UNICHECK

Ім'я користувача:
приховано налаштуваннями конфіденційності

ID перевірки:
1015562422

Дата перевірки:
12.06.2023 12:53:06 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
12.06.2023 13:21:58 EEST

ID користувача:
100011372

Назва документа: ОК-41 Нанай А.С

Кількість сторінок: 34 Кількість слів: 4903 Кількість символів: 37171 Розмір файлу: 1.44 MB ID файлу: 1015213413

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

7.77%
Схожість

Найбільша схожість: 3.22% з джерелом з Бібліотеки (ID файлу: 1003992029)

Пошук збігів з Інтернетом не проводився

7.77% Джерела з Бібліотеки | 119 ..... Сторінка 36

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування | 7 сторінок

1 АНАЛІЗ ЗАВДАННЯ ТА ОГЛЯД СУЧАСНИХ ПРОГРАМ-МЕНЕДЖЕРІВ

1.1 Аналіз завдання

В даному дипломному проэкті потрібно розробити десктопний додаток для ОС Windows. Основні вимоги до проектування:

- додаток повинен взаємодіяти з користувачем;
- повинні працювати локальні оповіщення від додатку;
- інтерфейс повинен бути зрозумілий, всі елементи повинні бути зручні для їх знаходження та розуміння;
- повинна бути присутня зручна навігація в додатку;
- повинна бути підтримка персоналізації з користувачем;
- повинна бути присутня локалізація;
- додаток повинен працювати в режимі офлайн.

При дотриманні цих вимог розробки десктопний додаток вийде зручним для користування на різних пристроях . Додаток буде підтримувати українську та англійську мови для зручності користувача.

Десктопний додаток повинен забезпечити рішення наступних задач:

- можливість створювати , редагувати , видаляти файли , завдання. Виведення повідомлення про визначені події;
- створення нотаток з можливістю додання назви, опису, та часу нагадування;
- перегляд розкладу пар;
- збереження інформації корисної для навчання.

1.2 Вибір етапів проектування

Для розробки десктопного додатку “Менеджера для навчання” слід дотримуватися етапів створення десктопних додатків:

- визначення цільової аудиторії;
- вибір платформи;
- розробка концепції;
- формування технічного завдання;
- розробка дизайну інтерфейсу;
- програмна розробка;
- тестування;
- відлагодження;
- реєстрація та реліз у Internet-магазинах;

1.3 Сучасні програми - менеджери

Десктопні додатки - це програми, які встановлюються та запускаються безпосередньо на комп'ютері користувача, призначені для виконання різних функцій та завдань. Вони надають розширені можливості та широкий функціонал, орієнтований на задоволення потреб користувачів у різних сферах життя та роботи.

Сучасні програми-менеджери відкривають безліч можливостей для ефективного керування проектами та завданнями. Вони допомагають заощадити час, поліпшують комунікацію, спільну роботу та досягнення поставлених цілей. Незалежно від розміру команди чи сфери діяльності, сучасні програми-менеджери

можуть стати незамінними інструментами для підвищення продуктивності та успіху вашого проекту. Вони дозволяють організувати робочі графіки, створювати списки завдань, пріоритети та розподіляти ресурси. Вони спрощують спілкування та спільну роботу у команді, дозволяючи обмінюватися даними, файлами та коментарями. Додатково, програми-менеджери можуть надавати аналітику та звітність щодо продуктивності, ресурсів та фінансів, допомагаючи ідентифікувати проблеми та приймати обґрунтовані рішення на основі даних. З їхньою допомогою ви можете більш ефективно організовувати свою роботу, зосереджуючись на найважливіших аспектах та досягаючи успіху в проектах. Ось деякі ключові аспекти, що характеризують сучасні програми-менеджери:

Централізоване керування завданнями. Сучасні менеджери дозволяють створювати, організовувати та призначати завдання різним учасникам команди. Вони надають зручний інтерфейс для створення списків завдань, пріоритезації, встановлення термінів та відстеження прогресу.

Комунікація та спільна робота. Сучасні програми-менеджери надають засоби для ефективної комунікації в команді. Вони можуть включати функції обміну повідомленнями, коментування завдань, спільного доступу до файлів та спільної роботи над проектами. Це дозволяє забезпечити взаємодію між учасниками команди, обмін знаннями та координацію зусиль.

Графіки та аналітика. Сучасні менеджери мають вбудовані інструменти для створення графіків, діаграм та звітів. Це дозволяє візуалізувати прогрес проекту, визначати критичні шляхи, виявляти затримки та відстежувати ефективність. Аналітичні функції допомагають менеджерам отримувати важливі дані для прийняття обґрунтованих рішень.

Інтеграція з іншими інструментами. Сучасні програми-менеджери можуть інтегруватися з іншими корисними інструментами, такими як електронна пошта, календарі, чат-програми, хмарні сховища та інші. Це спрощує обмін даними та забезпечує зручність роботи в одній централізованій системі.

Мобільність. Багато сучасних менеджерів мають мобільні додатки, що дозволяють працювати з ними на смартфонах та планшетах. Це дає можливість бути завжди підключеним до своїх проектів та завдань, отримувати сповіщення та оновлення навіть поза офісом.

