

КОМИТЕТ ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖНОЙ ПОЛИТИКИ
ВОЛГОГРАДСКОЙ ОБЛАСТИ
Государственное бюджетное профессиональное образовательное учреждение
«Волгоградский колледж управления и новых технологий имени Юрия Гагарина»
(ГБПОУ «ВКУиНТ им. Ю. Гагарина»)

Выпускная квалификационная работа

«Допустить к защите»
Начальник отдела УВР
образовательных программ
цифровых технологий
в промышленности

Александрина А.Ю. / _____ /
(ФИО, подпись)
« _____ » _____ 2023 г.

ДИПЛОМНЫЙ ПРОЕКТ

(ДП 09.02.07 ИСП-19 12 ПЗ)

Тема: Разработка автоматизированной информационной системы по учету
работы строительной организации

Студент Бибарцев В.Т. / _____ /
(ФИО, подпись)

Группа ИСП-19

Специальность 09.02.07 Информационные
системы и программирование
(код, наименование)

Руководитель: _____ / _____ /
(ФИО, подпись)

Консультант по экспериментальной части

_____ / _____ /
(ФИО, подпись)

Консультант по нормоконтролю

_____ / _____ /
(ФИО, подпись)

Рецензент _____
_____ / _____ /
(ФИО, должность, подпись)

Зав.кафедры ИТ и радиотехники

_____ / _____ /
(ФИО, подпись)

Волгоград
2023

УТВЕРЖДАЮ
Начальник отдела УВР
образовательных программ
цифровых технологий
в промышленности

(ФИО, подпись)

«13» 04 2023г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

Студенту 4 курса, группы ИСП-19, специальности 09.02.07 Информационные системы и программирование
(код, наименование)

Бибарцеву Вячеславу Тимуровичу
(фамилия, имя, отчество)

Тема дипломного проекта: Разработка автоматизированной информационной системы по учету работы строительной организации

Исходные данные: теоретический и практический материал по специальности 09.02.07 Информационные системы и программирование

Перечень решений, подлежащих разработке: Разработка автоматизированной информационной системы по учету работы строительной организации

Структура и содержание дипломного проекта:

ВВЕДЕНИЕ

Глава 1. Анализ предметной области

- 1.1 Обзор предметной области
- 1.2 Анализ существующих решений
- 1.3 Постановка задачи

Глава 2. Проектирование информационной системы

- 2.1 Построение бизнес-модели организации
- 2.2 Проектирование информационной системы

Глава 3. Разработка программных решений

- 3.1 Выбор средств и методов для разработки программных решений
- 3.2 Построение ERD-диаграммы
- 3.3 Разработка баз данных, объектов баз данных и импорт
- 3.4 Описание процесса создания desktop-приложения
- 3.5 Руководство пользователя

Глава 4. Тестирование программных решений

- 4.1 Функциональное тестирование
- 4.2 Руководство по установке и внедрению программного решения

Заключение

Список использованных информационных источников

Приложения

Список использованных информационных источников

1. Баранчиков, А.И. Организация сетевого администрирования: учебник / А.И. Баранчиков. – СПб., 2020.
2. Максимова, Н.В. Архитектура ЭВМ и вычислительных систем: Учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. – 5-е изд., перераб. и доп. – СПб., 2019. – (Профессиональное образование).
3. Шоломов, Л.А. Основы теории дискретных логических и вычислительных устройств: Учебное пособие / Л.А. Шоломов. – 3-е изд., испр. – СПб., 2020
4. Введение в инфокоммуникационные технологии: Учебное пособие / Л.Г. Гагарина и др. – СПб., 2021.
5. Веретехина, С.В. Информационные технологии. Пакеты программного обеспечения общего блока «ИТ-инструментарий»: Монография / С.В. Веретехина, В.В. Веретехин. – СПб., 2021.
6. Яшин, В.Н. Информатика: программные средства персонального компьютера: Учебное пособие / В.Н. Яшин. – СПб., 2022.
7. Исаченко, О.В. Программное обеспечение компьютерных сетей: Учебное пособие / О.В. Исаченко. – СПб., 2020.
8. Максимов, Н.В. Архитектура ЭВМ и вычислительных систем: Учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. – 5-е изд., перераб. и доп. – СПб., 2019.
9. Семенов, Ю.А. Алгоритмы телекоммуникационных сетей. В 3 ч. Ч.1. Алгоритмы и протоколы каналов и сетей передачи данных: Учебное пособие / Ю.А. Семенов. – СПб., 2019.
10. Семенов, Ю.А. Алгоритмы телекоммуникационных сетей. В 3 ч. Ч.2. Протоколы и алгоритмы маршрутизации в internet: Учебное пособие / Ю.А. Семенов. – СПб., 2022
11. Семенов, Ю.А. Алгоритмы телекоммуникационных сетей. В 3 ч. Ч.3. Процедуры, диагностика, безопасность : Учебное пособие. / Ю.А. Семенов. – СПб., 2019.
12. Верещагина, Е.А. Операционные системы: Учебно-методический комплекс. – М.: Проспект, 2020
13. Назаров, С.В. Операционные системы. Практикум (для бакалавров) / С.В. Назаров. – СПб., 2020.
14. Абрамов, В.Г. Введение в язык Pascal : Учебное пособие / В.Г. Абрамов. – СПб., 2022.
15. Грацианова, Т.Ю. Программирование в примерах и задачах / Т.Ю. Грацианова. – СПб., 2022.
16. Колдаев, В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев. – СПб., 2020
17. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – 4-е изд., перераб. и доп. – М., 2022.
18. Голицына, О.Л. Основы проектирования баз данных: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. – 2-е изд. – М., 2021.
19. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – 3-е изд., дополненное. – СПб., 2019.
20. Одинцов, И. Профессиональное программирование: системный подход / И. Одинцов. – СПб.: «bhn», 2019
21. Федорова, Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем: учебник / Г.Н. Федорова. – СПб., 2020.
22. Черников, Б.В. Управление качеством программного обеспечения: Учебник / Б.В. Черников. – СПб., 2019
23. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – 3-е изд., дополненное. – СПб., 2020.

24. Одинцов, И. Профессиональное программирование: системный подход / И. Одинцов. – СПб.: «bhn», 2020
25. Федорова, Г.Н Разработка программных модулей программного обеспечения для компьютерных систем: учебник / Г.Н. Федорова. – СПб., 2019.
26. Черников, Б.В. Управление качеством программного обеспечения: Учебник / Б.В. Черников. – СПб., 2019
27. Дронов, В.А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов / В.А. Дронов. – СПб., 2021.
28. Пьюривал, С. Основы разработки веб-приложений / С. Пьюривал. – СПб: Питер, 2021. – 272 с.
29. Робсон, Э. Изучаем HTML, XHTML и CSS / Э. Робсон. – 2-е изд. – СПб., 2020.
30. Черников, Б.В. Управление качеством программного обеспечения: Учебник / Б.В. Черников. – Спб., 2020. – (Высшее образование).

План – график выполнения дипломного проекта:

№ п/п	Разделы и их содержание	Срок выполнения
	Введение	24.05.2023
1	Глава 1. Анализ предметной области	28.05.2023
2	Глава 2. Проектирование информационной системы	29.05.2023
3	Глава 3. Разработка программных решений	29.05.2023
4	Глава 4. Тестирование программных решений	1.06.2023
5	Заключение	5.06.2023
6	Список использованных информационных источников	5.06.2023
7	Приложения	5.06.2023

Наименование предприятия преддипломной практики: ГБПОУ «ВКУиНТ им.Ю. Гагарина»

Руководитель дипломного проекта Ребро И.В. / _____ /

подпись

Ф.И.О.

Дата выдачи задания «__13__» __04__ 2023 г.

Срок сдачи студентом дипломного проекта «9» __06__ 2023 г.

Рассмотрено на заседании кафедры ИТ и радиотехники

«14» 12 2022 г.

Протокол № __4__

Зав. кафедрой Гладкова Е.М. / _____ /

подпись

Ф.И.О.

Студент Бибарцев В.Т. / _____ /

подпись

Ф.И.О.

«__13__» __04__ 2023 г.

Содержание

Введение	6
Раздел 1. Анализ предметной области	8
1.1. Организация работы строительной организации	8
1.2. Анализ существующих решений	11
1.3. Постановка задачи.....	13
Раздел 2. Проектирование информационной системы.....	20
2.1. Построение бизнес-модели организации	20
2.2. Проектирование информационной системы.....	20
Раздел 3. Разработка программных решений	22
3.1. Выбор средств и методов для разработки программных решений	22
3.2. Построение ERD диаграммы	23
3.3. Разработка баз данных, объектов баз данных и импорт	24
3.4. Описание процесса создания desktop – приложения.....	26
3.5. Руководство пользователя	35
Раздел 4. Тестирование программных решений.....	39
4.1. Функциональное тестирование.....	39
4.2. Руководство по установке и внедрению программного решения	44
Заключение	47
Список использованных источников	49
Приложение А	50
Приложение Б.....	51
Приложение Г	52
Приложение Д	55

Введение

В современном мире строительная отрасль является важнейшей среди других отраслей, а также является опорой национальной экономики страны. Главная роль заключается в том, что только из-за строительства появляется причина успешного развития основных производственных фондов и экономической ситуации страны в целом. Строительство является важным условием в жизни общества, оно сопровождает жизнь человечества с самого начала его существования, помогает удовлетворить потребности общества – главную потребность в жилье.

Строительная организация¹ - это олицетворенная производственная единица, основой которой является профессионально организованный трудовой коллектив, способный с помощью имеющихся в его распоряжении средств производства вырабатывать нужную потребителю строительную продукцию (дома, сооружения, строительные работы) и строительные услуги (ремонтно-строительные работы) соответствующего назначения, профиля, типа.

Строительная организация направлена на различные строительные услуги. Например, капитальный и текущий ремонт, реконструкция, реставрация, возведение сооружений².

Каждый человек хочет иметь "крышу над головой" - жильё, в котором будет всё удобно, красиво, современно, надёжно и прочно. Без строительных организаций это общечеловеческое желание реализовать никак не получится. То, что делают представители этих организаций, служит людям очень долго. И это не только жильё, а ещё и другие здания различного предназначения: заводы и больницы, школы и детские сады, стадионы и концертные залы, музеи и кинотеатры, ангары, склады и многое другое.

¹[https://studfile.net/preview/5166446/#:~:text=Строительная%20организация%20-%20это%20олицетворенная,работы\)%20соответствующего%20назначения%2C%20профиля%2C%20типа](https://studfile.net/preview/5166446/#:~:text=Строительная%20организация%20-%20это%20олицетворенная,работы)%20соответствующего%20назначения%2C%20профиля%2C%20типа)

²<https://studfile.net/preview/9160205/page:4/>

В современном мире строительные организации расширяются все больше и больше и используют для своей успешной работы автоматизированные информационные системы, которые помогают решать профессиональные проблемы. Основными проблемами строительных организаций является сложность введения учета услуг, планирования услуг на длительный период времени и мониторинг проведения услуг. Поэтому более благоразумно будет использовать АИС, которая поможет сотрудникам с их проблемами.

Актуальность выбранной темы выпускной квалификационной работы обусловлено тем, что она направлена на уменьшение ведения документооборота строительной организации.

Целью дипломного проекта является разработка автоматизированной информационной системы по учету работы строительной организации.

Для достижения поставленных целей поставлены такие задачи:

- Изучить предметную область;
- Осуществить анализ существующих решений;
- Определить основные требования к программному продукту;
- Спроектировать бизнес-модель организации;
- Спроектировать информационную систему;
- Необходимо сделать выбор средств и методов для разработки программных решений;
- Построить ER-диаграмму;
- Разработать базы данных, объекты баз данных и осуществить импорт;
- Создать desktop-приложение;
- Создать руководство пользователя;
- Протестировать разработанное приложение;
- Создать руководство по установке, внедрению программного решения.

Раздел 1. Анализ предметной области

1.1. Организация работы строительной организации

Строительная организация является сложной структурой, которая включает в себя различные подразделения, отделы, бригады, субподрядчиков и клиентов. Управление проектами в строительной организации требует точной и своевременной информации о выполненных работах, расходах материалов, затратах времени и бюджетах.

Современная строительная организация включает в себя множество подразделений, отвечающих за различные аспекты производственного процесса. Одним из важнейших направлений в работе строительной организации является учёт и контроль выполнения работ³.

Учёт работы строительной организации включает в себя учёт объёма работ, использованных материалов и оборудования, затрат рабочего времени и т.д. Для эффективного управления производственным процессом и принятия оперативных решений необходима автоматизированная информационная система учёта работы строительной организации.

Такая система позволяет сократить время на обработку и анализ данных, уменьшить риск ошибок, повысить прозрачность и контрольность процессов в организации. Кроме того, автоматизированный учёт работы строительной организации способствует оптимизации использования ресурсов, повышению эффективности работы персонала и сокращению затрат.

Сотрудник, который ведет учет строительной организации фиксирует на бумаге, в Excel-таблице или в специальной программе все операции, связанные с хранением и движением товаров. К ним относятся:

- приемка заказов от заказчиков;
- ввод данных о работе в базу или учетные документы;

³<https://www.kp.ru/guide/struktura-stroitel-noi-organizatsii.html>

- сдача объекта заказчику;
- перемещение;
- инвентаризация;
- информирование бухгалтерии об изменении закупочных цен;
- списание на собственные нужды, из-за порчи или недостачи;
- корректировка характеристик продукции в учетных карточках.

Благодаря учету в любой момент времени можно уточнить, какой объект где, в каком количестве и на какую сумму был сдан.

Если номенклатуры много, важно, чтобы сотрудники могли быстро найти нужные позиции. Поэтому учет предполагает группировку услуг по датам начала работы и ценам. Так удобнее регулировать ассортимент: выявлять, какие услуги пользуются спросом, а какие приводят к «затаренности», своевременно находить услуги, которые не удастся выполнить вовремя и предпринять нужные меры.

Когда сотрудники могут назвать точную дату начала и окончания работы по определенным категориям, это помогает строить более точные планы на будущие строительные услуги, контролировать показатели сдачи объектов: объем, сроки расчетов.

