

Руководство пользователя.....	28
Заключение	34
Список использованных источников	35
Приложение А	36
Приложение Б.....	38
Приложение В	40
Приложение Г	42
Приложение Д	44
Приложение Е.....	46
Приложение Ж.....	48
Приложение З	50

Введение

Тема проекта: Разработка программы для управления данными клиентов и заказов в сервисе по ремонту и обслуживанию компьютерной техники «GrodnoLaptop».

Цель проекта заключается в создании программы, которая автоматизирует процесс учета клиентов, обработки данных о заказах, предоставляемых услугах и выполненных работах. Программа рассчитана на использование в небольшом компьютерном сервисе.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем вы сможете ознакомиться с постановкой задачи, включающей исследование предметной области учета клиентов в компьютерном сервисе и изучение текущих методов решения подобных задач. Также в разделе будут описаны входные данные (например, данные клиента, информация об устройстве, список предоставляемых услуг) и выходные данные (отчеты, готовые списки выполненных заказов). В подразделе «Инструменты разработки» будет рассмотрена выбранная среда программирования, а также установлены минимальные требования к техническим средствам для работы программы.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки программы. Описывается структура хранения данных (например, таблицы клиентов, заказов, услуг) и способы их обработки. В этом разделе будет подробно описан пользовательский интерфейс программы: меню, элементы управления, формы ввода данных. Также будут составлены алгоритмы работы программы, включая добавление, редактирование и удаление данных.

«Реализация задачи» – это третий раздел пояснительной записки, в котором приводятся описание всех элементов программы, используемых для реализации функционала. Описываются основные функции приложения, например: регистрация новых клиентов, обработка информации о заказах, формирование отчетов. Также в разделе будут представлены аннотации к файлам и компонентам, входящим в состав программы.

Четвертый раздел – «Тестирование». В этом разделе будет описано тестирование функционала программы. Проверка охватит ввод, обработку и вывод данных, а также корректность выполнения пользовательских операций. Будут рассмотрены возможные сценарии работы, включая ввод некорректных данных и моделирование ошибок.

В разделе «Применение» будет описано назначение программы, её область применения (например, использование в локальных сервисных центрах), а также условия эксплуатации.

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		4

«Заключение» подведет итоги разработки, кратко сформулирует задачу, представит выводы о проделанной работе, а также оценит эффективность автоматизации учета клиентов.

В разделе «Литература» будет приведен список источников, использованных при разработке программы.

В приложениях к пояснительной записке будет представлен листинг программы с комментариями к коду. Графическая часть будет содержать схемы работы программы и структуры данных.

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		5

1 Анализ задачи

1.1 Постановка задачи

1.1.1 Организационно-экономическая сущность задачи

Наименование задачи: Программа для учета клиентов в компьютерном сервисе «GrodnoLaptop».

Цель разработки: Создание программы для облегчения ведения учета клиентов и их заказов в компьютерном сервисе.

Назначение: Данный программный продукт предназначен только для сотрудников сервиса «GrodnoLaptop» для ведения учета клиентов и их заказов.

Периодичность использования: По мере необходимости.

Источники и способы получения данных: Сотрудники компьютерного сервиса.

Обзор существующих аналогичных ПП: WireCrm – это ПП позволяющий формирование базы клиентов, заключаемых договоров, заказов, истории доставок по каждому клиенту отдельно.

Класс365 – это ПП позволяющий вести учет заказов, подходящий для компаний и интернет-магазинов в котором можно вести учет клиентов и получать информацию о товарах.

Qsystem CRM – это ПП для ведения учета заказов, предоставляющая такие функции как: учет товаров, учет выполненных заказов, учет контактов, учет накладных и прочего.

1.1.2 Функциональные требования

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Пользователь:

- 1) Добавление клиентов;
- 2) Редактирование информации о заказах;
- 3) Изменение статуса заказов;
- 4) Установка фильтров поиска клиентов;
- 5) Сортировка таблицы по разным столбцам.

1.1.3 Описание входной информации:

Перечень исходной информации:

- 1) ФИО клиента;
- 2) Тип устройства клиента;
- 3) Предоставляемые услуги;
- 4) Статус заказа;
- 5) Стоимость услуг;
- 6) Дата добавления заказа;
- 7) Номер телефона клиента;

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		6

8) Ответственный сотрудник;

9) Комментарий.

Форма представления информации:

Электронный файл, содержащий информацию.

В качестве формы ввода будет представлена структура экранных форм ввода, куда пользователь вручную вводит информацию.

Пример заполнения данных представлен в таблице 1:

Таблица 1 – Пример заполнения данных.

ФИО клиента	Тип устройства	Услуга	Статус	Стоимость	Дата	Номер телефона	Мастер	Комментарий
1	2	3	4	5	6	7	8	9
Иванов Сергей Алексеевич	Ноутбук	Ремонт материнской платы	Активен	250 р.	14.12.2024	+375293821973	Константин	
Петров Владислав Александрович	Телефон	Замена дисплея	Выполнен	0 р.	10.11.2024	+375334508714	Михаил	Отменен

Перечень пользователей исходной информации:

Пользователь: вводит данные о клиентах и устройствах при оформлении заказа и предоставляет данные о выполненных работах и состоянии заказа.

1.1.4 Описание выходной информации:

Перечень выходной информации:

1) Отчет о выполненных заказах (информация о клиентах, их устройствах, проведенных работах, стоимости услуг);

2) Список активных заказов (заказы, которые находятся в работе, с указанием текущего статуса);

3) Итоговая таблица для клиента.

Формы представления результатной информации:

– На экране программы: таблица с данными;

– В виде файлов: отчет сохраненный в формате .xlsx.

Периодичность и сроки представления:

Для сотрудников: информация о текущих заказах доступна в реальном времени через интерфейс программы.

Перечень пользователей результатной информации:

Пользователь: получает информацию о статусе заказов для уведомления клиентов, видит список текущих заказов и подробности о задаче.

1.1.5 Описание условно-постоянной информации:

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ			Лист
Изм.	Лист	№ докум.	Подпись	Дата				7

Перечень условно-постоянной информации:

- Справочник статусов заказа: выбор различных статусов заказа (например, «Активен», «Выполнен»);
- Справочник сотрудников: выбор имени сотрудника (например, Константин, Михаил);
- Таблица клиентов и заказов: данные о клиентах и их заказах которые остаются в базе;

Формы представления:

- В виде таблиц: справочник сотрудников и справочник статусов отображаются в таблицах с колонками (например, "Статус заказа", "Мастер");
- В выпадающих списках: Статусы заказа, сотрудников можно выбирать из выпадающего списка.

1.1.6 Нефункциональные (эксплуатационные) требования

Требования к применению:

Интерфейс должен быть интуитивно понятным и доступным даже для пользователей с минимальным уровнем технической подготовки.

Использование стандартных элементов управления (выпадающие списки, кнопки, формы ввода).

Минимальное количество действий для выполнения основной операции (например, регистрация клиента).

Программа должна сопровождаться подробным руководством с описанием всех функций.

При необходимости предоставляется краткое руководство по обучению сотрудников сервисного центра работе с программой.

Требования к производительности:

Программа должна обрабатывать запросы пользователя (например, поиск клиента, добавление заказа) за время не более 1 секунды. Программа должна корректно работать с базой данных, содержащей до 1000 записей о клиентах и заказах без заметного снижения производительности. Программа не должна потреблять более 100 МБ оперативной памяти в стандартных условиях работы. Время запуска программы — не более 5 секунд.

