informační systém bude rozdělen na dvě části. První část bude Rest API, které bude představovat back-endovou část aplikace. Tato část bude čistě nezávislá na front-endové části – tedy front-end je možný využít jakýkoliv (desktopovou aplikaci, mobilní, webovou). Druhá část bude front-endová část aplikace, která bude komunikovat se zmíněným API. Ta bude realizován formou webového rozhraní jako Single Page Application (SPA) – tento pojem byl vysvětlen v sekci 3.2.1. Pro persistenci a uchování dat bude použita relační databáze, která bude zastřešovat datovou vrstvu aplikace. Konkrétní detaily použitých technologií budou rozebrány více do detailu později v dalších podsekcích.

Jedná se o nově vytvořenou aplikaci, která bude nasazená formou Cloud VPS. Jako poskytovatel privátního cloudu je společnost Contabo, která nabízí širokou škálu cloudových řešení s rozumným výkonem, za rozumnou cenu. Cena byla zohledněna ve finančním zhodnocení této práce. Bude použito minimální řešení, protože se nepředpokládá velký počet požadavků na aplikaci v rámci jednoho dne. Denně se provede maximálně pár desítek operací na skladě. V případě, kdyby tento hosting nestačil, tak bude využít o úroveň výkonnější Cloud VPS. Servery se nachází pro evropské klienty v Německu, takže odezva bude v celku rychlá.

### 5.6.1 Databáze

Databáze je jedním ze tří pilířů této aplikace. Zastřešuje datovou vrstvu aplikace. Na databázi se přímo dotazuje později vysvětlené implementované API. Pro toto řešení byla vybrána konkrétně open-source relační databáze PostgreSQL, viz sekce 3.2.3.

Databázi lze vytvořit běžným klasickým přístupem modelování a pomocí SQL příkazů, nicméně v rámci tohoto řešení byla databáze vytvořena pomocí Entity Framework Core a jeho nástrojů a příkazů pro vytvoření migrací databáze, pomocí přístupu Code-First. EF Core framework byl popsán v sekci 3.2.5 a Code-First přístup v 3.2.5. Díky tomuto přístupu lze právě vše řídit pomocí programovacího jazyka, vytvořených modelů jako tříd. Také poskytuje kompletní historii a lze se v ní vracet ke starším schémátům databáze v případě zavedení chybné migrace. Pro migraci je automaticky vytvořena vlastní tabulka, kde lze vidět verzi, která byla použita – k tomu je vygenerován i zdrojový kód, dle kterého se migrace vykoná – lze vidět přesné změny. Tento mechanismus je velice výhodný při vývoji aplikací.

Schéma databáze a její migrace jsou součástí projektu, ve kterém se nachází implementace API – tento celek je uchovávan v repozitáři verzovacího systému git (sekce 3.2.7). Tento repozitář je soukromě uchovávaný na straně vývojáře.

V databázi jsou jednak uchovávány primární tabulky, které jsou základem pro aplikaci – jsou to tabulky vytvořené na základě modelů. Dále databáze obsahuje tabulku `RefreshToken`, která uchovává obnovující tokeny pro jednotlivé uživatele – je součástí autentizačního mechanismu, který bude vysvětlen v pozdějších sekcích. Nakonec také obsahuje tedy tabulku, která reprezentuje historii migrací. Toto schéma bylo navrženo na základě požadavků společnosti. Kompletní ER diagram databáze lze vidět na obrázku 5.5.

Mimo kompletní struktury ER diagramu jsou následovně popsány jednotlivé tabulky podrobněji:

- **Center**

Center je tabulka, která reprezentuje středisko firmy. Firma má totiž několik středisek výroby, ve kterých operuje a eviduje je. Nejedná se o tabulku, která by měla obsahovat větší množství záznamů, jelikož firma má jen několik jednotek středisek. Každé středisko má vlastní `id`, které má typ `uuid`<sup>2</sup>, název a lokaci. Lokaci stačí evidovat pouze jako řetězec znaků, nikoliv jako celou adresu, jelikož firma používá vlastní názvy lokací – zaměstnanci tak budou vědět, kde se středisko nachází. Každé středisko může mít několik skladů.

- **Warehouse**

Warehouse je tabulka reprezentující sklad. Je to místo, kde budou uchovávány veškeré položky. Každý sklad je unikátní a patří ke konkrétnímu jednomu středisku. Může obsahovat několik skladových položek 0 až n. Ty přímo nereferencují materiál, tedy tabulku `Item`, ale jsou dostupné skrze inventář, který funguje jako propojovací tabulka. Má své vlastní `id`, název, indikátor, jestli je prázdný nebo ne, a pak tedy cizí klíč, který odkazuje na středisko, ke kterému patří.

- **Item**

Item je tabulka, která uchovává informace o položce. Položka může být právě nějaký materiál (železný klín, ...), surovina (řepka, pšenice, ...) nebo skot. Skot se také bude evidovat jako skladová položka, v tuto chvíli je sklad spíše stáj/kravín. Položka nemá přímou relaci s tabulkou skladu, ale právě přes inventář. Má vlastní `id`, název, měrnou jednotku, jelikož některé položky mohou být v jednotkách kilogramů, litrů nebo se eviduje počet kusů. Má referenci na kategorii, pod kterou spadá.

- **Inventory**

Tato tabulka je propojovací tabulkou skladu a položky. Záznam v této tabulce se vytváří ve chvíli, kdy dojde k operaci na skladě, tj. naskladnění, přesun, vyskladnění a představuje aktuální položku dostupnou na skladě. Obsahuje vlastní `id`, pak `id` položky a skladu. Také obsahuje aktuální množství, které je na skladě.

- **User**

Tato tabulka uchovává informace o uživateli informačního systému. Každý uživatel má vlastní `id`, jméno, příjmení, také email, který používá k přihlašování. Obsahuje

---

<sup>2</sup><https://www.uuidgenerator.net/>

taky čas, kdy byl registrován, telefonní číslo a jestli jeho účet je stále aktivní. V případě odchodu zaměstnance se jeho účet deaktivuje, nemaže se hned, aby mu bylo možné přiřadit transakce. Každý uživatel má vlastní roli, která je důležitou součástí informačního systému, role stanovuje co uživatel může a nesmí. V této tabulce se taky uchovává hashované heslo uživatele.

