МИНИСТЕРСТВО ОБРАЗОВАНИЯ И МОЛОДЕЖНОЙ ПОЛИТИКИ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГАПОУ СО «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Тестирование кода информационной системы»

ТЕМА: «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 0 | 2 | 0 | 7 | - КП - |  |  |  |  |  |  |  |  |  |  |  |

ОЦЕНКА: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РУКОВОДИТЕЛЬ КУРСОВОГО ПРОЕКТА:

Грибова А.В.

СТУДЕНТ гр. ИП-01

Данилюк Е.М.

|  |
| --- |
| 2021 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № стр. | Формат | Обозначение | Наименование | Кол.  листов | № экз. | Примеч. |
| 1 | А4 | ЕМК 09.02.07. КП 3518-ИП81-18 ПЗ | Пояснительная записка | 50 | - |  |
| 2 |  | ЕМК 09.02.07.КП 3518- ИП81-18 ИС | «Тема» | 1 |  | CD-диск |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |
| 19 |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |
| 21 |  |  |  |  |  |  |
| 22 |  |  |  |  |  |  |
| 23 |  |  |  |  |  |  |
| 24 |  |  |  |  |  |  |
| 25 |  |  |  |  |  |  |
|  | | | | | | |

СОДЕРЖАНИЕ

Оглавление

[СОДЕРЖАНИЕ 3](#_Toc137486351)

[ВВЕДЕНИЕ 6](#_Toc137486352)

[Актуальность исследования 6](#_Toc137486353)

[Цель исследования 6](#_Toc137486354)

[Объект исследования 6](#_Toc137486355)

[Предмет исследования 6](#_Toc137486356)

[Задачи курсового проектирования: 6](#_Toc137486357)

[1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 8](#_Toc137486358)

[2. АНАЛИТИЧЕСКАЯ ЧАСТЬ 9](#_Toc137486359)

[Цели, функции и задачи системы: 9](#_Toc137486360)

[Модули системы 9](#_Toc137486361)

[Описание предметной области системы 9](#_Toc137486362)

[Определение пользователей системы: 10](#_Toc137486363)

[Проектные решения по реализации тестирования приложения: 11](#_Toc137486364)

[Проектные решения по реализации интерфейса приложения 20](#_Toc137486365)

[3. ПРОЕКТНАЯ ЧАСТЬ 21](#_Toc137486366)

[Модель информационной системы 21](#_Toc137486367)

[Файловая структура приложения: 28](#_Toc137486368)

[Реализация кода приложения 31](#_Toc137486369)

[Разработка dll-библиотек 31](#_Toc137486370)

[Работа с системой контроля версий 37](#_Toc137486371)

[Руководство по установке программы 38](#_Toc137486372)

[Документация по работе с информационной системой 40](#_Toc137486373)

[4. ГЛОССАРИЙ 47](#_Toc137486374)

[5. СПИСОК ИСТОЧНИКОВ 48](#_Toc137486375)

[ЗАКЛЮЧЕНИЕ 49](#_Toc137486376)

[ПРИЛОЖЕНИЕ А 50](#_Toc137486377)

[ПРИЛОЖЕНИЕ Б 51](#_Toc137486378)

ВВЕДЕНИЕ

Если начать рассматривать современные компании, которые работают с заявками от клиентов, то можно заметить, что каждая из них действует по своему алгоритму и использует персональное программное обеспечение, которое позволяет автоматизировать процесс обработки и упрощает процедуру отчета. Так или иначе, в современном мире без хорошего инструмента автоматизации уже далеко не уйти. Показательным примером станет то, что в момент открытия у одной компании клиентов очень мало, и учет, как таковой, вести проблем не составит. Но компания продолжает работать, клиентская база растет, заказы копятся, и что-то понять среди всего этого уже очень проблематично. В результате за неимением хорошей системы обработки заявок вся работа встает, заказы начинают отдаваться не вовремя, теряться, и вообще происходит хаос. Нетрудно догадаться, что ждет такую фирму, если она не пересмотрит свою политику работы.

Поэтому для достижения требуемого результата в работе мы решили ряд задач, которые в той или иной мере помогут достичь основной цели. По итогу, в работе мы получили готовый программный продукт, который должен выполнять возложенный на него спектр задач, а также поддерживать возможность модификации и дополнений.

Актуальность исследования: актуальность данного проекта заключается в том, что в современном мире, очень быстро развиваются информационные технологии, благодаря чему стало возможно автоматизировать работу сотрудников в сфере ремонта и сборки ПК.

Цель исследования: разработать АРМ приложение, которое позволит улучшить качество и эффективность работу сотрудников в ремонтных и компьютерных магазинах.

Объект исследования: Объектом исследования является Рабочее Место сотрудника компьютерного магазина оснащенный нашим (АРМ) приложением для упрощения работы учета заказов на ремонт и учету комплектующих ПК. АРМ по учету заказов на ремонт может использоваться в различных компьютерных и ремонтных магазинах, предоставляет широкий спектр функциональности, включая учет заказов на ремонт и сборку ПК.

Предмет исследования: Программы Visual Studio, MySQL Management Studio, Сайт GitHub

### Задачи курсового проектирования:

1. Провести анализ предметной области, изучив существующие информационные системы в сфере комп и рем магазинах и их функциональные возможности.
2. Основные требования и потребности ремонтных и компьютерных магазинов, выявить постоянные проблемы с которыми сталкиваются сотрудники в повседневной работы.
3. Разработать и реализовать АРМ по учету заказов на ремонт, учитывая выявленные требования и потребности. С этими задачами разработать функциональность приложения, создать простой и интуитивно понятный интерфейс пользователя.
4. Протестировать АРМ по учету заказов на ремонт и проверить его эффективность и преимущества сравнивая с приложениями конкурентов. На основе тестирования проанализировать рекомендации с учетом дальнейшего совершенствования

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

### Цели, функции и задачи системы:

Цели создания АРМ по учету заказов являются:

1. Оптимизация среды работы сотрудников компьютерных и ремонтных магазинов
2. Создание удобного приложения для работы
3. Улучшение качества работы

Функции АРМ по учету заказов являются:

1. добавление, редактирование, удаление информации о заказах на ремонт компьютера.
2. добавление, редактирование заказа клиента на сборку компьютера и всей необходимой информацией о комплектующих этого заказа.
3. автоматизированный вывод гарантийного талона, чека, накладной на заказ клиента.
4. Ввод новых пользователей и сотрудников.

Задачи АРМ по учету заказов являются:

1. Разработка простого пользовательского интерфейса
2. Управление заказами на ремонт и сборку

Модули системы:

- Авторизация

- Главная страница

- Страница добавления клиента

- Страница добавления сотрудников

- Страница управления заказами

- Страница учета заказов

- Страница заказов на ремонт

- Страница заказов на сборку

- Страница просмотра заказов

- страница управления заказами

-страница управления комплектующими

Описание предметной области системы:

Для разработки проекта была выбрана область ремонта техники. Система АРМ по учету заказов предназначена для упрощения и оптимизации работы сотрудников компьютерных и ремонтных магазинов. Исходные данные, которые были взяты в основу проекта - это данные, полученные у работников в сфере ремонта и сборки ПК. Были изучены различные источники информации и интернет ресурсы. Были проведены опросы и встречи с сотрудниками. С помощью опросов мы получили понимание какие есть проблемы и чего не хватает сотрудникам компьютерных и ремонтных магазинов.

Были изучены приложения для исходных данных проекта: Fillin, itsm365, serviceapp, prostoysoft.

