STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ

**TRACK TRAVELLER**

DESKTOPOVÁ APLIKÁCIA

**ANDER ABDULLAEV**

2024

**track traveller**

DESKTOPOVÁ APLIKÁCIA

OBHAJOBA VLASTNÉHO PROJEKTU

**ANDER ABDULLAEV**

Stredná priemyselná škola elektrotechnická, Hálova 16, Bratislava

Študijný odbor: 2561 M informačné a sieťové technológie

Konzultant: RNDr. Miloš Vojtek

Dátum odovzdania: 27. marca 2024

Bratislava 2024

Obsah

[Obsah 3](#_Toc161560605)

[Úvod 4](#_Toc161560606)

[1 Teoretická časť 6](#_Toc161560607)

[1.1 Programovacie jazyky použité v programe 6](#_Toc161560608)

[1.2 Použité programové vybavenie v práci 8](#_Toc161560609)

[1.3 Použité technológie v práci 10](#_Toc161560610)

[1.4 Porovnanie s inými produktmi 11](#_Toc161560611)

[2 Praktická časť 12](#_Toc161560612)

[2.1 Vytvorenie databázy a tabuliek 12](#_Toc161560613)

[2.2 Registračný formulár 13](#_Toc161560614)

[14](#_Toc161560615)

[2.3 Prihlasovací formulár a zmena hesla 16](#_Toc161560616)

[2.4 Nákup lístka 18](#_Toc161560617)

[2.5 Radenie vlaku 20](#_Toc161560618)

[2.6 Správa užívateľov a vlakov 22](#_Toc161560619)

[3 Záver 25](#_Toc161560620)

[Zoznam použitej literatúry 26](#_Toc161560621)

[Prílohy 27](#_Toc161560622)

[Príloha A – USB médium 27](#_Toc161560623)

Úvod

Desktopová aplikácia s názvom Track Traveller sa zaoberá kúpou lístka na vlak, vyhľadávaním radenia súpravy vlaku a správou užívateľov a vlakov. Je zameraná na intiutívne užívateľské rozhranie, aby sa táto aplikácia jednoducho a pohodlne používala.

Aplikácia je implementovaná v jazyku C Sharp vo vývojárskom prostredí Microsoft Visual Studio, v ktorom je implementovaná logika aplikácie a komunikácia s databázou a tabuľkami. Využíva grafické užívateľské rozhranie s wireframom a funkcionalitou pomocou technológie Windows Forms s dôrazom na vytvorenie prehľadného a užívateľsky dobrého prostredia. Databázy a tabuľky sú vytvorené v programe Microsoft SQL Server, v nich sú ukladané dáta a sú definované štruktúry tabuliek. V backend časti projektu je použitá knižnica ADO.NET, ktorá umožňuje efektívne spravovať prepojenie a manipuláciu s dátami.

Železnice sú jedným z druhov dopravy, a sú dôležitou súčasťou nášho života. Predstavujú dôležitú súčasť ekonomiky a spoločnosti, pokroky v technológiach, inováciach, rýchle presúvanie ľudí z jedného miesta na druhé. Jej vysoké užívanie dokáže odľahčiť dopravnú situáciu na cestách a diaľniciach. Je bezpečnejšia a spoľahlivejšia ako ostatné druhy dopravy. Preto by sme železnice nemali zanedbávať, ale mali by sme do nich pravidelne investovať. A treba rozvíjať aj online služby a aplikácie železníc.

Cieľom projektu je hlavne vytvoriť pre užívateľa pohodlný nákup lístka na vlak. Pre vlakových nadšencov je tu vyhľadanie radenia súpravy na základe čísla vlaku. Ďalším cieľom je motivovať užívateľov k tomu, aby využívali a dôverovali online službám železníc. Väčšie využívanie online služieb dokáže priniesť ekonomické prínosy pre železničnú spoločnosť v podobe odstránenia administratívnej záťaže pre zamestnancov pokladníc. Týmto môže dôjsť aj k zefektívneniu počtu pokladníc, a ich nahradenie použiteľnými automatmi na tých staniciach, kde je to dostatočné.

Už 5 rokov sa aktívne zaujímam o železnice, rušne a o vlakové spoje. Jazdím vlakmi v pravidelných intervaloch, fascinujú ma výlety vlakmi. Aktívne si pozerám radenia súprav vlakových spojov na stránkach vlakových fanúšikov. Je to jeden z hlavných dôvodov, prečo ako projekt robím desktopovú aplikáciu s názvom Track Traveller. Zaujímalo ma tiež to, ako funguje, a ako je naprogramovaná rezervácia miest na konkrétny vozeň a kúpa lístkov cez aplikáciu. Keďže som ešte nikdy neprogramoval

vlakovú aplikáciu, a Železničná spoločnosť Slovensko nemá v ponuke vyhľadanie radenia súpravy na základe čísla, tak som sa rozhodol, že moja maturitná práca bude zameraná na vlaky. Moja snaha spočíva v tom, aby aj u nás na oficiálnej stránke dopravcu bolo vyhľadávanie radenia súpravy, nielen na stránkach vlakových fanúšikov. Tiež, aby sa železničná doprava využívala vo väčšom rozsahu. Pri radení súpravy po zadaní čísla som sa inšpiroval stránkou vlakového dopravcu České dráhy. Očakávam, že pri spracovaní maturitnej práce sa naučím, ako sa robia aplikácie väčšieho rozsahu. Tieto nadobudnuté skúsenosti sa mi môžu stať v budúcnosti užitočnými pri plánovaní projektov alebo aplikácií pre vlakových ako aj ostatných dopravcov.