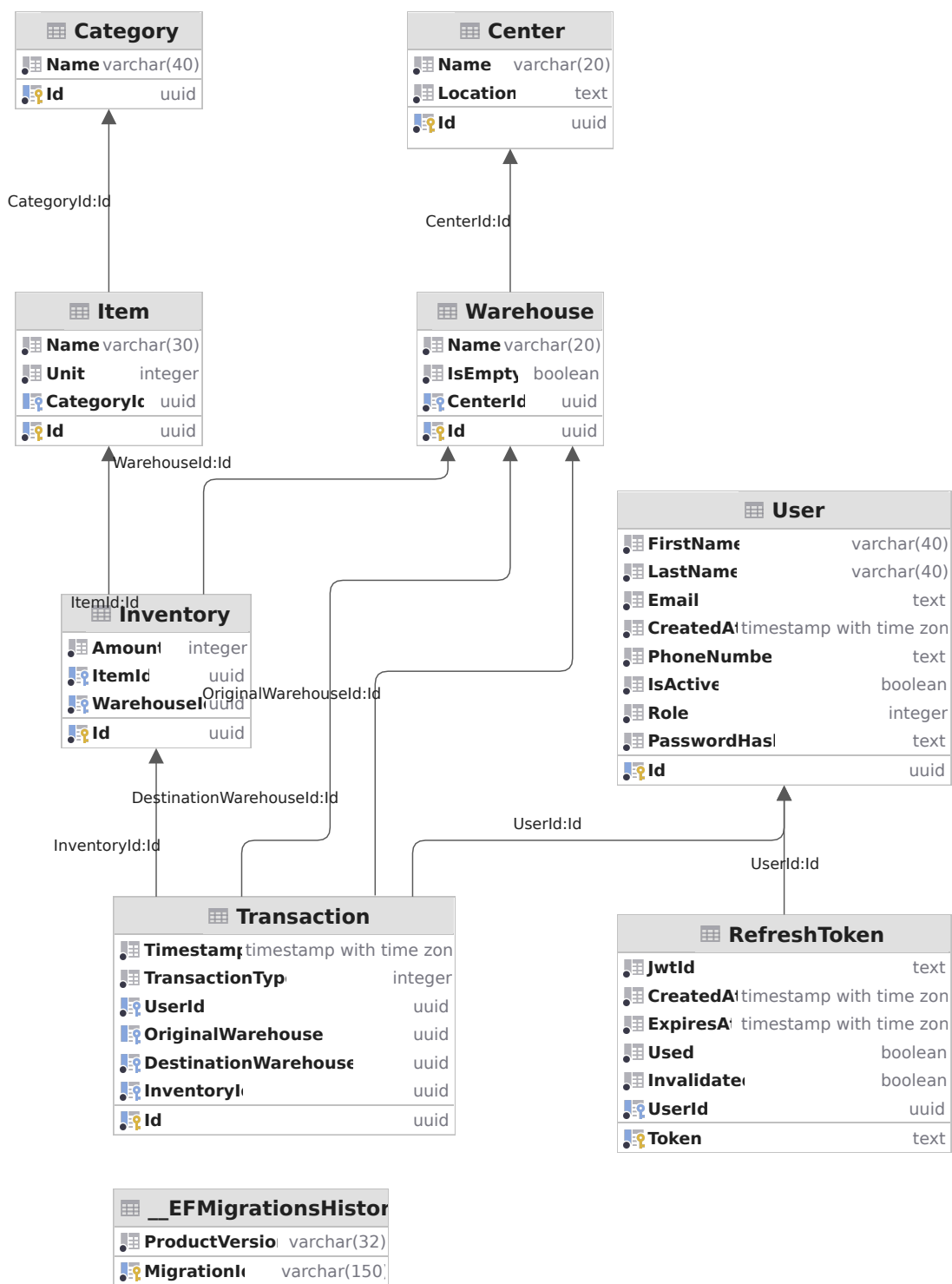
Když uživatel provádí operace na skladě, tak se vždy vytvoří transakce. Má tedy přímou relaci, kdy uživatel může provést 0 až n transakcí.

- **Transaction**

Tabulka transakcí představuje transakci provedenou na skladě. Ta se vytvoří pokaždé, kdy je provedena operace na skladě. Ve chvíli kdy se vytvoří, tak se k ní přidělí aktuální časová značka. Tím je možné mít kompletní historii toho, co se na skladě za celou dobu provedlo. Transakce se váže ke konkrétnímu uživateli, který operaci provedl. Také se váže ke skladu, na kterém operace byla provedena.

Má vlastní id, časovou značku, typ transakce, cizí klíč uživatele a položky na skladě. Výchozí sklad nemusí být vyplněn, ten je vyplněn jen v případě, že položka je přesunuta mezi sklady. Cílový sklad také nemusí být vyplněn, jen v případě, kdy dojde k vyskladnění materiálu.

Zbývající dvě tabulky `_EFMigrationsHistory` a `RefreshToken` zde nebudou podrobněji popsány, jedna obsahuje historii migrací a druhá obnovující tokeny pro JWT autentizaci.



Obrázek 5.5: ER Diagram databáze aplikace

## 5.6.2 Back-end

Další významnou částí aplikace je aplikační rozhraní (API). To je konkrétně implementováno jako ASP.NET Core Web API ve verzi .NET 6. Tento framework je založen na programovacím jazyce C#. Jako Object-Relation Mapper se používá Entity Framework Core. Jako data provider je použit NPSQL<sup>3</sup>, který poskytuje rozhraní pro práci s PostgreSQL relační databází.

Toto API je založené na MVC patternu, využívající protokol HTTP (sekce 3.2.2) pro komunikaci s klientem. Objekty jsou přenášeny a serializovány do formátu JSON (sekce 3.2.4). API je sice založené na MVC patternu, nicméně neobsahuje **View** část, tu bude obsahovat front-endová část. Tedy je implementováno pouze pomocí modelů a kontrolerů.

### Modely

Jak již bylo zmíněno v sekci 5.6.1, modely jsou tvořeny třídami v jazyce C#. Tyto modely jsou pak mapovány na tabulky v databázi. Modely jsou 1:1 stejné s tabulkami, které lze vidět na obrázku 5.5.

### Přístup k databázi

Přístup k databázi je zprostředkován přes třídu `SkladovacDbContext`, který dědí z třídy `DbContext`. Tento kontext se pak používá při tvoření migrací. V modelech a tomto kontextu jsou nastavena veškerá integritní omezení, pro správné užití databáze.

Tento kontext se pak využívá ve třídách, které zastřešují logiku pro vytváření, čtení, aktualizování a odstranění záznamů z databáze pomocí jednoduchých metod. Tyto třídy nepoužívají přímo psané SQL queries, ale metody Entity Framework Core, které poskytují značnou bezpečnost, hlavně proti SQL injection.

### Koncové body

Rozhraní API je postaveno na koncových bodech. Koncový bod představuje URI adresu, což je vstupní bod pro komunikaci s API. Klient se na tyto body dotazuje a API vykoná příslušný kód.

Struktura koncových bodů v této aplikaci je tvořena následovně:

1. Nejprve URI adresa obsahuje ip adresu nebo doménový název serveru.
2. Za první částí následuje řetězec `api/` a jeho verze, aktuálně `v1`.
3. Poslední částí je řetězec konkrétního koncového bodu, například `/items`.

Všechny koncové body zde nebudou vyjmenovány, protože jich je několik desítek. Každý koncový bod je přiřazen metodě v rámci konkrétního kontroleru. Celý mechanismus funguje tak, že na vstupní bod přijde požadavek, zkontroluje se, zda je správný, a pokud je správný,

---

<sup>3</sup><https://www.npgsql.org/>

tak se vykoná příslušný kód a vrátí se odpověď ze serveru. Odpověď je buď úspěšná s nějakými dalšími daty nebo neúspěšná s odůvodněním, k tomu se používají HTTP stavové kódy.

