

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Návrh a implementace podnikové webové aplikace

Vojtěch Martínek

© 2018 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vojtěch Martínek

Informatika

Název práce

Návrh a implementace podnikové webové aplikace

Název anglicky

Design and implementation of enterprise web application

Cíle práce

Hlavním cílem práce je analýza již existujících webových aplikací pro řízení skladových evidencí a zhodnocení výhod a nevýhod jednotlivých řešení. Z těchto poznatků bude navrhnutá a implementována jednoduchá webová aplikace, která převeze důležité funkce z komplexních systémů, ale bude snadno použitelná pro všechny menší podniky.

Metodika

Nastudujte informační zdroje a dle doporučených metod softwarového inženýrství proveďte analýzu uživatelských požadavků, návrh (konceptuální model), implementaci a řádné otestování webové aplikace pro podporu vybraných podnikových procesů ve zvolené organizaci. Na základě analýzy teoretických poznatků a výsledků praktické části zformulujte závěry bakalářské práce.

Doporučený rozsah práce

30-40 stran A4

Klíčová slova

webová aplikace, skladová evidence, podnikový informační systém, logistika, softwarové inženýrství

Doporučené zdroje informací

BAŠE, Ondřej. JQuery pro neprogramátory: průvodce využitím knihovny jQuery UI. Brno: Computer Press, 2012. ISBN 978-80-251-3750-5.

CASTAGNETTO, J. PHP : *programujeme profesionálně*. Praha: Computer Press, 2001. ISBN 80-7226-310-2.

CASTRO, E. – HYSLOP, B. *HTML5 a CSS3 : názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.



Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

Ing. David Buchtela, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 13. 3. 2018

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 13. 3. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 15. 03. 2018

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci na téma **Návrh a implementace podnikové webové aplikace** jsem vypracoval samostatně, pod vedením vedoucího bakalářské práce a za použití informačních pramenů, které cituji v seznamu použitých zdrojů na konci práce. Prohlašuji, že při vytváření práce jsem čerpal pouze z uvedených zdrojů a že jsem neporušil autorská práva třetích osob.

V Praze dne 15.3.2018

Vojtěch Martínek

Poděkování

Rád bych poděkoval všem, kteří se podíleli na zpracování této bakalářské práce, zejména vedoucímu práce Ing. Davidu Buchtelovi, Ph.D., za jeho cenné připomínky a rady. Chtěl bych poděkovat zejména za vymezení hranic v praktické části bakalářské práce a díky tomu k úspěšnému dokončení včetně plně funkční praktické části.

Návrh a implementace podnikové webové aplikace

Souhrn:

Bakalářská práce se zabývá problematikou evidence skladových zásob, zejména u podniků, které mají více prodejních kanálů. Takové podniky nemohou využít některé vestavěné skladové evidence, která poskytují například e-shopová řešení, jelikož by byl obsloužen pouze jeden prodejní kanál. Hlavním cílem práce je analyzovat současná možná řešení decentralizovaných skladových evidencí, sběr požadavků na systém a zjištění hlavních funkcionalit. V praktické části je následně navrženo a vyvíjeno vlastní řešení ve formě webové aplikace. Výsledná aplikace umožňuje vést decentralizovanou skladovou evidenci za pomoci webového prohlížeče nezávisle na konkrétním zařízení.

Klíčová slova:

webová aplikace, vývoj aplikace, skladový systém, decentralizovaný sklad, logistické řešení

Design and implementation of enterprise web application

Summary:

The bachelor thesis deals with main warehouse problem for business which has more than one sales channel. It is not possible to use built-in warehouse solution which are incorporated within e-shop solutions, but they need to use decentralized warehouse system. The main goal of the bachelor thesis is the analysis of existing solutions, collecting system requirements and identifying main functionalities of these systems. In second part of thesis online web application is designed and developed. Web application should allow to handle warehouse records from each connection point independent of the particular device.

Key words:

web application, application development, warehouse system, decentralized warehouse, logistics solution

Obsah

| | | |
|-------------|------------------------------------------|-----------|
| 1 | Úvod | 14 |
| 2 | Cíl a metodika práce..... | 15 |
| 2.1 | Cíl práce..... | 15 |
| 2.2 | Metodika práce | 15 |
| 3 | Teoretická východiska..... | 16 |
| 3.1 | Vymezení pojmů..... | 16 |
| 3.1.1 | Webová aplikace..... | 16 |
| 3.1.2 | Webový server | 16 |
| 3.1.3 | Relační databáze | 16 |
| 3.1.4 | CSS framework..... | 17 |
| 3.1.5 | PHP framework..... | 18 |
| 3.2 | Životní cyklus vývoje softwaru | 19 |
| 3.2.1 | Etapy vývoje | 19 |
| 3.2.1.1 | Analýza a specifikace | 19 |
| 3.2.1.2 | Návrh | 19 |
| 3.2.1.3 | Implementace..... | 20 |
| 3.2.1.4 | Testování..... | 20 |
| 3.2.1.5 | Zavedení do provozu | 20 |
| 3.2.1.6 | Provoz a údržba | 20 |
| 3.2.2 | Základní modely | 21 |
| 3.2.2.1 | Vodopádový model..... | 21 |
| 3.2.2.2 | Iterativní a inkrementální metodiky..... | 22 |
| 3.2.2.2.1 | Příklady agilních metodik | 22 |
| 3.2.2.2.1.1 | Extrémní programování | 22 |

| | | |
|-------------|----------------------------------------------------|-----------|
| 3.2.2.2.1.2 | Scrum | 23 |
| 3.2.2.3 | RUP – Rational Unified Process..... | 23 |
| 3.3 | Skladová evidence | 24 |
| 3.3.1 | Situace na trhu | 24 |
| 3.3.1.1 | Altus Vario..... | 24 |
| 3.3.1.1.1 | Možnosti skladové evidence | 25 |
| 3.3.1.1.2 | Výhody řešení | 25 |
| 3.3.1.1.3 | Nevýhody řešení..... | 25 |
| 3.3.1.1.4 | Cena řešení | 25 |
| 3.3.1.2 | ABRA FlexiBee..... | 26 |
| 3.3.1.2.1 | Možnosti skladové evidence | 26 |
| 3.3.1.2.2 | Výhody řešení | 26 |
| 3.3.1.2.3 | Nevýhody řešení..... | 27 |
| 3.3.1.2.4 | Cena řešení | 27 |
| 3.3.1.3 | Instantní sklad | 28 |
| 3.3.1.3.1 | Výhody řešení | 28 |
| 3.3.1.3.2 | Nevýhody řešení..... | 28 |
| 3.3.2 | Zhodnocení výsledků..... | 28 |
| 4 | Návrh vlastního řešení..... | 30 |
| 4.1 | Vzorové osoby | 30 |
| 4.2 | Vymezení funkcionalit..... | 33 |
| 4.2.1 | Prizpůsobení aplikace pro svůj obor podnikání..... | 33 |
| 4.2.2 | Základní přehled | 33 |
| 4.2.3 | Přidávání produktů..... | 33 |
| 4.2.4 | Editace produktů | 33 |

| | | |
|--------|-----------------------------------------------------|----|
| 4.2.5 | Režim vyskladňování..... | 34 |
| 4.2.6 | Historie skladu | 34 |
| 4.2.7 | Inventura | 34 |
| 4.3 | Logické rozdělení aplikace | 35 |
| 4.4 | Vytvoření sitemapy..... | 35 |
| 4.5 | Logický design (wireframy) | 38 |
| 4.5.1 | Úvodní strana webu | 38 |
| 4.5.2 | Vytvoření nového účtu..... | 39 |
| 4.5.3 | Přihlášení do aplikace | 40 |
| 4.5.4 | Průvodce nastavením nového uživatele – 1. krok..... | 41 |
| 4.5.5 | Průvodce nastavením nového uživatele – 2. krok..... | 41 |
| 4.5.6 | Průvodce nastavením nového uživatele – 3. krok..... | 42 |
| 4.5.7 | Úvodní stránka aplikace – základní přehled | 43 |
| 4.5.8 | Přidání nového produktu..... | 44 |
| 4.5.9 | Režim vyskladňování..... | 45 |
| 4.5.10 | Inventura | 45 |
| 4.5.11 | Historie skladu | 46 |
| 4.5.12 | Nastavení aplikace | 47 |
| 4.6 | Metodika řešení..... | 47 |
| 4.6.1 | Nette Framework | 48 |
| 4.6.2 | Bootstrap..... | 49 |
| 4.6.3 | jQuery | 49 |
| 4.6.4 | MySQL | 49 |
| 4.7 | Implementace..... | 50 |
| 4.7.1 | Prvotní nastavení..... | 50 |

| | | |
|----------|------------------------------------------------------------------------------|-----------|
| 4.7.2 | Návrh databáze | 51 |
| 4.7.3 | Rozdělení aplikace do modulů..... | 52 |
| 4.7.4 | Routování URL adres | 53 |
| 4.7.4.1 | Routování obecně | 53 |
| 4.7.4.2 | Routování URL adres do jednotlivých modulů | 53 |
| 4.7.5 | Ukázkový kód pro získání produktu dle ID..... | 54 |
| 4.7.6 | Ukázkový kód pro aktualizování množství produktů po provedené inventuře..... | 55 |
| 4.7.7 | Testování aplikace | 56 |
| 4.7.8 | Spuštění aplikace v produkčním prostředí..... | 57 |
| 5 | Závěr | 58 |
| 6 | Seznam použitých zdrojů | 59 |
| 7 | Přílohy..... | 60 |
| 7.1 | Příloha A Vytvořená aplikace – úvodní strana | 61 |
| 7.2 | Příloha B Vytvořená aplikace – zobrazení produktů | 62 |
| 7.3 | Příloha C Vytvořená aplikace – režim inventury..... | 63 |
| 7.4 | Příloha D Vytvořená aplikace – nastavení..... | 64 |
| 7.5 | Příloha E CD se zdrojovými kódy | 65 |

Seznam obrázků

| | |
|-----------------------------------------------------------------------------|----|
| Obrázek 1 - Vodopádový model..... | 22 |
| Obrázek 2 - Agilní model vývoje | 23 |
| Obrázek 3 - Sitemapa webové aplikace..... | 37 |
| Obrázek 4 - Wireframe úvodní obrazovky | 38 |
| Obrázek 5 - Wireframe pro vytvoření nového účtu..... | 39 |
| Obrázek 6 - Wireframe pro přihlášení do aplikace..... | 40 |
| Obrázek 7 - Wireframe pro průvodce nastavením - 1. krok | 41 |
| Obrázek 8 - Wireframe pro průvodce nastavením - 2. krok | 41 |
| Obrázek 9 - Wireframe pro základní přehled | 43 |
| Obrázek 10 - Wireframe pro přidání nového produktu | 44 |
| Obrázek 11 - Wireframe pro režim vyskladňování | 45 |
| Obrázek 12 - Wireframe pro inventuru..... | 45 |
| Obrázek 13 - Wireframe pro historii skladu | 46 |
| Obrázek 14 - Wireframe pro nastavení aplikace | 47 |
| Obrázek 15 - Životní cyklus stránky navrhnuté pomocí Nette Frameworku | 48 |
| Obrázek 16 - Grafický návrh databáze | 51 |
| Obrázek 17 - Upravená struktura aplikace do modulů | 52 |

Seznam tabulek

| | |
|-------------------------------------|----|
| Tabulka 1 - Vzorová persona 1 | 31 |
| Tabulka 2 - Vzorová persona 2..... | 31 |
| Tabulka 3 - Vzorová persona 3..... | 32 |

Seznam ukázkových kódů

| | |
|------------------------------------------------------------------------------|----|
| Zdrojový kód 1 – Přesměrování domovské stránky v souboru .htaccess | 50 |
| Zdrojový kód 2 - princip routování URL adres | 53 |
| Zdrojový kód 3 - Routování URL adres do odpovědných modulů | 54 |
| Zdrojový kód 4 - Ukázka metody, která vrací produkt dle jeho ID..... | 55 |
| Zdrojový kód 5 - Aktualizování množství produktů po provedené inventuře..... | 56 |

1 Úvod

Cílem každého podniku je tvorba zisku, který je generován prodejem produktu. Produkt je základní stavební jednotkou, na který navazuje celá řada oborů - od výroby produktu, jeho skladování, propagace, až po jeho distribuci k odběratelům. Optimalizováním každého z uvedených procesů se dospělo k závěru, že o každém produktu je potřeba evidovat informace o jeho stavu, množství, váze, variantách, dostupnosti, pozice ve skladu, ceně a mnoho dalších.

