

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



## **Bakalářská práce**

**Tvorba informačního systému pro vyhledávání zboží a  
vychystávání objednávek v online hypermarketovém  
skladu**

**Daria Milinchukova**

© 2023 ČZU v Praze

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Daria Milinchukova

Informatika

Název práce

**Tvorba informačního systému pro vyhledávání zboží a vychystávání objednávek v online hypermarketovém skladu**

Název anglicky

**Design of Information System for managing online hypermarket warehouse**

---

### Cíle práce

Hlavním cílem bakalářské práce je návrh a implementace informačního systému určeného ke zvýšení efektivity procesů vyhledávání zboží a vychystávání objednávek v online hypermarketovém skladu.

Dílí cíle bakalářské práce jsou:

- Analýza workflow ve skladu online obchodu,
- Analýza informačních potřeb koncových uživatelů,
- Vývoj modelu informačního systému,
- Vytvoření jediné centralizované databáze pro rychlý přístup k aktuálním informacím o umístění zboží ve skladu, kvalitě práce zaměstnanců.

### Metodika

Metodika bakalářské práce bude založena na základních principech návrhu aplikací (analýza, návrh, implementace, testování a provoz), studiu odborných zdrojů a vyhodnocení získaných poznatků.

V teoretické části autorka popíše teoretické základy informačních systémů. Budou vysvětleny základní pojmy a principy podnikových informačních systémů na základě vybrané literatury.

Popsána bude metodika vývoje a nástroje k dosažení hlavního cíle.

Bude provedena analýza potřeb firmy a sběr dat. Na základě této analýzy bude proveden návrh a implementace informačního systému určeného ke zefektivnění průběhu vychystávání objednávek.

Praktická část bude obsahovat analýzu a hlavně návrh systému samotného. Pro návrh modelu informačního systému budou použity metody softwarového inženýrství a jazyky HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JavaScript, SQL (Structured Query Language) a relační databáze.

Na základě shrnutí teoretických poznatků a výsledků vlastní části práce budou formulovány závěry bakalářské práce.



### **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "Tvorba informačního systému pro vyhledávání zboží a vychystávání objednávek v online hypermarketovém skladu" jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2023

---

### **Poděkování**

Ráda bych touto cestou poděkovala mému vedoucímu práce panu doc. Ing. Jan Tyrychtr, Ph.D. za odborné rady a konzultace během vedení této bakalářské práce.

# Tvorba informačního systému pro vyhledávání zboží a vychystávání objednávek v online hypermarketovém skladu

## Abstrakt

Tato bakalářská práce se zabývá návrhem informačního systému zajišťujícího zvýšení efektivity procesů vyhledávání zboží a vychystávání objednávek v online hypermarketovém skladu prostřednictvím dobře zavedené posloupnosti provozů a rozvinuté databáze vychystávacího procesu, které budou řízeny vyvinutým systémem.

V teoretické části této bakalářské práce jsou vysvětlovány a definovány základní pojmy týkající se dané problematiky. Dále se bude zabývat vysvětlením informačního systému a jeho částí. V poslední části teoretické práce budou popsány programovací jazyky, které budou použity pro vývoj webové stránky.

Praktická část této bakalářské práce se skládá ze dvou částí. První část obsahuje analýzu vychystávacího procesu, sběr a analýzu požadavků na vývoj informačního systému. Výsledkem je návrh informačního systému a databáze. Druhá část praktické práce spočívá ve vytvoření informačního systému pomocí HTML, kaskádových stylů jazyka a programovacího jazyka PHP.

Výsledný systém se skládá z několika částí. První je samotná webová aplikace, ve které bylo vyvinuto uživatelsky přívětivé rozhraní pro pracovníky skladu. Druhou částí systému je databáze, kde jsou uloženy všechny potřebné informace pro jeho provoz a jejímž prostřednictvím je řízeno a sledováno správné provádění pracovního procesu.

Na základě shrnutí teoretických poznatků a výsledků vlastní části práce budou formulovány závěry.

**Klíčová slova:** sklad, vychystávací proces, web, systém, informační systém, HTML, CSS, JavaScript, PHP, MYSQL, databáze, relační databáze, analýza, návrh, implementace.

# **Design of Information System for managing online hypermarket warehouse**

## **Abstract**

This bachelor thesis deals with the design of an information system ensuring the increase of efficiency of the processes of searching goods and picking orders in an online hypermarket warehouse through a well-established sequence of operations and a developed database of the picking process, which will be managed by the developed system.

In the theoretical part of this bachelor thesis, the basic concepts related to the subject are explained and defined. It will also deal with the explanation of the information system and its parts. In the last part of the theoretical work, the programming languages that will be used for the development of the website will be described.

The practical part of this bachelor thesis consists of two parts. The first part contains the analysis of the picking process, collection and analysis of the requirements for the development of the information system. The result is the design of the information system and database. The second part of the practical work consists of the development of the information system using HTML, cascading style sheets and PHP programming language.

The resulting system consists of several parts. The first is the web application itself, in which a user-friendly interface for warehouse workers has been developed. The second part of the system is the database where all the necessary information for its operation is stored and through which the correct execution of the work process is controlled and monitored.

Based on the summary of the theoretical knowledge and the results of the actual part of the work, conclusions will be formulated.

**Keywords:** warehouse, picking process, web, system, information system, HTML, CSS, JavaScript, PHP, MYSQL, database, relational database, analysis, design, implementation.

# Obsah

<b>1 Úvod .....</b>	<b>11</b>
<b>2 Cíl práce a metodika .....</b>	<b>12</b>
2.1 Cíl práce.....	12
2.1.1 Dílčí cíle bakalářské práce.....	12
2.2 Metodika.....	12
<b>3 Teoretická východiska .....</b>	<b>14</b>
3.1 Sklad.....	14
3.1.1 Funkce skladu .....	14
3.1.1.1 Přednostní funkce skladu.....	14
3.1.1.2 Negativní funkce skladu .....	15
3.1.2 Skladovací procesy.....	15
3.1.2.1 Příjem .....	15
3.1.2.2 Zaskladnění.....	15
3.1.2.3 Vyskladnění zboží.....	16
3.1.2.4 Packing .....	16
3.1.3 Metody vychystávání ve skladu .....	16
3.1.3.1 Vychystávání po 1. objednávce .....	16
3.1.3.2 Vychystávání po objednávkových dávkách .....	17
3.1.3.3 Zónové vychystávání .....	17
3.1.4 Značení skladových pozic .....	18
3.2 Data a informace .....	18
3.3 Informační systém.....	19
3.3.1 Druhy informačních systémů.....	19
3.3.2 Architektura informačního systému .....	19
3.3.2.1 Architektury podle metodiky MMDIS .....	20
3.3.3 Vývoj informačního systému.....	20
3.4 Web, webové stránky .....	21
3.5 Technologie .....	22
3.5.1 HTML.....	22
3.5.2 CSS.....	23
3.5.3 JavaScript.....	24
3.5.4 PHP .....	24
3.5.5 SQL .....	25
3.5.6 My SQL .....	25
3.5.7 Textový editor.....	25



<b>4 Vlastní práce .....</b>	<b>26</b>
4.1 Analýza procesu vychystávání objednávek pomocí workflow diagram .....	26
4.2 Požadavky na informační systém.....	27
4.2.1 Obdržené požadavky .....	27
4.3 Návrh IS.....	29
4.3.1 Návrh architektury .....	29
4.3.2 Návrh databáze. Entity relationship diagram.....	29
4.3.2.1 Tabulka „storehouse“ .....	29
4.3.2.2 Tabulka „goods“ .....	29
4.3.2.3 Tabulka „cart“ a „mobile“ .....	30
4.3.2.4 Tabulka „task“ .....	30
4.3.2.5 Tabulka „worker“.....	30
4.3.2.6 Tabulka „effectiveness“ .....	31
4.3.3 Grafický návrh uživatelského rozhraní .....	31
4.3.3.1 Přihlášení uživatele .....	32
4.3.3.2 Výběr sektoru.....	33
4.3.3.3 Kontrola výběru sektoru.....	33
4.3.3.4 Zadání čísla košíku.....	34
4.3.3.5 Vychystávání objednávek.....	35
4.3.3.6 Poslední stránka systému.....	36
4.4 Vývoj informačního systému .....	37
4.4.1 Vytvoření struktury databáze.....	37
4.4.2 Tvorba webové aplikace .....	39
4.4.2.1 Připojení do databáze .....	39
4.4.2.2 Nastavení HTML kódu.....	40
4.4.2.3 Implementace přihlášení uživatele.....	40
4.4.2.4 Implementace výběr sektoru a kontrola .....	43
4.4.2.5 Implementace zadání čísla košíku a kontrola .....	46
4.4.2.6 Implementace vychystávání objednávek.....	48
4.4.2.7 Implementace poslední stránky systému .....	52
<b>5 Zhodnocení výsledků .....</b>	<b>53</b>
<b>6 Závěr .....</b>	<b>54</b>
<b>7 Seznam použitých zdrojů .....</b>	<b>55</b>
<b>Seznam obrázků a zkratk.....</b>	<b>57</b>

7.1	Seznam obrázků .....	57
7.2	Seznam použitých zkratek .....	57
<b>Přílohy</b>	.....	<b>59</b>

# 1 Úvod

Firmy mají stále větší tendenci přesouvat své aktivity do online prostoru. Tento trend je zvláště výrazný v období covidovém a postcovidovém. Dnes má každý internetový podnik sklad, kde skladuje prodávané zboží, a zatímco v minulosti byl sklad vnímán jako prostor pro uskladnění přebytečného zboží, dnes je důležitou součástí podniku, na jehož efektivním řízení závisí úspěch firmy.

Podnikové sklady čelí výzvě efektivního řízení zásob a přesného a rychlého plnění objednávek, protože donedávna se práce ve skladu prováděla pomocí tištěných faktur a kompletačních listů. V této souvislosti se práce ve skladu prováděla s pomocí informačních systémů, které dokážou uchovávat velké množství informací a řídit skladové procesy, minimalizovat lidské chyby, zkrátit dodací lhůty a zvýšit přesnost.

Sklad zahrnuje velké množství procesů, z nichž každý vyžaduje univerzální informační systém. V tomto článku se budu zabývat procesem vyhledávání zboží a vychystávání objednávek. Před dvěma lety jsem pracovala ve skladu jako vychystávač a používala jsem podobné informační systémy, což mi umožnilo analyzovat, jaké nevýhody ty stávající mají. Motivací pro vývoj tohoto typu informačního systému je urychlit a zároveň i zefektivnit průběh vychystávání objednávek.

## 2 Cíl práce a metodika

### 2.1 Cíl práce

Hlavním cílem bakalářské práce je návrh a implementace informačního systému určeného ke zvýšení efektivity procesů vyhledávání zboží a vychystávání objednávek v online hypermarketovém skladu. Za pomoci tohoto programu by se měl urychlit a zároveň i zefektivnit průběh vychystávání objednávek.

#### 2.1.1 Dílčí cíle bakalářské práce

- Analýza workflow ve skladu online obchodu,
- Analýza informačních potřeb koncových uživatelů,
- Vývoj modelu informačního systému,
- Vytvoření jediné centralizované databáze pro rychlý přístup k aktuálním informacím o umístění zboží ve skladu, kvalitě práce zaměstnanců.

### 2.2 Metodika

Metodika bakalářské práce je založena na základních principech návrhu aplikací (analýza, návrh, implementace, testování a provoz), studiu odborných zdrojů a vyhodnocení získaných poznatků.

V teoretické části se zabývám teoretickými znalostmi pro informační systém. Vysvětluji základní pojmy a principy informačních systémů na základě vybrané literatury. Analyzuji potřeby firmy a sbírám data. Na základě této analýzy provádím návrh a implementaci informačního systému určeného ke zefektivnění průběhu vychystávání objednávek. Dále vysvětluji metodiku vývoje a nástroje k dosažení hlavního cíle. Studiem se seznámím s programovacími jazyky používanými k řešení. V práci samotné popisují jejich základní charakteristiky a k čemu jsou využívány.

Praktická část obsahuje analýzu vybraného procesu a hlavně návrh systému samotného.

Pro analýzu procesu vyhledávání zboží a vychystávání objednávek využívám diagram pracovních postupů (workflow diagram). Pro návrh modelu informačního systému používám metody softwarového inženýrství. Při vytvoření databáze používám ER diagram (Entity relationship diagram) pro identifikaci klíčových prvků a vyznačení vazeb mezi nimi a programovací jazyk SQL (Structured Query Language). Pro implementaci informačního systému používám jazyky HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JavaScript, PHP (Personal Home Page).

Na základě shrnutí teoretických poznatků a výsledků vlastní části práce budou formulovány závěry bakalářské práce.

## 3 Teoretická východiska

V dané kapitole se práce bude zabývat vysvětlením všech základních a důležitých pojmů nutných pro tvorbu informačního systému na vyhledávání zboží a vychystávání objednávek ve skladu. V první kapitole bude vysvětleno, co je sklad, jaké má funkce a jaké procesy v něm probíhají. V dalších kapitolách se bude věnovat informačnímu systému a využitým technologiím k jeho implementaci. Všechny znalosti teoretických východisek budou uplatněny k tvorbě závěrečného projektu.

