**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт Цифровых технологий моделирования в строительстве (ИЦТМС)

Кафедра Информационных систем, технологий и автоматизации в строительстве (ИСТАС)

**КУРСОВАЯ РАБОТА**

по дисциплине

«Архитектура прикладного программного обеспечения»

Тема:

«Разработка ИС проектной организации»

|  |  |
| --- | --- |
| Выполнил обучающийся | ИЦТМС 3-5 Аблязизов А.Э. |
|  | (институт, курс, группа, Ф.И.О.) |
| Руководитель курсовой работы | ст. преподаватель Садовский Б. С. |
|  | (учёное звание, учёная степень, должность, Ф.И.О.) |
| К защите |  |
|  | (дата, подпись руководителя) |
| Курсовая работа защищена с оценкой |  |
|  | (оценка цифрой и прописью) |
| Руководитель курсовой работы |  |
|  | (дата, подпись руководителя) |
| Председатель аттестационной комиссии |  |
|  | (учёное звание, учёная степень, должность, Ф.И.О.) |
| Члены комиссии: |  |
|  |  |
|  |  |
|  | (дата, подпись члена комиссии) |
|  | |

Москва

2023

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт Цифровых технологий моделирования в строительстве (ИЦТМС)

Кафедра Информационных систем, технологий и автоматизации в строительстве (ИСТАС)

Дисциплина Архитектура прикладного программного обеспечения

**ЗАДАНИЕ**

**НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ**

Студенту группы ИЦТМС-3-5 Аблязизову Арсену Эскендеровичу

1. Тема курсовой работы «Разработка ИС рекламной компании»
2. Исходные данные к курсовой работе:
   1. лекции, компьютерные практики и курсовая работа по курсу ООП;
   2. лекции и компьютерные практики по курсу «Архитектура прикладного ПО»;
   3. исходный код примеров решения отдельных деталей интерфейса Qt;
   4. документация по Qt;
   5. указания по оформлению отчёта;
   6. инструкция по использованию программы Doxygen.
3. Перечень подлежащих разработке вопросов:
   1. Программа должна быть написана на Си++ используя кроссплатформенный инструментарий разработки Qt или аналога.
      1. Программа должна хранить данные в БД, уметь работать по сети с СУБД PostgreSQL. То есть должна:
      2. Подключатся к СУБД.
      3. Отображать данные из таблиц и/или представлений.
      4. Добавлять данные в БД.
      5. Редактировать данные в БД.
      6. Удалять данные из БД.
   2. В БД должно храниться не менее 20-и записей в текстовом формате. Каждая запись — это один пункт списка таблицы. По каждой записи должно быть не менее 6 свойств (характеристик). Все свойства определяются целесообразностью их упоминания.
   3. Исходный текст программы должен быть полностью задокументирован с помощью комментариев и команд программы Doxygen.
   4. Программа должна состоять из главного окна и диалоговых окон (как минимум одного диалогового окна).
   5. В главном окне должно быть простое меню для реализации основных задач обработки данных.
   6. Объекты в окне не должны расползаться при изменении его размеров.
   7. В программе обязательно должно быть указано авторство (институт-курс-группа, фамилия, имя отчество полностью).
   8. Авторство указывается в диалоговом окне, вызываемым из меню главного окна. Оформить как пункт меню об авторе.
   9. Программа должна быть на русском языке, но при этом язык программы должен быть оформлен как перевод из файла qm. В файлах с исходным кодом (\*.cpp, \*.h, \*.hpp) всё должно быть на английском через функцию tr(). Аналогично в файлах с пользовательским интерфейсом (\*.ui) весь интерфейс должен быть на английском.
   10. Программа должна запоминать настройки интерфейса.
   11. Для хранения и отображения данных программа может использовать QTableWidget. Загрузка и работа с данными может быть реализована через QsqlQuery.
   12. Вместо использования QTableWidget, должен быть реализован архитектурный паттерн «Модель-представление-контроллер» (англ. Model-View-Controller, MVC).
   13. Для обработки корректности, вводимых пользователем данных, должны использоваться соответствующие объекты по написанным классам отображения элементов, а также рекомендуется применять различные валидаторы и маски ввода.
   14. Должна быть реализована возможность поиска и сортировки по отдельным данным.
   15. У таблиц должны быть реализованы защита данных от аномалий обновления. Например, должна быть связь по ключам.
   16. В БД необходимо реализовать схему из нескольких таблиц, связанных через внешние ключи.
   17. Должно быть разделение обработки данных. На стороне сервера должна быть реализована обработка объёмных данных, а на стороне клиента оставить простые операции. Например: расчёт зарплат сотрудникам должен быть на стороне сервера, а отображение цен на товары по курсу валюты, на стороне клиента.
   18. Загрузка данных на клиент должна быть по частям.
   19. При изменении данных в таблицах в несколько запросов, необходимо использовать транзакции.
   20. Соединение с СУБД PostgreSQL надо организовать через SSL.
4. Предоставить отчёт о проделанной работе оформленные согласно нормам ГОСТ, UML, положению о курсовых проектах и курсовых работах обучающихся (Выпуск 3) и др. В отчёте обязательно должны быть разделы:
   1. Введение.
   2. Основная часть.
      1. Входные данные.
      2. Интерфейс пользователя.
      3. Структура проекта.
      4. Описание классов и методов.
   3. Заключение.
   4. Библиографический список.
   5. Приложение (опционально).
   6. Основная часть отчёта формируется с использованием программы Doxygen на основе комментариев в исходном коде.