Táto práca pozostáva z dvoch častí. V prvej časti práce sú podrobne popísané technologické základy a použité technológie v aplikácii. Zaoberá sa ich využitím, výhodami, prínosmi a programovým vybavením na tvorbu tejto aplikácie. V tejto časti je uvedené, čo treba k práci na tejto aplikácii. Popísané sú v nej programovacie jazyky C Sharp a SQL, programy Microsoft Visual Studio a Microsoft SQL Server, a technológie Windows Forms a ADO.NET.

Druhá časť práce je zameraná na praktickú realizáciu desktopovej aplikácie. Táto časť ukazuje, ako bola naprogramovaná aplikácia a metódy v programovacom kóde, a tvorbu tabuliek a databázy, a ich naplňovanie údajmi o účtoch a vlakoch. Obsahuje ukážky, ako je vytvorený a umiestnený wireframe v častiach aplikácie. Ukazuje ako sa dokážem zaregistrovať, prihlásiť sa, vykonať kúpu lístka, spravovať užívateľov a vlakov a vyhľadať radenie vlaku prostredníctvom grafického užívateľského prostredia. Tiež ukazuje prepojenie s konkrétnou databázou, pohyb z jednej časti aplikácie na druhú pomocou tlačidiel.

Vyhlasujem, že som túto komplexnú odbornú prácu napísal samostatne pod odborným vedením konzultanta práce a použil iba uvedenú literatúru.

1. Teoretická časť

V rámci tohto projektu som sa spoliehal na viacero programovacích jazykov a nástrojov, ktoré mi umožnili spraviť desktopovú aplikáciu.

1. Programovacie jazyky použité v programe

**C Sharp**, často skracovaný ako C# je objektovo orientovaný, vysoko úrovňový a moderný programovací jazyk vyvinutý spoločnosťou Microsoft. Široko sa používa v spojení s .NET Framework, ktorý poskytuje bohatú knižnicu tried a nástrojov pre efektívny vývoj aplikácií. C Sharp vychádza z programovacích jazykov C a C++, čo umožňuje programátorom, ktorí ovládajú tieto programovacie jazyky, využiť ich skúsenosti. Táto vlastnosť sa volá interoperabilita.

C# má veľku výhodu v tom, že má dlhú podporu. Vyvíja ho spoločnosť Microsoft a sú pravidelne vydávané verzie, ktoré aktualizujú funkcie jazyka. To znamená, že C# sa bude využívať aj v budúcnosti a teda istotu pre rôzne programy a aplikácie naprogramované v tomto jazyku. Jazyk bol veľakrát testovaný, vďaka tomu existuje spoľah, že je aj bezpečný. Za tú dobu si jazyk C# obľúbilo množstvo programátorov, ktorí vedia tento jazyk.

Existuje množstvo kníh, videí, návodov o tomto jazyku, takže je ľahké si zaobstarať materiály o C#. Tiež existuje rozpis o tomto jazyku priamo od Microsoftu, veľmi dôkladne spracovaný, ktorý môže začiatočníkom pomôcť. Vo svete je navyše veľa programátorov, ktorí pomáhajú ostatným sa učiť tento jazyk.

C Sharp podporuje deklarácie premenných s rôznymi dátovými typmi, ako sú var, int, double, bool, char a string. Podporuje tiež tiež polia v tvare new[] s inicializátorom kolekcie. Dátový typ int reprezentuje celé čislo bez desatinných miest, zatiaľ čo double reprezentuje desatinné číslo s bodkou. Dátový typ bool vracia hodnotu true alebo false v závislosti od podmienky. Char predstavuje jeden znak alebo písmeno a string je reťazec znakov alebo viet. C# je typovo bezpečnejší ako C++. Neumožňuje globálne premenné alebo metódy, musia byť deklarované v triedach.

Mal som na výber jazyky PHP a C Sharp. V mojom projekte som si vybral programovanie v C Sharp z dôvodu, že s týmto programovacím jazykom mám najviac skúseností a je to pre mňa najznámejší programovací jazyk. Tiež som zobrali tento jazyk ako vhodný pre robenie desktopových aplikácií. Nechcel som webovú aplikáciu.

**Structured Query Language** – je počítačový jazyk, ktorý slúži na manipuláciu, správu a organizovanie dát, ktoré sú uložené v databázových systémoch. Využívajú ho ako aj užívatelia, tak aj vývojári aplikácií. Vďaka nemu dokážeme definovať dáta, umožňuje riadenie prístupu k dátam. Používa sa buď užívateľsky alebo interaktívne k riešeniu úloh, umožňuje získať odpovede takmer okamžite. Má množinový prístup k dátam, dáta chápe v podobe tabuliek, pracuje s relačnými databázami. Každá tabuľka je množinou dát, ktorá je reprezentovaná stĺpcami a riadkami.