### 3.1 Sklad

“Primární – hlavní funkcí skladu je expedovat materiál (zboží) v množství, kvalitě, skladbě, obalech a přepravních prostředcích, v čase (lhůtách, frekvenci) a v pořadí (sekvenci) podle požadavků odběratelů”. [1]

Dříve byla definice skladu omezená na jeho jedinou funkci – skladování přebytečného zboží. To určilo sklad jako neefektivní věc, protože vede k dočasnému přerušení materiálových toků a zbytečným nákladům firmy. Za celou dobu existence se sklady modernizovaly, měnily se principy jeho práce a účel. V dnešní době sklady zajišťují významnou část podnikových činností a plní mnoho funkcí, které pomáhají společnosti zvyšovat úroveň poskytovaných služeb. [1]

#### 3.1.1 Funkce skladu

Sklad plní mnoho funkcí a jako součást obchodního procesu mají tyto funkce pozitivní i negativní aspekty. Níže popíšeme nejdůležitější z nich.

##### 3.1.1.1 Přednostní funkce skladu

Za přednostní funkce skladu považují:

- Sklady poskytují úsporu nákladů. Díky své kapacitní funkci jsou schopné pracovat s velkým množstvím materiálových toků, umožňují firmě plně využívat své zdroje a zajišťovat dodávky velkého sortimentu.
- Sklad jako zprostředkovatel kvalitní interakce s klientem. Vhodně lokalizovaný sklad umožňuje zkrátit termín vyřízení objednávek a poskytuje lepší úroveň služeb.

- Sklad umožňuje skladování přebytků a vytvoření zásob, což pomáhá firmě reagovat s menšími ztrátami na náhodný charakter poptávky a omezené možnosti systému. [1]

#### 3.1.1.2 Negativní funkce skladu

Za negativní funkce skladu považují:

- Sklady přinášejí firmě dodatečné náklady. Jejich většinu tvoří náklady na plánování a jsou vynaloženy na údržbu a provoz skladu (spotřeba energie na osvětlení, náklady na zaměstnance, ztráty způsobené nedodržením podmínek skladování). [1]

#### 3.1.2 Skladovací procesy

Než se výrobek dostane ke spotřebiteli, prochází ve skladu postupně jednotlivými skladovými procesy. Hlavním manažerským cílem při údržbě skladu je vytvořit takový sled činností, který zajistí, že se výrobek dostane včas do rukou spotřebitele ve správném počtu a kvalitě.

„Základním předpokladem efektivního řízení skladu je podrobné zmapování veškerých procesů, které se v něm odehrávají, jejich pojmenování a náležitá implementace v systému“. [2]

##### 3.1.2.1 Příjem

Příjem je prvním a důležitým krokem ve skladovacím procesu, který se skládá z prvotní kontroly, tedy že zboží došlo ve správném počtu, v nepoškozeném balení a má všechny potřebné viněty. Každá chyba, která se zde uskuteční, se promítne v dalších procesech. [3]

##### 3.1.2.2 Zaskladnění

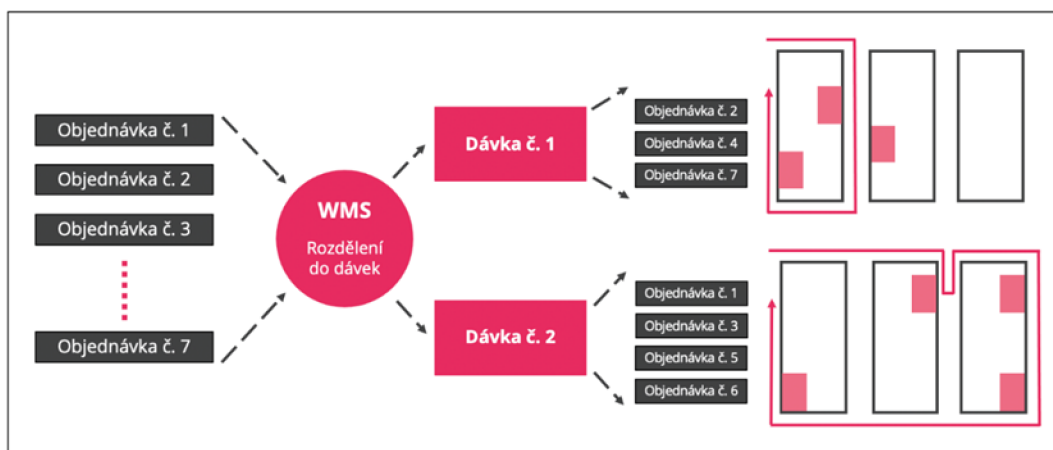
Proces zaskladnění – určení místa pro uskladnění příchozího zboží. Tento proces je zaměřen na členění řízeného skladu na zóny, sekce, uličky, regály, patra a pozice, ve kterých každá zásoba bude adresně umístěna. Zboží je umístěno do zón s možností minimalizovat nároky na prostor a zkrátit dobu hledání. [3]





### 3.1.3.2 Vychystávání po objednávkových dávkách

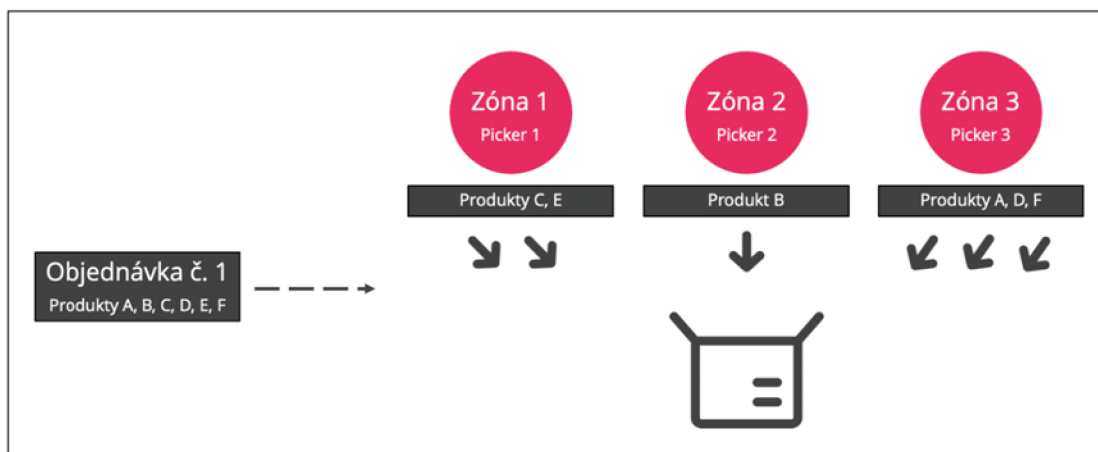
Jedná se o metodu, která je vhodná, pokud zboží jednotlivých objednávek obsahují stejná nebo blízká umístění. Tyto objednávky se slučují do jednoho seznamu a skladník navštíví skladovou oblast pouze jednou. Metoda se používá pro malé objednávky. Cílem této metody je zvýšení produktivity při minimalizaci délky tras. [4]



Obrázek 2: Ilustrace procesu "Vychystávání po objednávkových dávkách" [4]

### 3.1.3.3 Zónové vychystávání

Jedná se o metodu, která je vhodná, pokud jsou součástí objednávky produkty z různých oblastí. Každému pracovníkovi je přiřazena určitá oblast skladu, ze které vychystávají zboží pro všechny objednávky. Tato metoda je příjemná pro velké provozy, které odbavují větší objednávky s různými položkami. [4]



Obrázek 3: : Ilustrace procesu "Zónové vychystávání" [4]

### 3.1.4 Značení skladových pozic

Efektivní práce ve skladu vyžaduje označení jednotlivých skladových míst tak, aby pro skladníka bylo co nejvíce srozumitelné a aby po skladu zbytečně nebloudil. Pro označení jednotlivých zón se nejčastěji využívá podlahové značení nebo závěsné značky. Při využití mobilních terminálů se úsek obvykle označuje čárovým kódem a viditelným kontrolním číslem.

Nejčastěji je pracovník vybaven mobilním terminálem s vestavěným snímačem čárového kódu, který je bezdrátově napojen na systém. Na displeji terminálu dostává skladník instrukce k vykonání požadovaného úkolu, pomocí snímače čárového kódu přečte kód skladové pozice, na níž je zboží umístěno, tím provádí kontrolu správnosti dané pozice.

[5]



Obrázek 4: Čárový kód a kontrolní číslo [5]

## 3.2 Data a informace

Další část bakalářské práce je věnovaná tématu informačního systému, ale nejprve je třeba probrat dva základní pojmy, které s ním úzce souvisí – data a informace.

- **Data** jsou počítačově (elektronicky) zapsaná fakta, zprávy a údaje o všem možném, co se kdy komu podařilo zachytit a následně do nich zaznamenat (zakódovat) vhodným způsobem, a to podle nějakých pravidel (syntaxe). Data vypovídají o světě, tj. obsahují nějaká sdělení (informací) a za předpokladu, že jsou pravdivá a srozumitelná, může z nich mít příjemce nějaký užitek – produkt lidské činnosti.
- **Informace** vzniká v okamžiku účelového užití dat. Informací tedy rozumíme data, jimž jejich uživatel přisuzuje určitý význam, která uspokojují konkrétní objektivní informační potřebu svého příjemce. Nositelem informace jsou číselná data, text, zvuk, obraz atd.

Data jsou „surovinou“ pro tvorbu informací, z čehož vyplývá, že nesprávná (nepravdivá) data nám mohou poskytnout špatnou informaci. [6]

### 3.3 Informační systém

„Informační systém je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení. Jinak řečeno, systém je množina vzájemně propojených komponent, které musí pracovat dohromady pro celý systém tak, aby tento systém naplnil daný účel.“ [6]

Informační systém tedy pomáhá řídit práci a předávat znalosti v univerzální formě. Informační systém je obecně chápán jako úložiště informací, které jsou přijímány a zpracovávány prostřednictvím relační databáze. Jedním z hlavních cílů informačního systému je eliminovat přímou lidskou účast pomocí automatizací podnikových procesů. Automatizace práce podniku vytvořením informačního systému pomáhá nejen strukturovat důležitou informaci, která je uložena v tištěné podobě, ale také zjednodušit k ní přístup. Informační systém tedy pomáhá řídit práci a předávat znalosti v univerzální formě. [7]

#### 3.3.1 Druhy informačních systémů

Informační systémy ve firmě lze rozdělit do dvou hlavních skupin:

- **Externí IS** – jedná se o informační systém, který se zaměřuje na řízení vztahů s externími uživateli (zákazníky).
- **Interní IS** – jedná se o informační systém, který se zaměřuje na zlepšení vnitřních podnikových procesů. Takové informační systémy neposkytují služby, jsou k dispozici omezenému počtu uživatelů a slouží konkrétní organizační funkci. [8]

#### 3.3.2 Architektura informačního systému

Architektura informačního systému je významným faktorem stabilní a úspěšné struktury systému. Pomocí architektury popisujeme jednotlivé komponenty podnikových procesů podle technologických, ekonomických a dalších možností pro zohlednění hlavních požadavků na vlastnosti informačního systému. Architektura umožňuje založit integrovaný systém, kde všechny jeho komponenty budou dohromady spolupracovat. [9]

### 3.3.2.1 Architektury podle metodiky MMDIS

Tato metodika je založena na principu vrstvení, který zahrnuje tři úrovně architektury. Hlavním cílem MMDIS je vytvořit model, kde všechny komponenty budou vzájemně spolupracovat a vytvoří efektivní systém, který splňuje požadované vlastnosti. [9]

#### 3.3.2.1.1 Byznys architektura

Jedná se o architekturu, která nepatří do oboru informatiky. Jejím úkolem je analýza byznysu – pohled na celý podnik (jeho cíle, byznys procesy, strategii, základní pravidla apod.). Závěr této úrovně představuje mapovaný model podniku. Hlavní otázky: Co? Kdo? [9]

#### 3.3.2.1.2 Globální architektura

Druhá úroveň metodiky MMDIS je architektura, jejímž cílem je průměrný počet funkcionalit podnikových procesů: jednotlivých technických služeb podniku (hardware, software, data, lidé) a obchodních podmínek (bezpečnostní pravidla, cena, lokalita). Hlavní otázky objemové: Kolik? Kvalitní: Jak? Kdy? [9]

#### 3.3.2.1.3 Dílčí architektury

Poslední úroveň metodiky MMDIS je architektura, která v sobě zahrnuje doplňující architektury, poskytující příležitost hlubšího prozkoumání budoucího systému:

- **aplikační architektura** – hlavními komponentami jsou softwarové aplikace (jakými aplikacemi je pokryta celková funkcionalita informačního systému a jaké vazby jsou mezi těmito aplikacemi);
- **datová/informační architektura** – hlavními komponentami jsou datové objekty (analýza potřebných typů datových objektů a jejich vazeb);
- **softwarová architektura** – hlavními komponentami jsou programové moduly aplikace;
- **technologická architektura** – hlavními komponentami jsou hardwarové komponenty (servery, počítačové sítě) a komponenty základního programovaného vybavení (operační systémy, databázové systémy). [9]

### 3.3.3 Vývoj informačního systému

Jedním z důležitých kroků na začátku plánování informačního systému je určit, jakým způsobem bude probíhat vývoj informačního systému. Existují dvě základní metody jeho vývoje.

První způsob je **IASW – individuální aplikační software**. Je to způsob, při němž je aplikace vytvořena individuálně pro každý konkrétní podnik, podle jeho vlastních potřeb. Za výhodu považujeme, že funkcionalita tohoto softwaru optimálně podporuje činnosti každého specifického procesu podniku a dosažení specifických cílů na trhu. Takže tento způsob vývoje informačního systému zajišťuje snadnější výhodu nad konkurencí, protože podnik je jediným uživatelem daného softwaru. Je jasné že každý individuální produkt je dražší a jeho vývoj trvá delší dobu. Budu využívat tuto metodu.