График выполнения курсовой работы:

| № | Наименование этапа выполнения курсовой работы | Срок выполнения | Процент выполнения курсовой работы |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | Получение задания | 28.02.2022 |  |
| 2 | Создание БД, создание главного окна приложения и стандартные диалоговые окна. Меню программы | 02.03.2022 |  |
| 3 | Организован интерфейс приложения с учётом изменения геометрии окна и политик. Сохранение настроек | 16.03.2022 |  |
| 4 | Сделать подключение программы клиента к СУБД PostgreSQL | 30.03.2022 |  |
| 5 | Организация работы с таблицами БД. | 13.04.2022 |  |
| 6 | Поиск и сортировка. Шаблон проектирования MVC | 27.04.2022 |  |
| 7 | Доработка и устранение ошибок в приложении | 11.05.2022 |  |
| 8 | Написание отчёта | 25.05.2022 |  |

1. Дата выдачи задания 25.02.2022.

Задание получил студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Руководитель курсовой работы:

ст. преподаватель кафедры ИСТАС

Садовский Б. С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc106971477)

[1 Исходные данные 7](#_Toc106971478)

[2 Технические требования 9](#_Toc106971479)

[3 Функциональные возможности 11](#_Toc106971480)

[4 Интерфейс программы. 12](#_Toc106971481)

[4.1 Главное окно 12](#_Toc106971482)

[4.2 Работа с записями в документе 14](#_Toc106971483)

[5 Описание исходного кода программы 16](#_Toc106971484)

[5.1 Иерархия классов 16](#_Toc106971485)

[5.2 Описание классов и их методов 17](#_Toc106971486)

[5.2.1 Класс Employee 17](#_Toc106971487)

[5.2.2 Класс EmployeeDb 20](#_Toc106971488)

[5.2.3 Класс EmployeeDialog 22](#_Toc106971489)

[5.2.4 Класс MainWindow 24](#_Toc106971490)

[5.2.5 Класс ProxyModel 26](#_Toc106971491)

[6 Список файлов с исходным кодом и вспомогательных 27](#_Toc106971492)

[ЗАКЛЮЧЕНИЕ 28](#_Toc106971493)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 29](#_Toc106971494)

**ВВЕДЕНИЕ**

В данной работе требуется написать программу, с применением механизмов ООП языка Си++ в среде Qt. Программа должна иметь в себе функции учета параметров и разных элементов ПЭВМ.

Программа должна хранить данные в БД и уметь работать по сети с СУБД PostgreSQL. То есть должна:

* + - 1. Подключатся к СУБД.
      2. Отображать данные из таблиц и/или представлений.
      3. Добавлять данные в БД.
      4. Редактировать данные в БД.
      5. Удалять данные из БД.

В данном ПО должно иметься основное окно, через которое будет вестись работа с записями в БД. Диалоговые окна содержат в себе информацию о программе, а также предоставляют возможность редактировать и добавлять записи в БД. Должна быть возможность сортировать данные в таблице и осуществлять поиск по данным. Программа должна быть переведена на английский и русский языки.

**1 Исходные данные**

Входные данные приведены на рисунке 1, это таблица projects, содержащиеся в базе данных proects. Первичный ключ таблицы это id.