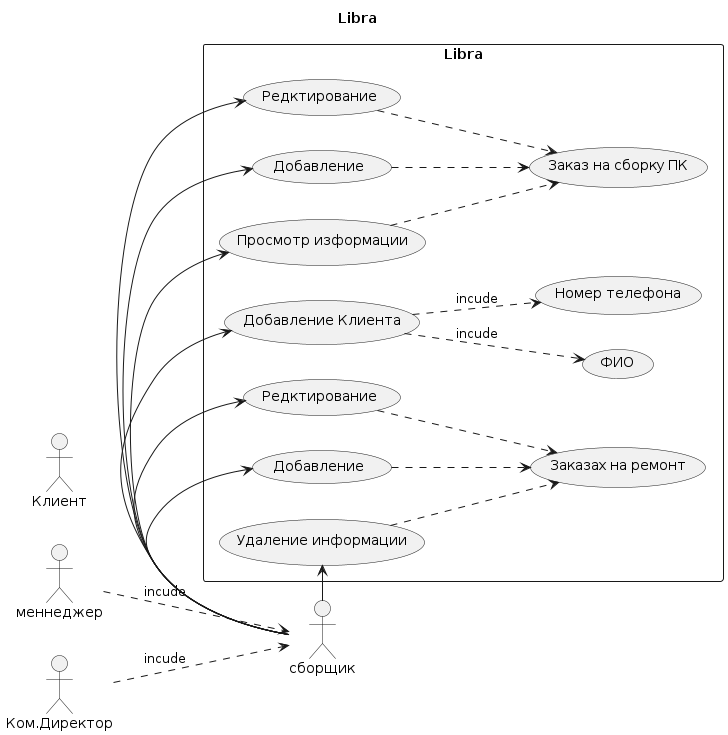
### Определение пользователей системы:

Пользователи системы

1. коммерческий директор
2. Менеджеры
3. Сборщик Пк

Разграничения доступа в приложении нет, все имеют одинаковый доступ к действиям внутри приложения

Диаграмма функционала приложения:



Связи между прецендентами неверные

Выделить прецендент «формирование заказа на ремонт»

### Проектные решения по реализации тестирования приложения:

|  |  |
| --- | --- |
| Requirements ID | Описание функции |
| Req\_1 | Проверка правильности заполнения ФИО клиента |
| Req\_2 | Проверка правильности заполнения номера телефона клиента |
| Req\_3 | Проверка правильности названия продукта |
| Req\_4 | Проверка правильности названия артикля |
| Req\_5 | Проверка правильности названия серийного номера продукта |
| Req\_6 | Проверка правильности стоимости продукта |
| Req\_7 | Проверка правильности заполнения ФИО сотрудника |
| Req\_8 | Проверка правильности заполнения логина сотрудника |
| Req\_9 | Проверка правильности заполнения пароля сотрудника |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| test case ID | описание объекта тестирован ия (requiremen ts) | стратегия тестирования (вид тестирования ) | Последовательност ь проведения тестирования в разрезе запланированных фаз разработки | Критерии начала тестировани я | Критерии окончани я тестирова ния |
| TC\_1 | Ввод данных при пустых значениях ФИО |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_2 | Ввод данных ФИО на английском языке |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_3 | Ввод данных ФИО с цифрами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_4 | Ввод данных ФИО с спец. символами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_5 | Ввод данных ФИО с маленькой буквы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_6 | Ввод данных ФИО со всеми большими буквами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_7 | Ввод данных ФИО с спец. символами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_8 | Ввод данных ФИО правильного формата |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_9 | Ввод данных при пустых значениях номера телефона |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_10 | Ввод данных прописными латинскими буквами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_11 | Ввод данных заглавных прописных латинских букв |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_12 | Ввод данных прописной кириллицей |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_13 | Ввод данных заглавных букв прописной кириллицей |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_14 | Ввод данных спец. символами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_15 | Ввод данных при неправильном формате телефона с плюсом |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_16 | Ввод данных при неправильном формате телефона с 8 в начале |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_17 | Ввод данных при неправильном формате телефона с пробелами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_18 | Ввод данных при правильном вводе телефона |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_19 | Ввод данных латинскими символами в нижнем регистре |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_20 | Ввод данных латинскими символами в верхнем регистре |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_21 | Ввод данных кириллицей в нижнем регистре |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_22 | Ввод данных кириллицей в верхнем регистре |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_23 | Ввод данных кириллицей с символами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_24 | Ввод данных латинскими символами с цифрами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_25 | Ввод данных латинскими символами и кириллицей с пробелами и цифрами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_26 | Ввод данных при пустых |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_27 | Ввод данных кириллицей |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_28 | Ввод данных артикула с пробелами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_29 | Ввод данных артикула с буквами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_30 | Ввод данных артикула с пробелами и цифрами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_31 | Ввод данных артикула с пробелами, цифрами и тире |  |  | Готовность тестового кода | Результат кода должен совпадать с ожидаемым результатом |
| TC\_32 | Ввод данных номера телефона буквами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_33 | Ввод данных номера телефона с буквами и цифрами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_34 | Ввод данных номера телефона с буквами, цифрами и тире |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_35 | Ввод данных неправильного формата цены без учета копеек |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_36 | Ввод данных цены с точкой |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_37 | Ввод данных цены с нулем в начале |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_38 | Ввод данных цены с большим кол-вом знаков после запятой |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_39 | Ввод данных цены правильного формата |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_40 | Ввод данных ФИО с пробелами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_41 | Ввод данных при слишком коротком логине |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_42 | Ввод данных при слишком длинном логине |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_43 | Ввод данных логина с маленькой латиницы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_44 | Ввод данных логина с большойй латиницы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_45 | Ввод данных логина с большой и маленькой латиницы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_46 | Ввод данных логина с большой и маленькой латиницы с цифрами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_47 | Ввод данных логина пустыми значениями |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_48 | Ввод данных логина пробелами |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_49 | Ввод данных логина кириллицей |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_50 | Ввод данных логина с большой кириллицы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_51 | Ввод данных логина с маленькой кириллицы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_52 | Ввод данных логина с большой и маленькой кириллицы |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |
| TC\_53 | Ввод данных логина только маленькой латиницей |  |  | Готовность тестового кода. | Результат кода должен совпадать с ожидаемым результатом |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Requirements ID | Req\_1 | Req\_2 | Req\_3 | Req\_4 | Req\_5 | Req\_6 | Req\_7 | Req\_8 | Req  \_9 |
| test case ID |  |  |  |  |  |  |  |  |  |  |
| TC\_1 |  | + |  |  |  |  |  |  |  |  |
| TC\_2 |  | + |  |  |  |  |  |  |  |  |
| TC\_3 |  | + |  |  |  |  |  |  |  |  |
| TC\_4 |  | + |  |  |  |  |  |  |  |  |
| TC\_5 |  | + |  |  |  |  |  |  |  |  |
| TC\_6 |  | + |  |  |  |  |  |  |  |  |
| TC\_7 |  | + |  |  |  |  |  |  |  |  |
| TC\_8 |  | + |  |  |  |  |  |  |  |  |
| TC\_9 |  |  | + |  |  |  |  |  |  |  |
| TC\_10 |  | + |  |  |  |  |  | + |  |  |
| TC\_11 |  | + |  |  |  |  |  | + |  |  |
| TC\_12 |  | + |  |  |  |  |  | + |  |  |
| TC\_13 |  | + |  |  |  |  |  | + |  |  |
| TC\_14 |  | + |  |  |  |  |  |  |  |  |
| TC\_15 |  |  | + |  |  |  |  |  |  |  |
| TC\_16 |  |  | + |  |  |  |  |  |  |  |
| TC\_17 |  |  | + |  |  |  |  |  |  |  |
| TC\_18 |  |  | + |  |  |  |  |  |  |  |
| TC\_19 |  |  |  | + |  |  |  |  |  |  |
| TC\_20 |  |  |  | + |  |  |  |  |  |  |
| TC\_21 |  |  |  | + |  |  |  |  |  | + |
| TC\_22 |  |  |  | + |  |  |  |  |  | + |
| TC\_23 |  |  |  | + |  |  |  |  |  | + |
| TC\_24 |  |  |  | + |  |  |  |  |  | + |
| TC\_25 |  |  |  | + |  |  |  |  |  | + |
| TC\_26 |  |  |  | + |  |  |  |  |  | + |
| TC\_27 |  |  |  | + |  |  |  |  |  | + |
| TC\_28 |  |  |  |  | + |  |  |  |  | + |
| TC\_29 |  |  |  |  | + |  |  |  |  |  |
| TC\_30 |  |  |  |  | + |  |  |  |  |  |
| TC\_31 |  |  |  |  | + |  |  |  |  |  |
| TC\_32 |  |  | + |  |  |  |  |  |  |  |
| TC\_33 |  |  | + |  |  |  |  |  |  |  |
| TC\_34 |  |  | + |  |  |  |  |  |  |  |
| TC\_35 |  |  |  |  |  |  | + |  |  |  |
| TC\_36 |  |  |  |  |  |  | + |  |  |  |
| TC\_37 |  |  |  |  |  |  | + |  |  |  |
| TC\_38 |  |  |  |  |  |  | + |  |  |  |
| TC\_39 |  |  |  |  |  |  | + |  |  |  |
| TC\_40 |  | + |  |  |  |  |  | + |  |  |
| TC\_41 |  |  |  |  |  |  |  |  | + |  |
| TC\_42 |  |  |  |  |  |  |  |  | + |  |
| +TC\_43 |  |  |  |  |  |  |  |  | + |  |
| TC\_44 |  |  |  |  |  |  |  |  | + |  |
| TC\_45 |  |  |  |  |  |  |  |  | + |  |
| TC\_46 |  |  |  |  |  |  |  |  | + |  |
| TC\_47 |  |  |  |  |  |  |  |  | + |  |
| TC\_48 |  |  |  |  |  |  |  |  | + |  |
| TC\_49 |  |  |  |  |  |  |  |  | + |  |
| TC\_50 |  |  |  |  |  |  |  |  | + |  |
| TC\_51 |  |  |  |  |  |  |  |  | + |  |
| TC\_52 |  |  |  |  |  |  |  |  | + |  |
| TC\_53 |  |  |  |  |  |  |  |  | + |  |