Všechny kontrolery vyžadují, aby uživatel byl autentizovaný a obsahují zabezpečení také pomocí autorizace na základě rolí. Například role **Worker**, tedy pracovník v terénu si nemůže například zobrazit transakce nebo provádět některé editace. Jediné koncové body, které nevyžadují autentizaci ani autorizaci jsou ty, přes které se uživatel registruje nebo přihlašuje.

## Autentizace a autorizace

Autentizace a autorizace je nedílnou součástí této aplikace. Uživatel musí být přihlášen, aby mohl provádět různé akce v rámci tohoto systému. Toto API využívá autentizační schéma pomocí JSON Web Tokenů (JWT), viz sekce 3.2.4.

Registrace jsou omezeny pouze na uživatele, kteří jsou pracovníci v terénu. Ostatní uživatelé jsou vytvářeni přímo vložením do databáze. O přihlašování i registraci se stará **IdentityService**. Při prvním přihlášení uživatele se vygeneruje token a obnovující token. Obnovující token se používá pouze k obnově tokenu, který se zasílá uživateli, obnovuje se jen v případě, kdy už expiroval. Expirace je nastavena na 1 den. Expirace obnovujícího tokenu je pak na 4 týdny, při jeho expiraci se musí uživatel znovu přihlásit. Obnovující token se neposílá uživateli z bezpečnostního hlediska. Token pak uživatel zasílá s každým požadavkem na API, aby mohl přistupovat k místům, ke kterým přístup v rámci role může mít.

Autorizace funguje na úrovni kontrolerů, v případě, že uživatel nemá oprávnění koncový bod využívat, bude mu vrácena odpověď formou HTTP se stavovým kódem 401 **Unauthorized**. API je tak tímto způsobem zabezpečeno proti neoprávněným přístupům. Obecně kontrolery používají atribut, viz 5.2 nebo se specifikovanou rolí, viz 5.1.

```
1 [Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme,  
2 Roles = nameof(UserRole.Economist))]
```

Výpis 5.1: Autorizační atribut se specifikovanou rolí

```
1 [Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme)]
```

Výpis 5.2: Obecný autorizační atribut

## Docker

Součástí projektu API je také **Dockerfile**, který umožňuje pomocí několika jednoduchých příkazů aplikaci kontejnerizovat (sekce 3.2.7). Vytvoří se tedy aplikace s minimální velikostí, optimální efektivitou a lze jej distribuovat na jakékoli zařízení, server nebo cloud.

## Skladové operace

Mimo výše popsané části aplikace je hlavním účelem tohoto API právě logika řízení skladových operací. To je jádrem této aplikace. Aplikace implementuje automatické zazname-

návání operací, které se dějí na skladě, formou transakcí. Transakce je vázaná na konkrétní sklad a provádí ji vždy jen jeden konkrétní uživatel. Proto jsou vytvořeny následující 4 koncové body:

- **[GET]** /api/v1/warehouses/{warehouseId}/inventory  
Vrací pole, které obsahuje položky, jenž jsou aktuálně na skladě s id `warehouseId`.
- **[POST]** /api/v1/warehouses/{warehouseId}/inventory  
Dojde k naskladnění položky na sklad
- **[DELETE]** /api/v1/warehouses/{warehouseId}/inventory  
Dojde k vyskladnění položky na skladě
- **[PUT]** /api/v1/warehouses/{warehouseId}/inventory  
Dojde k přesunu položky mezi sklady

Každá operace se provede buď úplně nebo vůbec, to znamená, že když dojde například k chybě při přesunu položky, tak se nepřesune nic a nevytvoří se ani žádná transakce. Id skladu se získává z cesty, ostatní parametry jsou získávány vždy z query cesty nebo z těla HTTP požadavku.

Skladové operace mohou provádět všechny role, primárně je provádí pracovníci z terénu, ale ekonom nebo vedoucí musí mít taky možnost tyto akce provádět, v případě, že dojde k chybám v záznamech.

Součástí implementace jsou taky filtry jednak na filtrování transakcí například pomocí datového rozmezí, nebo skladů podle středisek apod. Dalším důležitým filtrem, který je implementován, je stránkování. Neposílají se tak všechny záznamy, ale jen po určitém množství, které lze specifikovat v query požadavku. Nepochází tak k přenosu velkého objemu dat v případě velkého množství záznamů a aplikace je tak efektivnější.

### 5.6.3 Front-end

V této sekci je popsán front-end aplikace, jedná se o tu část aplikace, se kterou bude uživatel bezprostředně interagovat. Jak již bylo řečeno, je realizována formou Single Page Application (SPA). Pro implementaci je použita JavaScriptová knihovna React. Tato část je realizovaná jako samostatný projekt a jeho oddělený od back-endu, pouze zaslá požadavky na jeho koncové body, tváří se tedy jako klient.

Dle požadavků, je potřeba aby aplikace byla responzivní. Byla použitelná jak pro mobilní zařízení, tak pro běžné počítače. Proto byla zvolena webová aplikace. V následujících sekcích budou popsány jednotlivé hlavní části aplikace, včetně jejich ukázek.

Aplikace je jednostránková, má však pevně dané rozložení s hlavičkou a bočním menu, pomocí kterého se lze v aplikaci orientovat a dynamicky tak načítat patřičný obsah. Toto menu je sbalené v mobilní verzi, uživatel si jej tak musí pomoci „hamburger“ menu zobrazovat. V menu jsou pak odkazy, kterými se uživatel může přepínat mezi středisky, sklady, materiály a transakcemi. Každý typ uživatele však vidí jen stránky, ke kterým má oprávnění. V případě, že by se uživatel pokusil zadat nějakou cestu do url pole, tak mu bude zobrazena stránka, že neexistuje a dostane se zpět na původní stránku. Toto je jedno ze

zabezpečení aplikace. Pokud uživatel není přihlášen má pouze k dispozici stránku přihlášení nebo registrace.

## Komponenty

Jelikož je k implementaci použit React, tak veškeré grafické elementy jsou realizovány pomocí komponent. Ty jsou tvořeny JSX kódem, kde jsou použity klasické HTML elementy. Pro stylování vzhledu komponent není použito čisté CSS, ale framework Tailwind CSS, viz sekce 3.2.6. Komponenty se renderují dynamicky a pouze při změně vstupních parametrů. Aplikace je tak rychlá a poskytuje příjemnou zkušenost během používání aplikace. Design je ucelený a jednoduchý. Primárně jsou využity komponenty jako tabulky a formuláře, jiné komplexnější komponenty nejsou v tuto chvíli vyžadovány.

## Struktura

Nyní budou představeny jednotlivé části aplikace a toho, jak s nimi uživatel bude interagovat.

## Přihlášení

Přihlašovací stránka je první stránka, která se uživateli zobrazí. Systém nelze používat bez přihlášení.



