**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY

**Implementácia systému na správu tovaru v sklade s webovím rozhraním na prehľad aktuálne naskladnených položiek**

BAKALÁRSKA PRÁCA

Daniel Hlavatý

Cielom je vytvoriť systém na správu tovaru, jeho ceny poplatku za uskladanenie a ich majitelov. Doležitosť je jednoduchosť a rýchlosť aplikácie pri preberaní a vydávaní tovaru.

.) Analýza konkurenčných systémov

.) Analýza možnosti identifikácie zákazníkov pomocou čítačky dokladov alebo mobilného telefónu

.) analýza možnosti pridávania obrázkov k tovaru

.) výber vhodnej platformy, programovacieho jazyka a použiteľných frameworkov pre daný systém.

.) návrh štruktúry databázy

.)Implementácia systému

.)Testovanie a zhodnotenie.

**Abstrakt**

V práci bude vytvorený jednoducho ovládatelný systém na správu tovaru. Popíše postup, výber technológii a samotnú implementáciu tohto systému. Výsledkom je je jednoduchá aplikácia schopná pridávať a meniť údaje o tovare a zákazníkoch, a jednoduchá webová aplikácia, slúžiaca na prehliadanie tovaru práve dostupného na sklade.

OBSAH

1, Ako to momentalne funguje?

2.) Analýza konkurenčných riešení

3) Vyber platformy, jazyka, databazoveho systemu

4) Moznosti pridavania fotiek, tlace a identifikácie zákazníkov.

5.) navrh a implementácia aplikácie

6.) Webové rozhranie

7.) Zhodnotenie výsledku

* slovensko.sk

\*<https://www.e-iceblue.com/Introduce/free-pdf-component.html#.XLON1DAzY_w>

**Úvod**

--opisat tovar atd blablity, pri ako to funguje teraz

**1 Ako to funguje teraz**

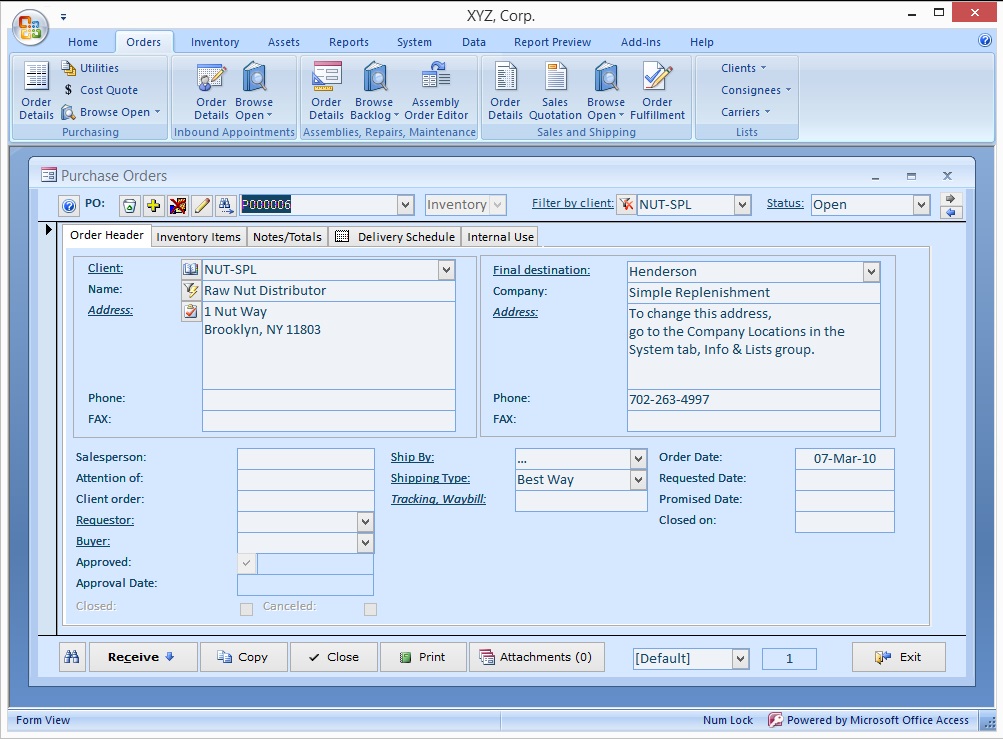
primanie tovaru, ziadne fotografie, zoznam tovaru, ide zakaznikov

**2 Analýza konkurenčných riešení**

Cieľom analýzy konkurenčných riešení je spoznať sa so základnými dizajnovými štandardami, ako aj vybrať riešenia ktoré by mohli riešiť problémi aj v našej aplikácií ako napríklad tlač či webkamera. Zhodnotíme ktoré prvky vyhovujú alebo nevyhovujú požiadavkám vyvýjanej aplikácie. Tento zoznam neobsahuje všetky voľne dostupné aplikácie riešiace problém inventárového manažmentu, ale vzorka dobre ukazuje na rozdieli medzi samotnými riešeniami.