Druhý způsob je **TASW - typový aplikační software**. Je to způsob, při němž je aplikace vytvořena a dále rozvíjena specializovaným výrobcem pro požadavky určité třídy podniků. Takový produkt zahrnuje v sobě funkcionalitu nejlepších praktik pro daný obor. Podniky berou pronájem aplikace TASW, přičemž cena licence je nižší než cena na vývoj osobní aplikace pro jednotlivý podnik (IASW). Takže doba instalace aplikace na technologie je kratší, protože podnik nakupuje již hotový produkt. Podniky ale často nepoužívají veškerou funkcionalitu této aplikace, což znamená, že nakupuje to, co nepotřebuje. Naopak od IASW jde o situace, kdy se podnik musí přizpůsobit aplikaci, nikoliv aplikace podniku. Tento způsob je vhodný pro vysoce standardizované procesy (účetnictví, řízení dokumentu). [9]

### 3.4 Web, webové stránky

Každý informační systém je založen na webu a využívá webové technologie. Proto je důležité definovat, co je to web.

Web je velký informační zdroj, na němž je uloženo velké množství webových stránek, což jsou hypertextové dokumenty. Webové stránky vznikly v podobě statického textu – jako elektronické noviny, kde se lidé mohli jenom seznámit s obsahem odborných článků nebo základními informacemi o nějaké firmě. Postupně se internet stával dynamičtější a dostupnějším, což umožnilo používat stránky i pro komunikaci s návštěvníky.

Vývojem webových standardů se zabývá společenství W3C (World Wide Web Consortium), vedené Timem Berners-Leem, jenž vynalezl web a jazyk HTML. Není možné vytvořit webovou stránku bez jazyka HTML. Jeho kód uchovává obsah stránky a popisuje jeho význam. Webové prohlížeče posléze zobrazují tento obsah obalený do značek jazyka HTML uživatelům. [10]

Web se skládá z několika částí. Většinu z těchto složek uživatelé nevidí, jsou totiž určeny spíše pro webové prohlížeče a vyhledávací roboty:

- HTML je značkovací jazyk pro tvorbu webových stránek.
- URL je způsob adresování webových stránek.
- HTTP je protokol, který umožňuje prohlížečům získávat webové stránky.

### 3.5 Technologie

Informační systém nemůže existovat bez informačních technologií, které se stávají hlavním vykonavatelem výstavby a provozu informačního systému. [6]

Při výběru technologie jsem vycházela z toho, co jsem nastudovala na univerzitě. Chtěla bych v této práci uplatnit znalosti získané během studia.

#### 3.5.1 HTML

HTML (HyperText Markup Language) je základem každé webové aplikace nebo webových stránek. Znalosti jazyka HTML tvoří základ pro stavbu efektivních webových stránek.

HTML funguje jako systém vkládání informací o obsahu do textového dokumentu. Tyto informace nazýváme značka a popisujeme s nimi význam obsahu neboli jeho sémantiku. Vývojáři neustále pracují na vývoji HTML. Aktuální verze je HTML5. Jazyk HTML5 klade důraz na sémantiku, přičemž veškerou změnu vzhledu nechává na jazyk CSS. U starších verzí jazyka HTML tomu tak ale vždy nebylo. Sémantika je v jazyce HTML důležitá. Kód s dobrou sémantikou je efektivnější, udržitelnější a lze snadněji měnit jeho vzhled.

Každý dokument HTML by měl obsahovat následující komponenty:

- Definici typu dokumentu – DOCTYPE. Prohlašuje naši stránku za dokument jazyka HTML5.
- Element `<html>` s atributem `lang` `<html lang="kod-jazyka">` zahájí samotný dokument HTML a zvolíme kód výchozího jazyka, v němž budeme psát obsah naší stránky.
- Element `<head>` uvozuje záhlaví dokumentu.
- Specifikaci znakové sady v elementu `meta` `<meta charset="UTF-8">` definuje nejuniverzálnější znakovou sadu UTF-8 pro náš dokument. Samozřejmě můžeme zvolit jinou znakovou sadu.

- Element <title> obsahuje název naší stránky.
- Element <body> uvozuje tělo dokumentu. Za tuto značku patří obsah naší stránky. Značka </body> ukončí tělo dokumentu. [10]

```

1  <!DOCTYPE html>
2  <html lang="cs">
3  <head>
4      <meta charset="UTF-8">
5      <title>Ahoj, světe</title>
6  </head>
7  <body>
8  <p>
9      Příklad textu
10 </p>
11 </body>
12 </html>

```

**Obrázek 5: Vzorek kódu HTML**

V této práci se pro tvorbu kompletního uživatelského rozhraní využívá právě jazyk HTML5 kvůli jeho aktuálnosti a jednoduché implementaci.

### 3.5.2 CSS

CSS (Cascading Style Sheets) je jazykem kaskádových stylů, je vlastně popisným jazykem strukturovaných dokumentů. Jazyky kaskádových stylů umožňují přímo ovlivňovat uspořádání vytvořeného kódu HTML. Jinak řečeno, CSS má za úkol zajistit, jak by se měly jednotlivé elementy jazyka HTML zobrazovat v prohlížeči.

První verze jazyka CSS se objevily až několik let po vzniku jazyka HTML. Vztah mezi jazykem CSS3 a jeho staršími verzemi je analogický ke vztahu jazyka HTML5 s jeho předchůdci – jazyk CSS3 je přirozeným rozšířením svých starších verzí.

CSS a HTML jsou dva nezávislé dokumenty a pro spolupráci musí být vzájemně připojené. To lze provést dvěma způsoby: externím připojením v podobě souboru nebo přímým uvedením v hlavičce dokumentu (vzor příkazu HTML pro připojení souborů CSS <link href="style.css" type="text/css" rel="stylesheet">).

Vznik CSS zefektivnil tvorbu webových stránek, protože umožnil od sebe oddělit obsah a vzhled. Tím se zmenšila velikost souboru HTML a změnil se vzhled samotného zdrojového kódu, což usnadnilo úpravu textů a stylů. [10]

```

1 ▼ html {
2     background: #e6e9e9;
3     background-image: linear-gradient(270deg, rgb(230, 233, 233) 0%, rgb(216, 221, 221) 100%);
4 }
5
6 ▼ body {
7     background: #fff;
8     box-shadow: 0 0 2px rgba(0, 0, 0, 0.06);
9     color: #545454;
10    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
11    font-size: 16px;
12    line-height: 1.5;
13    margin: 0 auto;
14    max-width: 800px;
15    padding: 2em 2em 4em;
16 }

```

Obrázek 6: Vzorek kódu CSS

### 3.5.3 JavaScript

JavaScript je kompletní programovací jazyk, který umožňuje přidat na webové stránky dynamiku. JavaScript je jedním z nejlepších způsobů, jak zajišťovat reagování na události, které vyvolává návštěvník nebo prohlížeč na webové stránky. Je to jediný jazyk, který lze použít ve většině prohlížečů a který pracuje přímo s prohlížečem. Ostatní jazyky jako například Java, Perl, PHP nebo C nemají přímý přístup do obrázků, formulářů ani do oken tvořících webové stránky.

Naučit se JavaScript je jednoduché. Nepotřebujete žádný speciální hardware ani software, nepotřebujete přístup na webový server. Všechno, co potřebujete, je webový prohlížeč a textový editor. [11]

JavaScript musí být zapsán do HTML stejným způsobem jako CSS, a to externím připojením v podobě souboru, nebo přímým uvedením v hlavičce dokumentu (vzor příkazu HTML pro připojení souborů JavaScript `<script type="text/javascript">` Zde je zapsán kód JavaScript. `Neni HTML!</script>`).

```

12 ▼ <script type="text/javascript">
13     function setColor()
14     { document.getElementById("demo").style.backgroundColor = "red"; }
15 </script>

```

Obrázek 7: Vzorek kódu JavaScript

### 3.5.4 PHP

PHP, což je zkratka pro "PHP: Hypertextový Preprocesor", je open source skriptovací jazyk, určený pro všeobecné užívání. PHP bylo vytvořeno pro programování na



webu ale i může být vnořen do HTML. Hlavním, nikoli však jediným účelem jazyka PHP je umožnit tvorbu dynamických webových stránek.

PHP je často používáno jako moduli na webovém serveru, který také poskytuje možnost integrace a komplexní operace s databázemi a stažením určité informace ze serveru pro následující zobrazení v prohlížeči. [12]

### 3.5.5 SQL

SQL (Structured Query Language) je strukturovaný dotazovací jazyk. Jazyk SQL byl vyvinut tak, aby sloužil principům relačního modelu databáze, jenž je nejoblíbenějším modelem datového úložiště. V relačním databázovém modelu jsou data oddělena do množin, které připomínají tabulkovou strukturu. Jedná se o způsob, jakým jazyk SQL zpracovává data – jako množiny nebo skupiny. Jedná se o jazyk neprocedurální – SQL nepopisuje, jak se má s daty pracovat, ale na jakých datech se má pracovat. Pomocí SQL můžeme vytvářet tabulky, přidávat data, mazat data, spojovat data dohromady, spouštět akce na základě změn v databázi a ukládat dotazy do programu či databáze.

Například k vytvoření relační databáze sestávající z dat o zaměstnancích můžete začít s tabulkou nazvanou EMPLOYEE, jež obsahuje následující informace: EMP\_ID (identifikátor zaměstnance), LNAME (příjmení), FNAME (jméno) a DOB (datum narození). Tyto čtyři části dat tvoří pole v tabulce EMPLOYEE.

Databáze se v té či oné formě používají ve většině organizací pro správu důležitých podnikových dat. Bez databází by organizace byly nuceny i nadále uchovávat data v tištěné formě. Informace jsou nyní přístupné pouhým stiskem tlačítka myši. [13]

### 3.5.6 My SQL

V této práci jsem si zvolila databázový systém MySQL. Jde o nejoblíbenější databázi s otevřeným zdrojovým kódem, kterou lze snadno stáhnout a používat.

### 3.5.7 Textový editor

Jako textový editor pro svůj projekt jsem si vybrala PhpStorm, který se zaměřuje na práci s jazyky HTML, CSS a JavaScript a PHP. Tento editor je docela funkční a má přátelský a srozumitelný software.

## 4 Vlastní práce

Tato část práce bude obsahovat tři hlavní části: analýzu procesu a požadavků uživatelů, návrh IS a samotný vývoj IS. V těchto částech podrobně popíšu práci, kterou jsem vykonala při tvorbě informačního systému, a budu schopna uceleně prezentovat mnou vytvořený IS.

### 4.1 Analýza procesu vychystávání objednávek pomocí workflow diagram

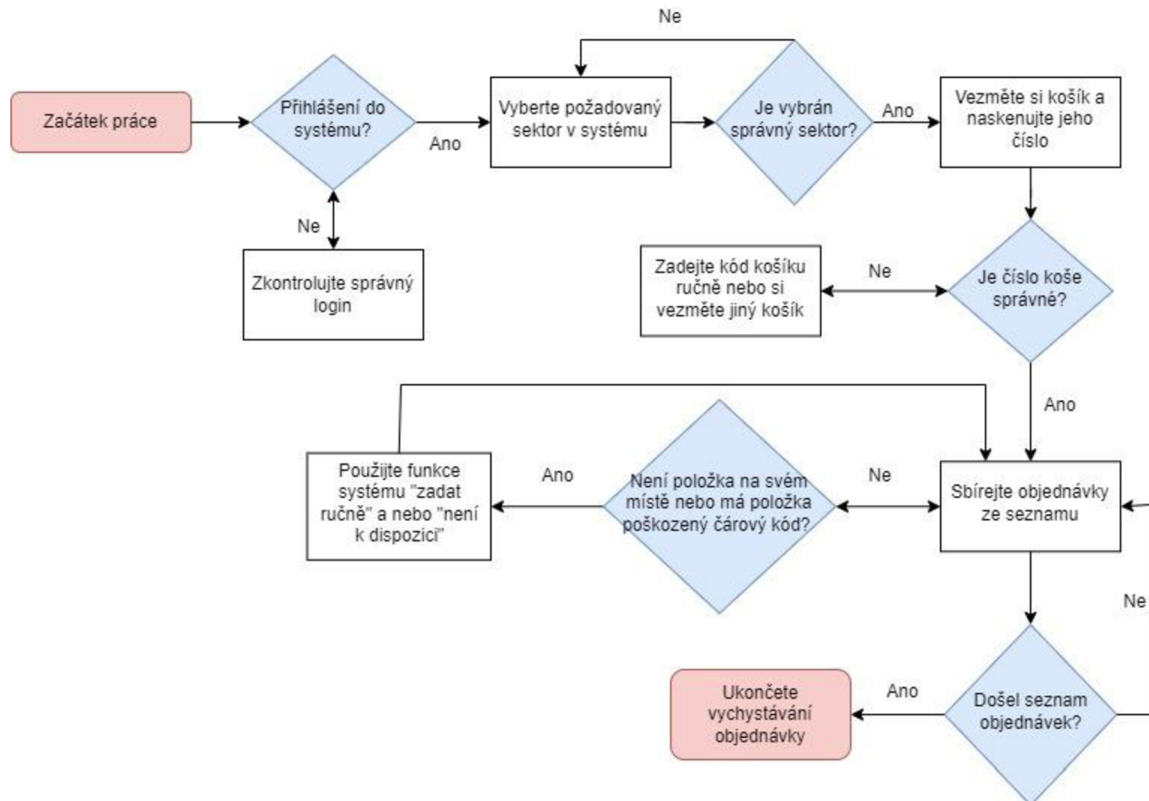
Pro vytvoření kvalitního informačního systému je třeba podrobně analyzovat vybraný proces vychystávání objednávek. Analýza se provede s využitím workflow diagramu, který umožňuje popsat pracovní postup od začátku do konce pomocí standardizovaných symbolů. V tomto diagramu budu používat tři symboly. Obdélník se zaoblenými okraji označuje začátek a konec procesu. Kosočtverec se používá k označení větvení, tj. situace, která má několik scénářů. Obdélníky označují akce.

