



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ Радиотехнический _____

КАФЕДРА _____ Системы обработки информации и управления _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:

**Автоматизированная система учета и анализа
телефонных контактов**

Студент **РТ5-81**
(Группа)

(Подпись, дата)

Житенев В.Г.
(И.О.Фамилия)

Руководитель ВКР

(Подпись, дата)

(И.О.Фамилия)

Консультант

(Подпись, дата)

(И.О.Фамилия)

Нормоконтролер

(Подпись, дата)

(И.О.Фамилия)

2019 г.

АННОТАЦИЯ

Квалификационная работа «Автоматизированная система учета и анализа телефонных контактов» посвящена разработке программного обеспечения для автоматизации работы с электронной записной книжкой. АСУ предназначена для автоматизации процесса управления и анализа данных. Главной особенностью проекта является возможность использования системы как на локальной базе данных, так и на удаленном сервере.

Расчётно-пояснительная записка включает в себя описание этапов и результаты разработки системы.

Конструкторская часть содержит обоснование и выбор технологий, программных средств, модели данных и архитектуры для реализации системы.

В технологической части представлены этапы разработки интерфейса пользователей системы и ключевые особенности проекта.

Расчётно-пояснительная записка содержит 81 страницу, 105 страниц с учётом 3 приложений:

- Приложение А. Техническое задание;
- Приложение Б. Программа и методика испытаний;
- Приложение В. Графическая часть;

Было использовано 11 источников, работа включает в себя 55 рисунков и 23 таблицы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. КОНСТРУКТОРСКАЯ ЧАСТЬ	7
1.1 Общетехническое обоснование разработки	7
1.1.1 Постановка задачи проектирования	7
1.1.2 Описание предметной области	8
1.1.3 Перечень процессов, подлежащих автоматизации.....	9
1.1.4 Выбор и обоснование критериев качества	10
1.1.5 Оценка критериев качества	11
1.2 Анализ аналогов и прототипов	14
1.3 Постановка задачи проектирования	15
1.4 Инфологическая модель	17
1.4.1 Описание инфологической модели предметной области	17
1.4.2 Описание сущностей и атрибутов	18
1.4.3 Связи между сущностями	20
1.5 Даталогическая модель.....	21
1.6 Структурная схема системы.....	24
1.7 Разработка архитектуры системы.....	27
1.7.1 Архитектура системы	27
1.7.2 Архитектура сети	30
1.8 Выбор оборудования.....	31
1.8.1 Выбор накопителя.....	31
1.8.2 Выбор сервера	35

1.9	Обоснование выбора программных средств	39
2	ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ.....	45
2.1	Разработка графа диалога.....	45
2.2	Разработка интерфейса пользователя	54
2.3	Краткий обзор классов.....	59
2.4	Ключевые особенности проектирования системы	61
2.4.1	Использование связанных списков	61
2.4.2	Использование пользовательских моделей отображения данных	63
2.4.3	Корректное формирование даты для запроса	66
2.4.4	Динамическое освобождение памяти при работе с формами	69
2.4.5	Система анализа	71
2.4.6	Работа с отдельными файлами разных форматов.....	75
	ЗАКЛЮЧЕНИЕ	80
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	81
	ПРИЛОЖЕНИЕ А Техническое задание.....	82
1	Наименование	85
2	Основание для разработки	85
3	Исполнитель	85
4	Назначение и цель разработки.....	85
5	Содержание работы	86
5.1	Задачи, подлежащие решению.....	86
5.2	Требования к функциональности программного изделия	86
2.4.7	Перечень процессов, подлежащих автоматизации.....	86
5.3	Требования к разрабатываемым компонентам	87
5.4	Требования к составу программного обеспечения.....	87

ПРИЛОЖЕНИЕ Б	Программа и методика испытаний	88
1	Объект испытаний.....	91
2	Цель испытаний.....	91
3	Требования к объекту испытаний	91
4	Требования к программной документации	91
5	Средства и порядок испытаний	92
6	Методика испытаний	92
ПРИЛОЖЕНИЕ В	Графическая часть	96

ВВЕДЕНИЕ

Прогресс не стоит на месте и на сегодняшний день трудно представить себе человека без мобильного телефона. Этот курс взят уже давно: на смену аналоговым телефонам приходят мобильные, а бумажные записные книги заменяются электронными, ведь учет данных в таком виде менее трудоемкий. Использование базы с хранимыми записными книжками открывает доступ к ряду возможностей, которые могут быть востребованы, например, в силовых структурах.

Основная цель работы силовых структур – соблюдение порядка в пределах территории государства. Расследование преступлений, выявление криминальных схем, а также предотвращение террористических актов – это лишь одни из ряда задач, стоящих перед данными ведомствами.

Применение проектируемой системы может существенно помочь в выполнении вышеописанных целей, благодаря функции анализа, суть которой заключается в выявлении связей между правонарушителем и людьми, задержанными ранее.

1. КОНСТРУКТОРСКАЯ ЧАСТЬ

1.1 Общетехническое обоснование разработки

1.1.1 Постановка задачи проектирования

Целью данной работы является проектирование и реализация программного обеспечения, основной задачей которого является ведение и анализ записных книжек.

Данное приложение предназначено для использования на персональных электронно-вычислительных машинах под управлением операционной системы семейства Microsoft Windows (XP, 7, 8 ,10) или семейства Linux.

Задачи проектирования могут быть сформулированы следующим образом:

- исследование предметной области;
- выбор и обоснование критериев качества
- сравнение аналогов и прототипов
- разработка структуры и архитектур системы
- разработка инфологической и даталогической модели базы данных
- разработка алгоритмов обработки информации
- разработка алгоритма-анализа записных книг;
- разработка алгоритма устранения дубликатов;
- разработка алгоритмов импорта и экспорта данных;
- выбор средства программной реализации;
- программная реализация;
- разработка графа диалога и экранных форм;
- разработка документации.

1.1.2 Описание предметной области

Система для учета и анализа телефонных номеров использовалась ранее в пределах одного силового ведомства с шифром «Записная книжка». Однако, ввиду устаревания этой системы, возникла потребность в ее модернизации и расширении функциональных возможностей. Ранее система состояла из множества рабочих мест, необъединенных в единую вычислительную сеть. Каждый пользователь работал с локальной базой, а обмен данными между офисами и рабочими станциями происходил с помощью переноса на внешних носителях. Именно поэтому востребована функция импорта и экспорта, а также возможности шифровки передаваемых файлов во избежание перехвата. Система новой топологии подразумевает собой работу множества пользователей с сервером базы данных в пределах ЛВС, однако взаимодействие между сетями старой и новой топологии все также остается на передачи данных на внешних носителях.

Проектируемая система имеет кодовый шифр «Записная книжка 2.0» и взаимодействует, как и ее предшественник, с записными книжками, хранимыми в базе. Записная книга представляет собой совокупность данных о ее владельце, его номерах телефона, а также списках контактов, привязанных к каждому из телефонов.

Особенностью проектирования базы данных является система отслеживания нарушений уникальности полей номеров телефонов, ведь у одного номера телефона не могут быть два владельца. Данная система не позволяет добавить номер, совпавший с одним из базы, при редактировании, добавлении и импорте записных книг.

Главной задачей АСУА «Записная книжка 2.0» является выполнение функции анализа записной книги, суть которой заключается в выявлении прямых и косвенных связей среди фигурантов базы. Данные связи представляют собой порядка трех различных вариантов знакомств

фигурантов: номер анализируемого находится в списке контактов другого фигуранта базы, в списке контактов анализируемого найден телефонный номер фигуранта базы и в случае, если номер контакта из записной книги анализируемого совпал с контактным номером фигуранта базы.

Взаимодействие элементов предметной области изображено на рисунке 1

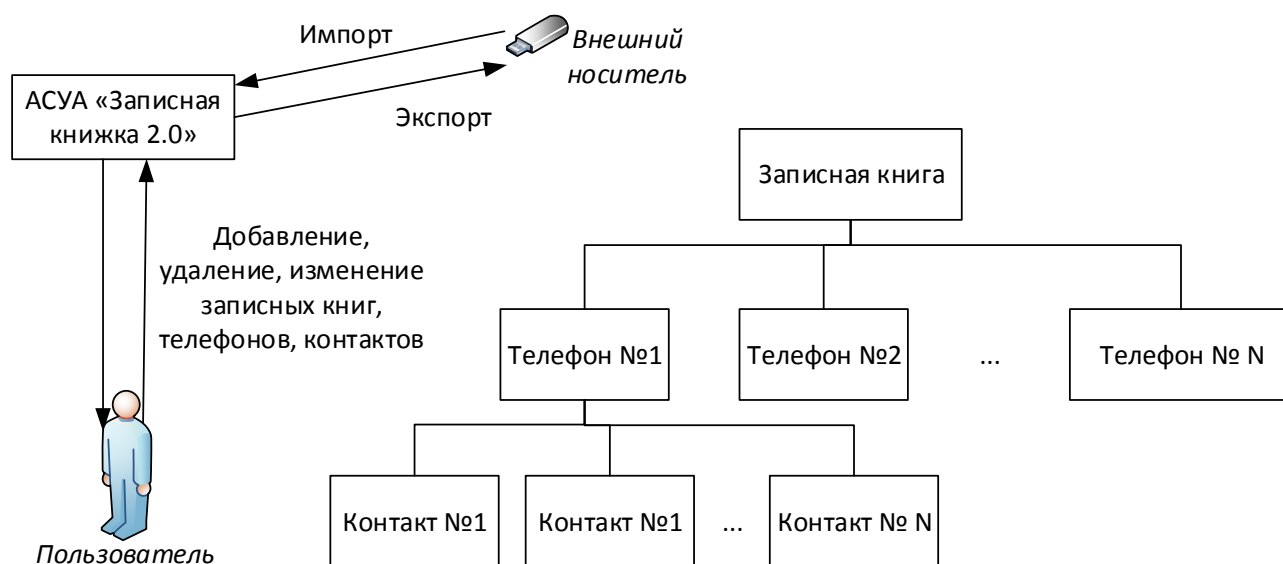


Рис. 1 Взаимодействие элементов предметной области

1.1.3 Перечень процессов, подлежащих автоматизации

1. добавление, редактирование и удаление записных книг;
2. добавление, редактирование и удаление телефонов;
3. добавление, редактирование и удаление контактов;
4. отображение данных;
5. анализ записных книг;
6. быстрый поиск по номеру записной книги;
7. расширенный поиск по реквизитам;

8. формирование справок по результатам поиска и анализа;
9. вывод информации о служебных телефонах на экран;
10. экспорт данных в файл;
11. шифрование экспортируемых данных;
12. импортирование данных из файла;

1.1.4 Выбор и обоснование критериев качества

При проектировании системы анализа данных важными являются следующие критерии качества:

1. время освоения;
2. скорость работы;
3. удобство работы;
4. функциональность;
5. трудозатраты на установку.

Время освоения

Пользователь должен быстро разобраться и освоить систему. Время освоения не должно превышать 24 часов.

Скорость работы

Скорость зависит от времени загрузки данных из файлов и выгрузки данных в файл, которое линейно зависит от объема перемещаемых данных

Удобство работы

Для эффективной работы пользователя в системе требуется наличие всей необходимой информации для выполнения задач на одном экране. Однако, при этом стоит избегать избыточности интерфейса – множество лишних окон и бессмысленной информации будет лишь путать пользователя.

Функциональность

Согласно проведенному анализу функций, подлежащих автоматизации, разрабатываемая система должна иметь основной функционал для анализа и изменения данных в базе. Наличие дополнительных функций приветствуется, однако это не является обязательным условием.

Трудозатраты на установку

Система должна быстро внедряться и быстро настраиваться до состояния полной готовности работы с ней.

1.1.5 Оценка критериев качества

По итогам вышесказанного, критериям качества можно присвоить следующие весовые коэффициенты, которые отражают в таблице 2.

Таблица 2. Критерии качества и их весовые коэффициенты.

Критерий	α
1. Время освоения.	0,15
2. Скорость работы.	0,3
3. Удобство работы.	0,2
4. Функциональность.	0,25
5. Трудозатраты на установку.	0,1

Выполнено следующее условие:

$$\sum \alpha_i = 0,15 + 0,3 + 0,2 + 0,25 + 0,1 = 1$$

Для расчета нормированного значения j-го варианта по i-ому критерию необходимо воспользоваться формулой (1.1): [7]

$$K_{ij} = \frac{x_{ij} - x_i^-}{x_i^+ - x_i^-}, \quad (1.1)$$

где x_{ij} - натуральное значение;

x_i^+ - максимальное значение;

x_i^- - минимальное значение.

Для расчета интегрального показателя необходимо воспользоваться формулой (1.2): [7]

$$K = \sum_{i=1}^m \alpha_i \cdot K_{ij}, \quad (1.2)$$

где m - количество критериев.

Оценка по критериям производится путём присуждения баллов в соответствии со шкалами (таблицы 3 – таблица 4):[7]

*Таблица 3. Перевод качественных оценок
в баллы для критерия
«Функциональность».*

<i>Лингвистическая оценка</i>	<i>Описание</i>	<i>Балл</i>
Отлично	Реализован механизм импорта и экспорта как зашифрованных, так и данных без пароля, а также анализа, расширенного и быстрого поиска данных в базе, настройки, изменения и отображения данных	1
Хорошо	Реализован механизм анализа, расширенного и быстрого поиска данных в базе, а также настройки, изменения и отображения данных	0,8
Удовл.	Реализован механизм быстрого поиска, настройки, изменения и отображения данных.	0,6
Не удовл.	Реализован только механизм изменения и отображения данных.	0,4

*Таблица 4. Перевод качественных оценок
в баллы для критерия «Удобство
пользовательского интерфейса».*

Оценка	Описание	Балл
Отлично	Интерфейс простой и удобный, интуитивно понятный	1
Хорошо	Интерфейс удобный, но неопытному пользователю нужна инструкция	0,8
Удовл.	Интерфейс не удобный, воспринимается приемлемо	0,6
Неудовл.	Интерфейс не удобный, сбивает с толку	0,4

*Таблица 5. Перевод качественных оценок
в баллы для критерия «Трудозатраты на
установку».*

Лингвистическая оценка	Описание	Балл
Отлично	Установка компонентов из установщика всех компонентов программы	1
Хорошо	Установка компонентов из установщика, требуется установка сторонних программ	0,8
Удовл.	Установка компонентов из установщика, требуются дополнительные настройки и установка сторонних программ	0,6
Не удовл.	Установка компонентов из установщика, требуются дополнительные настройки и установка сторонних программ, требуется компиляция	0,4

1.2 Анализ аналогов и прототипов

Аналогами проектируемой системы могут служить следующие системы:

- «Записная книжка». АСУА, использующаяся в пределах силовой структуры в сегодняшний день. Данное ПО морально и технически устарело, поскольку имеет существенные ограничения, не позволяющие его полноценному функционированию.
- Ручной учет данных в Microsoft Excel. Данный метод может использоваться в случае отсутствия специализированного ПО, так как функциональные особенности Microsoft Excel позволяют работать с таблицами данных. Однако функциональность данной альтернативы отстает от конкурентов.

Сравним аналоги с использованием оценок (таблица 6): [7]

Таблица.6. Сравнение аналогов с использованием оценок.

Критерий	«Записная книжка 2.0»	«Записная книжка»	Microsoft Excel
<i>Время освоения.</i>	Хорошо	Отлично	Удовлетворительно
<i>Скорость работы.</i>	Отлично	Хорошо	Хорошо
<i>Удобство работы.</i>	Отлично	Хорошо	Удовлетворительно
<i>Функциональность.</i>	Отлично	Хорошо	Удовлетворительно
<i>Трудозатраты на установку.</i>	Хорошо	Отлично	Отлично

Сравним аналоги с учетом весовых коэффициентов (табл. 7): [7]

Таблица 7. Сравнение аналогов с учетом
весовых коэффициентов.

Критерий	α	«Записная книжка 2.0»	«Записная книжка»	Microsoft Excel
<i>Время освоения</i>	0,15	0,8	1	0,6
<i>Скорость работы</i>	0,3	1	0,8	0,8
<i>Удобство работы</i>	0,2	1	0,8	0,6
<i>Функциональность</i>	0,25	1	0,8	0,6
<i>Время внедрения системы на производство</i>	0,1	0,8	1	1
Итого	1	<u>0,95</u>	0,85	0,7

Таким образом, разрабатываемая система «Записная книжка 2.0» является лучшей среди аналогов и оправдывает свое создание.

1.3 Постановка задачи проектирования

Автоматизированная система «Записная книжка 2.0» должна выполнять следующие основные функции:

1. Отображение связанных данных

Пользователю необходимо иметь перед глазами список всех имеющихся записных книг в базе. При нажатии на одну из книг, система выдаст перечень привязанных к выбранной книге телефонов, выбрав один из которых, пользователь сможет ознакомиться со списком контактов, который находится на телефоне.

2. Изменение данных

Должна быть реализована возможность добавления новых данных или изменения старых: редактирование информации о владельце записной книги, добавление, изменение и удаление привязанных телефонов и списков контактов.

3. Расширенный поиск по базе

Поиск осуществляется по следующим полям: номер записной книжки, номер телефона, привязанный к ней, фамилия, имя, отчество, дата рождения, адрес проживания владельца, дата формирования записной книги, а также поиск книг, сформированных за период.

4. Анализ записных книг.

Необходимо реализовать функцию анализа, суть которой заключается в поиске прямых и косвенных связей между владельцами записных книг, по результатам которого формируется справка. Под прямыми связями подразумеваются следующие варианты знакомства владельцев:

- Номер телефона, привязанный к анализируемой записной книге, найден в списке контактов, информация о владельце которого имеется в базе данных;
- В одном из списков контактов анализируемой записной книги найден номер, информация о владельце которого имеется в базе данных.

В свою очередь под косвенной связью понимается следующий вид знакомства:

- В списке контактов анализируемой записной книжки был найден телефон, который был так же обнаружен в списке контактов другой записной книги из базы.

5. Импорт и экспорт данных.

Должна быть реализована функция выгрузки данных из базы в локальный файл, зашифрованный паролем при необходимости. Также в свою очередь необходимо реализовать обратный процесс - загрузку данных из файла в базу.

6. Служебные телефоны

Необходимо реализовать функцию вывода на экран списка служебных телефонов

1.4 Инфологическая модель

Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки СУБД. Время, затраченное разработчиком на проектирование БД, обычно окупается высокой скоростью реализации проекта.

Для анализа предметной области необходимо выделить объекты представления предметной области, а также необходимо определить взаимосвязи между этими объектами. [2]

1.4.1 Описание инфологической модели предметной области

Инфологическая модель базы данных представляет собой модель предметной области, определяющая совокупность информационных объектов (сущностей), их атрибутов и отношений между объектами, динамику изменения предметной области, а также характер информационных потребностей пользователей. [2]

Сущность – это некоторая совокупность атрибутов предметной области, имеющая имя. Сущность – любой различимый объект, информацию о котором необходимо хранить в базе данных. [3]

Атрибут – это поименованная характеристика сущности. Атрибут принимает значение из некоторого множества, которое называют типом данных. Атрибуты используются для определения того, какая информация должна быть собрана о сущности. [3]

Ключ сущности – совокупность ее ключевых атрибутов. [3]

Связь – это символ, обозначающий отношение между сущностями. [3]

В результате анализа предметной области определены сущности, их атрибуты, взаимосвязь между ними и разработана инфологическая модель базы данных.

Для разработки инфологической модели была использована программа Visio 2010.

