МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Южно-Уральский государственный университет» (национальный исследовательский университет) Высшая школа экономики и управления

Кафедра «Информационные технологии в экономике»

**Разработка «Справочно-аналитическая база атрибутивных данных информационных моделей объектов архитектурно-строительного проектирования» для ООО ПБСК «СМАЛЬТА»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА ЮУрГУ

– 09.03.03.2023.\_\_\_.ВКР

Руководитель, ст. преп.

/С. Ю. Нестеренко/

« » 2023 г.

Автор,

студент группы ЭУ- 411

/Л.В. Ефремов/

« » 2023 г.

Нормоконтролер,

//

« » 2023 г.

Челябинск, 2023

**АННОТАЦИЯ**

Ефремов Л. В. Разработка справочно-аналитического программного обеспечения для ООО ПБСК «СМАЛЬТА» – Челябинск: ЮУрГУ, ЭУ-411, 2023. – 00 стр., 00 рис., 00 табл., 00 листинг, библиограф. список – 00 наим., 00 прил.

Выпускная квалификационная работа

Оглавление

[ВВЕДЕНИЕ 4](#_Toc135176875)

[1 ПОСТАНОВКА ЗАДАЧИ 5](#_Toc135176876)

[1.1 Описание предметной области 5](#_Toc135176877)

[1.2 Описание задачи 6](#_Toc135176878)

[1.3 Анализ существующих систем 8](#_Toc135176879)

[Вывод по первому разделу 13](#_Toc135176880)

[2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 14](#_Toc135176881)

[2.1 Анализ средств разработки 14](#_Toc135176882)

[2.2 Построение DFD – диаграмм 20](#_Toc135176883)

[2.3 Таблицы базы данных 31](#_Toc135176884)

[Вывод по второму разделу 46](#_Toc135176885)

[3 РЕАЛИЗАЦИЯ ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ 47](#_Toc135176886)

[3.1 Реализация моделей и таблиц базы 47](#_Toc135176887)

[3.2 Основной пользовательский интерфейс 50](#_Toc135176888)

[3.3 Раздел аналитики 64](#_Toc135176889)

[Вывод по третьему разделу 67](#_Toc135176890)

[4 ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ПРОГРАММЫ 68](#_Toc135176891)

[4.1 Затраты компании до внедрения системы 68](#_Toc135176892)

[4.2 Расчет затрат на разработку системы 68](#_Toc135176893)

[4.3 Экономическая эффективность, затраты после внедрения 68](#_Toc135176894)

[Вывод по четвертому разделу 68](#_Toc135176895)

[ЗАКЛЮЧЕНИЕ 68](#_Toc135176896)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 69](#_Toc135176897)

# ВВЕДЕНИЕ

С каждым годом информационные технологии расширяются спектр своих г возможностей. Благодаря данному росту технической мощи появляется возможность не просто облегчать работу человеку, но и выполнять то, что без технологий было бы невозможно, а именно автоматизацию процессов на крупных производствах.

Разработчики стараются производить типовые решения, рассчитанные на применения в большем числе предприятий, расположенных по всей территории Российской Федерации. Ориентируясь на типовыми решения, компании забывают о необходимости разработки нетиповых программ, для конкретных организаций.

Рано или поздно, каждое узконаправленное производство должно проходить процесс автоматизации, который позволит ускорить и облегчить работу своих сотрудников. Узкая специализированность предприятия во многом влияет на процесс автоматизации и количество затраченного времени на это, так как в основном готовые типовые решение, предлагаемые рынком, в большинстве случаев не удовлетворяют потребности заказчика.

Главной проблемой узкоспециализованных компаний является необходимость в разработке уникальной, порой специфической, программы, удовлетворяющей потребности только данной фирмы. В такой ситуации стоимость разработки может быть несколько завышенной. С этой проблемой сталкиваются строительные бюро, чья основная работа проходить в сертифицированном программном комплексе, а все вспомогательные и второстепенные обязанности сотрудники вынуждены выполнять вручную из раза в раз. Именно этот процесс будет автоматизировать в ходе дипломной работы для одной из таких организаций.

# ПОСТАНОВКА ЗАДАЧИ

## Описание предметной области

Организация ООО Проектное Бюро Сейсмостойких Конструкций «СМАЛЬТА» (далее ООО ПБСК «СМАЛЬТА») является один из таких узкоспециализированных фирм и занимается архитектурно-строительным проектированием.

Основное направление работы компании ООО ПБСК «СМАЛЬТА» - это разработка проектов сейсмических конструкций, устойчивых к природным катаклизмам. Данная сфера деятельности довольно популярна в областях сейсмических воздействий, характеризующиеся:

* сотрясениями земной поверхности;
* потерей несущей способности грунта;
* волнами цунами.

Основная работа компании начинается, когда заказчик заключил договор на разработку проекта, обговорив свои пожелания и согласовав с компанией стоимость работ. С того момента главный инженер подготавливает документацию по проекту для специалистов по планировке.

Работа специалистов выполняется в несколько этапов:

1 этап. Архитекторы разрабатывают архитектурно-строительное решение. Результатом работы архитектора является планировочное решение.

2 этап. Получив решение, инженера-конструкторы осуществляют расчет несущих способностей каркаса, разрабатывают конструктивные решение и составляют расчетные обоснования по ним.

3 этап. Специалисты систем инженерного обеспечения (водоснабжение, водоотведение, электроснабжения, отопление и вентиляция) выполняют итоговые расчеты.

После прохождения 3-х этапов работы разработка документации считается завершенной и передается для дальнейшей работы.

Проблемой данного механизма, требующей решения, считаются нецентрализованные действия, которые сотрудники организации выполняют изо дня в день, а именно:

1. Хранение данных. Детализованная информация по каждому из разработанных проектов хранится в отдельном файле Word, а смета в файле Excel. Использование только этого метода хранения информации считается небезопасным, т. к. файл в процессе передачи от специалиста к специалисту может затеряться, а резервной копии не будет под рукой. В таком случае на восстановление может уйти достаточно времени.
2. Просмотр данных. Данные хранятся в разных файлах, поэтому главный инженер не в силах из одного места просмотреть информацию по различным по старым проектам и провести анализ.
3. Составление отчетности. Процесс осуществляется в ручном формате, т. к. нет централизованной базы, управляемой программным обеспечением, нет возможности автоматически генерировать файлы c характеристика (Word) и сметой (Excel) и также быстро восстанавливать при потере.

Отсутствие единого, пополняемого хранилища данных и приложения, позволяющего просматривать, вносить правки и мгновенно формировать новую отчетность, тормозит работу сотрудников ООО ПБСК «СМАЛЬТА».

Для автоматизации работы и решения данной проблемы ответственное лицо строительного бюро назначило задачу, требующую реализации.

## Описание задачи

Необходима разработать программное обеспечение прикладного назначения: «Справочно-аналитическая база атрибутивных данных информационных моделей объектов архитектурно-строительного проектирования», на основе проектов, реализованных ООО ПБСК «СМАЛЬТА».

Основные задачи:

* Систематизация атрибутивных свойств информационного массива данных этапа проектирования жизненного цикла проектов, реализованных «ПБСК СМАЛЬТА» объектов строительства, прошедших государственную экспертизу, апробированных при строительстве, эксплуатации и подтвердившие, требуемые функциональные проектные характеристики зданий и сооружений.
* Автоматизация трудоемкой аналитической работы, связанной с поиском информации показателей проекта в «статичных», стандартных архивах проектов, хранимых в виде разнородных документов на бумажных носителях и в виде цифровых документов в формате: \*.doc; \*.pdf; \*.dwg; \*.xlsx и т.д.

Реализация программного обеспечения позволит оптимизировать работу, с имеющейся справочной информацией по комплексу показателей, характеризующих каждый проект.

Вторым функциональным назначением, разрабатываемой программы, являются аналитические возможности, позволяющие выполнять статистические и прогнозные расчеты показателей при планировании нового проекта, на основе статистики имеющейся в информационной базе. Возможность пополнения информационной базы повысит качество прогнозируемых показателей.

В целом программа позволит исключить непродуктивные трудозатраты на «ручной» анализ архивной информации по старым проектам для выполнения подготовительных вычислений по энергопотреблению, по затратам тепла, воды, водоотведению и трудозатраты на разработку разделов из состава планируемого проекта.

По каждому объекту будет храниться следующая информацию:

1. Функциональные показатели (геометрические габариты, площади, объемы, материалы и т.д.);
2. Финансово-экономические показатели (в т.ч. сметы, калькуляции и т.д.);
3. Ресурсные показатели (материалы, оборудование, трудоемкость и т.д.);
4. Климатическая область применения объектов;
5. Инженерно-геологические условия объектов;
6. Особые условия (учет сейсмичности, геофизических явлений, транспортно-логистических условий, наличие или отсутствие аварийной устойчивости (живучести) объектов и д.р.);
7. Использованное программное обеспечение.

Пожелания заказчика по интерфейсу и функционалу программного обеспечения:

* Интересный, не загромождённый дизайн;
* Понятный и доступный интерфейс, с всплывающими подсказками по функционалу.

Цель создания пополняемой информационно-аналитической базы - повышение производительности инженерного и экономического анализа аналогов во время подготовки к разработке новых проектов.

Итоговый документ системы – это текстовый документ (.docx, .xlsx) в виде пояснительной записки, характеризующий проект.

## Анализ существующих систем

Ознакомившись с предметной областью и задачей, поставленной компанией, необходимо произвести анализ существующих аналог. Это необходимо для понимания, есть ли необходимость в разработке программного обеспечения. Рассмотрим системы, которым пользуется не одно производство.

**Надстройка FillDocuments для MS Excel**

Надстройка FillDocuments предназначена для заполнения большого количества документов данными из Excel по нажатию одной кнопки.

Для работы программы требуется таблица Excel с информацией для подстановки и папка с шаблонными документами. В качестве заготовок можно использовать файлы Word и Excel.

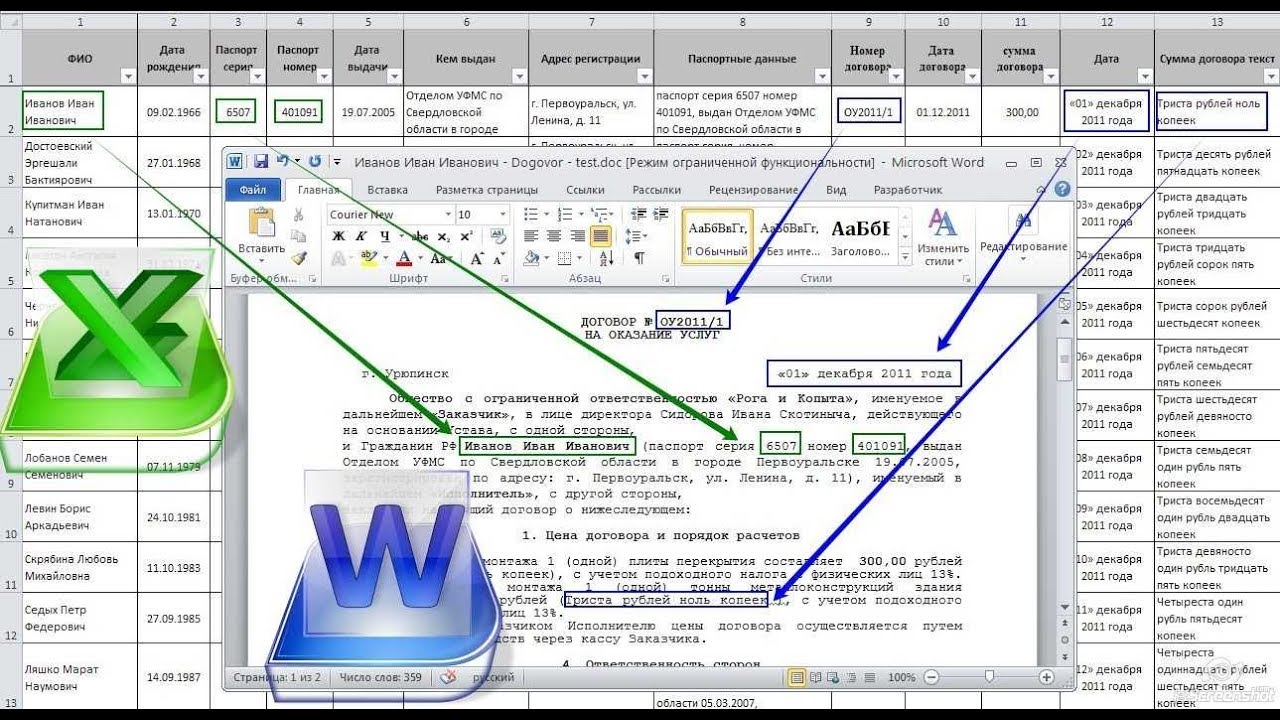
Результатом работы надстройки будет заполненный файл-шаблон исходного формата или PDF.

Особенности данной программы:

* Возможность настройки любых шаблонов, количество которых неограниченно. В качестве шаблонов могут быть файлы Word и Excel, а также текстовые документы;
* Возможность подстановки таблиц, картинок (фото, печати, подписи);
* Формирование красочных HTML-писем и рассылка почты;
* Дополнительной функцией по обработке данных считается склонение фамилий, дат, сумм прописью.

Недостатки надстройки:

* Ограниченный функционал;
* Отсутствие единой базы для хранения данных.

 Рисунок 1 – Пример заполнения документа при помощи надстройки FillDocuments

**Microsoft Dynamics NAV 2018**

Dynamics NAV 2018 – это интегрированная система управления предприятия для среднего и малого бизнесов. Она поставляется компанией Microsoft и объединяет в себе функции планирование ресурсов предприятия (ERP) и управления взаимоотношениями с клиентами(CRM).

Система была разработана для управления финансами, что позволяет анализировать слабые и сильные стороны компании, а также принимать важные управленческие решения. Система поддерживает частичную оплату, отмену по счетам и оценку задолженностей. Инструментарий соответствует всем правилам налогового и бухгалтерского учета. Программное обеспечение поддерживает разные политики производства (под заказ, на склад), а контроль за производственной мощностью способствует возможности распределения заданиями между рабочими филиалами. Особое место в системе занимает возможность составления планов и бюджетов для проектов, так как функционал создает идеальные условия для детального планирования задач.

Особенности данной программы:

* Наличие 42 локальных(адаптируемых) версии с поддержкой языка законодательных требований (в том числе и для России);
* Наличие сквозных процессов, благодаря которым каждая операция обновляет всю связанную с ней информацию во всех модулях;
* Возможность развернуть систему автономно, на облаке или на индивидуальном серверном оборудовании;
* Открытый исходный код бизнес-логики, поэтому появляется возможность расширить ее с помощью инструментов Visual Studio.

Недостатки системы:

* Широкий типовой функционал будет излишним для узкоспециализированных компаний;
* Не предназначен для крупных компаний;
* Сложный загроможденный интерфейс.

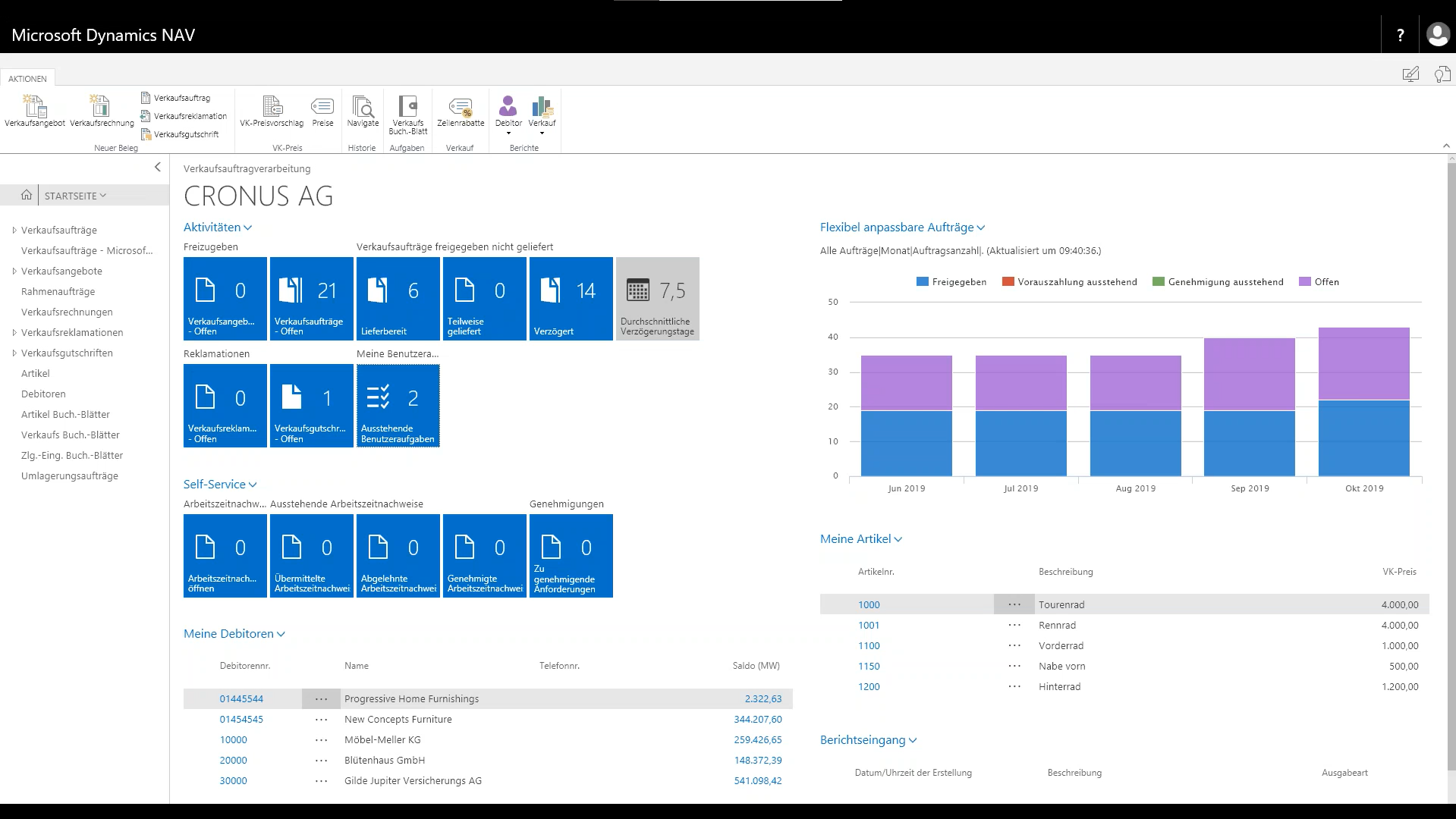
****

Рисунок - Главное окно Microsoft Dynamics NAV 2018

**1С: Документооборот**

1С: Документооборот – современное решение с широким набором инструментов для управления деловыми процессами, проектами и совместной работой сотрудников. Система поддерживает многопользовательскую работу как в локальной, так и онлайн сети.