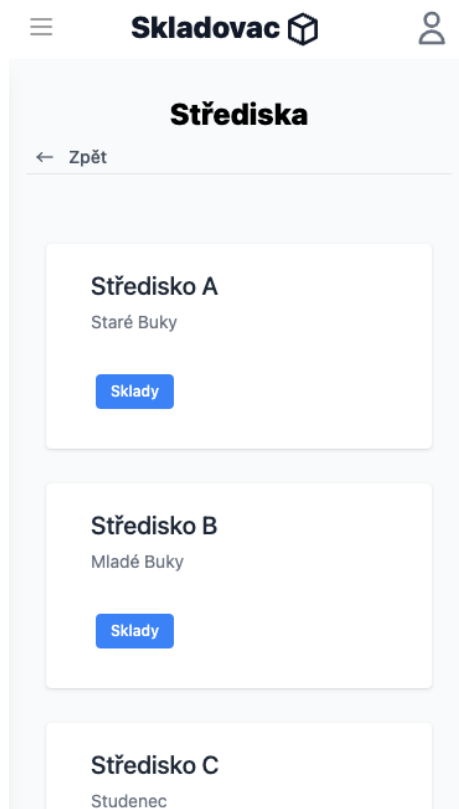
The image shows a mobile login page for an application named 'Skladovac'. At the top, the application name 'Skladovac' is displayed next to a cube icon, and a user profile icon is in the top right corner. Below this, there is a light blue rounded rectangle containing two input fields. The first field contains the email address 'worker@test.com'. The second field contains a masked password represented by six dots. Below the input fields is a prominent blue button with the white text 'Přihlásit se' (Log in).

Obrázek 5.6: Přihlašovací stránka (mobilní verze)

## Střediska

Po přihlášení je uživatel přesměrován na stránku středisek, neohledě na roli uživatele. Jedná se o stránku, ke které mají přístup všichni. Jedná se o seznam středisek, jsou realizovány jako karta. Každé středisko má název a svoji lokaci. Také má tlačítko **sklady**, které uživatele přesměruje na stránku se sklady patřící jednomu zvolenému středisku.

Střediska nelze z uživatelského rozhraní nijak přidávat, upravovat nebo odebírat. Jedná se o entitu, která je víceméně stálá, jejich počet není velký a společnost úpravy nevyžaduje. Byly vygenerovány při nasazení aplikace přímo do databáze.



Obrázek 5.7: Seznam středisek (mobilní verze)

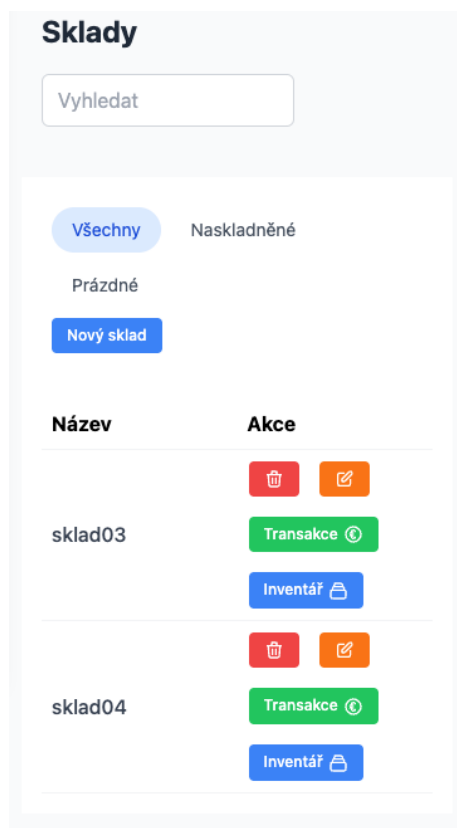
## Sklady

Sklady si lze zobrazit právě skrze konkrétní středisko, jak bylo popsáno výše. V takovém případě jsou vidět sklady pouze jednoho zvoleného střediska. Pokud si uživatel chce zobrazit všechny sklady, tak se k nim může navigovat skrze boční menu (hamburger menu v mobilní verzi). Zobrazení je realizováno formou tabulky. Sklady konkrétního střediska lze vidět na obrázku 5.8.

Jsou zde jednoduché filtry, kdy si lze zobrazit všechny, prázdné nebo naskladněné sklady. Sklad lze i vyhledat na základě názvu.

Každý sklad obsahuje různé akce. Lze si zobrazit transakce konkrétního skladu, nebo právě jeho inventář, tedy to co je aktuálně skladem. Lze ho odstranit, editovat nebo přidat. Pra-

covník v terénu má k dispozici pouze tlačítko, který jej přesměruje na inventář konkrétního skladu.



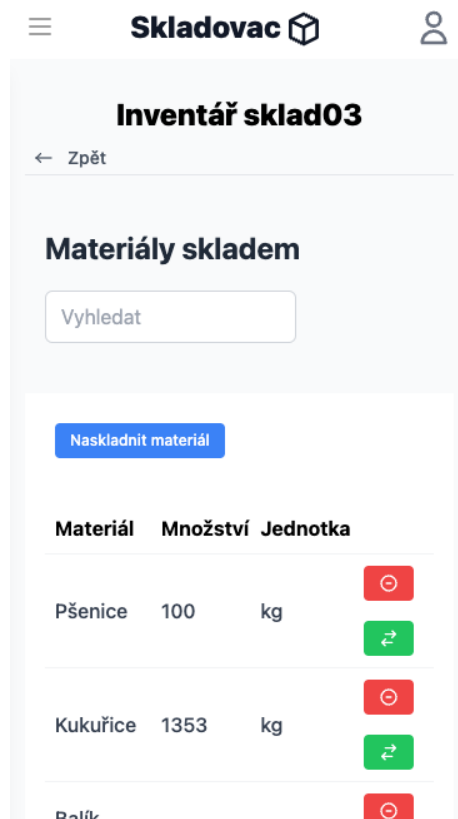
Obrázek 5.8: Seznam skladů střediska (mobilní verze)

### Inventář skladu

Inventář skladu představuje aktuální stav skladu. Inventář je zobrazen vždy pro konkrétní sklad, nikdy pro všechny. Na inventář skladu se lze navigovat pouze skrz seznam skladů. Zobrazení je také realizováno formou jednoduché tabulky.

Vzhledem k tomu, že firma má jen určité množství typu materiálu a nepotřebuje uchovávat větší komplexnější sestavy, tak lze filtrovat materiály na skladě jen na základě názvu materiálu.

Inventář skladu obsahuje tlačítko, které přesměruje uživatele na formulář, skrze který může naskladnit nový materiál. Dále každá položka skladu také obsahuje dvě akce, které přesměrují uživatele na příslušný formulář, jedná se o přesun položky mezi sklady a vyskladnění položky. U naskladnění položky může uživatel vybírat jen ze seznamu evidovaných materiálů, zadá množství a poznámku. U přesunu vybere ve formuláři ze seznamu příslušný sklad a přenášené množství. U vyskladnění zadá množství a poznámku.