Схема инфологической модели представлена в графической части дипломной работы на листе «Инфологическая модель».

1.4.2 Описание сущностей и атрибутов

Выделены следующие сущности предметной области. Описание сущностей и атрибутов представлено в таблицах ниже. [2]

Условные обозначения: PK (primary key) – первичный ключ; FK (foreign key) – внешний ключ.

Описание сущностей представлено в таблицах 8 - 11

Таблица 8. Сущность «Записная книга»

Имя атрибута	Описание атрибута
<u>ID записной книги (PK)</u>	Идентифицирующий атрибут
Фамилия	Фамилия владельца записной книги
Имя	Имя владельца записной книги
Отчество	Отчество владельца записной книги
Дата рождения	Дата рождения владельца записной книги
Город регистрации	Город регистрации владельца записной книги
Улица регистрации	Улица регистрации владельца записной книги
Дом регистрации	Дом регистрации владельца записной книги
Корпус регистрации	Корпус регистрации владельца записной книги
Квартира регистрации	Квартира регистрации владельца записной книги

Таблица 8. Сущность «Записная книга»

Имя атрибута	Описание атрибута
Город проживания	Город проживания владельца записной книги
Улица проживания	Улица проживания владельца записной книги
Дом проживания	Дом проживания владельца записной книги
Корпус проживания	Корпус проживания владельца записной книги
Квартира проживания	Квартира проживания владельца записной книги
Дата добавления	Дата добавления записной книги
Время добавления	Время добавления записной книги
Дополнительная информация	Дополнительная информация о владельце записной книги
Проверяется в интересах	Наименование службы в интересах которой проверяется владелец записной книги

Таблица 9. Сущность «Телефон»

Имя атрибута	Описание атрибута
<u>ID телефона (PK)</u>	Идентифицирующий атрибут
Номер телефона	Номер телефона, принадлежащий владельцу записной книги

Таблица 10. Сущность «Контакты»

Имя атрибута	Описание атрибута
<u>ID контакта (PK)</u>	Идентифицирующий атрибут
Номер телефона	Номер телефона контакта
Пометка контакта	Пометка, с которой записан телефонный номер в списке контактов

Таблица 11. Сущность «Служебные телефоны»

Имя атрибута	Описание атрибута
<u>ID служебного телефона</u>	Идентифицирующий атрибут
Номер служебного телефона	Телефонный номер служебного телефона
Название службы	Номер телефона контакта

1.4.3 Связи между сущностями

Рассмотрим связи между сущностями, выделенными и описанными выше (таблица 11):

Таблица 12. Описание связей между сущностями

Родительская сущность	Дочерняя сущность	Описание	Тип связи
Записная книга	Телефоны	К записной книге привязаны уникальные номера телефонов	1:M
Телефоны	Контакты	Телефонный номер содержит список контактов	1:M

Схема инфологической модели базы данных представлена на рис.13.

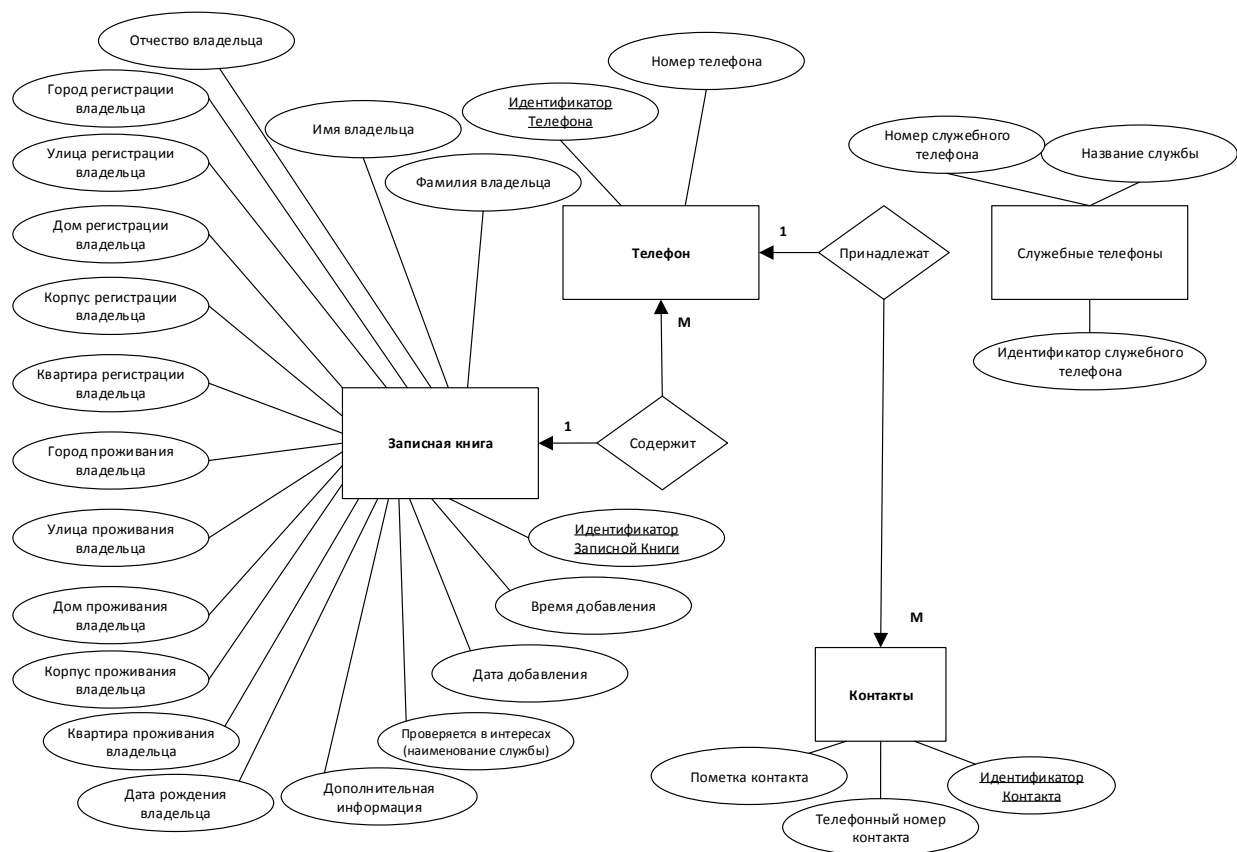


Рис. 13 Инфологическая модель базы данных

1.5 Даталогическая модель

Результатом даталогической модели является концептуальная схема БД, которая включает определение всех информационных единиц и связей, в том числе задание типов, характеристик и имен. [5]

В качестве СУБД для создания даталогической схемы базы данных была выбрана СУБД PostgreSQL. [8]

Описание таблиц представлено в таблицах 14 – 11

Таблица 14. Таблица «zk» (Записная книга)

Название поля	Типы данных	Ограничения
zk_id	INTEGER	PK, INCREMENT
lastname	VARCHAR(20)	NOT NULL
name	VARCHAR(20)	NOT NULL
mid_name	VARCHAR(20)	NOT NULL
check_for	VARCHAR(30)	
dop_info	VARCHAR(50)	
reg_city	VARCHAR(30)	
reg_street	VARCHAR(30)	
reg_home	VARCHAR(5)	
reg_corp	VARCHAR(5)	
reg_flat	VARCHAR(5)	
liv_city	VARCHAR(30)	
liv_street	VARCHAR(30)	
liv_home	VARCHAR(5)	
liv_corp	VARCHAR(5)	
liv_flat	VARCHAR(5)	
birth_date	DATE	NOT NULL
date_add	DATE	NOT NULL
time_add	VARCHAR(10)	NOT NULL

Таблица 15. Сущность «owners_tel» (Телефон)

Имя поля	Типы данных	Ограничения
telephone_id	INTEGER	PK, INCREMENT
telephone_num	VARCHAR(15)	NOT NULL, UNIQUE
fk_telephone_zk	INTEGER	FK, NOT NULL

Таблица 16. Сущность «contacts» (Контакты)

Имя поля	Типы данных	Ограничения
contact_list_id	INTEGER	PK, INCREMENT
cl_telephone	VARCHAR(15)	NOT NULL
cl_info	VARCHAR(50)	
fk_cl_telephone	INTEGER	FK, NOT NULL

Таблица 17. Сущность «off_tels» (Служебные телефоны)

Имя поля	Типы данных	Ограничения
of_t_id	INTEGER	PK, NOT NULL, UNIQUE
tel_num	VARCHAR(15)	NOT NULL, UNIQUE
service_name	VARCHAR(50)	NOT NULL

Для разработки даталогической модели была использована программа pgAdmin III. Схема даталогической модели представлена на рис. 18.

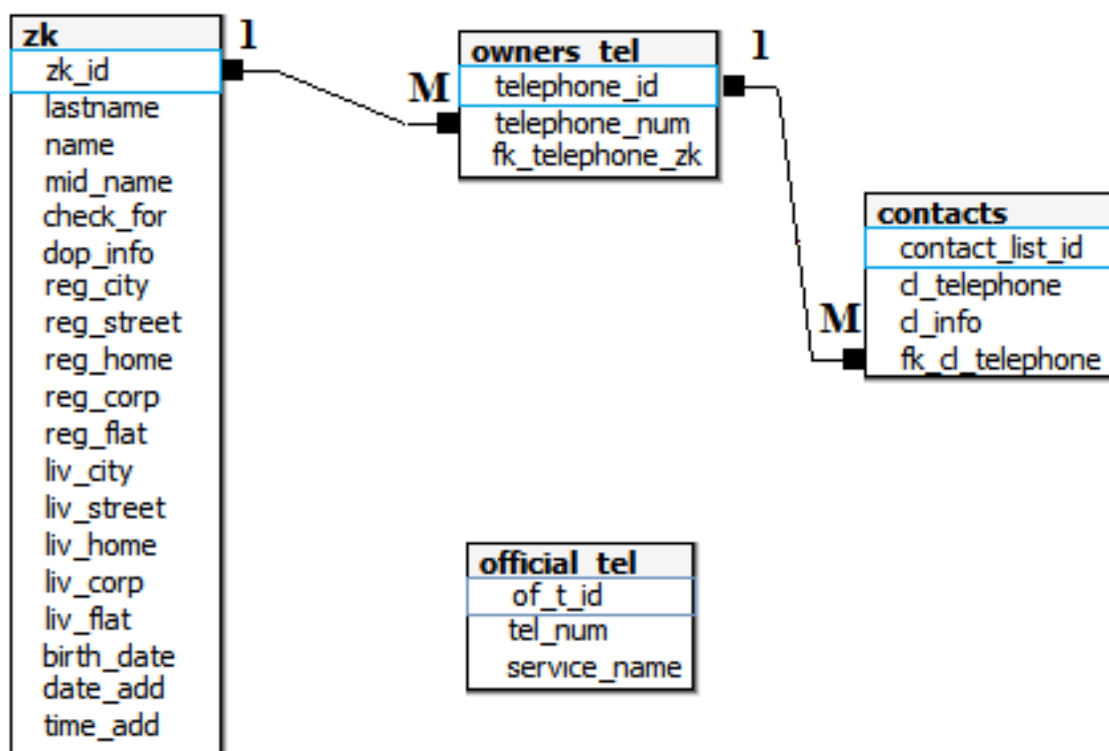


Рис. 18. Даталогическая модель базы данных

1.6 Структурная схема системы

Разрабатываемая система «Записная книжка 2.0» имеет модульную структуру. Количество и состав модулей зависит от требований к системе. Данная система состоит из 4 подсистем и 14 модулей (рис. 19):

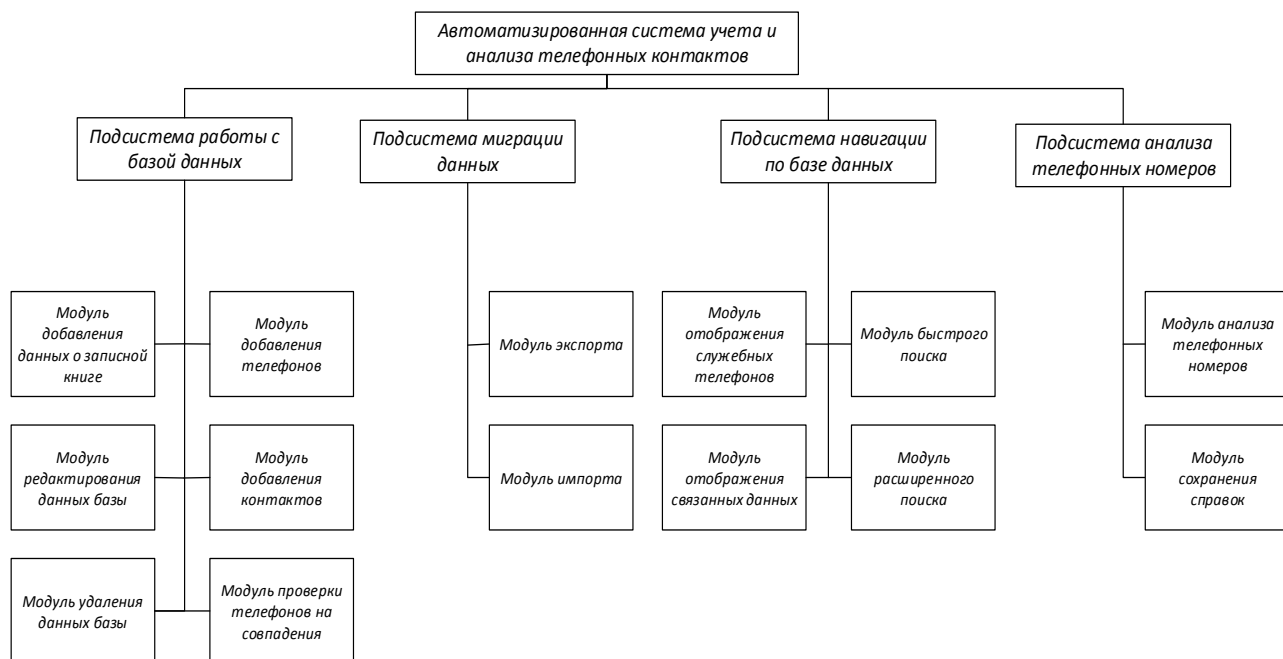


Рисунок 19. Структурная схема системы

Подсистема работы с базой данных

На данный компонент системы возлагаются функции по хранению информации. Сама подсистема состоит из шести модулей:

Модуль добавления данных о записной книге (позволяет добавлять новые записные книги);

Модуль добавления телефонов (позволяет создавать списки телефонных номеров во время добавления или редактирования записной книги);

Модуль добавления контактов (позволяет создавать списки контактов, привязанных к телефонным номерам, во время добавления или редактирования записной книги);

Модуль редактирования данных базы (позволяет изменять уже имеющиеся записные книги, привязанные телефонные номера, а также списки контактов).

Модуль удаления данных базы (позволяет удалять контакты, телефонные номера или записные книги).

Модуль проверки телефонов на совпадения (позволяет избежать ошибки нарушения уникальности полей телефонных номеров)

Подсистема миграции данных

На данный компонент системы возлагаются функции по загрузке и

выгрузке данных из базы. Сама подсистема состоит из двух модулей:

Модуль экспорта (служит для выгрузки данных из удаленной базы в файл на персональном компьютере. При необходимости файл может быть зашифрован паролем);

Модуль импорта (служит для загрузки данных из файла. Если файл был зашифрован, то пользователь увидит соответствующее сообщение).

Подсистема навигации по базе данных

На данный компонент системы возлагаются функции поиска и отображения данных. Сама подсистема состоит из четырёх модулей:

Модуль отображения связанных данных (служит для загрузки выбранных пользователем данных из базы);

Модуль отображения служебных телефонов (отвечает за отображения списка служебных телефонов, имеющих в базе).

Модуль быстрого поиска (служит для отображения данных по поисковому запросу с номером записной книги)

Модуль расширенного поиска (отвечает за формирование поискового запроса по реквизитам, составлении справки с результатами поиска или их отображением на главном окне)

Подсистема анализа телефонных номеров

На данный компонент системы возлагаются функции анализа телефонных номеров и формирование справок. Сама подсистема состоит из двух модулей:

Модуль анализа телефонных номеров (отвечает за поиск прямых и косвенных связей между фигурантами базы данных)

Модуль сохранения справок (отвечает за сохранение сформированных справок в текстовом формате)

1.7 Разработка архитектуры системы

1.7.1 Архитектура системы

На текущий момент система работает в локальном режиме с переносом данных на внешних носителях. Согласно требованиям технического задания, необходимо расширить данную сеть до локальной с применением клиент серверной технологии. Пользователи имеют возможность одновременно заходить в систему в разных, защищенных друг от друга сессиях сервера.

В роли клиента может выступать любой ПК, обладающий как минимум 150 Мб свободной оперативной памяти, так как на нем выполняются операции по сбору и временному хранению данных.

Для реализации системы был представлен вариант архитектуры, который включает в себя следующие компоненты:

- клиент;
- сервер БД.

Во время работы с программой, в большинстве выполняемых функций, будут формироваться запросы к базе данных, расположенной на персональном компьютере пользователя, а их результаты обрабатываться на персональном компьютере клиента. Также на стороне клиента могут выполняться процессы экспорта, импорта, изменения настроек и сохранения отчета по результатам поиска или анализа.

Схема взаимодействия с источниками данных представлена на рисунке (рис. 20):

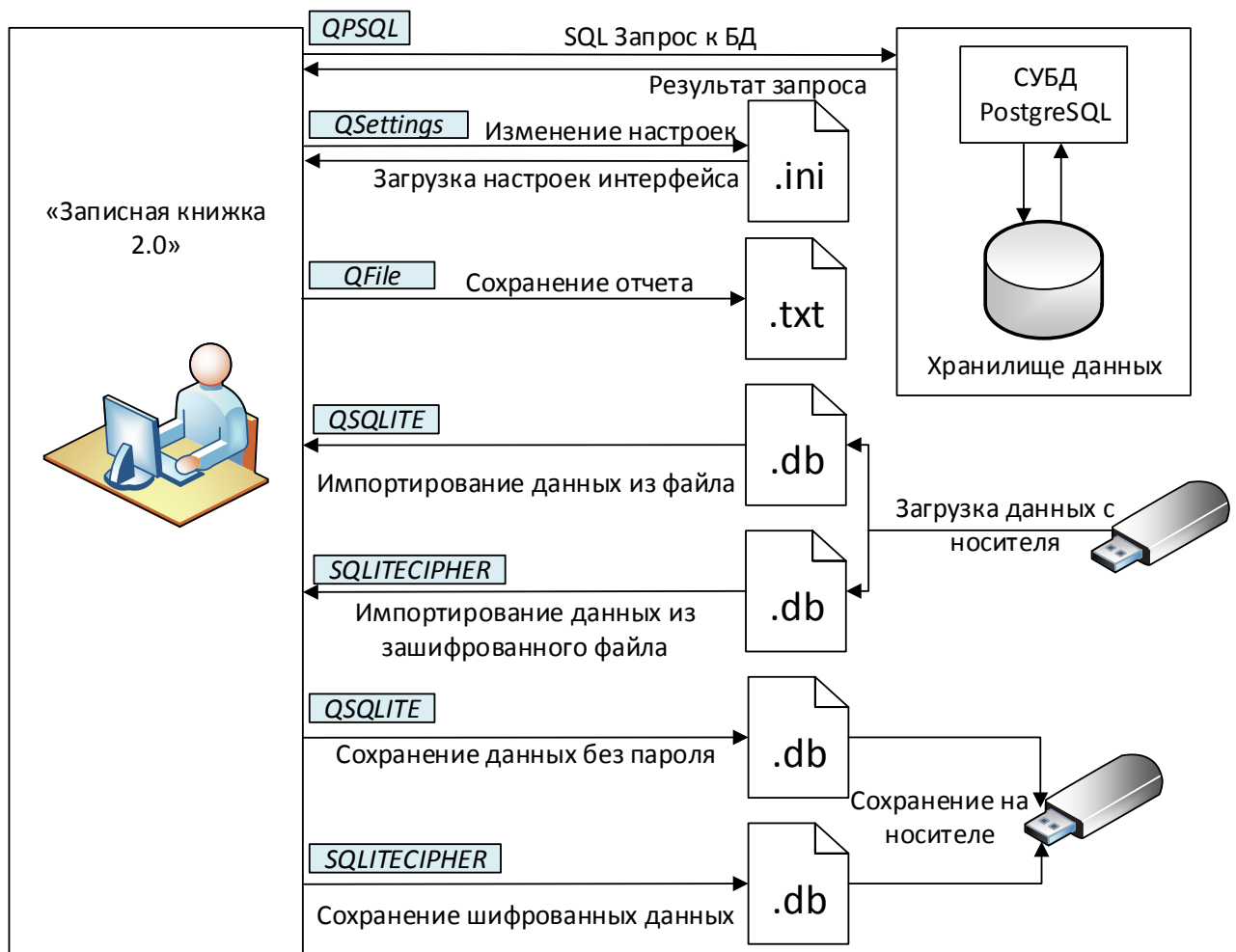
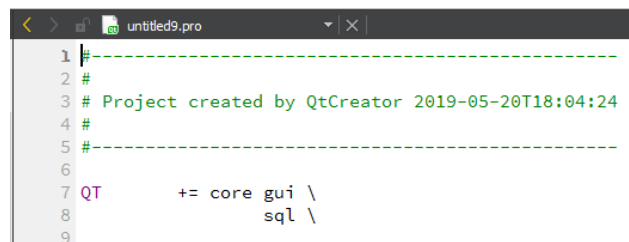


Рис. 20. Схема взаимодействия с источниками данных.

