Міністерство освіти і науки України

Чернівецький національний університет імені Юрія Федьковича

\_\_\_\_\_\_\_Програмного забезпечення комп’ютерних систем\_\_\_\_\_\_\_

(повна назва кафедри, циклової комісії)

**КУРСОВИЙ ПРОЄКТ**

з Бази даних та NoSQL системи

(назва дисципліни)

на тему: **Інформаційна система аеропорту** **засобами мови С# та БД MongoDB**

Виконав: студент (ка)     3     курсу, групи     343А

спеціальності           121

(шифр спеціальності)

\_\_\_\_Інженерія програмного забезпечення\_\_\_\_\_

(назва спеціальності)

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Матюхін О.К.\_\_\_\_

(підпис) (прізвище, ініціали)

Керівник \_\_\_\_\_\_\_Комісарчук В.В.\_\_\_\_\_\_

(прізвище, ініціали)

|  |  |  |
| --- | --- | --- |
|  |  | **Дата захисту** «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_р. |
|  |  | **Оцінка:** |
|  |  | за національною шкалою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (словами) |
|  |  | кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (цифра) |
|  |  | за шкалою ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (літера) |

Чернівці, 2024

Чернівецький національний університет імені Юрія Федьковича

Кафедра \_Програмного забезпечення комп’ютерних систем\_\_\_

Спеціальність \_Інженерія програмного забезпечення\_\_\_\_\_\_\_\_\_\_\_\_\_

**ЗАВДАННЯ**

**на курсовий проєкт студенту**

Матюхіну Олександру Костянтиновичу

(прізвище, ім’я, по-батькові)

1.Тема проєкту:

Інформаційна система аеропорту засобами мови С# та БД MongoDB

2. Вихідні дані до проєкту:

– реалізувати засобами системи управління базою даних MongoDB інформацію про персонал аеропорту, літаки, розклад авіарейсів та наявні квитки.

– реалізувати програмний додаток для роботи з базою даних за допомогою графічного інтерфейсу;

– забезпечити можливість збереження та зчитування даних з БД;

– реалізувати механізм адміністрування за допомогою форм додавання та ідентифікації, маніпулювання даними через запити до БД;

– механізм адміністрування повинен забезпечити можливість створення користувача БД із правами власника, адміністратора, оператора та користувача із дозволом перегляду даних;

­– механізм адміністрування повинен забезпечити використання логіну та паролю для користувачів програми з реалізацією forgot password;

3. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити):

– опис загальних вимог до програми;

– опис призначення та області застосування;

– структурно-організаційний аналіз предметної області;

– формування набору вимог до програмного продукту у форматах Product Backlog (UserStory), UseCase, SRS (Software Requirements Specification):

– виокремлення сутностей об’єктів структурного аналізу предметної області;

­­­­– опис модулів проекту та алгоритмів виконання;

– опис методів програми;

– опис користувацького інтерфейсу.

– опис блок-схем програми;

– обґрунтування розробки бази даних.

4. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень):

– блок-схеми структурно-організаційного аналізу роботи служб аеропорту

– блок-схеми роботи програмного продукту;

– скріншоти роботи програми.

Завдання прийняв (ла) до виконання \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис студента)

Керівник проєкту \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис керівника)

**РЕФЕРАТ**

В рамках курсового проєкту реалізований програмний додаток для керування базою даних «Інформаційна система аеропорту» для настільних та переносних пристроїв на базі операційної системи Windows.

Застосунок призначений для використання персоналом аеропорту без спеціальної підготовки в сфері інформаційних технологій.

Область застосування – програмне забезпечення для ведення обліку персоналу аеропорту, літаків, квитків та розкладу рейсів на базі Windows-пристроїв (ноутбуків або персональних комп'ютерів).

У проєкті сформовано набор вимог до продукту у форматах Product Backlog, UseCase, SRS, набор сутностей бази даних шляхом структурно-організаційного аналізу предметної області та виокремлення сутностей об’єктів. Структура програми реалізована за допомогою алгоритмів сортування, фільтрації, пошуку даних з використанням функціоналу додавання, збереження, видалення та редагування запису. Програма реалізована у формі додатка-клієнта та NoSQL бази даних з використанням хмарних технологій.

Продукт створено на базі інтегрованого середовища розробки Visual Studio на мові C# з використанням графічної підсистеми WPF для швидкого та якісного створення додатку в операційній системі Windows, та бази даних MongoDB, яка дозволяє зручно оброблювати велику кількість даних.

Архітектура програми дозволяє у майбутньому розширити додаток із додаванням нового функціоналу та видозміненою логіки обробки.

Курсовий проект містить:   74 с.,   34   рис.,   5   табл.,   2   додатка,

  26   джерел.

*ОПЕРАЦІЙНА СИСТЕМА, WPF, КОМП’ЮТЕРИ, MongoDB, VISUALSTUDIO, C#.*

**SUMMARY**

Within course project implemnted program aplication for control of Data Base "Music goods" for poartable and desktop devices based on Windows System.

Application made for using by regular user without expirience of work and special thing in the Infomational Technology area.

Use area is software for personal using and small business in selling music content based on Windows-devices (laptop and Pesonal Computers) with limited counting recources.

Program product developed with help of integrated development environment Visual Studio in C# with using graphical sybsystem Windows Forms, wich allow fast and quality develop applications in operational system Windows.

Architecture and structure of program application allow in future expand application with adding new functional and modified processing logic.

The course project contains:   74   p.,   34  fig.,   5   table.,   2   appendice,

  26   sources

OPERATING SYSTEM, *Windows Forms*, COMPUTERS, VISUAL STUDIO, C#.

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ…………………………………….…..…6

1. АРХІТЕКТУРНА РЕАЛІЗАЦІЯ ТА ФУНКЦІОНАЛ ПРОГРАМИ .….…7
   1. Загальні вимоги до програми……………………………………………7
   2. Призначення та область застосування………………………….…….…9
   3. Функціональні вимоги до програмного продукту……………..……… 9
2. ФОРМУВАННЯ НАБОРУ ВИМОГ ДО ПРОГРАМИ ТА СУТНОСТЕЙ БАЗИ ДАНИХ……………………………………………………………………..9
   1. Структурно-організаційний аналіз предметної області…...……………………………………………………….……...9
   2. Формування переліку вимог до програмного продукту у форматах Product Backlog (UserStory), UseCase, SRS (Software Requirements Specification)…………………………………………….………………...9
   3. Виокремлення сутностей об’єктів структурного аналізу предметної області……………………………………………………………………..9
3. СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ДОДАТКУ…………………………11
   1. Структура програми……………………………………………………..11
   2. Опис методів програми………………………………………………….13
   3. Засоби розробки.………………………………………………...………21
   4. Опис користувацького інтерфейсу………………………………..……22

ВИСНОВКИ……………………………………………………………...………36

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ………………………………..………37

ДОДАТКИ………………………………………………………………..………39

Додаток А. Блок-схеми структурно-організаційного аналізу аеропорту...39

Додаток Б. Блок-схема сутностей бази даних додатку…………………….39

Додаток В. Алгоритми роботи програми…………………………………...39

Додаток Г. Скролінг (текст) програми………………………………….….49

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ОС – операційна система.

GUI (Graphical User Interface) – графічний інтерфейс користувача.

БД – база даних.

MessageBox – модальне вікно з текстом для користувача, що блокує інші дії в додатку, доки користувач не закриє його. Окрім тексту може містити кнопки та символи для покращення взаємодії з користувачем.

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів.

Windows Forms – інтерфейс програмування програм (API), який є частиною програмної платформи Microsoft .NET Framework.

MongoDB – документоорієнтована система управління базами даних, яка не вимагає опису схеми таблиць. Вважається одним із класичних прикладів NoSQL-систем, використовує JSON-подібні документи та схему бази даних.

Windows Presentation Foundation (WPF, кодова назва — Avalon) —платформа інтерфейсу користувача, яка надає комплексний набір функцій розробки додатків, які включають мову XAML, елементи управління, прив'язку до даних, макет, двовимірну і тривимірну графіку, анімацію, стилі, шаблони, документи, мультимедіа, текст і типографічні функції.

ASP.NET Core – кроссплатформний фреймворк для створення веб-додатків на платформі .NET з відкритим вихідним кодом. Платформа розробляється корпорацією Майкрософт спільно з спільнотою і має високу продуктивність, модульну структуру та сумісна з такими операційними системами як Windows, Linux та macOS.

# АРХІТЕКТУРНА РЕАЛІЗАЦІЯ ТА ФУНКЦІОНАЛ ПРОГРАМИ

## **Загальні вимоги до програми**

### **Вимоги до графічного інтерфейсу користувача**

1. Обов’язковий україномовний інтерфейс.
2. Використання зрозумілої користувачеві термінології для елементів інтерфейсу.
3. Наявність форм для запитів створення, редагування, видалення та зчитування даних БД.
4. Використання додатково панелі графічних елементів для управління БД.
5. Наявність спливаючої інформації для графічних елементів управління.
6. Забезпечення функціонування «Посібник користувача» з інформацією про використання програми.
7. Обов’язкова наявність спливаючих діалогових вікон в разі некоректного вводу даних.

### **Вимоги до архітектури програми**

1. Реалізація NoSQL бази даних.
2. Реалізація програми у формі запитів додатку-клієнта до БД.
3. Забезпечення стійкості програми:
   1. програма має бути працездатною при будь-яких діях користувача, дані або їх цілісність не повинні втрачатись;
   2. програма повинна здійснювати перевірку коректності введення-виведення даних та обробку помилок.

### **Вимоги до функціональності додатка**

1. Реалізація збереження даних у NoSQL базі даних MongoDB;
2. Реалізація можливості змін та зчитування даних з бази користувачем за допомогою додатка-клієнта.
3. Реалізація механізму доступу до БД (адміністрування) за допомогою форми добавлення користувачів БД, форми ідентифікації користувача (підключення до БД з додатка), форм введення/виведення інформації БД.
4. Виконання стандартних операцій обробки елементів БД (обробка даних та управління БД).
5. Відображення результатів обробки даних в вигляді таблиці головного вікна програми.
6. Виконання операцій фільтрації, сортування та пошуку елементів БД за різними ключами, обраними користувачем, з відображенням результатів в графічному інтерфейсі.
7. Перевірка коректності введення даних, що передаються користувачем через графічний інтерфейс програмного додатку.
8. Інформування користувача за допомогою інформаційних та діалогових вікон під час роботи користувача з БД програмного додатку.

### **Вимоги до вихідного коду**

1. Вихідний код кожного з класів програми повинен міститися в окремому файлі.
2. Наявність коментарів (для класів – призначення класу; для методів – призначення методу, опис параметрів та значення, що повертається).

## **Призначення та область застосування**

Функціональним призначеннямєвикористання програмного додатку для зручного використання та адміністрування бази даних аеропорту (управління та перегляд зі внесенням змін до бази даних: додавання, редагування, видалення, фільтрація та сортування).