Proces vychystávání objednávek nevyžaduje mnoho různorodých činností, ale přesné provedení potřebných činností v daném pořadí zajišťuje úspěšné dokončení práce.

- Nejdříve se musí zaměstnanec přihlásit. Pokud se přihlásit nemůže, měl by zkontrolovat, zda je přihlašovací login správný.
- Pokud je login správný, dalším krokem je přechod do sektoru a výběr správného sektoru v systému. Pokud je vybrán nesprávný sektor, systém umožní pracovníkovi vrátit se zpět a vybrat správný.
- Po výběru správného sektoru je třeba vzít košík, do něhož bude objednávka vychystávaná, a zadat jeho číslo, aby systém věděl, kde se objednávka nachází. Pokud se číslo nepodaří zadat, musí se zadat číslo ručně nebo použít jiný košík.
- Po zadání čísla nákupního košíku do systému přejde pracovník k samotnému vychystávání objednávky. Zobrazí se mu velký seznam, podle něhož musí postupovat. Pokud se však vyskytnou problémy s tím, že čárový kód nelze zadat nebo že výrobek není k dispozici, musí pracovník využít funkce systému: zadat kód ručně, nebo upozornit na nepřítomnost zboží.

- Proces vychystávání objednávek bude ukončen, jakmile budou shromážděny všechny položky ze seznamu.

Níže je uveden diagram popsaného procesu.



Obrázek 8: Workflow diagram procesu vychystávání objednávek

## 4.2 Požadavky na informační systém

V úvodu své práce jsem napsala, že jsem pracovala ve skladu (Amazon, Rohlik) a ke shromáždění požadavků jsem využila kontakty na zaměstnance těchto dvou velkých firem.

Za účelem analýzy požadavku jsem provedla průzkum mezi hlavními uživateli navrhovaného IS – zaměstnanci skladu. Po zpracování dat jsem konzultovala s vedoucím skladu schválení dotazů a mohu poskytnout seznam hlavních dotazů na IS. Níže je uveden seznam prioritních dotazů, z průzkumu vyplynulo také několik dotazů, které by mohly být řešeny v budoucích verzích vyvíjeného IS.

### 4.2.1 Obdržené požadavky

- Jednoduché a přehledné rozhraní systému.

- Požadavek na jednoduché rozhraní bez zbytečných funkcí. Pracovník skladu by měl pracovat co nejefektivněji a neztrácet čas zbytečnými úkony v systému.
- Řešení: systém bude navržen se všemi prioritními funkcemi pro efektivní provoz. Systém bude rovněž navržen v jednobarevném provedení.
- Přihlášení do systému pod osobním číslem zaměstnance.
  - Tato systémová funkce umožní použití terminálu pro sběr dat různými pracovníky. Tato funkce bude také přiřazovat vykonanou práci jednotlivým pracovníkům.
  - Řešení: každý zaměstnanec má své vlastní číslo, které bude moci zadat do systému pomocí tlačítka „přihlásit se“ a „odhlásit se“.
- Možnost označit, že položka na požadované adrese došla.
  - Tato funkce pomůže zaměstnanci přejít k další položce a upozornit vedoucího směny, že je třeba položku dokončit. Tato funkce zvýší rychlost zaměstnance a zlepší komunikaci mezi zaměstnancem a vedoucím směny.
  - Řešení: speciální tlačítko „není k dispozici“.
- Možnost zobrazit adresu další pozice.
  - Tato funkce usnadňuje pracovníkovi orientaci v prostoru, může si v hlavě předem vytyčit nejlepší trasu.
  - Řešení: zobrazení všech objednávek na jedné stránce.
- Možnost pozastavit práci.
  - Zaměstnanci měli požadavek, aby mohli objednávku kdykoli zrušit, ale po konzultaci s vedoucím skladu se ukázalo, že to není možné. Dospěli jsme k závěru, že zaměstnanec může svou objednávku pozastavit, aby si mohl na chvíli odskočit.
  - Řešení: tlačítko „odhlásit se“ bude fungovat pouze v případě, že je objednávka dokončena. Pokud objednávka není dokončena, systém vygeneruje chybu.
- Pole pro ruční zadání kódu zboží.
  - Někdy se adresní kódy vymažou a terminál pro sběr dat není vždy schopen kód přečíst. Pro urychlení práce je třeba zadat pracovníka ručně.
  - Řešení: pole „nastavit adresu ručně“. Toto pole lze kdykoli vyvolat.

## 4.3 Návrh IS

### 4.3.1 Návrh architektury

Architektura vyvíjeného informačního systému se skládá z několika klíčových komponent:

- Databáze, ve které jsou uloženy informace o zboží na skladě, včetně skladových zásob, umístění zboží a dalších důležitých údajů.
- Uživatelské rozhraní, které umožňuje pracovníkům skladu komunikovat se systémem, zadávat data a přijímat pokyny k činnosti.
- Obchodní logika, v níž se zpracovávají údaje, jako jsou příchozí objednávky, výpočet úrovně zásob, vytváření úkolů pro pracovníka a koordinace toku zboží.
- Reporting a analytika: tento modul poskytuje data a analytiku o výkonu práce.
- Zabezpečte systém připojením k interní síti firmy pomocí samostatné přístupové politiky a systému hesel pro všechny účty.

### 4.3.2 Návrh databáze. Entity relationship diagram

Vytvořená databáze obsahuje všechny informace potřebné pro nastavení a provoz informačního systému.

#### 4.3.2.1 Tabulka „storehouse“

První tabulka (sklad) obsahuje údaje o skladu, které vytvářejí jeho číselnou mapu. Tato tabulka obsahuje informace o sekcích skladu (section), z nichž každá je rozdělena na ulice (street), které jsou označeny písmeny. Na každé ulici jsou pak 4 úrovně regálů, z nichž každá je očíslována vertikálně (shelf number) a horizontálně (box number).

Pracovník uvidí adresu ve formátu 1-A-2-3, což znamená, že požadované zboží se nachází v prvním sektoru, v ulici A, na druhém regálu v krabici číslo 3.

#### 4.3.2.2 Tabulka „goods“

Další tabulka (zboží) bude obsahovat informace o zboží na skladě. Každé zboží má svůj název (gd\_name), výrobce (producer). Každý výrobek má čárový kód (gd\_barcode), který je rovněž uložen v systému. Databáze obsahuje také informace o zásobách výrobků ve skladu, které jsou uloženy v atributu (leftovers).

Informace o přesném umístění výrobku je v atributu (gd\_idstoregouse), který je externím klíčem a jenž je pomocí id propojen s tabulkou sklad. Externí klíče a vazby mezi tabulkami umožňují nepřetěžovat tabulky v databázi, pokud je stejná informace potřebná v několika tabulkách.

#### 4.3.2.3 Tabulka „cart“ a „mobile“

Následující informace potřebné pro systém vozík i terminál pro sběr dat jsou uloženy v tabulkách “cart” a “mobile”. Každá z těchto pracovních položek má svůj vlastní čárový kód uložený v attributech (cart\_barcode, mb\_barcode).

Čárový kód vozíku je potřebný pro další sledování objednávky ve skladu. Čárový kód na terminálu pro sběr dat je potřebný pro sledování, který zaměstnanec použil který terminál.

#### 4.3.2.4 Tabulka „task“

Následující tabulka obsahuje informace, které představují konkrétní úkol pro pracovníka (task) a obsahují základní údaje potřebné k provedení sběru zakázek. Atribut (ts\_idgoods) je externím klíčem k tabulce zboží (goods), s jehož pomocí můžeme získat pouze ty informace o výrobku, které jsou nezbytné pro proces montáže (název, výrobce, čárový kód, umístění výrobku).

Atribut (quantity) informuje o tom, kolik konkrétního produktu má být odebráno pro konkrétní objednávku. Atribut (ts\_idcart) obsahuje informaci o čísle košíku použitého k odběru objednávky a je externím klíčem k tabulce košíku.

Atribut (ts\_idworker) obsahuje informace o pracovníkovi, který vybírá konkrétní zakázku. Tento atribut je externím kódem a je propojen s tabulkou worker, ze které získává všechny potřebné informace o pracovníkovi.

#### 4.3.2.5 Tabulka „worker“

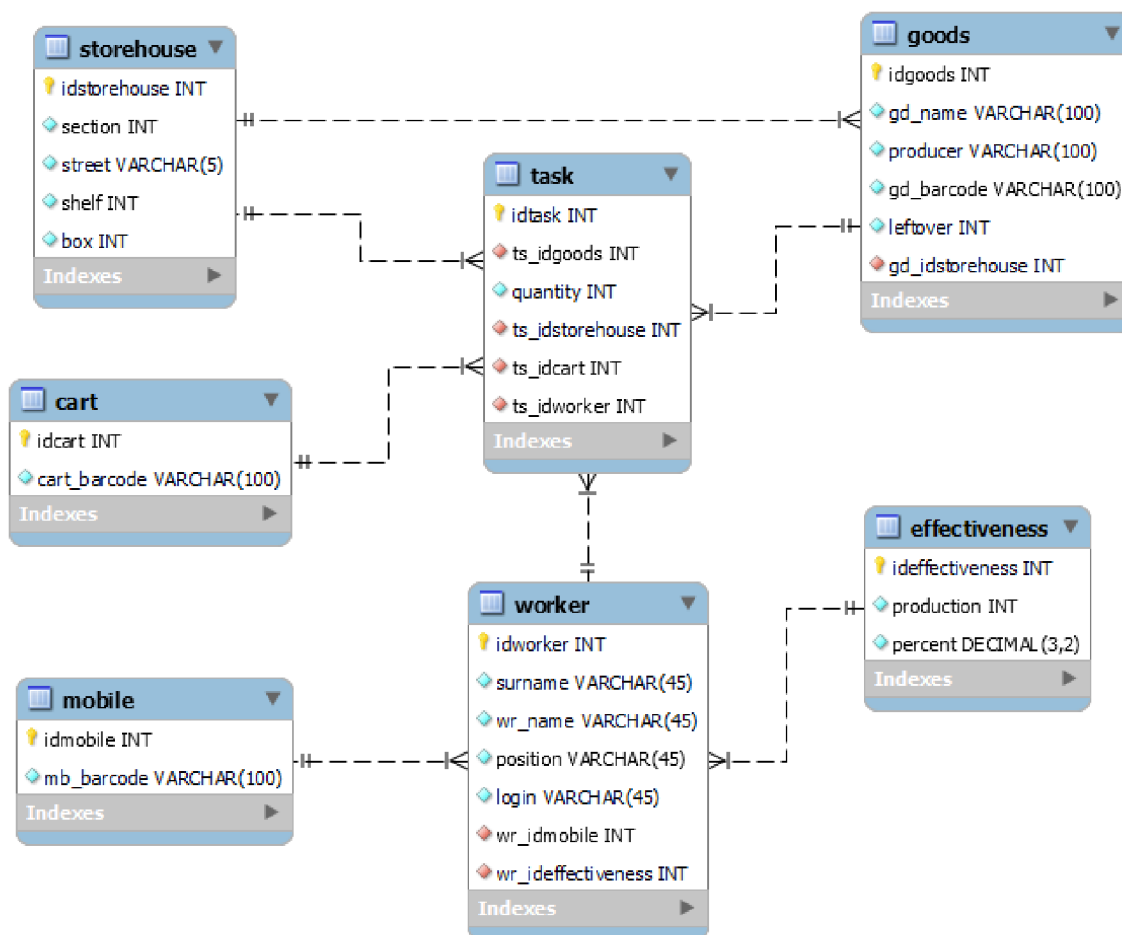
Následující tabulka „worker“ obsahuje informace o pracovníkovi – příjmení (surname), jméno (wr\_name) a pozici (position) ve skladu. Zaměstnanec musí mít k práci terminál pro sběr dat, do kterého se přihlašuje pomocí jedinečného přihlašovacího jména zaměstnance, jež je uloženo v atributu «login». Přihlašování pomocí jedinečného přihlašovacího jména je nezbytné pro zajištění bezpečnosti informačního systému.

Informace o tom, který zaměstnanec má terminál, je uložena v atributu „wr\_idmobile“, který je externím vodítkem a je propojen s tabulkou «mobile». Informace o efektivitě zaměstnance je uložena v atributu wr\_idefektivita a počítá se podle počtu odebraného zboží. Tento atribut je externím klíčem a je propojen s tabulkou „effectiveness“.

#### 4.3.2.6 Tabulka „effectiveness“

Poslední tabulka ukazuje výkonnost zaměstnance z hlediska produkce (production) a procenta (percent) odebraného zboží.

ERD popisované databáze je znázorněno níže . Diagram znázorňuje informační toky, které se používají ve vyvíjeném informačním systému, a vztahy mezi těmito daty.



