

Міністерство освіти і науки України  
Чернівецький національний університет  
імені Юрія Федьковича  
Інститут фізико-технічних та комп'ютерних наук  
(повна назва інституту/факультету)  
кафедра інформаційних технологій та комп'ютерної фізики  
(повна назва кафедри)

## **Автоматизована система бронювання квитків у кінотеатрі**

Дипломна робота  
Рівень вищої освіти - перший (бакалаврський)

Виконав:

студент 4 курсу, групи 417ск

спеціальності

126 – інформаційні системи та технології

(шифр і назва спеціальності)

Мартинюк А. Р.

(прізвище та ініціали)

Керівник: доктор фіз.-мат. наук, ст.наук.сп.,

Борча Мар'яна Драгошівна

(науковий ступінь, вчене звання, прізвище, ім'я, по-батькові)

Рецензент \_\_\_\_\_

(науковий ступінь, вчене звання, прізвище, ім'я, по-батькові)

До захисту допущено:

Протокол засідання кафедри № 19

від „ 17 ” червня 2021 р.

зав. кафедри \_\_\_\_\_ Борча М.Д.

Чернівці – 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Інститут фізико-технічних та комп'ютерних наук  
Кафедра інформаційних технологій та комп'ютерної фізики

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

докт. фіз.-мат. наук, доц.

\_\_\_\_\_ М. Д. Борча

“ \_\_\_\_ ” \_\_\_\_\_ 2021 р.

**АВТОМАТИЗОВАНА СИСТЕМА БРОНЮВАННЯ  
КВИТКІВ У КІНОТЕАТРІ**

ЛИСТ ЗАТВЕРДЖЕННЯ

**УЗГОДЖЕНО**

Керівник роботи

докт. фіз.-мат. наук, доцент

\_\_\_\_\_ М.Д. Борча

“ \_\_\_\_ ” \_\_\_\_\_ 2021 р.

Виконавець

студент 4-го курсу

\_\_\_\_\_ А. Р. Мартинюк

“ \_\_\_\_ ” \_\_\_\_\_ 2021 р.

## ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Мартинюку Андрію Романовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Автоматизована система бронювання квитків у кінотеатрі  
керівник роботи Борча Мар'яна Драгошівна, докт. фіз.-мат. наук, доцент,

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від “\_\_” \_\_\_\_ 202\_ року № \_\_\_\_

2. Строк подання студентом проекту (роботи) 27 травня 2021 р.

3. Вихідні дані до проекту (роботи) Мета роботи – розробка програмного засобу для автоматизації роботи касира у кінотеатрі, яке забезпечить ведення бази даних про фільми і сеанси, квитки і бронь на них, зали, персонал і користувачів, виконувати продаж квитків. Автоматизація перерахованих процесів виключить виникнення помилок при роботі з даними, позбавить від великих обсягів різної документації і прискорить роботу. Середовище розробки Visual Studio 2019, мова програмування – С#. 7.

База даних Microsoft SQL Server 2019.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) описати існуючі системи бронювання квитків у кінотеатрі

2) описати структуру та функції системи

3) створити програму для бронювання квитків у кінотеатрі

4) дослідити ефективність розробленої програми

5. Перелік графічного матеріалу

1) Структура таблиць бази даних

2) Приклад роботи програми

3) Діаграма прецедентів програми

Студент \_\_\_\_\_

(підпис)

Мартинюк А.Р.

(прізвище та ініціали)

Керівник проекту (роботи) \_\_\_\_\_

(підпис)

Борча М.Д.

(прізвище та ініціали)

## АНОТАЦІЯ

Кваліфікаційна робота виконана студентом групи 417 ск Мартинюком Андрієм Романовичем. Тема «Автоматизована система бронювання квитків у кінотеатрі». Робота направлена на здобуття ступеня бакалавр за спеціальністю 126 «Інформаційні системи та технології».

Метою кваліфікаційної роботи є створення та розробка інформаційної системи “ Автоматизована система бронювання квитків у кінотеатрі ”.

В результаті виконання кваліфікаційної роботи було створено програмний продукт бронювання квитків у кінотеатрі, який забезпечить потрібний для користувачів функціонал, зручний інтерфейс та швидкий доступ

Кількість сторінок – 34, таблиць – 7, рисунків – 9, джерел – 15.

Ключові слова: інформаційна система, бронювання, квитки в кінотеатр, база даних, критерії якості системи.

## ЗМІСТ

<b>ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ</b> .....	3
<b>ВСТУП</b> .....	6
<b>РОЗДІЛ 1</b> .....	8
<b>ЗАГАЛЬНІ ПОЛОЖЕННЯ</b> .....	8
1.1 Системний аналіз об'єкту дослідження та предметної області .....	8
1.1.1 Аналіз мети функціонування системи .....	8
1.1.2 Забезпечення якості .....	8
1.2 Постановка задачі .....	10
1.2.1 Вимоги до системи .....	10
1.2.2 Очікуванні ефекти від впровадження .....	12
<b>ВИСНОВКИ ДО РОЗДІЛУ</b> .....	12
<b>РОЗДІЛ 2</b> .....	13
<b>ПОБУДОВА СИСТЕМИ</b> .....	13
2.1 Моделювання системи .....	13
2.1.1 Вхідні дані .....	13
2.1.2 Вихідні дані .....	14
2.1.3 Функції та структура системи .....	14
2.2 Методи та засоби побудови системи .....	20
<b>ВИСНОВКИ ДО РОЗДІЛУ</b> .....	23
<b>РОЗДІЛ 3</b> .....	24
<b>ПРАКТИЧНА РЕАЛІЗАЦІЯ</b> .....	24
3.1 Засоби розробки .....	24
3.2 Опис реалізації системи .....	25
3.3 Аналіз отриманих результатів .....	28
3.3.1 Контрольний приклад .....	28
3.3.2 Аналіз критеріїв якості .....	31
<b>ВИСНОВКИ ДО РОЗДІЛУ</b> .....	32
<b>ЗАГАЛЬНІ ВИСНОВКИ</b> .....	33

## ВСТУП

На сьогоднішній день у світі активно реалізуються та впроваджуються інформаційні технології. Вже в багатьох сферах діяльності людини можна значну частину роботи довірити комп'ютеру, так як вони можуть зробити цю роботу не гірше людини, а в деяких випадках навіть краще. Персональні комп'ютери набагато швидше людини можуть виконати математичні обчислення, пошук інформації та рішення будь-яких інших завдань. Але без програмного забезпечення комп'ютери марні. Для кожної сфери діяльності людини необхідно своє програмне забезпечення. Сьогодні без цих помічників не обходиться жодна компанія, жодне підприємство.