Řada vývojářských firem začala problematiku skladových evidencí řešit vývojem vlastních aplikací, které jsou vyvíjeny buď na zakázku pro jeden konkrétní podnik, nebo jako univerzální řešení pro podniky ve všech oborech. Výhodou aplikací na zakázku je dokonalé přizpůsobení aplikace potřebám zákazníka. Naopak hlavní nevýhodou se může jevit vysoká cena. Univerzální aplikace jsou většinou tvořeny pomocí modulů, které si zákazník přikupuje dle své potřeby. Výhodou je nižší cena, než u aplikací na zakázku, poskytování technické podpory a pravidelná údržba aplikace.

Tato práce je složena ze dvou částí. V teoretické části bude provedena analýza již existujících řešení, vyhodnocení výhod a nevýhod jednotlivých řešení. V praktické části bude navrhnutá a vyvíjena vlastní webová aplikace, která bude dostupná online, nezávisle na konkrétním zařízení.

2 Cíl a metodika práce

2.1 Cíl práce

Hlavním cílem bakalářské práce je odhalení nedostatků již existujících řešení, na jehož základě bude vytvořena přehledná webová aplikace, která bude umožňovat decentralizovaně vést skladovou evidenci zásob.

Dalším z cílů této práce je navrhnutí aplikace takovým způsobem, aby ji mohly využívat subjekty z libovolných oborů. Skladová evidence by měla být přístupná ihned po úspěšné registraci na úvodní stránce projektu.

2.2 Metodika práce

Pro dosažení cílů bude využito odborných informačních zdrojů, které dle doporučených metod softwarového inženýrství povedou k návrhu a implementaci webové aplikace. Aplikace bude následně implementována na webhosting a řádně otestována reálnými produkty a základními operacemi s nimi. Po otestování budou navrhnuty možné úpravy a vylepšení, které skladovou evidenci budou do budoucna optimalizovat a podnikům tak umožní ještě efektivnější řízení zásob.

3 Teoretická východiska

3.1 Vymezení pojmů

3.1.1 Webová aplikace

Pod webovou aplikací si lze představit program, který lze spustit z libovolného zařízení pomocí webového prohlížeče bez nutnosti aplikaci instalovat. Webový prohlížeč působí v roli klienta, kterému webový server dynamicky zasílá webovou stránku k promítnutí.

Hlavní rozdíl mezi webovou stránkou a webovou aplikací spočívá v tom, že webová aplikace bývá zpravidla složitější a aktivně využívá databázi k načítání i ukládání dat. Postupný vývoj internetu zapříčinil rapidní vývoj webových aplikací, které v současné době svými funkcemi konkurují standardním aplikacím, které se musí instalovat. (1)

3.1.2 Webový server

Webový server lze definovat jako „program, který nabízí své služby jiným programům. Když si chceme prohlížet určitou webovou stránku, vyžádá si ji náš prohlížeč (ten zde vystupuje v roli klienta) od serveru.“ (2)

„Webový server se spoustou dalších služeb samozřejmě nabízejí všichni poskytovatelé hostingových služeb, a pokud provozujete nějaké vlastní webové stránky, jistě už jste jejich služby použili. Některé tyto služby jsou k dispozici zdarma; za jiné se platí, přičemž přidanou hodnotou bývá obvykle zvýšená dostupnost a technická podpora (3, s. 14)“ .

3.1.3 Relační databáze

Databáze představuje místo, kde je možné uchovávat potřebná data. Typickým příkladem databáze může být například knihovna nebo kartotéka. Pokud známe způsob uložení dat v těchto systémech, můžeme v nich efektivně vyhledávat potřebné informace.

Určitém podtypem databází, které se hojně využívají při tvorbě aplikací, jsou relační databáze. Základním prvkem relačních databází jsou databázové tabulky, které mezi sebou mají logické vazby neboli relace. Každá takto vytvořená tabulka je tvořena řádky a sloupci, přičemž sloupce představují vlastnosti, které se u prvků evidují a řádky představují jednotlivé záznamy. Každý řádek poté může představovat jednoho konkrétního žáka, jeden konkrétní nahraný obrázek či přidáný produkt.

Aby bylo možné jednotlivé řádky mezi sebou rozpoznat, je důležité mít definovaný jedinečný identifikátor, tzv. primární klíč. „Obvykle se jedná o číselné řady s tím, že každý záznam dostává číslo o jednotku vyšší než ten předchozí. Při návrhu tabulky lze pro tuto funkcionalitu použít parametr AUTO INCREMENT, který zajistí automatické číslování. (4)“

Pro vzájemné propojování relačních databází mezi sebou se využívají tzv. cizí klíče. Cizí klíč u konkrétního záznamu poté představuje primární klíč záznamu z propojené tabulky.

3.1.4 CSS framework

Obecnou definicí frameworku (ať už CSS frameworku nebo PHP frameworku) jsou soubory tříd, metod či celých objektů, které mají za úkol zjednodušit opakující se práci vývojářů. Úspora času vývojářů významně snižuje náklady, které vyvolává vývoj aplikací.

Vzhled webových aplikací se vždy upravoval a upravuje pomocí CSS (Cascading Style Sheets). CSS soubory představují určitý soubor vlastností a hodnot, které jsou přiděleny prvkům HTML souboru. Vlastnosti CSS souborů umožňují upravovat velikost či font písma, jeho stínování, pozici celého prvku či například jeho pozadí (5, s. 410-424).

Při vývoji rozsáhlejších projektů se zjistilo, že CSS styly se pro určité prvky opakují na každé webové stránce a je potřeba řešit stále stejné problémy. Neustálé zapisování vlastností a hodnot pro formátování prvků vedlo k vývoji prvních CSS frameworků, které obsahují již předdefinované CSS vlastnosti prvků. Prvkům se pouze přiřadí odpovědná třída (class) a díky tomu jsou na ně aplikovány vlastnosti a hodnoty, které CSS framework obsahuje. Velké problémy nastaly při tvorbě responzivních aplikací, tedy webových aplikací, které přizpůsobují svůj vzhled dle konkrétního zařízení při zachování přehlednosti

všech prvků na webové stránce. Pokud si webovou aplikaci tedy otevřeme například v prohlížeči mobilního telefonu, prvky již nejsou poskládány do šířky vedle sebe, ale pod sebe, což vede k vyššímu uživatelskému komfortu při procházení webovou aplikací.

Na problém při návrhu responzivních designů CSS frameworky reagovaly vytvořením tzv. grid systémů. Pod pojmem grid systém si můžeme představit mřížku typicky 12 sloupců, které představují 100% šířku zařízení. Například nejznámější CSS framework Bootstrap dále rozlišuje čtyři různé zařízení pomocí CSS tříd, a to na xs – pro mobilní telefony, sm – pro tablety, md – pro malé notebooky a lg pro velké notebooky a stolní počítače.

CSS frameworky pomocí předdefinovaných tříd velmi urychlují vývoj designu webových aplikací, umožňují efektivně vytvářet responzivní design a zachovávat tak uživatelsky přívětivý design aplikace. Mezi nejznámější CSS Frameworky patří Bootstrap na druhém místě poté Pure či Gumby.

3.1.5 PHP framework

PHP (Hypertext Preprocessor) je jedním z nejznámějších programovacích jazyků, který slouží pro dynamické generování webových aplikací na serveru a zaslání vygenerované stránky pro klienta, tedy webový prohlížeč (6, s. 6).

Od svého vzniku 8. června 1995 prošel celou řadou úprav a náročným vývojem. V současné době je nejaktuálnější verzí PHP 7.2.0., která byla vydána 30. listopadu 2017. Stejně jako u CSS, i v PHP se stále více opakovaly problémy, které bylo potřeba řešit u každého nového projektu. Díky této skutečnosti začaly vznikat první PHP frameworky, které představují soubor metod, tříd a doporučených principů.

PHP frameworky umožňují rychlejší vývoj webových aplikací, zachovávají znovupoužitelnost kódu, dodržují MVC¹ model a minimalizují bezpečnostní rizika

¹ MVC (Model View Controller) – model obsluhuje získávání a zpracování dat z databáze, view představuje šablonu, tedy html kód, který je uživateli zobrazen. Controller je mezivrstva mezi

webových aplikací. Mezi nejznámější PHP frameworky patří celosvětově Laravel, Symfony nebo český Nette framework.

3.2 Životní cyklus vývoje softwaru

3.2.1 Etapy vývoje

Vývoj softwaru se skládá z činností, které na sebe navzájem navazují. Jednotlivé činnosti nelze podložit fakty, stejně jako ve fyzice nebo chemii, ale vychází se ze zkušeností, které se během několika let vývoje staly standardem (tzv. best practices – osvědčené postupy).

3.2.1.1 Analýza a specifikace

Prvním krokem, který předchází samotnému vývoji softwaru, je analýza a specifikace požadavků od zákazníka. Je důležité co nejpodrobněji specifikovat, jaké jsou kladeny požadavky na software, zda je možné za daných podmínek (převážně finančních a časových) aplikaci vytvořit. Přesné vymezení požadavků na software zamezují změnám zadání a tím se minimalizuje riziko, že zákazník nebude s výsledkem spokojen.

3.2.1.2 Návrh

Pokud máme jasně definované cíle a požadavky na vyvíjenou aplikaci, následuje její návrh. Během návrhu dochází k logickému rozčlenění aplikace do modulů, zamyšlení se nad datovou strukturou a postupné zpodrobnění jednotlivých funkcí. Výsledkem návrhu je cenový odhad aplikace, jejich nároků na lidské zdroje a čas. Výstupem by také měly být podrobnější testy jednotlivých modulů. (7)

modelem a šablonou, zajišťuje, zda je uživatel oprávněn stránku zobrazit, získává data z modelu a předává je k vykreslení do šablony.

3.2.1.3 Implementace

Implementací se rozumí proces, který zahrnuje převedení myšlenek a návrhů na fungující systém. Je nezbytné, aby nově vytvořený systém měl implementovány všechny stanovené požadavky. Během implementace je navrhována dokumentace a popis řešení, které je v budoucnu nezbytnou součástí každého systému. Dokumentace musí být vždy aktuální ke zdrojovému kódu, který je již v produkci, aby se předešlo problémům, které by při následném udržování a rozšiřování kódu mohly nastat.

3.2.1.4 Testování

Po sloučení všech modulů se přechází na testování aplikace. Během testování jsou vytvářeny unitové testy, testovací skripty na relační databázi a vytváření nástrojů pro automatické testování funkcí. Pokud během testování dojde k chybě aplikace, zpravidla se vrací do předchozí etapy, kde se problém identifikuje, zjistí se příčina a dochází k opravě. Po opravě se opět přechází do etapy testování.

3.2.1.5 Zavedení do provozu

Během zavádění aplikace dochází k instalaci na server či konkrétní zařízení. V případě webové aplikace dochází k přesunutí aplikace na webhosting, kde se webová aplikace stává přístupnou ve světě internetu. Spolu s přesunutím aplikace na webhosting také migruje relační databáze na databázový server.

Zpravidla v prvních týdnech po zprovoznění aplikace vývojový tým přechází do režimu hypercare, tedy stavu neustálého monitorování a hlídání stavu systému. Během tohoto režimu jsou jakékoliv chyby aplikace okamžitě zachyceny a předávány k opravě.

3.2.1.6 Provoz a údržba

Aplikaci v produkčním prostředí je potřeba monitorovat, zachytávat chybové hlášky a reagovat na zpětnou vazbu uživatelů. K tomuto účelu se používají pravidelné kontroly, které jsou vývojářskou firmou poskytovány. Dále také může docházet ke sledování potřeb

uživatelů a rozvíjení funkčnosti aplikace, aby aplikace uspokojila co nejvíce potenciálních uživatelů.