Проектные решения по реализации интерфейса приложения (разработка руководства по стилю): Приведение всех кнопок к одному цвету является отличительным свойством приложения, интерфейс состоящий из 3 цветов делает его наиболее простым и лаконичным.

Цвет кнопок ярко выражен по сравнению с темой приложения, что делает взаимодействие пользователя с приложением простым и удобным.

Руководство по стилю:

<https://www.figma.com/file/aNBrth3F2zHSHpjNqt7TLj/Untitled?type=design&node-id=0-1&t=CVIDT0Z6XpupKu7Q-0>

прототип приложения:

<https://www.figma.com/file/1jt0qRrcxmG4hMbnqTDUk0/Untitled?type=design&t=S3gAi1Uyyktv2vrG-0>

3. ПРОЕКТНАЯ ЧАСТЬ

### Модель информационной системы



Схема базы данных

Таблица Clients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | idClients | int | N | Автозаполнение  с шагом 1 | Id клиента |
|  | ClientSurname | nvarchar(200) | N |  | Фамилия клиента |
|  | ClientName | nvarchar(200) | N |  | Имя клиента |
|  | ClientPatronymic | nvarchar(200) | N |  | Отчество клиента |
|  | ClientPhoneNumber | date | N |  | Номер телефона |

Таблица Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdUser | int | N |  | ID пользователя |
|  | UserName | nvarchar(200) | N |  | Имя пользователя |
|  | UserSurname | nvarchar(200) | N |  | Фамилия пользователя |
|  | UserPatronymic | nvarchar(200) | N |  | Отчество пользователя |
|  | UserLogin | nvarchar(200) | N |  | Лоигн пользователя |
|  | UserPassword | nvarchar(200) | N |  | Пароль пользователя |
| FK | UserRoleId | nvarchar(200) | N |  | ID-роль пользователя |

Таблица Orders

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdOrders | int | N | Автозаполнение  с шагом 1 | ID заказа |
|  | OrderDateStart | date | N |  | Дата начала заказа |
|  | OrderDateEnd | date | Y |  | Дата окончания заказа |
| FK | ClientId | int | N |  | ID клиента |
| FK | UserId | int | N |  | ID пользователя |
| FK | ProductId | int | N |  | ID продукта |
| FK | OrderTypeId | int | N |  | ID типа продукта |
|  | OrderCost | float | Y |  | Цена заказ |

Таблица OrdersType

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | OrdersTypesId | int | N | Автозаполнение  с шагом 1 | ID типа продукта |
|  | OrdersTypesName | nvarchar(200) | N |  | Наименование продукта |

Таблица Product

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdProduct | int | N | Автозаполнение  с шагом 1 | ID продукта |
|  | ProductName | nvarchar(500) | N |  | Наименование продукта |
|  | ProductArticle | nvarchar(500) | N |  | Артикул продукта |
|  | ProductSerialNumber | nvarchar(500) | N |  | Серийный номер продукта |

Таблица ComponentsName

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdComponentName | int | N |  | ID компонента |
| FK | ComponentName | nvarchar(200) | N |  | Наименование компонента |

Таблица ProductsComponents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdProductComponent | int | N |  | ID продукта компонента |
|  | ProductId | int | N |  | ID продукта |
| FK | ComponentId | int | N |  | ID компонета |

Таблица Components

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdComponent | int | N | Автозаполнение  с шагом 1 | ID компонента |
|  | ComponentNameId | int | N |  | ID наименования компонента |
|  | ComponentCost | Float | Y |  | Цена компонента |

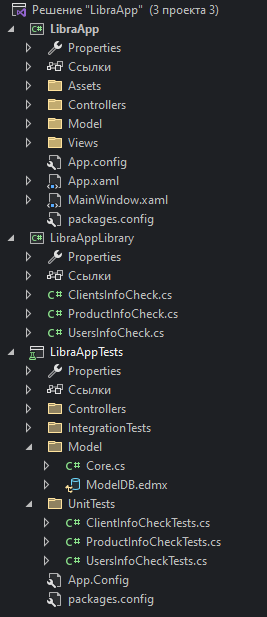
Таблица CharecteristicNames

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdCharacteristicsName | int | N | Автозаполнение  с шагом 1 | ID наименование характеристики |
|  | CharacteristicsName | nvarchar(200) | Y |  | Наименование характеристики |

Таблица Characteristic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ключ | Имя поля | Тип данных | Null | Дополнительно | Описание |
| PK | IdCharacteristics | int | N | Автозаполнение  с шагом 1 | ID характеристики |
| FK | CharacteristicNameId | nvarchar(50) | N |  | Наименование характеристики |
|  | CharacteristicValue | int | N |  | Данные характеристики |

### Файловая структура приложения:

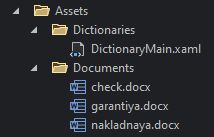


В папке Assets находятся все ресурсы для дизайна приложения.

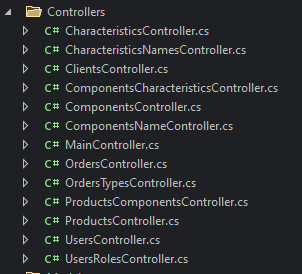
В папке Dictionaries содержит словари ресурсов для дизайна приложения.

