**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по учебной практике УП.02.01 (по профилю специальности)**

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 42919/7 группы

Романова Анна Андреевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

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

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на учебную практику (по профилю специальности)**

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 42919/7 группы

Романова Анна Андреевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

**Виды работ, обязательные для выполнения** *(переносится из программы, соответствующего ПМ):*

* Участие в выработке требований к программному обеспечению;
* Стадии проектирования программного обеспечения. Проектирование UML-диаграмм;
* Конструирование пользовательского интерфейса. Разработка модулей программного обеспечения;
* Разработка рабочего проекта и технологической документации.

**Индивидуальное задание: ВАРИАНТ 15**

Задание выдал «28» октября 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_ Хисамутдинова А.С.

(подпись) (Ф.И.О.)

Задание получил «28» октября 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_ Романова А.А (подпись) (Ф.И.О.)

**МИНОБРНАУКИ РОССИИ**

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 профилю специальности)**

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 42919/7 группы

Романова Анна Андреевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

Руководитель практики Хисамутдинова А.С.

(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** |
| 28.10.2024 | Анализ предметной области. Анализ методов решения. Анализ и выбор средств. |  |
| 29.10.2024 | Анализ и выбор средств. Исследовательские работы. Разработка технического задания. |  |
| 30.10.2024 | UML. Проектирование диаграммы вариантов использования и последовательности. UML. Проектирование диаграммы вариантов активности и состояния. |  |
| 31.10.2024 | Моделирование структуры ПО. Проектирование инфологической и даталогической модели данных. |  |
| 01.11.2024 | Проектирование интерфейса пользователя. |  |
| 02.11.2024 | Разработка схемы алгоритма программного продукта. |  |
| 05.11.2024 | Разработка модели данных. Разработка программного продукта. Создание функциональности ПО. |  |
| 06.11.2024 | Модульное тестирование. Создание тестовых случаев. |  |
| 07.11.2024 | Создание тестовых случаев. Отладка программных модулей. |  |
| 08.11.2024 | Разработка документа «Руководство пользователя» в соответствии с ЕСПД. Разработка документа «Программа и методика испытаний» в соответствии с ЕСПД. |  |
| 09.11.2024 | Подготовка отчета. |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc181920586)

[Задание № 1 «Разработка технического задания» 7](#_Toc181920587)

[Задание № 2 «Разработка алгоритмов и диаграмм» 11](#_Toc181920588)

[Задание № 3 «Разработка макетов программы» 22](#_Toc181920589)

[Задание № 4 «Разработка программного продукта» 30](#_Toc181920590)

[Задание № 5 «Тестирование программы» 36](#_Toc181920591)

[Задание № 6 «Выгрузка проекта на GitHub» 44](#_Toc181920592)

[ЗАКЛЮЧЕНИЕ 45](#_Toc181920593)

ВВЕДЕНИЕ

В условиях быстро растущего рынка услуг по ремонту бытовой техники автоматизация процессов учета и обработки заявок становится важным аспектом для повышения эффективности работы сервисных центров. Моя цель в данной практической работе – разработать информационную систему для ремонта и обслуживания бытовой техники, которая позволит не только упростить процесс управления заявками, но и значительно улучшить качество обслуживания клиентов.

Предметной областью данной системы является автоматизация учета заявок на ремонт бытовой техники, что подразумевает создание интерфейса для взаимодействия клиентов с сервисом, а также функционал для сотрудников, позволяющий эффективно управлять потоком заявок. Основная задача разрабатываемого модуля заключается в регистрации, хранении и мониторинге заявок, что даст возможность отслеживать текущее состояние каждого ремонта. Такой подход не только облегчает внутренние процессы, но и способствует повышению удовлетворенности клиентов благодаря прозрачности и быстроте обработки их запросов.

Задание № 1 «Разработка технического задания»

Описание предметной области

Основной целью модуля является автоматизация процесса учета, обработки и мониторинга заявок на ремонт бытовой техники. Это позволяет мастерам и сотрудникам сервисных центров эффективно управлять потоком заявок и отслеживать статус выполнения каждого ремонта.

Разработка программного модуля для учета заявок на ремонт бытовой техники — это процесс создания программного продукта, который позволяет сервису по ремонту бытовой техники эффективно отслеживать и ввести учет заявок на ремонт бытовой техники.

Основной задачей программного модуля является автоматизация процесса регистрации и хранения информации о заявках на ремонт, возможность отслеживания текущего состояния каждого ремонта, генерация отчетов для оценки производительности и качества оказанного сервиса.

Эффективная реализация данного программного модуля позволят улучшить внутренние процессы, повышает результативность работы и способствует удовлетворенности клиентов на рынке услуг по ремонту бытовой техники.

Основные функции и возможности модуля включают:

1. Заявка на ремонт: это информация, предоставленная клиентом о неисправности бытовой техники, который требует ремонта. Клиент оставляет новую заявку. Заявка может содержать данные о виде бытовой техники, модели, описании проблемы, личную информацию (ФИО клиента и номер телефона).

2. Регистрация заявки: этот процесс включает приём и регистрацию заявки оператором сервиса в системе учёта. Важными аспектами регистрации являются присвоение уникального идентификатора заявке, сохранение информации о заявке.

3. Обработка заявки: процесс, включающий анализ заявки, определение её приоритетности и назначение механика для ремонта.

4. Исполнение заявки: фактическое выполнение ремонта бытовой техники. На этом этапе назначенный механик ремонтирует технику, вносит необходимые изменения в статус заявки, заменяет неисправные запчасти.

5. Отчётность и информирование: важной составляющей учёта заявок на ремонт является фиксация и отчёт о выполненной работе. После завершения ремонта механик должен предоставить отчёт о проделанной работе, включая информацию о затраченных ресурсах (время, запчастях) и оказанной помощи.

6. Мониторинг и анализ: этот этап предполагает контроль и анализ процесса учёта заявок на ремонт. Важно отслеживать и анализировать время обработки заявок, качество выполненных работ, расходы и прочие параметры, которые могут помочь в оптимизации и улучшении процесса.

Техническое задание

1. Общие сведения

1.1. Наименование проекта: Разработка программного модуля для учета заявок на ремонт и обслуживание бытовой техники.

1.2. Заказчик: Старикова Яна Евгеньевна.

1.3. Исполнитель: Романова Анна Андреевна.

2. Функциональные требования

2.1. Возможность добавления заявок в базу данных с указанием следующих параметров:

- Номер заявки;

- Дата добавления;

- Вид бытовой техники;

- Модель бытовой техники;

- Описание проблемы;

- ФИО клиента;

- Номер телефона;

- Статус заявки (новая заявка, в процессе ремонта, завершена).

2.2. Возможность редактирования заявок:

- Изменение этапа выполнения (готова к выдаче, в процессе ремонта, ожидание автозапчастей);

- Изменение описания проблемы;

- Изменение ответственного за выполнение работ.

2.3. Возможность отслеживания статуса заявки:

- Отображение списка заявок;

- Получение уведомлений о смене статуса заявки;

- Поиск заявки по номеру или по параметрам.

2.4. Возможность назначения ответственных за выполнение работ:

- Добавление механика к заявке;

- Отслеживание состояния работы и получение уведомлений о ее завершении;

- Механик может добавлять комментарии на форме заявки и фиксировать информацию о заказанных запчастях и материалах.

2.5. Расчет статистики работы отдела обслуживания:

- Количество выполненных заявок;

- Среднее время выполнения заявки.

3. Нефункциональные требования

3.1. Кроссплатформенность:

- Поддержка работы на ОС семейства Windows.

3.2. Безопасность:

- Логин и пароль для доступа к приложению;

- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

3.3. Удобство использования:

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.

3.4. Производительность:

- Приложение должно иметь быстрый доступ к данным;

- Минимальное время отклика на запросы пользователя.

4. Требования к реализации

4.1. Язык программирования: C#

4.2. СУБД: MS SQL

5. Требования к документации

5.1. Техническое задание на разработку программного модуля.

6. Руководство по стилю

6.1. Шрифт: GOST type B.

6.2. Цветовая схема: #b8d8d8 – цвет фона, #eef5db – цвет текстового поля, #000000 – цвет текста, #fe5f55 – цвет текста в полях, #7a9e9f, #e5a679– остальные компоненты формы.

Задание № 2 «Разработка алгоритмов и диаграмм»

На изображении ниже представлена диаграмма вариантов использования. Она помогает идентифицировать различные пути взаимодействия пользователя с системой и является отличным способом визуализации функциональности продукта (Рисунок 1).



Рисунок 1 – диаграмма вариантов использования

Ниже приведена диаграмма последовательности. Она позволяет наглядно отобразить порядок выполнения операций между объектами или участниками в процессе (Рисунок 2).



Рисунок 2 – диаграмма последовательности

Также нужно изобразить диаграмму активности. На ней отображаются шаги, решения и потоки управления в удобной визуальной форме (Рисунок 3).



