**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

Лисин Станислав Александрович

(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

Лисин Станислав Александрович

(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 30**

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

(подпись)  (Ф.И.О.)

Задание получил «9» декабря 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_ Лисин С.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

Лисин Станислав Александрович

(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

[ВВЕДЕНИЕ 8](#_Toc185498796)

[Задание №1 «Разработка технического задания» 9](#_Toc185498797)

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

[Задание №3 «Разработка макетов» 25](#_Toc185498813)

[Задание №4 «Разработка программы» 32](#_Toc185498818)

[Задание №5 «Тестирование» 37](#_Toc185498824)

[Задание №6 «GitHub.Оценка проекта» 51](#_Toc185498828)

[ЗАКЛЮЧЕНИЕ 62](#_Toc185498834)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 63](#_Toc185498835)

[ПРИЛОЖЕНИЕ 1 64](#_Toc185498836)

ВВЕДЕНИЕ

Учебная практика проходила в ФГАОУ ВО СПбПУ Институт среднего профессионального образования в период с 09.12.24 по 21.12.24. Этот этап является важным компонентом подготовки квалифицированных специалистов, так как позволяет закрепить теоретические знания на практике и получить опыт работы в реальных условиях.

Учебная практика представляет собой вид учебно-вспомогательного процесса, направленного на развитие практических навыков и умений, необходимых для успешной профессиональной деятельности. В ходе практики студенты приобретают опыт работы с современными технологиями и инструментами, что способствует их профессиональному росту и адаптации к требованиям рынка труда.

Целью практики являлось закрепление практических умений и получение дополнительных навыков в разработке программных модулей программного обеспечения для компьютерных систем. Для достижения поставленной цели был выполнен ряд задач, включая разработку спецификаций отдельных компонент, конструирование пользовательского интерфейса, проектирование и разработку модулей программного обеспечения, модульное тестирование, а также документирование программных продуктов.

В ходе практики была работа над реальными задачами, что позволило им приобрести навыки, необходимые для работы в сфере информационных технологий. Результаты практики отражают 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**

# **Описание предметной области.**

Цель данного модуля является автоматизация процесса учета и обработки заявок на покупку/аренду/обмен недвижимости. Модуль позволяет сотрудникам агентства недвижимости контролировать процесс на всех стадиях до получения клиентом собственности.

Разработка программного модуля для учета заявок на покупку недвижимости — это процесс создания программного продукта, который позволяет агентству недвижимости контролировать процесс на всех стадиях до получения клиентом собственности.

Задача разработки такого программного модуля состоит в создании удобного и эффективного инструмента для проведения сделки, рекомендаций по более предпочтительным объектам в соответствии с пожеланиями клиента, учета и контроля состояния сделки, что упрощает подход к работе и позволяет минимизировать затраты времени на консультацию.

Программный модуль позволяет агентству недвижимости упростить подход к работе сотрудников, оптимизировать использование ресурсов и повысить удовлетворенность клиентов.

# **Основные функции и возможности модуля.**

1. Консультация: клиент хочет купить/арендовать/обменять недвижимость по его потребностям. Агент подбирает для покупателя предварительные варианты недвижимости и открывает статус сделки. Сотрудник заполняет данные о клиенте: ФИО, Номер телефона, email, критерии, квартиры, в которых заинтересован покупатель. Статус сделки: “открыта”.
2. Учет проведения сделки: после окончательного выбора агент занимается оформлением и предоставлением всего пакета документов. В случае покупки решается каким способом будем производиться оплата. В данном случае покупатель может взять ипотеку, тогда дополнительно надо заключать договор с банком. Статус сделки: “в процессе”.
3. Заключение договора и закрытие сделки: процесс, включающий работу юристов на проверку “чистоты” документов, передача подготовленных документов агенту и подписание окончательного договора для завершения сделки. Статус сделки меняется на “закрыта”. При срыве сделки она также будет иметь статус “закрыта”.
4. Внесение данных о новых объектах недвижимости: процесс, в котором производится заполнение новыми данными о недвижимости, что появилась недавно на рынке.
5. Согласование документов: юристы заполняют данные о каждом договоре для более быстрого доступа, а также, по запросу от агента, создают полный или частичный пакет документов для оформления, заключения сделки. В частных случаях может произвести изменения в договоре.

# **Общие сведения.**

1. Наименование проекта: Разработка программного модуля для учета и обработки заявок на покупку/аренду/обмен недвижимости.
2. Заказчик: Агентство недвижимости “КвартПлюс”.
3. Исполнитель: ООО “СофтФордж Решения”.

# **Функциональные требования.**

1. Возможность добавления клиентов в базу данных с указанием следующих параметров:

* Id клиента;
* ФИО клиента;
* Дата консультации;
* Номер телефона;
* Email;
* Вид операции (покупка, аренда, обмен);
* Критерии выбора недвижимости (район, площадь, цена, тип недвижимости и т.д.).

1. Возможность добавления объектов недвижимости в базу данных с указанием следующих параметров:

* Id объекта;
* Адрес;
* Цена;
* Тип недвижимости (Дом, Квартира);
* Площадь объекта;
* Статус объекта;
* Описание.

1. Возможность редактирования сделок:

* Изменение этапа выполнения (открыта, в процессе, закрыта);
* Обновление данных клиента и его предпочтений;
* Уточнение текущего состояния сделки.

1. Возможность добавления агентов в базу данных с указанием следующих параметров:

* Id агента;
* ФИО агента;
* Номер телефона;
* Email;
* Дата приема на работу.

1. Учет состояния сделок:

* Отображение списка сделок;
* Возможность добавления комментариев агентом;
* Получение уведомлений о смене статуса сделки.

1. Подбор недвижимости:

* Поиск недвижимости в базе данных по критериям клиента;
* Сохранение выбранных вариантов в рамках сделки.

1. Работа с документами:

* Согласование документов юристами;
* Генерация полного или частичного пакета документов по запросу;
* Хранение и управление договорами.

# **Нефункциональные требования.**

1. Кроссплатформенность:

* Поддержка работы на ОС семейства Windows.

1. Безопасность:

* Авторизация через логин и пароль;
* Ролевой доступ (агент, юрист, администратор).

1. Удобство использования:

* Интерфейс должен содержать кнопки, которые сразу дают понять их назначение;
* Система уведомлений о действиях и изменениях;

1. Производительность:

* Время отклика на запросы пользователя не более 2 секунд;
* Мгновенный поиск сделок и объектов недвижимости с задержкой не более 1 секунды.

# **Требования к реализации.**

1. Язык программирования: Python.
2. СУБД: PostgreSQL.

# **Требования к документации.**

1. Техническое задание на разработку программного модуля.
2. Руководство системному программисту.

# **Руководство по стилю.**

1. Шрифт: Inter 12.
2. Цветовая схема:

* Основной цвет - #404041;
* Дополнительный цвет 1 - #F6EDE3;
* Дополнительный цвет 2 - #E6C17A.

# **Этапы и сроки разработки**

1. Этапы выполнения работ:

* Анализ требований и проектирование – 4 дня;
* Разработка модуля – 7 дней;
* Тестирование и отладка – 1 день;
* Внедрение – 1 день.

1. Общий срок реализации: 2 недели с момента утверждения ТЗ.