Загалом, сучасні програми-менеджери надають широкі можливості для ефективного керування проектами та завданнями. Вони допомагають зберегти час, покращити комунікацію, спільну роботу та досягнення поставлених цілей. Незалежно від розміру команди або галузі діяльності, сучасні програми-менеджери можуть бути корисними інструментами для підвищення продуктивності та успіху вашого проекту.

1.4 Огляд способів розробки десктопних додатків

Засоби розробки десктопних додатків - це набір інструментів, які використовуються для створення програмного забезпечення, що працює на персональних комп'ютерах. Такі додатки можуть бути розроблені для різних операційних систем, таких як Windows, macOS або Linux, і можуть надавати різноманітний функціонал від стандартних офісних інструментів до важких наукових обчислень.

Популярні засоби розробки десктопних додатків:

1. Electron - це фреймворк, який дозволяє розробникам створювати десктопні додатки за допомогою веб-технологій, таких як HTML, CSS і JavaScript. Він базується на двох основних компонентах: веб-браузері Chromium і фреймворку Node.js. Завдяки цьому розробники можуть створювати багатоплатформні додатки, які працюють на різних операційних системах.
2. JavaFX - це набір бібліотек і інструментів для розробки десктопних додатків на мові програмування Java. Він надає розширені можливості для

створення графічного інтерфейсу користувача, а також підтримку мультимедіа і відео. JavaFX працює на різних платформах, включаючи Windows, macOS і Linux.

3. WinForms (Windows Forms) - це технологія розробки десктопних додатків для платформи Windows, використовуючи мови програмування, такі як C# або VB.NET. WinForms базується на бібліотеці класів .NET Framework і надає широкі можливості для створення інтерфейсу користувача, включаючи різноманітні елементи управління і зв'язок з базами даних.

4. WPF (Windows Presentation Foundation) - це ще одна технологія розробки десктопних додатків для платформи Windows, яка також працює з мовами програмування C# або VB.NET. WPF надає більш розширені можливості для створення багатоплатформних інтерфейсів користувача з використанням XAML (Extensible Application Markup Language) для опису інтерфейсу.

5. Qt - це платформонезалежний фреймворк для розробки десктопних додатків. Він підтримує різні мови програмування, такі як C++ і Python, і дозволяє розробникам створювати багатоплатформні додатки, які працюють на Windows, macOS, Linux і багатьох інших операційних системах.

Це лише кілька прикладів засобів розробки десктопних додатків. Існує багато інших фреймворків і інструментів, які можуть бути використані для створення десктопних додатків залежно від потреб і вимог розробки.

Середовища розробки десктопних додатків – це програми або набори інструментів, які допомагають розробникам створювати, тестувати і налагоджувати десктопні додатки. Вони забезпечують зручний і ефективний інтерфейс для написання коду, керування проектами, управління версіями, відлагодження програми і багато іншого.

Популярні середовища розробки (IDEs) для десктопних додатків:

1. Visual Studio - це інтегроване середовище розробки (IDE) від Microsoft. Воно підтримує розробку десктопних додатків для різних платформ, включаючи Windows, macOS і Linux. Visual Studio має розширений набір інструментів для

підтримки мов програмування, таких як C++, C#, VB.NET, Java і багатьох інших. Воно також надає можливості для створення графічного інтерфейсу користувача, відлагодження програми, керування версіями і багато іншого.

2. IntelliJ IDEA - це популярне середовище розробки для Java і інших мов програмування, таких як Kotlin, Groovy, Scala тощо. Воно має багато розширених функцій, таких як рефакторинг коду, підказки під час розробки, інструменти для тестування і налагодження, підтримку систем контролю версій, інтеграцію зі засобами збирання проектів і багато іншого.

3. Xcode - це IDE для розробки десктопних додатків під операційну систему macOS і додатків для iOS. Він має широкий набір інструментів для створення інтерфейсу користувача, розробки програмного коду, налагодження, тестування та профілювання додатків для платформ Apple. Xcode також надає доступ до різних фреймворків, документації і інших ресурсів, необхідних для розробки додатків для платформ Apple.

4. PyCharm - це IDE, спеціально розроблене для розробки додатків на мові програмування Python. Воно надає широкий набір інструментів для розробки Python-проектів, включаючи підказки коду, аналіз проекту, рефакторинг, відлагодження, тестування і багато іншого.

Інтегроване середовище розробки (ICP) - це комплексне програмне рішення, яке надає зручну і продуктивну робочу середу для розробки програмного забезпечення. Воно включає в себе різноманітні інструменти та функції, які сприяють ефективному процесу програмування.

Одним з найпопулярніших функціональних можливостей ICP є автодоповнення коду. Ця функція аналізує контекст роботи розробника і пропонує варіанти продовження коду на основі наявного коду, доступних бібліотек або інших ресурсів. Це дуже корисна функція, яка прискорює процес розробки та допомагає уникнути синтаксичних помилок.

Деякі середовища розробки містять компілятор, інтерпретатор. Деякі інтегровані середовища розробки містять систему керування версіями або інструменти для полегшення розробки графічного інтерфейсу користувача (XCode, Embarcadero Delphi). Багато сучасних IDE містять інспектор класів, інспектор об’єктів, схему ієрархії класів для полегшення об’єктно-орієнтованої розробки програмного забезпечення.