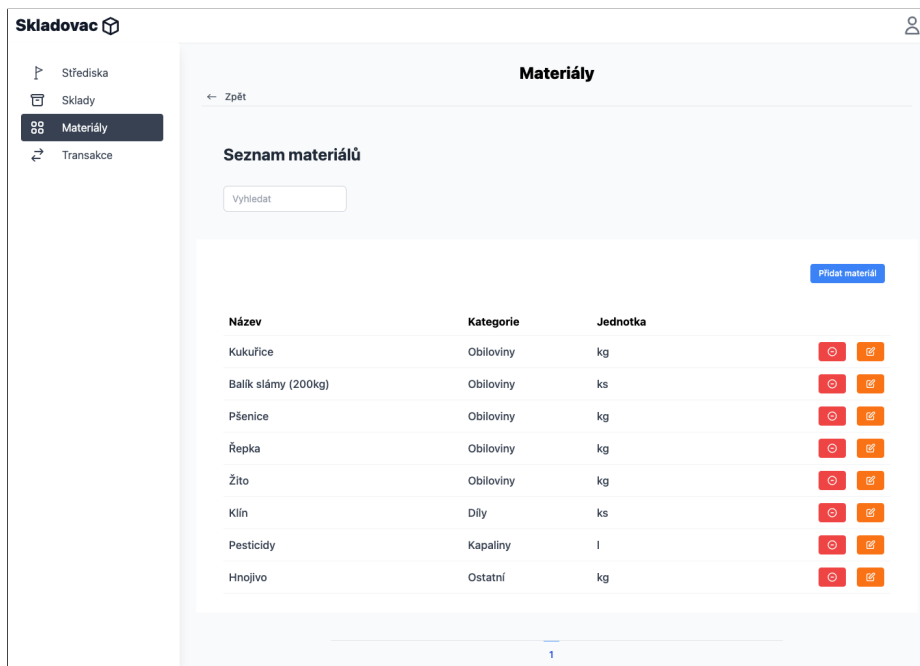
Obrázek 5.9: Inventář skladu (mobilní verze)

## Materiály

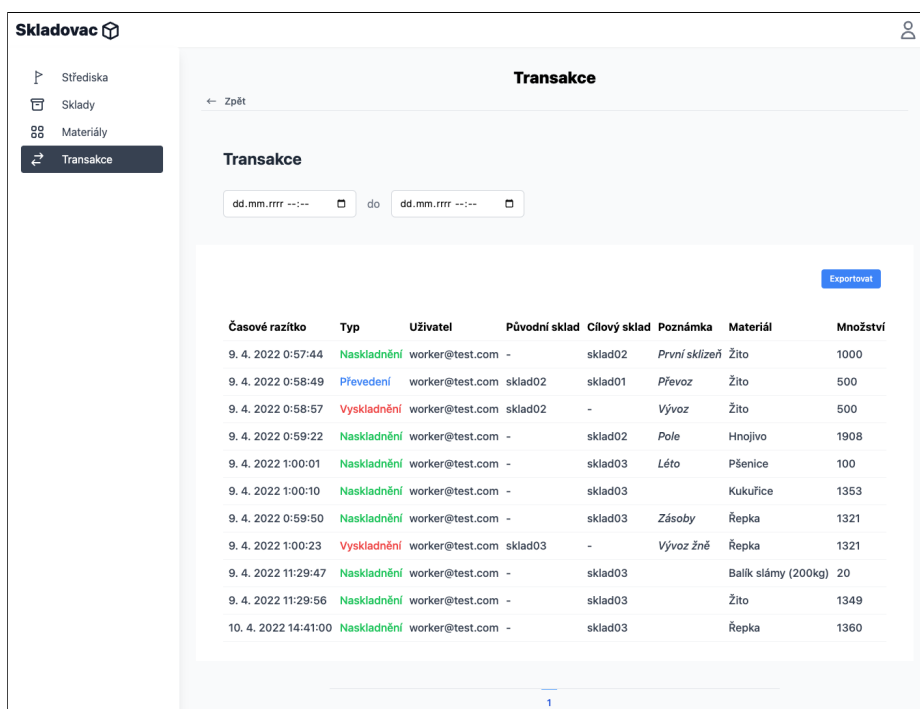
Tato stránka obsahuje seznam materiálů obecně, spadá do toho krmivo, suroviny, polotovary, kapaliny a další. Jsou zde záznamy materiálů, které firma běžně používá. Pro běžného pracovníka provádějící operace na skladě je tento seznam pouze pro čtení, ostatní role jako ekonom, vedoucí je mohou editovat, přidávat, odstraňovat. Stránku materiálu lze vidět na obrázku 5.10.

## Transakce

Tato stránka je určena přímo pro ekonoma, který potřebuje vytvářet na základě měsíčních transakcí přehledy. Může si vyfiltrovat konkrétní měsíc a data exportovat do CSV formátu. S exportovanými daty již pracuje podle své potřeby. Pracovník v terénu tuto stránku má jen pro čtení. Stránku transakcí lze vidět na obrázku 5.11.



Obrázek 5.10: Materiály



Obrázek 5.11: Provedené transakce na skladech

## 5.7 Audit výsledného řešení pomocí portálu ZEFIS

Po implementaci systému byl proveden jeho audit pomocí portálu ZEFIS, viz sekce 3.3.5.

### 5.7.1 Nedostatky, doporučení a řešení vybraného systému

Výsledek auditu systému je v celku solidní, až na nějaké chyby vysoké významnosti, které je ale relativně snadné napravit v rámci aktuálního řešení. Nejedná se o nic co by mělo v současné době příliš velký dopad na fungování systému, nicméně je třeba je primárně vyřešit.

#### Nízká kvalifikace pracovníků při práci s počítači - vysoká významnost

Podle hodnocení auditu se jedná o silnou slabou stránku pro firmu. Oblasti, ve kterých to působí problémy jsou: vyšší náklady na podporu, vyšší prostoje v práci, vyšší chybovost dat v ISu. Aktuálně jsou sice pracovníci v terénu převážně mladí lidé od 20 do 30 let, ale nachází se zde i starší, kteří tolik s mobilním telefonem příliš neumí. V rámci firmy má každý služební telefon a sadu aplikací, pro komunikaci mezi kolegy a další věci. Nicméně se stává, že se naučí jen několik úkonů a při nestandardní situaci, při chybě a dalších podobných případech, nemusí řešit daný problém, a tak jej jednoduše hodí za hlavu.

Nejdůležitějším doporučením je proškolení pracovníků na informační systém, to bude aplikováno i jako řešení. To by mělo probíhat při jeho zavedení a pokaždé, kdy nastoupí nový zaměstnanec, tak by měl být proškolen. V rámci školení by byla i demonstrace užití systému, aby to daný zaměstnanec viděl přímo v praxi. Ve chvíli, kdy by daný systém vůbec neznali a tápali v něm, tak se vystavujeme riziku chybně vložených dat.

#### Nastavení přístupových práv - vysoká významnost

Zde může být problém nastavování přístupových práv příliš pozdě. Ale také může být problémem zrušení práv, respektive deaktivace účtů, při výstupu zaměstnance. Uživatelé se mohou dostat do situace, kdy potřebují systém nezbytně využívat, ale nemohou se dostat k určitým věcem. Při výstupu mohou naopak napáchat nějaké škody, změnit informace apod.