**Дата составления**: 10.12.2024  
**Подпись**: Прокофьев А.А.

**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

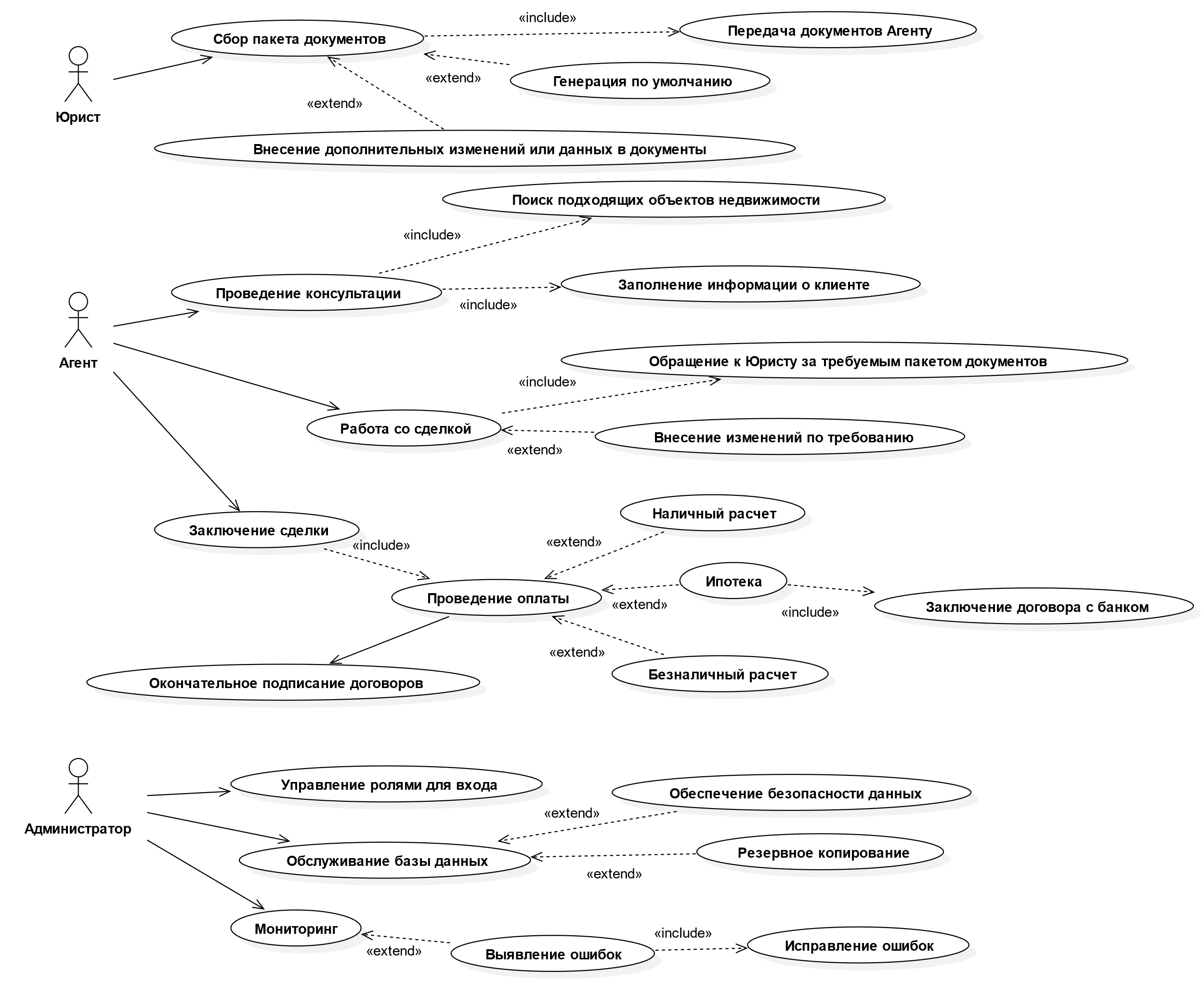


Рисунок 1 – Диаграмма вариантов использования

**Сценарий:** Взаимодействие Агента с программой

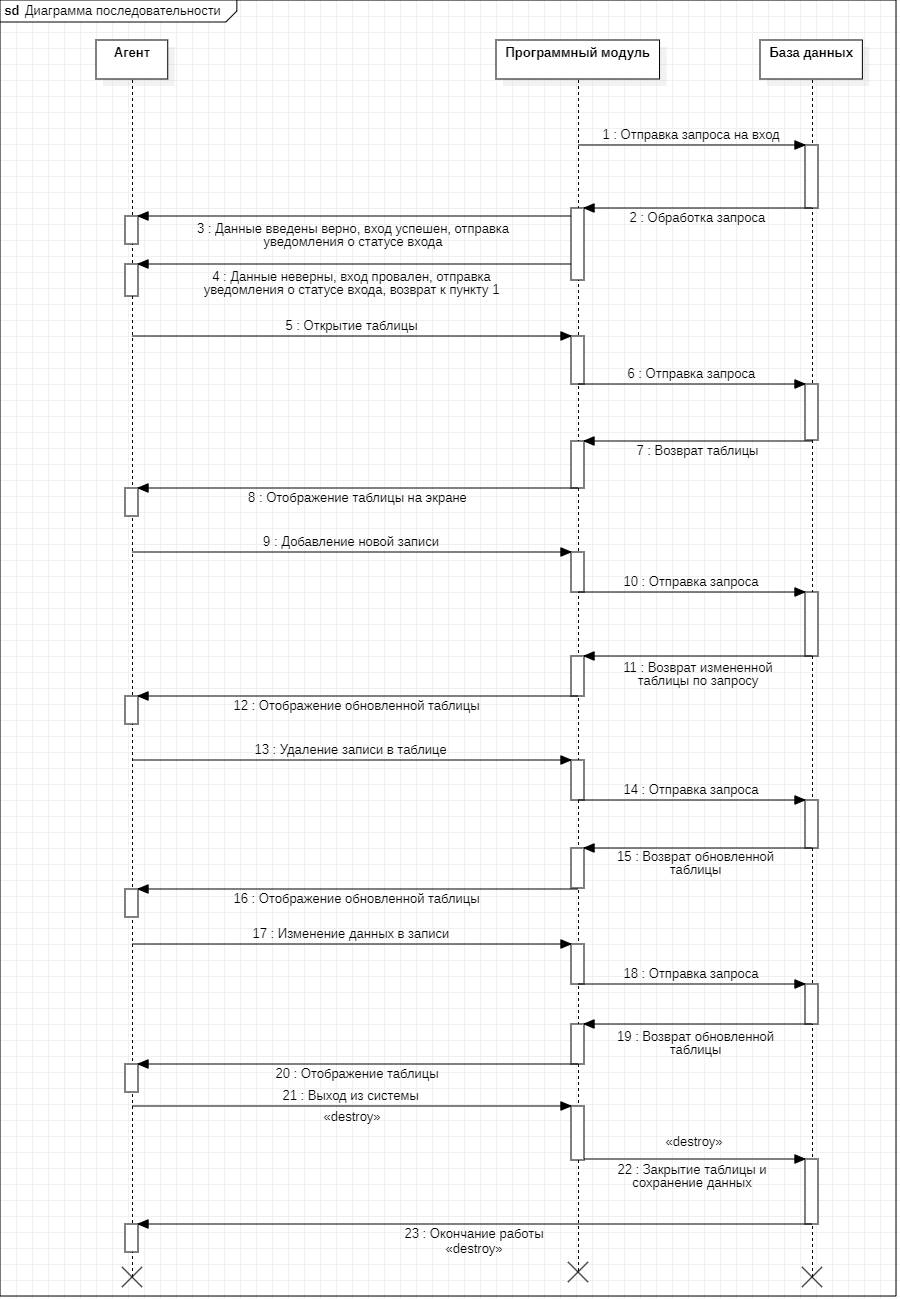


Рисунок 2 – Диаграмма последовательности

**Сценарий**: Процесс проведения сделки

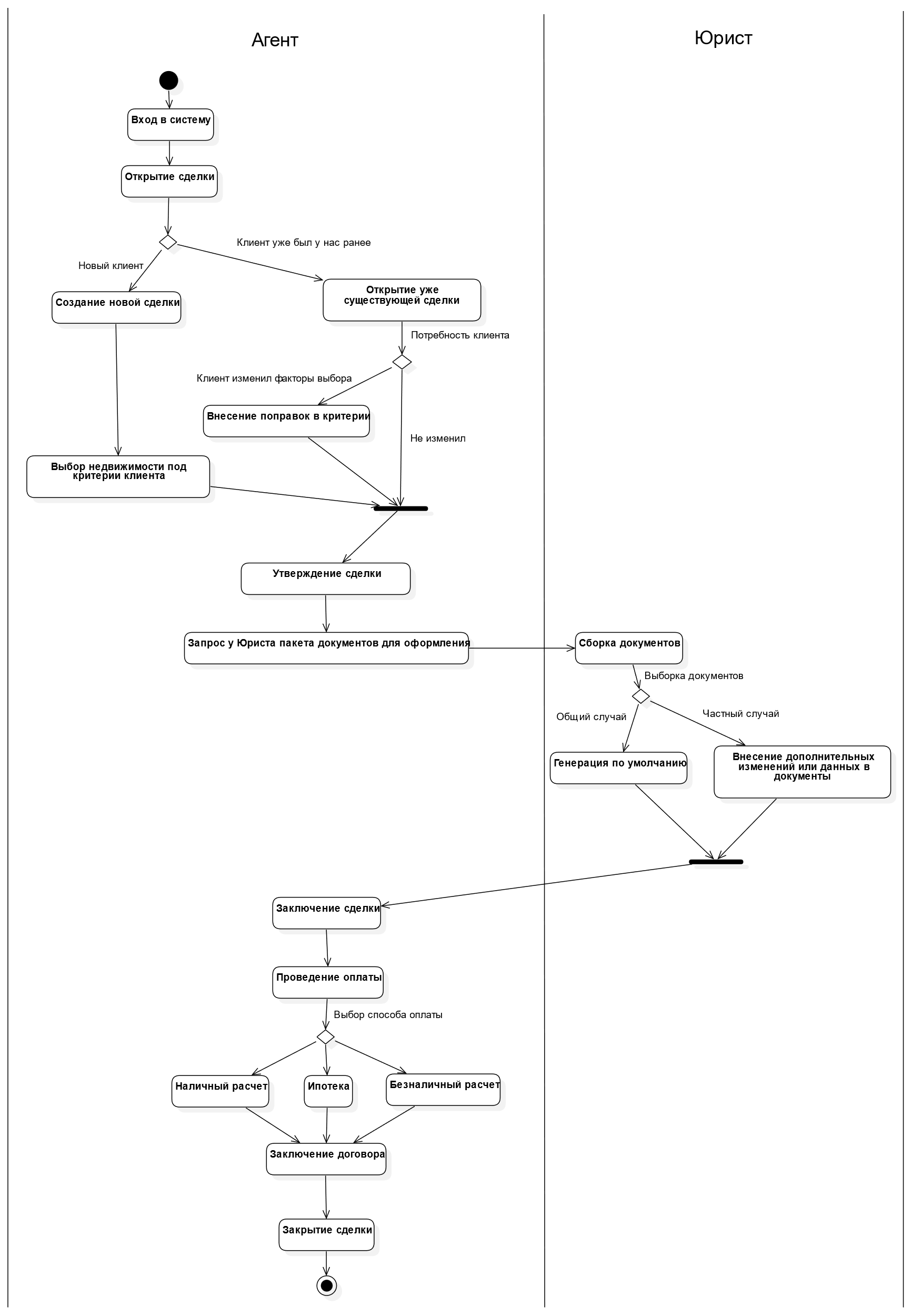


Рисунок 3 – Диаграмма активности

### **ER-диаграмма базы данных**

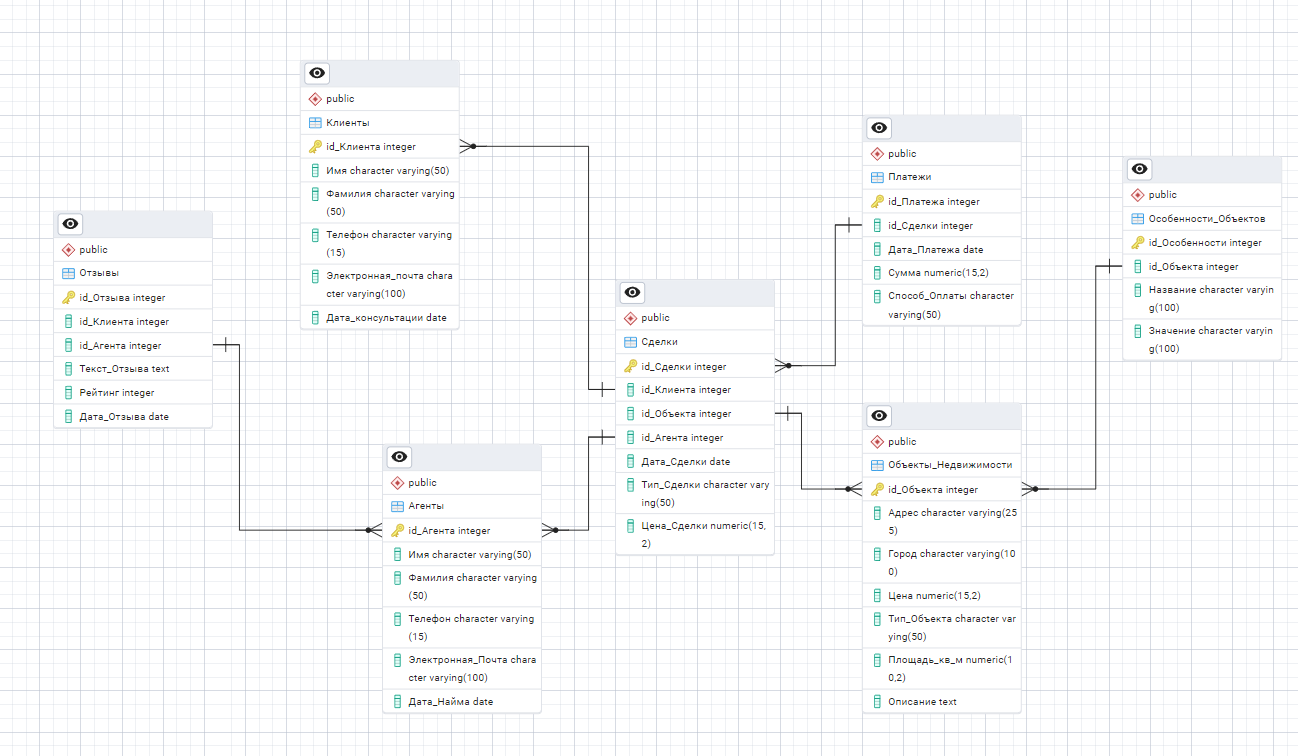


Рисунок 4 – ER-диаграмма базы данных

### **Словарь данных**

# **Таблица 1: Клиенты**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Клиента | SERIAL | Уникальный id клиента. | PRIMARY KEY |
| Имя | VARCHAR(50) | Имя клиента. | NOT NULL |
| Фамилия | VARCHAR(50) | Фамилия клиента. | NOT NULL |
| Телефон | VARCHAR(15) | Номер телефона клиента. | - |
| Электронная\_Почта | VARCHAR(100) | Адрес электронной почты клиента. | - |
| Дата\_Регистрации | DATE | Дата регистрации клиента в системе. | - |