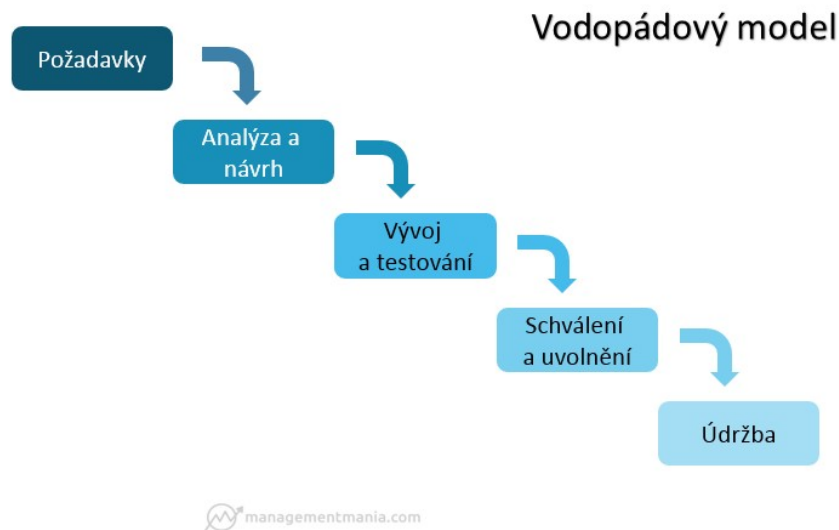
3.2.2 Základní modely

3.2.2.1 Vodopádový model

Vodopádový model vychází z předpokladu, že jednotlivé etapy vývoje na sebe vzájemně navazují. Jakmile dojde k dokončení první etapy, začne etapa druhá. Proces vodopádového modelu začíná důkladnou analýzou potřeb zákazníka. Pokud je analýza potřeb provedena důkladně, může následující etapy urychlit. Pokud se kritický problém odhalí již během analýzy požadavků nebo návrhu, dojde k úspoře nemalé finanční částky za vývoj aplikace. Pokud by naopak analýza systému neodhalila problém již v začátcích, ale během samotného vývoje, dochází k návratu do první etapy, a to je v současné době velmi finančně nákladné.

Hlavní výhodou vodopádového modelu je fakt, že pokud bude první etapa analýzy provedena důkladně, předejde se problémům, které by se jinak odhalily až během implementace. Autor knih o softwarovém inženýrství Steve McConnell odhaduje, že odstranění chyby v požadavcích, kterou se nepodaří odhalit až do fáze implementace nebo údržby, stojí 50krát až 200krát více, než kdyby se taková chyba odhalila a napravila již v etapě specifikace požadavků (8, s. 465).

Nevýhodou vodopádového modelu je nutnost přepracování celého modelu v případě, že zákazník změní během vývoje své požadavky. Stejně tak pokud je aplikace již dokončena a uživatel žádá změnu funkce aplikace, je potřeba přejít do první etapy vývoje a pojit celým procesem vývoje od začátku.



Obrázek 1 - Vodopádový model

Zdroj: <https://managementmania.com/cs/vodopadovy-model-waterfall-model>

3.2.2.2 Iterativní a inkrementální metodiky

Iterativní metodiky jsou oproti vodopádovému modelu založeny na časté diskuzi se zákazníkem. Konkrétně při navrhování a vývoji systému se hovoří o agilních metodikách. Díky neustálé komunikaci se zákazníkem agilní metodika umožňuje téměř okamžitou reakci na změnu požadavků a ubírání vývoje systému jiným směrem.

3.2.2.2.1 Příklady agilních metodik

3.2.2.2.1.1 Extrémní programování

Jedná se o metodiku programování v krátkých cyklech, kde se navrhuje jednotkové testy², píše se krátký a jednoduchý kód a provádí se neustálý refaktoring³.

² Jednotkové testy (unit testy) – představují samostatně testovatelnou část programu

³ Refaktoring – přepsání zdrojového kódu na čistější kód, vede ke zvýšení přehlednosti a budoucího rozšíření aplikace. Vnější chování kódu zůstává nezměněn.

3.2.2.2.1.2 Scrum

Technika, při které dochází ke každodenním krátkým setkáním týmu, kde každý člen prezentuje své činnosti z minulého dne, baví se o probíhajících činnostech a o činnostech budoucích. Jednotlivá setkání se seskupují do sprintů, které představují období dlouhé 1-4 týdny, po kterých je prezentován postup vývoje zákazníkovi, sponzorovi nebo investorům (9).



Obrázek 2 - Agilní model vývoje

Zdroj: <https://www.congesta.cz/Services/WebProgramming>

3.2.2.3 RUP – Rational Unified Process

Metodika RUP vznikla na základě nejlepších praktik softwarového vývoje. Metodika se skládá z iterativního vývoje, řízení požadavků (požadavky se mění), použití komponentové architektury, vizuálního modelování (UML), ověřování kvality software a řízení změn.

Metodika vychází z UML⁴ a je koncipována do čtyř etap (zahájení, projektování, realizace, předání). Model kvůli své složitější konstrukci nalezne využití zejména u velkých, komerčních projektů (10).

⁴ UML – jazyk pro vizualizaci, specifikaci a dokumentaci systémů.

3.3 Skladová evidence

Každý podnik, který se zabývá výrobou, distribucí či prodejem, dříve nebo později zjistí, že je potřeba vést evidenci svých produktů. Není prakticky možné, aby podnik nevedl skladovou evidenci, protože by se mohlo lehce stát, že nabídne koncovému zákazníkovi nebo odběrateli produkt, který dle představ měl být skladem, nicméně skladem již není. Taková společnost by v dnešní době na trhu nejspíše neobstála. Z tohoto důvodu téměř každý podnik takovou evidenci vede, aby měl přehled o každém konkrétním produktu a podnik mohl být na trhu o to obratnější. 21. století přímo vybízí k užití informačních technologií pro tvorbu takovéto evidence.

3.3.1 Situace na trhu

V současné době existuje velké množství systémů pro řízení skladové evidence. V první řadě se jedná o programy, které pracují v tzv. offline režimu. Takové systémy jsou dostupné pouze na zařízení, kde je aplikace nainstalována. Dalším řešením může být aplikace, kterou podnik provozuje na vlastním serveru či na cloudových řešeních. V neposlední řadě je možné provozovat evidenci jako webovou aplikaci. V principu řízení z libovolného zařízení vítězí právě poslední zmiňované řešení.

3.3.1.1 Altus Vario

Společnost Altus Software s.r.o., konkrétně produkt Altus Vario, patří k největším hráčům na trhu. Skladová evidence je jeden z mnoha modulů, které si zákazník může pořídit do své společnosti. K produktu Altus Vario je možné pořídit také moduly pro evidenci zákazníků, fakturaci, účetnictví, pro řízení mezd, pro provoz pokladny a mnohé další.

3.3.1.1.1 Možnosti skladové evidence

Hlavní předností Altus Varia je možnost nastavit nákupní strategii, která bere v potaz dobu dodání zásob, rychlost obratu zásob na skladě a řídit tak automatické dodávání zásob od dodavatele. Další funkcí je vytváření variant produktů, které umožní jeden konkrétní produkt rozlišit barvou, velikostí produktu, pozicí ve skladu, zkrátka je možné editovat varianty jako jednotlivé produkty. Dále Altus Vario ukazuje svou funkci v řízení dostupnosti zboží a rezervace produktů.

3.3.1.1.2 Výhody řešení

Modul skladové evidence je velmi vospělý, pro korporátní společnosti je velmi užitečný. Skladový modul může být propojen do dalších modulů, jako je například modul pro pokladnu, účetnictví, mzdy, atp. Další výhodou je ceník zpřístupněný uživatelům a možnost pronájmu či odkupu systému.

3.3.1.1.3 Nevýhody řešení

Komplexnost a obsáhlé funkce Altus Varia si žádají uživatelsky složitý a nepřívětivý design, což může být pro malé podniky (v řádu jednotek zaměstnanců) nepřehledné. Další nevýhodou se jeví fakt, že evidence lze řídit pouze ze zařízení, kde je nainstalovaný klient. Není tedy možné řešit skladovou evidence z libovolného zařízení.

3.3.1.1.4 Cena řešení

Cena tohoto řešení se odvíjí od počtu přikoupených modulů, velikosti podniku a volby prostředí, na kterém bude aplikována serverová část aplikace.

Drobní podnikatelé (1 uživatel)

- Cena od 5 000 Kč do 10 000 Kč
- Možnost pronájmu 180 Kč až 360 Kč za měsíc

Malé a střední firmy (1 až 10 uživatelů)

- Cena od 15 000 Kč do 250 000 Kč

Velké firmy a organizace (více než 10 uživatelů)

- Cena od 200 000 Kč do 2 000 000 Kč

3.3.1.2 ABRA FlexiBee

Další systém, který nabízí řízení skladové evidence se nazývá ABRA FlexiBee. Skladová evidence není jako v případě Altus Varia řešena jako dostupný modul, ale řízení skladových zásob je jednou z mnoha evidencí, které si zákazník do podniku zakoupí jako součást balíčku. Součástí balíčku může být také daňová evidence, řízení účetnictví a majetku, systém pro správu mezd atd.

3.3.1.2.1 Možnosti skladové evidence

Abra Flexibee v části skladové evidence nabízí funkce pro rezervaci zboží, evidence sériových čísel, šarží a datum spotřeby. Dále systém umí zobrazit fyzické umístění produktu ve skladu, hlídat minimální množství na skladě, které poté zasílá notifikace uživateli, případně dodavateli. K jednotlivým produktům lze přikládat přílohy, což je možné využít například u přikládání certifikátů o produktu, manuálů, atp.

3.3.1.2.2 Výhody řešení

Klienta je možné nainstalovat na vlastní počítač, případně server nebo provozovat skladovou evidenci v cloudovém řešení za měsíční poplatek. ABRA Flexibee má přehledný ceník balíčků bez nutnosti registrace.

3.3.1.2.3 Nevýhody řešení

Skladovou evidenci lze řídit pouze za pomoci nainstalovaného klienta, který je k dispozici pro Windows, Mac OS X nebo Linux, a to i při provozování skladové evidence v cloudovém řešení. Není tedy možné skladovou evidenci řídit z mobilního zařízení.

3.3.1.2.4 Cena řešení

Balíček One (pro 1 uživatele)

- Zdarma s omezenou funkcí
- Obsahuje v sobě skladovou evidenci

Balíček Basic

- Cena odkupu 3 950 Kč / 1 uživatel
- Cena pronájmu 295 Kč / 1 uživatel / 1 měsíc
- Stejně jako balíček One, rozšíření o majetkovou evidenci

Balíček Business

- Cena odkupu 6 950 Kč / 1 uživatel
- Cena pronájmu 595 Kč / 1 uživatel / 1 měsíc
- Rozšířené o mzdy, personalistiku, evidenci nabídky, poptávky a objednávky, užití pro příspěvkové a neziskové organizace

Balíček Premium

- Cena odkupu 9 950 Kč / 1 uživatel
- Cena pronájmu 795 Kč / 1 uživatel / 1 měsíc
- Rozšířeno o pokročilý obchod a sklady, přístupová práva, možnosti přizpůsobení

3.3.1.3 Instantní sklad

Instantní sklad je webová aplikace vyvíjená společností CCV, s.r.o. Není tedy potřeba investovat do hardwarové části a počáteční náklady jsou tedy výrazně nižší. Pokud se podnik rozhodne pro tento typ skladové evidence, je veden k prvnímu kontaktu s konzultantkou, se kterou společně evidenci nastaví dle oboru podnikání. Následně je evidence spuštěna jako webová aplikace.

3.3.1.3.1 Výhody řešení

Instantní sklad nabízí ke svému řešení také vlastní čtečku čárových kódů. Jejich skladová evidence je tedy této čtečce plně uzpůsobena. Skladová evidence může být přes API rozhraní napojena na ERP⁵ systém nebo i e-shop. Aplikace je provozována jako webová aplikace, tudíž může být řízena z libovolného zařízení připojeného k internetu.

3.3.1.3.2 Nevýhody řešení

Ceník řešení není pro uživatele viditelný, pro zjištění ceny řešení je uživatel nucen k domluvení konzultace. K tomuto kroku je uživatel nucen i v případě, kdy by chtěl skladovou evidence pouze vyzkoušet. Evidence je uzpůsobena pro podniky s obrovskými sklady a pro menší podnikatele může být nepřehledná.

3.3.2 Zhodnocení výsledků

Z výše uvedených skladových evidencí lze odvodit, že na trhu se nachází velké množství systémů, které slouží pro řízení skladových zásob. Svou funkčností jsou systémy velmi vyspělé, dokáží zohlednit dobu, za kterou pravděpodobně dojde k vyskladnění všech

⁵ ERP systém – systém s jehož pomocí podnik řídí většinu svých interních činností, jako je plánování, zásoby, nákup zboží, marketing a jiné.

produktů a tuto dobu zahrnout do předběžných dodávek produktů na sklad. Obsáhlostí funkcí se ale stávají velmi robustními a komplexnějšími, což vede k pomalejšímu přizpůsobování k trendům dnešní doby. Všechny aplikace jsou koncipovány na velké podniky, které mají více provozoven a řeší vyskladňování z jednoho centrálního skladu. Na trhu se tak nachází obrovský prostor mezi skladovou evidencí vedenou na papíře či v tabulkách a těmito komplexními systémy.

