ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА

[КОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ](http://knvsh.gov.spb.ru/)

Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

**«ПЕТРОВСКИЙ КОЛЛЕДЖ»**

(СПб ГБПОУ «Петровский колледж»)

*Отделение информационно – промышленных технологий и судостроения (прописывается наименование отделения)*

**ОТЧЕТ**

**по учебной практике обучающегося**

**по специальности** 09.02.07 Информационные системы и программирование

**группы** 50-02

Притчина Павла Викторовича

(фамилия, имя, отчество)

*ПМ.02. Осуществление интеграции программных модулей*

**Место прохождения практики:** \_ *СПб ГБПОУ «Петровский колледж»*

*(название организации, юридический адрес, телефон)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Период прохождения** *с 04.12.2023 по 16.12.2023 г.*

**Руководитель практики от образовательной организации** \_\_\_\_\_\_\_\_\_\_

Бережков А.В.

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись, дата)

Санкт - Петербург, 2023 г.

**Содержание отчета по учебной практике:**

1. Задание по учебной практике
2. Дневник по учебной практике (заверенный подписью наставника практики от профильной организации с печатью)
3. Характеристика (заверенная подписью наставника практики от профильной организации с печатью)
4. Аттестационный лист (заверенный подписью наставника практики от профильной организации с печатью)
5. Акт о выполнении пробной квалификационной работы (заверенный подписью наставника практики от организации с печатью)
6. Благодарственные письма, отзывы о работе обучающегося (при наличии)
7. Отчётный материал в соответствии с заданием:
   1. Описание предметной области
   2. Анализ существующих аналогов
   3. Требования к разрабатываемой информационной системе
   4. Разработка модели базы данных
   5. Проектирование программного обеспечения
   6. Реализация основных функций программного обеспечения
   7. Тестирование программного обеспечения

В отчет (Приложение) необходимо включить листинги программных кодов, где реализуются *функции программы*. Для описания работы приложения в отчете необходимо использовать *скриншоты*. Для защиты практики необходимо создать *презентацию разработанного ПО 12-14* слайдов.

1. **Пояснительная записка для обучающихся, проходящих учебную практику (по профилю специальности)**

Учебная практика (по профилю специальности) является составной частью ППССЗ СПО, обеспечивающей реализацию федерального государственного образовательного стандарта среднего профессионального образования (далее – ФГОС СПО) по специальности09.02.07 Информационные системы и программирование

Учебная практика направлена на формирование у обучающегося общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках профессиональных модулей ППССЗ СПО по каждому из видов профессиональной деятельности, предусмотренных ФГОС СПО по специальности.

В период прохождения практики на обучающегося распространяются требования охраны труда и правила внутреннего распорядка, действующие в профильной организации, а также трудовое законодательство, в том числе в части государственного социального страхования.

В период прохождения практики обучающиеся обязаны:

* выполнять правила внутреннего распорядка, требования техники безопасности нахождения в производственных мастерских образовательной организации;
* знать и соблюдать нормы охраны труда и правила пожарной безопасности;
* выполнять все виды работ, предусмотренные программой практики и заданием, соблюдая график прохождения практики, и ежедневно вносить в дневник практики запись о видах выполняемых работ, заверенную подписью руководителя практики от образовательной организации;
* по окончании практики оформить дневник и представить его непосредственному руководителю практики от образовательной организации для проверки в установленные сроки;
* своевременно проходить промежуточную аттестацию по практике.

Аттестация по итогам практики проводится с учетом (или на основании) результатов ее прохождения, подтверждаемых документами, подписанными руководителем практики от образовательной организации.

По результатам практики руководителем практики формируется аттестационный лист, содержащий сведения об уровне освоения обучающимся профессиональных компетенций в период прохождения практики.

**СПб ГБПОУ «ПЕТРОВСКИЙ КОЛЛЕДЖ»**

***Отделение информационных технологий***

**ЗАДАНИЕ ПО УЧЕБНОЙ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ) ПРАКТИКЕ**

|  |  |  |
| --- | --- | --- |
| **Вид деятельности** | **Виды работ** | **Требования к результатам освоения** |
| ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент | Создание проектной и технической документации на программный продукт при структурном или объектно-ориентированном подходе к программированию | **Иметь практический опыт:**  модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения  **уметь:**  использовать выбранную систему контроля версий; использовать методы для получения кода с заданной функциональностью и степенью качества  **знать:**  модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения |
| ПК 2.2. Выполнять интеграцию модулей в программное обеспечение | Разработка программных продуктов с помощью различных инструментальных средств. Сборка модулей программного обеспечения в программный комплекс. |
| ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств | Отладка программы в среде программирования |
| ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. | Разработка тестовых наборов и тестовых сценариев для программного обеспечения. |
| ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования | Инспектирование компонент разрабатываемого программного продукта на предмет соответствия стандартами кодирования |

Руководители практики от образовательной организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*\_\_\_\_\_\_\_\_\_\_\_\_\_*/ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА

[КОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ](http://knvsh.gov.spb.ru/)

Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

**«ПЕТРОВСКИЙ КОЛЛЕДЖ»**

(СПб ГБПОУ «Петровский колледж»)

***Отделение информационных технологий***

**ДНЕВНИК**

**учебной (по профилю специальности) практики обучающегося**

**Притчина Павла Викторовича**

(фамилия, имя, отчество)

**Специальность 09.02.07 Информационные системы и программирование**

***ПМ 02 Осуществление интеграции программных модулей***

Период прохождения: *с 04.12.2023 по 16.12.2023 г.*

Группа 50-02

**Руководитель практики от образовательной организации**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Бережков А.В.

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись, дата)

**Ведущий преподаватель** Бережков А.В

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись, дата)

Санкт - Петербург, 2023 г

|  |  |  |
| --- | --- | --- |
| **Дата** | **Виды выполняемой работы** | **Подпись руководителя практики** |
| 04.12 | Знакомство с заданием. Создание, заполнение базы данных. |  |
| 05.12 | Реализация авторизации пользователей по ролям. Создание рабочих окон для клиента и администратора. |  |
| 06.12 | Создание и настройка стилей страницы вывода данных. |  |
| 07.12 | Создание и настройка стилей страницы добавления/редактирования/удаления заявки на оказание услуг. |  |
| 08.12 | Создание добавления/редактирования/удаления заявки на оказание услуг. |  |
| 09.12 | Оформление отчетной документации |  |
| 11.12 | Создание добавления/редактирования/удаления заявки на оказание услуг. |  |
| 12.12 | Настройка фильтрации, сортировки и поиска на странице вывода данных |  |
| 13.12 | Тестирование программного обеспечения |  |
| 14.12 | Оформление отчетной документации |  |
| 15.12 | Оформление отчетной документации |  |
| 16.12 | Защита практики |  |