Zahŕňa v sebe dva kľúčové pojmy: Data Definition Language – používa sa na definovanie dát, štruktúry tabuliek, vytváranie nových tabuliek a databáz. Druhým je Data Manipulation Language – umožňuje vypĺňanie stĺpcov tabuliek údajmi, aktualizáciu a odstraňovanie údajov, a definovať vzťahy a organizáciu medzi zložkami. Používajú nasledovné príkazy, ktoré sú popísané v nasledovných tabuľkách:

**Tab. 1 Príkazy pre Data Definition Language**

|  |  |
| --- | --- |
| ALTER | Mení štruktúru tabuľky |
| CREATE | Vytvára novú databázu alebo tabuľku |
| DROP | Odstraňuje existujúcu tabuľku |
| TRUNCATE | Vytvára nový index |
| COMMENT | Mení index |

**Tab. 2 Príkazy pre Data Manipulation Language**

|  |  |
| --- | --- |
| SELECT | Získava dáta z tabuľky databázy |
| INSERT | Vloženie alebo pridanie dát do tabuľky |
| UPDATE | Úprava existujúcich údajov v tabuľke |
| DELETE | Odstránenie údajov v tabuľke |

Tento jazyk umožňuje nekomplikovaný priebeh v prípade, že prístup k dátam majú viacerí užívatelia. Je to interpretačný jazyk, čo znamená, že príkazy je možné preložiť do samostatne spustiteľného kódu. Nemá však riadiace programovacie konštrukcie (if else, for, while), a preto nie je považovaný za plnohodnotný programovací jazyk, akými sú C, C Sharp, C++. Každá tabuľka by mala mať primárny kľúč, ten slúži k identifikácii a tá istá hodnota v danom stĺpci sa môže vyskytnúť iba raz. Tiež každý stĺpec má dátový typ, ako sú varchar, int, datetime. Varchar predstavuje dĺžku slova alebo vety, int predstavuje číselnú hodnotu a datetime predstavuje dátum a čas.

V tomto prípade som nemal výber jazyka, hneď som začal so SQL. Bez tohto jazyka by som nemohol používať databázy a ukladať do nich údaje.

1. Použité programové vybavenie v práci

**Microsoft Visual Studio –** je bezplatný vývojársky program od spoločnosti Microsoft, ktorý programátori využívajú na vytváranie webových, desktopových aplikácií, webových stránok a služieb. Sú rôzne verzie, Visual Studio Professional je určený pre menšie podniky, a verzia Enterprise je vhodná pre veľké organizácie. Visual Studio má integrované vývojové prostredie (IDE), umožňuje vytvárať, písať upravovať a zostavovať programovací kód. Obsahuje nastavenia a nástroje na pomoc pri zostavovaní kódu, návrhár formulárov pre webové a GUI aplikácie. Tento program podporuje rôzne programovacie jazyky – hlavne C++ a C#, Python, JavaScript, XAML, TypeScript a ďalšie. Využíva platformy na vývoj softvéru, napríklad Windows Forms, Windows API, Windows Presentation Foundation. Obsahuje editor kódu, ktorý zvýrazňuje syntax pomocou IntelliSense pre premenné, metódy, cykly, podmienky a funkcie. IntelliSense poskytuje automatické dopĺňanie kódu a nápovedu v editore, výrazne uľahčuje prácu programátorom.

Na obrázku 1 je znázornený editor kódu. Konkrétne je to konzolová aplikácia, výhodou je, že pri vytvorení aplikácie sa vygenerujú základy kódu ako sú knižnice, namespace, základná trieda Program a v konzolovej aplikácii statická nenávratová metóda Main. Je to výhoda, keďže to netreba nastavovať ručne, ako je to napríklad v CodeBlocks. Ďalej sú tam samozrejme funkcionality ako Solution Explorer, Properties, vo Windows Forms aj Toolbox, z ktorého sa vyberajú grafické prvky. A zelená šípka obsahuje možnosť kompilovania a spustenia.

Obrázok, na ktorom je snímka obrazovky, softvér, text, multimediálny softvér

Automaticky generovaný popis

**Obr. 1 Editor kódu Microsoft Visual Studio**

Microsoft Visual Studio som si vybral ako vývojársky program z dôvodov, že s týmto mám najviac skúseností a považujem ho za prijateľný program pre prácu s projektom. Tiež preto, lebo podporuje programovací jazyk C# a má integráciu s Microsoft technológiami.

Je porovnateľný s ostatným vývojovým prostredím, ako je napríklad Xcode, lenže sú medzi nimi aj rozdiely. Xcode je zameraný na pre iOS a macOS a má integráciu s App Store, zatiaľ čo Visual Studio hlavne pre Windows. Xcode používa minimalistické rozhranie s dôrazom na jednoduchosť a podporuje programovacie jazyky Swift a Objective-C, zatiaľ čo Visual Studio používa komplexnejšie prostredie a podporuje hlavne C#.

**Microsoft SQL Server** – Je to systém správy relačných databáz (RDBMS), tie používajú tabuľky na ukladanie dát. Každá tabuľka má svoje vlastné meno a definíciu štruktúry. Aplikácie sa pripájajú k inštancii alebo k databáze SQL Server a komunikujú pomocou Transact-SQL (T-SQL). Ten poskytuje rozšírenia a doplnky k SQL, a umožňuje lepšiu integráciu s technológiami od spoločnosti Microsoft. SQL Server je v 4 edíciach, prvá je Developer, ďalej Express, Enterprise a Standard.

Ako integrované prostredie sa používa SQL Server Management Studio (v skratne SSMS). Pomocou neho je možné konfigurovať, spravovať a pristupovať ku všetkým častiam serveru.

Využíva sa na operačných systémoch Windows a Linux, to znamená, že na macOS program nepobeží.

Tento program som použili v našom projekte pre možnosť uchovávania údajov v databáze. Mal som na výber programy databáz ako sú Oracle, MySQL a Microsoft SQL Server. Vybral som si práve ten Microsoft kvôli skúsenostiam s týmto programom a zdal sa mi najužitočnejší.

