

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Informační systém pro správu firemního majetku
Bc. Jan Korbel

Diplomová práce
2024

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jan Korbel**
Osobní číslo: **I22163**
Studijní program: **N0613A140007 Informační technologie**
Téma práce: **Informační systém pro správu firemního majetku**
Zadávající katedra: **Katedra softwarových technologií**

Zásady pro vypracování

Cílem diplomové práce je navrhnout a implementovat webový informační systém s primárním zaměřením na skladové zásoby a inventarizaci majetku.
Teoretická část práce se bude zabývat analýzou a návrhem aplikace společně s rozborem zvolených technologií ve kterých bude aplikace vytvořena (předpokladem je využití programovacího jazyka PHP).
Praktická část práce bude obsahovat samotnou tvorbu systému, jeho implementaci a nasazení. Součástí řešení budou také jednotkové a integrační testy pro back-end část a také vybrané testy front-end části.

Rozsah pracovní zprávy: **cca 60 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. ARLOW, Jim a NEUSTADT, Ila. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
2. SKLAR, David. PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web. Přeložil Jan POKORNÝ. Encyklopedie Zoner Press. Brno: Zoner Press, 2018. ISBN 978-80-7413-363-3.

Vedoucí diplomové práce: **doc. Ing. Michael Bažant, Ph.D.**
Katedra softwarových technologií

Datum zadání diplomové práce: **8. listopadu 2023**
Termín odevzdání diplomové práce: **17. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 30. listopadu 2023

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 16. 5. 2024

Bc. Jan Korbel

PODĚKOVÁNÍ

Tímto bych rád poděkoval doc. Ing. Michaelu Bažantovi, Ph.D. za vedení diplomové práce a za všechny poskytnuté rady. Veliké poděkování patří mé rodině a blízkým za trpělivost a podporu během mého studia. V neposlední řadě bych také rád poděkoval firmě "NEVŠÍMAL – voda, topení, revize" za možnost zpracovávat toto téma diplomové práce.

ANOTACE

Tato diplomová práce se zabývá vývojem webové aplikace ve frameworku Nette. Jedná se o informační systém pro správu majetku menší, dynamicky se rozvíjející firmy. Teoretická část se věnuje analýze potřeb podniku a již existujících řešení. Na základě získaných specifických požadavků a vyhodnocení současné firemní situace je popsán význam a výhody vlastního informačního systému reflektujícího specifické potřeby firmy. V praktické části práce je představen návrh systému, který zahrnuje jak funkční, tak nefunkční požadavky, a popisuje použité technologie různých typů a velikostí. Finální kapitoly se věnují nasazení systému na platformu Raspberry Pi, což v praxi přináší specifické výhody, a testování aplikace pomocí různých nástrojů a technik. Závěrem jsou formulována doporučení pro další vývoj a možné rozšíření systému.

KLÍČOVÁ SLOVA

Doctrine, Informační systém, Nette, PHP, Systém pro správu majetku, Webová aplikace

TITLE

Information system for corporate asset management

ANNOTATION

This diploma thesis focuses on the development of a web application in the Nette framework. It concerns an information system for asset management within a smaller, dynamically developing company. The theoretical part is devoted to the analysis of the company's needs and existing solutions. Based on the specific requirements and the analysis of the current company situation, the importance and advantages of a custom information system reflecting the specified needs of the company are described. The practical part of the thesis presents the concept of the system, which includes both functional and non-functional requirements, and describes the technologies used of different types and sizes. The final chapters focus on implementing the system on the Raspberry Pi platform, which brings specific benefits in practice, and testing the application using various tools and techniques. Finally, recommendations for further development and possible extension of the system are formulated.

KEYWORDS

Asset Management System, Doctrine, Nette, PHP, Information System, Web application

OBSAH

Seznam obrázků.....	9
Seznam tabulek	10
Seznam zkratk.....	11
Úvod	12
1 Motivace k řešení	13
1.1 Závěrečné shrnutí.....	14
2 Analýza současné situace.....	16
2.1 Proces inventarizace	16
2.2 Správa katalogů a cen	16
2.3 Výhody a nevýhody současného systému	17
2.4 Potenciál pro zlepšení	17
2.5 Shrnutí a závěr	17
3 Analýza již hotových řešení	18
3.1 Univerzální řešení	18
3.1.1 Microsoft Dynamics 365	18
3.1.2 SAP S/4 HANA	18
3.1.3 Oracle ERP	19
3.1.4 Acumatica	19
3.1.5 Odoo.....	20
3.1.6 ABRA	20
3.2 Zhodnocení	21
4 Návrh informačního systému.....	24
4.1 Funkční požadavky	24
4.2 Nefunkční požadavky	25
4.2.1 Use Case	25
4.2.2 ER diagram	27
4.2.3 Datový model.....	28
4.2.4 Diagram aktivit přidání produktu do košíku.....	28
5 Použité Technologie	30
5.1 PHP	30
5.2 Nette.....	30
5.3 Latte	32
5.4 Doctrine/Netrine	32
5.5 MVC	32
5.5.1 Model.....	33
5.5.2 View.....	33
5.5.3 Controller.....	33
5.6 jQuery	33
5.7 MariaDB	34
6 Platforma	35

6.1	Nízké náklady	35
6.2	Flexibilita a rozšiřitelnost	35
6.3	Kontrola nad prostředím	35
6.4	Bezpečnost a soukromí	35
6.5	Zálohování a obnova systému.....	35
7	Prostředí	37
7.1	Dashboard	37
7.2	Tabulka skladových položek	38
7.3	Pokladna.....	42
7.4	Inventura	45
7.5	Flash zprávy	49
7.6	Uživatelé a role	50
8	Testování.....	53
8.1.1	Nette Tester.....	53
8.1.2	Selenium	53
8.1.3	PHPStan	54
8.1.4	Mockery	54
8.1.5	Testy databáze.....	54
8.2	Nasazení systému:.....	55
8.2.1	Nasazení na vybavení firmy	56
9	Budoucnost	58
9.1	Automatizace faktur.....	58
9.2	Obrázky.....	58
9.3	Filtry.....	59
9.4	Propojení s více systémy.....	59
9.5	QR platba	59
9.6	Napojení na e-shop	60
9.7	Kamerové systémy.....	60
	Závěr	61
	Použitá literatura	62
	Přílohy.....	64

SEZNAM OBRÁZKŮ

Obrázek 1 Use case diagram.....	26
Obrázek 2 ER diagram.....	27
Obrázek 3 Datový model	28
Obrázek 4 Diagram pro přidání produktu.....	29
Obrázek 5 Hlavní stránka aplikace	37
Obrázek 6 Skladová tabulka	39
Obrázek 7 Přidávání položek.....	40
Obrázek 8 Hlídané položky	41
Obrázek 9 Graf historie cen.....	42
Obrázek 10 Úprava cen v košíku.....	44
Obrázek 11 Pokladna	45
Obrázek 12 Přehled inventur	46
Obrázek 13 Detail inventury.....	46
Obrázek 14 Export inventury.....	47
Obrázek 15 Načítaná položek inventury.....	48
Obrázek 16 Notifikace	49
Obrázek 17 Vytváření role.....	51
Obrázek 18 Ukázka implementace na vybavení firmy.....	57

SEZNAM TABULEK

Tabulka 1 Aktuální ceník Abra flexi [10].....	22
Tabulka 2 Aktuální ceník Oddo [11]	23
Tabulka 3 Tabulka oprávnění nad stránkami.....	52

SEZNAM ZKRATEK

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CSS	Cascading Style Sheets
DI	Dependency injection
DOM	Document Object Model
DPH	Daň z přidané hodnoty
EAN	European Article Number
ER	Entity-relationship
ERP	Enterprise Resource Planning
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
MVC	Model view controller
ORM	Objektově relační mapování
PDF	Portable Document Format
SD	Secure Digital
SSD	Solid-state drive
SQL	Structured Query Language
URL	Uniform Resource Locator

ÚVOD

V dnešní době, kdy dochází k digitalizaci veškerých úřadů, školství i průmyslu, je vhodné, aby tuto možnost využívaly i menší firmy. V některých podnicích se stále pracuje s formuláři tištěnými na papíru, provádí se inventury skrze hledání v dlouhém papírovém seznamu, nebo se uchovávají data ve strukturách a programech, které k tomu nejsou určeny. Pravděpodobně nejrozšířenějším nástrojem mezi většími aplikacemi je Excel, případně jeho alternativy, které jsou cenově dostupnější, nebo také zdarma. Vést firmu tímto způsobem, snažit se držet data ve stejných tabulkách a neadaptovat se, může být sice proveditelné, ale časově náročné a neefektivní. Zároveň je s touto možností spjata komplikované sdílení dat mezi zaměstnanci. To se obvykle řeší posíláním souborů prostřednictvím cloudových služeb, emailů, nebo přenosu na externím paměťovém médiu. Velké množství firem, které takto, případně podobně nevhodným způsobem fungují, často tvrdí, že jim současné řešení stačí. To může být zapříčiněno tím, že se ještě nedostali do situace, kdy by bylo zřejmé, že je tento přístup nevhodný nebo nedostačující, nebo nemají porovnání, aby si to uvědomili. V případě, že se v této skupině najde firma, která by s tím chtěla něco udělat, nechce a většinou ani nemůže vynakládat příliš velké prostředky na realizaci změny.

Tato situace se týká i firmy, pro kterou je v rámci této diplomové práce vyvíjena webová aplikace pro správu firemního majetku. Jedná se o menší rodinnou firmu, která však roste velmi rychlým tempem, a postupně se snaží expandovat na nové trhy a rozvíjet své podnikání. Návrh systému reflektuje specifické potřeby této firmy. Nejdříve se zaměřuji na důvody, proč je dobré mít vlastní webový informační systém, a následně provádím analýzu současné situace ve firmě. Neopomínám ani stále rostoucí trh s informačními systémy a jejich jak malé, tak velké zástupce, kteří by mohli teoreticky pokrýt alespoň část kladených požadavků. Zhodnocuji zde také související finanční výdaje za měsíční licence. Dále popisuji stanovené funkční a nefunkční požadavky, na základě kterých probíhal další návrh aplikace a tvorba modelů.

Jelikož jsem pro vývoj použil velké množství technologií různých typů a velikostí, rozhodl jsem se čtenáře seznámit s jejich možnostmi, unikátními specifiky a historií, a v případě užití i význam v této aplikaci. Zajímavou kapitolou je i výsledné nasazení na platformu Rappery Pi s popisem výhod tohoto řešení. Finální kapitoly popisují jednotlivé moduly aplikace společně s jejich možnostmi a použitím za doprovodu grafické dokumentace. S tím souvisí nezbytná část otestování celé aplikace, a to hned několika testovými nástroji a různými technikami. Finální část popisuje některé moduly, které jsou nadále zpracovávány mimo rozsah diplomové práce do systému a formulují tak její budoucnost a použití.

1 MOTIVACE K ŘEŠENÍ

Realizace vlastního webového systému na správu skladových zásob, nebo i obecných systémů, může být motivována řadou podnětů, které reflektují specifické potřeby a cíle podniku. Klíčové faktory, které vedly ke zvolení vlastního řešení místo implementace některého z již existujících systémů v tomto případě, jsem zmínil v následujících bodech.

- Přizpůsobení se obchodním potřebám

Běžné systémy na trhu zpravidla nikdy dokonale neodpovídají unikátním procesům a potřebám konkrétní firmy. Naopak kvalitně provedené řešení na míru umožňuje plné přizpůsobení se interním procesům a požadavkům tak, aby bylo co nejefektivnější, a firmě šetřilo čas i finance.

- Flexibilita a rozšiřitelnost

Vlastní systémy nabízejí vyšší úroveň flexibility a rozšiřitelnosti. Firma může systém kdykoliv upravit nebo rozšířit o další funkce podle aktuálních potřeb a trendů trhu bez závislosti na externím dodavateli, samozřejmě s ohledem na možné náklady spojené s dalším vývojem. Kromě obsahových změn lze systém kdykoli přizpůsobit vzhledově, aby byl uživatelsky co nejpřívětivější a nejpřehlednější, a reflektoval například měnící se priority a četnost jednotlivých úkonů.

- Integrace s existujícími systémy

Integrace vlastního systému s již používanými nástroji a systémy v rámci firmy může být hladší a efektivnější. Tento přístup minimalizuje potenciální problémy s kompatibilitou a umožňuje lepší využití stávající IT infrastruktury.

- Konkurenční výhoda

Vlastní systém, který je přímo navržen pro podporu unikátních procesů firmy, může představovat významnou konkurenční výhodu. Specifické funkce a zefektivnění procesů mohou zapříčinit znatelný rozdíl v rychlosti, kvalitě a nákladech.

- Kontrola nad bezpečností

Bezpečnostní aspekty lze při vývoji na míru detailně adresovat a přizpůsobit požadavkům firmy. To zahrnuje jak fyzickou bezpečnost dat, tak ochranu proti kybernetickým hrozbám. Systém se může plně přizpůsobit možnostem, které místní síťové prostředí nabízí, a napojit tak veškerá specializovaná zařízení zapojená do této sítě.

- Nákladová efektivita v dlouhodobém horizontu

I když počáteční investice do vývoje vlastního systému může být vyšší než koupě jednoho ze standardních řešení, v dlouhodobém horizontu může být tento krok nákladově efektivnější díky eliminaci poplatků za licence, přizpůsobení a rozšíření. Kromě samotných licencí za software je třeba brát v potaz ceny serverů, na kterých tyto softwary a jejich služby běží.

- Nezávislost na dodavateli

Další výhodou interně vyvíjeného webového systému pro správu skladových zásob je nezávislost na externích dodavatelích software a jejich časových harmonogramech pro aktualizace a podporu. Díky vlastnímu software získá firma plnou kontrolu nad celým systémem a jeho budoucím vývojem, což je klíčové pro rychlé a individuální reagování na požadavky trhu.

1.1 Závěrečné shrnutí

Rozhodnutí pro vývoj vlastního webového systému na správu skladových zásob bylo pečlivě zváženo firemními zástupci i webovým vývojářem s ohledem na specifické potřeby a cíle firmy. Byla provedena hloubková analýza interních procesů, požadavků, které jsou zmíněny v dalších pasážích této práce, bezpečnostních standardů a nákladů spojených s vývojem a údržbou systému. Individuální přístup při tvorbě systému přináší firmě významné výhody v podobě plné přizpůsobitelnosti, kontroly a možnosti integrace, které posilují konkurenceschopnost a efektivitu práce.

V kontextu rychlého růstu a dynamických změn ve firemním prostředí je schopnost rychle se adaptovat na nové požadavky a výzvy nezbytná. Integrace s ostatními systémy a aplikacemi v rámci firmy je u vlastního systému rovněž zjednodušena, což vede k lepšímu propojení dat a efektivnějšímu rozhodování. Dále je dobré zmínit, že některá robustnější řešení mohou být pro specifické potřeby firmy příliš robustní a komplikované.

Dalším kritickým faktorem je bezpečnost, a to zvláště v dnešní době, kdy jsou kybernetické útoky a úniky dat častější. Je proto nespornou výhodou mít veškerá svá data pod svou vlastní kontrolou, a nevystavovat je zbytečnému riziku.

Nákladová efektivita vlastního systému se projevuje zejména v dlouhodobém horizontu, kdy se investice do vývoje a implementace systému vrátí ve formě zvýšené efektivity, snížení chyb a optimalizace skladových procesů.

Zatímco standardní řešení mohou nabízet rychlé nasazení a nižší počáteční náklady, různá omezení a nemožnost přizpůsobení se specifickým potřebám firmy může vést k vyšším

provozním nákladům a kompromisům v oblasti funkcionalit a bezpečnosti v dlouhodobém horizontu.

2 ANALÝZA SOUČASNÉ SITUACE

Tato práce se zabývá návrhem webového systému pro správu skladových zásob malého podniku. Jedná se o rodinný podnik v oboru stavebnictví s počtem zaměstnanců do 10 osob, avšak s neustále se rozšiřující nabídkou produktů i služeb. Pro účely podnikání byl vystavěn prodejní sklad, kde jsou umístěny materiály, které se používají při montáži, ale i prodejné zboží pro širokou veřejnost. Vzhledem k aktuální situaci firmy, ale i její vizi do budoucna, je změna procesu správy zásob nutná.

Současný systém správy inventury a katalogů ve firmě je založen na ručních procesech. Inventura je prováděna zaměstnanci, kteří zapisují údaje na papír a poté je přepisují do Excelových tabulek. Tento proces byl zaveden před několika lety jako jednoduché řešení pro správu zásob a katalogů, ale s postupem času a přibývajícimi zásobami na skladě, se stal velmi časově náročným a nese s sebou vysoké riziko chyb. Následující analýza zkoumá klíčové aspekty tohoto systému a identifikuje oblasti, kde by mohlo dojít ke zlepšení.

2.1 Proces inventarizace

Inventura ve firmě probíhá jednou za rok. Tým zaměstnanců prochází sklady a zapisuje počty produktů do předtištěných formulářů. Jelikož si jsou některé položky velmi podobné, je při procházení skladu velmi obtížné přiřadit jednotlivé položky samotným názvům na seznamu. Především někteří brigádníci nemají zkušenosti s těmito produkty, a jejich nalezení je pro ně o to složitější. Tato situace často končí dotazováním na ostatní pracovníky a společnou snahou o rozpoznání produktu. Tento ruční proces je velmi časově náročný a náchylný k chybám. Po dokončení inventury jsou údaje přepsány do Excelových tabulek, kde jsou dále analyzovány a zpracovávány. Za tento proces je zodpovědný vedoucí skladu, který také zajišťuje správnost přepisu dat. V tomto procesu je velké množství lidských faktorů, při kterém může dojít k chybám jako jsou špatné identifikování a zařazení produktu, chyba v přepisu, ale i špatná čitelnost rukopisu pracovníků.

2.2 Správa katalogů a cen

Katalogy produktů včetně specifikací a ceny jsou vedeny v Excelových tabulkách. Každý produkt má vlastní řádek, kde jsou zaznamenány informace o názvu, popisu, ceně a dalších detailech. Těchto informací není mnoho a samotné vyhledávání tak funguje přes funkce filtrace a vyhledávání implementované Excelem. Aktualizace cen probíhají ručně na základě pohybu cen na trhu. Tuto činnost provádí administrativní tým, který aktualizuje data podle potřeby. Tento systém je sice jednoduchý, ale současně je náchylný k chybám, zejména při manuálních

úpravách. Excelové tabulky mohou být dobrým nástrojem pro vedení skladových zásob malých podniků, jelikož umožňují částečnou automatizaci prostřednictvím vestavěných funkcí a vzorečků. V případě této rychle rostoucí firmy však považují tento přístup za velmi omezující, časově náročný a z dlouhodobého hlediska neudržitelný. Skladový sortiment se rozšiřuje každým měsícem a již nyní jsou ve skladových zásobách stovky ba i tisíce druhů položek.