Существует три основных вида учета строительной организации⁴:

- оперативно-технический является текущим (оперативным) учетом, непосредственно определяющим технико-экономическое состояние строительного производства. Данному виду учета подлежат следующие характеристики, определяющие ход реализации строительства:
 - Учет выполненных работ;
 - Учет труда и заработной платы;
 - Учет материально-технических ресурсов и т.д.

⁴<https://kamala-soft.com/blog/uchet-v-stroitelstve-osobennosti-i-poryadok-avtomatizatsii/>

- Бухгалтерский является сплошным и непрерывным учетом всех показателей, связанных, прежде всего, с движением денежных средств, основных фондов и оборотных средств. Бухгалтерский учет должен обеспечить
 - Получение сводных данных о производственной деятельности строительных организаций;
 - Контроль за движением материальных ценностей;
 - Группировку материальных издержек и т.д.
- Статистический – система регистрации, обобщения и изучения массовых, качественно однородных социально-экономических явлений в масштабе предприятия, отрасли, экономического региона или страны. Распространенный и самый простой вариант ведения учета строительной организации — ручной учет в Excel-таблицах.

Сотрудники в свободной форме вносят информацию о приходе и движении запасов. В Excel обработка данных технически возможна только на самом примитивном уровне — подсчеты сумм.

Плюс этого способа — бесплатно. Пожалуй, он единственный. Минусов гораздо больше:

- риск потери документов или данных, если сотрудник случайно удалил из таблицы что-то важное;
- отсутствие автоматизации приводит к рутине, вследствие чего инвентаризация может затянуться на много часов;
- так как все процессы учтываются вручную, легко забыть про услугу, которую не удастся вовремя выполнить, допустить ошибки (и обнаружить их слишком поздно) в документах.

Альтернатива ручного способа — автоматизация с помощью специальной программы. Она решает все проблемы, перечисленные выше, так как:

1. Все документы в электронном виде. Благодаря распределению прав доступа можно ограничить просмотр и редактирование.

2. Документы заполняются с помощью встроенных шаблонов и справочников, что сокращает риск ошибок из-за человеческого фактора.
3. Разгрузка от бумажной волокиты экономит время на складские операции, в том числе инвентаризацию.

1.2. Анализ существующих решений

Существуют различные информационные системы для управления строительными проектами, включая универсальные и специализированные системы. Перечисленные программы имеют свои особенности, индивидуальные характеристики, предназначение. Рассмотрим более детально каждую программу для автоматизации строительства.

БИТ. СТРОИТЕЛЬСТВО 365⁵ – облачный сервис для управления строительством на базе 1С. Решение состоит из модулей, каждый из которых закрывает весь цикл задач одного из подразделений. Данные хранятся на российских серверах, что обеспечивает надежность и независимость от работы зарубежных сервисов.

первый.бит ПРОДУКТЫ 1С ОТРАСЛЕВЫЕ РЕШЕНИЯ УСЛУГИ КРУПНОМУ БИЗНЕСУ БЛОГ О НАС КОНТАКТЫ +7 (495) 677-01-77
Центр автоматизации строительных компаний

1С / Программы 1С / БИТ.СТРОИТЕЛЬСТВО 365

БИТ.СТРОИТЕЛЬСТВО 365

Российский облачный сервис для автоматизации учета и отчетности строительной компании

- ✓ Оперативная и достоверная отчетность по объектам
- ✓ Автоматическое заполнение форм М-19, М-29, КС-2, КС-3
- ✓ Сокращение времени на ввод первичной документации
- ✓ Удобное и быстрое подключение без первоначальных затрат

Рисунок 1 - АИС БИТ.СТРОИТЕЛЬСТВО 365

Возможности программы:

⁵<https://www.1cbit.ru/1csoft/bit-stroitelstvo-365/>

- Бухгалтерский и налоговый учет строительной организации;
- Управление материально-производственными запасами;
- Расчет зарплаты по объектам строительства;
- Учет работы спецтехники и т.д.

Procure⁶- управляет строительными процессами, ресурсами и финансами от планирования проекта до его завершения. Эта платформа связывает каждого участника проекта с решениями, разработанными специально для отрасли - для владельца, для генерального подрядчика и для специализированного подрядчика.



Рисунок 2 –Окно программы Procure

Возможности программы:

- Учет в строительстве;
- Составление смет;
- CRM для строительной отрасли;
- Незавершенные работы (Punch list);
- BIM-программы для проектирования зданий;
- Календарное планирование строительства;

⁶<https://picktech.ru/product/procure>

- Управление строительными заявками.

Buildertrend⁷- это облачное программное обеспечение для управления проектами.

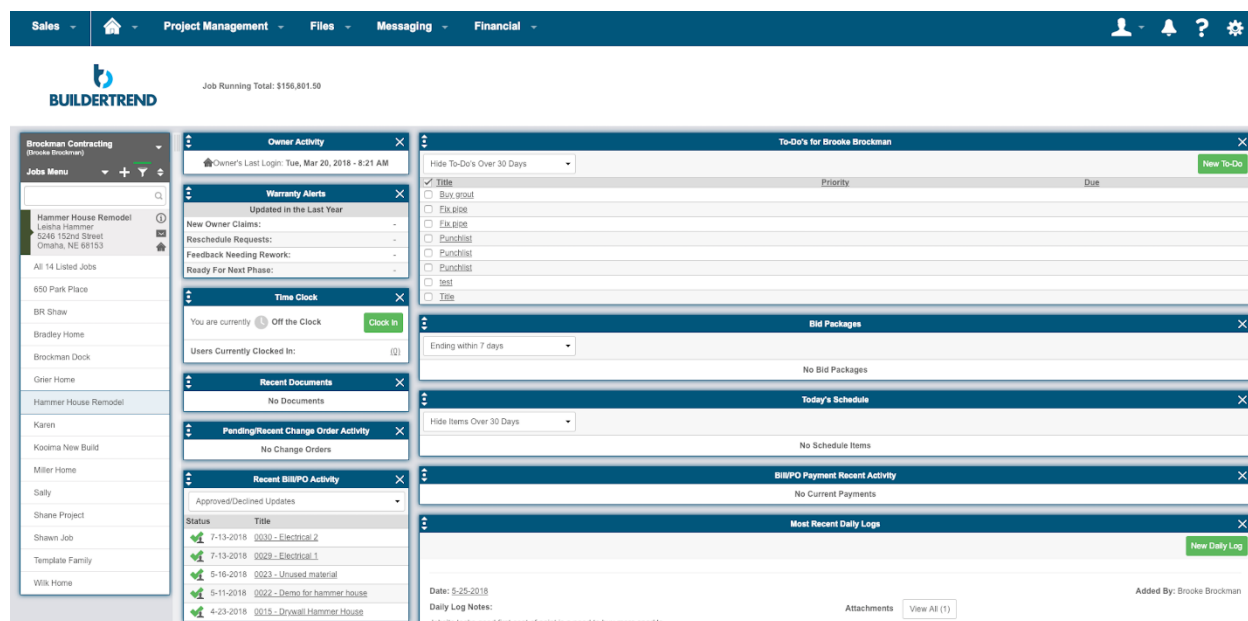


Рисунок 3 - Buildertrend

Возможности программы:

- Управление проектами;
- CRM для строительной отрасли;
- Ландшафтные работы;
- Календарное планирование строительства;

После рассмотрения программ аналогов можно понять какие функции необходимо реализовать:

- Учет бюджета строительной организации;
- Учет времени начала и окончания работ;
- Сколько сотрудников будет затронуто на выполнение строительной услуги.

1.3. Постановка задачи

Название АИС “СтройДом–строительная организация”.

⁷<https://picktech.ru/product/buildertrend>

В рамках выполнения выпускной квалификационной работы необходимо разработать основные модули информационной системы по учету работы строительной организации. Основная цель разработанной АИС заключается в повышении оперативности и качества принимаемых управленческих решений сотрудниками Заказчика. Основным назначением АИС является автоматизация ведения учета строительной организации.

Данная АИС должна выполнять соответствующие требования:

Описание бизнес-процесса

Необходимо построить схему бизнес-процесса, с помощью нотации BPMN. Данная схема необходима для подробного описания логики выполнения бизнес-процесса, в том числе для отражения деталей процессов, таких как: события, исполнители каждого из действий, используемые и создаваемые документы и другие объекты, использующиеся в качестве входных данных для тех или иных действий или создающиеся в результате их выполнения.

Проектирование информационной системы

При обобщении поставленных задач необходимо преобразовать требования в диаграмму использования (UseCase), с помощью которой моделируется приложение. И построить функциональную структуру разрабатываемой программной системы

Средства и методы разработки АИС

Осуществить анализ средств и методов для разработки автоматизированных информационных систем, а именно СУБД, язык программирования, аппаратно-программный комплекс.

Проектирование, разработка баз данных и импорт

На основе описания предметной области, имеющихся требований, необходимо построить ERD- диаграмму. Для реализации подсистемы хранения данных должна использоваться СУБД <указывается название и версия СУБД>

Также необходимо заполнить базу данных тестовыми данными (не менее 3 записей).

Файловая структура

Файловая структура проекта должна отражать логику, заложенную в приложение. Все формы содержатся в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.

Структура проекта

Каждая сущность должна быть представлена в программе как минимум одним отдельным классом. Классы должны быть небольшими, понятными и выполнять одну единственную функцию (Single responsibility principle).

Для работы с разными сущностями необходимо использовать разные формы, где это уместно.

Логическая структура

Логика представления (работа с пользовательским вводом/выводом, формы, обработка событий) не должна быть перемешана с бизнес-логикой, а также не должна быть перемешана с логикой доступа к базе данных (SQL-запросы, запись, получение данных). В идеале это должны быть три независимых модуля.

Функциональные требования

Менеджер может:

- Сортировать, фильтровать, просматривать информацию;

Администратор может:

- Редактировать информацию о количестве строительных услуг, цене на них;
- Удалять и добавлять любую информацию о строительных услугах;

При запуске приложения окно входа – первое, что видит пользователь.

На ней пользователю предлагается ввести свой логин и пароль и пройти авторизацию.

После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа. При переходе в любую учетную запись в интерфейсе (правый верхний угол) должны отображаться ФИО пользователя.

После первой попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации, а затем помимо ввода логина и пароля просит ввести captcha, состоящую из 4 символов.

САРТСНА - должна содержать минимум 4 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты либо наложены друг на друга.

После попытки неудачной авторизации выводится captchas блокировкой кнопки “Авторизация”, при таком раскладе пользователь сможет авторизоваться только тогда, когда он правильно введет captcha.

Список товаров

Очень важно постоянно получать актуальную информацию об незаконченных строительных услугах, поэтому необходимо реализовать вывод услуг, которые хранятся в базе данных, согласно предоставленному макету. При отсутствии изображения необходимо вывести картинку-заглушку из папки Images (no-image.png) .

Вывод услуг должен осуществляться списком.

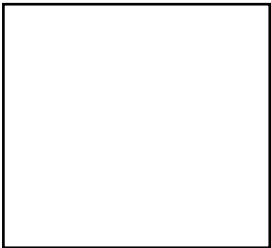
	Наименование услуги
	<i>Тип услуги</i>
	Цена:
	Бригада:

Рисунок 4 - Макет интерфейса

Должна выводиться вся информация о предоставляемой услуги из базы данных. Пользователь должен иметь возможность искать услуги,

используя поисковую строку. Поиск должен осуществляться по всем доступным данным, в том числе по нескольким атрибутам одновременно. Пользователь должен иметь возможность отсортировать услуги (по возрастанию и убыванию) по стоимости и т.д.

Руководство по стилю

Визуальные компоненты должны соответствовать руководству по стилю, необходимо соблюдать:

- Цветовую схему (основной фон: HEX (#FFFF), дополнительныйHEX (#66c2ff).
- Размещение логотипа, с заголовком,
- Использование шрифтов (шрифт Comic Sans MS).
- Установку иконки приложения.
- Необходимо использовать соответствующие названия для приложений и файлов.

Макет и технические характеристики

Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

- разметка и дизайн (предпочтение отдается масштабируемой компоновке;
- последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);

Обратная связь с пользователем

Необходимо уведомлять пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, запрашивать подтверждение перед удалением, предупреждать о неотвратимых операциях, информировать об отсутствии результатов поиска.

Обработка ошибок

В случае несоответствия типа данных или размера поля введенному

значению, пользователю приходит оповещение о совершенной им ошибке.

Оформление кода

Идентификаторы переменных, методов и классов должны отражать суть и цель их использования, в том числе и наименования элементов управления не допустимо чтобы у значений по умолчанию стоял тип Form1, button3).

Хранение данных

Необходимо построение концептуальной модели предметной области, заключающееся в выявлении сущностей и связей между ними, а также отображение этой информации в виде ER-диаграммы.

Требования к ПО и техническому обеспечению

ИС по организации работы сотрудников строительной организации требует для своей работы установки следующего ПО:

1. На сервере ИС должны быть установлены:

- Операционная система: Microsoft Windows XP, Vista, 7, 8, 10, 11
- СУБД MICROSOFT SQL SERVER MANAGEMENT STUDIO 18

2. На рабочей станции пользователя необходимо установить:

- Операционная система: Microsoft Windows XP, Vista, 7, 8, 10, 11
- ИС “для организации работы сотрудников строительной организации”.

Для функционирования ИС необходимо:

- локальная вычислительная сеть на основе протокола TCP/IP с пропускной способностью 10/100 Мбит/с.

Сервер должен удовлетворять следующим минимальным требованиям:

- процессор Celeron-500MHz или аналогичный,
- 1 Gb и более оперативной памяти;
- 80 Gb – жесткий диск
- Монитор – SVGA;
- Клавиатура - 101/102 клавиши;

- Манипулятор типа «мышь».

Требования, предъявляемые к конфигурации клиентских станций:

- процессор, с тактовой частотой не менее 400 MHz,
- 256 Mb оперативной памяти;
- Монитор – SVGA;
- Клавиатура - 101/102 клавиши;
- Манипулятор типа «мышь».

Требования к методическому обеспечению

Необходимо создать «Руководство пользователя и руководство по установке АИС по организации работы сотрудников строительной компании.

Тестирование АИС

Тестирование должно быть проведено по следующим направлениям:

- тестирование интерфейса и функциональности АИС при помощи ввода данных;
- нагрузочное тестирование АИС.