В папке Fonts содержит шрифты для текста в приложении.

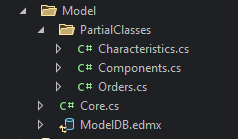
В папке Documents содержатся Docx файлы для вывода гарантий, чеков, накладной



В папке Controllers дополняет классы созданные с entity frameworks по нашей БД.



В папке Model держит базу данных, классы для таблиц баз данных



В папке Page содержит страницы приложения.

- AddAssemblyOrderPage – Страница добавления заказа на сборку

- AddComponentPage – Станица добавления клиента

- AddComponentPage – Страница добавления комплектующих

- AddRepairOrderPage – Страница добавления заказа на ремонт

- AddUserPage – Страница добавления работника

- AuthPage – Страница входа

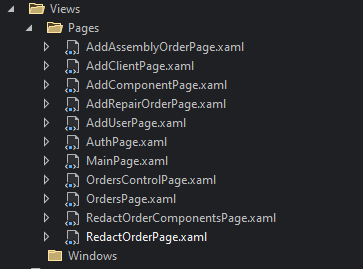
- MainPage – Главная страница

- OrdersControlPage – Страница учета заказов

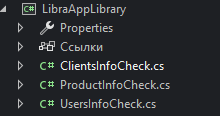
- OrdersPage – Страница просмотра заказов

- RedactOrderComponentsPage – Страница управления комплектующими

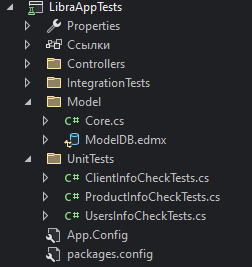
- RedactOrderPage – Страница управления заказами



В LibraAppTests содержатся функции для проверки ввода данных

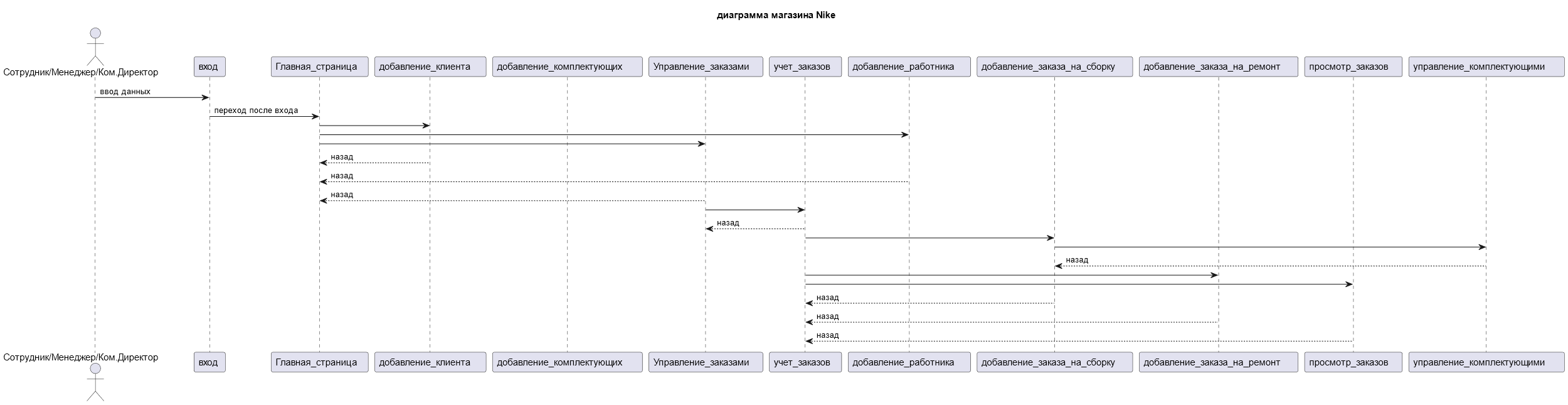


В LibraAppTests содержатся тесты для проверки ввода данных в самом приложении



Реализация кода приложения:

Диаграмма последовательности:



### Разработка dll-библиотек

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода ФИО |
| Название теста | CheckFIOInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен ФИО или нет |
| Тестовые данные | Правильный ввод данных:   1. string entry = "Сидоров";   не правильный ввод данных:   1. string entry = "ivanov"; 2. string entry = "Сидоров3"; 3. string entry = "Сидоров#"; 4. string entry = "сидоров"; 5. string entry = "СИДОРОВ"; 6. string entry = ""; |
| Ожидаемый результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False |
| Фактический результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. Pass |
| Комментарии | Названия тестовых методов   1. CheckFIOInfo\_EmptyFIOInfo\_ReturnFalse 2. CheckFIOInfo\_LatinFIOInfo\_ReturnFalse 3. CheckFIOInfo\_DigitFIOInfo\_ReturnFalse 4. CheckFIOInfo\_SymbolsFIOInfo\_ReturnFalse 5. CheckFIOInfo\_LowerFIOInfo\_ReturnFalse 6. CheckFIOInfo\_UpperFIOInfo\_ReturnFalse 7. CheckFIOInfo\_RightFIOInfo\_ReturnTrue |

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода номера телефона |
| Название теста | CheckPhoneInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен номер телефона или нет |
| Тестовые данные | Правильный ввод данных:   1. string entry = "79456473625";   не правильный ввод данных:   1. string entry = ""; 2. string entry = "asdasd"; 3. string entry = "ASDASD"; 4. string entry = "фывфыв"; 5. string entry = "ФЫВФЫВ"; 6. string entry = "%%^&$#@!!@#$"; 7. string entry = "+79456473625"; 8. string entry = "89456473625"; 9. string entry = "7 945 647 36 25"; |

|  |  |
| --- | --- |
| Ожидаемый результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Фактический результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. Pass 8. Pass 9. Pass 10. pass |
| Комментарии | Названия тестовых методов   1. CheckPhoneInfo\_EmptyPhoneInfo\_ReturnFalse 2. CheckPhoneInfo\_LatinPhoneInfo\_ReturnFalse 3. CheckPhoneInfo\_LatinUpperPhoneInfo\_ReturnFalse 4. CheckPhoneInfo\_CyrillicLowerPhoneInfo\_ReturnFalse 5. CheckPhoneInfo\_CyrillicUpperPhoneInfo\_ReturnFalse 6. CheckPhoneInfo\_SynbolsPhoneInfo\_ReturnFalse 7. CheckPhoneInfo\_PlusPhoneInfo\_ReturnFalse 8. CheckPhoneInfo\_EightPhoneInfo\_ReturnFalse 9. CheckPhoneInfo\_SpacePhoneInfo\_ReturnFalse 10. CheckPhoneInfo\_RightPhoneInfo\_ReturnTrue |

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода наименования продукта |
| Название теста | CheckNameInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введено название продукта |
| Тестовые данные | Правильный ввод данных:   1. string entry = "samsung"; 2. string entry = "Samsung AE"; 3. string entry = "самсунг"; 4. string entry = "Самсунг"; 5. string entry = "Самсунг 07"; 6. string entry = "Samsung 07"; 7. string entry = "Монитор Samsung 07";   Не правильный ввод данных   1. string entry = ""; 2. string entry = "№;% !"; |
| Ожидаемый результат | 1. True 2. True 3. True 4. True 5. True 6. True 7. True 8. False 9. False |
| Фактический результат | 1. True 2. True 3. True 4. True 5. True 6. True 7. True 8. False 9. False |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. Pass 8. Pass 9. pass |
| Комментарии | Названия тестовых методов   1. CheckNameInfo\_EmptyNameInfo\_ReturnFalse 2. CheckNameInfo\_SymbolsNameInfo\_ReturnFalse 3. CheckNameInfo\_LatinLowerNameInfo\_ReturnTrue 4. CheckNameInfo\_LatinUperAndLowerNameInfo\_ReturnTrue 5. CheckNameInfo\_CyrillicLowerNameInfo\_ReturnTrue 6. CheckNameInfo\_CyrillicLowerAndUpperNameInfo\_ReturnTrue 7. CheckNameInfo\_CyrillicAndDigitNameInfo\_ReturnTrue 8. CheckNameInfo\_LatinAndDigitNameInfo\_ReturnTrue 9. CheckNameInfo\_LatinAndCyrillicNumbersSpaceNameInfo\_ReturnTrue |