# **Таблица 2: Объекты\_Недвижимости**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Объекта | SERIAL | Уникальный id объекта недвижимости. | PRIMARY KEY |
| Адрес | VARCHAR(255) | Адрес объекта недвижимости. | NOT NULL |
| Город | VARCHAR(100) | Город, в котором находится объект недвижимости. | NOT NULL |
| Цена | DECIMAL(15, 2) | Цена объекта недвижимости. | NOT NULL |
| Тип\_Объекта | VARCHAR(50) | Тип объекта (Квартира, Дом). | NOT NULL |
| Площадь\_кв\_м | DECIMAL(10, 2) | Площадь объекта в квадратных метрах. | NOT NULL |
| Описание | TEXT | Описание объекта недвижимости. | - |

# **Таблица 3: Агенты**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Агента | SERIAL | Уникальный id агента. | PRIMARY KEY |
| Имя | VARCHAR(50) | Имя агента. | NOT NULL |
| Фамилия | VARCHAR(50) | Фамилия агента. | NOT NULL |
| Телефон | VARCHAR(15) | Номер телефона агента. | - |
| Электронная\_Почта | VARCHAR(100) | Адрес электронной почты агента. | - |
| Дата\_Найма | DATE | Дата найма агента в агентство. | - |

# **Таблица 4: Сделки**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Сделки | SERIAL | Уникальный id сделки. | PRIMARY KEY |
| Id\_Клиента | INT | Id клиента, участвующего в сделке. | FOREIGN KEY (Клиенты.Id\_Клиента) |
| Id\_Объекта | INT | Id объекта недвижимости, участвующего в сделке. | FOREIGN KEY (Объекты\_Недвижимости.Id\_Объекта) |
| Id\_Агента | INT | Id агента, участвующего в сделке. | FOREIGN KEY (Агенты.Id\_Агента) |
| Дата\_Сделки | DATE | Дата совершения сделки. | - |
| Тип\_Сделки | VARCHAR(50) | Тип сделки (Продажа, Аренда). | NOT NULL |
| Цена\_Сделки | DECIMAL(15, 2) | Цена сделки. | NOT NULL |

# **Таблица 5: Особенности\_Объектов**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Особенности | SERIAL | Уникальный id особенности объекта. | PRIMARY KEY |
| Id\_Объекта | INT | Id объекта недвижимости, к которому относится особенность. | FOREIGN KEY (Объекты\_Недвижимости.Id\_Объекта) |
| Название | VARCHAR(100) | Название особенности (например, Бассейн, Гараж). | NOT NULL |
| Значение | VARCHAR(100) | Значение особенности (например, "Есть" или "Нет"). | - |

# **Таблица 6: Платежи**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Платежа | SERIAL | Уникальный id платежа. | PRIMARY KEY |
| Id\_Сделки | INT | Id сделки, к которой относится платеж. | FOREIGN KEY (Сделки.Id\_Сделки) |
| Дата\_Платежа | DATE | Дата совершения платежа. | - |
| Сумма | DECIMAL(15, 2) | Сумма платежа. | NOT NULL |
| Способ\_Оплаты | VARCHAR(50) | Способ оплаты (Наличные, Карта, Ипотечный кредит). | NOT NULL |

# **Таблица 7: Отзывы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** | **Ограничения** |
| Id\_Отзыва | SERIAL | Уникальный id отзыва. | PRIMARY KEY |
| Id\_Клиента | INT | Id клиента, оставившего отзыв. | FOREIGN KEY (Клиенты.Id\_Клиента) |
| Id\_Агента | INT | Id агента, о котором оставлен отзыв. | FOREIGN KEY (Агенты.Id\_Агента) |
| Текст\_Отзыва | TEXT | Текст отзыва. | - |
| Рейтинг | INT | Рейтинг отзыва (от 1 до 5). | CHECK (Рейтинг BETWEEN 1 AND 5) |
| Дата\_Отзыва | DATE | Дата оставления отзыва. | - |