Підприємства малого чи середнього бізнесу, незалежно від кількості їх працівників, є системами з складними взаємозв'язками між об'єктом управління і управлінською системою. Тому підвищення рівня управління є важливим фактором збільшення ефективності господарської діяльності підприємств. Досягти цього можна завдяки застосуванню інформаційних технологій, які допоможуть ефективно проаналізувати, спрогнозувати та прийняти управлінські рішення.

На ринку програмного забезпечення відзначається стійкий інтерес до програмних продуктів, здатних забезпечити ефективне управління різної області компаній і підприємств. Більшість керівників розуміють, що для успішної організації виробничого процесу необхідно застосовувати сучасні інформаційні технології. У зв'язку з глобальною інформатизацією у всіх сферах життєдіяльності людини, а також зі зростаючим рівнем потреби до програмних продуктів, здатних забезпечити ефективне управління виробничим процесом, надається можливість створення програмного засобу, яке вирішить поставлену задачу, застосовуючи сучасні інформаційні технології.

Саме тому завданням на дипломну роботу я обрав розробку програмного засобу для автоматизації роботи касира у кінотеатрі, яке повинно забезпечувати ведення бази даних про фільми і сеанси, квитки і бронь на них,

зали, персонал і користувачів, виконувати продаж квитків. Автоматизація перерахованих процесів виключить виникнення помилок при роботі з даними, позбавить від великих обсягів різної документації і прискорить роботу.

В результаті роботи над дипломним проектом буде розроблено програмний засіб і складена пояснювальна записка, яка включає в себе три розділи, що містять необхідну і достатню інформацію щодо використання даного програмного засобу та докладний його опис. У висновку буде проведений аналіз створеного програмного додатку, визначиться ступінь відповідності поставленого завдання і виконаної роботи.

## **РОЗДІЛ 1**

### **ЗАГАЛЬНІ ПОЛОЖЕННЯ**

#### **1.1 Системний аналіз об'єкту дослідження та предметної області**

Об'єктом дослідження розв'язуваної задачі є робота касира у кінотеатрі. Така робота пов'язана з накопиченням великої кількості інформації про сеанси та квитки, через що може бути проблематично виконати пошук інформації на традиційних паперових носіях які, зазвичай, використовуються у кінотеатрах. Тому для полегшення та прискорення роботи касира у кінотеатрі буде створене програмне забезпечення.

##### **1.1.1 Аналіз мети функціонування системи**

Сьогодні існує безліч аналогічних програм, призначені для складання і ведення баз даних. Однак розробка програми, спрямованої на вирішення завдання в певній сфері, не тільки полегшує працю касира, але і забезпечує правильність і безпеку зберігання і редагування даних, які обробляються цією програмою. Не дивлячись на те, що програма націлена на касира, її може використовувати і адміністратор кінотеатру, у якого буде більше функцій для роботи з програмою.

##### **1.1.2 Забезпечення якості**

У програмі буде організований інтерфейс, зрозумілий на інтуїтивному рівні як користувачу, що не володіє глибокими знаннями комп'ютерних технологій і навичками роботи з персональним комп'ютером, так і досвідченому користувачеві. Необхідно відзначити, що даний програмний продукт має засоби, що обмежують доступ до нього, а саме, при запуску програми необхідно буде ввести пароль.

Доцільним буде відзначити те, що даний програмний додаток не володіє засобами, що перешкоджають вільному поширенню програми. Для роботи з



додатком не потрібна установка додаткових компонентів і спеціальних технічних і програмних ресурсів.

## 1.2 Постановка задачі

Традиційно інформація зберігається на паперових носіях, при цьому важко здійснити швидкий пошук і виправлення помилково зазначених даних, а також продаж квитків займає у касира доволі багато часу. Таким чином, автоматизація роботи касира у кінотеатрі, є актуальною і перспективною.

### 1.2.1 Вимоги до системи

З усіх завдань, які вирішуватиме програмне забезпечення, можна виділити ряд основних:

- зберігати дані про сеанси, квитки, фільми, зали, персонал та користувачів;
- здійснювати продаж і бронювання квитків;
- виконувати пошук, сортування, фільтрацію даних.

Ця програма направлена на підвищення ефективності та швидкості роботи касира. Також вона повинна спростити роботу, пов'язану з пошуком, редагуванням і додаванням інформації про сеанси на фільми. Ці функції будуть доступні для касира кінотеатру.

В функції касира входять:

- робота з сеансами;
- продаж квитків на сеанси;
- бронювання квитків на сеанси.

Адміністратор може створити користувача «касир» в необмеженій кількості. Функції адміністратора:

- робота з сеансами;
- продаж квитків на сеанси;
- бронювання квитків на сеанси;
- робота з фільмами;
- робота з залами;
- робота з персоналом;

- робота з користувачами.

Існують загальні вимоги, які пред'являються до програмних Windows-додатків:

- відповідність стандартам організації інтерфейсу: використання багатовіконного підходу, реалізація управління роботою програмного додатка за допомогою елементів управління;
- вимоги до обраної колірної схеми робочого екрану;
- виконання однієї функції за допомогою різних елементів управління;
- надання довідкової інформації.

Відповідно до загальних вимог до графічного інтерфейсу програмний додаток має:

- містити звичні і зрозумілі користувачеві пункти меню, що відповідають повноваженням обробки;
- орієнтуватися на користувача, який спілкується з програмою на зовнішньому рівні взаємодії;
- задовольняти правилу «шести» в одну лінійку меню включати не більше шести понять, кожне з яких містить не більше шести опцій;
- зберігати стандартизоване призначення та місце розташування на екрані графічних об'єктів, що працюють в середовищі Windows.

Сучасні програмні додатки працюють в діалоговому режимі. В даному програмному засобі дані вимоги враховані.

В роботі програми передбачені деякі ситуації, які повинні попереджати користувача, щоб він виконував ті чи інші дії для продовження роботи програми. Для цього існують системні повідомлення, наприклад, якщо користувач почав додавати дані про фільм, але не заповнив усі обов'язкові поля, то в цьому випадку виводиться на екран повідомлення з попередженням про неможливість продовження роботи, так як не всі поля заповнені.

Розроблений додаток повинен мати зрозумілий інтерфейс, досить простий в застосуванні, і в той же час дозволяти зменшити витрати часу і матеріальних ресурсів, використовуваних при автоматизації роботи касира кінотеатру. Програма повинна мати невеликий розмір, бути оптимізована для швидкого запуску і мінімального завантаження системи. Доступний дружній інтерфейс забезпечує наочність простоти в управлінні програми і не містить складних налаштувань.

### **1.2.2 Очікуванні ефекти від впровадження**

На жаль, на сьогоднішній день усі подібні програми платні або недоступні для вільного доступу, тому порівняння цього додатку з аналогами не вийде.

Програмне забезпечення, яке розробляється під час дипломного проєктування, націлене на два типи користувачів: касир, адміністратор. Завдяки зрозумілому інтерфейсу, під час користування програмою, у касира не виникне труднощів у додаванні сеансів, бронюванні та продажі квитків на фільм.