Личные наблюдения обучающегося:

В течение учебной практики по интеграции программных модулей в колледже я столкнулся с множеством интересных и сложных задач, которые требовали от меня не только теоретических знаний, но и навыков практической реализации. В частности, я принимал участие в проекте, направленном на объединение различных программных компонентов для создания более эффективной и современной системы.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Одной из ключевых задач была интеграция базы данных с пользовательским интерфейсом. В процессе работы я понял, насколько важно тщательно планировать структуру данных, чтобы она соответствовала потребностям приложения. Я изучил методы работы различными протоколами передачи данных, что позволило эффективно обмениваться информацией между модулями.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Характеристика деятельности обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Притчина Павла Викторовича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО)

при прохождении учебной практики

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Общие компетенции | Основные показатели оценки результата | Уровень (низкий, средний, высокий) |
| ОК 01 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам | Умения: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;  составить план действия; определить необходимые ресурсы;  владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) |  |
| ОК 02 | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности | Умения: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска |  |
| ОК 03 | Планировать и реализовывать собственное профессиональное и личностное развитие. | Умения: определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования |  |
| ОК 04 | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. | Умения: организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности |  |
| ОК 05 | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. | Умения: грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе |  |
| ОК 06 | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей. | Умения: описывать значимость своей специальности |  |
| ОК 07 | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. | Умения: соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности |  |
| ОК 08 | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | Умения: использовать достижения жизненных и профессиональных целей; пользоваться средствами профилактики перенапряжения характерными для данной специальности |  |
| ОК 09 | Использовать информационные технологии в профессиональной деятельности | Умения: применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение |  |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языках. | Умения: понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы |  |
| ОК 11 | Планировать предпринимательскую деятельность в профессиональной сфере | Умения: выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования |  |

Руководитель практики от образовательной организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

|  |  |  |  |
| --- | --- | --- | --- |
| **аттестационный лист по практике**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Притчина Павла Викторовича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,  *ФИО*  обучающийся на 4 курсе по специальности СПО 09.02.07 Информационные системы и программирование,прошел(ла) учебную (по профилю специальности) практику по профессиональному модулю *ПМ .02 Осуществление интеграции программных модулей*  в объеме **72** часа с «05» декабря 2022 г. по «17» декабря 2022 г. | | | |
| Виды работ | Основные показатели оценки результата | Код компетенции  (ПК) | Оценка |
| Создание проектной и технической документации на программный продукт при структурном или объектно-ориентированном подходе к программированию | Проектная и техническая и техническая документация составлена в соответствии с ГОСТ.  Охвачены все информационные объекты предметной области и взаимосвязи между ними.  При создании документации соблюдены все правила и графические нотации | ПК 2.1 |  |
| Разработка программных продуктов с помощью различных инструментальных средств. Сборка модулей программного обеспечения в программный комплекс. | Разработанное программное обеспечение охватывает все необходимые функции.  Инструментарий используется верно | ПК 2.2 |  |
| Отладка программы в среде программирования | Приложение выполняется без критических ошибок.  Отладка программы в среде программирования производилась в соответствии с правилами. | ПК 2.3 |  |
| Разработка тестовых наборов и тестовых сценариев для программного обеспечения. | Тестовые сценарии охватывают весь функционал программного обеспечения. | ПК 2.4 |  |
| Инспектирование компонент разрабатываемого программного продукта на предмет соответствия стандартами кодирования | Компоненты разрабатываемого программного обеспечения соответствуют стандартами кодирования | ПК 2.5 |  |
| Итоговая оценка по практике ПМ 02 Осуществление интеграции программных модулей | | |  |

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_Бережков А. В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

руководитель практики от образовательной организации

* 1. Описание предметной области

База данных "Учёт заявок в студии звукозаписи" предназначена для хранения информации о заявках на оказание услуг, которые поступают в студию звуокзаписи. Каждая заявка содержит следующие данные: логин клиента, его изображение, тип услуги, количестве услуг, которые надо оказать, итоговую стоимость (количество \* стоимость одной услуги), а также статус заявки.

После авторизации пользователь получает доступ к остальным модулям системы:

* клиент;
* администратор;
* гость.

Поиск должен осуществляется по логину клиента, сортировка по номеру заявки и логину клиента (по возрастанию и убыванию), а фильтрация производится по типу услуги. Поиск, сортировка и фильтрация должны осуществляться одновременно.

* 1. Анализ существующих аналогов

В таблице номер 1 приведено сравнение функциональных возможностей разрабатываемой ИС и её аналогов. В данной таблице параметром «1» обозначены значения, которые имеются в ИС; параметр «0» - отсутствующие.

Таблица 1 – Сравнение аналогов разработанной ИС

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Критерий | ИС «Intradesk» | ИС «happydesk» | ИС  «HubEx» | Разрабатываемая ИС |
| 1 | Наличие формы авторизации | 1 | 0 | 1 | 1 |
| 2 | Поиск клиентов в каталоге | 1 | 1 | 0 | 1 |
| 3 | Сортировка заявок по номеру или клиенту | 0 | 1 | 0 | 1 |
| 4 | Переход к редактированию заявки | 1 | 1 | 1 | 1 |

* 1. Требования к разрабатываемой информационной системе

Функционал разделяется на две категории: функционал для клиентов и администраторов.

Клиент может:

* просмотреть все свои заявки;
* найти товар по названию;
* отсортировать товары по номеру заявки;
* отфильтровать товары по типу услуги
* оставить заявку на оказание услуг.

Для администраторов круг действий шире. Кроме возможностей клиента, администратору доступно:

* редактирование данных об услуге;
* удаление заявок;
* просмотр заявок от всех клиентов;
* удалить заявку.

