

α.3 Databázový systém

Správa obchodu

Radek Vinařský

Kontakt na autora: radek.vinarsky@post.cz

Vytvořeno během studia na SPŠE Ječná, 31.1.2023

Obsah:

1. Úvod
2. Uživatelská část - ovládání, možnosti, UI, funkce
 - a. Hlavní menu
 - b. Význam jednotlivých sekcí, funkce
 - i. Editace dat
 - ii. Přidávání dat
 - iii. Mazání dat
 - c. Import
3. Technická část - třídy, vazby, databáze, připojení, error handling
 - a. Struktura Databáze
 - b. Struktura tříd
 - c. Návrhové vzory
 - d. Formuláře
 - e. Konfigurace
 - f. HW a SW požadavky
 - g. Chybové hlášky
4. Závěr

Úvod

Program je určen pro správu obchodu. Byl napsán v jazyce C# v prostředí Visual studio. Pro databázi byl použit MySQL server.

Návod pro spuštění je v README.txt.

Uživatelská část

Hlavní menu

Hlavní menu je levá část aplikace a umožňuje orientaci v aplikaci. Je součástí hlavního formuláře. Slouží pro přepínání mezi tabulkami, s jejíž daty uživatel může manipulovat. Tyto tabulky jsou zaměstnanci, produkty, zákazníci, objednávky, pobočky. Sekce import je pouze pro import nových produktů.

Význam jednotlivých sekcí, funkce

Sekcí označuji měnící formuláře na pravé straně aplikace (zaměstnanci, produkty, zákazníci, atd).

Sekce umožňují přehledné zobrazení dat, přidávání, mazání nebo úpravu dat. Ne všechny sekce ale umožňují všechny operace.

Přidávání

Pokud sekce umožňuje přidávání dat, bude pro to formulář s tlačítkem 'Přidat'. Je nutné vkládat validní data, jinak je aplikace nepřijme.

Úprava

Úprava dat se provede jednoduchým poklikáním na políčko, které chceme upravit. Po úpravě je NUTNÉ změnu potvrdit klávesou ENTER.

Mazání

Mazat lze zvolením jakéhokoli políčka na řádku které chceme smazat a klikneme na tlačítko 'Smazat'.

Sekce '**Zaměstnanci**' ukazuje jméno, příjmení, email a datum nástupu do práce. Zápisy můžeme mazat, přidávat a upravovat. Datum nástupu se vloží automaticky jako datum vytvoření zápisu. Datum pak nelze v aplikaci měnit.

V sekci '**Produkt**' opět můžeme provádět všechny operace, přidávat, mazat, upravovat. Tabulka obsahuje všechny produkty, které obchod prodává. Tlačítkem 'Aktualizovat' lze znovu načíst data. Toto se použije v případě, že jste naimportovali nové produkty, které se ještě nezobrazili. Je možné přidat produkt s kategorií z listu nebo s novou kategorií. Aby se tak stalo, je nutno zaškrtnout políčko 'Použít novou kategorii' a pole 'Nová kategorie' musí mít název nové kategorie. Nová kategorie se ihned zobrazí v listu kategorií. Tlačítkem 'Upravit kategorii' lze změnit kategorii produktu na kategorii zvolenou v listu kategorií, nebo napsanou v poli 'Nová kategorie', opět musí být zaškrtnuto políčko 'Použít novou kategorii'.

Sekce 'Zákazníci' umožňuje pouze úpravu a mazání již popisovaným způsobem.

V sekci 'Objednávka' je možné pouze číst data. Nelze je měnit, mazat nebo přidávat. Klikáním na jednotlivé objednávky se v levé tabulce zobrazí seznam produktů ve zvolené objednávce.