Інтегровані середовища розробки створені для того, щоб максимізувати продуктивність програміста, надавши йому пов'язані інструменти розробки зі схожими інтерфейсами як одну програму, в якій відбуватиметься весь процес розробки й яка надає необхідні функції для модифікації, компілювання, розгортання та налагодження програмного забезпечення. Протилежним до цього є підхід до розробки ПЗ, під час якого використовуються окремі інструменти, так як vi, GCC або make.

Одним із завдань IDE є зменшення часу, необхідного на конфігурацію різноманітних інструментів розробки, натомість пропонуючи той самий набір, як єдине ціле. Це може збільшити продуктивність розробника, у випадку, коли навчання тому, як працює інтегроване середовище розробки є швидшим, ніж освоєння всіх інструментів зокрема. Крім того, більша інтеграція між вбудованими інструментами потенційно може сприяти додатковому збільшенню продуктивності. Наприклад, синтаксичний аналіз коду може відбуватися безпосередньо під час його редагування, тим самим виявляючи помилки ще до трансляції коду.

Існує категорія інтегрованих середовищ розробки, спеціалізованих на підтримці певних мов програмування або групи схожих мов. Ці IDE, такі як PhpStorm, Xcode, Xojo і Delphi, надають набір функціональностей, які краще відповідають парадигмі програмування цих мов.

Більшість сучасних IDE мають графічний інтерфейс користувача, що спрощує роботу з ними. Раніше, до появи систем, таких як Microsoft Windows та X

Window System (X11), широко використовувалися консольні IDE, наприклад Turbo Pascal. Вони використовували функціональні клавіші та комбінації клавіш для запуску команд або макросів, які часто використовувалися в розробці.

Ці інтегровані середовища розробки різняться за функціональністю, інтерфейсом та набором інструментів, проте всі вони призначені для сприяння зручній і продуктивній розробці програмного забезпечення. Кожен розробник може вибрати IDE, яке найбільше відповідає його потребам і мові програмування, з якою він працює.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ

2.1 Вибір мови програмування

Існує безліч мов програмування, які можна використовувати для розробки різноманітних програм і додатків. Далі я перерахую деякі популярні мови програмування :

Java є однією з найпопулярніших мов програмування, використовуваних у багатьох галузях. Вона є мовою загального призначення і використовується для розробки десктопних, мобільних, веб- та корпоративних додатків.

PPython - це проста, легкочитана мова програмування, яка стала дуже популярною у сферах науки про дані, штучного інтелекту, веб-розробки та багатьох інших. Вона має багато сторонніх бібліотек і фреймворків, що робить її потужним інструментом для розробки різних програм.

C++ - це мова програмування загального призначення, яка використовується для створення продуктивного програмного забезпечення, включаючи десктопні додатки, ігри, вбудовані системи і багато іншого. Вона надає велику контроль над ресурсами і ефективність виконання.

C# (C-Sharp) - це мова програмування, розроблена компанією Microsoft, яка часто використовується для розробки десктопних та веб-додатків, а також ігор для платформи Windows.

JavaScript - це мова програмування, що використовується в основному для розробки веб-додатків. Вона дозволяє створювати динамічні інтерактивні елементи на веб-сторінках, і зараз також використовується для розробки десктопних додатків за допомогою фреймворків, таких як Electron.

Я обрав мову програмування ,,C#,, .

Існує кілька причин, чому я обрав мову програмування C# замість

інших мов. Ось деякі з них:

- Розширена підтримка платформи Windows: C# розроблена компанією Microsoft і має глибоку інтеграцію з платформою Windows. Це означає, що C# надає розширені можливості для розробки десктопних додатків, служб Windows, веб-сайтів і інших програмних продуктів, спеціально призначених для роботи на платформі Windows.
- Широкі можливості розробки: C# підтримує розробку різних типів додатків, включаючи десктопні, веб- та мобільні додатки. Ви можете розробляти додатки для платформи Windows, а також використовувати фреймворки, такі як Xamarin і Unity, для розробки мобільних додатків для Android і iOS. Це дає вам можливість писати код один раз і використовувати його на різних платформах.
- Об'єктно-орієнтована природа: C# є об'єктно-орієнтованою мовою програмування, що дозволяє розробникам використовувати принципи ООП для створення чистого, модульного і легкозрозумілого коду. Це сприяє розробці більш структурованих і масштабованих додатків.
- Багата екосистема та підтримка: C# має багату екосистему, включаючи різні фреймворки, бібліотеки і інструменти, які полегшують розробку і підтримку додатків. Крім того, мова має велике співтовариство розробників, яке надає допомогу, ресурси, бібліотеки коду і відповіді на запитання.
- Широке використання в індустрії: C# широко використовується в індустрії, особливо для розробки десктопних додатків, корпоративних рішень і веб-сайтів. Це означає, що ви можете знайти багато робочих місць, проектів і ресурсів для розвитку вашої кар'єри в цій області.