Na trhu se v současné době nenachází žádná webová aplikace, která by umožňovala menším podnikatelům (v řádu jednotek pracovníků) vést skladovou evidenci rychle a přehledně. V této práci bych se chtěl vydat opačnou cestou. Mým cílem je vytvořit jednoduchou webovou aplikaci, která začínajícím podnikatelům umožní evidovat jejich produkty na skladě pouze pomocí webového prohlížeče. Důležitým aspektem bude možnost přizpůsobení aplikace přímo pro svůj obor podnikání, bez nutnosti užívat další nepotřebné moduly.

Z velkých hráčů na trhu lze vyčíst problém neprůhledného ceníku řešení, které je podnikům odkryto až po konzultaci. Ve většině případů není možné skladovou evidenci bezplatně vyzkoušet, jelikož na přizpůsobení evidence pro konkrétní obor je potřeba pověřený pracovník, který provádí zaškolování.

4 Návrh vlastního řešení

Druhá část této práce se zabývá převedením získaných informací k vytvoření vlastního řešení ve formě webové aplikace. Aplikace je vytvářena od úplného počátku, od zamyšlení se nad funkcionalitou a logickým rozdělením aplikace z pohledu budoucího vývoje. Dále bude následovat vytvoření vzorových person, které představují, kdo by měl aplikaci využívat a pro koho je určena, vymezení používaných technologií, grafické návrhy řešení, samotný vývoj webové aplikace až po důsledné otestování v běžném podnikatelském světě. Aplikace bude navrhována s responzivním designem, které umožní řídit zásoby na stolním počítači, tabletu i na mobilním zařízení.

4.1 Vzorové osoby

S pojmem vzorová persona se můžeme velmi často setkat v obsahovém marketingu, kde si pomocí person vyhraníme, kdo je naše cílová skupina, její pohlaví, jaké jsou její záliby, jak probíhá typický den person a případně jak daná osoba může využít naši službu či produkt. Pokud správně určíme naši cílovou skupinu, jsme připraveni pro ni přizpůsobit reklamní kampaň a zvýšit tak konverzí poměr naší cílové skupiny.

Je důležité říci, že tvorba vzorových person není užitečná pouze v marketingu, ale její tvorba by měla předcházet tvorbě jakéhokoliv produktu, v tomto případě při tvorbě naší aplikace. Vytvoříme si tedy 3 fiktivní osoby, pro které by mohla být aplikace užitečná.

Tabulka 1 - Vzorová persona 1

| | |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jméno | Kristýna Makarovová |
| Pohlaví | Žena |
| Věk | 21 |
| Koníčky | Focení, turistika, ruční výroba kosmetiky |
| Typický den | Probuzení, osobní hygiena, dopoledne strávené ve škole. Odpoledne výroba kosmetických produktů, focení, vyřizování objednávek. Večer strávený s rodinou, osobní hygiena, spánek. |
| Historie | Absolvování obchodní akademie. Nyní studuje na vysoké škole obor Veřejná správa. Při studiu ručně vyrábí a prodává kosmetické výrobky na inzertních portálech. |

Tabulka 2 - Vzorová persona 2

| | |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jméno | Věra Martínková |
| Pohlaví | Žena |
| Věk | 48 |
| Koníčky | Tanec, in-line bruslení, výroba a prodej dětského oblečení, provozování vlastního e-shopu. |
| Typický den | Probuzení, osobní hygiena, odeslání balíků na poštu. Příchod do dílny, šití dětského oblečení, obsluha zákazníků na prodejně. Odpoledne vyřízení příchozích objednávek a příprava balíků k odeslání. Večer strávený s rodinou na zahradě, osobní hygiena, spánek. |
| Historie | Absolvování základní školy a vyučení jako prodavačka na střední odborné škole. Po střední škole zaměstnání 3 roky jako prodavačka v papírnictví. Ukončení kvůli mateřské dovolené. Poté se stala podnikatelkou, provozuje vlastní dílnu a prodejnu s dětským oblečením. |

Tabulka 3 - Vzorová persona 3

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jméno | Petr Novák |
| Pohlaví | Muž |
| Věk | 41 |
| Koníčky | Sportovní aktivity, cestování s rodinou, práce na vlastním autě. |
| Typický den | Probuzení, osobní hygiena, odvezení dcery do školky. Cesta do Německa, kde nakoupí velké množství nářadí, které prodává na svém internetovém obchodě a jezdí po trzích po celé České republice. O volném čase si přivydělává opravou automobilů. |
| Historie | Absolvování základní školy a střední průmyslové školy jako strojař. Po střední škole nastoupil do zaměstnání do výroby, kde mu nevyhovoval pracovní režim a chtěl si řídit svůj čas sám. Zřízení živnostenského oprávnění, nákup a prodej nářadí na trzích po celé ČR. |

4.2 Vymezení funkcionalit

4.2.1 Přizpůsobení aplikace pro svůj obor podnikání

Po úspěšné registraci si uživatel pomocí průzkumníku přizpůsobí aplikaci pro svůj obor podnikání. Uživateli musí být umožněno vytvořit kategorie produktů, společné parametry pro všechny kategorie, tzn. například pozice ve skladu. Dále je důležité, aby si uživatel mohl vytvořit parametry produktů dle odpovídající kategorie. Díky těmto možnostem si uživatel může vytvořit údaje, které budou evidovány u všech produktů, ale také údaje, které budou vyplňovány pouze u konkrétní kategorie.

4.2.2 Základní přehled

Základní přehled tvoří úvodní stránku, která je zobrazena po přihlášení do aplikace. Jedná se o statistiku a stav skladu, důležité je zobrazení celkového počtu produktů, celkové množství produktů včetně jednotlivých variant a celková hodnota produktů kvůli vázanosti peněz v naskladněných produktech.

4.2.3 Přidávání produktů

Prvním krokem je volba kategorie, do které nově přidávaný produkt spadá. Pokud uživatel vybere kategorii, načtou se konkrétní údaje, které odpovídají zvolené kategorii. Uživateli musí být umožněno vytvářet varianty produktu. Každá varianta má poté svůj název, kterým může být například čárový kód nebo jiné označení. K produktu je dále potřeba funkce pro možnost nahrání náhledových obrázků ve formátu JPEG nebo PNG.

4.2.4 Editace produktů

Po kliknutí na produkt se uživateli zobrazí detail produktu, kde je mimo jiné uvedeno, kdy byl produkt do skladové evidence nahrán, jaké je celkové množství a celková cena celého produktu. Při této příležitosti musí být uživateli umožněno změnit libovolné atributy u produktu, případně produkt jako takový odstranit.

4.2.5 Režim vyskladňování

Pokud uživatel vyskladňuje více než jednu položku, není potřeba si každý produkt zobrazit a manuálně změnit množství. K tomuto účelu slouží funkce s názvem Režim vyskladňování. Uživateli se zobrazí vstupní pole, kde uživatel vyhledá produkt pomocí názvu, kódu produktu nebo kódu varianty. Nalezené varianty se mu zobrazí v seznamu, kde si zvolí, který produkt se má vyskladnit. Zvolené produkty se postupně budou přidávat do seznamu. K finálnímu vyskladnění uživatel poté klikne na tlačítko Vyskladnit.

Je vhodné, aby si uživatel mohl zaškrtnout volbu pro využití čtečky čárových kódů. V takovém případě uživatel naskenuje čárový kód, systém vyhledá produkt a automaticky ho přiřadí do seznamu k vyskladnění. Finální vyskladnění je opět potřeba provést kliknutím na tlačítko Vyskladnit.

4.2.6 Historie skladu

Každá změna, kterou uživatel provede, je potřeba zaznamenávat v historii skladu. U každé položky je potřeba zobrazit datum, kdy ke změně došlo, co a s jakým produktem se dělo a který uživatel změnu provedl. Jedná se o důležitou funkci, která dopomáhá nalézt produkty, které byly v minulosti již vyskladněny.

4.2.7 Inventura

Režim inventury je nedílnou součástí každého systému pro řízení skladové evidence. Je žádoucí, aby si uživatel v prvním kroku vybral kategorie produktů, na které se má spustit inventura. V následujícím kroku se uživateli zobrazí tabulka s evidovaným množstvím a políčko, kam uživatel zadá reálný stav produktu. Po vyhodnocení inventury je uživateli umožněno změny množství uložit, případně inventuru ukončit bez aktualizování množství u jednotlivých produktů.

4.3 Logické rozdělení aplikace

Webovou aplikaci lze logicky rozdělit do třech základních modulů, které na sebe vzájemně navazují. Moduly jsou shodné pro většinu online aplikací, kde si uživatel vytváří uživatelský účet a s webovou aplikací pracuje.

Nový uživatel přistupuje na úvodní obrazovku celé aplikace, kde se s aplikací seznámí, případně vyplní svoji e-mailovou adresu a vytvoří si nový účet. Tento modul je dále v práci nazvaný jako Public modul.

Druhým, a zároveň nejdůležitějším, modulem celé aplikace je Warehouse modul. V tomto modulu uživatel pracuje se svojí skladovou evidencí, přidává a vyskladňuje produkty, řídí své zásoby a zpracovává inventury.

Třetím – posledním modulem je Admin modul, který spolupracuje s Warehouse modulem. Admin modul je jakýmsi rozhraním, které umožňuje zobrazuje registrované uživatele, řídí jejich uživatelské účty, obnovuje ztracená data a vede si své přehledné grafy o využívání aplikace.

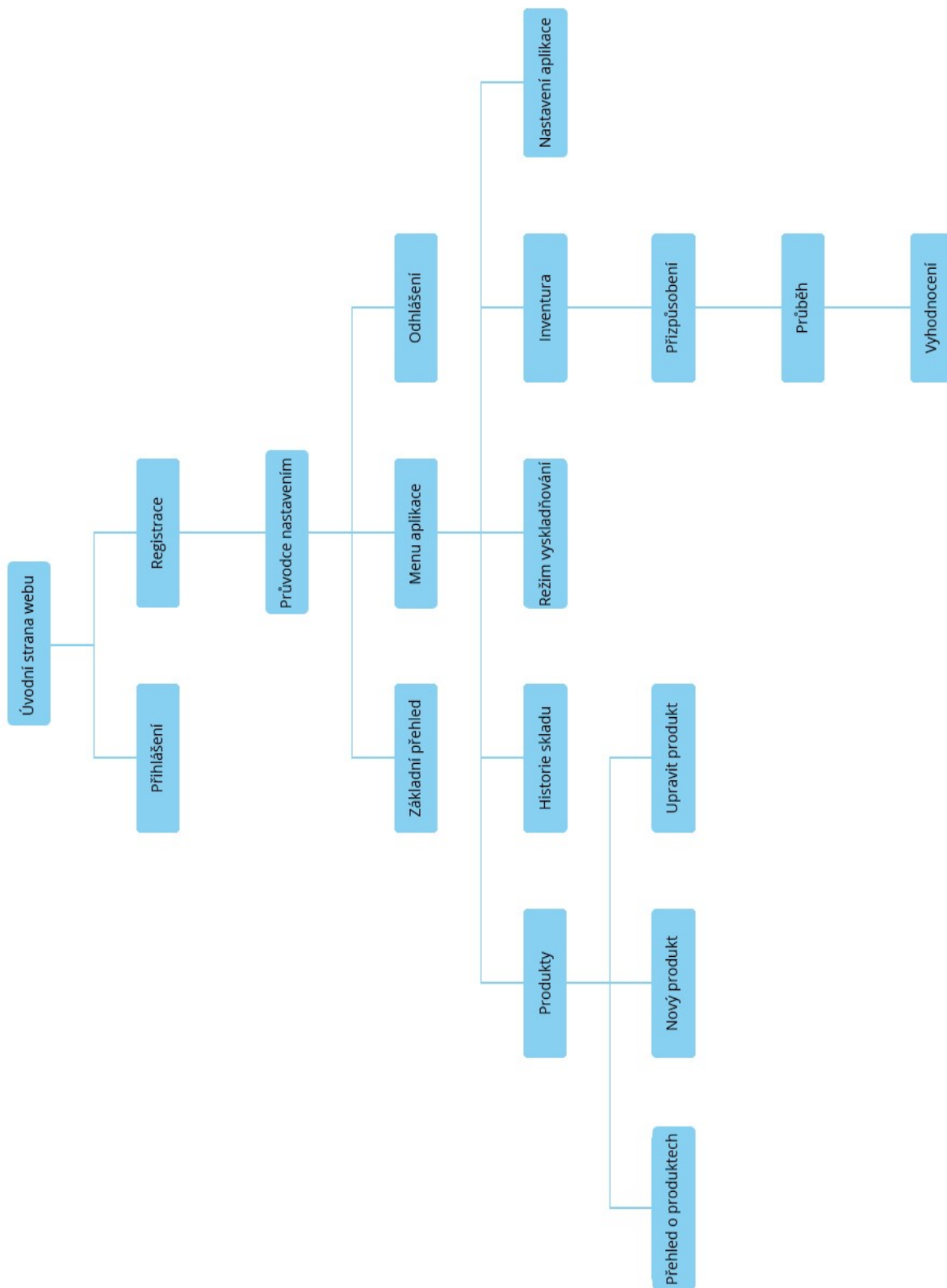
V rámci bakalářské práce je aplikace rozdělena pouze na dva základní moduly, a to na Public modul a Warehouse modul. Admin modul je vynechán záměrně, jelikož webová aplikace nevzniká za účelem komerčního využití, a není tedy potřeba spravovat uživatelské účty uživatelů.