Рисунок 3 – диаграмма активности

На изображение ниже показана ER-диаграмма. На такой диаграмме отображаются сущности (таблицы), атрибуты (поля) и связи между ними для наглядного понимания структуры базы данных (Рисунок 4).

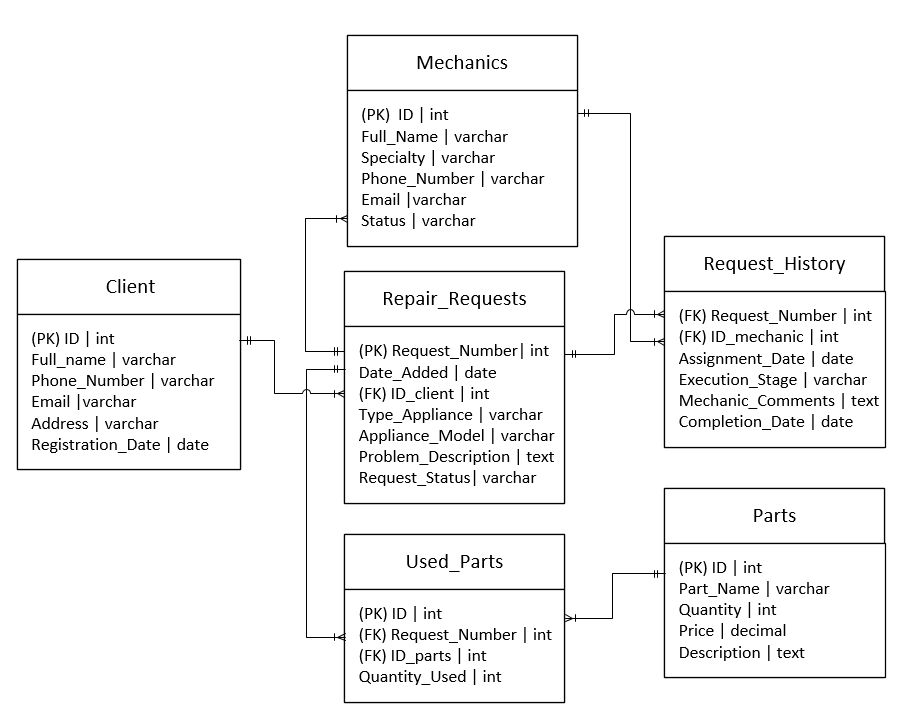


Рисунок 4 – ER-диаграмма

Таблица 1. Словарь данных

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Столбец | Тип данных | Описание |
| Mechanics | ID (PK) | INT | Уникальный идентификатор механика |
| Full\_Name | VARCHAR | Полное имя механика |
| Specialty | VARCHAR | Специальность механика |
| Phone | VARCHAR | Номер телефона механика |
| Email | VARCHAR | Адрес электронной почты механика |
| Status | VARCHAR | Статус механика (активен/неактивен) |
| Client | ID (PK) | INT | Уникальный идентификатор клиента |
| Full\_Name | VARCHAR | Полное имя клиента |
| Phone\_Number | VARCHAR | Номер телефона клиента |
| Email | VARCHAR | Адрес электронной почты клиента |
| Address | VARCHAR | Адрес клиента |
| Registration\_Date | DATE | Дата регистрации клиента |
| RepairRequests | Request\_Number (PK) | INT | Уникальный номер заявки на ремонт |
| Date\_Added | DATE | Дата добавления заявки |
| ID\_client (FK) | INT | Идентификатор клиента (ссылка на Client) |
| Type\_of\_Appliance | VARCHAR | Тип бытовой техники |
| Appliance\_Model | VARCHAR | Модель бытовой техники |
| Problem\_Description | TEXT | Описание проблемы |
| Request\_Status | VARCHAR | Статус заявки |
| RequestHistory | Request\_Number (FK) | INT | Номер заявки (ссылка на RepairRequests) |
| ID\_mechanic (FK) | INT | Идентификатор механика (ссылка на Mechanics) |
| Assignment\_Date | DATE | Дата назначения механика |
| Execution\_Stage | VARCHAR | Этап выполнения |
| Mechanic\_Comments | TEXT | Комментарии механика |
| Completion\_Date | DATE | Дата завершения |
| Parts | ID\_parts (PK) | INT | Уникальный идентификатор запчасти |
| Part\_Name | VARCHAR | Наименование запчасти |
| Quantity\_In\_Stock | INT | Количество запчастей в наличии |
| Price | DECIMAL | Цена запчасти |
| Description | TEXT | Описание запчасти |
| Used\_Parts | ID\_record (PK) | INT | Уникальный идентификатор записи |
| Request\_Number (FK) | INT | Номер заявки (ссылка на Repair\_Requests) |
| ID\_parts (FK) | INT | Идентификатор запчасти (ссылка на Parts) |
| Quantity\_Used | INT | Количество использованных запчастей |

Данные SQL-запросы создают таблицы с соответствующими столбцами, включающими первичные и внешние ключи, а также различные ограничения. Каждая таблица заполняется 5 строками тестовых данных.

Создание таблицы Механики

CREATE TABLE Mechanics (

ID\_mechanic INT IDENTITY(1,1) PRIMARY KEY,

Full\_Name VARCHAR(255) NOT NULL,

Specialty VARCHAR(255) NOT NULL,

Phone VARCHAR(15),

Email VARCHAR(255),

Login VARCHAR(50) NOT NULL,

Password VARCHAR(255) NOT NULL,

Role VARCHAR(20) DEFAULT 'сотрудник'

);

Вставка данных в таблицу Механики

INSERT INTO Mechanics (Full\_Name, Specialty, Phone, Email, Login, Password, Role) VALUES

('Иванов Иван Иванович', 'Электрик', '89001234567', 'ivanov@example.com', 'ivanov', 'password123', 'сотрудник'),

('Петров Петр Петрович', 'Механик', '89007654321', 'petrov@example.com', 'petrov', 'mypassword', 'сотрудник'),

('Сидоров Сидор Сидорович', 'Слесарь', '89002345678', 'sidorov@example.com','sidorov', '12345678', 'сотрудник'),

('Кузнецов Николай Николаевич', 'Ремонтник', '89003456789', 'kuznetsov@example.com', 'kuznetsov', 'qwerty', 'сотрудник'),

('Сафонов Алексей Александрович', 'Инженер', '89009876543', 'safonov@example.com', 'safonov', 'admin123', 'сотрудник');

Результат создания таблицы Механики (Рисунок 5).

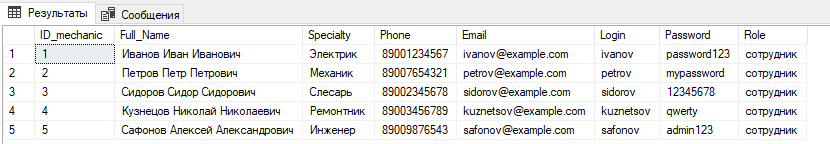


Рисунок 5 – таблица Механики

Создание таблицы Клиенты

CREATE TABLE Clients (

ID\_client INT PRIMARY KEY,

Full\_Name VARCHAR(255) NOT NULL,

Phone\_Number VARCHAR(15) NOT NULL,

Email VARCHAR(255),

Address VARCHAR(255),

Login VARCHAR(50) NOT NULL,

Password VARCHAR(255) NOT NULL,

Role VARCHAR(20) DEFAULT 'клиент'

);

Вставка данных в таблицу Клиенты

INSERT INTO Clients (ID\_client, Full\_Name, Phone\_Number, Email, Address, Login, Password, Role) VALUES

(1, 'Александр Смирнов', '89005551234', 'smirnov@example.com', 'Москва, ул. Ленина', 'smirnov', 'password888', 'клиент'),

(2, 'Мария Иванова', '89007775678', 'ivanova@example.com', 'Питер, пр. Невский', 'ivanova', 'mypassword234', 'клиент'),

(3, 'Елена Петрова', '89009990000', 'petrova@example.com', 'Казань, ул. Баумана', 'petrova', '123456782334d', 'клиент'),

(4, 'Дмитрий Кузнецов', '89002223344', 'kuznetsov@example.com', 'Новосибирск, ул. Красный проспект', 'kuznetsov', 'qweadty553', 'клиент'),

(5, 'Татьяна Соколова', '89003334455', 'sokolova@example.com', 'Екатеринбург, ул. Малышева', 'sokolova', '12dfg3', 'клиент');

Результат создания таблицы Клиенты (Рисунок 6).