2.3 Výhody a nevýhody současného systému

Výhodou současného systému je jeho nízká cena a relativní jednoduchost. Excel je běžně používaný software, který je dostupný všem zaměstnancům. Nevýhodou je však riziko chyb způsobených ručním přepisem dat a časová náročnost procesu. Dále může být problémem nedostatek integrace s jinými systémy, což ztěžuje sdílení dat a spolupráci mezi odděleními. Obecně je tento systém využíván pouze na evidenci dat, nikoli na jejich velké zpracování. Sdílení mezi zařízeními nebo uživateli se musí řešit prostřednictvím cloudových služeb, nebo čistým přenosem, například přes e-mail. Tím se riskuje neaktuálnost dat na některých zařízeních a následné chyby v naceňování pro zákazníka.

2.4 Potenciál pro zlepšení

Zavedení nového informačního systému představuje značný potenciál pro zlepšení efektivity a přesnosti ve správě zásob. Automatizace procesu inventarizace by snížila riziko chyb a zkrátila čas potřebný k dokončení. Digitalizace katalogů a cen by usnadnila správu a sdílení dat. Informační systém by také mohl umožnit lepší integraci s jinými odděleními, což by podpořilo koordinaci a spolupráci. Dále by byla firma lépe připravena na případné zřízení e-shopu, jelikož data by byla již zpracována.

2.5 Shrnutí a závěr

Současný systém je jednoduchý, ale má několik významných nevýhod. Ruční procesy jsou náchylné k chybám a časově náročné. Zlepšení může být dosaženo prostřednictvím automatizace a digitalizace. Doporučuje se zvážit implementaci informačního systému, který by zvýšil efektivitu, přesnost a usnadnil spolupráci mezi odděleními. Další kroky by měly zahrnovat detailní plánování a výběr vhodného řešení, které splňuje potřeby firmy.

3 ANALÝZA JIŽ HOTOVÝCH ŘEŠENÍ

V současné době se na trhu nachází široké spektrum ERP systémů, které slouží k efektivní správě skladových zásob a dalších firemních procesů. Systémy se liší nejen svými funkcionalitami, ale i rozsahem použití, uživatelským prostředím, možnostmi integrace a cenou. Mezi nejznámější a nejčastěji používané ERP systémy patří Microsoft Dynamics 365, SAP, Oracle ERP, Acumatica a Odoo, každý s vlastními specifickými výhodami a zaměřením. [1]

3.1 Univerzální řešení

V následujících kapitolách popisují již hotová řešení, která se v současné situaci nacházejí na trhu. Zmiňují zde zástupce světových lídrů, ale také menší projekty s kořeny v České republice.

3.1.1 Microsoft Dynamics 365

Dynamics 365 představuje integrovanou sadu cloudových aplikací pro řízení podnikových procesů, vyvíjenou a distribuovanou společností Microsoft Corporation. Tento komplexní softwarový ekosystém poskytuje širokou škálu nástrojů pro správu vztahů se zákazníky (CRM), správu operací (ERP), analýzu dat a další oblasti podnikání. Jeho flexibilita, modularita a schopnost snadné integrace se staly atraktivními faktory pro podniky různých velikostí a odvětví. Microsoft Dynamics 365 je významným hráčem na trhu ERP systémů, který je známý svou integrací s dalšími produkty Microsoftu, jako jsou Office 365, Azure a Power BI. Díky tomu jsou jeho silnou stránkou automatizace a analýza dat, což významně podporuje firemní operace. Jeho implementace je obecně jednodušší a náklady na zavedení jsou ve srovnání s konkurenčními systémy nižší. [2]

3.1.2 SAP S/4 HANA

SAP S/4 HANA je považována za lídra trhu ERP systémů, zejména díky svým rozsáhlým funkcím a hluboké integraci specifických oborových řešení. SAP S/4 HANA poskytuje lokalizační schopnosti pro více než 130 zemí a je vhodný pro široké spektrum průmyslových odvětví.

SAP S/4HANA, zavedená společností SAP SE, představuje nejnovější generaci podnikových zdrojových plánovacích (ERP) systémů. Byla navržena specificky pro využití vysoké výkonnosti in-memory databáze SAP HANA. Tento systém se od svých předchůdců významně liší z hlediska zpracování transakcí a analýzy dat v reálném čase, což umožňuje podnikům zefektivnit operace a zlepšit uživatelskou zkušenost.

Jednou z klíčových inovací SAP S/4HANA je zjednodušení datového modelu. Díky tomu se zmenšuje celkový objem dat, což vede k nižším nákladům na úložiště a posílení výkonu systému. SAP S/4HANA také integruje analýzy a transakce do jediného systému, což eliminuje potřebu oddělených systémů pro reportování a transakční zpracování. Toto integrované prostředí zapříčiňuje lepší rozhodování díky poskytování přesnějších a aktuálnějších informací. SAP S/4HANA je dostupná v několika verzích a cloudových řešení, což umožňuje podnikům vybrat si model nasazení, který nejlépe vyhovuje jejich specifickým potřebám. Cloudová verze, SAP S/4HANA Cloud, nabízí výhody v podobě nižších počátečních investic, flexibilního škálování a automatických aktualizací, což zjednodušuje správu a udržování systému.[3]

3.1.3 Oracle ERP

Oracle ERP, součást rodiny cloudových aplikací Oracle, je komplexní podnikový zdrojový plánovací systém navržený pro organizace všech velikostí. Jakožto kompletní, moderní a bezpečný cloudový ERP systém, Oracle ERP Cloud poskytuje širokou škálu funkcí pokrývajících hlavní oblasti podnikového plánování, včetně financí, projektového managementu, obchodních transakcí, dodavatelského řetězce a výroby. Klíčovým prvkem tohoto systému je jeho schopnost poskytovat detailní analytické informace, které podporují lepší rozhodování a optimalizaci obchodních procesů.

Oracle ERP Cloud je vybaven pokročilými technologiemi, jako je umělá inteligence (AI), strojové učení a automatizace procesů, které pomáhají zvyšovat efektivitu, snižovat ruční práci a zlepšovat přesnost dat. Díky cloudové infrastruktuře Oracle nabízí systém vysokou míru škálovatelnosti, bezpečnosti a spolehlivosti.[4]

3.1.4 Acumatica

Acumatica je cloudový podnikový systém, který byl vyvinut s cílem plnit neustále se měnící potřeby středních firem a jejich zákazníků. Nabízí snadnou integraci s již existujícími nástroji a systémy, což minimalizuje přerušení běžících procesů. Díky intuitivnímu uživatelskému rozhraní se zlepšuje efektivita práce a produktivita týmu. Moderní workflow podporuje spolupráci napříč různými oblastmi podniku. Přístupnost k datům z libovolného zařízení s internetovým připojením umožňuje práci odkudkoliv a kdykoliv. Možnost personalizace systému bez nebo s minimálním kódováním umožňuje přizpůsobit systém konkrétním potřebám podniku. Díky škálovatelnosti je možné snadno reflektovat požadavky na uživatele a zdroje podle aktuálních potřeb. Tyto vlastnosti činí Acumaticu atraktivní volbou pro střední

podniky hledající moderní a flexibilní řešení pro zlepšení efektivity, inovace a konkurenceschopnosti.[5]

3.1.5 Odoo

Odoo se v oblasti firemního software etablovalo jako významná platforma, jež nabízí širokou škálu aplikací určených k podpoře a zlepšení operací společností různých velikostí a sektorů. Odoo umožňuje společnostem snadné uzpůsobení či rozšíření systémů dle individuálních potřeb a růstových fází. Dále klade důraz na uživatelskou přívětivost a komplexnost řešení, flexibilitu a škálovatelnost systému, díky čemuž vyhovuje potřebám jak malých start-upů, tak velkých korporací. Platforma se může chlubit více než 30 hlavními aplikacemi a více než 16,000 dalšími aplikacemi vyvinutými aktivní komunitou přesahující 1500 členů. Díky tomu dokáže Odoo pokrýt i velmi specifické a rozmanité podnikové potřeby.

Díky své open-source povaze je Odoo neustále rozvíjeno a udržováno rozsáhlou komunitou vývojářů, což zajišťuje neustálé inovace a adaptaci na měnící se podnikové prostředí a požadavky uživatelů. Cenová přístupnost, včetně možnosti využití některých modulů zdarma, umožňuje i malým společnostem vstup do světa sofistikovaného firemního software bez nutnosti významných počátečních investic. S více než 5 miliony uživatelů po celém světě a širokým spektrem aplikací, Odoo zaujímá vedoucí pozici v oblasti podnikových řešení. Jedná se o oblíbenou platformu mezi společnostmi hledající efektivní, flexibilní a cenově dostupné podnikové software řešení. [6]

3.1.6 ABRA

Společnost ABRA Software se řadí mezi přední české vývojáře informačních systémů, které najdou uplatnění v širokém spektru podnikatelských aktivit. S více než dvacetiletou historií a zaměřením na inovativní přístupy k softwarovému vývoji, ABRA Software představuje klíčového hráče na trhu ERP systémů v České republice. Díky svým produktům ABRA Gen a ABRA Flexi, firma nabízí komplexní řešení pro správu podnikových procesů různě velkých společností.

ABRA Gen je výkonný, flexibilní a intuitivní ERP systém určený pro střední a velké podniky, který pokrývá všechny klíčové oblasti firemního řízení od CRM, přes výrobu až po poprodejní servis. Systém je navržen tak, aby ušetřil čas díky automatizaci mnoha procesů, které by jinak vyžadovaly ruční práci. Jeho adaptabilita umožňuje přizpůsobení specifickým potřebám každé firmy, čímž podporuje její jedinečnost a podporuje a růst.

ABRA Flexi, na druhou stranu, je cloudově orientované řešení zaměřené na potřeby malých a středních podniků. Jeho klíčové vlastnosti zahrnují snadnou integraci s jinými systémy díky výkonnému API, inteligentní vyhledávání a filtrování, přehledný dashboard pro rychlý přehled o finančním zdraví firmy a vstřícnou uživatelskou podporu společně s aktivní komunitou uživatelů. Flexi je navrženo tak, aby poskytovalo pohodlí a povzbuzovalo radost z práce díky intuitivnímu uživatelskému rozhraní a automatizaci běžných úkolů. [7]

3.2 Zhodnocení

Při výběru ERP systému je klíčové zvážit specifické potřeby firmy, včetně velikosti podniku, průmyslového odvětví, požadavků na integraci a rozpočtu. Analýza a porovnání jednotlivých systémů od různých poskytovatelů, může pomoci určit, které řešení nejlépe vyhovuje konkrétním firemním potřebám. Bryce Emley nabízí přehledný seznam osmi nejlepších ERP software pro rok 2024, který zahrnuje Oracle NetSuite ERP, Odoo ERP, SAP Business a další.[8] Při analýze a porovnání systémů pro správu skladových zásob se objevují různá ERP řešení, která nabízejí širokou škálu funkcionalit pro různé podnikové potřeby. Tyto systémy se liší svými přístupy, zaměřením a integrací, což může podnikům přinést jedinečné výhody v závislosti na jejich specifických potřebách.

V současné době je na trhu přítomna široká škála univerzálních informačních systémů, jak bylo zmíněno v předchozích částech této práce, stejně jako modulárních alternativ. Kromě těchto existují také řešení, která jsou navržena s cílem uspokojit specifičtější podnikové požadavky. Mezi tyto systémy patří, mimo jiné, Katana, Zoho a FISHBOWL, které reprezentují různorodost dostupných aplikací na trhu. Přesto se ukázalo, že ani tyto, ani jiné dříve zmíněné systémy, nebyly schopny plně uspokojit unikátní potřeby konkrétního podniku, buď kvůli jejich nadměrné komplexnosti a robustnosti, nebo naopak kvůli absenci některých klíčových funkcionalit. Dalšími z negativních faktorů byly také vysoké provozní náklady. [9]

Z analytického průzkumu bylo zjištěno, že nejvhodnějšími kandidáty, které by mohly vyhovovat specifickým potřebám podniku, jsou systémy Odoo a ABRA Flexi. Odoo se vyznačuje flexibilitou, která umožňuje využívání jednoho z jeho modulů zcela bezplatně, což představuje významnou výhodu pro malé podniky nebo začínající firmy hledající efektivní řešení svých počátečních informačních potřeb. Na druhou stranu, ABRA Flexi nabízí možnost integrace s libovolnými lokálními technologiemi prostřednictvím již zmíněného aplikačního programovacího rozhraní (API), což umožňuje vyšší míru personalizace a adaptability systému k specifickým požadavkům podniku. Tyto charakteristiky činí Odoo a ABRA Flexi potenciálně

nejvhodnějšími řešeními pro podniky hledající způsoby, jak optimalizovat své procesy pomocí informačních technologií.

Bohužel ani tyto známé produkty nepředstavují ideální volbu. Žádný systém nemůže splňovat všechny přímé požadavky každého zákazníka. Výsledkem pak může být, že uživateli chybí některé potřebné funkce, ale je zahlcen jinými, které jsou pro něj zbytečné. Takový problém se často řeší kompromisem, a sice že je každá část řešena jinou aplikací, přičemž takové řešení není optimální. Druhým problémem je stabilita. Téměř nikdo nemůže garantovat stoprocentní dlouhodobou dostupnost a podporu. Stejně tak může být problém s dalším růstem firmy, kdy budou přibývat další potřeby, které univerzální software nedokáže uspokojit.

Posledním důvodem je cena. Oba zmínění zástupci poskytují své aplikace za měsíční poplatky. Cena je stanovena podle počtů modulů a uživatelů. Při zaokrouhlení na celé stovky a pomnutí možných slev je třeba počítat s 400 Kč/osoba/měsíc za základní balíček modulů. Podrobnější přehled cen je vyobraz v následujících tabulkách.

Tabulka 1 Aktuální ceník Abra flexi [10]

Licence	Moduly	Cena za uživatele na měsíc
Basic	Účetnictví Sklady Fakturace Banka a pokladna Majetek	395 Kč
Business	Vše v Basic a navíc: Mzdy a Personalistika Uživatelské reporty Importy dat z Excelu Nabídky, objednávky, zakázky	995 Kč
Premium	Vše v Business a navíc: Pokročilý obchod a sklady EDI komunikace Pravidelná automatická fakturace Kusovník	1295 Kč

Tabulka 2 Aktuální ceník Odoo [11]

Licence	Aplikace	Cena za uživatele
Jedna aplikace zdarma	Jedna libovolná aplikace	Zdarma
Standardní	Všechny aplikace Odoo online	14,80 €
Pokročilý	Veškeré aplikace Odoo Online / Odoo.sh** Odoo Studio Multi-společnosti Externí API	22,40 €

Při předpokladu nedostatečné funkcionality jediné aplikace u Odoo, se roční náklady mohou vyšplhat na nemalé částky v závislosti na počtu zaměstnanců. Navíc je zde velká nevýhoda v přístupnosti uživatelů. Pokud bude chtít firma provést velkou skladovou inventuru a najme si na pomoc několik brigádníků, bude muset zaplatit licence jednorázově i pro tyto účely. Naopak u vlastní implementace jsou náklady větší na počátku, ale z dlouhodobého hlediska se návratnost investice vyplatí. Navíc toto řešení přináší ideální pokrytí požadavků a uživatelských scénářů.

4 NÁVRH INFORMAČNÍHO SYSTÉMU

V rámci projektu byly definovány funkční a nefunkční požadavky (dle [12]), které slouží jako klíčová vodítka pro vývoj a implementaci systému. Tyto požadavky jsou zásadní pro určení očekávané funkcionality a kvality systému, zajišťující, že výsledný produkt bude splňovat potřeby uživatelů a požadavky podniku.

Funkční požadavky popisují konkrétní operace, služby nebo úkoly, které systém musí provádět. Tyto požadavky jsou zaměřeny na očekávanou funkcionalitu, která zahrnuje správu informací o skladovaných produktech, inventarizaci majetku, definování uživatelských rolí, správu číselníků, a mnoho dalších operací nezbytných pro efektivní správu skladových zásob a obchodních transakcí.

Na druhé straně, nefunkční požadavky specifikují kritéria, která lze použít k posouzení fungování systému, ale nepopisují přímo jeho chování. Tato kritéria zahrnují aspekty jako jsou podpora a údržba systému po jeho nasazení, kompatibilita s různými zařízeními a prohlížeči, bezpečnost a uživatelská přívětivost. Tyto požadavky jsou klíčové pro zajištění, že systém bude spolehlivý, bezpečný, a snadno použitelný.

Následující požadavky byly vytvořeny na základě zadání vyhotovené firmou. Veškeré větší úpravy byly průběžně konzultovány a odsouhlaseny, a mělo by tak dojít k přesnému pokrytí uživatelských potřeb.

4.1 Funkční požadavky

- FR1: Systém umožní uživatelům spravovat informace o skladovaných produktech, včetně jejich názvu, popisu, množství a umístění.
- FR2: Systém umožní provádět inventarizaci majetku, tj. aktualizovat stav skladovaných produktů a provádět inventární operace, jako je přidání nového produktu, odebrání, či změna jeho stavu.
- FR3: Systém umožní definovat různé uživatelské role s příslušnými oprávněními pro přístup a manipulaci se skladovými zásobami.
- FR4: Systém umožní správu číselníků.
- FR5: Systém bude obsahovat pokladnu/nákupní košík.
- FR6: Systém umožní naskenovat položky s čárovým kódem.
- FR7: Systém umožní importovat produkty z již existujících excelových seznamů.

- FR8: Systém umožní exportovat data do několika formátů, včetně Excelu.
- FR9: Systém umožní generovat a následně tisknout jednotlivé štítky k produktům s čárovým kódem.
- FR10: Systém umožní vyhledávání a filtrování produktů.
- FR11: Systém umožní generovat PDF fakturu.
- FR12: Systém umožní generovat a následně tisknout účtenky na účtenkové tiskárně v terminálu.
- FR13: Systém umožní provádět inventury na skladové zásoby
- FR14: Systém bude zpracovávat data a zobrazovat je v jednoduché statistice prostřednictvím grafů
- FR15: Systém bude ukládat historii majetku v čase.