Obrázek 9: Entity relationship diagram

#### 4.3.3 Grafický návrh uživatelského rozhraní

Grafický návrh je vzorem toho, jak by měly webové stránky vypadat. Pomáhá při plánování budoucího projektu: kde budou na webu umístěny objekty nebo jak bude vypadat uživatelské rozhraní. Návrh uživatelského rozhraní je vizuální „rozvržení“ informačního

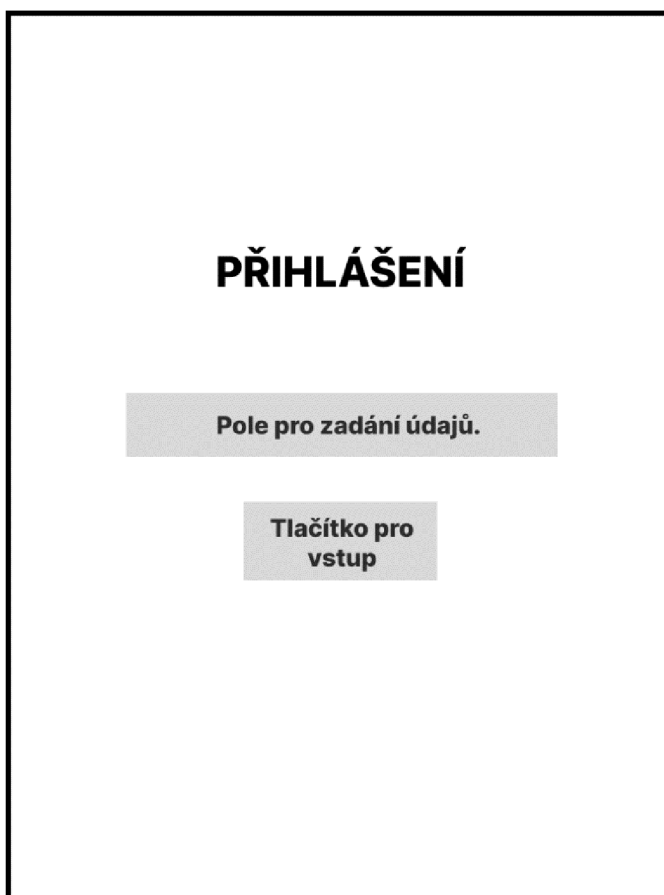
systemu. V konečném důsledku se informační systém nebude funkčně lišit od navrženého rozvržení.

Uživatelské rozhraní je součástí systému, který vidí uživatel. Zaměstnanec komunikuje se systémem prostřednictvím navrženého rozhraní, a proto je tak důležité vytvořit efektivní a srozumitelné rozhraní, které bude plnit hlavní funkce projektu – usnadňovat a zjednodušovat zaměstnanci práci.

Níže vyvinutý grafický návrh je jednoduchý a přímočarý, protože tento systém nevyžaduje složitý design, ale dobře navrženou funkčnost.

#### 4.3.3.1 Přihlášení uživatele

Obrázek č. 10 zobrazuje grafický návrh úvodní přihlašovací stránky k informačnímu systému. Uprostřed stránky je záhlaví, pod ním pole pro zadání individuálního přihlašovacího jména zaměstnance a tlačítko, po jehož stisknutí se dostanete na další stránku informačního systému, pokud je přihlašovací jméno správné.



**Obrázek 10: Grafický návrh - Přihlašovací stránka**



#### 4.3.3.2 Výběr sektoru

Obrázek č. 11 ukazuje grafiku na druhé straně systému – výběr sektoru. Tato část systému umožňuje vybrat sektor, ve kterém bude probíhat práce. Obvykle je sklad rozdělen do různých sektorů v závislosti na skladování výrobků. V tomto příkladu je sklad rozdělen do tří sektorů. Počet sektorů se liší v závislosti na velikosti a rozsahu poskytovaného skladu.

V levém horním rohu vidíte individuální číslo pečovatele, který se přihlásil. Tato funkce pomůže zaměstnanci, aby si nepletl terminál pro sběr dat, se kterým pracoval. Pod středem obrazovky je záhlaví, pod ním jsou tlačítka, z nichž každé vede k objednávce v požadovaném sektoru. Nejspodnějším tlačítkem lze potvrdit výběr sektoru.

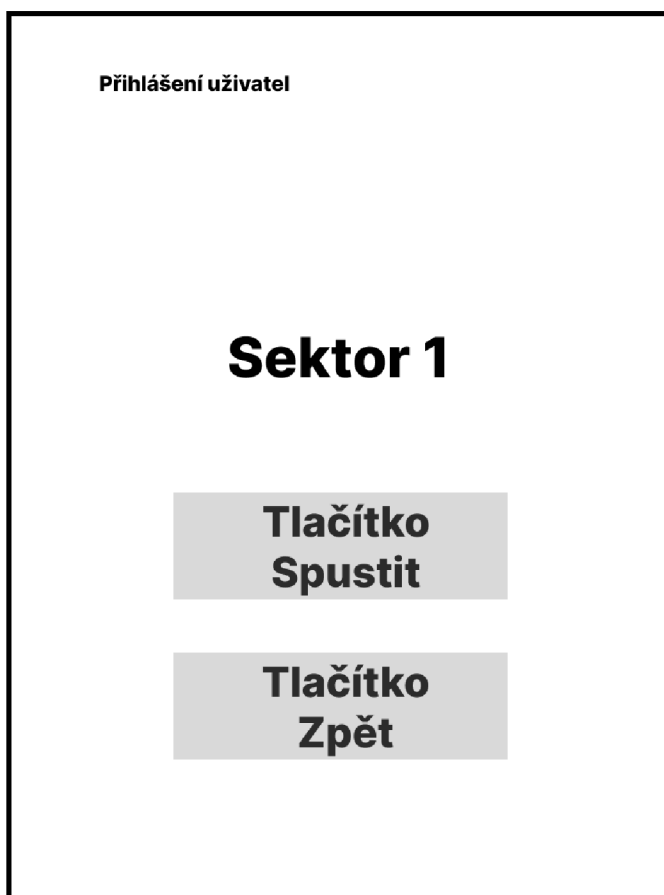


Obrázek 11: Grafický návrh - Výběr sektoru

#### 4.3.3.3 Kontrola výběru sektoru

Obrázek č. 12 ukazuje grafický návrh třetí strany vyvíjeného IS. Po výběru sektoru vyzve systém zaměstnance k zahájení práce ve vybraném sektoru. Stisknutím tlačítka „Spustit“ může zaměstnanec potvrdit, že zvolil správný sektor, a pokračovat ve sběru zakázek. Pokud se zaměstnanec spletl, může stisknout tlačítko „Zpět“, vrátit se na stránku a

vybrat sektor znovu. Tato funkce pomáhá rovnoměrně rozdělovat pracovníky do sektorů a předcházet chybám.



**Obrázek 12: Grafický návrh - Kontrola výběru sektoru**

#### 4.3.3.4 Zadání čísla košíku

Obrázek č. 13 ukazuje grafický návrh čtvrté strany IS. Na starém obrázku je zobrazeno záhlaví, pod ním je pole pro zadání čísla nákupního košíku a tlačítko, kterým lze přejít na další stránku.

V této části systému musí zaměstnanec vzít košík a naskenovat číslo, tím se objednávka přiřadí ke konkrétnímu košíku v systému, což v budoucnu usnadní vyhledání odebrané objednávky. Pokud se číslo na koši vymaže, může pracovník zadat číslo ručně, aby neztrácel čas hledáním nového koše.

Přihlášení uživatel

**ZADEJTE ČÍSLO  
KOŠÍKU**

POLE PRO ZADÁNÍ ČÍSLA

TLAČÍTKO DALŠÍ

**Obrázek 13: Grafický návrh - Zadání čísla košíku**

#### 4.3.3.5 Vychystávání objednávek

Obrázek č. 14 zobrazuje grafický návrh další stránky IS. Na stránce vidíme fotografii zboží, všechny potřebné informace o něm, pole pro zadání čárového kódu a dvě tlačítka. Pokud zboží na dané adrese chybí, může zaměstnanec kliknout na tlačítko „není k dispozici“ a systém ho přesune na další položku. Pokud je naopak čárový kód výrobku poškozen, může pracovník zadat kód ručně pomocí tlačítka „zadat kód ručně“.

Stránka je dlouhý seznam, v němž jsou položky za sebou. To pomáhá zaměstnanci zjistit, kterou položku má vyzvednout jako další a kolik položek je v pořadí. Tyto funkce urychlí práci zaměstnance.

**Přihlášení uživatel**



**Adresa zboží**  
**Název zboží**  
**Výrobce**  
**Množství**

**pole pro zadávání čárového kódu**

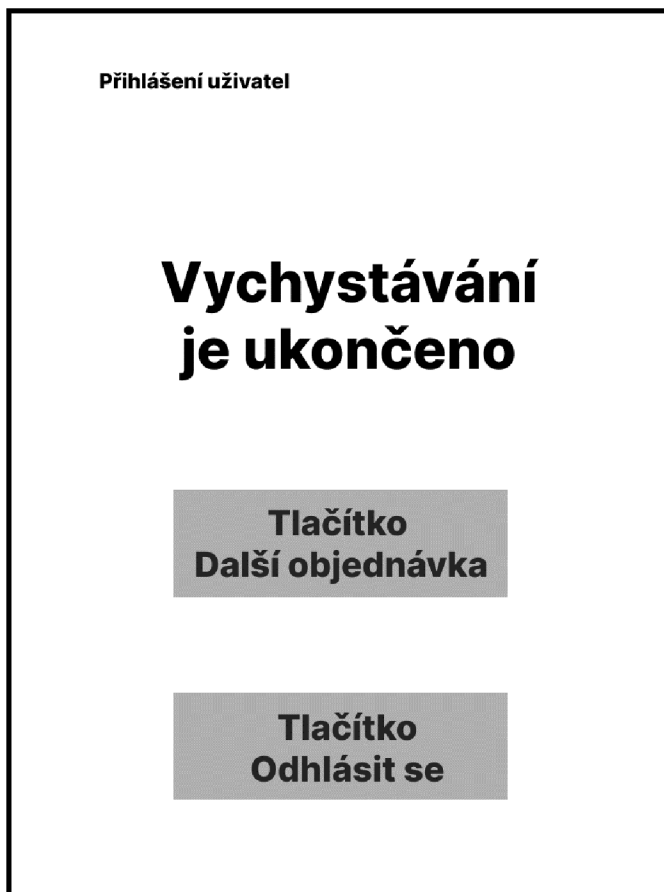
**Tlačítko  
není k dispozici**

**Tlačítko  
ručně zadat kod**

**Obrázek 14: Grafický návrh - Vychystávání objednávek**

#### 4.3.3.6 Poslední stránka systému

Na obrázku č. 15 je zobrazen grafický návrh poslední stránky webové aplikace. Stránka informuje zaměstnance, že vychystávání objednávek je ukončeno, a vyzývá ho, aby přešel na další zakázku. V takovém případě se dostane na stránku pro výběr sektoru a může začít znovu vybírat zakázku (obrázek číslo 11). Pokud chce zaměstnanec zakázku dokončit nebo přejít do jiného sektoru, bude se moci ze systému odhlásit.



**Obrázek 15: Grafický návrh - Poslední stránka systému**

#### 4.4 Vývoj informačního systému

V následujících kapitolách se seznámíte s tvorbou aplikace. Předvedu důležité části kódu a komunikaci s databází. Pro implementaci byl využit software PhpStorm, pro práci s databází byla využita webová služba PhpMyAdmin, jež je součástí webového serveru.

##### 4.4.1 Vytvoření struktury databáze

Práce s databází probíhala pomocí lokálního webového serveru Open Server Panel, který slouží k vytvoření lokálního vývojového prostředí. Vývoj informačního systému jsem zahájila tvorbou databázových tabulek. Přistupovala jsem k tomu přes PhpMyAdmin, který mi umožnil naplnit databázové tabulky položkami a vytvoření vazeb mezi nimi.

Vyplnila jsem všechny tabulky v databázi, abych vytvořila informace o skladu, se kterými mohu pracovat. Pokud se podíváte na všechny tabulky, je jasné, že všechny informace, které procházejí skladem, musí být uloženy v databázi, přidány do ní a upraveny. Díky dobře vytvořené databázi je možné efektivně řídit skladové zásoby a sledovat práci zaměstnanců.

Pomocí příkazu SQL “SELECT \* FROM nazev tabulky” vypisují všechny tabulky.