Důležité je pečlivě prověřit či doplnit procesy změn přístupových práv tak, aby nemohlo dojít k opomenutí v nastavování a odebírání práv pracovníků. Tato nastavení je občas potřeba podrobit kontrole, zda odpovídají reálnému stavu. Nelze připustit, aby měl pracovník přístup k datům, která se ho netýkají a není tedy důvod, aby s nimi pracoval jakoukoli formou. Přístupy do informačního systému budou tedy zakomponovány do vstupní a výstupní prověrky zaměstnance.

#### Nejsou aktualizovaná hesla uživatelů - vysoká významnost

Systém aktuálně trpí nedostatkem a to je ten, že uživatelé nemusí měnit pravidelně svá hesla. Uživatelé, kteří mají tedy slabší hesla, mohou znamenat hrozbu pro celý systém. Útočník může snadno tohoto vstupního bodu využít a heslo různými metodami prolomit.

Systém by měl vyžadovat periodické aktualizování hesel, přičemž by tato hesla mít minimální požadavky, jako je například minimální počet znaků, speciální znak, číslo. Dostatečně

silné heslo mívá větší počet znaků, například mezi 10 až 20. Aktuální systém vyžaduje silná hesla, expirace hesel pro uživatele lze dodatečně implementovat do stávajícího řešení snadno, přidá se záznam expirace do tabulky při registraci uživatele a po expiraci se vyžádá nové heslo. Zmíněné řešení by mělo trvat jeden den a náklady jsou jen mzda vyplacená vývojáři.

## 5.8 Budoucí vylepšení

V implementovaném systému je prostor pro zlepšení, také díky zvolené architektuře. V této sekci jsou navrženy možná budoucí vylepšení systému jako samotného a systému z pohledu integrace s ostatními systémy. Další integrace a vylepšení budou možná, až ve chvíli, kdy implementovaná aplikace bude spolehlivě fungovat na produkci v následujících měsících.

### 5.8.1 Mobilní aplikace

Jelikož je aplikace rozdělena na část webového API a front-endovou část, tak API je ve výsledku samostatná aplikace, na kterou lze napojit jakýkoliv front-end řešení. Lze tak vytvořit mobilní aplikaci jak na iOS, tak na Android, která by komunikovala s API.

Další implementace v podobě mobilní aplikace by byla relativně rychlá, jelikož by šlo pouze o front-end část, která by se dotazovala již na stávající API. Náklady by tvořily primárně mzdu vyplacenou vývojáři aplikace, přičemž implementace takové aplikace by mohla trvat zhruba okolo dvou až tří týdnů. Pokud by byla vyvíjena pro android, tak je snadné je dostat do obchodu s aplikacemi, v případě iOS systému by se náklady rozšířily o poplatky na deployment aplikace do App Store.

### 5.8.2 Integrace s dalšími systémy

Implementované API lze použít pro integraci s dalšími systémy. Aktuálně lze z aplikace exportovat měsíční transakce do formátu CSV. S těmito daty ekonom dále pracuje, kontroluje, upravuje a následně zadává do účetního systému, který aktuálně používají. Zde je právě prostor pro zlepšení, kdy automatizací a integrací systému s účetním systémem by mohlo ještě proces tvoření měsíčních výkazů pro ekonomu zlepšit. Řešení a výše nákladů by závisely na možnostech, které ostatní systémy poskytují v rámci jejich integrace. Některé systémy mají dostupná API, takový případ by byl ideální.

## 5.9 Přínosy výsledného řešení

Hlavním problémem firmy byla absence informačního systému v oblasti řízení skladových zásob. Nově implementovaný systém má několik pozitivních přínosů pro firmu.

Jelikož zde chyběl jakýkoliv systém pro řízení zásob, tak zde existovala velká spousta problémů spojená s prováděním operací na skladě. Aplikace velmi usnadňuje procesy pro všechny zaměstnance, primárně pro pracovníky, kteří operace na skladě provádí. Tento pracovník již nemusí nadále předávat informaci dalšímu zaměstnanci, který má na starosti inventuru skladů a zápis pohybů na skladě do tištěných přejímek. Zaměstnanec, který zapisoval pohyby na skladě do tištěných přejímek, nemusí již tyto informace ugovat od pracovníků a zapisovat je. Rovněž nemusí provádět tak často inventuru. Je také zbaven zodpovědnosti předat včas přejímku před koncem měsíce ekonomovi. Ekonom má také

usnadněnou práci, jelikož nemusí urgovat předání přejímky nebo spoléhat na její předání na konci měsíce. Z této přejímky vždy musel záznamy přepisovat do účetního systému ručně, což mu zabralo velké množství času, jelikož přejímky obsahovaly desítky, někdy stovky záznamů. Nyní data pouze exportuje do CSV souboru, se kterými může pracovat dál podle své potřeby, většinou je ještě kontroluje, upravuje a poté zadává do účetního systému.

Došlo tedy k významnému zefektivnění procesů. Díky tomuto systému jsou mnohem lépe stanoveny zodpovědnosti. Nebylo jasné, kdo má nést přímou zodpovědnost za předání informace o pohybu na skladě. Zda to měl být pracovník, který operaci provedl, nebo osoba zapisující záznamy do přejímky. Aktuálně za použití informačního systému tato zodpovědnost připadá pracovníkovi, který operace provádí. Osoba přepisující záznamy do přejímek je zbavena zodpovědnosti úplně, do systému má přístup čistě kvůli kontrole při inventurách. Ekonomovi zodpovědnost zůstává stejná, jen již nemusí přebírat přejímky.

V rámci lepší efektivity procesů a zbavení zodpovědnosti v procesech některých zaměstnanců je hlavním přínosem ušetření času všech zaměstnanců. Pracovník, který provede operaci na skladě ji zaznamená snadno pomocí mobilního telefonu během několika vteřin. Obecně je přínosem lepší tok informací mezi zaměstnanci. Všichni, kteří mají do systému přístup vidí aktuální stavy skladů a provedené transakce.

Nyní lze zjistit aktuální stavy skladů pouhým vstupem do systému. Je zvýšena integrita dat a dochází také k menší chybovosti v souvislosti s aktuálním stavem skladů. Dříve pokud někdo informaci nepředal, což se naneštěstí občas dělo, docházelo při inventurách k zjištění, že skladové zásoby neseďí.

Aplikace je vcelku jednoduchá, uživatelům by tedy nemělo dělat problém ji používat. Z finančního hlediska je v aktuální situaci těžké říci, zda dojde k nějakému markantnímu ušetření nákladů, předpokládá se to, ale nebyl to hlavní důvod, proč byla aplikace vyžadována.

## Kapitola 6

### Závěr

Cílem této práce bylo navrhnout a implementovat softwarovou aplikaci ve zvolené firmě. Pro tuto práci byl zvolen zemědělský podnik, který se zabývá rostlinnou a živočišnou prvovýrobou.

Práce se opírá o teoretickou část, kde jsou popsány veškeré nástroje, metody, postupy a technologie, která tato práce využívá. Aby byl tento projekt realizovatelný a úspěšný, tak byly nejdříve provedeny strategické analýzy podniku a analýza současného řešení. Strategické analýzy byly provedeny pro obecné okolí firmy, oborové okolí a z hlediska vnitřních faktorů firmy. Z analýz vyplynulo několik nedostatků ve firmě, kterým by měla být věnována pozornost. Tato práce se zaměřila na nedostatek v oblasti řízení skladových zásob.