Основные возможности системы подразделяют на категории: первая - учет, хранение документов (регистрация входящих и сходящих документов, учет персональных данных, поддержка бумажного делопроизводства, хранение документов любого вида, начиная от просто текста и заканчивая аудиофайлами и приложениями); вторая – работа с документами (просмотр и редактирование, формирование задач исполнителям, быстрый поиск, настраиваемая маршрутизация между разными видами проектов и возможность коллективной работы с одним и тем же файлом).

Достоинства:

* Автоматизация процессов, применимая ко всем функция компании;
* Система подходит для производственных и проектных организаций;
* Продуманная организация документооборота.

Недостатки:

* Сложный загроможденный интерфейс;
* Долгая адаптация сотрудников к функционалу при работе с системой;
* Дополнительные настройки и поддержание системы требует участие сотрудника 1С (дополнительные траты).

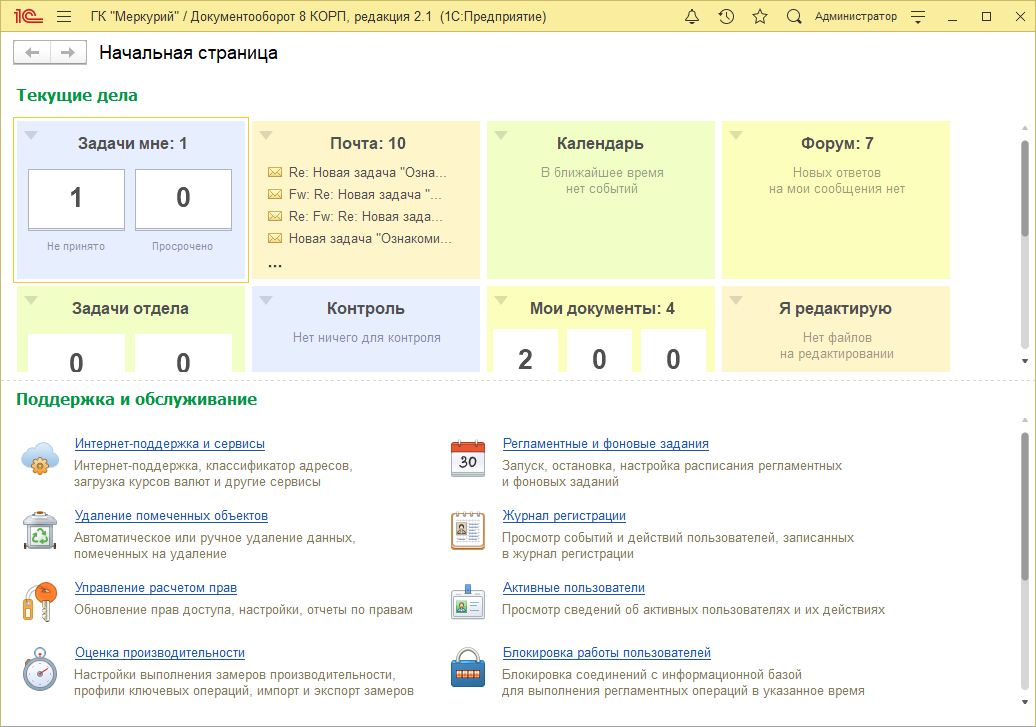


Рисунок 3 - Начальная страница 1С: Документооборот

**Сравнение решений**

Ознакомившись с существующими системам, проведем общий сводный анализ по основным критериям и выявим лучший способ реализации.

Таблица 1 – Сводная таблица по существующим системам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Надстройка FillDocuments для Excel | Microsoft Dynamics NAV 2018 | 1С: Документо-оборот | Наша система |
| Главная цель | Заполнения документов данными из Excel | Управление ресурсами и клиентами | Управление документами | Хранение данных по строи-тельным проектам |
| Управление проектами | – | + | + | + |
| Формирование отчетов в Word/Excel | +/– | + | + | +/+ |
| Специализирован на хранение данных о строительных проектах | –/+ | – | – | + |
| Аналитика данных | – | + | + | + |
| Автоматизированный ввод повторяющихся данных | – | + | + | + |
| Простой не загромождённый функционал | + | – | – | + |

Таблица наглядно отражает достоинства и недостатки рассмотренных систем, которые необходимо учитывать при выборе программного обеспечения для ООО ПБСК «СМАЛЬТА». В нашем случае можно считать, что желание разработать индивидуальное программное обеспечения для компании является обоснованным. В данной ситуации только индивидуальная разработка сможет закрыть потребности компании. При этом функциональность программы будет узкоспециализированой (ничего лишнего), так как готовые типовые решения имею лишние опции, за которые в конечном счете пришлось бы переплатить.

## Вывод по первому разделу

Результатами работы по первому разделу является ознакомление с предметной областью и задачей, выявление цели создания программного обеспечение и определение требований к хранимой информации. В ходе анализа существующих систем не было найдено решения, удовлетворяющего требованиям поставленной задачи, а настройка FillDocuments не автоматизирует работу сотрудников ООО ПБСК «СМАЛЬТА». Результат исследований подтвердил необходимость разработки единственной в своем роде программной системы для решения узконаправленных задач компании.

# ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

## Анализ средств разработки

Перед началом проектирования системы необходимо рассмотреть варианты реализации представленных требований. В связи с узконаправленной спецификой компании осуществить задачу, поставленную доверенным лицом ООО ПБСК «СМАЛЬТА», можно одним из следующих способов:

1. Приобрести надстройку для MS Excel;
2. Разработать конфигурацию на базе 1С, удовлетворяющую индивидуальные потребности компании;
3. Разработать программное обеспечение в среде Visual Studio на базе Windows Forms или Windows Presentation Foundation;
4. Разработать десктопное приложение на базе JavaScript-фреймворков.

**Конфигурация на базе платформы 1С: Предприятие 8**

«1С: Предприятие» - это универсальная платформа, на которой базируется любая конфигурация «1С». Данная платформа позволяет создавать и запустить систему «1С». Без нее будет невозможно работать в программе.

Созданные конфигурации могут подразделяться на типовые, отраслевые и специализированные (нетиповые), то есть выполненные на заказ.

Достоинствами типовых конфигураций:

* Универсальность. Возможность установки и использования в любом формате бизнеса, настроив под потребности компании;
* Автоматизация обновлений программы без помощи программиста;
* Синхронизация с сайтами и другими продуктами «1С»;
* Широкий арсенал функций;
* Ведение учета в различных сферах деятельности.

Недостатки типовых конфигураций:

* Отсутствие индивидуального комплекса инструментов;
* Перегруженность, обусловленная широким функционалом.

Нетиповые конфигурации в основном создаются на заказ для работы в узкоспециализированных компаниях. Главным и решающим отличием от типовых считается метод обновления версий: в нетиповых системах изменения следует вносит точечно. В данных программных обеспечений также есть свои плюсы и минусы.

Достоинства нетиповых конфигураций:

* Специализированность. Сотрудники, использующие этот тип системы, имеют возможность в процессе выполнения работы применять специально разработанные инструменты, «заточенные» на выполнение узкоспециализированных задач;
* Простота. Они не требуют сложных настроек и просты в использовании.
* Производительность. Работа выполняется намного быстрее, так как количество объектов и их взаимосвязей меньше.

Недостатки нетиповых конфигураций:

* Конфигурации не проходят специальные тестирования специалистами «1С»;
* Неавтоматизированный процесс обновления.

Основными достоинствами как типовых, так и нетиповых считаются:

* Индивидуальная настройка интерфейса форм под каждого сотрудника (управление видимостью и порядком колонок);
* Возможность хранения изображений и документов;
* Ускоренный массовый ввод благодаря возможности выбора данных из справочников;
* Настройки отчетности (отбор и сортировка), возможность отображения в виде таблиц и диаграмм (гистограмм).

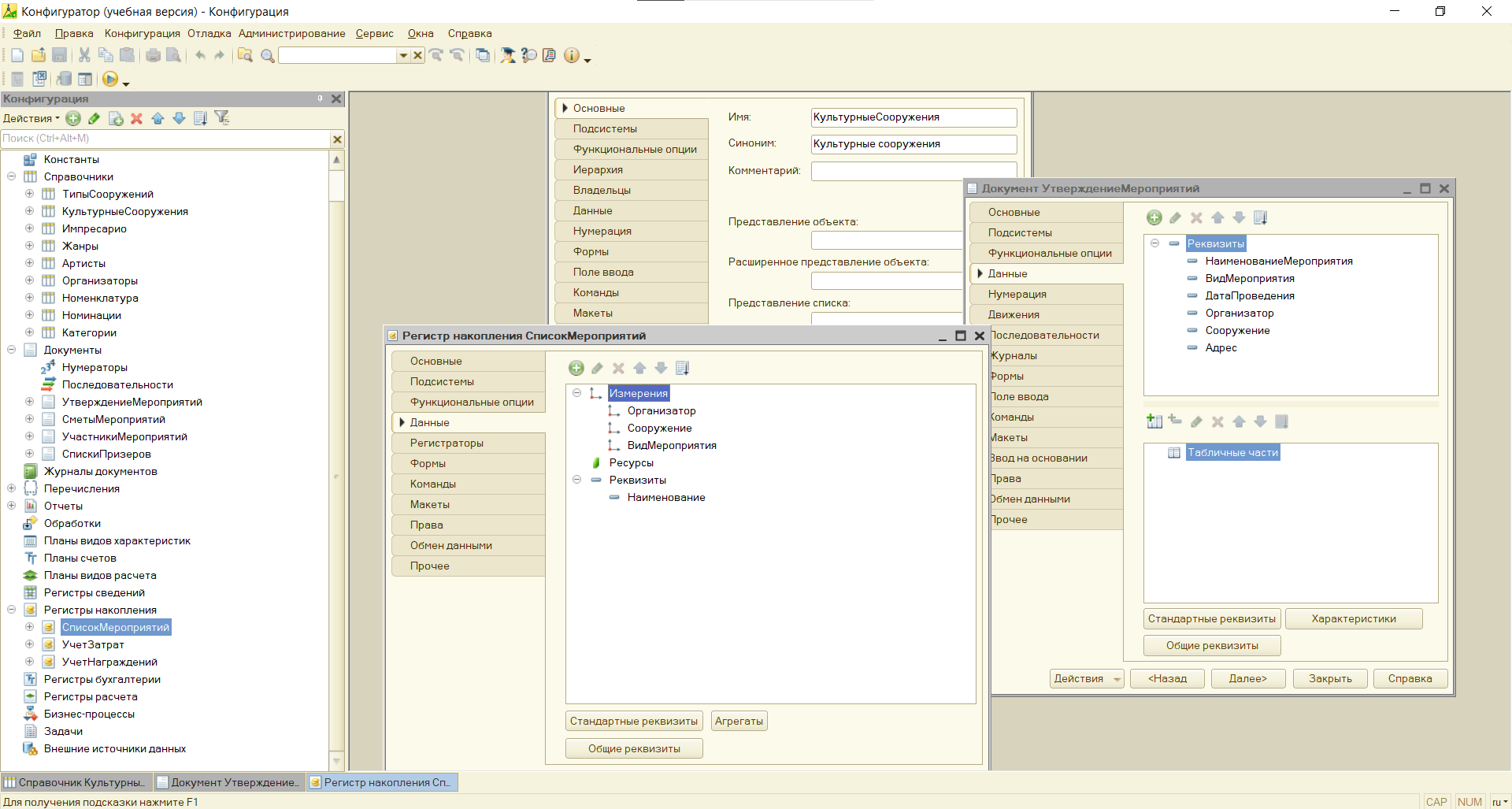


Рисунок 4 – Пример окна конфигурации 1С

**ПО на базе Windows Presentation Foundation**

Windows Presentation Foundation (далее WPF) – это аналог WinForms; система для построения клиентских приложений с визуально привлекательными возможностями взаимодействия с пользователем. В основе WPF лежит векторный механизм визуализации, способный использовать все преимущества современного графического оборудования.

WPF позволяет разрабатывать приложения при помощи языков разметки XAML (определение внешнего вида) и программирования С# (реализация поведения интерфейса). Разделение позволяет:

1. Уменьшить затраты на разработку;
2. Повышение эффективности разработки, так как дизайнеры и разработчики имеют возможность работать параллельно.

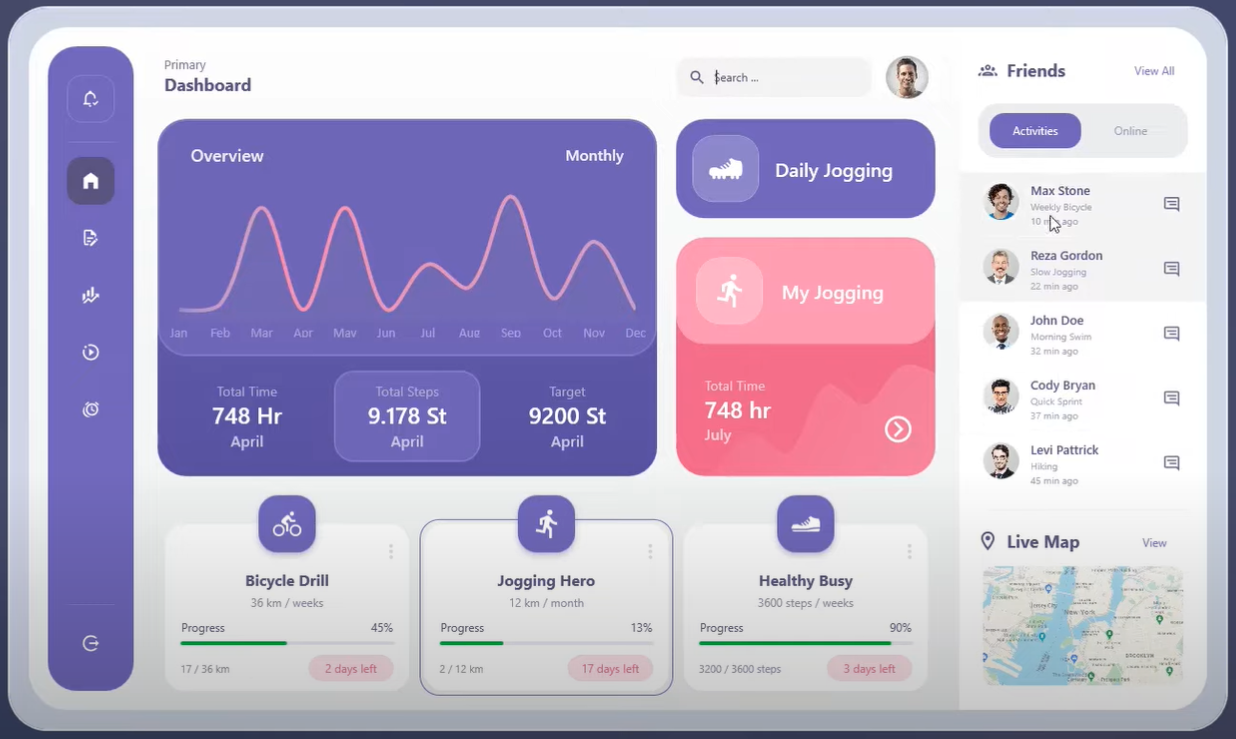


Рисунок 5 – Пример интерфейса созданного на языке XAML

Достоинства WPF:

* Возможность использования стилей и шаблонов для создания привлекательного интерфейса;
* Поддержка анимации, аудио и видео;
* Возможность отображения расширенного стилизованного текста;
* Наличие богатой модель рисования.

Недостатки WPF:

* Чувствительность к драйверам видеокарты;
* По сравнению с WinForms слабая поддержка дизайнера окон, в основном используется ручной вод XAML-кода;
* Неудачные названия многих классов.

**Десктопное приложение на базе Electron**

Electron – фреймворк с открытым исходным кодом. Для создания десктопных приложений с помощью Electron необходимо знать 3 веб-технологии: JavaScript, HTML и CSS.

Десктопное приложения, созданные на базе Electron, ведут себя как веб-приложения, но при этом могут читать и записывать данные в файловой системе компьютера.