Раздел 2. Проектирование информационной системы

2.1. Построение бизнес-модели организации

Для подробного описания логики выполнения бизнес-процесса, в том числе для отражения деталей процессов, таких как: события, исполнители с действиями каждого, используемые и создаваемые документы и другие объекты, использующиеся в качестве входных данных для тех или иных действий или создающиеся в результате их выполнения используют нотация бизнес-моделирования BPMN. В приложении А изображена данная диаграмма.

В работе выступают два пользователя – администратор и менеджер. Менеджер после удачной авторизации может просматривать, сортировать и фильтровать услуги. А администратор в свою очередь может просматривать, редактировать, добавлять и удалять услуги строительной организации. Если пользователь вводит неверные данные, то авторизация не происходит и на экране пользователя появляется ошибка. Оба пользователя авторизуются и просматривают данные через одну базы данных. Все данные хранятся в базе данных.

2.2 Проектирование информационной системы

При обобщении поставленных задач можно преобразовать требования в диаграмму использования, с помощью которой моделируется автоматизированная система. На рисунке 5 изображена диаграмма прецедентов.

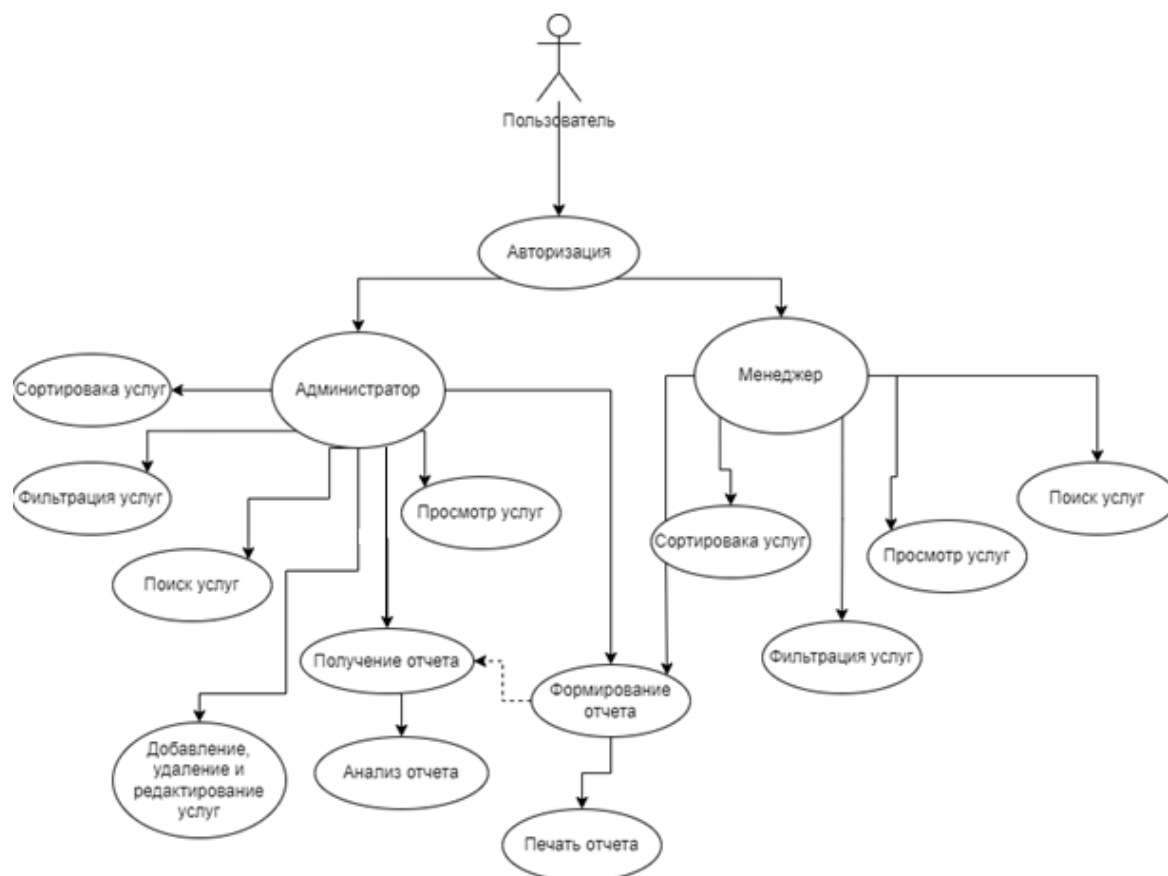


Рисунок 5 - Диаграмма прецедентов

Подсистема аутентификации пользователя, позволяет пройти авторизацию пользователя с помощью ввода логина и пароля, эту процедуру могут пройти сотрудники, персональные данные которых хранятся в базе данных.

Поиск заказа может осуществлять менеджер по работе с клиентами, с помощью поисковой строки. Фильтрация услуг также осуществляется менеджером с помощью выпадающего списка, в котором можно отфильтровать услуги по типам.

Добавление услуг осуществляет администратор с помощью нового окна, в котором будет нужный функционал для добавления услуги. Редактирование и удаление услуг делает также администратор. После авторизации у администратора рядом с услугой появятся две кнопки, которые отвечают за удаление и редактирование. Просматривать услуги могут все авторизованные пользователи.

Раздел 3. Разработка программных решений

3.1. Выбор средств и методов для разработки программных решений

В настоящее время существует огромное количество программных продуктов средств и методов, позволяющих в сжатые сроки эффективно и качественно разработать автоматизированную информационную систему для различных предметных областей.

В качестве целевой СУБД была выбрана MSSQL Server. Достоинства выбранной СУБД заключается в масштабируемости и производительности. База данных менее уязвима. Инструменты бизнес-аналитики с поддержкой самообслуживания предоставляет такие опции, как высокая доступность, усиленная безопасность, улучшенное сжатие данных, сервисы интеграции. Высокая доступность обеспечивается в первую очередь за счёт отказоустойчивой кластеризации.

Таким образом, выбранная целевая СУБД удовлетворяет наши требования и позволит изготовить качественный программный продукт.

Клиентское приложение было разработано в среде Microsoft Visual Studio 2022. Эта среда использует технологию программирования .NET, которая вместе со связанной с ней средой .NET Framework, является одной из самых важных технологий для разработчиков ПО за много лет. Технология .NET спроектирована как новая среда, в рамках которой можно разработать практически любое приложение для Windows. Данная версия среды Visual Studio использует .NET Framework 4.8. Далее мы вкратце перечислим преимущества технологии .NET перед другими технологиями разработки: безопасные типы и общее повышение безопасности приложений, единая модель обработки ошибок, межъязыковое взаимодействие, единая среда разработки, позволяющая проводить межъязыковую отладку, расширенные возможности повторного использования кода.

В качестве языка программирования, был выбран язык программирования C#, так как, он является простым, надежным и масштабируемым языком программирования. Также он является динамически типизированным, облегчает разработчикам поиск ошибок в коде. C# устраняет проблему утечки памяти.

Для создания пользовательского интерфейса была выбрана платформа Windows Presentation Foundation, с помощью которой можно создать более гибкий интерфейс, в WPF можно создавать стили и шаблоны, что облегчит разработку интерфейса.

В заключение можно сказать, что выбор средств разработки является важной задачей при создании автоматизированной информационной системы, а выбранные нами средства позволили создать современное АИС при минимуме усилий, что подтверждает правильность выбора.

3.2. Построение ERD диаграммы

Первым шагом при создании логической модели БД является построение диаграммы ERD, которая состоит из трех частей: сущностей, атрибутов и взаимосвязей.

ERD-диаграмма позволяет рассмотреть систему целиком и выяснить требования, необходимые для ее разработки, касающиеся хранения информации. ERD-диаграмма графически представляет структуру данных проектируемой информационной системы. Сущности отображаются при помощи прямоугольников, содержащих имя.

Взаимосвязи созданы при помощи линий, соединяющих отдельные сущности. Взаимосвязь показывает, что данные одной сущности ссылаются или связаны с данными другой.

При создании концептуальной модели были определены сущности и атрибуты сущностей и составлена логическая модель базы данных, представленной в приложении Б.

Сущностями в ERD-диаграммах выступают таблицы. Атрибуты – это поля в таблицах. Все сущности связаны между собой связями.

Атрибутами в сущности “User” являются: ID, FirstName, LastName, MiddleName, Login, Password, RoleID.

Атрибутами в сущности “Role” являются: ID, RoleName.

Таблица “User” связывается с таблицей “Role” связью “Один ко многим”.

Атрибутами в сущности “Services” являются: ID, ServiceName, ServiceTime, Serviceprice, ImageService, ServiceTypeId, WorkerId.

Атрибутами в сущности “TypeServices” являются: ID, TypeService.

Атрибутами в сущности “Workers” являются: ID, FullName, KindOfWork, BrigadeId.

Таблица “Services” связывается с таблицами “TypeServices” и “Workers” связью “Один ко многим”.

Атрибутами в сущности “Brigade” являются: ID, BrigadeName.

Таблица “Workers” связывается с таблицей “Brigade” связью “Один ко многим”.

3.3. Разработка баз данных, объектов баз данных и импорт

С помощью графического интерфейса в среде SQL Server Management Studio была создана база данных BuildDb, данная база данных содержит шесть таблиц.

1. Таблица “User” соответствует сущности пользовательской учетной записи.
 - ID (автоинкрементируемый первичный ключ): int, not null, primary key;
 - FirstName (имя пользователя): varchar(50), not null;
 - LastName (фамилия пользователя): varchar(50), not null;
 - MiddleName (отчество пользователя): varchar(50);

- Login (логин пользователя): varchar(50), not null;
- Password (пароль пользователя): varchar(50), not null;
- RoleID (внешний ключ на таблицу “Role”): int, not null.

Атрибуты таблицы:

ID, FirstName, LastName, MiddleName, Login, Password, RoleID. Роль пользователя определяется по внешнему ключу RoleId, который является ссылкой на таблицу “Role”.

2.Таблица “Role” соответствует сущности роли пользователя в системе.

- ID (автоинкрементируемый первичный ключ): int, not null, primary key;
- RoleName (наименование роли): varchar(50), not null.

Атрибуты таблицы:ID, RoleName.

3.Таблица “Services” соответствует сущности предоставляемых услуг.

- ID (автоинкрементируемый первичный ключ): int, not null, primary key;
- ServiceName (наименование услуги): varchar(50), not null;
- ServiceTime (примерное время выполнения услуги в часах): int, not null;
- ServicePrice (цена услуги): int, not null;
- ImageService (титульная картинка услуги): image;
- ServiceTypeId (внешний ключ на таблицу “ServiceType”): int, not null;
- WorkerId (внешний ключ на таблицу “Workers”): int, not null.

Атрибуты таблицы: ID, ServiceName, ServiceTime, ServicePrice, ImageService, ServiceTypeId, WorkerId.

4.Таблица “TypeServices” соответствует сущности типов услуг.

- ID (автоинкрементируемый первичный ключ): int, not null, primary key;

- TypeService (наименование типа услуг): varchar(50), not null.

Атрибуты таблицы: ID, TypeService.

5. Таблица “Workers” соответствует рабочим, которые будут выполнять услугу.

- ID (автоинкрементируемый первичный ключ): int, not null, primary key;

- FullName (ФИО рабочих): varchar(100), not null;
- KindOfWork (вид выполняемой работы): varchar(40), not null;
- BrigadeID (внешний ключ на таблицу “Brigade”): int, not null.

Атрибуты таблицы: ID, FullName, KindOfWork, BrigadeID.

6. Таблица “Brigade” соответствует сущности бригад, в которые вступают рабочие.

- ID (автоинкрементируемый первичный ключ): int, not null, primary key;

- BrigadeName (наименование бригады): varchar(50), not null.

Атрибуты таблицы: ID, BrigadeName.

Заказчик предоставил нам данные для заполнения таблиц базы данных, которые были структурированы в таблицах Excel. Предоставленные данные были успешно импортированы в ранее разработанные таблицы.

3.4. Описание процесса создания desktop – приложения

База данных подключается к программе с помощью добавления файла “Модель ADO.NETEDM”. База данных хранится в папке Model.

```

Ссылка: 9
public partial class BuildDbEntities : DbContext
{
    Ссылка: 4
    public BuildDbEntities()
        : base("name=BuildDbEntities")
    {
    }

    Ссылка: 0
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        throw new UnintentionalCodeFirstException();
    }

    Ссылка: 0
    public virtual DbSet<Brigade> Brigade { get; set; }
    Ссылка: 0
    public virtual DbSet<Role> Role { get; set; }
    Ссылка: 9
    public virtual DbSet<Service> Service { get; set; }
    Ссылка: 3
    public virtual DbSet<TypeService> TypeService { get; set; }
    Ссылка: 1
    public virtual DbSet<User> User { get; set; }
    Ссылка: 1
    public virtual DbSet<Worker> Worker { get; set; }
}

```

Рисунок 6 - код БД

В папке Images хранятся картинки, иконки и т.д.

В файле App.xaml задаются стили, которые будут использоваться в UI-интерфейсе.

```

<Application.Resources>
    <BitmapImage x:Key="NullImage" UriSource="/Images/NullImage.jpg"/>

    <SolidColorBrush x:Key="BackColor" Color="#FFFF"/>
    <SolidColorBrush x:Key="BorderBackColor" Color="#66c2ff"/>

    <Style x:Key="MainText" TargetType="TextBlock">
        <Setter Property="Foreground" Value="#003d99"/>
        <Setter Property="FontSize" Value="26"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="FontFamily" Value="Comic Sans MS"/>
    </Style>

    <Style x:Key="Text" TargetType="TextBlock">
        <Setter Property="Foreground" Value="#0066ff"/>
        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontFamily" Value="Comic Sans MS"/>
    </Style>

    <Style x:Key="BtnExitStyle" TargetType="Button">
        <Setter Property="Foreground" Value="#0066ff"/>
        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontFamily" Value="Comic Sans MS"/>
        <Setter Property="Background" Value="FFFFFF"/>
        <Setter Property="BorderBrush" Value="#002db3"/>
    </Style>
</Application.Resources>

```

Рисунок 7 - код в файле App.xaml

В файле MainWindow.xaml прописан основной интерфейс программы и добавлен Frame, в котором будут выводиться страницы.