Procesy byly zastaralé, k evidenci záznamů jednotlivých pohybů se používala tištěná přejímka a docházelo ke ztrátě nebo zpoždění informací. Nebyly jasně dané zodpovědnosti a předávání informací bylo zdlouhavé. Také nebyla dostupná přesná historie všech provedených operací s konkrétním časem, ale jen v rámci dní. Nedostatkem bylo také pracné přepisování záznamů z přejímek při měsíčních výkazech, které musel ekonom provádět.

Na základě analýz bylo navrženo řešení pomocí webové aplikace. Změna byla zavedena pomocí Lewinova modelu změn, přičemž byla provedena analýza rizik pomocí skórovací metody. Časový plán byl vypracován metodou PERT. Poté byla popsána implementovaná aplikace, její možná budoucí vylepšení a přínosy pro firmu. Aplikace byla nasazena do firmy, přičemž splňuje požadavky firmy jako zadavatele. Aplikace zefektivňuje některé procesy spojené s řízením skladových zásob, šetří čas, je bezpečná a zajišťuje integritu dat.

# Literatura

- [1] JUREVICIUS, O. *McKinsey 7S Model* [online]. Strategic Management Insight, 2021 [cit. 14. dubna 2022]. Dostupné z: <https://strategicmanagementinsight.com/tools/mckinsey-7s-model-framework/>.
- [2] LUTTIN, D. R. R. A. S. *Overview to ASP.NET Core* [online]. Microsoft, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0>.
- [3] *Code First to a New Database* [online]. Microsoft, 2020 [cit. 18. dubna 2022]. Dostupné z: <https://docs.microsoft.com/en-us/ef/ef6/modeling/code-first/workflows/new-database>.
- [4] LARKIN, R. A. K. *Tutorial: Create a web API with ASP.NET Core* [online]. Microsoft, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio>.
- [5] *CSS basics* [online]. Mozilla Foundation, 2020 [cit. 18. dubna 2022]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics).
- [6] ELLINGWOOD, J. *What are databases?* [online]. Prisma.io, 2022 [cit. 17. dubna 2022]. Dostupné z: <https://www.prisma.io/dataguide/intro/what-are-databases>.
- [7] ELLINGWOOD, J. *What are database migrations?* [online]. Prisma.io, 2022 [cit. 17. dubna 2022]. Dostupné z: <https://www.prisma.io/dataguide/types/relational/what-are-database-migrations>.
- [8] ELLINGWOOD, J. *Comparing database types: how database types evolved to meet different needs* [online]. Prisma.io, 2022 [cit. 17. dubna 2022]. Dostupné z: <https://www.prisma.io/dataguide/intro/comparing-database-types>.
- [9] *What is Database Management System (DBMS)? – FAQ, Types, and Details* [online]. erp-information.com, 2022 [cit. 17. dubna 2022]. Dostupné z: <https://www.erp-information.com/database-management-system.html>.
- [10] *Use containers to Build, Share and Run your applications* [online]. Docker Inc., 2022 [cit. 18. dubna 2022]. Dostupné z: <https://www.docker.com/resources/what-container/>.
- [11] *Dockerfile reference* [online]. Docker Inc., 2022 [cit. 18. dubna 2022]. Dostupné z: <https://docs.docker.com/engine/reference/builder/>.

- [12] *Struktura dotačních zdrojů* [online]. Ministerstvo zemědělství, 2022 [cit. 12. dubna 2022]. Dostupné z: <https://eagri.cz/public/web/mze/dotace/>.
- [13] *Entity Framework Core* [online]. Microsoft, 2021 [cit. 18. dubna 2022]. Dostupné z: <https://docs.microsoft.com/en-us/ef/core/>.
- [14] RANJAN, R. *What is a Framework in Programming & Why You Should Use One* [online]. net solutions, 2021 [cit. 16. dubna 2022]. Dostupné z: <https://www.netsolutions.com/insights/what-is-a-framework-in-programming/>.
- [15] *Git Branches: List, Create, Switch to, Merge, Push, & Delete* [online]. Noble Desktop, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://www.nobledesktop.com/learn/git/git-branches>.
- [16] HANA, S. P. K. *Informační systémy v podnikové praxi: 2. vydání*. Brno: Computer Press, 2000. ISBN 978-80-251-2878-7.
- [17] HARDCASTLE, E. *Business Information Systems*. Ventus Publishing ApS, 2008. ISBN 978-87-7681-463-2.
- [18] *HTML: HyperText Markup Language* [online]. Mozilla Foundation, 2020 [cit. 18. dubna 2022]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
- [19] *An overview of HTTP* [online]. Mozilla Foundation, 2020 [cit. 18. dubna 2022]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>.
- [20] *Ceny výrobců v zemědělství* [online]. Český statistický úřad, 2022 [cit. 12. dubna 2022]. Dostupné z: [https://www.czso.cz/csu/czso/ceny\\_vyrobcu\\_ekon](https://www.czso.cz/csu/czso/ceny_vyrobcu_ekon).
- [21] JANKOVÁ, Z. *Časová a síťová analýza: IV. Metoda PERT* [cvičení]. prezentace. 2022 [cit. 16. dubna 2022].
- [22] JANKOVÁ, Z. *Riziková politika: III. Řízení rizik* [cvičení]. prezentace. 2022 [cit. 16. dubna 2022].
- [23] *JavaScript basics* [online]. Mozilla Foundation, 2020 [cit. 18. dubna 2022]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics).
- [24] *Introducing JSON* [online]. json.org, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://www.json.org/json-en.html>.
- [25] *Introduction to JSON Web Tokens* [online]. jwt.io, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://jwt.io/introduction>.
- [26] *How to secure your LoopBack 4 application with JWT authentication* [online]. loopback.io, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://loopback.io/doc/en/lb4/Authentication-tutorial.html>.
- [27] ÚŘAD VLÁDY ČESKÉ REPUBLIKY. *Vnitřní trh zemědělství* [online]. Euroskop.cz, 2022 [cit. 12. dubna 2022]. Dostupné z: <https://euroskop.cz/evropska-unie/politiky-eu/vnitri-trh/zemedelstvi/>.