Для эффективного взаимодействия элементов системы используются следующие драйверы и библиотеки:

- **QPSQL**

Драйвер для работы с СУБД на PostgreSQL. Для подключения с данным типом драйвера необходимо указать адрес хоста, номер порта, название базы данных, имя пользователя и пароль при необходимости. В рамках данной системы нет необходимости разграничивать права доступа у клиентов системы, однако при появлении таковой потребности сделать это не составит труда. Данные для подключения изменяются в файле настроек и используются в качестве по умолчанию при последующем запуске программы. Подключение QPSQL осуществляется с помощью добавления модуля `sql` в проектный файл с расширением `.pro` во время проектировки данной системы в Qt.



```

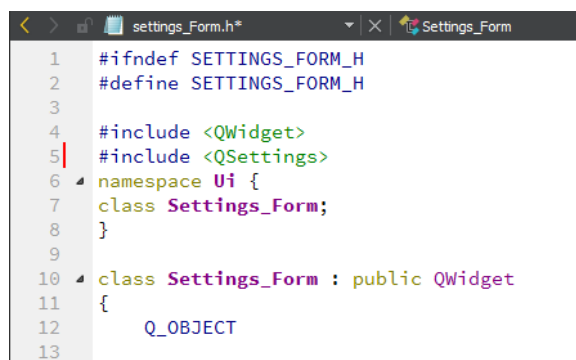
1 #-----
2 #
3 # Project created by QtCreator 2019-05-20T18:04:24
4 #
5 #-----
6
7 QT       += core gui \
8           sql \
9

```

Рис. 21 Подключение модуля sql во время проектировки в Qt.

- *QSettings*

Библиотека Qt QSettings предназначена для создания и изменения файлов платформонезависимых настроек приложений. Подключение происходит во время проектировки классов, обращающихся к методам данной библиотеки.



```

1 #ifndef SETTINGS_FORM_H
2 #define SETTINGS_FORM_H
3
4 #include <QWidget>
5 #include <QSettings>
6 namespace Ui {
7     class Settings_Form;
8 }
9
10 class Settings_Form : public QWidget
11 {
12     Q_OBJECT
13

```

Рис. 22 Подключение библиотеки QSettings в определении класса формы изменения настроек.

1. *QFile*

Библиотека Qt QFile содержит методы для работы с файлами: открытия, закрытия, чтения и записи данных. В рамках проектируемой системы, класс QFile применяется для сохранения справки в текстовом формате. Подключение происходит во время проектировки классов, обращающихся к методам данной библиотеки

```

1  #ifndef AN_RESULT_H
2  #define AN_RESULT_H
3  #include <QFile>
4  #include <QWidget>
5  #include <QDate>
6  #include "for_analysis.h"
7
8
9  namespace Ui {
10     class An_result;
11 }
12
13 class An_result : public QWidget
14 {
15     Q_OBJECT
16

```

Рис. 23 Подключение библиотеки QFile в определении класса формы отображения результатов поиска и анализа.

2. *QSQLITE*

Драйвер для работы с файлами базы данных на SQLite. При подключении данный драйвер не требует данных, достаточно иметь путь к файлу в формате .db, расположенному на персональном компьютере клиента или внешнем носителе. Подключение QSQLITE аналогично подключению любого другого драйвера базы данных: в файле проектирования указывается модуль sql.

3. *QSQLITECIPHER*

Не входит в список стандартных драйверов баз данных, предлагаемых для установки при работе с Qt. Выполняет функцию подобную QSQLITE – работа с SQLite, однако с возможностью использования пароля для шифровки данных. Пароль вводит клиент во время работы с системой при экспорте или импорте данных. Несмотря на необходимость сторонней установки QSQLITECIPHER, подключается он аналогично остальным драйверам.

1.7.2 Архитектура сети

Каждый пользователь АСУА «Записная книжка 2.0» в пределах офисов работает с базой данных, расположенной на персональных компьютерах. Обмен данными осуществляется с помощью сохранения экспортируемых данных в файл, перемещении файла на внешний носитель и передача его к другой рабочей

станции. Такая топология имеет одно преимущество – низкая вероятность перехвата конфиденциальных данных

Работники головного офиса используют усовершенствованную технологию, позволяющую работать с удаленным сервером в пределах одной ЛВС. Каждый участник сети также может взаимодействовать с другими офисами по принципу старой топологии – обмен данными с использованием внешних носителей.

Сетевая архитектура системы в пределах головного офиса строится на базе стандарта Ethernet 100 Base-TX. Этот стандарт является оптимальным по соотношению цена/производительность. Пропускная способность каналов связи составляет 100 Мбит/с. Схема взаимодействия рабочих станций представлена на рисунке 24:

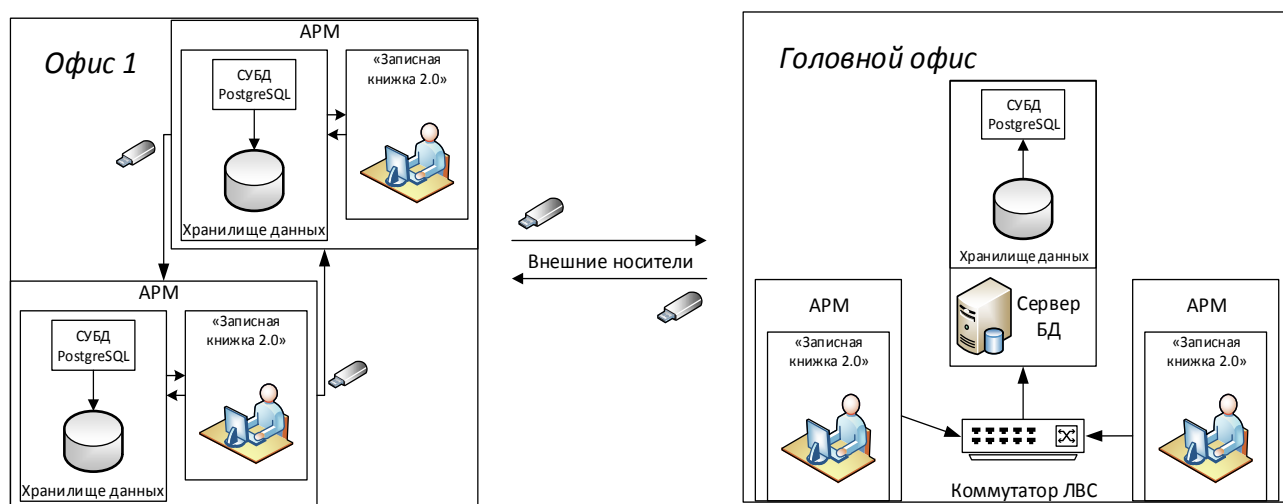


Рисунок 24. Схема взаимодействия рабочих станций «Записной книжки 2.0»

1.8 Выбор оборудования

1.8.1 Выбор накопителя

Главным звеном в системе взаимодействия рабочих станций является внешний носитель. На сегодняшний день дискеты и оптические диски сильно уступают по многим параметрам USB-флеш-накопителям, именно поэтому

только они заслуживают детального сравнения. В качестве вариантов USB-флеш-накопителей рассмотрим следующие: SANDISK Cruzer Fit, Transed Jetflash 790, Transed Jetflash 300

Проведем численную оценку вариантов накопителей. Для оценки будем использовать следующие критерии:

- 1) Стоимость
- 2) Объем
- 3) Протокол
- 4) Устойчивость к внешним воздействиям

Дадим качественную и количественную характеристику по перечисленным выше критериям в таблице 25

Таблица 25. Качественные характеристики

Система Характеристика	SANDISK Cruzer Fit	Transed Jetflash 790	Transed Jetflash 300
Стоимость	260 р.	360 р.	350 р.
Объем	16 Gb	16 Gb	8 Gb
Протокол	USB2.0	USB3.0	USB2.0
Устойчивость к внешним воздействиям	Отл.	Удовл.	Хор.

Переведем качественное значение характеристик в количественное значение в соответствии со шкалой перевода (таблица 26).

Таблица 26. Шкала перевода качественных характеристик в количественные.

Качественные параметры	Удовл.	Хор	Отл
Количественные параметры	0,60	0,80	1

Получаем таблицу характеристик, которые имеют количественное значение (таблица 27).

Таблица 27. Таблица характеристик с количественными значениями.

Система Характеристика	SANDISK Cruzer Fit	Transed Jetflash 790	Transed Jetflash 300
Стоимость	1	0.8.	0.8.
Объем	1	1	0.8
Протокол	0.8	1	0.8
Устойчивость к внешним воздействиям	1	0.6	0.8

Проведем оценку на основании нормированных значений с учетом весового коэффициента. Весовой коэффициент выберем в соответствии с актуальностью характеристики.

Весовой коэффициент для характеристики «Стоимость» возьмем равным 0,15, так как она является наименее важной на фоне остальных. Приобретение

нового накопителя оправдано лишь поломкой старого или его утратой. Поэтому аккуратное пользование позволит сэкономить, ведь пользователь заплатит небольшую цену единожды.

«Объем» является куда более важной характеристикой. Несмотря на то, что объем файлов базы и справок не отличается большим размером, возьмем коэффициент равным 0.25.

«Протокол», пожалуй, одна из наиболее важных характеристик, так как она отвечает за скорость записи информации. Для нее возьмем коэффициент равный 0.3.

В связи с тем, что накопитель выполняет функцию транспортировки информации между пользователями и офисами, характеристика «устойчивость к внешним воздействиям» имеет высокий приоритет. Для нее возьмем коэффициент равный 0.3.

Нормированные значения критериев и весовые коэффициенты представлены в таблице 28.

Таблица 28. Таблица нормированных значений.

Система Характеристика	Весовой коэффициент	SANDISK Cruzer Fit	Transed Jetflash 790	Transed Jetflash 300
Количество ядер*частота	0.15	1.	0.8.	0.8.
Объем оперативной памяти	0.25	1	1	0.8
Объем жесткого диска	0.3	0.8	1	0.8
Стоимость	0.3	1	0.6	0.8
ИТОГ		0,94	0,85	0,85

Расчет:

1) SANDISK Cruzer Fit

$$0.15*1+0.25*1+0.3*0.8+0.3*1=0,94$$

2) Transed Jetflash 790

$$0.15*0.8+0.25*1+0.3*1+0.3*0.6=0,85$$

3) Transed Jetflash 300

$$0.15*0.8+0.25*0.8+0.3*0.8+0.3*0.8=0,85$$

По результатам расчета предпочтительным USB-флеш-накопителем является SANDISK Cruzer Fit.

1.8.2 Выбор сервера

Выберем сервер для ЛВС в головном офисе из нескольких вариантов, которые удовлетворяют требуемым характеристикам. В качестве вариантов серверов рассмотрим следующие: Dell PowerEdge T140, HP ProLiant ML30, Dell PowerEdge T130

Проведем численную оценку вариантов серверов. Для оценки будем использовать следующие критерии:

- 1) количество ядер * частота процессора;
- 2) объем оперативной памяти;
- 3) объем жесткого диска;
- 4) стоимость.

Дадим качественную и количественную характеристику по перечисленным выше критериям (таблица 29).

Таблица 29. Качественные характеристики

Система Характеристика	HP ProLiant ML30	Dell PowerEdge T140	Dell PowerEdge T130
Количество ядер*частота	1*3,3 ГГц	1*3,5 ГГц	1*3,8 ГГц
Объем оперативной памяти	8 Gb	8 Gb	16 Gb
Объем жесткого диска	2 ТБ	1 ТБ	1 ТБ
Стоимость	86 232 р.	93 647 р.	111 228 р.

Переведем качественное значение характеристик в количественное значение в соответствии со шкалой перевода (таблица 30).

Таблица 30. Шкала перевода качественных характеристик в количественные.

Качественные параметры	Удовл.	Хор	Отл
Количественные параметры	0,60	0,80	1

Получаем таблицу характеристик, которые имеют количественное значение (таблица 31).

*Таблица 31. Таблица характеристик с
количественными значениями.*

Система Характеристика	HP ProLiant ML30	Dell PowerEdge T140	Dell PowerEdge T130
Количество ядер*частота	0.8	0.8	1
Объем оперативной памяти	0.8	0.8	1
Объем жесткого диска	1	0.8	0.8
Стоимость	1	0.8	0.6

Проведем оценку на основании нормированных значений с учетом весового коэффициента. Весовой коэффициент выберем в соответствии с актуальностью характеристики.

Весовой коэффициент для характеристики «Объем оперативной памяти» возьмем равным 0.3, так как для системы важна производительность сервера, ведь чем больше клиентов в сети, тем большее количество запросов приходится обрабатывать. Данный показатель влияет напрямую на производительность сервера.

«Количество ядер*частота» является второй характеристикой, влияющей на производительность, но менее важной. Для нее коэффициент возьмем равным 0.25.

В связи с тем, что на сервере будет обрабатываться информация в больших объемах, весовой коэффициент для характеристики «Объем жесткого диска» берем равным 0.3

Необходимо понимать, что большие растраты на оборудование не

рациональны. Однако, если стоит выбор между стоимостью и производительностью, то в реалиях крупной организации потребность в качественном выполнении большого объема задач будет в приоритете. Именно поэтому для характеристики «стоимость» весовой коэффициент получает значение равное 0.15

Нормированные значения критериев и весовые коэффициенты представлены в таблице 32.

Таблица 32. Таблица нормированных значений.

Система Характеристика	Весовой коэффициент	HP ProLiant ML30	Dell PowerEdge T140	Dell PowerEdge T130
Количество ядер*частота	0.3	0.3	0.8	0.8
Объем оперативной памяти	0.25	0.25	0.8	0.8
Объем жесткого диска	0.3	0.3	1	0.8
Стоимость	0.15	0.15	1	0.8
ИТОГ		0.89	0.8	0.88

Расчет:

1) HP ProLiant ML30

$$0.8*0.3+0.8*0.25+0.3*1+1*0.15=\mathbf{0.89}$$

2) Dell PowerEdge T140

$$0.8*0.3+0.8*0.25+0.8*0.3+0.8*0.15=0.8$$

3) Dell PowerEdge T130

$$1*0.3+1*0.25+0.8*0.3+0.6*0.15=0.88$$

По результатам расчета предпочтительным сервером для реализации системы «Записная книжка 2.0» является сервер HP ProLiant ML30.

1.9 Обоснование выбора программных средств

Обоснованию выбора подлежат следующие программные средства:

1. Среда разработки и язык программирования
2. Система управления базами данных.
3. Средства документации

Среда разработки и язык программирования

Перед проектированием АСУА «Записная книжка 2.0» появилась задача выбора среды разработки и языка программирования. После анализа всех доступных фреймворков, выбор пал на Qt и C++, так как они обладают рядом внушительных преимуществ.

Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования. [11]

Объектно-ориентированный подход в программировании позволяет решать задачи на уровне моделирования информационных объектов. C++ обладает интересными особенностями и открывает широкий спектр возможностей. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Один из базовых принципов C++ — «не платишь за то, что не

используешь» — то есть ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования. Имеется возможность работы с памятью на низком уровне. [1]

Исходя из вышеописанных особенностей выбранных компонентов проектирования, было принято решение использовать среду разработки Qt и C++ в качестве языка программирования.

Система управления базами данных

Так как старая версия АСУА не поддерживала работу в ЛВС, все рабочие станции взаимодействовали с локальными базами данных. Обновление «Записная книжка 2.0» должно расширить функциональные возможности старой версии. Для работы с БД в пределах ЛВС необходимо выбрать СУБД с целостным хранением данных, а также поддержкой их хранения в большом объеме. Исходя из этих потребностей, была выбрана СУБД PostgreSQL.

Созданный с использованием объектно-реляционной модели, PostgreSQL поддерживает сложные структуры и широкий спектр встроенных и определяемых пользователем типов данных. PostgreSQL обеспечивает расширенную ёмкость данных и заслужил доверие бережным отношением к целостности данных. [8]

Средства документации

Визуализация данных – отличный способ быстро и ёмко донести информацию до читающего. Именно поэтому в ходе разработки сложных проектов приходится прибегать к сторонним средствам визуализации программного кода и составления документации. Во время создания системы «Записная книжка 2.0» выбор пал на использование программ Umbrello Project и Doxygen.

Umbrello Project

Так как одной из главных особенностей объектно-ориентированного программирования является наследование, то программы, разрабатываемые на этих языках, обладают очень сложной структурой. Создание пользовательских заголовочных файлов и их подключение к другим файлам проекта используется

чаще наследования, но также усложняет структуру разрабатываемой системы. Для того чтобы иметь четкое представление о зависимостях и наследованиях пользовательских классов, был разработан графический язык моделирования UML.

UML призван поддерживать процесс моделирования программных систем на основе объектно-ориентированного подхода, организовывать взаимосвязь концептуальных и программных понятий, отражать проблемы масштабирования сложных систем. Модели на UML используются на всех этапах жизненного цикла программной системы, начиная с бизнес-анализа и заканчивая сопровождением системы. Разные организации могут применять UML по своему усмотрению в зависимости от своих проблемных областей и используемых технологий.

Рынок обладает большим количеством предлагаемых проектных решений для разработки UML диаграмм. Однако на фоне всех остальных уже многие годы выделяется Umbrello Project по следующим причинам:

- Кроссплатформенность
- Функция экспортирования. Достаточно подключить к Umbrello необходимую папку, и программа построит автоматически UML диаграмму зависимостей, основываясь на исходном коде.
- Поддержка широкого списка языков программирования.