**2.1 ABC Inventory**

Jeden z najpopulárnejších riešení. Prvotný dojem aplikácie by som rozdelil na dva stupne, najprv príjemný aplikácia je vlažnej modrej farby ktorá predpokladám neunavý oči pri ôsmich hodinách práce, potom pocit zmatenia, aplikácia ponúka mnoho skrytej funkcionality v hornom menu a preto pôsobí neprehladne. Vyzerá že uprednostnila voľné miesto pred prehľadnosťou.



Ukážka ABC Inventory okna

Zmena stavu objednávky sa vykonáva v samostatných oknách pri každej zmene užívateľ musí doplňiť údaje. Pri archivácií údaju je celkom zložité túto zmenu vrátiť. Pri zapísaní alebo zmene údajov v ktorej koľvek tabulke nie je jasné či sú údaje uložené alebo nie. Pri prekliknutí na iné okno, údaje zmiznú čo by mohlo byť matúce.

Vyhľadávanie tovaru je problémové, software neponúka zoznam všetkého tovaru, iba prehľad tovaru pre daného užívateľa. Neponúka žiadnu históriu úprav.

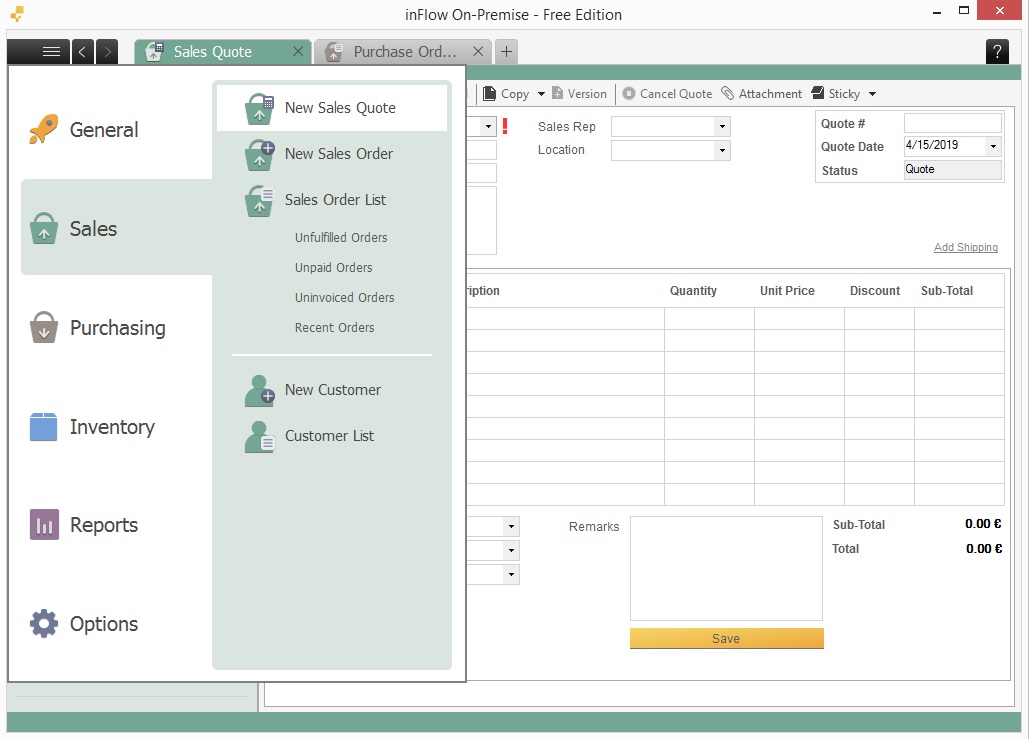
Aplikácia neponúka možnosť fotenia tovaru ani pridávania obrázkov, ale ponúka možnosť tlače. Užívateľ si vyberie objednávku ktorú chce vytlačiť v danom okne a aplikácia automaticky doplní text do PDF pripravená k tlači. Pri stlačení zapínača aplikácia vyberie prvú aktívnu tlačiareň a tlačí, ak žiadna nie je pripojené, tlačidlo stále svieti ale pri stlačení vypíše chybu. Fakt že tlačiť sa dá len v tomto okne môže byť pomalé pretože užívateľ sa musí preklikať.