Після початку використання даного додатку вдосконалиться:

- процес бронювання квитків;
- процес продажу квитків;
- процес пошуку інформації загалом;
- швидкість роботи касира;
- якість роботи адміністратора;
- зручність купівлі та бронюванні квитків клієнтами даного закладу.

## **ВИСНОВКИ ДО РОЗДІЛУ**

В даному розділі було виконано аналіз предметної області та її опис, поставлення задачі до виконання дипломного проєкту, розкрито питання щодо актуальності та перспективності розробки програмного забезпечення.

## РОЗДІЛ 2

### ПОБУДОВА СИСТЕМИ

#### 2.1 Моделювання системи

##### 2.1.1 Вхідні дані

До вхідних даних можна адресувати дані, що заносяться в базу даних при додаванні або редагуванні записів про сеанси, квитки, зали, фільми, користувачів або про персонал.

Структура вхідної інформації повинна відповідати структурі даних, визначеній на етапі проектування бази даних, і відповідати типам даних, необхідних у визначенні фільтра і ключа пошуку. На рисунку 2.1 зображена схема бази даних.

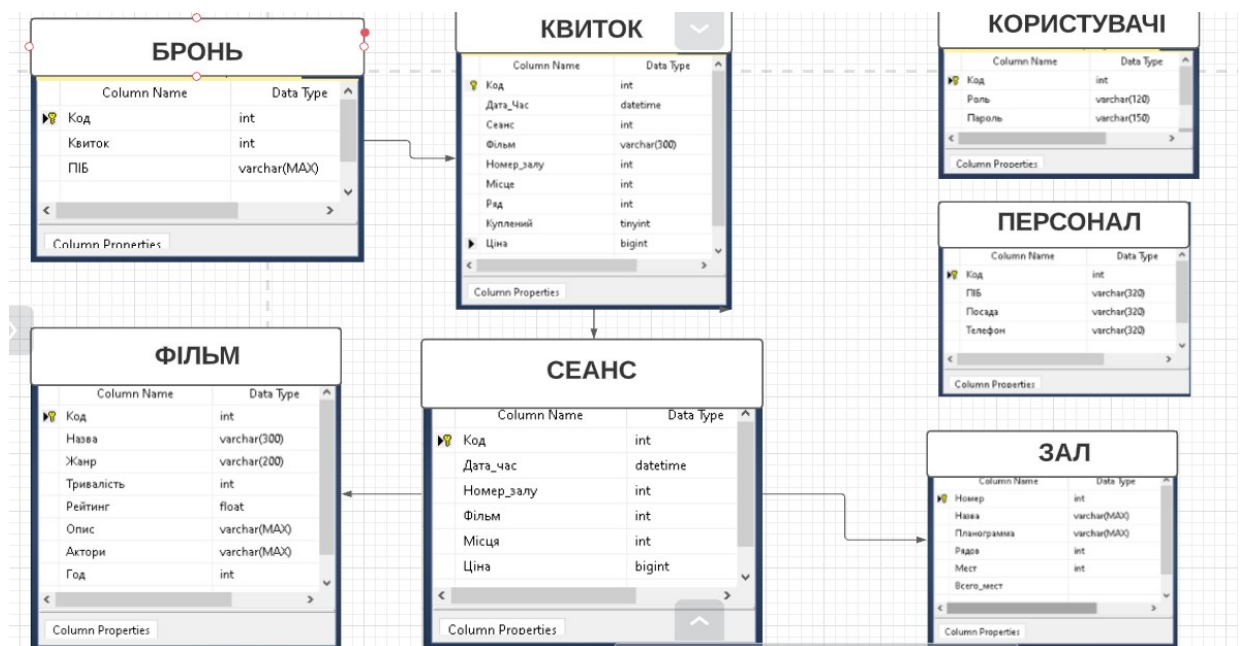


Рисунок 2.1 – Схема бази даних.

Всі ці дані в кодї програми перетворюються і записуються в базу даних. Перевірка коректності даних, що вводяться здійснюється в програмі, тому користувачеві не доведеться турбуватися про неправильному введенні.

Також до вхідної інформації можна віднести інформацію, що задається користувачем при фільтрації і пошуку даних.

### 2.1.2 Вихідні дані

До вихідних даних можна віднести інформацію, яка переглядається користувачем на екрані на будь-якій формі програмного засобу. Наприклад, назви фільмів, доступні сеанси, заброньовані квитки і тому подібне.

Організація даних має на увазі створення моделі даних, головними елементами якої є сутності та їх зв'язки. Реляційна модель заснована на математичному понятті відносин, поданням якого є таблиця. У реляційній моделі відносини використовуються для зберігання інформації про об'єкти, представлених в базі даних. Відношення має вигляд двомірної таблиці, в якій рядки відповідають записам, а стовпці - атрибутам. Кожен запис має однозначно характеризуватися в таблиці. Для цього використовують первинні і вторинні ключі. Перевагою реляційної моделі є простота і зручність фізичної реалізації.

### 2.1.3 Функції та структура системи

На даному етапі були виділені і реалізовані функції, які дозволяють вирішити завдання, що забезпечують досягнення мети курсового проекту.

Функція продажу квитка на сеанс здійснюється в момент натискання на кнопку «Продати квиток» форми з сеансами і викликається методом «AddBilet ()». При виконанні даної функції будуть формуватися запити, додаватися в базу інформація про новий квиток, буде зменшуватися кількість вільних місць на сеанс. Так само ця функція служить для бронювання квитків. Код даної функції представлений нижче.

```
private void addBilet(bool bronirovanie)
{
    if (квитокDataGridView.RowCount > 0)
    {
        for (int i = 0; i < квитокDataGridView.RowCount; i++)
        {
```

```

    if
    (Convert.ToInt32(квитокDataGridView.Rows[i].Cells["dataGridViewTextBoxColumn15"].Value) == Convert.ToInt32(numericUpDown2.Value) &&
        Convert.ToInt32(билетDataGridView.Rows[i].Cells["dataGridViewTextBoxColumn12"].Value) == Convert.ToInt32(numericUpDown1.Value))
    {
        MessageBox.Show("Це місце вже зайняте");
        return;
    }
}
byte kuplen = 1;
if (bronirovanie)
{
    kuplen = 0;
}
квитокTableAdapter.Insert(Convert.ToDateTime(сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn2"].Value.ToString()),
    Convert.ToInt32(сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn1"].Value.ToString()),
    Convert.ToString((сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn4"] as DataGridViewComboBoxCell).FormattedValue.ToString()),
    Convert.ToInt32(сеансDataGridView.CurrentRow.Cells["Номер_залу"].Value.ToString()),
    Convert.ToInt32(numericUpDown1.Value),
    kuplen,
    Convert.ToInt32(сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn6"].Value.ToString()), Convert.ToInt32(numericUpDown2.Value));
    string new_id = "";
    string sql = "SELECT top 1 * from Квиток order by Код DESC";
    using (SqlConnection c = new SqlConnection(Connection))
    {