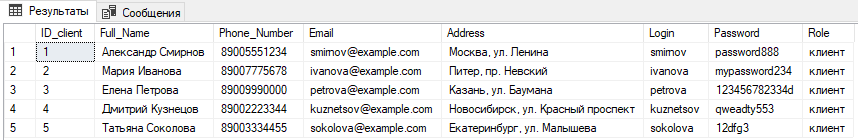


Рисунок 6 – таблица Клиенты

Создание таблицы Заявки на ремонт

CREATE TABLE Repair\_Requests (

Request\_Number INT IDENTITY(1,1) PRIMARY KEY,

ID\_client INT,

Type\_of\_Appliance VARCHAR(255) NOT NULL,

Appliance\_Model VARCHAR(255) NOT NULL,

Problem\_Description TEXT NOT NULL,

Request\_Status VARCHAR(20),

FOREIGN KEY (ID\_client) REFERENCES Clients(ID\_client)

);

Вставка данных в таблицу Заявки на ремонт

INSERT INTO Repair\_Requests (ID\_client, Type\_of\_Appliance, Appliance\_Model, Problem\_Description, Request\_Status) VALUES

(1,'Стиральная машина', 'LG 12345', 'Не включается', 'новая'),

(2, 'Холодильник', 'Samsung RCA', 'Шумит', 'новая'),

(3, 'Телевизор', 'Sony LCD', 'Нет изображения', 'новая'),

(4, 'Микроволновая печь', 'Bosch M20', 'Не греет', 'в процессе'),

(5, 'Пылесос', 'Philips FC', 'Слабая мощность', 'завершена');

Результат создания Заявки на ремонт (Рисунок 7).

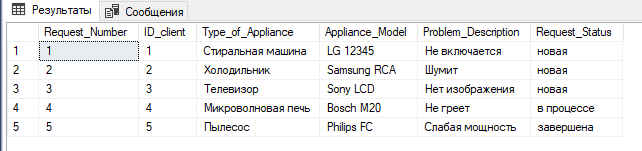


Рисунок 7 – таблица Заявки на ремонт

Создание таблицы История обработки заявок

CREATE TABLE Request\_History (

Request\_Number INT,

ID\_mechanic INT,

Assignment\_Date DATE NOT NULL,

Execution\_Stage VARCHAR(30) NOT NULL,

Mechanic\_Comments TEXT,

Completion\_Date DATE,

PRIMARY KEY (Request\_Number, ID\_mechanic),

FOREIGN KEY (Request\_Number) REFERENCES Repair\_Requests(Request\_Number),

FOREIGN KEY (ID\_mechanic) REFERENCES Mechanics(ID\_mechanic)

);

Вставка данных в таблицу История обработки заявок

INSERT INTO Request\_History (Request\_Number, ID\_mechanic, Assignment\_Date, Execution\_Stage, Mechanic\_Comments, Completion\_Date) VALUES

(1, 1, '2023-01-16', 'в процессе ремонта', 'Проверка нагревательного элемента', NULL),

(2, 2, '2023-02-21', 'ожидание запчастей', 'Заказаны запчасти', NULL),

(3, 3, '2023-03-26', 'в процессе ремонта', 'Диагностика проблемы', '2023-04-01'),

(4, 4, '2023-04-01', 'готова к выдаче', 'Ремонт завершен', '2023-04-05'),

(5, 5, '2023-05-06', 'в процессе ремонта', 'Работа продолжается', NULL);

Результат создания таблицы История обработки заявок (Рисунок 8).

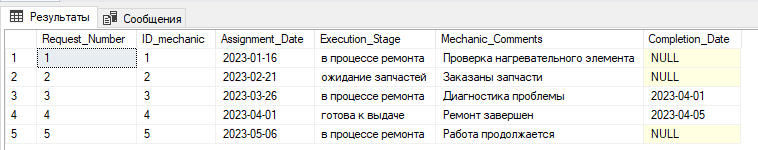


Рисунок 8 – таблица История обработки заявок

Создание таблицы Запчасти

CREATE TABLE Parts (

ID\_parts INT IDENTITY(1,1) PRIMARY KEY,

Part\_Name VARCHAR(255) NOT NULL,

Quantity\_In\_Stock INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

Description TEXT

);

Вставка данных в таблицу Запчасти

INSERT INTO Parts (Part\_Name, Quantity\_In\_Stock, Price, Description) VALUES

('Модуль управления', 10, 5000.00, 'Контрольная плата для стиральных машин'),

('Компрессор', 5, 15000.00, 'Компрессор для холодильников'),

('Экран', 8, 3000.00, 'Экран для телевизоров Sony'),

('Картридж', 15, 1200.00, 'Картридж для пылесосов Philips'),

('Микроволновая печь', 2, 6000.00, 'Запчасть для микроволновых печей Bosch');

Результат создания таблицы Запчасти (Рисунок 9).



Рисунок 9 – таблица Запчасти

Создание таблицы Используемые запчасти

CREATE TABLE Used\_Parts (

ID\_record INT IDENTITY(1,1) PRIMARY KEY,

Request\_Number INT,

ID\_parts INT,

Quantity\_Used INT NOT NULL,

FOREIGN KEY (Request\_Number) REFERENCES Repair\_Requests(Request\_Number),

FOREIGN KEY (ID\_parts) REFERENCES Parts(ID\_parts)

);

Вставка данных в таблицу Используемые запчасти

INSERT INTO Used\_Parts (Request\_Number, ID\_parts, Quantity\_Used) VALUES

(1, 1, 1),

(2, 2, 1),

(3, 3, 1),

(4, 4, 2),

(5, 1, 1);

Результат создания таблицы (Рисунок 10).

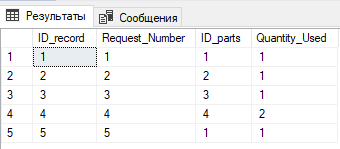


Рисунок 10 – таблица Используемые запчасти

Получившаяся база данных ремонт и обслуживание бытовой техники (Рисунок 11).

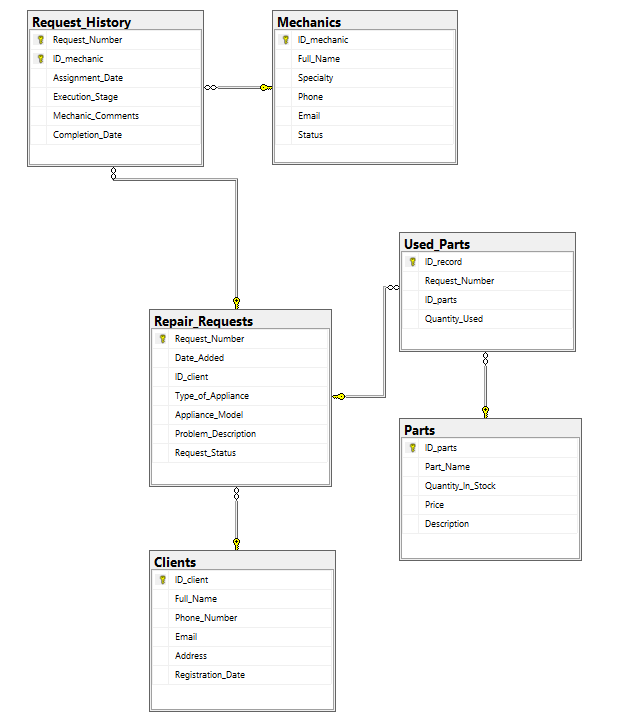


Рисунок 11 – база данных

Задание № 3 «Разработка макетов программы»

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. Разработать карту навигации приложения.

Для начала нужно разработать макеты Wireframe (черно-белые) в графическом редакторе Figma.

Макет формы авторизации (Рисунок 12).



Рисунок 12 – форма авторизации

Макет формы регистрации (Рисунок 13).



Рисунок 13 – форма регистрации

Макет формы для заполнения заявки (Рисунок 14).

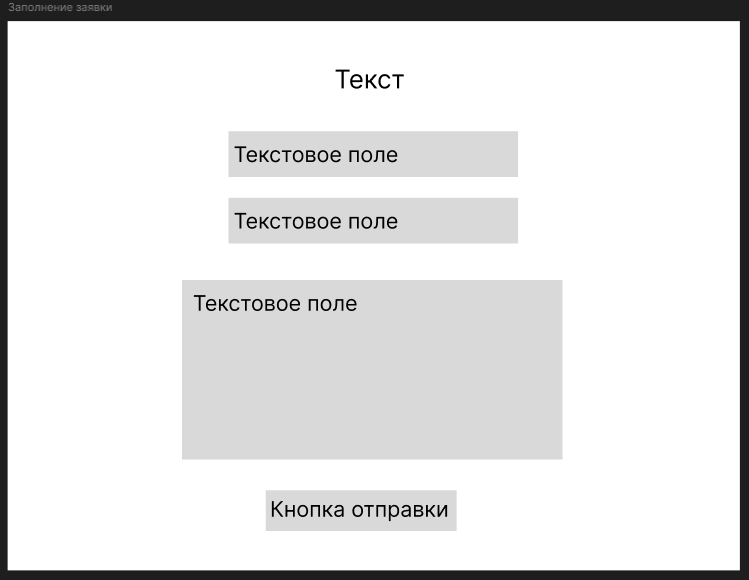


Рисунок 14 – заполнение заявки

Макет формы, которая будет отображаться у клиента при входе в приложение (Рисунок 15).

