



Omega

Závěrečná práce 4. ročníku

Radek Vinařský

Zadání práce

Zpracujte vlastní originální softwarový projekt, který bude splňovat níže uvedená kritéria:

- Software musí být spustitelný na jednom, nebo více školních PC v učebně, kde se vyučuje předmět PV, nebo na jiném školním zařízení v jiné učebně po dohodě s vyučujícím.
- Software musí mít smysluplné reálné použití.
- Vaše autorské zdrojové kódy musí souhrnně obsahovat minimálně 2500 (slovy dva a půl tisíce) řádků kódu ve Vámi zvolených programovacích a skriptovacích jazycích. Měření proběhne programem CLOC (<http://cloc.sourceforge.net/>).
- Vaše autorské zdrojové kódy nesmí obsahovat více než 300 řádků zdrojového kódu/skriptu v jednom souboru. Měření proběhne programem CLOC s konfigurační volbou --by-file (<http://cloc.sourceforge.net/>).
- Veškerý zdrojový kód, který není autorský (tj. Vámi vytvořený) musí být v samostatné složce, která bude vhodně nazvaná, například *lib*, nebo *vendor*. Nelze tedy do jednoho souboru umístit jak Váš autorský kód tak i cizí kód.
- Software musí obsahovat alespoň jednu knihovnu třetí strany, nebo cizí zdrojový kód, který bude vždy v samostatné složce viz. předchozí bod.

Zvolené téma

Okenní aplikace usnadňuje správu různých oblastí v hotelu, jako jsou například rezervace pokojů, správa zaměstnanců, správa zákaznických účtů. Důležitými funkcemi, které by měla aplikace pro hotel management obsahovat, jsou:

- Snadná správa pokojů, včetně jejich typů, cen a dostupnosti
- Možnost správy zákaznických účtů
- Možnost správy zaměstnanců
- Možnost vytvářet nové rezervace
- Přehled o rezervacích

Důležitou součástí aplikace pro hotel management je také uživatelské rozhraní, které by mělo být intuitivní a snadno ovladatelné pro jakéhokoli uživatele. Aplikace taky musí být bezpečná a chránit osobní údaje zákazníků a zaměstnanců.

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
3. Technická část - třídy, vazby, databáze, připojení, error handling
 - a. Struktura Databáze
 - b. Struktura tříd
 - c. Formuláře
 - d. Konfigurace
 - e. HW a SW požadavky
 - f. Chybové hlášky
4. 4. Závě

Úvod

Projektem je okenní aplikace v C#. Je to aplikace určená pro správu hotelu vyvinuta ve Visual studio za pomoci .net5 frameworku. 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 sekcemi, s jejíž daty uživatel může manipulovat. Tyto sekce jsou přehled, nové rezervace, uživatel, zaměstnanec, hosté a pokoje. 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 umožňuje všechny operace.

Přidávání

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

Úprava

Úprava dat se provede jednoduchým kliknutím 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 pro smazání.

Části UI

Sekce "Přehled" poskytuje uživateli přehled o aktuálních a nadcházejících rezervacích, zahrnující informace o hostech, kteří jsou v nich zapojeni.

Sekce "Nová rezervace" umožňuje uživateli vytvářet nové rezervace, včetně určení data začátku a konce rezervace, přidání hostů do rezervace a přiřazení pokoje. Automaticky se zde také vypočítává cena rezervace.

Sekce "Uživatel" umožňuje přihlášenému uživateli manipulovat s jeho vlastními údaji, jako jsou například osobní údaje a heslo.