**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).

### **Wireframe-макеты**

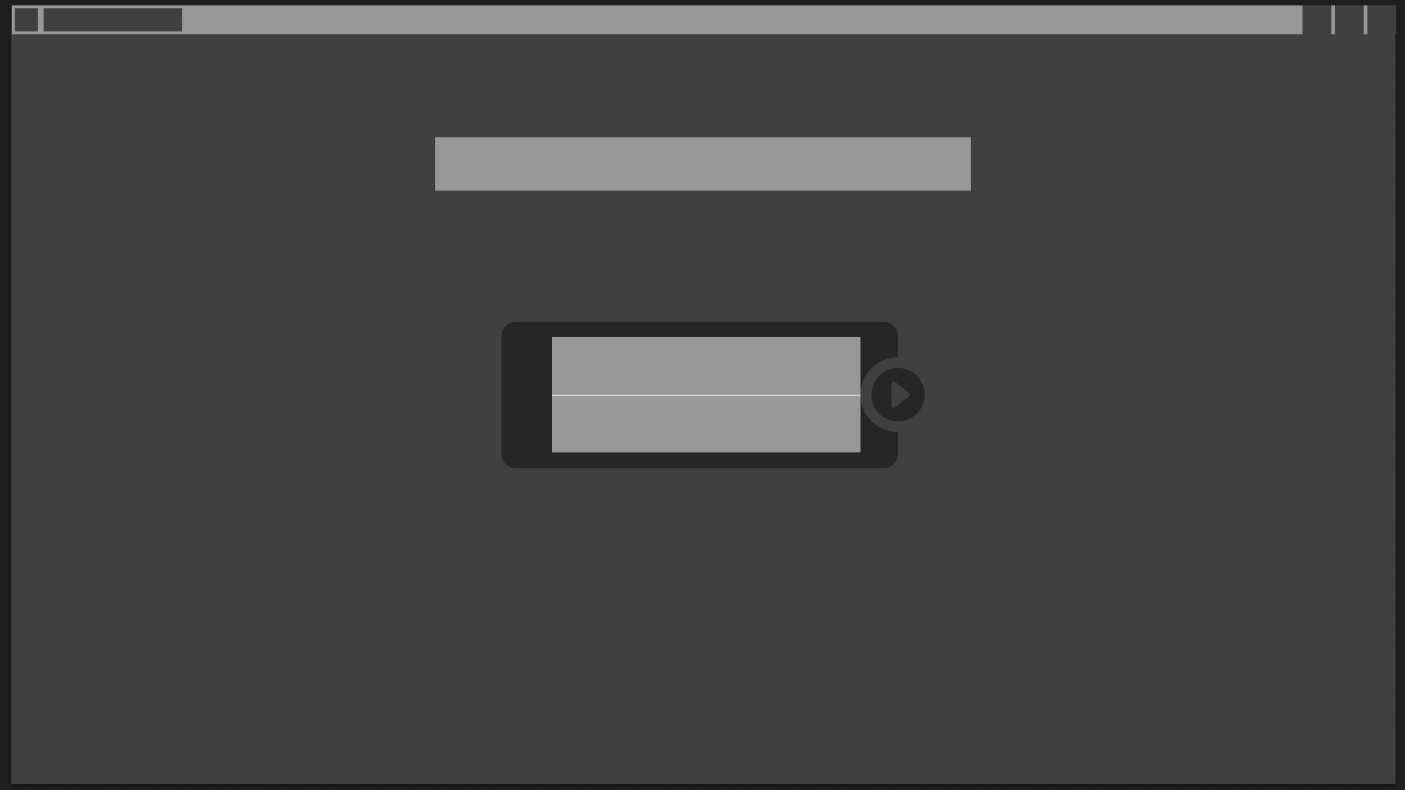
****

Рисунок 5 – Wareframe-макет окна авторизации

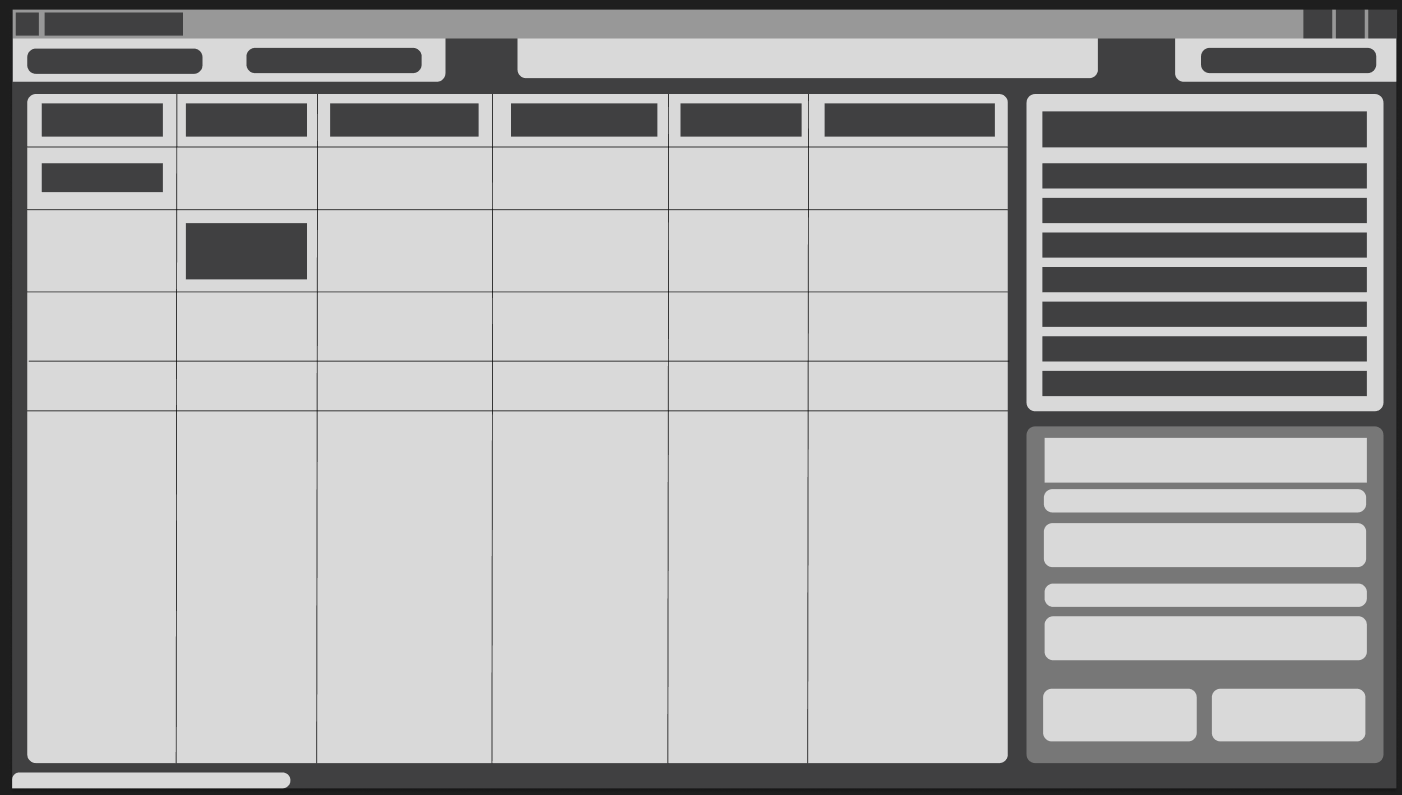
****

Рисунок 6 – Wareframe-макет главного окна



Рисунок 7 – Wareframe-макет окна вывода ошибки

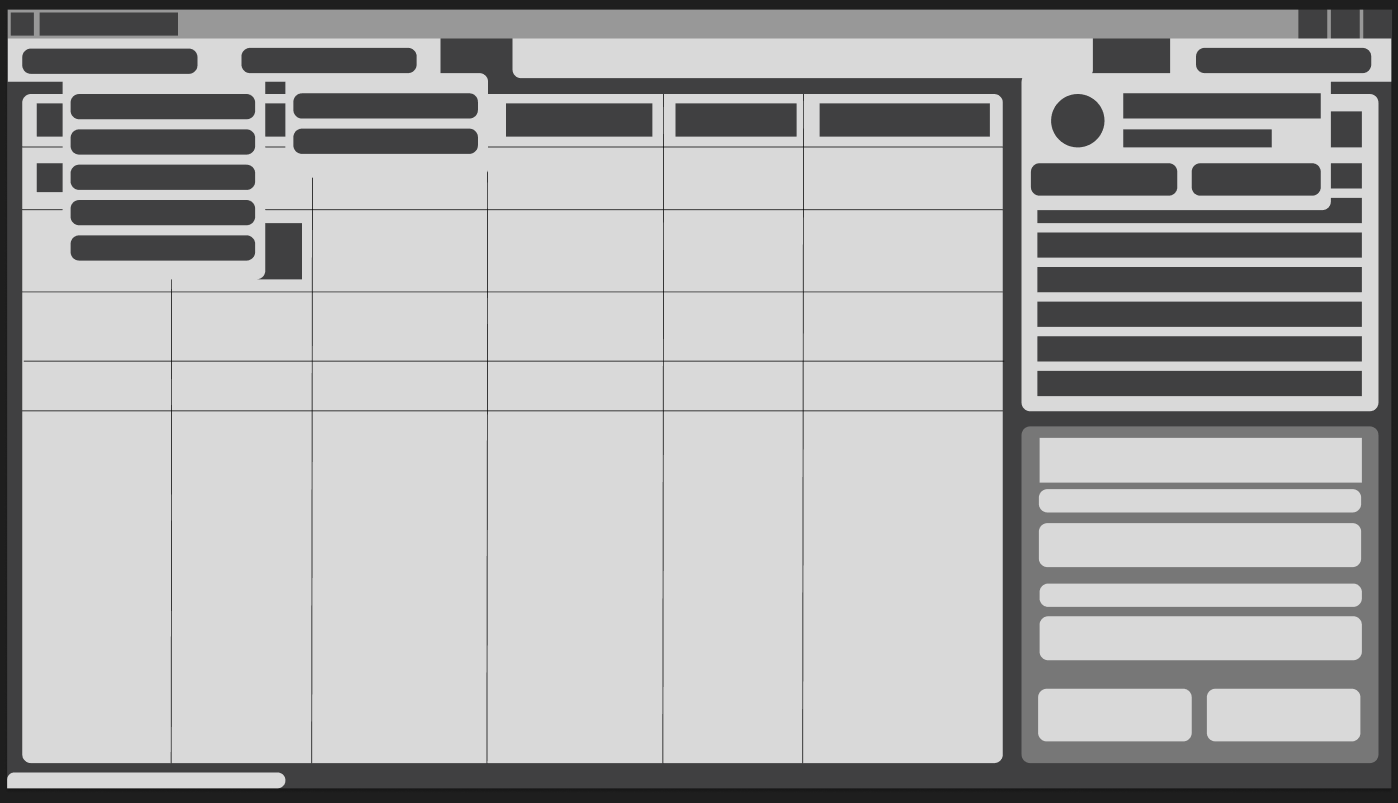


Рисунок 8 – Wareframe-макет главного окна при раскрытии меню

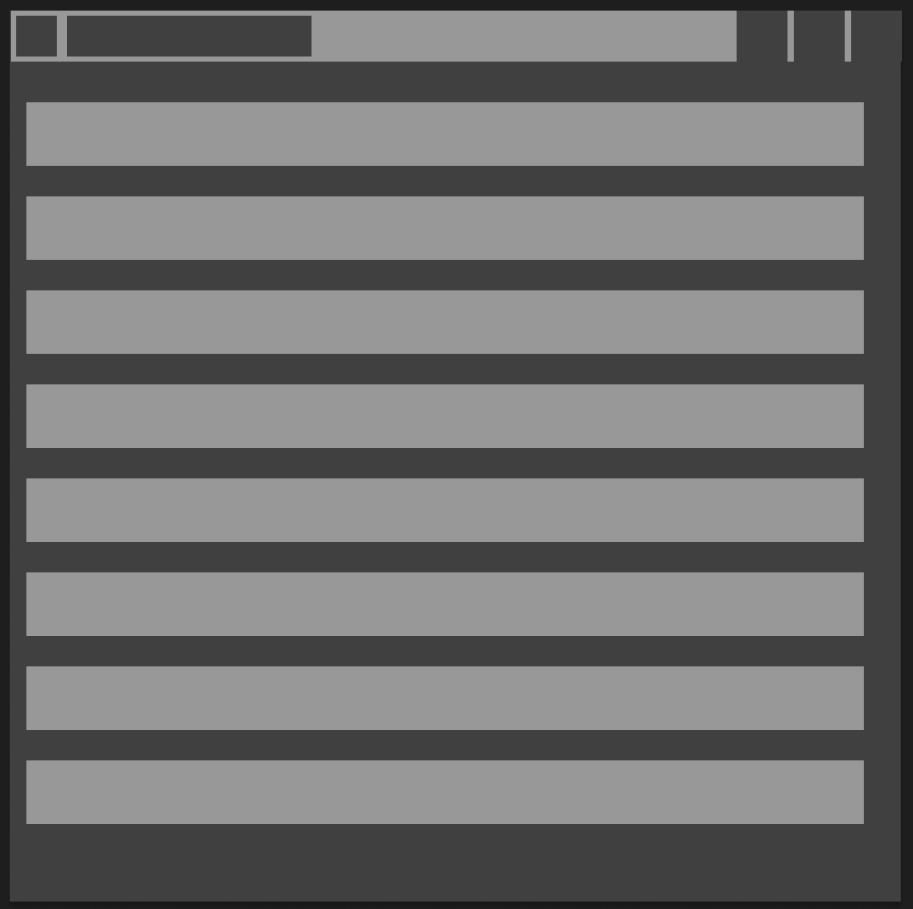


Рисунок 9 – Wareframe-макет окна вывода «О программе»

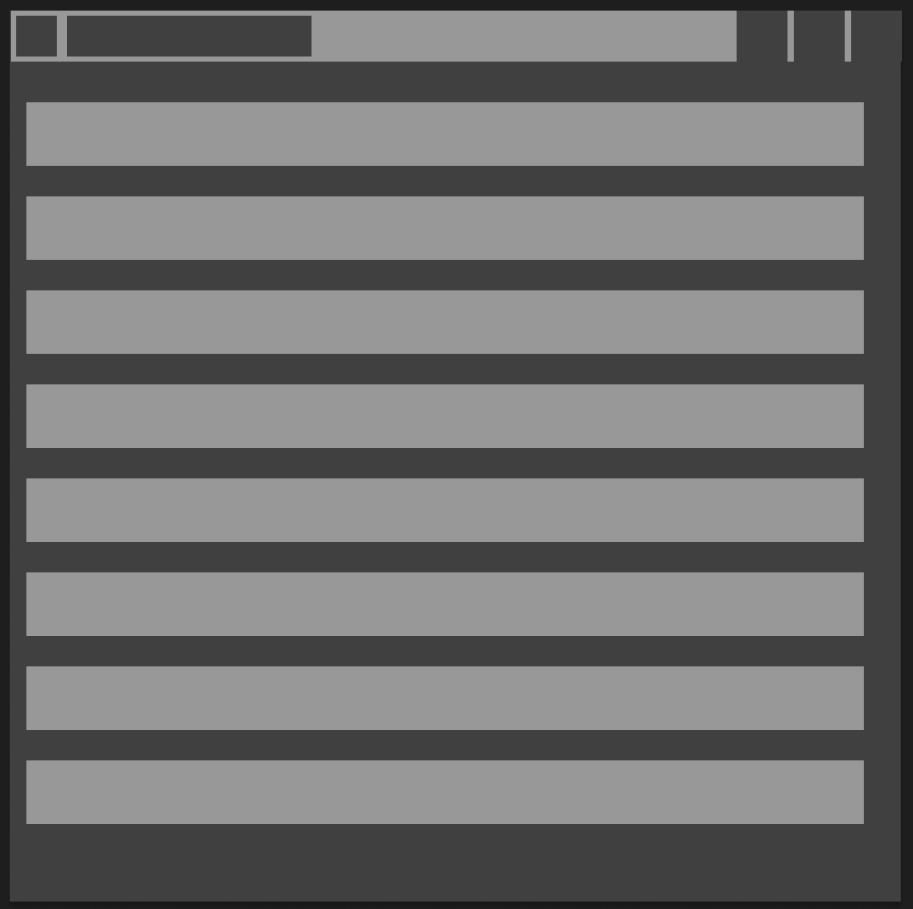


Рисунок 10 – Wareframe-макет окна вывода «FAQ»