Областю застосування є експлуатація програмного додатку персоналом у сфері авіаперевезень з метою покращення ефективності функціонування інфраструктури аеропорту на базі Windows-пристроїв (ноутбуків або персональних комп'ютерів).

## **Функціональні вимоги**

1. Створення бази даних.
2. Вхідні дані подаються користувачем на GUI у вигляді форм введення даних.
3. Результати запитів та обробки БД подаються в табличному вигляді.
4. Дані бази даних мають оброблятись шляхом:
   1. збереження даних у хмарному сховищі;
   2. зчитування даних з хмарного сховища;
5. Меню та панель управління повинні забезпечувати у відповідності з адміністративними правами наступні можливості:
6. оновлення даних таблиці;
7. відображення елементів БД;
8. редагування даних БД;
9. додавання даних до файлу БД;
10. видалення даних з БД;
11. Меню повинно забезпечувати можливість перегляду посібника користувача у віконному режимі.
12. Сортування інформації за зростанням та спаданням, що відображується в графічному інтерфейсі користувача у вигляді переліку записів БД за запитом користувача згідно з адміністративними правами.
13. Фільтрація інформації, що відображується в графічному інтерфейсі користувача у вигляді переліку записів БД за запитом користувача згідно з адміністративними правами.
14. Пошук інформації, що відображується в графічному інтерфейсі користувача у вигляді переліку записів БД за запитом користувача згідно з адміністративними правами за назвою;
15. Отримання та відображення підсумкової інформації в результаті виконання спеціальних запитів користувача до база даних.
16. Забезпечення перевірки допустимості даних, що вводяться користувачем.
17. Видача користувачу попереджуючих та інформаційних повідомлень.

# ФОРМУВАННЯ НАБОРУ ВИМОГ ДО ПРОГРАМИ ТА СУТНОСТЕЙ БАЗИ ДАНИХ

## **Структурно-організаційний аналіз предметної області**

Основою аналізу предметної області обрано організаційну структуру реального аеропорту згідно [1] та [2].

Базуючись на одержаних даних та умовах до курсового проекту доцільно сформовати наступну блок-схему структурно-організаційного аналізу аеропорту (див. рис.А.1). За схемою, основою ієрархії структурної організації служб аеропорту є наступні департаменти: експлуатаційний, повітряного руху, технічного обслуговування, спеціалізованих служб, адміністрації та додаткових служб.

Кожен департамент сформований згідно з його зоною відповідальності.

Так, експлуатаційний департамент відповідає за підтримку роботи аеропорту, департамент повітряного руху забезпечує планування, навігаційну підтримку та проведення польотів, департамент технічного обслуговування підтримує інфраструктуру та техніки аеропорту у гарному стані. Відповідно, департамент спеціалізованих служб відповідає за забезпечення комплексної безпеки і здоров'я, як відвідувачів аеропорту так і робочого персоналу, департамент адміністрації та додаткових служб здійснює загальне управління організацією, управління персоналом, правове забезпечення та документообіг.

Наступним щаблем організаційної ієрархії об’єкту є структурно-організаційні одиниці, які підпорядковуються департаментам – служби та відділи. Наступні блок-схеми відображають відповідну організаційну структуру кожного з департаментів:

1. Економічний департамент складається з (див. рис.А.2) планово-економічного, фінансового, відділу праці та служби продажу квитків. Економісти планово-економічного відділу відповідають за розробку і контроль фінансових планів, аналіз бюджету та ефективності, управління ресурсами і підготовку економічної звітності. Зоною відповідальності бухгалтерів-фінансистів фінансового відділу є управління фінансами організації, включаючи бюджетування, фінансове планування, облік та контроль витрат. Нормувальники відділу праці займаються управлінням трудовими відносинами, включаючи організацію праці, оплату праці та забезпечення дотримання трудового законодавства.
2. Департаменту експлуатаційних служб підпорядковуються (див. рис.А.3) служба паливно-мастильних матеріалів, служба спецтранспорта, клінінгова та рятувальна служби. Заправники служби паливно-мастильних матеріалів забезпечують ефективне управління, зберігання і подачу паливних і мастильних матеріалів для безперебійної роботи техніки та транспорту. Заправники об’єднуються у бригади для зручного розподілу між авіарейсами. Водії вантажного, аварійного транспорту та оператори машин для технічного обслуговування забезпечують безперервне функціонування спеціального транспорту. Прибиральники клінінгової служби відповідають за чистоту аеропорту та повітряних транспортних засобів. Для зручного розподілу між рейсами персонал об’єднуються у бригади прибирання аеропорту та у бригади прибирання літаків. Бригади рятувальників відповідають за оперативне реагування на надзвичайні ситуації, аварії та інциденти, забезпечуючи безпеку пасажирів і персоналу.
3. Департамент повітряного руху (див. рис.А.4) включає в себе диспетчерську, аеронавігаційну, льотну служби та обслуговування радіозв’язку. Бригади диспетчерів злітно-посадкової смуги та навігаційного обслуговування координують всі аспекти авіаційного руху, забезпечуючи безпеку і ефективність повітряного руху. Спеціалісти з планування польотів та метеорологи служби забезпечують точне планування маршрутів польотів і надають актуальну метеорологічну інформацію з метою підвищення безпеки та ефективності авіаційних операцій. Інженери з радіозв’язку служби обслуговування радіозв’язку забезпечують безперебійне функціонування і технічне обслуговування радіозв'язкових систем. Льотна служба, яка складаються з льотних бригад пілотів та бортпровідників, відповідає за підтримання комфорту і безпеки під час транспортування пасажирів до місця призначення. Льотні бригади закріплюються за певними рейсами.
4. Департамент технічного обслуговування (див. рис.А.5) складається з служб авіатехнічного обслуговування, обслуговування аеродрому, ремонту спецприладдя та обслуговування автотранспорту. Механіки з обслуговування двигунів, спеціалісти по структурному ремонту та інженери-авіоники служби авіатехнічного обслуговування здійснюють ремонт, та теохогляд авіаційного обладнання з метою забезпечення безпеки та надійності літаків. Персонал організований у бригади техогляду літаків та бригади ремонту. Бригади для обслуговування інфраструктури та злітної смуги здійснюють технічну підтримку та утримання інфраструктури аеродрому (ремонт і обслуговування будівель, електричних систем, водопроводу та дорожнього покриття і т.д.). До складу бригад входять будівельники, електрики, сантехніки та асфальтобетонники. Автомеханіки служби обслуговування автотранспорту займаються ремонтом, технічним обслуговуванням і діагностикою автомобілів, забезпечуючи їхню справність і безпеку на дорозі. Техніки обслуговування системи безпеки та інженери обслуговування обладнання аеропорту служби ремонту спец-приладдя здійснюють ремонт, технічне обслуговування та налаштування спеціалізованого обладнання аеропорту, забезпечуючи його належну функціональність і безпеку.
5. Департаменту спеціалізованих служб (див. рис.А.6) підпорядковуються служба безпеки, митна, медико-санітарна та служба реєстрації. Бригади охоронців служби безпеки здійснюють охорону території аеропорту, контроль доступу, перевірка пасажирів і багажу, а також реагування на інциденти для забезпечення загальної безпеки та порядку. Інспектори митної служби контролюють переміщення товарів і пасажирів через митний кордон, здійснюють перевірку документації та огляд вантажів і багажу. Агенти реєстрації білетів та агенти реєстрації багажу служби реєстрації здійснюють процес реєстрації пасажирів на рейси, видають посадкові талони, приймають та обробляють багаж, а також забезпечують коректність і ефективність перевірки документів. Працівники служби об'єднуються в бригади реєстрації. Бригади лікарів та медсестр медико-санітарної служби надають медичну допомогу пасажирам, здійснюють медичні огляди персоналу аеропорту, надають першу допомогу у випадку екстрених ситуацій.
6. Департамент адміністрації та додаткових служб (див. рис.А.7) включає в себе адміністрацію, відділ кадрів, юридичний відділ та канцелярію. Начальники департаментів у складі адміністрації відповідають за прийняття рішень та контроль функціонування департаментів аеропорту. Спеціалісти з підбору персоналу відділу кадрів займаються набором, оцінкою та відбором кандидатів на вакансії, організацією процесів, адаптації нових співробітників, а також керуванням питаннями кадрового обліку та розвитку персоналу. Працівники юридичного відділу аеропорту забезпечують правову підтримку аеропорту, включаючи консультування з питань авіаційного права, здійснюють підготовку та перевірку юридичних документів, управління правовими ризиками, а також представляють інтереси аеропорту у судах та інших юридичних органах. Секретарі канцелярії займаються організацією документації, управлінням кореспонденцією, веденням розкладу зустрічей, підготовкою звітів і документів, а також забезпеченням ефективної комунікації між різними відділами та зовнішніми партнерами.

## **Формування переліку вимог до програмного продукту у форматах Product Backlog (UserStory), UseCase, SRS (****Software Requirements Specification)**

### **Обґрунтування переліку запитів для реалізації у форматах Product Backlog (UserStory), UseCase, SRS (Software Requirements Specification)**

Згідно з завданням до проєкту, програмний продукт повинен мати можливість здійснити наступні запити:

1. Одержати перелік і загальне число авіарейсів зазначеної категорії, у певному напрямку, із зазначеним типом літака.
2. Одержати перелік і загальне число вільних і заброньованих місць на зазначеному рейсі, на конкретний день, по зазначеному маршруту, за ціною, за часом вильоту; загальне число зданих квитків на деякий pейс, у зазначений день, по певному маршруту, за ціною квитка, за віком, статтю.
3. Одержати перелік і загальне число літаків приписаних до аеропорту, що перебувають у ньому в зазначений час, за часом надходження в аеропорт, за кількістю здійснених рейсів; що пройшли техогляд за певний період часу, відправлених у ремонт у зазначений час, ремонтованих задане число pазів, за кількістю здійснених рейсів до ремонту, за віком літака.
4. Одержати перелік і загальне число пасажирів на даному рейсі, що полетіли зазначений день, що полетіли за кордон в зазначений день, за ознакою здачі речей у багажне відділення, за статевою ознакою, за віком.
5. Одержати перелік і загальне число пілотів, що пройшли медогляд або не пройшли його в зазначений рік, за статевою ознакою, віком, розміром заробітної платні.
6. Одержати перелік і загальне число рейсів по зазначеному маршруту, за тривалістю перельоту, за ціною квитка й по всім цим критеріям разом.
7. Одержати перелік і загальне число рейсів, якими літають літаки заданого типу й середню кількість проданих квитків на визначені маршрути, за тривалістю перельоту, за ціною квитка, часом вильоту.
8. Одержати перелік і загальне число робітників у бригаді, по всіх відділах, у зазначеному відділі, що обслуговують конкретний рейс, за віком, сумарною (середньою) зарплатнею у бригаді.
9. Одержати перелік і загальне число скасованих рейсів повністю, у зазначеному напрямку, по зазначеному маршруту, за кількістю невикористаних місць, за відсотковим співвідношенням невикористаних місць; затриманих рейсів повністю, за зазначеною причиною, по зазначеному маршруту, і кількість зданих квитків під час затримки.
10. Одержати список і загальне число всіх робітників аеропорту, начальників відділів, робітників зазначеного відділу, за стажем роботи в аеропорту, статевою ознакою, віком, ознакою наявності й кількості дітей, за розміром заробітної платні.