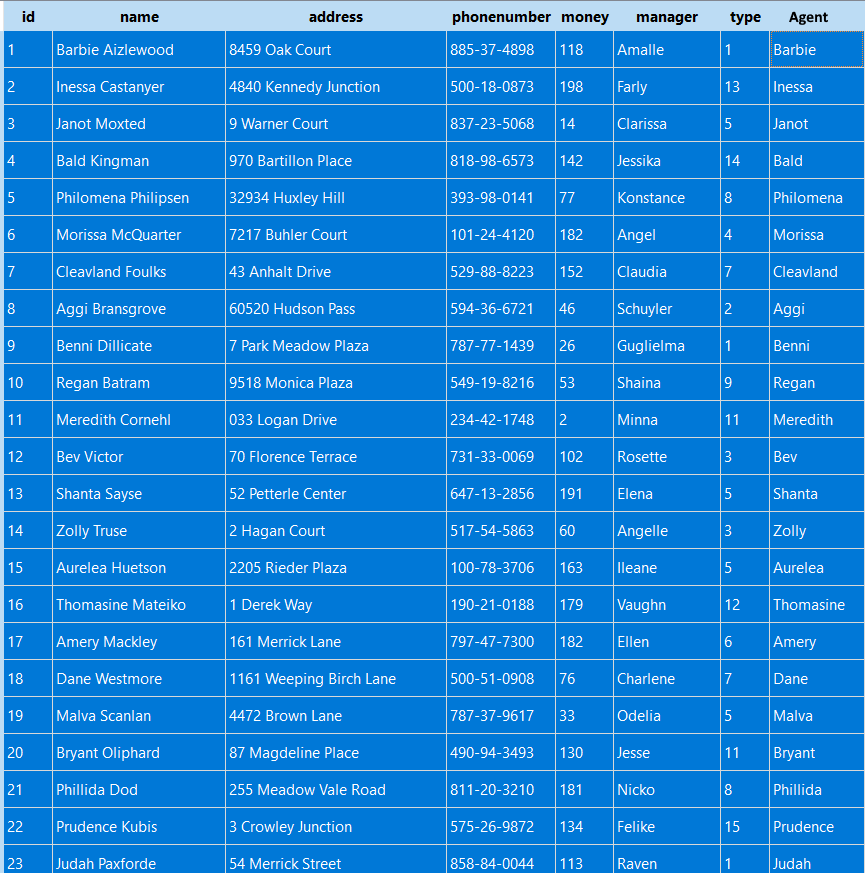


Рисунок 1– Входные данные из таблицы projects

**2 Технические требования**

* 1. Программа должна быть написана на Си++ используя кросс-платформенный инструментарий разработки Qt или аналога. Программа должна хранить данные в БД, уметь работать по сети с СУБД PostgreSQL. То есть должна:
     + 1. Подключатся к СУБД.
       2. Отображать данные из таблиц и/или представлений.
       3. Добавлять данные в БД.
       4. Редактировать данные в БД.
       5. Удалять данные из БД.
  2. В БД должно храниться не менее 1000-и записей в текстовом формате. Каждая запись — это один пункт списка таблицы. По каждой записи должно быть не менее 6 свойств (характеристик). Все свойства определяются целесообразностью их упоминания.
  3. Исходный текст программы должен быть полностью задокументирован с помощью комментариев и команд программы Doxygen.
  4. Программа должна состоять из главного окна и диалоговых окон (как минимум одного диалогового окна).
  5. В главном окне должно быть простое меню для реализации основных задач обработки данных.
  6. Объекты в окне не должны расползаться при изменении его размеров.
  7. В программе обязательно должно быть указано авторство (институт-курс-группа, фамилия, имя отчество полностью).
  8. Авторство указывается в диалоговом окне, вызываемым из меню главного окна. Оформить как пункт меню об авторе.
  9. Программа должна быть на русском языке, но при этом язык программы должен быть оформлен как перевод из файла qm. В файлах с исходным кодом (\*.cpp, \*.h, \*.hpp) всё должно быть на английском через функцию tr(). Аналогично в файлах с пользовательским интерфейсом (\*.ui) весь интерфейс должен быть на английском.
  10. Программа должна запоминать настройки интерфейса.
  11. Для хранения и отображения данных программа может использовать QTableWidget. Загрузка и работа с данными может быть реализована через QSqlQuery.
  12. Вместо использования QTableWidget, должен быть реализован архитектурный паттерн «Модель-представление-контроллер» (англ. Model-View-Controller, MVC).
  13. Для обработки корректности, вводимых пользователем данных, должны использоваться соответствующие объекты по написанным классам отображения элементов, а также рекомендуется применять различные валидаторы и маски ввода.
  14. Должна быть реализована возможность поиска и сортировки по отдельным данным.
  15. У таблиц должны быть реализованы защита данных от аномалий обновления. Например, должна быть связь по ключам.
  16. В БД необходимо реализовать схему из нескольких таблиц, связанных через внешние ключи.
  17. Должно быть разделение обработки данных. На стороне сервера должна быть реализована обработка объёмных данных, а на стороне клиента оставить простые операции. Например: расчёт зарплат сотрудникам должен быть на стороне сервера, а отображение цен на товары по курсу валюты, на стороне клиента.
  18. Загрузка данных на клиент должна быть по частям.
  19. При изменении данных в таблицах в несколько запросов, необходимо использовать транзакции.
  20. Соединение с СУБД PostgreSQL надо организовать через SSL.