Pro vyplnění databázových tabulek jsem použila příkaz SQL “INSERT INTO nazev tabulky (atribut, atribut) values (hodnoty, hodnoty)”.

idstorehouse	section	street	shelf	box
1	1	A	1	1
2	1	A	2	5
3	1	A	3	10
4	1	B	2	8
5	1	B	4	3
6	1	C	1	6
7	1	C	3	5
8	1	D	3	10
9	1	B	3	5
10	1	D	2	5

**Obrázek 16: Naplnění tabulky "storehouse"**

INSERT INTO `storehouse` (`idstorehouse`, `section`, `street`, `shelf`, `box`)  
VALUES (1, 1, 'A', 1, 1),(2, 1, 'A', 2, 5),(3, 1, 'A', 3, 10) atd..

idgoods	gd_name	producer	gd_barcode	leftover	gd_idstorehouse	foto
1	Jihočeské mléko lahodné polotučné 1,5% 1L	Madeta	8593803222301	30	1	1
2	Chléb toastový světlý 500g	Ölz	9005300011535	25	2	2
3	Ochucený pomelo/grep nealkoholické pivo 500ml	Birell	8594404009520	59	3	3
4	Čokoláda Triple caramel 90g	Milka	7622210516107	46	4	4
5	Jablko 100% džus 1L	Relax	8595646200638	25	5	5
6	Mouka hladká dortová 1kg	Babiččina volba	8594060910284	58	6	6
7	Rýže basmati 500g	Fine Life	8595237400508	35	7	7
8	Víno Modrý Portugal 1L	Mikulov	8594044835787	65	8	8
9	Voda Červené jablko neperlivá 1,5L	Dobrá voda	8594001405978	47	9	9
10	Fromage chipsy 215g	Lay's	5900259097163	34	10	10

**Obrázek 17: Naplnění tabulky "goods"**

INSERT INTO `goods` (`idgoods`, `gd\_name`, `producer`, `gd\_barcode`, `leftover`,  
`gd\_idstorehouse`, `foto`) VALUES

(1, 'Jihočeské mléko lahodné polotučné 1,5% 1L', 'Madeta', '8593803222301', 30, 1, '1'),

(2, 'Chléb toastový světlý 500g', 'Ölz', '9005300011535', 25, 2, '2'),

(3, 'Ochucený pomelo/grep nealkoholické pivo 500ml', 'Birell', '8594404009520', 59, 3, '3')

atd.

idworker	surname	wr_name	position	login	wr_idmobile	wr_ideffectiveness
1	Svoboda	Marek	picker	125svomar764	1	1
2	Procházka	Filip	picker	582profil753	2	2
3	Horák	Karel	picker	573horkar474	3	3
4	Veselý	Dominik	picker	256vesdom858	4	5

**Obrázek 18: Naplnění tabulky "worker"**

INSERT INTO `worker` (`idworker`, `surname`, `wr\_name`, `position`, `login`,  
`wr\_idmobile`, `wr\_ideffectiveness`) VALUES

(1, 'Svoboda', 'Marek', 'picker', '125svomar764', 1, 1),

(2, 'Procházka', 'Filip', 'picker', '582profil753', 2, 2),  
 (3, 'Horák', 'Karel', 'picker', '573horkar474', 3, 3) atd.

idtask	ts_idgoods	quantity	ts_idstorehouse	ts_idcart	ts_idworker
1	1	2	1	1	1
2	2	1	2	1	1
3	3	3	3	1	1
4	4	1	4	1	1
5	5	2	5	1	1
6	6	1	6	1	1
7	7	1	7	1	1
8	8	2	8	1	1
9	9	1	9	1	1
10	10	1	10	1	1

**Obrázek 19: Naplnění tabulky "task"**

```
INSERT INTO `task` (`idtask`, `ts_idgoods`, `quantity`, `ts_idstorehouse`,
`ts_idcart`, `ts_idworker`) VALUES (1, 1, 2, 1, 1, 1), (2, 2, 1, 2, 1, 1), (3, 3, 3, 3, 1, 1).
```

Tabulka Task je tvořena především externími klíči, a proto obsahuje tolik odkazů na «id» jiných tabulek.

idmobile	mb_barcode
1	100386
2	100546
3	100765
4	100543

**Obrázek 20: Naplnění tabulky "mobile"**

idcart	cr_barcode
1	12345678
2	23579076
3	46942807
4	70396838

**Obrázek 21: Naplnění tabulky "cart"**

#### 4.4.2 Tvorba webové aplikace

Pro implementaci IS byl využit software PhpStorm. Vývoj byl prováděn s pomocí jazyků HTML, CSS, JavaScript a PHP.

##### 4.4.2.1 Připojení do databáze

Prvním krokem při vytváření systému je spojení webové aplikace s databází. Spojení bylo navázáno pomocí této části kódu, příkaz je součástí každé stránky a kontroluje spojení. Pokud spojení není navázáno, systém vygeneruje chybu.

```
<?php
$conn = mysqli_connect("127.0.0.1", "root", "", "storehouse", "3306");
if (!$conn) {
    die('Error');
}
```

```
session_start();  
?>
```

#### 4.4.2.2 Nastavení HTML kódu

Rozložení každé stránky bude zapsáno v jazyce HTML a tento kód musí být správně nastaven.

```
<!DOCTYPE html>
```

Nastavení verze znamená, že web je vytvořen pomocí značek html5, a říká prohlížeči, aby poskytl přístup ke všem vlastnostem a prvkům html5.

```
<html lang="cs">
```

Nastavení jazyka, ve kterém bude aplikace vytvořena. V mém případě je to čeština.

```
<head>
```

```
<meta charset="UTF-8">
```

Kódování UTF-8 je standardizované kódování textu, které umožňuje ukládat znaky Unicode.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Značka `<meta="viewport">` řídí měřítko zobrazení stránky a umožňuje otevření webové aplikace na různých zařízeních bez narušení rozvržení stránky.

```
<link rel="stylesheet" href="/css/normalize.css">
```

```
<link rel="stylesheet" href="/css/style.css">
```

```
<script src="js/script1.js"></script>
```

Na závěr přidám validní odkazy, jež umožňují připojení k validním souborům CSS a JavaScript.

#### 4.4.2.3 Implementace přihlášení uživatele

Po zajištění a kontrole připojení k databázi přejdu k dalšímu kroku – vytvoření přihlašovací stránky. Rozložení bylo provedeno pomocí jazyka HTML a jazyka kaskádových stylů CSS.

```
<body>
```

```
<div class="system">
```

```
<header>
```

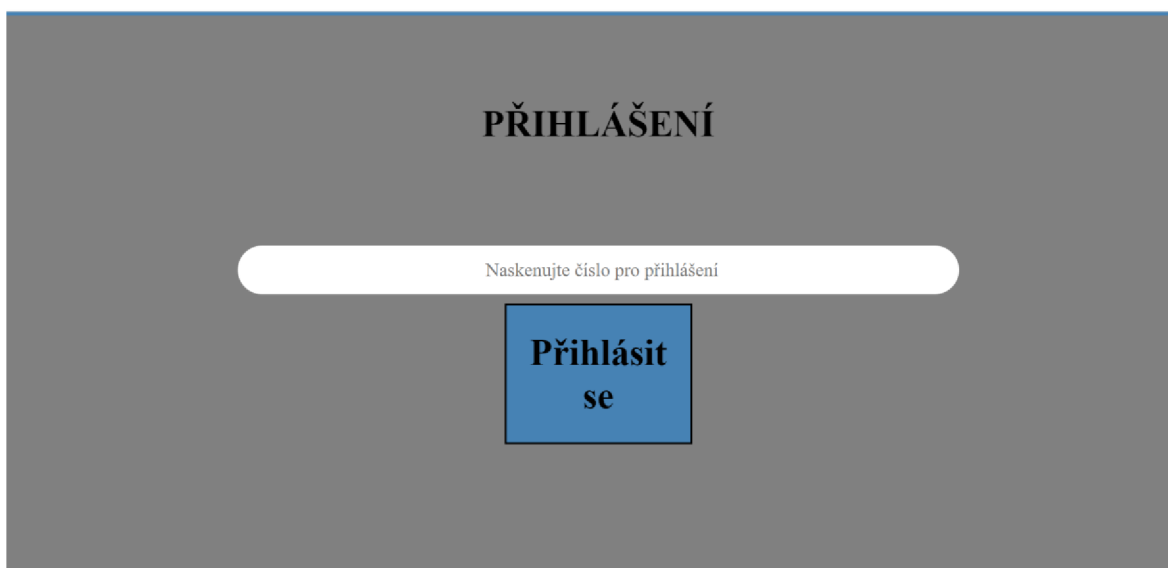


```

<div class="header-container"></div>
</header>
<div class="start">
  <form id="form" method="POST" action="inc/vstup.php">
    <h1>PŘIHLÁŠENÍ</h1>
    <label class="form-container">
      <input type="text" name="login" placeholder="Naskenujte číslo pro přihlášení">
    </label>
    <button type="submit" name="submit">Přihlásit se</button>
  </form>
</div>
</div>

```

Byla vytvořena hlavička stránky `<header>` a pomocí jazyka CSS byla stylizována do malého modrého pruhu. Bylo vytvořeno pole pro zadávání dat `<input type="text">` a tlačítko pro přihlášení `<button type="submit" >`. Tlačítku jsem přiřadila `name="submit"` a poli `name="login"`, aby je serverový jazyk PHP našel v kódu HTML. Formuláři, který obsahuje veškerý kód HTML, jsem přiřadila `<method="POST">` a `<action="inc/vstup.php">`, který odkazuje na soubor, v němž je uložen kód PHP, a zajišťuje provedení akcí nastavených tímto jazykem na straně serveru.



**Obrázek 22: Screenshot dokončeného systému - přihlášení uživatele**

Dalším krokem je zajistit, aby byla do pole zadána platná hodnota. To vyžaduje vyřešit načítání uživatele z databáze a vytvořit kód, který po kliknutí na tlačítko ověří, zda hodnota zadaná do pole existuje v databázi. Teprve po nalezení shody bude moci systém přejít na další stránku. Níže je uveden kód PHP, který tuto kontrolu provede.

Pak jsem napsala příkaz SQL, který říká, že se má vybrat taková hodnota „login” z tabulky „worker”, která bude odpovídat hodnotě zadané v poli. Následně jsem napsala podmínku, která říká, že pokud bude nalezena alespoň jedna shoda, musím přejít na další stránku. Pokud se nenajde žádná shodná hodnota, systém zůstane na stejné stránce.

Upozorním, že odkaz na ostatní stránky je zapsán ve formátu «../», což znamená, že soubor, na který je odkazováno, je v jiném adresáři a systém musí přejít ze složky, ve které se nachází, do hlavního adresáře, aby tento soubor našel.

```
if (isset($_POST['submit'])) {
    $login = $_POST['login'];

    $sql = "SELECT * FROM worker WHERE login = '$login'";
    $result = mysqli_query($connect, $sql);

    if (mysqli_num_rows($result) > 0) {
        $user=mysqli_fetch_assoc($result);
        $_SESSION['user']=[
            "login"=>$user['login']
        ];
        header('Location: ../sector.php');
        exit;
    } else {
        $error_message = "Invalid login";
        header('Location: ../index.html');
    }
    unset($error_message);
}

mysqli_close($connect);
```

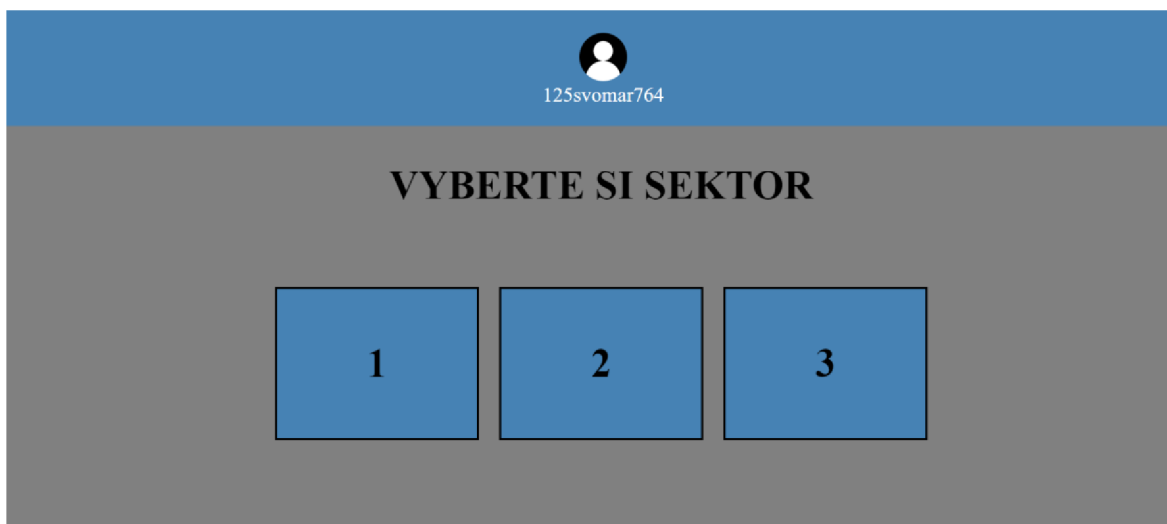
#### 4.4.2.4 Implementace výběr sektoru a kontrola

Rozložení HTML stránek pro výběr sektoru a jejich validace jsou umístěné v jednom souboru a jsou rozděleny do samostatných částí pomocí elementu `<fieldset>`. Níže je uveden příklad rozvržení hlavní stránky výběru a doplňkové stránky s ověřením vybraného Sektoru 1.

```
<fieldset class="form_fieldset" id="sector1-fieldset" style="display:block;">
  <header>
    <div class="header-container">
      <div class="system_head_user">
        
        <p><?= $_SESSION['user']['login'] ?></p>
      </div>
    </div>
  </header>
  <div class="container">
    <h1 class="title">Sektor 1</h1>
    <div class="buttons">
      <button class="continue">Spustit</button>
      <button class="log-out" id="log-out1">Zpět</button>
    </div>
  </div>
</fieldset>
```

Formát tohoto souboru, ve kterém je kód zapsán, je `.php`, což umožňuje zapsat kód PHP uvnitř kódu HTML. Například `<p><?= $_SESSION['user']['login'] ?></p>` umožňuje zobrazit v záhlaví profilu přihlašovací jméno, které bylo zadáno do systému.

Každý `<fieldset>` a tlačítko mají individuální `id`, které je nezbytné pro další práci s JavaScriptem.



**Obrázek 23: Screenshot dokončeného systému - výběr sektoru**

Dále je třeba umožnit přepínání mezi prvky `<fieldset>` kliknutím na příslušná tlačítka. K tomu slouží kód JavaScriptu, který pracuje s vytvořenými proměnnými.