Також з метою відображення системою реальних умов функціонування аеропорту при розробці програмного продукту додатковими умовами є (згідно вимог до проекту):

1. Працівників аеропорту можна розділити на пілотів, диспетчерів, техніків, касирів, робітників служби безпеки, довідкової служби й інших, що адміністративно відносяться кожний до свого відділу.
2. У відділах існує розбиття робітників на бригади.
3. Відділи очолюються начальниками, які представляють собою адміністрацію аеропорту. У функції адміністрації входить планування рейсів, складання розкладів, формування кадрового складу аеропорту.
4. За кожним літаком закріплюється бригада пілотів, техніків й обслуговуючого персоналу. Пілоти зобов’язані проходити щороку медогляд, тих, що не пройшли медогляду необхідно перевести на іншу роботу.
5. Літак повинен своєчасно оглядатися техніками й при необхідності ремонтуватися. Підготовка до рейсу містить у собі технічну частину (техогляд, заправка необхідної кількості палива) і обслуговуючу частину (прибирання салону, запас продуктів харчування й т.п.).
6. В розкладі вказується тип літака, рейс, дні відльоту, час відльоту й прильоту, маршрут (початковий і кінцевий пункти призначення, пункт пересадки), вартість квитка.
7. Квитки на авіарейси можна придбати завчасно або забронювати в авіакасах.
8. Ціна квитка залежить не тільки від маршруту, але й від часу відльоту (у незручний час – вночі або рано вранці – ціна квитка нижче).
9. До відправлення рейсу, якщо в цьому є необхідність, квиток можна повернути.
10. Авіарейси можуть бути затримані через погодні умови, технічні неполадки, а також можуть бути скасовані, якщо продано менше встановленого мінімуму квитків.
11. Авіарейси можна розділити на наступні категорії: внутрішні, міжнародні, чартерні, вантажоперевезення, спеціальні рейси.
12. Під час посадки в літак пасажир повинен пред’явити квиток та паспорт, а для міжнародного рейсу, окрім вказаного, має пройти митний контроль.
13. Пасажири можуть здавати свої речі в багажне відділення.
14. Hа рейси вантажоперевезень і спеціальні рейси квитки не продаються.
15. Для спец. рейсів не існує розкладу.
16. Квитки на чартерні рейси розповсюджує те агентство, яке його організувало.

Враховуючи вимоги до проєкту та результати структурного аналізу предметної області (пункт 2.1) сформуємо перелік вимог у форматах у форматах Product Backlog (UserStory), UseCase, SRS (Software Requirements Specification).

### **Реалізація запитів у форматі Product Backlog (UserStory)**

Перелік вимог у форматі Product Backlog (UserStory) наведено у таблиці 2.1.

Таблиця 2.1 – Product Backlog (UserStory)

|  |
| --- |
| User Story: |
| Назва: Перелік авіарейсів і загальне число авіарейсів за категорією, напрямком і типом літака. |
| Як користувач системи (касир, начальник служби продажу квитків),  Я хочу отримати перелік авіарейсів за певною категорією, напрямком і типом літака,  Щоб я міг знайти рейси, які відповідають моїм критеріям. |
| Критерії приймання:  Користувач може вказати категорію, напрямок і тип літака.  Система відображає перелік рейсів, що відповідають зазначеним критеріям.  Система надає загальне число знайдених рейсів. |
| User Story: |
| Назва: Інформація про місця на рейсі. |
| Як користувач системи (касир, агент реєстрації квитків, агент реєстрації багажу),  Я хочу дізнатися про перелік та кількість вільних і заброньованих місць на рейсі,  Щоб виконувати професійні обов’язки з продажу реєстрації квитків, багажу. |
| Критерії приймання:  Користувач може вказати такі критерії, як рейс, дату, маршрут, ціну та час вильоту, вік, стать.  Система відображає перелік та кількість вільних і заброньованих місць за обраними критеріями.  Система надає загальне число зданих квитків за обраними критеріями. |
| User Story: |
| Назва: Інформація про літаки в аеропорту. |
| Як користувач системи (диспетчер злітно-посадкової смуги, диспетчер навігаційного обслуговування, начальник служби авіатехнічного обслуговування),  Я хочу отримати перелік та загальне число літаків, які перебувають в аеропорту в зазначений час, за часом надходження в аеропорту, за кількістю здійснених рейсів,  Щоб я міг перевірити статус літаків і їх технічний стан. |
| Критерії приймання:  Користувач може вказати аеропорт і час.  Система відображає перелік літаків, які перебувають в аеропорту.  Система надає інформацію про техогляд і ремонт літаків. |
| User Story: |
| Назва: Інформація про пасажирів на рейсі. |
| Як користувач системи (начальник служби реєстрації, начальник митної служби),  Я хочу дізнатися про перелік та кількість пасажирів на рейсі в певний день,  Щоб я міг аналізувати дані про пасажирів. |
| Критерії приймання:  Користувач може вказати внутрішній або міжнародний рейс, дату, наявність багажу, стать, вік.  Система відображає перелік пасажирів за вищевказаними критеріями.  Система підраховує загальне число пасажирів за вищевказаними критеріями. |
| User Story: |
| Назва: Інформація про пілотів за медоглядом. |
| Як користувач системи (начальник медико-санітарної служби, начальник відділу кадрів),  Я хочу отримати перелік пілотів, які пройшли або не пройшли медогляд у зазначеному році,  Щоб я міг перевірити медичний статус пілотів. |
| Критерії приймання:  Користувач може вказати рік медогляду.  Система відображає перелік та число пілотів з медоглядом.  Система надає перелік та число пілотів за різними критеріями (стать, вік, зарплатня). |
| User Story: |
| Назва: Інформація про рейси за маршрутом, тривалістю перельоту, ціною квитка й по всім цим критеріям разом. |
| Як користувач системи (диспетчер злітно-посадкової смуги, диспетчер навігаційного обслуговування, начальник планово-економічного відділу),  Я хочу дізнатися про перелік та число рейсів за певними критріями,  Щоб я міг знайти рейси, що відповідають моїм критеріям. |
| Критерії приймання:  Користувач може вказати маршрут, тривалість перельоту, ціну квитка  Система відображає перелік та число рейсів за вказаними критеріями. |
| User Story: |
| Назва: Інформація про рейси для літака певного типу та кількість проданих квитків за певними критеріями. |
| Як користувач системи (економіст, бухгалтер - фінансист , начальник диспетчерської служби, начальник аеронавігаційної служби),  Я хочу дізнатися про всі рейси, виконані літаками певного типу, продані квитки за набором критеріїв,  Щоб я міг аналізувати використання літаків цього типу, середню кількість проданих квитків за певними критеріями. |
| Критерії приймання:  Користувач може вказати тип літака, визначений маршрут, тривалість перельоту, ціну квитка.  Система відображає перелік рейсів для літака цього типу.  Система надає середню кількість проданих квитків за вказаними користувачем критеріями. |
| User Story: |
| Назва: Інформація про робітників у відділі, бригаді, що обслуговує рейс за певними критеріями. |
| Як користувач системи (начальник відділу кадрів),  Я хочу одержати перелік та загальне число робітників у бригаді, що обслуговують конкретний рейс, зазначеному відділі за певними критеріями  Щоб я міг перевірити склад бригади, відділу та зарплатню їх робітників. |
| Критерії приймання:  Користувач може вказати відділ, бригаду та рейс, вік, середню зарплатню.  Система відображає перелік робітників у бригаді або відділі за вищевказаними критеріями.  Система підраховує загальне число робітників у бригаді або відділі за вищевказаними критеріями. |
| User Story: |
| Назва: Інформація про скасовані і затримані рейси. |
| Як користувач системи (касир, диспетчер злітно-посадкової смуги, диспетчер навігаційного обслуговування, спеціаліст з планування польотів),  Я хочу дізнатися про всі скасовані та затримані рейси,  Щоб я міг аналізувати причини і наслідки скасувань та затримок. |
| Критерії приймання:  Користувач може вказати напрямок і маршрут, кількість невикористаних місць, відсоткове співвідношення невикористаних місць.  Користувач може вказати причину затримки, маршрут, кількість зданих квитків під час затримки  Система відображає перелік скасованих і затриманих рейсів. |
| User Story: |
| Назва: Інформація про всіх робітників аеропорту. |
| Як користувач системи (Начальник відділу кадрів),  Я хочу отримати перелік всіх робітників аеропорту за певними критеріями,  Щоб я міг переглядати дані про їхній стаж, стать, вік, наявність дітей і зарплатню. |
| Критерії приймання:  Користувач може вказати посаду, стаж, стать, вік, наявність дітей, розмір зарплатні.  Система відображає перелік та загальне число робітників відповідно до зазначених критеріїв.  Система підраховує загальне число робітників відповідно до зазначених критеріїв. |

### **Реалізація запитів у форматі UseCase**

Перелік вимог у форматі UseCase наведено у таблиці 2.2.

