**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

УТВЕРЖДАЮ

Зам. директора по УПР

С.Ю. Назаров\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_20\_\_\_г.

**ОТЧЕТ**

**по учебной практике УП.02.01**

По профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) 4 курса 42919/4 группы

Алексеева Полина Александровна

(Фамилия, имя, отчество)

Место прохождения практики: ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса д. 23

(наименование и адрес организации)

Период прохождения практики

с «9» декабря 2024 г. по «21» декабря 2024 г.

Руководитель практики Прокофьев А.А.

(подпись) (расшифровка подписи)

Итоговая оценка по практике\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на учебную практику** **УП.02.01**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/4 группы

Алексеева Полина Александровна

(Фамилия, имя, отчество)

Место прохождения практики: ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, Энгельса 23

(наименование и адрес организации)

Период прохождения практики

с «9» декабря 2024 г. по «21» декабря 2024 г.

**Виды работ, обязательные для выполнения**

1. Участие в выработке требований к программному обеспечению
2. Стадии проектирования программного обеспечения
3. Разработка модулей программного обеспечения
4. Тестирование программных модулей и их интеграции
5. Разработка программной документации. Стандарты кодирования

**Индивидуальное задание: ВАРИАНТ 7**

Задание выдал «9» декабря 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Прокофьев А.А.

(подпись)  (Ф.И.О.)

Задание получил «9» декабря 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_ Алексеева П.А.(подпись)  (Ф.И.О.)

СОГЛАСОВАНО

Председатель ПЦК

\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения учебной практики УП 02.01**

По профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/4 группы

Алексеева Полина Александровна

(Фамилия, имя, отчество)

Место прохождения практики ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса д. 23

(наименование и адрес организации)

Период прохождения практики

с «9» декабря 2024 г. по «21» декабря 2024 г.

Руководитель практики Прокофьев А.А.

(подпись) (расшифровка подписи)

Итоговая оценка по практике

Санкт-Петербург

2024

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| **1** | **2** | **3** |
| 09.12.24 | Анализ предметной области. |  |
| 09.12.24 | Выявление требований к программе. |  |
| 10.12.24 | Разработка технического задания. |  |
| 10.12.24 | UML. Проектирование диаграммы вариантов использования. |  |
| 11.12.24 | UML. Проектирование диаграммы последовательности. |  |
| 11.12.24 | UML. Проектирование диаграммы активности. |  |
| 12.12.24 | Моделирование структуры ПО. |  |
| 12.12.24 | Проектирование инфологической и даталогической модели данных. |  |
| 13.12.24 | Проектирование интерфейса пользователя. Создание Wireframe эскизов. |  |
| 13.12.24 | Разработка дизайна программы в соответствии с руководством по стилю. |  |
| 14.12.24 | Разработка базы данных. |  |
| 14.12.24 | Разработка словаря данных. |  |
| 16.12.24 | Создание приложения. Форма авторизации. |  |
| 16.12.24 | Создание приложения. Форма заказов. |  |
| 17.12.24 | Создание приложения. Основные формы приложения. |  |
| 17.12.24 | Разработка библиотеки классов. |  |
| 18.12.24 | Подготовка отчетов и выгрузка документов для печати. |  |
| 18.12.24 | Отладка программных модулей. |  |
| 19.12.24 | Модульное тестирование. |  |
| 19.12.24 | Создание тестовых случаев. |  |
| 20.12.24 | Интеграционное тестирование. |  |
| 20.12.24 | Разработка самодокументирующегося кода. |  |
| 21.12.24 | Инспекция кода на соответствие стандартам кодирования. |  |
| 21.12.24 | Подготовка отчета и размещение результатов в репозитории контроля версий. |  |

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| **1** | **2** | **3** |
| 09.12.24 | Участие в выработке требований к программному обеспечению. |  |
| 11.12.24 | Стадии проектирования программного обеспечения. |  |
| 13.12.24 | Разработка модулей программного обеспечения. |  |
| 19.12.24 | Тестирование программных модулей и их интеграции. |  |
| 20.12.24 | Разработка программной документации. Стандарты кодирования. |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc185498796)

[Задание №1 «Разработка технического задания» 8](#_Toc185498797)

[Задание №2 «Разработка алгоритмов и диаграмм» 17](#_Toc185498803)

[Задание №3 «Разработка макетов» 31](#_Toc185498813)

[Задание №4 «Разработка программы» 68](#_Toc185498818)

[Задание №5 «Тестирование» 91](#_Toc185498824)

[Задание №6 «GitHub.Оценка проекта» 105](#_Toc185498828)

[ЗАКЛЮЧЕНИЕ 116](#_Toc185498834)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 117](#_Toc185498835)

[ПРИЛОЖЕНИЕ 1 118](#_Toc185498836)

ВВЕДЕНИЕ

Учебная практика проводилась в ФГАОУ ВО СПбПУ, Институт среднего профессионального образования, в период с 09.12.2024 по 21.12.2024. Этот этап обучения является важным звеном в подготовке квалифицированных специалистов, поскольку он позволяет применить теоретические знания на практике и получить опыт работы в реальных условиях.

Учебная практика представляет собой форму учебно-вспомогательной деятельности, направленной на развитие практических навыков и компетенций, необходимых для успешной профессиональной деятельности. В процессе практики студенты осваивают современные технологии и инструменты, что способствует их профессиональному развитию и адаптации к требованиям рынка труда.

Целью практики было закрепление практических умений и приобретение дополнительных навыков в области разработки программных модулей для компьютерных систем. Для достижения этой цели были выполнены задачи, такие как разработка спецификаций отдельных компонентов, проектирование пользовательского интерфейса, создание программных модулей, их модульное тестирование и документирование программных продуктов.

Во время практики студенты решали реальные задачи, что позволило им развить навыки, необходимые для работы в сфере информационных технологий. Результаты практики подтверждают уровень подготовки студентов и их готовность к профессиональной деятельности.

Задание №1 «Разработка технического задания»

**Цель работы:** развитие навыков разработки технического задания (ТЗ) и понимания процесса взаимодействия между заказчиком и исполнителем.

Поделиться на пары: каждый из вас будет и исполнителем, и заказчиком. Заказчику необходимо разработать задание для исполнителя согласно его варианту, которое будет состоять из описания предметной области со всеми пожеланиями к готовому программному продукту, включая внешний вид (желаемый цвет фона, шрифт и т.д.). После составления предметной области необходимо поменяться готовым заданием друг с другом и приступить к обсуждению, по результатам которого исполнитель, так же каждый из вас, должен будет составить подробное техническое задание со всеми функциями, описанные заказчиком. Во время обсуждения предметная область может дорабатываться. Учтите при составлении предметной области и технического задания следующие требования:

* Минимум 3 роли (например, администратор, сотрудник и клиент);
* Возможность реализовать не менее 5 таблиц в БД;
* Программный продукт должен включать основные функции, такие как: удаление/добавление/изменение всех ключевых таблиц, их корректное отображение, вывод ошибок и предупреждений, авторизация, масштабирование, составление и выгрузка отчетов, админ панель.

Варианты:

* 1. Торговля
  2. Турагентство
  3. Библиотека
  4. Поликлиника
  5. ГАИ
  6. Цветочный магазин
  7. Типография
  8. Авиакомпания
  9. Аптека
  10. Склад
  11. Фабрика
  12. Банк
  13. Прокат автомобилей
  14. Салон красоты
  15. Школа
  16. Почта
  17. Фитнес-клуб
  18. Спортивный клуб (Футбольный клуб/Баскетбольный)
  19. Автосалон
  20. Музей
  21. Ресторан
  22. Поставка товаров
  23. Страховая компания
  24. Компьютерная мастерская
  25. Компьютерный клуб
  26. Ломбард
  27. Гостиница
  28. Химчистка
  29. Театр
  30. Агентство недвижимости

Выполнение задания

Вариант «Типография»

### **Описание предметной области**

Основной целью данного модуля является автоматизация процесса учета и обработки заказов в типографии. Модуль позволяет сотрудникам типографии прозрачно и эффективно управлять заказами, отслеживать их текущий статус.

Разработка программного модуля для учета заказов типографии — это процесс создания программного продукта, который позволяет типографиям эффективно отслеживать, учитывать и управлять заказами клиентов на печатную продукцию.

Задача разработки такого программного модуля состоит в создании удобного и эффективного инструмента для учета, отслеживания и управления заказами, что помогает оптимизировать производственные процессы и повысить удовлетворенность клиентов.

Программный модуль позволяет типографии повысить точность учета заказов и сократить время обработки.

**Основные функции и возможности модуля включают:**

1. **Прием заказа**:

Клиент оформляет новый заказ, указывая параметры продукции (тип продукции, тираж, формат).

1. **Регистрация заказа**:

Оператор принимает заказ, присваивает уникальный идентификатор, вносит данные в систему.

1. **Изменение заказа:**

Возможность удалить заказ до того, как оператор примет его в обработку.

1. **Исполнение заказа**:

Выполнение этапов работы: печать, постпечатная обработка (резка, ламинирование, переплет).

1. **Отчётность и информирование**:

Изменение статуса заказа.

### **Техническое задание**

1. **Общие сведения**

**Наименование проекта:** Разработка программного модуля для учета заказов типографии.

* 1. **Заказчик:** ИП Рождественская Анна.
  2. **Исполнитель:** Компания «TechPrint».