**3 Функциональные возможности**

Программа написана на Си++ с использованием кроссплатформенного инструментария разработки Qt. Программа хранит данные в БД и умеет работать по сети с СУБД PostgreSQL. То есть она:

* Подключается к СУБД.
* Отображает данные из таблиц.
* Добавляет данные в БД.
* Редактирует данные в БД.
* Удаляет данные из БД.

В БД хранятся 1000 записей. Каждая запись — это один пункт списка таблицы. По каждой записи 8 свойств.

Программа состоит из главного окна и диалоговых окон.

В главном окне простое меню для реализации основных задач обработки данных. Объекты в окне не расползаются при изменении его размеров.

Авторство указано в диалоговом окне, вызываемым из меню главного окна.

Весь интерфейс программы на английском языке, но есть возможность переключиться на русский. Программа запоминает настройки интерфейса.

Вместо использования QTableWidget, реализован архитектурный паттерн «Модель-представление-контроллер» (англ. Model-View-Controller, MVC).

Реализована возможность поиска и сортировки по отдельным данным.

**4 Интерфейс программы.**

**4.1 Главное окно**

Далее приведены скриншоты с демонстрацией интерфейса приложения на рисунках 4.1, 4.2 и 4.3.

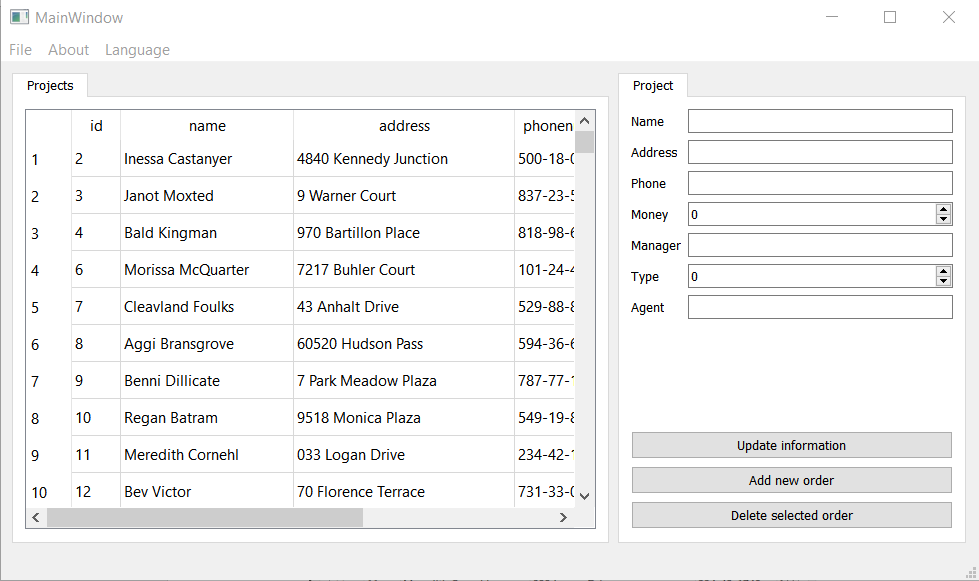


Рисунок 4.1 – Скриншот основного окна

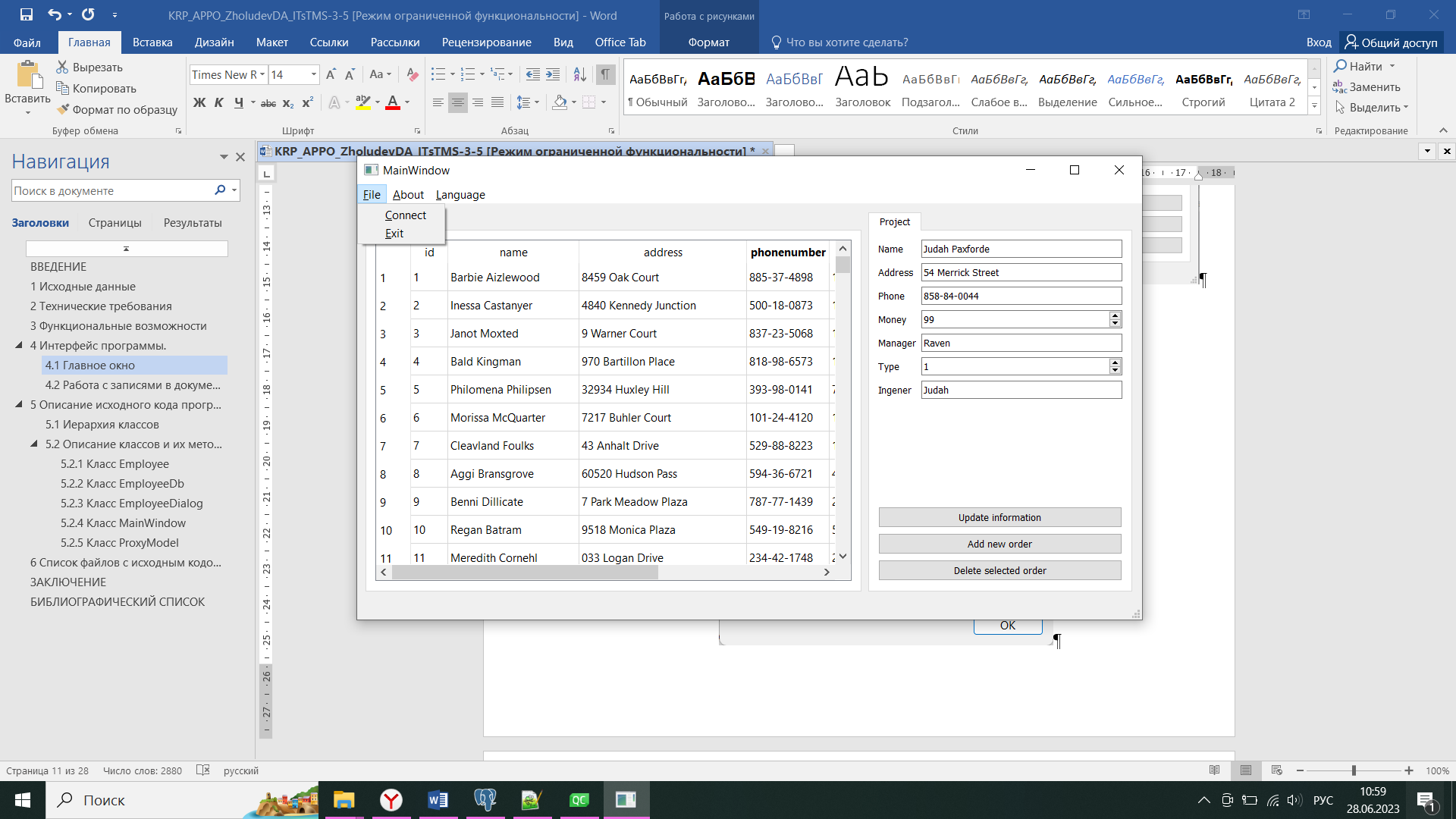


Рисунок 4.2 – Меню «Файл»

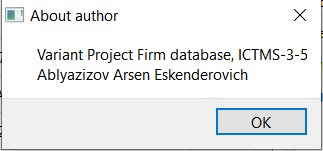


Рисунок 4.3 – Скриншот окна с отображением информации о авторе

**4.2 Работа с записями в документе**

Добавление новой записи происходит после открытия БД в приложении и заполнении формочки «Проект», нажатием на кнопку «Добавить новый заказ», изображенное на рисунке 4.4.