Компанія Microsoft розробила кілька фреймворків для мови програмування C#, які сприяють розробці різних типів додатків. Ось кілька з них:

- . NET Framework є одним з найпоширеніших фреймворків для розробки додатків з використанням мови C#. Він надає засоби для створення десктопних додатків, веб-додатків, служб Windows і багатьох інших типів програм. .NET Framework має широкий набір бібліотек і вбудованих функцій, які полегшують розробку, а також підтримує різні мови програмування, включаючи C#.
- . NET Core є відкритим і переносимим фреймворком, який дозволяє розробляти додатки на платформах Windows, macOS і Linux. Він надає швидкодіюче і легковагове середовище для розробки додатків з використанням C# і інших мов програмування. .NET Core також підтримує розробку додатків для контейнерів і хмарних платформ.
- ASP.NET - це фреймворк для розробки веб-додатків з використанням мови C#. Він надає засоби для створення потужних і масштабованих веб-додатків, включаючи сторінки, веб-служби, API і багато іншого. ASP.NET має різні підходи, такі як ASP.NET Web Forms, ASP.NET MVC і ASP.NET Core, які відповідають різним потребам розробників.
- Entity Framework є ORM (Object-Relational Mapping) фреймворком, який спрощує взаємодію з базами даних з використанням C#. Він надає високорівневий інтерфейс для роботи з даними, що дозволяє зосередитися на логіці додатку, а не на деталях бази даних. Entity Framework підтримує різні бази даних і дозволяє використовувати підходи Code First, Database First і Model First.

За допомогою таких фрейворків можна розробляти програми з різноманітними властивостями і можливостями , наприклад : Доступ до баз данних , або створення графічного інтерфейсу .

2.2 Вибір баз данних

Існує велика кількість баз даних, які можна використовувати для зберігання та керування даними в програмах. Ось кілька популярних баз даних:

- MySQL є однією з найпопулярніших відкритих реляційних баз даних. Вона має велику спільноту користувачів і широкий набір функцій, які полегшують роботу з даними.
- Oracle є однією з найпоширеніших комерційних реляційних баз даних. Вона відома своєю масштабованістю, надійністю і широким набором функцій для управління даними.
- Microsoft SQL Server є комерційною реляційною базою даних, розробленою компанією Microsoft. Вона є ключовою складовою екосистеми Microsoft і надає широкі можливості для управління даними, включаючи розширену підтримку для платформи Windows.
- PostgreSQL є потужною відкритою реляційною базою даних з високим рівнем надійності і розширюваності. Вона підтримує велику кількість функцій, включаючи географічні дані, текстовий пошук і багато іншого.
- MongoDB є документ-орієнтованою нереляційною базою даних. Вона забезпечує гнучку схему, горизонтальне масштабування і високу продуктивність для роботи з даними.
- SQLite є вбудовуваною реляційною базою даних, яка використовує один файл для збереження всіх даних. Вона легка, швидка і поширена в мобільних додатках та вбудованих системах.

Я обрав для збереження данних якими оперує мій додаток базу данних

„ Microsoft SQL Server„ ця база данних має переваги перед іншими які зумовили мій вибір для дипломного проекту .

Широкий функціонал: SQL Server надає широкий спектр функцій і можливостей для управління даними. Він підтримує реляційну модель даних, забезпечує потужність і гнучкість для роботи з великими обсягами даних і має вбудовану підтримку для різних типів даних, індексів, транзакцій і безпеки.

Масштабованість і продуктивність: SQL Server може масштабуватися від невеликих проектів до великих підприємств, забезпечуючи високу продуктивність та швидкодію. Він підтримує різні режими роботи, включаючи одиночний сервер, кластери та розподілені системи, що дозволяє пристосовувати його до потреб проекту.

Інструменти для розробки і адміністрування: SQL Server надає багато інструментів для розробки, адміністрування та моніторингу баз даних. Наприклад, SQL Server Management Studio (SSMS) - це потужний інструмент для розробки і управління базами даних SQL Server, який надає широкі можливості для написання запитів, профілювання, настроювання і налагодження.

Робочий інтерфейс Microsoft SQL Server зображений на рисунку 2.1:

Рисунок 2.1 – Робочий інтерфейс Microsoft SQL Server

Інтеграція з екосистемою Microsoft: SQL Server має глибоку інтеграцію з іншими продуктами та технологіями Microsoft. Наприклад, він добре поєднується

з розробкою додатків на базі .NET Framework і має вбудовану підтримку мови C# для написання збережених процедур, функцій і тригерів.

Безпека і надійність: SQL Server надає різні механізми безпеки для захисту даних, включаючи автентифікацію, авторизацію, шифрування та аудит. Він також має механізми резервного копіювання і відновлення, які дозволяють забезпечити надійність даних і можливість відновлення в разі випадкового видалення або помилкових дій.

Підтримка: Microsoft SQL Server має широку спільноту користувачів і активну підтримку від компанії Microsoft. Існує багато документації, форумів, розробників інструментів, а також оновлення та патчі для забезпечення безпеки і функціональності.

2.3 Вибір середовища розробки

Існують різні середовища розробки Ось кілька популярних середовищ розробки (IDEs) для десктопних додатків:

- Visual Studio - це інтегроване середовище розробки (IDE) від Microsoft. Воно підтримує розробку десктопних додатків для різних платформ, включаючи Windows, macOS і Linux. Visual Studio має розширений набір інструментів для підтримки мов програмування, таких як C++, C#, VB.NET, Java і багатьох інших. Воно також надає можливості для створення графічного інтерфейсу користувача, відлагодження програми, керування версіями і багато іншого.
- IntelliJ IDEA - це популярне середовище розробки для Java і інших мов програмування, таких як Kotlin, Groovy, Scala тощо. Воно має багато розширених функцій, таких як рефакторинг коду, підказки під час розробки, інструменти для

тестування і налагодження, підтримку систем контролю версій, інтеграцію зі засобами збирання проектів і багато іншого.