1. **Функциональные требования**
   1. Возможность добавления заказов в базу данных с указанием следующих параметров:

* Номер заказа;
* Дата оформления;
* Тип продукции (листовки, брошюры, визитки, книги);
* Формат печати (A4, A3, нестандартный);
* Тираж;
* Цветность (монохромная/цветная);
* Вид отделки (ламинирование, переплет, порезка);
* ФИО клиента;
* Контактный номер телефона;
* Статус заказа (новый, в производстве, выполнен, выдан клиенту).
  1. Возможность отслеживания статуса заказа:
* Отображение списка заказов;
* Изменение статуса заказа;
* Редактирование заказа;

# 3. Нефункциональные требования

## **3.1. Кроссплатформенность:**

* Поддержка работы на ОС Windows.

## **3.2. Безопасность:**

* Авторизация по логину и паролю;
* Пароли должны храниться в зашифрованном виде;
* Ограничение доступа к функциям приложения в зависимости от роли пользователя (администратор, сотрудник, клиент).

## **3.3. Удобство использования:**

* Интуитивно понятный интерфейс с подсказками (использование простых элементов управления (кнопки, выпадающие списки, поля ввода), минимизация числа действий для выполнения базовых операций);
* Уведомления о важной информации и ошибках.

## **3.4. Производительность:**

* Минимальное время отклика на пользовательские запросы.

# 4. Требования к реализации

**4.1.** Язык программирования: C# **10.0 и выше** (совместимость с .NET 6 и выше).

**4.2.** Фреймворк: .NET 8.0 **Windows Forms**.

**4.3.** СУБД: MS SQL Server 2019 и выше.

# 5. Требования к документации

**5.1.** Техническое задание на разработку программного модуля.

**5.2.** Руководство системному программисту с описанием структуры базы данных и функционала приложения.

1. **Руководство по стилю**
   1. **Оформление интерфейса:**

* Использование фиолетовых и белых цветов.
* На основных окнах должно быть лого (Приложение В).

Задание №2 «Разработка алгоритмов и диаграмм»

**Цель работы**: разработка и реализация системы проектирования и моделирования, включающей создание алгоритмов в виде блок-схем и UML-диаграмм, а также проектирование базы данных, соответствующей требованиям третьей нормальной формы (3НФ) с обеспечением ссылочной целостности. Работа направлена на развитие навыков анализа предметной области, проектирования архитектуры системы, разработки алгоритмов и баз данных, а также на практическое применение методологий моделирования и проектирования.

По составленному техническому заданию необходимо разработать алгоритмы в виде блок-схем и UML-диаграммы:

* диаграммы вариантов использования;
* диаграммы последовательности;
* диаграммы активности.

Все диаграммы и блок-схемы должны подробно отображать разные процессы.

Также необходимо разработать ER-диаграмму для БД + словарь данных. Обязательно 3 нормальная форма с обеспечением ссылочной целостности. При разработке диаграммы обратите внимание на согласованную осмысленную схему именования, создайте необходимые первичные и внешние ключи, определите ограничения внешних ключей, отражающие характер предметной области.

Создайте все необходимые сущности, определите отношения, создайте ограничения на связи между сущностями (при наличии всех связей), приведите базу данных к 3НФ (при наличии всех сущностей и связей).

Создайте базу данных, используя предпочтительную платформу, на сервере баз данных. Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения.

Выполните названия таблиц и полей в едином стиле. Заполните таблицы данными.

Выполнение задания

### **Диаграммы вариантов использования**

Рисунок 1 – Диаграмма вариантов использования

### **Диаграммы последовательности**

**Сценарий:** Регистрация новой заявки

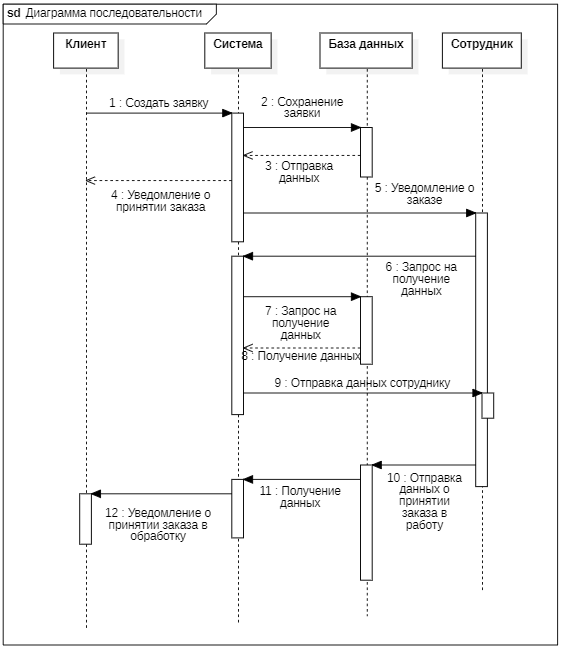


Рисунок 2 – Диаграмма последовательности

### **Диаграммы активности**

**Сценарий**: Формирование штрафа

Рисунок 3 – Диаграмма активности

### **ER-диаграмма базы данных**

Рисунок 4 – ER-диаграмма базы данных

### **Словарь данных**

Таблица 1 – Словарь данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сущность | Атрибут | Тип данных | Описание | Пример |
| Users | UserID | INT (PK) | Уникальный идентификатор пользователя. | 1, 2, 3 |
|  | Username | VARCHAR(50) | Имя пользователя. | "john\_doe", "admin" |
|  | PasswordHash | VARCHAR(255) | Хэш пароля пользователя. | "5e884898da28047151d0e56f8dc6292773" |

Продолжение таблицы 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RoleID | INT (FK) | Идентификатор роли пользователя. | 1 |
| Roles | RoleID | INT (PK) | Уникальный идентификатор роли. | 1, 2, 3 |
|  | RoleName | VARCHAR(50) | Название роли (например, "Администратор", "Сотрудник ГАИ", "Клиент"). | "Администратор", "Клиент" |
| Drivers | DriverID | INT (PK) | Уникальный идентификатор водителя. | 1, 2, 3 |
|  | FullName | VARCHAR(100) | Полное имя водителя. | "Иванов Иван Иванов" |
|  | DateOfBirth | DATE | Дата рождения водителя. | "1985-05-15" |
|  | LicenseNumber | VARCHAR(20) | Номер водительского удостоверения. | "1234567890" |
|  | UserID | INT (FK) | Идентификатор пользователя, представляющего водителя. | 1 |
| Brands | Brand | INT (PK) | Уникальный идентификатор марки автомобиля. | 1, 2, 3 |
|  | Description | VARCHAR(100) | Описание марки автомобиля. | "Toyota", "Ford" |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Vehicles | VehicleID | INT (PK) | Уникальный идентификатор транспортного средства. |
|  | Brand | INT (FK) | Идентификатор марки автомобиля. |
|  | Model | VARCHAR(50) | Модель автомобиля. |
| Vehicles | VehicleID | INT (PK) | Уникальный идентификатор транспортного средства. |
|  | Brand | INT (FK) | Идентификатор марки автомобиля. |
| ViolationType | VARCHAR(50) | Тип нарушения. | "Превышение скорости", "Парковка" |
| FineID | INT (PK) | Уникальный идентификатор штрафа. | 1, 2, 3 |
| DriverID | INT (FK) | Идентификатор водителя, получившего штраф. | 1 |
| DateOfViolation | DATE | Дата нарушения. | "2023-10-01" |
| ViolationID | INT (FK) | Идентификатор типа нарушения. | 1 |
| Amount | DECIMAL(10, 2) | Сумма штрафа. | 500.00 |

Продолжение таблицы 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | IsPaid | BIT | Флаг, указывающий, оплачен ли штраф (0 — не оплачен, 1 — оплачен). | 0 |
| Accidents | AccidentID | INT (PK) | Уникальный идентификатор ДТП. | 1, 2, 3 |
|  | Date | DATE | Дата ДТП. | "2023-09-15" |
|  | Time | TIME | Время ДТП. | "14:30:00" |
|  | Location | VARCHAR(100) | Место происшествия. | "ул. Ленина, 10" |
|  | Description | TEXT | Описание ДТП. | "Столкновение на перекрестке" |
|  | DamageAmount | DECIMAL(10, 2) | Сумма ущерба. | 2500.00 |
| Reports | ReportID | INT (PK) | Уникальный идентификатор отчета. | 1, 2, 3 |
|  | ReportType | VARCHAR(50) | Тип отчета. | "Штрафы за месяц", "ДТП" |
|  | DateGenerated | DATE | Дата создания отчета. | "2023-10-05" |
|  | Content | TEXT | Содержимое отчета. | "Список штрафов за сентябрь" |

Продолжение таблицы 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cameras | CameraID | INT (PK) | Уникальный идентификатор камеры. | 1, 2, 3 |
|  | Location | VARCHAR(100) | Местоположение камеры. | "ул. Мира, 5" |
|  | Status | VARCHAR(50) | Статус камеры. | "Online", "Offline" |
| PaymentMethods | PaymentMethodID | INT (PK) | Уникальный идентификатор способа оплаты. | 1, 2, 3 |
|  | PaymentMethodName | VARCHAR(50) | Название способа оплаты. | "Карта", "Электронный кошелек" |
| Payments | PaymentID | INT (PK) | Уникальный идентификатор платежа. | 1, 2, 3 |
|  | FineID | INT (FK) | Идентификатор штрафа, который оплачен. | 1 |
|  | UserID | INT (FK) | Идентификатор пользователя, совершившего платеж. | 1 |
|  | PaymentDate | DATETIME | Дата и время платежа. | "2023-10-01 12:00:00" |