1. Použité technológie v práci

**Windows Forms** – je to UI framework, ktorý umožňuje jednoduchú tvorbu Windows desktopových aplikácií pomocou grafického uživateľského prostredia (GUI). Poskytuje jeden zo spôsobov produktívneho vytvárania aplikácií vo vývojárskom prostredí Microsoft Visual Studio. Vizuálne ovládacie prvky, ktoré sa dajú premiestňovať myšou, rôzne spôsoby vytvárania dizajnu uľahčujú vytváranie aplikácií. Vďaka tejto technológii bolo vytvorených množstvo desktopových aplikácií, ktoré je možné aktualizovať, zmeniť a tak ďalej. Tie môžu pristupovať k lokálnemu hardvéru na počítači, na ktorom je daná aplikácia spustená.

Dôležitou súčasťou Windows Forms je vizuálna plocha, na ktorej sa zobrazujú vizuálne ovládacie prvky, dizajn a informácie používateľovi. Pri práci s aplikáciou sa pridávajú ovládacie prvky, tie sa nachádzajú v zozname zvanom Toolbox. Podporuje tiež vytváranie vlastných prvkov pomocou triedy UserControl. Prvky buď zobrazujú alebo akceptujú vstup údajov. Pokiaľ používateľ vykoná niečo s ovládacím prvkom tak sa vygeneruj udalosť.

Na obrázku 2 je vidno rozmiestnené prvky ako sú button naľavo. Button vygeneruje nejakú udalosť na základe stlačenia tlačidla. V strede je textbox, do neho sa zapisuje text či už číslicami alebo písmenami, je znázornený ako string. Napravo je checkBox, ktorý je niečo ako bool – pokiaľ je fajka, tak je stav true, pokiaľ nie je, tak je stav false. To sú iba jedny z mála prvkov, ktoré môžu byť zahrnuté vo Windows Forms.

Obrázok, na ktorom je elektronika, text, snímka obrazovky, displej

Automaticky generovaný popis

**Obr. 2 Windows Forms**

**ADO.NET** – technológia, ktorá poskytuje prístup k zdrojom údajom ako napríklad SQL Server, Oracle a XML. Aplikácie zdieľajúce dáta môžu využiť ADO.NET na pripojenie k databázovému serveru a tým pádom získavať, aktualizovať a mazať dáta. Pozostáva z riadených tried, tie umožňujú .NET aplikáciam sa pripojiť k serveru. Poskytuje širokú škálu funkcií, ako je pripojený a odpojený režim, v pripojenom režime sú to živé dáta a v odpojenom režime sú dáta poskytované z dátového úložiska. Je navrhnutý pre riešenie dvoch spôsobov práce s údajmi. Prvý je pristupovať raz k údajom a ukladať údaje za behu. Druhý je ten, že vždy ideme do databázy, aby sme údaje získali.

1. Porovnanie s inými produktmi

Prvým významným rozdielom oproti iným podobným produktom je to, že Track Traveller je navrhnutý na offline používanie, čo znamená, že funguje aj bez pripojenia na internet. To umožňuje použiť tento produkt kedykoľvek a tiež poskytuje viac flexibility. Táto možnosť umožňuje vyniknúť medzi online aplikáciami, pretože online aplikácie vyžadujú pripojenie na internet.

Druhým významným rozdielom je to, že tento produkt je naprogramovaný v programovacom jazyku C#, zatiaľ čo mnohé podobné produkty sú naprogramované vo webových technológiach, ako sú HTML, PHP a CSS. Jazyk C# pomúka pre projekt Track Traveller výhody čo sa týka výkonnosti a bezpečnosti. Keďže využíva hadrvérové zdroje, tak sa rýchlejšie načítavajú dáta pre používateľov, než pri webových produktoch a nezávisí to od internetového pripojenia.

Ďalším významným rozdielom je kompabilita s operačnými systémami. Keďže Microsoft SQL Server beží iba na operačnom systéme Windows alebo Linux, to znamená, že na operačnom systéme MacOS nepobeží. To môže byť nevýhodou pre tých, ktorí používajú iMac alebo Macbook.

1. Praktická časť
2. Vytvorenie databázy a tabuliek

V tomto programe som v programe Microsoft SQL Server vytvoril databázu s názvom SpravaVlakov a v nej tri tabuľky, prvá je dbo.ZoznamUctov, druhá je dbo.ZoznamVlakov a tretia je ZoznamListky. Tabuľka ZoznamUctov slúži na uchovávanie údajov o užívateľoch a následne na prístup daných údajov v desktopovej aplikácii. Má 5 stĺpcov. Tabuľka ZoznamVlakov slúži na uchovávanie údajov o vlakoch a tiež na prístup. Má 11 stĺpcov. Tabuľka ZoznamListkov slúži na záznam o kúpených lístkoch. V programe je inštancia SQL Serveru **DESKTOP-HV06JV3\SQLEXPRESS, kde ľavá časť od opačného lomítka znamená názov počítača, na ktorom beží server.**

Na obrázku 3 sme vytvorili databázový diagram, ktorý zahŕňa všetky tri tabuľky. Na tomto obrázku sú názvy tabuliek, štruktúra tabuliek a ich stĺpce. Diagram má názov dbo.Diagram.