- Xcode - це IDE для розробки десктопних додатків під операційну систему macOS і додатків для iOS. Він має широкий набір інструментів для створення інтерфейсу користувача, розробки програмного коду, налагодження, тестування та профілювання додатків для платформ Apple. Xcode також надає доступ до різних фреймворків, документації і інших ресурсів, необхідних для розробки додатків для платформ Apple.
- PyCharm - це IDE, спеціально розроблене для розробки додатків на мові програмування Python. Воно надає широкий набір інструментів для розробки Python-проектів, включаючи підказки коду, аналіз проекту, рефакторинг, відлагодження, тестування і багато іншого.
- Objective-C використовують інтегроване середовище розробки. Деякі середовища розробки містять компілятор, інтерпретатор та можливість автозаповнення коду, що полегшує роботу з кодом. забезпечення.

Я обрав Visual Studio для розробки свого додатку через ряд переваг :

Інтегроване середовище розробки (IDE): Visual Studio надає повноцінне інтегроване середовище для розробки додатків на платформі Microsoft. Воно містить набір потужних інструментів, таких як редактор коду, візуальний дизайнер і відладчик, які спрощують процес розробки.

Підтримка мов програмування: Visual Studio підтримує широкий спектр мов програмування, включаючи C#, VB.NET, F# і багато інших. Це дозволяє розробникам вибирати мову, з якою вони найбільш знайомі і комфортні для роботи.

Платформна підтримка: Visual Studio має вбудовану підтримку для різних платформ, включаючи Windows, Android, iOS і хмарні платформи, такі як Azure. Це дозволяє розробникам створювати додатки для різних цільових платформ з використанням одного інструменту.

Розширюваність: Visual Studio має широкий вибір розширень і додаткових компонентів, які дозволяють налаштувати середовище розробки під свої потреби. Розробники можуть встановлювати розширення для підтримки конкретних фреймворків, інструментів тестування, систем контролю версій і багато іншого.Інтеграція з іншими інструментами Microsoft: Visual Studio гармонійно поєднується з іншими інструментами та технологіями Microsoft, такими як Azure, SQL Server, Team Foundation Server і GitHub. Підтримка спільної роботи: Visual Studio має вбудовані інструменти для спільної роботи, що дозволяють розробникам працювати в команді над проектами.

Робочий інтерфейс Visual Studio зображений на рисунку 2.2:

Рисунок 2.2 – Робочий інтерфейс Visual Studio

Visual Studio досить часто оновляється , а компанія Microsoft регулярно робить оновлення для фреймворків які використовуються мовами програмування.

У Visual Studio є купа корисних інструментів одним із них є браузер який

дозволяє під час роботи за умови що є інтернет підключення прямо у вікні програми знайти і встановити пакет додаткових бібліотек для роботи з тим чи інакшим фреймворком чи ще чимось . Він дозволяє під час встановлення пакету бібілотек вибрати їх версію а також проек до якого їх потрібно доєднати , реалізований перегляд популярності різних пакетів .

Також є зручний Solution Explorer який дозволяє переглядати файли які знаходятья у проекті взаємодіяти з програмними файлами і змінювати назви , мізце розташування і прив'язку цих фалів. Це полегшує роботу з файловим менеджером , тепер майже все що потрібно можна зробити у середовищі розробки.

Системи контролю версій (СКВ) є інструментами, які дозволяють відстежувати зміни в файловій системі та керувати версіями файлів і проектів. Вони є незамінними для розробки програмного забезпечення і спрощують спільну роботу над проектом у команді. За допомогою Git Hub або Git Lab або інших засобів можна використовувати систему контролю версій у самому середовищі розробки зі всіма можливостями .

Зручність перегляду та взяємодії з кодом , можливість змінювати положення вікон з додатковимим засобами розробки , відкріплення їх від вікна середовища заміна і широкий вибір для виконання різних задач і багато інших переваг допомогли мені обрати саме це середовище розробки для мого додатку .

3 РОЗРОБКА КОМП'ЮТЕРНОГО ДОДАТКА-МЕНЕДЖЕРА ДЛЯ НАВЧАННЯ

3.1 Розробка структури програми та опис її функціональних можливостей

Головний клас додатку - містить точку входу в додаток та ініціалізує головну форму додатку.

Головна форма (вікно) додатку: Відображає інтерфейс користувача та надає доступ до функціональності додатку. Містить розклад занять, поле для введення подій та кнопку для додавання нагадувань. Забезпечує відображення розкладу занять, списку нагадувань та довідкових матеріалів. Меню навігації: Зазвичай розташоване у верхній частині вікна і містить різні пункти меню, які дозволяють користувачу переходити до різних функціональних модулів додатку, таких як:

- Нотатки;
- Розклад;
- Нагадування;
- Файли;
- Календар.