Do našej aplikácie pravdepodobne odzrkadlíme fungovanie tlačenia, pretože je veľmi jednoduché bez zásahu užívateľa. Vyhľadávanie tovaru by bolo ideálnejšie, keby bolo na dvoch miestach aj u špecifického zákazníka aj samostatne všetok tovar. Musíme myslieť na to, že archivovaný tovar je možné zobraziť.

**2.2 InFlow** // zaciatocne nastav. pri 1 behu aplikacie – do zaveru ako vylepsenie.

Zadarmo riešenie postavené na Microsoft SQL servery. Pred prvím zapnutím aplikácie prebehnú prvé nastavenia, ako napríklad mena. Pri vstupne vyskočí tutoriál helper, ktorý zaberie vadšinu okna, teda nedá sa ignorovať. Táto vlastnosť môže byť dobrá pri programoch cielených pre vadšiu cieľovú skupinu, ako táto, naša však taká nie je.

Ukážka okna InFlow



Pri vytváraní objednávky, tovar musí byť pridaný naraz čo nie je dobré riešenie, chýba možnosť archivácie. Vyhľadávanie aj tlačenie chýba. Samotné okná sú prehľadejšie a menej nadvazujú.

Menu je vyriešené hamburger menu, čo je omnoho úhladnejšie ako v ABC Inventory. Toto menu sa nenachádza vo všetkých oknách aplikácie čo uvoľnuje priestor.

Aplikácia ponúka históriu, kreslí graf a umožňuje prehliadať zmeny. K Samotným údajom užívateľa alebo tovaru sa z tohoto okna dostať nedá.

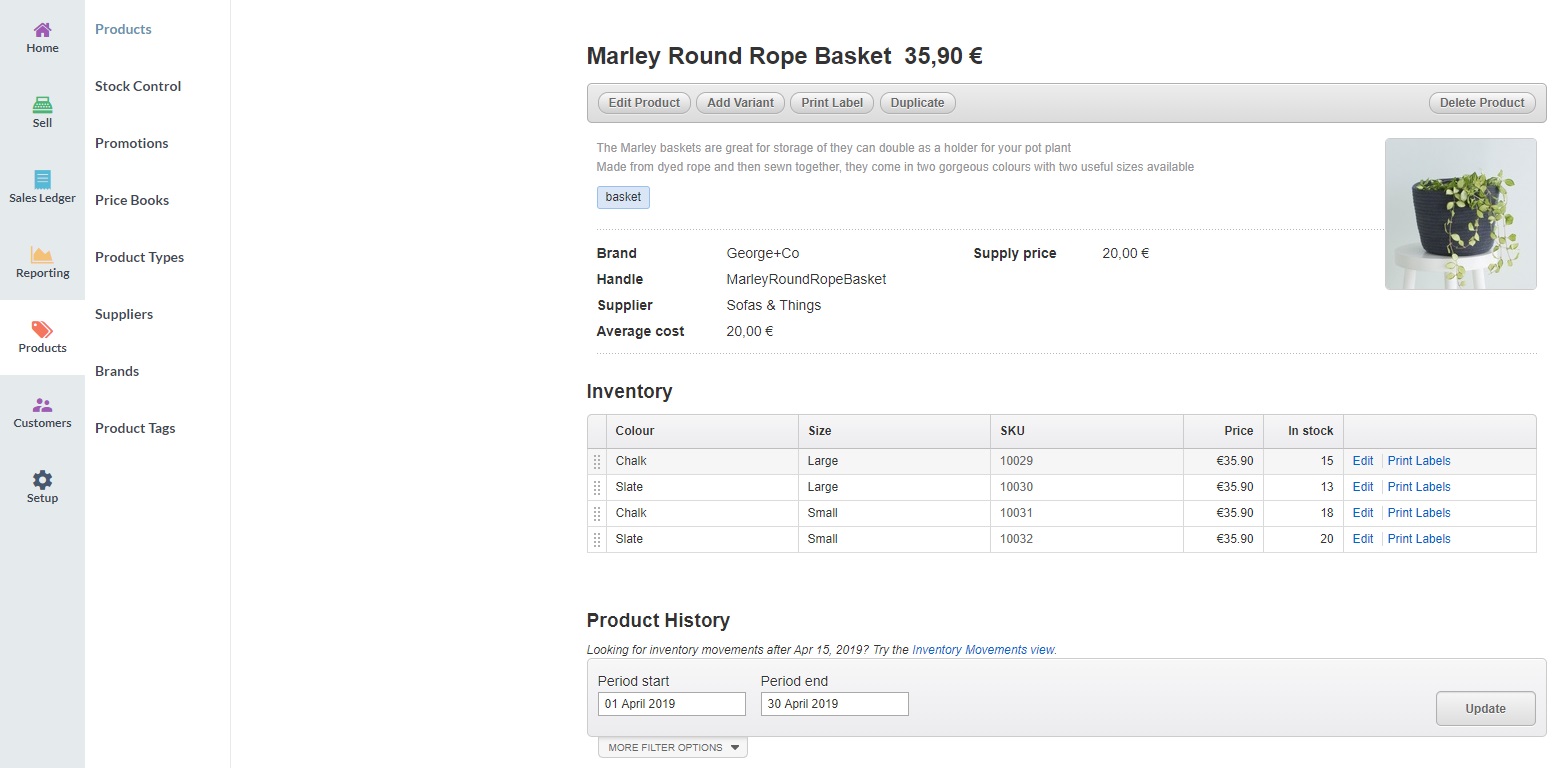
Tiež nie je jasné či upravené údaje sú uložené alebo nie. Popis tovaru sa nachádza v tabuľke vedľa ostatných údajov, avšak obyčajne býva dlhší, môže byť neprehľadné.