Obrázok, na ktorom je text, snímka obrazovky, číslo, písmo

Automaticky generovaný popis

**Obr 3. Databázový diagram databázy SpravaVlakov**

Je prvý spôsob, ako som si vytvoril tabuľku. Spočíva v príkaze CREATE, ktorý som uviedol v podkapitole 1.1. Zahrnul som tam názvy stĺpcov a dátové typy. Nvarchar(50) je ako string s maximálne 50 znakmi. Uvádzam príklad, ako som vytvoril tabuľku ZoznamUzivatelov:

CREATE TABLE ZoznamUzivatelov (

meno nvarchar(50),

priezvisko nvarchar(50),

email nvarchar(50),

heslo nvarchar(50),

pheslo nvarchar(50)

);

Pri vytváraní databáz, diagramov a tabuliek je tiež jednoduchší spôsob akým som ich vytváral. Naľavo v Object Explorer pri zložke „Databases“ som klikol na „New Database“ a nastavil som mu názov a tak som vytvoril. Podobne pri tabuľkách v zložke „Tables“. Pri diagramoch som klikol na „Database Diagrams“ že chcem nový, a automaticky mi ponúklo zoznam tabuliek, a po stlačení sa do diagramu automaticky pridali.

Pri vytváraní tabuliek som narazil na menší problém, bol v tom, že som použil pomlčník v názvoch stĺpcoch, napríklad (e-mail). To som ale vyriešil ľahko, keď som odstránil spojovníky, tak spojenie s tabuľkami bolo bez problémov.

1. Registračný formulár

Ako prvý Form v rámci aplikácie som vytvoril s názvom Registracia.cs. Zahŕňa vypĺňanie údajov uživateľa, ako sú meno, priezvisko, e-mail, heslo a potvrdzujúce heslo. Po stlačení tlačidla sa tieto údaje odošlú do tabuľky ZoznamUzivatelov. V rámci ochrany sme tiež nakódovali šifrovanie hesla pomocou náhodného generovania.

Na obrázku 4 som pridal prvky z Toolboxu pomocou myši, a týmto spôsobom som zrobil grafickú štrukrúru daného Formu. Naľavo a napravo som pridal 2 panely, kvôli oddeleniu. ozdelil som ich na pomer polovica-polovica. Na ľavej strane som nechal bielu farbu, a pridal som tam BackgroundImage ako logo Železničnej spoločnosti Slovensko. Pravý panel som nastavil na farbu Navy, čo je tmavomodrá farba. V ňom som zahrnul všetky labely, tlačidlá a textboxy. Kvôli dizajnovej príťažlivosti som labely dával nad textboxy, aby užívateľ videl, čo má napísať do konkrétneho textboxu. Hore je label s popisom „Registrácia užívateľského účtu“. Dal som tam tiež jeden linklabel, lebo je to odkaz na prihlasovanie užívateľa v prípade, že je už zaregistrovaný. Napravo od textboxu s heslom je tlačidlo s ikonou oka, slúži na odkrytie hesla od znakov.A samozrejme dole je tlačidlo, ktoré má väčšiu veľkosť.

Obrázok, na ktorom je text, snímka obrazovky, softvér, operačný systém

Automaticky generovaný popis

**Obr. 4 Wireframe Formu Registracia.cs**

Čo sa týka kódu, na začiatok deklarujeme premenné typu string, ktoré obsahujú údaje z daných TextBoxov pre zjednodušenie zápisu. Nie je to nutné, ale rozhodol som sa kvôli lepšej čitateľnosti.

string pouzivatelskeMeno = register\_meno.Text;

string pouzivatelskePriezvisko = registracia\_priezvisko.Text;

string pouzivatelskyEmail = registracia\_email.Text;

string pouzivatelskeHeslo = registracia\_heslo.Text;

string pouzivatelskeOverenieHesla = registracia\_potvrdenieHesla.Text;

Následne zamedzím registráciu užívateľa v prípade, že nemá vyplnené všetky textové polia pomocou operátora && (AND), podmienky != a vlastnosti triedy String.Empty. Pokiaľ nemá vyplnené všetky textboxy, tak nebude zaregistrovaný a údaje nebudú odoslané do tabuľky.

if(register\_meno.Text != String.Empty && registracia\_priezvisko.Text != String.Empty && registracia\_email.Text != String.Empty && registracia\_heslo.Text != String.Empty && registracia\_potvrdenieHesla.Text != String.Empty)

Blok kódu, ktorý slúži na registráciu užívateľa v prípade, že má vyplnené správne a všetky textboxy som obalil do try-catch pre vychytanie chýb. Najprv som vytvoril string premennú sifrovaneHeslo, v hodnote sa volá funkcia SifrovanieHesla s parametrom pouzivatelskeHeslo. Databáza sa otvorí pomocou metódy Open(). Pomocou príkazu INSERT uvedeného v kapitole 1.1.2 a metódy Add sa údaje vložia do tabuľky. Príkaz ExecuteNonQuery() zabezpečí, že sa dané údaje zobrazia v tabuľke. A tiež po stlačení tlačidla sa presmeruje na Form Prihlasenie.cs. Po presmerovaní sa tento Form zatvorí pomocou príkazu this.Hide(). V prípade, že je tam chyba, tak sa zobrazí MessageBox so správou.

try

{

string sifrovaneHeslo = SifrovanieHesla(pouzivatelskeHeslo);

con.Open();

string vlozenie = "INSERT INTO ZoznamUzivatelov (meno, priezvisko, email, heslo, pheslo) VALUES (@meno, @priezvisko, @email, @heslo, @pheslo)";

SqlCommand insertCommand = new SqlCommand(vlozenie, con);

insertCommand.Parameters.Add("@meno", pouzivatelskeMeno);

insertCommand.Parameters.Add("@priezvisko", pouzivatelskePriezvisko);

insertCommand.Parameters.Add("@email", pouzivatelskyEmail);

insertCommand.Parameters.Add("@heslo", sifrovaneHeslo);

insertCommand.Parameters.Add("@pheslo", sifrovaneHeslo);

insertCommand.ExecuteNonQuery();