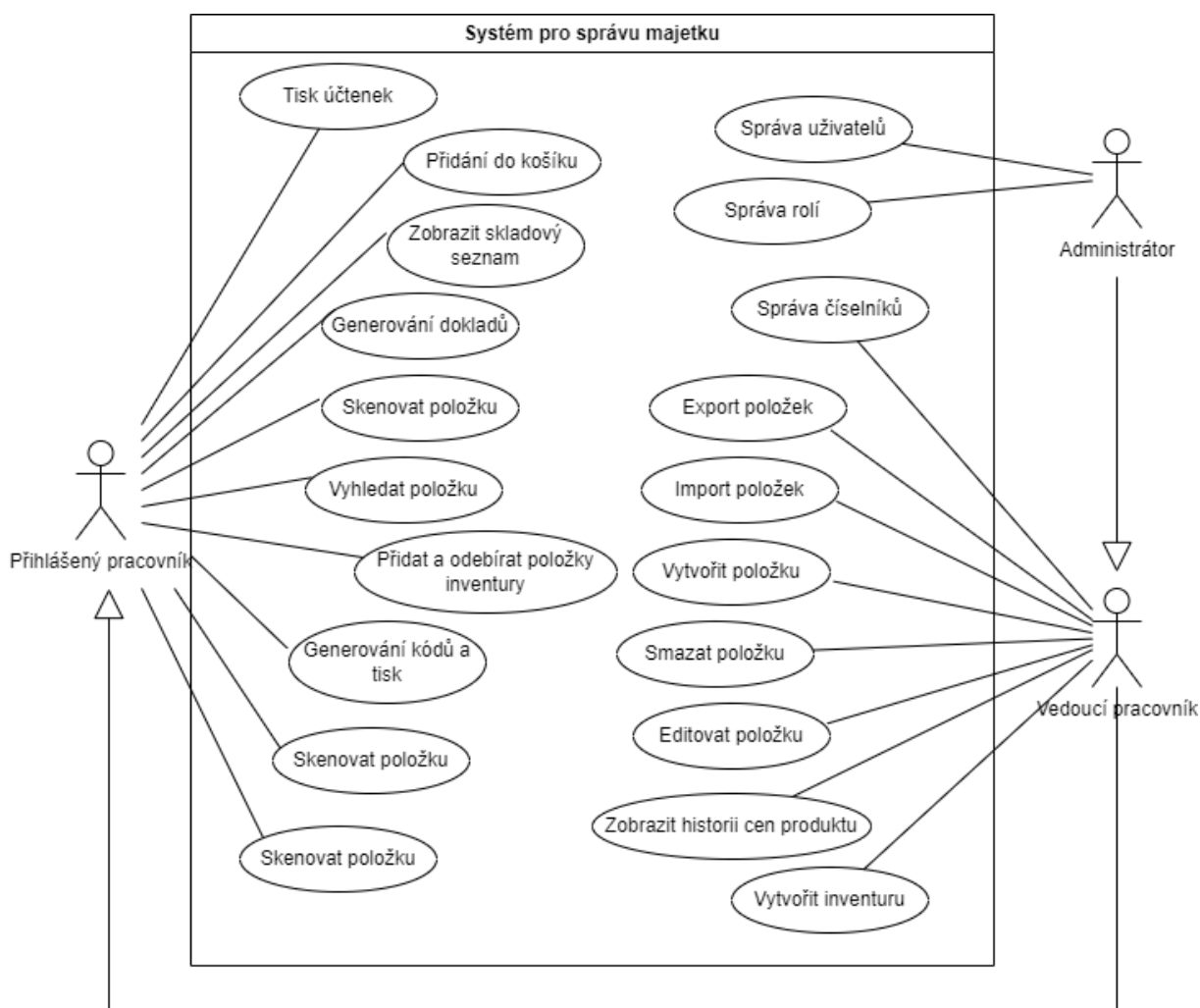
4.2 Nefunkční požadavky

- NR1: Zajištění podpory a údržby systému po jeho nasazení, včetně oprav chyb a aktualizací.
- NR2: Kompatibilita systému s různými webovými prohlížeči a operačními systémy pro uživatelskou flexibilitu.
- NR3: Systém bude možné provozovat na Raspberry Pi 4.
- NR4: Systém bude průběžně testován.
- NR5: Systém bude nasazen na lokální síti firmy.
- NR6: Vysoká úroveň bezpečnosti, včetně ochrany dat a autentizace uživatelů.
- NR7: Systém musí být navržen s důrazem na uživatelskou přívětivost a intuitivní navigaci.
- NR8: Systém bude pravidelně zálohován

4.2.1 Use Case

V předloženém use case modelu je prezentováno základní konfigurační nastavení aplikace, které je navrženo tak, aby odrazilo tři základní uživatelské role, jež jsou klíčové pro její počáteční fázi využití. Tyto role jsou definovány jako Administrátor, Vedoucí pracovník a Pracovník, přičemž každá z nich disponuje specifickými oprávněními a přístupy k různým

funkcím systému. Tato struktura nejenže zjednodušuje administraci a delegaci úkolů v rámci organizace, ale také zajišťuje, že každý uživatel může efektivně plnit své povinnosti s ohledem na svou přidělenou roli od administrátora.



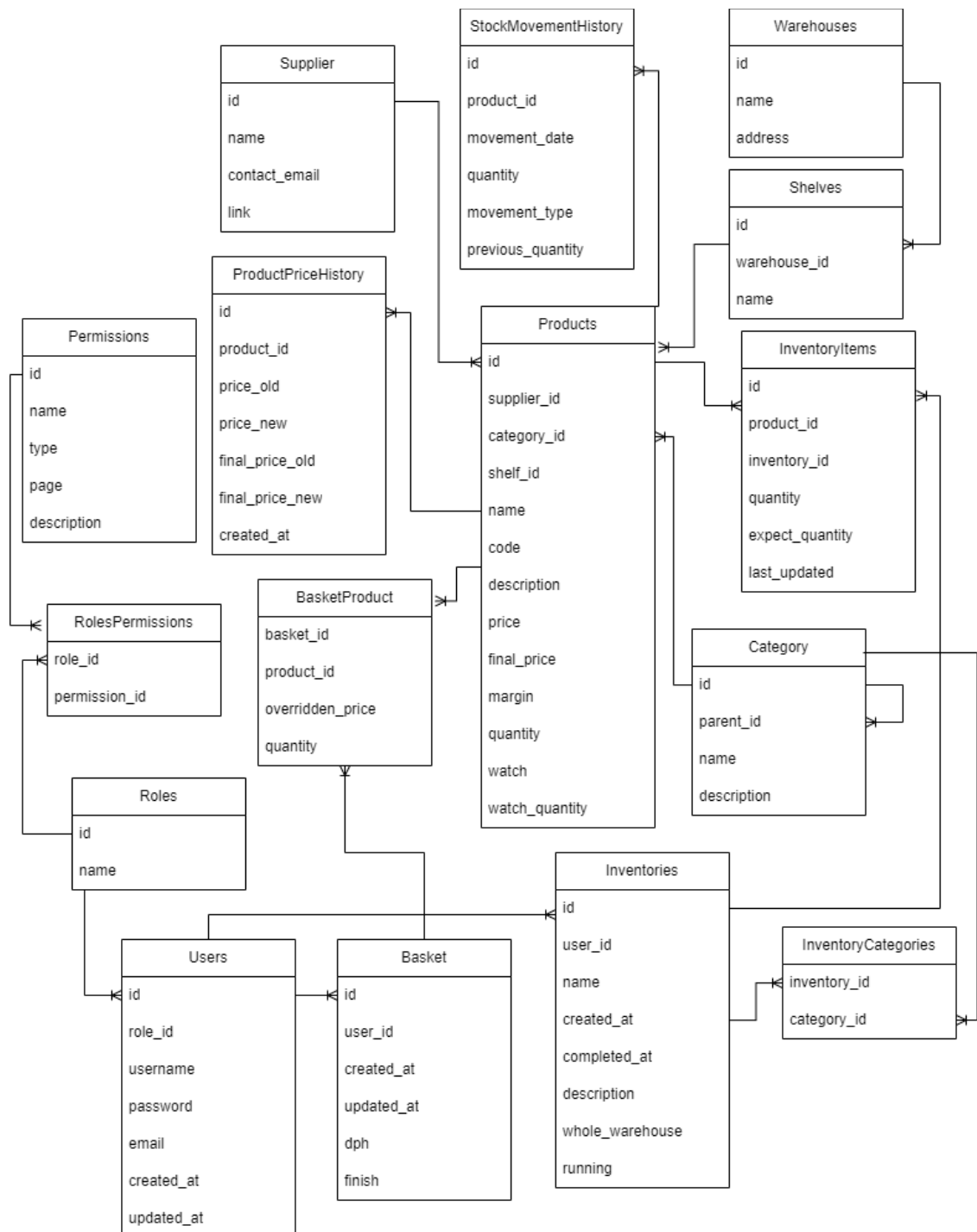
Obrázek 1 Use case diagram

Přestože jsou v modelu zahrnuty pouze tři základní role, systém je navržen s předvídáním budoucího rozvoje a umožňuje vytváření nových uživatelských rolí s přizpůsobenými oprávněními. Tato funkcionality je nezbytná pro podporu dynamického vývoje pracovních procesů a postupů, neboť dovoluje organizacím rychle reagovat na nové výzvy a příležitosti.

Implementace systému s možností přizpůsobení a rozšiřování rolí tak podporuje strategickou flexibilitu a inovace. Díky tomu je aplikace schopna nejenže efektivně sloužit současným potřebám, ale také se adaptovat na potřeby budoucí, což přispívá k dlouhodobé udržitelnosti systému a rozvoji firmy.

4.2.2 ER diagram

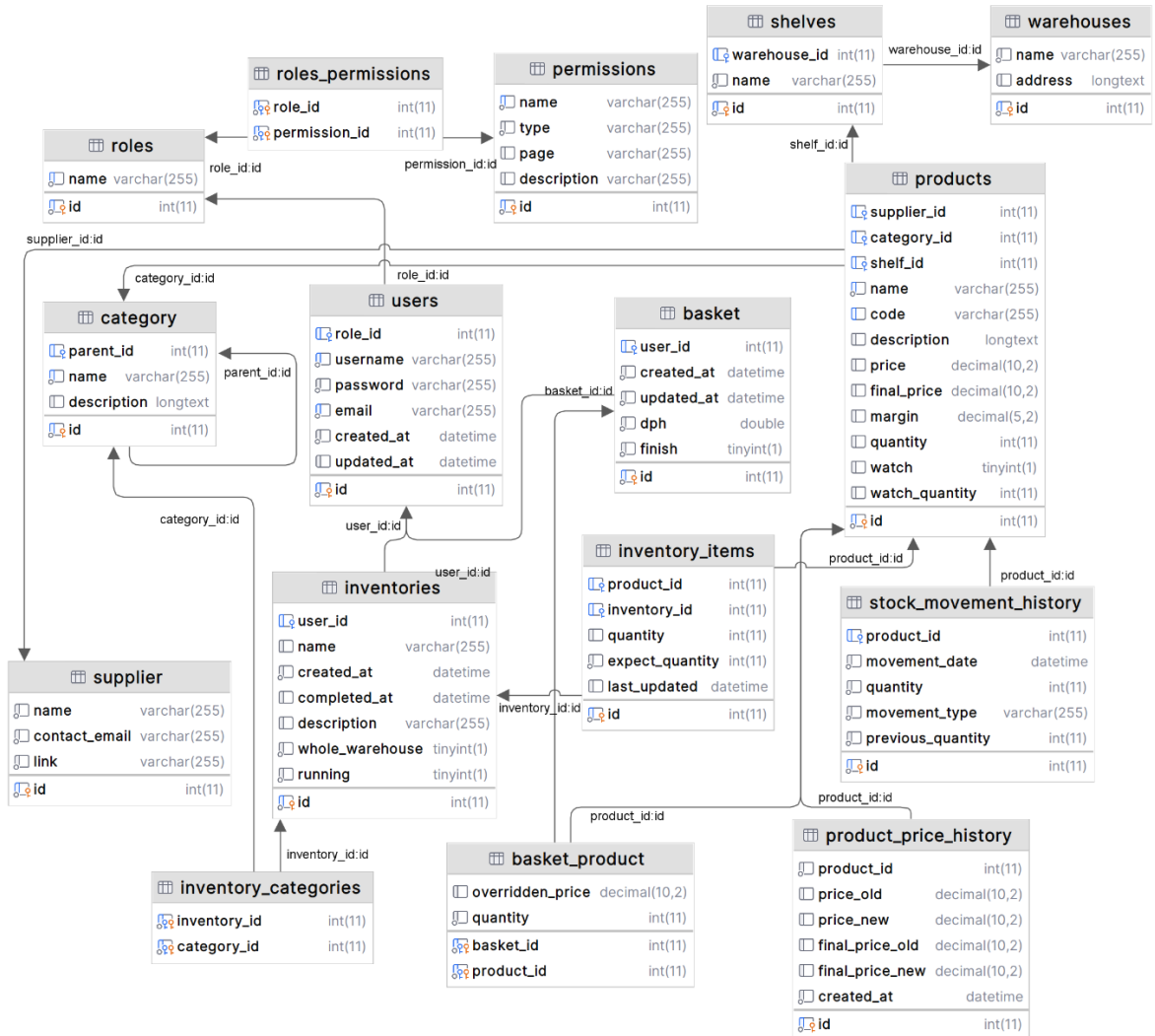
Následující obrázek ukazuje navržený ER diagram pro celou aplikaci. Tento diagram má za úkol vyobrazit jednotlivé relace mezi entitami. Jednou ze zajímavých relací je relace u entity „Category“. Ze schématu lze vyčíst, že je zde relace realizována na stejnou entitu, ze které vychází. To je zapříčiněno tím, že bylo třeba zajistit možnost řazení kategorií do podkategorií a dále je větvit.



Obrázek 2 ER diagram

4.2.3 Datový model

Na následujícím obrázku je vyobrazen datový model celé aplikace. Jednotlivé tabulky představují Doctrine entity, které jsou specifikovány ve zdrojovém kódu. Na rozdíl od ER diagramu jsou zde uvedeny finální datové typy a jednotlivé identifikátory.

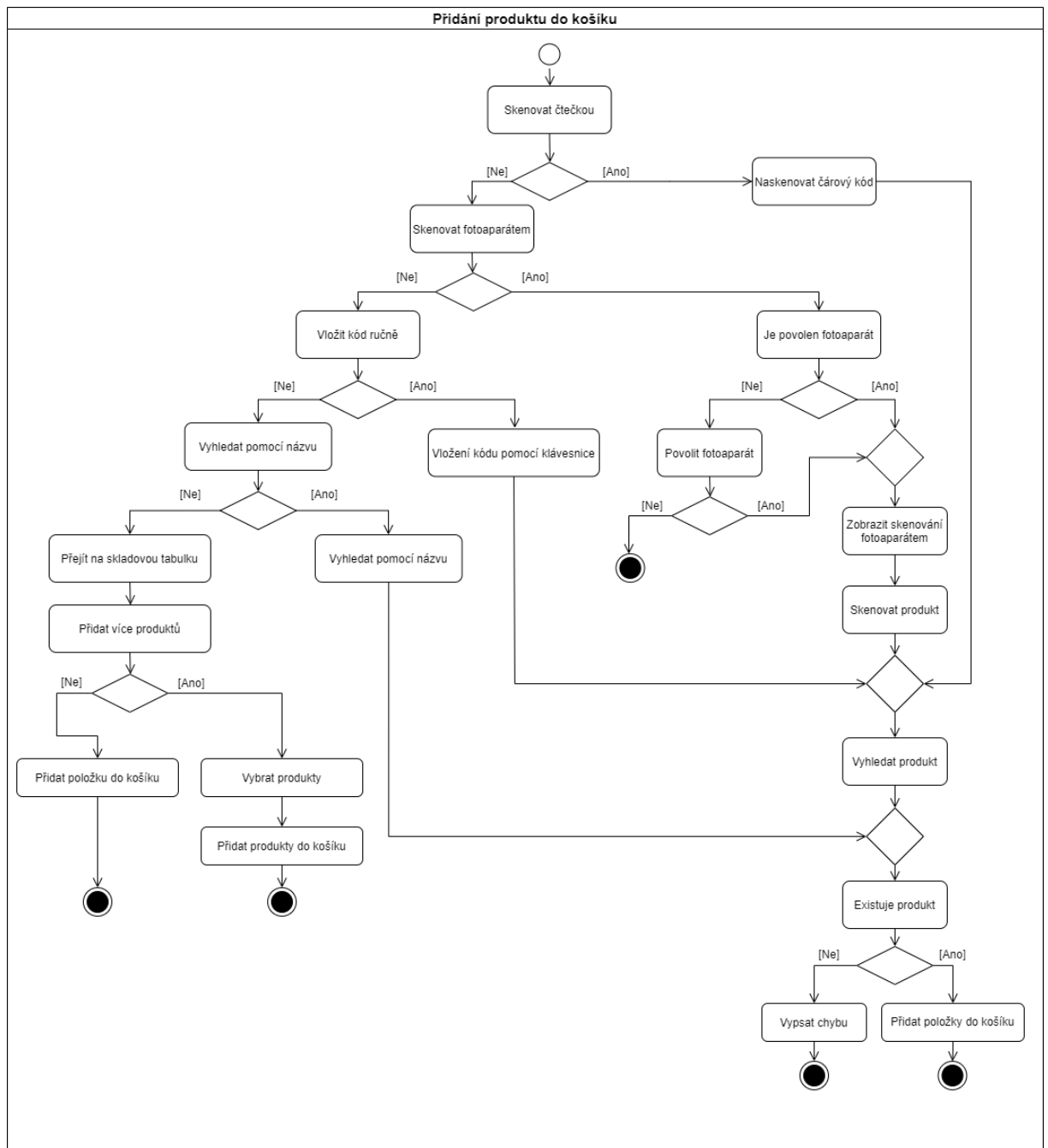


Obrázek 3 Datový model

4.2.4 Diagram aktivit přidání produktu do košíku

Proces přidání produktu do košíku je jedním z rozsáhlejších a složitějších procesů, které jsou obsaženy v této práci. Proto jsem se pro lepší přiblížení čtenáři rozhodl tento proces vyobrazit ve formě aktivity diagramu. Tento diagram zachycuje obrázek číslo 4. Diagram ukazuje několik možných cest, které pracovníkovi firmy umožní přidat produkty do pokladního košíku za účelem prodeje. Jednou z prvních cest jsou skenovací periferie. Zde záleží, jestli chce uživatel používat pro načítání některé z přímo na to určených zařízení, například čtečku, nebo si vystačí

s vestavěným fotoaparátem. Podmínkou pro použití vestavěného fotoaparátu je povolení přístupu k této části periferie, o které se stará samotný prohlížeč, a bez něhož nelze pokračovat ve skenování. Dále jsou zde zahrnuty ruční metody. Při jedné z nich uživatel vloží EAN kód přímo do textového pole. V jiném případě může uživatel použít vyhledávání v horní liště, kde lze u nalezeného produktu podle jména nebo kódu zvolit možnost přidání do košíku. Ostatní přidávání se provádí přímo ve skladové tabulce, kde lze přidávat produkty jednotlivě, nebo pomocí výběru více produktů a následného přidání.