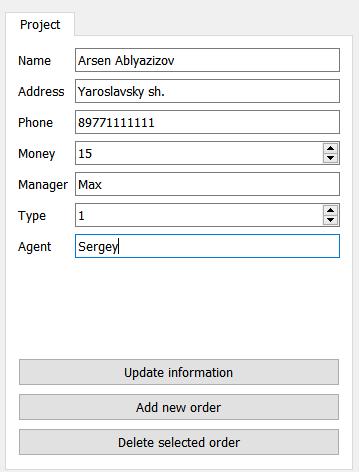


Рисунок 4.4 – Скриншот окна с добавлением записи

Редактирование записи происходит после открытия БД в приложении и выбора нужной записи для редактирования. Изменения вносятся в формочку аналогичную той, что использовалась для добавления нового проекта, Изменения сохраняются нажатием на кнопку «Обновить информацию» изображенное на рисунке 4.5.

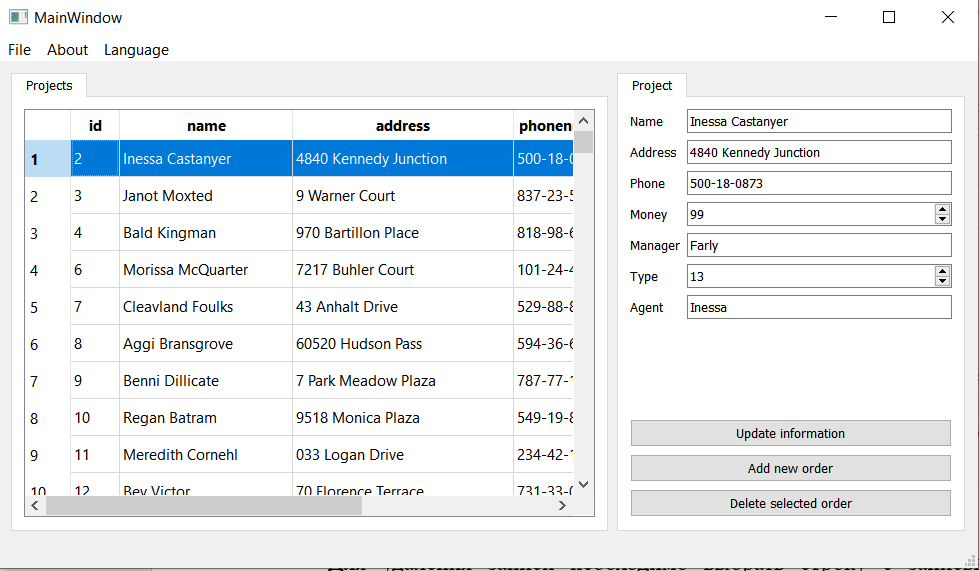


Рисунок 4.5 – Скриншот окна с редактированием записи

Для удаления записи необходимо выбрать строку с записью, которую необходимо удалить, и нажать на кнопку «Удалить выбранный заказ», рисунок 4.6.

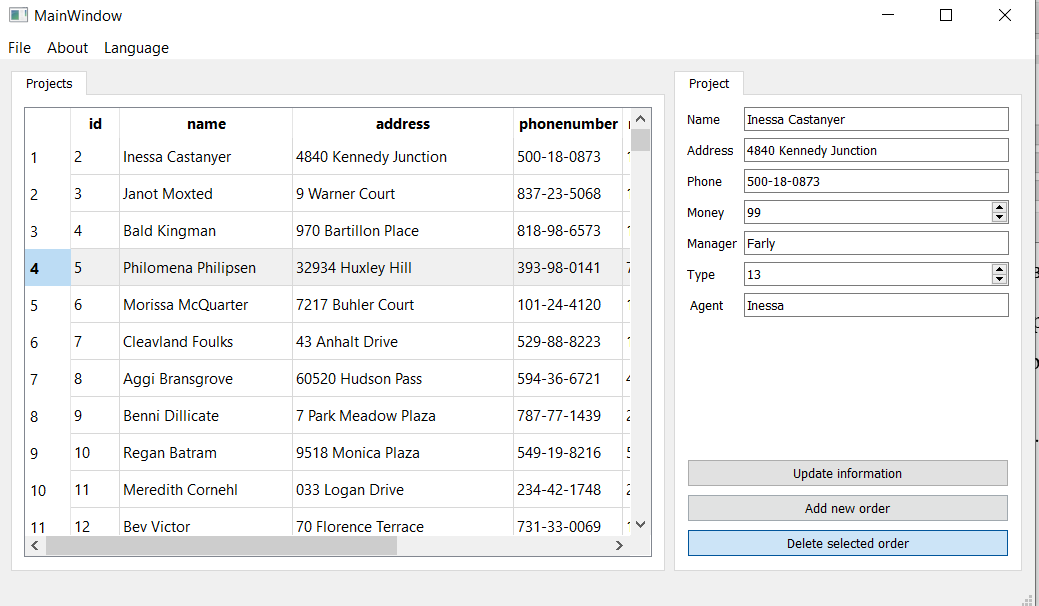


Рисунок 4.6 – Скриншот удаления

**5 Описание исходного кода программы**

**5.1 Иерархия классов**

Ниже представлена иерархия классов, созданных в проекте.