MessageBox.Show("Registrácia prebehla úspešne.");

con.Close();

Prihlasenie prihlasenie = new Prihlasenie();

prihlasenie.Show();

this.Hide();

}

catch (Exception)

{

MessageBox.Show("Stala sa nejaká chyba pri procese registrácie účtu.");

}

Pre šifrovanie hesla som vytvoril metódu SifrovanieHesla s modifikátorom private a parametrom (string heslo).

private string SifrovanieHesla(string heslo)

{

using (var sha256 = SHA256.Create())

{

byte[] zasifrovaneHeslo = sha256.ComputeHash(Encoding.UTF8.GetBytes(heslo));

return Convert.ToBase64String(zasifrovaneHeslo);

}

}

Použili sme kryptografickú funkciu SHA256, konvertoval som heslo na pole bytov pomocou UTF-8 kódovania. Slúži tiež na výpočet hashu. Hash je funkcia, ktorá slúži na transformovanie textu na iný reťazec. Vráti to prevod na Base64 pomocou metódy Convert.ToBase64String. Base64 je spôsob premeny binárnych dát do textovej formy.

Nastavili sme pre Label registracia\_heslo možnosť, aby heslo bolo zakryté a aby sa odkrylo až po stlačení tlačidla. V tejto časti sme narazili na problém, keď sme v Properties nastavili UseSystemPasswordChar na true, tak sa to neodrazilo na reálnom zakrytí hesla. Čiže sme museli zakrytie hesla znakmi naprogramovať ručne v konštruktore, pre tento Label sme nastavili vlastnosť PasswordChar = '\*'.

registracia\_heslo.PasswordChar = '\*';

1. Prihlasovací formulár a zmena hesla

Vytvoril som Form Prihlasenie.cs, ktorého cieľom je možnosť prihlásenia sa pre užívateľa, ktorý sa už registroval, a má v databáze uložené svoje údaje. Po zadaní údajov do TextBoxov a stlačení tlačidla sa užívateľ prihlási.

Pri prihlasovacom formulári sme pridali naľavo Panel s názvom panel2. Nastavili sme ho presne ako vo Forme Registracia.cs. Napravo som pridal Panel s názvom panel1, v ktorom sme pridali nasledovné prvky: hore Panel s popisom „Prihlásenie sa do užívateľského účtu“, ďalej 2x TextBox s názvami prihlasenie\_email a prihlasenie\_heslo a nad nimi Labely s popismi „E-mail“ a „Heslo“. Napravo od TextBoxu s heslom je Button zobrazenie\_hesla, funkciou ktorého je zobraziť heslo. Ďalej LinkLabel zle\_heslo, ktorý odkazuje na zmenu hesla a LinkLabel odkaz\_na\_registraciu, ktorý odkazuje na registráciu. A samozrejme úplne dobe Button prihlasenie\_tlacidlo, ktorý odkazuje na nákup lístka.

**Obrázok, na ktorom je text, snímka obrazovky, softvér, operačný systém

Automaticky generovaný popis**

**Obr. 5 Wireframe Formu Prihlasenie.cs**

Ako v kapitole 2.22 sme deklarovali premenné typu string pre zjednodušenie zápisu.

string prihlasovaciEmail = prihlasenie\_email.Text;

string prihlasovacieHeslo = prihlasenie\_heslo.Text;

if(prihlasenie\_email.Text != String.Empty && prihlasenie\_heslo.Text != String.Empty)

{

try

{

con.Open();

string query = "SELECT COUNT(\*) FROM ZoznamUctov WHERE email = @email AND heslo = @heslo";

SqlCommand prikaz = new SqlCommand(query, con);

prikaz.Parameters.Add("@email", prihlasovaciEmail);

prikaz.Parameters.Add("@heslo", prihlasovacieHeslo);

prikaz.ExecuteScalar();

con.Close();

MessageBox.Show("Prihlásenie bolo úspešné.");

HlavnyPanel hlavna = new HlavnyPanel();

hlavna.Show();

this.Hide();

}

catch (Exception)

{

MessageBox.Show("Stala sa nejaká chyba pri procese prihlasovania účtu.");

}

}

else

MessageBox.Show("Nie sú vyplnené všetky požadované údaje. Treba ich vyplniť.");

1. Nákup lístka

Tento Form s názvom HlavnyPanel.cs zahŕňa nákup lístka na základe zadaného mena, priezviska a čísla vlaku. Keď zadám číslo vlaku, tak sa zobrazia ostatné údaje dole. Nákup lístka potom prebehne cez tlačidlo s popisom „Kúpíť“, údaje sa uložia do tabuľky ZoznamListkov a tlačidlo s popisom „Zobraziť kúpené lístky“ zobrazí na ďalšom Forme danú tabuľku.

Obrázok, na ktorom je text, snímka obrazovky, displej, softvér

Automaticky generovaný popis

**Obr. 6 Wireframe Formu HlavnyPanel.cs**

V rámci Formu HlavnyPanel.cs sme pridali ľavý Panel s názvom panel3. Ten oddeľuje ľavú časť s odhlásením sa a buttonmi, ktoré odkazujú na ostatné časti, a pravú časť, v ktorej prebieha tá kúpa lístka. V panel3 som použil nasledovné prvky: Button s popisom „Správa vlakov a užívateľov“, Button s popisom „Rezervácia miest vo vozni“ a Button s popisom „Radenie vlaku“. Všetky tieto buttony odkazujú na presne tieto časti. Dole som pridal LinkLabel s názvom odhlasit\_sa, po jeho stlačení prebehne presmerovanie na registračný formulár, konkrétne Form Registracia.cs. Hore som dal PictureBox s názvom pictureBox1, v ktorom som pridal tiež logo Železničnej spoločnosti Slovensko, akurát v inom dizajne. Tesne pod ním sme dali Label s popisom „ZSSK“. V tomto paneli som BackColor nastavil na bielu a ForeColor na čiernu práve kvôli dizajnovej príťažlivosti.