Окончание таблицы 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PaymentAmount | DECIMAL(10, 2) | Сумма платежа. | 500.00 |
|  | PaymentMethodID | INT (FK) | Идентификатор способа оплаты. | 1 |

### **Алгоритмы программы**

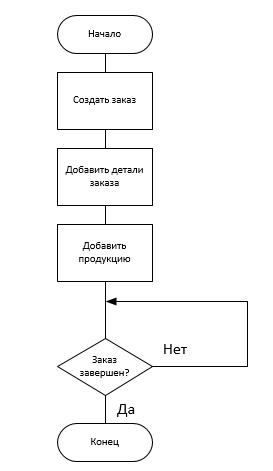


Рисунок 9 – Диаграмма метода создания заказа

Задание №3 «Разработка макетов»

**Цель работы:** изучение и применение различных видов макетов (Wireframe, Mockup, Prototype) для проектирования интерфейса программного продукта, соответствующего требованиям заказчика. Работа направлена на развитие навыков визуализации и прототипирования пользовательских интерфейсов, а также на создание удобной и интуитивно понятной карты навигации приложения. Дополнительно целью является освоение анимации в прототипах для повышения интерактивности и удобства использования будущего продукта.

Изучить информацию о разных видах макетов. Ссылка на ресурс: <https://blog.sibirix.ru/mockup-wireframe-prototype-sketch/>

Необходимо разработать макеты интерфейса будущего программного продукта, опираясь на требования заказчика. Вы должны представить макеты Wireframe (черно-белые) и Mockup (цветные). Работу выполнять в графическом редакторе, например Figma. За скрины форм с Visual Studio максимум 3 балла.

Разработать карту навигации приложения.

Дополнительное задание: разработать прототипы (макеты с анимацией).