### **Mockup-макеты**

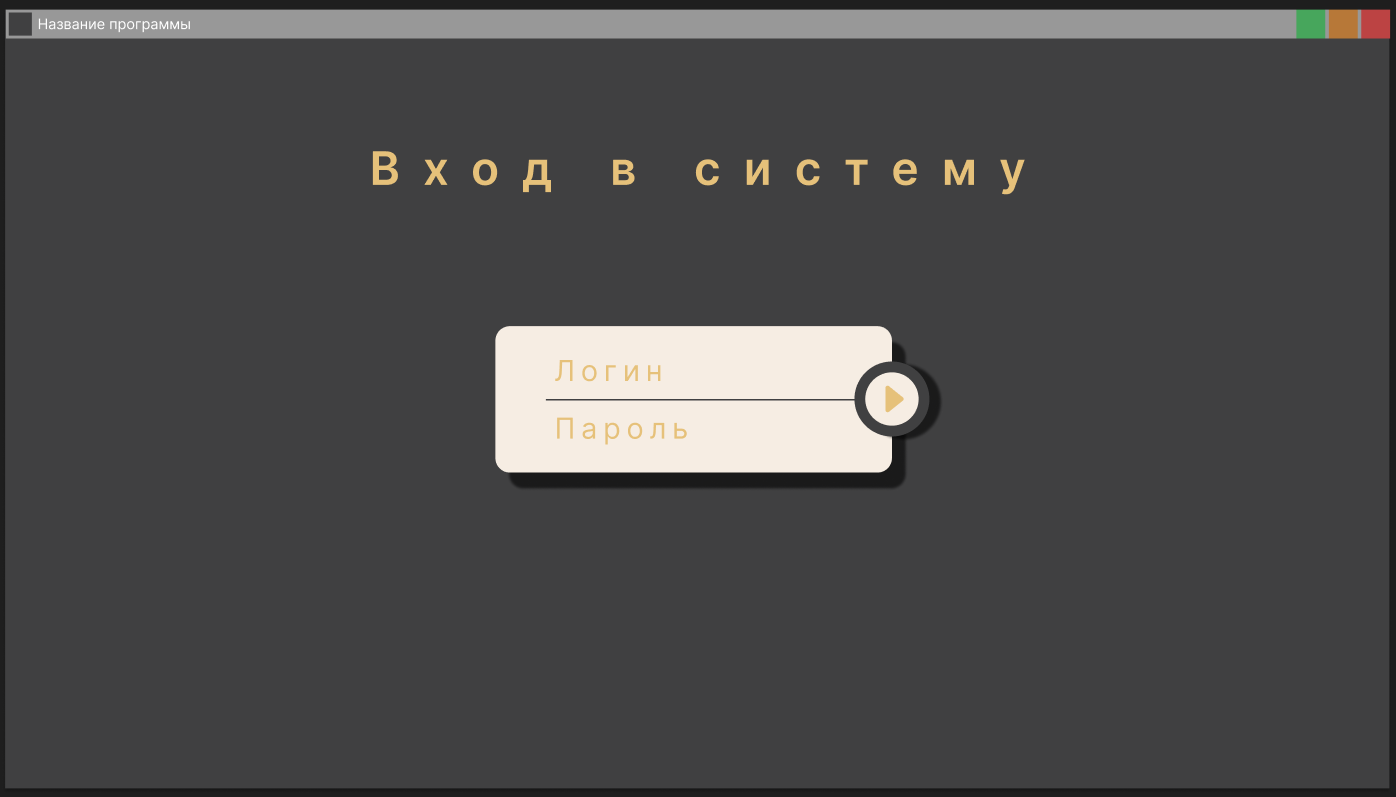
****

Рисунок 11 – Mockup-макет главного окна приложения



Рисунок 12 – Mockup-макет окна авторизации

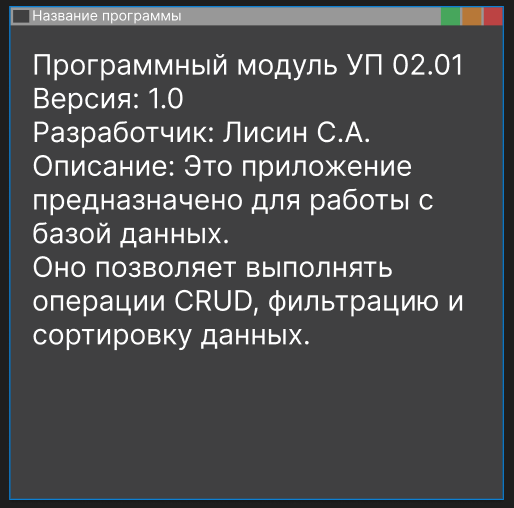


Рисунок 13 – Mockup-макет главного окна роли «Водитель»

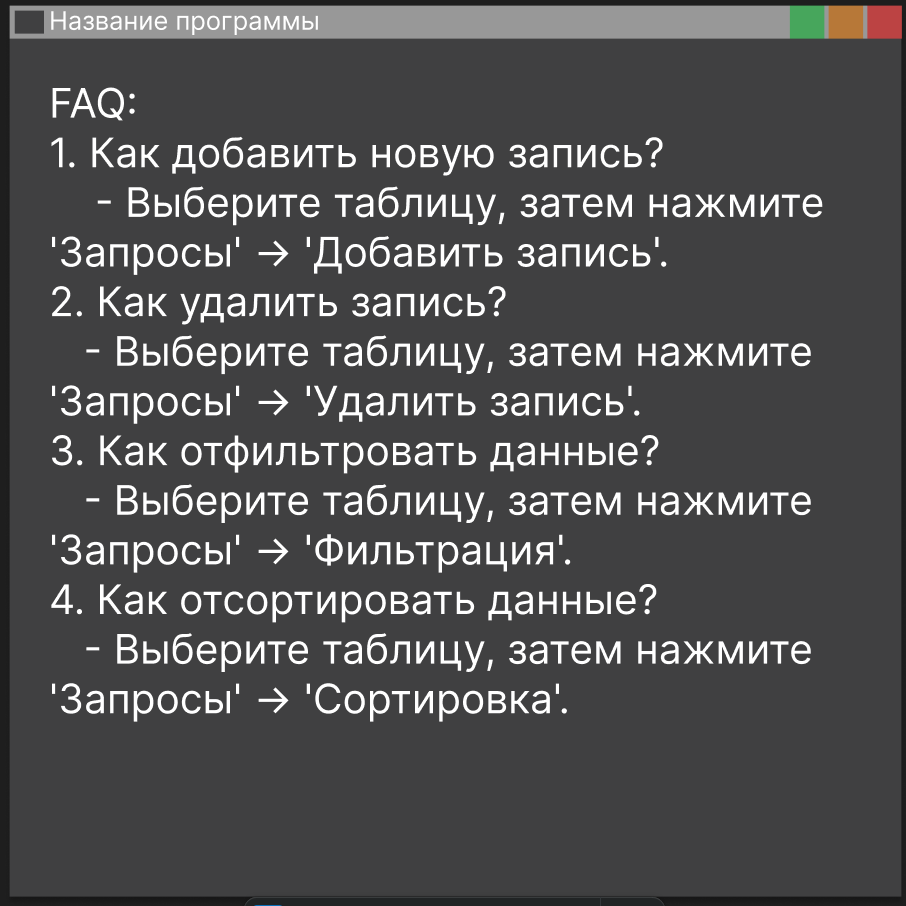
****

Рисунок 14 – Mockup-макет окна просмотра информации о ТС роли «Водитель»

### **Карта навигации приложения**



Рисунок 15 – 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:

### **Окно авторизации**



Рисунок 16 – Главное окно приложения

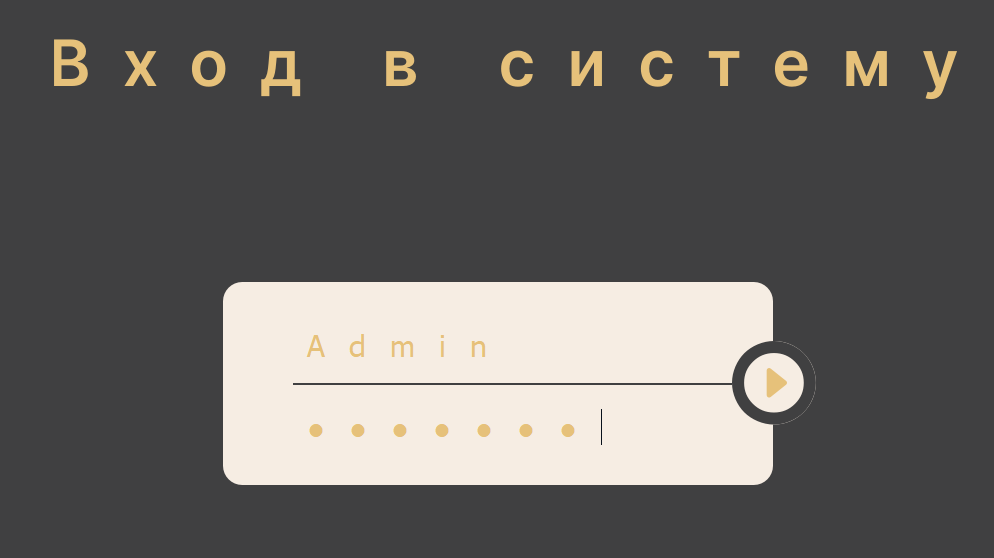


Рисунок 17 – Окно авторизации

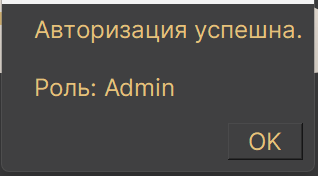


Рисунок 18 – вывод уведомления об успешном входе

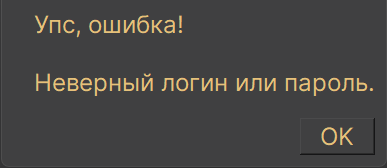


Рисунок 19 – вывод уведомления об ошибки входа

### **Окна работы с таблицами**

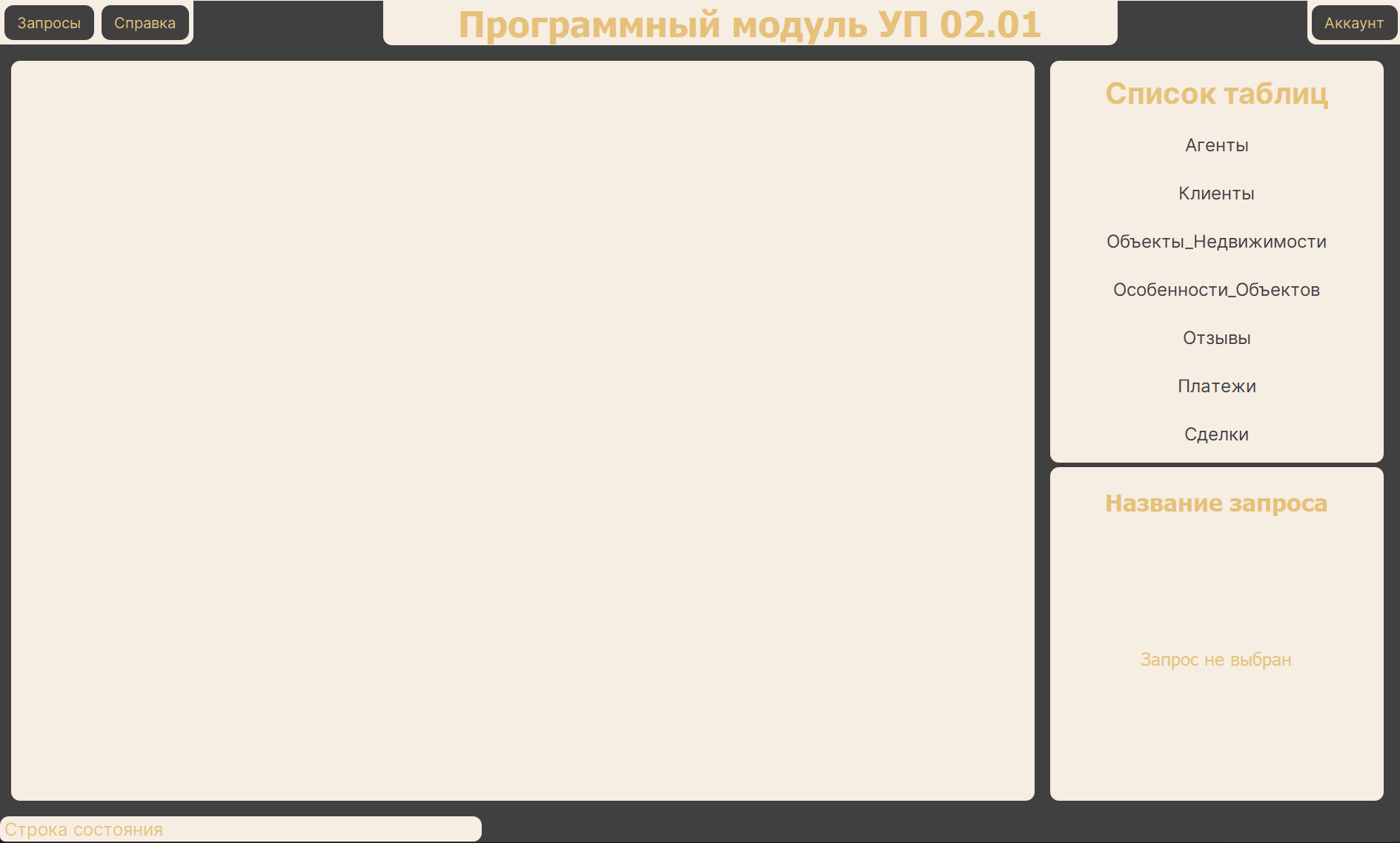


Рисунок 20 – главное окно

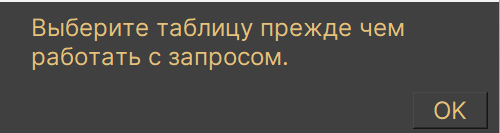


Рисунок 21 – вывод ошибки при выборе запроса, но таблица еще не открыта

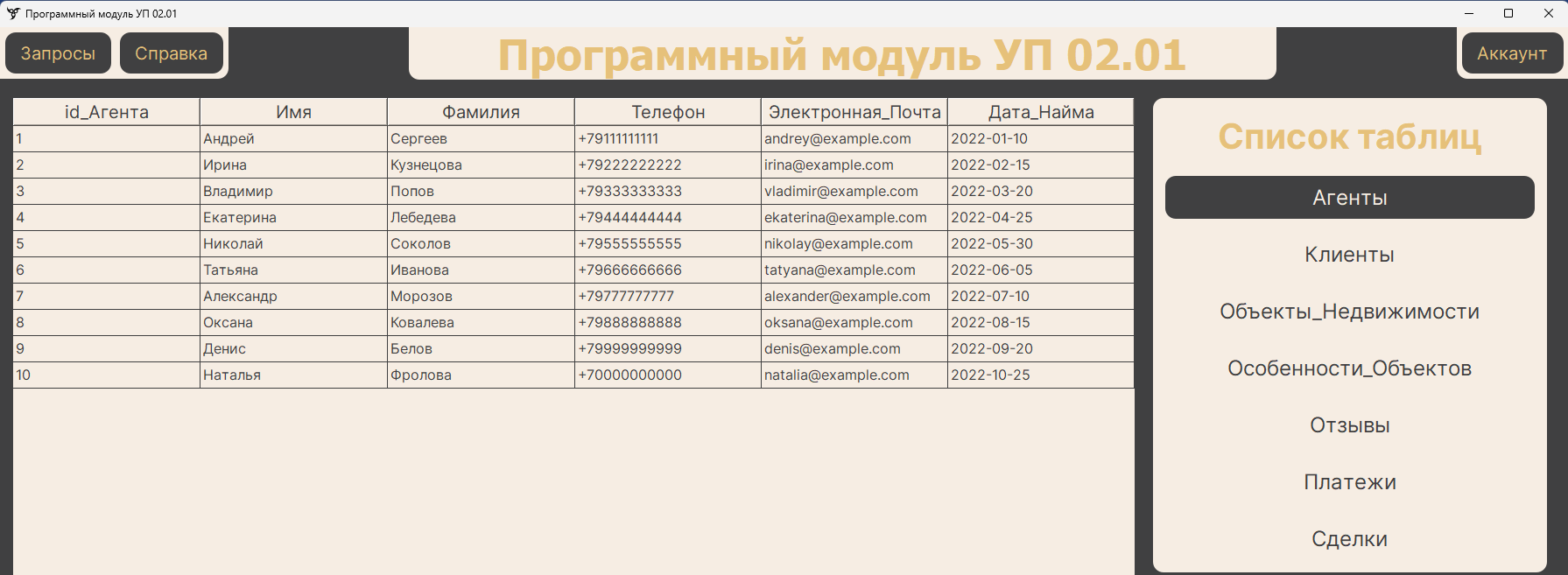


Рисунок 22 – вывод выбранной из списка таблицы

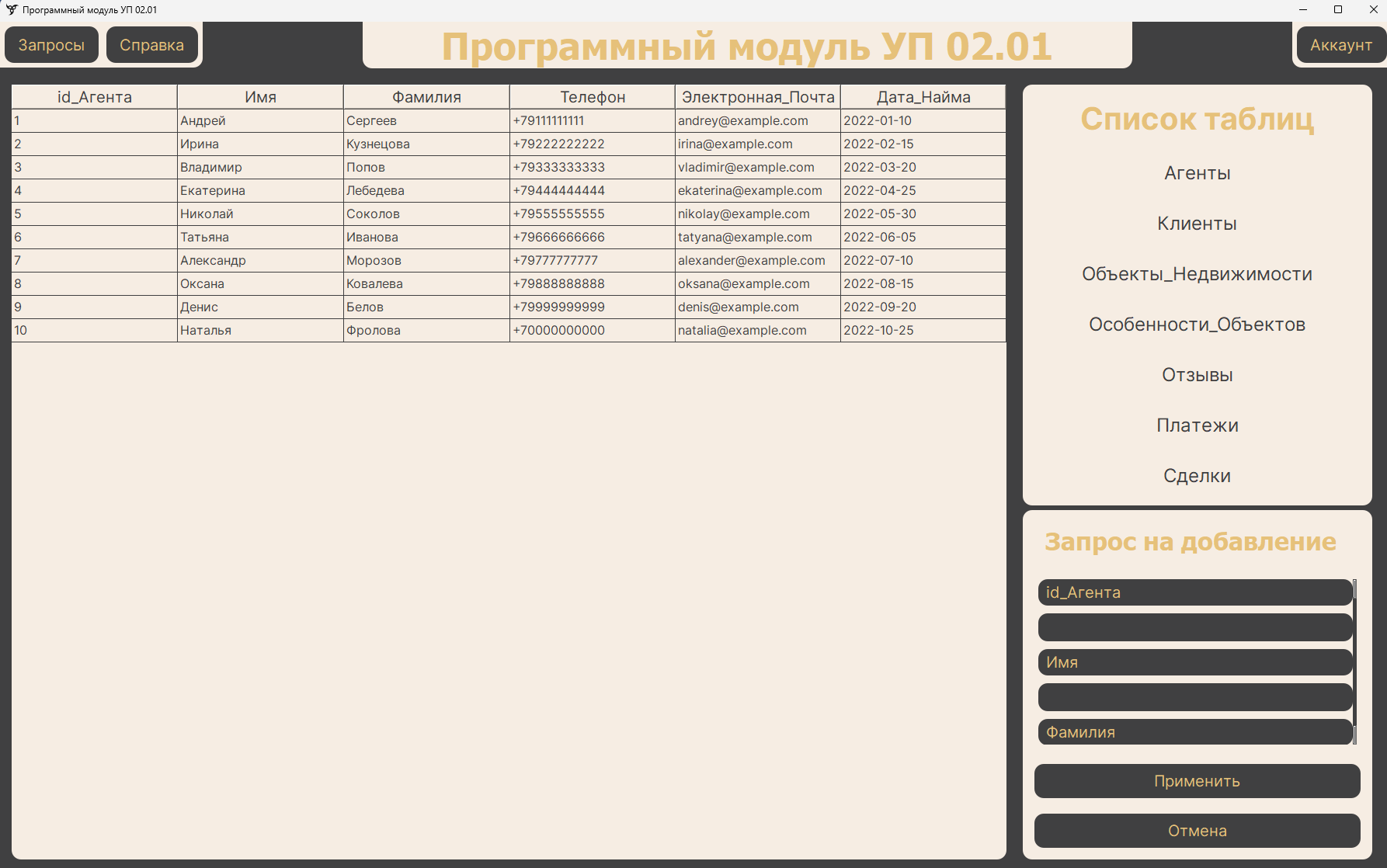


Рисунок 23 – выбор запроса на добавления

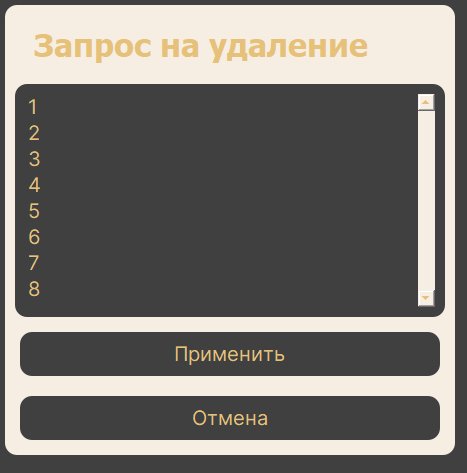


Рисунок 24 – таким же образом запрос на удаление

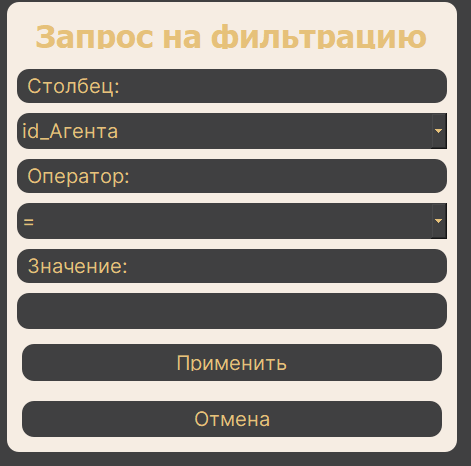


Рисунок 25 –запрос на фильтрацию

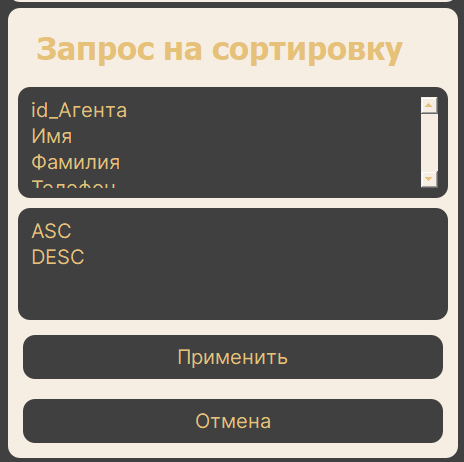


Рисунок 26 –запрос на сортировку

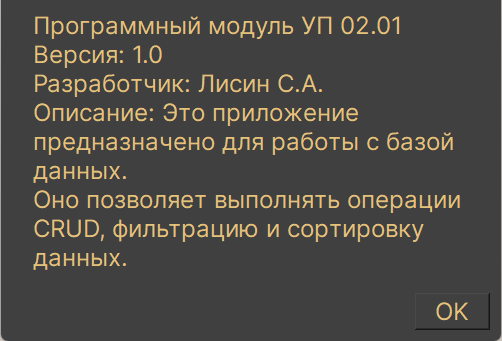


Рисунок 27 –вывод информации о программе

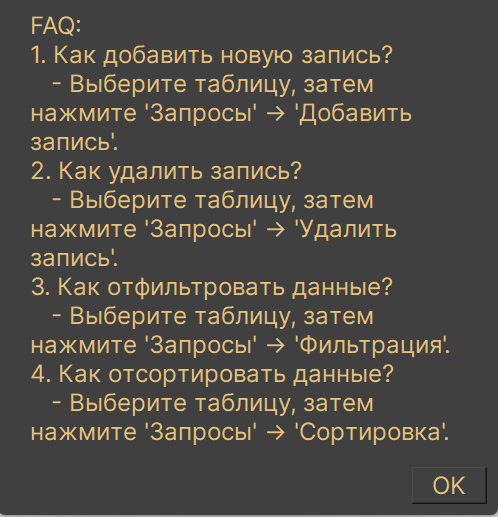


Рисунок 28 –вывод информации в формате вопрос-ответ

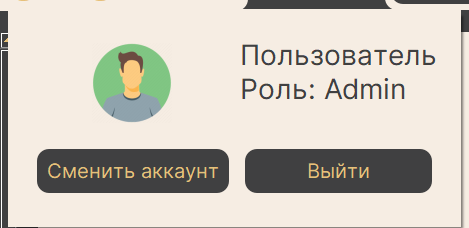


Рисунок 29 – меню аккаунта

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.

### **Tests-case**

|  |  |
| --- | --- |
| **Название проекта** | UP |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Лисин Станислав Александрович |
| **Дата(ы) теста** | 16.12.2024 |

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_AUTH\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Авторизация с корректными данными пользователя |
| **Краткое изложение теста** | Проверка успешной авторизации пользователя с корректными логином и паролем. |
| **Этапы теста** | 1. Запустить приложение. 2. Ввести корректный логин в поле "Логин". 3. Ввести корректный пароль в поле "Пароль". 4. Нажать кнопку "Войти". |
| **Тестовые данные** | * Логин: Admin * Пароль: neadmin |
| **Ожидаемый результат** | Сообщение "Авторизация успешна. Роль: Admin" и переход к основному интерфейсу приложения. |
| **Фактический результат** | Сообщение "Авторизация успешна. Роль: Admin" и переход к основному интерфейсу приложения. |
| **Статус** | Зачет (Приложение А. Рисунок 1) |
| **Предварительное условие** | Пользователь с логином Admin и паролем neadmin существует в системе. |
| **Постусловие** | Пользователь авторизован, отображается основной интерфейс приложения. |
| **Примечания/комментарии** | - |