Таблиця 2.2 – UseCase

|  |
| --- |
| Назва Use Case: Одержати перелік авіарейсів за категорією, напрямком і типом літака. |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вибирає категорію рейсу.  Користувач вказує напрямок рейсу.  Користувач вказує тип літака.  Система відображає перелік рейсів, що відповідають критеріям.  Система підраховує загальне число знайдених рейсів. |
| Альтернативні сценарії:  Якщо не знайдено рейсів, система відображає повідомлення про відсутність результатів. |
| Результат: Перелік рейсів і загальне число рейсів. |
| Назва Use Case: Одержати інформацію про місця на рейсі |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує рейс, дату, маршрут, ціну і час вильоту.  Система відображає перелік вільних і заброньованих місць на рейсі.  Система підраховує загальне число вільних і заброньованих місць.  Користувач вказує параметри для аналізу зданих квитків (день, маршрут, ціна, вік, стать).  Система відображає загальне число зданих квитків за зазначеними параметрами. |
| Альтернативні сценарії:  Якщо рейс не знайдено, система відображає повідомлення про помилку. |
| Результат: Перелік місць, загальне число місць та зданих квитків. |
| Назва Use Case: Одержати інформацію про літаки в аеропорту |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі |
| Основний сценарій:  Користувач вказує кількість здійснених рейсів, час і аеропорт.  Система відображає перелік літаків, що перебувають в аеропорту в зазначений час.  Система підраховує кількість літаків.  Користувач вказує період техогляду та ремонту, кількість ремонтів, кількість здійснених рейсів до ремонту, вік літака.  Система відображає перелік літаків.  Система підраховує кількість літаків. |
| Альтернативні сценарії:  Якщо вказаний час чи аеропорт не знайдені, система відображає відповідне повідомлення. |
| Результат: Перелік літаків, їх стан і кількість. |
| Назва Use Case: Одержати інформацію про пасажирів рейсу |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі |
| Основний сценарій:  Користувач вказує рейс і дату та додактові параметри параметри (польоти за кордон, багаж, стать, вік).  Система відображає перелік пасажирів на рейсі.  Система підраховує загальне число пасажирів на ресі. |
| Альтернативні сценарії:  Якщо рейс або дата не знайдені, система відображає повідомлення про помилку. |
| Результат: Перелік пасажирів і загальне число пасажирів. |
| Назва Use Case: Одержати інформацію про пасажирів рейсу |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі |
| Основний сценарій:  Користувач вказує рейс і дату та додактові параметри параметри (польоти за кордон, багаж, стать, вік).  Система відображає перелік пасажирів на рейсі.  Система підраховує загальне число пасажирів на ресі. |
| Альтернативні сценарії:  Якщо рейс або дата не знайдені, система відображає повідомлення про помилку. |
| Результат: Перелік пасажирів і загальне число пасажирів. |
| Назва Use Case: Одержати інформацію про пілотів за медоглядом |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує рік для медогляду та додаткові параметри (стать, вік, зарплата).  Система відображає перелік пілотів, що пройшли або не пройшли медогляд.  Система підраховує загальне число пілотів за зазначеними параметрами. |
| Альтернативні сценарії:  Якщо дані не знайдено, система відображає відповідне повідомлення. |
| Результат: Перелік пілотів і загальне число пілотів. |
| Назва Use Case: Одержати інформацію про рейси за маршрутом |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує маршрут та додаткові парметри, (тривалість, ціна).  Система відображає перелік рейсів по зазначеному маршруту.  Система підраховує загальне число рейсів. |
| Альтернативні сценарії:  Якщо маршрут не знайдено, система відображає повідомлення про помилку. |
| Результат: Перелік рейсів і загальне число рейсів за параметрами. |
| Назва Use Case: Одержати інформацію про рейси для літака певного типу |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує тип літака.  Система відображає перелік рейсів, виконаних літаками цього типу.  Система відображає загальне число рейсів, якими літають літаки заданого типу.  Користувач вказує маршрут, тривалість перельоту, час вильоту, ціну.  Система підраховує середню кількість проданих квитків на визначені маршрути. |
| Альтернативні сценарії:  Якщо дані не знайдено, система відображає відповідне повідомлення. |
| Результат: Перелік рейсів і середня кількість проданих квитків. |
| Назва Use Case: Одержати інформацію про робітників у бригаді |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує тип запиту ( всі робітники, по відділам або по бригадам), та додаткові параметри (відділ, рейс, бригду вік та середню зарплату).  Система відображає перелік і загальне число робітників за вказаними параметрами. |
| Альтернативні сценарії:  Якщо бригада не знайдена, система відображає повідомлення про помилку. |
| Результат: Перелік робітників і загальне число робітників у бригаді. |
| Назва Use Case: Одержати інформацію про скасовані і затримані рейси |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує напрямок, маршрут, кількість невикористаних місць, відсоток невикористаних місць.  Система відображає перелік та загальне число скасованих рейсів.  Користувач вказує причину, маршрут та кількість зданих квитків під час затримки.  Система відображає перелік та загальне число затриманих рейсів з причинами затримки. |
| Альтернативні сценарії:  Якщо рейси не знайдено, система відображає відповідне повідомлення. |
| Результат: Перелік скасованих і затриманих рейсів, а також додаткова інформація. |
| Назва Use Case: Одержати інформацію про всіх робітників аеропорту |
| Актор: Користувач системи |
| Передумови: Користувач авторизований у системі. |
| Основний сценарій:  Користувач вказує обирає тип пошуку (всі робітники, начальники відділів, робітники зазначеного відділ) та додатковими параметрами (стаж роботи в аеропорту, статева ознака, вік, ознака наявності й кількості дітей, розмір заробітної платні.).  Система відображає перелік та загальне число робітників. |
| Альтернативні сценарії:  Якщо робітників не знайдено, система відображає відповідне повідомлення. |
| Результат: Перелік робітників, які працюють в зазначених структурних одиницях та мають певний статус. |

### **Реалізація запитів у форматі SRS**

Перелік вимог у форматі Software Requirements Specification наведено у таблиці 2.3.

Таблиця 2.3 – Software Requirements Specification

|  |
| --- |
| Ідентифікатор: REQ-001 |
| Назва: Одержання переліку авіарейсів за категорією, напрямком і типом літака. |
| Опис: Система повинна надати можливість користувачеві отримати перелік авіарейсів, що відповідають зазначеній категорії, напрямку і типу літака. Система також повинна надати загальне число знайдених рейсів. |
| Вхідні дані:  Категорія рейсу.  Напрямок.  Тип літака. |
| Вихідні дані:  Перелік рейсів.  Загальне число рейсів. |
| Сценарії використання:  Користувач вводить категорію рейсу, напрямок і тип літака.  Система генерує перелік рейсів, що відповідають критеріям.  Система підраховує і відображає загальне число рейсів. |
| Параметри успіху:  Перелік рейсів відповідає зазначеним критеріям.  Загальне число рейсів точно відображено. |
| Ідентифікатор: REQ-002 |
| Назва: Одержання інформації про місця на рейсі. |
| Опис:  Система повинна надати інформацію про вільні і заброньовані місця на зазначеному рейсі для конкретного дня за критеріями.  Система повинна також надати загальне число зданих квитків за різними критеріями. |
| Вхідні дані:  Номер рейсу, дата, маршрут, ціна, час вильоту (для місць)  День, маршруту, ціна квитка, вік, стать. |
| Вихідні дані:  Перелік і загальне число місць.  Загальне число зданих квитків. |
| Сценарії використання:  Користувач вводить номер рейсу, дату, маршрут і час вильоту (для місць), День, маршруту, ціна квитка, вік, стать (для квитків).  Система генерує перелік вільних і заброньованих місць за зазначеними критеріями.  Система підраховує загальне число місць і зданих квитків за зазначеними критеріями. |
| Параметри успіху:  Інформація про місця відображається правильно.  Загальне число місць і зданих квитків точно підраховане. |
| Ідентифікатор: REQ-003 |
| Назва: Одержання інформації про літаки за ознакою приписки до аеропорту та за ознакою техогляду. |
| Опис: Система повинна надати перелік літаків за певними критеріями. |
| Вхідні дані:  Час, час надходження в аеропорту, за кількістю здійснених рейсів ( для літаків, приписаних до аеропорту).  Період часу, дата, кількість здійснених рейсів до ремонту, вік літака, кількість ремонтів (для літаків за ознакою техогляду). |
| Вихідні дані:  Перелік літаків.  Загальне число літаків. |
| Сценарії використання:  Користувач вводить аеропорт і час.  Система генерує перелік літаків в аеропорту за припискою та ознакою техогляду.  Система підраховує кількість літаків в аеропорту за припискою та ознакою техогляду. |
| Параметри успіху:  Перелік літаків відповідає вказаним критеріям.  Загальне число літаків точно підраховане. |
| Ідентифікатор: REQ-004 |
| Назва: Одержання інформації про пасажирів на рейсі. |
| Опис: Система повинна надати інформацію про пасажирів на даному рейсі за різними критеріями. |
| Вхідні дані:  Зазначений день, тип рейсу, наявність багажу стать і вік. |
| Вихідні дані:  Перелік пасажирів.  Загальне число пасажирів. |
| Сценарії використання:  Користувач вводить номер рейсу.  Система генерує перелік пасажирів та підраховує загальне число пасажирів за зазначеними критеріями. |
| Параметри успіху:  Перелік пасажирів відповідає вказаним критеріям.  Загальне число пасажирів точно підраховане. |
| Ідентифікатор: REQ-005 |
| Назва: Одержання інформації про пілотів за медоглядом. |
| Опис: Система повинна надати перелік пілотів, які пройшли або не пройшли медогляд у зазначеному році за різними критеріями. |
| Вхідні дані:  Рік медогляду, статева ознака, вік, розмір заробітної плати. |
| Вихідні дані:  Перелік пілотів.  Загальне число пілотів. |
| Сценарії використання:  Користувач вводить рік медогляду та вищезгадані критерії.  Система генерує перелік пілотів та загальне число. |
| Параметри успіху:  Перелік пілотів відповідає вказаним критеріям.  Загальне число пілотів точно підраховане. |
| Ідентифікатор: REQ-006 |
| Назва: Одержання інформації про рейси по маршруту. |
| Опис: Система повинна надати перелік рейсів по зазначеному маршруту за ціною квитка, тривалістю перельотів, кількістю рейсів та за всіма зазначеними критеріями. |
| Вхідні дані:  Маршрут, ціна квитка, тривалість перельотів, кількість рейсів. |
| Вихідні дані:  Перелік рейсів.  Загальне число рейсів. |
| Сценарії використання:  Користувач вводить маршрут.  Система генерує перелік рейсів та загальне число рейсів за зазначеними критеріями. |
| Параметри успіху:  Перелік рейсів відповідає вказаним критеріям.  Загальне число рейсів точно підраховане. |
| Ідентифікатор: REQ-007 |
| Назва: Одержання інформації про рейси для літака певного типу. |
| Опис: Система повинна надати перелік та число рейсів, виконаних літаками певного типу.  Система повинна надати середню кількість проданих квитків на визначені маршрути за різними критеріями. |
| Вхідні дані:  Тип літака ,тривалість перельоту, ціна квитка, час вильоту (для отримання переліку та кількості рейсів)  Тривалість перельоту, за ціна квитка, часом вильоту (для середньої кількості проданих). |
| Вихідні дані:  Перелік рейсів.  Середня кількість проданих квитків. |
| Сценарії використання:  Користувач вводить тип літака ,тривалість перельоту, ціна квитка, час вильоту.  Система генерує перелік та загальне число рейсів для літака цього типу.  Користувач вводить тривалість перельоту, за ціна квитка, часом вильоту.  Система підраховує середню кількість проданих квитків за зазначеними критеріями. |
| Параметри успіху:  Перелік рейсів відповідає вказаним критеріям.  Загальне число рейсів точно підраховане.  Середня кількість проданих квитків точно підрахована. |
| Ідентифікатор: REQ-008 |
| Назва: Одержання інформації про робітників у бригаді. |
| Опис: Система повинна надати перелік та число робітників у бригаді за певними критеріями. |
| Вхідні дані:  Відділ, рейс, вік, середня зарплата. |
| Вихідні дані:  Перелік робітників.  Загальне число робітників.  Середня зарплата. |
| Сценарії використання:  Користувач вводить відділ, рейс, вік, середня зарплата.  Система генерує перелік робітників у бригаді.  Система підраховує загальне число робітників. |
| Параметри успіху:  Перелік робітників відповідає вказаним критеріям.  Загальне число робітників точно підраховане. |
| Ідентифікатор: REQ-009 |
| Назва: Одержання інформації про перелік та загальне число скасованих і затриманих рейси. |
| Опис: Система повинна надати перелік скасованих і затриманих рейсів. |
| Вхідні дані:  Напрям, маршрут, кількість невикористаних місць, відсоткове співвідношення, невикористаних місць |
| Вихідні дані:  Перелік скасованих рейсів.  Перелік затриманих рейсів.  Число скасованих рейсів.  Число затриманих рейсів. |
| Сценарії використання:  Користувач вводить вищезгадані критерії.  Система генерує перелік скасованих рейсів.  Система генерує перелік затриманих рейсів.  Система підраховує число скасованих рейсів.  Система підраховує число затриманих рейсів. |
| Параметри успіху:  Перелік рейсів відповідає вказаним критеріям.  Загальне число рейсів точно підраховане. |
| Ідентифікатор: REQ-010 |
| Назва: Одержання інформації про всіх робітників аеропорту |
| Опис: Система повинна надати перелік та загальне число робітників аеропорту за певними критеріям |
| Вхідні дані:  Статус, стаж роботи, стать, вік, наявність дітей, зарплата |
| Вихідні дані:  Перелік робітників.  Загальне число робітників. |
| Сценарії використання:  Користувач вводить вищезгадані критерії.  Система генерує перелік робітників.  Система підраховує загальне число робітників. |
| Параметри успіху:  Перелік робітників відповідає вказаним критеріям.  Загальне число робітників точно підраховане. |