Aplikácia neponúka možnosť pridávania obrázkov. Ponúka možnosť tlače a to v akomkoľvek štádiu objednávky. Pri stlačení zapínača vyskočí okno podobné tlači Windows Word o výbere tlačiarne a počte strán. Náš sklad má dostupnú len jednu tlačiareň, preto toto okno eliminujeme.

Hamburger menu je ideálnym úhľadným riešením aj pre našu aplikáciu, pri úpravách zákazníka alebo tovaru môže tiež zmiznúť. Narozdiel od ABC Inventory jednotlivé okná sú menej prepojené a lahšie dostupné, čím sa aplikácia ľahšie ovláda. História tejto aplikácia má za úlohu zbierať všetky informácie o zákazníkoch a tovare, v našom prípade sledujeme prácu skladníka a zmeny ceny. Z histórie by sa mohlo dať dostať priamo k tovaru ktorý bol upravený, pre jednoduchšiu zmenu spať.

**2.3 Vend**

Prvá z uvedených ktorá je web aplikáciou, čo by bolo ideálne, mohla by naraz umožňovať aj skladníkovu prácu aj webové rozhranie pre prehľad tovaru na sklade. Výhodou webovej aplikácie je kompatibility, funguje všade kde funguje podporovaný prehliadač. Nevýhodou je že je neporovnateľne pomalšia práve preto že beží v prehliadači, napríklad pri tejto aplikácií, prechod medzi oknami tovaru a zákazníka je relatívne dlhá operácia, ktorá by mala byť takmer instatná.



Ukážka okna Vend

Aplikácia neponúka možnosť fotenia tovaru, ale ponúka možnosť zmeny obrázku jednoducho spolu zo zmenami ostatných údajov. Obrázok tovaru je schovaný a otvorí sa až pri otvorení detailu tovaru. Má prehľadné hamburger menu, ktoré je aktívne vo všetkých oknách čo pomáha pri prechodoch medzi oknami.

Tovar ktorý je predaný aplikácia nenávratne odloží preč z prehľadu čím uvoľnuje miesto. Umožňuje samostatný prehľad všetkého tovaru alebo tovaru na sklade, ale v samostatných oknách.

Výzor aplikácie je velmi príjemný, pre prehľad tovaru môžeme odzrkadliť funkcionalitu prehliadania tovaru, s takou zmenou, aby obrázok tovaru bol vždy viditeľný.

**3 Výber platformy, jazyka, frameworku a databázového systému**

Aplikácia musí bežat na počítači skladníka, a preto bude bežať len na operačnom systéme windows. Pri výbere platformy jazyka a databázového systému myslíme na rýchlosť aplikácie, potrebnú podporu dalších zariadení (webkamera, čítanie občianskeho, tlač). Dôležité sú aj moje znalosti, inklinujem k technológiám ktoré sú mi bližšie, tým že boli vyučované na škole.

**3.1 Platforma, jazyk**

Spravil som menší prehľad momentálne najpoužívajneších platforiem ktoré sú využívané na malé GUI aplikácie. Pri výbere myslíme na to, že samotná rýchlosť jazyka či platformy ja zanedbateľna, aplikácia vadšinu času zaberú operácie ako prístup do databázy.

**3.1.1 QT a C++**

C++ je jazyk známy svojou rýchlosťou, ale aj komplexnosťou. Napriek tomu, že je sním perfektne možné vyvíjať GUI aplikácie, jeho kód je dlhšie písaný. V praxi sa obyčajne využíva pri vyvíjaní rozsiahlejších aplikácií, kde rýchlosť samotného kódu hrá velkú rolu. QT by svojou rozsiahlou knižnicou uľahčilo prácu z webkamerou , čítaním kariet či tlačou.