Розділ «Нотатки» містить текстове поле яке дозволяє вводити текст у досить великій кількості , з можливістю збереження тексту кнопкою . Також програма самостійно автоматично зберігає данні з текстового поля при завершенні роботи . Цей модуль програми потрібен для того щоб спростити роботу користува , для того щоб не потрібно було переключатись між вкладками , все на одному вікні .

Розділ «Розклад» містить відображення розкладу занять студента. Він може бути представлений у вигляді таблиці або календаря, де кожен день має свої відповідні заняття. Користувач може переглядати розклад на поточний тиждень або переключатись на інші тижні.

Розділ «Нагадування» дозволяє користувачу створювати нагадування для певних подій або занять у розкладі. Користувач може встановлювати дату, час та повідомлення для нагадування. При наближенні до встановленого часу нагадування, додаток повинен виводити сповіщення користувачеві.

Розділ «Файли» дозволяє користувачу переглядати та відкривати різні файли, такі як текстові файли, зображення, документи тощо. Користувач може переглядати список файлів, обираючи їх з деревовидного перегляду або списку, та відкривати їх за допомогою програм, встановлених на комп'ютері за замовчуванням.

Модуль роботи з базою даних: Забезпечує збереження та вибірку даних з бази даних MS SQL Server. Містить сутність (модель) для розкладу занять і нагадувань. Здійснює створення та підключення до локальної бази даних у форматі .mdf. Забезпечує функції для збереження розкладу занять, нагадувань та інших даних у базі даних.

Модуль нагадувань у мому додатку є інструментом, який дозволяє користувачам створювати нагадування про певні події з розкладу. Цей модуль забезпечує збереження нагадувань у базі даних і їх відображення на головній формі додатку.

Один з ключових функціональних можливостей модуля нагадувань - це можливість створювати нові нагадування. Користувач може ввести необхідну інформацію, таку як заголовок нагадування, дату і час події, а також додаткові деталі, які він вважає за потрібне. Після збереження нагадування воно автоматично додається до бази даних.

Метод який забезпечує скачування данних про розклад з бази данних зображений на рисунку 3.1:

Рисунок 3.1 – Метод завантаження розкладу з локальної бази данних

Одним з важливих аспектів модуля нагадувань є перевірка активних нагадувань. Програма регулярно перевіряє базу даних на наявність активних нагадувань, тобто тих, які повинні бути виконані в даному моменті.

Методи оновлення і перевірки повідомлень зображені на рисунку 3.2:

Рисунок 3.2 – Методи оновлення і перевірки повідомлень

Можливістю модуля нагадувань є редагування та видалення нагадувань. Користувач може відредагувати інформацію про нагадування, змінити дату і час події, або видалити нагадування з бази даних, якщо воно вже не потрібне.

Цей модуль допомагає забезпечити користувачам зручний та ефективний спосіб керування нагадуваннями про події з розкладу.

Модуль роботи з файлами: Забезпечує відкриття файлів програмами за замовчуванням. Здійснює збереження файлів, що пов'язані з додатком, у папці з додатком. Відповідає за роботу з різними типами файлів (текстові файли, зображення, документи тощо).

Модуль довідкових матеріалів: Забезпечує доступ до довідкових матеріалів, таких як фотографії підручників, текстові пояснення тощо. Здійснює відображення довідкових матеріалів у вікні додатку.

В результаті, поєднання даних модулів, структура програми має вигляд як показано на рис.3.3:

Рисунок 3.3 – Структура програми

3.2 Розробка та опис інтерфейсу програми

Розробка інтерфейсу додатку включає створення елементів, які користувач буде бачити і з якими буде взаємодіяти. Це включає розташування елементів на екрані, визначення їх зовнішнього вигляду, навігаційні елементи і функціональні кнопки, які забезпечують різні можливості додатку.

Головне вікно додатку буде представляти собою основний контейнер для інших елементів і функцій.

Інтефейс програми зображений на рисунку 3.4:

Рисунок 3.4 – Інтерфейс програми

Розклад: У розділі розкладу можна відобразити таблицю або календар з днями тижня або місяцями, в залежності від обраного періоду. Кожен день має відповідні заняття, які можна відображати у вигляді текстових блоків або піктограм. Користувач може переглядати розклад на поточний тиждень або переключатись на інші тижні.

Методи для відкриття файлів зображені на рисунку 3.5:


```
}
private void OpenFile()
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "All Files (*.*)|*.*";
    openFileDialog.RestoreDirectory = true;

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        string filePath = openFileDialog.FileName;
        OpenFileWithDefaultProgram(filePath);
    }
}

private void OpenFileWithDefaultProgram(string filePath)
{
    try
    {
        Process.Start(filePath);
    }
    catch (Exception ex)
    {
        // Обробка помилок відкриття файлу
        MessageBox.Show("Помилка відкриття файлу: " + ex.Message);
    }
}

private void openFileButton_Click(object sender, EventArgs e)
{
    OpenFile();
}
```

Рисунок 3.5 – Методи для відкриття файлів

Нагадування: У розділі нагадувань користувач може переглядати список нагадувань, які він створив. Кожне нагадування може мати назву, дату та час,

інформацію про подію і опції повтору. Користувач може створювати, редагувати та видаляти нагадування.