Требования к реализации:

Разработка на языке Python. Поддержка операционных систем Windows 10 и выше. Использование стандарта SQL для работы с реляционной базой данных (например, SQLite). Среда для разработки Microsoft Visual Studio Code.

Требования к надежности:

Программный продукт должен быть стабильным и надежным в работе, минимизируя возможные сбои, ошибки и аварийные ситуации. Программа должна работать стабильно при стандартных нагрузках, допуская не более 1 сбоя на 1000

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

операций. Необходимо реализовать механизмы обеспечения отказоустойчивости программы, позволяющие восстанавливаться после сбоев и восстановить работоспособность. В случае аварийного завершения работы программа должна корректно завершать открытые сессии и сохранять данные до последнего выполненного действия. Программа после сбоя должна запускаться повторно. Должна быть доступна в любой момент времени.

Требования к интерфейсу:

Пользователь должен иметь доступ ко всем функциям программы. Данные вводятся пользователями через графический интерфейс. Результаты выводятся на экран, сохраняются в файлы (например, MS Excel).

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		9

1.2 Диаграмма вариантов использования

В качестве актеров представлен пользователь, который взаимодействует с системой. Ему представлены варианты использования:

- Просмотр списка заказов – пользователь просматривает существующие заказы;
- Поиск в таблице – поиск информации в таблице с заказами и клиентами. Включает поиск по фильтрам, что позволяет уточнить отображение списка заказов;
- Редактирование списка клиентов и заказов. Включает: добавление заказа, удаление заказа, редактирование информации о заказах;
- Создание отчета – пользователь генерирует отчет;
- Сортировка таблицы, включает сортировку данных по столбцам таблицы;
- Открытие справочной информации.

Разработанная диаграмма вариантов использования представлена в приложении А.

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		10

1.3 Выбор стратегии разработки и модели жизненного цикла

Для разработки программы для управления данными клиентов и заказов в сервисе по ремонту и обслуживанию компьютерной техники «GrodnoLaptop» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 2 – Выбор модели жизненного цикла на основе характеристик требований

Критерии категории требований	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7
Являются ли требования к проекту легко определяемыми и реализуемыми?	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет	Нет
Могут ли требования быть сформулированы в начале ЖЦ?	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет
Часто ли будут изменяться требования на протяжении ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да
Нужно ли демонстрировать требования с целью их определения?	<u>Нет</u>	<u>Нет</u>	Да	<u>Нет</u>	Да	Да
Требуется ли проверка концепции программного средства или системы?	<u>Нет</u>	<u>Нет</u>	Да	<u>Нет</u>	Да	Да
Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Нужно ли реализовать основные требования на ранних этапах разработки?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>

Вычисления: 5 за каскадную, 5 за V-образную, 4 за RAD, 6 за инкрементную, 2 за быстрого прототипирования и 2 за эволюционную.

Итог: на основе результатов заполнения табл. 2 подходящей является инкрементная модель.

Таблица 3 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

Критерии категории команды разработчиков проекта	Каскадная	V-образная	RAD	Инкрементная	Быстро о прототипирован ия	Эволюционная
1	2	3	4	5	6	7
Являются ли проблемы предметной области проекта новыми для большинства разработчиков?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да
Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков?	Да	Да	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да
Изменяются ли роли участников проекта на протяжении ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
Является ли структура процесса разработки более значимой для разработчиков, чем гибкость?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	Нет	Нет
Важна ли легкость распределения человеческих ресурсов проекта?	Да	Да	Да	Да	<u>Нет</u>	<u>Нет</u>
Приемлет ли команда разработчиков оценки, проверки, стадии разработки?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>

Вычисления: 4 за каскадную, 4 за V-образную, 3 за RAD, 4 за инкрементную, 3 за быстрого прототипирования и 2 за эволюционную.

Итог: на основе результатов заполнения табл. 3 подходящими являются каскадная, V-образная и инкрементная модели.

Таблица 4 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

Критерии категории коллектива пользователей	Каскадная	V-образная	RAD	Инкрементная	Быстро го прототипирован ия	Эволюционная
1	2	3	4	5	6	7
Будет ли присутствие пользователей ограничено в ЖЦ разработки?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	Нет	<u>Да</u>
Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>

Продолжение таблицы 4

1	2	3	4	5	6	7
Будут ли пользователи вовлечены во все фазы ЖЦ разработки?	Нет	Нет	<u>Да</u>	Нет	<u>Да</u>	Нет
Будет ли заказчик отслеживать ход выполнения проекта?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

Вычисления: 1 за каскадную, 1 за V-образную, 1 за RAD, 2 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: на основе результатов заполнения табл. 4 подходящей является модель быстрого проектирования и эволюционная.

Таблица 5 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

Критерии категории типов проекта и рисков	Каскадная	V-образная	RAD	Инкрементная	Быстро о прототипирования	Эволюционная
1	2	3	4	5	6	7
Разрабатывается ли в проекте продукт нового для организации направления?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Будет ли проект являться расширением существующей системы?	Да	Да	Да	Да	<u>Нет</u>	<u>Нет</u>
Будет ли проект крупно- или среднемасштабным?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
Ожидается ли длительная эксплуатация продукта?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	Нет	<u>Да</u>
Необходим ли высокий уровень надежности продукта проекта?	Нет	<u>Да</u>	Нет	<u>Да</u>	Нет	<u>Да</u>
Предполагается ли эволюция продукта проекта в течение ЖЦ?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Велика ли вероятность изменения системы (продукта) на этапе сопровождения?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
Является ли график сжатым?	<u>Нет</u>	<u>Нет</u>	Да	Да	Да	Да
Предполагается ли повторное использование компонентов?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>

Продолжение таблицы 5

1	2	3	4	5	6	7
Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

Вычисления: 3 за каскадную, 4 за V-образную, 2 за RAD, 6 за инкрементную, 6 за быстрого прототипирования и 8 за эволюционную.

Итог: на основе результатов заполнения табл. 5 подходящей является эволюционная модели.

Общие вычисления: 13 за каскадную, 14 за V-образную, 10 за RAD, 18 за инкрементную, 14 за быстрого прототипирования и 15 за эволюционную.

Общий итог: в итоге заполнения табл. 2 – 5 наиболее подходящей является инкрементная модель.

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

1.4 Инструменты разработки

Для разработки данного проекта будет выбрана среда разработки Visual Studio Code, которая является наиболее актуальной средой для создания приложений данного типа.