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода наименования артикля |
| Название теста | CheckArticleInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен артикль |
| Тестовые данные | 1. string entry = ""; 2. string entry = "СУПЕРартикль"; 3. string entry = "article"; 4. string entry = "ARTICLE ARTICLE"; 5. string entry = "ARTICLE"; 6. string entry = "ARTICLE1"; 7. string entry = "ARTICLE-1-ARTICLE"; |
| Ожидаемый результат | 1. False 2. False 3. False 4. False 5. True 6. True 7. True |
| Фактический результат | 1. False 2. False 3. False 4. False 5. True 6. True 7. True |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. Pass 8. Pass |
| Комментарии | Названия тестовых методов   1. CheckArticleInfo\_EmptyArticleInfo\_ReturnFalse 2. CheckArticleInfo\_CyrillicArticleInfo\_ReturnFalse 3. CheckArticleInfo\_LatinLowerArticleInfo\_ReturnFalse 4. CheckArticleInfo\_RightArticleWithSpacesInfo\_ReturnFalse 5. CheckArticleInfo\_RightArticleWithLatinOnlyInfo\_ReturnTrue 6. CheckArticleInfo\_RightArticleWithDigitInfo\_ReturnTrue 7. CheckArticleInfo\_RightArticleInfo\_ReturnTrue |

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода написания номера телефона |
| Название теста | CheckNumberInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен артикль |
| Тестовые данные | 1. string entry = ""; 2. string entry = "СУПЕРномер"; 3. string entry = "number"; 4. string entry = "NUMBER NUMBER"; 5. string entry = "NUMBER"; 6. string entry = "NUMBER1"; 7. string entry = "NUMBER-1-NUMBER"; |
| Ожидаемый результат | 1. False 2. False 3. False 4. False 5. True 6. True 7. True |
| Фактический результат | 1. False 2. False 3. False 4. False 5. True 6. True 7. True |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. pass |
| Комментарии | Названия тестовых методов   1. CheckNumberInfo\_EmptyNumberInfo\_ReturnFalse 2. CheckNumberInfo\_CyrillicNumberInfo\_ReturnFalse 3. CheckNumberInfo\_LatinLowerNumberInfo\_ReturnFalse 4. CheckNumberInfo\_RightNumberWithSpacesInfo\_ReturnFalse 5. CheckNumberInfo\_RightNumberWithLatinOnlyInfo\_ReturnTrue 6. CheckNumberInfo\_RightNumberWithDigitInfo\_ReturnTrue 7. CheckNumberInfo\_RightNumberInfo\_ReturnTrue |

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода наименования цены |
| Название теста | CheckCostInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен артикль |
| Тестовые данные | Правильный ввод данных:   1. string entry = "220,00";   Не правильный ввод данных:   1. string entry = ""; 2. string entry = "приветПРИВЕТ"; 3. string entry = "helloHELLO"; 4. string entry = "@#$%"; 5. string entry = " "; 6. string entry = "200"; 7. string entry = "200.00"; 8. string entry = "020,00"; 9. string entry = "220,002"; |
| Ожидаемый результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Фактический результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |

|  |  |
| --- | --- |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. pass 6. pass 7. pass 8. pass 9. pass 10. pass |
| Комментарии | Названия тестовых методов   1. CheckCostInfo\_EmptyCostInfo\_ReturnFalse 2. CheckCostInfo\_CyrillicUpperAndLowerCostInfo\_ReturnFalse 3. CheckCostInfo\_LatinUpperAndLowerCostInfo\_ReturnFalse 4. CheckCostInfo\_SymbolsCostInfo\_ReturnFalse 5. CheckCostInfo\_SpaceCostInfo\_ReturnFalse 6. CheckCostInfo\_WrongFormatCostInfo\_ReturnFalse 7. CheckCostInfo\_DotCostInfo\_ReturnFalse 8. CheckCostInfo\_ZeroStartCostInfo\_ReturnFalse 9. CheckCostInfo\_MoreThanTwoCostInfo\_ReturnFalse 10. CheckCostInfo\_RightFormatTwoCostInfo\_ReturnTrue |

|  |  |
| --- | --- |
| Тестовый пример # | Тест на проверку правильного ввода данных логина |
| Название теста | CheckLoginInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен логин |
| Тестовые данные | 1. string entry = "AZz0"; 2. string entry = ""; 3. string entry = "логинЛОГИН"; 4. string entry = "@#$"; 5. string entry = " "; 6. string entry = "az"; 7. string entry = "azzzazazazzaczcvhfghjhhjbgj"; 8. string entry = "azz"; 9. string entry = "AZZ"; 10. string entry = "AZz"; |
| Ожидаемый результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Фактический результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. Pass 8. Pass 9. pass |
| Комментарии | Названия тестовых методов   1. CheckCostInfo\_EmptyCostInfo\_ReturnFalse 2. CheckCostInfo\_CyrillicUpperAndLowerCostInfo\_ReturnFalse 3. CheckCostInfo\_LatinUpperAndLowerCostInfo\_ReturnFalse 4. CheckCostInfo\_SymbolsCostInfo\_ReturnFalse 5. CheckCostInfo\_SpaceCostInfo\_ReturnFalse 6. CheckCostInfo\_WrongFormatCostInfo\_ReturnFalse 7. CheckCostInfo\_DotCostInfo\_ReturnFalse 8. CheckCostInfo\_ZeroStartCostInfo\_ReturnFalse 9. CheckCostInfo\_MoreThanTwoCostInfo\_ReturnFalse 10. CheckCostInfo\_RightFormatTwoCostInfo\_ReturnTrue |

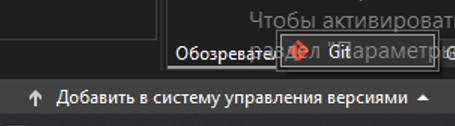
|  |  |
| --- | --- |
| Тестовый пример # | TC\_7 |
| Название теста | CheckPasswordInfo |
| Краткое изложение теста | Тест проводится для того, чтобы понять корректно ли введен пароль |
| Тестовые данные | Правильные данные:   1. string entry = "Password10$";   не правильные данные:   1. string entry = ""; 2. string entry = " "; 3. string entry = "пароль"; 4. string entry = "ПАРОЛЬ"; 5. string entry = "парольПАРОЛЬ"; 6. string entry = "Пароль10"; 7. string entry = "password"; 8. string entry = "Password"; 9. string entry = "Password1"; |
| Ожидаемый результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Фактический результат | 1. True 2. False 3. False 4. False 5. False 6. False 7. False 8. False 9. False 10. False |
| Статус | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass 7. Pass 8. Pass 9. Pass 10. pass |
| Комментарии | Названия тестовых методов   1. CheckPasswordInfo\_EmptyPasswordInfo\_ReturnFalse 2. CheckPasswordInfo\_SpacesPasswordInfo\_ReturnFalse 3. CheckPasswordInfo\_LowerCyrillicPasswordInfo\_ReturnFalse 4. CheckPasswordInfo\_UpperCyrillicPasswordInfo\_ReturnFalse 5. CheckPasswordInfo\_LowerAndUpperCyrillicPasswordInfo\_ReturnFalse 6. CheckPasswordInfo\_LowerAndUpperCyrillicAndDigitPasswordInfo\_ReturnFalse 7. CheckPasswordInfo\_LowerLatinPasswordInfo\_ReturnFalse 8. CheckPasswordInfo\_LowerAndUpperLatinPasswordInfo\_ReturnFalse 9. CheckPasswordInfo\_LowerAndUpperLatinAndDigitPasswordInfo\_ReturnFalse 10. CheckPasswordInfo\_LowerAndUpperLatinAndDigitPasswordInfo\_ReturnTrue |