**2.3. Виокремлення сутностей об’єктів структурного аналізу предметної області**

# СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ДОДАТКУ

## **Структура програми**

### **Модулі програми**

### BlackThemeMainView – клас який реалізує введення даних користувачем у елементи управління для виконання операцій та відображення результатів у таблиці DataGridView.

BlackThemeModifData – клас, який відображує довідкову інформацію, щодо внесення змін до БД (видалення, додавання, редагування) через таблицю головного вікна програми.

BlackThemeAboutAutor – клас який відображує довідкову інформацію, про автора в темній колірній гамі.

BlackThemePanelElments – клас, який відображує довідкову інформацію, щодо роботи елементів панелі управління.

BlackThemeSeletionData – клас, який відображує довідкову інформацію, щодо відбору даних в таблиці головного вікна програми (сортування, фільтрація, пошук).

IMainView – інтерфейс для зв’язку класів View (BlackThemMainView та LightThemeMainView) з класом MainWindowPresentor.

MainPresentor – клас, який реалізує логічні операції програмного додатку та виводить результати в головне вікно програми, завдяки підписці на події та отриманню параметрів View за допомогою інтерфейсу IMainView.

LightThemeMainView – клас, який наслідується від BlackThemeMainView та має видозмінену світлу кольорову гаму, без зміни логіки роботи.

LightThemeModifData – клас, який наслідується від BlackThemeModifData та має видозмінену світлу кольорову гаму.

LightThemePanelElements – клас, який наслідується від BlackThemePanelElements та має видозмінену світлу кольорову гаму.

LightThemeSeletionData – клас, який наслідується від BlackThemeSeletionData та має видозмінену світлу кольорову гаму.

Albom – клас, який реалізує прототип альбому, який має властивості: назва, виконавець перелік пісень, кількість пісень.

MemStorage – клас, який наслідується від класу Albom та реалізує прототип носію інформації, який має властивості: тип носія, час звучання.

Product – клас, який наслідується від класу MemStorage та реалізує прототип товару, який має властивості: каталоговий номер, ціна.

ListBase – клас, що містить всередині список, в який зчитуються дані з текстового файлу під час виконання програми, та записує й зберігає дані в текстовий файл, при внесенні змін в список, реалізуючи модель БД.

ListTable – клас, що функціонально копіює поле-список з класу ListBase, для реалізації операцій фільтрації, сортування та пошуку, редагування й додавання елементів таблиці головного вікна, без збереження змін у записи текстового файлу.

### **Алгоритми роботи програми**

Загальний алгоритм роботи основної програми та підпрограм описані у вигляді блок-схем в Додатку.А:

1. Рисунок А.1 – Загальна блок-схема алгоритму роботи програми.
2. Рисунок А.2 – Узагальнена блок-схема алгоритму сортування даних БД.
3. Додаток А.3 – Узагальнена блок-схема алгоритму фільтрації даних БД.
4. Додаток А.4 – Узагальнена блок-схема алгоритму пошуку записів БД.
5. Додаток А.5 – Узагальнена блок-схема алгоритму зміни вмісту комірки таблиці головного вікна.
6. Додаток А.6 – Узагальнена блок-схема додавання пустого рядка в таблицю.
7. Додаток А.7 – Узагальнена блок-схема додавання запису в БД.
8. Додаток А.8 – Узагальнена блок-схема збереження редагування запису в БД.
9. Додаток А.9 – Узагальнена блок-схема видалення елементу з БД.
10. Додаток А.10 – Узагальнена блок-схема роботи меню головного вікна.

## **Опис методів програми**

Список методів класу BlackThemeMainView та їх опис наведено в табл.1.

Таблиця 1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | AssociateAndRaiseViewEvents() | Здійснює виклик подій головного вікна та їх прив’язку до подій, що передаватимуться у MainWindowPresentor.. |
| 2 | dataGridView1\_CellBeginEdit(object sender,DataGridViewCellCancelEventArgs e) | Здійснює запам’ятовування даних комірки на початку редагування, для використання перевірки даних комірки у функції класу MainWindowPresentor. |
| 3 | ChooseSearchButton\_Click (object sender, EventArgs e) | Здійснює переключення меню обробки даних на вкладку «Пошук». |

Продвження табл.1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | СhooseSortButton\_Click(object sender, EventArgs e) | Здійснює переключення меню обробки даних на вкладку «Сортування». |
| 5 | ChooseFiltrButton\_Click(object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Фільтрація» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 6 | SelectionToolStripMenuItem\_Click\_1(object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Відбір даних» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 7 | ModifyDataToolStripMenuItem\_Click\_1  (object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Зміна даних» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 8 | ElementPanelToolStripMenuItem\_Click\_1  (object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Елементи панелі меню» при натисненні на відповідний підпункт, пункту панелі меню «Про програму». |
| 9 | AboutDeveloperToolStripMenuItem\_Click (object sender, EventArgs e) | Здійснює відкриття інформаційного вікна «Про розробника» при натисненні на відповідний пункт панелі меню. |
| 10 | LightThemToolStripMenuItem\_Click(object sender, EventArgs e) | Здійснює заміну темної теми на світлу при натисненні на відповідний підпункт пункту панелі меню «Зміна теми». |

Продвження табл.1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 11 | RestartToolStripMenuItem\_Click(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Оновити» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 12 | AddElToolStripMenuItem\_Click(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Додати новий елемент в базу» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 13 | RedactToolStripMenuItem(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Зберегти редагування в базу» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 14 | RemoveToolStripMenuItem1\_Click(object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Видалити» при натисненні на відповідний підпункт пункту панелі меню «Інструменти». |
| 15 | AddRowInTableToolStripMenuItem\_Click  (object sender, EventArgs e) | Здійснює виклик події натиснення кнопки «Додати рядок в таблицю» при натисненні на відповідний підпункт пункту панелі меню «Інструменти» . |
| 16 | SearchingElment\_KeyDown(object sender, KeyEventArgs e) | Прибирає фокус з поля пошуку при натисненні кнопки «Enter». |

Продвження табл.1 - Основні методи класу BlackThemeMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 17 | FiltrationByPriceFrom\_KeyDown(object sender, KeyEventArgs e) | Прибирає фокус з поля верхньої границі фільтрації при натисненні кнопки «Enter». |
| 18 | FiltrationByPriceTo\_KeyDown(object sender, KeyEventArgs e) | Прибирає фокус з поля нижньої границі фільтрації при натисненні кнопки «Enter». |

Список методів інтерфейсу IMainView та їх опис наведено в табл.2.

Таблиця 2 – Основні методи інтерфейсу IMainView

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | SetTableSoure(BindingSource TABELELIST) | Здійсню виклик подій головного вікна та їх прив’язку до подій, що передаватимуться у MainWindowPresentor.. |
| 2 | IfDataInTableSelected() | Здійснює запам’ятовування даних комірки на початку редагування, для використання перевірки даних комірки у функції MainWindowPresentor. |
| 3 | GetCatalogNumberOFSelectedElement() | Здійснює передачу значення каталогового номера виділеного елемента у MainWindowPresentor. |
| 4 | GetFiltrationByPriceFrom() | Здійснює передачу значення верхньої границі фільтрації у MainWindowPresentor. |
| 5 | GetFiltratiц33 onByPriceTo() | Здійснює передачу значення верхньої границі фільтрації у MainWindowPresentor. |
| 6 | GetEditingСolumnIndex() | Здійснює передачу в MainWindowPresentor номеру стовпчика змінюваної комірки. |

Продовження табл.2 – Основні методи інтерфейсу IMainView

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 7 | GetPreviousValue() | Здійснює передачу у MainWindowPresentor значення комірки перед зміною. |
| 8 | GetChangedValue() | Здійснює передачу у MainWindowPresentor значення комірки таблиці після редагування. |
| 9 | GetSelectedRowIndex() | Здійснює передачу у MainWindowPresentor номера виділеного рядка. |
| 10 | GetSearchingElment() | Здійснює передачу у MainWindowPresentor текст поля пошуку. |
| 11 | GetSortDirection() | Здійснює передачу у MainWindowPresentor напряму сортування. |
| 12 | GetTypeOfSorting() | Здійснює передачу у MainWindowPresentor типу сортування. |
| 13 | CancelChangesInTableCell() | Здійснює скасування змін у комірці таблиці. |
| 14 | СlearAllFields() | Здійснює очищення у MainWindow всіх полів обробки даних. |
| 15 | СlearSortFields() | Здійснює очищення у MainWindow всіх полів обробки даних. |
| 16 | ClearFiltraionFields() | Здійснює очищення у MainWindow поля фільтрації. |

Список методів класу MainPresentor та їх опис наведено в табл.3.