Разработка будет осуществляться на языке Python с применением библиотеки PyQt5.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

- Microsoft Visual Studio Code – это текстовый редактор, разработанный для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

- Qt Designer – среда для разработки графических интерфейсов программ, использующих библиотеку Qt;

- WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;

- Microsoft Word 2021 – это универсальное приложение для обработки текстовых документов. Оно предоставляет широкий спектр возможностей для создания, редактирования и форматирования текста. С помощью Microsoft Word легко создавать разнообразные документы, такие как письма, резюме, отчеты, статьи, презентации и многое другое. Программа обладает удобным и интуитивно понятным интерфейсом, что делает её доступной для пользователей всех уровней навыков. Благодаря возможностям автоматизации, шаблонам и интеграции с другими приложениями Microsoft Office, такими как Excel и PowerPoint, Word является незаменимым инструментом для работы с текстовой информацией в офисной среде, образовании и повседневной жизни. Microsoft Word необходим для написания документации;

- Microsoft Visio — это мощный инструмент для создания диаграмм, схем и визуализации данных, который входит в линейку продуктов Microsoft Office. Visio активно используется в бизнесе, инженерии, IT и других областях для создания структурированных схем и графиков;

- Figma – это популярный облачный инструмент для проектирования пользовательских интерфейсов и совместной работы над дизайном. Он выделяется своей простотой, доступностью и возможностью работать непосредственно в браузере, не требуя установки сложного программного обеспечения;

- Adobe Photoshop 2023 – это мощный графический редактор, который широко используется для редактирования и обработки изображений. Включает в

себя широкий спектр инструментов и функций, позволяющих профессионалам и любителям создавать и редактировать изображения любой сложности. Необходим для оформления интерфейса и для работы с текстом, включая различные шрифты, стили и эффекты. Photoshop предоставляет множество инструментов для редактирования цветов, яркости, контрастности и других характеристик изображений;

- Smart Install Maker – это программа для создания установочных пакетов для программного обеспечения под операционные системы Windows. Она обладает простым и интуитивно понятным интерфейсом, который позволяет пользователям легко создавать профессиональные установщики для своих программ;

С помощью Smart Install Maker можно создавать нестандартные установочные пакеты, включая различные опции установки, параметры, создание ярлыков, добавление лицензионных соглашений и многое другое. Программа поддерживает автоматическое создание систем восстановления и резервного копирования, что обеспечивает дополнительную гибкость и надежность процесса установки;

Smart Install Maker также предоставляет возможность создания многоязычных установочных пакетов, что позволяет программам быть доступными для широкой аудитории. Кроме того, программа поддерживает сжатие файлов и защиту паролем, обеспечивая безопасность и компактность установочных пакетов;

- Dr.Explain – это инструмент для создания документации и руководств пользователя. Он автоматизирует процесс написания документации для ПО и веб-сайтов;

- Xmind – для создания системы меню и навигации по программе.

Разработка ведется на ноутбуке Asus, который имеет следующие системные характеристики:

- Процессор Amd Ryzen 5 4600H;
- Видеокарта Nvidia Geforce RTX 3050;
- SSD-накопитель на 500 гб;
- Объём ОЗУ 16 гб;
- Операционная система Windows 11 Pro.

Однако программа не очень требовательна к системным ресурсам и способна работать на любых современных компьютерах.

1.5 Разработка плана работы над проектом

Диаграмма Ганта – это наглядное отображение всех задач, которые нужно сделать для завершения проекта. Она представляет собой горизонтальную шкалу времени, на которой отображаются задачи проекта, их последовательность и продолжительность.

Она наглядно разбивает проект на задачи с указанием временных рамок. Горизонтальная ось – это временная шкала. Вертикальная ось – перечень задач или этапов проекта. Прямоугольные блоки на диаграмме – временные интервалы выполнения задач.

Разработанная диаграмма Ганта представлена в приложении Б.

					УП ТРПО 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		17

2 Проектирование задачи

2.1 Разработка системы меню и навигации по программе

Система меню и навигации является важным компонентом пользовательского интерфейса, обеспечивающим удобный доступ к функциональности и разделам приложения. Ее главная задача — помочь пользователю быстро находить необходимую информацию и перемещаться между различными страницами. Для этого меню должно быть интуитивным, доступным и логичным.

При разработке такой системы необходимо учитывать адаптивный дизайн, чтобы меню корректно отображалось на любых экранах. Также важно документировать структуру и работу навигации, чтобы пользователи могли быстро освоиться. Хорошо продуманная система меню способствует повышению эффективности работы с приложением и улучшению общего пользовательского опыта.

Структура меню строится на основе иерархии разделов: главное меню включает ключевые пункты, которые состоят из таких кнопок как «Заказ», «Редактировать», «Удалить», «Справка», «Экспорт», «Поиск» и «Очистить». Для удобства взаимодействия активные элементы навигации визуально выделяются, а плавные анимации при переходах делают процесс более естественным.

Разработанная система меню и навигации представлена в приложении В.

2.2 Разработка UML-диаграмм

2.2.1 Диаграмма последовательности

Диаграмма последовательности используется для визуализации взаимодействия между объектами в системе. Она показывает порядок сообщений, которыми объекты обмениваются друг с другом во времени и их жизненный цикл. Это самый простой и удобный инструмент для демонстрации всех интеграций и взаимодействий в рамках проектируемого бизнес-процесса.

На диаграмме показана последовательность действий для редактирования заказа в таблице.

Разработанная диаграмма последовательности представлена в приложении Г.

2.2.2 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

На диаграмме показаны используемые в программе классы.

С диаграммой классов можно ознакомиться в приложении Д.

2.2.3 Диаграмма деятельности

Диаграмма деятельности – это UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний.

Разработанная диаграмма деятельности показана в приложении Е

2.3 Разработка пользовательского интерфейса

Одним из важнейших элементов проектирования является описание внешнего интерфейса разрабатываемого программного продукта.

Для разработки дизайна использовалась стилистика сайта компьютерного сервиса «www.grodnolaptop.by».

Для обеспечения эффективной работы пользователя нужно создать целостное приложение, охватывающее конкретную предметную область, где все компоненты будут сгруппированы по их функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс. Приложение должно позволять пользователю решать задачи с минимальными усилиями по сравнению с работой с разрозненными объектами.

Прототип – это модель пользовательского интерфейса. В сущности, это «черновик», созданный на основе представления разработчика о потребностях пользователя.

В создании интерфейса были задействованы такие шрифты как: Jost, Roboto и Inter. Для текста были использованы черные, серые и белые цвета. Логотип программы был взят с сайта компьютерного сервиса.

Прототипы UX и UI представлены в приложении Ж.

Со всеми разработанными UX прототипами можно ознакомиться по ссылке: <https://www.figma.com/design/oNrCchc0JtiRPtP1RlhV1x/UX-grodnolaptop?node-id=19-929&t=TUUC4CmCCMCXF0Rm-1>

Со всеми разработанными UI прототипами можно ознакомиться по ссылке: <https://www.figma.com/design/3GHvnghgIk40WlOCdggTm4/UI-grodnolaptop?node-id=0-1&t=tii2SYD3kZEc9haf-1>

3 Реализация

3.1 Руководство программиста

Программный продукт разработан на языке программирования Python с использованием библиотеки PyQt5. Эта библиотека позволяет создавать оконные программы на языке Python и обладает функционалом для настройки и добавления элементов, редактирования их стилей и настройки функционала. Для создания графического интерфейса была использована среда разработки Qt Designer. Для написания функционала и связывания файлов интерфейса был использован текстовый редактор Microsoft Visual Studio Code.

Для создания базы данных в файле DB_connection.py был использован следующий фрагмент кода:

```
path=os.path.dirname(os.path.abspath(__file__))
db=QtSql.QSqlDatabase.addDatabase('SQLITE') #Установка соединения с
базой данных SQLITE3
db.setDatabaseName(path+r"\orders_db.db') #Определение имени базе данных
Здесь устанавливается драйвер СУБД создается файл базы данных с
указанием пути файла.
```

Для создания таблицы в базе данных был использован следующий SQL-запрос:

```
"""CREATE TABLE IF NOT EXISTS orders (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    client TEXT NOT NULL,
    device TEXT NOT NULL,
    service TEXT NOT NULL,
    status TEXT NOT NULL,
    price INTEGER,
    date TEXT NOT NULL,
    number INTEGER,
    master TEXT NOT NULL,
    comment TEXT NOT NULL)
"""
```

В этом коде объявляются поля таблицы и их типы данных.