### Работа с системой контроля версий

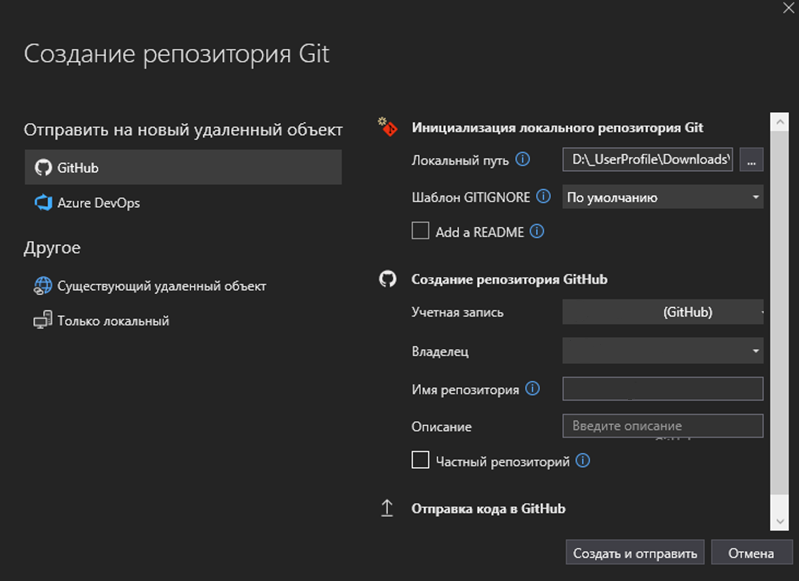
Приложение сохраняется и устанавливается на сайте GitHub.

Для добавления приложения на GH, в Visual Studio необходимо нажать на кнопку “Добавление в систему управления версиями

Добавление в систему управления версиями



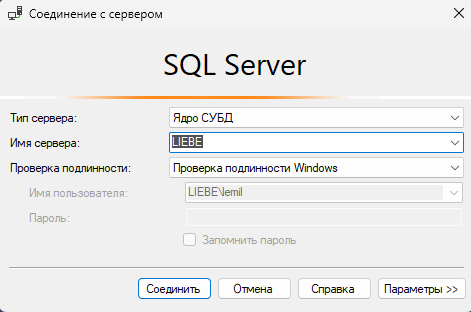
Кнопки добавления



Ссылка на работу в GitHub: <https://github.com/Vara1on1/kyrsach>

### Руководство по установке программы

1. Скачать проект по указанной выше ссылке на GH
2. Запустить Visual Studio
3. Импортировать с Visual studio Базу данных
4. Запускайте MySQL Management Studio и войдите на сервер.

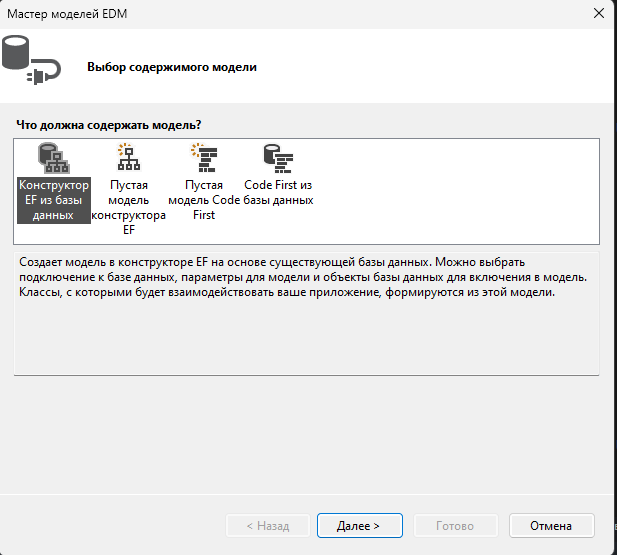


Вход на сервер

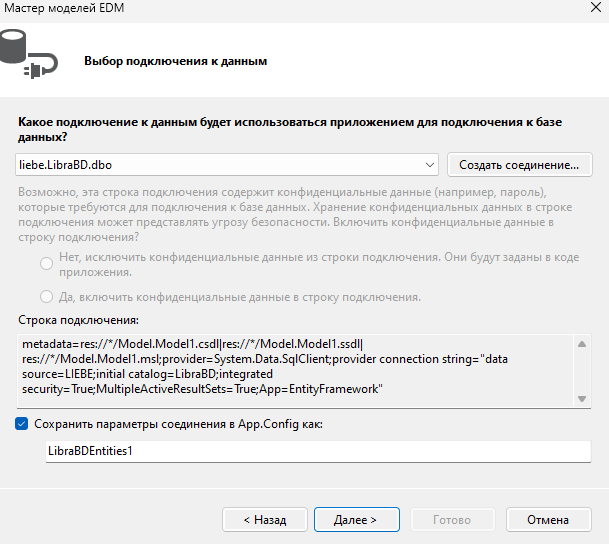
1. Импортировать файл SQL в SQL Management Studio, перенеся в программу
2. Импортировать Базу данных в Visual Studio



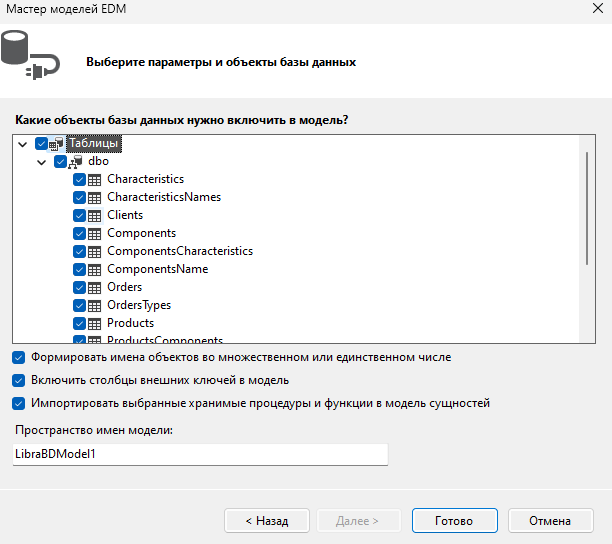
Нажмите на “Модель ADO.Net EDM”