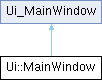
Иерархия классов:

* Ui

MainWindow 24

**5.2 Описание классов и их методов**

**5.2.1 Класс MainWindow**



#include <QMainWindow>

*объявление типа модели SQL-запросов*

class QSqlQueryModel;

*действие пункта подключения к БД*

void on\_actionConnect\_triggered();

*действие сбора значений из строки таблицы*

*Проекты в lineEdit/spinBox*

void fillingInGroupBoxprojectsFields();

*удаление строки из таблицы Проекты*

void on\_pushButtonDeleteprojects\_clicked();

*добавление новой строки в таблицу*

*Проекты и перенос в нее данных из lineEdit/spinBox*

void on\_pushButtonAddprojects\_clicked();

*внесение изменений в строки таблицы*

*Проекты путем переноса значений из*

*lineEdit/spinBox*

void on\_pushButtonUpdateprojects\_clicked();

*вызов окна с информацией об авторе*

void on\_actionAbout\_author\_triggered();

*объект по работе с БД*

QSqlDatabase db;

*указатели, отвечающие за работу с SQL-моделью*

QSqlQueryModel \*queryModelprojects;

SqlQueryModel \*queryModelDataBase;

*переменные, хранящие номер строки, выделенной*

*пользователем в таблицах*

int selectedRowprojects;

void changeEvent (QEvent \*event);

*функция переключение языков*

void changeTranslator(QString postfix);

#endif // MAINWINDOW\_H

**6 Список файлов с исходным кодом и вспомогательных**

На рисунке 6.1 изображена вся файловая структура программы, включая вспомогательные файлы.

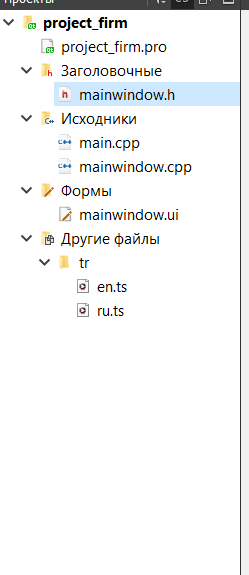


Рисунок 6.1 – Файловая структура программы

**ЗАКЛЮЧЕНИЕ**

Выполняя эту работу, были углублены навыки использования Qt Framework на языке C++, навыки работы с Qt Designer и перевода текстовых элементов интерфейса с помощью Qt Linguist. Были улучшены навыки программирования на языке C++ и использования PostgreSQL.

Данная программа может:

* Программа хранит данные в БД, умеет работать по сети с СУБД PostgreSQL. То есть должна:
* Подключается к СУБД.
* Отображает данные из таблиц и/или представлений.
* Добавляет данные в БД.
* Редактирует данные в БД.
* Удаляет данные из БД.

В БД для примера содержится 1000 записей, в каждой из них по 8 свойств. В созданном ПО имеется область для отображения и формочка. Основное меню программы представлено на главном окне, а в формочке можно редактировать или добавлять запись в БД. Нажав «О программе» в главном окне, можно узнать информацию о создателе программы и о самой программе. Программа имеет возможность менять язык на Русский и Английский.

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Сайт с документацией классов Qt. [Электронный ресурс] URL: https://doc.qt.io (дата обращения 18.05.2023)
2. Основы программирования на языках Си и C++ для начинающих [Электронный ресурс] URL: http://cppstudio.com (дата обращения 18.05.2023)
3. Форум программистов, системных администраторов, администраторов баз данных, компьютерный форум, форум по электронике и бытовой технике, обсуждение софта. [Электронный ресурс] URL: https://www.cyberforum.ru (дата обращения 21.05.2023)
4. Форум программистов и системных администраторов [Электронный ресурс] URL: https://stackoverflow.com (дата обращения 21.05.2023)