Napravo sme pridali Panel s názvom panel2, v ktorom sú zahrnuté Labely, TextBoxy a Buttony ktoré sa týkajú kúpy lístka.

V rámci kódu sme najprv deklarovali premenné typu string pre zjednodušenie zápisu.

string zaciatok = Zaciatok.Text;

string koniec = Koniec.Text;

string cisloVlaku = cislo\_vlaku.Text;

string pociatocnaStanica = poc\_stanica.Text;

string koncovaStanica = kon\_stanica.Text;

string datum = datum\_cas.Text;

string cena = cena\_vlaku.Text;

Vytvorili sme metódu s názvom NajdenieUdajov, má modifikátor public, čiže znamená, že táto metóda je voľne dostupná. Použil som v metóde kolekciu List a v nej dátový typ string. Táto metóda slúži na nájdenie údajov čo sa týka času odchodu, príchodu, počiatočnej a koncovej stanice po zadaní čísla vlaku a stlačení Buttonu kupit.

Vytvorili sme inštanciu s názvom udaje, do ktorej sa následne uchovávajú údaje z databázy. Následne sme použili try-catch blok, pomocou ktorého dokážeme zistiť, či počas procesu vyhľadávania údajov sa nestala nejaká chyba. Pokiaľ áno, zobrazí sa MessageBox s výpisom, že sa stala chyba.

Použijeme príkaz na otvorenie databázy con.Open(), deklarovali sme premennú typu string s názvom vyhladanie, v ktorej je SELECT príkaz na vyhľadanie potrebných údajov kde je údaj CisloVlaku rovný parametru dotazu @cislo.

public List<string> NajdenieUdajov(string cisloVlaku)

{

List<string> udaje = new List<string>();

try

{

con.Open();

string vyhladanie = "SELECT Odchod, Prichod, PociatocnaStanica, KoncovaStanica, CenaVlaku FROM ZoznamVlakov WHERE CisloVlaku = @cislo";

SqlCommand prikaz = new SqlCommand(vyhladanie, con);

using (SqlDataReader reader = prikaz.ExecuteReader())

{

if (reader.Read())

{

udaje.Add(((DateTime)reader["Odchod"]).ToString());

udaje.Add(((DateTime)reader["Prichod"]).ToString());

udaje.Add(reader["PociatocnaStanica"].ToString());

udaje.Add(reader["KoncovaStanica"].ToString());

udaje.Add(((Decimal)reader["CenaVlaku"]).ToString());

con.Close();

}

}

}

catch (Exception)

{

MessageBox.Show("Stala sa nejaká chyba pri procese nákupu lístka.");

}

return udaje;

}

Pre uloženie údajov do tabuľky databázy ZoznamListkov sme vytvorili metódu s názvom UlozenieUdajovDoTabulky, ktorá nemá návratovú hodnotu a ako parameter sme uviedli zoznam reťazcov List s názvom udaje.

public void UlozenieUdajovDoTabulky(List<string> udaje)

{

try

{

con.Open();

string pridanie = "INSERT INTO ZoznamListkov (meno, priezvisko, cisloVlaku, casOdchodu, casPrichodu, datum, cena) VALUES (@meno, @priezvisko, @cislovlaku, @casodchodu, @casprichodu, @datum, @cena)";

SqlCommand cmd = new SqlCommand(pridanie, con);

cmd.Parameters.AddWithValue("@meno", meno.Text);

cmd.Parameters.AddWithValue("@priezvisko", priezvisko.Text);

cmd.Parameters.AddWithValue("@cislovlaku", cislo\_vlaku.Text);

cmd.Parameters.AddWithValue("@casodchodu", Zaciatok.Text);

cmd.Parameters.AddWithValue("@casprichodu", Koniec.Text);

cmd.Parameters.Add(new SqlParameter("@datum", SqlDbType.Date)).Value = Convert.ToString(datum\_cas.Text);

cmd.Parameters.AddWithValue("@cena", udaje[4]);

cmd.ExecuteNonQuery();

con.Close();

}

catch (Exception)

{

MessageBox.Show("Stala sa nejaká chyba pri procese nákupu lístka.");

}

}

1. Radenie vlaku

Tento Form s názvom RadenieVlaku.cs zahŕňa zobrazenie radenia vlaku na základe zadaného čísla vlaku a stlačenia tlačidla.