Выберете пункт “Конструктор EF”, Нажмите далее



Нажмите далее

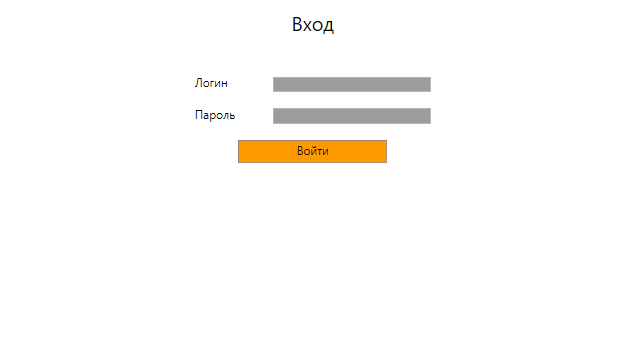


Выберите пункт таблицы(должна появиться галочка), нажмите готово

Прототип готов к запуску

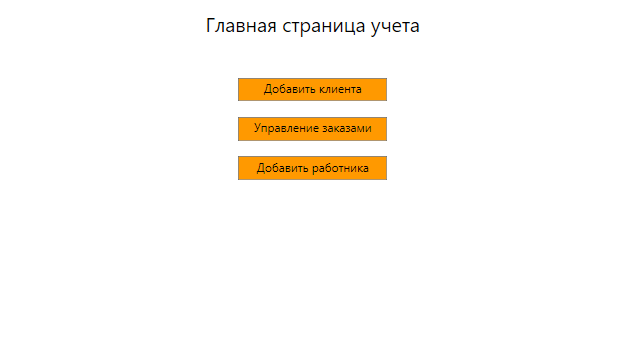
### Документация по работе с информационной системой

1. Первое что предстоит сотруднику, менеджеру, Ком. Директору это пройти страницу авторизации



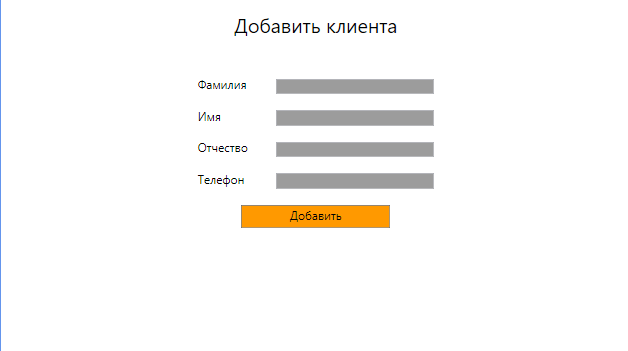
1. после авторизации они попадают на “Главную страницу” приложения

Где есть функции добавления клиента, управления заказами и добавление нового сотрудника.



3.Выбрав кнопку "Добавления клиента” вы попадаете на эту страницу.

Куда нужно ввести ФИО и номер телефона клиента



После выполнения всех условий нажмите на кнопку “сохранить”, вас выведет в главное меню приложения

4.Выбрав кнопку “Добавить сотрудника” вы попадаете на эту страницу.

Куда нужно ввести ФИО, логин, пароль и роль сотрудника



рис

После выполнения всех условий нажмите на кнопку “Добавить”, вас выведет в главное меню приложения

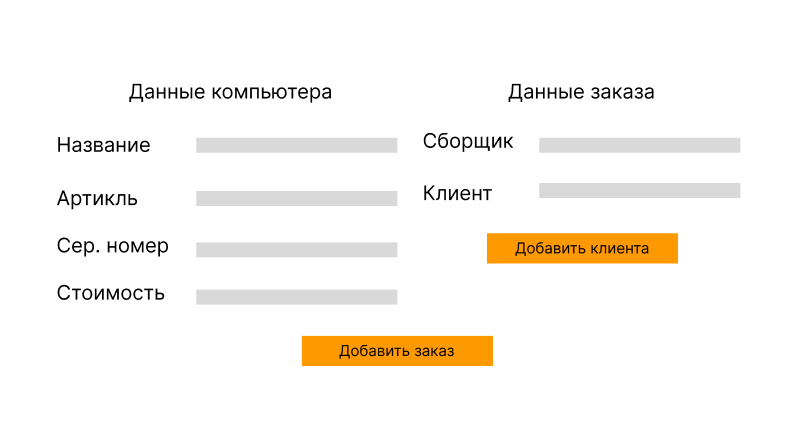
1. Выбрав кнопку “Управление заказами” вы попадаете на страницу “Учета заказов”



На этой странице есть 3 кнопки “Заказ на ремонт”. “Заказ на сборку”. “Просмотр заказов”.

5.Нажав на кнопку “Заказ на ремонт” вы попадаете на эту станицу.

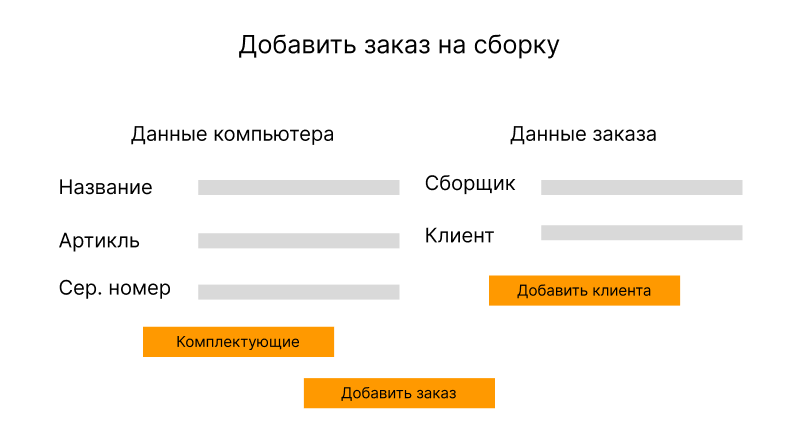
Куда нужно ввести Название ремонта, артикль, сер. Номер, стоимость, выбрать сборщика, и самого клиента. Если клиент не был зарегистрирован ранее нажмите на копку “Добавить клиента” и вас перенесет на страницу Добавления клиента.



После выполнения всех условий нажмите на кнопку “Добавить заказ”, вас выведет в Главное меню

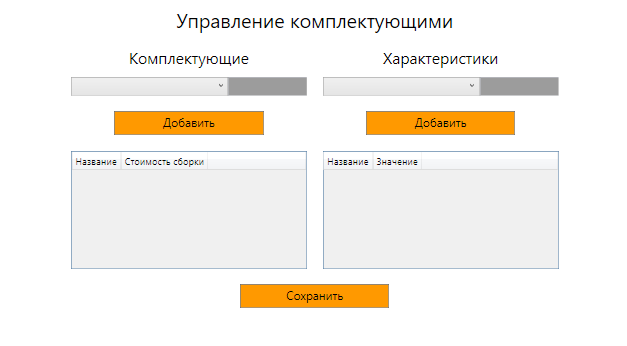
6.Нажав на кнопку “Заказ на сборку” вы попадаете на эту станицу.

Куда нужно ввести Название сборки, артикль, сер. Номер, выбрать сборщика, и самого клиента. Если клиент не был зарегистрирован ранее нажмите на копку “Добавить клиента” и вас перенесет на страницу Добавления клиента.



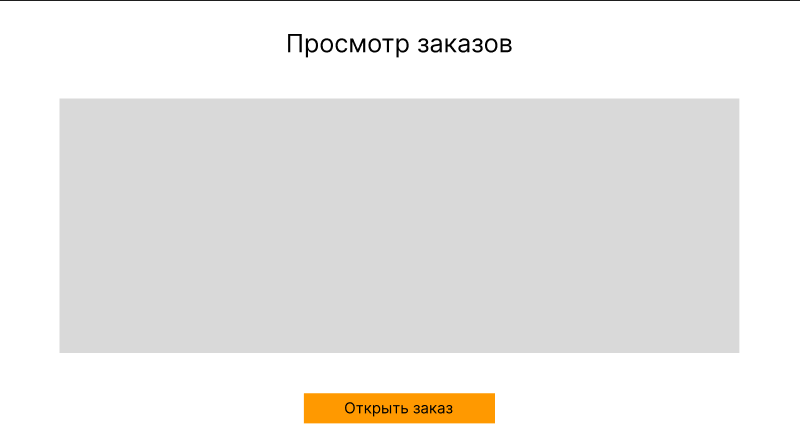
Нажав на кнопку “Комплектующие” вас перенесет на эту страницу.