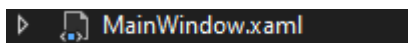


Рисунок 8 –Файл с листингом кода разработанного интерфейса

В файле MainWindow.xaml.cs прописаны основные настройки окна программы (запуск по центру экрана, минимальные размеры и т.д.).

```
WindowStartupLocation = WindowStartupLocation.CenterScreen;  
  
MinHeight = 500;  
MinWidth = 815;  
  
MainFrame.Navigate(new Authorization());
```

Рисунок 9 - настройки MainWindow

Также в файле MainWindow.xaml.cs прописан код кнопки для выхода из системы.

```
private void LogoutButton_Click(object sender, RoutedEventArgs e)  
{  
    MainFrame.Navigate(new Authorization());  
    AuthedPanel.Visibility = Visibility.Hidden;  
    MessageBox.Show("До свидания !");  
}
```

Рисунок 10 - код кнопки 'Выход'

Авторизация осуществляется в файлах Authorization.Authorization.xaml отвечает за интерфейс авторизации.

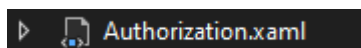


Рисунок 11 –Файл с листингов кода интерфейса авторизации

В файле Authorization.xaml.cs прописана вся логика авторизации и captcha. За генерацию captcha в файле Authorization.xaml.cs отвечает отдельный метод под названием GenerateCaptcha().

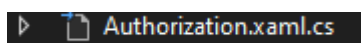


Рисунок 12 –Файл с листингом кода авторизации

Authorization.xaml.cs

```
var currentUser = _context.User.ToList();
```

```

        var auth = currentUser.FirstOrDefault(x => x.Login == InputLogin.Text
        && x.Password == InputPassword.Password);

        if(auth != null)
        {
            NavigationService.Navigate(new ListServices(auth.RoleID));

            ((MainWindow)Window.GetWindow(this)).AuthedPanel.Visibility =
            Visibility.Visible;

            ((MainWindow)Window.GetWindow(this)).AuthedUserTxt.Text =
            $"Здравствуйте {auth.FirstName} {auth.LastName}";
        }

        else
        {
            MessageBox.Show("Вы ввели некорректные данные!",
            "Внимание");
            CapchaPanel.Visibility = Visibility.Visible;

            GenerateCapcha();

            if (InputCapchaTxt.Text != CapchaTxt.Text)
            {
                AuthAsUserbtn.IsEnabled = false;

                DispatcherTimer timer = new DispatcherTimer()
                {
                    Interval = new TimeSpan(0, 0, 3)
                };

                timer.Tick += (bt, ea) =>
                {
                    AuthAsUserbtn.IsEnabled = true;
                    timer.Stop();
                };

                timer.Start();
            }
        }
    }

```

За генерацию captcha в файле Authorization.xaml.cs отвечает отдельный метод под названием GenerateCapcha(). Фрагмент листинга кода изображен на рисунке 13.

```
Ссылка 1
private void GenerateCapcha()
{
    Random rand = new Random();

    CapchaTxt.Text = string.Empty;

    for (int i = 0; i < 2; i++)
    {
        CapchaTxt.Text += (char)('a' + rand.Next(0, 26));
        CapchaTxt.Text += (char)('A' + rand.Next(0, 26));
        CapchaTxt.Text += (char)('0' + rand.Next(0, 9));
    }
}
```

Рисунок 13 - генерация captcha

Кнопка AuthAsUserbtn отвечает за проверку введенного пароля и логина, если данные совпадают с данными из БД, то осуществляется переход на следующую страницу. Если данные не совпадут, то на экране пользователя появится captchas таймером, блокирующем кнопку авторизации. Если пользователь авторизуется в качестве администратора, то у него будет расширенный функционал программы. После чего пользователя перенаправляет на страницу ListServices.

Кнопка AuthAsGuestBtn предназначена для того, чтобы пользователь вошел в программу в качестве гостя, в таком случае он сможет только просматривать услуги. После нажатия по кнопке пользователь переходит на страницу ListServices с минимальными возможностями, конкретнее просмотр услуг.

```
Ссылка 1
private void AuthAsGuestBtn_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new ListServices(1));
    ((MainWindow)Window.GetWindow(this)).AuthedPanel.Visibility = Visibility.Visible;
    ((MainWindow)Window.GetWindow(this)).AuthedUserTxt.Text = "Вы вошли как гость";
}
```

Рисунок 14 - Код кнопки 'AuthAsGuestBtn'

Файл ListServices предназначен для вывода и обработки информации об услугах. В файле ListServices.xaml прописан основной интерфейс блока программы с выводом услуг. Сюда входит основное поле ListView, благодаря которому выводятся все услуги из БД. Также в файле ListServices располагаются кнопки для администратора: удаление, добавление и редактирование. А для остальных пользователей располагается меню с поиском, фильтрацией и сортировкой услуг.

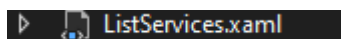


Рисунок 15 –Файл с листингом кода интерфейса для вывода услуг

В файле ListServices.xaml.cs прописана логика взаимодействия с услугами. В конструктор данного файла передается роль авторизованного пользователя. Если авторизованный пользователь является администратором, то у него будут доступны такие функции, как редактирование, удаление и добавление услуг. Также в этом файле прописана логика поиска, фильтрации и сортировки услуг. Также в конструкторе прописано заполнение ComboBox для сортировки услуг по типам.

```
public ListServices(int? RoleID)
{
    InitializeComponent();

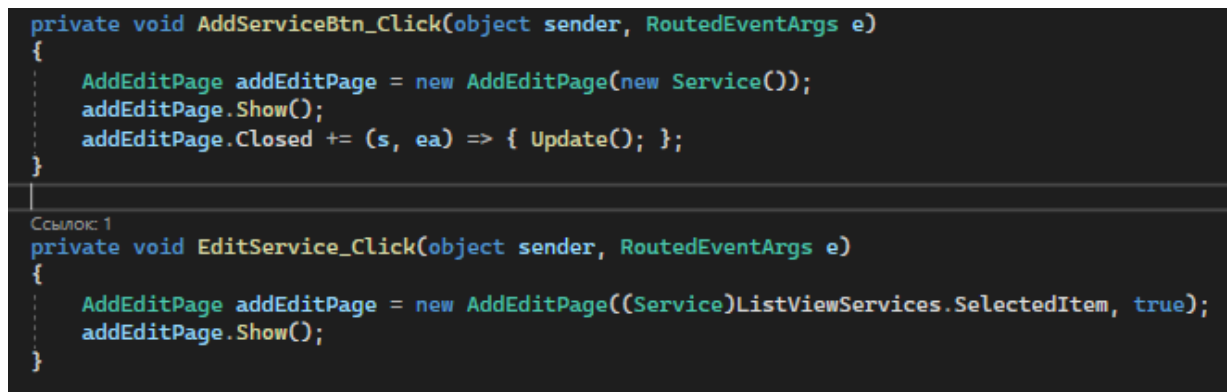
    if(RoleID == 1)
    {
        AdminPanel.Visibility = Visibility.Visible;
        PrintReport.Visibility = Visibility.Visible;
    }

    if (RoleID == 2)
    {
        PrintReport.Visibility = Visibility.Visible;
    }

    var allTypes = _context.TypeService.ToList();
    allTypes.Insert(0, new TypeService
    {
        ServiceType = "Все типы"
    });
    ComboTypesServices.ItemsSource = allTypes;
    ComboTypesServices.SelectedIndex = 0;
}
```

Рисунок 16 - код конструктора класса

Когда авторизованный пользователь с ролью администратор нажмет на кнопку “Добавить услугу” или “Редактировать услугу”, то у него откроется окно AddService, в которой будет осуществляться редактирование или добавление услуг, в зависимости от выбранного действия.



```
private void AddServiceBtn_Click(object sender, RoutedEventArgs e)
{
    AddEditPage addEditPage = new AddEditPage(new Service());
    addEditPage.Show();
    addEditPage.Closed += (s, ea) => { Update(); };
}

Ссылка 1
private void EditService_Click(object sender, RoutedEventArgs e)
{
    AddEditPage addEditPage = new AddEditPage((Service)ListViewServices.SelectedItem, true);
    addEditPage.Show();
}
```

Рисунок 19 - код на перенаправление на страницу AddService

Удаление услуги осуществляется путем выбора услуги, которую необходимо удалить, далее нажатием по кнопке “Удалить услугу”.

Код для удаления услуги:

```
MessageBoxResult deleteService = MessageBox.Show($"Удалить  
{((Service)ListViewServices.SelectedItem).ServiceName} ?", "Удаление",  
MessageBoxButton.YesNo);

if (deleteService == MessageBoxResult.Yes)
{
    try
    {
        _context.Service.Remove((Service)ListViewServices.SelectedItem);
        _context.SaveChanges();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Неудается удалить услугу  
{ex.Message.ToString()}");
    }
}
```



```

    }
    return;

```

Также на странице ListServices есть кнопка, которая по нажатию открывает страницу для формирования отчета. Формировать отчет может администратор и менеджер.

```

private void GoPrintReportBtn_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new PrintReport());
}

```

Рисунок 20 - код на открытие страницы PrintReport

Интерфейс для формирования отчета разделен на две части. В первой части располагается кнопка выйти и ComboBox для сортировки услуг по типам.

Во второй части располагается DataGrid в котором выводятся услуги табличным способом

► PrintReport.xaml

Рисунок 21—Файл с листингом кода интерфейса печати отчета

В коде C# прописана логика работы формирования отчета. В конструкторе класса прописано заполнение ComboBox для сортировки услуг по типам. В методе ComboTypes_SelectionChanged прописана логика фильтрации, а метод PrintReportBtn_Click отвечает за печать отчета, либо за конвертирование таблицы в PDFформат.

► PrintReport.xaml.cs

Рисунок 22 –Файл с логикой работы формирования отчета

PrintReport.xaml.cs

```

System.Windows.Controls.PrintDialog Printdlg = new
System.Windows.Controls.PrintDialog();

if ((bool)Printdlg.ShowDialog().GetValueOrDefault())
{
    Size pageSize = new Size(Printdlg.PrintableAreaWidth,
Printdlg.PrintableAreaHeight);

    DataGridServices.Measure(pageSize);

```

```

        DataGridServices.Arrange(new Rect(5, 5, pageSize.Width,
        pageSize.Height));

        Printdlg.PrintVisual(DataGridServices, Title);
    }

```

Окно AddEditPage отвечает за добавление и редактирование услуг администратором.

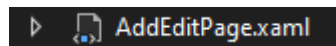


Рисунок 23 –Файл с листингом кода интерфейса окна AddEditPage

Данное окно отвечает сразу за добавление и редактирование услуг. Данный подход помогает уменьшить вес программы за счет того, что не создаются дополнительные файлы с дополнительной логикой.

```

Ссылка 2
public AddEditPage(Service currentService, bool editMode = false)
{
    InitializeComponent();
    _currentService = currentService;
    _editMode = editMode;
    DataContext = currentService;

    ComboType.ItemsSource = _context.TypeService.ToList();
    ComboWorker.ItemsSource = _context.Worker.ToList();
}

Ссылка 1
private void SaveAddServiceBtn_Click(object sender, RoutedEventArgs e)
{
    if (_editMode)
    {
        var initialService = _context.Service.FirstOrDefault(x => x.ID == _currentService.ID);
        if(initialService != null)
        {
            initialService.ServiceName = _currentService.ServiceName;
            initialService.ServicePrice = _currentService.ServicePrice;
            initialService.ServiceTime = _currentService.ServiceTime;
            initialService.TypeService = _currentService.TypeService;
            initialService.Worker = _currentService.Worker;
        }
    }
    else
    {
        _context.Service.Add(_currentService);
    }

    try
    {
        _context.SaveChanges();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Неудается добавить или редактировать услугу {ex.Message.ToString()}");
    }

    Close();
}

```

Рисунок 24 - код для добавления и редактирования

Код для добавления картинок при редактировании и добавлении услуг.

```
private void AddImage_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog fileDialog = new OpenFileDialog()
    {
        Title = "Выберите фото",
        Filter = "Файлы фотографий|*.png;*.jpg",
        Multiselect = false,
    };
    fileDialog.ShowDialog();
    if (fileDialog.FileName == "")
        return;
    byte[] fileAsBytes = File.ReadAllBytes(fileDialog.FileName);
    _currentService.ImageService = fileAsBytes;
    ServiceImage.Source = new BitmapImage(new Uri(fileDialog.FileName));
}
```

Рисунок 25 - код для добавления картинок

Первоначальной страницей запуска программы является авторизация.

Все страницы открываются в основном окне MainWindow и загружаются в Frame.

3.5. Руководство пользователя

Приложение “СтройДом”- автоматизированная информационная система учета работы строительной организации. Оно предназначено для более удобного введения учета услуг строительной организации. Основные функции данного приложения: просмотр услуг, а также их поиск, фильтрация и сортировка. Кроме того, данное приложение может формировать отчеты по услугам в PDFформат.

Приложение “СтройДом” позволяет просматривать все услуги, которые предоставляет строительная организация. С помощью этого приложения можно найти интересующую услугу, отфильтровать все услуги по их типам, в рамках данного приложения есть все два типа – ремонт и строительство. Также данное приложение позволяет отсортировать все услуги по алфавиту, цене и времени выполнения. Для администратора предусмотрены дополнительные функции, такие как добавление, редактирование и удаление услуг.

Интерфейс приложения выполнен в приятных для глаз цветах, стилистически подобранными для строительной организации.. Текст приложения легко выделяется на остальном фоне приложения. Интерфейс интуитивно понятен для пользователя. Все кнопки расположены на удобных местах для пользователя.

Для того, чтобы начать пользоваться приложением нужно выполнить следующие шаги:

1. Запустить приложение;
2. Авторизоваться;
 - а. Если пользователь введет неверные данные, то нужно будет ввести капчу для повторной авторизации.
3. Войти в систему.

После успешной авторизации от имени менеджера можно будет:

1. Просматривать информацию об услугах;
2. Выполнять поиск по услугам;
3. Фильтровать услуги по типам;
4. Сортировать услуги по алфавиту, цене и времени работы.

После успешной авторизации от имени администратора можно будет выполнять все вышеперечисленные действия, а также:

1. Добавлять услуги;
2. Редактировать услуги;
3. Удалять услуги.

Основные функции приложения:

1. Поиск по названию услуги;
2. Фильтрация услуг;
3. Сортировка услуг;
4. Добавление услуг;
5. Редактирование услуг;
6. Удаление услуг;
7. Формирование отчета.