Для облегчения обращения к таблице была создана функция для обработки SQL-запросов:

```
def create_query(self, sql_query, sql_query_values=None):#Метод для
обработки SQL-запросов
    query=QtSql.QSqlQuery()
    query.prepare(sql_query)
    if sql_query_values is not None:
```

```

for query_value in sql_query_values:
    query.addBindValue(query_value)

```

Данная функция получает текст SQL-запроса и список элементов для передачи информации в таблицу базы данных.

В во всех окнах интерфейса программы используются нестандартные шрифты, которые программа загружает из файлов при помощи следующих команд:

```

path=os.path.dirname(os.path.abspath(__file__))
QFontDatabase.addApplicationFont(path+"\\fonts\\Inter-Medium.otf")
QFontDatabase.addApplicationFont(path+"\\fonts\\Jost-SemiBold.ttf")
QFontDatabase.addApplicationFont(path+"\\fonts\\Roboto.ttf")

```

Для настройки обработки событий кнопок пользовательского интерфейса был использован фрагмент кода:

```

self.ui.order_btn.clicked.connect(self.new_order_window)
self.ui.edit_btn.clicked.connect(self.new_order_window)
self.ui.del_btn.clicked.connect(self.delete_selected_order)
self.ui.search_Button.clicked.connect(self.apply_filter)
self.ui.clear_Button.clicked.connect(self.delete_filter)
self.ui.help_btn.clicked.connect(self.open_manual)
self.ui.export_btn.clicked.connect(lambda:self.export_to_excel_with_pandas(self.ui.tableView, f"{self.desktop_path}\\Данные.xlsx"))

```

За инициализацию интерфейса главного окна и подключения к нему базы данных отвечает следующий фрагмент кода:

```

def __init__(self):
    super(GrodnoLaptop,self).__init__()
    self.ui = Ui_MainWindow()
    self.ui.setupUi(self)
    self.con = Data() #Экземпляр класс Data из файла DB_connection
    self.view_table()

```

Здесь создается объект класса Ui_mainWindow() для создания интерфейса и его элементов и с помощью метода SetupUi() связывается с текущим классом.

Создается объект класса Data(), который используется для работы с базой данных.

Функция view_table() отображает содержимое таблицы.

Для создания диалогового окна используется функция:

```

def new_order_window(self):
    self.new_window = QDialog()
    self.ui_window=Ui_Dialog()
    self.ui_window.setupUi(self.new_window)

```

Здесь создается объект new_window класса QDialog() и объект ui_window класса Ui_Dialog() из файла графического интерфейса диалогового окна. Функция SetupUi() связывает ui_window с классом QDialog().

Для передачи данных добавления заказа в базу данных из диалогового окна создана функция:

```
def add_order(self):
    self.con.add_new_order(client, device, service, status, price, date, number,
        master, comment)
    self.view_table()
    self.new_window.close()
```

Функция передает аргументы полученные из текстовых полей в функцию add_new_order() класса Data() для создания SQL-запроса. После вызывается функция view_table() для обновления таблицы и закрывается диалоговое окно.

Для создания Excel документа используется функция использующая библиотеку Pandas:

```
def export_to_excel_with_pandas(self, table_view, file_path):
    model = table_view.model()
    if not model:
        return
```

Функция получает в параметр table_view модель данных таблицы, в параметр file_path путь создания файла.

Далее из модели данных в список собираются заголовки и строки с данными:

```
data = []
    headers = [model.headerData(col, Qt.Horizontal) for col in
        range(model.columnCount())]

    for row in range(model.rowCount()):
        row_data = [model.index(row, col).data() for col in
            range(model.columnCount())]
        data.append(row_data)
```

Далее создается DataFrame и преобразовывается в документ, после чего появляется информационное окно:

```
df = pd.DataFrame(data, columns=headers)
df.to_excel(file_path, index=False)
QMessageBox.about(None, "Выполнено", "Файл успешно создан на
    рабочем столе")
```

Для открытия справочника используется соответствующая функция:

```
def open_manual(self):
```

```

path=os.path.dirname(os.path.abspath(__file__))+"manual.chm"
if os.path.exists(path):
    os.startfile(path)

```

В переменную path записывается относительный путь файла запуска программы в котором находится файл справки. Далее проверяется существует ли по такому пути файл и в случае истинности он запускается.

Для центрального выравнивания элементов в таблице был переопределен класс QSqlTableModel:

```

class CenteredSqlTableModel(QSqlTableModel):    #Подкласс класса
QsqlTableModel для выравнивания элементов по центру
def data(self, index, role):
    if role == Qt.TextAlignmentRole:
        return Qt.AlignCenter # Выравниваем данные по центру
    return super().data(index, role)

```

Был создан класс CenteredSqlTableModel() наследующий класс QSqlTableModel(). Был переопределен метод data() обрабатывающий параметр role если он равен Qt.TextAlignmentRole, метод возвращает значение Qt.AlignCenter, указывающее что текст должен быть по центру. Если role не равна Qt.TextAlignmentRole, то data() вызывает реализацию из базового класса.

Был создан класс CustomFilterProxyModel() наследующий класс QSortFilterProxyModel():

```

class CustomFilterProxyModel(QSortFilterProxyModel):
def __init__(self):
    super().__init__()
    self.filters = {
        'client': "",
        'device': "",
        'service': "",
        'status': "",
        'price': "",
        'phone': "",
        'master': ""
    }

```

```

def set_filter(self, column, value):#Метод для установки фильтра по
столбцу
    self.filters[column] = value

```

```

def filterAcceptsRow(self, source_row, source_parent):

```

Этот класс используется для получения из модели данных таблицы модель сортировки данных с целью их сортировки. Для работы с используемой таблицей в классе были переопределены методы `set_filter()` и `filterAcceptsRow()` под используемые столбцы.

3.1.1 Организация данных

Проект состоит из следующих файлов:

- `Main.py` – основной файл запуска программы;
- `Base_Window.py` – файл создания интерфейса главного окна;
- `Splashscreen.py` – файл создания интерфейса заставки;
- `Dialog_Window.py` – файл создания интерфейса диалогового окна;
- `DB_connection.py` – модуль для работы с базой данных;
- `icons_dialog.py` – файл с ресурсами для диалогового окна;
- `icons_res.py` – файл с ресурсами для главного окна;
- `splash_screen_res.py` – файл с ресурсами для заставки;
- `orders_db.db` – файл хранения базы данных;
- `manual.chm` – файл для открытия справочной системы.

В качестве СУБД использовалась SQLite3. Основной таблицей является `orders`. Она содержит следующие поля:

- `id`: целое число, ключевое поле (идентификатор заказа);
- `client`: текст (ФИО клиента);
- `device`: текст (наименование устройства клиента);
- `service`: текст (оказываемые клиенту услуги);
- `status`: текст (статус заказа);
- `price`: целое число (стоимость услуг);
- `date`: текст (дата добавления заказа);
- `number`: целое число (номер телефона клиента);
- `master`: текст (назначенный мастер).