```

```

c.Open();
using (SqlDataAdapter a = new SqlDataAdapter(sql, c))
{
    DataTable t = new DataTable();
    a.Fill(t);
    new_id = t.Rows[0]["Код"].ToString();
}
}
try
{
    using (SqlConnection con = new SqlConnection(Connection))
    {
        con.Open();
        using (SqlCommand command = new SqlCommand("Update Сеанс set
Місяць = Місяць-1 where Код=" +
сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn1"].Value.
ToString(), con))
        {
            command.ExecuteNonQuery();
        }
        this.сеансTableAdapter.Fill(this.kinoDataSet.Сеанс);
        con.Close();
    }
}
catch (SystemException ex)
{
    MessageBox.Show(string.Format("An error occurred: {0}", ex.Message));
}

if (bronirovanie)

```



```

{
броньTableAdapter.Insert(Convert.ToInt32(new_id), textBox1.Text);
MessageBox.Show("Квиток заброньований");
}
else {
MessageBox.Show("Квиток заброньований");
printBilet(Convert.ToInt32(new_id));
}

this.квитокTableAdapter.Fill(this.kinoDataSet.Квиток);
}

```

Функція пошуку по таблиці сеансів обробляється після натискання кнопки «Шукати» після вибору критеріїв пошуку і введення слів для пошуку. Ця функція викликається методом «button4\_Click», закріпленим за подією «Click». Текст даної функції представлений нижче.

```

private void button4_Click(object sender, EventArgs e)
{
string filter = "Код = Код ";
bool error = false;
if (checkBox1.Checked || checkBox2.Checked || checkBox3.Checked)
{
if (checkBox1.Checked)
{
filter += String.Format("and Дата_час >= '{0:yyyy-MM-dd}' AND Дата_час
< '{1:yyyy-MM-dd}'",          dateTimePicker1.Value.ToShortDateString(),
dateTimePicker2.Value.AddDays(1).ToShortDateString());
}
if (checkBox3.Checked)
{
filter += " and Номер_залу = " + comboBox2.SelectedValue + " ";
}
}
}

```

```

}
if (checkBox2.Checked)
{

filter += " and Фільм = " + comboBox3.SelectedValue + " ";
}

if (error)
{
MessageBox.Show("Заповніть поле пошуку для вибраного критерію!");
return;
}
else
{
this.сеансBindingSource.Filter = filter; //фільтрація
}

}
else
MessageBox.Show("Виберіть хоча б один критерій пошуку!");
}

```

Функція повернення квитка на сеанс здійснюється в момент натискання на кнопку «Повернути квиток» на формі з квитками. Ця функція викликається методом «button2\_Click», закріпленим за подією «Click». Текст даної функції представлений нижче.

```

private void button2_Click(object sender, EventArgs e)
{
int id =
Convert.ToInt32(билетDataGridView.CurrentRow.Cells["dataGridViewTextBox
Column1"].Value);

```

```

try
{
using (SqlConnection con = new SqlConnection(Connection))
{
con.Open();
using (SqlCommand command = new SqlCommand("Delete from Квиток
where Код=" + id, con))
{
command.ExecuteNonQuery();
MessageBox.Show("Квиток повернуто");
this.квитокTableAdapter.Fill(this.kinoDataSet.Квиток);
}
con.Close();
}
}
catch (SystemException ex)
{
MessageBox.Show(string.Format("An error occurred: {0}", ex.Message));
}
}

```

Функція друку квитка на сеанс здійснюється в момент натискання на кнопку «Друк» на формі з печаткою квитка після його продажу. Ця функція викликається методом «button2\_Click», закріпленим за подією «Click». Текст даної функції представлений нижче.

```

private void CaptureScreen()
{
Graphics myGraphics = this.CreateGraphics();
Size s = this.Size;
memoryImage = new Bitmap(s.Width, s.Height-20, myGraphics);

```

```

Graphics memoryGraphics = Graphics.FromImage(memoryImage);
memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, s);
}

```

## 2.2 Методи та засоби побудови системи

Для реалізації дипломного проекту було обране інтегроване програмне середовище Microsoft Visual Studio, а мовою програмування С#. Виходячи з дослідження предметної області завдання прийнято рішення про організацію програмного засобу у вигляді бази даних MS SQL SERVER. У програмі передбачається створення навігаційного меню, що складається з пунктів «Сеанси», «Квитки», «Фільми», «Зали», «Персонал», «Заброньовані квитки», «Користувачі».

Структура бази даних розроблюваного програмного засобу включає сім таблиць. Їх структура, і короткий опис наводиться в таблицях 2.1-2.7.

Таблиця «Квиток» призначена для зберігання інформації про квитки.

Таблиця 2.1 – Структура таблиці «Квиток»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Код	int	Довге ціле	Код
Дата_час	datetime	64	Дата та час покупки
Сеанс	int	Довге ціле	Код сеансу
Фільм	varchar	300	Назва фільму
Номер_залу	int	Довге ціле	Номер залу
Місце	int	Довге ціле	Місце
Ряд	int	Довге ціле	Ряд
Куплений	tinyint	1	Флаг купленого квитку
Ціна	bigint	Довге ціле	Ціна квитку

Таблиця «Бронь» призначена для зберігання інформації про заброньовані квитки.

Таблиця 2.2 – Структура таблиці «Бронь»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Код	int	Довге ціле	Код
Квиток	int	Довге ціле	Код квитка
ПІБ	varchar	65535	ПІБ бронюючого

Таблиця «Персонал» призначена для зберігання інформації про персонал.

Таблиця 2.3 – Структура таблиці «Персонал»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Код	int	Довге ціле	Код
ПІБ	varchar	320	ПІБ робітника
Посада	varchar	320	Посада
Телефон	varchar	320	Телефон

Таблиця «Зал» призначена для зберігання інформації про зали.

Таблиця 2.4 – Структура таблиці «Зал»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Номер	int	Довге ціле	Код залу
Назва	varchar	65535	Назва залу
Планограма	varchar	65535	Планограма
Місце	int	Довге ціле	Кількість вільних місць

Рядів	int	Довге ціле	Рядів
Всього_місць	Вираховуюче	-	Всього місць

Таблиця «Фільм» призначена для зберігання інформації про фільми.

Таблиця 2.5 – Структура таблиці «Фільм»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Код	int	Довге ціле	Код
Назва	varchar	300	Назва фільму
Жанр	varchar	200	Жанр
Тривалість	int	Довге ціле	Тривалість фільму
Рейтинг	float	Довге з плаваючою точкою	Рейтинг фільму
Опис	varchar	65535	Опис фільму
Актори	varchar	65535	Актори
Рік	int	Довге ціле	Рік виходу

Таблиця «Користувач» призначена для зберігання інформації про користувачів.