Гость может:

* просмотреть все типы услуг и их стоимость;
  1. Разработка модели базы данных

В ходе разработки модели базы данных, были выделены следующие таблицы (рисунок 1-5 соответственно): Пользователь (Users), Роль пользователя (Roles), Заявка (Applications), Тип услуги в заявке (ApplicationTypes), Статус заявки (ApplicationStatuses)

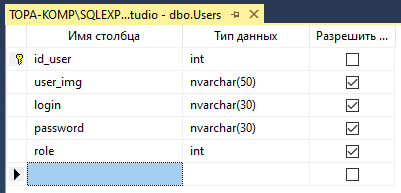


Рисунок 1 – Таблица Пользователь

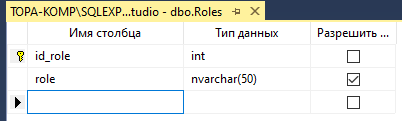


Рисунок 2 – Таблица Роль пользователя

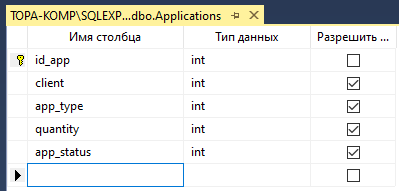


Рисунок 3 – Таблица Заявка

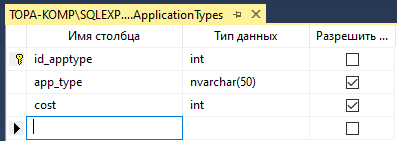


Рисунок 4 – Таблица Тип услуги в заявке

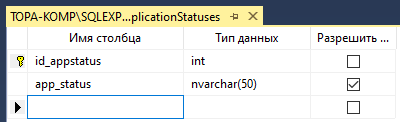


Рисунок 5 – Таблица Статус заявки

Отношение находится во 3NF. Схема базы данных «Учет продаж в магазине одежды» представлена на рисунке 6.

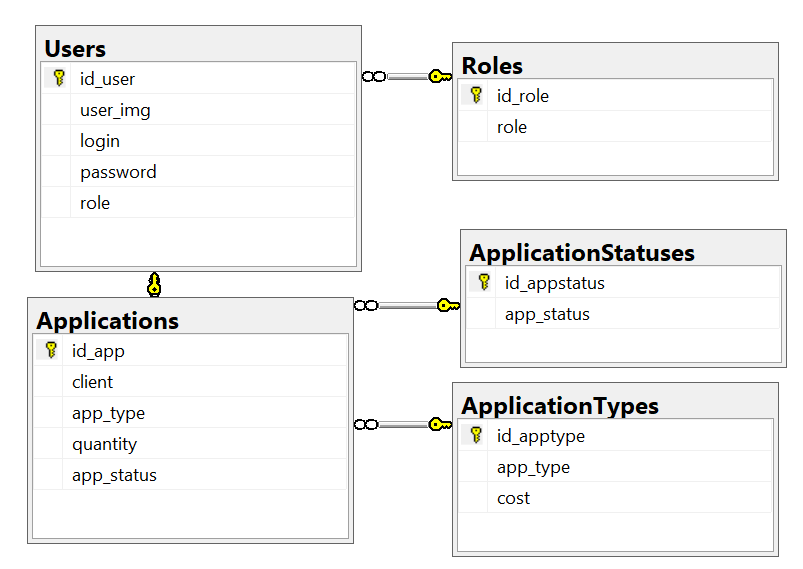


Рисунок 6 – Схема базы данных «Учет продаж в магазине одежды»

* 1. Проектирование программного обеспечения

Для каталога товаров был создан макет, на основе которого будет создано приложение (рисунок 7).

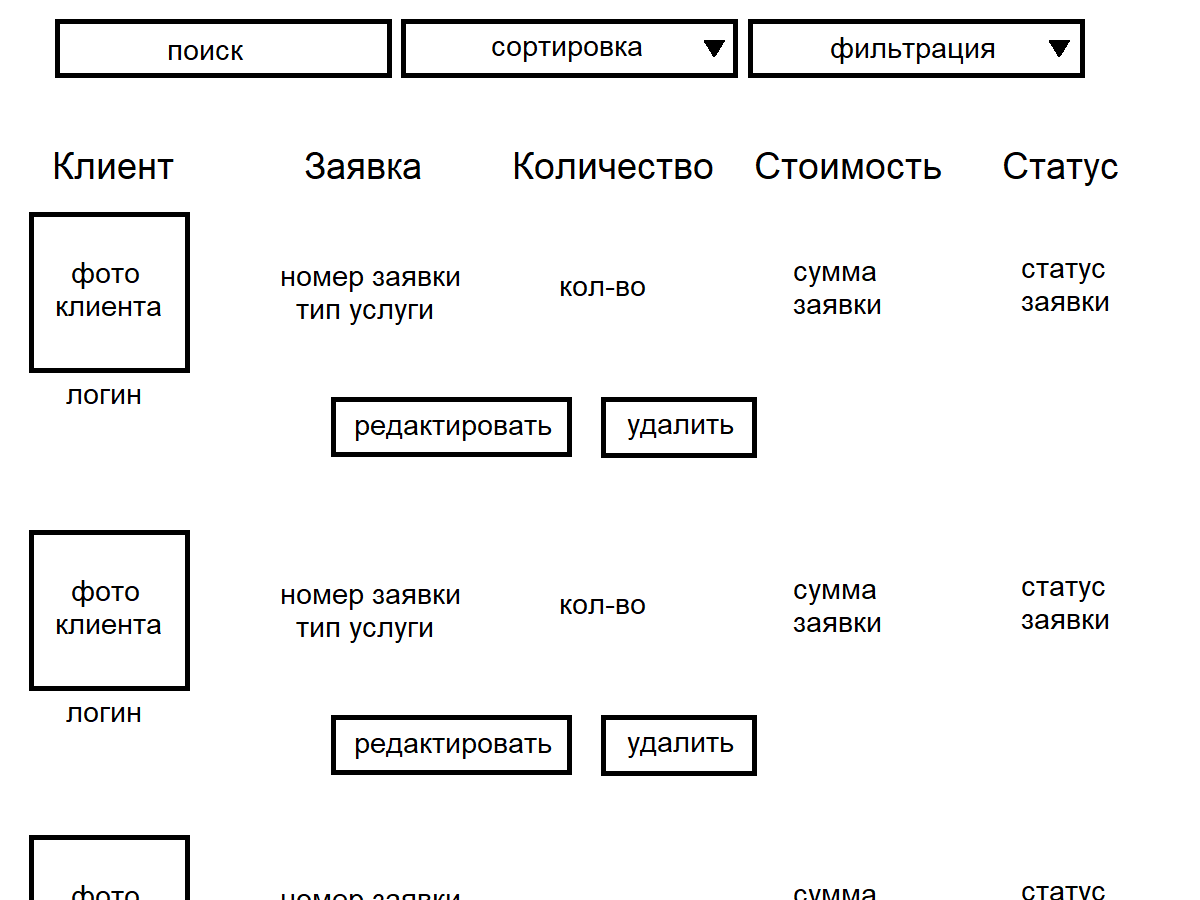


Рисунок 7 – Макет вывода заявок

На основе разработанного макета был написан интерфейс вывода заявок для создаваемой информационной системы. Этот интерфейс применяет технологию WPF для формирования клиентских приложений и язык разметки XAML (рисунок 8).