Выполнение задания

### **Wireframe-макеты**

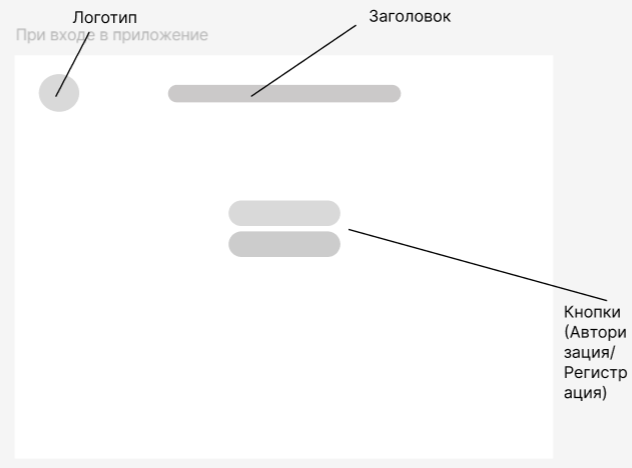
****

Рисунок 11 – Wareframe-макет главного окна приложения

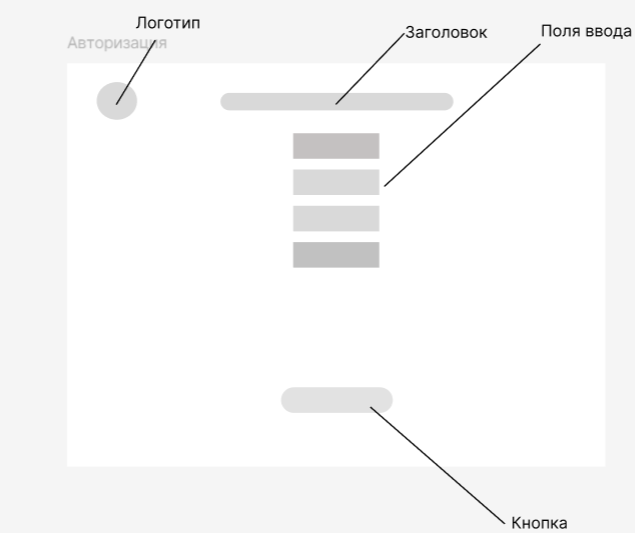
****

Рисунок 12 – Wareframe-макет окна авторизации

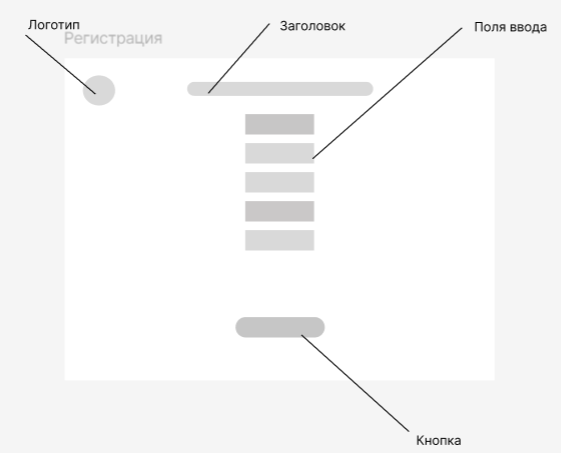
****

Рисунок 13 – Wareframe-макет главного окна регистрации

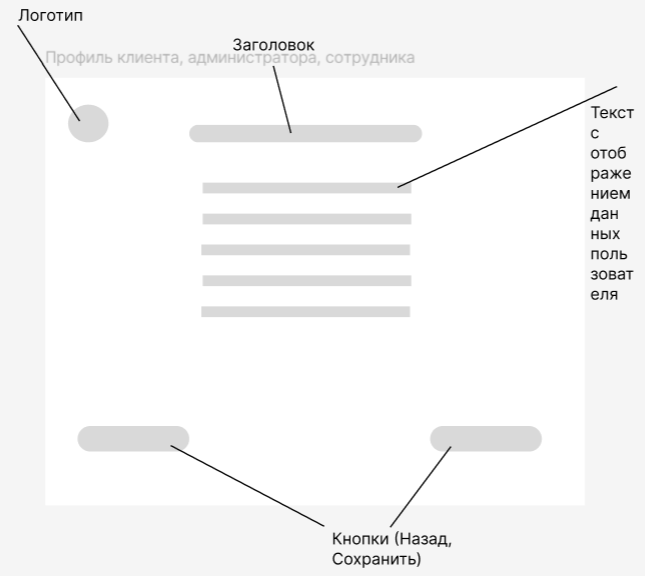
****

Рисунок 14 – Wareframe-макет окна просмотра информации о пользователе

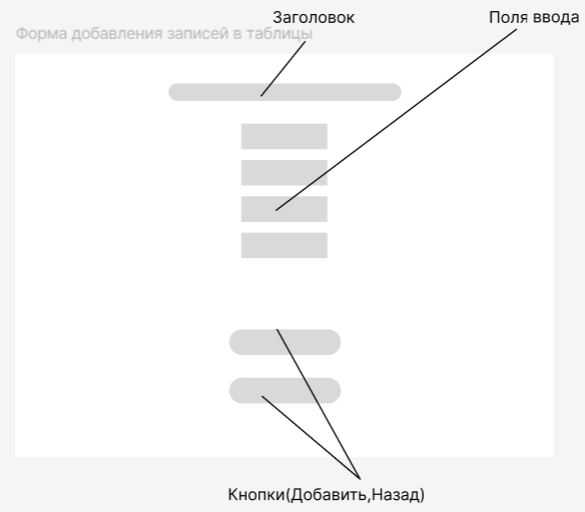
****

Рисунок 15 – Wareframe-макет окна добавления записей в таблицы

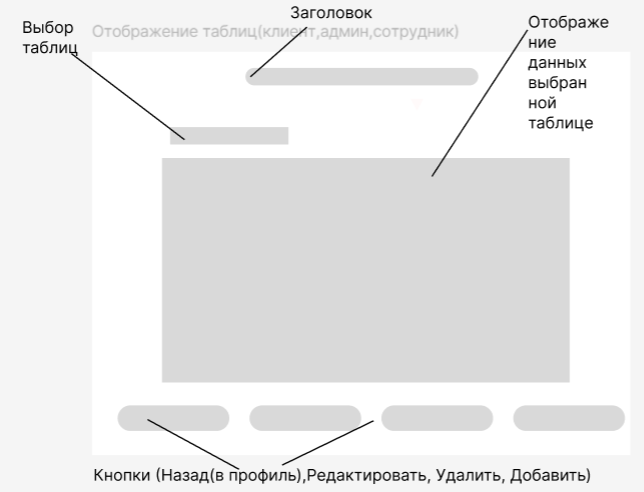
****

Рисунок 16 – Wareframe-макет окна просмотра основной формы

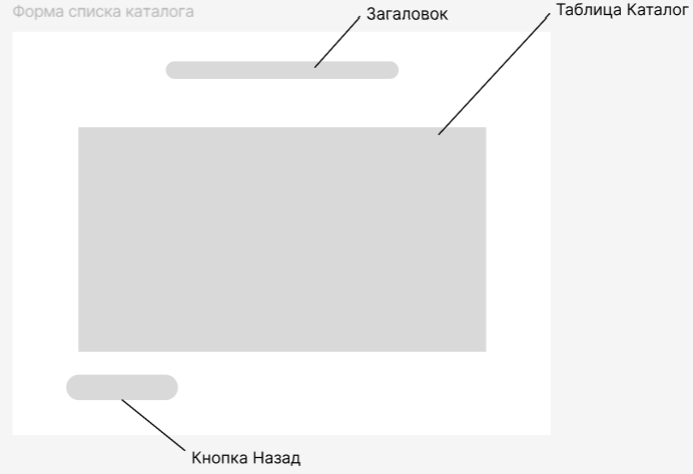
****

Рисунок 17 – Wareframe-макет окна просмотра каталога

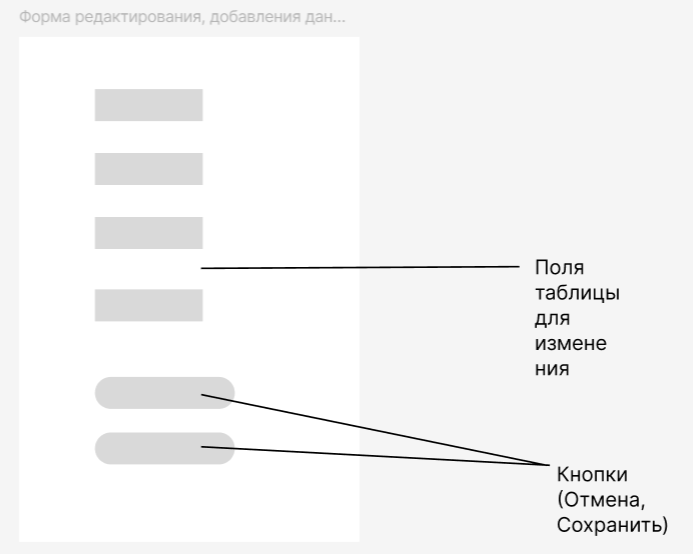
****

Рисунок 18 – Wareframe-макет окна добавления, редактирование записей

### **Mockup-макеты**

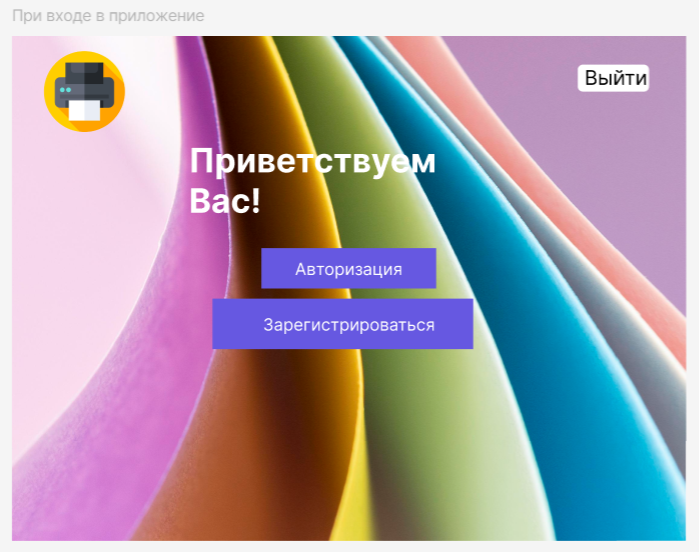
****

Рисунок 28 – Mockup-макет главного окна приложения

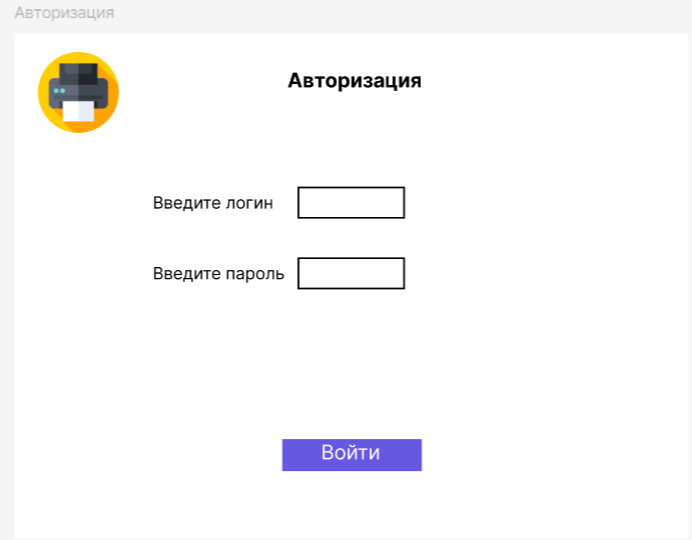


Рисунок 29 – Mockup-макет окна авторизации

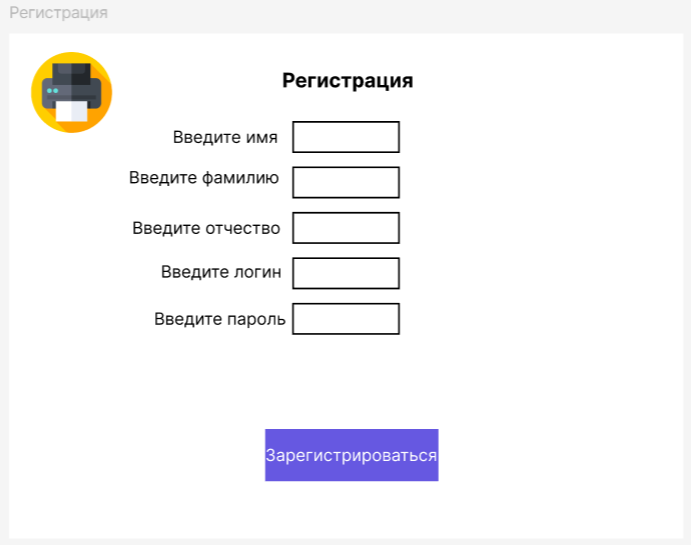


Рисунок 30 – Mockup-макет окна регистрации

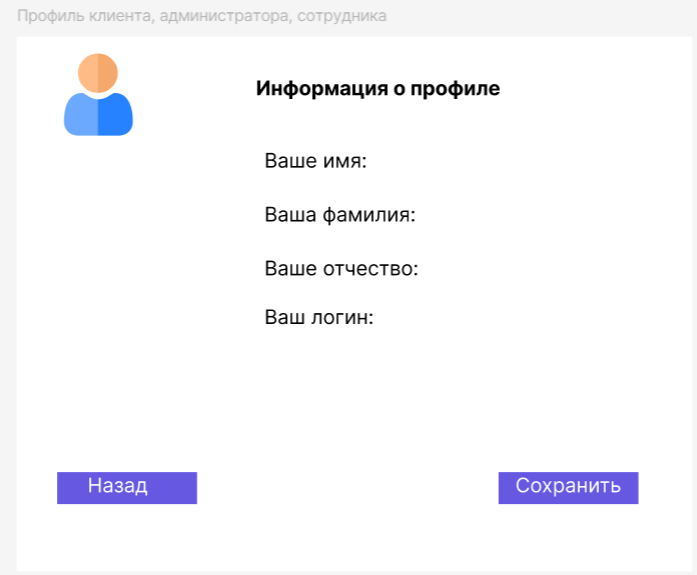
****

Рисунок 31 – Mockup-макет окна просмотра информации о пользователе

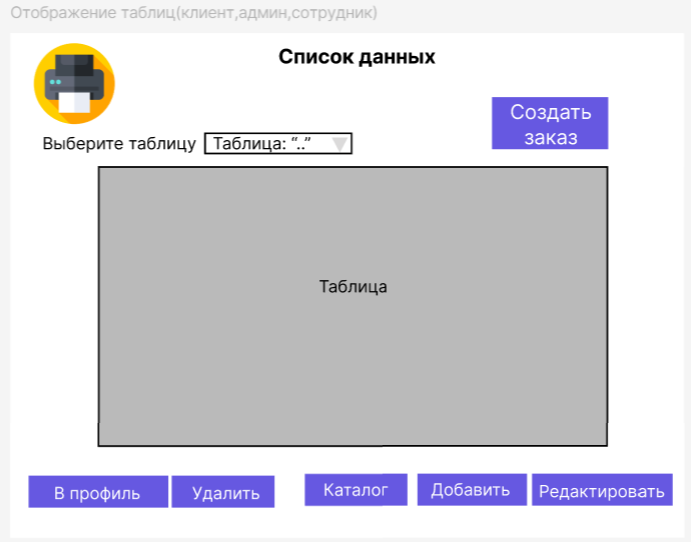
****

Рисунок 32 – Mockup-макет главного окна

****

Рисунок 33 – Mockup-макет окна просмотра каталога

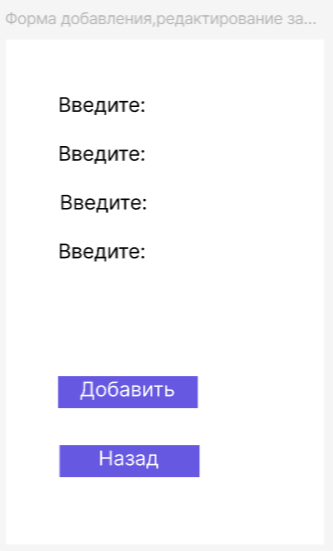
****

Рисунок 34 – Mockup-макет окна создания, редактирование записей в таблицы

### **Карта навигации приложения**



Рисунок 49 – Карта навигации

Задание №4 «Разработка программы»

**Цель работы:** разработка программного продукта, соответствующего составленным алгоритмам, диаграммам, макетам и техническому заданию. Работа направлена на создание последовательного и удобного пользовательского интерфейса, обеспечение единообразия внешнего вида компонентов системы, а также реализацию обработки исключительных ситуаций для повышения стабильности и удобства использования приложения. Целью также является соблюдение стандартов кодирования, написание чистого и понятного кода, а также проведение отладки и тестирования программного обеспечения для обеспечения его корректной работы.