Таблиця 3 – Основні методи класу MainPresentor

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | ReloadViewTable() | Здійснює відміну всіх незбережених змін у таблиці. |
| 2 | ReloadAllInformation() | Здійснює відміну всіх незбережених змін у таблиці та очищує всі поля обробки даних. |
| 3 | AddNewElementToTable() | Здійснює додавання нового рядка в таблицю. |
| 4 | ControlDataInCell() | Здійснює перевірку даних таблиці та скасовує зміни при введенні некоректних значень. |
| 5 | ErrorInData() | Здійснює вивід помилки при вводі некоректних типів даних у комірки таблиці головного вікна. |
| 6 | RedactElementInBase() | Здійснює збереження змін рядка таблиці головного вікна у базу даних. |
| 7 | ReloadDataInTable() | Здійснює перезавантаження даних у таблицю головного вікна. |
| 8 | ChangingBySearchLine() | Здійсню оновлення таблиці та скидання полів сортування та фільтрації головного вікна при повному стиранні значення пошукового поля. |
| 9 | Sort() | Здійсню сортування даних таблиці головного вікна. |
| 10 | RemoveElement() | Здійснює видалення елементу з бази даних. |
| 11 | AddElementInBase() | Здійснює додавання нового елементу у базу даних. |
| 12 | CancelChangesInTableCell() | Здійснює скасування змін у комірці таблиці. |

Список методів класу ListTable та їх опис наведено в табл.4.

Таблиця 4 – Основні методи класу ListTable

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 | 3 |
| 1 | RewriteDataInViewsByDataFromDatabase  (ListBase ChangedBase) | Здійснює перезапис даних списку-копії даними з БД. |
| 2 | TestIfSelectedElementNotNull(int NumberOfRowSelectElement) | Здійснює перевірку чи заповнений виділений рядок таблиці. |
| 3 | IfElementChanged(Product TestingElement, ListBase DatabaseData) | Здійснює перевірку чи даний елемент таблиці змінено. |
| 4 | IfThereNonUnsavedChangesInTable(ListBase DatabaseData) | Здійснює перевірку чи немає незбережених змін елементів таблиці. |
| 5 | IfThereNonUnsavedElementInTable(ListBase DatabaseData) | Здійснює перевірку чи немає нових незбережених елементів таблиці. |
| 6 | SortingByCatalogNumber(ref List<Product> ListBeSorted,string Direction) | Здійснює сортування списку за каталоговим номером |
| 7 | SortingByPrice(ref List<Product> ListBeSorted, string Direction) | Здійснює сортування списку за ціною. |
| 8 | SortingByNumberOfSongs(ref List<Product> ListBeSorted, string Direction) | Здійснює сортування списку за кількістю пісень. |
| 9 | SortingByGroupName(ref List<Product> ListBeSorted, string Direction | Здійсню сортування даних таблиці головного вікна за назвою групи. |
| 10 | SearchByName(string Name, ListBase DataFromBase) | Здійснює пошук в базі даних за назвою носія. |
| 11 | SearchByCatalogNumber(string CatalogNumber, ListBase DataFromBase) | Здійснює пошук в базі даних за каталоговим номером. |
| 12 | SearchBySongName(string SongName, ListBase DataFromBase) | Здійснює пошук в базі даних за назвою пісні. |

Продовження табл.4 – Основні методи класу ListTable

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 13 | SearchByGroupName(string Name, ListBase DataFromBase) | Здійснює пошук в базі даних за назвою групи. |
| 14 | public bool TestCountOfSongs(Product ProductItem) | Здійснює перевірку чи відповідності кількості пісень, до реальної. |
| 15 | TestIFCatalogNumberExist(int CatatlogNumber, ListBase DataFromBase) | Здійснює перевірку чи запис з даним каталоговим номером є в базі даних. |

Список методів класу ListBase та їх опис наведено в табл.5.

Таблиця 5 – Основні методи класу ListBase

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Опис методу |
| 1 | 2 |  |
| 1 | RemoveElementFromBase(int DeleteCatalogNumber) | Здійснює видалення елементу з БД за каталоговим номером. |
| 2 | WriteData( List<Product> storage) | Здійснює запис зі списку даних БД у текстовий файл. |
| 3 | WriteToFIle(string PathToFile, List<List<string>> list) | Здійснює запис списку рядків у текстовий файл бази даних і є допоміжним для методу WriteData. |
| 4 | IFElementExistInBase(int CatalogNumberOfElmenet) | Здійснює перевірку чи товар з даним каталоговим номером існує в БД. |
| 5 | ReadData() | Здійснює зчитування даних з файлу БД в список даних БД. |

Продовження табл.5 – Основні методи класу ListBase

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 7 | public void FormatingData(Product ProductItem) | Здійснює форматування даних перед збереженням у текстовий файл. |
| 6 | ReadInformationFromDataFile(string pathtofile) | Здійснює зчитування даних з текстового файлу у список рядків та є допоміжним для методу ReadData. |

## **Засоби розробики**

Програмний додаток створено на мові С# за допомогою середовища розробки програмних продуктів Visual Studio. Ряд продуктів компанії Microsoft під торговою маркою Microsoft Visual Studio являє собою потужний засіб, який містить у собі інтегроване середовище розробки (IDE) програмного забезпечення та низку інших корисних інструментальних засобів.

Ігри та програми з графічним інтерфейсом (також з підтримкою технології Windows Forms, UWP), веб-сайти, веб-додатки, веб-служби в коді всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI та інших – все це можливо реалізувати за допомогою вищевказаної лінійки продуктів.

IDE Visual Studio об’єднує можливість написання, редагування, налагодження та складання коду, а також подальше розгортання програми. Крім того Visual Studio включає компілятори, засоби завершення коду, систему керування версіями, розширення та багато інших функцій для покращення кожного етапу розробки програмного забезпечення. Зокрема редактор вихідного коду, технологію IntelliSense, налагоджувач, редактор форм, веб-редактор, дизайнер класів значно спрощують створення програмного продукту.

Таким чином, Microsoft Visual Studio є зручним для виконання всього циклу розробки різноманітного ПЗ в одному місці для різноманітних платформ Windows.

## **Опис користувацького інтерфейсу**

Головне вікно, яке відкривається після запуску програми, є основним інтерфейсом для взаємодії користувача з БД, і містить наступні елементи (див. рис.2.1):

* панель меню (позначено цифрою 1);
* панель з графічними елементами управління (позначено цифрою 2);
* вкладки функціоналу обробки даних БД (позначено цифрою 3);
* таблиця для відображення записів БД та результатів взаємодії з даними (позначено цифрою 4).

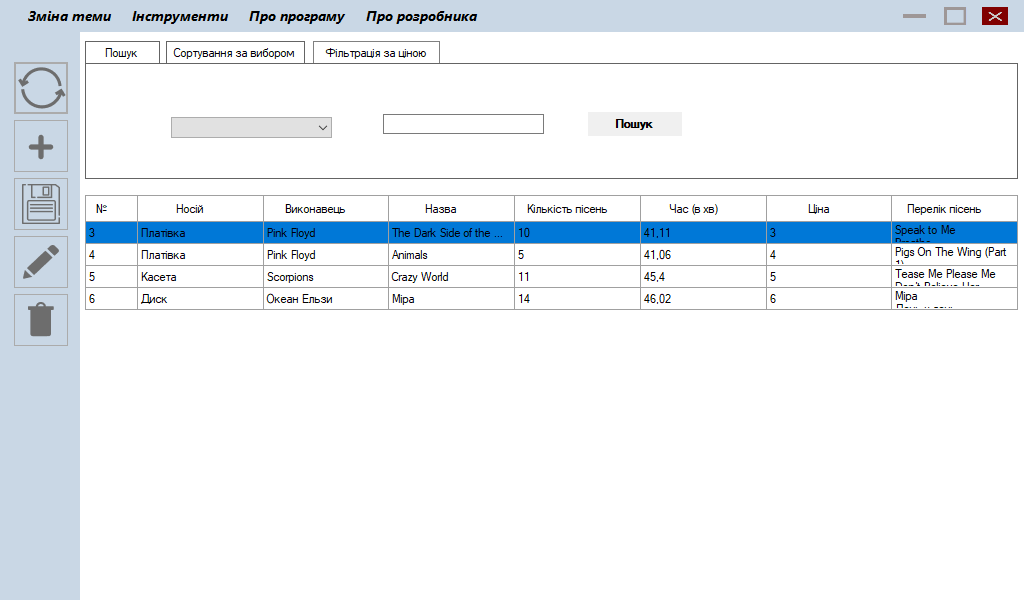


Рисунок 2.1 – Структурні елементи головного вікна

**2**

**3**

**1**

**4**

В панелі меню можна вибрати світлу та темну теми інтерфейсу програми (див. рис.2.2).



Рисунок 2.2 – Пункт меню «Зміна теми»

Пункт меню «Інструменти» (див. рис.2.3) дозволяє формувати БД (додавати, редагувати, видаляти елементи БД і т.д.). Функціонал пункту меню дублює панель з графічними елементами меню управління.

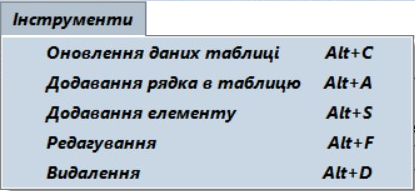


Рисунок 2.3 – Пункт меню «Інструменти»

Пункт меню «Про програму» (див. рис.2.4) містить інформацію про роботу програми (кожен підпункт містить посилання на відповідне інформаційне вікно).

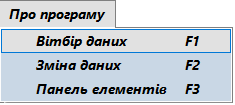


Рисунок 2.4 – Пункт меню «Про програму»

Пункт меню «Про розробника» посилається на вікно, яке містить інформацію про розробника програми.

В панелі графічних елементів управління (див. рис.2.5) знаходяться наступні функціональні кнопки:

* «Оновити» (позначено цифрою 1);
* «Додати ряд у таблицю» (позначено цифрою 2);
* «Додати до бази» (позначено цифрою 3);
* «Редагувати» (позначено цифрою 4);
* «Видалити» (позначено цифрою 5).



Рисунок 2.5 – Панель з графічними елементами управління

Кнопка «Оновити» перезавантажує записи таблиці даними з БД, використовується для відміни змін таблиці, незбережених у базу і скидання до необробленого пошуком, фільтрацією і сортуванням вигляду. При цьому виводиться за допомогою MessageBox попереджувальне повідомлення про втрату незбережених даних (див. рис.2.6).

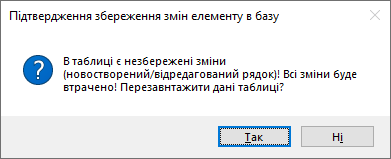


Рисунок 2.6 – Діалогове вікно до функції «Оновити»

Кнопка «Додати ряд у таблицю» додає новий незаповнений рядок у таблицю (за умови якщо немає попередньо доданого рядка, незбереженого у базу). У випадку наявності незбереженого рядка при натисканні на кнопку виводиться відповідне інформаційне повідомлення (див. рис.2.7).

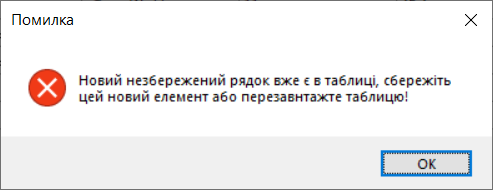


Рисунок 2.7 – Інформаційне повідомлення до функції «Додати ряд у таблицю»

Кнопка «Додати до бази» дозволяє зберегти новий рядок у базу. Після натискання кнопки виводиться за допомогою MessageBox діалогове вікно для підтвердження додавання у базу (див. рис.2.8), в якому треба натиснути кнопку «Так» для збереження даних у базу.