Для того, чтобы начать поиск по услугам необходимо:

1. Авторизоваться в системе;
2. Нажать на текстовое поле в верхней части экрана;
3. Начать вводить необходимое название услуги.

Для фильтрации услуг необходимо:

1. Авторизоваться в системе;
2. Нажать на выпадающий список фильтрации в верхней части экрана;
3. Выбрать нужный тип фильтрации.

Для сортировки услуг необходимо:

1. Авторизоваться в системе;
2. Нажать на выпадающий список сортировки в верхней части экрана;
3. Выбрать нужный тип сортировки.

Добавление и редактирование услуг происходит в одном окне. Для добавления услуги необходимо:

1. Авторизоваться в системе в качестве администратора;
2. Нажать на кнопку “Добавить услугу”;
3. В текстовые поля, открывшегося окна, ввести необходимые данные;
4. Нажать на кнопку “Сохранить”.

Для редактирования услуги необходимо:

1. Авторизоваться в системе в качестве администратора;
2. Выбрать услугу, которую необходимо отредактировать;
3. Нажать на кнопку “Редактировать услугу”;
4. В текстовые поля, открывшегося окна, ввести необходимые данные;
5. Нажать на кнопку “Сохранить”.

Для удаления услуг необходимо:

1. Авторизоваться в системе в качестве администратора;

2. Выбрать услугу, которую необходимо удалить;
3. Нажать на кнопку “Удалить услугу”;

Для формирования отчета необходимо:

1. Авторизоваться в системе;
2. Нажать на кнопку “Сформировать отчет”;
3. Отфильтровать услуги в отчете, если это нужно;
4. Нажать на кнопку “Сформировать отчет”;
5. Далее необходимо выбрать действие с отчетом: распечатать его или сформировать в PDFформат;

Требования к изображениям:

1. Формат изображения только .png или .jpg;
2. Размер изображения не имеет смысла, так как программа сама настроит размеры изображения.

Раздел 4. Тестирование программных решений

4.1. Функциональное тестирование

Для проведения функционального тестирования нами были разработаны таблицы, которые содержат перечень сценариев тестирования программного продукта (test cases) с подробным описанием шагов. Каждый шаг сценария тестирования характеризуется действиями пользователя и ожидаемыми результатами – ответной реакции автоматизированной информационной системы на эти действия.

Таблица 1 “Кнопка авторизации”

Тестовый пример #	№1
Приоритет тестирования	Средний
Заголовок/название теста	Авторизация
Краткое изложение теста	Пользователь должен войти в систему
Этапы теста	<ol style="list-style-type: none">1. Ввести логин2. Ввести пароль3. Нажать на кнопку “Авторизоваться”4. Войти в систему
Тестовые данные	<ol style="list-style-type: none">1. Логин пользователя: slava666777@bk.ru2. Пароль пользователя: qwe123!
Ожидаемый результат	Пользователь должен войти в систему
Фактический результат	Пользователь зашел в систему
Предварительное условие	В базу должны быть загружены данные
Постусловие	Пользователь зашел в систему

Таблица 2 “Кнопка удаления услуги”

Тестовый пример #	№2
Приоритет тестирования	Средний
Заголовок/название теста	Удаление услуги
Краткое изложение теста	Услуга должна без ошибок удалиться из таблицы услуги
Этапы теста	<ol style="list-style-type: none"> 1. Выделить услугу 2. Нажать на кнопку “Удалить услугу” 3. Проверить наличие удаленной услуги в таблице
Тестовые данные	Название: Тестовая услуга Тип услуги: Строительство Время выполнения от: 24ч Цена от: 1,000.00 руб/кв.м
Ожидаемый результат	Запись должна быть удалена из списка услуг без ошибок и исключений
Фактический результат	Запись удалена
Предварительное условие	В базу должен быть загружен тестовый продукт
Постусловие	Услуга должна удалиться из списка круизов

Таблица 3 “Кнопка Добавление услуги”

Тестовый пример #	№3
Приоритет тестирования	Средний
Заголовок/название теста	Добавление услуги
Краткое изложение теста	Услуга должна без ошибок добавиться в таблицу услуги
Этапы теста	<ol style="list-style-type: none"> 1. Нажать на кнопку “Добавить услугу” 2. Ввести необходимые данные 3. Нажать на кнопку “Сохранить”

	4. Проверить наличие добавленной записи в таблице
Тестовые данные	Название: Тестовая услуга Тип услуги: Строительство Время выполнения от: 24ч Цена от: 1,000.00 руб/кв.м
Ожидаемый результат	Запись должна быть добавлена в список услуг без ошибок и исключений
Фактический результат	Запись добавлена
Предварительное условие	Нет
Постусловие	Услуга должна добавиться в список услуг

Таблица 4“Кнопка Редактирование услуги”

Тестовый пример #	№4
Приоритет тестирования	Средний
Заголовок/название теста	Редактирование услуги
Краткое изложение теста	Услуга должна без ошибок редактироваться в таблице услуги
Этапы теста	<ol style="list-style-type: none"> 1. Нажать на кнопку “Редактировать услугу” 2. Ввести необходимые данные 3. Нажать на кнопку “Сохранить” 4. Проверить наличие отредактированной записи в таблице
Тестовые данные	Название: Тестовая услуга редактирование Тип услуги: ремонт Время выполнения от: 42ч Цена от: 1,500.00 руб/кв.м
Ожидаемый результат	Запись должна быть отредактирована в списке услуг без ошибок и исключений
Фактический результат	Запись отредактирована
Предварительное условие	В базу должны быть загружен тестовый продукт

Постусловие	Услуга должна редактироваться в список услуг
-------------	--

Таблица 5 “Текстовое поле Поиск услуги”

Тестовый пример #	№5
Приоритет тестирования	Средний
Заголовок/название теста	Поиск услуги
Краткое изложение теста	Услуга без ошибок должна быть найдена
Этапы теста	1. Нажать на текстовое поле для поиска услуг 2. Ввести необходимые данные
Тестовые данные	Название: Тестовая услуга Тип услуги: Строительство Время выполнения от: 24ч Цена от: 1,000.00 руб/кв.м
Ожидаемый результат	Запись должна быть найдена в списке услуг без ошибок и исключений
Фактический результат	Запись найдена
Предварительное условие	В базу должны быть загружены тестовые продукты
Постусловие	Услуга должна быть найдена

Таблица 6 “Выпадающий список Сортировка услуг”

Тестовый пример #	№6
Приоритет тестирования	Средний
Заголовок/название теста	Сортировка услуг
Краткое изложение теста	Услуги должны без ошибок сортироваться по алфавиту
Этапы теста	1. Нажать на выпадающий список 2. Выбрать “Сортировать по алфавиту”
Тестовые данные	Название: Тестовая услуга Тип услуги: Строительство

	Время выполнения от: 24ч Цена от: 1,000.00 руб/кв.м
Ожидаемый результат	Запись должна быть отсортирована в списке услуг без ошибок и исключений
Фактический результат	Запись отсортирована
Предварительное условие	В базу должны быть загружен тестовый продукт
Постусловие	Услуга должна быть отсортирована

Таблица 7 “Выпадающий список Фильтрация услуг по типу”

Тестовый пример #	№7
Приоритет тестирования	Средний
Заголовок/название теста	Фильтрация услуг по типу
Краткое изложение теста	Услуги должны без ошибок фильтроваться по типу
Этапы теста	1. Нажать на выпадающий список 2. Выбрать необходимый тип услуг
Тестовые данные	Название: Тестовая услуга Тип услуги: Строительство Время выполнения от: 24ч Цена от: 1,000.00 руб/кв.м
Ожидаемый результат	Запись должна быть отфильтрована в списке услуг без ошибок и исключений
Фактический результат	Запись отфильтрована
Предварительное условие	В базу должны быть загружен тестовый продукт
Постусловие	Услуга должна быть отфильтрована

Таблица 8 “Кнопка Сформировать отчёт”

Тестовый пример #	№8
Приоритет тестирования	Средний
Заголовок/название теста	Формирование отчёта
Краткое изложение теста	Отчёт должен сформироваться без ошибок
Этапы теста	<ol style="list-style-type: none"> 1. Нажать на кнопку “Сформировать отчет” 2. Отфильтровать услуги, если надо 3. Нажать на кнопку “Сформировать отчет” 4. Выбрать средство печати отчета 5. Распечатать
Тестовые данные	Название: Тестовая услуга Тип услуги: Строительство Время выполнения от: 24ч Цена от: 1,000.00 руб/кв.м
Ожидаемый результат	Отчёт должен сформироваться без ошибок и исключений
Фактический результат	Отчёт сформирован
Предварительное условие	В базу должны быть загружен тестовый продукт
Постусловие	Отчёт должен быть сформирован

4.2. Руководство по установке и внедрению программного решения

Руководство по внедрению может быть использовано при установке системы с привлечением консультантов, так и при самостоятельном внедрении.

План внедрения:

1. Ознакомьтесь руководством пользователя данной автоматизированной системы (п.3.5.);
2. Рассчитайте, сколько необходимо компьютеров, для работы в данной АИС;

Для работы необходимо от двух рабочих мест (компьютеров). Одно это рабочее место администратора, другое место менеджера.

3. Программный комплекс не требователен к оборудованию и может работать на любых компьютерах, которые выпускаются с 2015 года. Рекомендуемая конфигурация представлена в п 1.3.

4. Рекомендуется подключить высокоскоростной интернет для ускорения процесса обмена данными.

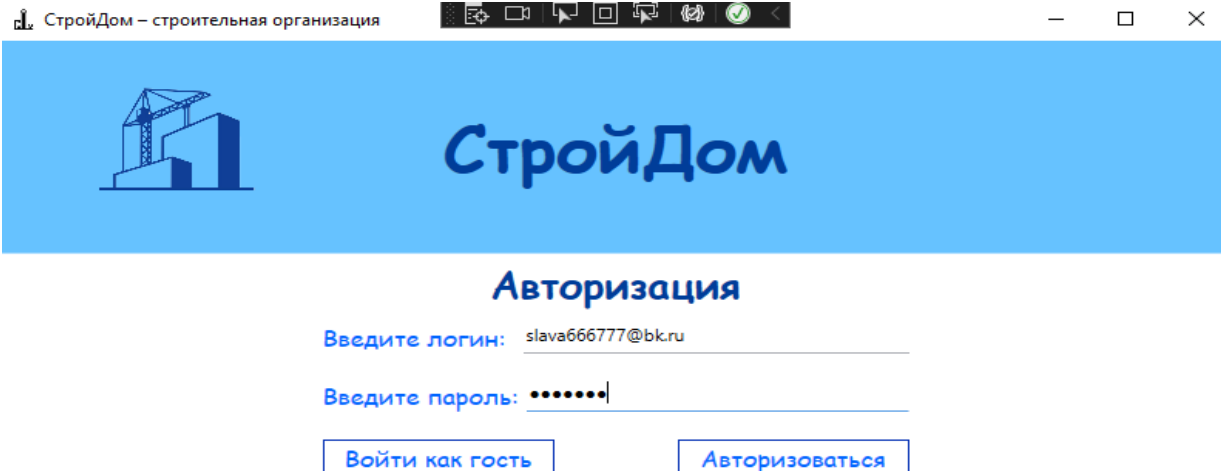
5. Чтобы зайти в данную АИС требуется:

5.1. Запустить exe файл программы.


База данных храниться в SQL Server Management Studio Management Studio 19.

К ней имеет доступ Администратор.

6. Для доступа к функционалу АИС необходимо осуществить вход в программу под учетной записью администратора. Имя пользователя slava666777@bk.ru Пароль qwe123!



СтройДом – строительная организация

 **СтройДом**

Авторизация

Введите логин:

Введите пароль:

7. Раздать пароли работникам.

Для реализации внедрения разработанной АИС, необходимо руководствоваться программой внедрения, которая включает в себя:

- адаптация программного продукта под задачи бизнеса, в том числе частичную доработку функционала;
- настройку программного продукта;
- обучение пользователей программы;
- поддержку пользователей и сопровождение процесса внедрения до получения запланированного эффекта.

Заключение

В данном дипломном проекте была проведена разработка автоматизированной информационной системы для учёта работы строительной организации, с учетом необходимых минимальных потребностей, возникающих в результате работы любой строительной организации.

В итоге создания АИС для учета работы строительной организации основная цель проекта была достигнута и результатом представлен удобный и функциональный программный продукт для управления учетом работ строительной организации.

В ходе написания дипломного проекта были решены поставленные задачи. В первом разделе работы произведен анализ предметной области, выявлена необходимость автоматизации процесса работы учета строительной организации, предложены выбор и обоснование своего варианта решения задачи. Во втором разделе разработано концептуальное проектирование информационной системы. В третьем разделе описана реализация информационной системы, разработка руководства по использованию системы. В четвертом разделе представлено тестирование разработанной автоматизированной информационной системы.

В результате произведенных исследований и анализа предметной области был определен перечень функциональных и нефункциональных требований к системе. Для выполнения поставленных задач было использовано современное программное обеспечение Visual Studio 2022 и технологии баз данных MSSQL Server 2022.