Vytvořím proměnné pro všechna tlačítka "sector1Btn" a pro všechny prvky `<fieldset>`, které mi umožní nastavit jim jednotlivé příkazy. Po vytvoření proměnných nastavím příkaz, které říká, že při otevření stránky bude viditelný pouze první `<fieldset>` = blok a všechny ostatní budou skryté = none.

```
document.addEventListener('DOMContentLoaded', function () {
  "use strict";

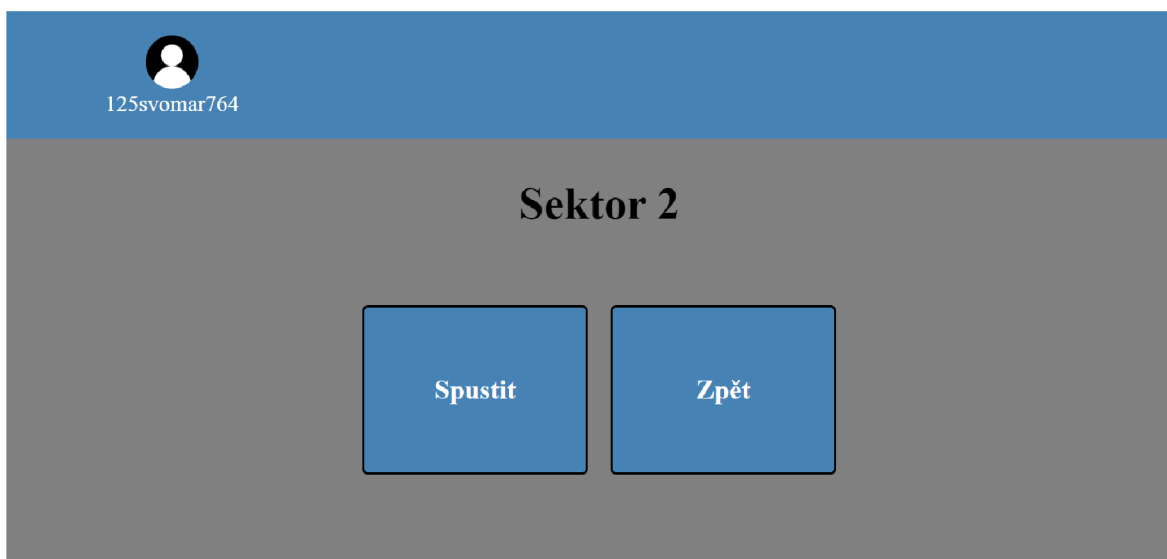
  const sector1Btn = document.getElementById("sector1-btn");
  const sector2Btn = document.getElementById("sector2-btn");
  const sector3Btn = document.getElementById("sector3-btn");
  const sectorFieldset = document.querySelectorAll('.form_fieldset')[0]; //veber
  const sector1Fieldset = document.querySelectorAll('.form_fieldset')[1]; //sektor 1
  const sector2Fieldset = document.querySelectorAll('.form_fieldset')[2]; //sektor 2
  const sector3Fieldset = document.querySelectorAll('.form_fieldset')[3]; //sektor 3

  sectorFieldset.style.display = "block";
  sector1Fieldset.style.display = "none";
  sector2Fieldset.style.display = "none";
  sector3Fieldset.style.display = "none";
```

Dalším krokem přiřadím každému z tlačítek akci „click“. Kliknutím na '1' tlačítko se zviditelní pouze druhý element <fieldset> a všechny ostatní budou zakázány.

```
sector1Btn.addEventListener('click', function () {  
    sectorFieldset.style.display = 'none'  
    sector2Fieldset.style.display = 'none';  
    sector3Fieldset.style.display = 'none';  
    sector1Fieldset.style.display = 'block';  
});
```

Stránky s potvrzením vybraného sektoru obsahují dvě tlačítka, jejichž akce je rovněž třeba nastavit pomocí JavaScriptu.



**Obrázek 24: Screenshot dokončeného systému - kontrola výběru sektoru**

Pro každé z tlačítek vytvořím proměnné v jazyce JavaScript, tentokrát prostřednictvím tříd, které byly určeny v kódu HTML. To umožňuje zadat stejnou akci všem tlačítkům v určené třídě s jediným pořadím (id prvků v HTML musí být individuální).

Tlačítka „log out“ aktivují první <fieldset> s výběrem sektoru a deaktivují ostatní.

Tlačítko s třídou „continue“ odkazuje a přenese uživatele na další soubor.

```
const backSectorsBtn = document.querySelectorAll('.log-out');  
  
backSectorsBtn.forEach(btn => {  
    btn.addEventListener("click", (event) => {  
        event.preventDefault();  
        sectorFieldset.style.display = "block";  
    });  
});
```

```

sector1Fieldset.style.display = "none";
sector2Fieldset.style.display = "none";
sector3Fieldset.style.display = "none";
});
});

const continueSectorsBtn = document.querySelectorAll(".continue");

continueSectorsBtn.forEach(btn => {
  btn.addEventListener("click", () => {
    window.location.href = "kosik.php";
  });
});
});

```

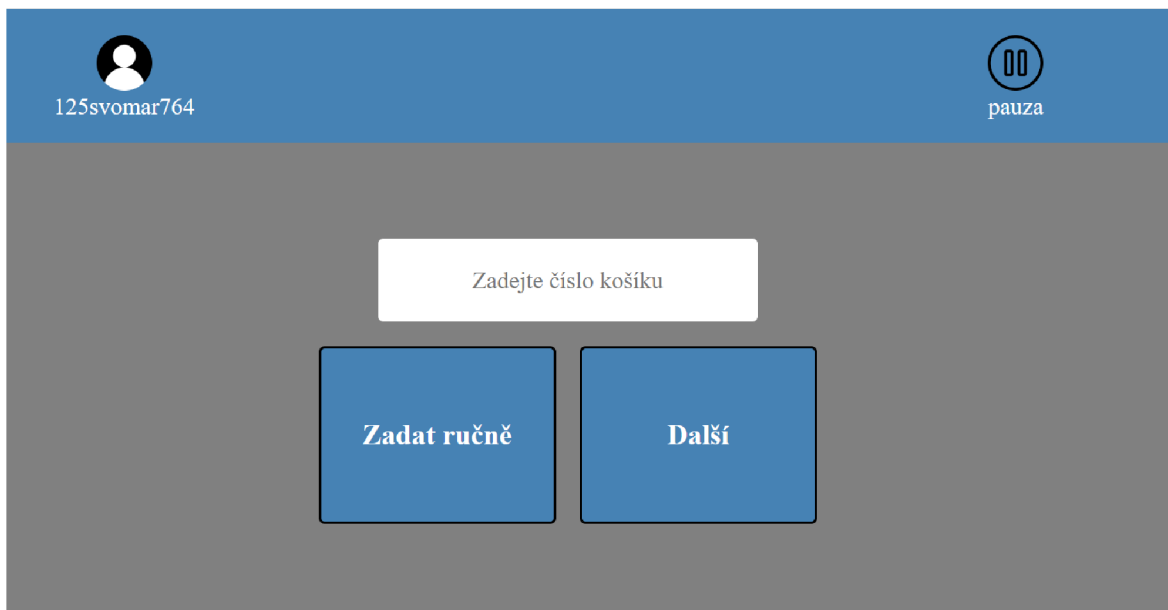
#### 4.4.2.5 Implementace zadání čísla košíku a kontrola

V rozložení HTML na další stránce bylo vytvořeno pole pro zadávání dat a dvě tlačítka. Aby bylo možné zkontrolovat zadané číslo košů, což se provede po stisknutí tlačítka s individuálním názvem `<name="next">`, je nutné vyřešit načítání čárových kódů všech košů z databáze. Za tímto účelem přiřaďte individuální název poli pro zadávání údajů `<input name="cart">` a vytvořte odkaz na soubor s kódem PHP `<action="inc/datakosik.php">`.

```

<h1>Naskenujte číslo košíku</h1>
<form class="form-container" method="POST" action="inc/datakosik.php">
  <label><input type="text" id="cart-number" name="cart" placeholder="Zadejte číslo košíku"></label>
  <div class="buttons">
    <button class="log-out" type="button" id="manual-input">Zadat ručně</button>
    <button class="continue" name="next">Další</button>
  </div>
</form>

```



**Obrázek 25: Screenshot dokončeného systému - zadání čísla košíku a kontrola**

Kód PHP je téměř totožný s kódem v kapitole 4.4.2.3 Implementace přihlášení uživatele, změnily se pouze hodnoty.

```
if (isset($_POST['next'])) {  
    $cartBarcode = $_POST['cart'];  
    $sql = "SELECT * FROM cart WHERE cr_barcode = '$cartBarcode'";  
    $result = mysqli_query($connect, $sql);  
    if (mysqli_num_rows($result) > 0) {  
        header("Location: ../task.php");  
        exit();  
    }  
    else {  
        $error_message = "Invalid login";  
        header('Location: ../kosik.php');  
    }  
    mysqli_close($connect);  
}
```

Pomocí JavaScriptu jsem nastavila akce tlačítka s přiřazeným `<id="manual-input">`. Vytvořila jsem dvě proměnné: první proměnná je zodpovědná za tlačítko, druhá proměnná je zodpovědná za pole pro zadávání dat. Vytvořila jsem příkaz, který říká, že po kliknutí na

tlačítko se pole stane aktivním. A pokud akce proběhne na zařízení bez klávesnice, objeví se klávesnice, s jejíž pomocí se do pole zadá požadovaná hodnota.

```
document.addEventListener('DOMContentLoaded', function () {
  "use strict";

  const manualInputButton = document.getElementById("manual-input");
  const inputField = document.getElementById("cart-number");

  manualInputButton.addEventListener("click", () => {
    inputField.focus();
  });
});
```

#### 4.4.2.6 Implementace vychystávání objednávek

Dále jsem v systému vytvořila stránku, která slouží k vychystávání objednávek. Implementace této stránky spočívala ve vytvoření seznamu produktů pro sběr objednávek. Rozložení HTML vytvoří prostor pro zobrazení potřebných informací, vytvoří pole pro zadávání údajů a potřebná tlačítka.

Zobrazení informací na obrazovce je vytvořeno pomocí kódu PHP. To umožnilo, aby byl systém univerzální, závislý na databázi a změnách v ní, nikoli na kódu HTML.

```
<div class="photo container order"></div>
<div class="address container">Adresa: <?=$show_adress = $adresa['section'] . " " .
$adresa['street'] . " " . $adresa['shelf'] . " " . $adresa['box']; ?> </div>
<div class="product-name container">Název zboží: <?=$gd_name['gd_name']; ?> </div>
<div class="product-name container"> Výrobce: <?=$gd_name['producer']; ?> </div>
<div class="product-name container"> Množství: <?=$quantity['quantity']; ?> kus.</div>
  <div class="product-name container"><!--<?=$gd_name['gd_barcode']; ?>--> </div>

<form method="post">
  <label class="form-container">
    <input type="text" class="product-number" name="barcode"
placeholder="Naskenujte čárový kód">
```



```

</label>
<div class="buttons">
  <button>Není k dispozici</button>
  <button class="manual-input1">Ručně zadat kod</button>
  <button type="submit" name="verify">Ověřit</button>
</div>
</form>

```

Jako první krok při spouštění kódu PHP na této stránce jsem vytvořila příkazy, které lze použít k získání požadovaných informací z databáze.

První příkaz načte všechna data z tabulky „goods“ pomocí externího klíče „ts\_idgoods“ tabulky „task“.

```

1. $result_nazev=mysqli_query($connect, "SELECT * FROM goods WHERE
idgoods IN (SELECT ts_idgoods FROM task WHERE ts_idgoods = idgoods)");

```

Druhý příkaz načte všechny údaje potřebné k sestavení přesného umístění výrobku z tabulky „storehouse“ pomocí externího klíče „gd\_idstorehouse“ tabulky „goods“.

```

2. $result_adresa=mysqli_query($connect, "SELECT section, street, shelf, box
FROM storehouse WHERE idstorehouse IN (SELECT gd_idstorehouse FROM
goods WHERE gd_idstorehouse=idstorehouse)");

```

Třetí příkaz byl použit k získání požadovaného množství zboží z tabulky „task“ pomocí atributu „quantity“.

```

3. $result_quantity=mysqli_query($connect, "SELECT quantity FROM task ");

```

Všechny tyto dotazy jsem pak převedla na nové proměnné pomocí funkce `mysqli_fetch_assoc()`, která pomáhá získat konkrétní řádek dat z datového souboru, který jsem získala jako výsledek výše uvedených příkazů. Nyní jsem schopna zobrazit požadované údaje zadáním konkrétního potřebného array. Například toto vložení kódu PHP do kódu HTML mi umožňuje vypsat údaje o výrobci zboží: `<?= $gd_name['producer']; ?>`.

```

$adresa = mysqli_fetch_assoc($result_adresa);
$gd_name = mysqli_fetch_assoc($result_nazev);
$quantity = mysqli_fetch_assoc($result_quantity);