Помимо этого, визуализация классов в Umbrello является достаточно наглядной. Пример отображения класса `An_result` в среде программы Umbrello Project можно увидеть на рисунке 33

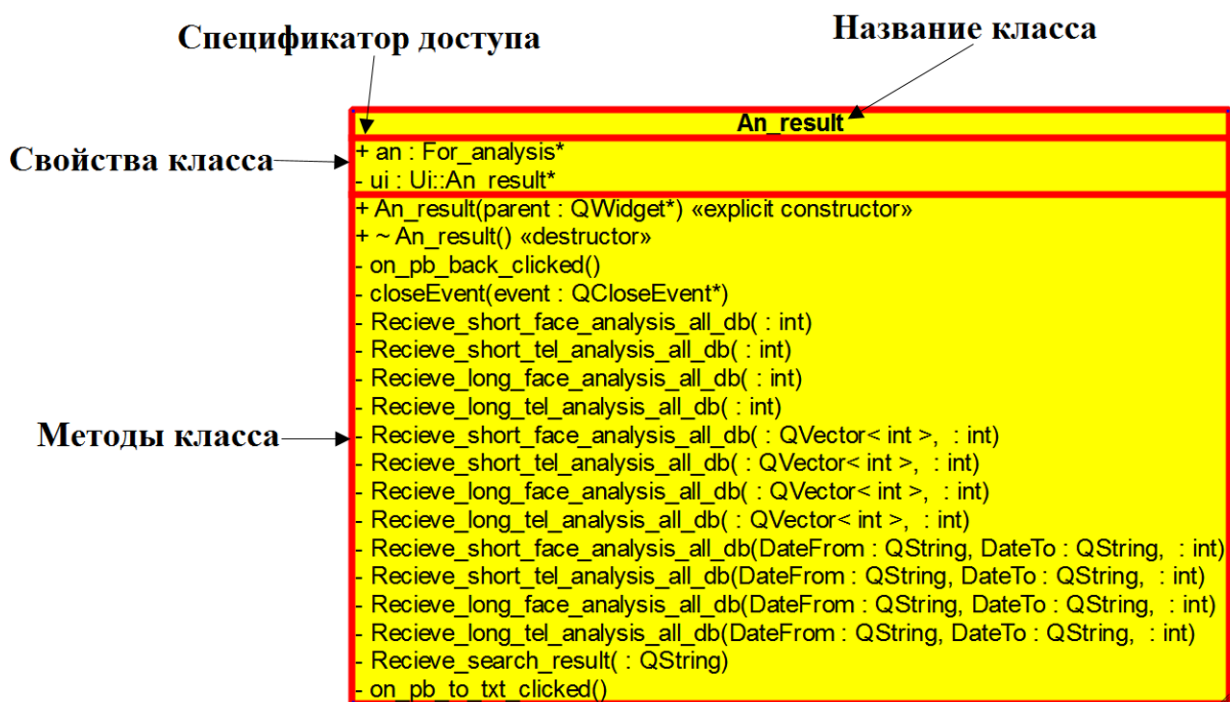


Рис 33. Отображение класса *An_reult* в программе *Umbrello Project*

Класс представлен в нотации UML. Особое внимание следует обратить на обозначение спецификаторов доступа:

- + - свойство или метод находится в public
- - - свойство или метод находится в private
- # - свойство или метод находится в private

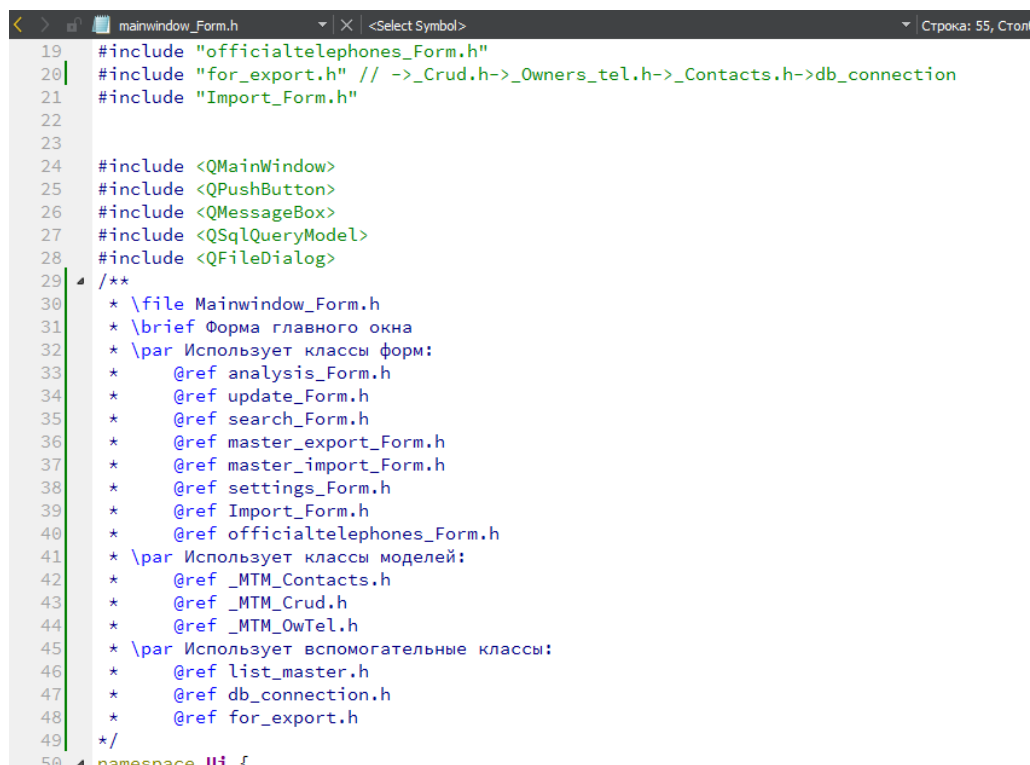
Основываясь на вышеописанных особенностях, для визуализации классов системы «Записная книжка 2.0» было выбрано данное программное средство

Doxygen

Комментирование кода — является не только хорошим тоном программирования, но и необходимостью в случае разработки крупных проектов, основанных на объектно-ориентированном подходе. Особенность этого подхода — разделение крупных проектов на небольшие задачи, что позволяет подключить большее число программистов. Отсюда вытекает следующая проблема: каждый человек, работающий над проектом, имеет свои субъективные взгляды на написание кода, что препятствует пониманию

синтаксиса даже в пределах некрупных задач. Наличие комментариев упрощает понимание, однако в случае крупных проектов с существенным количеством заимствований и наследований, программист все так же рискует потерять драгоценное время из-за отсутствия наглядной документации по разрабатываемой системе. Именно в таких случаях на помощь приходит система автоматической документации.

Так как данный проект разрабатывался на языке программирования C++, выбор программы-генератора документации был не велик. Doxygen является лучшим выбором ввиду простоты реализации и настройки. Для того, чтобы получить справку по разрабатываемому проекту в формате HTML, достаточно лишь указать путь к файлам проекта и придерживаться определенных правил комментирования кода. Пример синтаксиса комментариев Doxygen и результата составления документации приведен на рисунках 34 и 35



```
19 #include "officialtelephones_Form.h"
20 #include "for_export.h" // ->_Crud.h->_Owners_tel.h->_Contacts.h->db_connection
21 #include "Import_Form.h"
22
23
24 #include <QMainWindow>
25 #include <QPushButton>
26 #include <QMessageBox>
27 #include <QSqlQueryModel>
28 #include <QFileDialog>
29 /**
30  * \file Mainwindow_Form.h
31  * \brief Форма главного окна
32  * \par Использует классы форм:
33  *   @ref analysis_Form.h
34  *   @ref update_Form.h
35  *   @ref search_Form.h
36  *   @ref master_export_Form.h
37  *   @ref master_import_Form.h
38  *   @ref settings_Form.h
39  *   @ref Import_Form.h
40  *   @ref officialtelephones_Form.h
41  * \par Использует классы моделей:
42  *   @ref _MTM_Contacts.h
43  *   @ref _MTM_Crud.h
44  *   @ref _MTM_OwTel.h
45  * \par Использует вспомогательные классы:
46  *   @ref list_master.h
47  *   @ref db_connection.h
48  *   @ref for_export.h
49 */
50 namespace UI {
```

Рис.34 Пример применения документирования в нотации Doxygen

Detailed Description

Форма главного окна

Использует классы форм:

`analysis_Form.h` `update_Form.h` `search_Form.h` `master_export_Form.h` `master_import_Form.h` `settings_Form.h` `Import_Form.h` `officialtelephones_Form.h`

Использует классы моделей:

`_MTM_Contacts.h` `_MTM_Crud.h` `_MTM_OwTel.h`

Использует вспомогательные классы:

`list_master.h` `db_connection.h` `for_export.h`

Рис.35 Результат документирования в отчете Doxygen

Такой вид генерирования документации с использованием Doxygen имеет большой ряд преимуществ на фоне обычного комментирования кода. Время, затрачиваемое на понимание работы основных методов и взаимодействий классов, сокращается в разы, так как просмотр HTML-документации более нагляден чем чтение обычных комментариев в коде. [6]

Основываясь на вышесказанном, было принято решение использовать программу Doxygen как средство автоматического генерирования документации. [6]

2 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

2.1 Разработка графа диалога

ВВОДНЫЕ СЛОВА

ЧЕТКО В ФОРМАТЕ ЕСЛИ ТО

Пользователь запускает «Записная книга 2.0» и попадает в главное меню

0. «Главное меню системы».

Главная страница имеет следующие пункты меню:

0.1 Создание ЗК

0.2 Выбор ЗК из списка имеющихся

0.3 Быстрый поиск ЗК

0.4 Анализ ЗК

0.5 Расширенный поиск ЗК

0.6 Экспорт ЗК

0.7 Импорт ЗК

0.8 Служебные телефоны

0.9 Изменение настроек отображения

0.10 Выход

Если клиент осуществит 0.2 Выбор ЗК из списка имеющихся, то система автоматически выполнит загрузку по требованию – в таблице телефонов появятся телефоны выбранного владельца. После этого клиент получит доступ к следующим функциям системы:

2.1 Изменение выбранной ЗК

2.2 Удаление выбранной ЗК

2.3 Выход

Если пользователь нажмет на кнопку 2.2. Удаление выбранной ЗК, система потребует подтверждения данного действия. В случае, если был получен положительный ответ, выбранная записная книга будет удалена, а отображаемый список будет обновлен.

Кнопка 2.1. Изменение выбранной ЗК, так же, как и кнопка 0.1. Создание ЗК, открывает доступ к форме редактирования/добавления записи. Изменив поля с полученными данными и нажав на кнопку «Редактировать» (или добавив данные в поля и нажав на кнопку «Добавить новую ЗК», в случае формы добавления), программа осуществит проверку телефонных номеров на совпадения. Если проверка прошла удачно, то данные сохранятся. Если же в базе данных найдено совпадение, то будет открыто меню 11. Вывод информации о совпавшем телефонном номере:

11.1 Изменение ЗК с совпавшим номером

11.2 Изменение добавляемой/редактируемой ЗК

11.3 Закрывать ЗК без сохранения

С помощью кнопки 11.1 Изменение ЗК с совпавшим номером пользователь попадает на форму редактирования записной книги, которая содержит совпавший телефонный номер. Кнопка 11.2 Изменение добавляемой/редактируемой ЗК закроет текущее меню и вернет пользователя обратно на форму добавления/редактирования. Кнопка 11.3. Закрывать ЗК без сохранения закроет форму добавления/редактирования и вернет клиента на 0. «Главное меню системы».

На 0. «Главное меню системы» пользователь может воспользоваться функцией 0.3. Быстрый поиск ЗК, принцип работы которой заключается в следующем: приложение осуществляет запрос в базу данных, основываясь на введенном номере записной книги. Если результат был получен, то список записных книг на главной форме отображает результаты поиска. В ином случае, система выдает сообщение об ошибке.

При нажатии на кнопку 0.4. Анализ ЗК, пользователь попадает на форму 4. Анализ ЗК. Для того чтобы выполнить анализ записной книги, пользователю необходимо осуществить: выбор анализируемой ЗК (ввести номер книги, телефонные номера и списки которой будут анализироваться), выбор области анализа ЗК (определить какие записные книги будут сравниваться с анализируемой: со всей базы данных, только с выбранными номерами или сформированные за определенный период.), выбор размера справки (краткая или полная) , выбор типа привязки анализа (анализ с привязкой к телефонным номерам или лицам). После нажатия кнопки «Анализ» осуществляется анализ ЗК по введенным параметрам, в результате которого сформируется справка. Пользователь увидит на своем экране текст справки, а его в свою очередь можно сохранить в текстовом формате на компьютере. После работы с формой анализа, пользователь возвращается на 0. «Главное меню системы».

Нажав на кнопку 0.5. Расширенный поиск ЗК, пользователь попадает на форму 5. Расширенный поиск:

5.1. Поиск по введенным параметрам

5.2.. Сброс результатов поиска

Данная форма выполняет функцию поиска записей в базе данных. Нажав на кнопку «Поиск», программа выполнит 5.1. Поиск по введенным параметрам, который представляет из себя подготовку и выполнение запроса к базе данных с введенными параметрами. Ввод параметров поиска служит для корректировки поискового запроса по следующим реквизитам: номер записной книжки, номер телефона, привязанный к ней, фамилия, имя, отчество, дата рождения, адрес проживания владельца, дата формирования записной книги, а также поиск книг, сформированных за период. По результатам поиска пользователь может сформировать справку или вывести их на главный экран. Осуществив 5.2. Сброс результатов поиска, система перезагрузит содержимое таблицы с отображаемыми ЗК.

Выбрав на главной форме 0.6. Экспорт ЗК, пользователь попадает на форму 6. Экспорт ЗК:

6.1. Выбор пути для файла экспортируемых данных

6.2. Отправка данных на экспорт

Отметив необходимые данные для экспорта, клиент переходит к 6.1. Выбор пути для файла экспортируемых данных, где ему открывается диалоговое окно с выбором места сохранения файла с данными из базы. Создав новый файл или перезаписав старый, пользователь переходит к условию: выбора пароля для файла экспортируемых данных, суть которого заключается в вводе пароля, которым будет шифроваться файл с данными. Нажав на кнопку «Экспорт», программа приступает к 6.2. Отправка данных на экспорт, функция которой заключается в выгрузке данных из базы в оперативную память, а затем в создании таблиц в локальном файле и загрузка в них выгруженных данных.

Выбрав на главной форме 0.7. Импорт ЗК, пользователь попадает на форму 7. Импорт ЗК

В самом начале работы с данной формой клиент должен выбрать путь для файла с данными импорта, в результате чего откроется диалоговое окно с выбором местоположения файла с импортируемыми данными. Если пользователю известно, что данные зашифрованы паролем, необходимо ввести пароль для файла с импортируемыми данными, что позволит приступить к импорту зашифрованных данных, иначе система выдаст ошибку либо в случае неверного пароля, либо в случае отсутствия надобности в нем. Нажав кнопку «Приступить к импорту», система начнет выполнять импорт данных из выбранного файла, суть которого представляет собой подготовку списка для импорта. Данный список получается в ходе следующих операций: сперва данные из файла выгружаются в оперативную память, на чем работа с файлом заканчивается, затем выполняется процесс сравнения выгруженных данных и имеющихся в основной базе. В случае совпадения импортируемых данных и основных, система открывает 20. Мастер импорта в случае совпадения записей:

20.1 Сохранение импортируемой ЗК

20.2. Сохранение имеющейся ЗК

20.3. Пропуск импорта совпавшей ЗК

20.4. Пропуск всех совпавших ЗК

20.5. Изменение импортируемой ЗК

20.6. Отмена

В данной форме пользователь осуществляет корректировку импортируемого списка. Нажав на кнопку 20.1 Сохранение импортируемой ЗК, система подготовит к удалению имеющуюся запись в основной базе (список на удаление пополнится) и продолжит выполнять сравнение. Нажав на кнопку 20.2. Сохранение имеющейся ЗК, система поступит иначе – из списка в оперативной памяти удалится запись, а сравнение продолжится далее. Активировав кнопку 20.3. Пропуск импорта совпавшей ЗК, список импорта сместит свой итератор вперед на одно значение, если это возможно, и продолжит сравнение далее. Используя кнопку 20.4. Пропуск всех совпавших ЗК, система удалит из списка импорта совпавшую запись и будет выполнять тоже самое со следующими, до тех пор, пока сравнение записей не будет успешным или список кончится. Помимо всего прочего, у клиента есть возможность отредактировать импортируемую запись, перейдя на форму добавления/редактирования по кнопке 20.5. Изменение импортируемой ЗК. В результате такого редактирования, запись сохранится в импортируемом списке, а система начнет повторное сравнение записей.

Если в ходе выполнения последнего сравнения записей система не обнаружила повторений, то она переходит в стадию импорта, суть которой заключается в добавлении новых записей из импортируемого списка и удаления старых, в случае если список на удаление не пуст. По результатам выполнения импорта система выдаст сообщение.

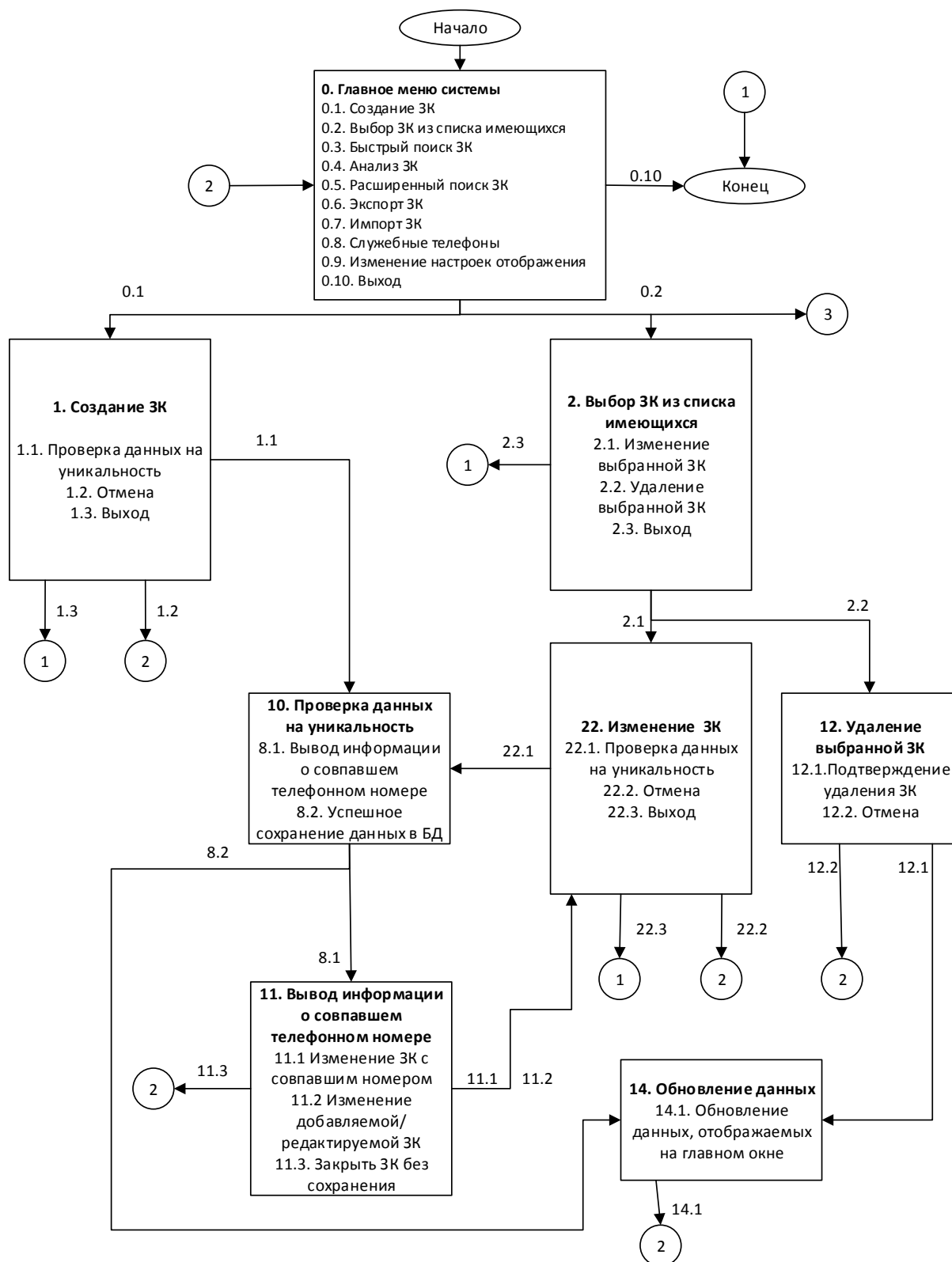
В случае если пользователь на главном меню нажмет на кнопку 0.8. Служебные телефоны, то система откроет форму со справочными телефонами, основная функция которой – отображение имеющихся в базе служебных телефонов.