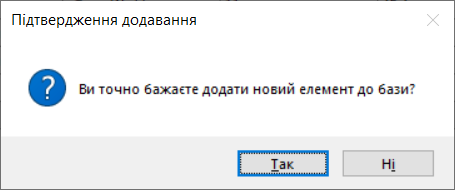


Рисунок 2.8 – Діалогове вікно до функції «Додати до бази»

У випадку якщо рядок незаповнений або такий запис вже є в базі, виведуться відповідні інформаційні повідомлення за допомогою MessageBox (див. рис.2.9).

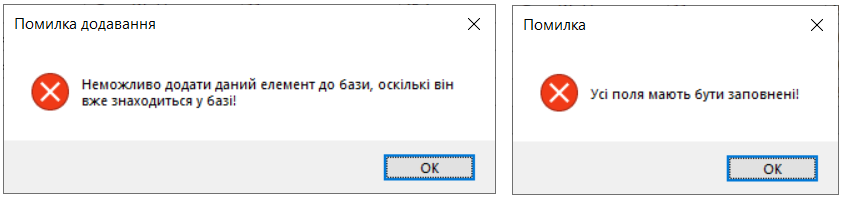


Рисунок 2.9 – Інформаційні повідомлення до функції «Додати до бази»

я

Редагування елементу бази даних здійснюється наступним чином. Спочатку вносяться зміни в обраний рядок. Для збереження внесених змін потрібно натиснути кнопку «Редагувати», після чого відбувається збереження виділеного рядка. У випадку, якщо рядок відсутній у базі, незаповнений або зміни не внесені, виведуться відповідні інформаційні повідомлення за допомогою MessageBox (див. рис.2.10).

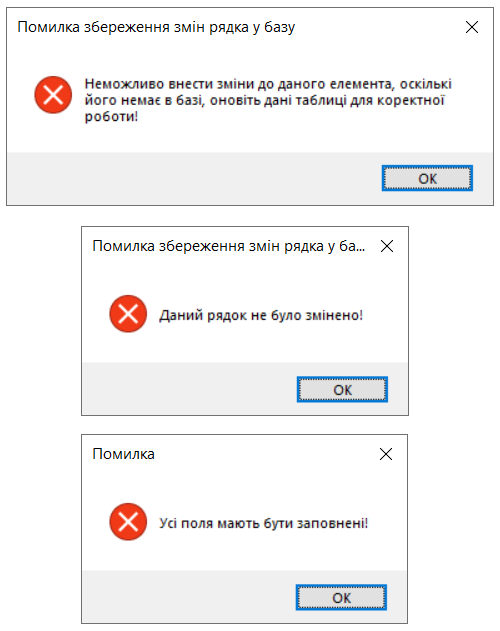


Рисунок 2.10 – Інформаційні повідомлення до функції «Редагувати»

Кнопка «Видалити» дозволяє виключити обраний в таблиці рядок з бази. Після натискання кнопки за допомогою MessageBox виводиться діалогове вікно для підтвердження виконання дії видалення (див. рис.2.11).

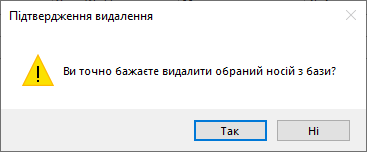


Рисунок 2.11 – Діалогове вікно до функції «Видалити»

При у мові якщо рядок доданий до таблиці, але незбережений в БД, або не має не заповнені комірки, за допомогою MessageBox виведуться відповідні інформаційні повідомлення (див. рис.2.12).

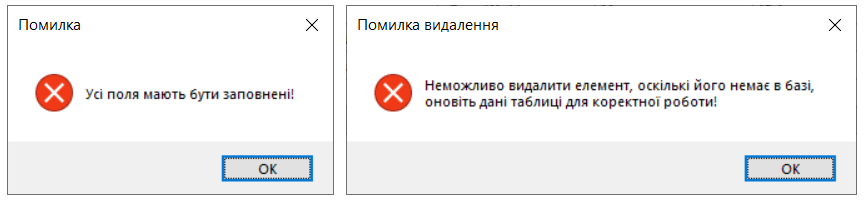


Рисунок 2.12 – Інформаційні повідомлення до функції «Видалити»

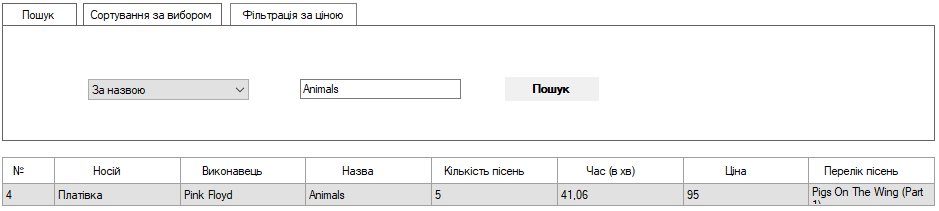
Меню функціоналу обробки БД містить наступні елементи:

* вкладка «Пошук»;
* вкладка «Сортування»;
* вкладка «Фільтрація за ціною».

Вкладка «Пошук» (див. рис.2.13) містить наступні елементи:

* поле «Тип пошуку»;
* поле «Елемент пошуку»;
* кнопка «Пошук».

Поле «Тип пошуку» (позначено цифрою 1) дозволяє обрати певні критерії, за якими буде виконуватись відбір необхідних користувачу записів БД. Поле «Елемент пошуку» (позначено цифрою 2) використовується для введення даних, які відповідають критеріям типу пошуку. Після вибору необхідного критерія та елементу пошуку, при натисканні на кнопку «Пошук» (позначено цифрою 3), здійснюється процедура відбору певних записів БД.



**1**

**2**

**3**

Рисунок 2.13 – Вкладка «Пошук»

Якщо поле «Тип пошуку» або «Елемент пошуку» не заповнені, виведуться відповідні інформаційні повідомлення (див. рис.2.14)

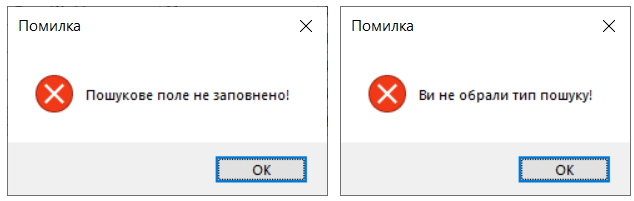
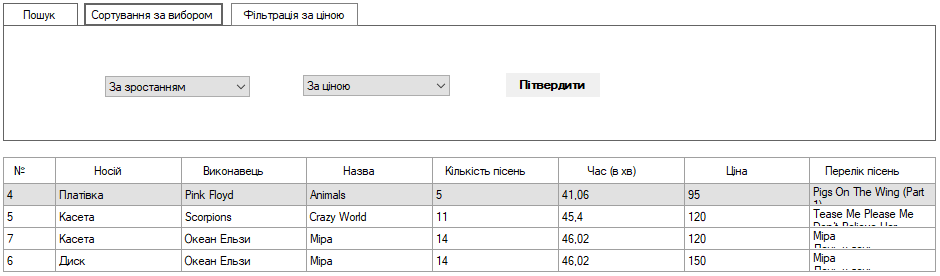


Рисунок 2.14 – Інформаційні повідомлення до функції «Пошук»

Вкладка «Сортування» (див. рис.2.15) містить наступні елементи:

* поле «Напрям сортування»;
* поле «Тип сортування»;
* кнопка «Підтвердити».

Поле «Напрям сортування» (позначено цифрою 1) використовується для сортування за зростанням або спаданням критерію сортування. Поле «Тип сортування» (позначено цифрою 2) дозволяє вибрати критерій сортування. Після натискання кнопки «Підтвердити» (позначено цифрою 3) виконується процедура сортування з виведенням результату в таблицю.



**1**

**2**

**3**

Рисунок 2.15 – Вкладка «Сортування за вибором»

У випадку некоректного заповнення полів «Напрям сортування» або «Тип сортування» або відсутності даних в одному з полів виведуться відповідні інформаційні повідомлення (див. рис.2.16).

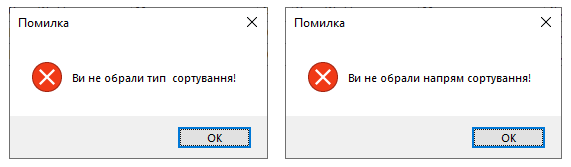
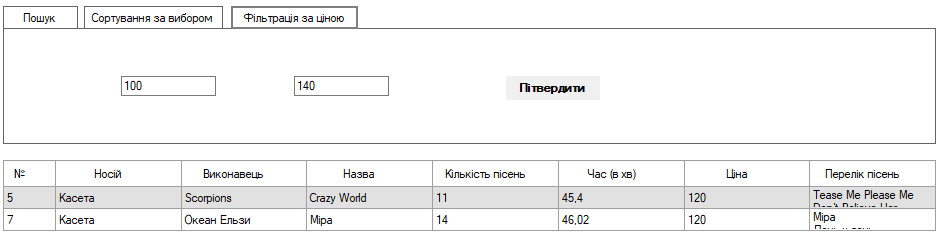


Рисунок 2.16 – Інформаційні повідомлення до функції «Сортування»

Вкладка «Фільтрація за ціною» (див. рис.2.17) містить наступні елементи:

* поле «Від»;
* поле «До»;
* кнопка «Підтвердити»

Поля «Від» (позначено цифрою 1) та «До» (позначено цифрою 2) використовуються для введення верхньої та нижньої границь фільтрації. При натисканні кнопки «Підтвердити» (позначено цифрою 3) виконується процедура фільтрації з виведенням результату в таблицю.



**1**

**2**

**3**

Рисунок 2.17 – Вкладка «Фільтрація за ціною»

У випадку некоректного заповнення полів «Від» або «До» або відсутності даних в обох полях виведуться відповідні інформаційні повідомлення (див. рис.2.18). При наявності заповненого одного з полів, процедура фільтрації виконується в будь-якому випадку.

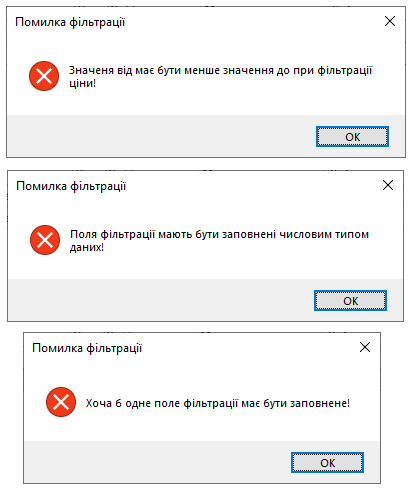


Рисунок 2.18 – Інформаційні повідомлення до функції «Фільтрація»

Таблиця (див. рис.2.19) як елемент головного вікна призначений для відображення всіх записів, результатів їх обробки (за допомогою функціоналу обробки БД) та внесення змін (за допомогою меню інструментів) до БД.