**3.1.2 JAVA**

Java ponúka tiež niekoľko zadarmo dostupných riešení, ako Swing či JavaFX. Štandardné používateľské okno pôsobí ďaleko od toho windowsového, čo by mohlo pôsobiť odpudivo na skladníka. Prácu z webkamerou a tlačiarňou by som musel riešiť cez voľne dostupné knižnice, ktoré však nie sú štandardom.\*\* V čase písania práce je už dostupná informácia o skončení zadarmo podpory od Oracle, a aj preto som sa však tomuto jazyku vyhol.

**3.1.3 .NET**

.Net podporuje viacero jazykov, najpoužívanejšími sú C# a Visual Basic, niekedy pri vyvýjaní napríklad vnútornej logiky stránky sa používa F#. C# a Visual Basic sú si schopnosťami velmi podobné, líšia sa najme v syntaxi. Kedže aplikácia má bežať len na operačnom systéme Windows, môžeme využiť Windows Presentation Foundation, ktorý nám uľahčí prácu z kamerou, tlačou, dokonca aj pri pridávaní fotografií, okná budú skladníkovy už používateľsky známe, a tak sa z ich používaním už nebude musieť oboznamovať.

**3.2 Windows presentation foundation a WinForms**

Najvadším rozdielom medzi nimi je fakt, že WinForm je vlastne len vrstva postavená na štandardných windows ovládačoch(win controls). WPF nie je postavený na nich, a teda nie je na nich závislí. Môže to vyzerať ako malí rozdiel, ale v skutočnosti nie je.

Dobrým príkladom je napríklad zapínadlo s obrázkom a textom, pretože to nie je v štandardných Windowsových ovládačoch WinForms to neumožnuje jednoducho, musel by som si implementovať vlastný zapínač alebo využiť nejaký voľne dostupný. Vo WPF, zapínač môže obsahovať hocičo, pretože je to vlastne rám (border) s obsahom (content). V tomto prípade by stačilo vytvoriť zapínač, a vytvoriť dnu ovládač textu a obrázku. Naša aplikácia má byť jednoduchá na ovládanie, a bude obsahovať množstvo tabuliek ktoré budú upravované, tieto zmeny sú jednoduchšie uskutočniteľné vo WPF. Jeho výhodou je aj XAML ktoré rozdeľuje dizajn a vnútornú stránku aplikácie.

**3.3 Databáza** [file:///C:/Users/Daniel/Downloads/zaverecna\_prace%20(2).pdf](file:///C:\Users\Daniel\Downloads\zaverecna_prace%20(2).pdf) 5,2

Naše dáta budú mať svoju štruktúru, môžeme využiť relačné databázy, ktoré zabránia duplicite niektorých dát. Ich mínusom sú v podstate len neštrukturované dáta, ktoré nemáme. Dáta sú jednoducho uložené a vieme ich získať spať pomocou SQL queries. Výber z štandardných populárnych riešení, ktoré som vybral na základe webu DB-engines. :

**3.3.1 Oracle**

Oracle je momentálne lídrom databázových riešení od Oracle Corporation, je overená časom, ponúka výkon a prepracovanú funkcionalitu. Zadarmo je iba Oracle Express edícia, ktorá je oproti mySQL možnosťami obmedzená.

**3.3.2 MySQL**

MySQL je databázový systém sponzorovaný spoločnosťou MySQL AB, ale bola MySQL bolo kúpené taktiež Oracle Corporation. Je to open source, neponúka riešenia pre velké projeky, nepodporuje XML ale od verzie 5.7 podporuje JSON. Pri práci je možné pracovať s viacerími storage enginmi, štandardne InnoDB alebo MyISAM. InnoDB podporuje foreign klúče a stým bude táto funkcionalita pre našu aplikáciu dostatočnou.

**3.3.3 Microsoft SQL server**

Je relačný databázový systém vyvýjaný spoločnosťou Microsoft. Ponúkajú niekoľko edícií, zadarmo je edícia Express, ktorá je obmedzenou verziou oproti štandardnej verzií aj MySQL.

**3.3.4 PostgreSQL**

PostreSQL je plnohodnotný výkonný komunitou oblúbený open source relačný databázový systém vývíjaný PostreSQL Global Development Group. Plne podporuje triggery, procedúry môžu byť programované v jazykoch ako Java C/C++ Python a dalšie.

**4 Možnosti pridávanie fotiek, tlače a identifikácie zákazníkov**

**4.1 Možnosti a problémy indetifikácie zákazníkov**

Pri každom príchode zákazníka ako prvá prebieha jeho identifikácia. Medzi najčastejšie sposoby, ktoré môžeme zahliadnuť v okolí, patrí identifikácia občianskym preukazom. Spôsoby implementácie sa líšia.