В качестве входной информации используются данные, вводимые пользователем через интерфейс (например, добавление нового заказа, выбор фильтров), данные из базы данных.

В качестве выходной информации используются данные отображаемые из таблицы с учетом сортировки и фильтров.

3.1.2 Структура программы

Файлы `Base_Window.py`, `Splashscreen.py`, `Dialog_Window.py` отвечают за создание главного окна, заставки и диалогового окна соответственно. Элементы интерфейса связаны с событиями (например, кнопки, поля ввода).

Файл `orders_db.db` управляет подключением к базе данных, выполняет запросы `SELECT`, `INSERT`, `UPDATE` и `DELETE`.

Файлы `splash_screen_res.py`, `icons_res.py`, `icons_dialog.py` хранят необходимые ресурсы для файлов интерфейсов (например, картинки, иконки).

Файл `main.py` связывает между собой файлы интерфейсов создавая окна приложений, связывая действия пользователя с функциями обработки данных.

Логика работы с интерфейсом отделена от логики взаимодействия с базой данных и от логики обработки событий.

3.1.3 Структура и описание процедур и функций пользователя

Основные функции файла `main.py`:

- `view_table()` – загружает в таблицу данные из базы данных;
- `apply_filter()` – применяет выбранные к данным фильтры;
- `delete_filter()` – убирает все примененные фильтры;
- `new_order_window()` – запускает диалоговое окно создания либо редактирования заказа;
- `add_order()` – отвечает за передачу данных заказа в базу данных;
- `edit_selected_order()` – отвечает за передачу отредактированных данных в базу данных;
- `delete_selected_order()` – функция удаляющая выбранный заказ;
- `export_to_excel_with_pandas()` – функция создающая Excel документ с данными из базы данных;
- `open_manual()` – функция отвечающая за открытие справочника.

Основные функции файла `DB_connection.py`:

- `connection()` – функция для подключения к базе данных;
- `create_query(sql_query, sql_query_values=None)` – функция для создания SQL-запроса;
- `add_new_order(client, device, service, status, price, date, phone_number, master)` – функция добавления заказа в базу данных;
- `edit_order(client, device, service, status, price, date, phone_number, master, id)` – функция редактирования заказа в базе данных;
- `delete_order(id)` – функция удаления заказа из базы данных;
- `get_change(id)` – функция получения информации определенного заказа из базы данных.

4 Тестирование

4.1 Тесты на использование

При разработке программы возникали различные ошибки и недоработки, которые исправлялись в ходе реализации. После завершения реализации программы было проведено полное тестирование программы. Тестирование должно гарантировать стабильную и бесперебойную работу программы.

Разработанные тест-кейсы и статус их выполнения представлены в приложении 3.

Расписание работ над проектом представлено в таблице 6.

Таблица 6 – Расписание работы над проектом

Имя	Дата	Деятельность	Продолжительность, ч
Козейко Максим	06.12.2024	Разработка тестов	2
Козейко Максим	06.12.2024	Тестирование программы	2
Козейко Максим	08.11.2024	Составление отчетов о найденных дефектах	1
Козейко Максим	09.11.2024	Исправление найденных ошибок	3
Козейко Максим	11.11.2024	Проведение регрессионного тестирования	2
Козейко Максим	11.11.2024	Составление отчета о результатах тестирования	2

4.2 Отчёт о результатах тестирования

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

Статистика по всем дефектам представлена в таблице 7.

Таблица 7 – Статистика по всем дефектам

Статус	Количество	Важность			
		Низкая	Средняя	Высокая	Критическая
Найдено	0	0	0	0	0
Исправлено	0	0	0	0	0
Проверено	0	0	0	0	0
Открыто заново	0	0	0	0	0
Отклонено	0	0	0	0	0

5 Руководство пользователя

Программа для управления данными клиентов и заказов в сервисе по ремонту и обслуживанию компьютерной техники «GrodnoLaptop» создана для применения на персональных компьютерах с операционной системой Windows.

Для инсталляции программы необходимо запустить инсталлятор setup.exe представленный на рисунке 1.

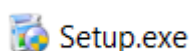


Рисунок 1 – Инсталлятор программы

Далее открывается страница приветствия (рисунок 2), где необходимо выбрать кнопку «Далее» для продолжения установки либо «Отмена» для прекращения установки.

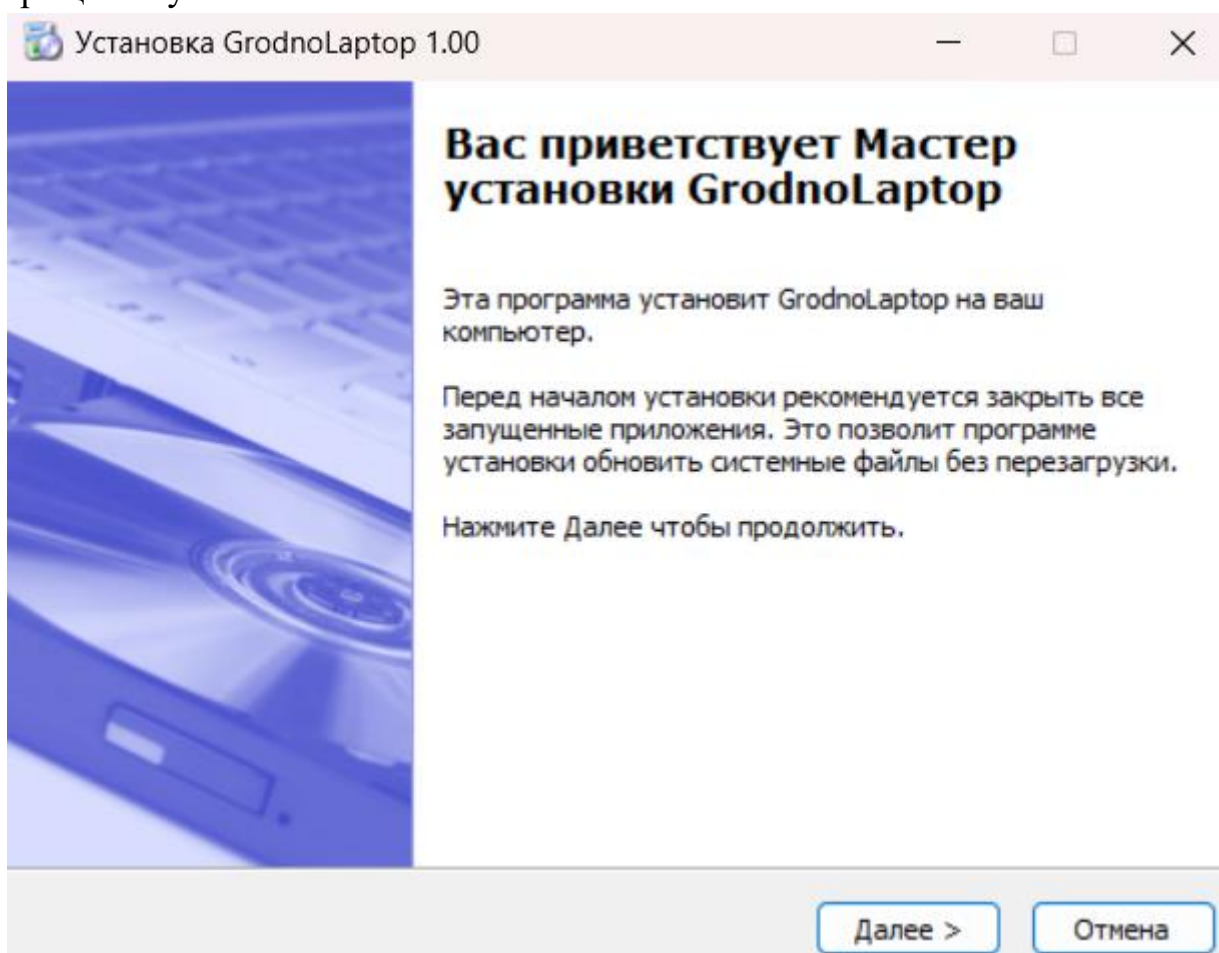


Рисунок 2 – Страница приветствия

Следующим открывается страница выбора папки (рисунок 3), где необходимо указать путь к папке в которую установится программа.