Obrázek 4 Diagram pro přidání produktu

5 POUŽITÉ TECHNOLOGIE

V následující kapitole se zaměřím na podrobný přehled technologií, které byly klíčové pro vývoj mé webové aplikace. Každá z těchto technologií představuje základní stavební kámen celého projektu. Specificky se budu věnovat jazyku PHP, jehož široká podpora udržuje jeho popularitu jako volby pro dynamické webové aplikace. Podrobněji popíši Nette Framework, který přináší moderní přístupy k vývoji aplikací v PHP, společně s šablonovacím systémem Latte, který efektivně odděluje logiku aplikace od její prezentace. Neopomenu ani rozšíření pro práci s databázemi, jako jsou Doctrine a Nettrine, a architektonický vzor MVC, který umožňuje udržitelný vývoj a organizaci kódu. V neposlední řadě zmíním jQuery knihovnu, která výrazně ulehčuje některé procesy na straně uživatele.

5.1 PHP

PHP je skriptovací jazyk na straně serveru, vytvořený v roce 1994 Rasmusem Lerdorfem, původně známý jako "Personal Home Page", dnes "PHP: Hypertext Preprocessor". Primárně slouží k rychlému vývoji dynamicky generovaných webových aplikací. PHP je kompatibilní s mnoha databázovými systémy a operačními systémy, což umožňuje vysokou flexibilitu při vývoji. Integrace do HTML, podpora objektově orientovaného programování od verze 5 a další vylepšení v novějších verzích, jako je zlepšený výkon a typová deklarace, udržují PHP relevantní i ve světě moderního webového vývoje. PHP, spravováno PHP Group, se těší širokému používání a popularitě, také díky jeho open source povaze pod licencí PHP. [13]

5.2 Nette

Nette Framework, který se začal formovat v roce 2004 díky iniciativě Davida Grudla, sehrál důležitou roli v evoluci PHP frameworků. Motivací pana Grudla pro tvorbu vlastního řešení byla nespokojenost s tehdejšími dostupnými nástroji pro vývoj aplikací v PHP. Jeho cílem bylo vytvořit framework, který by podporoval efektivní vývoj webových aplikací a zároveň respektoval principy bezstavového protokolu HTTP.

V kontextu vývoje softwaru byly využívány různé přístupy. Například frameworky v jazycích Java a .NET využívaly znovupoužitelné UI komponenty. Tyto metody však Grudla nepřesvědčily, především kvůli jejich neefektivnímu zpracování stavovosti nad bezstavovým HTTP. Inspiraci pro Nette tak čerpal nejen z existujících webových technologií, ale i z desktopového vývoje, konkrétně z nástroje Delphi, jenž umožňoval skládání aplikací z komponent.

První veřejná prezentace Nette proběhla v roce 2007 na akci v Praze, kde Grudl představil základní principy a funkcionalitu svého frameworku. Mezi klíčové prvky patřil například komponentový model a obousměrný router, což byly inovace, jež značně ovlivnily další vývoj PHP frameworků.

Další vývoj Nette byl charakterizován snahou o integraci principů dependency injection (DI), což byl významný odchyl od dříve používaného service locatoru. Diskuze s Václavem Purchartem, odborníkem na DI, pomohly Grudlovi vytvořit DI kontejner pro Nette, který se stal klíčovým prvkem frameworku.

Rozdělení Nette na samostatné balíčky, reagující na popularitu nástroje Composer, umožnilo vývojářům lépe spravovat a rozvíjet jednotlivé komponenty. Tento přístup nejenže zjednodušil údržbu a aktualizace, ale také zvýšil flexibilitu v různých projektech.

Uznání, které Nette obdrželo v rámci komunity vývojářů, včetně vysokého umístění v anketách o nejlepší PHP frameworky, odráží jeho význam a vliv na vývoj moderních webových aplikací. Průběžný vývoj a inovace, jakými prošel od svého zrodu, činí z Nette jednu z klíčových platforem pro vývoj v PHP.

Ve své diplomové práci jsem se rozhodl využít framework Nette především kvůli jeho pokročilé podpoře asynchronních volání handlerů. Klíčovou vlastností, která stojí za výběrem Nette, jsou snippety, tedy dynamické výstřižky, jež umožňují aktualizovat konkrétní části webové stránky bez nutnosti jejího celkového znovunačítání. Tento přístup nejen značně urychluje interakci s aplikací, ale také poskytuje uživatelům plynulejší a příjemnější zážitek z prohlížení. [14]

Snippety v rámci Nette fungují na jednoduchém principu. Při návštěvě stránky se načtou všechny její části včetně snippetů. Když pak uživatel provede nějakou interakci, jako je odeslání formuláře nebo kliknutí na tlačítko, vygeneruje se místo celostránkového načítání specifický AJAXový požadavek. Tento požadavek je zpracován na serveru, kde se rozhodne, které části stránky, neboli které snippety, je třeba aktualizovat. Následně jsou tyto aktualizované snippety odeslány zpět do prohlížeče ve formě JSON pole. Klient pak tyto data použije k obnovení pouze změněných částí stránky, což vede k výrazné úspoře dat a zlepšení reakční doby aplikace. Pokud se zkombinuje technologie snippetů s knihovnou Naja, pak lze lehce proměnit klasické volání na asynchronní. K tomu stačí pouze přidat třídu s názvem „ajax“ do HTML elementu a Naja se o vše postará. Na backendové části stačí pouze určit které snippety se mají překreslit.[15]

5.3 Latte

Sofistikovaný a výkonný šablonovací systém představuje neoddělitelnou součást frameworku Nette. Latte revolucionizuje způsob, jakým se vytváří a spravují dynamické webové aplikace. Jeho promyšlené syntaxe a široké možnosti umožňují vývojářům efektivně a elegantně oddělit prezentaci od logiky aplikace. Srozumitelnost, a snadná rozšiřitelnost činí z Latte ideální volbu pro tvorbu šablon ve webových projektech. Tento šablonovací nástroj nabízí bohatou sadu funkcí, včetně dědičnosti šablon, zanořených bloků, podmínek, cyklů a automatické úpravy výstupu, což umožňuje vývojářům vytvářet komplexní a dynamické uživatelské rozhraní s minimálním úsilím. Pro účely této práce byl nástroj Latte využit pro tvorbu šablonové vrstvy webové aplikace, přičemž jsem se zaměřil na jeho klíčové vlastnosti jako jsou dědičnost šablon a makra. [16]

5.4 Doctrine/Netrine

Doctrine a Netrine jsou v rámci frameworku Nette klíčové knihovny pro práci s databází. Poskytují efektivní a elegantní řešení pro správu a manipulaci s daty. Doctrine je objektově relační mapovací (ORM) nástroj, který umožňuje mapovat objektové modely na relační databázové tabulky a pracovat s nimi pomocí objektově orientovaných operací, což zjednodušuje práci s databází a snižuje potřebu psaní SQL dotazů ručně. Netrine pak poskytuje integraci Doctrine do frameworku Nette, což umožňuje vývojářům využívat výhody Doctrine ORM přímo v rámci svých Nette aplikací.

Doctrine je výkonný ORM nástroj napsaný v PHP, který nabízí pokročilé funkce pro práci s databází, včetně správy relací, migrací schémat, dotazování a správy transakcí. S využitím Doctrine mohou vývojáři definovat entity reprezentující datový model aplikace a pracovat s nimi pomocí objektových operací, což zvyšuje produktivitu a snižuje riziko chyb. Doctrine umožňuje inteligentně cashovat získaná data a optimalizovat tak přístupy do databáze, nebo optimalizovat samotné dotazy. [17][18]

V rámci své práce se zaměřuji na využití Doctrine/Netrine pro práci s databází a správu datové vrstvy aplikace. Kombinace těchto nástrojů poskytuje robustní a flexibilní řešení pro práci s databází v rámci frameworku Nette, díky čemuž lze efektivně spravovat a manipulovat data v aplikaci.

5.5 MVC

Model-View-Controller (MVC) je architektonický vzor, který se stal standardem při vývoji softwarových aplikací díky své schopnosti oddělit různé aspekty aplikace a zlepšit tak

modularitu, udržitelnost a obnovitelnost kódu. Tento vzor rozděluje aplikaci do tří hlavních komponent:

5.5.1 Model

Reprezentuje datový model aplikace a zajišťuje manipulaci s daty a obchodní logiku. Modely jsou zodpovědné za získávání, ukládání a manipulaci s daty a poskytují také rozhraní pro komunikaci s databází nebo jinými zdroji dat. Jejich účelem je oddělit logiku aplikace od prezentace a udržet data konzistentní a spravovatelná.

5.5.2 View

Reprezentuje uživatelské rozhraní aplikace a zajišťuje zobrazení dat uživateli. View komponenta zpracovává prezentaci dat a interakci s uživatelem. Jejím úkolem je vykreslovat data z modelu uživateli a zobrazovat je ve vhodné formě, často ve formě HTML stránek nebo uživatelského rozhraní.

5.5.3 Controller

Zpracovává vstupy uživatele a rozhoduje, jaká akce má být provedena. Controller komponenta funguje jako prostředník mezi modelem a pohledem, zajišťuje tok dat mezi nimi a řídí běh aplikace na základě uživatelských požadavků. Jeho úkolem je přijímat vstupy od uživatele, interpretovat je a následně aktualizovat stav modelu nebo vybrat odpovídající pohled pro zobrazení výsledků. V případě Nette frameworku se používá Presenter, který nese stejný význam, ale jiné pojmenování.

MVC architektura umožňuje oddělení obchodní logiky, prezentace a uživatelské interakce, což vede k čitelnějšímu, udržitelnějšímu a škálovatelnějšímu kódu. Tento přístup také usnadňuje testování a údržbu aplikace a umožňuje vícečlenný tým pracovat na různých částech aplikace současně. [19]

5.6 jQuery

jQuery je populární a mocná knihovna JavaScriptu, která zjednodušuje manipulaci s HTML dokumenty, událostmi, animacemi a AJAX interakcemi ve webových aplikacích. Byla vytvořena s cílem zjednodušit a zkrátit psaní kódu v JavaScriptu a zajistit konzistentní chování na různých webových prohlížečích.

Jedním z hlavních důvodů popularity jQuery je její jednoduchá syntaxe, která umožňuje snadnou navigaci v DOM stromu a manipulaci s prvky na stránce. Díky selektorům podobným těm, které jsou známé z CSS, mohou vývojáři rychle vyhledávat a manipulovat s prvky na

stránce. To znamená, že lze snadno přidávat události, animace, efekty a dynamicky měnit obsah stránky bez zbytečného opakování kódu.

Další významnou funkcí jQuery jsou události, které umožňují vývojářům reagovat na interakce uživatele s webovou stránkou. Knihovna jQuery poskytuje jednotné rozhraní pro zachycení a zpracování událostí, jako jsou kliknutí, změny stavu, klávesové stisky atd., což výrazně usnadňuje práci s interaktivními prvky na stránce.

Knihovna jQuery rovněž zjednodušuje práci s asynchronními žádostmi na server pomocí AJAX technologie. Poskytuje jednoduché metody pro odesílání a přijímání dat ze serveru bez nutnosti obnovovat celou stránku, což vede k lepší uživatelské zkušenosti a větší dynamice webových aplikací. [20]

5.7 MariaDB

MariaDB Server patří mezi nejoblíbenější databázové servery na světě. Vznikl díky původním vývojářům MySQL. Mezi jeho významné uživatele patří giganti jako Wikipedia, WordPress.com a Google.

MariaDB se využívá k přeměně dat do strukturovaných informací v různorodých aplikacích, od bankovníctví až po webové stránky. Původně koncipován jako vylepšená a plně kompatibilní alternativa k MySQL, MariaDB Server si získal popularitu díky své rychlosti, škálovatelnosti a odolnosti. Jeho bohatý ekosystém úložných enginů, pluginů a dalších nástrojů z něj činí velmi flexibilního hráče pro mnoho různých účelů.

MariaDB Server je vyvíjen jako open source software a jakožto relační databáze poskytuje uživatelům SQL rozhraní pro přístup k datům. Nejnovější verze MariaDB Serveru nabízí také pokročilé funkce pro práci s geografickými informačními systémy a formátem JSON. [21]

6 PLATFORMA

Umístění webového serveru s informačním systémem na lokální server, jako je Raspberry Pi, přináší několik klíčových výhod, které mohou být zásadní pro malé až středně velké podniky nebo pro individuální projekty. Tento přístup kombinuje nízké náklady, flexibilitu a kontrolu nad prostředím, což vytváří jedinečné benefity pro uživatele a správce systému.

6.1 Nízké náklady

Raspberry Pi je cenově dostupná platforma, jejíž pořizovací cena tvoří pouze zlomek ceny tradičních serverových řešení. Kromě nízkých pořizovacích nákladů jsou i náklady na provoz minimální, vzhledem k nízké spotřebě elektrické energie. Toto řešení je ideální pro malé projekty nebo start-upy, které potřebují efektivní řešení s omezeným rozpočtem.

6.2 Flexibilita a rozšiřitelnost

Raspberry Pi poskytuje významnou míru flexibility. Díky své kompaktní velikosti a modulární povaze může být snadno upraven nebo rozšířen o další komponenty v závislosti na potřebách projektu. To umožňuje uživatelům přizpůsobit si server přesně podle požadovaných specifikací, což znamená velkou výhodu oproti některým předkonfigurovaným řešením.

6.3 Kontrola nad prostředím

Hostování informačního systému na lokálním serveru umožňuje plnou kontrolu nad prostředím. To zahrnuje možnost přímého přístupu k hardware, což je výhodné pro účely údržby, bezpečnosti a přizpůsobení. Administrátoři mají plnou kontrolu nad softwarovým stackem, což umožňuje instalaci a konfiguraci specifického softwaru dle potřeby, včetně bezpečnostních aktualizací a patchů.

6.4 Bezpečnost a soukromí

Vlastnictví lokálního serveru také zvyšuje bezpečnost a soukromí dat. V době rostoucích obav z úniku a zneužití dat na internetu poskytuje lokální hosting jistotu, že data nejsou sdílena nebo ukládána na externích serverech, nad kterými nemají organizace přímou kontrolu. Kromě toho je možné implementovat pokročilé bezpečnostní protokoly a firewally, které jsou specificky navrženy pro ochranu citlivých informací.

6.5 Zálohování a obnova systému

Jelikož je celý systém implementován na SD kartě, byla vytvořena kopie této karty, a to včetně všech nastavení. Tím by mělo být zajištěno rychlé obnovení systému v případě poškození

uložiště. Prakticky stačí v případě komplikací vyměnit kartu a následně obnovit databázi ze zálohy. Databáze je v pravidelných intervalech zálohována; jednak na připojené externí uložení, přímo propojené se zařízením, a dále také na vzdálené zabezpečené uložení. K zálohování aktuálně dochází jednou denně, což by mělo být dostačující. Jelikož je zařízení a aplikace stále ve fázi testování, nedošlo k dimenzování uložení. Do budoucna se bude přecházet na systém běžící na klasickém SSD disku, který má lepší vlastnosti a kapacitu. Nyní je připojen jako sekundární uložení.

Použití Raspberry Pi jako lokálního serveru pro hostování webového serveru s informačním systémem nabízí jedinečnou kombinaci nízkých nákladů, flexibility, kontrolu nad prostředím, zvýšenou bezpečnost a výukový potenciál. Tato řešení jsou ideální pro malé a středně velké projekty, kde je důležitá kontrola nad daty a infrastrukturou a přijatelná cena.

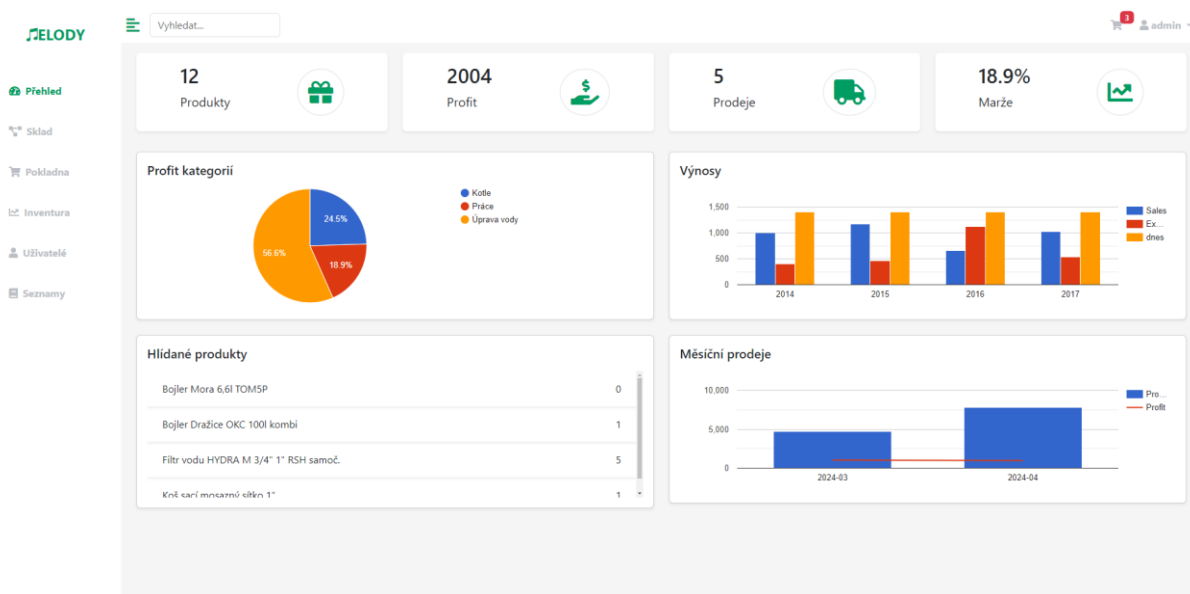
7 PROSTŘEDÍ

V kapitole prostředí se zaměřuji na funkcionalitu navrhované aplikace a její rozhraní implementované v rámci této práce. Vysvětluji, jakým způsobem jsem řešil některé scénáře a jaké kroky jsem podstupoval pro lepší uživatelský zážitek.