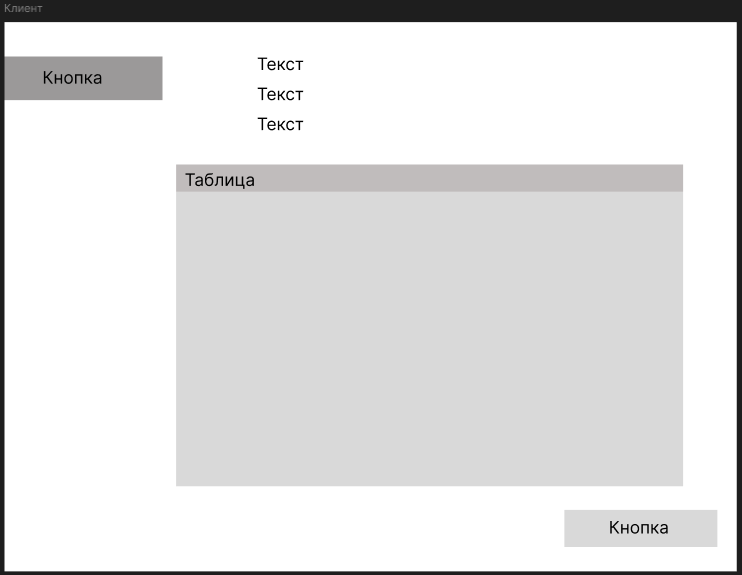


Рисунок 15 – форма для клиента

Макет формы, которая будет отображаться у механика при входе в приложение (Рисунок 16).

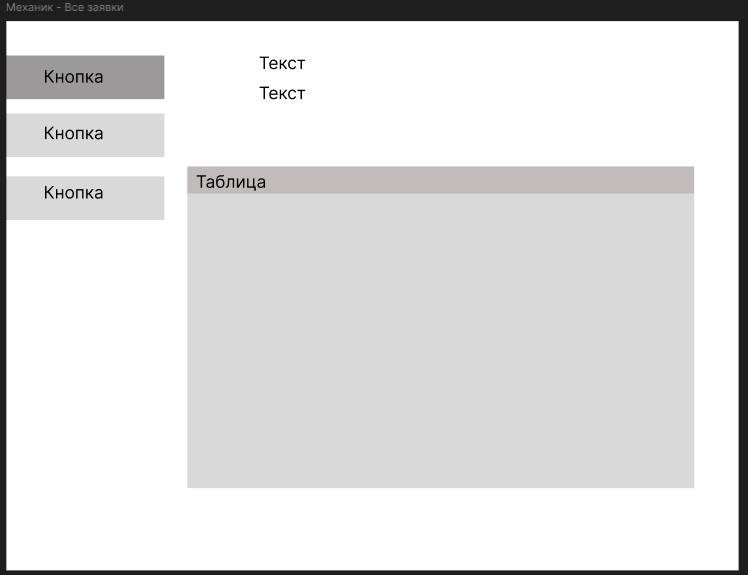


Рисунок 16 – форма для механика

Макет формы, которая будет отображаться у администратора при входе в приложение (Рисунок 17).

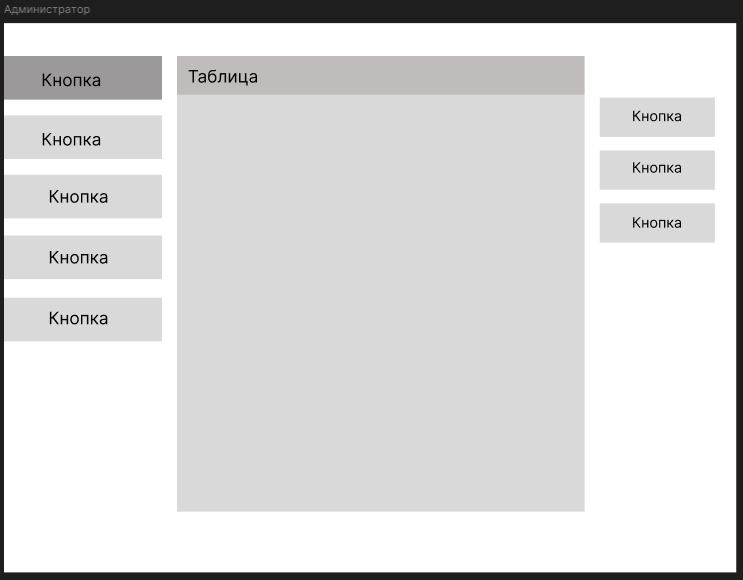


Рисунок 17 – форма для администратора

Далее нужно создать Mockup (цветные) макеты. Ниже представлен макет формы авторизации (Рисунок 18).



Рисунок 18 – форма авторизации

Макет формы регистрации (Рисунок 19).

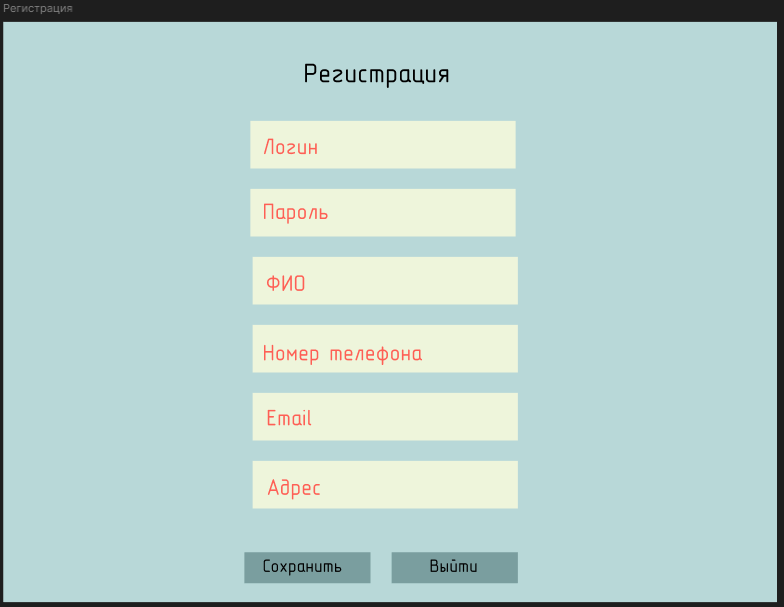


Рисунок 19 – форма регистрации

Макет формы заполнение заявки (Рисунок 20).

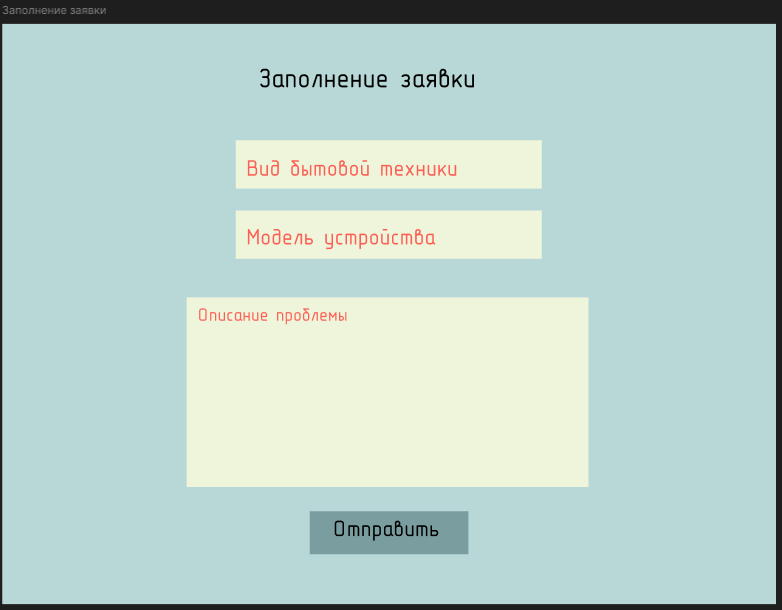


Рисунок 20 – заполнение заявки

Макет формы, которая будет отображаться при входе клиента в программу (Рисунок 21).



Рисунок 21 – форма для клиента

Макет формы, которая будет отображаться при входе механика в программу, в ней находятся три вкладки (Рисунок 22 - 24).