Главным минусом разработки программы с помощью Electron – это большое потребление памяти. Данный недостаток может стать решающим, но не отменяет того факта, что на данный момент многие популярные приложения написаны на Electron (Skype, Discord, VSCode и т.д.).

Достоинства:

* Возможность переноса наработок веб-приложений на десктоп;
* Отработанная концепция создания: дизайнер-верстальщик-программист;
* Позволяет создать кроссплатформенное программное обеспечение;
* Наличие богатого набора встроенных и сторонних компонентов.

Недостатки:

* Для разработки необходимо знание 3-х языков: JavaScript, HTML и CSS;
* Большое потребление памяти (пустой проект занимает 100-200 Мб);



Рисунок 6 - Интерфейс Skype

**Сравнение решений**

Ознакомившись с возможными решениями поставленной задачи, проведем общий сводный анализ по основным критериям и выявим лучший способ реализации.

Таблица 2 – Сводная таблица по существующим решениям

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Конфигурация 1С (1С:Платформа) | WF/WPF(Visual Studio) | Система на базе Electron (JavaScript) |
| Способ | Инд. разработка | Инд. разработка | Инд. Разработка |
| Знание языков программирования | 1С | С#, XAML | JavaScript, HTML, CSS |
| Возможность создания отчетов в Word (Excel) | + (+) | + (+) | + (+) |
| Небольшое потребление памяти | + | + | - |
| Ускоренный массовый ввод | + | + | + |
| Аналитика | + | + | +/- |
| Оптимизация хранения данных | + | + | + |
| Уникальный пользовательский дизайн | - | + | + |

Таблица наглядно показывает всевозможные достоинства и недостатки выдвинутых решений. Сводные данные четко дают понять, что наилучшим способом реализации данной задачи является создание программы на базе Windows Forms/ Windows Presentation Foundation в среде Visual Studio.

Индивидуальная разработка системы позволит:

* Автоматизировать процессы заполнения данных разработчиками объята и передачу информации по проекту ответственному лицу для дальнейшего формирования общего отчета и отправки его заказчику;
* Оптимизировать систему хранения архивных данных (уход от бумажных носителей);
* Сэкономить время на формировании отчетов;
* Разработать программу с специализированным функционалом.

## Построение DFD – диаграмм

Проектирование приложение – это один из важнейших этапов в процессе разработки больших программных обеспечений.

Первое, что необходимо сделать во время моделирования системы, это построить диаграмму потоков данных (DFD). В этой методологии проводится анализ, описывающий взаимодействие внешних сущностей с разрабатываемой системой, и определяется логика функций (структурные подразделения программы), потоки информации (входных в выходные и наоборот) и хранилище данных.

Модель по методологии DFD разработана с помощью приложения BPwin, позволяющий произвести действия, описанные ранее.

**Контекстная диаграмма**

Контекстная диаграмма (верхний уровень DFD диаграммы) отображает взаимодействие системы с внешними сущностями, определяет границы и потоки входных-выходных данных.

На рисунке 00 представлена контекстная диаграмма, разрабатываемого приложения. На ней отображен 1 функциональный блок – Информационно-аналитическое программное обеспечение, 2 внешних сущности, с которыми взаимодействует система, а именно:

* с главным инженером, который вносит информацию с программное обеспечение, работает с ней и производит анализ существующих данных при проектировании новых заказов;
* с заказчиком посредством передачи отчетных материалом полученных из программы в виде документа Word и Excel.



Информационно-аналитическое программное обеспечение

Рисунок 7 - Контекстная диаграмма приложения

В таблице 00 приставлены потоки данных, определяющие информацию, передаваемую пользователем в программу и наоборот.

Таблица 3 – Потоки данных контекстной диаграммы

| Наименование | Определение | Заметки |
| --- | --- | --- |
| Исходные данные по объекту | Передаются новые исходные данные по объекту пользователем | Данные записей таблицы "Проекты":  Наименование - text; фото - blob; город - int (Таблица "Города"); год - datetime; разработчик - int (Таблица "Разработчики"); роль - int; заказчик - int (Таблица "Заказчики"); № договора - text; дата заключения – datetime; документация - text;  Данные записей таблицы "Клим-ские особенности":  Климатический район – int, подрайон - int;  температура ХП - int, температура ХС - int; снеговой район - int (Таблица "Снеговых районов"); ветровой район - int (Таблица "Ветровые районы"); уровень ответственности - int (Таблица "Ответственности"); карта сейсмичности - text; сейсмичность по проекту - int; cсейсмичность по тесту - int. |
| Перечень исходных данных по проекту | Передаются существующие исходные данные по объекту пользователю |
| Данные сметы по проделанной работе | Передаются новые пункты сметы по проделанной работе пользователем | Данные записей таблицы "Сметы":  Проект - int (Таблица "Проекты"); марка раздела – int (Таблица «Марки»); наименование - text; трудоемкость - int; выплата - double. |
| Смета по проекту | Передаются существующие пункты сметы по проделанной работе пользователю |
| Характеристики здания по проекту | Передаются новые характеристики здания по проекту пользователем | Данные записей таблицы "Характеристики здания":  Проект - int (Таблица "Проекты"); категория долговечности - int (Таблица "Ответственности"); S застройки - double; общая S помещений в здании - double; полезная S помещений в здании - double; строительный объем здания - double; число наземных этажей - int; число цокольных этажей - int; общая высота здания - double  Данные записей таблицы "Площади помещений":  Проект - int (Таблица "Проекты"); S магазинов - double; S офисов - double; S жилая - double; S прочих помещений - double; полезная S магазинов - double; полезная S офисов - double; полезная S жилая - double; полезная S прочих помещений - double;  Данные записей таблицы "Строительные объемы здания" для объема здания:  Проект - int (Таблица "Проекты"); объем наземной части - double; объем подвала - double; объем входной части - double;  Данные записей таблицы "Высоты зданий" для общей высоты здания:  Проект - int (Таблица "Проекты); кол-во типовых этажей - int, кол-во подземных этажей - int; высота 1 этажа - double; высота 2 этажа - double; высота типового этажа - double; высота подвала - double; высота технического этажа - double; высота нулевого этажа - double. |
| Перечень характеристик здания по проекту | Передаются существующие характеристики здания по проекту пользователю |
| Функциональное назначение здания | Передача новых данных по назначениям здания пользователем | Данные записей таблицы "Функциональные назначение":  Проект - int (Таблица "Проекты"); наименование - int (Таблица «Типы этажей»); назначение - int (Таблица "Назначения). |
| Перечень функциональных назначений здания | Передача существующих данных по назначениям здания пользователю |
| Показатели земельного участка | Передаются новые данные по земельному участку пользователем | Данные записей таблицы "Земельный участок":  Проект - int (Таблица "Проекты"); S участка - double; S асфальтобетонного покрытия - double; S озеленения - double. |
| Перечень показателей земельного участка | Передача существующие данные по земельному участку пользователю |
| Показатели ограждающих конструкций | Передаются новые данные по показателям огр. конструкциям пользователем | Данные записей таблицы "Ограждающие конструкции":  Проект - int (Таблица "Проекты"); СТ внешних стен - double; СТ покрытия - double; СТ окон - double.  СТ- сопротивление теплопередач |
| Перечень показателей ограждающих конструкций | Передаются данные по показателям огр. конструкциям пользователю |
| Данные об огнестойкости несущих конструкций | Передаются новые данные по огнестойкости несущих конструкций пользователем | Данные записей таблицы "Огнестойкости несущих конструкций":  Проект - int (Таблица "Проекты);  сечение 600х600 - int; сечение 500х500 - int; сечение 400х400 - int; ригели - int; стены - int; перекрытия - int; лестничные марши - int; перегородки - int; вертикальные связи - int. |
| Информация об огнестойкости несущих конструкций | Передаются существующие данные по огнестойкости несущих конструкций пользователю |
| Данные по конструктивному решению | Передаются новые данные по конструктивному решению пользователем | Данные записей таблицы "Строительная схема":  Проект - int (Таблица "Проекты); форма - text; cложность - text; длина в осях - double; ширина в осях - double; тип здания - int (Таблица «Типы зданий»); поперечная схема - int (Таблица «Типы конструкций»); кол-во поперечных пролетов - int; шаги поперечных пролетов - text; продольная схема - int (Таблица «Типы конструкций»); кол-во продольных пролетов - int; шаги продольных пролетов - text; вертикальная устойчивость - int (Таблица «Типы устойчивости»); горизонтальная устойчивость - int (Таблица «Типы устойчивости»); поперечная устойчивость - int (Таблица «Типы устойчивости»); продольная устойчивость - int (Таблица «Типы устойчивости»).  Данные записей таблицы "Конструктивное решение ниже отметки 0.00":  Проект - int (Таблица "Проекты); фундамент - int (Таблица «Типы фундамента»); технология СМР - int (Таблица «Типы технологий»); плита - int (Таблица «Типы конструкций»); толщина плиты - double; плита с надбалкой - int (Таблица «Типы конструкций»); толщина плиты с надбалкой - double; подколонники - int (Таблица «Типы конструкций»); размерность подколонников - text; колонны - int (Таблица «Типы конструкций»); размерность колонн - text; стены - int (Таблица «Типы конструкций»); толщина стен - double; перекрытия - int (Таблица «Типы конструкций»); толщина перекрытий - double; ригели - int (Таблица «Типы конструкций»); толщина ригель - double; бетон класса B - int; бетон класса F - int; бетон класса W - int; марка стали - text.  Данные записей таблицы "Конструктивное решение выше отметки 0.00":  Проект - int (Таблица "Проекты); несущие конструкции - int (Таблица «Типы конструкций»); колонны - int (Таблица «Типы конструкций»); перекрытия - int (Таблица «Типы конструкций»); ригели - int (Таблица «Типы конструкций»); стены - int (Таблица «Типы конструкций»); бетон класса B - int; бетон класса F - int; бетон класса W - int; марка стали - text; лестница - int (Таблица «Типы лестниц»); косоуры - text; шахта лифта - int (Таблица «Типы шахт»); толщина лифта - double; крыша - int (Таблица «Типы крыш»); водосток - text; кровля - int (Таблица «Типы кровли»); материал кровли - int (Таблица «Типы материалов кровли»); высота парапета - double; примечания по парапету - text; бетон класса B - int; бетон класса F - int; бетон класса W - int; »); толщина бетона - double; подготовка - text; толщина подготовки - double; вент шахта - int (Таблица «Типы вент шахт»); наружная отделка фасада - int (Таблица «Типы отделки»); утеплитель - int (Таблица «Типы утеплителя»); толщина утеплителя - double; плотность утеплителя - text; |
| Информация по конструктивному решению | Передаются существующие данные по конструктивному решению пользователю |
| Расчетные обоснования конструктивного решения | Передаются новые данные по расчетным обоснованиям конструктивного решения пользователем | Данные записей таблицы "Основные данные обоснования":  Проект - int (Таблица "Проекты"); РДМ модель - blob; диаграмма - blob; программный продукт – int (Таблица "Типы ПО");  Данные записей таблицы "ИГЭ выработки":  Проект - int (Таблица "Проекты"); отчет по выработкам - text; год выполнения - int; организация- text;  Данные записей таблицы "Марки ИГЭ":  Выработка - int (Таблица "ИГЭ выработки"), наименование - text; мощность - double; удельный вес - double; модуль деформации - double; удельно сцепление - double; угол внутреннего трения - double; показатель текучести - double; примечания - text.  Данные записей таблицы "Колебания":  Проект - int (Таблица "Проекты"); кол-во форм - int; W - double; F - double; T - double; |
| Информация по расчетным обоснованиям конструктивного решения | Передаются существующие данные по расчетным обоснованиям конструктивного решения пользователю |
| Показатели систем инженерного обеспечения | Передаются новые данные по показателям систем инженерного обеспечения пользователем | Данные записей таблицы "Водоснабжение":  Проект - int (Таблица "Проекты); водоканал - text; договор - text; улица - text; диаметр врезки – int; холодная м3/сут - double; холодная м3/час - double; холодная л/сек - double; холодная пожаротушение - double; горячая м3/сут - double; горячая м3/час - double; горячая л/сек - double;  Данные записей таблицы "Энергопотребление":  Проект - int (Таблица "Проекты"); энерг-ние магазинов - double; энерг-ние офисов - double; энерг-ние квартир - double; энерг-ние прочих помещений - double;  Данные записей таблицы "Тепловые нагрузки":  Проект - int (Таблица "Проекты"); тепло-ние магазинов - double; тепло-ние офисов - double; тепло-ние квартир - double; тепло-ние прочих помещений - double; |
| Информация по системам инженерного обеспечения | Передаются существующие данные по показателям систем инженерного обеспечения пользователю |

**Диаграммы декомпозиции**

Диаграмма декомпозиции представлять 2 уровень диаграммы потоков и показывает более детальную структуры, разрабатываемой системы. На данном уровне отображаются основные разделы программы и как между ними распределяются входные-выходные данные, упомянутые ранее.

Потоки информации были рассмотрены во время представления диаграммы композиции, далее набор входной-выходной информации изменен не будет, поэтому далее в таблице 00 будет представлено описание функциональных блоков программы.

Таблица 4 – Функциональные блоки диаграммы декомпозиции

| Наименование блока | Определение |
| --- | --- |
| Раздел «Исходные данные» | Отвечает за хранение, редактирование и вывод основной информации по объекту и климатическим особенностям местности по отдельно взятому проекту. |
| Раздел «Характеристики здания по проекту» | Отвечает за хранение, редактирование и вывод информации по зданию (а именно габариты, высота, объем, площадь здания и функциональные назначения этажей) по отдельно взятому проекту. |
| Раздел «Земля и ограждающие конструкции» | Отвечает за хранение, редактирование и вывод данных по земельному участку и ограждающим конструкция по отдельно взятому проекту. |
| Раздел «Конструктивное решение» | Отвечает за хранение, редактирование и вывод данных конструктивному решению (а именно объемно-планировочное решение, характеристика конструктивной схемы и конструкция здания выше(ниже) отметки 0.00) по отдельно взятому проекту. |
| Раздел «Обоснование конструктивного решения» | Отвечает за хранение, редактирование и вывод данных по инженерно-геологическим выработкам основания фундамента, колебаниям форм здания (включая эпюры и рдм-проекты) по отдельно взятому проекту. |
| Раздел «Инженерные системы» | Отвечает за хранение, редактирование и вывод данных по инженерным системам (а именно водоснабжение, энергоснабжение, теплоснабжение) по отдельно взятому проекту. |
| Раздел «Смета по проекту» | Отвечает за хранение, редактирование и вывод данных по затрате трудовых ресурсов на осуществление пунктов сметы отдельно взятому проекту |
| Раздел «Аналитика» | Отвечает за анализ продолжительности проектов, отслеживание динамики выполненных заказов (по годам, документация), расчет прибыли по годам, выявление максимального дохода (год, кол-во проектов, сумма). |



Информационно-аналитическое программное обеспечение

Рисунок 8 - Диаграмма декомпозиции

Далее рассмотрим пару примеров 3 уровня. На них отображаются технические возможности разделов программы с помощью функциональных блоков, которые олицетворяют команды, предоставляющие родительским модулем, т. е. соответствующим разделом.

На рисунке 00 показан 3 уровень блока «Исходные данные». Данный раздел это целое окно, информация на котором посвящена основным данным по договору и климатическим показателям местности. На диаграмме показано, что по запросу пользователя поток данных из базы будет передаваться на просмотр, а при добавлении или редактировании программа сохранит текущие изменения. Также страница обрабатывает запрос на вывод информации по всем разделам, относящейся к рассматриваемому проекту, в Word.



Раздел «Исходные данные по проекту»

Рисунок 9 - Диаграмма блока "Исходные данные по проекту"

Ознакомимся еще с одним примером реализации 3 уровня. На рисунке 00 отображен модуль «Инженерные системы». После рассмотрения потоков информации, служащих входными/выходными данными для функциональных блоков, можно сделать вывод, что в представленном разделе программного обеспечения размещены данные по водоснабжению, энергоснабжению и теплоснабжению. По запросу пользователя информация трех подразделов одновременно (при наличии) отображаются на экране, а добавление и редактирование можно осуществить отдельно по одному из подразделов, так как они независимы друг от друга. После заполнения данными программа проверить введенную информацию на корректность, сформирует запрос на добавление/изменение и направит его в базу с последующим сохранением.



Раздел «Инженерные системы»

Рисунок 10 - Диаграмма блока "Инженерные системы"

В завершении рассмотрим рисунок 00 - детализацию небольшого модуля «Смета по проекту». На диаграмме размещены три функциональных блока: первый из осуществляет вывод списка сметы по текущему проекту; второй - обрабатывает изменение информации по отдельно взятому пункту сметы и формирует запрос на сохранение данные; третий – принимает запрос пользователя на формирование отчетности по представленной смете в Word. Таким образом можно сделать вывод, что в данном окне размещен список для просмотра/редактирования и пользователь поочередно может работать с каждой записью данного списка.



Раздел «Смета по проекту»

Рисунок 11 - Диаграмма блока "Смета по проекту"

## Таблицы базы данных

Главная цель создания справочного обеспечение – это хранение большого объема информации в структурированном виде с возможностью редактирования. Наличие внутренней структуры в базе данных обеспечивают таблицы, хранящие информацию в виде логически организованных строк и столбцов. Рассмотрим содержание всех таблиц (в сумме 44) базы SQLite, определяя к какому разделу они относятся.