Алгоритм роботи з додатком показаний на рисунку 3.6

Рисунок 3.6 – Алгоритм роботи з додатком

3.3 Налаштування баз данних

Локальна база даних MS SQL Server - це рішення для зберігання та управління даними, яке працює на вашому локальному комп'ютері. Вона забезпечує широкі можливості зберігання та обробки даних, а також підтримку складних операцій із базами даних. Нижче наведений код у форматі “ xml ” у якому проводиться підключення локальної бази данних по абсолютному шляху .

<?xml version="1.0" encoding="utf-8" ?>

```
<configuration>
    <connectionStrings>
        <add name="conStr" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Users\Admin\Duplomna\
Duplomna\bin\Debug\Diplom_DB.mdf;Integrated Security=True;Connect Timeout=30"
providerName="System.Data.SqlClient"/>
    </connectionStrings>
</configuration>
```

Ця стрічка використовується для створення зв’язку з локальною базою данних для подальших запитів для отриманя , внесення чи зміни данних .

Кожен раз коли потрібно щось зробити з базою данних треба відкривати нове з'єднання . Не можна тримати впродовж роботи програми одне вічно відкрите зєднання . Це зєднання може залишитись відкритим після закриття програми і закрити його можна буде тільки вимкненням комп'ютера . Також вічно відкрите одне під'єднання до бази данних сильно навантажує комп'ютер . Метод що використовує стрічку для підєднання до БД зображений на рисунку 3.5:

Рисунок 3.5 – Метод що використовує стрічку для підєднання до БД

Одна з головних переваг локальної бази даних MS SQL Server полягає у її надійності та стабільності. Вона дозволяє зберігати великі обсяги даних і

забезпечує високу швидкодію доступу до них. База даних MS SQL Server використовує ефективні алгоритми для оптимізації запитів і забезпечення швидкого доступу до даних.

Для роботи з локальною базою даних MS SQL Server ви можете використовувати різні інструменти, зокрема Microsoft SQL Server Management Studio (SSMS), який надає вам інтерфейс для створення та керування базами даних. Ви також можете використовувати різні програмні інтерфейси, такі як ADO.NET або Entity Framework, для взаємодії з базою даних з вашого додатку.

Локальна база даних MS SQL Server дозволяє вам створювати таблиці для зберігання даних і встановлювати відношення між ними. Ви можете визначити типи даних для полів таблиць, обмеження цілісності, індекси та інші параметри, що дозволяють забезпечити правильність та ефективність роботи з даними.

Локальна база даних MS SQL Server підтримує мову структурованих запитів SQL, що дозволяє вам створювати складні запити для отримання, оновлення та видалення даних. Ви можете використовувати різні оператори, функції та групування для виконання різних операцій з даними.

При роботі з локальною базою даних MS SQL Server важливо забезпечити безпеку даних. Ви можете встановити різні рівні доступу до бази даних, обмеження прав користувачів та ролей. Також, база даних MS SQL Server надає можливість резервного копіювання та відновлення даних, що дозволяє забезпечити збереження та відновлення даних у разі випадкового видалення або втрати.

Окрім цього, локальна база даних MS SQL Server підтримує транзакції, що дозволяють виконувати групу операцій як єдину атомарну операцію. Це забезпечує цілісність даних і дозволяє вам відновлювати базу даних у випадку помилки або відміни операції.

Усі ці функціональні можливості локальної бази даних MS SQL Server роблять її потужним інструментом для розробки додатків, які потребують зберігання та обробки даних. Вона надає надійну та ефективну інфраструктуру для роботи з даними та дозволяє легко масштабувати ваш додаток у майбутньому.

Після підключення бази даних я здійснив важливий крок, а саме використання спеціальних класів, які відповідають таблицям бази даних. Ці класи працюють як невід'ємні помічники, що допомагають вам перетворити дані з бази даних у зручний для мене формат - об'єкти. Такий підхід полегшує подальшу роботу з цими даними.

Один з основних компонентів вашого додатку - це цикл "while". Цей цикл використовується для пройдення крізь таблицю бази даних. Кожна ітерація циклу дозволяє мені отримати наступний рядок даних з таблиці. І це ще не все! Я використовую блок "using" для забезпечення належного закриття з'єднання з базою даних після завершення циклу. Це необхідно для звільнення ресурсів та ефективного управління пам'яттю, що є важливим аспектом будь-якої програми.

Одним із способів зберігання даних є використання списків. Я вирішив використовувати "List", який містить об'єкти одного типу, описані класами, що відповідають таблицям бази даних. Цей список дозволяє мені зручно зберігати дані у відповідному форматі та з легкістю працювати з ними, зокрема за допомогою різноманітних операцій списків.

Останній аспект, але не менш важливий, полягає в методах, які я розробив для оновлення ListBox. ListBox відображає списки розкладів і даних, ці методи дозволяють мені оновлювати вміст ListBox, щоб відображати актуальні дані з бази даних. Це важливий механізм для забезпечення користувачеві завжди оновленої та точної інформації.

В цілому, мій додаток має структуру, яка сприяє зручній та ефективній роботі з базою даних, а також забезпечує відображення актуальних даних для

користувача. Такий підхід робить мій додаток потужним та високопродуктивним, дозволяючи ефективно використовувати дані та забезпечувати задоволення від використання програми користувачам.