Необходимо разработать программный продукт, опираясь на составленные алгоритмы, диаграммы, макеты и техническое задание.

  Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий макетам, а также следующим требованиям:

- последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);

- соответствующий заголовок на каждом окне приложения.

Выполнить исходный код модуля в соответствии гайдлайну: идентификаторы должны соответствовать соглашению об именовании, например (CodeConvention), стилю CamelCase (для C# и Java), snake\_case (для Python) и <https://its.1c.ru/db/v8std#browse:13:-1:31> (для 1C).

Допустимо использование не более одной команды в строке.

Необходимо использовать комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода. Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения.

Реализовать программные обработки исключительных ситуаций в приложении. Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных.

Выполнить отладку модуля.

Выполнить отладку программного обеспечения с использованием инструментальных средств. Сохранить и представить результаты в скриншотах.

Выполнение задания

На основе Mockup-макетов (Рисунок 30–50) были реализованы следующие окна приложения:

### **Главное окно, авторизация и регистрация**

Рисунок 50 – Главное окно приложения

Рисунок 51 – Окно авторизации

Рисунок 51 – Окно регистрации

### **Главное окно**

Рисунок 52 – Главное окно роли «Водителя»

### **Окно профиля**

Рисунок 58 – Главное окно роли «Сотрудник ГАИ»

### **Окно каталога**

Рисунок 67 – Главное окно роли «Администратор»

**Окна редактирование записи, создание записи и создание заказа(для клиента)**Рисунок 67 – Главное окно роли «Администратор»

Рисунок 67 – Главное окно роли «Администратор»

Рисунок 67 – Главное окно роли «Администратор»

Задание №5 «Тестирование»

**Цель работы:** разработка и документирование 5 Test-Case, которые проверяют наиболее важный процесс в системе. Эти тест-кейсы должны охватывать различные сценарии и ситуации, чтобы обеспечить полное тестирование функциональности. Выбранный процесс должен быть критически важным для функционирования системы, и тест-кейсы должны быть оформлены в соответствии с предоставленным шаблоном.

Подготовьте 5 Test-Case, проверяющие один наиболее важный на ваш взгляд процесс в системе. Важно, чтобы тестовые данные предусматривали различные ситуации. При оформлении используйте представленный ниже шаблон.

Аннотация теста:

|  |  |
| --- | --- |
| **Название проекта** |  |
| **Рабочая версия** |  |
| **Имя тестирующего** |  |
| **Дата(ы) теста** |  |

Расшифровка тестовых информационных полей:

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Название тестируемого проекта |
| **Рабочая версия** | Версия проекта/программного обеспечения (первый тест считается 1.0). |
| **Имя тестирующего** | Имя того, кто проводил тесты |
| **Дата(ы) теста** | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста. |
| **Тестовый пример #** | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например,‘TC\_UI\_1′ означает‘user interface test case #1′ ( ТС\_ПИ\_1: тестовый случай пользовательского интерфейса#1) |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет. |
| **Заголовок/название теста** | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| **Краткое изложение теста** | Описание того, что должен достичь тест. |
| **Этапы теста** | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея. |
| **Тестовые данные** | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа. |
| **Ожидаемый результат** | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране. |
| **Фактический результат** | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста. |
| **Предварительное условие** | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая. |
| **Постусловие** | Каким должно быть состояние системы после выполнения теста? |
| **Статус**  *(Зачет/Незачет)* | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено. |
| **Примечания/комментарии** | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами). |

Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** |  |
| **Приоритет тестирования** |  |
| **Заголовок/название теста** |  |
| **Краткое изложение теста** |  |
| **Этапы теста** |  |
| **Тестовые данные** |  |
| **Ожидаемый результат** |  |
| **Фактический результат** |  |
| **Статус** |  |
| **Предварительное условие** |  |
| **Постусловие** |  |
| **Примечания/комментарии** |  |

Модульные тесты

В рамках проекта разработайте тесты, максимально полно покрывающие функционал метода. Ничего страшного, если ваш метод работает не совсем идеально и тесты могут быть не пройдены в связи с этим - в данном модуле это не так важно. Обратите внимание, что имена тестов должны отражать их суть, т.е. вместо TestMethod1() тест следует назвать, например, GetQuantityForProduct\_NonExistentProductType() для тестирования случая передачи несуществующего типа продукции.

Реализуйте 10 unit-тестов. Важно, чтобы тестовые данные предусматривали различные ситуации.

Выполнение задания

### **Tests-case**

|  |  |
| --- | --- |
| **Название проекта** | ПринтПлюс |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Алексеева Полина Александровна |
| **Дата(ы) теста** | 20.12.2024 |

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация нового пользователя с пустыми полями |
| **Краткое изложение теста** | Проверка поведения системы при попытке регистрации пользователя с незаполненными полями. |
| **Этапы теста** | 1. Открыть окно регистрации. 2. Оставить все поля пустыми. 3. Нажать кнопку "Зарегистрировать". |
| **Тестовые данные** | * ФИО: "" * Логин: "" * Пароль: "" |
| **Ожидаемый результат** | Появление сообщения: "Пожалуйста, заполните все поля!" |
| **Фактический результат** | Сообщение: "Пожалуйста, заполните все поля!" |
| **Статус** | Зачет (Приложение А. Рисунок 2) |
| **Предварительное условие** | Открыто окно регистрации. |
| **Постусловие** | Данные в базе данных не изменились. |
| **Примечания/комментарии** | - |

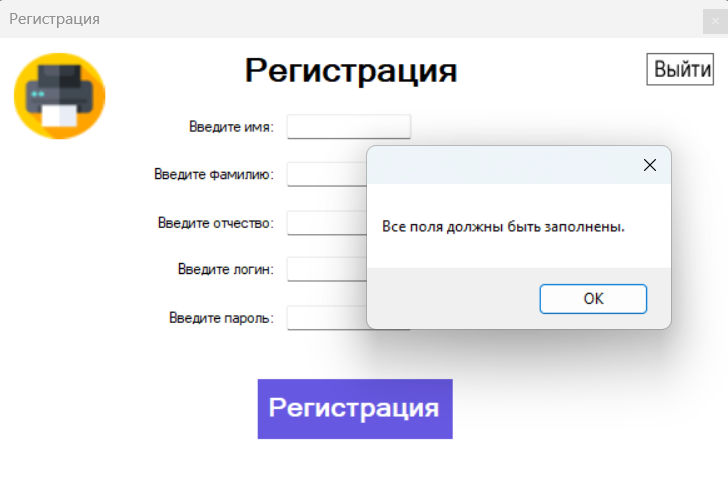


Рисунок 74 – Успешный вывод ошибки

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Создание нового заказа с корректными данными |
| **Краткое изложение теста** | Проверка корректного создания нового заказа с валидными данными. |
| **Этапы теста** | 1. Открыть окно создания заказа. 2. Выбрать продукцию из выпадающего списка (например, "Листовки"). 3. Ввести валидные данные в поля:  * Тираж: "20" * Формат: «А4»  1. Проверка автоматического заполнения полей:   Цена:  Общая стоимость:   1. Нажать кнопку "Зарегистрировать". |
| **Тестовые данные** | * ТипПродукции: "Листовки" * Формат: "А4" * Тираж: "20" * Цена: "50" * Общая стоимость: "1000" |
| **Ожидаемый результат** | Появление сообщения: "Заказ создан!"  Закрытие окна создание заказа. |
| **Фактический результат** | Сообщение: "Заказ создан!"  Окно регистрации закрыто. |
| **Статус** | Зачет (Приложение А. Рисунок 3) |
| **Предварительное условие** | Открыто окно создания заказа. |
| **Постусловие** | В базе данных появилось новый заказ с указанными данными. |
| **Примечания/комментарии** | - |