Таблиця 2.6 – Структура таблиці «Користувач»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Код	int	Довге ціле	Код
Роль	varchar	120	Роль
Пароль	varchar	150	Пароль

Таблиця «Сеанс» призначена для зберігання інформації про сеансах.

Таблиця 2.7 – Структура таблиці «Сеанс»

Ім'я поля	Тип поля	Розмір поля	Опис поля
Код	int	Довге ціле	Код
Дата_час	datetime	64	Дата і час сеансу
Номер_зала	int	Довге ціле	Номер залу
Фільм	int	Довге ціле	Код фільма
Места	int	Довге ціле	Вільні місця
Цена	bigint	Довге ціле	Ціна квитка на сеанс

### ВИСНОВКИ ДО РОЗДІЛУ

В даному розділі було зображено схему бази даних, подано структури таблиць бази даних, за допомогою яких працює інформаційна система. Також роз'яснено теоретичну частину про вхідні та вихідні дані, і наведено код основних функцій програмного забезпечення.

## РОЗДІЛ 3

### ПРАКТИЧНА РЕАЛІЗАЦІЯ

#### 3.1 Засоби розробки

Для створення дипломного проекту обране інтегроване середовище розробки Microsoft Visual Studio. В цілому середовище має багато переваг:

- забезпечення інтуїтивно зрозумілим, що розширюється, уніфікованим середовищем для мови, конструкторів і інструментальних засобів;
- надання розробникам набору модернізованих взаємодіючих елементів, відповідних наявними у розробників навичкам;
- надання високопродуктивних інструментальних засобів для всіх етапів життєвого циклу розробки - від визначення вимог і планування до подальшого супроводу продукту.

В якості мови програмування був обраний C #.

Переваги мови програмування C #:

- Справжня об'єктна орієнтованість (будь-яка мовна сутність претендує на те, щоб бути об'єктом);
- компонентно-орієнтоване програмування;
- безпечний (в порівнянні з мовами C і C++) код;
- уніфікована система типізації;
- підтримка подієво-орієнтованого програмування;
- «рідна» мова для створення додатків в середовищі .NET;
- об'єднання кращих ідей сучасних мов програмування: Java, C++, Visual Basic та ін.

C # це мова програмування, що поєднує об'єктно-орієнтовані і аспектно-орієнтовані концепції. Розроблено в 1998-2001 роках групою інженерів під керівництвом Андерса Хейлсберг в компанії Microsoft як основна мова розробки додатків для платформи Microsoft .NET. Компілятор з C # входить в стандартну установку самої .NET, тому програми на ньому можна створювати і компілювати навіть без інструментальних засобів, на кшталт Visual Studio.



C # відноситься до сім'ї мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C ++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато від своїх попередників - мов C ++, Java, Delphi, Модула і Smalltalk - C #, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем: так, C # не підтримує множинне успадкування класів ( на відміну від C ++).

C # розроблявся як мова програмування прикладного рівня для CLR і, як такий, залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C #, яка відображає FCL. Присутність або відсутність тих чи інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована в відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C #; подібної взаємодії слід очікувати і в подальшому. CLR надає C #, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, прибирання сміття не реалізована в самому C #, а проводиться CLR для програм, написаних на C # точно так же, як це робиться для програм на VB.NET, J #.

База даних була створена під системою управління базами даних Microsoft SQL Server.

### **3.2 Опис реалізації системи**

Програма розроблена для середньостатистичного користувача, який не має навичок у сфері користування комп'ютером. Для цього було створено навігаційне меню, яке має такі пункти: «Сеанси», «Квитки», «Фільми», «Зали», «Персонал», «Звіти(тимчасово не працює)», «Заброньовані квитки», «Користувачі» і «Довідка».

Пункт меню «Сеанси» буде викликати форму з сеансами, де користувач зможе додати, редагувати сеанси, здійснити пошук і сортування, а так само продати квиток на сеанс.

Пункт меню «Квитки» буде викликати форму з квитками, де користувач зможе здійснити перегляд і редагувати квитки, а також здійснити повернення.

Пункт меню «Фільми» буде викликати форму з фільмами, де користувач зможе здійснити додавання і редагування фільму, а також здійснити пошук і сортування.

Пункт меню «Зали» буде викликати форму з залами, де користувач зможе здійснити редагування залів.

Пункт меню «Персонал» буде викликати форму з персоналом, де користувач зможе здійснити перегляд, додати і редагувати працівників, а також здійснити пошук і сортування.

Пункт меню «Звіти» буде викликати форму з меню на якому знаходяться підпункти з різними видами звітів.

Пункт меню «Заброньовані квитки» буде викликати форму з заброньованими квитками, де користувач зможе здійснити перегляд, а так само видалення квитка з бронню і продаж заброньованого квитка.

Пункт меню «Користувачі» буде викликати форму з користувачами програми, де користувач зможе здійснити перегляд, додати а так само змінити паролі у користувачів.

Пункт меню «Довідка» буде викликати довідку по користуванню програмою.

На формах будуть розташовуватися різні елементи управління, призначені для певних завдань. Наприклад, стандартний елемент керування Select, призначений для вибору потрібних даних зі списку програмного додатка і буде здійснювати вибір користувачів. За компонентом Button будуть закріплені різні дії для виконання певних функцій, таких як: видалення, додавання і редагування даних. Стандартний елемент управління Textbox буде

використаний для введення і виведення різноманітних даних і для відображення тексту на сторінках.

Проект складається з двадцяти шести форм, кожна з яких це діалогове вікно, у якому користувач може виконати ту чи іншу дію. На рисунку 3.1 зображено організацію архітектури проекту.