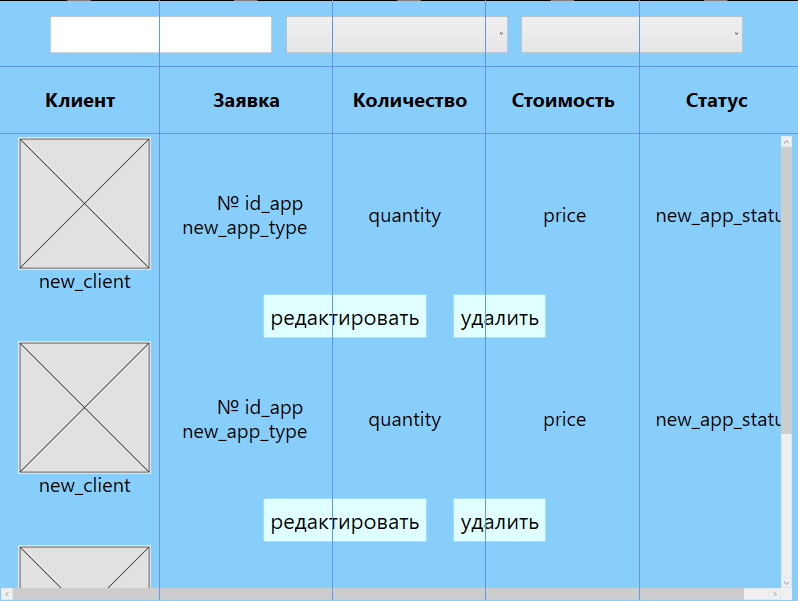


Рисунок 8 – Форма вывода заявок на основе макета

* 1. Реализация основных функций программного обеспечения

При запуске приложения появляется окно и страница авторизации (рисунок 9), где пользователь может авторизоваться или зайти как гость. Листинги а.1 и а.2 представлены в Приложение А.

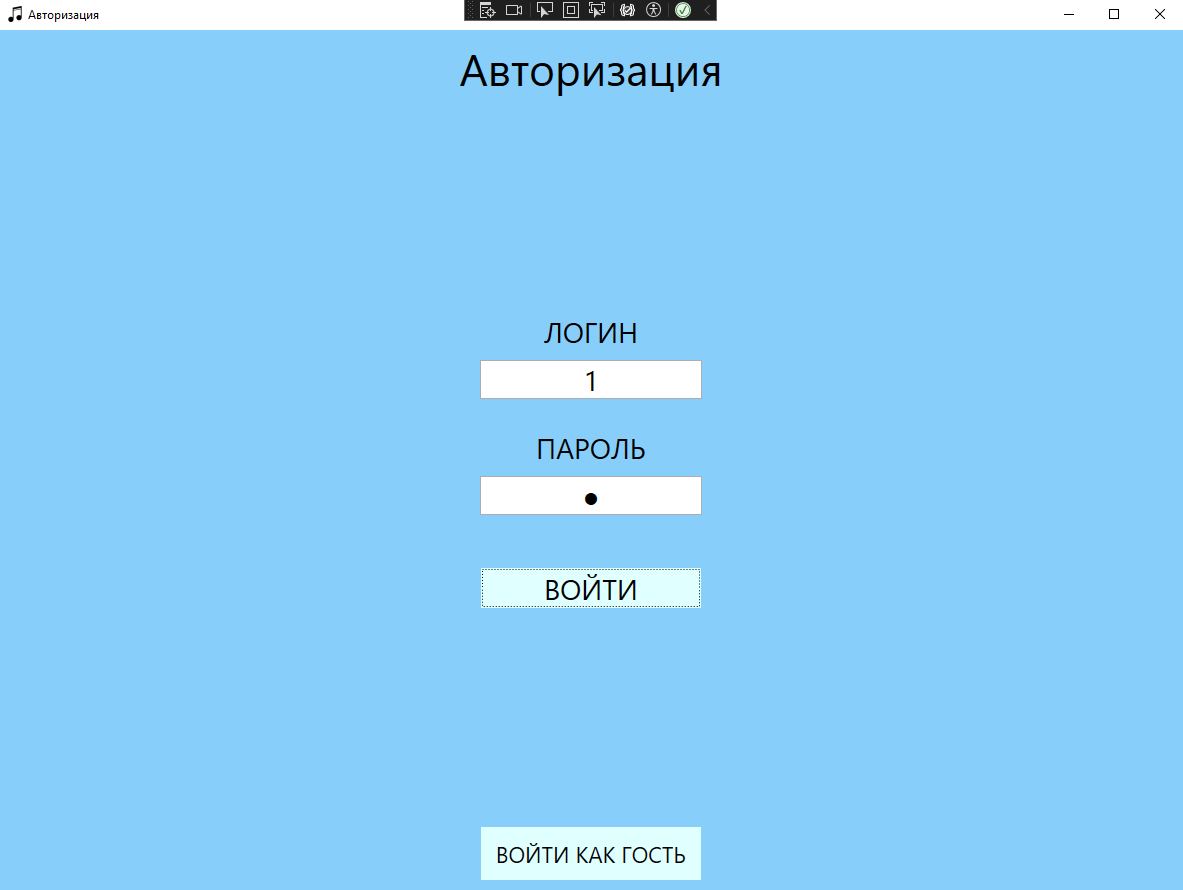


Рисунок 9 – Форма авторизации

Если пользователь нажал кнопку «войти как гость», то он увидит видит окно и страницу, которые представлены на рисунке 10. Листинги а.3 и а.4 представлены в Приложение А.



Рисунок 10 – Форма вывода всех услуг для гостя

После авторизации пользователь переходит на страницу заявок, где он может просмотреть все, если это администратор, или только свои, если это клиент, заявки, найти заявку по логину клиента, отсортировать по номеру заявки или логину клиента и отфильтровать по типу услуги в заявке (рисунки 11 и 12). Также, администратору видны кнопки «редактировать» и «удалить». Листинг а.5 представлен в приложении А.



Рисунок 11 – Форма вывода всех заявок для клиента с логином «1»



Рисунок 12 – Форма вывода всех заявок для администратора

При нажатии на кнопку «добавить», открывается форма добавления заявки. Для клиента видны только поле выбора типа услуги и поле для ввода количества услуг, которые ему нужны, а также есть текстовое поле, в котором отображается сумма заявки (рисунок 13). Для администратора дополнительно доступны поле выбора клиента из списка и поле для установки статуса у заявки (рисунок 14). Также, эта форма используется для редактирования данных и в этом случае становится виден номер редактируемой заявки (рисунок 15). При нажатии на кнопку «удалить», появляется всплывающее окно, в котором надо подтвердить действие. Заявка удаляется после подтверждения (рисунок 16 и 17) Листинги а.6 и а.7 представлены в приложении А.