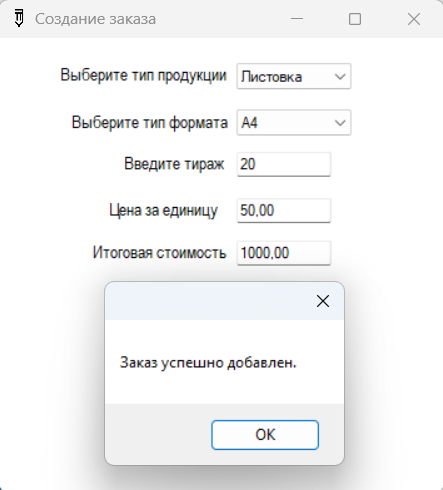
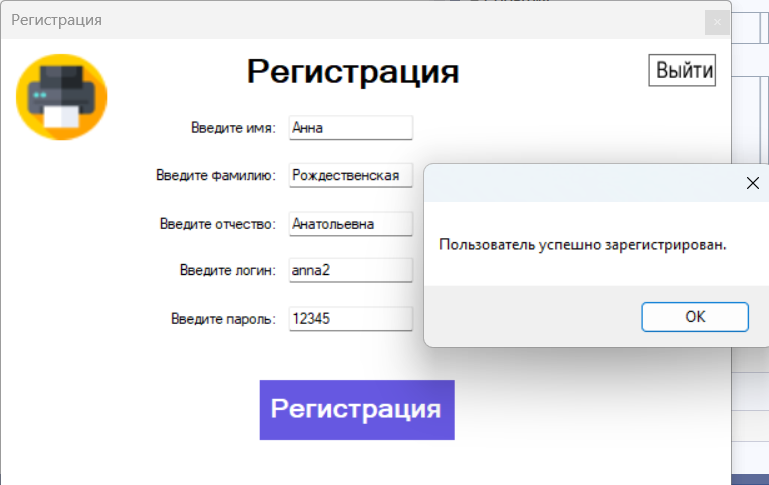
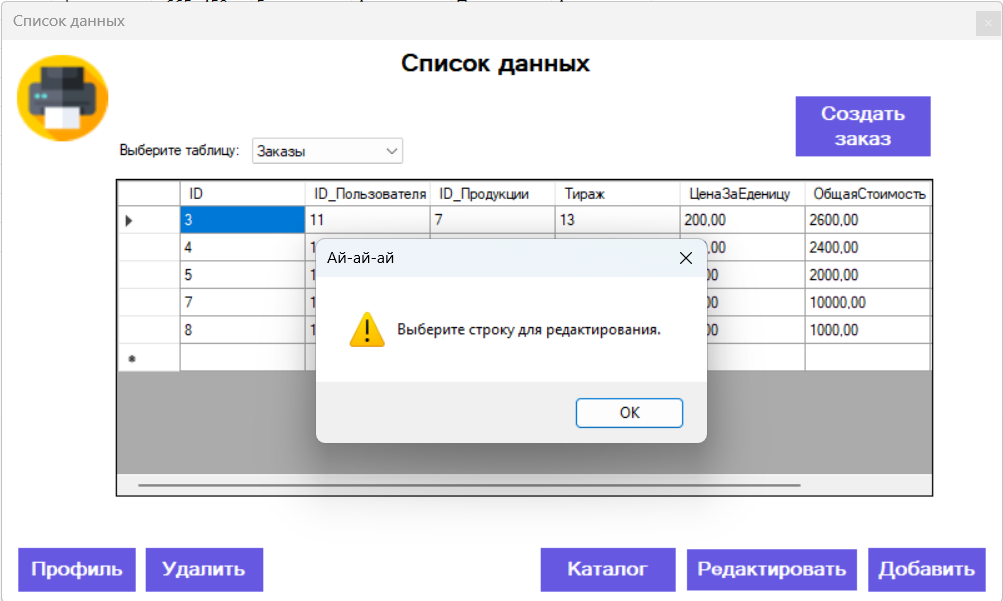


Рисунок 75 – Успешный вывод сообщения

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_3 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка изменения полей профиля пользователя |
| **Краткое изложение теста** | Проверка поведения системы при изменении данных пользователя. |
| **Этапы теста** | 1. Открыть окно профиля после регистрации пользователя. Данные пользователя:  * ФИО: " Рождественская Анна Анатольевна " * Логин: " anna2"  1. Ввести новые валидные данные в поле:  * Логин: "anna20"  1. Нажать кнопку "Сохранить". |
| **Тестовые данные** | * ФИО: " Рождественская Анна Анатольевна " * Логин: " anna20" |
| **Ожидаемый результат** | Появление сообщения: "Данные профиля сохранились!" |
| **Фактический результат** | Сообщение: "Данные профиля сохранились!" |
| **Статус** | Зачет (Приложение А. Рисунок 4) |
| **Предварительное условие** | Открыто окно профиля. |
| **Постусловие** | Данные в базе данных изменились. |
| **Примечания/комментарии** | - |

Рисунок 76 – Успешная регистрация

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_4 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Проверка вывода ошибки при нажатии на кнопку «Редактировать». |
| **Краткое изложение теста** | Проверка вывода ошибки при не выбранной строке таблицы и нажатия на кнопку «Редактировать» |
| **Этапы теста** | 1. Открыть окно «Список Данных». 2. Не выбирать всю строку в таблице, только ячейку или столбец. 3. Нажать кнопку "Редактировать". |
| **Тестовые данные** | Пользователь с ролью "Администратор". |
| **Ожидаемый результат** | Появление сообщения: «Выберите строку для редактирования» |
| **Фактический результат** | Сообщение: "Выберите строку для редактирования! |
| **Статус** | Зачет (Приложение А. Рисунок 5) |
| **Предварительное условие** | Открыто основное окно. |
| **Постусловие** | Данные в базе данных не изменились. |
| **Примечания/комментарии** | - |

Рисунок 77 – Успешный вывод ошибки

### **Unit-tests**

Рисунок 79 – Успешное прохождения unit-tests.

Задание №6 «GitHub.Оценка проекта»

**Цель работы:** выполнение задания по выгрузке готового клиентского приложения, приложения-тестировщика (unit-test) и отчета в репозиторий GIT, а также проведение оценки проекта по заданным критериям. В рамках этой работы также необходимо предоставить заказчику оценку готовой программы, сравнив ожидания с реальностью. Это позволит продемонстрировать навыки работы с системой контроля версий GitHub, умение создавать и оформлять проекты, а также проводить анализ и оценку качества разработанного программного обеспечения

В репозиторий GIT выгрузите получившееся клиентское приложение,

приложение-тестировщик unit-test и отчет. В названии отчета укажите ваши номер группы и ФИО.

Провести оценку проекта по таблице (файл «Критерии профиль»). Заказчику также нужно дать свою оценку готовой программе, на сколько ожидания совпали с реальностью.

Выполнение задания

### **GitHub**

****Рисунок 80 – Заходим на аккаунт GitHub и создаем репозиторий

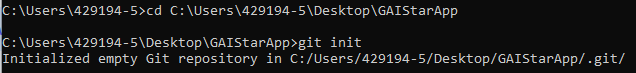


Рисунок 82 – Инициализируем Git в локальной папке

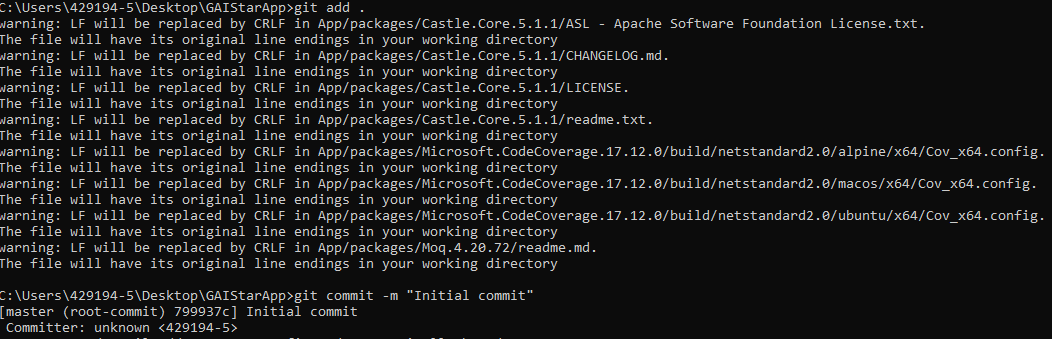


Рисунок 83 – Добавим все файлы из папки в индекс Git



Рисунок 84 – Подключаем локальный репозиторий к GitHub



Рисунок 85 – Отправляем все в GitHub

### **Ссылка на проект**

[**https://github.com/AlexsandraZahz/GAIStarApp.git**](https://github.com/AlexsandraZahz/GAIStarApp.git)

**Оценка проекта по критериям**

**А1Д1: Анализ технического задания, составление краткой спецификации разрабатываемого модуля (входные и выходные данные), формирование алгоритмов в соответствии с техническим заданием**

* **Балл: 3**
  + Техническое задание проанализировано.
  + Составлена краткая спецификация, описаны входные и выходные данные.
  + Алгоритмы разработаны и соответствуют техническому заданию.

**А2Д1: Оформление алгоритмов представлено одним из трех способов и выполнено в соответствии со стандартами (ГОСТ 19.701-90 или ГОСТ 2.105 или ГОСТ 24.301)**

* **Балл: 3**
  + Алгоритмы оформлены в соответствии со стандартами.

**А3Д1: Реализация последовательности алгоритма по этапам (входные данные равны выходным данным), разработка алгоритма с использованием данных, определенных в техническом задании**

* **Балл: 3**
  + Рассмотрены все варианты работы алгоритма.
  + Реализована последовательность алгоритма по этапам.
  + Алгоритм использует все необходимые данные.

**Б1Д1: Программный модуль разрабатывается по имеющемуся алгоритму в среде разработки и полностью соответствует техническому заданию**

* **Балл: 6**
  + Программный модуль разработан по имеющемуся алгоритму.
  + Модуль полностью соответствует техническому заданию.