7.1 Dashboard

Systemový dashboard byl navržen s ohledem na specifikované požadavky a poskytuje statistický přehled skladových zásob a prodejů. Pro realizaci layoutu celého dashboardu byl zvolen přístup založený na implementaci karet, což zajišťuje jeho plnou responzivitu na různých zařízeních. Tento přístup umožňuje uživatelům efektivní navigaci a zlepšuje celkovou uživatelskou přívětivost aplikace.

V rámci dashboardu jsou prezentovány číselné statistiky, které reflektují výkonnost v oblasti prodejů a marží, spolu s obecnými přehledy relevantních dat. Pro vizualizaci grafů a diagramů byla zvolena knihovna Google Charts. Tato knihovna byla vybrána na základě její schopnosti poskytovat srozumitelné, vizuálně atraktivní a animované zobrazení dat, což přispívá k lepší interpretaci a porozumění předkládaných informací. Využití knihovny Google Charts dále umožňuje snadnou integraci a adaptabilitu v rámci webových aplikací, což představuje zásadní výhodu pro dynamickou prezentaci statistických dat.



Obrázek 5 Hlavní stránka aplikace

7.2 Tabulka skladových položek

Centrálním modulem aplikace je tabulka skladových položek, která je implementována pomocí knihovny jQuery. Tato klíčová funkčnost poskytuje komplexní přehled důležitých informací o položkách určených k prodeji. Design tabulky je pečlivě navržen tak, aby zajistil její plnou responzivitu a podporoval interaktivitu uživatele prostřednictvím integrovaných tlačítek, která nabízí různé možnosti správy inventáře.

Aplikace klade značný důraz na optimalizaci uživatelského zážitku, čehož je dosaženo využitím asynchronních požadavků realizovaných skrze AJAX. Tato metodika umožňuje dynamické aktualizace relevantních segmentů stránky bez nutnosti jejího celkového přenačítání, což významně zlepšuje plynulost a rychlost interakce s aplikací. Snippetová architektura frameworku Nette je v tomto procesu využívána pro efektivní manipulaci s obsahem, což zvyšuje reaktivitu aplikace.

Pro asynchronní odesílání formuláře bez dodatečných úprav jsem využil rozšíření Naja. Pro použití tohoto volání stačí do tříd prvku přidat třídu s názvem ajax a formulář je odesílán ajaxově. Dále je třeba inicializovat rozšíření ve scriptu pomocí:

```
naja.initialize();
```

V případě potřeby zachování čisté URL adresy lze ještě použít následující část kódu, což odstraní stopy po volání v URL adrese.

```
$this->payload->postGet = true;  
$this->payload->url = $this->link('this');
```

Počáteční zobrazení tabulky primárně reflektuje skladové zásoby. Přepínač umístěný v horní části uživatelského rozhraní však umožňuje rozšíření zobrazení o položky momentálně nedostupné na skladě. Jelikož jsou data do tabulky načítána pomocí asynchronního volání, je možné do budoucna vytvořit více podobných filtrů. Tento adaptivní přístup zajišťuje, že uživatelé mají kompletní přehled o všech položkách, včetně těch s nulovým stavem zásob.

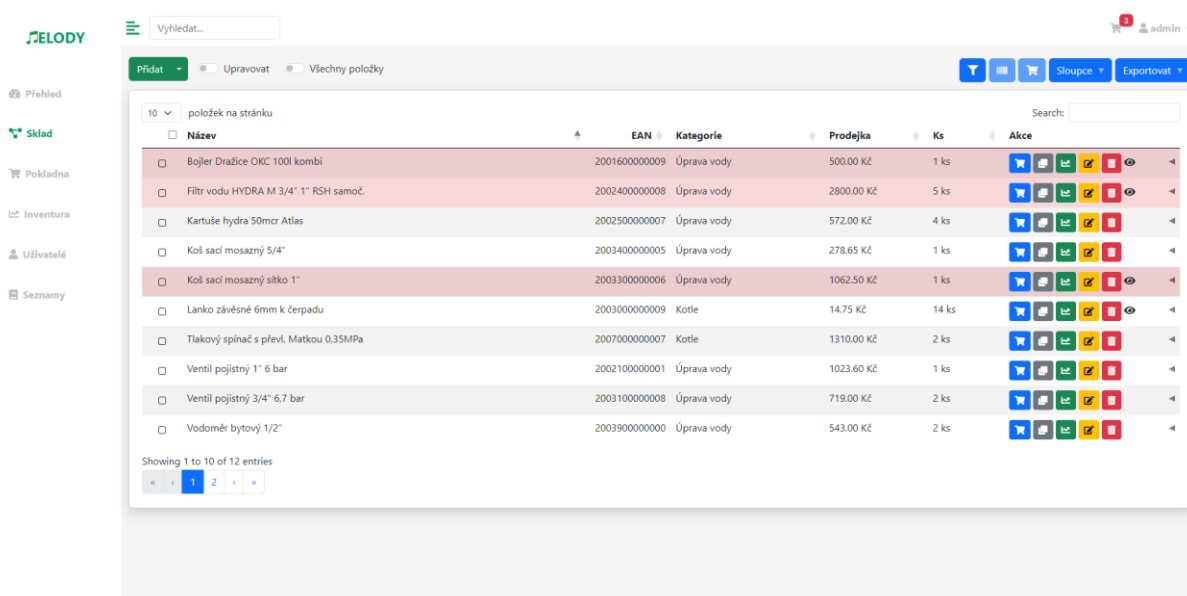
Integrace tlačítek pro editaci, kopírování, úpravu a mazání přímo do záznamů v tabulce přispívá k vysoké míře uživatelského pohodlí a efektivity při správě zásob. Možnosti, jako jsou generování EAN-13 kódů či přidávání položek do nákupního košíku, dále rozšiřují funkční spektrum aplikace. Podpora pro vyhledávání a filtraci podle kategorií produktů umožňuje rychlou a přesnou selekci dat.

Výše uvedené charakteristiky skladové tabulky a přístupy k její integraci do aplikace ilustrují závazek k poskytování intuitivního, efektivního a ergonomicky optimalizovaného uživatelského rozhraní.

Nad primární skladovou tabulkou je umístěn přepínač, který umožňuje aktivovat režim editace pro jakýkoliv záznam v tabulce. Po aktivaci této funkce je uživateli umožněno kliknout na libovolnou buňku v tabulce, která se následně transformuje na editovatelné textové pole. Uživatel může do tohoto pole vložit nové hodnoty, a po odstranění fokusu z buňky (například kliknutím mimo ni) jsou aktualizovaná data asynchronně odeslána do databáze prostřednictvím AJAXu. Na backend systému se odešle pouze unikátní identifikátor produktu, název atributu, který se mění a jeho nová hodnota. Tento proces je navržen tak, aby byl pro uživatele co nejméně rušivý, přičemž ho o výsledku informuje notifikace o úspěšné aktualizaci položky.

Aby bylo zajištěno, že data zůstanou konzistentní a správně kategorizovaná, existuje výjimka při editaci kategorií záznamů. V případě, že uživatel klikne na buňku obsahující kategorii, místo transformace na běžné textové pole se zobrazí výběrový seznam (select element) obsahující všechny dostupné kategorie. Po výběru požadované kategorie z tohoto seznamu jsou data aktualizována stejným mechanismem, jaký je použit u ostatních buněk. V případě uživatelského smazání záznamu, kde se nachází číslo pro čárový kód EAN, je prostřednictvím backendu vygenerován nový na základě unikátního identifikátoru.

Tento intuitivní mechanismus editace zvyšuje uživatelskou efektivitu a zjednodušuje správu dat bez nutnosti opustit kontext skladové tabulky. Celkově je tento přístup navržen tak, aby poskytoval hladký a bezproblémový uživatelský zážitek při aktualizaci dat, zatímco zároveň zachovává integritu a správné kategorizování informací.



Název	EAN	Kategorie	Prodejka	Ks	Akce
Bojler Dražice OKC 100l kombi	2001600000009	Úprava vody	500.00 Kč	1 ks	[edit] [delete] [refresh] [add]
Filtr vody HYDRA M 3/4" 1" RSH samoč.	2002400000008	Úprava vody	2800.00 Kč	5 ks	[edit] [delete] [refresh] [add]
Kartuše hydra 50mcr Atlas	2002500000007	Úprava vody	572.00 Kč	4 ks	[edit] [delete] [refresh] [add]
Koší sací mosazný 5/4"	2003400000005	Úprava vody	278.65 Kč	1 ks	[edit] [delete] [refresh] [add]
Koší sací mosazný sítko 1"	2003300000006	Úprava vody	1062.50 Kč	1 ks	[edit] [delete] [refresh] [add]
Lanko závěsné 6mm k čerpadu	2003300000009	Kotle	14.75 Kč	14 ks	[edit] [delete] [refresh] [add]
Tlakový spínač s pleví. Matkou 0.35MPa	2007000000007	Kotle	1310.00 Kč	2 ks	[edit] [delete] [refresh] [add]
Ventil pojistný 1" 6 bar	2002100000001	Úprava vody	1023.60 Kč	1 ks	[edit] [delete] [refresh] [add]
Ventil pojistný 3/4" 6,7 bar	2003100000008	Úprava vody	719.00 Kč	2 ks	[edit] [delete] [refresh] [add]
Vodoměr bytový 1/2"	2003900000000	Úprava vody	543.00 Kč	2 ks	[edit] [delete] [refresh] [add]

Obrázek 6 Skladová tabulka

V rámci implementace informačního systému pro správu firemního majetku je uživatelům umožněno zadávat položky majetku buď individuálně nebo s použitím funkce hromadného vkladu. Pro účely hromadného vkladu byl vyvinut specifický algoritmus umožňující import dat z excelových souborů, které jsou formátovány v souladu s firemními dokumentačními standardy. Tyto dokumenty jsou kategorizovány dle typu majetku, což usnadňuje jejich zpracování. Při importu souboru systém automaticky identifikuje název kategorie a provádí vyhledání odpovídající kategorie v databázi. V případě, že kategorie existuje, je přiřazena k importovaným položkám. Neexistuje-li daná kategorie, je vytvořena nová. Analogicky, systém vyhledává existenci produktů podle názvu; neexistuje-li záznam, je vytvořena nová položka včetně zařazení do příslušné kategorie. Vzhledem k tomu, že EAN-13 kódy dosud nebyly evidovány, jsou generovány na základě unikátního identifikátoru přiděleného databázi. Pro vkládání individuálních položek byl vytvořen uživatelsky přívětivý formulář, umístěný v bočním panelu uživatelského rozhraní. Tento formulář je navržen tak, aby zjednodušil proces zadávání nových položek tím, že požaduje pouze nezbytné informace. Pro detailnější specifikace položek je možné využít rozšiřující menu, které odhalí další volby.

Marže
10.28 %
26.13 %
20.17 %
36.83 %
62.50 %
25.53 %
10.08 %
17.17 %
20.03 %
20.13 %

Přidání nového produktu ×

Název:

Sledovat:

Kategorie:

Množství: Kód:

Skenovat položku

Nákupka:	Prodejní cena:	Marže:
<input type="text" value="12"/>	<input type="text" value="12.00"/>	<input type="text" value="0.00"/>
21%	12%	Profit
14.52 Kč	13.44 Kč	0.00 Kč

[Skrýt detaily](#)

Dodavatel:

Shelf:

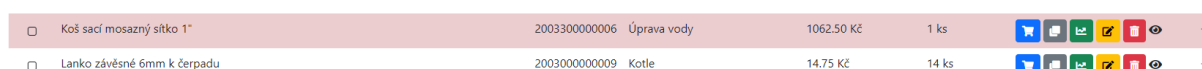
Popis:











Obrázek 7 Přidávání položek

Cenová kalkulace představuje klíčový prvek v procesu definování nové položky. K zjednodušení tohoto procesu byl do formuláře integrován kalkulátor, umožňující uživatelům

zadat nákupní cenu a rozhodnout se mezi aplikací procentuální marže či stanovením pevné prodejní ceny. Na základě zvolené metody jsou automaticky vypočítány a zobrazeny doplňující finanční údaje včetně daně.

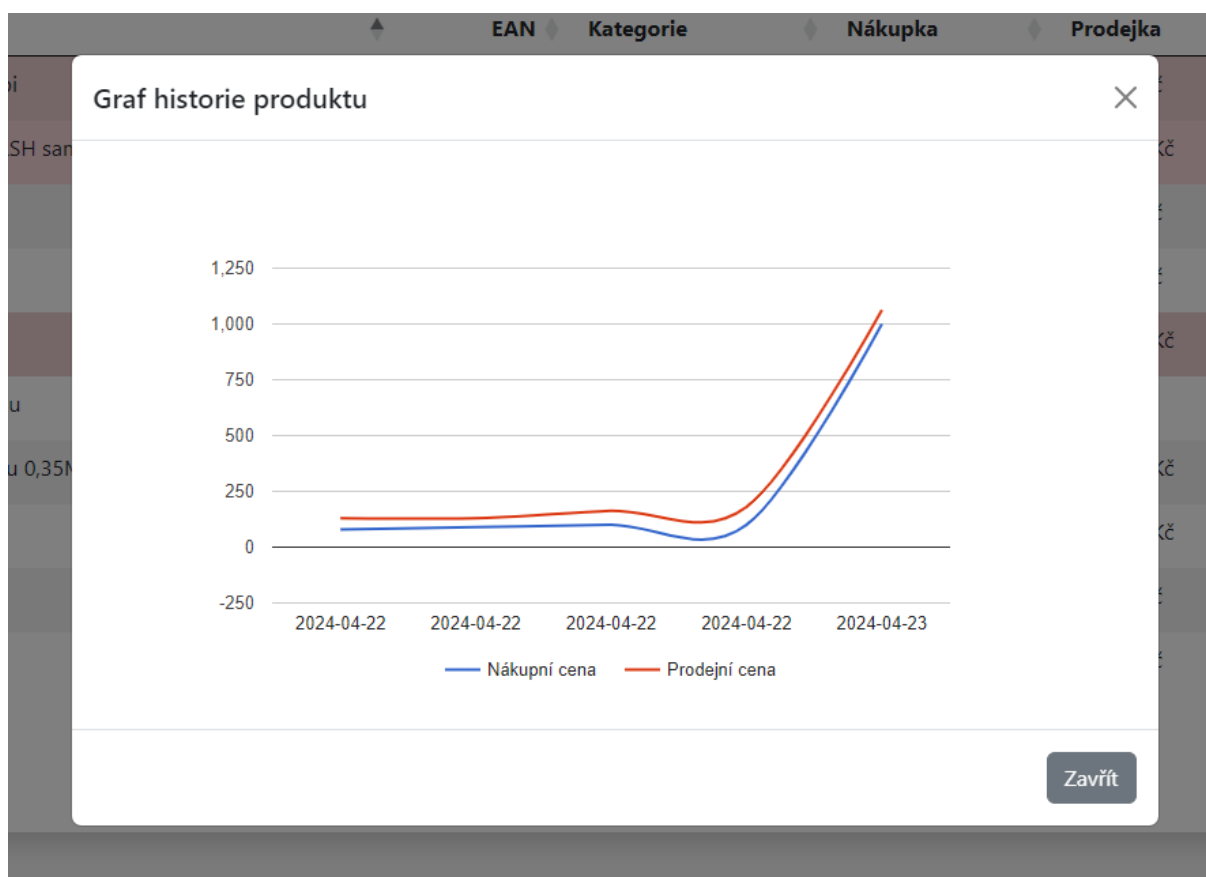
Jak je vidět na předchozím obrázku, při přidávání, nebo upravování produktu, je uživateli umožněno zvolit možnost sledovat produkt. Při zvolení této možnosti se zobrazí dodatečné pole, kde si uživatel zvolí, při jakém množství se má produkt zvýraznit. Pokud sledovaný produkt klesne na sledovanou hodnotu, případně níže, je následně zvýrazněn ve skladové tabulce a také v tabulce sledovaných produktů na domovské stránce. Tato situace je znázorněna v následujícím obrázku, kde máme sledovaný produkt znázorněn červeně, protože kleslo jeho množství až na hranici. Po tímto produktem je druhý, který je sledován, což je znázorněno ikonou oka, ale má v pořádku skladové zásoby.



<input type="checkbox"/>	Koš sací mosazný sítko 1"	2003300000006 Úprava vody	1062.50 Kč	1 ks	    
<input type="checkbox"/>	Lanko závěsné 6mm k čerpadu	2003000000009 Kotle	14.75 Kč	14 ks	    

Obrázek 8 Hlídané položky

Jednou z poslední přidaných funkcí je zobrazení historických cen u jednotlivých produktů. U každé položky nalezneme ikonu grafu. Po kliknutí na tlačítko se odešle požadavek na získání dat o historii cen produktů. Z backendu přijde JSON odpověď s daty, ke které se připojí hlavička s titulkem sloupců. Následně jsou data vykreslena pomocí Google Charts na frontendu. Potřebná data pro tuto historii se získávají prostřednictvím databázových triggerů. Na tabulku produktu je přidělen trigger, který v případě, že se upravuje cena, nebo finální cena, přidá záznam do tabulky s historií. Díky tomu je možné zaznamenávat pohyb ceny jednotlivých produktů v poměru s jejich nákupní cenou. Kromě tohoto triggeru existuje ještě jeden a to pro ukládání historie pohybu skladových zásob. V rámci toho triggeru se v případě editace množství, ať už v rámci naskladnění, nebo prodeje, uloží množství počáteční a změněné. Díky těmto datům je možné sledovat pohyby produktů. V aktuální verzi nemá uživatelské rozhraní možnost tato data vykreslit, ale již nyní mohou administrátorovy pomoci v případě řešení problémů ohledně skladových zásob, například v případě nesrovnalostí při inventuře. Do budoucna budou i tato data zobrazena v rámci administrace produktu a jsou komunikována s firmou společně s dalšími vylepšeními.