Разработанная АИС имеет ряд преимуществ перед аналогами, такие как, простота и понятность интерфейса, удобный механизм поиска и фильтрации информации, создания отчетов, выгрузка в PDF для печати и редактирования, а также для загрузки в другие программные продукты и

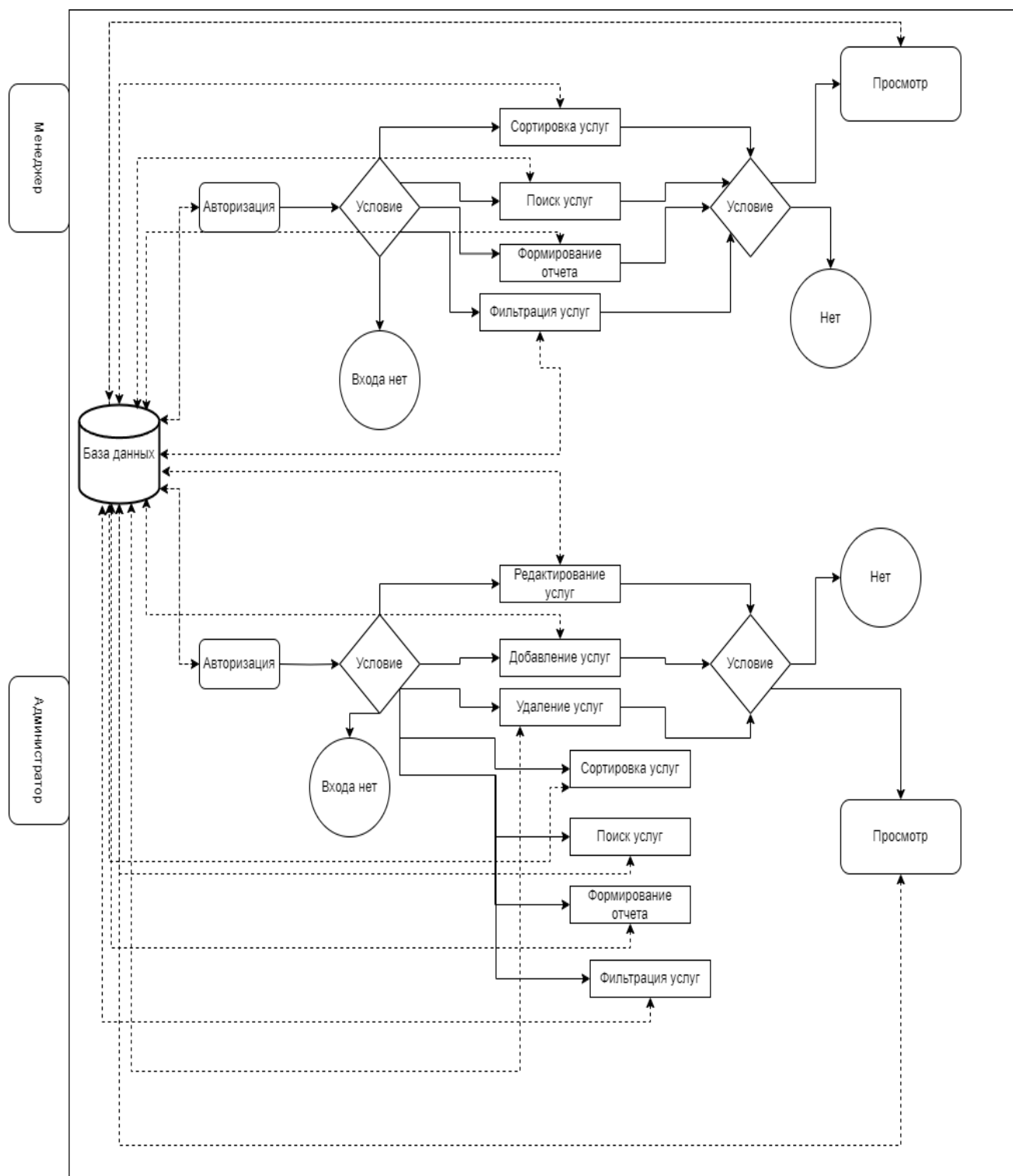
т.д. Представляет собой совокупность данных, предназначена для обработки информации и принятия управленческих решений.

В итоге, применение разработанной АИС для учёта работы строительной организации позволит упростить и ускорить процесс учета работы строительной организации, минимизировать ошибки в учете работы. Данный проект имеет высокий потенциал для успешной коммерциализации в сфере строительства.

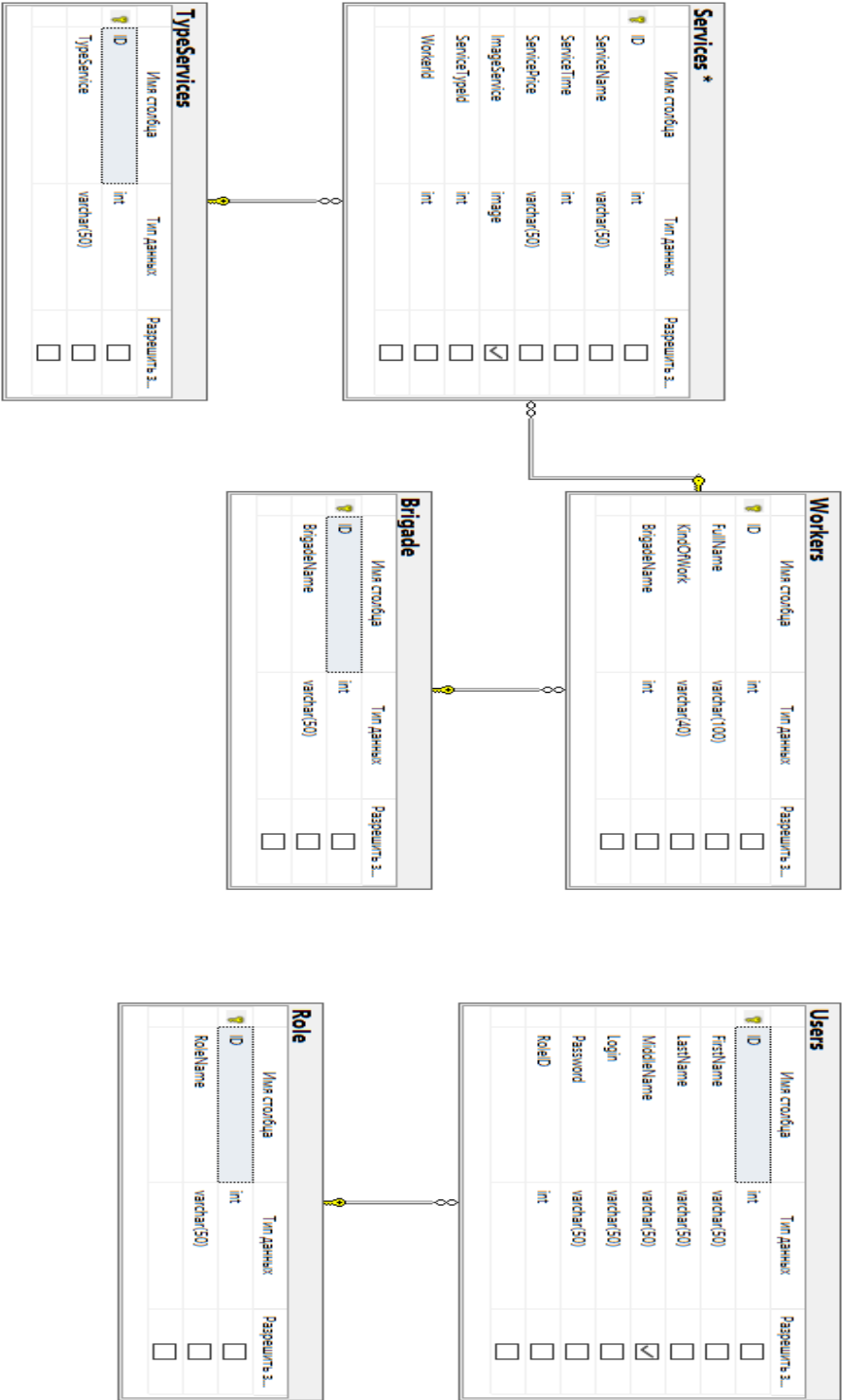
Список использованных информационных источников

1. Грацианова, Т.Ю. Программирование в примерах и задачах / Т.Ю. Грацианова. – СПб., 2022.
2. Колдаев, В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев. – СПб., 2020
3. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – 4-е изд., перераб. и доп. – М., 2022.
4. Голицына, О.Л. Основы проектирования баз данных: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. – 2-е изд. – М., 2021.
5. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – 3-е изд., дополненное. – СПб., 2019.
6. Одинцов, И. Профессиональное программирование: системный подход / И. Одинцов. – СПб.: «bhv», 2019
7. Федорова, Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем: учебник / Г.Н. Федорова. – СПб., 2020.
8. Черников, Б.В. Управление качеством программного обеспечения: Учебник / Б.В. Черников. – СПб., 2019
9. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – 3-е изд., дополненное. – СПб., 2020.
10. [https://studfile.net/preview/5166446/#:~:text=Строительная%20организация%20-%20это%20лицетворенная,работы\)%20соответствующего%20назначения%20С%20профиля%20С%20типа](https://studfile.net/preview/5166446/#:~:text=Строительная%20организация%20-%20это%20лицетворенная,работы)%20соответствующего%20назначения%20С%20профиля%20С%20типа)
11. <https://studfile.net/preview/9160205/page:4/>
12. <https://www.kp.ru/guide/struktura-stroitel-noi-organizatsii.html>
13. <https://kamala-soft.com/blog/uchet-v-stroitelstve-osobennosti-i-poryadok-avtomatizatsii/>

14. [https://studfile.net/preview/5166446/#:~:text=Строительная%20организация%20-%20это%20лицетворенная,работы\)%20соответствующего%20назначения%2C%20профиля%2C%20типа](https://studfile.net/preview/5166446/#:~:text=Строительная%20организация%20-%20это%20лицетворенная,работы)%20соответствующего%20назначения%2C%20профиля%2C%20типа)
15. <https://www.1cbit.ru/1csoft/bit-stroitelstvo-365/>
16. <https://picktech.ru/product/procore>
17. <https://picktech.ru/product/buildertrend>



Приложение Б ERD-диаграмма



Приложение В

Скрипт БД

```
USE [BuildDb]
GO
/***** Object: Table [dbo].[Brigade]  Script Date: 16.05.2023 21:05:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Brigade](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [BrigadeName] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Brigade] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
    ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[NumberOfWorkers]  Script Date: 16.05.2023 21:05:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[NumberOfWorkers](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [NumberOfWorkers] [int] NOT NULL,
    [BrigadeName] [int] NOT NULL,
    CONSTRAINT [PK_NumberOfWorkers] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
    ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Role]  Script Date: 16.05.2023 21:05:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Role](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [RoleName] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Role] PRIMARY KEY CLUSTERED
```

```

(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
    ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Services]  Script Date: 16.05.2023 21:05:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Services](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [ServiceName] [varchar](50) NOT NULL,
    [ServiceTime] [int] NOT NULL,
    [ServicePrice] [varchar](50) NOT NULL,
    [ImageService] [image] NULL,
    [ServiceTypeId] [int] NOT NULL,
    [NumberOfWorkersId] [int] NOT NULL,
    CONSTRAINT [PK_Services] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
    ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[TypeServices]  Script Date: 16.05.2023 21:05:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TypeServices](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [TypeService] [varchar](50) NOT NULL,
    CONSTRAINT [PK_TypeServices] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
    ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Users]  Script Date: 16.05.2023 21:05:02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE TABLE [dbo].[Users](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [FirstName] [varchar](50) NOT NULL,
    [LastName] [varchar](50) NOT NULL,
    [MiddleName] [varchar](50) NULL,
    [Login] [varchar](50) NOT NULL,
    [Password] [varchar](50) NOT NULL,
    [RoleID] [int] NOT NULL,
    CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
    ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[NumberOfWorkers] WITH CHECK ADD CONSTRAINT [FK_NumberOfWorkers_Brigade] FOREIGN
    KEY([BrigadeName])
REFERENCES [dbo].[Brigade] ([ID])
GO
ALTER TABLE [dbo].[NumberOfWorkers] CHECK CONSTRAINT [FK_NumberOfWorkers_Brigade]
GO
ALTER TABLE [dbo].[Services] WITH CHECK ADD CONSTRAINT [FK_Services_NumberOfWorkers] FOREIGN
    KEY([NumberOfWorkersId])
REFERENCES [dbo].[NumberOfWorkers] ([ID])
GO
ALTER TABLE [dbo].[Services] CHECK CONSTRAINT [FK_Services_NumberOfWorkers]
GO
ALTER TABLE [dbo].[Services] WITH CHECK ADD CONSTRAINT [FK_Services_TypeServices1] FOREIGN
    KEY([ServiceTypeId])
REFERENCES [dbo].[TypeServices] ([ID])
GO
ALTER TABLE [dbo].[Services] CHECK CONSTRAINT [FK_Services_TypeServices1]
GO
ALTER TABLE [dbo].[Users] WITH CHECK ADD CONSTRAINT [FK_Users_Role] FOREIGN KEY([RoleID])
REFERENCES [dbo].[Role] ([ID])
GO
ALTER TABLE [dbo].[Users] CHECK CONSTRAINT [FK_Users_Role]
GO

```

MainWindow.xml

```
<Window x:Class="PDPBuild.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:PDPBuild"
    mc:Ignorable="d"
    Title="СтройДом – строительная организация" Height="500" Width="800" Icon="/Images/Icons/Icon.ico">
    <Grid Background="{StaticResource BackColor}">
    <Grid.RowDefinitions>
    <RowDefinition Height="1*"/>
    <RowDefinition Height="2*"/>
    </Grid.RowDefinitions>
    <Grid Grid.Row="0" Background="{StaticResource BorderBackColor}">
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <StackPanel Grid.Column="0" Orientation="Horizontal">
    <Image Source="Images/Logo.png"/>
    </StackPanel>
    <Viewbox Grid.Column="1" StretchDirection="Both" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Width="258">
    <StackPanel Orientation="Horizontal">
    <TextBlock Text="СтройДом" Margin="10" Style="{StaticResource MainText}"/>
    </StackPanel>
    </Viewbox>
    <Grid x:Name="AuthedPanel" Visibility="Hidden" Margin="10" HorizontalAlignment="Stretch" Grid.Column="2">
    <Viewbox StretchDirection="DownOnly" HorizontalAlignment="Stretch" VerticalAlignment="Top">
    <TextBlock x:Name="AuthedUserTxt" Style="{StaticResource Text}" FontSize="30" VerticalAlignment="Top"/>
    </Viewbox>
    <Button x:Name="LogoutButton" Click="LogoutButton_Click" Style="{StaticResource BtnExitStyle}" FontSize="18"
        Content="Выйти" Width="200" VerticalAlignment="Bottom"/>
    </Grid>
    </Grid>
    <Frame x:Name="MainFrame" Grid.Row="1" NavigationUIVisibility="Hidden"/>
    </Grid>
</Window>
```

MainWindow.xaml.cs

```
public MainWindow()
{
    InitializeComponent();
}
```



```

        WindowStartupLocation = WindowStartupLocation.CenterScreen;
        MinHeight = 500;
        MinWidth = 815;
        MainFrame.Navigate(new Authorization());
    }
    private void LogoutButton_Click(object sender, RoutedEventArgs e)
    {
        MainFrame.Navigate(new Authorization());
        AuthedPanel.Visibility = Visibility.Hidden;
        MessageBox.Show("До свидания !");
    }