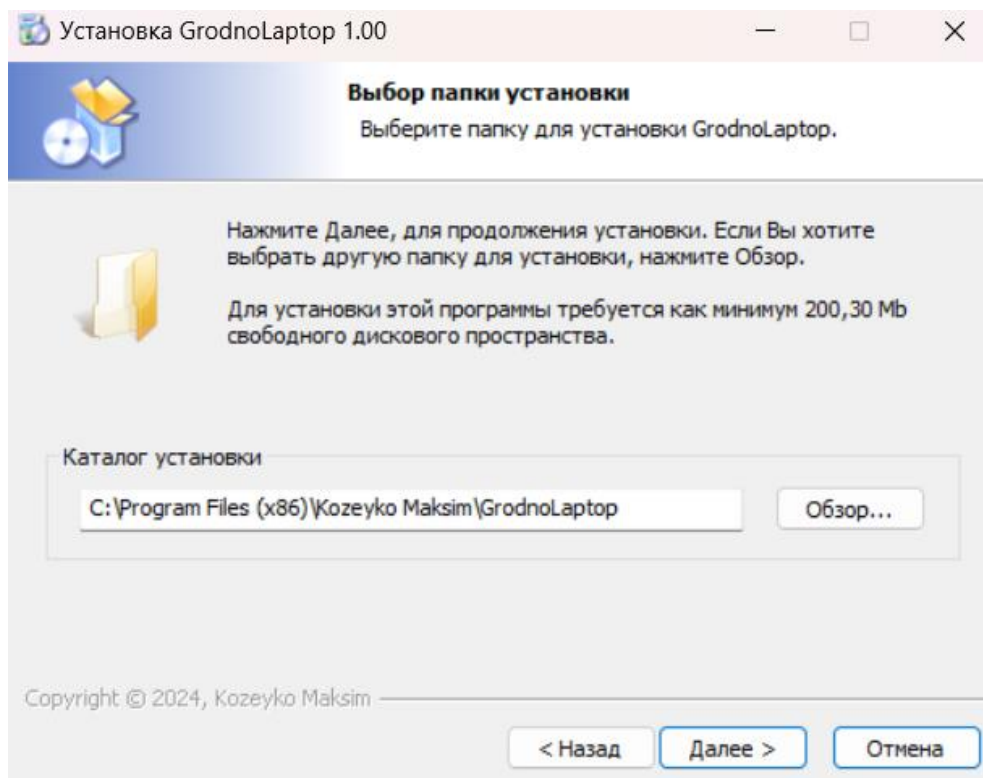


Рисунок 3 – Страница выбора папки

После выбора пути папки и нажатия кнопки «Далее» откроется страница выбора дополнительных ярлыков (рисунок 4), где можно выбрать установку ярлыка на рабочем столе.

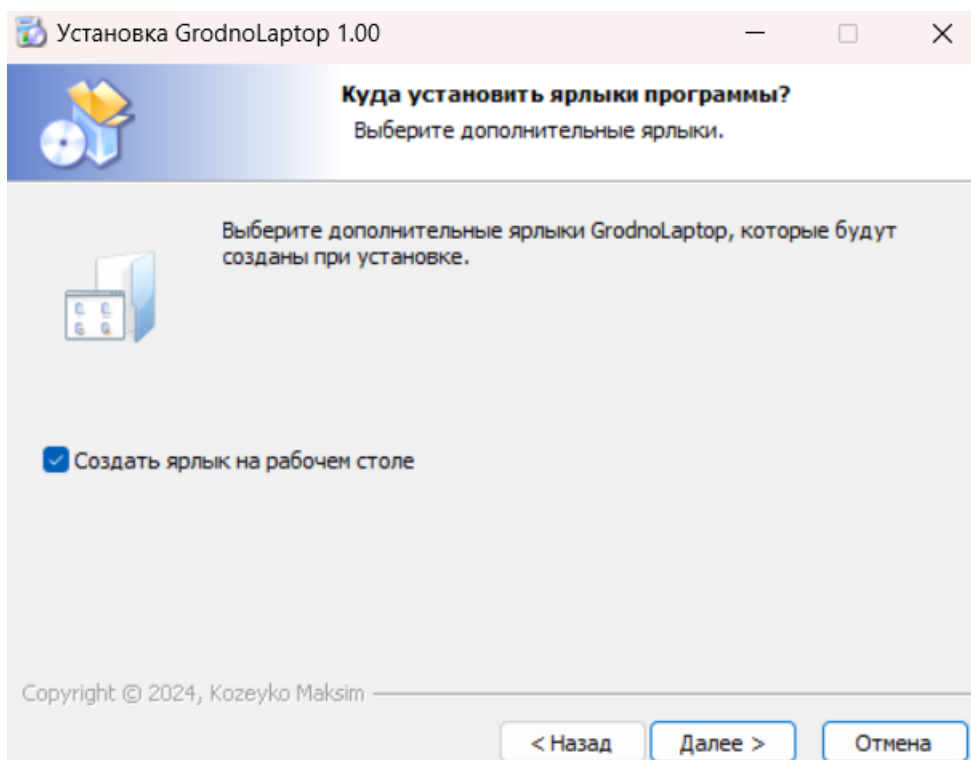


Рисунок 4 – Страница выбора дополнительных ярлыков

После чего откроется страница готовности к установке (рисунок 5), там можно проверить правильность выбранных параметров и нажать кнопку «Установить» для запуска установки программы.