Помимо всего вышеописанного, пользователь может изменить отображаемые данные о записных книгах на главной форме, нажав на 0.9.

Изменение настроек отображения. В открывшейся форме настроек пользователю доступно изменение количества отображаемых записей на одной странице, а также атрибутов таблицы записных книг. Настройки сохраняются после изменения в файле с расширением .ini.

Выход из системы осуществляется по кнопке 0.10 «Выход».

Граф диалога с пользователем проектируемой системы представлен на рисунках 36 -38. [4]



ПРИМЕЧАНИЕ: ЗК – записная книга
БД – база данных

Рис.36 Граф диалога ч.1

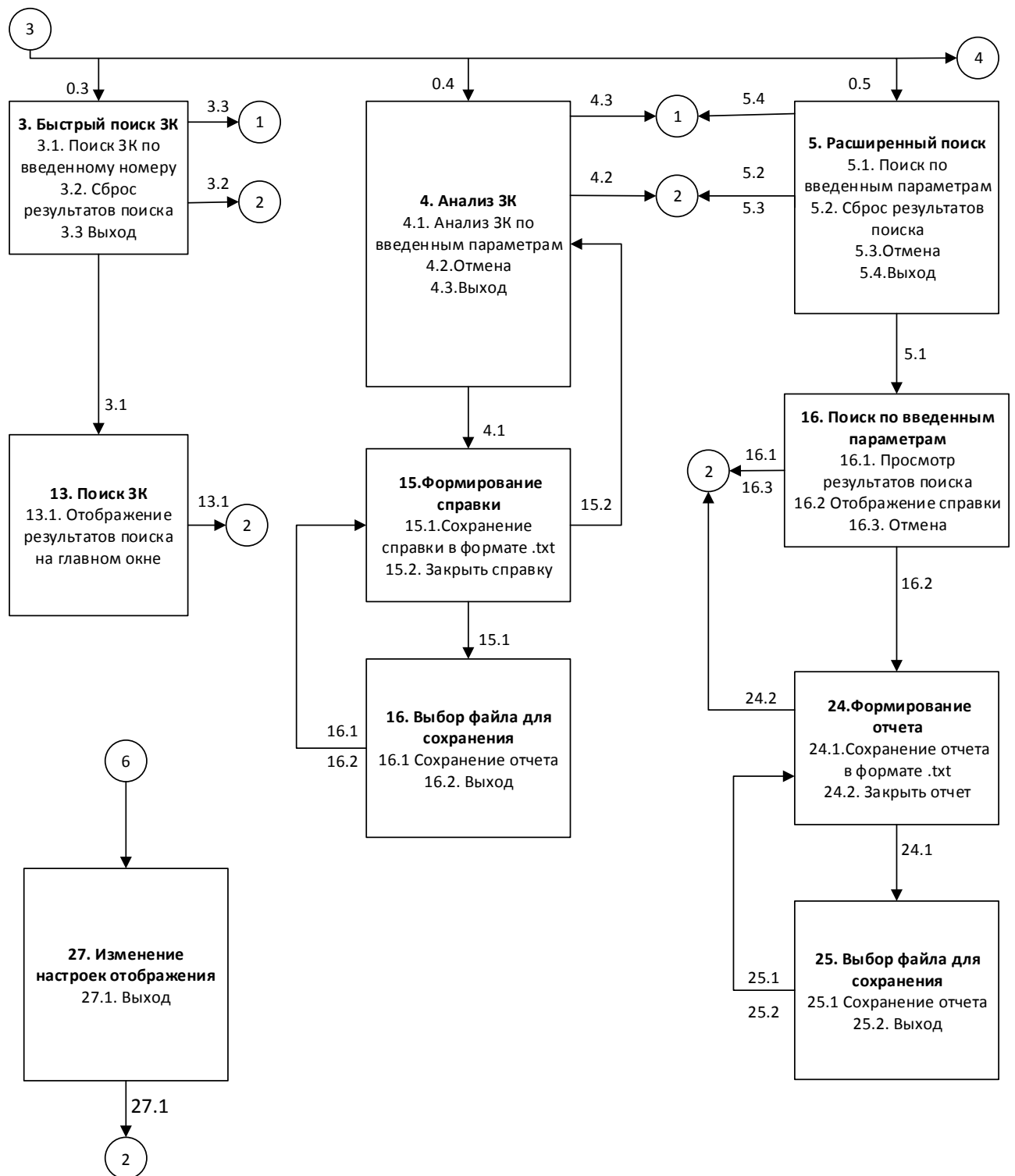


Рис.37 Граф диалога ч.2

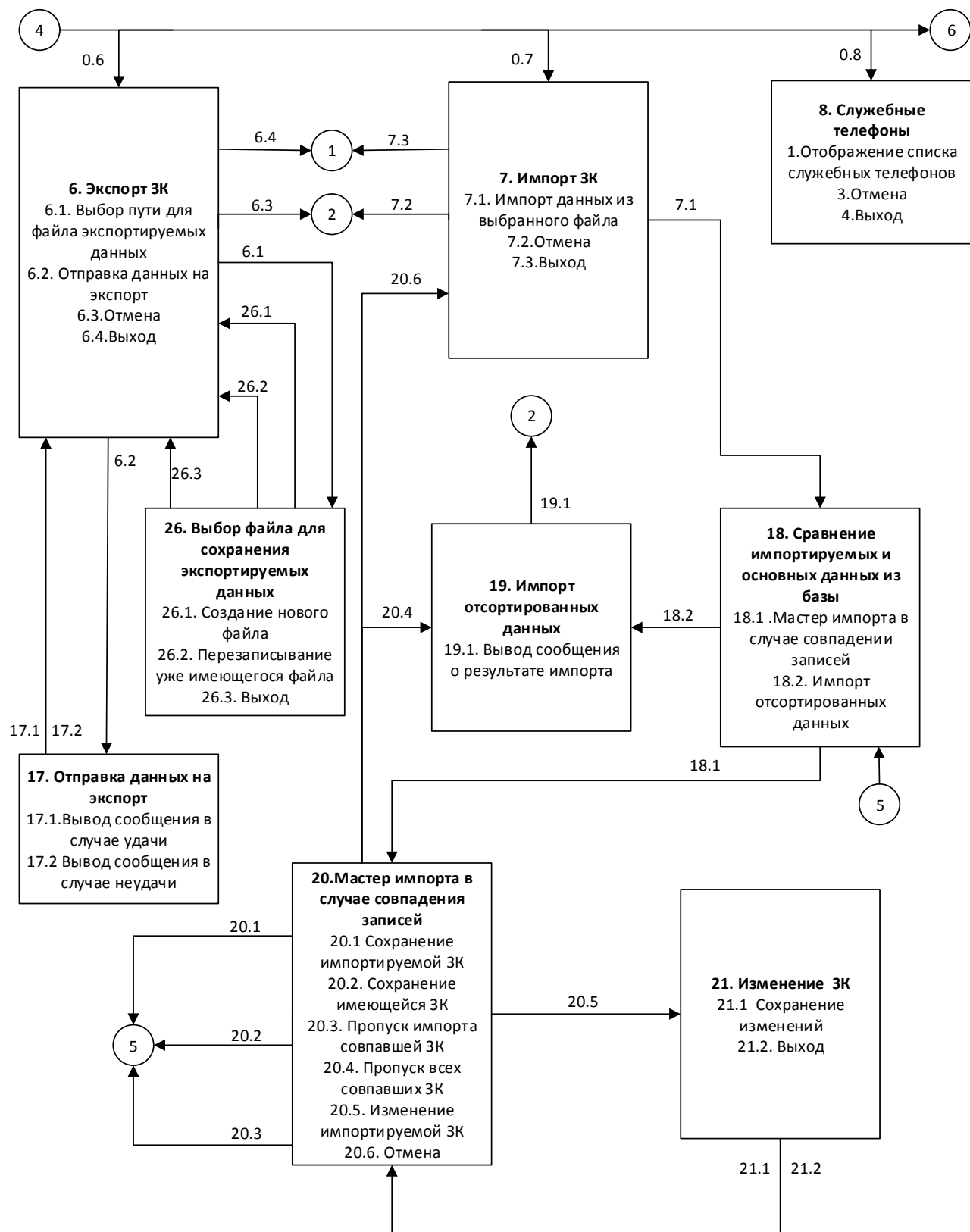


Рис.38 Граф диалога ч.3

2.2 Разработка интерфейса пользователя

Интерфейс взаимодействия пользователя с системой разрабатывается для отображения связи пользователя с системой. Интерфейс должен быть интуитивно понятным для пользователя, дружелюбным, обладать эстетической привлекательностью и соответствовать эргономическим требованиям.

Форма, представленные в данном разделе, соответствуют формам, описанным в разделе «Разработка графа диалога» (п. 2.1).

На рис. 39-50 представлены основные формы работы АСУА «Записная книжка 2.0»:

Номер ЗК	Фамилия	День добавления	Дата рождения	Имя	Отчество	Время добавления
<input type="checkbox"/> 1	Житенев	2017-05-22	1997-10-23	Владислав	Германович	
<input type="checkbox"/> 2	Сумин	2016-07-12	1995-03-22	Сергей	Иванович	23:31:22
<input type="checkbox"/> 3	Савина	2012-05-17	2000-12-02	Елизавета	Ивановна	14:00:06
<input type="checkbox"/> 4	Желов	2013-04-02	2012-02-04	Евгений	Петрович	10:11:26
<input type="checkbox"/> 5	Васнецов	2019-06-17	1990-01-13	Олег	Иванович	10:11:45

Номер телефона
1 89258558882
2 8914578888

Номер телефона	Пометка
1 8911111155	ТЕСТ
2 8911111111	2 ВАРИАНТ

Рис 39. главная страница «Записная книжка 2.0»

The screenshot shows the 'Записная книга 2.0' application window. The 'Редактирование' (Editing) tab is active, showing a form for editing a contact entry. The form includes fields for:

- Family name: Житнев
- Name: Владислав
- Patronymic: Германович
- Additional information: TEST
- Birth date: 23/10/1997
- Registration address: Moscow, Street: улица, House: дом, Corridor/Building: 1, Apartment: 200
- Residence address: Tver, Street: Горького, House: 23, Corridor/Building: а, Apartment: 200
- Phone numbers: 89258558882, 8914578888
- Contact list: (empty)

 The left sidebar contains a 'Быстрый поиск' (Quick search) section with a search button. The bottom of the form has 'Редактировать' (Edit) and 'Отмена' (Cancel) buttons.

Рис.40 Форма Редактирования

The screenshot shows a warning dialog box titled 'ВНИМАНИЕ' (Attention). The message states: 'Введенный телефонный номер 7777777 обнаружен принадлежим владельцу записной книжки № 3' (The entered phone number 7777777 has been found to belong to the owner of notebook No. 3). Below the message are three buttons: 'Перейти к записной книжке № 3' (Go to notebook No. 3), 'Редактировать телефонный номер' (Edit phone number), and 'Заккрыть карточку без сохранения' (Close card without saving).

Рис.41 Оповещение в случае нарушения уникальности

The screenshot shows the 'Записная книга 2.0' application window with the 'Добавление новой записной книжки' (Add new notebook) form. The form includes fields for:

- Family name, Name, Patronymic
- Additional information
- Birth date: Day, Month, Year
- Registration address: City, Street, House, Corridor/Building, Apartment
- Residence address: City, Street, House, Corridor/Building, Apartment
- Phone numbers: 1
- Contact list: (empty)

 The left sidebar contains a 'Быстрый поиск' (Quick search) section. The bottom of the form has 'Добавить новую записную книжку' (Add new notebook) and 'Отмена' (Cancel) buttons.

Рис.42 Форма Добавления

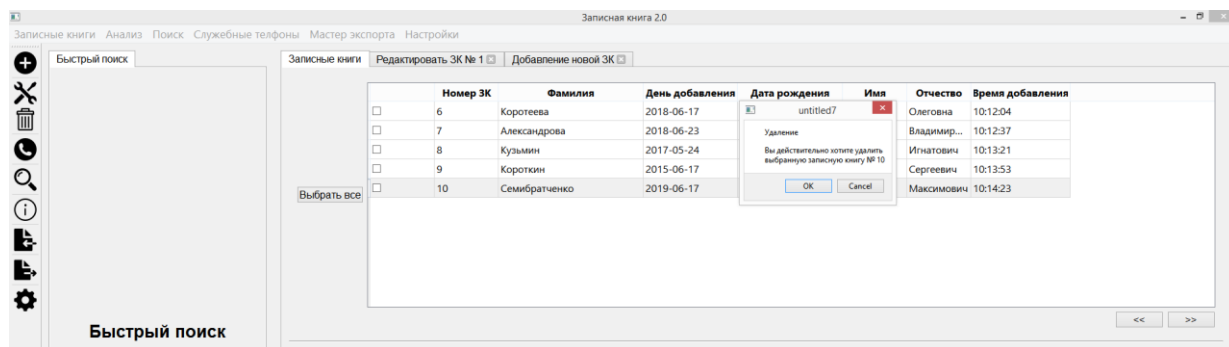


Рис. 43. Удаление выбранной записи

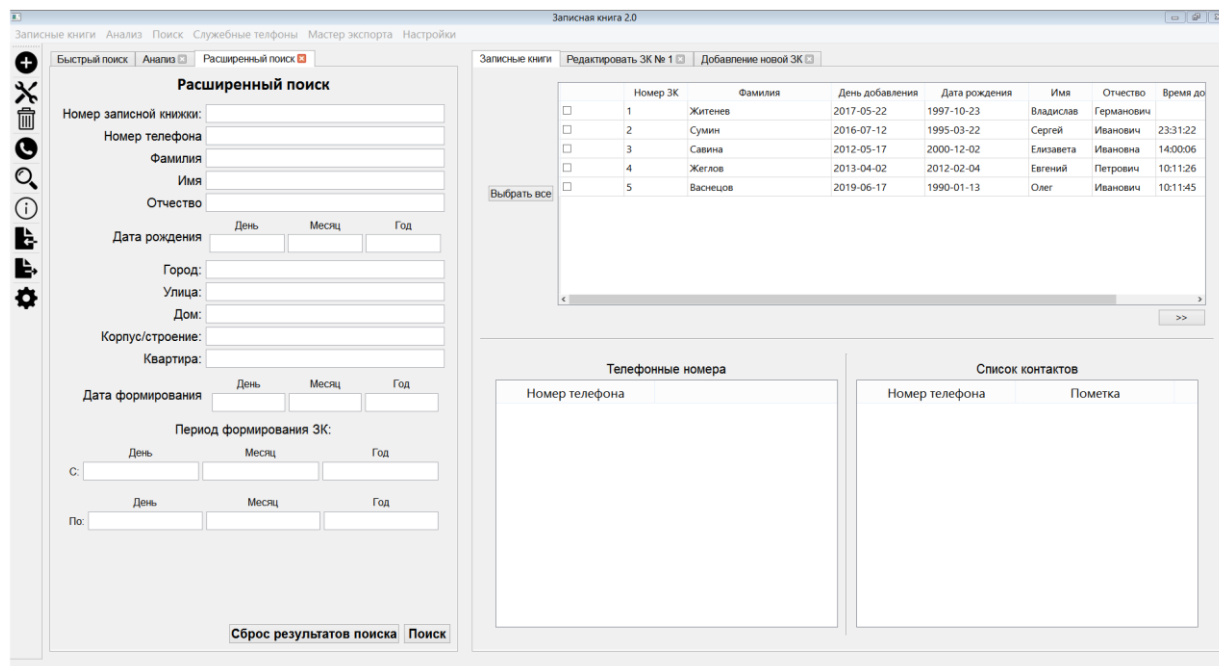


Рис.44 Форма Расширенного поиска

Записная книга 2.0

Записные книги Анализ Поиск Служебные телефоны Мастер экспорта Настройки

Быстрый поиск Служебные телефоны

Служебные телефоны

Наименование службы	Номер телефона
1 Главное отделение ФСБ	8(499)-720-56-22

Записные книги

Номер ЗК	Фамилия	чь добавлен	дата рождения	Имя	Отчество	дата добавления
<input type="checkbox"/>	1	Житенев	2017-05-22	1997-10-23	Владислав Германович	
<input type="checkbox"/>	2	Сумин	2016-07-12	1995-03-22	Сергей Иванович	23:31:22
<input type="checkbox"/>	3	Савина	2012-05-17	2000-12-02	Елизавета Ивановна	14:00:06
<input type="checkbox"/>	4	Жеглов	2013-04-02	2012-02-04	Евгений Петрович	10:11:26
<input type="checkbox"/>	5	Васнецов	2019-06-17	1990-01-13	Олег Иванович	10:11:45

Выбрать все

Телефонные номера

Номер телефона	Номер
1 89258558882	
2 8914578888	

Список контактов

Рис.45 Форма Служебных телефонов

Записная книга 2.0

Записные книги Анализ Поиск Служебные телефоны Мастер экспорта Настройки

Быстрый поиск Служебные телефоны Экспорт данных

Экспорт данных

Наименование службы	Номер телефона
1 Главное отделение ФСБ	8(499)-720-56-22

Записные книги

Номер ЗК	Фамилия	чь добавлен	дата рождения	Имя	Отчество	дата добавления
<input type="checkbox"/>	1	Житенев	2017-05-22	1997-10-23	Владислав Германович	
<input type="checkbox"/>	2	Сумин	2016-07-12	1995-03-22	Сергей Иванович	23:31:22
<input type="checkbox"/>	3	Савина	2012-05-17	2000-12-02	Елизавета Ивановна	14:00:06
<input type="checkbox"/>	4	Жеглов	2013-04-02	2012-02-04	Евгений Петрович	10:11:26
<input type="checkbox"/>	5	Васнецов	2019-06-17	1990-01-13	Олег Иванович	10:11:45

Выбрать все

Телефонные номера

Номер телефона	Номер
1 89258558882	
2 8914578888	

Список контактов

Рис.46 Форма Экспорта данных

Импорт данных

Выберите путь

Введите пароль

Прислупить к импорту

Номер ЗК	Фамилия	день добавления	дата рождения	Имя	Отчество	время добавления
<input type="checkbox"/> 1	Житенев	2017-05-22	1997-10-23	Владислав	Германович	
<input type="checkbox"/> 2	Сумин	2016-07-12	1995-03-22	Сергей	Иванович	23:31:22
<input type="checkbox"/> 3	Савина	2012-05-17	2000-12-02	Елизавета	Ивановна	14:00:06
<input type="checkbox"/> 4	Жеглов	2013-04-02	2012-02-04	Евгений	Петрович	10:11:26
<input type="checkbox"/> 5	Васнецов	2019-06-17	1990-01-13	Олег	Иванович	10:11:45

Выбрать все

Телефонные номера

Номер телефона

1 89258558882

2 8914578888

Список контактов

Рис.47 Форма Импорта данных

Найдено совпадение

Телефонный номер **89258558882** принадлежащий владельцу импортируемой ЗК № 1 совпадает с телефонным номером владельца уже имеющейся ЗК № 1

Изм.