Obrázek 9 Graf historie cen

7.3 Pokladna

Sekce pokladny v této aplikaci je klíčovým prvkem pro správu prodejních a fakturačních procesů. Umožňuje načítání položek buď přímo při prodeji nebo během tvorby faktur a je závěrečným krokem před generováním účtenek a faktur. Tento modul poskytuje uživatelům možnost vytvoření a správy více košíků, které mohou reprezentovat specifické zakázky nebo zákazníky. Uživatelé mají možnost mezi těmito košíky plynule přepínat, což umožňuje efektivní úpravu položek podle aktuálních potřeb.

Přidávání položek do košíku je možné realizovat několika způsoby, z nichž každý odpovídá různým provozním situacím a technickému vybavení. Jednou z možností je naskenování čárového kódu pomocí dedikovaného skeneru čárových kódů, což přímo zapisuje informace do vstupního pole systému. V případě, že je aplikace používána na zařízeních bez tiskárny, jako jsou mobilní telefony nebo tablety, je možné čárový kód naskenovat prostřednictvím integrovaného fotoaparátu. Tento proces je podporován externí knihovnou, která ve spolupráci s jQuery přetváří obraz čárového kódu na číselné údaje, jež jsou následně odesílány do databáze k získání odpovídajícího produktu. Přidání produktu může být úspěšné, ale může také vyústit v chybu, pokud skenovaný kód není v databázi rozpoznán, což může být způsobeno například

skenováním položky, která není v systému zalistována, nebo skenováním čárového kódu z jiných systémů.

Další možností je přidání zboží prostřednictvím vyhledávání podle čárového kódu nebo názvu produktu, kdy se v interaktivním vyskakovacím okně zobrazí tlačítko pro přidání položky do košíku. Stejná funkce je dostupná i v tabulce skladových položek, což usnadňuje přístup k nejčastěji prodávaným nebo potřebným produktům.

Každá položka v košíku obsahuje základní informace jako jsou název, množství a cena. U každé položky je možné kdykoli upravit množství, což automaticky aktivuje AJAX požadavek, který aktualizuje celkovou cenu v závislosti na novém počtu položek. Tyto změny se zaznamenávají do databáze, což umožňuje budoucí úpravy košíku nebo sdílení informací o košíku mezi více zařízeními.

System také řeší potenciální problémy spojené s uchováváním neuzavřených košíků, které mohou být kdykoli upraveny. Typickým příkladem je situace, kdy zákazník obdrží cenovou nabídku, ale rozhodne se o jejím přijetí s časovým odstupem. Pokud se v tomto období změní cena produktů, systém upozorní uživatele pomocí výstražného symbolu ve dvou formách. První forma zobrazuje trojúhelník s varováním, že cena v košíku se liší od aktuálně nastavené prodejní ceny, což může být oznámeno ve vyskakovacím okně po kliknutí na symbol. Druhý stav signalizuje změnu barvy položky na světle červenou, což naznačuje, že položka je prodávána za cenu nižší, než je její nákupní cena. V takových případech je vhodné položku v košíku upravit, což lze provést pomocí tlačítka, které otevře modální okno pro editaci ceny.

Úprava ceny ✕

Cena v košíku: 2800 Kč

Aktuální prodejní cena: 2800 Kč

Aktuální nákupní cena: 2220 Kč

Změnit cenu:

Zobrazit podrobnosti o zisku

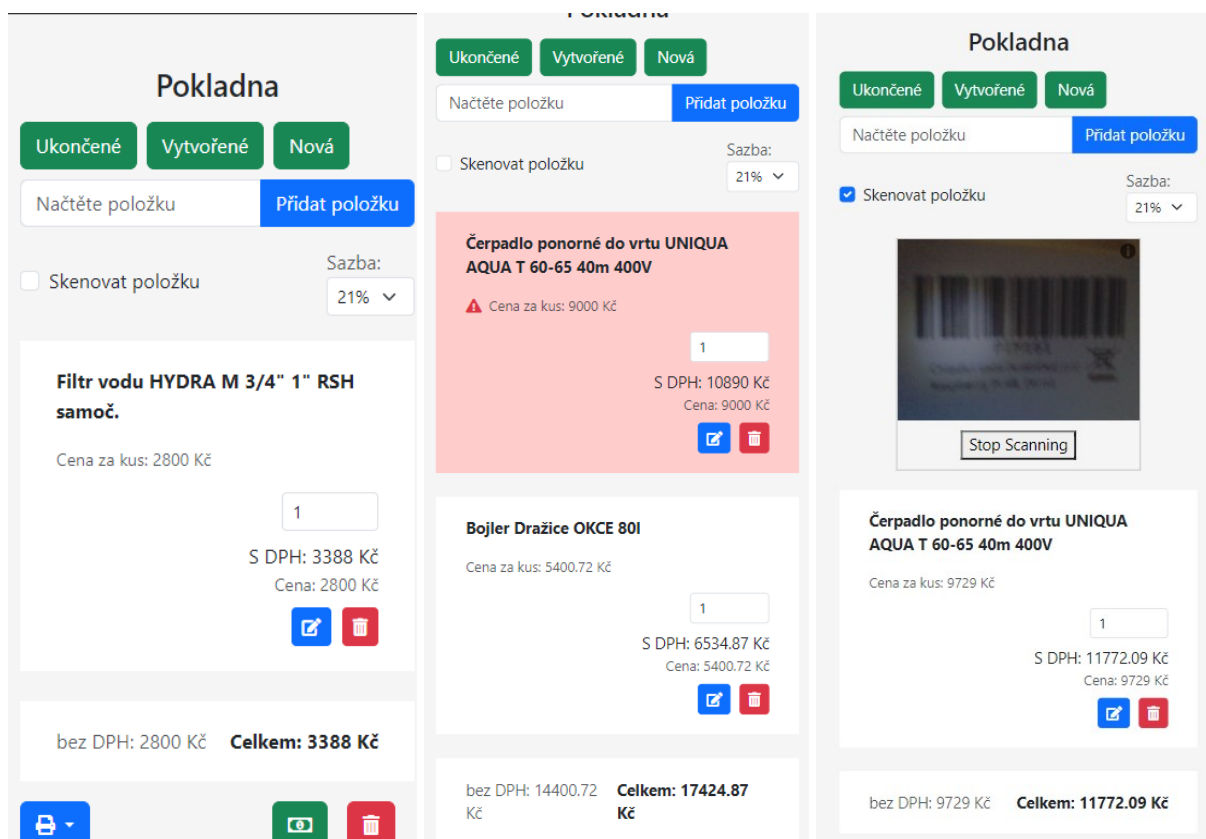
Marže (%): 26.13%

Zisk (Kč): 580.00 Kč

Close Uložit

Obrázek 10 Úprava cen v košíku

Pro jednodušší ladění ceny se zde automaticky přepočítává a zobrazuje případný zisk v korunách společně s procenty marže na produktu. Dále je možné změnit sazbu DPH; v případě, že je to potřebné, ceny se znovu přepočítají podle této nové sazby. V poslední řadě pokladna umožňuje tisknout účtenku přímo na tiskárně terminálu. V systému je tato možnost realizována naformátovaným PDF layoutem s předvolbou tisku. Další kroky jako samotný tisk jsou realizovány vnitřní implementací zařízení. Na následujících obrázcích lze vidět pokladnu ve výchozím zobrazení s jedním produktem. Dále je vyobrazeno samotné skenování čárového kódu a poslední obrázek ukazuje situaci, kdy se položka prodává se ztrátou.

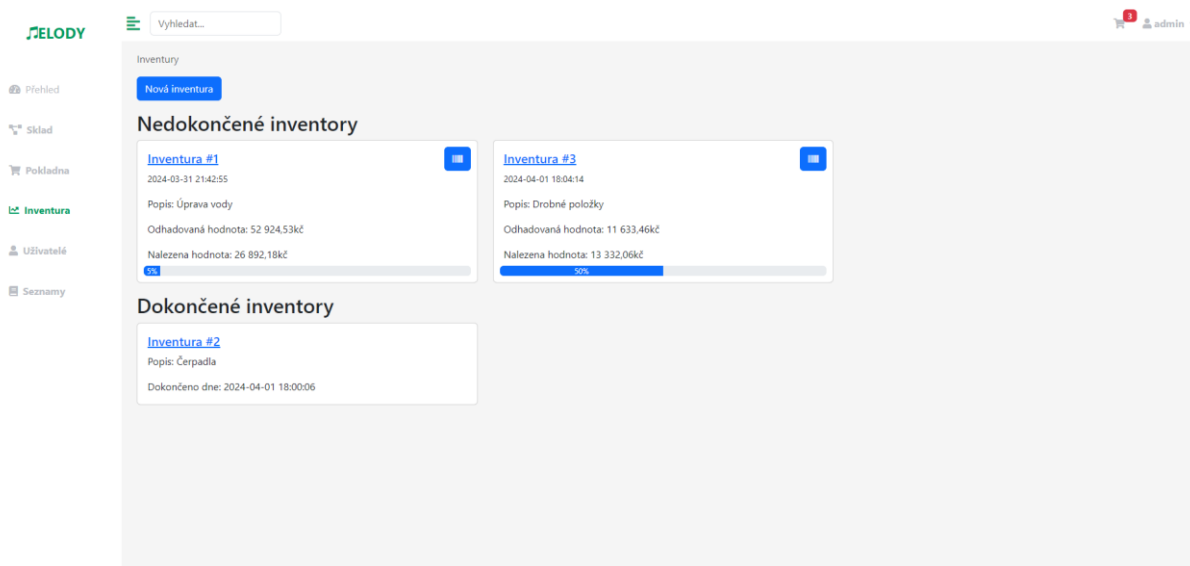


Obrázek 11 Pokladna

7.4 Inventura

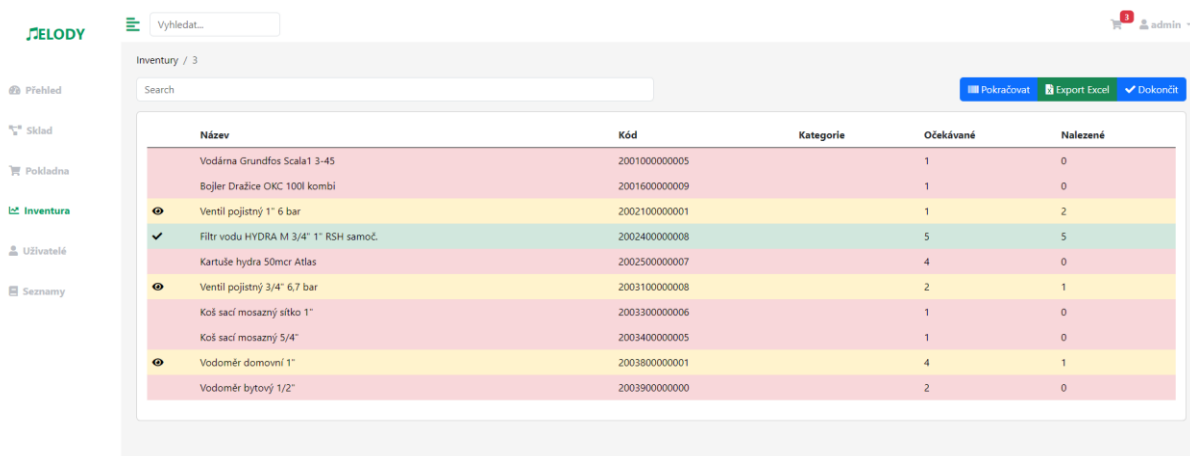
Stránka s inventurou má hned několik funkčních úkolů. Rozděluje se na 4 podstránky. První a zároveň hlavní podstránka obsahuje výpis všech inventur. Vyobrazení jsem volil formou kartiček s akcemi a informacemi. Rozlišuji zde 2 typy kartiček. Prvním z nich jsou nedokončené inventury. V těchto kartičkách se ukazují všechny rozpracované inventury. Jako první lze na kartě vidět identifikátor, který je odkazem na detail inventury. Níže jsou zobrazeny ostatní základní informace jako popis, odhadovaná částka a nalezená částka. Odhadovaná částka je celková suma cen položek, které by měly být nalezeny během inventury. Oproti tomu nalezená hodnota odpovídá hodnotě dosud nalezených produktů. Pro rychlý přehled je zde naimplementován progress bar, který ukazuje procentuální splnění inventury. Jedná se pouze o informační údaj, který je vypočítán na základě očekávaných a nalezených kusů. Jinými slovy, pokud budou nějaké rozdíly na skladu, například uživatel nalezne více položek, může se mu stát, že progress bar bude ukazovat sto procent, ale zároveň bude zaměstnanec stále nacházet položky do této inventury. Tyto situace by měly nastávat minimálně. Pokud má inventura sto procent, neznámá to, že se automaticky stane ukončenou. Tuto akci musí provést uživatel

manuálně, v ideálním případě společně s důkladnou kontrolou výsledků této inventury.



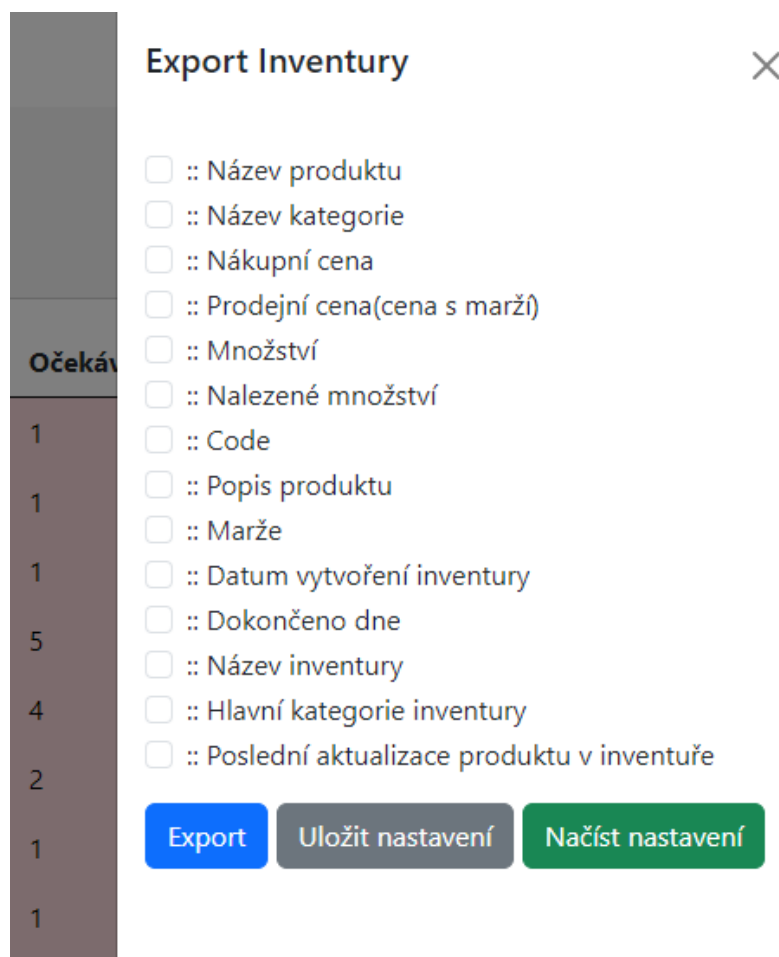
Obrázek 12 Přehled inventur

Je možné se proklikat i k detailu, který lze vyvolat jak na nedokončené tak na dokončené inventury. Tento detail je rychlý přehled všech položek, které by měly být nalezeny během inventury. V tomto tabulkovém seznamu je možné vyhledávat jak podle názvu, tak podle EAN kódu produktu. Na obrázku níže jsou znázorněny jednotlivé řádky tabulky, které mají rozdílné barvy a také různé ikony na začátku řádku. Tímto způsobem jsou vyobrazeny jednotlivé stavy produktu. Prvním a zároveň nejjednodušším stavem je pouze červený řádek bez ikony. Produkty, které jsou takto vyobrazené signalizují, že během inventury nebyl nalezen ani jeden kus této položky. Dalším stavem, do kterého se produkt může dostat, je žluté zbarvení s ikonou oka. Tento typ signalizuje nutnost kontroly položky. Nastane tehdy, když je nalezeno méně či více kusů, než je očekáváno. Posledním stavem je zelený řádek s ikonou zaškrtnutí, který značí, že vše u tohoto produktu sedí a není třeba žádná speciální kontrola, nebo dohledávání.



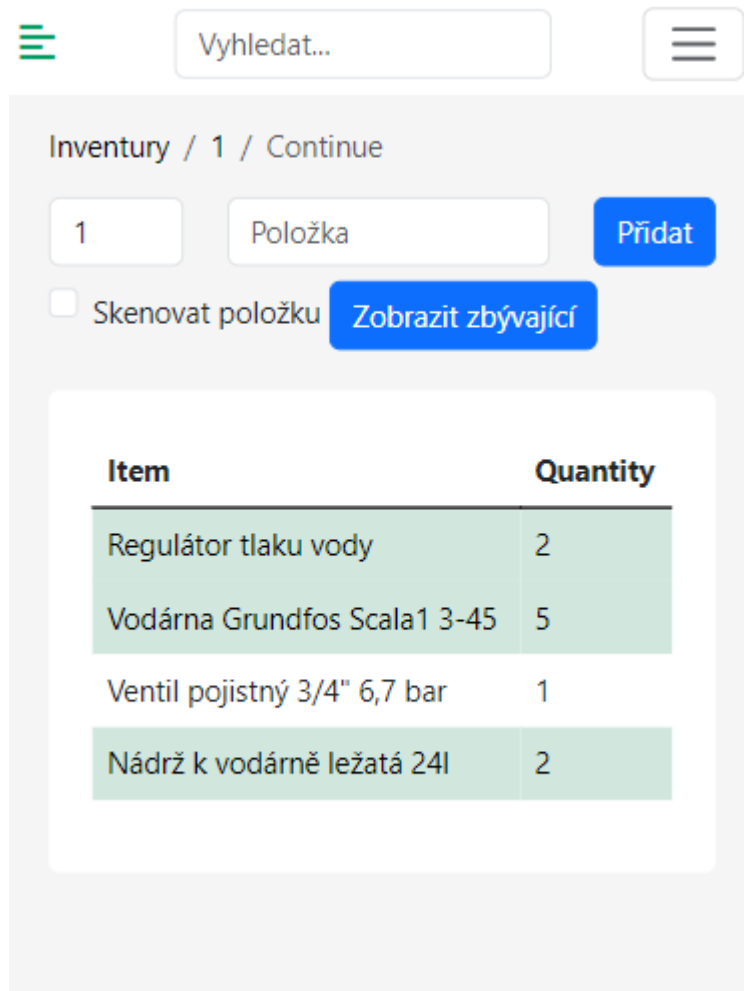
Obrázek 13 Detail inventury

Celý tento přehled je možné exportovat do excelového listu s vlastním nastavením. Lze zde ovlivnit, jaká data budou exportována pomocí zaškrťovacího pole u vlastností, a případně také měnit jejich pořadí. Pokud plánuje uživatel export se stejnými parametry používat častěji, může si uložit nastavení exportu a při příští návštěvě si preset načíst. Okno pro export je vyobrazeno na následujícím obrázku.