- [28] MALLYA, T. *Základy strategického řízení a rozhodování*. 1. vyd. Praha: Grada, 2007. Expert. ISBN 978-80-247-1911-5.
- [29] MICROSOFT. *The history of C#* [online]. Microsoft, 2021 [cit. 20. prosince 2021]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history>.
- [30] MICROSOFT. *A tour of the C# language* [online]. Microsoft, 2021 [cit. 20. prosince 2021]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
- [31] SMITH, S. *Architect Modern Web Applications with ASP.NET Core and Azure* [online]. Microsoft Developer Division, .NET, and Visual Studio product teams, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>.
- [32] WOLF, P. *Jaké mzdy nařizuje stát v zemědělství?* [online]. KupníSíla.cz, 2013 [cit. 13. dubna 2022]. Dostupné z: <https://www.kupnisila.cz/zarucene-mzdy-zemedelstvi/>.
- [33] *The PESTLE business analysis technique* [online]. Knowledge Train @, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://www.knowledgetrain.co.uk/business-analysis/business-analyst-course/business-analysis-technique-pestle>.
- [34] *Ceny pohonných hmot v ČR* [online]. Český statistický úřad, 2022 [cit. 12. dubna 2022]. Dostupné z: <https://www.czso.cz/csu/czso/setreni-prumernych-cen-vybranych-vyrobkou-pohonne-hmoty-a-topne-oleje-casove-rady>.
- [35] *Porter's 5 Forces* [online]. Investopedia, 2020 [cit. 14. dubna 2022]. Dostupné z: <https://www.investopedia.com/terms/p/porter.asp>.
- [36] *PORTER ANALÝZA* [online]. EVOLUTION MARKETING, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://www.evolutionmarketing.cz/marketingovy-slovník/porter-analyza/>.
- [37] *What is PostgreSQL?* [online]. The PostgreSQL Global Development Group, 2022 [cit. 18. dubna 2022]. Dostupné z: <https://www.postgresql.org/about/>.
- [38] PRETTYMAN, S. *Learn PHP 7 : object oriented modular programming using HTML5, CSS3, Javascript, XML, JSON, and MySQL*. Berkeley, CA New York, NY: Apress, 2015. ISBN 978-1-4842-1730-6.
- [39] *Vývoj ceny pšenice* [online]. Kurzy.cz, 2022 [cit. 12. dubna 2022]. Dostupné z: <https://www.kurzy.cz/komodity/psenice-graf-vyvoje-ceny/11-czk-1-rok>.
- [40] *Tutorial: Intro to React* [online]. Meta Platforms, Inc., 2022 [cit. 18. dubna 2022]. Dostupné z: <https://reactjs.org/tutorial/tutorial.html>.
- [41] SKLENÁK, V. *Data, informace, znalosti a Internet*. Vyd. 1. V Praze: C.H. Beck, 2001. C.H. Beck pro praxi. ISBN 80-7179-409-0.
- [42] *An Overview of SLEPT Analysis With Steps To Conduct Analysis* [online]. Indeed Editorial Team, 2021 [cit. 14. dubna 2022]. Dostupné z: <https://www.indeed.com/career-advice/career-development/slept-analysis>.

- [43] SMEJKAL, V. *Řízení rizik ve firmách a jiných organizacích*. 3., rozš. a aktualiz. vyd. Praha: Grada, 2010. Expert (Grada). ISBN 978-80-247-3051-6.
- [44] SAGAR, P. *What Is a Single Page Application? Meaning, Pitfalls & Benefits* [online]. Excellent Webworld, 2020 [cit. 18. dubna 2022]. Dostupné z: <https://www.excellentwebworld.com/what-is-a-single-page-application/>.
- [45] SVOZILOVÁ, A. *Projektový management: systémový přístup k řízení projektů: 3., aktualizované a rozšířené vydání*. Praha: Grada, 2016. ISBN 978-80-271-0075-0.
- [46] KENTON, W. *Strength, Weakness, Opportunity, and Threat (SWOT) Analysis* [online]. Investopedia, 2021 [cit. 14. dubna 2022]. Dostupné z: <https://www.investopedia.com/terms/s/swot.asp>.
- [47] *Rapidly build modern websites without ever leaving your HTML*. [online]. Tailwind Labs Inc., 2022 [cit. 18. dubna 2022]. Dostupné z: <https://tailwindcss.com/>.
- [48] *Válka na Ukrajině ohrožuje globální potravinovou bezpečnost. Chybí pšenice* [online]. idnes.cz, 2022 [cit. 12. dubna 2022]. Dostupné z: [https://www.idnes.cz/ekonomika/zahranicni/ukrajina-rusko-psenice-ceny-staty-indie-dodavky-vyvoz.A220405\\_120235\\_eko-zahranicni\\_jla](https://www.idnes.cz/ekonomika/zahranicni/ukrajina-rusko-psenice-ceny-staty-indie-dodavky-vyvoz.A220405_120235_eko-zahranicni_jla).
- [49] DHADUK, H. *An Ultimate Guide to Web Application Architecture* [online]. simform.com, 2022 [cit. 17. dubna 2022]. Dostupné z: <https://www.simform.com/blog/web-application-architecture/>.
- [50] *ZEFIS – audit informačních systémů* [online]. Zefis, 2018 [cit. 14. dubna 2022]. Dostupné z: <https://www.zefis.cz/>.
- [51] ZUZANA, G. L. P. J. Šedivá. *Podniková informatika: 2., přepracované a aktualizované vydání*. 2., přeprac. a aktualiz. vyd. Praha: Grada Publishing, 2009. ISBN 978-80-247-2615-1.

# Seznam obrázků

3.1	Komponenty informačního systému . . . . .	6
3.2	Systém řízení báze dat . . . . .	9
3.3	Rozdíl mezi životními cykly komunikace tradičních webových aplikací a jednostránkových aplikací . . . . .	11
3.4	JSON Web token . . . . .	15
3.5	Dekódovaný JSON Web token . . . . .	15
3.6	Diagram autentizace pomocí JWT . . . . .	16
3.7	Vygenerované modely . . . . .	19
3.8	Ukázka větvení v gitu . . . . .	23
3.9	Model SLEPTE . . . . .	24
3.10	Porterův model . . . . .	26
3.11	Model 7S McKinsey . . . . .	27
3.12	SWOT matice . . . . .	29
3.13	Základní kroky procesu modelování řízené změny ve firmě . . . . .	31
3.14	Mapa rizik skórovací metody . . . . .	32
4.1	Organizační struktura společnosti . . . . .	34
4.2	Přejímka . . . . .	35
4.3	Graf cen pohonných hmot . . . . .	36
5.1	Mapa rizik před opatřeními . . . . .	51
5.2	Mapa rizik po opatřeních . . . . .	51
5.3	Síťový graf PERT, část první . . . . .	54
5.4	Síťový graf PERT, část druhá . . . . .	55
5.5	ER Digram databáze aplikace . . . . .	60
5.6	Přihlašovací stránka (mobilní verze) . . . . .	64
5.7	Seznam středisek (mobilní verze) . . . . .	65
5.8	Seznam skladů střediska (mobilní verze) . . . . .	66
5.9	Inventář skladu (mobilní verze) . . . . .	67
5.10	Materiály . . . . .	68
5.11	Provedené transakce na skladech . . . . .	68

# Seznam tabulek

5.1	Analýza silového pole . . . . .	43
5.2	Seznam úkolů změny . . . . .	46
5.3	Pravděpodobnost výskytu . . . . .	47
5.4	Metrika dopadu . . . . .	47
5.5	Identifikace hrozeb a scénářů . . . . .	48
5.6	Ohodnocení rizik . . . . .	49
5.7	Návrh opatření . . . . .	50
5.8	Seznam činností, jejich návaznost a časové odhady . . . . .	52
5.9	PERT tabulka . . . . .	53
5.10	Jednorázové náklady . . . . .	56
5.11	Opakující se náklady . . . . .	56