**Таблицы для раздела «Исходные характеристики объекта»:**

**AllProjects** – таблица, отображающая список всех объектов по проектам, спроектированных компанией.

Таблица 5 – Структура таблицы AllProjects

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | Primary key, int | Ключевой столбец таблицы |
| Image | Blob | Изображение по проекту (фото, проект 3D-CAD или РМД fea, вид визуализации) |
| Name | Text | Наименование проекта |
| Year | Int | Год строительства |
| ID\_city | Foreign key, int | Код места расположения |
| ID\_developer | Foreign key, int | Код разработчика проекта |
| Role | Text | Роль разработчика в проекта |
| ID\_customer | Foreign key, int | Код заказчика |
| Treaty | Text | № или шифр договора |
| DateOfConclusion | Text | Дата заключения договора |
| Documentation | Text | Проектная документация |

**Cities** – таблица для хранения списка городов, структура отображена в таблице.

Таблица 6 – Структура таблицы Cities

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование города |

**Customers** – таблица, содержащая список заказчиков когда-либо сотрудничающих с компанией.

Таблица 7 – Структура таблицы Customers

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Abbreviation | Text | Сокращенное наименование компании-заказчика |
| Name | Text | Полное наименование компании- заказчика |

**Developers** – таблица, хранящая весь список компаний-разработчиков, так как в справочнике могут хранится данные по проектам других компаний.

Таблица 8 – Структура таблицы Developers

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Abbreviation | Text | Сокращенное наименование компании-разработчика |
| Name | Text | Полное наименование компании-разработчика |

**ClimaticConditions** – таблица, содержащая данные по климатическим условиям района застройки объекта.

Таблица 9 – Структура таблицы ClimaticConditions

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| Region | Text | Климатический район |
| Subregion | Text | Климатический подрайон |
| Temperature\_coldD | Int | Температура холодной пятидневки |
| Temperature\_coldW | Int | Температура наиболее холодных суток |
| ID\_SnowyRegionId | Foreign key, int | Код снегового района |
| ID\_WindRegion | Foreign key, int | Код ветрового района |
| ID\_Responsibility | Foreign key, int | Код уровня ответственности |
| Map | Text | Карта сейсмичности |
| SeismicityTest | Int | Сейсмичность по результатам микросейсморайонированию |
| SeismicityProject | Int | Сейсмичность по проекту |

**Responsibility** – таблица, хранящая данные по уровням ответственности

Таблица 10 – Структура таблицы Responsibility

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Level | Text | Уровень ответственности |
| Class | Text | Класс сооружения |
| Coefficient | Double | Коэффициент надежности |

**SnowyRegions** – таблица содержит данные по нормативным значениям веса снегового покрова в соответствии с районом.

Таблица 11 – Структура таблицы SnowyRegions

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Номер района |
| Value | Double | Числовое значение |

**WindRegions** – таблица содержит данные по нормативным ветровым нагрузкам в соответствии с районом.

Таблица 12 – Структура таблицы WindRegions

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Номер района |
| Value | Double | Числовое значение |

**Таблицы для раздела «Характеристики здания по проекту»:**

**Characteristics** – таблица, в которой размещены основные (не обобщаемые) данные по зданию.

Таблица 13 – Структура таблицы Characteristics

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| Category | Text | Категория долговечности |
| Lifetime | Text | Нормативный срок эксплуатации |
| FireResistance | Int | Огнестойкость здания |
| BuiltUpArea | Double | Площадь застройки |

**BuildingAreas** – таблица хранит характеристику общих и полезных площадей помещений.

Таблица 14 – Структура таблицы BuildingAreas

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ShopArea | Double | Площадь помещений под магазины |
| OfficesArea | Double | Площадь офисных помещений |
| ApartmentsArea | Double | Площадь жилых помещений |
| OtherArea | Double | Площадь прочих |
| EffecShopArea | Double | Полезная площадь помещений под магазины |
| EffecOfficesArea | Double | Полезная площадь офисных помещений |
| EffectApartmentsArea | Double | Полезная площадь жилых помещений |
| EffectOtherArea | Double | Полезная площадь прочих |

**BuildingVolumes**– таблица хранит информацию по строительному объему здания (сооружения).

Таблица 15 – Структура таблицы BuildingVolumes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| GroundPart | Double | Объем наземной части |
| Basement | Double | Объем цокольной части |
| EnrancePart | Double | Объем входной части |

**BuildingHeight** – таблица содержит данные о характеристиках высоты каждого типа этажа.

Таблица 16 – Структура таблицы BuildingHeight

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| CountGroundFloors | Int | Количество наземных этажей |
| CountBasements | Int | Количество цокольных этажей |
| FirstFloor | Double | Высота 1-го этажа |
| SecondFloor | Double | Высота 2-го этажа |
| TypicalFloor | Double | Высота типового этажа |
| GroundFloor | Double | Высота цокольного этажа |
| TechnicalFloor | Double | Высота технического этажа |
| NullFloor | Double | Высота уровня 0.00 от уровня земли |

**Таблицы для раздела «Функциональное назначение здания»:**

**FunctionalPurposes** – таблица предназначена для хранения информации по функциональным назначениям здания.

Таблица 17 – Структура таблицы FunctionalPurposes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ID\_Appointment | Foreign key, int | Код функционального назначения |
| ID\_FloorType | Foreign key, int | Код типа помещения |

**FloorTypes** – таблица содержит перечень типов помещений.

Таблица 18 – Структура таблицы FloorTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа помещения |

**Appointments** – таблица хранит список функциональных назначений.

Таблица 19 – Структура таблицы Appointments

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование функциональных назначений |

**Таблицы для раздела «Земля и ограждающие конструкции»:**

**LandAndFences** – таблица с характеристиками земельного участка и показатели энергетической эффективности ограждающих конструкций.

Таблица 20 – Структура таблицы LandAndFences

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| LandArea | Double | Площадь участка |
| AsphaltArea | Double | Площадь асфальта |
| Greenarea | Double | Площадь озеленения |
| WallsST | Double | СТ внешних стен |
| CoatingsST | Double | СТ покрытия |
| WindowsST | Double | СТ окон |

**FireResistanceOfStructures** – таблица, хранящая информацию по огнестойкости несущих конструкций.

Таблица 21 – Структура таблицы FireResistanceOfStructures

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| Section600 | Int | Сечение железобетонных колон несущего каркаса (600х600) |
| Section500 | Int | Сечение железобетонных колон несущего каркаса (500х500) |
| Section400 | Int | Сечение железобетонных колон несущего каркаса (400х400) |
| Beams | Int | Ригели (балки) несущего каркаса |
| Walls | Int | Стены наружные и внутренние |
| Coatings | Int | Междуэтажные перекрытия и покрытия |
| FlightsOfStairs | Int | Лестничные марши и площадки |
| Partitions | Int | Внутренние стены и перегородки |
| Connections | Int | Вертикальные связи |

**Таблицы для раздела «Конструктивные решения»:**

**BuildingSchemes** – таблица, хранящая характеристику конструктивной схемы (сооружения).

Таблица 22 – Структура таблицы BuildingSchemes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| Form | Text | Форма здания |
| Complexity | Text | Сложность здания |
| Width | Double | Ширина в осях здания в плане |
| Length | Double | Длина в осях здания в плане |
| ID\_ConstructiveType | Foreign key, int | Код конструктивного типа здания |
| ID\_LaterScheme | Foreign key, int | Код схеме (поперечной) |
| CountLateralSpan | Int | Количество пролетов по ширине |
| LateralSteps | Text | Шаги пролетов по ширине |
| ID\_LongitudinalScheme | Foreign key, int | Код схеме (продольной) |
| CountLongitudinalSpan | Int | Количество пролетов по длине |
| LongitudinalSteps | Text | Шаги пролетов по длине |
| ID\_VerticalStability | Foreign key, int | Код типа устойчивости (вертикальная устойчивость) |
| ID\_HorizontalStability | Foreign key, int | Код типа устойчивости (горизонтальная устойчивость) |
| ID\_LateralStability | Foreign key, int | Код типа устойчивости (поперечная устойчивость) |
| ID\_LongitudinalStability | Foreign key, int | Код типа устойчивости (продольная устойчивость) |

**BuildingFoundations** – таблица, хранящая характеристику конструкций здания (сооружения) ниже отметки 0.00.

Таблица 23 – Структура таблицы BuildingFoundations

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ID\_FoundationType | Foreign key, int | Код типа фундамента |
| ID\_TechnologyType | Foreign key, int | Код типа технологии СМР |
| ID\_Slab | Foreign key, int | Код типа конструкции для плиты |
| ThicknessSlab | Double | Толщина плиты |
| ID\_SlabTwo | Foreign key, int | Код типа конструкции для плиты с надбалкой |
| ThicknessSlabTwo | Double | Толщина плиты с надбалкой |
| ID\_UnderСolumns | Foreign key, int | Код типа конструкции для подколонников |
| DimensionUnderСolumns | Text | Размерность подколонников |
| ID\_Сolumns | Foreign key, int | Код типа конструкции для колонн |
| DimensionСolumns | Text | Размерность колонн |
| ID\_Walls | Foreign key, int | Код типа конструкции для стен |
| ThicknessWalls | Double | Толщина стен |
| ID\_Floors | Foreign key, int | Код типа конструкции для перекрытий |
| ThicknessFloors | Double | Толщина перекрытия |
| ID\_Сrossbars | Foreign key, int | Код типа конструкции для ригели |
| ThicknessСrossbars | Double | Толщина ригели |
| ConcreteB | Int | Бетон класса В |
| ConcreteF | Int | Бетон класса F |
| ConcreteW | Int | Бетон класса W |
| SteelGrade | Text | Марка стали |

**BuildingStructures** – таблица, хранящая характеристику конструкций здания (сооружения) выше отметки 0.00.

Таблица 24 – Структура таблицы BuildingStructures

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ID\_BearingStructures | Foreign key, int | Код типа конструкции несущих конструкций |
| ID\_Сolumns | Foreign key, int | Код типа конструкции для колонн |
| ID\_Floors | Foreign key, int | Код типа конструкции для перекрытий |
| ID\_Сrossbars | Foreign key, int | Код типа конструкции для ригели |
| ID\_Walls | Foreign key, int | Код типа конструкции для стен |
| ConcreteB | Int | Бетон класса В для нес. конст-ций |
| ConcreteF | Int | Бетон класса F для нес. конст-ций |
| ConcreteW | Int | Бетон класса W для нес. конст-ций |
| SttelGrade | Text | Марка стали |
| ID\_Stair | Foreign key, int | Код типа конструкции лестницы |
| Beam | Text | Косоуры |
| ID\_Lift | Foreign key, int | Код типа конструкции лифта |
| ThicknesLift | Double | Толщина шахты лифта |
| ID\_HouseTop | Foreign key, int | Код типа конструкции крыши |
| Gully | Text | Тип водостока |
| ID\_RoofType | Foreign key, int | Код типа конструкции кровли |
| ID\_RoofingMaterial | Foreign key, int | Код материала кровли |
| ParapetHeight | Double | Высота парапета |
| ParapetNote | Text | Примечание к парапету |
| ConcreteB\_2 | Int | Бетон класса В для входа |
| ConcreteF\_2 | Int | Бетон класса F для входа |
| ConcreteW\_2 | Int | Бетон класса W для входа |
| ThicknesConcrete | Double | Толщина бетона |
| Preparation | Text | Подготовка |
| ThicknesPreparation | Double | Толщина подготовки |
| ID\_AirShaftTtype | Foreign key, int | Код типа конструкции вент шахты |
| ID\_ExteriorFinish | Foreign key, int | Код типа наружной отделка фасада |
| ID\_InsulationMaterial | Foreign key, int | Код типа материала утеплителя |
| ThicknesInsulation | Double | Толщина утеплителя |
| Density | Text | Плотность утеплителя |

**ConstructiveTypes** – таблица хранит список конструктивных типов зданий.

Таблица 25 – Структура таблицы ConstructiveTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование конструктивного типа здания |

**SchemeTypes** – таблица хранит список типов конструктивных схем здания.

Таблица 26 – Структура таблицы SchemeTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа конструктивной схемы |

**StabilityTypes** – таблица хранит список типов конструкций для пространственной устойчивости.

Таблица 27 – Структура таблицы StabilityTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа пространственной устойчивости |

**FoundationTypes** – таблица хранит список типов фундамента.

Таблица 28 – Структура таблицы FoundationTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа фундамента |

**TechnologyTypes** – таблица хранит список типов фундамента.

Таблица 29 – Структура таблицы TechnologyTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа технологии СМР |

**ConstructionTypes** – таблица хранит список типов конструкций (стен, ригелей, колонн, перекраден).

Таблица 30 – Структура таблицы ConstructionTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа конструкции |

**StairTypes** – таблица хранит список типов конструкций лестницы.

Таблица 31 – Структура таблицы StairTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа конструкции лестницы |

**LiftTypes** – таблица хранит список типов конструкций шахты лифта.

Таблица 32 – Структура таблицы LiftTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа конструкции шахты лифта |

**HouseTopTypes** – таблица хранит список типов крыши.

Таблица 33 – Структура таблицы HouseTopTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа крыши |

**RoofTypes** – таблица хранит список типов кровли.

Таблица 34 – Структура таблицы RoofTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа кровли |

**RoofingMaterials** – таблица хранит список типов материалов для кровли.

Таблица 35 – Структура таблицы RoofingMaterials

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа материала |

**AirShaftTypes** – таблица хранит список типов вентиляционных шахт.

Таблица 36 – Структура таблицы AirShaftTypes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа вентиляционной шахты |

**ExteriorFinishes** – таблица хранит список типов материалов для внешней отделки фасада.

Таблица 37 – Структура таблицы ExteriorFinishes

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа материала для внешней отделки фасада |

**InsulationMaterials** – таблица хранит список типов материалов утеплителя.

Таблица 38 – Структура таблицы InsulationMaterials

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование типа материала утеплителя |

**Таблицы для раздела «Расчетные обоснования конструктивного решения»:**

**RationaleInfos** – таблица с основными данными по расчетному обоснованию конструктивного решения.

Таблица 39 – Структура таблицы RationaleInfos

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ImageOne | Blob | Fea-проект |
| ImageTwo | Blob | Эпюра деформации основания фундамента |
| ID\_Software | Foreign key, int | Код программного комплекса |

**Software** – таблица со списком программных комплексов.

Таблица 40 – Структура таблицы Software

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Name | Text | Наименование комплекса |
| Developer | Text | Разработчик |
| Certificate | Text | Сертификат |

**IgeWorkings** – таблица для основных сведений по ИГЭ выработкам.

Таблица 41 – Структура таблицы IgeWorkings

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| Report | Text | Номер отчета по выработкам |
| Year | Int | Год выполнения выработок |
| Organization | Text | Наименование организации |

**GeologicaWork** – таблица для сведений по инженерно-геологическим выработкам основания фундамента.

Таблица 42 – Структура таблицы GeologicaWork

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_IgeWorking | Foreign key, int | Код на отчет по выработке |
| Name | Text | Наименование ИГЭ |
| Power | Double | Мощность ИГЭ |
| Weight | Double | Объемный вес |
| Deformation | Double | Модуль деформации |
| Clutch | Double | Удельное сцепление |
| Friction | Double | Угол внутреннего трения |
| Fluidity | Double | Показатель текучести |
| Note | Text | Заметки |

**ProjectFluctuations** – таблица c информацией по динамическим характеристика собственных форм колебаний проекта здания.

Таблица 43 – Структура таблицы ProjectFluctuations

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| CountForms | Int | Количество учтенных форм |
| W | Double | W (рад/с) |
| F | Double | F (Гц) |
| T | Double | Т (с) |

**Таблицы для раздела «Инженерные системы»:**

**WaterParameters** – таблица хранит данные по системе водоснабжения и водоотведения.

Таблица 44 – Структура таблицы WaterParameters

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| Vodokanal | Text | Водоканал |
| Treaty | Text | Номер технических условий |
| Street | Text | Улица |
| Diameter | Int | Диаметр вырезки |
| Second\_cw | Double | Секундный расход холодной воды |
| Hour\_cw | Double | Часовой расход холодной воды |
| Day\_cw | Double | Суточный расход холодной воды |
| Firefighting\_cw | Double | Расход воды в целях пожаротушения |
| Second\_hw | Double | Секундный расход горячей воды |
| Hour\_hw | Double | Часовой расход горячей воды |
| Day\_hw | Double | Суточный расход горячей воды |

**ElectricityParameters** – таблица содержит информацию по энергопотреблению оборудования здания.

Таблица 45 – Структура таблицы ElectricityParameters

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ShopArea | Double | Торговые помещения |
| OfficesArea | Double | Административные помещения |
| ApartmentsArea | Double | Жилые помещения |
| OtherArea | Double | Прочие |

**HeatParameters** – таблица хранит данные по тепловым нагрузкам на отопление и вентиляцию здания.

Таблица 46 – Структура таблицы HeatParameters

| Название столбца | Тип данных | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ShopArea | Double | Торговые помещения |
| OfficesArea | Double | Административные помещения |
| ApartmentsArea | Double | Жилые помещения |
| OtherArea | Double | Прочие |

**Таблицы для раздела «Смета по проекту»:**

**Estimates** – предназначена для хранение разделов сметы по всем проектам.

Таблица 47 – Структура таблицы Estimates

| Название столбца | Тип данных столбца | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| ID\_project | Foreign key, int | Код объекта |
| ID\_chapter | Foreign key, int | Код перечня работ |
| Pay | Double | Выплата за работу |
| Labor\_costs | Int | Трудозатратность |

**Chapters** – хранит перечень работ, которые могли быть учтены в смете по проекту.

Таблица 48 – Структура таблицы Chapters

| Название столбца | Тип данных столбца | Описание |
| --- | --- | --- |
| ID | PrimaryKey, int | Ключевой столбец таблицы |
| Abbreviation | Text | Аббревиатура вида работ |
| Name | Text | Полное название вида работ |

## Вывод по второму разделу

Во втором разделе были рассмотрены системы для разработки программного обеспечение. Было решено, что система будет реализована в среде Visual Studio на базе Window Forms/Windows Presentation Foundation. Это позволит разработать более интересный пользовательский интерфейс.