Рисунок 13 – Форма добавления заявки для клиента

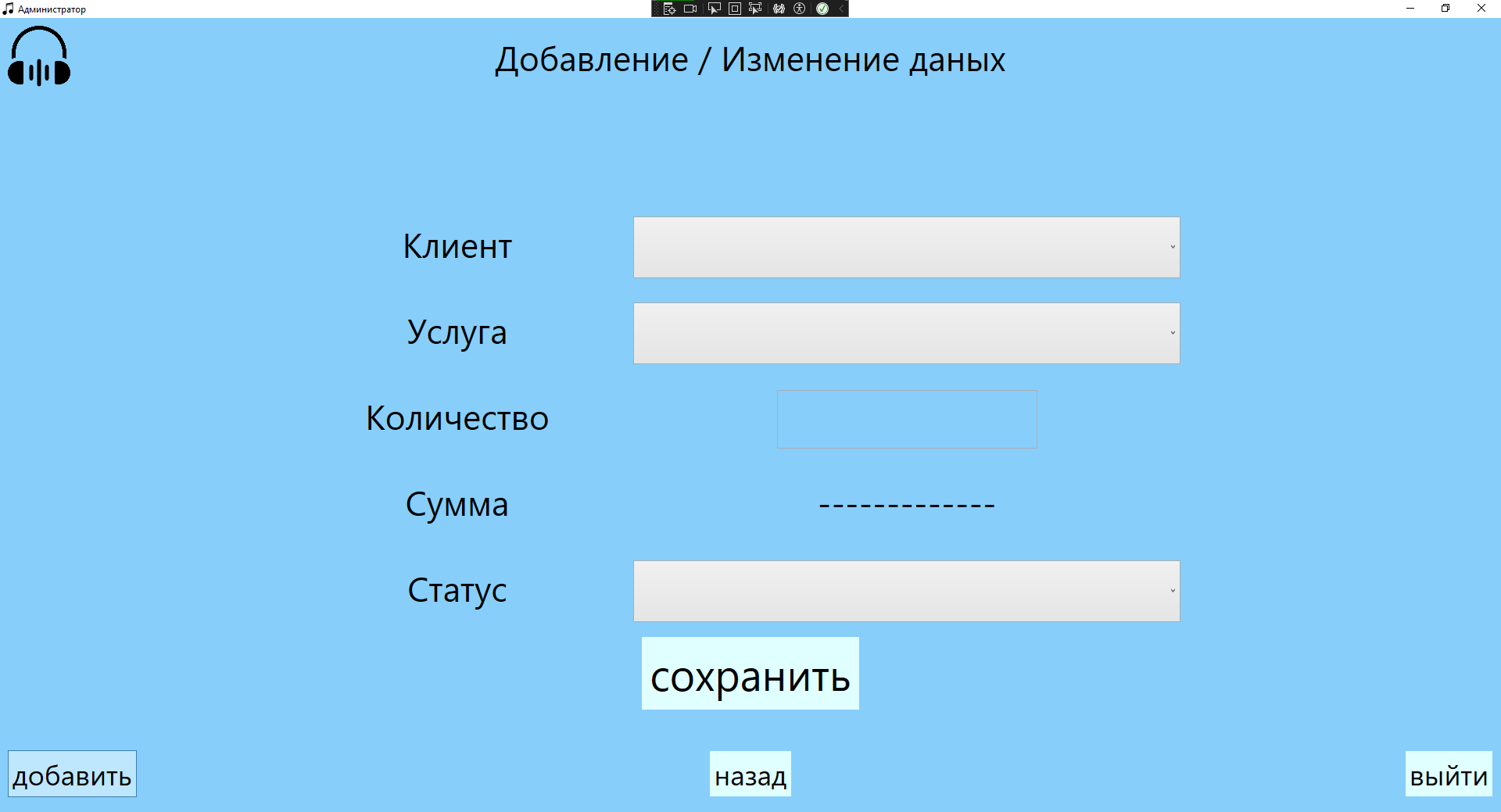


Рисунок 14 – Форма добавления заявки для администратора

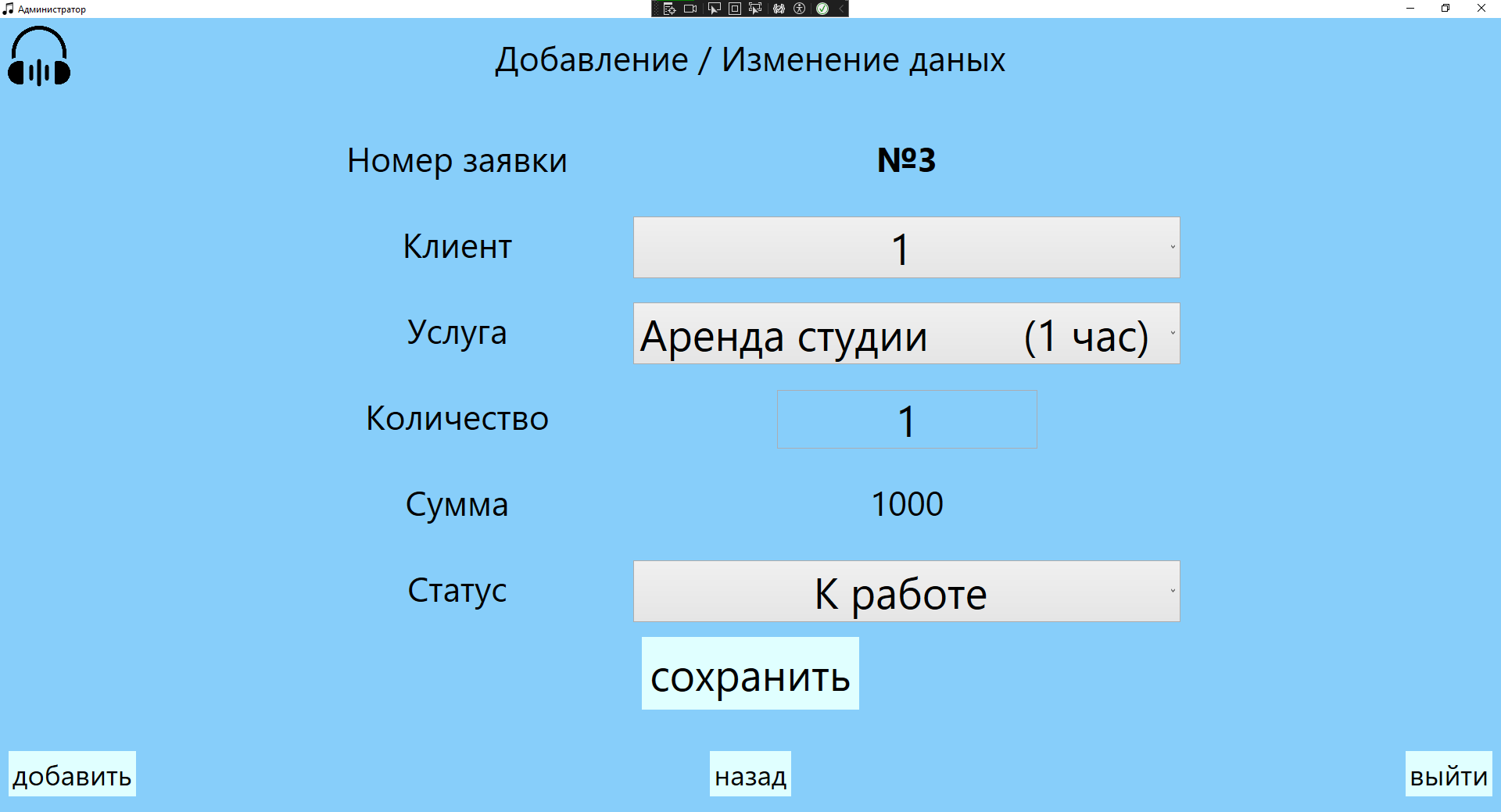


Рисунок 15 – Форма редактирования заявки для администратора

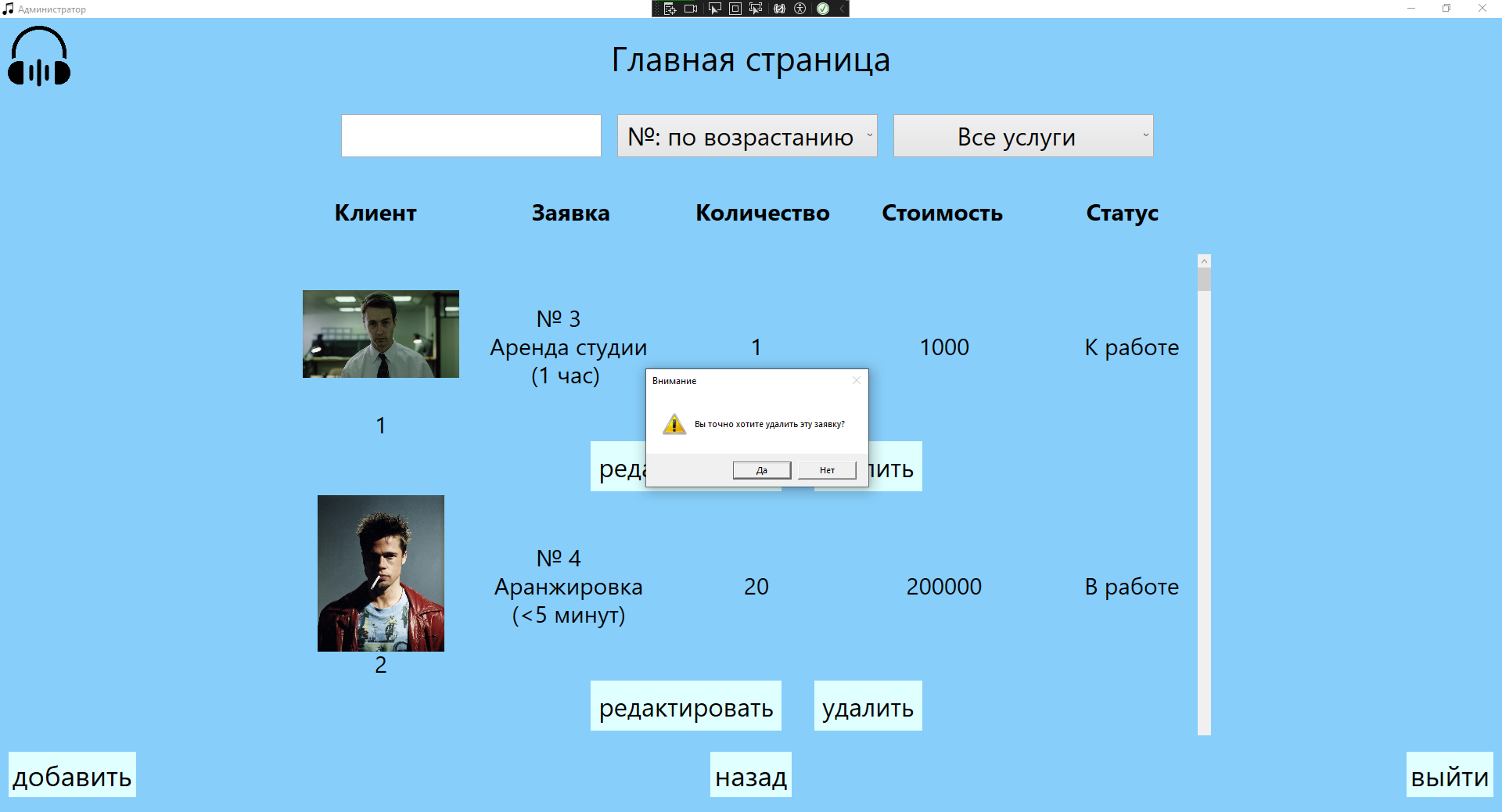


Рисунок 16 – Подтверждение удаления заявки



Рисунок 17 – Заявка удалена

* 1. Тестирование программного обеспечения

В ходе учебной практики было выполнено задание на работу с системой контроля версий Git и созданы модульные тесты (таблица 2). Результаты этих задач показаны в Приложении Б.