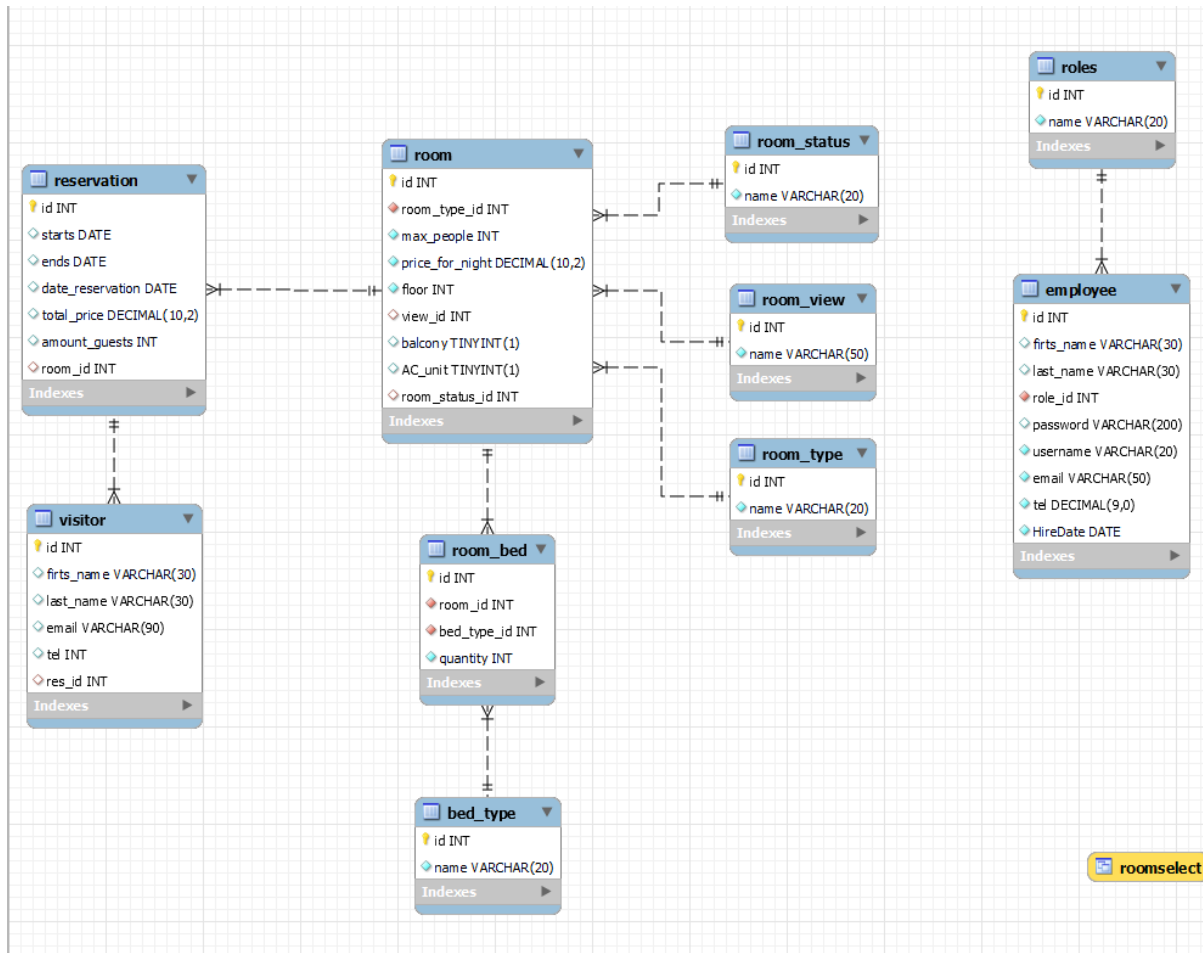
Sekce "Zaměstnanci" je chráněná vyšší úrovní oprávnění a umožňuje vytvářet, mazat a upravovat údaje zaměstnanců.

Sekce "Hosté" je podobná sekci "Zaměstnanci", ale je zaměřena na hosty, kteří mají odlišné údaje a oprávnění.

Sekce "Pokoje" umožňuje uživateli filtrovat pokoje podle svých potřeb a změnit stav pokojů. Zde také může vidět informace o postelích, které jsou v pokoji k dispozici.

Technická část

Struktura databáze



Seznam entit:

- bed_type: obsahuje typy postelí, které jsou k dispozici v pokojích
- employee: obsahuje informace o zaměstnancích
- reservation: obsahuje informace o rezervacích pokojů
- roles: obsahuje seznam rolí, které zaměstnanci mohou mít
- room: obsahuje informace o jednotlivých pokojích, které jsou k dispozici
- room_bed: obsahuje informace o postelích v jednotlivých pokojích

Tabulka bed_type:

- id: unikátní identifikátor typu postele
- name: typu postele

Tabulka employee:

- id: unikátní identifikátor zaměstnance
- first_name: jméno zaměstnance
- last_name: příjmení zaměstnance
- role_id: identifikátor role zaměstnance, odkazuje se na ID v tabulce roles
- password: heslo zaměstnance
- username: uživatelské jméno zaměstnance
- email: emailová adresa zaměstnance
- tel: telefonní číslo zaměstnance
- HireDate: datum nástupu zaměstnance

Tabulka reservation:

- id: unikátní identifikátor rezervace
- starts: datum začátku pobytu
- ends: datum konce pobytu
- date_reservation: datum kdy byla rezervace vytvořena
- total_price: celková cena pobytu
- amount_guests: počet hostů v rezervaci
- room_id: identifikátor pokoje, odkazuje se na ID v tabulce room

Tabulka roles:

- id: unikátní identifikátor role
- name: název role

Tabulka room:

- id: unikátní identifikátor pokoje
- room_type_id: identifikátor typu pokoje, odkazuje se na tabulku room_type
- max_people: maximální počet hostů na pokoji
- price_for_night: cena za noc na pokoji
- floor: patro, na kterém se pokoj nachází
- view_id: identifikátor výhledu z pokoje, odkazuje se na tabulku room_view
- balcony: boolean hodnota, zda má pokoj balkon nebo ne
- AC_unit: boolean hodnota, zda má pokoj klimatizaci nebo ne
- room_status_id: identifikátor stavu pokoje, odkazuje se na tabulku room_status

Tabulka room_bed:

- id: unikátní identifikátor postele v pokoji
- room_id: identifikátor pokoje, odkazuje se na tabulku room
- bed_type_id: identifikátor typu postele, odkazuje se na tabulku bed_type
- quantity: počet postelí daného typu v pokoji.

Databáze obsahuje jeden pohled, pro vypsání pokojů

Struktura tříd

Prvky, 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 User, 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 je vidět v \img složce.

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 Omega/App.config. V appSettings se dá změnit přihlašovací údaje k databázi a umístění a název souboru pro logy.

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 hodnota je neplatná. V případě, že vznikne chyba na straně databáze, program opět vyhodí chybovou zprávu, že něco selhalo. Nejpravděpodobnější příčina chyby jsou špatná vstupní data o nových zápisech.

Závěr

Výsledek práce zcela neodpovídá stanovenému předpokladu. Když bych práci mohl rozšířit, tak bych se podíval na design aplikace a zavedl bych správu skladových zásob, fakturování, check-in, check-out a evidence práce zaměstnanců.

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