```

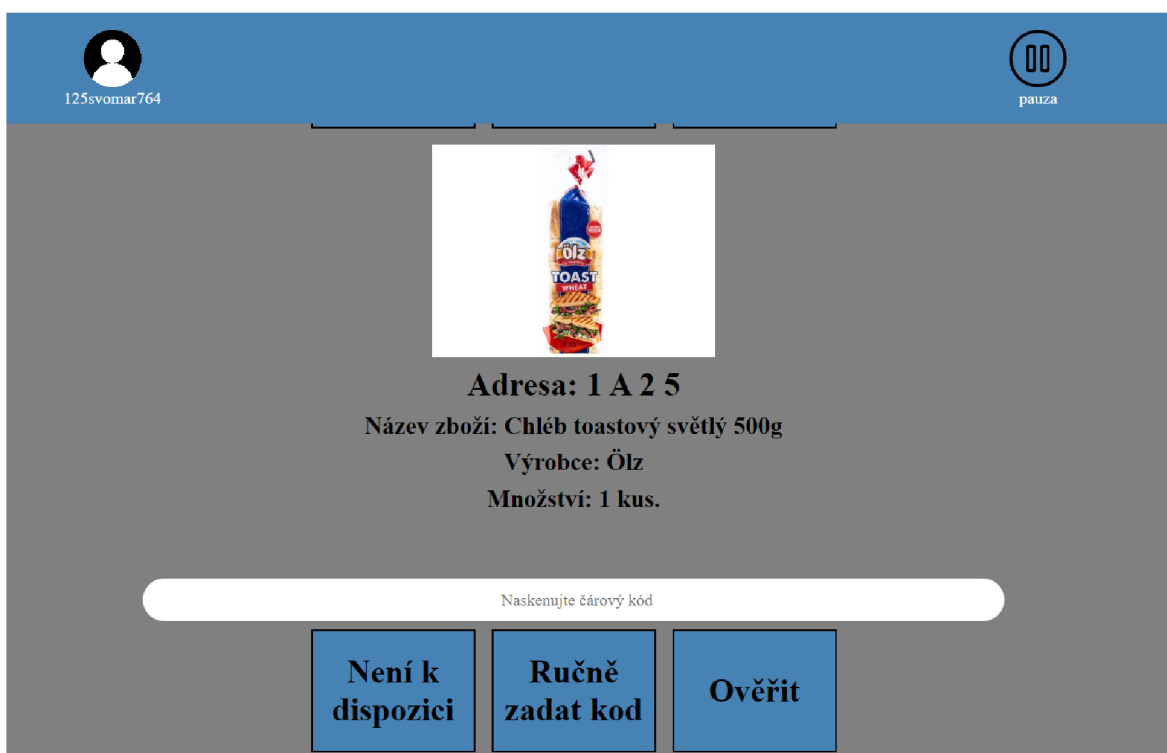
Dalším krokem bylo vytvoření cyklu, který by z databáze dostal všechna existující data a zkrátil kód HTML. Tento cyklus říká, že má provést kód HTML tolikrát, kolik dat je v zadaných proměnných. Tímto způsobem jsem byla schopna zobrazit všechna data na jedné obrazovce s neopakujícím se kódem.

```
while ($adresa && $gd_name && $quantity) {  
    ?>  
  
    HTML KOD  
  
    <?php  
    $adresa = mysqli_fetch_assoc($result_adresa);  
    $gd_name = mysqli_fetch_assoc($result_nazev);  
    $quantity = mysqli_fetch_assoc($result_quantity);  
  
    }  
    ?>
```

Dále jsem se při stisknutí tlačítka „Ověřit“ musela zkontrolovat, že zadaný čárový kód je v databázi a odpovídá čárovému kódu zobrazenému u každé položky v kódu HTML. Tento čárový kód jsem pro uživatele zneviditelnila <!--<?= \$gd\_name['gd\_barcode']; ?>--> pomocí komentovaného kódu PHP, ale jeho přítomnost v kódu je nutná pro následující kontrolu.



Obrázek 26: Screenshot dokončeného systému - vychystávání objednávek (1)



Obrázek 27: Screenshot dokončeného systému - vychystávání objednávek (2)

Po dokončení všech úkolů jsem na konec kódu umístila tlačítko, které přeneše na další stránku.

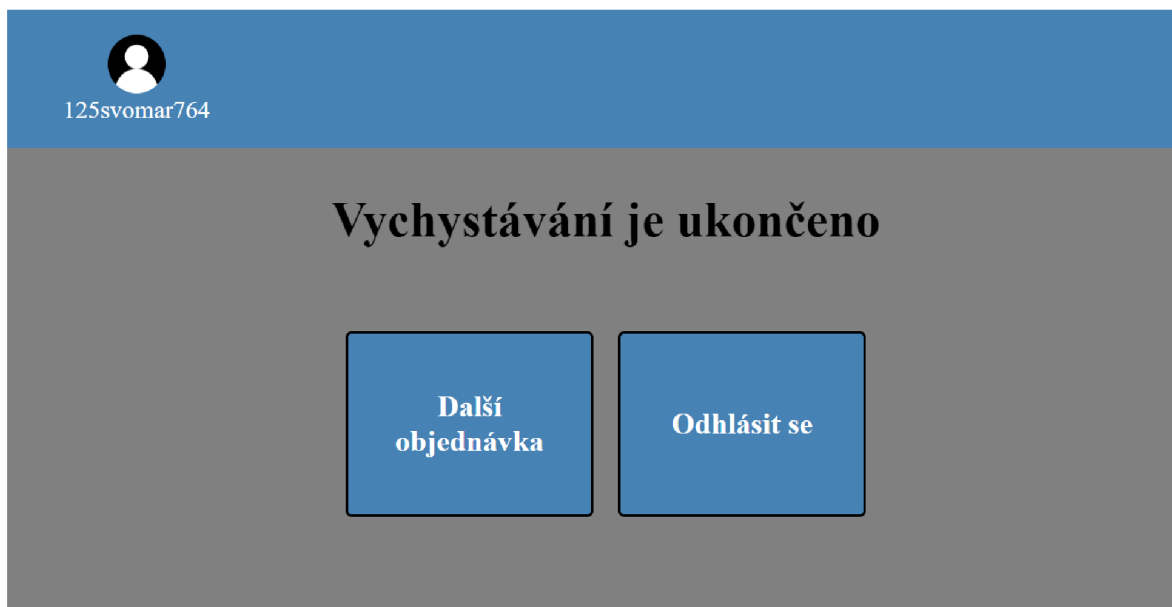
```
<button type="submit" name="continue" class="continue">Dále</button>
```

Přechod na další stránku se provádí pomocí kódu JavaScriptu.

```
const continueBtn = document.querySelector(".continue");
continueBtn.addEventListener("click", () => {
  window.location.href = "finish.php";
});
```

#### 4.4.2.7 Implementace poslední stránky systému

Poslední stránka obsahuje dvě tlačítka. Tlačítko „Další objednávka“ je zpracováno kódem JavaScript a odešle uživatele na stránku s výběrem sektoru. Tlačítko „Odhlásit se“ je zpracováno kódem JavaScriptu a odesílá uživatele na první stránku.



**Obrázek 28: Screenshot dokončeného systému - poslední stránka systému**

## 5 Zhodnocení výsledků

Vytvořený informační systém pomůže společnosti zlepšit kvalitu její práce. Systém může sloužit ke strukturování potřebných informací ve vytvořené databázi, která pomůže kontrolovat zboží a výkon zaměstnance ve skladu. Informační systém má jednoduchý design a nemá mnoho funkcí, což zvýší rychlost vychystávání objednávek a usnadní zaměstnancům orientaci v systému. Jeden velký seznam objednávek pomůže zaměstnanci lépe se orientovat, promyslet si trasu a vykonávat své funkce. Přihlašování pomocí loginu zvyšuje bezpečnost systému.

## 6 Závěr

V teoretické části bakalářské práce byl čtenář seznámen se základními pojmy, které jsou potřebné a důležité pro tvorbu informačního systému zabývajícího se vyhledáváním zboží a vychystáváním objednávek ve skladu. Byly vysvětleny pojmy sklad, typy prací, které se ve skladu vykonávají, a se zvláštní pozorností byl popsán proces vychystávání (v kapitole 3.1). Dále byly vysvětleny základní pojmy související s vývojem informačních systémů na webu: data, informace, architektury informačních systémů, web a webové stránky (v kapitolách 3.2 až 3.4). V dalších kapitolách byly představeny technologie, které se týkají vývoje kompletního, dynamického informačního systému a databáze: HTML, CSS, JavaScript, PHP, SQL (k nalezení v kapitolách 3.5.1 až 3.5.6). Bylo popsáno vývojové prostředí, ve kterém byl systém naprogramován, pro výsledny IS bylo využito PHPStorm (kapitola 3.5.7).

První část vlastní práce spočívá v provedení podrobné analýzy procesu vychystávání objednávek vytvořením workflow diagramu (v kapitole 4.1), byla provedena analýza stávajících problémů v procesu. Dale, prováděním průzkumu, byly získány požadavky na informační systém od uživatelů a bylo sestaveno možné řešení získaných požadavků (v kapitole 4.2). Druhá část vlastní práce je zaměřena na vytvoření návrhu informačního systému, který se drží na zjištěných poznatků z teoretické části. Byl proveden návrh architektury IS, návrh databáze vytvořením ER diagramu a grafický návrh uživatelského rozhraní (v kapitole 4.3). Dále jsem začala vývoj struktury databáze a systému samotného. V práci jsem uvedla důležité části implementace systému a vložila jsem do toho zdrojový kód jazyků HTML, JavaScript a PHP a podrobně popsala funkce, které jednotlivé části kódu, prezentovaného v této bakalářské práci, plní v systému (v kapitole 4.4).

## 7 Seznam použitých zdrojů

- [1] GROSS IVAN a kolektiv. Velká kniha logistiky. Praha: Vysoká škola chemicko-technologická v Praze , 2016. ISBN 978-80-7080-952-5.
- [2] LADÝŘ MICHAL. Čím se sklad stává opravdu řízeným?: WMS je základním předpokladem efektivního řízení skladu. IT Systems, Roč. 20, č. 10 (2018), s. 16-19 ISSN 1802-002X.
- [3] ZAHÁLKA LUDĚK. Warehouse management : Může být WMS konkurenční výhodou? IT Systems , Roč. 11, č. 9 (září 2009), s. 30-31. ISSN 1802-002X.
- [4] SKLADON. Fulfillment: Metody vychystávání, jejich (ne)výhody a doporučení. [online]. Dostupné: <https://skladon.cz/cs/blog/fulfillment-metody-vychystavani-jejich-ne-vyhody-a-doporuceni>
- [5] PŘÍHODA JAN. Jak efektivně řídit sklad? IT Systems , Roč. 17, č. 5 (2015), s. 10-11. ISSN 1802-002X.
- [6] MOLNÁR ZDENĚK. Podnikové informační systémy. Vyd. 2., přeprac. Praha: České vysoké učení technické, 2009. ISBN 978-80-01-04380-6.
- [7] ŠILEROVÁ EDITA a HENNYEYOVÁ KLÁRA. Informační systémy v podnikové praxi. Druhé vydání. Praha: Powerprint, 2017. ISBN 978-80-7568-065-5.
- [8] SODOMKA PETR, KLČOVÁ HANA. Informační systémy v podnikové praxi. Brno: Computer Press, 2010 ISBN: 978-80-251-2878-7.
- [9] BRUCKNER, VOŘÍŠEK, BUCHALCEVOVÁ. Tvorba informačních systémů. Principy, metodiky, architektury. Praha: Grada, 2012. ISBN 978-80-247-4153-6.
- [10] CASTRO ELIZABETH. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2022. ISBN 978-80-251-5045-0.
- [11] THAU DAVE. Velký průvodce JavaScriptem: tvorba interaktivních webových stránek v praxi. Praha: Grada, 2009. ISBN 978-80-247-2211-5.

- [12] LEISS OLIVER. PHP v praxi : pro začátečníky a mírně pokročilé, Praha: Grada, 2010. ISBN 978-80-247-3060-8.
- [13] STEPHENS RYAN K. Naučte se SQL za 28 dní. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.



## Seznam obrázků a zkratk

### 7.1 Seznam obrázků

Obrázek 1: Ilustrace procesu "Vychystávání po 1 objednávce" [4].....	16
Obrázek 2: Ilustrace procesu "Vychystávání po objednávkových dávkách" [4] .....	17
Obrázek 3: ; Ilustrace procesu "Zónové vychystávání" [4].....	17
Obrázek 4: Čárový kód a kontrolní číslo [5] .....	18
Obrázek 5: Vzorek kódu HTML .....	23
Obrázek 6: Vzorek kódu CSS .....	24
Obrázek 7: Vzorek kódu JavaScript .....	24
Obrázek 8: Workflow diagram procesu vychystávání objednávek.....	27
Obrázek 9: Entity relationship diagram.....	31
Obrázek 10: Grafický návrh - Přihlašovací stránka.....	32
Obrázek 11: Grafický návrh - Výběr sektoru .....	33
Obrázek 12: Grafický návrh - Kontrola výběru sektoru.....	34
Obrázek 13: Grafický návrh - Zadání čísla košíku.....	35
Obrázek 14: Grafický návrh - Vychystávání objednávek .....	36
Obrázek 15: Grafický návrh - Poslední stránka systému .....	37
Obrázek 16: Naplnění tabulky "storehouse" .....	38
Obrázek 17: Naplnění tabulky "goods" .....	38
Obrázek 18: Naplnění tabulky "worker" .....	38
Obrázek 19: Naplnění tabulky "task" .....	39
Obrázek 20: Naplnění tabulky "mobile" .....	39
Obrázek 21: Naplnění tabulky "cart" .....	39
Obrázek 22: Screenshot dokončeného systému - přihlášení uživatele .....	41
Obrázek 23: Screenshot dokončeného systému - výběr sektoru.....	44
Obrázek 24: Screenshot dokončeného systému - kontrola výběru sektoru .....	45
Obrázek 25: Screenshot dokončeného systému - zadání čísla košíku a kontrola .....	47
Obrázek 26: Screenshot dokončeného systému - vychystávání objednávek (1).....	51
Obrázek 27: Screenshot dokončeného systému - vychystávání objednávek (2).....	51
Obrázek 28: Screenshot dokončeného systému - poslední stránka systému .....	52

### 7.2 Seznam použitých zkratk

**IS** - Informační systém.

**HTML** - Hypertext Markup Language.

**CSS** - Cascading Style Sheets.

**PHP** - Hypertext Preprocessor, původně Personal Home Page.

**WEB** - World Wide Web.

**W3C** - World Wide Web Consortium.

**URL** - Uniform Resource Locator.

**HTTP** - Hypertext Transfer Protocol.

**MySQL/ SQL** - My Structured Query Language.

**MMDIS** - Multidimensional Management and Development of Information Systems

**ERD** - Entity relationship diagram

**IASW** – Individuální aplikační software

**TASW** - Typový aplikační software

## **Přílohy**

Do UIS jsem nahrála všechny přílohy se zdrojovým kódem.