Рисунок 22 – вкладка все заявки

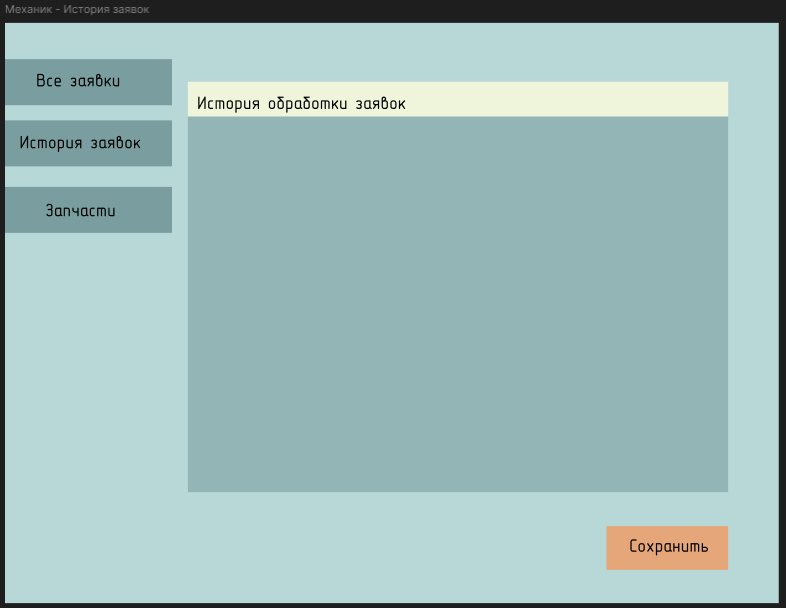


Рисунок 23 – вкладка история заявок

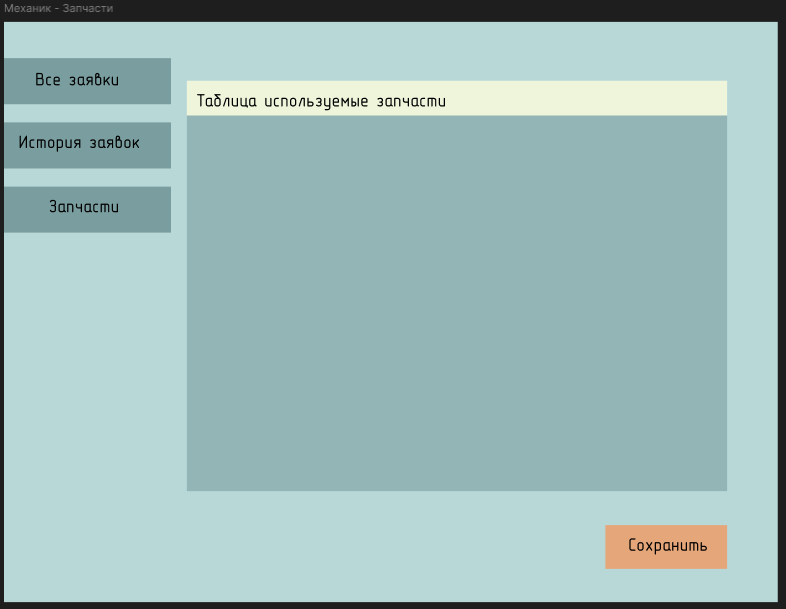


Рисунок 24 – вкладка запчасти

Макет формы, которая будет отображаться у администратора при входе в приложение (Рисунок 25).

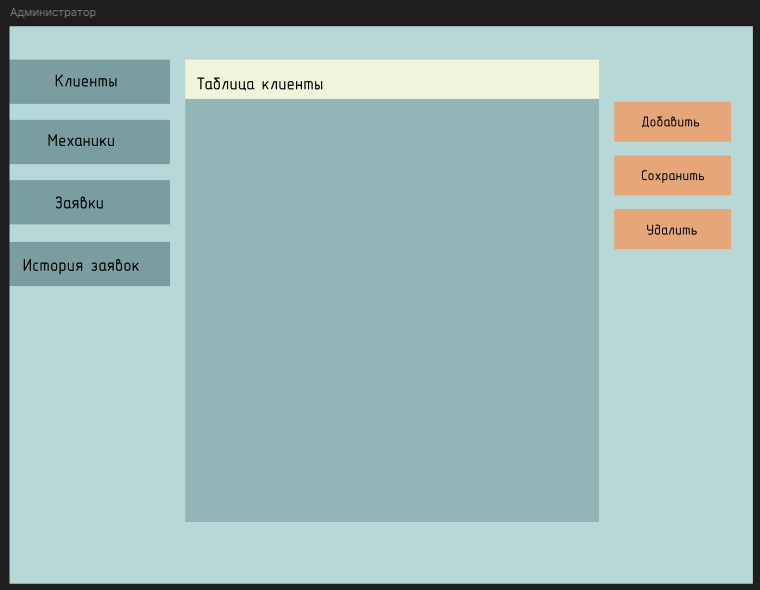


Рисунок 25 – форма для администратора

Следующим шагом необходимо разработать карту навигации приложения (Рисунок 26).

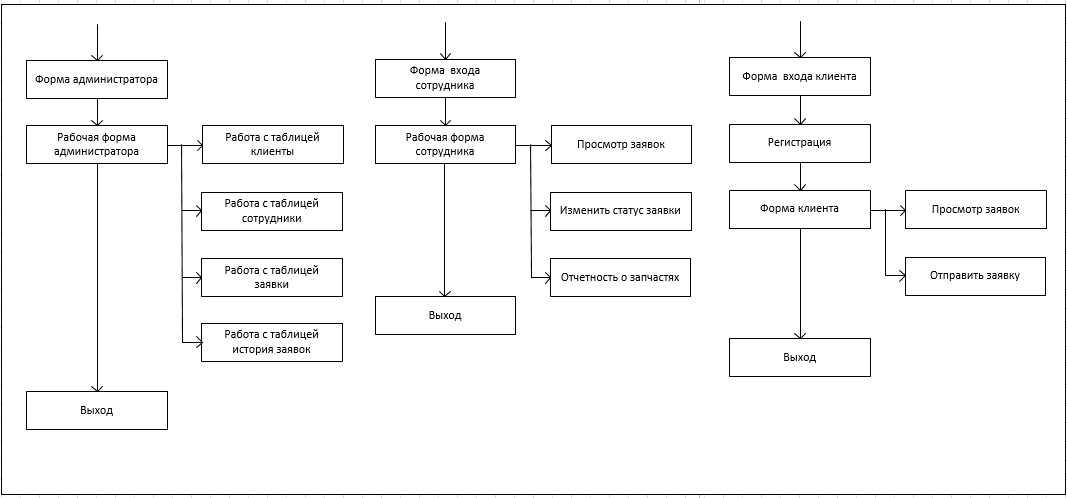


Рисунок 26 – карта навигации

Задание № 4 «Разработка программного продукта»

Для начала нужно разработать форму для авторизации, на которой можно будет перейти на форму регистрации для новых пользователей (Рисунок 27 - 28).

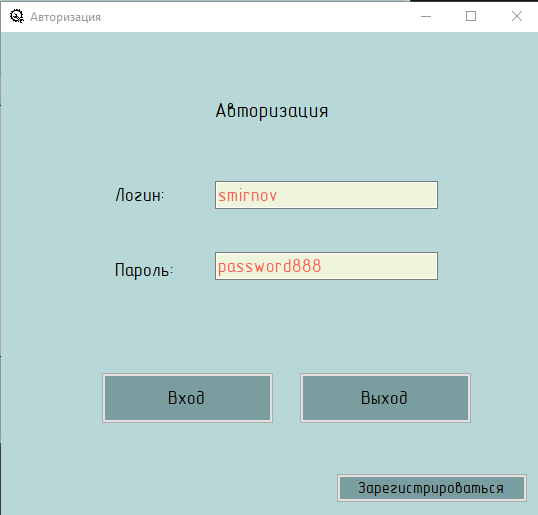


Рисунок 27 – авторизация

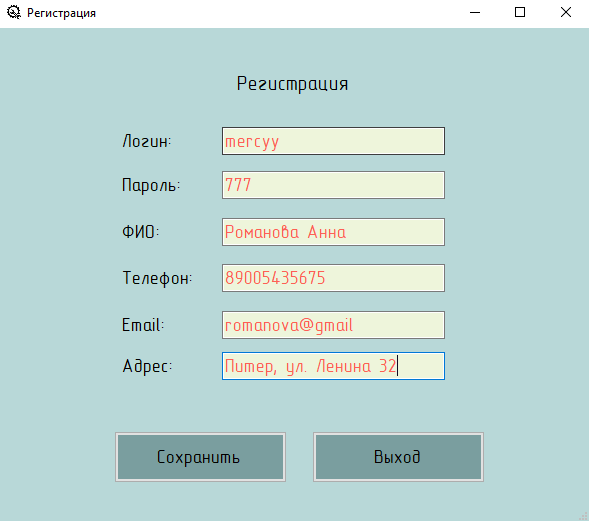


Рисунок 28 – регистрация

Если одно из полей не будет заполнено, программа выдаст предупреждение, о том, что требуется его заполнить (Рисунок 29).