Рисунок 30 – Успешная регистрация водителя

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_AUTH\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Авторизация с некорректными данными пользователя |
| **Краткое изложение теста** | Проверка реакции системы на ввод некорректных логина и пароля. |
| **Этапы теста** | 1. Запустить приложение. 2. Ввести некорректный логин в поле "Логин". 3. Ввести некорректный пароль в поле "Пароль". 4. Нажать кнопку "Войти". |
| **Тестовые данные** | * Логин: user123 * Пароль: password123 |
| **Ожидаемый результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Фактический результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Статус** | Зачет |
| **Предварительное условие** | Пользователь с логином user123 и паролем password123 не существует в системе. |
| **Постусловие** | Пользователь не авторизован, остается на экране авторизации. |
| **Примечания/комментарии** | - |

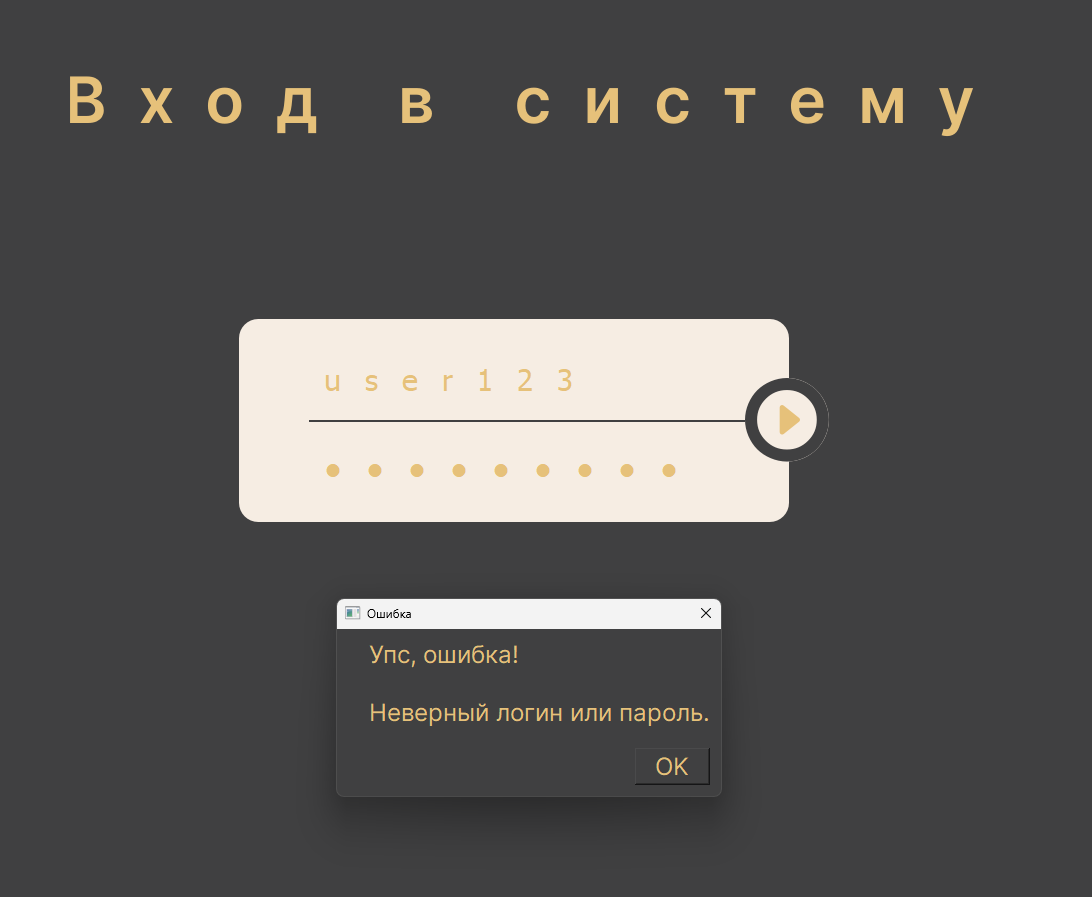


Рисунок 31 – Успешная авториация

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_AUTH\_3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Авторизация с пустыми полями логина и пароля |
| **Краткое изложение теста** | Проверка реакции системы на попытку авторизации без ввода данных. |
| **Этапы теста** | 1. Запустить приложение. 2. Оставить поле "Логин" пустым. 3. Оставить поле "Пароль" пустым. 4. Нажать кнопку "Войти". |
| **Тестовые данные** | * Логин: пусто * Пароль: пусто |
| **Ожидаемый результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Фактический результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | Пользователь не авторизован, остается на экране авторизации. |
| **Примечания/комментарии** | - |

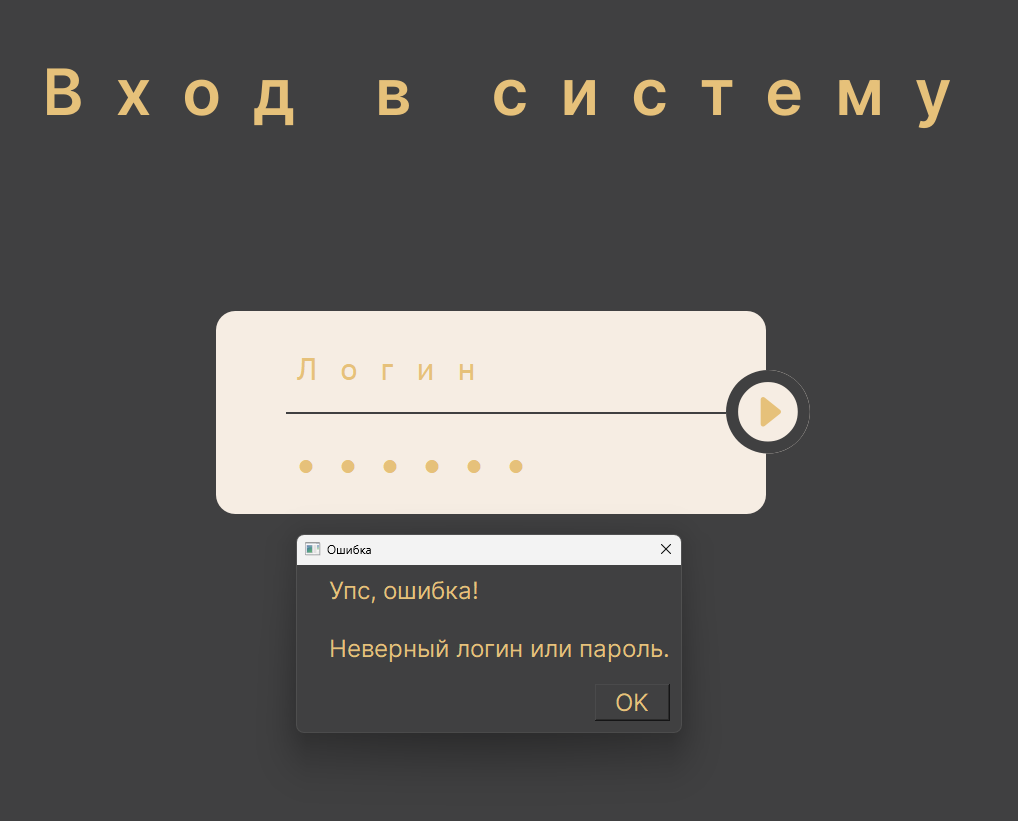


Рисунок 32 – Успешный вывод ошибки

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_AUTH\_4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Авторизация с пустым полем логина |
| **Краткое изложение теста** | Проверка реакции системы на попытку авторизации без ввода логина. |
| **Этапы теста** | 1. Запустить приложение. 2. Оставить поле "Логин" пустым. 3. Ввести корректный пароль в поле "Пароль". 4. Нажать кнопку "Войти". |
| **Тестовые данные** | * Логин: пусто * Пароль: neadmin |
| **Ожидаемый результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Фактический результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | Пользователь не авторизован, остается на экране авторизации. |
| **Примечания/комментарии** | - |