Где нужно выбрать комплектующие и назначить им цену, нужно выбрать характеристики к комплектующим.



После выполнения всех условий нажмите на кнопку “Сохранить”, вас выведет в меню Добавить заказ на сборку, где нужно нажать на кнопку “Добавить заказ”, вас выведет в главное меню

7.Нажав на кнопку “Просмотреть заказы” вы попадаете на эту страницу



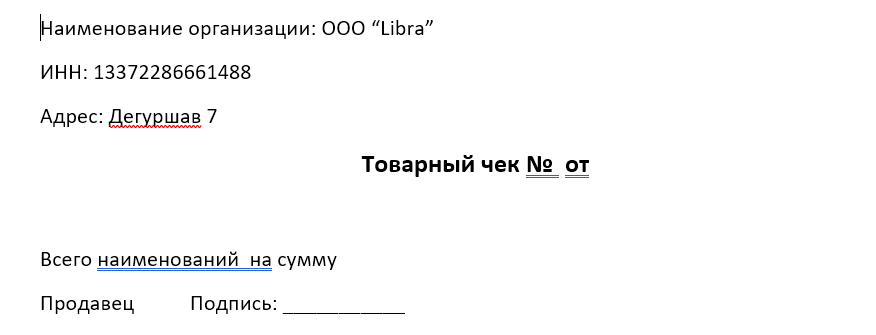
Здесь будут выведены все заказы как на ремонт так и на сборку, выбрав пункт заказа и нажав на кнопку “Открыть заказ” вы попадете на страницу “Управления заказом”



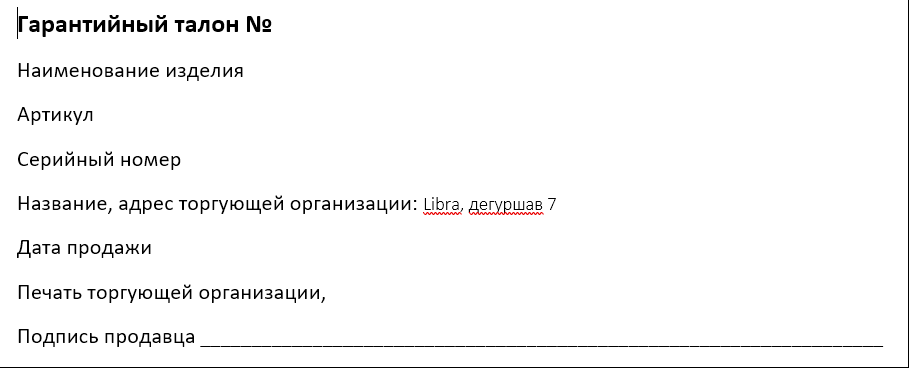
Тут вы можете посмотреть “Комплектующие” к заказу, можете “Удалить заказ”, “Завершить заказ”, посмотреть “Гарантийный талон”, “Накладную заказа”, “Товарный чек”. Информация о дате начала будет датой когда был заказан ремонт или сборка, дата окончания высветится только после завершения заказа, “Удалить заказ” можно всегда.

Так же “Гарантийный талон”, “Накладная заказа”, “Товарный чек” можно просмотреть только после завершения заказа

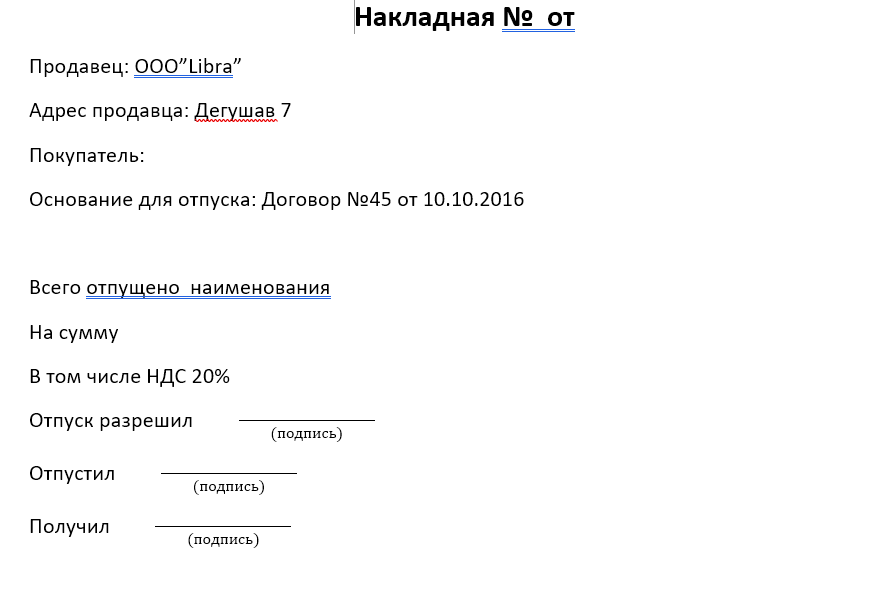
Товарный чек



Гарантийный талон



Накладная заказа



4. ГЛОССАРИЙ

БД – база данных

АРМ – автоматизированное рабочее место

СУБД MS SQL SERVER – система управления базами данных Microsoft SQL Server

VS– Visual studio

GH– GitHub

5. СПИСОК ИСТОЧНИКОВ

1. [WPF и C# | Полное руководство - Metanit](https://metanit.com/sharp/wpf/) [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/wpf/>
2. Браузер API .NET [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/>
3. Н. А. Тюкачев, В. Г. Хлебостроев C#. Основы программирования. Учебное пособие (+ CD) - СПб.: ЭБС ЛАНЬ, 2017. – 272 c.
4. в руководство по классическим приложениям - learn.microsoft.com
5. Github- github.com
6. Wikipedia - ru.wikipedia.org
7. Microsoft вики - microsoft.fandom.com
8. YouTube.com
9. <http://188.234.244.32> - <http://188.234.244.32/pluginfile.php/62441/mod_label/intro/1.pdf>
10. <http://188.234.244.32/pluginfile.php/62441/mod_label/intro/2.pdf>
11. <http://188.234.244.32/pluginfile.php/62441/mod_label/intro/3.pdf>
12. <http://188.234.244.32/pluginfile.php/62441/mod_label/intro/4.pdf>
13. <http://188.234.244.32/pluginfile.php/62441/mod_label/intro/5.pdf>
14. <http://188.234.244.32> – презентации
15. <http://188.234.244.32> – Docx Документы
16. Основа WPF- <https://professorweb.ru/my/WPF/base_WPF/level1/base_WPF_index.php>
17. c# - stackoverflow.com
18. PlantUML в двух словах [Электронный ресурс] - Режим доступа: <https://plantuml.com/ru/>
19. SQL Server connection strings [Электронный ресурс] – Режим доступа: <https://www.connectionstrings.com/sql-server/>
20. Создание запросов LINQ на языке C# [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/csharp/linq/write-linq-queries

ЗАКЛЮЧЕНИЕ

В этой курсовой работе было продемонстрировано приложение, её структура, наглядно было показано применение приложения и что для кого предназначено.

Была продемонстрирована работа с Visual studio.

Была продемонстрирована база данных и как её подключать.

Были продемонстрированы тесты и их ожидаемые результаты

Был продемонстрирован способ подключения приложения к сайте GitHub.

# ПРИЛОЖЕНИЕ А

# ПРИЛОЖЕНИЕ Б