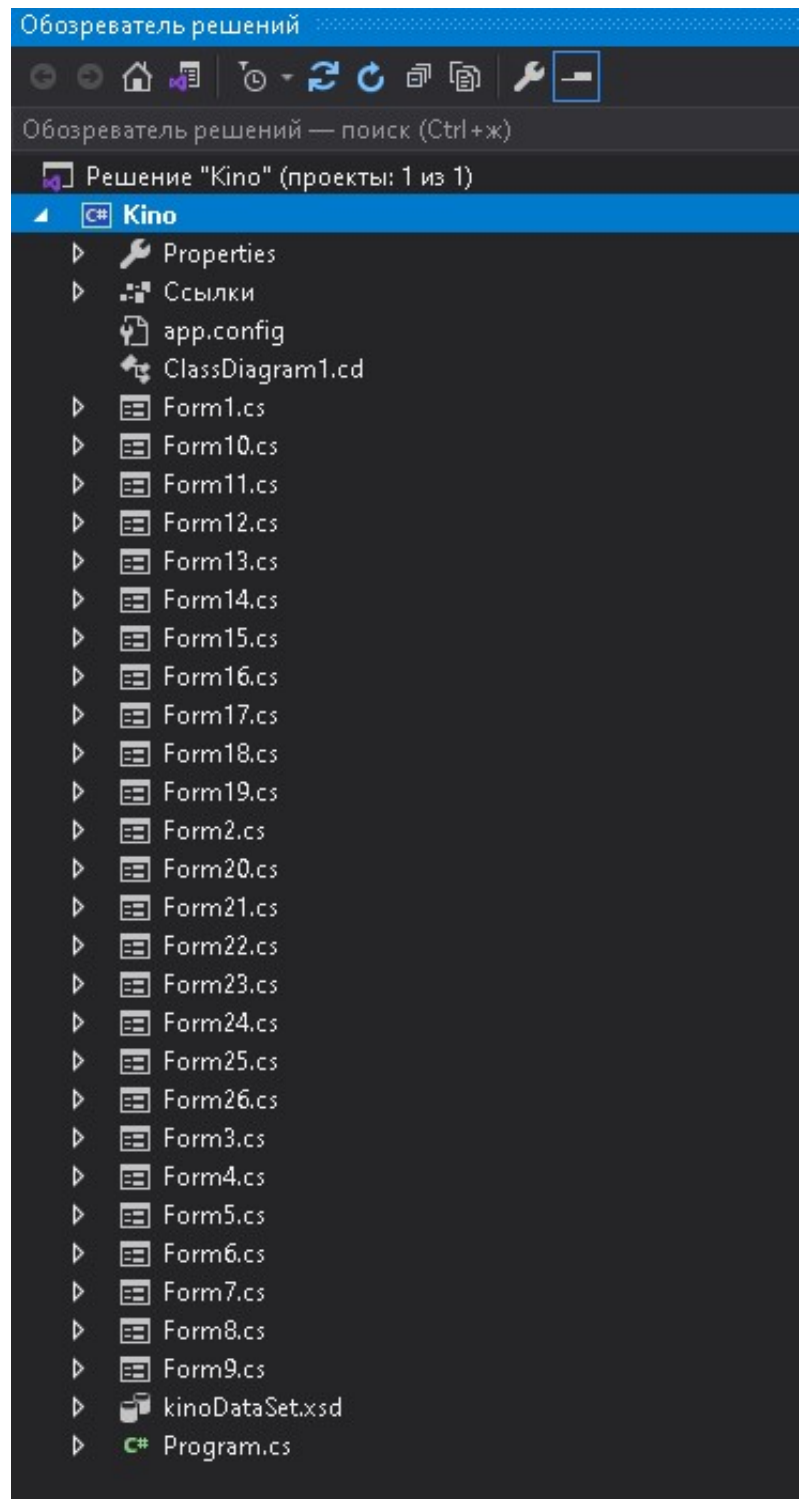


Рисунок 3.1 – Структура архітектури проекту

### 3.3 Аналіз отриманих результатів

#### 3.3.1 Контрольний приклад

Після запуску програми відкриється вікно авторизації, де потрібно вибрати користувача, ввести пароль та натиснути кнопку «Увійти». Вікно входу у систему зображено на рисунку 3.2.

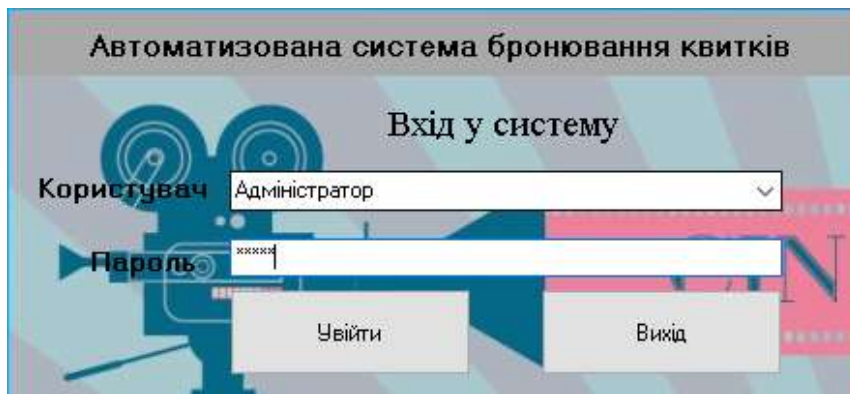


Рисунок 3.2 – Вікно авторизації.

При неправильному введенні пароля користувачу буде виведено відповідне повідомлення на екран. Після вибору адміністратора і введення правильного паролю, відкриється форма з меню. Зовнішній вигляд форми зображено на рисунку 3.3.



Рисунок 3.3 – Меню адміністратора

Для того, щоб подивитися сеанси, необхідно натиснути на кнопку «Сеанси». Відкриється форма, вид якої представлений на рисунку 3.4.

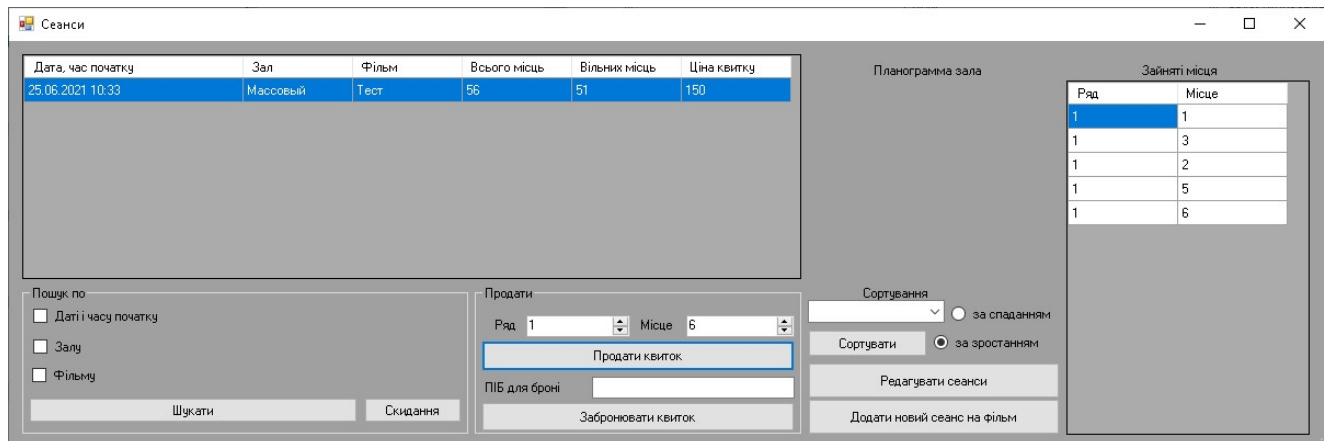


Рисунок 3.4 – Робота з сеансами

Для того, щоб здійснити пошук необхідно вибрати критерії і ввести слова для пошуку, після чого натиснути на кнопку «Шукати». Скинути пошук можна натиснувши на кнопку «Скидання».