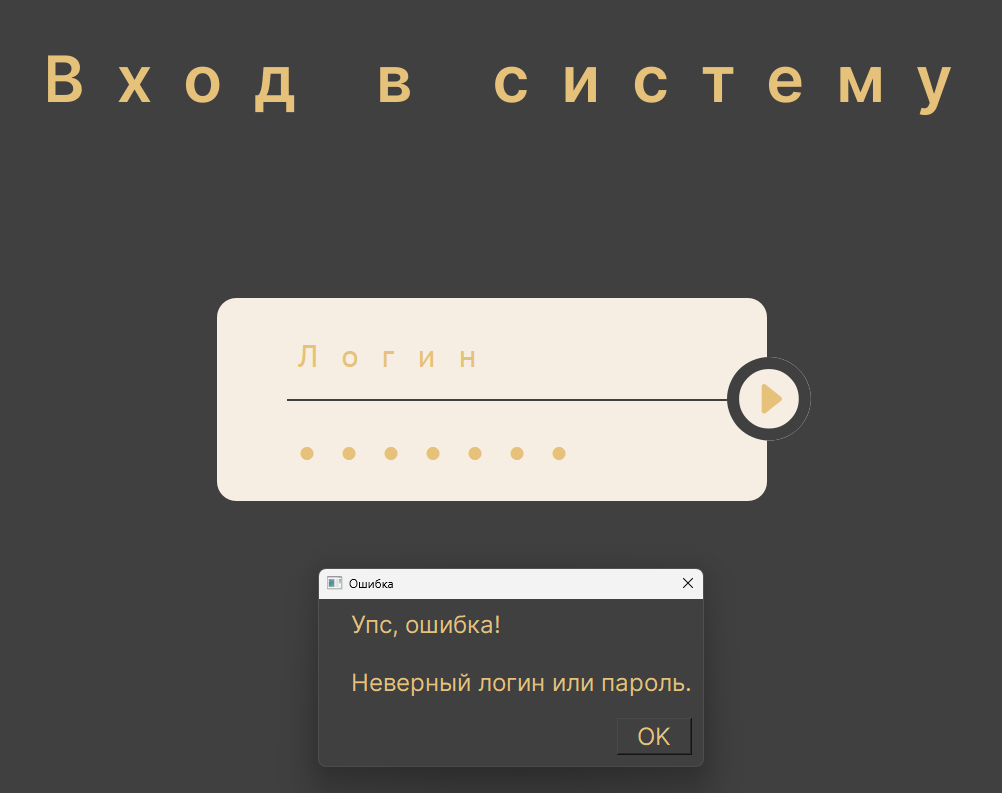


Рисунок 33 – Успешный вывод ошибки

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_AUTH\_5 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Авторизация с пустым полем пароля |
| **Краткое изложение теста** | Проверка реакции системы на попытку авторизации без ввода пароля. |
| **Этапы теста** | 1. Запустить приложение. 2. Ввести корректный логин в поле "Логин". 3. Оставить поле "Пароль" пустым. 4. Нажать кнопку "Войти". |
| **Тестовые данные** | * Логин: Admin * Пароль: пусто |
| **Ожидаемый результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Фактический результат** | Сообщение "Упс, ошибка! Неверный логин или пароль." |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | Пользователь не авторизован, остается на экране авторизации. |
| **Примечания/комментарии** | - |

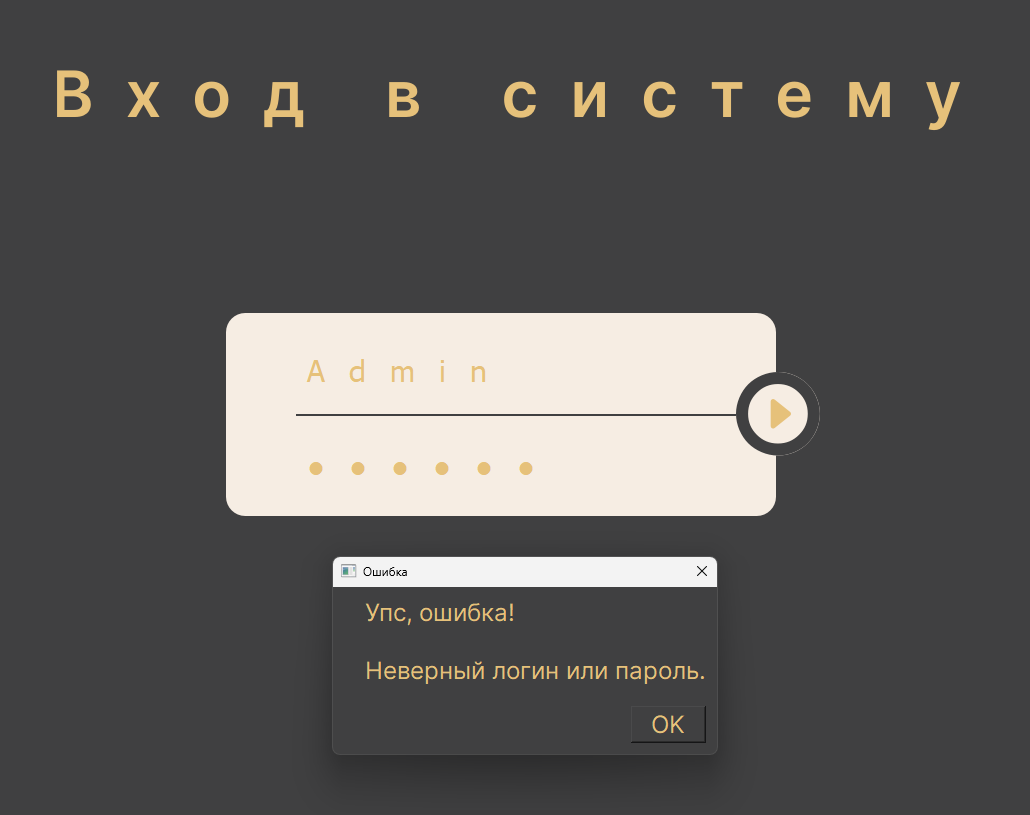


Рисунок 34 – Успешный вывод ошибки

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.

### **GitHub**

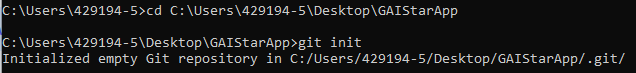


Рисунок 35 – Инициализируем Git в локальной папке

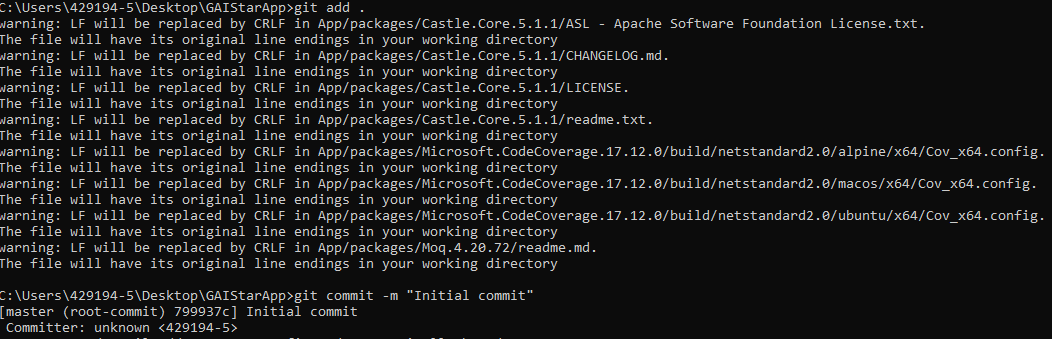


Рисунок 36 – Добавим все файлы из папки в индекс Git



Рисунок 37 – Подключаем локальный репозиторий к GitHub



Рисунок 38 – Отправляем все в GitHub

### **Ссылка на проект**

**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: 0**
  + 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**
  + 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: 2**
  + 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: 6**
  + 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: 65/80**

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

**1. Соответствие функциональным требованиям**

* **Оценка: 0/5**
  + Проект полностью реализует все функциональные требования, описанные в ТЗ.
  + Функции добавления новых клиентов, объектов недвижимости, редактирования сделок, учет состояния сделок и подбор недвижимости реализованы корректно.

**2. Удобство интерфейса**

* **Оценка: 0/5**
  + Интерфейс соответствует требованиям ТЗ: используется палитра цветов, 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: 0/5**
  + 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: 0/5**
  + 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: 0/5**
  + Используется язык программирования Python 3.11.
  + Интерфейс разработан с использованием PyQT5.
  + База данных реализована на PostgreSQL.
  + Система работает на Windows 10 и выше, что соответствует требованиям заказчика.

**6. Гибкость и масштабируемость**

* **Оценка: 0/5**
  + Система легко адаптируется к изменениям, что подтверждается реализацией дополнительных функций, не описанных в ТЗ.
  + Однако для полной масштабируемости рекомендуется доработка архитектуры системы.

**7. Качество кода**

* **Оценка: 0/5**
  + Код хорошо структурирован, соответствует стандартам и гайдлайнам.
  + Реализована обработка ошибок и исключений, что повышает надежность системы.

**8. Тестирование и отладка**

* **Оценка: 0/5**
  + 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: 0/5**
  + Предоставлено описание архитектуры системы.
  + Документация соответствует стандартам и помогает в понимании работы системы.

**10. Соответствие срокам и бюджету**

* **Оценка: 0/5**
  + Проект выполнен в не установленные сроки и в рамках бюджета.
  + Заказчик не удовлетворен результатом и считает, что проект не полностью оправдал ожидания.

**Итоговая оценка: 0/50**

Проект выполнен на высоком уровне, полностью соответствует требованиям технического задания и ожиданиям заказчика.

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.

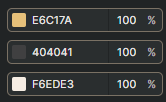
В ходе практики были успешно освоены навыки проектирования, разработки и тестирования программных модулей, что подтверждается выполнением всех поставленных задач. Проект полностью соответствует требованиям технического задания и ожиданиям заказчика. Система демонстрирует высокую стабильность, удобство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

Результаты практики подтверждают, что разработанная система готова к внедрению в эксплуатацию.

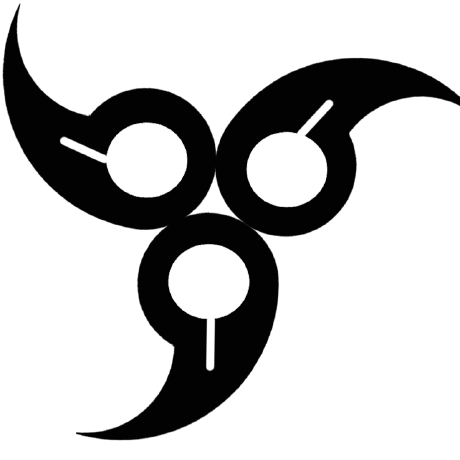
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Девятко Н.С. Учебное пособие по МДК 01.02 Прикладное программирование – СПб: ФГАОУ ВО СПбПУ Институт среднего профессионального образования, 2019, 95 с.
2. ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
3. ГОСТ 24.301-80. Информационная технология. Комплекс стандартов на автоматизированные системы. Общие положения.
4. Оглышев А.А. Основы проектирования информационных систем. – М.: Интернет-Университет Информационных Технологий, 2018. – 320 с.

ПРИЛОЖЕНИЕ A



ПРИЛОЖЕНИЕ Б

**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

Лисин Станислав Александрович

(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