Во второй части была построена DFD-диаграмма. В результате было сформировано четкое понимание взаимодействия системы с внешними сущностями, определены потоки данных, определены функциональные составляющие программного обеспечения.

В заключение была разработана база данных, содержащая 44 таблицы. Для работы с базой была выбрана компактная встраиваемая СУБД - SQLite.

# РЕАЛИЗАЦИЯ ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ

Реализация программного обеспечение преследуется методику написания кода, которая использует подход «Model First», представляющий один из принципов к работе с Entity Framework (технология для доступа к базам данных). Суть данного подхода в том, что в первую очередь разрабатываются модели программы, по которым строится база данных, далее проектируется интерфейсная часть. Завершающим этапом считается соединение интерфейса и моделей через прослойку (модель-представления), отвечающую за логику системы. Данное функциональное разделение на три составляющие является итогом использования паттерна MVVM (Model-View-ViewModel).

При работе с WPF в тандеме с применением паттерна MVVM появляется возможность использования механизма привязок (building) данных к элементам управления в коде XAML. Для применения данного инструмента модель-преставление должна реализовать интерфейс INotifyPropertyChanged, требующийся модулю привязки WPF. Он отвечает за оповещения системы в момент изменения свойств элементов на форме.

Второй важной настройкой для реализации паттерна MVVM являются команды, предоставляемые интерфейсом ICommand. Команды привязываются к элементу в коде XAML и определяют поведение системы в качестве отклика на действия пользователя.

## Реализация моделей и таблиц базы

Данный подраздел посвящен первому этапу подхода «Model First», а именно созданию моделей, по которым будет спроектирована база данных. Ранее, во время планирования, были разработаны таблицы, которые необходимо реализовать. На примере небольшой модели Estimate, представленного на рисунке 00, рассмотрим структуру класса, по которому будет строится соответствующая таблица (Estimates) в базе данных:

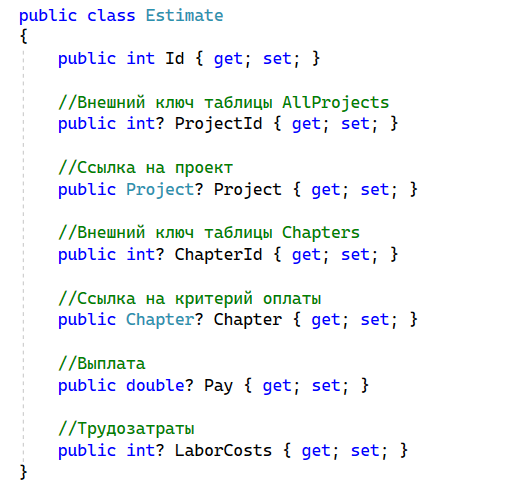


Рисунок 12 - Модель Estimate

Code-First видит навигационное свойство в коде модели и автоматически добавляет отношение между таблицами. Отношения определяются только с помощью свойств. Они могут указываться как в двух таблицах (двусторонняя связь), так и в одной (односторонняя).

При реализации модели использовалось одно из соглашений Code-First, а именно явное указание внешних ключей и ссылку на таблицу (public int? ProjectId {get; set}; public Project? Project {get; set};), но и без явного указания они будут генерироваться автоматически, так как используется соглашения при поиске внешнего ключа с свойствах модели. Данные правила основаны на имени свойства и следующих шаблонов:

[Имя первичного ключа внешней таблицы]

[Имя внешней таблицы] + [Имя первичного ключа внешней таблицы]

[Имя навигационного свойства] + [Имя первичного ключа внешней таблицы]

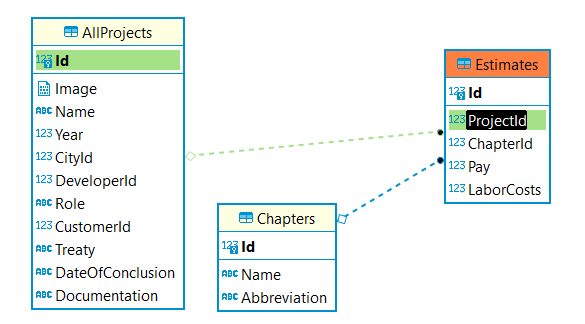


Рисунок 13 – Связи созданные с помощью соглашения Code-First

По данному принципу были созданы 44 модели, по которым, при обращении к классу производному от DbContext, строится база данных (при условии, если она не была построена ранее). DbContext является неотъемлемой частью Entity Framework, так как предоставляет сеанс с базой, используемый для запроса на получение информации и сохранения изменений сущностей.

Класс ApplicationContext включает в себя свойства DbSet<TEntity> для каждой сущности, а также переопределяет метод OnConfiguring, позволяющий выбрать и настроить источник данных, использующийся с контекстом. Время жизни данного класса начинается с создания экземпляра в файле логики (модель-представление) и заканчивается во время удаления экземпляра. Основными методами производного класса от DbContext считаются Add (добавление нового объекта), Enrty (обеспечивает доступ к информации), Find (находит сущность с заданным значением первичного ключа), Remove (устанавливает состояние удаления для указанного объекта), Update (устанавливает состояние изменения для указанного объекта), SaveChanges (выполняет сохранение объектов в состоянии добавляемый, изменяемый, удаляемый).

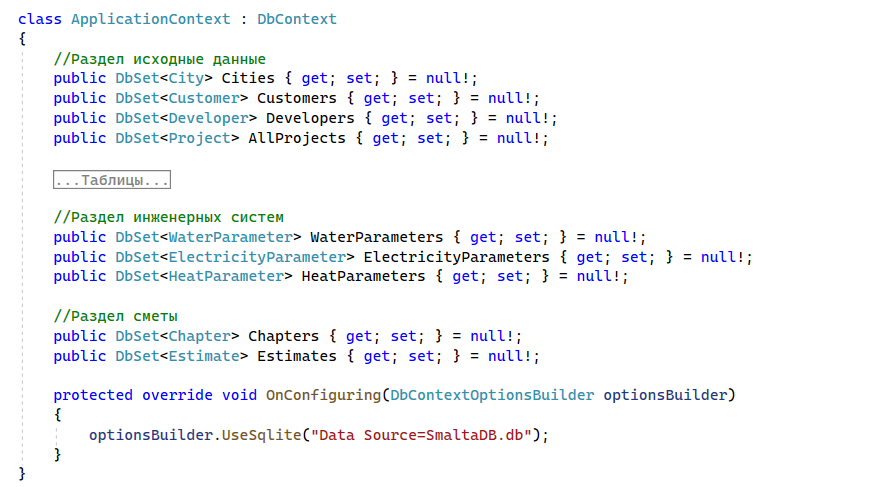


Рисунок 14 – Класс контекста ApplicationContext

## Основной пользовательский интерфейс

WPF считается аналогом WinForms, но главным отличием данной библиотеки считается то, что система нацелена на создание клиентских десктопных приложений с визуально привлекательным современным интерфейсом. Технология WPF предлагает разработчикам уже знакомые элементы управление (TextBox, TextBlock, Button и др.), а также позволяет создавать собственные стили для любого из них. Стилевое оформление элементов осуществляется через разметку XAML (расширенный язык разметки, основанный на XML).

Для данного проекта были созданы индивидуальные стили для button, checkbox, combobox, datagrid, scrollview, tabcontol, textblock, textbox, tooglebuton. Основными цветами программного обеспечения стали бежевый, коричневый и белый. На рисунке 00 изображено сравнение стандартных элементов и выполненных для текущего приложения.

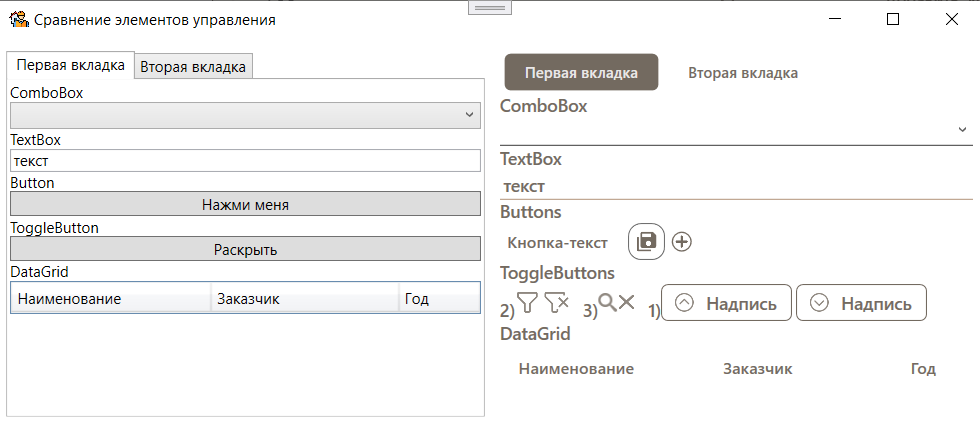


Рисунок 15 - Сравнение стилей элементов управления

Разберем интерфейс главной страницы приложения, показанной на рисунке 00. В начальном окне расположен общий список проектов, размещенных в справочной базе. При выборе одного из проектов в левой части страницы отображается основная информация (фото, город, заказчик, разработчик, договор и дата заключения). Над списком находится основной функционал окна:

1. Поле поиска по наименованию;
2. Поля для отбора записей. Можно произвести фильтрацию данных по одному из критериев (год, город, документация, заказчик);
3. Кнопки редактирования для добавления информации по новому проекту или редактирования уже имеющейся.
4. Переход к разделу аналитики данных.

Для интерфейса данного окна создан класс логики, в котором обозначены свойства полей, команды для кнопок и источник для списка. Примеры привязок отображены в листинге 00.

Листинг 1 – Пример использования механизма привязок (building)

**Код текстового поля в разметки XAML (файл AllProjects.xaml):**

<TextBox Style="{DynamicResource AllTextBox}"

Text="{Binding FilterText, Mode=TwoWay}"/>

**Свойство для текстового поля в классе логики (файл AllProjectsVM.cs):**

string filterText = string.Empty;

public string FilterTex{

get {return filterText ;}

set {filterText = value;

OnPropertyChanged ("FilterText") ;}

}

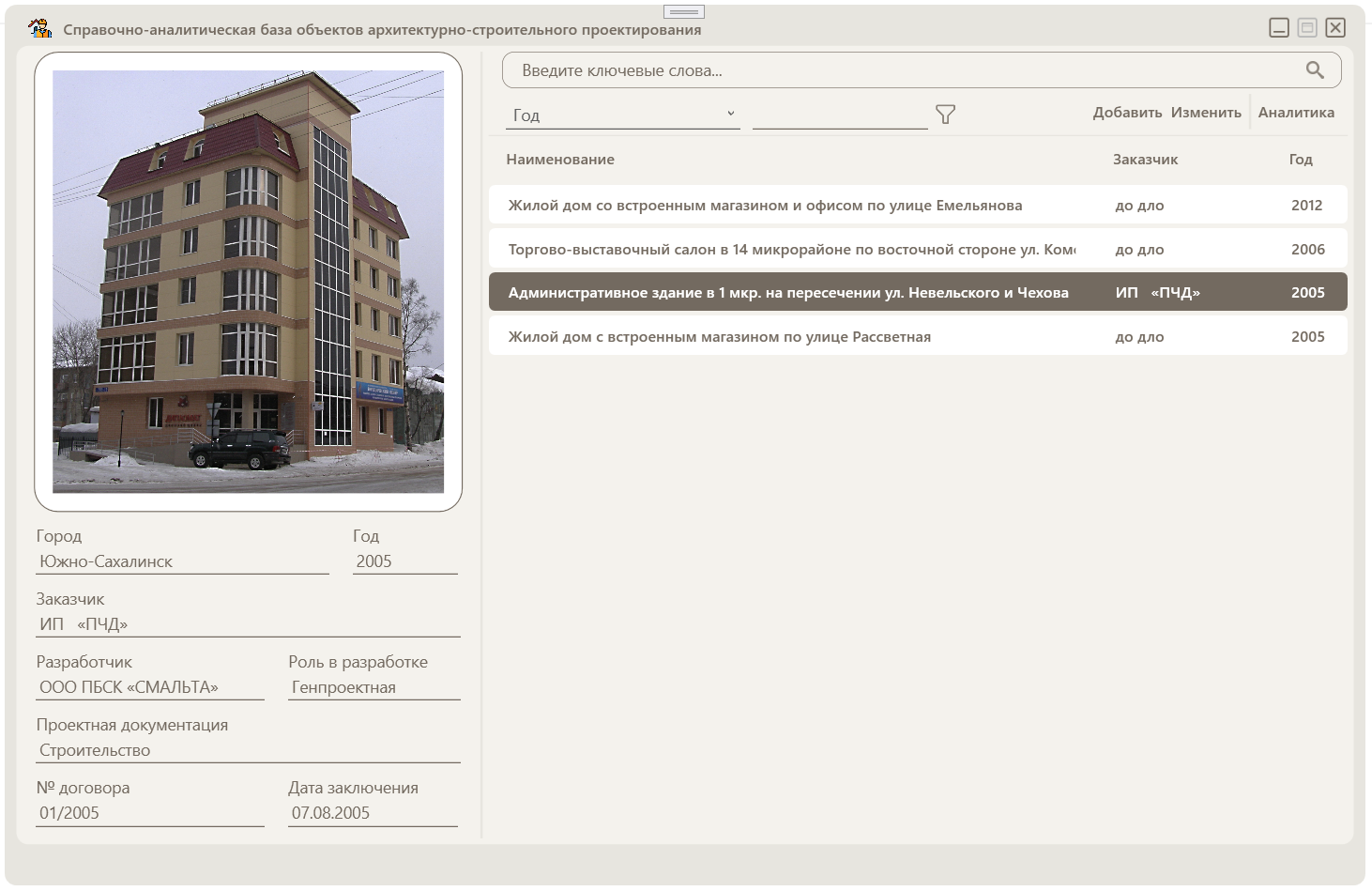
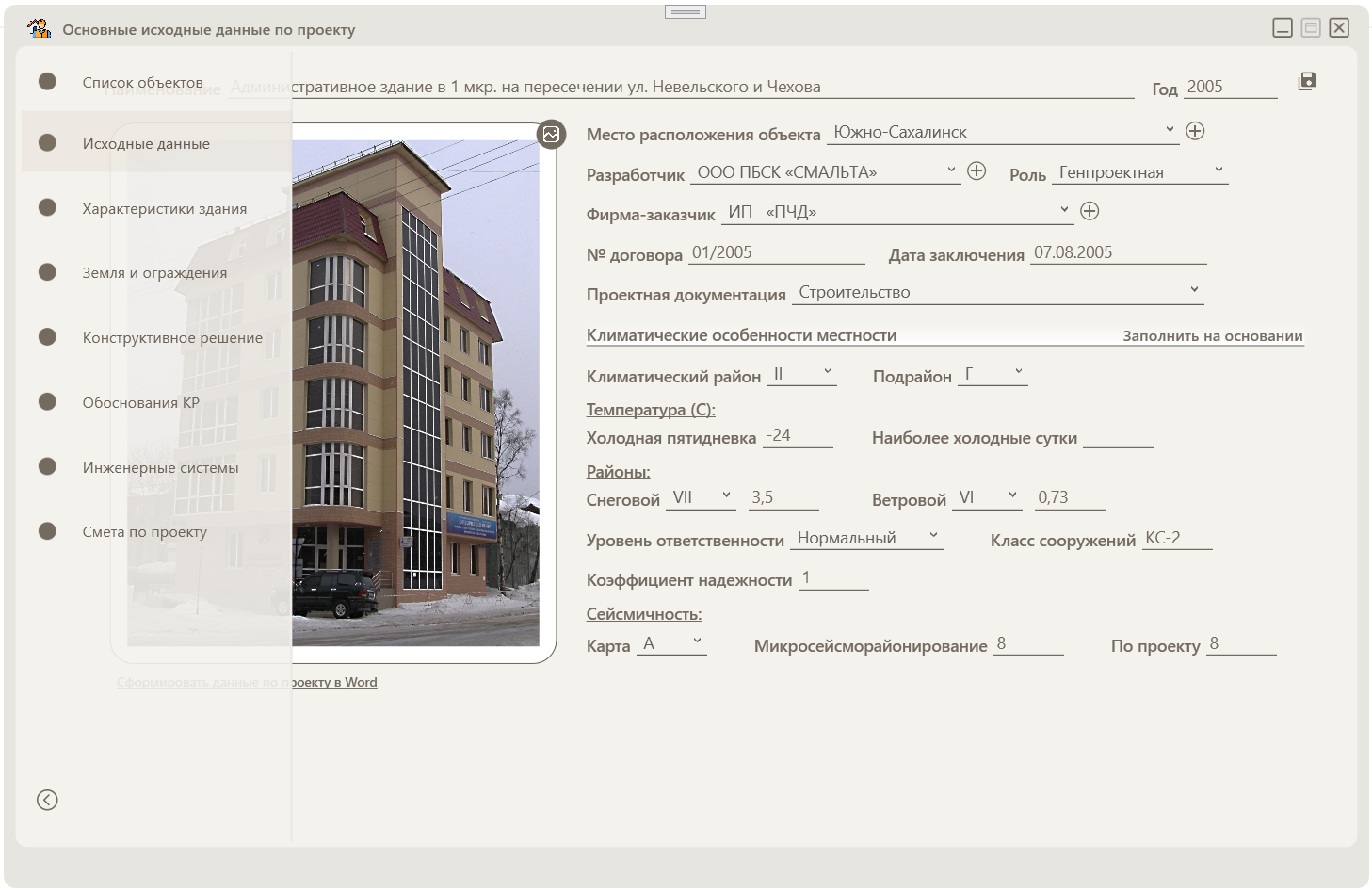


Рисунок 16 - Интерфейс главного окна

Все вкладки, которые будут рассмотрены далее, отображаются в элементе Frame, расположенном на странице. Принцип перехода по вкладкам заключается в том, что при выборе раздела в боковом меню вызывается команда, в котором происходит изменение содержимого в главной рамке (Frame). Данный метод отображение использует идею вложенности страниц, которую приветствует WPF.