Номер ЗК	Фамилия	день добавления	дата рождения	Имя	Отчество	время добавления
<input type="checkbox"/> 1	Житенев	2017-05-22	1997-10-23	Владислав	Германович	

Импортируемая БД

Номер ЗК	Фамилия	день добавления	дата рождения	Имя	Отчество	время добавления
<input checked="" type="checkbox"/> 1	Житенев	2017-05-22	1997-10-23	Владислав	Германович	

Основная БД

Сохранить импортируемую ЗК

Пропустить импорт этой ЗК

Сохранить имеющуюся ЗК

Соединить обе ЗК в одну

Пропустить импорт этой и последующих совпадающих ЗК

Рис.48 Форма Мастера импорта

Form

Настройки

Выберите отображаемые данные для ЗК

- ☒ Номер ЗК
- ☒ Фамилия
- ☒ Имя
- ☒ Отчество
- ☒ Дата рождения
- ☒ Проверяется в интересах
- ☒ Дополнительная информация
- ☒ Город проживания
- ☒ Улица проживания
- ☒ Дом проживания
- ☒ Корпус проживания
- ☒ Квартира проживания
- ☒ Дата добавления
- ☒ Город регистрации
- ☒ Улица регистрации
- ☒ Дом регистрации
- ☒ Корпус регистрации
- ☒ Квартира регистрации
- ☒ Время добавления

Количество отображаемых ЗК 5

Сохранить настройки

Рис. 50 Форма настроек

2.3 Краткий обзор классов

Из-за большого количества функций, выполняемых системой, пришлось прибегнуть к созданию большого количества классов. С одной стороны, это усложняет структуру проекта из-за большого числа заимствований, однако в тоже время позволяет использовать меньше кода из-за возможности использовать различные методы, реализованные в разных классах. [10]

Свернутая диаграмма зависимостей классов и краткий обзор классов представлены на рисунках 51-46

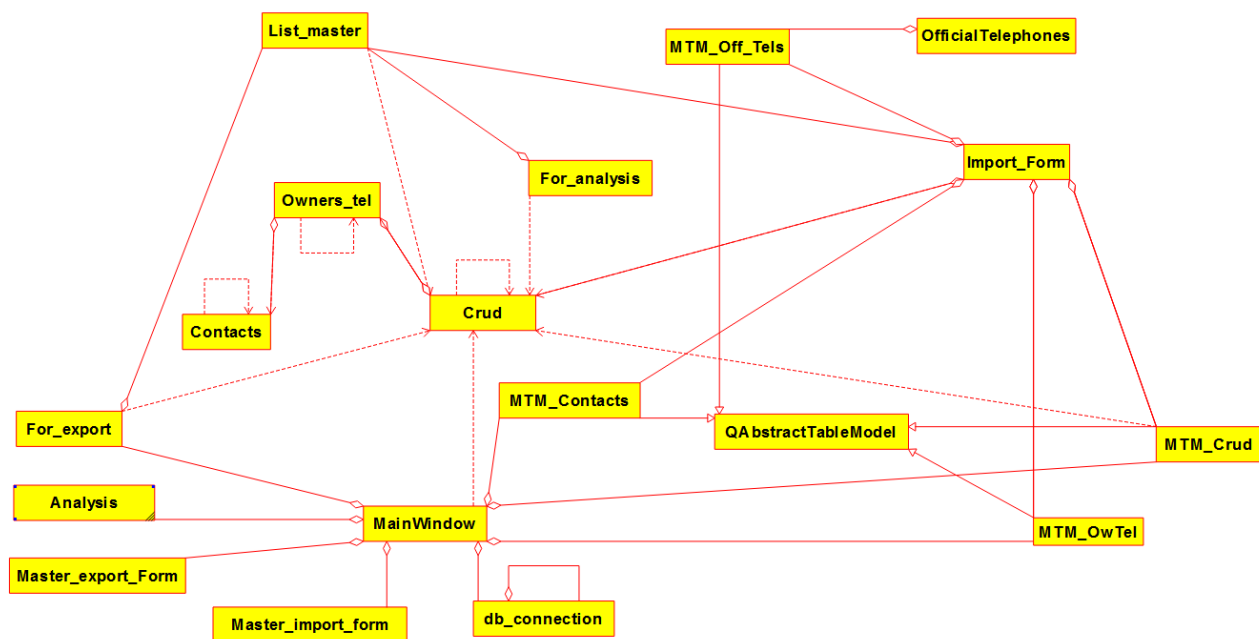


Рис.51 Свернутая диаграмма зависимостей классов проекта
«Записная книжка 2.0»

▼ Form_Headers	
an_result_Form.h	Форма вывода справок
analysis_Form.h	Форма анализа
Import_Form.h	Форма импорта
mainwindow_Form.h	Форма главного окна
master_export_Form.h	Форма экспорта данных
master_import_Form.h	Форма импорта данных
officialtelephones_Form.h	Форма служебных телефонов
search_Form.h	Форма поиска
settings_Form.h	Форма настроек
update_Form.h	Форма редактирования/добавления записных книг
▼ Model_Headers	
_MTM_Contacts.h	Модель отображения данных о контактах
_MTM_Crud.h	Модель отображения данных о ЗК
_MTM_Off_tels.h	Модель отображения данных о служебных телефонах
_MTM_OwTel.h	Модель отображения данных о номерах телефонов
▼ Table_Headers	
_Contacts.h	Класс сущности "Контакты"
_Crud.h	Класс сущности "Записная книга"
_Off_tels.h	Класс сущности "Служебные телефоны"
_Owners_tel.h	Класс сущности "Телефон"
db_connection.h	Класс подключения к базе данных
for_analysis.h	Класс методов анализа
for_export.h	Класс методов экспорта
list_master.h	Класс работы со списками

Рис.52 Краткий обзор классов в среде генератора документации
Doxygen.

2.4 Ключевые особенности проектирования системы

Во время разработки системы «Записная книжка 2.0» было необходимо сделать акцент на создание программного продукта с широким спектром функциональных возможностей и с компактным и эргономичным интерфейсом. Однако, учитывая, что данное программное обеспечение будет внедряться в компанию с большим количеством клиентов, персональные компьютеры которых не отличаются высокой производительностью, при создании данной системы пришлось сделать упор на оптимизацию.

Для осуществления вышеописанных потребностей, были выполнены следующие особенности проектирования системы:

1. Использование связанных списков
2. Использование пользовательских моделей отображения данных
3. Корректное формирование даты для запроса
4. Динамическое освобождение памяти при работе с формами
5. Система анализа
6. Работа с отдельными файлами разных форматов

2.4.1 Использование связанных списков

Во время работы с приложением, клиенту регулярно приходится обращаться к базе хранимых данных (запросы на отображение и анализ записных книг, добавление телефонных номеров и контактов, экспорт и т.д.). Частая отправка SQL-запросов к удалённому серверу базы данных плохо сказывается на производительности, так как к выполнению этих процессов добавляются сторонние факторы, такие как работоспособность удаленного сервера и качество соединения с ним. Для того чтобы решить вопрос оптимизации приложения при работе с удалённым сервером, было принято решение использовать связанные

списки. Такой подход позволяет отказаться от многочисленного обращения к серверу базы данных и переместить основное поле работы на оперативную память персонального компьютера клиента. [1]

Одна из ключевых особенностей C++ - это работа с указателями, которые позволяют осуществлять динамическое выделение памяти. Именно поэтому при создании системы было принято решение сделать наибольший упор на грамотное использование указателей. [10]

Проектирование классов сущностей базы данных представляло собой задачу определения свойств в соответствии атрибутам сущности, к которой относится класс. Также были определены списки указателей на зависимые классы, аналогично зависимостям сущностей. Таким образом:

- В классе записных книг Crud определен список указателей на класс телефонов Owners_tel.
- В классе телефонов Owners_tel определен список указателей на класс контактов Contacts.

Результат создания классов сущностей изображен на рисунке 53

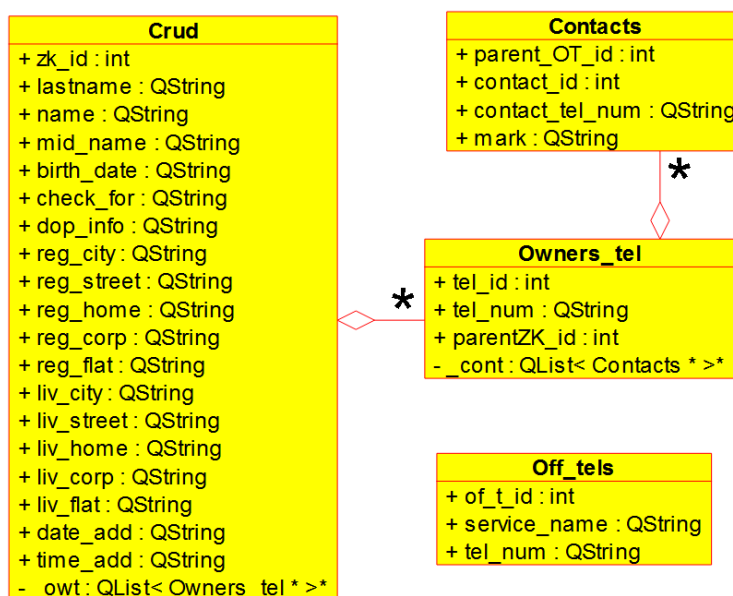


Рис.53 Классы сущностей

Такой подход в программировании позволяет осуществить загрузку по требованию: данные загружаются только по требованию пользователя, выборочно. Такой механизм дает возможность существенно ослабить

потребление свободной оперативной памяти приложением.

2.4.2 Использование пользовательских моделей отображения данных

Эргономичное отображение данных и создание интуитивно понятного интерфейса – это одни из важнейших задач проектировки системы «Записная книжка 2.0». Опыт пользования подобным программным обеспечением показал, что применение таблиц для отображения данных – самый оптимальный вариант решения. Однако таблицы в Qt имеют одну особенность – все данные, отображаемые в них, представляют собой модель, которую настраивает сам проектировщик. Были рассмотрены все доступные стандартные библиотеки Qt для работы с моделями:

QSqlQueryModel. Модель данного класса представляет собой выполненный SQL-запрос. Данные невозможно изменить, таблица служит для просмотра результатов запроса. [9]

QSqlTableModel. Данный класс имеет скудный функционал, так как выполняет задачу отображения данных из одной таблицы напрямую. Записи можно добавлять, удалять и изменять, а после отправки соответствующего сигнала, записи из модели синхронизируются с базой. [9]

Обе вышеописанные библиотеки не подходят под требования поставленной задачи. Именно поэтому было принято решение создать классы-наследники от библиотеки абстрактных моделей QAbstractTableModel [9].

Количество проектируемых классов определилось числом сущностей, данные которых необходимо отобразить в табличном представлении. Были созданы следующие классы:

- MTM_Crud – модель отображения данных сущности «Записная книга»
- MTM_OwTel – модель отображения данных сущности «Телефон»
- MTM_Contacts – модель отображения данных сущности «Контакты»

- MTM_Off_Tels – модель отображения данных сущности «Служебные телефоны»

Результат создания классов можно увидеть на рисунке 54

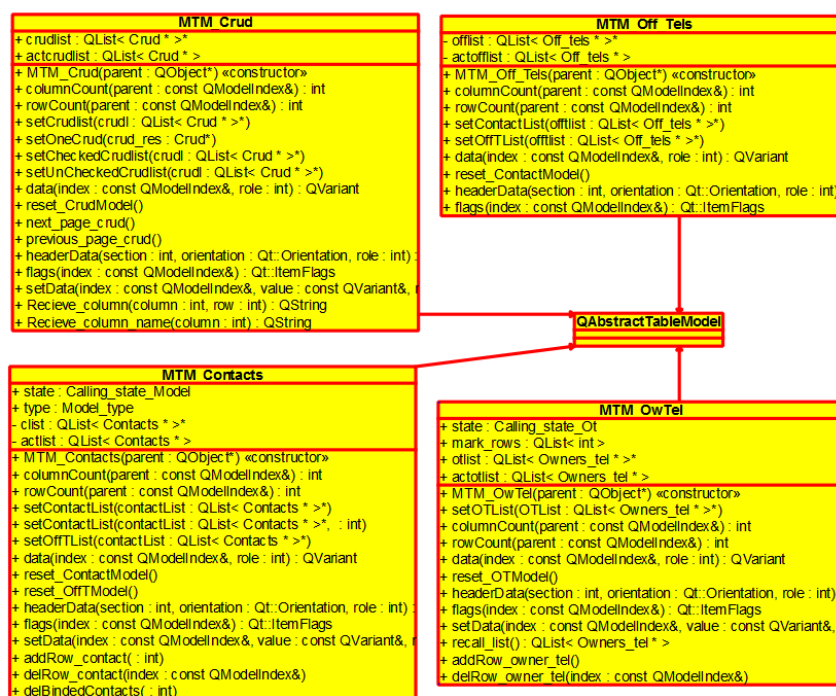


Рис.54 Создание классов моделей данных

Каждый класс наследуется от QAbstractTableModel с спецификатором доступа public. Несмотря на такое число, казалось бы, разных наследников, методы и свойства у них примерно одинаковые, так как выполняется перегрузка виртуальных методов базового класса. Появляются разные методы, в зависимости от выполняемых функций. Также принцип работы моделей остается одинаковым: каждая из них имеет список указателей для изменения и отображения, которые в свою очередь зависят от сущности с которой работает модель данных. Описание методов класса MTM_Crud изображено на рисунке 55.[10]

Public Member Functions

	MTM_Crud (QObject *parent=nullptr)
virtual int	columnCount (const QModelIndex &parent=QModelIndex()) const Переопределение количества колонок модели
virtual int	rowCount (const QModelIndex &parent=QModelIndex()) const Переопределение количества строк модели
virtual void	setCrudlist (QList< Crud * > *crudl) Получение списка записных книг моделью
virtual void	setOneCrud (Crud *crud_res) Получение одной записной книги моделью (Быстрый поиск)
virtual QVariant	data (const QModelIndex &index, int role=Qt::DisplayRole) const Получение списка записных книг моделью More...
void	reset_CrudModel () Сброс отображаемого списка модели
virtual void	next_page_crud () Загрузка записных книг следующей страницы
virtual void	previous_page_crud () Загрузка записных книг предыдущей страницы
virtual QVariant	headerData (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const Переопределение заголовков таблицы
virtual Qt::ItemFlags	flags (const QModelIndex &index) const Установка флагов
virtual bool	setData (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) Метод изменения данных в таблице
virtual QString	Recieve_column (int column, int row) const Метод получения данных для отображения таблицы (изменение настроек)
virtual QString	Recieve_column_name (int column) const Метод получения заголовков для отображения таблицы (изменение настроек)

Рис 55 Описание методов класса MTM_Crud

Использование такого подхода имеет внушительный ряд преимуществ на фоне подключения к базовым библиотекам. Рассмотрим на примере все той же MTM_Crud. Наследование от класса абстрактных моделей позволило существенно расширить функциональные возможности модели отображаемых данных. Появились функции пагинации, динамического перестроения таблицы после изменения настроек, а также возможность использования виджета в качестве данных колонки таблицы. Работа класса MTM_Crud изображена на рисунке 56 [11]

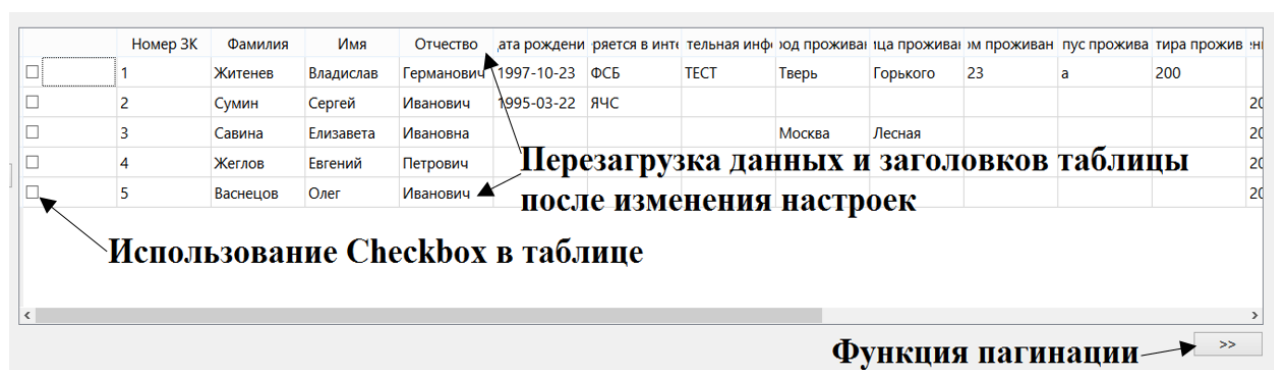


Рис.56 Работа класса MTM_Crud

2.4.3 Корректное формирование даты для запроса

Работа с полями даты имеет одну интересную особенность. В отличие от числового и символьного типа данных, дата привязана к формату вида YYYY-MM-DD (год-месяц-день). Также, необходимо учитывать потребности пользователя, например, в выводе записных книг, сформированных за определенные годы. Именно поэтому возникает потребность в составлении правильного SQL-запроса при работе с полями такого типа.