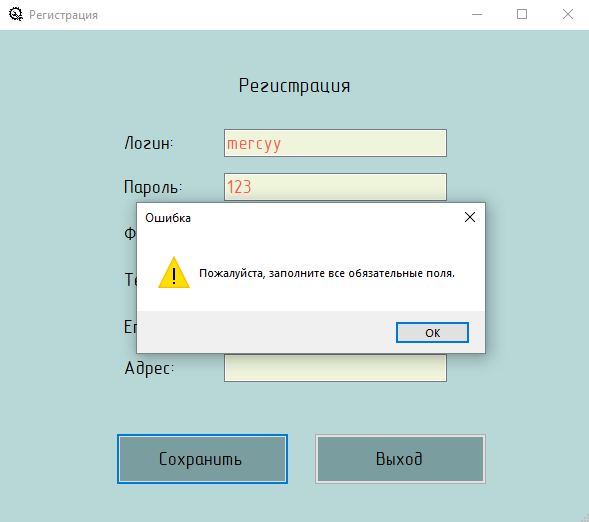


Рисунок 29 – вывод предупреждения

Следующим шагом, нужно разработать форму для пользователя, которая будет отображаться после его авторизации. На ней можно просмотреть заявки клиента, а также создать новую (Рисунок 30).

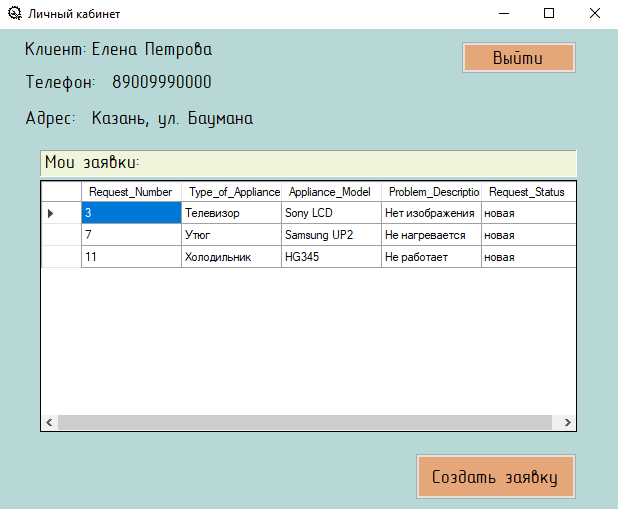


Рисунок 30 – форма пользователя

При создании заявки необходимо заполнить все поля, иначе будет выдано исключение (Рисунок 31 -32).

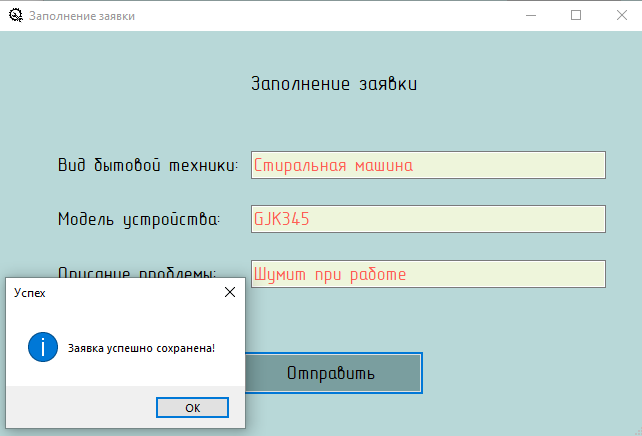


Рисунок 31 – успешное создание заявки

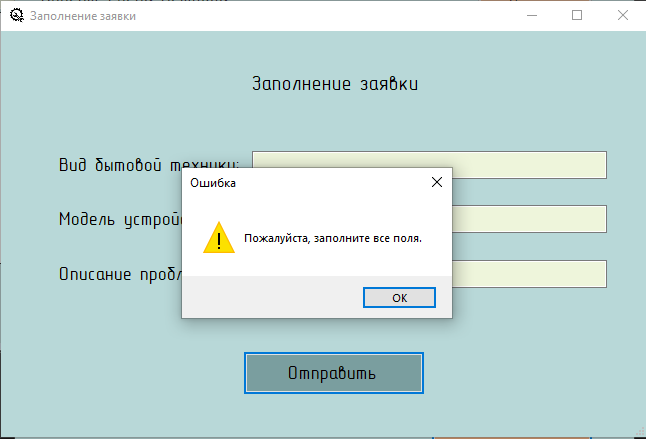


Рисунок 32 – вывод предупреждения

Далее нужно сделать форму для сотрудников, которая будет отображаться после авторизации. На данной форме сотруднику видны все заявки и таблица с запчастями (Рисунок 33).

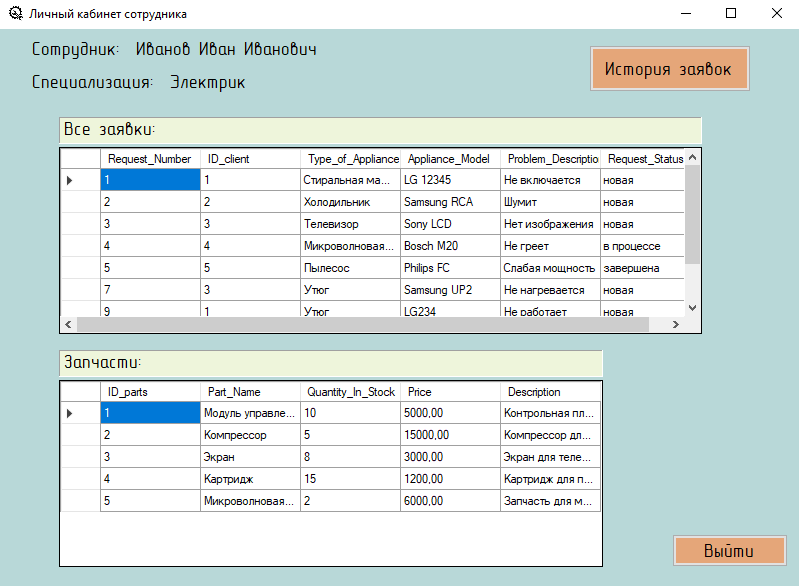


Рисунок 33 – форма сотрудника

С формы сотрудника при нажатии на кнопку «История заявок» осуществляется переход на форму история обработки заявок. Где можно отредактировать уже существующую заявку (Рисунок 34).

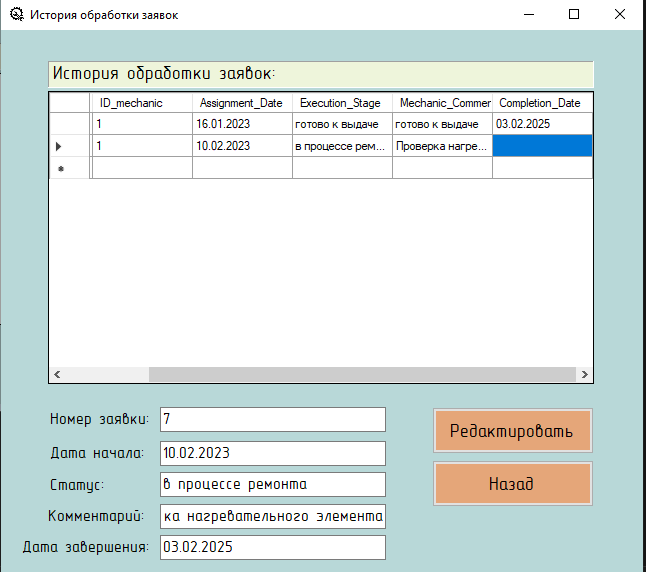


Рисунок 34 – форма истории заявок

Добавление даты завершения заявки (Рисунок 35).

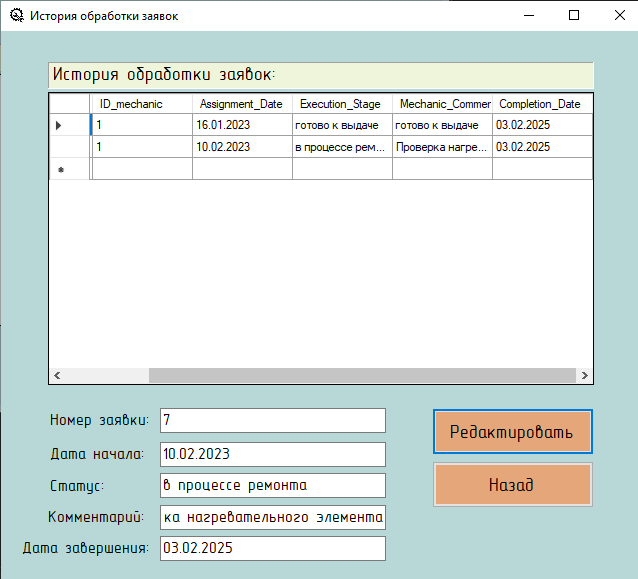


Рисунок 35 – редактирование заявки

Также необходимо сделать форму для администратора, который будет иметь возможность добавлять, обновлять, удалять данные в таблицах. Ещё нужно добавить возможность составить диаграмму, которую в последствии можно сохранить в файл (Рисунок 36).