4.4 Vytvoření sitemapy

V kapitole 4.3 jsme si logicky rozdělili aplikaci na dva základní moduly. Předtím, než začneme kreslit logické návrhy (wireframy) a případně grafické návrhy, vytvoříme si tzv. sitemapu. Sitemapu webové stránky si lze představit jako stromovou strukturu, která přesně ukazuje strukturu webové aplikace a jak budou jednotlivé obrazovky na sebe navazovat.

Základním důvodem, proč tvořit sitemapu ještě před logickým či grafickým designem, je uvědomění si uživatelských požadavků na skladový systém a roztřídění informací do jednotlivých obrazovek. Poté je možné pokračovat ve tvorbě logických designů, které se už zabývají konkrétními stránkami aplikace, které jsme si právě sitemapou vyhraničili.

Pro tvorbu sitemapy a logických designů jsem zvolil online webovou aplikaci mockflow.com, která je nabízena ve 3 verzích, z nichž základní verze je zdarma. Tvorba sitemap je u verze zdarma limitována pouze zlehka. V placené verzi aplikace je možné navíc přidávat komentáře a přílohy, obsahuje více grafických prvků a má výkonnější export projektů. Pro naše účely je verze zdarma pro tvorbu sitemapy více než dostačující.



Obrázek 3 - Sitemapa webové aplikace

4.5 Logický design (wireframy)

Pro tvorbu logických designů neboli wireframů jsem stejně jako u sitemapy zvolil online webovou aplikaci mockflow.com. Ve verzi zdarma je u logických designů omezení větší než u tvorby sitemap. Pravděpodobně největší omezení je vytvoření maximálně 1 projektu, a v něm v současné chvíli maximálně 3 logických designů. Tuto malou nevýhodu ve své práci řeším postupným exportováním již navrhnutých designů do formátu PNG, a tudíž je mohu v nástroji odstranit a pracovat tak na dalším, novém designu. Dále u verze zdarma je omezení kapacity na 1 GB, který je nicméně pro navrhování maximálně 3 designů dostačující. Za příplatkovou verzi Premium (cena 19\$ / 1 měsíc), TeamPack (cena 39\$ / 1 měsíc) nebo Enterprise (cena 160\$ / 1 měsíc) uživatel získá možnost pracovat v týmu současně, neomezené množství projektů a neomezený počet logických designů. Všechny přidávané hodnoty placených verzí lze najít na oficiálních stránkách <https://mockflow.com/pricing/>.

4.5.1 Úvodní strana webu



Obrázek 4 - Wireframe úvodní obrazovky

Minimalisticky navržená úvodní obrazovka je určena k představení hlavních funkcí webové aplikace. Zároveň je zde zobrazen řádek s číselnými údaji, které představují již dosažené cíle skladové evidence. Po zadání e-mailové adresy je uživatel přesměrován na vytvoření nového účtu, kde je e-mailová adresa již předvyplněna.

4.5.2 Vytvoření nového účtu

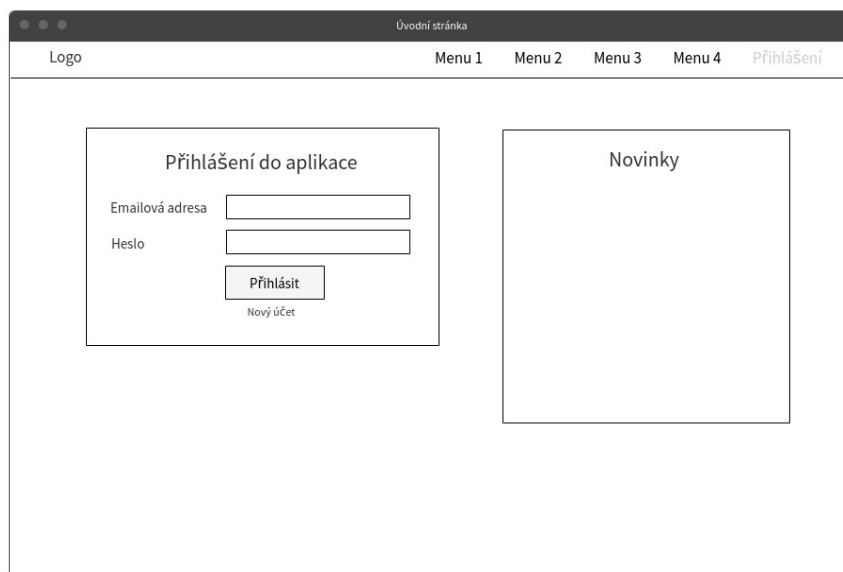
The image shows a wireframe of a web application window titled "Přihlášení". The window has a header bar with "Logo" on the left and "Menu 1", "Menu 2", "Menu 3", "Menu 4", and "Přihlášení" on the right. The main content area contains a centered form titled "Vytvoření nového účtu". The form includes the following fields and a button:

- E-mail:
- Jméno:
- Příjmení:
- Společnost:
- Heslo:
- Heslo znovu:
- Vytvořit:

Obrázek 5 - Wireframe pro vytvoření nového účtu

Princip webové aplikace je založen na vytvořeném uživatelském účtu, ke kterému se uživatel přihlašuje pomocí e-mailové adresy a hesla. Dále při vytváření nového účtu je potřeba vyplnit jméno a příjmení uživatele a společnost, která bude webovou aplikaci využívat. Díky nově vytvořenému účtu si uživatel může přizpůsobit skladovou evidenci dle svých preferencí a potřeb ve vlastním oboru podnikání.

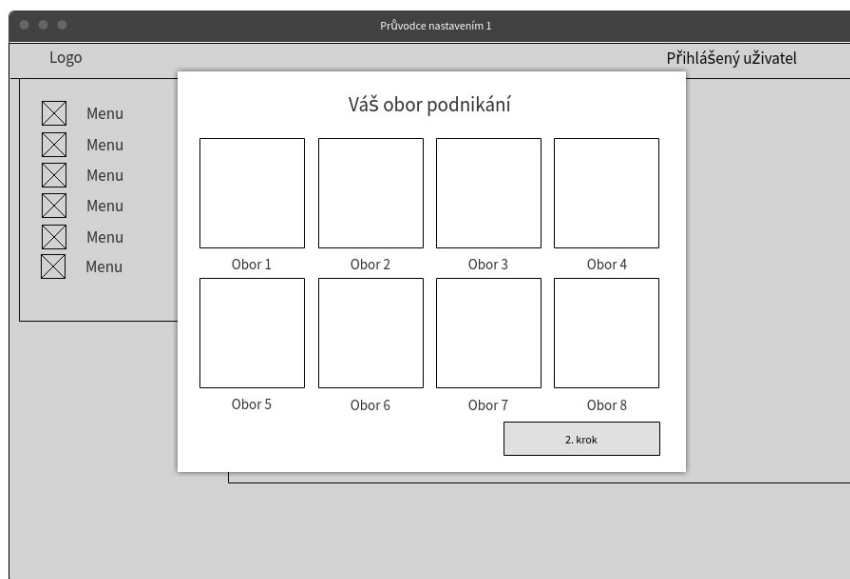
4.5.3 Přihlášení do aplikace



Obrázek 6 - Wireframe pro přihlášení do aplikace

Přihlášení je nedílnou součástí webové aplikace. Uživatel se do svého uživatelského účtu přihlašuje pomocí e-mailové adresy a hesla. Na přihlašovací stránce jsou také zobrazeny poslední novinky, které se ve webové aplikaci naposledy udály. Jedná se o přidání nových funkcí, jejich vylepšení, případně o opravy funkcí již existujících.

4.5.4 Průvodce nastavením nového uživatele – 1. krok



Obrázek 7 - Wireframe pro průvodce nastavením - 1. krok

Po vytvoření nového uživatelského účtu je uživateli zobrazen průvodce prvotním nastavením. V prvním kroku je uživateli zobrazeno sedm různých oborů. Pokud uživatel pochází z jiného oboru, je možnost zvolit osmý obor s názvem "Ostatní".

4.5.5 Průvodce nastavením nového uživatele – 2. krok



Obrázek 8 - Wireframe pro průvodce nastavením - 2. krok

Ve druhém kroku průvodce nastavením si uživatel vytváří kategorie produktů. První možností je kategorie nevytvářet. Tuto možnost využijí převážně uživatelé, kteří prodávají pouze jeden druh produktu. Druhou možností je kategorie pro své produkty vytvořit. Výhody vytvoření kategorií jsou prezentovány v následujícím, třetím kroku průvodce.

4.5.6 Průvodce nastavením nového uživatele – 3. krok

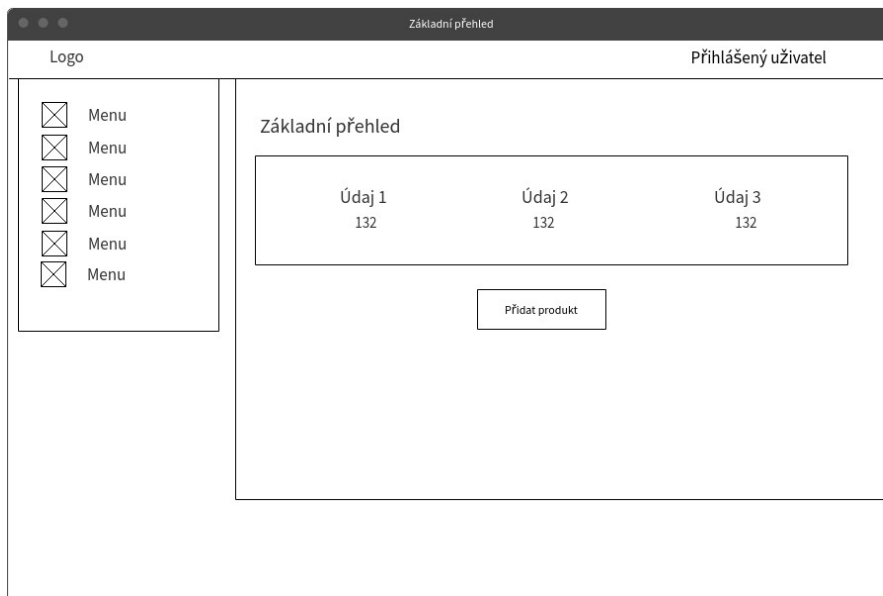
Obrázek 9 - Wireframe pro průvodce nastavením - 3. Krok

Ve třetím a zároveň posledním kroku průvodce nastavení si uživatel přizpůsobí jednotky množství a měny, se kterými bude následně skladová evidence pracovat.

Pomocí společných parametrů si uživatel může přidat vlastní údaje, které bude chtít u svých produktů evidovat. Společné parametry produktů se vztahují ke všem vytvořeným kategoriím v předchozím kroku, ale také pro možnost, kdy uživatel žádné kategorie nevytvořil. V takovém případě se kolonky ve společných parametrech promítají do všech produktů.

Pokud si uživatel v předchozím, tedy druhém kroku, kategorie vytvořil, může si pomocí parametrů kategorií vytvořit kolonky pro každou kategorii zvlášť. Tuto možnost ocení uživatelé, které budou ve skladové evidenci evidovat odlišné produkty.

4.5.7 Úvodní stránka aplikace – základní přehled



Obrázek 10 - Wireframe pro základní přehled

Po úspěšném dokončení prvotního průvodce, případně po přihlášení do svého účtu, je uživatel přesměrován na stránku se základním přehledem. Zde jsou zobrazeny číselné statistiky o všech naskladněných produktech, jako je například celkové množství či celková cena všech produktů.

4.5.8 Přidání nového produktu

The wireframe shows a web application window titled "Nový produkt". On the left, there is a sidebar with a "Logo" placeholder and a list of six "Menu" items, each with a checked checkbox. The main content area is titled "Nový produkt" and contains an "Uložit" (Save) button in the top right. Below the title, there are four input fields labeled "Kód", "Název", "Výrobce", and "Materiál". Underneath these is a section titled "Varianty produktu" which contains a table with two rows and four columns. Below the table is a "Přidat variantu" (Add variant) button. At the bottom of the form, there is an "Obrázek" (Image) label and a "Nahrát" (Upload) button. The top right corner of the window shows "Přihlášený uživatel" (Logged in user).