```

Authorization.xaml

```

<Page x:Class="PDPBuild.Pages.Authorization"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:PDPBuild.Pages"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="Authorization">
    <Grid Background="{StaticResource BackColor}">
    <Border Width="400" Height="300">
    <StackPanel >
    <TextBlock Text="Авторизация" Style="{StaticResource MainText}" HorizontalAlignment="Center"/>
    <StackPanel Orientation="Horizontal" Margin="10">
    <Grid>
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="130"/>
    <ColumnDefinition Width="1*" />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Введите логин: " Style="{StaticResource Text}"/>
    <TextBox Grid.Column="1" x:Name="InputLogin" Width="250" BorderThickness="0 0 0 1"/>
    </Grid>
    </StackPanel>
    <StackPanel Orientation="Horizontal" Margin="10">
    <Grid>
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="2*" />
    <ColumnDefinition Width="1*" />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Введите пароль: " Style="{StaticResource Text}"/>
    <PasswordBox Grid.Column="1" x:Name="InputPassword" Width="250" BorderThickness="0 0 0 1"/>
    </Grid>
    </StackPanel>
    </Grid>

```

```

<Button x:Name="AuthAsGuestBtn" Click="AuthAsGuestBtn_Click" Content="Войти как гость" Style="{StaticResource
    BtnExitStyle}" Width="150" Margin="10" HorizontalAlignment="Left"/>
<Button x:Name="AuthAsUserbtn" Click="AuthAsUserbtn_Click" Content="Авторизоваться" Style="{StaticResource
    BtnExitStyle}" Width="150" Margin="10" HorizontalAlignment="Right"/>
</Grid>
<StackPanel x:Name="CapchaPanel" Margin="10" Visibility="Hidden">
<TextBlock x:Name="CapchaTxt" Style="{StaticResource Text}" Background="{StaticResource BorderBackColor}"
    FontStyle="Italic" Padding="10" Margin="10" HorizontalAlignment="Center" TextDecorations="Strikethrough"
    FontSize="30"/>
<TextBox x:Name="InputCapchaTxt" Width="250"/>
</StackPanel>
</StackPanel>
</Border>
</Grid>
</Page>

```

Authorization.xaml.cs

```

public partial class Authorization : Page
{
    private readonly BuildDbEntities _context = new BuildDbEntities();
    public Authorization()
    {
        InitializeComponent();
    }
    private void AuthAsUserbtn_Click(object sender, RoutedEventArgs e)
    {
        var currentUser = _context.User.ToList();
        var auth = currentUser.FirstOrDefault(x => x.Login == InputLogin.Text && x.Password == InputPassword.Password);
        if(auth != null)
        {
            NavigationService.Navigate(new ListServices(auth.RoleID));
            ((MainWindow)Window.GetWindow(this)).AuthedPanel.Visibility = Visibility.Visible;
            ((MainWindow)Window.GetWindow(this)).AuthedUserTxt.Text = $"Здравствуйте {auth.FirstName} {auth.LastName}";
        }
        else
        {
            MessageBox.Show("Вы ввели некорректные данные!", "Внимание");
            CapchaPanel.Visibility = Visibility.Visible;
            GenerateCapcha();
            if (InputCapchaTxt.Text != CapchaTxt.Text)
            {
                AuthAsUserbtn.IsEnabled = false;
                DispatcherTimer timer = new DispatcherTimer()
                {
                    Interval = new TimeSpan(0, 0, 3)
                };
                timer.Tick += (bt, ea) =>

```

```

        {
            AuthAsUserbtn.IsEnabled = true;
            timer.Stop();
        };
        timer.Start();
    }
}
}
private void AuthAsGuestBtn_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new ListServices(null));
    ((MainWindow)Window.GetWindow(this)).AuthedPanel.Visibility = Visibility.Visible;
    ((MainWindow)Window.GetWindow(this)).AuthedUserTxt.Text = "Вы вошли как гость";
}
private void GenerateCapcha()
{
    Random rand = new Random();
    CapchaTxt.Text = string.Empty;
    for (int i = 0; i < 2; i++)
    {
        CapchaTxt.Text += (char)('a' + rand.Next(0, 26));
        CapchaTxt.Text += (char)('A' + rand.Next(0, 26));
        CapchaTxt.Text += (char)('0' + rand.Next(0, 9));
    }
}
}
}

```

ListServices.xaml

```

<Page x:Class="PDPBuild.Pages.ListServices"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:PDPBuild.Pages"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="ListServices">
    <Grid Background="{StaticResource BackColor}">
    <Grid.RowDefinitions>
    <RowDefinition Height="30"/>
    <RowDefinition Height="*/>
    <RowDefinition Height="30"/>
    </Grid.RowDefinitions>
    <Viewbox Grid.Row="0" StretchDirection="Both" HorizontalAlignment="Stretch" VerticalAlignment="Center">
    <StackPanel Orientation="Horizontal" Margin="5 0">
    <StackPanel Orientation="Horizontal" HorizontalAlignment="Left" Margin="5 0">

```

```

<TextBlock Text="Названиеуслуги: " Style="{StaticResource Text}" VerticalAlignment="Center"/>
<TextBox x:Name="SearchTxt" TextChanged="SearchTxt_TextChanged" Width="150" FontSize="16" Height="25"/>
<Button x:Name="ClearTxt" Click="ClearTxt_Click" Style="{StaticResource BtnExitStyle}" Content="X" Margin="1 0 0 0"/>
</StackPanel>
<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="5 0">
<TextBlock Text="Типуслуг: " Style="{StaticResource Text}" VerticalAlignment="Center"/>
<ComboBox
        x:Name="ComboTypesServices"
        SelectionChanged="ComboTypesServices_SelectionChanged"
        DisplayMemberPath="ServiceType" Width="150"></ComboBox>
</StackPanel>
<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" Margin="5 0">
<TextBlock Text="Сортироватьпо: " Style="{StaticResource Text}" VerticalAlignment="Center"/>
<ComboBox x:Name="ComboSortServices" SelectionChanged="ComboSortServices_SelectionChanged" Width="130">
<TextBlock Text="Алфавиту"/>
<TextBlock Text="Времениработы"/>
<TextBlock Text="Цене"/>
</ComboBox>
</StackPanel>
</StackPanel>
</Viewbox>
<ListView
        x:Name="ListViewServices"
        Grid.Row="1"
        Background="{StaticResource BackColor}"
        ScrollViewer.HorizontalScrollBarVisibility="Disabled">
<ListView.ItemsPanel>
<ItemsPanelTemplate>
<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"/>
</ItemsPanelTemplate>
</ListView.ItemsPanel>
<ListView.ItemTemplate>
<DataTemplate>
<Border BorderBrush="{StaticResource BorderBackColor}" BorderThickness="2" Margin="10">
<Grid Margin="30" Width="450">
<Grid.ColumnDefinitions>
<ColumnDefinition Width="70"/>
<ColumnDefinition Width="1*"/>
<ColumnDefinition Width="125"/>
</Grid.ColumnDefinitions>
<Image Grid.Column="0" Source="{Binding ImageService, TargetNullValue={StaticResource NullImage}}"/>
<StackPanel Grid.Column="1" Margin="10 0 0 0">
<TextBlock Text="{Binding ServiceName}" Style="{StaticResource Text}" FontWeight="Bold"/>
<TextBlock Text="{Binding TypeService.ServiceType}" Style="{StaticResource Text}" FontStyle="Italic"/>
<TextBlock Text="{Binding ServiceTime, StringFormat={ }Времявыполненияот: {0} ч.}" Style="{StaticResource Text}"/>
<TextBlock Text="{Binding ServicePrice, StringFormat={ }Ценаот: {0:N2} руб/кв. м.}" Style="{StaticResource Text}"/>
</StackPanel>
</Grid>
</Border>
</DataTemplate>
</ListView.ItemTemplate>

```

```

</ListView>
<Grid Grid.Row="2">
<StackPanel x:Name="AdminPanel" Visibility="Hidden" Orientation="Horizontal">
<Button      x:Name="EditService"      Click="EditService_Click"      Style="{StaticResource      BtnExitStyle}"
      Content="Редактировать услугу" Margin="5 0"/>
<Button      x:Name="DelServiceBtn"      Click="DelServiceBtn_Click"      Style="{StaticResource      BtnExitStyle}"
      Content="Удалить услугу" Margin="5 0"/>
<Button      x:Name="AddServiceBtn"      Click="AddServiceBtn_Click"      Style="{StaticResource      BtnExitStyle}"
      Content="Добавить услугу" Margin="5 0"/>
</StackPanel>
<StackPanel x:Name="PrintReport" HorizontalAlignment="Right" Margin="5 0" Visibility="Hidden">
<Button x:Name="GoPrintReportBtn" Click="GoPrintReportBtn_Click" Content="Сформировать отчет" Style="{StaticResource
      BtnExitStyle}"/>
</StackPanel>
</Grid>
</Grid>
</Page>

```

ListServices.xaml.cs

```

private BuildDbEntities _context = new BuildDbEntities();
public ListServices(int? RoleID)
{
    InitializeComponent();
    if(RoleID == 1)
    {
        AdminPanel.Visibility = Visibility.Visible;
        PrintReport.Visibility = Visibility.Visible;
    }
    if (RoleID == 2)
    {
        PrintReport.Visibility = Visibility.Visible;
    }
    var allTypes = _context.TypeService.ToList();
    allTypes.Insert(0, new TypeService
    {
        ServiceType = "Всё типы"
    });
    ComboTypesServices.ItemsSource = allTypes;
    ComboTypesServices.SelectedIndex = 0;
}
private void SearchTxt_TextChanged(object sender, TextChangedEventArgs e)
{
    UpdateServices();
}
private void ComboTypesServices_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    UpdateServices();
}

```

```

    }
    private void UpdateServices()
    {
        var currentServices = _context.Service.ToList();

        if (ComboTypesServices.SelectedIndex > 0)
            currentServices = currentServices.Where(x => x.TypeService == ComboTypesServices.SelectedItem).ToList();

        currentServices = currentServices.Where(x => x.ServiceName.ToLower().Contains(SearchTxt.Text.ToLower())).ToList();

        ListViewServices.ItemsSource = currentServices;
    }
    private void ComboSortServices_SelectionChanged(object sender, SelectionChangedEventArgs e)
    {
        if (ComboSortServices.SelectedIndex == 0)
        {
            var result = _context.Service.OrderBy(x => x.ServiceName).ToList();
            ListViewServices.ItemsSource = result;
        }
        if (ComboSortServices.SelectedIndex == 1)
        {
            var result = _context.Service.OrderBy(x => x.ServiceTime).ToList();
            ListViewServices.ItemsSource = result;
        }
        if (ComboSortServices.SelectedIndex == 2)
        {
            var result = _context.Service.OrderBy(x => x.ServicePrice).ToList();
            ListViewServices.ItemsSource = result;
        }
    }
    private void ClearTxt_Click(object sender, RoutedEventArgs e)
    {
        SearchTxt.Text = string.Empty;
    }
    private void AddServiceBtn_Click(object sender, RoutedEventArgs e)
    {
        AddEditPage addEditPage = new AddEditPage(new Service(), false, "Добавление");
        addEditPage.Show();
        addEditPage.Closed += (s, ea) => { Update(); };
    }
    private void DelServiceBtn_Click(object sender, RoutedEventArgs e)
    {
        MessageBoxResult deleteService = MessageBox.Show($"Удалить  

        (((Service)ListViewServices.SelectedItem).ServiceName) ?", "Удаление", MessageBoxButton.YesNo);

        if (deleteService == MessageBoxResult.Yes)
    
```

```

    {
        try
        {
            _context.Service.Remove((Service)ListViewServices.SelectedItem);
            _context.SaveChanges();
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Неудается удалить услугу {ex.Message.ToString()}");
        }
    }
    return;
}
private void Update()
{
    if (Visibility == Visibility.Visible)
    {
        ListViewServices.ItemsSource = _context.Service.ToList();
    }
}
private void EditService_Click(object sender, RoutedEventArgs e)
{
    AddEditPage addEditPage = new AddEditPage((Service)ListViewServices.SelectedItem, true, "Редактирование");
    addEditPage.Show();
}
private void GoPrintReportBtn_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new PrintReport());
}
}

```

PrintReport.xaml

```

<Page x:Class="PDPBuild.Pages.PrintReport"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:PDPBuild.Pages"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="PrintReport">
    <Grid Background="{StaticResource BackColor}">
    <StackPanel>
    <Grid>
    <ComboBox x:Name="ComboTypes" SelectionChanged="ComboTypes_SelectionChanged" DisplayMemberPath="ServiceType"
        Width="150" Margin="10" VerticalAlignment="Top" HorizontalAlignment="Left"/>

```

```

<Button      x:Name="GoBackBtn"      Click="GoBackBtn_Click"      Content="Назад"      VerticalAlignment="Top"
      HorizontalAlignment="Right" Margin="10" Style="{StaticResource BtnExitStyle}"/>
</Grid>
<DataGrid x:Name="DataGridServices" AutoGenerateColumns="False" ItemsSource="{Binding}" AlternationCount="1">
<DataGrid.Columns>
<DataGridTextColumn Width="*" Header="Наименованиеуслуги" Binding="{Binding ServiceName}"/>
<DataGridTextColumn Width="*" Header="Времявыполненияуслуги" Binding="{Binding ServiceTime, StringFormat={}{0}
      часов}"/>
<DataGridTextColumn Width="*" Header="Ценауслуги" Binding="{Binding ServicePrice, StringFormat={}{0:N2} руб.}"/>
<DataGridTextColumn Width="*" Header="Типуслуги" Binding="{Binding TypeService.ServiceType}"/>
<DataGridTextColumn Width="*" Header="Рабочий" Binding="{Binding Worker.FullName}"/>
</DataGrid.Columns>
</DataGrid>
<Button      x:Name="PrintReportBtn"      Click="PrintReportBtn_Click"      Content="Сформироватьотчет"      Style="{StaticResource
      BtnExitStyle}" Width="200" Margin="0 10"/>
</StackPanel>
</Grid>
</Page>