Obrázek 14 Export inventury

Pokud uživatele nezajímá detail inventury a raději by rovnou přešel k jejímu pokračování, může tak učinit pomocí ikonky čárového kódu u inventury, případně z detailu skrze stejnou ikonku.



Obrázek 15 Načítaná položek inventury

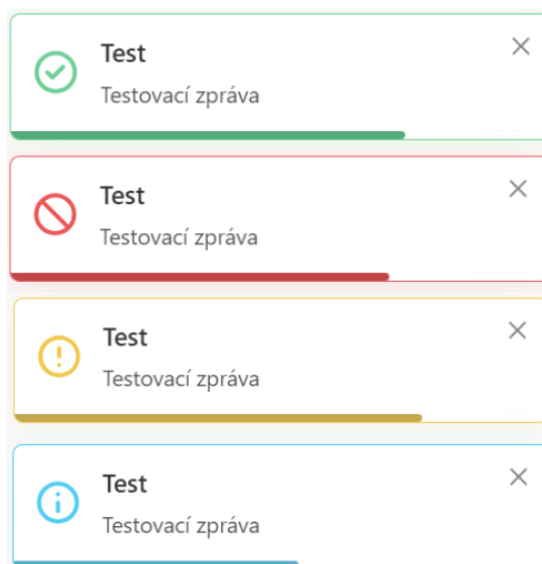
Na obrázku je ukázka grafického rozhraní zobrazeného na mobilním zařízení při volbě možnosti pokračovat ve skenování inventury. Zde se uživateli zobrazí seznam položek, které naskenoval on či jiný uživatel s rozpracovanou stejnou inventurou. Dalším krokem při pokračování inventury je volba množství, které bylo u dané položky skutečně nalezeno; ve výchozím stavu je hodnota nastavena na 1. Následně je vhodné naskenovat čárový kód produktu pomocí externího skeneru. Pokud by byla inventura prováděna na mobilním zařízení, lze položku skenovat přímo pomocí fotoaparátu zařízení stejným způsobem jako tomu je u načítání položky do košíku. Po samotném načtení je seznam bez obnovení stránky aktualizován a seřazen. Poslední naskenovanou položku systém zobrazí úplně nahoře. Také v tomto přehledu má uživatel možnost rozpoznat, jestli dané počty sedí. Jeho prvním indikátorem je zelené zbarvení položky. Je tu však drobná změna. Položky zezelenají i v případě že bylo nalezeno více kusů, než bylo původně očekáváno. Je to z důvodu, že inventura by měla být správně prováděna takzvaně naslepo, kdy zaměstnanci neznají množství, které by měli najít, ale skenují pouze reálné kusy na skladě. Tato indikace slouží jako rychlý přehled, že vše sedí, nikoli aby

se tím zaměstnanec řídil. Pokud má uživatel dostatečná oprávnění, je mu umožněno zobrazit zbývající položky, které ještě nenašel. Toto vyobrazení je realizováno prostřednictvím bočního vyjíždějícího menu se seznamem, ve kterém lze položky vyhledávat a také kontrolovat.

Pokud chce uživatel vytvořit novou inventuru a má na to oprávnění, přesune se přes odkaz z přehledu na vytvoření nové inventury. Při prvním načtení je uživateli zobrazen seznam všech položek, které jsou aktuálně na skladě společně s jejich počty. Po levé straně je připravený formulář na vytvoření nové inventury. Pokud je ve výchozím stavu spuštěna inventura bez jakékoli úpravy, spustí se tak jedna velká inventura, která zahrnuje celý sklad. Pokud by uživatel chtěl dělat inventuru pouze jednu konkrétní kategorii, vybere ze stromového menu s kategoriemi právě vybranou kategorii. Tímto způsobem se automaticky vyplní popis inventury na název dané kategorie a stromové menu se změní na menu s podkategoriemi. Pokud je na inventuru vybrána kategorie, je třeba myslet na to, že všechny potomci neboli podkategorie jsou v inventuře zahrnuti také. Pokud nějaké existují, zobrazí se ve stromovém menu, jehož kořenem je nyní zvolená kategorie. Také seznam položek se upraví tak, aby obsahoval veškeré položky podkategorií. Pokud je uživatel s výběrem spokojen, může inventuru vytvořit a zadat ji pracovníkovi ke zpracování.

7.5 Flash zprávy

Aby bylo možné uživatelům jasně prezentovat odpovědi na formuláře a jiné akce v aplikaci, přidal jsem do projektu systém vyskakovacích oken s časovačem pro zobrazení různých zpráv. Projekt rozlišuje čtyři typy těchto zpráv, které jsou prezentovány v následujícím obrázku.



Obrázek 16 Notifikace

Tyto zprávy mohou být generovány klasickými flash messages z backendu aplikace, ale rozhodl jsem se vytvořit alternativu umožňující volání z JavaScriptu, což je užitečné například pro zpracování chyb z asynchronních volání. Tento přístup zlepšuje uživatelský zážitek, protože časově omezené zprávy nevyžadují přímé zavírání uživatelem. Nicméně pokud mu způsobují obtíže, může je zavřít před vypršením času i ručně. K vytvoření takové zprávy stačí zadat její typ a obsah. V rámci frameworku Nette se flash zprávy posílají prostřednictvím relací (sessions). Vzhledem k tomu, že nedochází vždy k úplnému obnovení stránky, jsem musel začlenit i část kódu pro vykreslení zpráv do snippetů, což umožňuje kdykoli zprávy aktualizovat nebo obnovovat; případně se dá využít zmíněné volání ze scriptu.

7.6 Uživatelé a role

Ve webové aplikaci jsem ve spolupráci s firmou vytvořil univerzální přístupový systém, který podporuje správu uživatelských rolí a jejich oprávnění. Při počátečním nasazení budou k dispozici pouze dvě základní role: administrátor a vedoucí pracovník. Obě role mají podobná oprávnění, ale administrátor má navíc pravomoc vytvářet nové uživatele a role. Pokud bude potřeba vytvořit další uživatele nebo role, systém to umožní dynamicky.

Pro usnadnění přidávání uživatelů a rolí byl navržen formulář s přednastavenými zaškrtačovacími poli, která obsahují všechny možné oprávnění. Oprávnění jsou pevně definována, protože jejich ověření je začleněno přímo do kódu. Při přihlášení uživatele se jeho role a oprávnění načítají do jeho identity, čímž se eliminuje potřeba opakovaného načítání dat. Oprávnění jsou kategorizována podle jednotlivých stránek, ke kterým se vztahují, a tyto stránky jsou staticky definovány v konfiguračním souboru.

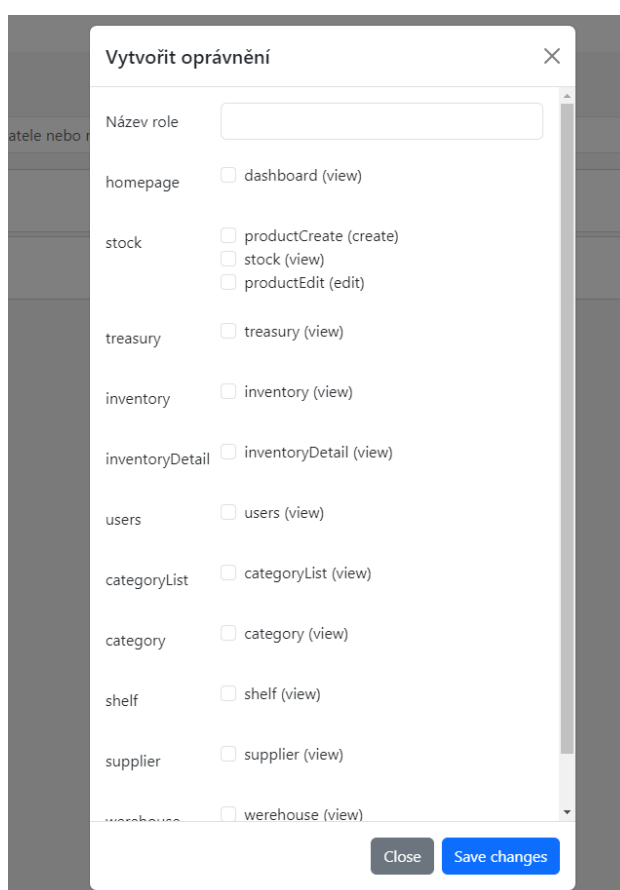
Pro kontrolu oprávnění k určitým akcím jsem vytvořil třídu pro rozšíření stávajícího bezpečnostního systému ve frameworku Nette. Tato třída používá systém Nette ACL (Access Control List), který je konstruován na základě údajů přihlášeného uživatele. Framework Nette už obsahuje metody pro kontrolu rolí, ale jelikož systém vyžaduje univerzální přístupová oprávnění, bylo nutné přidat podporu pro zdroje (resource) a oprávnění (permission). Pro ověření oprávnění lze použít metodu:

```
$user->isAllowed('users', 'view');
```

V tomto příkladu je 'users' zdroj definovaný v konfiguračním souboru a 'view' je specifické oprávnění k tomuto zdroji. Přístup je povolen pouze uživateli, který má přiřazené příslušné zdroje a oprávnění. Zdroj a oprávnění lze také kombinovat s třetím parametrem představujícím roli, ale tato funkcionality není v tomto systému využívána.

Každá stránka má předdefinovaná oprávnění pro různé funkce jako jsou zobrazení, uložení, či mazání. Při tvorbě nové role stačí v zaškrťovacím seznamu vybrat požadovaná oprávnění. Poté lze tuto novou roli přiřadit příslušným uživatelům. Tento systém umožňuje přidělit každému uživateli pouze jednu roli, která odpovídá jeho konkrétním potřebám, ale lze ji snadno upravit nebo vytvořit novou podle požadavků. Tento model také umožňuje přístup ke specifickým částem systému například brigádníkům, kteří mají přiřazené role podle svých pracovních úkolů. Vytváření nových rolí a uživatelů je však výhradně v kompetenci administrátorů.

Na obrázku níže je znázorněn formulář pro vytvoření nové role spolu s ukázkou možných oprávnění, která lze přidělit novým uživatelům.



Vytvořit oprávnění

Název role

homepage dashboard (view)

stock productCreate (create)
 stock (view)
 productEdit (edit)

treasury treasury (view)

inventory inventory (view)

inventoryDetail inventoryDetail (view)

users users (view)

categoryList categoryList (view)

category category (view)

shelf shelf (view)

supplier supplier (view)

warehouse warehouse (view)

Close Save changes

Obrázek 17 Vytváření role

Každému přihlášenému uživateli, respektive jeho identitě, je přidělena doba, po kterou je jeho identita validní a uživatel je přihlášen. Tato možnost jde upravovat prostřednictvím konfiguračního souboru na serveru. Názvy zdrojů aktuálně odpovídají názvům stránek pro lepší přehlednost. Do budoucna je možné, že se data doplní o jasnější překlad, nicméně pro aktuální účely je tento přístup dostačující. Při zavádění systému byly deklarovány základní zdroje a oprávnění, které se budou během testování rozšiřovat podle potřeby. Chtěl jsem tak předejít

velké granularitě oprávnění, které by zpomalovalo načítání systému a zároveň dělalo vytváření nových rolí složitějším. Přestože vytváření nových rolí není nejčastější prováděnou aktivitou, bylo mi z prostředí firmy potvrzeno, že k větší granularitě zatím není třeba přistupovat. Následující tabulka obsahuje základní nastavení, které lze přiřadit jednotlivým rolím.

Tabulka 3 Tabulka oprávnění nad stránkami

Zdroje	Oprávnění
Dashboard	View
Stock	View
Treasury	View, Edit, Create
Inventory	View, Create, Edit, Continue
Users	View, Create, Edit
CategoryList	View, Create, Edit
Category	View, Create, Edit
Shelf	View, Create, Edit
Supplier	View, Create, Edit
Warehouse	View, Create, Edit
ProductEdit	Edit

V případě potřeby lze jednotlivá tlačítka a metody opatřit ochranou formě oprávnění. Záleží jaké budou v budoucnu potřeby firmy v oddělování oprávnění uživatelů.

8 TESTOVÁNÍ

V oblasti vývoje webových aplikací je testování nezbytnou součástí zajištění kvality, bezpečnosti a uživatelské spokojenosti. V kontextu Nette Frameworku, který klade důraz na produktivitu a bezpečnost, je integrace robustního testovacího procesu klíčová. V tomto textu se zaměřím na využití Nette Testeru a Selenium pro testování webových aplikací, přičemž každý z těchto nástrojů slouží specifickým účelům a společně tvoří komplexní testovací strategii.

8.1.1 Nette Tester

Nette Tester je nástroj pro automatizované testování v PHP, který je navržen tak, aby byl jednoduše použitelný, a přitom nabízel dostatečnou funkcionalitu pro pokrytí širokého spektra testovacích scénářů. Jeho integrace do ekosystému Nette Frameworku umožňuje vývojářům efektivně testovat jednotlivé komponenty aplikace, modely, presentery a další logiku aplikace bez nutnosti spuštění v prohlížeči nebo závislosti na externím prostředí. [22]

8.1.2 Selenium

Zatímco Nette Tester skvěle slouží pro testování backendové logiky, Selenium přichází na scénu jako nástroj pro automatizaci webových prohlížečů, což umožňuje simulovat interakce uživatelů s aplikací ve skutečném prohlížečovém prostředí. Tato možnost je neocenitelná pro testování uživatelského rozhraní. Selenium umožňuje testovat, jak aplikace vypadá a chová se ve webových prohlížečích, a to včetně testování responzivity, formulářů a navigace. Provedení scénářů od začátku do konce, jako je například proces registrace uživatele nebo objednávky v e-shopu, poskytuje ucelený pohled na funkčnost a uživatelskou zkušenost.

Testování webových aplikací ve frameworku Nette s využitím Nette Testeru a Selenium představuje komplexní přístup k zajištění kvality softwaru. Tento přístup umožňuje rychle identifikovat a opravit chyby, zvyšuje spolehlivost a uživatelskou spokojenost. Integrace těchto nástrojů do vývojového procesu a jejich kombinace s principy Continuous Integration a Continuous Deployment vede k efektivnějšímu vývoji a udržitelnějším webovým projektům.

Úspěch v testování a celkové kvalitě aplikace závisí na pečlivém plánování, konzistentní implementaci testů a neustálé revizi testovacích strategií, aby byly v souladu s vývojem aplikace a technologií. Využití Nette Testeru a Selenium poskytuje vývojářům silné nástroje, které pomáhají dosáhnout těchto cílů, a umožňuje budování bezpečnějších, výkonnějších a uživatelsky přívětivějších webových aplikací. [23]

8.1.3 PHPStan

V dynamickém prostředí vývoje webových aplikací, kde se prioritizuje efektivita, bezpečnost a rychlost doručení, představuje statická analýza kódu fundamentální nástroj pro udržení a zlepšení kvality softwarových řešení. PHPStan je pokročilý nástroj pro statickou analýzu PHP kódu, navržený k detekci chyb a nekonzistencí ve zdrojovém kódu aplikace bez jejího spuštění. Jeho implementace do vývojového procesu umožňuje včasnou identifikaci a řešení širokého spektra potenciálních problémů, jako jsou typové nesrovnalosti, nedefinované proměnné, chybějící návratové typy funkcí a metody, které interagují s neexistujícími třídami či objekty. [24]

8.1.4 Mockery

Mockery je pružný a jednoduchý framework pro vytváření mock objektů v PHP, který je navržen tak, aby podporoval jednotkové testování v rámci různých frameworků, jako jsou PHPUnit a PHPSpec. Hlavním cílem Mockery je poskytnout framework pro testovací dvojice, který využívá stručné API umožňující detailně specifikovat všechny operace a interakce objektu v lidsky čitelném doménově specifickém jazyce. Mockery funguje jako bezproblémová alternativa ke knihovně phpunit-mock-objects od PHPUnit, která se snadno integruje právě s PHPUnit a je kompatibilní s phpunit-mock-objects bez jakýchkoliv problémů.

V oblasti jednotkových testů mock objekty simulují funkčnost skutečných objektů. Jsou klíčové pro zajištění izolace testů, nahrazují objekty, které ještě neexistují, nebo pomáhají při návrhu API tříd před skutečnou implementací. Ve svém projektu využívám Mockery pro psaní některých jednodušších jednotkových testů. [25]

8.1.5 Testy databáze

Integrační testy spojují více modulů a pokrývají rozsáhlejší scénáře aplikace, a proto nelze použít Mockery, jak je to možné u některých jednotkových testů. Ve frameworku Nette se závislosti ukládají do kontejneru, což jsem využil při psaní integračních testů. Vytvořil jsem speciální konfiguraci pro testovací databázi, která obsahuje nastavení přihlašovacích údajů a další potřebné parametry. Tuto konfiguraci a závislosti předávám pomocí kontejnerů přímo do testovacích tříd. Díky tomu je možné snadno získat přístup ke všem modelovým třídám a databázovému připojení. Vzhledem k tomu, že je důležité testy přiblížit co nejvíce k realitě, je pro testování přesnější používat stejnou databázi, jaká bude v produkčním prostředí. Před spuštěním testů je nutné do této databáze nahrát databázové schéma pomocí Doctrine, a to pouze v případě, že dojde ke změnám v tabulkách. Následně stačí spustit Nette tester, který

zahájí jak jednotkové, tak integrační testy. V rámci integračních testů například zkoumám přidávání produktů do košíku nebo jejich odebrání. Veškerá testovaná data se do testovací databáze nahrají automaticky. Eliminuje se tak potřeba manuálně spouštět SQL skripty.

8.2 Nasazení systému:

Pro úspěšné nasazení systému je nejprve vyžadováno klonování zdrojového kódu projektu z verzovacího systému GitHub. Tento krok zajišťuje, že všechny potřebné soubory a zdrojové kódy jsou staženy do lokálního vývojového prostředí. Po dokončení klonování je klíčové provést konfiguraci systému prostřednictvím lokálního konfiguračního souboru. V tomto souboru se nastavují důležité parametry včetně údajů pro připojení k databázi. Z důvodu ochrany citlivých informací se tento soubor nikdy nesdílí na veřejných repozitářích ani prostřednictvím jiných online kanálů.