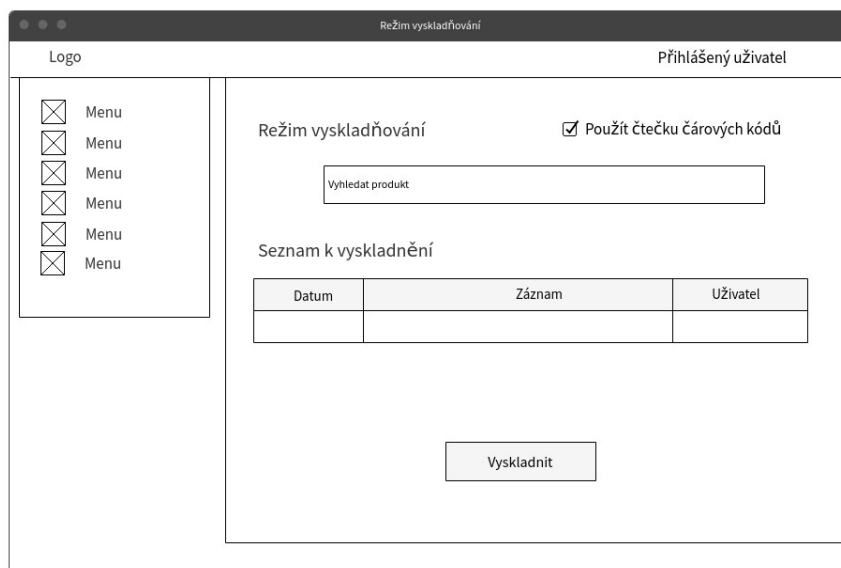
Obrázek 11 - Wireframe pro přidání nového produktu

Přidávání produktu patří ve skladové evidenci k těm nejdůležitějším funkcím. Při vytváření nového produktu je kladen důraz na rychlé a přehledné zpracování. U každého produktu je vyžadováno zadání kódu nebo názvu produktu a vyplnění alespoň jedné varianty.

Varianty jsou složeny ze tří základních pilířů. Prvním pilířem jsou přednastavená – neměnná políčka, kterými jsou kód varianty, její množství a cena. Druhým pilířem jsou společné parametry, které uživatel vytvořil v průvodci nastavení. Třetím a zároveň posledním pilířem jsou individuální parametry, které jsou aplikovány na uživatelem zvolenou kategorii produktu. Všechny parametry jsou pro správné vytvoření variant nezbytné. Variant produktu lze přidávat neomezené množství pomocí tlačítka “Přidat variantu”.

Při úpravě produktu je zobrazen naaprostu stejný formulář jako při přidávání produktu, včetně náhledové fotografie, která byla při přidávání produktu nastavena. Všechny údaje lze úpravou produktu změnit, tedy i vlastní kategorie produktu, včetně individuálních parametrů.

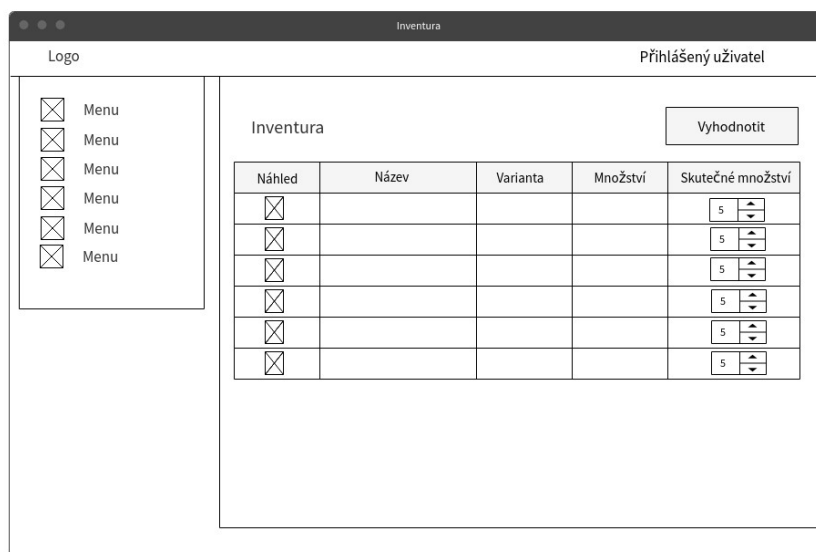
4.5.9 Režim vyskladňování



Obrázek 12 - Wireframe pro režim vyskladňování

Režim vyskladňování je koncipován s maximální jednoduchostí. Uživateli se zobrazí jedno vstupní pole, kde uživatel vyhledá produkt, který bude chtít vyskladnit. Všechny zvolené produkty se postupně přidávají do seznamu k vyskladnění. Pro konečné vyskladnění je požadováno stisknutí tlačítka “Vyskladnit”.

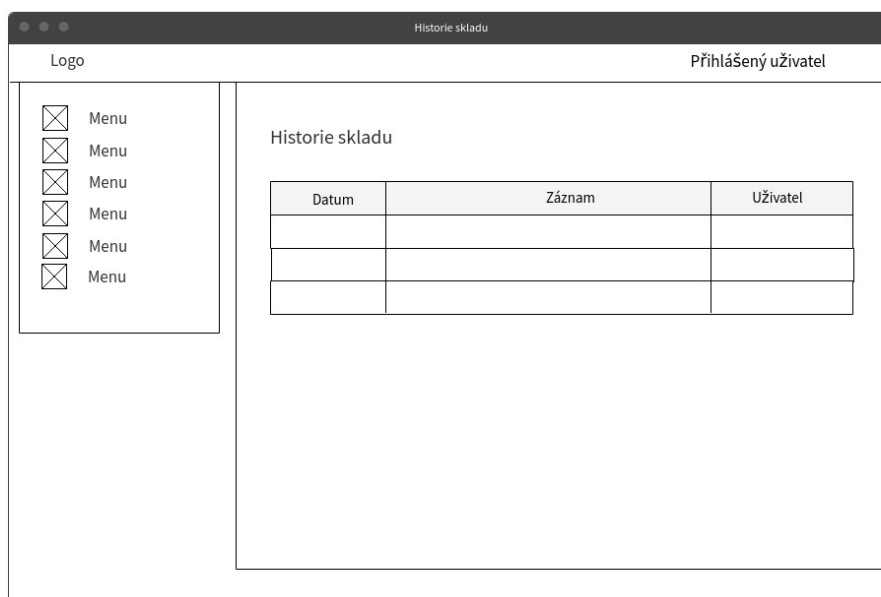
4.5.10 Inventura



Obrázek 13 - Wireframe pro inventuru

Inventura je zobrazení všech variant produktů, včetně uvedení jejich množství. Každá varianta produktu je zobrazována v tabulce jako jeden řádek. Uživatel postupně provádí inventuru a zanáší skutečné množství do krajního políčka u každé varianty. Pro dokončení je k dispozici tlačítko “Vyhodnotit”, které porovná údaje dle skladové evidence se zadaným skutečným množstvím.

4.5.11 Historie skladu



Obrázek 14 - Wireframe pro historii skladu

Přidání produktu, odstranění produktu i jeho upravování je zaznamenáváno v historii skladu. Každý řádek v historii skladu se skládá z data, kdy ke změně došlo, záznamu, který popisuje, co přesně a s jakým produktem se stalo. Posledním údajem je e-mailová adresa uživatele, který změnu produktu provedl.

4.5.12 Nastavení aplikace

The wireframe shows a settings window titled "Nastavení". At the top, there is a "Logo" placeholder on the left and "Přihlášený uživatel" on the right. A sidebar on the left contains six "Menu" items, each with a checkbox. The main content area is titled "Nastavení" and includes a "Uložit" button in the top right. The settings are organized into three sections: "Společnost" (Company) with a text input for "Název společnosti" and a dropdown for "Obor podnikání"; "Osobní údaje" (Personal data) with text inputs for "Jméno" and "Příjmení"; and "Kategorie" (Categories) with three text inputs labeled "Kategorie 1", "Kategorie 2", and "Kategorie 3", and a "Přidat kategorii" button below them.

Obrázek 15 - Wireframe pro nastavení aplikace

Nastavení, které uživatel provedl v prvotním průvodci nastavení, lze změnit v nastavení aplikace. V nastavení je možné změnit název společnosti, jednotky měny i množství, osobní údaje, vytvářet kategorie i společné parametry všech kategorií. Pro vytvořené kategorie lze také upravovat individuální parametry.

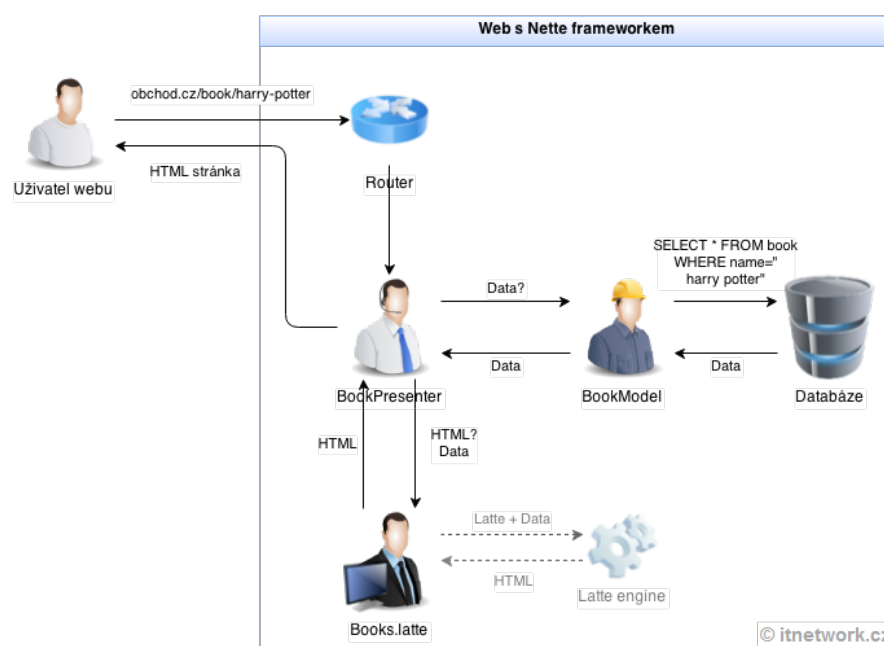
4.6 Metodika řešení

Při samotném vývoji webové aplikace je využito předešlých zkušeností při vývoji aplikací. Aplikace je proto navrhována pomocí ověřených nástrojů a postupů, které vedou k efektivnímu způsobu vývoje webových aplikací.

4.6.1 Nette Framework

Serverová část aplikace je navrhována pomocí vyspělého Nette Frameworku. Využitím frameworku je minimalizováno bezpečnostní riziko, které při programování webových aplikací v klasickém PHP vzniká. Dále framework umožňuje velmi rychle vytvářet webové aplikace a soustředit se na vývoj robustních, rozšiřitelných aplikací, bez nutnosti soustředění se na menší problémy, jako je například validace každého formulářového prvku na straně klienta, tak i na straně serveru a jiné opakující se problémy. Webové aplikace je navrhována za pomoci Nette Frameworku verze 2.4.

Nette Framework vyznává princip Model-View-Controller. Model slouží k získávání a zapisování dat do databázových tabulek, controller poté k ověřování totožnosti uživatelů a předávání získaných dat z modelu do šablony, tedy do tzv. View. Nette Framework obsahuje Latte šablony, které z předem získaných dat generují HTML kód, který je následně prohlížeči zasílán k zobrazení (11).



Obrázek 16 - Životní cyklus stránky navrhnuté pomocí Nette Frameworku

Zdroj: <https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>

4.6.2 Bootstrap

Nejrozšířenější CSS framework na celém světě je Bootstrap. Obrovskou výhodou tohoto frameworku je ucelený systém pro tvorbu responzivních webů pomocí tzv. grid systému. Bootstrap obsahuje předem předdefinované rozměry (breakpointy), pomocí kterých se rozeznává rozlišení zařízení a také jeho typ.

Kromě velmi vyspělého grid systému Bootstrap pohodlně obsluhuje vyskakovací okna, tzv. modal okna, které nejčastěji naleznou své uplatnění v potvrzovacích oknech.

4.6.3 jQuery

jQuery představuje JavaScriptový⁶ framework, který má, stejně jako jiné frameworky, za úkol maximálně zjednodušit opakující se úkoly. jQuery velmi významným způsobem zjednodušuje obsluhu událostí, jako je například kliknutí na tlačítko nebo stisknutí klávesy, umožňuje vytvářet animace prvků a v neposlední řadě umožňuje využívat AJAXové⁷ metody.