Для решения этой задачи был разработан алгоритм `get_date`, изображенный на рисунке 57

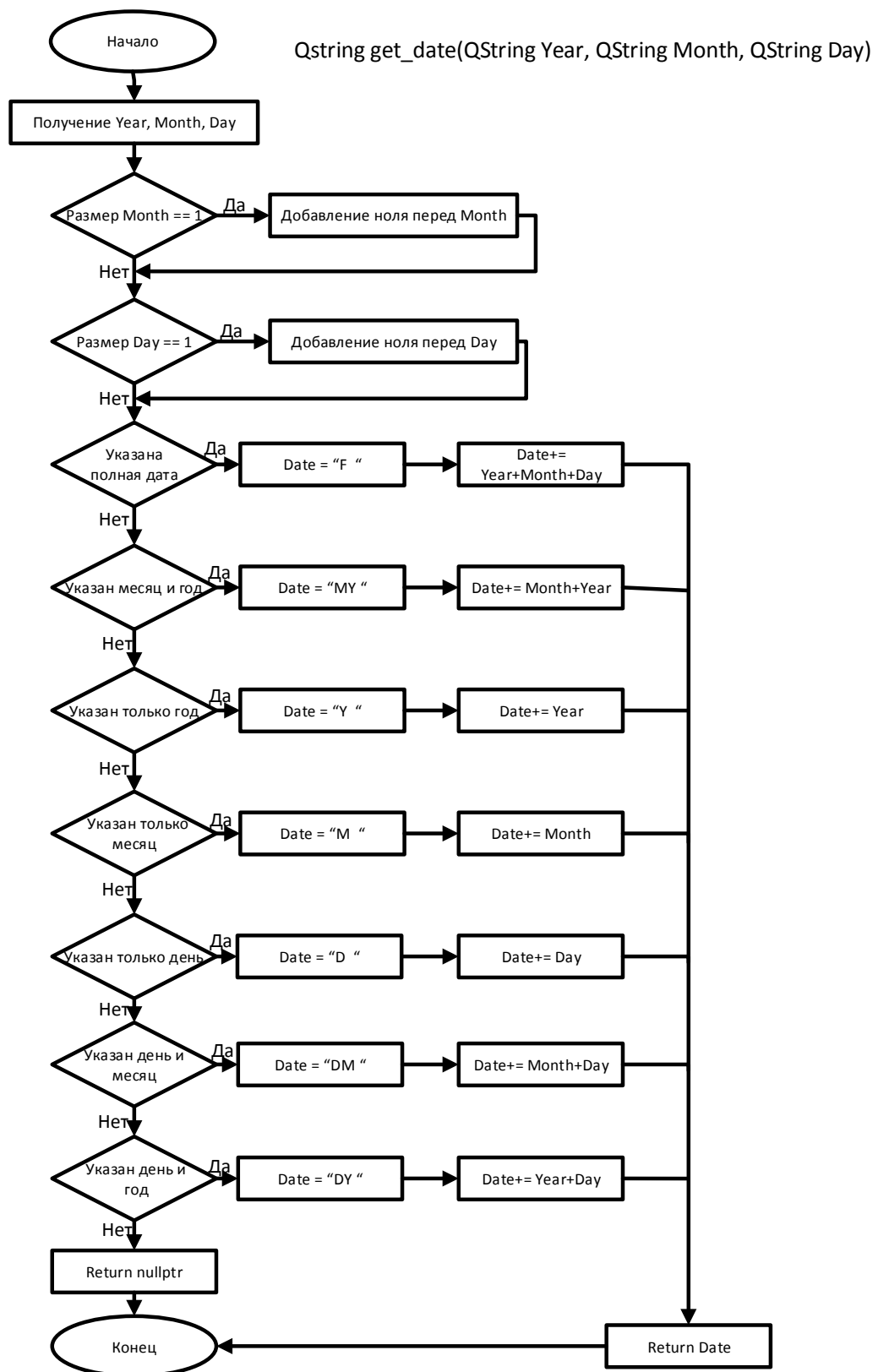


Рис.57 Алгоритм создания строки даты в необходимом формате.

Механизм алгоритма представляет собой следующее: полученные введенные пользователем данные, в первую очередь проверяются на корректное

соблюдение формата (если в поле дня или месяца указана одна цифра, то алгоритм добавит ноль в начало этих строк). В зависимости от того, какие поля были заполнены, система добавит соответствующую метку, основываясь на которой будет сформирован запрос.

Таким образом в случае если клиент не знает полную дату формирования записи, а только, например, месяц и год, то система расшифрует префикс строки даты и выполнит запрос к базе данных в корректном виде. Пример расшифровки и результат выполнения запроса на рисунках 58-59 [8]

```
if (temp == "MY ")
{
    switch (flag) {
        case 1:
            query = " AND EXTRACT(MONTH FROM zk.Birth_date) = ('"+date.left(2)+"') "
                " AND EXTRACT(YEAR FROM zk.Birth_date) = ('"+date.right(4)+"')";
            return query;
        case 2:
            query = " AND EXTRACT(MONTH FROM zk.date_add) = ('"+date.left(2)+"') "
                " AND EXTRACT(YEAR FROM zk.date_add) = ('"+date.right(4)+"')";
            return query;
    }
}
```

Рис. 58 Пример расшифровки префикса для месяца и года

The screenshot shows the 'Записная книга 2.0' application. The 'Расширенный поиск' (Advanced Search) panel on the left contains various input fields for searching records, including 'Номер записной книжки', 'Номер телефона', 'Фамилия', 'Имя', 'Отчество', 'Дата рождения', 'Город', 'Улица', 'Дом', 'Корпус/строение', 'Квартира', 'Дата формирования', and 'Период формирования ЗК'. The 'Записные книги' (Address Books) panel on the right displays a table with the following data:

	Номер ЗК	Фамилия	Имя	Отчество	День добавлен
<input checked="" type="checkbox"/>	4	Жеглов	Евгений	Петрович	2013-04-02

Below the table is a 'Выбрать все' (Select all) button. At the bottom of the interface, there are sections for 'Телефонные номера' (Phone numbers) and 'Список контактов' (List of contacts), each with a 'Номер телефона' (Phone number) input field.

Рис. 59 Результат выполнения поискового запроса

2.4.4 Динамическое освобождение памяти при работе с формами

Как было сказано ранее, работа с указателями – одна из ключевых особенностей C++. Помимо связанных списков, им также есть отличное применение в создании классов форм [11]. Использование динамического выделения и освобождения памяти существенно ускоряет работу программы.

При проектировании АСУА «Записная книжка 2.0» было принято решение использовать механизм обнуления указателей. Его применение можно увидеть в работе класса главной формы, так как все формы связаны с ней.

Инициализация нулевых указателей показана на рисунке 60.

```
56 class MainWindow : public QMainWindow
57 {
58     Q_OBJECT
59
60 public:
61     Search *sr = nullptr;           //форма поиска
62     Master_export_Form *expirt = nullptr; //форма экспорта
63     Master_import_form *impirt = nullptr; //форма импорта
64     OfficialTelephones *of = nullptr; //форма служебных телефонов
65     class Analysis *an = nullptr;    //форма анализа
66 }
```

Рис 60. Инициализация нулевых указателей на классы форм.

Сразу после инициализации указателям присваивается нулевой адрес в памяти. Данное действие позволяет избежать присвоение указателям случайных адресов. Затем, если пользователь нажмет кнопку необходимой ему формы, система выделит место для соответствующего класса формы с помощью оператора new. Алгоритм открытия форм изображен на рисунке.61



Рис.61 Алгоритм открытия форм

Однако при работе с указателями необходимо следить за выделяемой памятью, чтобы избежать ее переполнения «мусорными» данными. Так как классы форм наследуются от библиотеки `QWidget`, то событие закрытия оконной формы лишь скрывает ее от пользователя, а память так и остается выделенной. Наилучшим решением послужит перегрузка [1] базового метода `CloseEvent`, который будет выполнять удаление данных с помощью оператора `delete`. Алгоритм работы метода представлен на рисунке 62

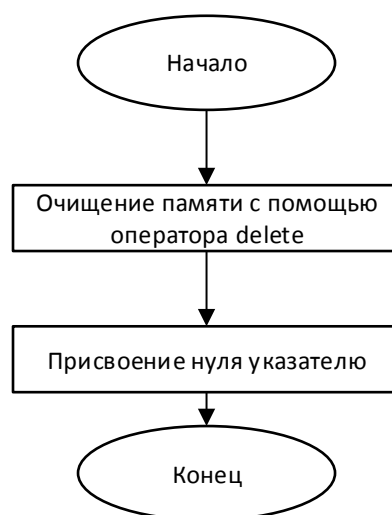


Рис 62. Алгоритм работы метода `CloseEvent` после перезагрузки

2.4.5 Система анализа

Система анализа является одной из основных особенностей системы. Основное предназначение – поиск связей между фигурантами базы. Всего существует три разных варианта знакомств фигурантов – два прямых и один косвенный:

Первый вариант знакомства. Телефонный номер, принадлежащий анализируемому владельцу, был найден среди контактов другой записной книги. В результате формируется справка о прямой связи фигурантов. Схема первого варианта связи и текст SQL-запроса изображен на рисунках 63-64

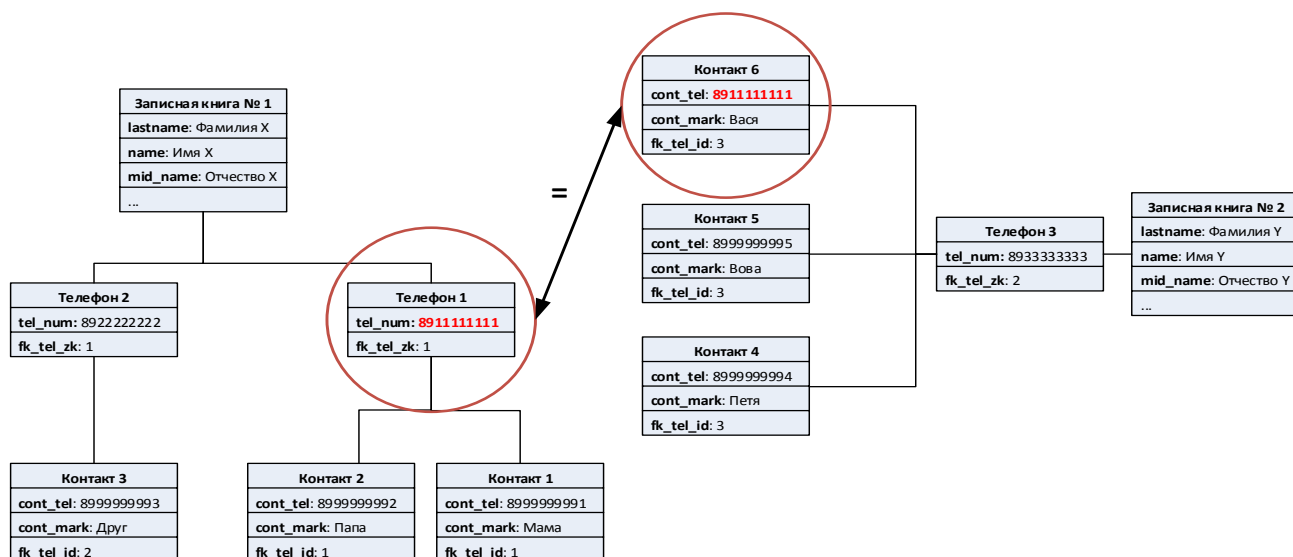


Рис 63. Выявление первого варианта связи при анализе записной книги №1. В контактах владельца Y обнаружен телефонный номер владельца X.

```
QString tempSQL = " SELECT zk,zk_id, S.TEL_NUM, S.CONTACT_MARK,"
    " S.OW_TEL_NUM "
    " FROM zk "
    " INNER JOIN (SELECT "
    " owners_tel.telephone_num AS TEL_NUM,"
    " Q.CONTACT_MARK AS CONTACT_MARK,"
    " owners_tel.telephone_id AS TEL_ID, Q.TEL_NUM AS OW_TEL_NUM"
    " FROM Owners_tel"
    " INNER join(SELECT contacts.cl_telephone AS CONTACT_NUM,"
    " contacts.cl_info AS CONTACT_MARK,"
    " contacts.fk_cl_telephone AS FK, "
    " OW.TEL_NUM AS TEL_NUM"
    " FROM contacts"
    " INNER JOIN (SELECT owners_tel.telephone_num AS TEL_NUM"
    " FROM owners_tel"
    " where owners_tel.fk_telephone_zk = (:zk_id)) AS OW"
    " ON OW.TEL_NUM = contacts.cl_telephone) AS Q"
    " ON Q.FK = owners_tel.telephone_id) AS S"
    " ON S.TEL_ID = zk.zk_id "
    " WHERE zk.zk_id >0 ";
```

Рис.64 SQL-запрос, выполняемый системой для определения первого варианта связи.

Второй вариант знакомства. Телефонный номер, находящийся в одном из списков контактов анализируемой записной книги, был найден принадлежащим владельцу другой записной книги. В результате формируется справка о прямой связи фигурантов. Схема второго варианта связи и текст SQL-запроса изображен на рисунках 65-66

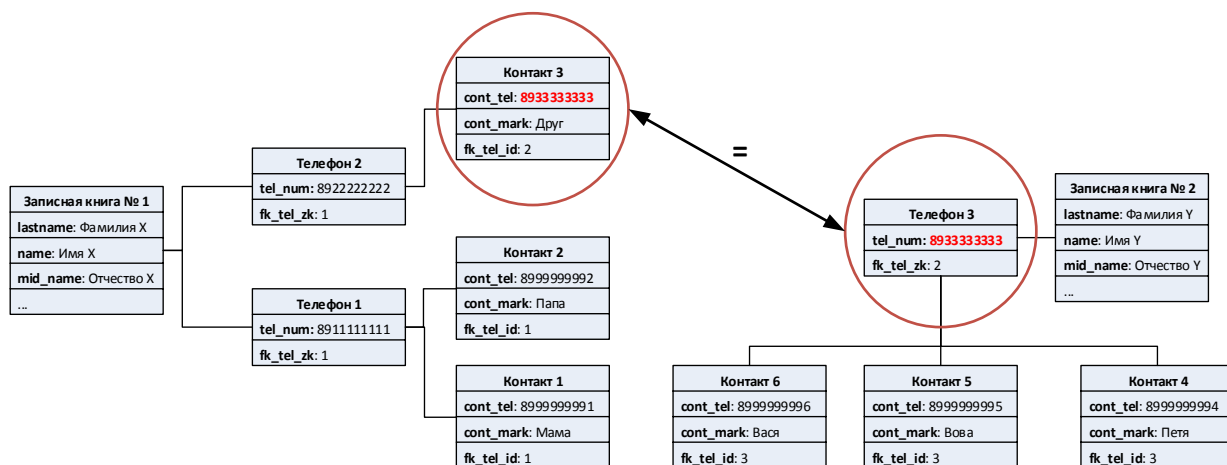


Рис.65 Выявление второго варианта связи при анализе записной книги №1. В контактах владельца X обнаружен телефонный номер владельца Y.

```
tempSQL += " SELECT zk.zk_id, OWT.OWNER_TEL,"
           " OWT.TEL_NUM, OWT.CONT_MARK"
           " FROM zk"
           " INNER JOIN (SELECT owners_tel.fk_telephone_zk AS FK_ZK,"
           " owners_tel.telephone_num AS TEL_NUM,"
           " CONT.CONT_MARK AS CONT_MARK,"
           " CONT.OWNER_TEL AS OWNER_TEL"
           " FROM owners_tel"
           " INNER JOIN (SELECT contacts.cl_telephone AS CONT_TEL,"
           " contacts.cl_info AS CONT_MARK,"
           " tels.OWNER_TEL AS OWNER_TEL"
           " FROM contacts "
           " INNER JOIN(SELECT DISTINCT owners_tel.telephone_id as tel_id,"
           " owners_tel.telephone_num AS OWNER_TEL"
           " FROM owners_tel"
           " WHERE owners_tel.fk_telephone_zk = (:zk_id)) as tels"
           " ON tels.tel_id = contacts.fk_cl_telephone) AS CONT"
           " ON CONT.CONT_TEL = owners_tel.telephone_num) AS OWT"
           " ON OWT.FK_ZK = zk.zk_id"
           " WHERE ZK.zk_id>0 ";
```

Рис.66 SQL-запрос, выполняемый системой для определения второго варианта связи.

Третий вариант знакомства. Телефонный номер, находящийся в одном из списков контактов анализируемой записной книги, был найден в списке, принадлежащем иному владельцу. В результате формируется справка о косвенной связи фигурантов

Схема третьего варианта связи и текст SQL-запроса изображен на рисунках 67-68

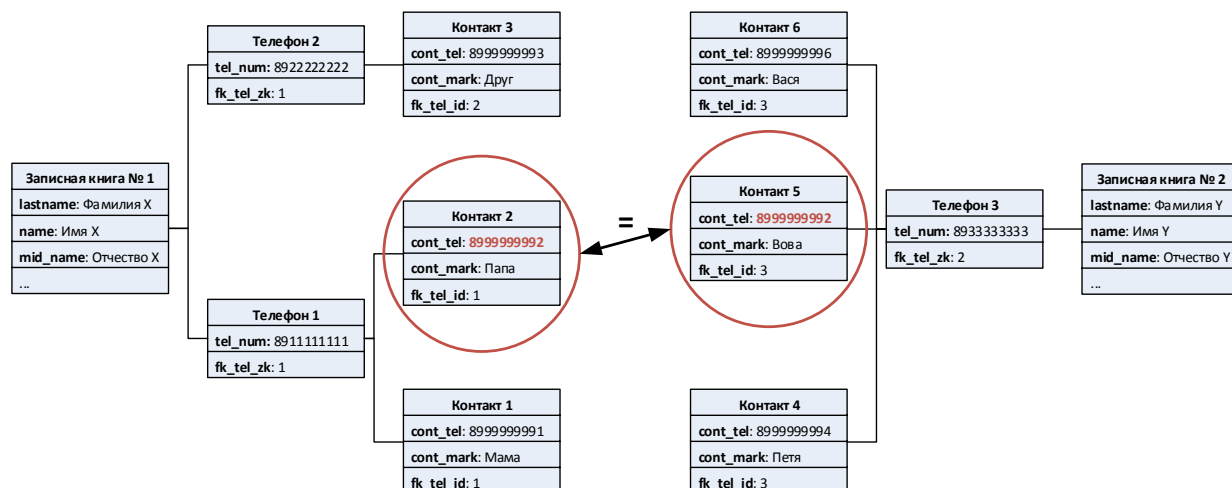


Рис.67 Выявление третьего варианта связи при анализе записной книги №1. В контактах владельца X обнаружен телефонный номер из списка контактов владельца Y.

```
tempSQL += " SELECT zk.zk_id, OW.TEL_NUM, OW.CONTACT_TEL, OW.CONTACT_MARK "
" FROM zk"
" INNER JOIN(SELECT DISTINCT owners_tel.fk_telephone_zk as ID,"
" owners_tel.telephone_num as TEL_NUM,"
" Q.CONT_MARK AS CONTACT_MARK, "
" Q.CONT_TEL AS CONTACT_TEL"
" FROM owners_tel"
" INNER JOIN"
" (SELECT DISTINCT contacts.cl_telephone as CONT_TEL,"
" contacts.contact_list_id as CONT_ID,"
" contacts.fk_cl_telephone as CONT_FK,"
" contacts.cl_info as CONT_MARK "
" FROM contacts"
" INNER JOIN (SELECT DISTINCT contacts.cl_telephone as TELEPHONE"
" , contacts.contact_list_id as ID "
" FROM contacts, owners_tel"
" WHERE owners_tel.fk_telephone_zk = (:zk_id)) AS S"
" ON S.TELEPHONE = contacts.cl_telephone"
" WHERE S.ID != contacts.contact_list_id ) AS Q"
" ON Q.CONT_FK = owners_tel.telephone_id"
" WHERE owners_tel.fk_telephone_zk != (:zk_id)) AS OW"
" ON OW.ID = zk.zk_id"
" WHERE zk_id>0 ";
```

Рис.68 SQL-запрос, выполняемый системой для определения третьего варианта связи.

После выполнения всех трех запросов к базе данных и получения результата, система готовит справку в текстовом виде. Пример текстовой справки на рисунке 69.