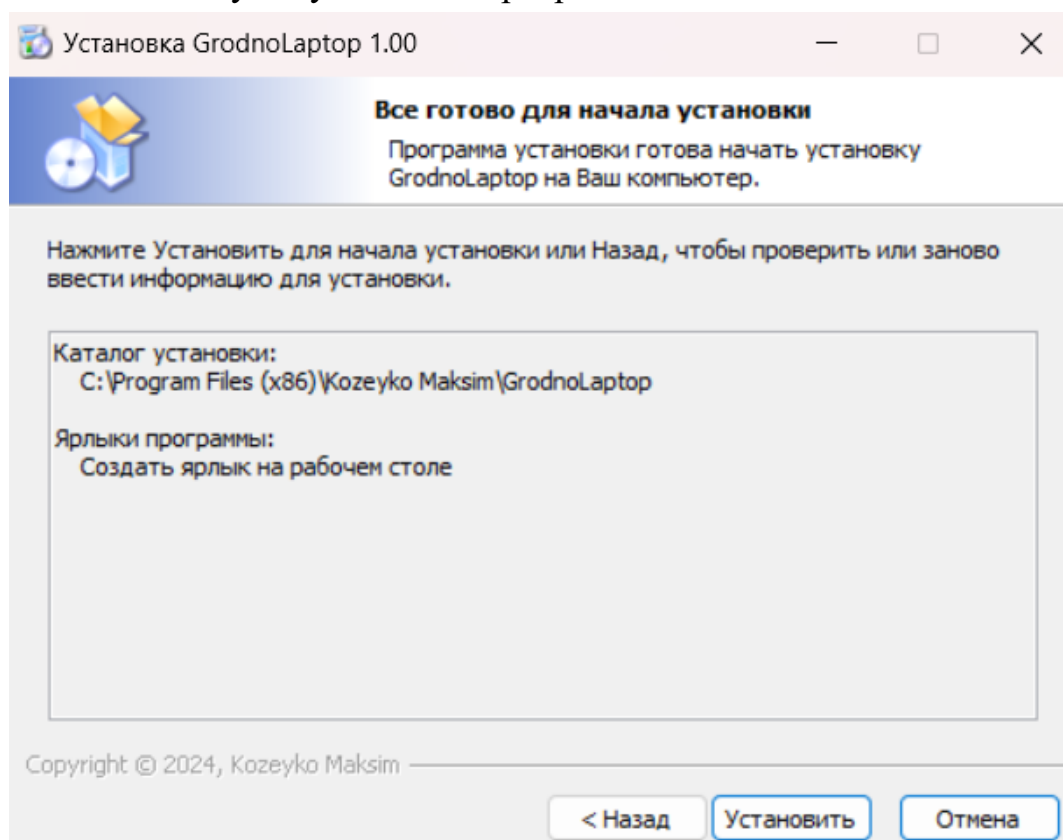


Рисунок 5 – Страница готовности к установке

По окончании установки откроется страница завершения установки (рисунок 6) и сообщит об успешности установки программы.

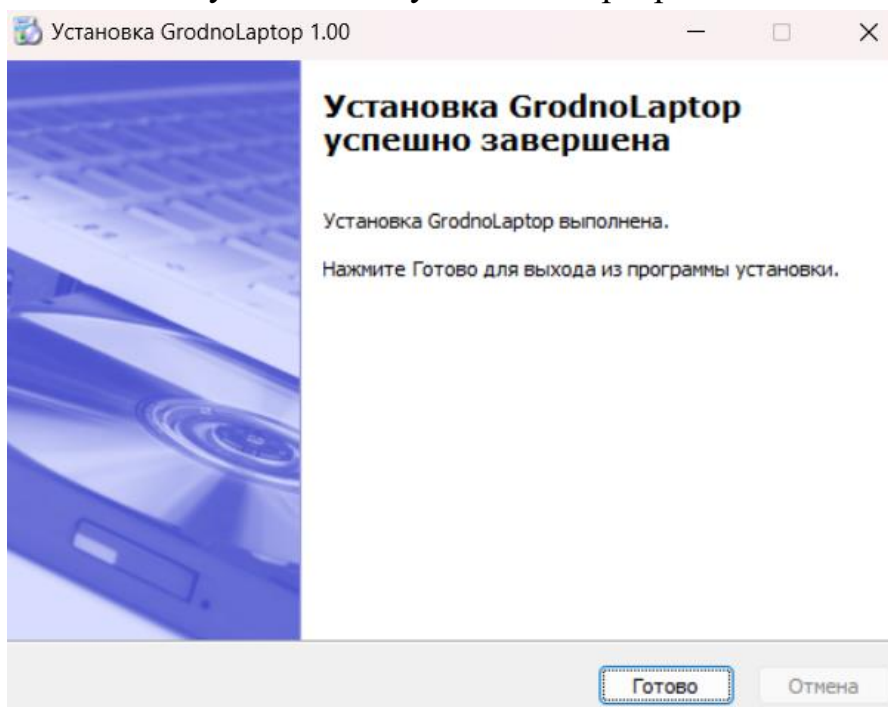


Рисунок 6 – Страница завершения установки

После выполненных действий программа готова к использованию.

Для запуска программы необходимо нажать на ярлык программы на рабочем столе (рисунок 7).

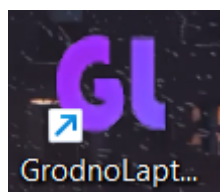


Рисунок 7 – Ярлык программы

При запуске программы вы попадаете на главное окно (рисунок 8).

GrodnoLaptop

GI GRODNOLAPTOP Справка

Заказы Заказ Редактировать Удалить Экспорт

Фильтры

Клиент

Устройство

Услуга

Стоимость

Статус

Номер телефона

Мастер

Номер	Клиент	Устройство	Услуга	Статус	Цена	Дата	Телефон	Мастер
27	Иванов Алексей Петрович	Ноутбук Acer Predator Vortex	Установка windows, замена материнской платы, прочистка	Активен	275	05.05.2024	375336543210	Константин
30	Смирнова Мария Сергеевна	Смартфон Samsung Galaxy Edge S20	Замена дисплея	Активен	150	15.10.2024	375345687234	Константин
31	Кузнецов Виктор Иванович	Планшет Lenovo TabX Pro	Замена корпуса	Активен	100	27.01.2023	375258901234	Михаил
35	Попова Анастасия Михайловна	Моноблок HP Pavilion CoreX	Перепрошивка	Активен	95	01.07.2024	375333456789	Константин
39	Васильев Дмитрий Олегович	Смартфон Honor SkyView 12	Замена камеры	Активен	196	01.12.2024	375291234567	Михаил
43	Соколов Артем Владимирович	Смартфон Realme NeoMax 5	Установка защитного стекла	Выполнен	15	17.12.2024	375339876543	Константин
46	Козлова Ольга Алексеевна	Игровой ноутбук Razer Blade Strike 16	Переустановка windows	Выполнен	70	15.08.2024	375255647890	Михаил
47	Новиков Павел Аркадьевич	Планшет Samsung Galaxy NoteTab S7	Замена батареи	Выполнен	150	15.04.2024	375294561234	Михаил
50	Морозова Елена Викторовна	Графический планшет Wacom SketchPad 13	Замена дисплея	Активен	70	13.02.2023	375332345678	Константин
51	Петрова Ирина Павловна	Ноутбук Asus ZenBook Flux 14	Замена матрицы	Выполнен	217	18.06.2024	375332109876	Константин
52	Киселев Андрей	Умная колонка Yandex Station	Замена чипа	Активен	78	19.03.2024	375253216789	Константин

Рисунок 8 – Главное окно

Главным элементом является таблица (рисунок 9) хранящая заносимые пользователем данные:

Номер	Клиент	Устройство	Услуга	Статус	Цена	Дата	Телефон	Мастер
27	Иванов Алексей Петрович	Ноутбук Acer Predator Vortex	Установка windows, замена материнской платы, очистка	Активен	275	05.05.2024	375336543210	Константин
30	Смирнова Мария Сергеевна	Смартфон Samsung Galaxy Edge S20	Замена дисплея	Активен	150	15.10.2024	375345687234	Константин
31	Кузнецов Виктор Иванович	Планшет Lenovo TabX Pro	Замена корпуса	Активен	100	27.01.2023	375258901234	Михаил
35	Попова Анастасия Михайловна	Моноблок HP Pavilion CoreX	Перепрошивка	Активен	95	01.07.2024	375333456789	Константин
39	Васильев Дмитрий Олегович	Смартфон Honor SkyView 12	Замена камеры	Активен	196	01.12.2024	375291234567	Михаил
43	Соколов Артем Владимирович	Смартфон Realme NeoMax 5	Установка защитного стекла	Выполнен	15	17.12.2024	375339876543	Константин
46	Козлова Ольга Алексеевна	Игровой ноутбук Razer Blade Strike 16	Переустановка windows	Выполнен	70	15.08.2024	375255647890	Михаил
47	Новиков Павел Аркадьевич	Планшет Samsung Galaxy NoteTab S7	Замена батареи	Выполнен	150	15.04.2024	375294561234	Михаил
50	Морозова Елена Викторовна	Графический планшет Wacom SketchPad 13	Замена дисплея	Активен	70	13.02.2023	375332345678	Константин
51	Петрова Ирина Павловна	Ноутбук Asus ZenBook Flux 14	Замена матрицы	Выполнен	217	18.06.2024	375332109876	Константин
52	Киселев Андрей	Умная колонка Yandex Station	Замена чипа	Активен	78	19.03.2024	375253216789	Константин

Рисунок 9 – Таблица с данными

Слева от таблицы находится панель фильтров (рисунок 10), где пользователь может искать по определенным параметрам необходимые заказы:

Фильтры

Клиент

Устройство

Услуга

Стоимость

Статус

Номер телефона

Мастер

Поиск

Очистить

Рисунок 10 – Панель фильтров

В заголовках столбцов предусмотрена стрелка сортировки (Рисунок 11) каждого столбца по убыванию либо по возрастанию:

Номер ▾

Рисунок 11 – Стрелка сортировки
Выше таблицы находиться 4 кнопки (Рисунок 12):

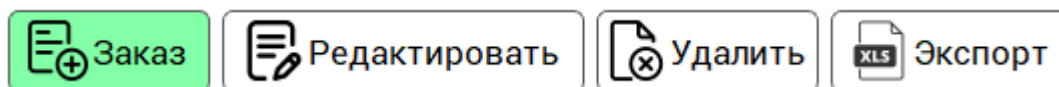


Рисунок 12 – Кнопки

Кнопка "Заказ" открывает меню добавления заказа в котором пользователь может ввести данные для нового заказа. Кнопка "Редактировать" открывает меню редактирования выбранного в таблице заказа. Кнопка "Удалить" удаляет выбранный в таблице заказ. Кнопка "Экспорт" создаёт на рабочем столе Excel-документ с переносом в него содержимого таблицы.

Для завершения работы в программе необходимо нажать кнопку закрытия окна в правом верхнем углу.

Для использования справочной системы необходимо нажать на кнопку «Справка» (рисунок 13).

❓ Справка

Рисунок 13 – Кнопка вызова справки

В открытом окне справки (рисунок 14) можно переходить по разделам для получения необходимой информации.

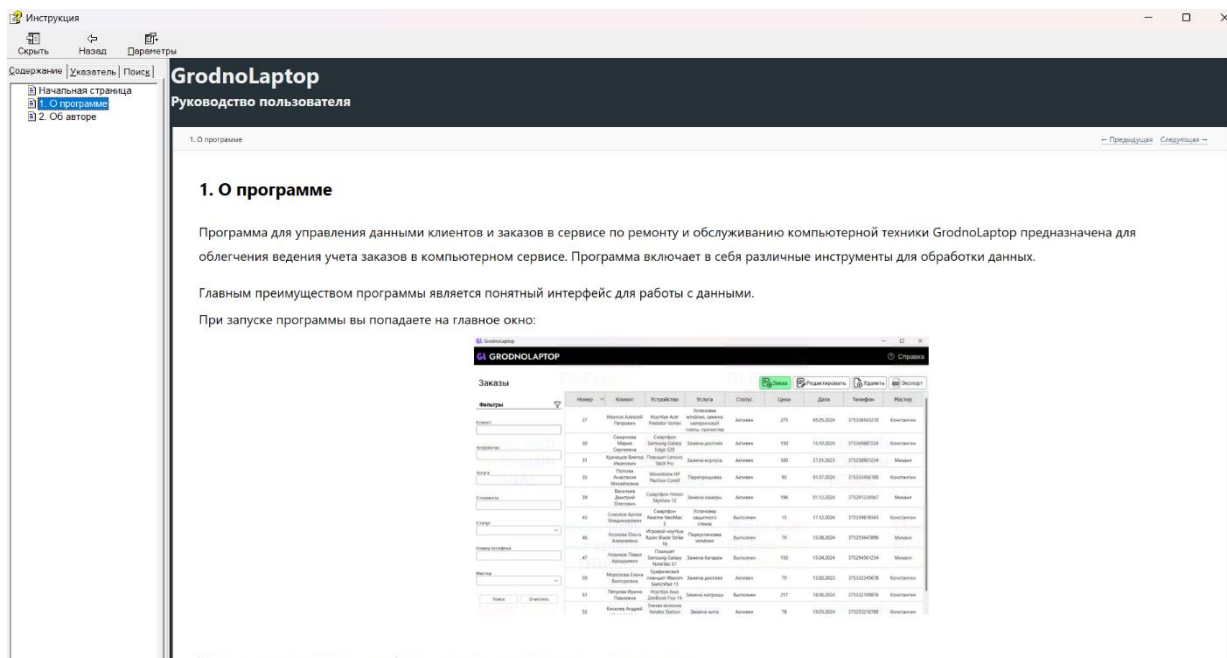


Рисунок 14 – Окно справки

Заключение

В ходе данного проекта была выполнена разработка программного продукта, предназначенного для ведения учета заказов в компьютерном сервисе «GrodnoLaptop». Проект предоставляет приложение позволяющее хранить информацию о заказах, а также взаимодействовать с ней.

В ходе реализации были закреплены и использованы на практике знания языка программирования Python, а также курс предмета «Технология разработки программного обеспечения».

Поставленная задача выполнена в соответствии с требованиями заказчика, созданы и протестированы все необходимые функции и компоненты проекта.

В ходе тестирования все исключительные ситуации были обработаны. Проект работает без сбоев и ошибок. В поставленной задаче был реализован простой и понятный пользовательский интерфейс.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

Список использованных источников

1. Qt for Python [Электронный ресурс]. – Режим доступа: <https://doc.qt.io/qtforpython-6/index.html> – Дата доступа: 15.11.2024.
2. Руководство по CSS - CSS: каскадные таблицы стилей | MDN [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/CSS/Reference> – Дата доступа: 19.11.2024.
3. Обработка баз данных SQL с помощью PyQt: Основы [Электронный ресурс]. – Режим доступа: <https://is20-2019.susu.ru/utkinaea/2021/03/15/obrabotka-baz-dannyh-sql-s-pomoshhyu-pyqt-osnovy/> – Дата доступа: 20.11.2024.
4. Работа с SQLite в Python (для чайников) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/754400/> – Дата доступа: 20.11.2024.
5. Qt Widgets Designer Manual [Электронный ресурс]. – Режим доступа: <https://doc.qt.io/qt-6/qtdesigner-manual.html> – Дата доступа: 17.11.2024.
6. Разработка Desktop приложений на Python и ... [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/799037/> – Дата доступа: 17.11.2024.