4.6.4 MySQL

MySQL je jeden z nejzákladnějších databázových systémů pro řízení relačních tabulek. Komunikace s relační databází probíhá – jak už název napovídá – pomocí jazyka SQL (4). V rámci této práce je využíván systém phpMyAdmin, který umožňuje jednoduchou a přehlednou správu databáze pomocí webového rozhraní.

⁶ JavaScript – jazyk, který pracuje na úrovni prohlížeče a obsluhuje prvky HTML dokumentu

⁷ AJAX (Asynchronous Javascript and XML) – představuje JavaScript, kterým lze zpracovávat data bez nutnosti znovunačtení stránky. Velké využití nalézá například u interaktivního získávání dat z databáze opět bez nutnosti překleslení celé webové stránky.

4.7 Implementace

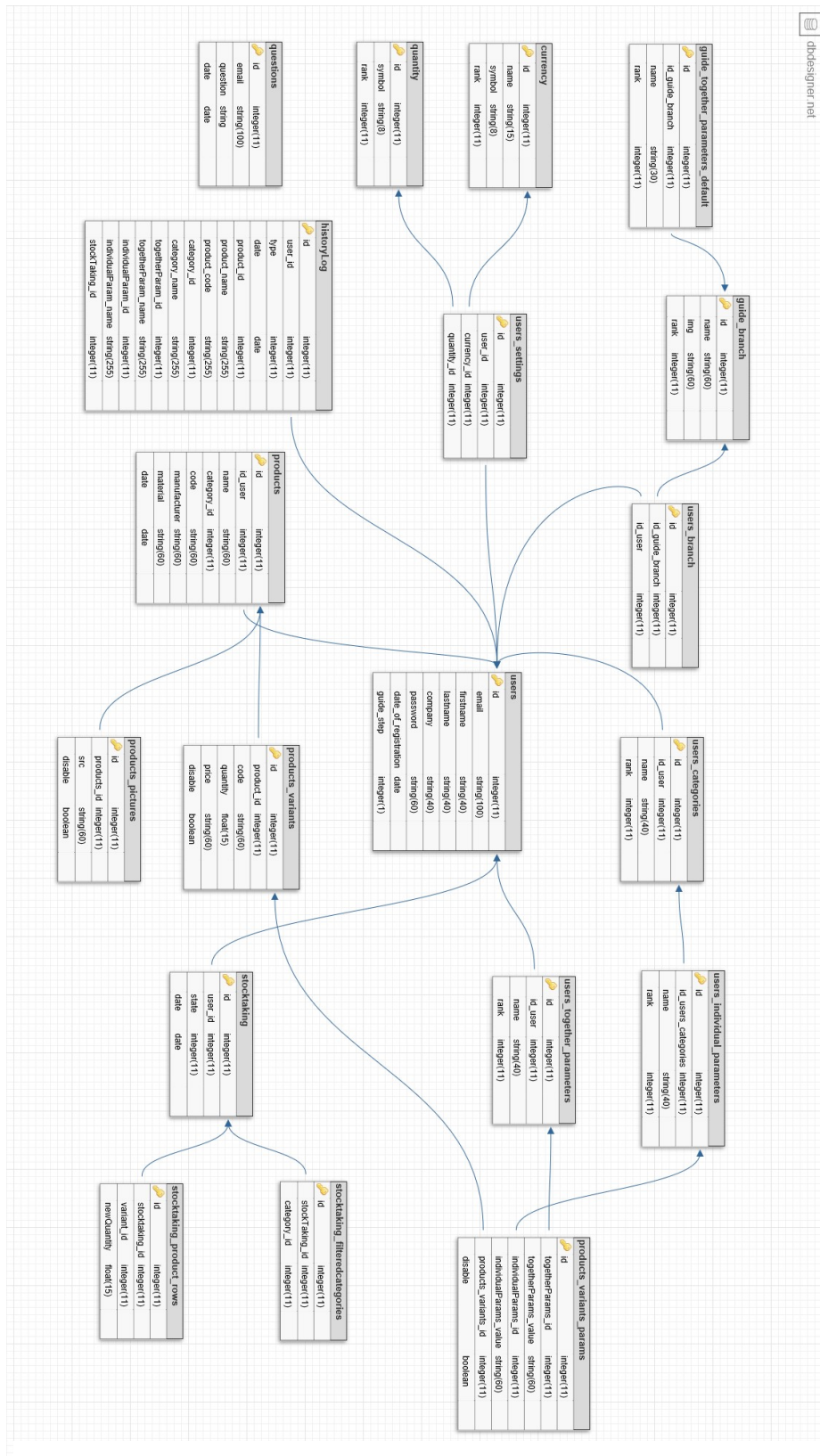
4.7.1 Prvotní nastavení

Jako první přichází přesměrování domovské stránky z `http://localhost/projekt/www` na `http://localhost/projekt`.

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteRule ^$ www/ [L]
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_URI} !^www/
    RewriteRule ^(.*)$ www/$1
</IfModule>
```

Zdrojový kód 1 – Přesměrování domovské stránky v souboru .htaccess

4.7.2 Návrh databáze



Obrázek 17 - Grafický návrh databáze

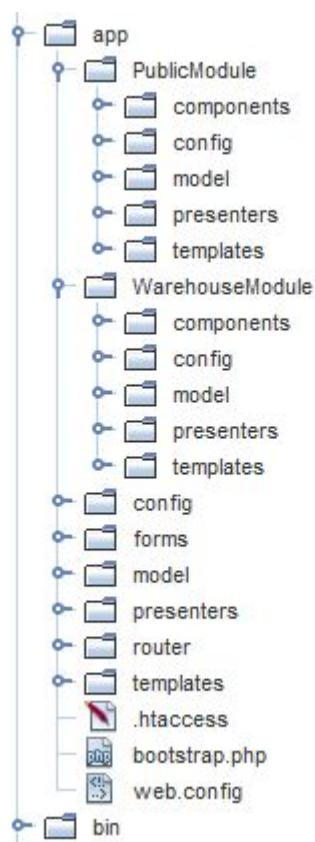
Zdroj: vlastní návrh pomocí aplikace <https://dbdesigner.net/>

4.7.3 Rozdělení aplikace do modulů

Základní kostra aplikace je rozdělena na 2 základní moduly. Prvním modulem je tzv. PublicModule, který zahrnuje tu část aplikace, která je přístupná nepřihlášeným uživatelům. Jedná se tedy o úvodní obrazovku neboli landing page, který slouží pro seznámení uživatele s aplikací a k první konverzi – registrace do aplikace.

Zbylá část webové aplikace, tedy vše, co se odehrává po úspěšném přihlášení do aplikace, je obsažena v modulu, který se nazývá WarehouseModule. V tomto modulu uživatel pracuje se svojí skladovou evidencí, přidává a upravuje produkty, vytváří a spouští inventury nebo si v nastavení aplikaci přizpůsobuje pro své potřeby.

Aby bylo možné aplikaci do modulů rozdělit, v samotném začátku je potřeba výchozí kostru aplikace, kterou nám Nette v základu nabízí, upravit. V kořenové složce app\ jsme vytvořili již zmiňované složky PublicModule a WarehouseModule. Každý z vytvořených modulů má poté své složky components, config, model, presenters a templates.



Obrázek 18 - Upravená struktura aplikace do modulů

4.7.4 Routování URL adres

4.7.4.1 Routování obecně

Routování neboli směrování URL adres je způsob, kterým lze z URL adresy přečíst, který presenter se má načíst a jaké má parametry. Abychom mohli odkazovat na nově vytvořené moduly, je potřeba zaregistrovat, pod jakým tvarem URL adresy se má naslouchat a na který presenter adresa odkazuje (12).

```
http://example.com/product/detail/123
```

Zdroj: <https://doc.nette.org/cs/2.4/routing>

Zdrojový kód 2 - princip routování URL adres

URL adresu uvedenou výše Nette automaticky rozpozná a zajistí, aby se zavolala presenter `ProductPresenter` a na něj poté šablona s názvem `detail`. Znaky `123` v tomto případě slouží jako identifikační prvek produktu, který je předán do konstruktoru.

4.7.4.2 Routování URL adres do jednotlivých modulů

Aby bylo možné pomocí URL adresy rozponat, o který modul uživatel žádá, je potřeba přizpůsobit routování v souboru `app\router\RouterFactory.php`.

Ve zdrojovém kódu níže jsou vytvořena pravidla pro routování, které zpřístupní `warehousemodule` na URL adrese `www.domena.cz/sklad/` a `PublicModule` poté bez přípony na `www.domena.cz/`. Obdobným způsobem lze vytvářet nové a nové moduly, které by mohly být ve tvaru `www.domena.cz/admin/`, `www.domena.cz/blog/` apod.

```

public static function createRouter()
{
    $router = new RouteList;

    // WarehouseModule
    $warehouseRouter = new RouteList('Warehouse');
    $warehouseRouter[] = new Route('sklad/<presenter>?<action>[/<id>]',
    'Prehled:default');
    $router[] = $warehouseRouter;

    // PublicModule
    $publicRouter = new RouteList('Public');
    $publicRouter[] = new Route('<presenter>/<action>[/<id>]',
    'Homepage:default');
    $router[] = $publicRouter;

    return $router;
}

```

Zdrojový kód 3 - Routování URL adres do odpovědných modulů

4.7.5 Ukázkový kód pro získání produktu dle ID

Zdrojový kód představuje funkci `getProduct($product_id)`, která ve svém parametru obdrží ID hledaného produktu. Konstanta `TABLE_PRODUCTS` je zástupným znakem pro název databáze. Dle konvencí přehledného vývoje je zvykem názvy databázových tabulek a jejich sloupce zapisovat do konstant v horní části zdrojového souboru. Názvy jsou tak velmi přehledné, a pokud se mění názvy databázových polí, stačí je změnit pouze jednou, a to právě v zástupné konstantě. V uvedeném příkladu je stejným způsobem obsazený název sloupce, kde se nachází ID produktu, a to konkrétně v konstantě `COLUMN_PRODUCTS_ID`. Metoda vrací první (a také jedinečný) prvek nalezeného produktu. Pokud při dotazování dojde k chybě, blok `try-catch` výjimku zachytí a metoda vrací `false`.

```

public function getProduct($product_id) {
    try {
        $product = $this->database->table(self::TABLE_PRODUCTS)->where([
            self::COLUMN_PRODUCTS_ID => $product_id
        ])->fetch();

        return $product;
    }
    catch (\Exception $e) {
        return false;
    }
}

```

Zdrojový kód 4 - Ukázka metody, která vrací produkt dle jeho ID

4.7.6 Ukázkový kód pro aktualizování množství produktů po provedené inventuře

Pokud uživatel zahájí inventuru, jsou mu zobrazeny všechny varianty produktů v řádcích pod sebou. U každé varianty je zobrazeno množství, které eviduje sklad a kolonka pro reálný stav skladu. U každé varianty uživatel zaznamenává reálné množství produktu na skladě a poté přechází k vyhodnocení inventury.

Jakmile inventuru vyhodnotí, jsou mu zobrazeny změny, které inventura zaznamenala. Uživatel může tyto změny ignorovat a ukončit tak inventuru, nebo zaznamenané reálné množství aktualizovat u všech produktů. K tomu slouží metoda `updateWarehouse($stockTaking_id)` zobrazena níže.