Následuje instalace závislostí pomocí nástroje Composer. Composer prověří a nainstaluje všechny požadované knihovny a moduly podle specifikace v souboru `composer.json`. Zvláštní pozornost je potřeba věnovat rozšířením, která nejsou ve výchozím nastavení povolena, jako je například grafické rozšíření 'Imagick'. Tyto závislosti musí být explicitně povoleny v konfiguračním souboru webového serveru, aby bylo možné zprovoznit všechny funkce aplikace. V posledních verzích frameworku je nezbytné v konfiguračním souboru nastavit místo relativní cesty do složky `temp` cestu absolutní. Bez této úpravy se aplikace nepodaří spustit a bude tuto úpravu vyžadovat.

Po dokončení konfigurace je důležité vytvořit schéma databáze a provést databázové migrace. Tyto operace lze spustit pomocí příkazů, které zde vyžadují specifikaci. Také je nutné provést SQL skript pro vytvoření databázových triggerů a inicializaci uživatelů s předdefinovanými rolemi, aby bylo možné přihlášení do systému a plné využití jeho funkcionalit.

Během vývoje bylo potřeba systém pravidelně testovat a optimalizovat, což vedlo k průběžnému nasazování verzí na produkční server. Jelikož jsem nasazování a ladění prováděl přímo ve společnosti, nebyl vyžadován instalační skript. Po zprovoznění aplikace na webovém serveru jsem následně provedl import dat produktů z excelových souborů, což aplikaci uvedlo do plně funkčního stavu. Další vývoj aplikace již přesahuje rámec této diplomové práce.

Je třeba zdůraznit, že pro optimální využití aplikace je doporučeno používat protokol HTTPS, zejména v případech, kdy je potřeba zabezpečit přenos dat, jako je skenování čárových kódů nebo stahování souborů. V případě, že není možné použít HTTPS, je možné v prohlížeči nastavit výjimku pro IP adresu nebo doménové jméno serveru, což umožní použití kamery

i přes HTTP. Tento postup je však považován za nouzové řešení a obecně se nedoporučuje. Po dokončení všech popsanych kroků je webová aplikace plně připravena k použití.

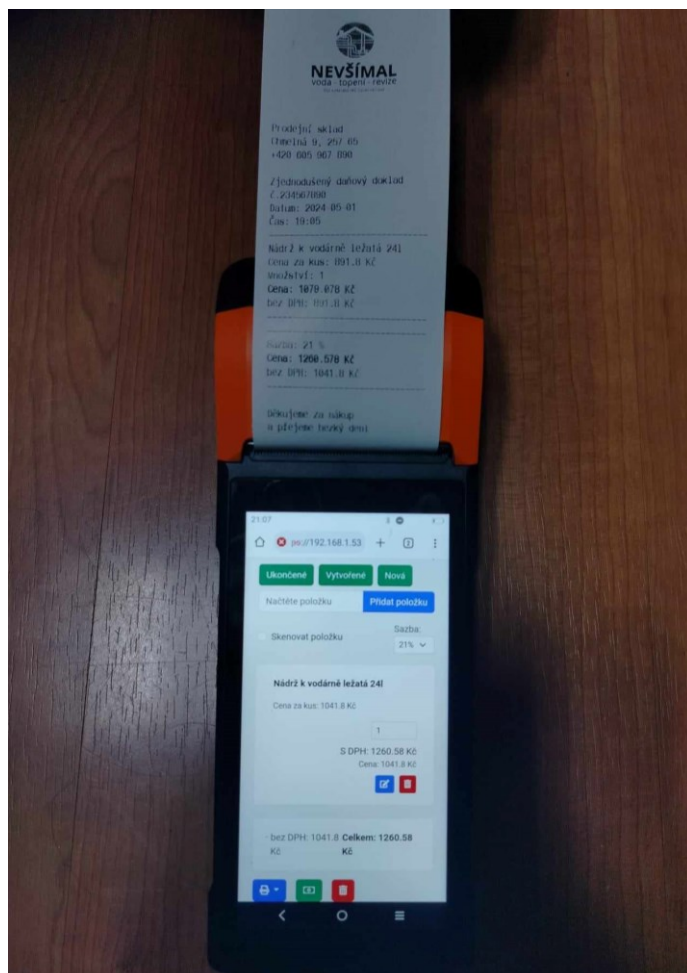
8.2.1 Nasazení na vybavení firmy

V rámci implementace v podnikovém prostředí bylo nezbytné navrhnout systém tak, aby byl schopen fungovat na menších a přenosných zařízeních, jako jsou mobilní telefony a terminály. Během procesu vývoje této aplikace byla rozšířena škála zařízení v majetkovém portfoliu společnosti, především o čtečky čárových kódů a terminály s integrovanými tiskárnami. Konkrétně na těchto zařízeních byla aplikace testována a bude využívána jako hlavní prodejní nástroj. Jedná se o mobilní zařízení s operačním systémem Android. Hlavním rozdílem je robustní konstrukce, vylepšená kamera pro skenování čárových kódů a integrovaná tiskárna pro tisk účtenek.

Původně byl navrhován modul pro přímé ovládání integrované tiskárny, avšak kvůli omezené dostupnosti dokumentace k danému zařízení by bylo nutné vyvíjet další aplikaci nebo skript v jazyce Java. Pro možnost tisku účtenek byla nakonec využita externí aplikace, která při pokusu o otevření PDF souboru automaticky přeměruje výstup na interní tiskárnu a provádí případné úpravy fontů a grafiky, aby bylo možné tisknout bez problémů. Pro tisk účtenek stačí pouze zvolit možnost náhledu, a aplikace se postará o odeslání na tiskárnu.

Pro formátování obsahu účtenky bylo nutné přizpůsobit PDF soubor rozměrům tiskové plochy. K tomuto účelu byla využita knihovna pro PHP, která umožňuje převést HTML kód na PDF soubor se zachováním definovaných stylů. Obsah účtenky byl konzultován s vedením firmy a obsahuje informace o zakoupených produktech, jejich ceně, ceně bez DPH, množství, čísla

účtenky a údajích o firmě. Tato účtenka slouží jako zjednodušený daňový doklad pro zákazníky a pro případné reklamace.



Obrázek 18 Ukázka implementace na vybavení firmy

9 BUDOUCNOST

V této kapitole se zaměřím na potenciální vylepšení a rozšíření systému, která by v budoucnu mohla významně zlepšit jeho funkcionalitu a efektivitu. Tyto možnosti nebyly zahrnuty v současné verzi systému, protože nebyly součástí původních požadavků, které byly definovány po konzultacích s firmou. Některá z navrhovaných rozšíření jsou technicky složitější a vyžadují náročnější implementaci a větší časovou dotaci. Pokud firma projeví zájem o rozvoj těchto funkcí, může dojít k jejich postupnému začleňování do systému. Některá ze zmíněných rozšíření už jsou aktuálně zpracovávána a budou zahrnuta v další verzi.

9.1 Automatizace faktur

V rámci rozvoje nových modulů se aktuálně pracuje na automatizaci načítání faktur. Cílem tohoto modulu je importovat dodací listy nebo faktury do systému a automaticky extrahovat veškeré informace o produktech uvedených v objednávce. Informace o produktech umožní automatické připsání zásob do skladu. V případě, že se jedná o dosud nezalistované zboží, vytvoří se v systému nově.

Tento proces nebude plně autonomní; bude vyžadovat kontrolu uživatele. Hlavní výzvou zůstává extrakce informací o produktech z faktur, které se mohou výrazně lišit svým formátem a stylováním, což komplikuje univerzální zpracování dat. Po načtení těchto informací bude uživateli zobrazen seznam položek s možnostmi dalšího postupu, jako je přidání zboží do skladu nebo vytvoření nového produktu. Uživatel bude mít možnost ovlivnit, jaké akce se s produkty provedou.

Další komplikací je různorodost názvů produktů od různých dodavatelů. Stejně zboží od různých dodavatelů může mít odlišné názvy, což ztěžuje jednoznačnou identifikaci. Z tohoto důvodu je nezbytné implementovat systém procentuální shody názvů. Pro produkty, u kterých není zřejmé, ke které skladové položce je přiřadit, se zobrazí seznam možností seřazený podle procentuální shody. Uživatel si pak může vybrat nejvhodnější odpovídající položku.

Tato funkcionalita by měla v budoucnu výrazně zrychlit proces naskladňování produktů. V současné rozpracované verzi je již implementováno získávání části dat z nahraného PDF souboru faktury.

9.2 Obrázky

Do současnosti nebyl využíván žádný systém pro skladové zásoby podporující obrázky, pouze excelové tabulky, a proto chyběla vizuální data, která by bylo možné přiřadit k jednotlivým

produktům. Přidání obrázků k produktům by mohlo významně usnadnit identifikaci při inventurách. Pro některé pracovníky, včetně brigádníků, nemusí být odborné termíny a názvy součástí zřejmé, a proto možnost nahlédnout na příkladový obrázek produktu představuje značný přínos a úsporu času vedoucího pracovníka.

V reakci na tuto potřebu jsem se rozhodl v budoucnu rozšířit možnosti systému o manuálně nahrávané soubory a fotografie. Tím bude umožněno u každého produktu vést galerii fotografií, online manuály a produktové dokumentace. V počáteční fázi bude implementováno Google API, které umožní získat první tři obrázky produktu podle jeho názvu jako vzorové. Tato funkce bude dostupná při vyhledávání produktů. V pozdějších fázích bude možné tyto obrázky přímo ukládat k jednotlivým produktům, což eliminuje potřebu opakovaného dotazování na Google API.

9.3 Filtry

V rámci skladové tabulky jsou implementovány filtry, které je možné v současnosti rozsáhle nastavovat. Zatím však systém nepodporuje ukládání přednastavených filtrů pro budoucí použití. V následujících verzích je již v plánu tuto možnost zahrnout, což umožní ukládat specifické filtry a zobrazit je v rychlé nabídce pro snadnější přístup.

Očekává se také rozšíření současných filtrů o nové možnosti, které budou odpovídat zvýšeným požadavkům uživatelů. Tyto inovace nebudou omezeny pouze na skladovou tabulku, ale budou se týkat i dalších relevantních dat, jako jsou inventury, jejich výsledky a číselníky. Rozšíření filtrů v těchto oblastech zvýší flexibilitu a efektivitu práce s daty.

9.4 Propojení s více systémy

Během realizace této diplomové práce došlo ke změně obsazení pozice účetní. Nově nastoupivší zaměstnankyně začala využívat účetní systém Pohoda. V případě, že by se tento systém osvědčil a vzbudil zájem o jeho další používání, byla by možná jeho integrace se systémem vyvinutým v rámci této práce.

9.5 QR platba

Jednou z funkcí pokladního systému je možnost exportu faktur ve formátu PDF. Do budoucna by mohl být do tohoto souboru integrován QR kód pro přednastavené fakturační platby. Toto jednoduše implementovatelné rozšíření by znamenalo značné zjednodušení platebních procesů pro nakupující uživatele, což by mohlo zvýšit jejich spokojenost a efektivitu při platbách za fakturované služby.

9.6 Napojení na e-shop

Firma aktuálně posiluje své aktivity v oblasti prodeje na pobočce a v dlouhodobějším horizontu zvažuje expanzi prostřednictvím zřízení e-shopu. Systém je již na tuto změnu připraven a vše závisí na bezprostřední implementaci e-shopu a případném přizpůsobení pro přenos dat mezi interním systémem a nově vytvořeným e-shopem.

9.7 Kamerové systémy

V rámci této práce se zmiňuji o objektu firmy, který je vybaven kamerovým systémem. Při jeho implementaci jsem měl možnost asistovat. Zvolený systém umožňuje integraci externích aplikací a podporuje odesílání notifikací nebo správu systému z externích zdrojů. Jako jedno z posledních vylepšení bylo rozhodnuto o implementaci kamerových logů, což umožní uživatelům retrospektivně sledovat záznamy a upozorní je na detekovaný pohyb. Ačkoliv je stávající systém pro správu adekvátní, tento nový modul by přinesl možnost částečné integrace obou systémů do jednoho ucelenějšího řešení pro základní použití.

ZÁVĚR

Cílem diplomové práce bylo vytvořit a nasadit webovou aplikaci na správu firemního majetku pro konkrétní podnik, což se úspěšně podařilo. Aplikace splnila veškeré stanovené požadavky na realizaci a byla obohacena o další funkce nad jejich rámec. Ty reflektují zpětně zjištěné potřeby, které vyplynuly z rozhovorů se zaměstnanci, které proběhly v různých fázích vývoje systému. Tato aplikace si kladla za cíl zlepšit administrátorskou, ale také běžnou pracovní rutinu v menší firmě, a šetřit tak zaměstnancům cenný čas. Veškeré řízení a procesy byly sjednoceny do jednoho aplikačního rozhraní. Věřím, že až se uživatelé naučí využívat plný potenciál systému, bude jejich práce mnohonásobně rychlejší, přehlednější a efektivnější, minimálně v rozsahu, který systém pokrývá.

Po technické stránce byl software napsán tak, aby byl připraven na další rozšíření a úpravy. Z tohoto důvodu vidím potenciál v nabídce systému dalším podobným firmám, které by si tak mohly dovolit modernizaci s relativně nízkými provozními náklady. Aplikace přináší uživatelům hodnotu zejména v oblasti správy skladových zásob a inventur, kde odproští uživatele od klasické metody tužky a papíru a zpříjemní jim pracovní proces inventarizace.

Aplikace je v současné době úspěšně nasazena v lokální síti firmy, kde nadále probíhá její další vývoj a uživatelské testování. Nadále se zanášejí další moduly, které jsem nastínil v jedné z posledních kapitol a stává se tak plnohodnotnou součástí zmíněné firmy. Pracovníci firmy, kteří se systémem již pracovali, se k němu vyjádřili velmi kladně.

POUŽITÁ LITERATURA

- [1] JUBEL, Eric. Top 10 ERP Systems of 2024. LinkedIn [online]. 2024, 30.06.2023 [cit. 2024-05-15]. Dostupné z: <https://www.linkedin.com/pulse/top-10-erp-systems-2024-eric-kimberling/>.
- [2] What is dynamics 365. MICROSOFT. Microsoft [online]. c2024 [cit. 2024-03-29]. Dostupné z: <https://dynamics.microsoft.com/cs-cz/what-is-dynamics365/>.
- [3] Sap [online]. b. r. [cit. 2024-03-29]. Dostupné z: <https://www.sap.com/cz/index.html>
- [4] Oracle Enterprise Resource Planning (ERP). ORACLE. Oracle [online]. c2024 [cit. 2024-03-29]. Dostupné z: <https://www.oracle.com/erp/>.
- [5] Acumatica [online]. c2008 — 2024 [cit. 2024-03-29]. Dostupné z: <https://www.acumatica.com/>.
- [6] Odoo [online]. b. r. [cit. 2024-03-29]. Dostupné z: https://www.odoo.com/cs_CZ
- [7] ABRA SOFTWARE A.S. Abra [online]. c2024 [cit. 2024-05-15]. Dostupné z: <https://www.abra.eu/>.
- [8] <https://zapier.com/>. EMLEY, Bryce. Zapier [online]. 2023, 18.10.2023 [cit. 2024-03-29]. Dostupné z: <https://zapier.com/blog/best-erp-software/>.
- [9] Nejlepší software pro správu zásob (2023). CARTEROVÁ, Rebeka. Ecommerce platforms [online]. 2023, 1.10.2023 [cit. 2024-03-29]. Dostupné z: <https://ecommerce-platforms.com/cs/articles/best-inventory-management-software>.
- [10] ABRA Flexi: Ekonomický software pro moderní byznys. ABRA SOFTWARE A.S. Abra [online]. c2024 [cit. 2024-03-29]. Dostupné z: <https://www.abra.eu/flexi/>.
- [11] Pricing plan. Odoo [online]. b. r. [cit. 2024-03-29]. Dostupné z: https://www.odoo.com/cs_CZ/pricing-plan.
- [12] ARLOW, Jim a NEUSTADT, Ila. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9
- [13] PHP [online]. b. r. [cit. 2024-05-15]. Dostupné z: <https://www.php.net>
- [14] GRUDL, David. Nette [online]. c2024 [cit. 2024-04-02]. Dostupné z: <https://nette.org/cs/>
- [15] Naja [online]. 2024 [cit. 2024-04-02]. Dostupné z: <https://naja.js.org/>
- [16] GRUDL, David. Latte [online]. c2008-2024 [cit. 2024-04-02]. Dostupné z: <https://latte.nette.org/cs/>
- [17] Doctrine [online]. b. r. [cit. 2024-05-15]. Dostupné z: <https://www.doctrine-project.org>
- [18] STRAFY, Rick. Ako použiť Doctrine ORM v Nette Frameworku. Nette [online]. 2020 [cit. 2024-04-02]. Dostupné z: <https://blog.nette.org/cs/ako-pouzit-doctrine-orm-v-nette-frameworku>

- [19] HARTINGER, David. MVC [online]. c2024 [cit. 2024-04-02]. Dostupné z: <https://www.itnetwork.cz/php/mvc/objektovy-mvc-redakcni-system-v-php-popis-architektury>
- [20] JQuery [online]. c2024 [cit. 2024-04-02]. Dostupné z: <https://jquery.com/>
- [21] About MariaDB Server. MariaDB [online]. c2009-2024 [cit. 2024-05-04]. Dostupné z: <https://mariadb.org/about/>.
- [22] GRUDL, David. Tester [online]. c2008-2024 [cit. 2024-05-15]. Dostupné z: <https://tester.nette.org/cs/>.
- [23] Selenium [online]. c2024 [cit. 2024-05-15]. Dostupné z: <https://www.selenium.dev/>.
- [24] MIRTES, Ondřej. PHPStan [online]. c2016–2024 [cit. 2024-05-15]. Dostupné z: <https://phpstan.org/>.
- [25] PÁDRAIC, Brady, Dave MARSHALL a contributors. Mockery [online]. c2024 [cit. 2024-05-15]. Dostupné z: <https://docs.mockery.io/en/latest/>.

PŘÍLOHY

Příloha A – Zdrojové kódy	65
---------------------------------	----

PŘÍLOHA A – ZDROJOVÉ KÓDY

K práci jsou přiloženy zdrojové kódy aplikace.