Таблица 2 – Модульные тесты ИС

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Название теста | Описание | Тестовые данные | Ожидаемый результат | Фактический результат |
| 1 | ValidateLgnPswTest | Ввод пустых полей на этапе авторизации | логин «»; пароль «» | Возвращается значением «false» | Возвращается значением «false» |
| 2 | ValidateLgnPswTest2 | Ввод корректных значений в поля на этапе авторизации | логин «1»; пароль «1» | Возвращается значением «true» | Возвращается значением «true» |
| 3 | ValidateLgnPswTest3 | Ввод не всех полей на этапе авторизации | логин «1»; пароль «» | Возвращается значением «false» | Возвращается значением «false» |
| 4 | BtnChekTest | Возможность переход на предыдущую страницу по кнопке «Назад» | Нажатие на кнопку «Назад» | Возвращается значением «true» | Возвращается значением «true» |
| 5 | ValidateQuantityTest | Ввод пустого значения в поле «Количество» | «» | Возвращается значением «false» | Возвращается значением «false» |
| 6 | ValidateQuantityTest2 | Ввод текстового значения в поле «Количество» | «bebra» | Возвращается значением «false» | Возвращается значением «false» |
| 7 | ValidateQuantityTest3 | Ввод дробного числового значения в поле «Количество» | «6,9» | Возвращается значением «false» | Возвращается значением «false» |
| 8 | ValidateQuantityTest4 | Ввод целочисленного значения в поле «Количество» | «3» | Возвращается значением «true» | Возвращается значением «true» |

Результат прохождения тестов можно увидеть на рисунке 18. Благодаря тестированию программного обеспечения, были исправлены ошибки в работе приложения, и разработанная ИС была оптимизирована.

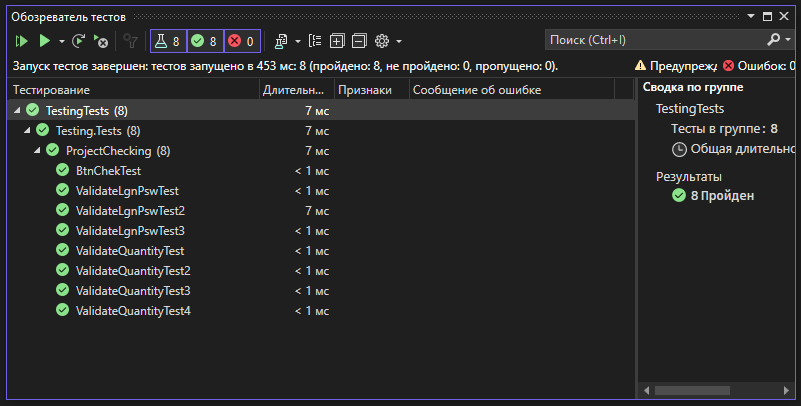


Рисунок 18 – Результат прохождения тестов

Приложение А

Листинг а.1 – Код окна авторизации

public partial class AuthorizationWindow : Window

{

public AuthorizationWindow()

{

InitializeComponent();

FrameAuthoriz.Navigate(new Authorization());

}

private void btnLoginGuest\_Click(object sender, RoutedEventArgs e)

{

GuestWindow gWindow = new GuestWindow(0);

gWindow.Show();

GetWindow(this).Close();

}

}

Листинг а.2 – Код страницы авторизации

public partial class Authorization : Page

{

public Authorization()

{

InitializeComponent();

}

private void btnLogin\_Click(object sender, RoutedEventArgs e)

{

try

{

var user\_log = App.Context.Users.Where(u => u.login == txtLogin.Text.Trim()).ToList();

if (user\_log.Count == 1)

{

if (user\_log[0].password.ToString() == txtPsw.Password.ToString())

{

var currentUser = App.Context.Users.Where(u => u.login == txtLogin.Text).FirstOrDefault();

App.CurrentUser = currentUser;

if (user\_log[0].role == 2)

{

AdminWindow adminWindow = new AdminWindow();

adminWindow.Show();

}

else

Продолжение листинга а.2

{

if (user\_log[0].role.ToString().Trim() == "")

{

GuestWindow gWindow = new GuestWindow(1);

gWindow.Show();

}

else

{

GuestWindow gWindow = new GuestWindow(1);

gWindow.Show();

}

}

Window.GetWindow(this).Close();

}

else

{

txtError.Text = "Неправильный пароль. Попробуйте снова.";

}

}

else

{

txtError.Text = "Неправильный логин. Попробуйте снова.";

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void txtLogin\_TextChanged(object sender, TextChangedEventArgs e)

{

txtError.Text = "";

}

private void txtPsw\_PasswordChanged(object sender, RoutedEventArgs e)

{

txtError.Text = "";

}

}

Листинг а.3 – Код окна для гостя

public partial class GuestWindow : Window

Продолжение листинга а.3

{

public GuestWindow(int x)

{

InitializeComponent();

if (FrameGuest.Content != null && FrameGuest.Content.ToString() == "SoundStudio.Pages.AddEditApps")

{

btnAdd.Visibility = Visibility.Hidden;

}

if (x == 0)

{

FrameGuest.Navigate(new Guestpage());

}

else if (x == 1)

{

FrameGuest.Navigate(new Homepage());

}

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

if (FrameGuest.CanGoBack)

FrameGuest.GoBack();

}

private void FrameGuest\_ContentRendered(object sender, EventArgs e)

{

}

private void btnExit\_Click(object sender, RoutedEventArgs e)

{

App.CurrentUser = null;

AuthorizationWindow authorizationwindow = new AuthorizationWindow();

authorizationwindow.Show();

this.Close();

}

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

FrameGuest.Navigate(new AddEditPage()); }

private void Image\_MouseDown (object sender, MouseButtonEventArgs e){

if(role != 0) {

FrameGuest.Navigate(new Homepage());

}

}

}

Листинг а.4 – Код страницы для гостя

public partial class Guestpage : Page

{

public Guestpage()

{

InitializeComponent();

TypesLoad();

}

public void TypesLoad()

{

var types = App.Context.ApplicationTypes.ToList();

LVTypes.ItemsSource = null;

LVTypes.ItemsSource = types;

}

private void btnEdit\_Click(object sender, RoutedEventArgs e)

{

}

}

Листинг а.5 – Код страницы вывода заявок

List<Applications> apps;

public Homepage()

{

InitializeComponent();

var types = App.Context.ApplicationTypes.Select(x => x.app\_type).ToList();

foreach (var type in types)

{

cbChooseType.Items.Add(type);

}

cbChooseType.SelectedIndex = 0;

cbSortNumCl.SelectedIndex = 0;

if (App.CurrentUser.role == 2)

{

apps = App.Context.Applications.ToList();

}

else

{

apps = App.Context.Applications.Where(a => a.client == App.CurrentUser.id\_user).ToList();

}

AppsLoad(apps);

}

public void AppsLoad(List<Applications> apps)

{

txtError.Visibility = Visibility.Hidden;

Продолжение листинга а.5

switch (cbChooseType.SelectedIndex)

{

case 0:

apps = App.Context.Applications.ToList();

break;

case 1:

apps = apps.Where(a => a.app\_type == 1).ToList();

break;

case 2:

apps = apps.Where(a => a.app\_type == 2).ToList();

break;

case 3:

apps = apps.Where(a => a.app\_type == 3).ToList();

break;

case 4:

apps = apps.Where(a => a.app\_type == 4).ToList();

break;

case 5:

apps = apps.Where(a => a.app\_type == 5).ToList();

break;