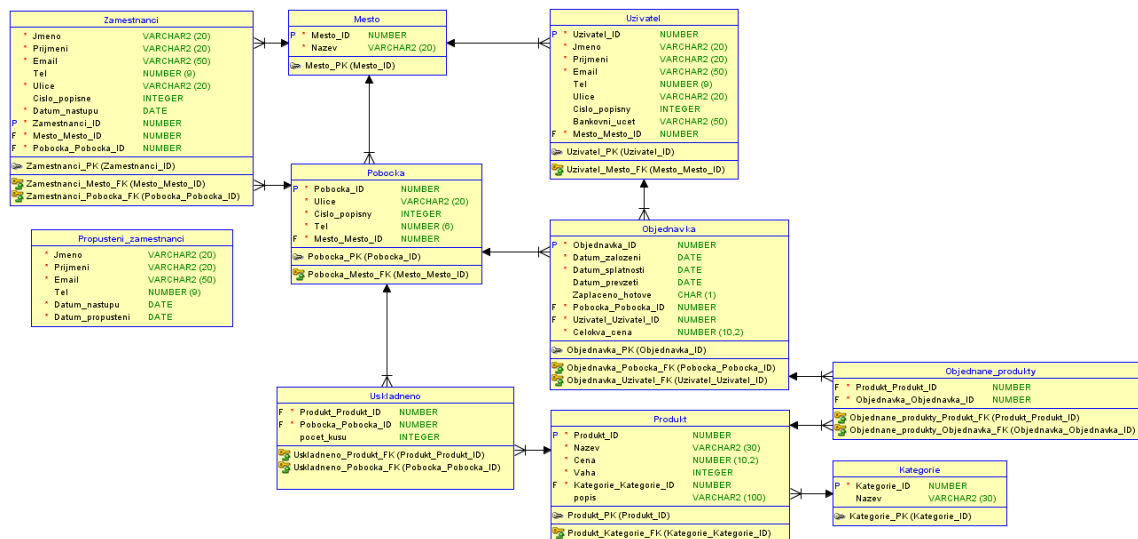
Sekce 'Pobočka' funguje podobně jako sekce 'Objednávka'. Nelze data přidávat, mazat nebo měnit. Po kliknutí na některé políčko vámi zvolené pobočky se zobrazí produkty, které má na skladě společně s počtem. Záporný počet reprezentuje nedostatek již objednaných produktů, které musí být dovezeny. Pro dovážení je tu funkce. Zvolíme pobočku odkud se má vyvézt a v levé tabulce zvolíme produkt, který se má vyvézt. V poli 'Počet pro přesun' zvolíme počet který se má vyvézt, do pole 'ID pobočky, kam se má item přesunout' dáme ID pobočky, kam chceme produkt převézt. ID poboček jsou vidět v pravé tabulce v levém sloupci. Potom je potvrdíme tlačítkem 'Přesunout'. Pobočka v systému reprezentuje obchod se skladem, kde si zákazníci mohou zboží vyzvednout.

Import

Sekce 'Import' umožní zobrazit data o nových produktech v soubor. Po kontrole dat je potom můžeme nahrát do databáze. Pokud produkt má kategorii, která ještě není v databázi, tak se tam sama vloží. Soubor pro import musí být uložen v hlavním adresáři programu, společně s Alpha3.sln, doc, img, atd. Pro ukázkou je tam uložen soubor Produkt.xml. Název souboru pro import lze změnit v Alpha3\App.config.

Technická část

Struktura Databáze



Entity a atributy jsou popsány v dokumentaci databáze. Veškeré informace jsou v /sql složce. Diagram lze nalézt v \img složce.

Struktura tříd

Každý prvek, se kterým se v aplikaci pracuje, má svojí vlastní třídu, k této třídě je i DAO a formulář. V aplikaci se např. pracuje se zaměstnanci. Je tu třída Customer, která definuje, jak zákazník bude vypadat. K této třídě je DAO, jenž obsahuje pouze read metody. Ostatní CRUD metody, jako update nebo delete, jsou nad samotnými objekty, které DAO vytvořilo. Takže Customer má update, create, delete přímo v sobě. Jak na sebe třídy navazují lze nalézt v \img složce.

Návrhové vzory

Pro tahání dat z databáze používám třídy DAO. Každá tabulka, ze které potřebuji tahat data, kromě vazebních, má svou DAO třídu. Tabulka 'uskladneno' a 'objednane_produkty' nemají DAO třídu. Metody read dostávají jen ty nejdůležitější data, jako jméno nebo email. Např. bydliště nebo popis produktu bylo vynecháno. K třídám DAO potom existují třídy, ze kterých se vytváří objekty s daty od DAO třídy. Např. ProduktDAO, která má pouze read metody, a k ní je Produkt třída s metody update, create a delete.

Pro připojení k databázi se používá singleton.

Formuláře

Formuláře slouží jako hlavní prvek pro interakci s uživatelem. Formuláře se skládají z více ovládacích prvků, které spouštějí různé eventy. Na pozadí každého formuláře je script, který spojuje formulář s jinou částí programu. Např. při načtení formuláře se spustí event, který zavolá read metodu z DAO třídy. Aplikace má jeden hlavní formulář, který má panel, ve kterém se střídají jiné formuláře podle volby uživatele

Konfigurace

Konfigurační soubor se nachází v /Alpha3/Alpha3/App.config. V appSettings se dá změnit přihlašovací údaje k databázi a umístění a název souboru pro import. Importovat se mohou pouze nové produkty.

HW a SW požadavky

Pro spuštění aplikace je nutný MySQL server, kam importovat databázi, a MySQL Workbench pro provedení importu.

Chybové hlášky

Aplikace neobsahuje kódy chyb. Vždy, když se něco nepovede, vyskočí error okno. Okno obsahuje informace o tom, co se nepovedlo. V případě vložení moc krátkého jména vyskočí okno se zprávou, že jméno je moc krátké. V případě, že vznikne chyba na straně databáze, program opět vyhodí chybovou zprávu, že se něco selhalo. Nejpravděpodobnější příčina chyby jsou špatná vstupní data o nových zápisech.

Závěr

Výsledek práce odpovídá stanovenému předpokladu. Když bych práci mohl rozšířit, tak bych se podíval na design aplikace a změnil bych způsob hlášení chyb. Zavedl bych kód chyby lepší zpětnou vazbu od programu, co přesně se nepovedlo a jak to opravit.

Projekt byl vytvořen během studia na SPŠE Ječná pro předmět PV - Programové vybavení ve 4. ročníku jako školní projekt.