Najrýchlejším a najelegantnejším spôsobom je využitie kartovej čítačky ktorá číta eiD číp na občianskom preukaze. Bohužiaľ napriek tomu že eid je dnes na Slovensku štandardom, nie každý občan ho má a preto tento spôsob nie je spoľahlivý.

Miesto eiD čipu by sme mohli využiť strojovo čítateľnú zónu(MRZ), čo je zóna na občianskom preukaze, kde sú údaje o držiteľovi a doklade zaznamenané vo forme opticky rozoznateľných znakov\*. Táto zóna je čítateľna takz. štrbinovou čítačkou dokladov. Cena takejto čítačky sa pohybuje od 400 eur a vyššie. Vzhľadom na to, že je to veľká investícia pre malý sklad, táto možnosť zamietnutá.

Ďalšou možnosťou by bolo občiansky preukaz odfotiť, a využiť jednu z knižníc na prekladanie textu z obrázkov. Najlpoužívanejšia z nich je knižnica IRONOCR alebo Tesseract od Google. Po otestovaní oboch som však zistil, že sú prekvapivo nespoľahlivé. Stačí poškriabaný alebo pošpinený preukaz, nesprávny uhol pod ktorým je fotené, a výsledkom je systém, ktorý si vyžaduje veľkú pozornosť pri fotení občianskeho preukazu človekom obstarávajúcim sklad. OCR je v praxi obyčajne využívaný na zbieranie dát z faktúr, kde štruktúra dokumentu a farba pozadia je jednodná.

Identifikácia telefónom môže byť tiež problémom. Takmer všetky telefónu dnes sú toho schopné, ale majú dve požiadavky. Prvou je, zákazník musí mať nainštalovanú danú aplikáciu. Druhou je samotná registrácia, zákazník nemá priradený identifikačný kód pri prvom príchode. Kedže hľadáme najjednoduchší spôsob považoval som za rozumné nevyužiť ani túto možnosť.

Poslednou a najjednoduchšou variantou, je identifikácia občianskym bez akýchkoľvek dalších systémov okrem skladovej aplikácie. Každému novému zákazníkovy bude idenfifikačný kód priradení, pri vytvorení zákazníka v systéme. Osoba pracujúca z aplikáciou si tak môže opakovaných návštevníkov jednoducho nájsť v aplikácii podľa mena, za predpokladu, že mená su relatívne jedinečné, alebo podľa jedinečného priradeného identifikačného kódu.

Vybral poslednú možnosť, pretože je používateľsky najjednoduchšia, osoby využívajúce sklad sa nemusia oboznamovat zo žiadnymi novými zariadeniami, a nevyžaduje žiadnu investíciu do čítacích zariadení.

**4.2 Možnosti pridávania fotiek**

Skladník pri príchode tovaru alebo dodatočne môže chcieť pridať príslušnú fotografiu.

Pri výbere obrázku z počítača, aplikácia by mala otvoriť štandardné windows okno pomocou ktorej vyberie cestu ktorú aplikácia uloží do databázy.

Rýchlejším spôsobom pre beh skladu by mohla byť webkamera, ktorú by mal skladník vždy pri počítači. WPF obsahuje triedy ImageView a Capture, s ktorými je jednoduché spraviť fotografiu. Urýchlilo by to proces pridávania fotiek, pretože by sa týmto eliminoval proces prenášania fotografií medzi nejakým zariadením a počítačom.

**4.3 Možnosti tlače**

Aplikácia má byť schopná vytlačiť faktúru zákazníkovy s jeho aktuálnym tovarom, pre jednoduchosť bez akéhokoľvek skladníkového zásahu. Vzhľadom na štruktúru dokumentu, ktorá bola tažko upraviteľná do .doc dokumentu, upravujem do PDF. \* Využívam knižnicu FreeSpire.PDF ktorá je zadarmo, a umožnuje zápis do pdf dokumentov podľa pozície, čo je ideálnym riešením. Takýchto knižník je viac, otestoval som aj knižnicu od synfusion a fungovali velmi podobne.

Na sklade sa nachádza len jedna tlačiareň, a preto pre jednoduchosť aplikácia neobsahuje menu výberu tlačiarne. Aplikácia vyberie prvú tlačiareň pripravenú na tlač a tlačí.