```

PrintReport.xaml.cs

```

private readonly BuildDbEntities _context = new BuildDbEntities();

public PrintReport()
{
    InitializeComponent();
    var allTypes = _context.TypeService.ToList();
    allTypes.Insert(0, new TypeService
    {
        ServiceType = "Все типы"
    });
    ComboTypes.ItemsSource = allTypes;
    ComboTypes.SelectedIndex = 0;

    DataGridServices.ItemsSource = _context.Service.ToList();
}

private void ComboTypes_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    var currentTypeService = _context.Service.ToList();

    if(ComboTypes.SelectedIndex > 0)
        currentTypeService = currentTypeService.Where(x => x.TypeService == ComboTypes.SelectedItem).ToList();

    DataGridServices.ItemsSource = currentTypeService;
}

private void GoBackBtn_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}

```



```

private void PrintReportBtn_Click(object sender, RoutedEventArgs e)
{
    System.Windows.Controls.PrintDialog Printdlg = new System.Windows.Controls.PrintDialog();
    if ((bool)Printdlg.ShowDialog().GetValueOrDefault())
    {
        Size pageSize = new Size(Printdlg.PrintableAreaWidth, Printdlg.PrintableAreaHeight);
        DataGridServices.Measure(pageSize);
        DataGridServices.Arrange(new Rect(5, 5, pageSize.Width, pageSize.Height));
        Printdlg.PrintVisual(DataGridServices, Title);
    }
}
}

```

AddEditPage.xaml

```

<Window x:Class="PDPBuild.Windows.AddEditPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:PDPBuild.Windows"
    mc:Ignorable="d"
    Title="AddEditPage" Height="500" Width="800">
    <Grid>
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <StackPanel Margin="5 0">
    <TextBlock x:Name="TitleText" Style="{StaticResource MainText}" FontSize="20" HorizontalAlignment="Center" Margin="0
        10"/>
    <Grid Margin="0 5">
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="150"/>
    <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <TextBlock Text="Название услуги: " Style="{StaticResource Text}"/>
    <TextBox Text="{Binding ServiceName}" Grid.Column="1" x:Name="ServiceNameTxt"/>
    </Grid>
    <Grid Margin="0 5">
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="150"/>
    <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <TextBlock Text="Время выполнения услуги: " Style="{StaticResource Text}" TextWrapping="Wrap"/>
    <TextBox Text="{Binding ServiceTime}" Grid.Column="1" x:Name="ServiceTimeTxt"/>
    </Grid>
    </Grid Margin="0 5">

```

```

<Grid.ColumnDefinitions>
<ColumnDefinition Width="150"/>
<ColumnDefinition Width="*/>
</Grid.ColumnDefinitions>
<TextBlock Text="Цена услуги: " Style="{StaticResource Text}"/>
<TextBox Text="{Binding ServicePrice}" Grid.Column="1" x:Name="ServicePriceTxt"/>
</Grid>
<Grid Margin="0 5">
<Grid.ColumnDefinitions>
<ColumnDefinition Width="150"/>
<ColumnDefinition Width="*/>
</Grid.ColumnDefinitions>
<TextBlock Text="Тип услуги: " Style="{StaticResource Text}"/>
<ComboBox x:Name="ComboType" SelectedItem="{Binding TypeService}" DisplayMemberPath="ServiceType"
Grid.Column="1"/>
</Grid>
<Grid Margin="0 5">
<Grid.ColumnDefinitions>
<ColumnDefinition Width="150"/>
<ColumnDefinition Width="*/>
</Grid.ColumnDefinitions>
<TextBlock Text="Рабочий: " Style="{StaticResource Text}"/>
<ComboBox x:Name="ComboWorker" SelectedItem="{Binding Worker}" DisplayMemberPath="FullName" Grid.Column="1"/>
</Grid>
<Grid Margin="0 5">
<Grid.ColumnDefinitions>
<ColumnDefinition Width="150"/>
<ColumnDefinition Width="*/>
</Grid.ColumnDefinitions>
<TextBlock Text="Изображение: " Style="{StaticResource Text}"/>
<Button x:Name="AddImage" Grid.Column="1" Click="AddImage_Click" Style="{StaticResource BtnExitStyle}"
Content="Выбрать изображение" />
</Grid>
<Button x:Name="SaveAddServiceBtn" Click="SaveAddServiceBtn_Click" Content="Сохранить" Style="{StaticResource
BtnExitStyle}" Width="300" Margin="10"/>
</StackPanel>
<StackPanel Grid.Column="1">
<TextBlock Text="Выбранное изображение: " Style="{StaticResource Text}" HorizontalAlignment="Center" Margin="0 5"/>
<Border Grid.Column="1" BorderBrush="{StaticResource BorderBackColor}" BorderThickness="1" Height="300" Width="300"
Margin="0 10">
<Image x:Name="ServiceImage" Source="{Binding ImageService, TargetNullValue={StaticResource NullImage}}"/>
</Border>
</StackPanel>
</Grid>
</Window>

```

AddEditPage.xaml.cs

```

private readonly BuildDbEntities _context = new BuildDbEntities();
private readonly Service _currentService = new Service();
private bool _editMode;
public AddEditPage(Service currentService, bool editMode = false, string titleText = "")
{
    InitializeComponent();
    _currentService = currentService;
    _editMode = editMode;
    DataContext = currentService;
    TitleText.Text = titleText;
    ComboType.ItemsSource = _context.TypeService.ToList();
    ComboWorker.ItemsSource = _context.Worker.ToList();
}
private void SaveAddServiceBtn_Click(object sender, RoutedEventArgs e)
{
    if (_editMode)
    {
        var initialService = _context.Service.FirstOrDefault(x => x.ID == _currentService.ID);
        if(initialService != null)
        {
            initialService.ServiceName = _currentService.ServiceName;
            initialService.ServicePrice = _currentService.ServicePrice;
            initialService.ServiceTime = _currentService.ServiceTime;
            initialService.TypeService = _currentService.TypeService;
            initialService.Worker = _currentService.Worker;
        }
    }
    else
    {
        _context.Service.Add(_currentService);
    }
    try
    {
        _context.SaveChanges();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Неудается добавить или редактировать услугу {ex.Message.ToString()}");
    }
    Close();
}
private void AddImage_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog()
    {
        Title = "Выберите фото",
    }
}

```

```

        Filter = "Файлы фотографий|.png;*.jpg",
        Multiselect = false,
    };
    fileDialog.ShowDialog();
    if (fileDialog.FileName == "")
        return;
    byte[] fileAsBytes = File.ReadAllBytes(fileDialog.FileName);
    _currentService.ImageService = fileAsBytes;
    ServiceImage.Source = new BitmapImage(new Uri(fileDialog.FileName));
    }
}

```

App.xaml

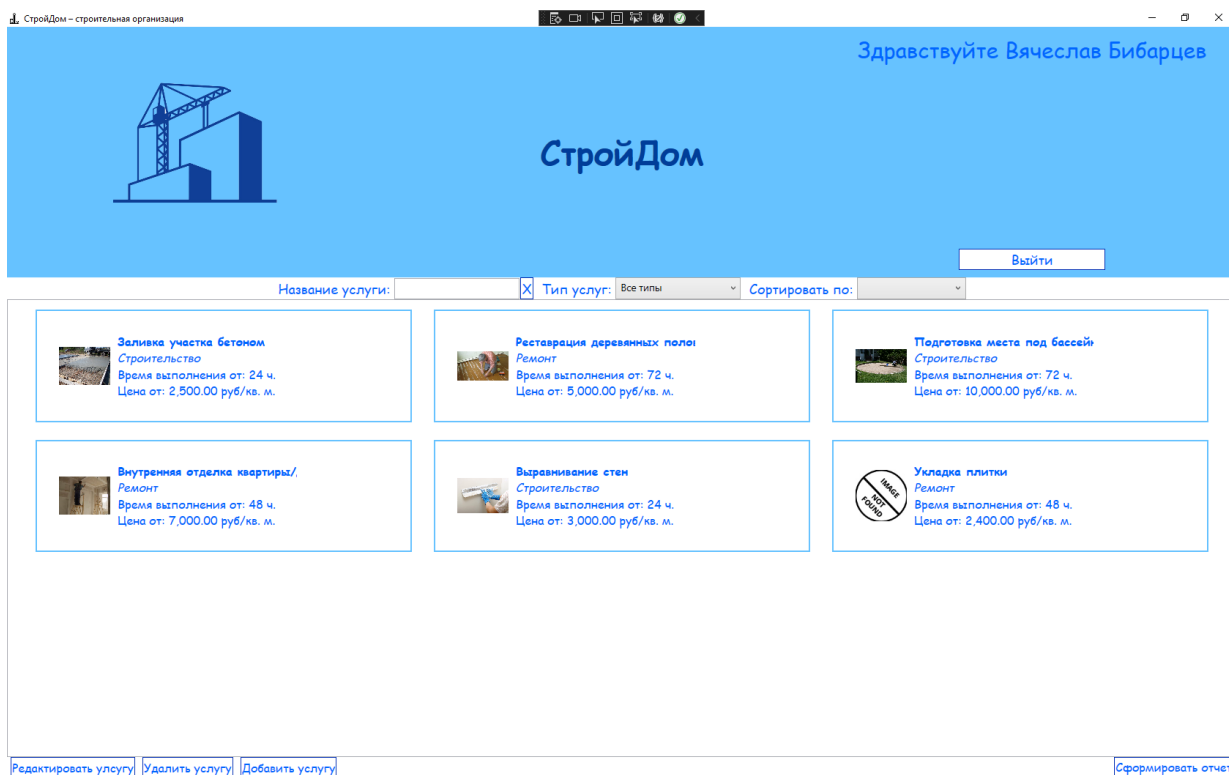
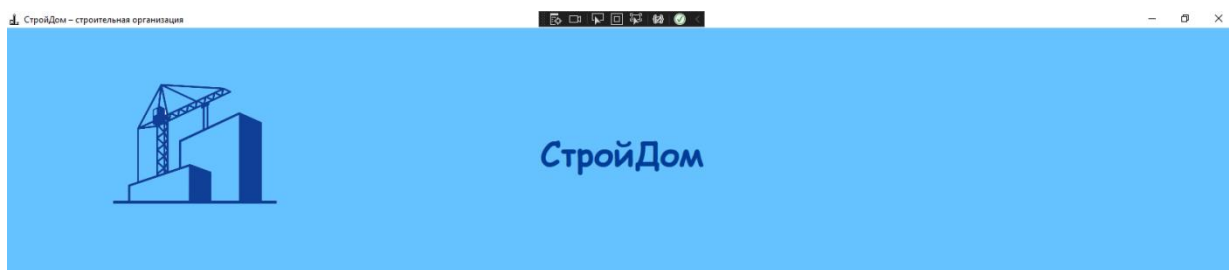
```

<Application x:Class="PDPBuild.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:PDPBuild"
    StartupUri="MainWindow.xaml">
    <Application.Resources>
        <BitmapImage x:Key="NullImage" UriSource="/Images/NullImage.jpg"/>
        <SolidColorBrush x:Key="BackColor" Color="#FFFF"/>
        <SolidColorBrush x:Key="BorderBackColor" Color="#66c2ff"/>
        <Style x:Key="MainText" TargetType="TextBlock">
            <Setter Property="Foreground" Value="#003d99"/>
            <Setter Property="FontSize" Value="26"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="FontFamily" Value="Comic Sans MS"/>
        </Style>
        <Style x:Key="Text" TargetType="TextBlock">
            <Setter Property="Foreground" Value="#0066ff"/>
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontFamily" Value="Comic Sans MS"/>
        </Style>
        <Style x:Key="BtnExitStyle" TargetType="Button">
            <Setter Property="Foreground" Value="#0066ff"/>
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontFamily" Value="Comic Sans MS"/>
            <Setter Property="Background" Value="#FFFFFF"/>
            <Setter Property="BorderBrush" Value="#002db3"/>
        </Style>
    </Application.Resources>
</Application>

```

Приложение Г

Экранные формы АИС



Окно добавления услуги

AddEditPage

Добавление

Название услуги:

Время выполнения услуги:


Цена услуги:

Тип услуги:

Рабочий:

Изображение:

Выбранное изображение:



Окно редактирования услуги

AddEditPage

Редактирование

Название услуги:

Время выполнения услуги:


Цена услуги:

Тип услуги:

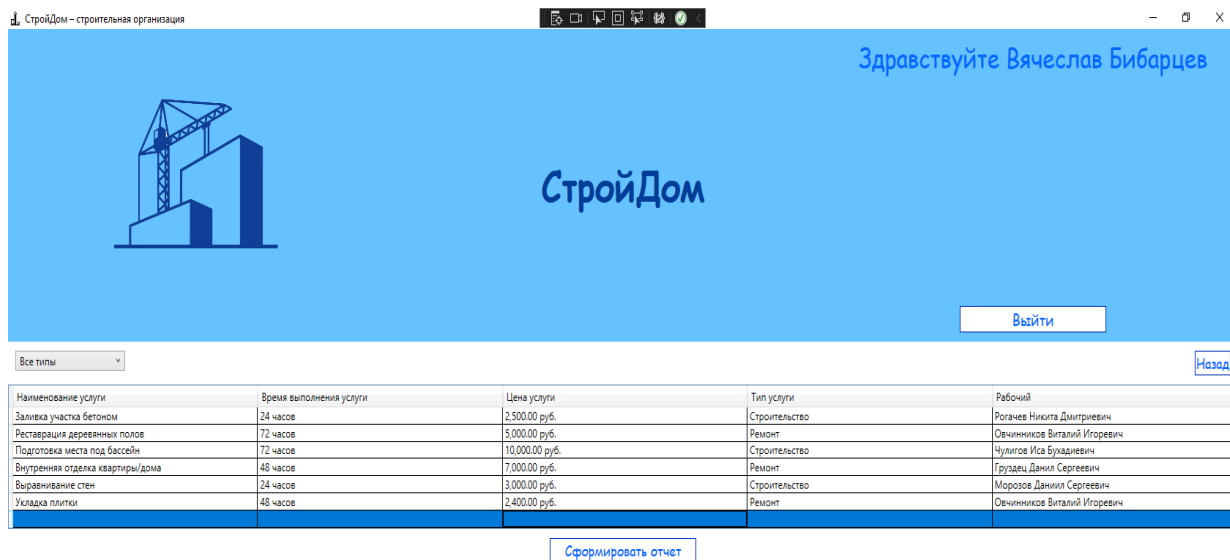
Рабочий:

Изображение:

Выбранное изображение:



Окно для формирования отчета



Окно печати отчета