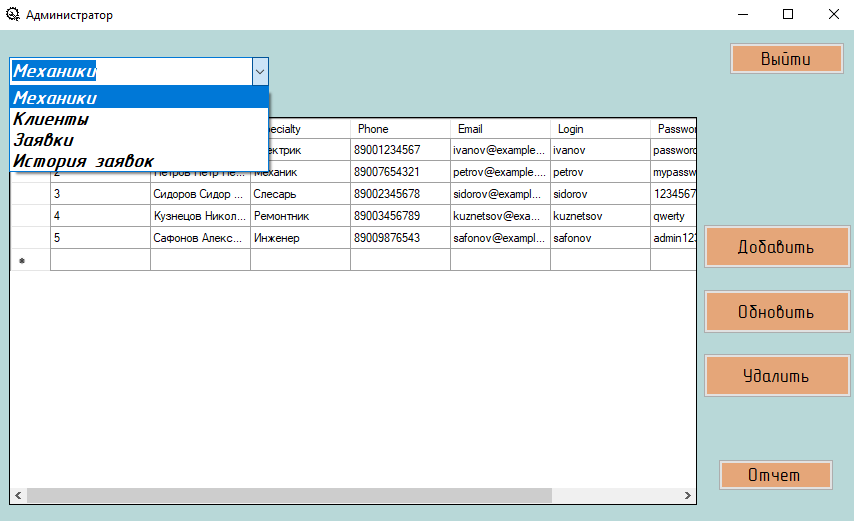


Рисунок 36 – форма администратора

При нажатии на кнопку «Отчет», открывается форма с диаграммой (Рисунок 37).

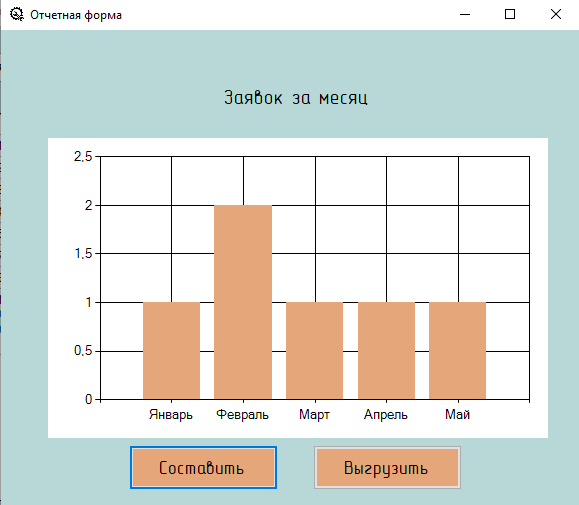


Рисунок 37 –отчетная форма

Задание № 5 «Тестирование программы»

Для тестирования данного программного продукта, необходимо подготовить 5 Test-Case, проверяющие один наиболее важный процесс в системе.

А также нужно реализовать 10 unit-тестов. Тестовые данные должны предусматривать различные ситуации.

Аннотация теста:

|  |  |
| --- | --- |
| **Название проекта** | Ремонт бытовой техники |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Романова Анна |
| **Дата(ы) теста** | 06.11.2024 - 07.11.2024 |

Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_LT\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка корректности отображения таблиц |
| **Краткое изложение теста** | Проверка, что форма корректно загружает и отображает таблицы БД на экране формы |
| **Этапы теста** | 1. Открытие формы тестирования; 2. Загрузка данных из таблицы БД; 3. Проверка отображения данных из таблицы БД на форме. |
| **Тестовые данные** | Таблицы базы данных |
| **Ожидаемый результат** | Успешное отображение таблиц из базы дынных на форме |
| **Фактический результат** | Успешное отображение таблиц из базы дынных на форме |
| **Статус** | Зачет |
| **Предварительное условие** |  |
| **Постусловие** |  |
| **Примечания/комментарии** | - |

Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UA\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка корректности авторизации пользователя |
| **Краткое изложение теста** | Данный тест проверяет процесс авторизации пользователя в системе на корректность |
| **Этапы теста** | 1. Запуск приложения; 2. Ввод корректного логина и пароля пользователя; 3. Нажатие кнопки «Вход»; 4. Проверка успешности авторизации. |
| **Тестовые данные** | Логин: smirnov  Пароль: password888 |
| **Ожидаемый результат** | Отображение основного окна пользователя |
| **Фактический результат** | Отображение основного окна пользователя |
| **Статус** | Зачет |
| **Предварительное условие** | Пользователь зарегистрирован в системе |
| **Постусловие** | Пользователь успешно авторизован в системе |
| **Примечания/комментарии** | - |

Тестовый пример #3:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_EC\_3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка функции выхода из системы |
| **Краткое изложение теста** | Проверка корректной работы кнопки выхода из аккаунта пользователя. |
| **Этапы теста** | 1. Нахождение в основном окне пользователя; 2. Нажатие на кнопку "Выход"; 3. Подтверждение действия выхода из системы. |
| **Тестовые данные** | Пользователь авторизован в системе |
| **Ожидаемый результат** | Пользователь выходит из системы и перенаправляется на экран авторизации |
| **Фактический результат** | Пользователь выходит из системы и перенаправляется на экран авторизации |
| **Статус** | Зачет |
| **Предварительное условие** | Пользователь авторизован в системе |
| **Постусловие** | Пользователь вышел из системы |
| **Примечания/комментарии** | - |

Тестовый пример #4:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UR\_4 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Регистрация нового пользователя |
| **Краткое изложение теста** | Тест проверяет возможность успешной регистрации нового пользователя в программе. |
| **Этапы теста** | 1. Открыть форму авторизации; 2. Нажать на кнопку "Зарегистрироваться"; 3. Заполнить форму регистрации нового пользователя корректными данными (логин, пароль, ФИО, номер телефона, адрес электронной почты, адрес); 4. Нажать кнопку "Сохранить"; 5. Проверить, что появилось уведомление об успешной регистрации. |
| **Тестовые данные** | Логин: romanova  Пароль: password888  ФИО: Романова Анна Андреевна  Номер телефона: 89005434567  Адрес электронной почты: [romannova8866@gmail.com](mailto:romannova8866@gmail.com)  Адрес: Питер, ул. Ленина 5 |
| **Ожидаемый результат** | Успешная регистрация нового пользователя с предоставленными данными |
| **Фактический результат** | Успешная регистрация нового пользователя с предоставленными данными |
| **Статус** | Зачет |
| **Предварительное условие** | Форма авторизации должна быть доступна |
| **Постусловие** |  |
| **Примечания/комментарии** | - |

Тестовый пример #5:

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UR\_5 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка пустых значений в полях при создании заявки пользователем |
| **Краткое изложение теста** | Тест проверяет обработку попытки заполнения заявки с пустыми значениями в полях и вывод предупреждения при этом |
| **Этапы теста** | 1. Открыть форму заполнение заявки;  2. Оставить одно или несколько полей пустыми;  3. Заполнить остальные поля формы;  4. Нажать кнопку "Отправить";  5. Проверить, что система выдает предупреждение о необходимости заполнения всех полей. |
| **Тестовые данные** | Пусто |
| **Ожидаемый результат** | Система должна выдавать предупреждение о необходимости заполнения всех полей при попытке отправки заполненной заявки с пустыми значениями |
| **Фактический результат** | Система выдает предупреждение о необходимости заполнения всех полей при попытке отправки заполненной заявки с пустыми значениями |
| **Статус** | Зачет |
| **Предварительное условие** |  |
| **Постусловие** |  |
| **Примечания/комментарии** | - |

Реализация 10 unit-тестов, максимально полно покрывающие функционал метода.

Метод TestLoginAsAdmin\_OpenForm1 предназначен для тестирования процесса авторизации администратора в системе. Он создает экземпляр формы авторизации и использует заранее определенные логин и пароль администратора, чтобы попытаться выполнить вход (Рисунок 38).

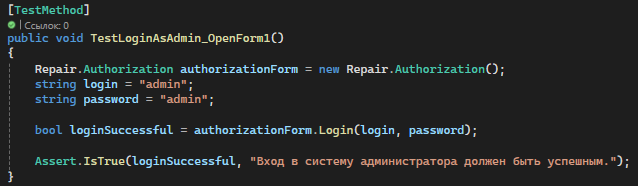


Рисунок 38 – тест TestLoginAsAdmin\_OpenForm1

Метод TestLoginAsClient\_OpenForm2 выполняет тестирование процесса авторизации клиента в приложении. Он создает экземпляр формы авторизации, затем пытается выполнить вход с заранее заданными учетными данными (логин "ivanov" и пароль "password123"). После попытки входа метод проверяет, была ли операция успешной, с помощью проверки булевого значения loginSuccessful, и ожидает, что результат будет истинным, что подтверждает успешный вход клиента в систему (Рисунок 39).