**5 Návrh a implementácia aplikácie**

**5.1 Vyhradenie funkcionality**

**5.1.1 Akcie a vlastnosti zákazníka**

Zákazník musí mať automaticky generovaný identifikačný klúč ktorý ho bude daľej reprezentovať, skladá sa z dvoch častí, posledných dvoch cifier aktuálneho roku a najmenšieho voľného trojciferného čísla vadšieho ako 100. Musí obsahovať údaje prvé meno, druhé meno, telefónne číslo, adresa, a počet tovaru ktorý má. Užívateľ ich musí byť schopný pridávať, upravovať všetky údaje okrem identifikačného klúča, nesmie ich však vedieť mazať. Z tabuľky zákazníka sa musíme vedieť dostať k zoznamu jeho tovaru. Z okna zákazníka skladník musí vedieť vytlačiť faktúru, ktorá bude obsahovať doplnené údaje.

**5.1.2 Akcie a vlastnosti tovaru**

Tovar musí byť označený majiteľom, musí obsahovať popis jednoducho prepísateľný aj v tabuľkách. Musí mať meno, veľkosť, a cenu. Tovar musí vedieť prechádzať medzi jednotlivými stavmi nepredaný, predaný kartou, predaný hotovosťou, vrátený, zaplatený, archivovaný. Tovar sa môže vždy nachádzať len v jednom z týchto stavov s výnimkou predaného a zaplateného. Tovar môže mať obrázok, ktorý môže byť odfotený priamo webkamerou. Stavy tovaru musia byť farebne odlíšené, a musia sa dať zobraziť aj archivované údaje.

**5.1.3 Akcie a vlastnosti záznamov**

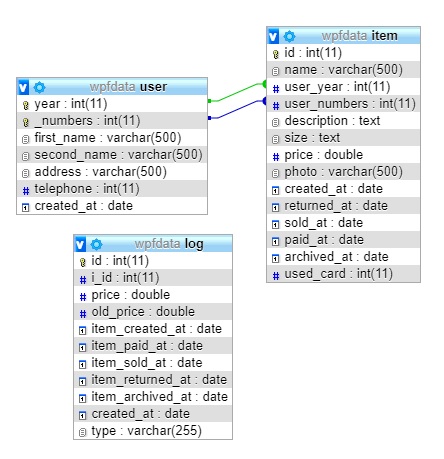
Záznamy v aplikácií slúžia len ako kontrola pre šéfa skladu, alebo môžu byť použité pri hľadaní vynníka chyby. Každý záznam musí obsahovať čas zmeny, popis akcie a ak bola zmenená cena, záznam uloží starú aj novú. Z každého záznamu sa musí dať prekliknúť priamo na okno detailu tovaru. Záznami sa tvoria len pre tovar. Musí sa medzi nimi dať prehľadávať podľa dátumu.

**5.2 Databáza // TO DO login zo stranky**

Záznami budú tvorené automaticky pomocou onupdate/oninsert triggerov. Foreign key pomôže zo zaručením priradenia tovaru majiteľovy. Stavy tovaru bude získavať aplikácia na základe prázdnych a vyplnených dátumov tovaru. Stav aplikácia zistí nasledovne :

Dátum created\_at je zapísaný pri vytvorení tovaru, a nikdy sa nemaže

* Začiatočný stav – všetky údaje sú prázdne
* Predaný kartou/hotovosťou - vyplnený je len dátum a sold\_at, ak bol predaný kartou, nastaví bool hodnotu card\_used na 1
* Vrátený – vyplnený je len dátum returned\_at
* Zaplatený – vyplnené len dátumy sold\_at a paid\_at
* Archivovaný – vyplnený je len dátum archived\_at



**5.3 Návrh okien**

Všetky tabuľky musia mať komboboxy z vyhľadávaním. Pridávanie niečoho sa vykonáva vždy v modálnych oknách, pri neuložených zmenách sa otvára popup window, ostatné operácie majú okná samostatné.

**5.3.1 Material Design**

Material design je dizajnovací jazyk vyvíjaný googlom a používa sa na dizajn stránok či aplikácií. Ponúka implementované štýly pre prvy WPF ako textbox. Ušetrí vela času zo štýlovaním aplikácie a aplikácia vyzerá jednotne.

**5.3.2 Hlavné okno**

Hlavné okno bude obsahovať hamburger menu na ľavej strane aplikácie, ktoré bude viditeľné len v hlavnom okne. Bude obsahovať základné prehľady, zoznam všetkých zákazníkov, zoznam všetkého tovaru a zoznam všetkých záznamov a pomocou hamburger menu sa medzi nimi bude dať ľubovolne pohybovať. Pri zapnutí aplikácií bude zobrazený zoznam užívateľov.

Zoznam všetkych zákazníkov bude tabuľka a zapínač slúžiaci na pridávanie nových zákazníkov. Tabuľka bude jednoduchá s možnosťou vyhľadávania podľa identifikačného čísla alebo mena. Pri dvojitom kliku na špecifický záznam sa otvorí okno užívateľa.