Obrázok, na ktorom je text, snímka obrazovky, softvér, displej

Automaticky generovaný popis

**Obr. 7 Wireframe a funkčnosť Formu RadenieVlaku.cs**

V tomto kóde sme do metódy button\_vyhladat\_Click dali try-catch bloky na zachytenie chyby. V try bloku je con.Open();, pomocou ktorého sa otvára prístup k databáze. Ďalej je deklarovaná string premenná najdenie, v ktorom je príkaz SELECT, ten som použil na všetky stĺpce tabuľky ZoznamVlakov a nastavili sme ho na základe čísla vlaku v Labeli cisloVlaku. Uviedli sme tam triedu SqlCommand, v nej sme deklarovali premennú cmd a tiež je tam parameter (najdenie, con). Pomocou toho sme vytvorili inštanciu triedy.

private void button\_vyhladat\_Click(object sender, EventArgs e)

{

try

{

con.Open();

string najdenie = "SELECT \* FROM ZoznamVlakov WHERE CisloVlaku = '" + cisloVlaku.Text + "'";

SqlCommand cmd = new SqlCommand(najdenie, con);

using (SqlDataReader citac = cmd.ExecuteReader())

{

if (citac.Read())

label\_radenieVlaku.Text = citac["RadenieVlaku"].ToString();

con.Close();

}

}

catch (Exception)

{

MessageBox.Show("Stala sa nejaká chyba pri zobrazení radenia vlaku.");

}

}

1. Správa užívateľov a vlakov

Obrázok, na ktorom je text, displej, snímka obrazovky, softvér

Automaticky generovaný popisFormy SpravaVlakov.cs a SpravaUzivatelov.cs zahŕňajú automatické zobrazenie údajov z danej tabuľky databázy cez DataGridView. Umožňujú pridanie riadku na základe vpísania údajov do TextBoxu a stlačenia tlačidla. Následne úprava údajov v riadku na základe toho, že stlačíme riadok v DataGridView, vypíšu sa údaje z riadka do TextBoxov a po úprave vybraného textu stlačíme tlačidlo a riadok je upravený. A tiež obsahuje zmazanie riadku tiež na základe údajov v TextBoxe a stlačenia tlačidla.

**Obr. 8 Wireframe Formu SpravaUzivatelov.cs**

Wireframe sme riešili nasledovnými spôsobmi: menu naľavo sme ponechali také isté. Hore napravo sme pridali panel a nad ním DataGridView v takej veľkosti, aby bolo možné vidieť dostatočne tie riadky. Následne sme vpravo na dolnej časti pridali ďalší panel, ktorému sme nastavili bielu farbu. V ňom sú zahrnuté všetky Labely, TextBoxy a tlačidlá. TextBoxy sme rozložili pod seba tak, aby to bolo dizajnovo príťažlivé a aby to užívateľ videl. Samozrejme naľavo od nich sme pridali Labely s popismi čo sa má do TextBoxov zapisovať. V paneli vľavo hore je popis „Užívateľ“, čiže je známe, že je to správa užívateľov. Dole sú tlačidlá ako pridať, zmazať, upraviť, ktoré sú základom tej správy užívateľov. A ešte tlačidlo, ktoré odkazuje na zoznam vlakov.

Vo Forme SpravaVlakov sme wireframe vyriešili veľmi podobným spôsobom. Rozdiely sú tie, že sme pridali viac TextBoxov a Labelov, keďže tabuľka ZoznamVlakov obsahuje viac stĺpcov. Pri zoraďovaní Labelov sme mali problém, že sme nevedeli ako presne ich rozmiestniť tak, aby sa zmestili. Ale podarilo sa nám ten problém vyriešiť spôsobom, že sme individuálne nastavili šírku TextBoxov.

Obrázok, na ktorom je text, displej, snímka obrazovky, softvér

Automaticky generovaný popis

**Obr. 9 Wireframe Formu SpravaVlakov.cs**

Ako prvý krok v programovaní kódu sme vytvorili privátnu metódu bez návratovej hodnoty s názvom ZobrazenieUdajov. Neobsahuje parameter.

Ako vždy sme použili try-catch blok, v try bloku sme použili príkaz SELECT na vyhľadanie všetkých stĺpcov z tabuľky ZoznamUctov pomocou znaku \*.

private void ZobrazenieUdajov()

{

try

{

con.Open();

string vyhladanie = "SELECT \* FROM ZoznamUctov";

SqlCommand cmd = new SqlCommand(vyhladanie, con);

using (SqlDataReader citanie = cmd.ExecuteReader())

{

DataTable udaje = new DataTable();

udaje.Load(citanie);

dataGridView.DataSource = udaje;

con.Close();

}

}catch (Exception)

{

MessageBox.Show("Stala sa nejaká chyba pri procese zobrazenia.");

}

}

1. Záver

Zoznam použitej literatúry

1. Kolektív autorov: *Visual Studio IDE*. [online] Microsoft. Dátum čerpania: 25. 11. 2023. Dostupné na internete: < <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>>
2. https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/
3. SADOVSKI: *SQL*. [online] VSB. Dátum čerpania: 20. 11. 2023. Dostupné na internete: < <http://books.fs.vsb.cz/SQLReference/Sadovski/SQL-PRVN.HTM>>
4. Kolektív autorov: *C Sharp advantages*. [online] Codeguru. Dátum čerpania: 20. 12. 2023. Dostupné na internete: < <https://www.codeguru.com/csharp/c-sharp-advantages/>>
5. SZABÓ, Peter: *SQL*. [online] SPŠE KE. Dátum čerpania: 20. 11. 2023. Dostupné na internete: < <https://spseke.sk/tutor/prednasky/dbs/sql.html>>
6. TOELE, Dennis : *Visual Studio – everything you need to know*. [online] Licenseq. Dátum čerpania: 1.12. 2023. Dostupné na internete: < <https://licenseq.com/visual-studio-everything-you-need-to-know/>>
7. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-8.0>
8. <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview>
9. <https://www.geeksforgeeks.org/introduction-to-ado-net/>
10. <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16>

Prílohy

Zoznam príloh záverečnej práce:

* Príloha A – USB médium

Príloha A – USB médium

Priložené USB médium obsahuje text záverečnej práce vo formáte PDF.