При переходе в окно с подробной информацией о выбранном проекте пользователь попадает на страницу с основными исходными данными. Все сведения по проекту размещены в общем счете на 7 вкладках, с которыми можно ознакомится на рисунке 00. Данные дополнительные разделы недоступны для пользователя до того момента, пока он не заполнит и не сохранить исходные данные.

В разделе основных сведений кроме информации по исполнителю и исполняющему также отражены данные о местности, на которой будет произведена постройка сооружения, и фото проекта (эпюра, проекция или фото готового здания) (код добавления в базу отражен в листинге 00). По возможно климатические особенности можно заполнить на основании уже имеющихся данных в справочнике, т.е. программа убедится в том, что пользователь выбрал город, и проверит список проектов на наличие записей, закрепленных за данной местностью, если таковые есть – поля заполнятся автоматически. Также на форме размещена кнопка для формирования отчетности в файл Word: по нажатию программа открывает ранее подготовленный шаблон и заполняет закладки документа необходимой информацией. Пример заполнения полей и создание таблицы представлен в листинге 00, а готовый отчет на рисунке 00.



Меню

Frame

Рисунок 17 - Вкладка «Основные исходные данные по проекту»

Листинг 2 – Добавление фотографии (присваивание свойству модели)

public RelayCommand? EditCommandImage {

get

{

return editCommandImage ??= new RelayCommand ((obj) =>

{

OpenFileDialog openFileDialog = new ()

{

Filter = "Image files (\*.png;\*.jpg)|\*.png;\*.jpg|All files (\*.\*)|\*.\*"

};

if (openFileDialog.ShowDialog() == true)

{

BasicData.myImageProject!.Source = new BitmapImage (new Uri (openFileDialog.FileName));

FileStream fileStream = new (openFileDialog.FileName.ToString (), FileMode.Open, FileAccess.Read);

BinaryReader binaryReader = new (fileStream);

Project!.Image = binaryReader.ReadBytes ((int) fileStream.Length);

}

});

}}

Листинг 3 – Заполнение вкладок в шаблоне Word

**//Заполнение закладки текстом**

document!.Bookmarks["Наименование"].Range.Text = project.Name;

**//Заполнение закладки фотографией**

if (project.Image != null) {

MemoryStream memory = new (project.Image);

BitmapImage bitmap = new ();

bitmap.BeginInit (); bitmap.StreamSource = memory; bitmap.EndInit();

Clipboard.SetImage(bitmap);

document!.Bookmarks["ФотоПроекта"].Range.Paste();

}

else document!.Bookmarks["ФотоПроекта"].Range.Text = "(ФОТО ОТСУТСТВУЕТ)";

**//Заполнение закладки таблицей**

List<FunctionalPurpose>? functionalPurposes = db.FunctionalPurposes.Where(c => c.ProjectId == project.Id).ToList();

if (functionalPurposes.Count != 0)

{

Word.Table table1 = document.Tables.Add (document.Bookmarks ["ТаблицаНазначений"].Range, functionalPurposes.Count + 1, 2);

table1.Select();

table1.Borders.Enable = 1;

table1.Cell (1, 1).Range.Text = "Назначение этажа";

table1.Cell (1, 2).Range.Text = "Тип этажа";

for (int i = 0; i < functionalPurposes.Count; i++)

{

table1.Cell (i + 2, 1).Range.Text = functionalPurposes[i]. Appointment?.Name;

table1.Cell (i + 2, 2).Range.Text = functionalPurposes[i]. FloorType?.Name;

}

}

else {

document.Bookmarks ["ТаблицаНазначений"].Range.Text = "Данные отсутствуют";}

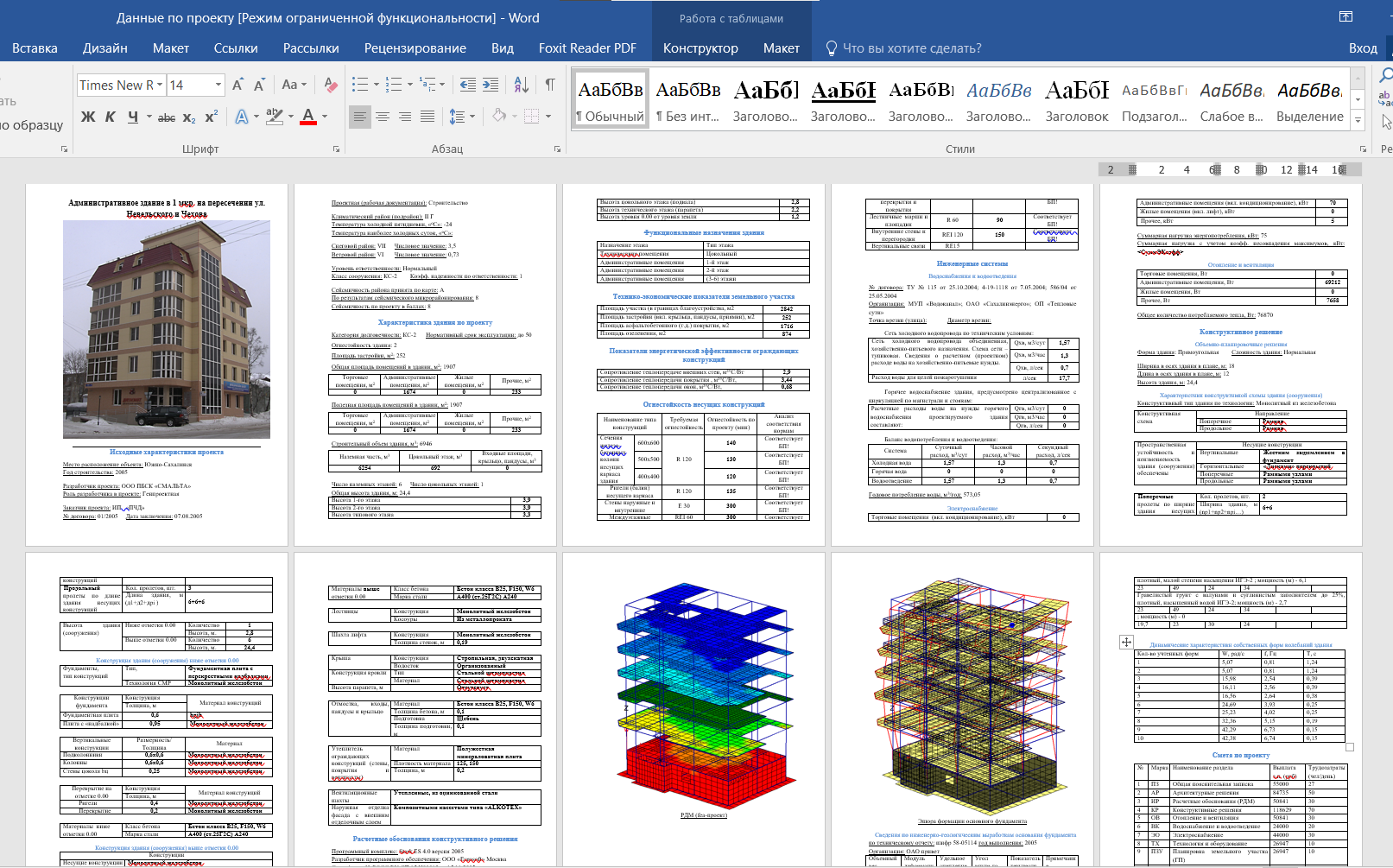


Рисунок 18 - Готовый отчет по проекту в Word

На второй вкладке проекта (рисунок 00) размещена информация о характеристики здания. Данный раздел хранит в себе сведения о таких технико-экономических характеристиках здания, как объемы и высота здания (количество этажей, высоты разных типов этажей), площади помещений, а также перечень функциональных назначений объекта поэтажно. Общие суммы просчитываются автоматически при изменении значений в полях ввода.

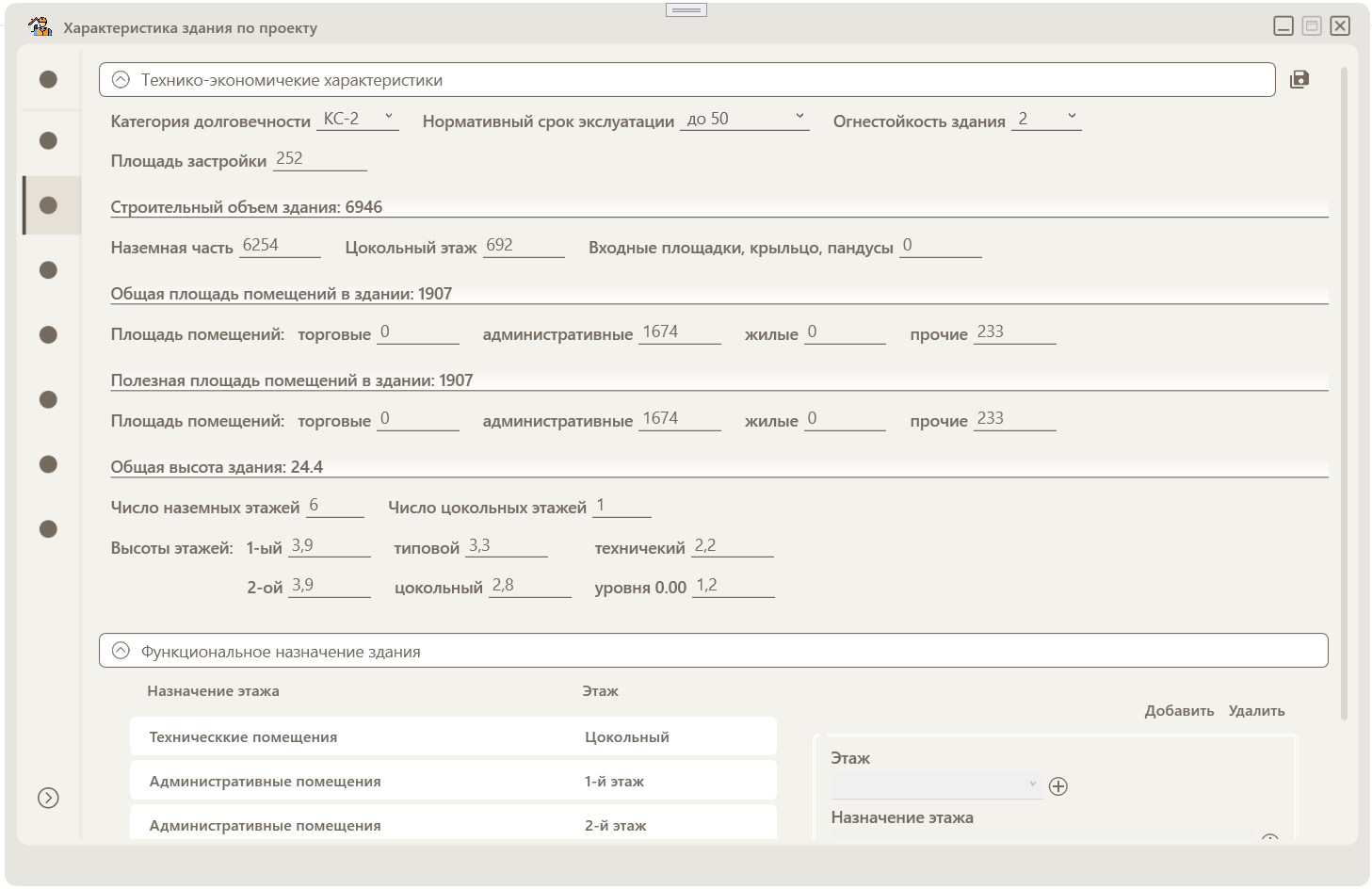


Рисунок 19 – Вкладка «Характеристика здания по проекту»

В третьей вкладке, показанной на рисунке 00, хранится информация по земельному участку, сведения о эффективности ограждающих конструкций и показатели огнестойкости несущих конструкций. Величина площади застройки – неизменяема, так как данный параметр заполняется в разделе характеристика здания, а тут просто выводится на просмотр.

При работе с показателями огнестойкости несущих конструкций выполняется автоматический анализ соответствия введённых значений нормам, прописанным в технических регламентированных актах. Результат проверки пользователь видит сразу после ввода в TextBlock, размещенном около поля ввода (TextBox). Пример проверки отображен в листинге 00.

Листинг 4 - Анализ значение огнестойкости при вводе данных

public string? Section600 {

get

{

if (FireResistance!.Section600 == null)

return string.Empty;

else

return FireResistance!.Section600.ToString ();

}

set

{

if (value != string.Empty)

{

FireResistance!.Section600 = Convert.ToInt32 (value);

if (FireResistance!.Section600 < 120) Section600Text = "НЕ соответствует ПБ!";

else Section600Text = "Соответствует ПБ!";

}

else

{

FireResistance!.Section600 = null;

Section600Text = string.Empty;

}

OnPropertyChanged("Section600");

}

}

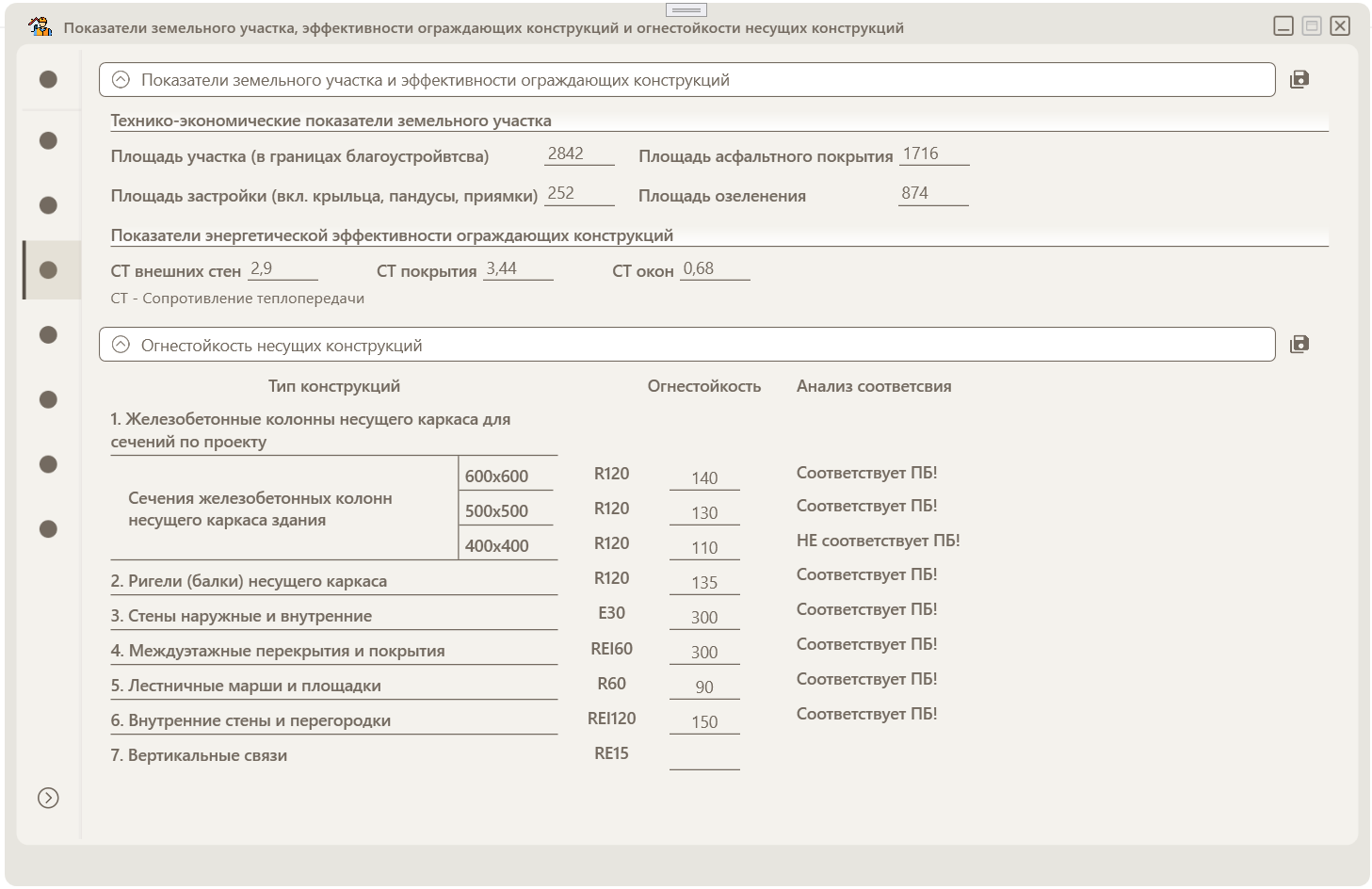


Рисунок 20 – Вкладка «Показатели земельного участка и конструкций»

Во вкладке четыре (рисунок 00) хранится информация о конструктивное решение проекта. Все данные раздела можно разделить на 3 составляющие:

1. Характеристика конструктивной схемы здания (сооружения) – включает в себя информацию по объёмно-планировочные решения, отображает на просмотр данные по высоте здания и определяет конструктивный тип сооружения;
2. Конструкция здания (сооружения) ниже отметки 0.00 – хранит данные о типе фундамента, технологии строительно-монтажных работ, вертикальных конструкциях и перекрытия ниже отметки 0.00, материалах;
3. Конструкция здания (сооружения) выше отметки 0.00 – содержит информацию о несущих конструкциях, типах лестниц, шахт лифтов, крыш (в том числе кровли, парапетов), входных частей (включая пандусы, крыльцо и отмостку), наружной отделка фасадов, утеплителя и вент шахт.