}

switch (cbSortNumCl.SelectedIndex)

{

case 0:

apps = apps.OrderBy(a => a.id\_app).ToList();

break;

case 1:

apps = apps.OrderByDescending(a => a.id\_app).ToList();

break;

case 2:

apps = apps.OrderBy(a => a.client).ToList();

break;

case 3:

apps = apps.OrderByDescending(a => a.client).ToList();

break;

}

apps = apps.Where(a => a.userName.ToLower().Contains(tbxSearch.Text.ToLower().Trim())).ToList();

LVApps.ItemsSource = null;

LVApps.ItemsSource = apps;

if (LVApps.Items.Count == 0)

{

txtError.Visibility= Visibility.Visible;

txtError.Text = "Заявок нет";

}

}

Продолжение листинга а.5

private void btnEdit\_Click(object sender, RoutedEventArgs e)

{

var button = sender as Button;

var app = button.DataContext as Applications;

NavigationService.Navigate(new AddEditPage(app));

}

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

var button = sender as Button;

var currentApp = button.DataContext as Applications;

if (MessageBox.Show($"Вы точно хотите удалить эту заявку?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

App.Context.Applications.Remove(currentApp);

App.Context.SaveChanges();

MessageBox.Show("Заявка была удалена");

}

AppsLoad(apps);

}

private void cb\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

AppsLoad(apps);

}

private void tbxSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

AppsLoad(apps);

}

}

Листинг а.6 – Код страницы добавления и редактирования заявок

public partial class AddEditPage : Page

{

public Applications current\_app = null;

public AddEditPage(Applications x)

{

InitializeComponent();

if(App.CurrentUser.role == 1)

{

txtClient.Visibility = Visibility.Hidden;

cbClients.Visibility = Visibility.Hidden;

txtStatus.Visibility = Visibility.Hidden;

cbStatuses.Visibility = Visibility.Hidden;

}

current\_app = x;

InfoLoad();

Продолжение листинга а.6

ComboBoxesLoad(x);

}

public AddEditPage()

{

InitializeComponent();

txtIdApp.Visibility = Visibility.Hidden;

if(App.CurrentUser.role == 1)

{

txtClient.Visibility = Visibility.Hidden;

cbClients.Visibility = Visibility.Hidden;

txtStatus.Visibility = Visibility.Hidden;

cbStatuses.Visibility = Visibility.Hidden;

}

InfoLoad();

}

public void InfoLoad()

{

cbClients.ItemsSource = App.Context.Users.Select(c => c.login).ToList();

cbTypes.ItemsSource = App.Context.ApplicationTypes.Select(c => c.app\_type).ToList();

cbStatuses.ItemsSource = App.Context.ApplicationStatuses.Select(c => c.app\_status).ToList();

}

public void ComboBoxesLoad(Applications x)

{

var app = App.Context.Applications.Where(a => a.id\_app == x.id\_app).FirstOrDefault();

var client = App.Context.Users.Where(c => c.id\_user == app.client).FirstOrDefault();

var type = App.Context.ApplicationTypes.Where(c => c.id\_apptype == app.app\_type).FirstOrDefault();

var status = App.Context.ApplicationStatuses.Where(c => c.id\_appstatus == app.app\_status).FirstOrDefault();

txtIdAppNum.Text = "№" + app.id\_app.ToString();

cbClients.SelectedIndex = client.id\_user - 1;

cbTypes.SelectedIndex = type.id\_apptype - 1;

txtQuantuty.Text = app.quantity.ToString();

cbStatuses.SelectedIndex = status.id\_appstatus - 1;

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (cbTypes.Text != "")

{

if (Int32.TryParse(txtQuantuty.Text, out int x))

Продолжение листинга а.6

{

int id\_client, id\_status;

if (App.CurrentUser.role == 1)

{

id\_client = App.CurrentUser.id\_user;

id\_status = 1;

}

else

{

var user = App.Context.Users.Where(u => u.login == cbClients.Text).FirstOrDefault();

id\_client = user.id\_user;

var status = App.Context.ApplicationStatuses.Where(s => s.app\_status == cbStatuses.Text).FirstOrDefault();

id\_status = status.id\_appstatus;

}

var type = App.Context.ApplicationTypes.Where(t => t.app\_type == cbTypes.Text).FirstOrDefault();

int id\_type = type.id\_apptype;

var new\_app = new Applications

{

client = id\_client,

app\_type = id\_type,

quantity = Int32.Parse(txtQuantuty.Text),

app\_status = id\_status

};

if (txtIdAppNum.Text.ToString() == "")

{

App.Context.Applications.Add(new\_app);

App.Context.SaveChanges();

MessageBox.Show("Заявка успешно добавлена");

}

else

{

current\_app.client = id\_client;

current\_app.app\_type = id\_type;

current\_app.quantity = Int32.Parse(txtQuantuty.Text);

current\_app.app\_status = id\_status;

App.Context.SaveChanges();

MessageBox.Show("Данные по заявке успешно обновлены");

}

NavigationService.Navigate(new Homepage());

}

else

{

MessageBox.Show("В поле 'Количество' введено не число");

Продолжение листинга а.6

}

}

else

{

MessageBox.Show("В поле 'Услуга' ничего не выбрано");

}

}

private void txtQuantuty\_TextChanged(object sender, TextChangedEventArgs e)

{

if (Int32.TryParse(txtQuantuty.Text, out int x) == true && cbTypes.SelectedItem.ToString() != "")

{

var type = App.Context.ApplicationTypes.Where(t => t.app\_type == cbTypes.SelectedItem.ToString()).FirstOrDefault();

Amountcount.Text = (int.Parse(txtQuantuty.Text) \* type.cost).ToString();

}

}

}

Листинг а.7 – Код удаления заявки

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

var button = sender as Button;

var currentApp = button.DataContext as Applications;

if (MessageBox.Show($"Вы точно хотите удалить эту заявку?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

App.Context.Applications.Remove(currentApp);

App.Context.SaveChanges();

MessageBox.Show("Заявка была удалена");

}

AppsLoad(apps);

}

Приложение Б

Листинг б.1 – Методы для тестирования

public class ProjectChecking

{

public static bool ValidateUser(string username, string password)

{

if (username == "" || password == "") { return false; }

else { return true; }

}

public static bool BtnCheck(bool btn\_click)

{

if (!btn\_click) { return false; } else { return true; }

}

public static bool ValidateQuantity(string quantuty)

{

if(int.TryParse(quantuty, out int q)) { return true; } else { return false;}

}

}