```

public function updateWarehouse($stockTaking_id) {
    $rows = $this->getFinishedStockTakingRows($stockTaking_id);
    if($rows){
        foreach($rows as $row){
            $this->productManager->updateVariantQuantity(
                $row[self::COLUMN_STOCKTAKING_PRODUCT_ROWS_VARIANT_ID],
                $row[self::COLUMN_STOCKTAKING_PRODUCT_ROWS_NEW_QUANTITY]);
        }
        return true;
    }
}

```

Zdrojový kód 5 - Aktualizování množství produktů po provedené inventuře

Každá inventura, která je ve skladové evidenci spuštěna, má přiděleno jedinečné ID. Pokud uživatel po vyhodnocení inventury hodlá aktualizovat množství produktů dle provedené inventury, je spuštěna metoda `updateWarehouse`, která přijímá jako parametr ID provedené inventury. V prvním kroku si metoda získá všechny řádky reálných stavů, které uživatel při inventuře vyplnil a uloží si je do proměnné `$rows`. V následujícím kroku je každý řádek iterován pomocí `foreach` cyklu. Pro každý řádek se následně volí metoda `updateVariantQuantity`, která v databázi aktualizuje množství jednoho konkrétního produktu.

4.7.7 Testování aplikace

Pro účely testování byly do skladové evidence nahrány produkty firmy V-MART, která se zabývá výrobou a prodejem dětského oblečení. Společnost disponuje vícero prodejními kanály, a to konkrétně prodejnou (výdejním místem), e-shopem a prodejem na bazarovém portálu `mimibazar.cz`. Z těchto důvodů společnost nemůže využít vestavěné skladové evidence, které jí e-shopové řešení nabízí.

Ve skladové evidenci bylo vytvořeno pět kategorií dle typu produktů. Konkrétně byla vytvořena kategorie Pyžama, Tepláky, Turkey, Legíny a Trička. Společným parametrem pro všechny kategorie je velikost. Individuální parametry kategorií nebyly

využity. Do skladové evidence bylo nahráno celkem 39 druhů zboží ve 192 variantách. Po úspěšném nahrání všech produktů do evidence byla zahájena a vyhodnocena inventura, která vedle k úpravě množství dle reálného stavu.

4.7.8 Spuštění aplikace v produkčním prostředí

Po úspěšném otestování byla webová aplikace spuštěna v produkčním prostředí, odkud je k dispozici z kteréhokoliv zařízení. K 15.3.2018 je webová aplikace k dispozici na adrese <http://vojtechmartinek.cz/skladovaevidence>. Zdrojové kódy jsou uloženy na CD jako příloha této práce.

5 Závěr

Hlavním tématem této bakalářské práce byla problematika evidence skladových zásob u podniků s více prodejními kanály. Jelikož tyto podniky nemohou využít integrované skladové evidence e-shopových řešení, bylo cílem vytvořit decentralizovanou evidenci skladových zásob za pomoci responzivní webové aplikace.

Teoretická část byla věnována obecně vývoji webových aplikací dle současných metodik návrhu a vývoje softwaru. Dále byl kladen důraz na analýzu dostupných řešení pro řízení skladové evidence, díky níž byl zjištěn výrazný nedostatek – na trhu zcela chybí možnost decentralizované správy skladu pomocí webové aplikace.

Praktická část byla věnována návrhu a implementaci webové aplikace, která by odpovídala požadavkům na správu skladu pro menší podniky. Aplikace mnou vytvořená splnila veškeré nároky, které byly uvedeny v prvotních požadavcích na webovou aplikaci. V ostrém testování konkrétním podnikem V-MART, který se zabývá výrobou a prodejem dětského oblečení splnila veškerá očekávání.

Ačkoliv původní zadání bylo splněno, aplikaci lze neustále vylepšovat a aktualizovat. V budoucnosti by bylo možné tuto aplikaci rozšířit o hromadnou úpravu produktů, možnost importu a exportu produktů pomocí XML feedu. Dále by bylo vhodné přidat do aplikace modul pro správu vytvořených účtů (řešení hesel, přidávání a odebrání uživatelů v rámci podniku...).

6 Seznam použitých zdrojů

1. Webová aplikace (Web Application). In: ManagementMania.com [online]. [cit. 2018-01-21]. Dostupné z: <https://managementmania.com/cs/webova-aplikace-web-application>
2. KOSEK, Jiří. Webový server. *HTML guru* [online]. [cit. 2018-02-8]. Dostupné z: <http://htmlguru.cz/vystaveni-webovy-server.html>.
3. BAŠE, Ondřej. *jQuery pro neprogramátory*. Brno : Computer Press, 2012. stránky 14-15. ISBN 978-80-251-3750-5.
4. Základy relačních databází, jejich využití v programování webu. *Grafická a multimediální laboratoř VŠE* [online]. [cit. 2018-02-8]. Dostupné z: <http://gml.vse.cz/data/oppa-webdesign/zaklady-db.html>
5. CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
6. CASTAGNETTO, Jesus. *Programujeme PHP profesionálně*. Praha: Computer Press, 2001. ISBN 80-7226-310-2.
7. KALÁB, Jan. Životní cyklus softwaru. *Kalábovic wikina* [online]. [cit. 2018-02-10]. Dostupné z: https://kalabovi.org/pitel:isz:zivotni_cyklus_softwaru
8. MCCONNELL, Steve. *Rapid development: taming wild software schedules*. Redmond, Wash.: Microsoft Press, c1996. ISBN 1-55615-900-5.
9. STREULEA, Thomas, Nino MISERINIA, Olin BARTLOMÉB, Michael KLIPPELC a Borja SOTO. *Implementation of Scrum in the Construction Industry* [online]. [cit. 2018-02-10]. Dostupné z: <https://www-sciencedirect-com.infozdroje.czu.cz/science/article/pii/S1877705816339601>
10. *Rational Unified Process: Best Practices for Software Development Teams* [online]. [cit. 2018-03-9]. Dostupné z: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
11. GRUDL, David. MVC aplikace & presentery. *Nette framework* [online]. [cit. 2018-03-9]. Dostupné z: <https://doc.nette.org/cs/2.4/presenters>
12. GRUDL, David. Routování URL. *Nette framework* [online]. [cit. 2018-03-11]. Dostupné z: <https://doc.nette.org/cs/2.4/routing>

7 Přílohy

- Příloha A Vytvořená aplikace – úvodní strana
- Příloha B Vytvořená aplikace – zobrazení produktů
- Příloha C Vytvořená aplikace – režim inventury
- Příloha D Vytvořená aplikace – nastavení
- Příloha E CD se zdrojovými kódy

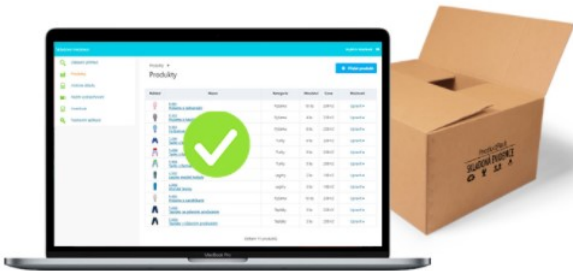
7.1 Příloha A Vytvořená aplikace – úvodní strana

Skladová evidence

O evidenci Ceník Reference Kontakt Přihlásit se

ONLINE PŘEHLED O VAŠICH PRODUKTECH?

Zadejte Váš e-mail [Zdarma vyzkoušet](#)





JAK SI VEDEME V ČÍSLECH?


| | | |
|---------------------|------------------------|------------------|
| 50 | 135 | 139280 |
| skladových evidencí | naskladněných produktů | hodnota produktů |

O EVIDENCI

Vlastníte internetový obchod, máte svojí prodejnu nebo snad vyrábíte své produkty? Právě pro Vás jsme vytvořili jednoduchou aplikaci, díky které získáte přehled o všech Vašich produktech. Nebudete již potřebovat žádný papír a tužku nebo snad tabulky. Získáte rychlou a přehlednou evidenci, kterou si otevřete na svém telefonu, na tabletu i notebooku. Je to čistě na Vás, jdete do toho?

 **Váš produkt**

 **Přidejte ho do evidence**

 **Získejte online přehled**

CENÍK

Webovou aplikaci **naskladnime.cz** jsme pro Vás vytvořili, aby Vám usnadnila podnikatelský život a Vy jste se tak mohli posunout na vyšší úroveň. Nezáleží na tom, zda jste velký podnik, malý podnik a nebo maminka na mateřské, která prodává ručně šité oblečení. Aplikace je přístupná pro všechny **zdarma**.

Zdroj: vlastní zpracování

7.2 Příloha B Vytvořená aplikace – zobrazení produktů

Skladová evidence Vojtěch Martinek

- Základní přehled
- Produkty**
- Historie skladu
- Režim vyskládávání
- Inventura
- Nastavení aplikace

Produkty > + Přidat produkt

Produkty

| Náhled | Název | Kategorie | Množství | Cena | Možnosti |
|---------------------------------------|---------|-----------|-----------|---------|----------|
| P-1004 Pyžama šedí hadí | Pyžama | 15 ks | od 239 Kč | Upravit | |
| P-1007 Pyžama zelený fotbal | Pyžama | 8 ks | od 239 Kč | Upravit | |
| P-1008 Stavební stroje | Pyžama | 5 ks | od 255 Kč | Upravit | |
| P-1009 Fotbalové pyžamo | Pyžama | 10 ks | 239 Kč | Upravit | |
| P-1010 Dinosauri | Pyžama | 5 ks | 239 Kč | Upravit | |
| P-1013 Pyžama s kopačkami | Pyžama | 3 ks | 239 Kč | Upravit | |
| P-1014 Pyžama s míči | Pyžama | 6 ks | od 239 Kč | Upravit | |
| P-1015 Pyžama s pandičkami | Pyžama | 16 ks | od 239 Kč | Upravit | |
| P-1016 Pyžama s modrými hadími | Pyžama | 21 ks | od 239 Kč | Upravit | |
| P-1017 Pyžama černočervené kozičky | Pyžama | 18 ks | od 239 Kč | Upravit | |
| P-1018 Pyžama vesmír | Pyžama | 10 ks | od 245 Kč | Upravit | |
| P-1019 Pyžama jednohroždi | Pyžama | 20 ks | od 245 Kč | Upravit | |
| P-1002 Pyžama sovičky | Pyžama | 14 ks | od 245 Kč | Upravit | |
| T-2009 Turky s medvídky | Turky | 6 ks | 209 Kč | Upravit | |
| T-2010 Turky s letadly | Tepláky | 6 ks | 209 Kč | Upravit | |
| T-2011 Turky s autíčky | Turky | 4 ks | 209 Kč | Upravit | |
| T-2012 Turky s princeznami | Turky | 6 ks | 209 Kč | Upravit | |

Celkem 17 produktů.

Zdroj: vlastní zpracování

7.3 Příloha C Vytvořená aplikace – režim inventury

Skladová evidence Vojtěch Martinek

Inventura > Nová inventura Vyhodnotit

Inventura

Vyvořeno: 15.3.2018 01:46
Kategorie: Pyžama, Tepláky, Turky, Legíny, Trička, Nezařazeno

| Náhled | Název | Kód varianty | Množství | Skutečné množství |
|--------|-----------------------------|--------------|----------|-------------------|
| | P-1006 Pyžamo šedí hasiči | 92 | 2 | 0 |
| | P-1006 Pyžamo šedí hasiči | 98 | 2 | 0 |
| | P-1006 Pyžamo šedí hasiči | 104 | 2 | 0 |
| | P-1006 Pyžamo šedí hasiči | 110 | 1 | 0 |
| | P-1006 Pyžamo šedí hasiči | 122 | 4 | 0 |
| | P-1006 Pyžamo šedí hasiči | 128 | 2 | 0 |
| | P-1006 Pyžamo šedí hasiči | 140 | 2 | 0 |
| | P-1007 Pyžamo zelený fotbal | 92 | 1 | 0 |
| | P-1007 Pyžamo zelený fotbal | 98 | 2 | 0 |
| | P-1007 Pyžamo zelený fotbal | 104 | 2 | 0 |
| | P-1007 Pyžamo zelený fotbal | 110 | 1 | 0 |
| | P-1007 Pyžamo zelený fotbal | 116 | 1 | 0 |
| | P-1007 Pyžamo zelený fotbal | 128 | 1 | 0 |
| | P-1008 Stavební stroje | 92 | 3 | 0 |
| | P-1008 Stavební stroje | 128 | 2 | 0 |
| | P-1009 Fotbalové pyžamo | 92 | 1 | 0 |
| | P-1009 Fotbalové pyžamo | 98 | 1 | 0 |
| | P-1009 Fotbalové pyžamo | 104 | 1 | 0 |
| | P-1009 Fotbalové pyžamo | 110 | 3 | 0 |
| | P-1009 | 116 | 4 | 0 |

Zdroj: vlastní zpracování

7.4 Příloha D Vytvořená aplikace – nastavení

Skladová evidence Vojtěch Martinek

[Základní přehled](#)
[Produkty](#)
[Historie skladu](#)
[Režim vyskládování](#)
[Inventura](#)
[Nastavení aplikace](#)

Nastavení aplikace Uložit

Společnost

Název společnosti
Obor podnikání

Osobní údaje

Emailová adresa
Vaše jméno
Vaše příjmení

Kategorie produktů

| | |
|--------------------------------------|---------|
| <input type="text" value="Pyžama"/> | Upravit |
| <input type="text" value="Tepláky"/> | Upravit |
| <input type="text" value="Turky"/> | Upravit |
| <input type="text" value="Legíny"/> | Upravit |
| <input type="text" value="Trička"/> | Upravit |

[Přidat kategorii](#)

Společné parametry produktů

| | |
|---------------------------------------|---------|
| <input type="text" value="Velikost"/> | Upravit |
|---------------------------------------|---------|

[Přidat parametr](#)

Zdroj: vlastní zpracování

7.5 Příloha E CD se zdrojovými kódy