3.4 Тестування розробленої програми

Тестування мого додатку є важливим етапом в процесі розробки, яке дозволяє виявити й виправити помилки та недоліки, забезпечити якість та надійність роботи програми. Тестування має на меті перевірити функціональність, коректність роботи та взаємодію з іншими компонентами системи.

Інтеграційне тестування спрямоване на перевірку взаємодії різних компонентів вашого додатку. Я можете перевірити, чи працюють компоненти разом коректно та передають дані один одному без помилок. Це можна зробити шляхом створення тестових сценаріїв, які симулюють різні ситуації та перевіряють очікувані результати.

Прийомочне тестування проводиться для перевірки додатку в реальних умовах його використання. Це може включати тестування різних сценаріїв взаємодії з користувачем, введення тестових даних та перевірку коректності відповідей та реакцій програми. Прийомочне тестування допомагає перевірити, чи задовольняє додаток вимоги та очікування користувачів.

Крім типів тестування, важливим аспектом є створення набору тестових даних, які використовуються під час тестування. Я можете створити різні сценарії з варіаціями даних, щоб перевірити, чи правильно обробляє додаток різні вхідні дані.

Тестування мого додатку допомогає забезпечити його якість та надійність, виявити та виправити помилки. Це важливий етап у розробці любого додатку, який дозволяє створити продукт, що задовольняє потреби користувачів та забезпечує їхнє задоволення від використання.

Схожість

Джерела з Бібліотеки 119

1	Студентська робота	ID файлу: 1003992029	Навчальний заклад: Lviv Polytechnic National University	13 Джерело	3.22%
2	Студентська робота	ID файлу: 1009721551	Навчальний заклад: Lviv Polytechnic National University	6 Джерело	3.2%
3	Студентська робота	ID файлу: 1005734997	Навчальний заклад: National Technical University of Ukra	7 Джерело	0.73%
4	Студентська робота	ID файлу: 1014906563	Навчальний заклад: Ternopil Volodymyr Hnatiuk National Pedagog...		0.41%
5	Студентська робота	ID файлу: 1009687432	Навчальний заклад: Yuriy Fedkovych Chernivtsi Nationa	32 Джерело	0.41%
6	Студентська робота	ID файлу: 1015119125	Навчальний заклад: National University of Life and Envir	10 Джерело	0.35%
7	Студентська робота	ID файлу: 1015168814	Навчальний заклад: Lviv Polytechnic National University	18 Джерело	0.22%
8	Студентська робота	ID файлу: 1015106154	Навчальний заклад: National University Ostroh Academy	3 Джерело	0.2%
9	Студентська робота	ID файлу: 1015194471	Навчальний заклад: Uzhhorod National University	2 Джерело	0.2%
10	Студентська робота	ID файлу: 1015010030	Навчальний заклад: Tavria State Agrotechnological University		0.2%
11	Студентська робота	ID файлу: 1014769143	Навчальний заклад: Lviv Polytechnic National University	2 Джерело	0.2%
12	Студентська робота	ID файлу: 1015081164	Навчальний заклад: National University of Life and Environmenta...		0.18%
13	Студентська робота	ID файлу: 1015084384	Навчальний заклад: National Aviation University		0.18%
14	Студентська робота	ID файлу: 1015168127	Навчальний заклад: National Technical University of Ukraine “Kyi...		0.18%
15	Студентська робота	ID файлу: 1015194571	Навчальний заклад: Uzhhorod National University	3 Джерело	0.18%
16	Студентська робота	ID файлу: 1003905430	Навчальний заклад: National Technical University of Ukra	2 Джерело	0.18%
17	Студентська робота	ID файлу: 1015048391	Навчальний заклад: Donetsk National Technical University		0.18%
18	Студентська робота	ID файлу: 1015062635	Навчальний заклад: National Technical University of Ukra	2 Джерело	0.18%
19	Студентська робота	ID файлу: 1015153382	Навчальний заклад: Lviv Polytechnic National University		0.16%
20	Студентська робота	ID файлу: 1015068571	Навчальний заклад: Lutsk National Technical University		0.16%

21	**Студентська робота**	ID **файлу**: 1015088284	**Навчальний заклад**: National Aviation University		0.16%
22	**Студентська робота**	ID **файлу**: 1015080986	**Навчальний заклад**: National University of Life and Envir	2 Джерело	0.16%
23	**Студентська робота**	ID **файлу**: 1015196287	**Навчальний заклад**: Lviv Polytechnic National University		0.16%
24	**Студентська робота**	ID **файлу**: 1008281955	**Навчальний заклад**: Taras Shevchenko National University of Kyiv		0.16%
25	**Студентська робота**	ID **файлу**: 1014972723	**Навчальний заклад**: Lutsk National Technical University		0.16%
26	**Студентська робота**	ID **файлу**: 1015197438	**Навчальний заклад**: Lviv Polytechnic National University		0.16%
27	**Студентська робота**	ID **файлу**: 1114076	**Навчальний заклад**: Lviv Polytechnic National University		0.16%
28	**Студентська робота**	ID **файлу**: 1012748447	**Навчальний заклад**: National Aviation University		0.16%
29	**Студентська робота**	ID **файлу**: 1015151308	**Навчальний заклад**: Lviv Polytechnic National University	2 Джерело	0.16%