В разделе достаточно много полей со списком (combobox). При выборе значения из данного элемента управления может возникнуть ситуация, когда необходимый параметр отсутствует в перечне вариантов выбора. В этом случае пользователь может добавить новый параметр через диалоговое окно (рисунок 00), открывающееся по нажатию кнопки, расположенной справа от combobox. Такая функциональная возможность разработана для каждого списка, который содержит непостоянный набор значений.

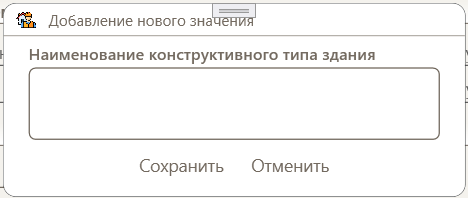


Рисунок 21 – Пример диалогового окна для добавления параметра

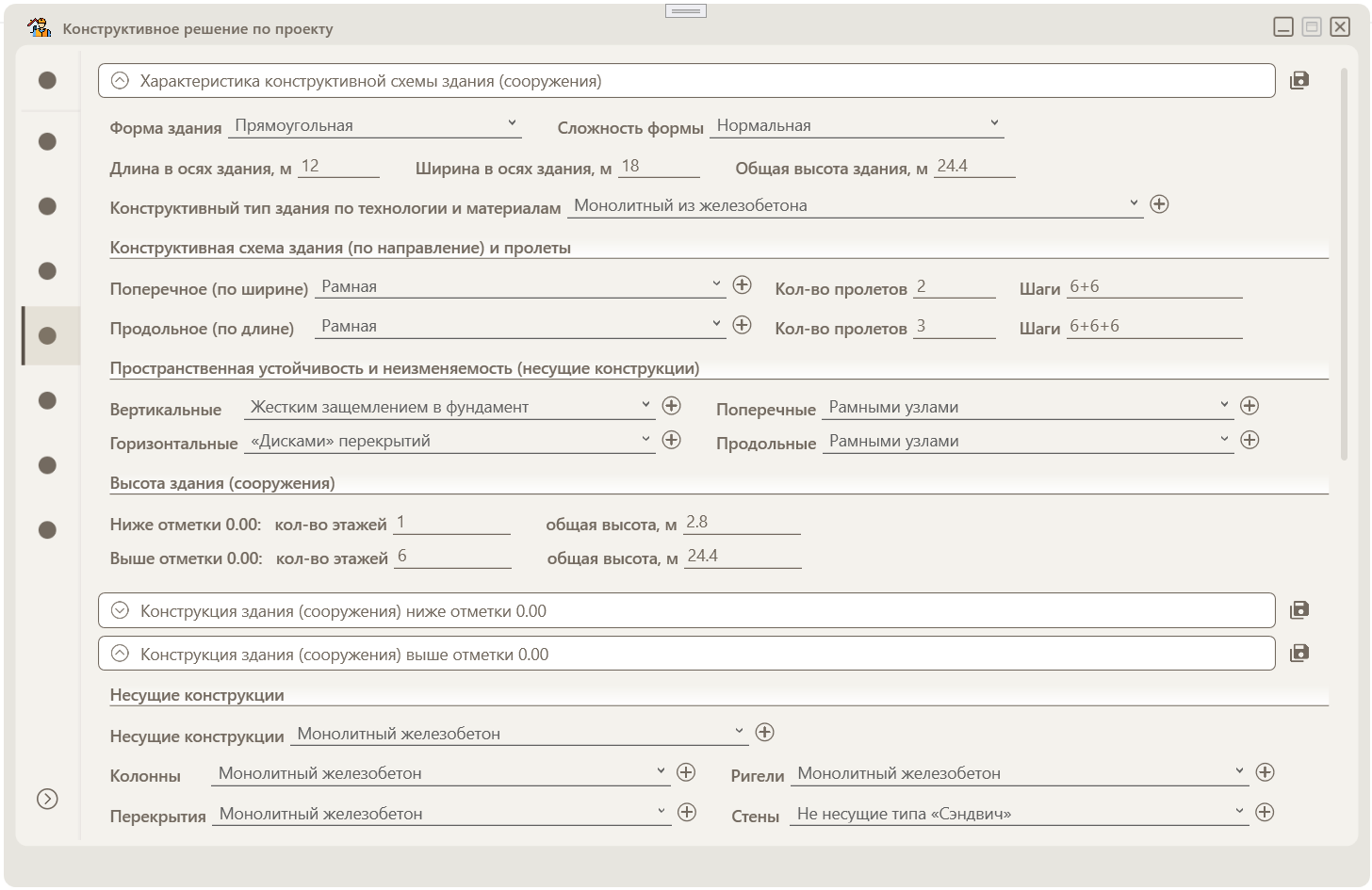


Рисунок 22 – Вкладка «Конструктивное решение здания»

Пятая вкладка (рисунок 00) посвящена обоснованию конструктивного решения, полученного при работе с данными с раздела четыре (конструктивное решение по проекту). В ней хранятся данные по результатам исследований на колебания собственных форм проекта здания (в том числе эпюра деформации и fea-проект), а также сведения по инженерно-геологическим выработкам основания фундамента.

На примере данного раздела рассмотрим реализацию кнопку сохранения, которая расположена на каждой вкладке. К ней привязана команда, которая рассчитана, как на добавление записи в таблицу базы данных, так и на изменение значений записи. Код команды представлен в листинге 00.

Листинг 5 – Код команды сохранения данных

public RelayCommand? EditCommandRationaleInfo {

get {

return editCommandRationaleInfo ??= new RelayCommand ((obj) => {

if (RationaleInfo!.Id != 0)

db.Entry(RationaleInfo).State = EntityState.Modified;

else

{

db.RationaleInfos.Add(RationaleInfo);

}

db.SaveChanges();

MainWindow.myChangeStatus!.Text = "Основная информация была успешно сохранена(обновлена)";

dispatcherTimer.Start();

});

}

}

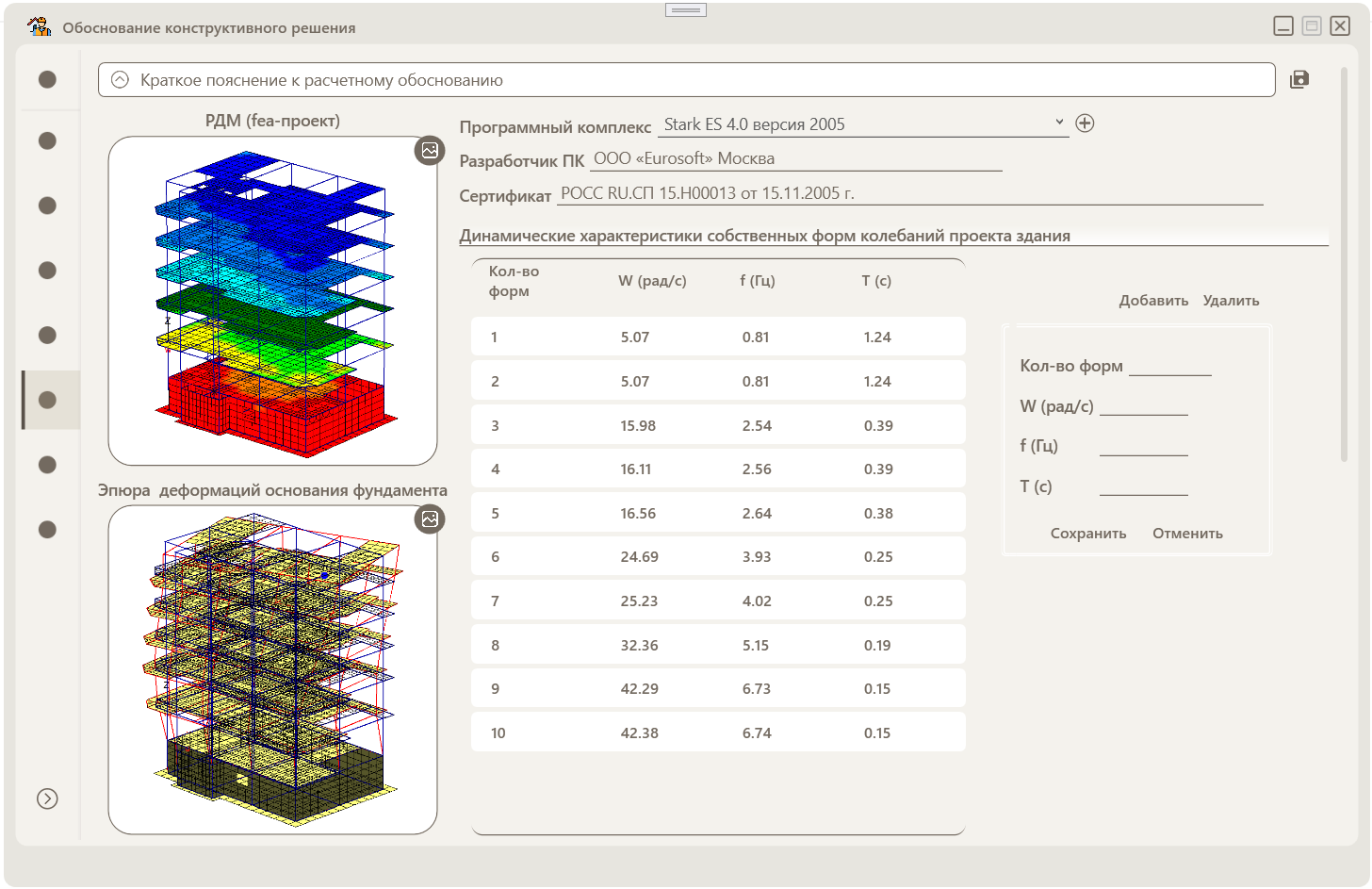


Рисунок 23 – Вкладка «Обоснование конструктивного решения»

На шестой вкладке вводятся данные по инженерным системам, а именно водо-, энерго- и теплоснабжению. В основном в качестве показателей пользователь вводит числовые значения, характеризующие среднее расчетное потребление благ (вода, электроэнергия и тепло). Для отслеживания корректности ввода данных к полям вкладки (в том числе и к полям других разделов) были применены события Text\_PreviewTextInput (срабатывает при вводе, отлавливает вводимые символы) и TextBox\_LostKeyboardFocus (срабатывает на при потере фокуса, проверяет корректность). Код событий представлен в листинге 00.

Листинг 6 – События для textbox

private void Text\_PreviewTextInput (object sender, TextCompositionEventArgs e) {

e.Handled = "0123456789,".IndexOf (e.Text) < 0;

}

private void TextBox\_LostKeyboardFocus (object sender, KeyboardFocusChangedEventArgs e) {

TextBox txt = (TextBox)e.Source;

if (txt.Text != "")

{

try {

Convert.ToDouble(txt.Text); }

catch {

txt.Focus();}

}

}

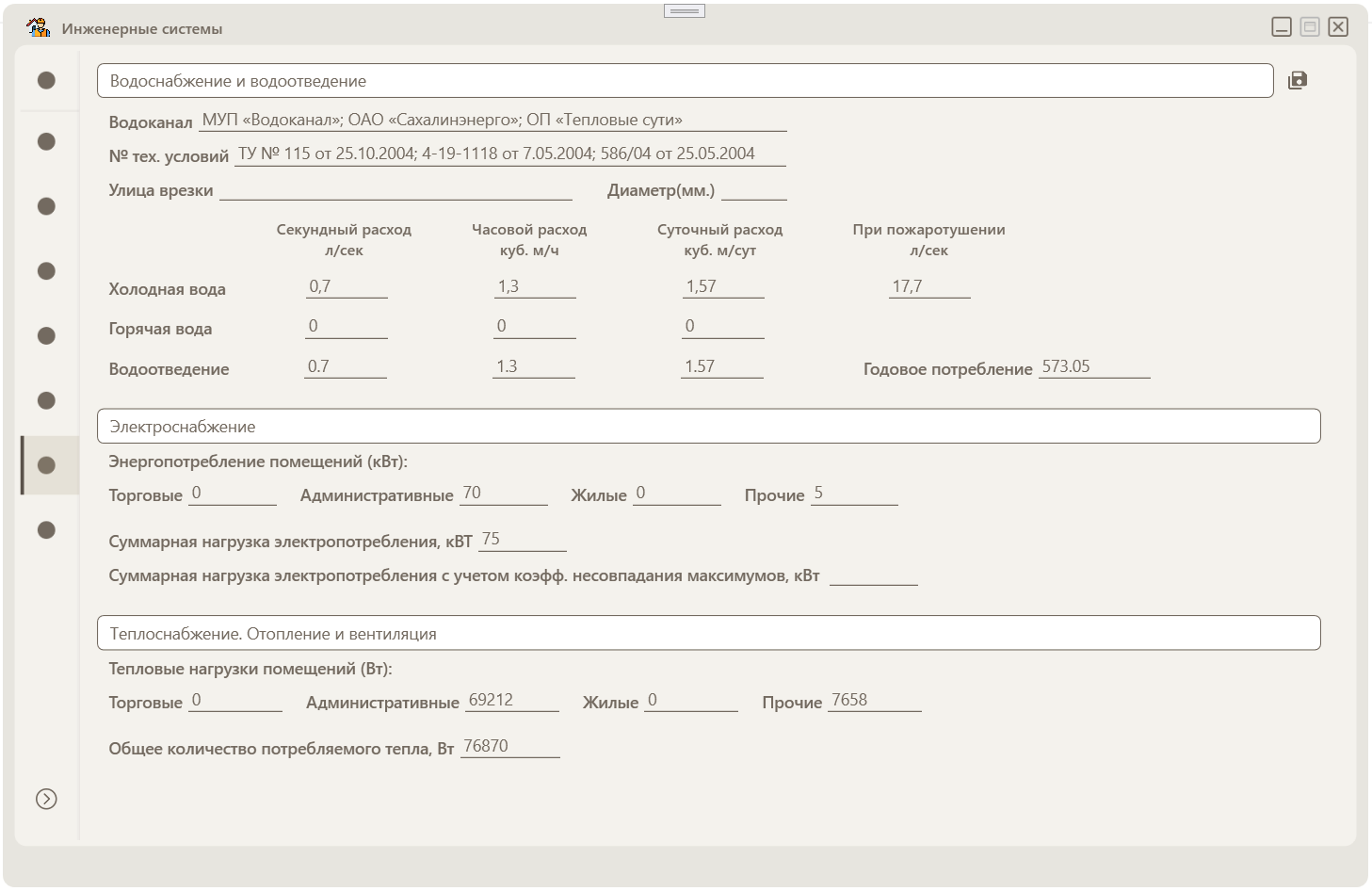


Рисунок 24 – Вкладка «Инженерные системы»

Седьмая вкладка, отображенная на рисунке 00, предназначена для информации по смете проекта. На изображении показан общий список пунктов сметы, которые заполняются из проекта в проект. Для ускорения процесса работы было решено автоматически создавать перечень записей для текущего проекта. В таком случай удастся избежать повторения разделов, ведь пользователю будет только вносить изменения в параметры выплата и трудозатраты. Код проверки на наличие записей и добавления в случае отсутствия представлен в листинге 00.

Листинг – Проверка и заполнение перечня разделов сметы по проекту

Estimates = db.Estimates.Where(c => c.ProjectId == ProjectId).ToList();

if (Estimates.Count == 0)

{

foreach (Chapter chapter in Chapters)

{

db.Estimates.Add(new()

{

ProjectId = projectID,

Chapter = chapter,

});

}

db.SaveChanges ();

Estimates = db.Estimates.Where(c => c.ProjectId == ProjectId).ToList();

}

Рассмотренное выше автоматическое заполнение также помогает в заполнении отчетности в Excel, по кнопке, размещенной на вкладке. По нажатию программа открывает ранее подготовленный шаблон и заполняет пустые ячейки документа необходимой информацией. Пример заполнения ячеек представлен в листинге 00, а готовый отчет на рисунке 00.