Таблиця містить наступні стовбці:

* «Каталоговий номер»;
* «Носій»;
* «Виконавець»;
* «Кількість пісень»;
* «Час»;
* Ціна;
* «Перелік пісень».



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

Рисунок 2.19 – Таблиця в головному вікні

Для внесення змін в таблиці потрібно двічі натиснути лівою кнопкою миші на відповідну комірку обраного для змін рядка. Комірка перейде у режим редагування і користувач зможе ввести та зберегти зміни, натиснувши Enter. Зміни попередньо зберігаються тільки в таблиці без збереження в БД (для збереження в базу змін таблиці використовуються відповідні пункти меню інструментів).

При введені некоректних даних в комірку стовбця «Каталоговий номер» (позначено цифрою 1), після натиснення «Enter» виводяться відповідні інформаційні повідомлення (див. рис.2.20), в залежності від типу помилки.

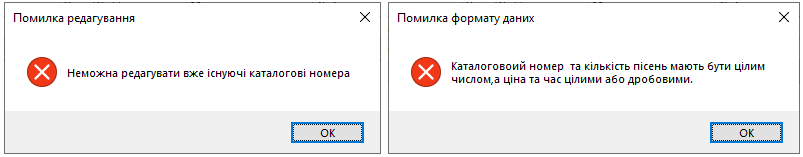


Рисунок 2.20 – Інформаційні повідомлення до комірки стовбця «Каталоговий номер»

Введення некоректних даних в комірку стовбця «Носій» (позначено цифрою 2) після натиснення «Enter» викликає повідомлення з інформацією про можливі значення комірки (див. рис.2.21).

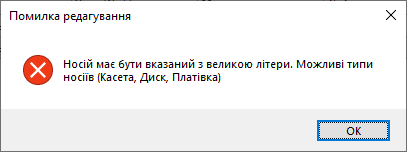


Рисунок 2.21– Інформаційне повідомлення до комірки стовбця «Носій»

Формат введених даних в комірки стовбців «Виконавець» та «Назва» (позначено цифрами 3 та 4) є довільним та не викликає повідомлень з інформацією про можливі помилки.

Комірка стовбця «Кількість пісень» (позначено цифрою 5) може містити тільки цілочисельний тип даних та викликає відповідне інформаційне вікно (див. рис.2.22) при некоректному вводі.

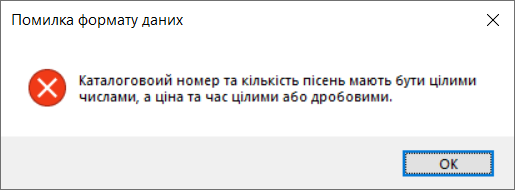


Рисунок 2.22 – Інформаційне повідомлення до комірки стовбця «Кількість пісень»

Формат введених даних в комірки стовбців «Час» та «Ціна» (позначено цифрами 6 та 7) є цифровим з довільним значенням та викликає відповідне інформаційне вікно (див. рис.2.23) при некоректному вводі.

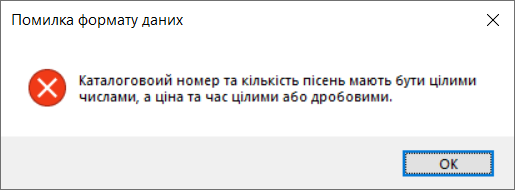


Рисунок 2.23 – Інформаційне повідомлення до комірок стовбців «Час» та «Ціна»

На відміну від попередніх стовбців, комірка стовбця «Перелік пісень» має особливий формат введення даних. Через свій великий об’єм текст не відображається в комірці в повному обсязі. Щоб одержати повну інформацію про вміст комірки достатньо навести курсор на комірку. В такому випадку з’являється спливаюче вікно з повним переліком пісень (див. рис.2.24).

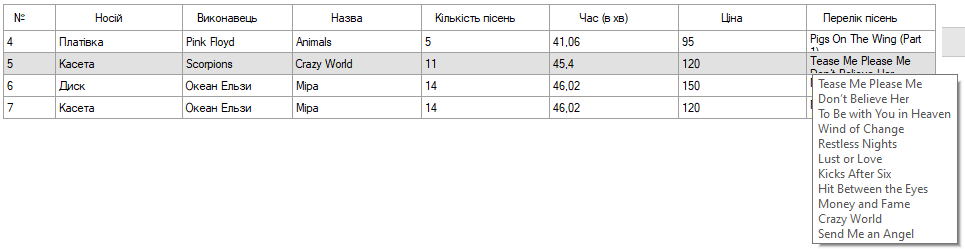


Рисунок 2.24 – Спливаюча інформація для комірки стовбця «Перелік пісень»

Дані в комірку стовбця «Перелік пісень» додаються за певною процедурою. Після введення чергової назви пісні натискання комбінації клавіш «Shift+Enter» дозволяє зробити перенесення курсору на наступний рядок без виходу з режиму редагування комірки. Якщо користувачу потрібно повернутись до попереднього рядка комірки, достатньо використати клавішу «Вгору». Для завершення редагування потрібно натиснути «Enter».

**ВИСНОВКИ**

В процесі виконання курсового проекту на тему «Створення інформаційно-довідкової системи «Музичні товари» засобами мови С#» створено програмний додаток з функціоналом керування БД з певною спеціалізацією.

Отримано цінні навички:

* роботи з середовищем розробки Microsoft Visual Studio на мові С#;
* самостійного пошуку методів розв’язку проблем в ході виконання поставлених задач;
* практичної роботи щодо процесу керування та обробки записів БД;
* виконання оформлення результатів роботи в вигляді докладного та добре аргументованого тексту пояснювальної записки;

Набутий досвід дає повноцінне уявлення про:

* можливості створення додатку на мові С# на базі Windows Forms;
* використання принципів ООП;
* формування архітектури та структури додатка;
* особливості функціоналу програмного продукту.

Додаток є повноцінним застосунком:

* із забезпеченням вимог стійкості до будь-яких можливих дій користувача;
* з архітектурою та структурою, яка дає можливість у майбутньому розширити програму шляхом додавання нового функціоналу та видозмінення логіки обробки.

Одержаний програмний продукт повністю готовий до використання пересічним споживачем без спеціальної підготовки. Застосунок призначений для особистого використання та малого бізнесу в сфері торгівлі музичним контентом. При створені додатку враховано необхідність його використання на базі Windows-пристроїв з обмеженими обчислювальними ресурсами.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

Електронні джерела:

1. <https://studfile.net/preview/9703158/page:15/>
2. <https://studfile.net/html/2706/981/html_tl7tvfxi1C.1fap/htmlconvd-4_BCVC_html_f4cb27953e5e2fd0.jpg>

**ДОДАТКИ**

**ДОДАТОК А**

Департамент повітряного руху

Диспетчерська служба

Аеронавігаційна служба

Льотна служба

Служба радіозв'язку

Служба авіатехнічного обслуговування

Служба обслуговування аеродрому

Служба ремонту спец-приладдя

Департамент технічного обслуговування

Служба безпеки

Митна служба

Медико – санітарна служба

Департамент спеціалізованих служб

Адміністрація

Відділ кадрів

Юридичний відділ

Канцелярія

Департамент адміністративних та додаткових служб

Експлуатаційний департамент

Служба паливо-мастильних матеріалів

Служба спецтранспорта

Рятувальна служба

Економічний департамент

Планово-економічний відділ

Фінансовий відділ

Відділ праці

Служба продажу квитків

Клінінгова служба

Служба безпеки

Рисунок А.1 – Структурно-організаційна блок-схема служб аеропорту

Планово-економічний відділ

Відділ праці

Служба продажу квитків

Фінансовий відділ

Економісти

Бухгалтери-фінансисти

Нормувальники

Касири

Начальник економічного департаменту

Рисунок А.2 – Структурно-організаційна блок-схема служб економічного департаменту

Служба паливо-мастильних матеріалів

Служба спецтранспорта

Рятувальна служба

Бригади заправників

Водії вантажного транспорту

Пожежники-рятувальники

Бригади рятувальників

Начальник департаменту експлуатаційних служб

Перевізники палива

Кілінгова служба

Бригади прибирання салону

Бригади прибирання аеропорту

Прибиральники

Прибиральники

Заправники

Водії аварійного транспорту

Оператори машин для технічного обслуговування

Рисунок А.3 – Структурно-організаційна блок-схема департаменту експлуатаційних служб

Аеронавігаційна служба

Диспетчерська служба

Служба обслуговування радіозв'язку

Льотна служба

Диспетчерські бригади

Спеціалісти з планування польотів

Інженери з радіозв’язку

Пілоти

Навігаційні бригади

Начальник департаменту повітряного руху

Льотні бригади

Стюардеси

Диспетчери навігаційного обслуговування

Диспетчери злітно-посадкової смуги

Метеорологи

Рисунок А.4 – Структурно-організаційна блок-схема департаменту повітряного руху

Стюардеси

Бригади обслуговування злітної смуги

Асфальтобетонники

Начальник департаменту технічного обслуговування

Служба авіатехнічного обслуговування

Служба обслуговування аеродрому

Служба ремонту спец-приладдя

Будівельники

Служба обслуговування автотранспорту

Бригади технічного обслуговування літаків

Електрики

Сантехніки

Бригади технічного огляду літаків

Бригади автомеханіків

Автомеханіки

Бригади з обслуговування інфраструктури

Механіки з обслуговування двигунів

Спеціалісти з структурного ремонту

Інженери-авіоніки

Інспектори з технічного стану

Інспектори з авіаційної безпеки

Механіки з обслуговування

Бригада обслуговування спецтехніки

Інженери з обслуговування обладнання аеропорту

Техніки з обслуговування системи безпеки

Рисунок А.5 – Структурно-організаційна блок-схема департаменту повітряного руху

Служба безпеки

Медико – санітарна служба

Митна служба

Інспектори

Лікарі

Бригади інспекторів митної служби

Бригади охоронців

Бригади медиків

Начальник департаменту спеціалізованих служб

Медсестри

Служба реєстрації

Агенти реєстрації білетів

Агенти реєстрації багажу

Бригади з реєстрації

Охоронці

Служба реєстрації

Бригади з реєстрації

Агенти реєстрації білетів

Агенти реєстрації багажу

Рисунок А.3 – Структурно-організаційна блок-схема департаменту спеціалізованих служб

Начальник департаменту адміністрації та додаткових служб

Адміністрація

Юридичний відділ

Канцелярія

Відділ кадрів

Керівники департаментів

Спеціалісти з підбору персоналу

Юридичні консультанти

Секретаріат

Юристи

Рисунок А.7 – Структурно-організаційна блок-схема департаменту адміністрації та додаткових служб