Zoznam všetkého tovaru bude tabuľka, ktorá bude mať v každom riadku tri zapínače. Samostatný zapínač pre detail toho špecifického tovaru, samostatný zapínač pre detail toho špecifického zákazníka, a samostatný zapínač pre popis aktivovanie ktorého odhalí popis tovaru v novom riadku. Aplikácia sa nebude pýtať na zmeny popisu, ale bude ich ukladať pri každej zmene.

Zoznam všetkých záznamov bude tabuľka s možnosťou vybratia dátumu dňa ako filtrovanie. Tabuľka musí obsahovať stĺpec s textom ktorý bude zložený na základe logov v databáze.

Pri vrátení sa k tomuto oknu na iné okno, alebo pri preklikávaní v hamburger menu medzi tabuľkami, tabuľka bude vždy aktualizovaná.

**5.3.3 Okno užívateľa**

Okno užívateľa musí obsahovať jednoduché menu pre možnosť zmeny osobných údajov zákazníka.

Musí obsahovať zoznam tovaru, s zapínačovím menu ktorým užívateľ môže meniť stav tovaru v logickej postupnosti. Môže upraviť popis a otvoriť detail tovaru rovnako ako v hlavnom okne, z doplnenou možnosťou archivovania. Archivovaný tovar sa nezobrazí v tabuľke ak užívateľ nerozhodne inak, a pri tlači sa do PDF dokumentu nezapisuje.

Musí obsahovať možnosť pridávania nového tovaru. V tomto okne sa bude realizovať aj tlač, môže poskytovať náhlaď na generovaný PDF dokument.

**5.3.4 Okno detailu tovaru**

Jednoduché okno z možnosťami zmeny údajov tovaru, vrátane stavu aj spatne. Obsahuje rovnaké popup okno na kontrolu uložených údajov. Detail tovaru musí podporovať pridávanie existujúcich obrázkov a pridávanie obrázkov za behu pomocou webkamery.

**5.3.5 Okná na pridávanie modal window**

Jediné okná aplikácie s tlačidlom spať, nemá štandardné okraje známe pre windows. Pri odchode z okna bez uložených údajov okno popup nevyhodí, po úspešnom pridaní sa okno automaticky nezatvára. Musí tiež vedieť pridávať fotografie oboma spomínanými spôsobmi

**5.3.6 Popup pri zmene údajov**

Pri neuložených zmenách a pokuse odchodu z okna, aplikácia sa opýta či zmeny uloží. Ak používateľ chce zmeny uložiť a z nejakého dôvodu operácia nepredbehne úspešne, používateľa vráti naspať do okna užívateľa. Ak údaje neopraví a pokúsi sa okno opustiť znova, aplikácia mu to bez opýtania dovolí. Ak zmeni uložiť nechce, bez oznamu odchádza.

**5.4 Využité prvky WPF a výsledná aplikácia**

**5.4.1 Webkamera**

Open source EMGU CV je knižnica ktorá ponúka objekty ImageViewer a VideoCapture ktorými môžeme robiť fotky pomocou webkamery. ImageViewer je trieda do ktorej obrázok z webkamery načítame, pomocou neho ho potom uložíme fyzicky do pc. Trieda VideoCapture je schopná urobiť fotku z dostupnej webkamery. Ak je webkamera vypnutá tak ju zapne. Zadarmo verzia knižnice neponúka možnosť webkameru vypínať a zapínať, preto po urobení fotografie kamera zostáva zapnutá kým sa automaticky nevypne.

**5.4.2 Tlač**

Po stlačení zapínača tlače, aplikácia najprv vytvorí PDF dokument z doplnenými údajmi zákazníka a jeho tovaru. Využíva knižnicu FreeSpirePDF ktorá podporuje pridávanie textu podľa pozície. Nepridáva archivované, a ak sa všetok tovar nevojde na jednu stranu, automaticky tlačí dve alebo viac. Aplikácia hľadá prvú voľnú aktívnu aplikáciu a ňou automaticky tlačí. Tlačiareň hľadá pomocou triedy PrinterSettings, ktorá obsahuje všetky tlačiarne ktoré dané zariadenie pozná, ako atribút má meno tlačiarne, a atribút ktorý značí či je tlačiareň pripravená tlačiť. Užívateľovy len oznámi úspech alebo neúspech

**Rozpisat aj design? ako su gridy ako to drzi? upravy gridov, ze virtualizacia je vypnuta atd, musi to byt zdokumentovane?**

**5.4.3 Výsledná aplikácia**

screenshoty a popis

**6 Webové rozhranie**

**7 Zhodnotenie projektu**

Porovnanie s konkurencnymi, moznosti vylepsenia do buducnosti, objavene problemi po ceste