Для сортування необхідно вибрати критерій сортування, вибрати тип сортування і натиснути на кнопку «Сортувати».

Для продажу квитка на сеанс необхідно ввести ряд і місце, після чого натиснути «Продати квиток». У разі, якщо це місце зайняте, користувачеві буде виведено відповідне повідомлення. У разі, якщо помилок немає, користувачеві буде показаний квиток, вид якого представлений на малюнку 3.5.

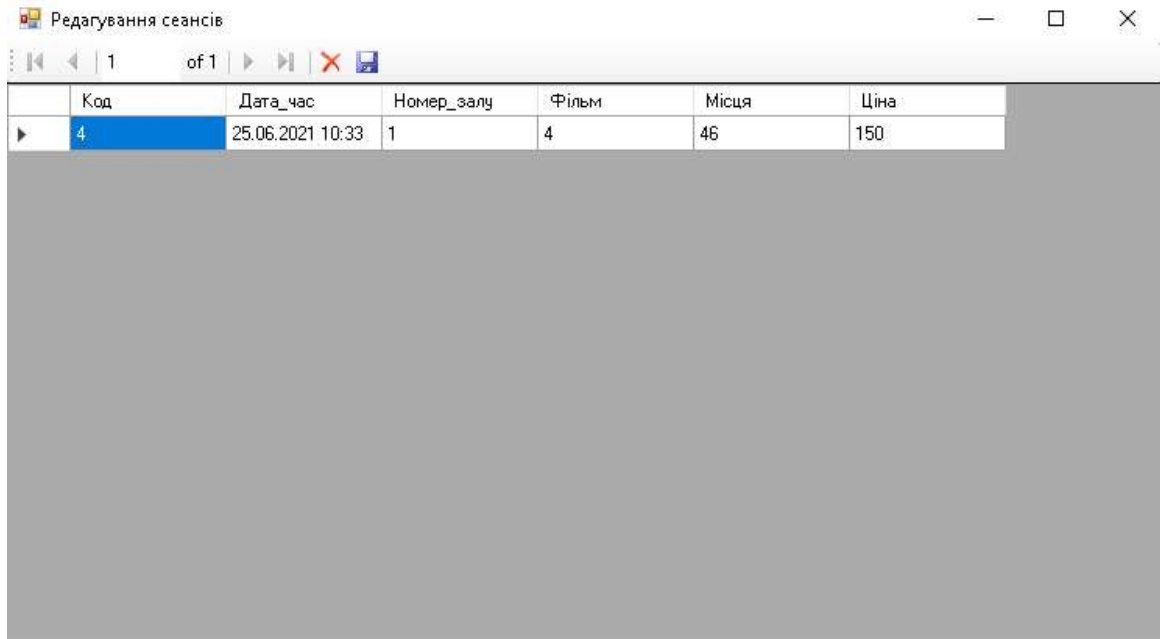


Рисунок 3.5 – Проданий квиток

Для друку квитка необхідно натиснути на відповідну кнопку на формі.

Для бронювання квитка на сеанс необхідно ввести ряд, місце і ПІБ для броні, після чого натиснути на кнопку «Забронювати квиток».

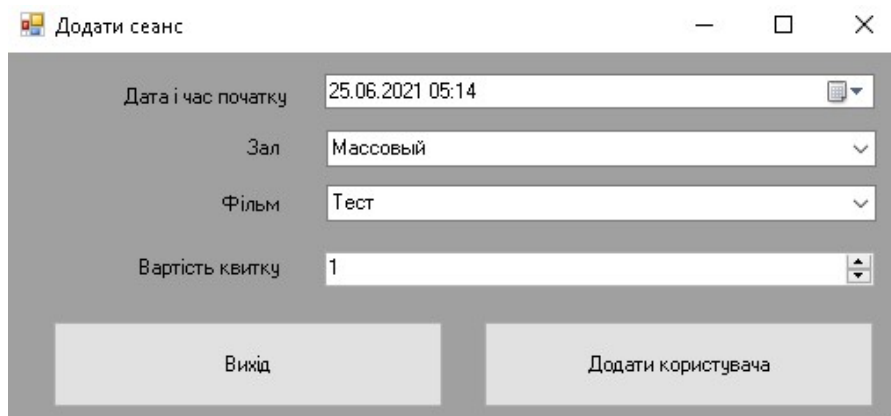
Для редагування сеансів необхідно натиснути на кнопку «Редагування сеансів». Буде відкрита форма, на якій користувач зможе змінити інформацію про сеансах. Вид форми представлений на малюнку 3.6.



Код	Дата_час	Номер_залу	Фільм	Місця	Ціна
4	25.06.2021 10:33	1	4	46	150

Рисунок 3.6 – Форма редагування сеансів

Для додавання сеансів необхідно натиснути на кнопку «Додати новий сеанс на фільм». Буде відкрита форма, на якій користувач зможе внести інформацію про сеанс, після чого необхідно натиснути на кнопку «Додати». Вид форми представлений на малюнку 3.7.