**Б2Д1: Выполнение исходного кода модуля в соответствии гайдлайну (название приложения, четкость, смысл именования форм и переменных очевидны)**

* **Балл: 4**
  + Исходный код модуля соответствует гайдлайну.
  + Название приложения, форм и переменных четко отражают их назначение.

**В1Д1: Использование отладки и программной обработки исключительных ситуаций, что позволяет исключить фатальные ошибки при работе приложения, а также приложение аварийно не завершает работу**

* **Балл: 4**
  + Выполнена отладка модуля и программная обработка исключительных ситуаций.
  + Фатальные ошибки не возникают, приложение аварийно не завершает работу.

**В2Д1: Выполнение отладки программного обеспечения с использованием инструментальных средств, с пояснением особенностей отладочных классов, сохранение и представление результатов в виде текстового документа**

* **Балл: 3**
  + Выполнена отладка программного обеспечения с использованием инструментальных средств.
  + Пояснены особенности отладочных классов.
  + Результаты представлены в виде текстового документа.

**Г1Д1: Оформление результатов тестирования в виде протокола тестирования в соответствии со стандартами**

* **Балл: 3**
  + Результаты тестирования оформлены в виде протокола тестирования, соответствуют стандартам.

**Г2Д1: Определение наборов входных данных, выполнение функционального тестирования модуля по определенному сценарию, проведение тестирования для проверки функциональности программы (хотя бы 1 тест на 1 функцию), представление результатов тестирования**

* **Балл: 1,5**
  + Созданы тесты для проверки функциональности программы (хотя бы 1 тест на 1 функцию).
  + Наборы входных данных определены верно.
  + Результаты тестирования представлены.

**Г3Д1: Использование инструментальных средств для тестирования функциональности программы**

* **Балл: 3**
  + Выполнено тестирование программного продукта с помощью инструментальных средств.

**Д1Д1: Проведение анализа поставленной задачи и проектирования базы данных (ERD модели) с применением case-средств, создание всех необходимых сущностей, определение отношений, создание ограничений на связи между сущностями (при наличии всех связей), приведение базы данных к 3НФ (при наличии всех сущностей и связей)**

* **Балл: 3**
  + Спроектирована и нормализована база данных в соответствии с поставленной задачей.
  + Уровень нормализации соответствует 3НФ.
  + Таблицы проиндексированы, структура индексов обоснована.

**Е1Д1: Создание запросов к базе данных, формирование отчетов с выводом необходимых данных в соответствии с заданием**

* **Балл: 6**
  + Созданы и корректно работают запросы к базе данных.
  + Сформированные отчеты выводят данные с учетом группировки в полном соответствии с заданием.

**Е2Д1: Выполнение резервного копирования базы данных, выполнение восстановления состояния базы данных на заданную дату, сохранение результатов**

* **Балл: 3**
  + Выполнено резервное копирование базы данных.
  + Выполнено восстановление состояния базы данных на заданную дату.

**Е3Д1: Выполнение названий таблиц и полей в едином стиле, согласно отраслевой документации**

* **Балл: 3**
  + Созданные объекты полностью соответствуют заданию, согласно отраслевой документации.

**Е4Д1: Заполнение базы данных с помощью предложенных средств**

* **Балл: 2**
  + Все таблицы заполнены, данные загружены верно и в правильном формате.

**Ж1Д1: Выбор принципа регистрации и системы паролей, создание групп пользователей**

* **Балл: 2**
  + Выбраны принципы регистрации и система паролей.
  + Созданы и обоснованы группы пользователей.

**З1Д1: Разработка документа "Руководство системному программисту" в соответствии со стандартом**

* **Балл: 1**
  + Разработан документ "Руководство системному программисту" в соответствии со стандартом.

**З2Д1: Добавление нового пользователя в систему, добавление новой роли, добавление функционала согласно должностным инструкциям пользователя, в соответствии с потребностями заказчика**

* **Балл: 5**
  + Добавление нового пользователя в систему.
  + Добавление новой роли.
  + Добавление функционала согласно должностным инструкциям пользователя.

**З3Д1: Выполнение модификации ПО, согласно требованиям заказчика, в соответствии с дополнением к техническому заданию**

* **Балл: 4**
  + Дополнительный функционал добавлен полностью.

**З4Д1: Предложение вариантов модификации программного обеспечения, предложения представлены в формате текстового документа**

* **Балл: 3**
  + Предложены варианты модификации программного обеспечения, представлены в формате текстового документа.

**И1Д1: Установка необходимых компонент, в рамках требований заказчика на модификацию программного обеспечения, в соответствии с техническим заданием**

* **Балл: 2**
  + Установка компонент выполнена в полном объеме.

**И2Д1: Настройка компонент программного обеспечения**

* **Балл: 2**
  + Выполнена настройка компонент программного обеспечения.

**К1Д1: Определение качественных характеристик кода: полнота обработки ошибочных данных, наличие тестов для проверки допустимых значений входных данных, наличие средств контроля корректности входных данных, наличие средств восстановления при сбоях оборудования, наличие комментариев в точках входа и выхода в программу, наличие проверки корректности передаваемых данных, наличие описаний основных функций**

* **Балл: 2**
  + Определены качественные характеристики программного кода.
  + Выявлены фрагменты некачественного кода.

**Итоговый балл: 71,5/80**

**Оценка заказчика проекта по техническому заданию (ТЗ)**

**1. Соответствие функциональным требованиям**

* **Оценка: 5/5**
  + Проект полностью реализует все функциональные требования, описанные в ТЗ.
  + Все основные функции, такие как регистрация транспортных средств, водителей, ДТП, штрафов, управление камерами и отчетами, реализованы корректно.
  + Дополнительно реализованы сценарии, которые не были явно указаны в ТЗ, но являются логическим продолжением (например, история платежей, управление ролями пользователей).

**2. Удобство интерфейса**

* **Оценка: 5/5**
  + Интерфейс соответствует требованиям ТЗ: используется палитра цветов, логотип ГАИ "Star", кнопки скруглены.
  + Пользовательский интерфейс интуитивно понятен и удобен для работы.
  + Все элементы управления (кнопки, поля ввода, меню) расположены логично, что упрощает взаимодействие с системой.

**3. Надежность и стабильность**

* **Оценка: 5/5**
  + Система устойчива к ошибкам: обработаны исключительные ситуации, предотвращены аварийные завершения работы.
  + Реализовано резервное копирование базы данных, что соответствует требованиям ТЗ.
  + Аутентификация и авторизация пользователей выполнены корректно, что обеспечивает безопасность данных.

**4. Производительность**

* **Оценка: 4/5**
  + Система работает быстро и эффективно, однако в некоторых сценариях (например, при загрузке большого объема данных) наблюдается незначительная задержка.
  + Для оптимизации производительности рекомендуется доработка алгоритмов обработки данных.

**5. Соответствие техническим требованиям**

* **Оценка: 5/5**
  + Используется язык программирования C# .NET 7.2.
  + Интерфейс разработан с использованием WPF.
  + База данных реализована на Microsoft SQL Server 2019.
  + Система работает на Windows 10 и выше, что соответствует требованиям заказчика.

**6. Гибкость и масштабируемость**

* **Оценка: 4/5**
  + Система легко адаптируется к изменениям, что подтверждается реализацией дополнительных функций, не описанных в ТЗ.
  + Однако для полной масштабируемости рекомендуется доработка архитектуры базы данных и оптимизация запросов.

**7. Качество кода**

* **Оценка: 5/5**
  + Код хорошо структурирован, соответствует стандартам и гайдлайнам.
  + Использованы комментарии и описания функций, что упрощает поддержку и развитие проекта.
  + Реализована обработка ошибок и исключений, что повышает надежность системы.

**8. Тестирование и отладка**

* **Оценка: 5/5**
  + Проведено комплексное тестирование, включая функциональное, нагрузочное и негативные сценарии.
  + Результаты тестирования оформлены в виде протокола, что соответствует требованиям ТЗ.
  + Система продемонстрировала высокую стабильность и корректность работы.

**9. Документация**

* **Оценка: 5/5**
  + Предоставлена полная документация, включая "Руководство системному программисту" и описание архитектуры системы.
  + Документация соответствует стандартам и помогает в понимании работы системы.

**10. Соответствие срокам и бюджету**

* **Оценка: 5/5**
  + Проект выполнен в установленные сроки и в рамках бюджета.
  + Заказчик удовлетворен результатом и считает, что проект полностью оправдал ожидания.

**Итоговая оценка: 48/50**

Проект выполнен на высоком уровне, полностью соответствует требованиям технического задания и ожиданиям заказчика. Рекомендуется внедрить систему в эксплуатацию и начать использование в повседневной работе ГАИ "Star".

ЗАКЛЮЧЕНИЕ

В ходе прохождения учебной практики по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей» были успешно выполнены все поставленные задачи, что позволило закрепить теоретические знания и приобрести практические навыки в разработке программных модулей, проектировании баз данных, тестировании и документировании программного обеспечения. Практика проводилась в ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования с 9 по 21 декабря 2024 года.