Листинг 8 - Заполнение пустых ячеек Excel

using ExcelHelper excel = new(Directory.GetCurrentDirectory() + @"\Расчетная форма проекта.xlsx");

var sheet = excel.application!.Workbooks[1].Worksheets[1];

sheet.Cells[2, 9].Value = DateTime.Now.ToShortDateString();

sheet.Cells[4, 4].Value = db.AllProjects.First(c => c.Id == ProjectId).Name;

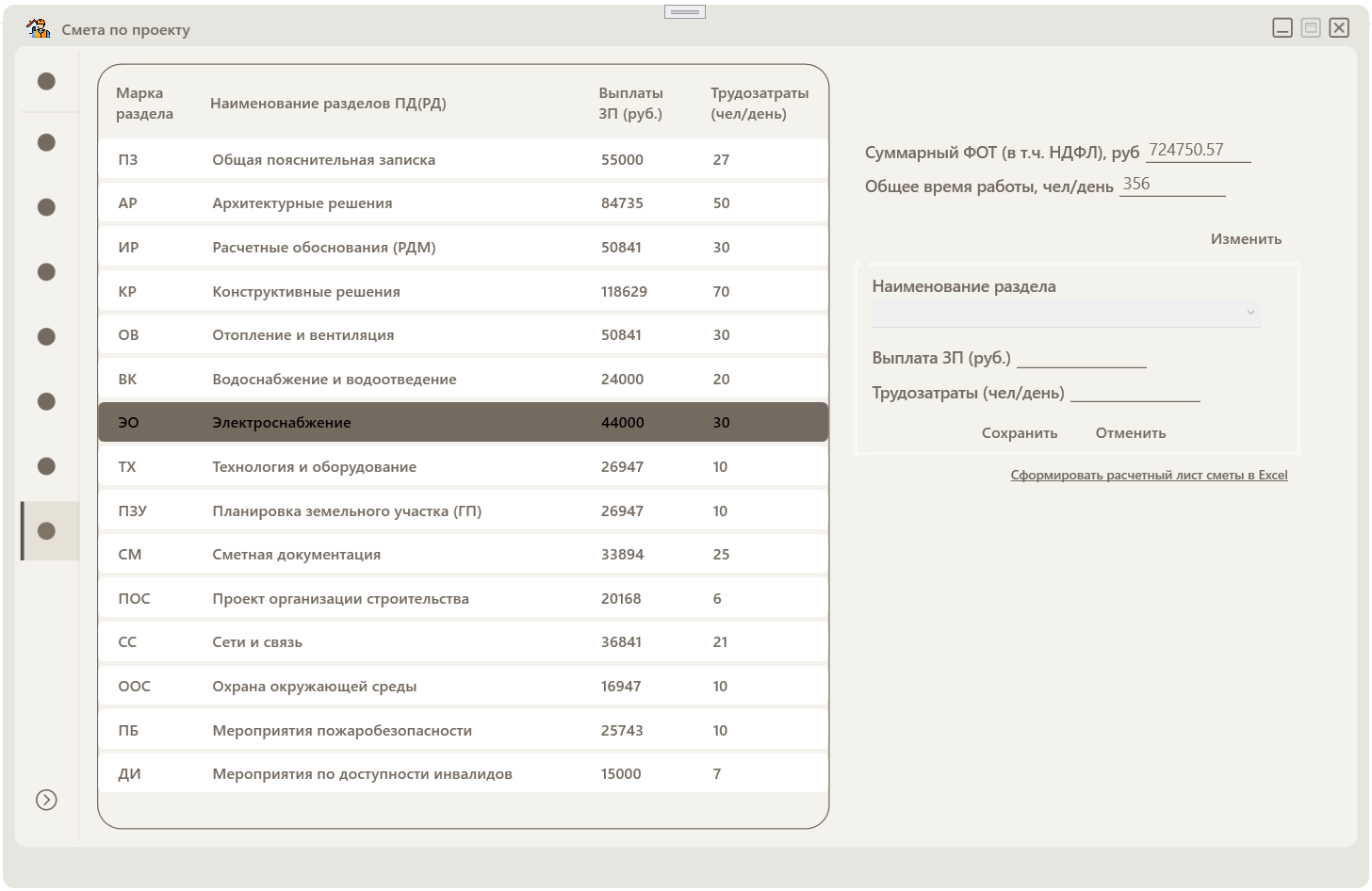
for (int i = 0; i < Estimates!.Count; i++)

{

sheet.Cells[8 + i, 5].Value = Estimates[i].LaborCosts;

sheet.Cells[8 + i, 9].Value = Estimates[i].Pay;

}

Рисунок – Вкладка «Смета по проекту»

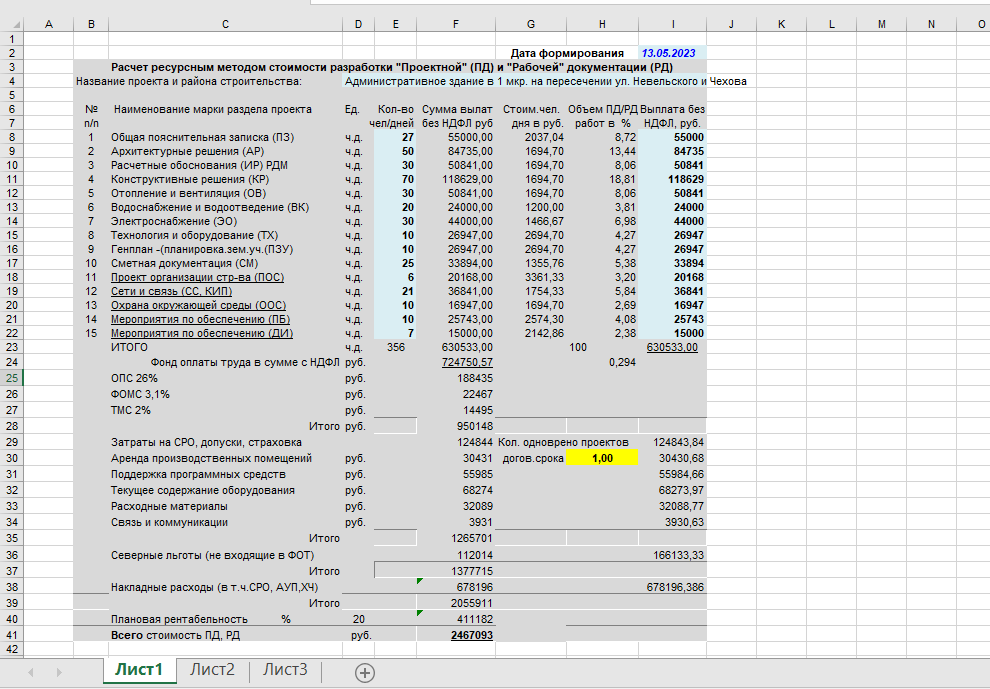


Рисунок 26 – Отчет по смете в Excel

## Раздел аналитики

Раздел аналитики подразделяется на три составляющие: динамика проектов, анализ трудозатрат и оценка выручки с проектов.

В первой вкладке главный инженер может просматривать динамику движения проектов по годам за выделенный период. Функционал вкладки рассчитан на то, что пользователь определяет временной период, учитывая который предполагается рассмотреть количество выполненных проектов по годам. Дополнительно можно определить типы документаций, которые необходимо учесть при отображении данных на основном (столбчатом) графике.

Кроме главного графике на вкладке расположены две дополнительный круговые диаграммы. Одна из них выполняется роль детализации количество проектов по одной из документаций (тип можно выбрать из выплывающего списка) по годам, а вторая показывает количество проектов за выбранный период, где компания выполняла генпроектную или субподрядную роли.



Рисунок 27 - Вкладка "Динамика проектов"

Вторая вкладка предполагает анализ трудозатрат. С помощью данного раздела главный инженер может осуществлять оценку затрат человеческого труда и при этом определять примерные сроки выполнения текущего проекта по той или иной документации для оглашения заказчику.

Функционал предполагает, что для сравнения и расчета трудозатрат пользователь будет рассматривать проекты в разрезе одной документации. Данный принцип рассчитан на рациональный расчет средних значений без выхода за рамки одной документации, так как в зависимости от ее вида время выполнения пунктов сметы могут сильно отличаться.

Общий список проектов фильтруется при изменении документации, а при выборе одного из проектов, информация сметы по нему попадает на линейный график, на котором в удобном формате можно оценить каждый пункт. Во время обновление перечня рассматриваемых проектов осуществляется пересчет средних значений по каждому пункту сметы. При желании можно очистить все сведения изменить тип документации и вновь произвести анализ.

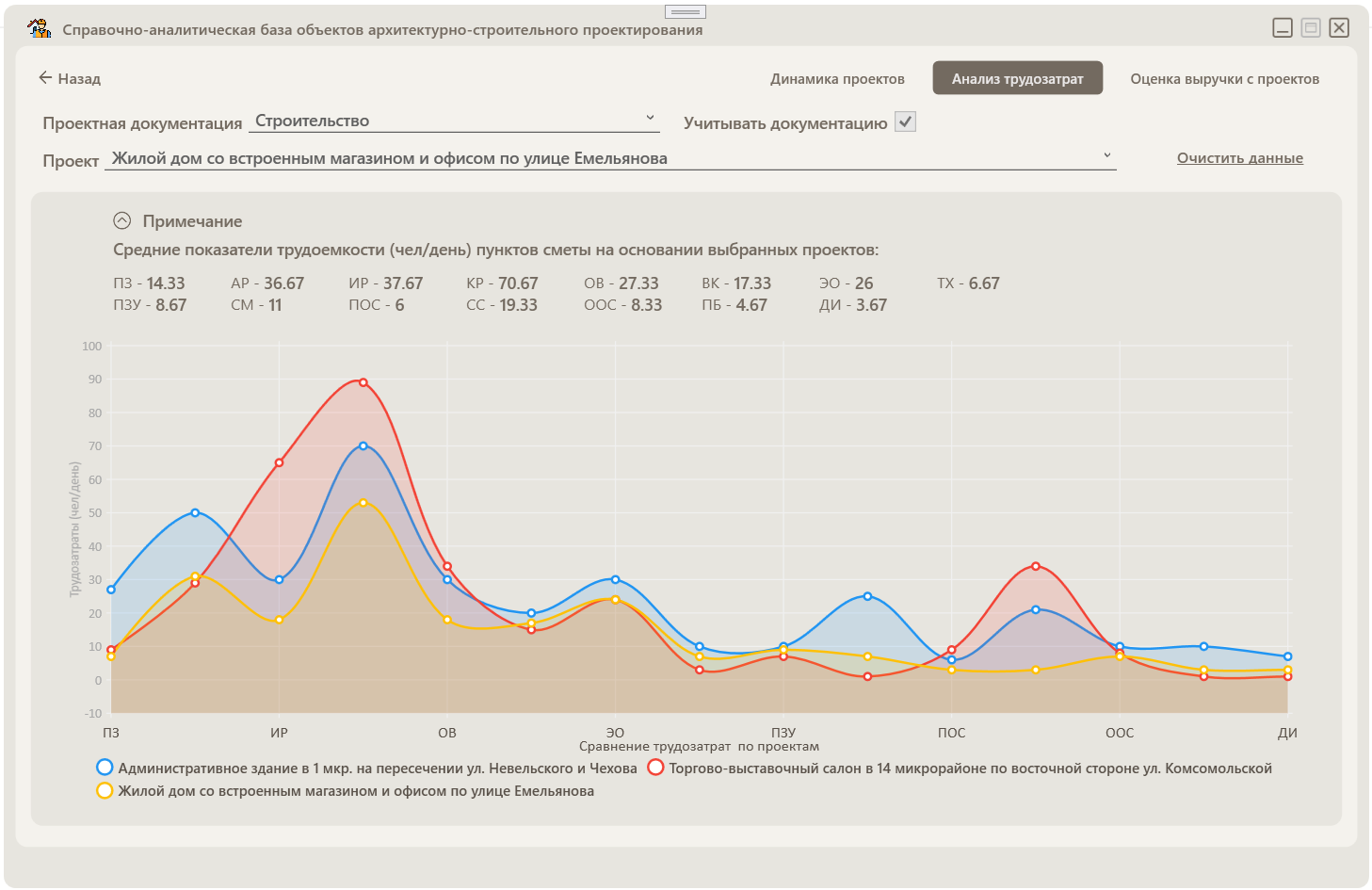


Рисунок 28 - Вкладка "Анализ трудозатрат"

На третьей вкладке можно оценить прибыть компании по годам. Как и в разделе номер один пользователь определяет временной промежуток, по которому хочет просмотреть информацию. Формирование графика проходит в 4 этапа (каждый соответствует одному из типов документации). На каждом этапе осуществляется отбор проектов, а далее программа поочередно обращается к смете каждого проекта и вычисляет общую сумму прибыли. Благодаря такому методу формирования графика пользователь может оценить прибыль по каждому году в разрезе всех документаций, понять какие проекты приносят большую прибыль. Также во время формирования графика программа определяет год максимальной(минимальной) прибыли за рассмотренный период, указывает сумму и количество выполненных проектов

В качестве дополнения на вкладке расположена возможность детального вывода списка проектов по указанному году. Пользователю необходимо только ввести год, по которому ему необходимо узнать названия проектов и сумму выручки по каждому их них.

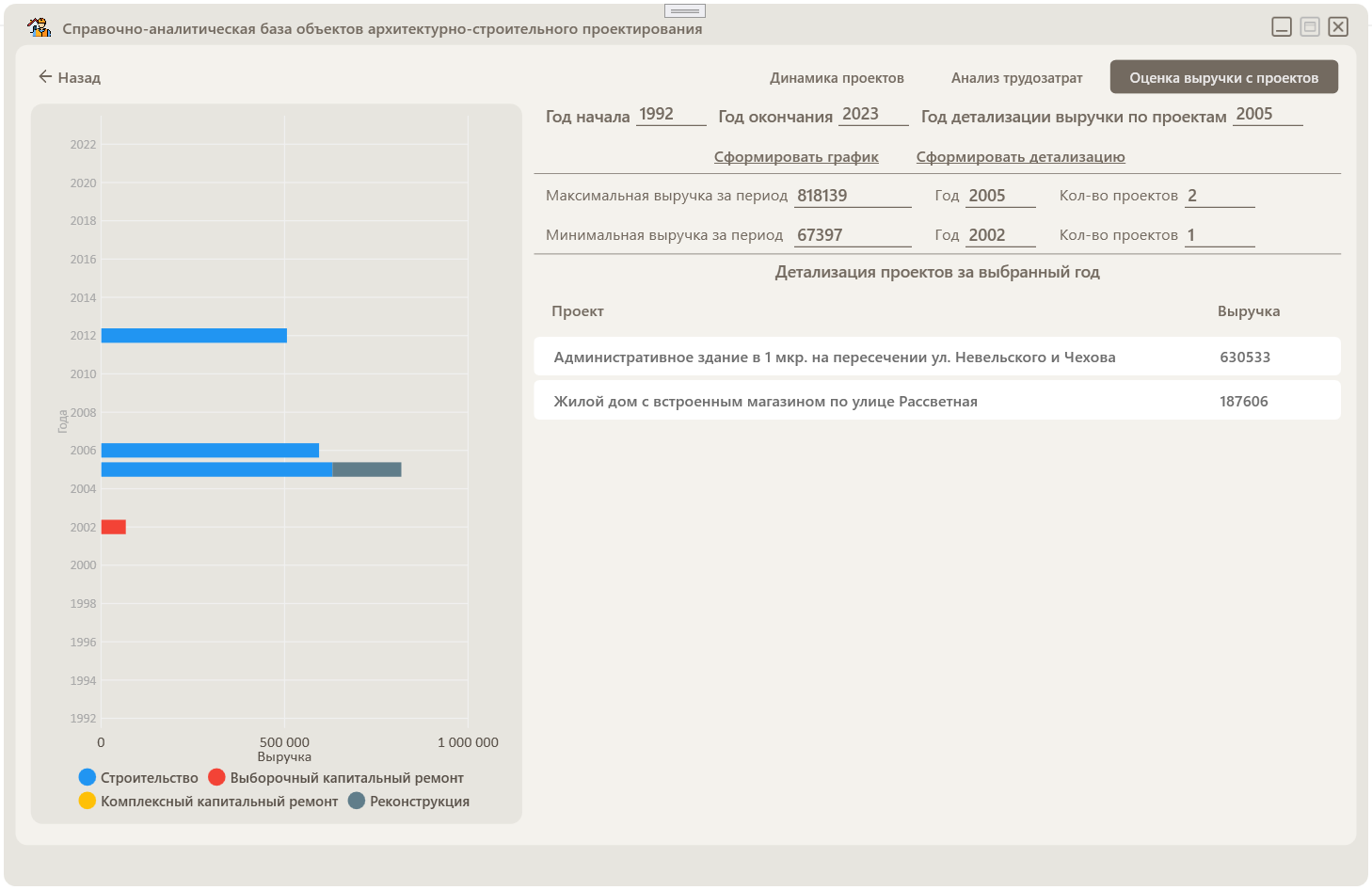


Рисунок 29 - "Оценка выручки с проектов"

## Вывод по третьему разделу

В ходе разработки была создана справочно-аналитическая база атрибутивных данных информационных моделей объектов архитектурно-строительного проектирования на основе Windows Presentation Foundation в среде Visual Studio. В основу разработки лег паттерн MVVM (Model-View-ViewModel) с использованием подхода «Model First».

В первой части разработки были реализованы модели классов для создания таблиц базы с помощью технологии Entity Framework и дальнейшей работы с данными; во второй – разработаны индивидуальные стили элементов управления и создано стилевое оформление страниц для основных данных программного обеспечения, а также для раздела аналитики; в третьей – была разработана прослойка (модель-представление), отвечающая за логики программы при помощи которой осуществляется соединение интерфейса и моделей.

# ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ПРОГРАММЫ

## Расчет затрат на разработку системы

## Затраты компании до внедрения программы

## Экономическая эффективность, затраты после внедрения

## Вывод по четвертому разделу

# ЗАКЛЮЧЕНИЕ

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

<https://professorweb.ru/my/WPF/documents_WPF/level36/36_5.php>

<https://www.entityframeworktutorial.net/efcore/entity-framework-core-dbcontext.aspx>