Лабораторная работа №2

Тема:Проектирование и макетирование программного продукта

Цель: Научиться проектировать простейшие системы и составлять документацию по проектированию программного продукта.

1. Архитектура системы

Тип архитектуры: Клиент-серверная, трёхуровневая

Уровни:

Представление (Windows Forms/WPF)

Бизнес-логика (C# классы)

Данные (PostgreSQL)

Основные модули:

TicketService — поиск, бронь, покупка, возврат

ScheduleManager — управление расписанием

ReportGenerator — формирование отчётов

UserAuth — аутентификация по ролям (пассажир, кассир, админ)

2. Макет интерфейса

Главное окно пассажира:

Поля: «Откуда», «Куда», «Дата»

Кнопка: «Найти»

Таблица: список рейсов с типами вагонов и местами

Кнопки: «Купить», «Забронировать», «Вернуть билет»

Окно кассира:

Те же элементы, но с доступом к возврату и печати билетов

Окно администратора:

Вкладки: «Расписание», «Отчёты», «Пользователи»

Кнопки: «Добавить рейс», «Удалить рейс», «Сформировать отчёт»

3. Фрагменты SDD-документа

3.1 Структура проекта

```
RailwayTicketSystem/
├── GUI/
│   ├── MainForm.cs
│   ├── AdminForm.cs
├── BusinessLogic/
│   ├── TicketService.cs
│   ├── ScheduleManager.cs
│   ├── ReportGenerator.cs
├── DataAccess/
│   ├── DBContext.cs
├── Models/
│   ├── Train.cs
│   ├── Ticket.cs
│   ├── User.cs
```

3.2 Пример класса TicketService

```
public class TicketService
{
   public bool BuyTicket(int trainId, int seatNumber, string passengerName)
   {
      if (!IsSeatAvailable(trainId, seatNumber)) return false;
      // Логика покупки
      DBContext.SaveTicket(new Ticket
      {
         TrainId = trainId,
         SeatNumber = seatNumber,
         PassengerName = passengerName,
         PurchaseDate = DateTime.Now
      });
      return true;
   }
```

```
    public bool IsSeatAvailable(int trainId, int seatNumber)
    {
        return !DBContext.Tickets.Any(t => t.TrainId == trainId && t.SeatNumber == seatNumber);
    }
}
```

4. Программная реализация (фрагмент)

Модель поезда

```
public class Train
{
    public int Id { get; set; }
    public string Number { get; set; }
    public string Route { get; set; }
    public DateTime DepartureTime { get; set; }
    public DateTime ArrivalTime { get; set; }
    public int TotalSeats { get; set; }
    public int FreeSeats { get; set; }
}
```

Форма поиска рейсов

```
private void btnSearch_Click(object sender, EventArgs e)
{
    var trains = ScheduleManager.SearchTrains(txtFrom.Text, txtTo.Text, dtpDate.Value);
    dataGridView1.DataSource = trains;
}
```

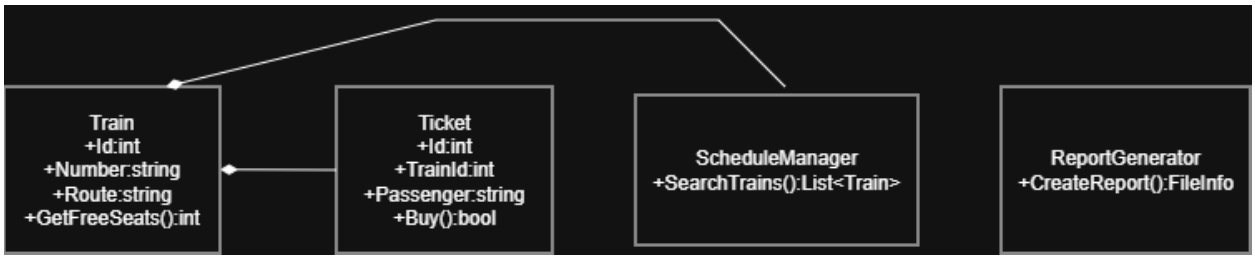
5 Детальная архитектура

5.1 Слой представления: MainForm, AdminForm (WinForms).

5.2 Слой бизнес-логики: TicketService, ScheduleManager, ReportGenerator.

5.3 Слой данных: PostgreSQL, EF Core, репозитории.

6 Диаграмма классов



7 Диаграмма последовательности «Покупка билета»

8 Структура БД