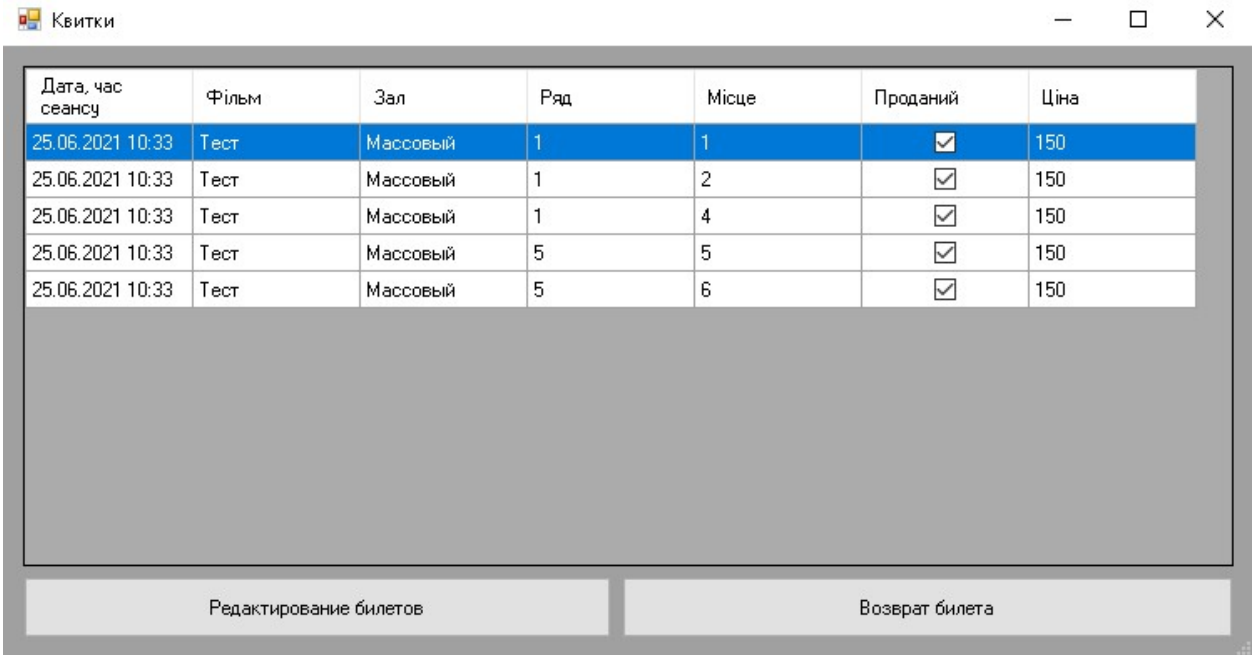


Дата і час початку	25.06.2021 05:14
Зал	Массовый
Фільм	Тест
Вартість квитку	1

Вихід      Додати користувача

Рисунок 3.7 – Форма додавання сеансу на фільм

Для того, що б подивитися квитки необхідно вибрати на головному меню пункт «Квитки». Відкриється форма, вид якої представлений на малюнку 3.8.



Дата, час сеансу	Фільм	Зал	Ряд	Місце	Проданий	Ціна
25.06.2021 10:33	Тест	Массовый	1	1	<input checked="" type="checkbox"/>	150
25.06.2021 10:33	Тест	Массовый	1	2	<input checked="" type="checkbox"/>	150
25.06.2021 10:33	Тест	Массовый	1	4	<input checked="" type="checkbox"/>	150
25.06.2021 10:33	Тест	Массовый	5	5	<input checked="" type="checkbox"/>	150
25.06.2021 10:33	Тест	Массовый	5	6	<input checked="" type="checkbox"/>	150

Редактирование билетов      Возврат билета

Рисунок 3.8 – Продані квитки

### 3.3.2 Аналіз критеріїв якості

Для застосування даного програмного забезпечення необхідні такі технічні вимоги:

- процесор Pentium 800 або вище;
- мінімальний обсяг оперативної пам'яті 128 Мб;
- операційна система Windows XP і вище;
- пакет оновлень Framework v4.0;
- рекомендується монітор типу VGA або з кращою роздільною здатністю;
- клавіатура;
- миша.

Комп'ютер повинен працювати під управлінням операційної системи, починаючи з Windows XP і вище. Найбільш зручною операційною системою для проведення випробувань є Windows 7, так як вона орієнтована на

максимальне використання всіх можливостей ПК, мережевих ресурсів і забезпечення комфортних умов роботи.

Програма має невеликий розмір, оптимізована для швидкого запуску і мінімального завантаження системи. Доступний дружній інтерфейс забезпечує наочність простоти в управлінні програми і не містить складних налаштувань.

Для початку роботи з програмним засобом необхідно запустити файл «Kino.exe», розмір якого становить 1 002КБ. Для функціонування програмного засобу необхідно встановити платформу .NET Framework 4.0. Програмний засіб не потребує встановлення.

### **ВИСНОВКИ ДО РОЗДІЛУ**

У цьому розділі були наведені мінімальні вимоги для запуску розробленого додатку, виконався його тест на належне функціонування. Провівся огляд пунктів меню, які були детально описані.



## ЗАГАЛЬНІ ВИСНОВКИ

В рамках дипломного проектування було розроблено «Автоматизовану систему бронювання квитків у кінотеатрі».

Програмний засіб має ряд переваг: простий і зрозумілий інтерфейс, невеликий обсяг пам'яті, займаний додатком на різних носіях інформації, зручний для швидкої обробки інформації.

У програмі реалізовані такі завдання, як: редагування даних, фільтрація даних, формування звітів, продаж і бронювання квитків на сеанси.

Програма реалізована в повному обсязі і відповідно до заданих вимог. Повністю налагоджена і протестована. Поставлені завдання виконані.

У програмному засобі, завдяки середовищі розробки Microsoft Visual Studio 2019 був розроблений зручний, зрозумілий і простий у використанні інтерфейс.

Програма реалізована повністю відповідно до поставленим завданням дипломного проекту. Додаток було протестовано і налагоджено.

Проект був реалізований за допомогою середовища розробки Microsoft Visual Studio 2019 з використанням мови C # і системи управління базами даних Microsoft SQL Server.

Програмний засіб готовий до практичного використання. Цей додаток може бути доповнено й модернізовано.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рихтер Джеффри. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. 4-е изд. / Джеффри Рихтер. - Питер, 2013. - 896 с.
2. Петцольд Ч. Программирование для MicrosoftWindows на C# – М.: Издательско-торговый дом «Русская редакция», 2002.
3. Троелсен Эндрю. Язык программирования C# 2010 и платформа.NET 4 / Эндрю Троелсен. - Вильямс, 2010. - 1392 с.
4. Microsoft Visual Studio // Материал з Вікіпедії - вільної енциклопедії. [Електронний ресурс] - Режим доступу: [http://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://ru.wikipedia.org/wiki/Microsoft_Visual_Studio).
5. Мова програмування C# [Електронний ресурс]- Режим доступу: [https://uk.wikipedia.org/wiki/C\\_Sharp](https://uk.wikipedia.org/wiki/C_Sharp).
6. Бьюли А. Изучаем SQL / Алан Бьюли. - Символ-Плюс, 2007. - 312 с.
7. Адам Фримен - ASP.NET Core MVC с примерами на C# для профессионалов. 2017р.
8. Карпов Р.Н. Microsoft Office 2007 – Мн.: Бином, 2007
9. Офіційний сайт TheOldReader // TheOldReader. [Електронний ресурс]- Режим доступу: <https://theoldreader.com/pages/apps>
10. Приёмы объектно-ориентированного проектирования. Паттерны проектирования / Э.Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – Харьков: Питер, 2001. – 368 с
11. Сайт SQLite // SQLite Home Page [Електронний ресурс]- Режим доступу: <http://sqlite.org/>
12. Бураков П.В. Введение в системы баз данных. Учебное пособие / П.В. Бураков., В.Ю. Петров. - СПбГУ ИТМО, 2010. 129 с.
13. Стиллмен Э. Изучаем C#. Включая C#.NET 4.0 и Visual Studio 2010. 2-е издание / Э. Стиллмен., Дж. Грин - O'Reilly, 2012. - 689 с.

14. Скiт Д. C# in Depth / Джон Скiт., 2013. – 582 с
15. Галерея додаткiв VisualStudio // NuGet [Електронний ресурс]-  
Режим доступу: <https://www.nuget.org/packages/>