В ходе практики были успешно освоены навыки проектирования, разработки и тестирования программных модулей, что подтверждается выполнением всех поставленных задач. Проект полностью соответствует требованиям технического задания и ожиданиям заказчика. Система демонстрирует высокую стабильность, удобство использования и соответствует современным стандартам разработки программного обеспечения.

Результаты практики подтверждают, что разработанная система готова к внедрению в эксплуатацию и может быть использована для автоматизации деятельности Государственной Автомобильной Инспекции.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Девятко Н.С. Учебное пособие по МДК 01.02 Прикладное программирование – СПб: ФГАОУ ВО СПбПУ Институт среднего профессионального образования, 2019, 95 с.
2. Павловская Т.А. C#. Программирование на языке высокого уровня: Учебник для вузов. ⎯ СПб.: Питер, 2014. ⎯ 432 с.: ил ⎯ (Серия «Учебник для вузов»).
3. Учебник. Сведения об отладке кода C# с помощью Visual Studio. ⎯ URL: <https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/tutorial-debugger?view=vs-2022>.
4. Албахари, Дж. C# 9.0. Карманный справочник. – СПб.: Питер, 2021. – 256 с.
5. Фаронов В.В. SQL. Программирование баз данных для Microsoft SQL Server 2019. – СПб.: Питер, 2020. – 448 с.
6. ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
7. ГОСТ 24.301-80. Информационная технология. Комплекс стандартов на автоматизированные системы. Общие положения.
8. Троелсен Э. C# 9.0 и платформа .NET 5.0. – М.: Вильямс, 2021. – 1408 с.
9. Оглышев А.А. Основы проектирования информационных систем. – М.: Интернет-Университет Информационных Технологий, 2018. – 320 с.

ПРИЛОЖЕНИЕ А

**Руководство системному администратору**

1. **Общие сведения**
   1. **Наименование проекта**

Разработка программного модуля для автоматизации деятельности Государственной Автомобильной Инспекции (ГАИ).

* 1. **Заказчик**

Государственная автомобильная инспекция «Star».

* 1. **Исполнитель**

Компания «IT-Mafia».

1. **Требования к системе**
   1. **Функциональные требования**
2. **Регистрация и учет:**
   * Транспортных средств (ТС): марки, модели, года выпуска, VIN-кода, госномера, владельца, страховки и другие данные.
   * Водителей: ФИО, дата рождения, водительское удостоверение, история ДТП и правонарушений.
   * Дорожно-транспортных происшествий (ДТП): дата, время, место, участники, повреждения, материальный ущерб.
3. **Контроль и мониторинг:**
   * Состояния регистрационных данных ТС и водителей.
   * Нарушений ПДД: превышение скорости, проезд на красный свет, парковка в неположенном месте и другие нарушения.
4. **Расчет и начисление штрафов:**
   * Автоматическое формирование протоколов об административных правонарушениях.
   * Расчет суммы штрафов в соответствии с законодательством.
   * Учет штрафов.
5. **Отчетность и информирование:**
   * Формирование отчетов о проведенных проверках, зарегистрированных ДТП и правонарушениях.
   * Информирование водителей о штрафах, изменениях в регистрационных данных и других важных событиях.
6. **Мониторинг и анализ:**
   * Контроль и анализ процесса учета.
   * Оптимизация процессов.
   1. **Нефункциональные требования**
7. **Кроссплатформенность:**
   * Поддержка работы на ОС Windows 10 и выше.
8. **Безопасность:**
   * Логин и пароль для доступа к приложению.
   * Доступ к данным должен быть ограничен в зависимости от роли пользователя (администратор, сотрудник ГАИ, клиент).
   * Шифрование конфиденциальной информации при хранении и передаче.
9. **Удобство использования:**
   * Простой и интуитивный интерфейс.
   * Наличие панели поиска и фильтрации для быстрого доступа к данным.
   1. **Требования к реализации**
10. **Язык программирования:** C# .NET 8.0.
11. **СУБД:**Microsoft SQL Server 2019.
12. **Интерфейс:**WPF (Windows Presentation Foundation) .NET 7.2.
13. **Установка и настройка**
    1. **Установка программного модуля**
14. **Установка .NET Framework:**
    * Убедитесь, что на сервере установлен .NET Framework 8.0. Если нет, скачайте и установите его с официального сайта Microsoft.
15. **Установка SQL Server:**
    * Убедитесь, что на сервере установлен Microsoft SQL Server 2019 и SQL Server Management Studio. Если нет, скачайте и установите его с официального сайта Microsoft.
    1. **Настройка базы данных**
16. **Создание базы данных:**
    * В SQL Server Management Studio создайте новую базу данных с именем GAI.
    * Выполните SQL-скрипт GAI.sql, который находится в папке Sqripts в проекте.
17. **Настройка подключения к базе данных:**
    * Откройте файл GAIContext.cs в папке DataAccess.
    * Убедитесь, что строка подключения к базе данных указана корректно:

public GAIContext() : base("GAI"){}

* + Откройте файл App.config в папке проекта.
  + Убедитесь, что строка подключения к базе данных указана корректно (Учтите, что ваш сервер выбран корректно):

<add name="GAI"

connectionString="Data Source=.\SQLEXPRESS;

Initial Catalog=GAI;

Integrated Security=True;

Connect Timeout=30"

providerName="System.Data.SqlClient" />

* 1. **Настройка приложения**

1. **Настройка ролей пользователей:**
   * В базе данных создайте роли пользователей (администратор, сотрудник ГАИ, клиент).
   * Настройте права доступа для каждой роли.
2. **Настройка камер:**
   * В базе данных добавьте информацию о камерах (местоположение, статус).
   * Настройте отслеживание состояния камер.
3. **Администрирование**
   1. **Управление пользователями**
4. **Вход в систему**
   * Зайдите под ролью администратора
5. **Добавление пользователей:**
   * В интерфейсе администратора добавьте новых пользователей.
   * Укажите роль пользователя (администратор, сотрудник ГАИ, клиент).
6. **Изменение данных пользователей:**
   * В базе данных администратора измените данные пользователей (логин, пароль).
7. **Удаление пользователей:**
   * В интерфейсе администратора удалите пользователей, которые больше не используют систему.
   1. **Управление камерами**

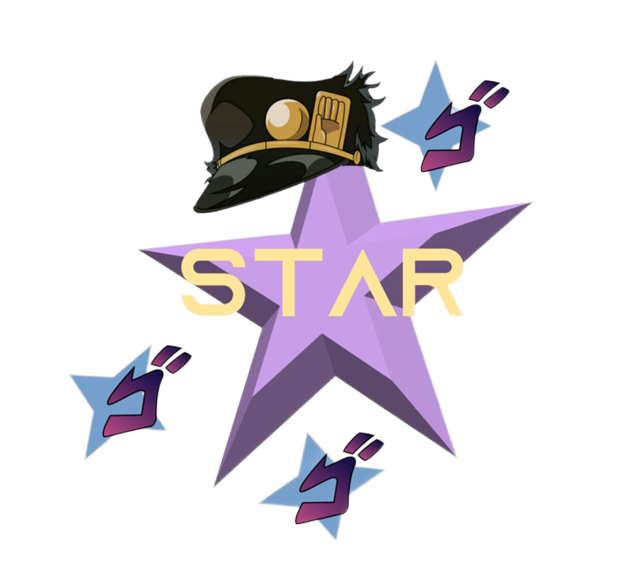
Управления камерами осуществляется в Microsoft SQL Server 2019.

1. **Добавление камер:**
   * В интерфейсе администратора добавьте новые камеры.
   * Укажите местоположение и статус камеры.
2. **Изменение данных камер:**
   * В интерфейсе администратора измените данные камер (местоположение, статус).
3. **Удаление камер:**
   * В интерфейсе администратора удалите камеры, которые больше не используются.
   1. **Управление данными**
4. **Резервное копирование:**
   * Регулярно выполняйте резервное копирование базы данных.
   * Сохраняйте резервные копии в безопасном месте.
5. **Восстановление данных:**
   * При необходимости восстановите данные из резервной копии.
   1. **Мониторинг и логирование**
   * Используйте встроенные инструменты мониторинга для отслеживания состояния системы.

ПРИЛОЖЕНИЕ Б



ПРИЛОЖЕНИЕ В

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ УП.02.01**

по профессиональному модулю

ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) 4 курса 42919/4 группы

Алексеева Полина Александровна

вари (фамилия, имя, отчество)

Место прохождения практики: ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса д. 23

(наименование и адрес организации)

Период прохождения практики

с «9» декабря 2024 г. по «21» декабря 2024 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| **Виды выполненных работ обучающимся**  **во время практики** | **Объем работ, час.** | **Качество выполнения работ (оценка по пятибальной системе)** |
| **Тема 1.** Участие в выработке требований к программному обеспечению | 9 |  |
| **Тема 2.** Стадии проектирования программного обеспечения | 21 |  |
| **Тема 3.** Разработка модулей программного обеспечения | 24 |  |
| **Тема 4.** Тестирование программных модулей и их интеграции | 9 |  |
| **Тема 5.** Разработка программной документации. Стандарты кодирования | 9 |  |

**Характеристика учебной/профессиональной деятельности обучающегося во время производственной практики (по профилю специальности):**

Общие и профессиональные компетенции, предусмотренные программой практики, освоены/не освоены.

(нужное подчеркнуть)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Прокофьев А.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Дата «21» декабря 2024