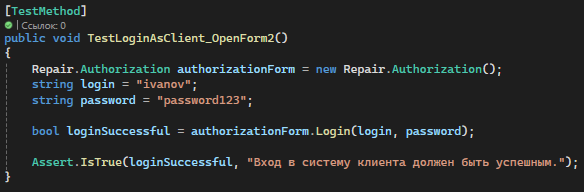


Рисунок 39 – тест TestLoginAsClient\_OpenForm2

В этом коде выполняется тест метода входа в систему для сотрудника с заданными логином и паролем. Тест проверяет, что вход был успешным, и если это не так, выводится сообщение об ошибке (Рисунок 40).

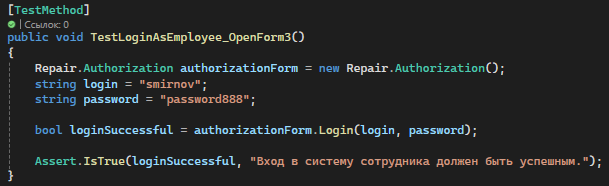


Рисунок 40 – тест TestLoginAsEmployee\_OpenForm3

В данном коде реализован юнит-тест, который проверяет, изменяется ли выбранная таблица в форме Form1 при установке нового значения в комбобоксе. Метод SetComboBoxSelection принимает индекс элемента, представляющего таблицу клиентов (индекс 1), и задает значение переменной selectedTable, так же происходит с таблицей механиков и заявок (Рисунок 41).

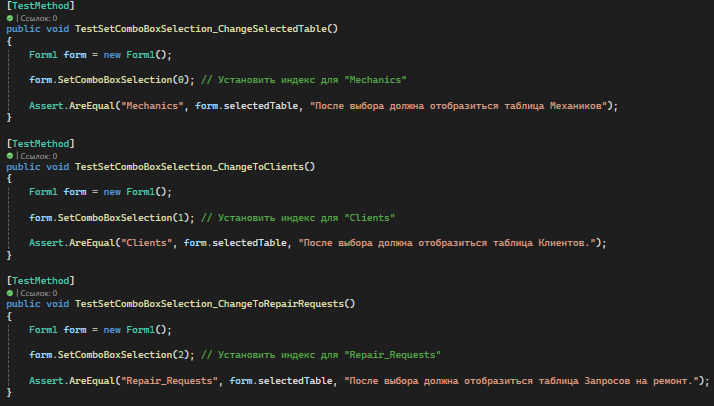


Рисунок 41 – тесты для вывода таблиц

Метод TestSetComboBoxSelection\_InvalidIndex тестирует поведение формы Form1, когда пытается установить выбор в комбобоксе с неверным индексом (Рисунок 42).

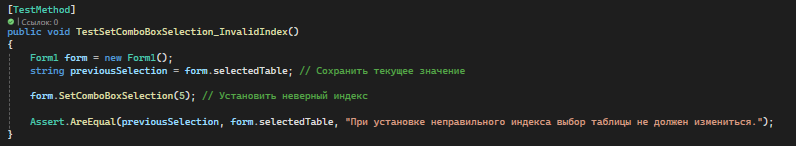


Рисунок 42 – тест TestSetComboBoxSelection\_InvalidIndex

Данный тест проверяет, что при попытке регистрации с пустыми полями (логин и номер телефона) пользователю выводится предупреждающее сообщение о необходимости заполнить все обязательные поля. Тест устанавливает значения в текстовые поля формы, инициирует нажатие кнопки регистрации и сравнивает фактическое сообщение с ожидаемым (Рисунок 43).

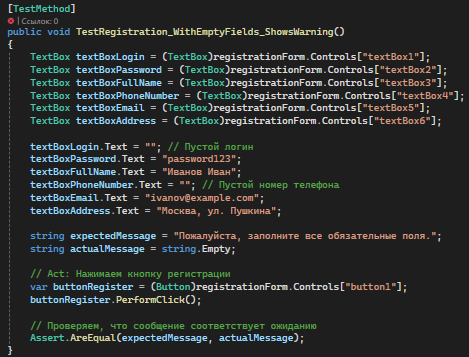


Рисунок 43 – тест TestRegistration\_WithEmptyFields\_ShowsWarning

Этот тест проверяет, что попытка входа в систему с неверными учетными данными (логин и пароль) не приводит к успешной аутентификации. В случае неправильного входа метод Login должен возвращать false, и тест завершится успешно, если это условие выполнено (Рисунок 44).

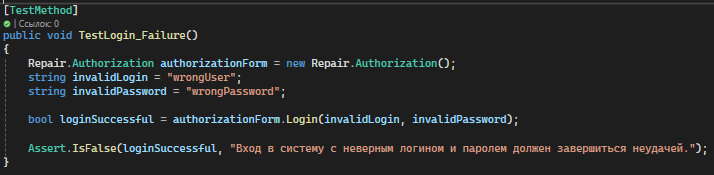


Рисунок 44 – тест TestLogin\_Failure

Приведенный тест ниже проверяет метод ExitApplication в форме регистрации, который должен закрыть форму. После вызова метода тест удостоверяется, что экземпляр формы был корректно закрыт (Рисунок 45).

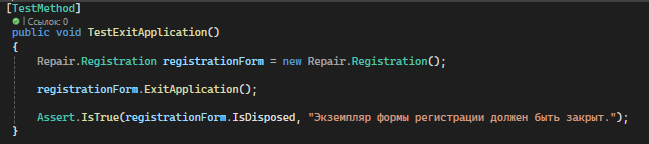


Рисунок 45 – тест TestExitApplication

На изображении ниже показан результат запуска всех тестов (Рисунок 46).

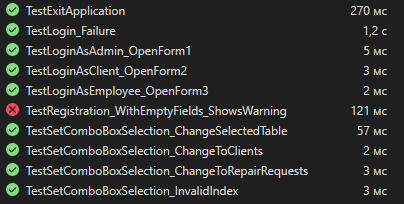


Рисунок 46 –запуск тестов

Задание № 6 «Выгрузка проекта на GitHub»

Готовый проект и отчет необходимо выгрузить на GitHub (Рисунок 47).

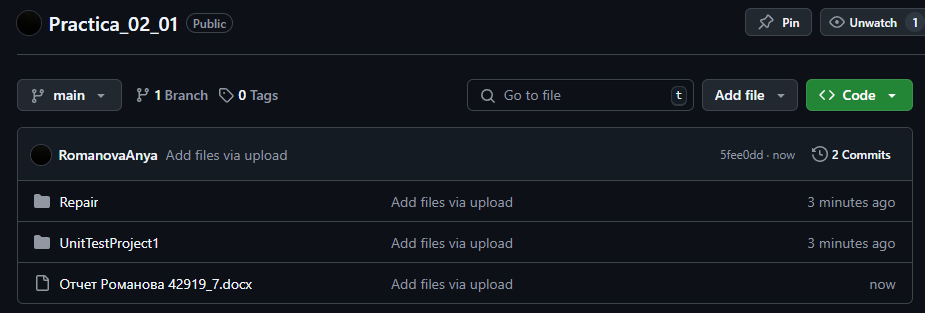


Рисунок 47 – выгрузка проекта

ЗАКЛЮЧЕНИЕ

В ходе разработки информационной системы для ремонта и обслуживания бытовой техники были рассмотрены ключевые аспекты автоматизации учета и обработки заявок. Проведенный анализ предметной области позволил выявить основные требования к системе и определить функциональные компоненты, необходимые для обеспечения высоких стандартов обслуживания. Автоматизация процессов способствует улучшению работы сервисных центров и повышает удовлетворенность клиентов.

Проведение анализа предметной области, а также разработка технического задания обеспечили основу для структурированного проектирования на основе UML. Создание диаграмм вариантов использования и последовательности, а также моделирование активности и состояния помогли лучше понять взаимодействия внутри системы и ее потенциал для дальнейшего развития.

Эффективная реализация всех этапов разработки, включая модульное тестирование и создание тестовых случаев, гарантирует высокое качество и надежность программного обеспечения.

Таким образом, разработанная система оптимизирует процесс обслуживания и ремонта бытовой техники, делает его более прозрачным и эффективным.

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ УП.02.01 (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

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 42919/7 группы

Романова Анна Андреевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| **Виды выполненных работ обучающимся**  **во время практики** | **Объем работ, час.** | **Качество выполнения работ (оценка по пятибалльной системе)** |
| **Тема 1.** Участие в выработке требований к программному обеспечению. | 12 |  |
| **Тема 2.** Стадии проектирования программного обеспечения. Проектирование UML-диаграмм. | 15 |  |
| **Тема 3.** Конструирование пользовательского интерфейса. Разработка модулей программного обеспечения. | 25 |  |
| **Тема 4.** Тестирование и отладка программных модулей. Модульное тестирование. | 12 |  |
| **Тема 5.** Разработка программной документации. | 9 |  |

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

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

(нужное подчеркнуть)

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

Руководитель практики Хисамутдинова А.С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Дата «09» ноября 2024 г.