ФОРМИРОВАНИЕ ПОЛНОЙ СПРАВКИ С ПРИВЯЗКОЙ К ЛИЦАМ

Владелец записной книжки Житенев Владислав Германович возможно знаком со следующими лицами:

Сумин Сергей Иванович, 1995-03-22, адрес регистрации: , / , , адрес проживания: , / , телефон: 895558888, 891111111, окраска ???, дополнительная информация: , Дата формирования 2016-07-12 в интересах: ФСБ

Вывод о знакомстве сделан на основании того, что:

1 ВАРИАНТ Номер телефона 8914578888, принадлежащий Житенев Владислав Германович, обнаружен в записной книжке, владельцем которой является Сумин Сергей Иванович с пометкой Друг

1 ВАРИАНТ Номер телефона 89258558882, принадлежащий Житенев Владислав Германович, обнаружен в записной книжке, владельцем которой является Сумин Сергей Иванович с пометкой Кон3

2 ВАРИАНТ Номер телефона 891111111 с пометкой 2 ВАРИАНТ в записной книжке, владельцем которой является Житенев Владислав Германович, обнаружен, как принадлежащий владельцу записной книжки: Сумин Сергей Иванович

3 ВАРИАНТ Номер телефона 88888888 с пометкой Контакт2 был обнаружен в ЗК, принадлежащей Сумин Сергей Иванович, так же как и в ЗК, принадлежащей Житенев Владислав Германович

3 ВАРИАНТ Номер телефона 89258558882 с пометкой Кон3 был обнаружен в ЗК, принадлежащей Сумин Сергей Иванович, так же как и в ЗК, принадлежащей Житенев Владислав Германович

3 ВАРИАНТ Номер телефона 7777777 с пометкой Контакт4 был обнаружен в ЗК, принадлежащей Сумин Сергей Иванович, так же как и в ЗК, принадлежащей Житенев Владислав Германович

Сохранить в .txt Закреть

Рис.69 Пример текстовой справки по результатам выполнения анализа.

Полный алгоритм работы системы анализа представлен в приложении настоящего документа на листе «Блок-схема работы системы анализа»

2.4.6 Работа с отдельными файлами разных форматов

Для расширения функциональных возможностей при проектировании АСУА «Записная книжка 2.0», понадобилось прибегнуть к использованию файлов сторонних форматов.

Текстовые файлы

Формирование справок – одна из главных функций проектируемой системы. Выбор формата для файлов текстовых данных являлся важной задачей, ведь хранимые справки должны занимать наименьший объем памяти жесткого диска. После анализа всех альтернатив, было принято решение использовать формат txt. Пример и результат работы с текстовым файлом на рисунках 70-71

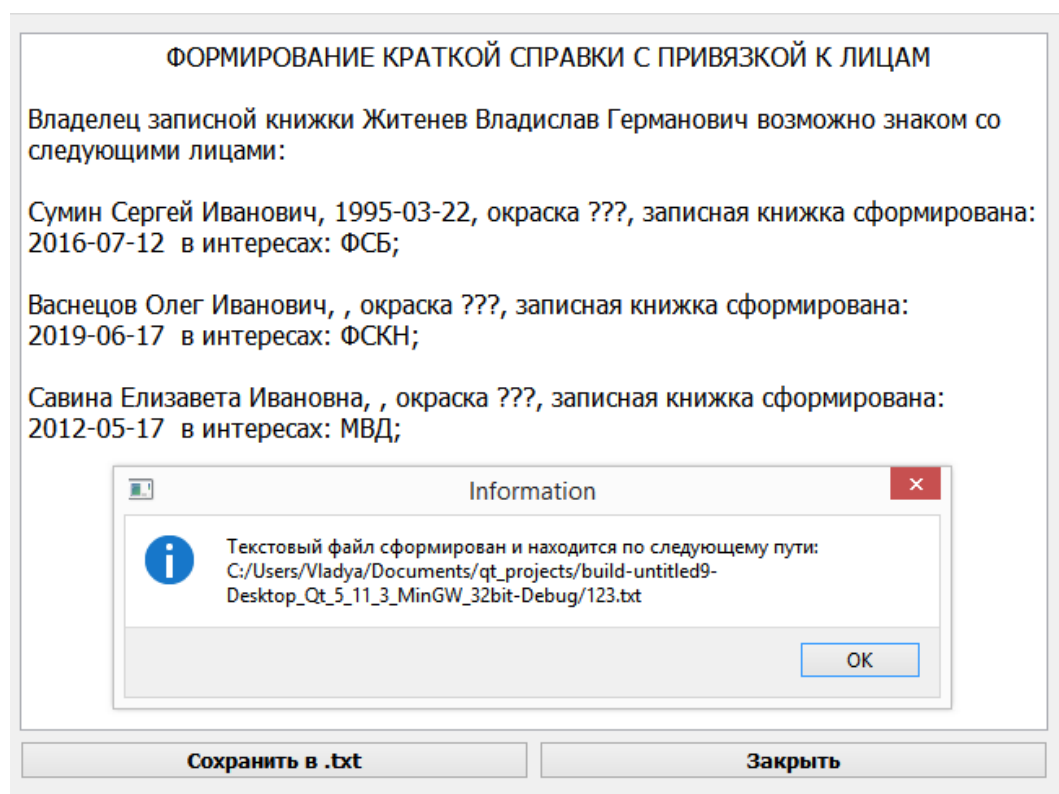


Рис.70 Успешное сохранение текста справки в файле 123.txt

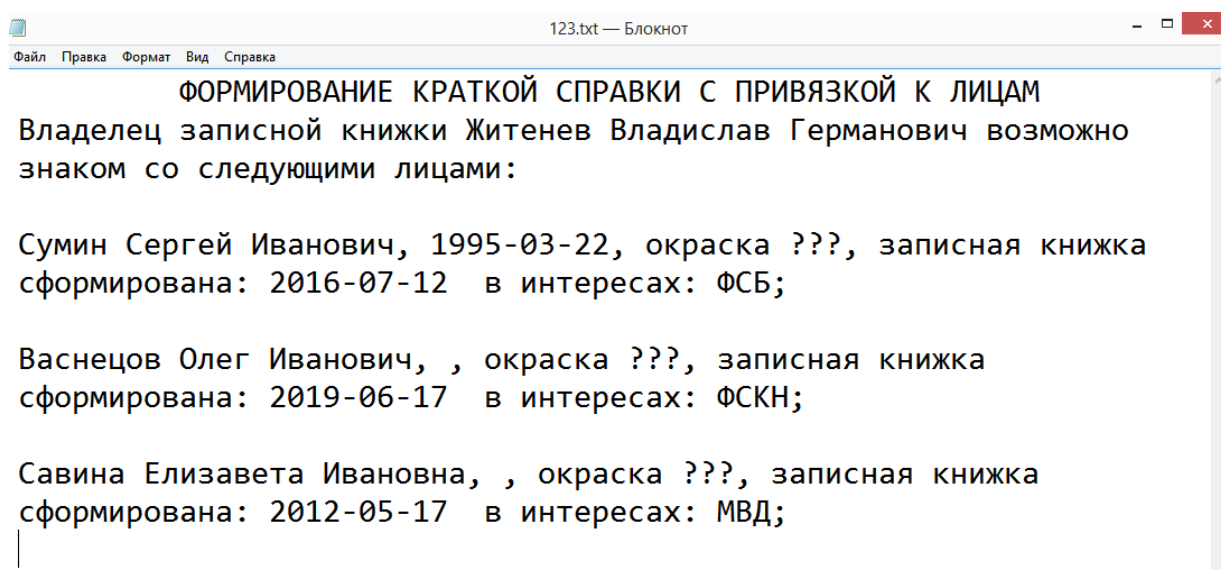


Рис.71 Текст справки, хранимый в файле 123.txt

Файлы базы данных

Для осуществления работы функций импорта и экспорта появилось необходимость выбора формата хранения данных базы. Самым оптимальным решением стал один из форматов SQLite [5] – db, так как эти файлы занимают

мало места на жестком диске. Пример создания и результат работы с файлом базы данных приведен на рисунках 72-73

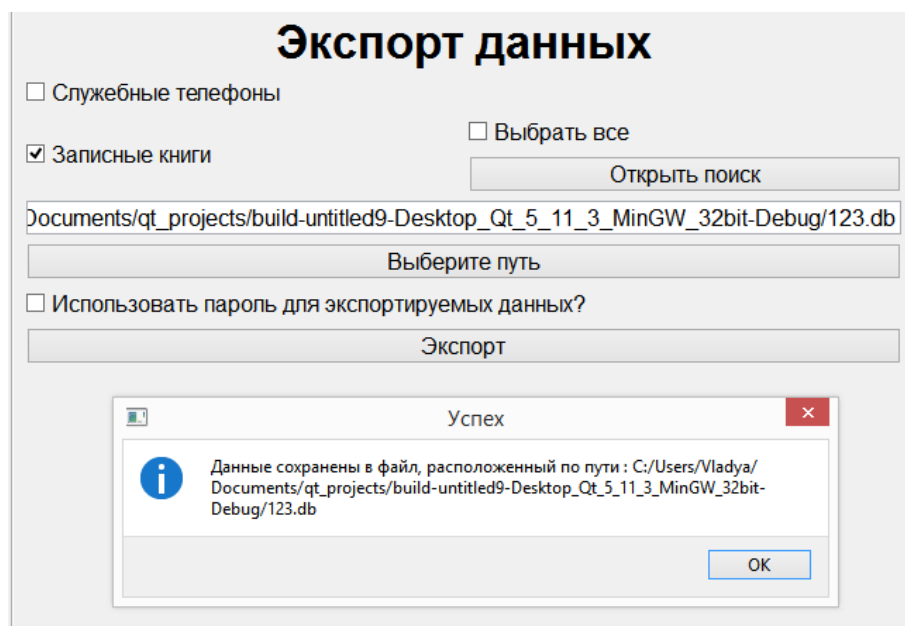


Рис.72 Успешное экспортирование данных в 123.db

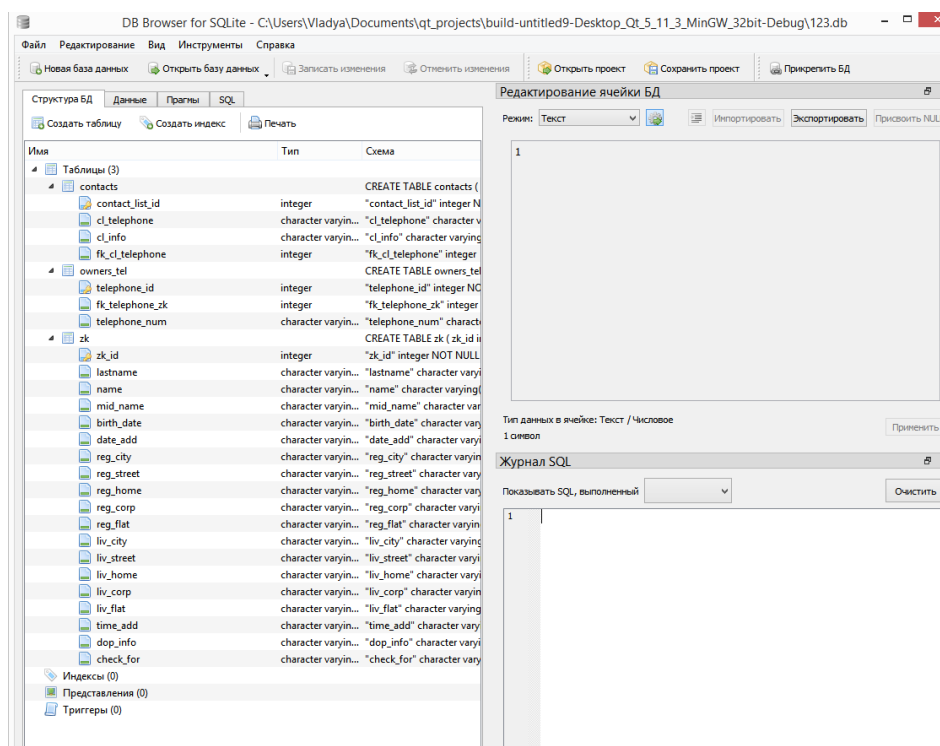


Рис.73 Открытие файла 123.db в программе DB Browser

Вторая особенность этого формата данных — это возможность использовать пароль для шифрования. Пример создания файла с паролем и

результат его открытия на рисунках 74-75

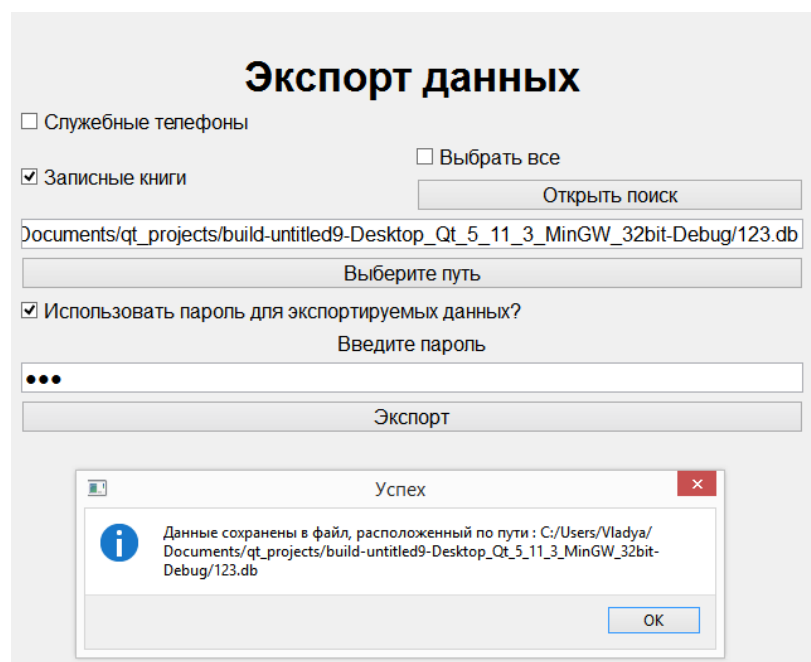


Рис.74 Успешное экспортирование данных с паролем в 123.db

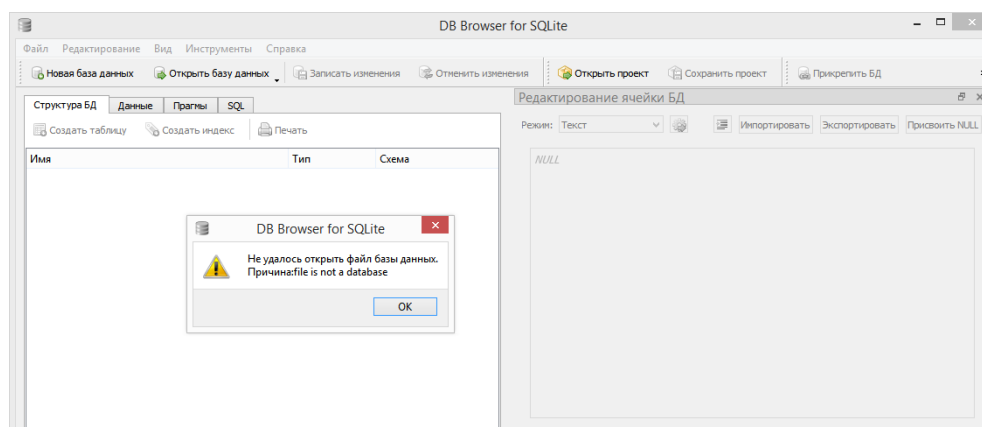


Рис. 75 Неудачная попытка открыть 123.db в программе DB Browser

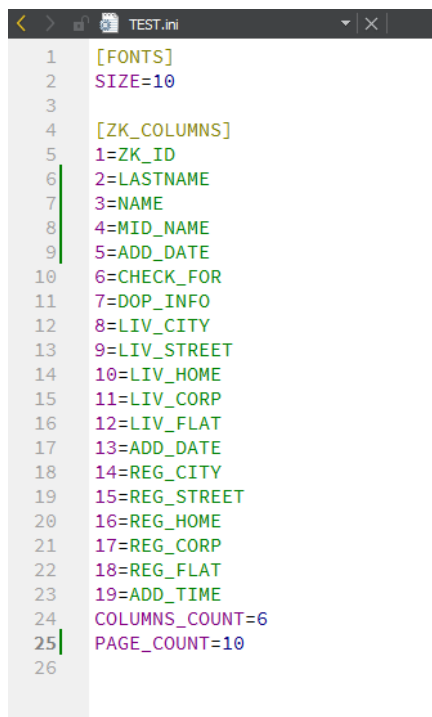
Открытие файла, зашифрованного паролем невозможно в сторонних программах для отображения табличных данных. Доступ к таким файлам открывается лишь при использовании операции импорта и вводе корректного пароля.

Файл настроек

Использование файла с хранимыми конфигурациями – правильный подход при работе с системами, подлежащими изменениям. Способ конфигурирования не пришлось долго выбирать – самым оптимальным вариантом стало

использование отдельного файла с форматом ini и регулярного подключения к нему.

Файлы формата ini изначально являются текстовыми, но имеют одну особенность - групповую структуру. Так как на назначение названий группам не существует регламента, то открывается широкий спектр возможностей при проектировании. Например, в файле с хранимыми настройками TEST.ini раздел FONTS отвечает за шрифт интерфейса, а параметр SIZE за его размер. Содержимое файла TEST.ini изображено на рисунке 76



```
1  [FONTS]
2  SIZE=10
3
4  [ZK_COLUMNS]
5  1=ZK_ID
6  2=LASTNAME
7  3=NAME
8  4=MID_NAME
9  5=ADD_DATE
10 6=CHECK_FOR
11 7=DOP_INFO
12 8=LIV_CITY
13 9=LIV_STREET
14 10=LIV_HOME
15 11=LIV_CORP
16 12=LIV_FLAT
17 13=ADD_DATE
18 14=REG_CITY
19 15=REG_STREET
20 16=REG_HOME
21 17=REG_CORP
22 18=REG_FLAT
23 19=ADD_TIME
24 COLUMNS_COUNT=6
25 PAGE_COUNT=10
26
```

Рис.76 Содержимое файла настроек TEST.ini

Исходя из вышеописанного, работа с файлом этого формата позволяет создать гибкую систему конфигурации, что куда удобнее чем скрытое (программное) изменение настроек внутри приложения.

ЗАКЛЮЧЕНИЕ

За время выполнения выпускной работы были решены все поставленные задачи из ТЗ такие, как:

- Разработка структуры и архитектур системы
- Разработка инфологической и даталогической модели базы данных
- Разработка алгоритмов обработки информации
- Разработка алгоритма-анализа записных книг;
- Разработка алгоритма устранения дубликатов;
- Разработка алгоритмов импорта и экспорта данных;
- Выбор средства программной реализации;
- Программная реализация;
- Разработка графа диалога и экранных форм
- Разработка документации.

Были применены навыки из следующих дисциплин:

- Модели данных, базы данных;
- Методы проектирования АСОИиУ;
- Эксплуатация АСОИиУ;
- Основы программирования;
- Модели данных;

Результатом ВКРБ является отлаженный программный продукт, отвечающий требованиям технического задания. Система является масштабируемой, т.к. клиент-серверная архитектура системы позволяет без труда увеличить количество рабочих станций. Также данная система может применяться и в устаревшей топологии с использованием локальных баз данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лафоре Р. Объектно-ориентированное программирование в C++, 4-е издание, 2004 год
2. Лекция 7. Инфологическое моделирование.
URL: <https://www.intuit.ru/studies/courses/1001/297/lecture/7411> (Дата обращения: 20.04.2019)
3. Григорьев Ю.А., Ревунков Г.И. Банки данных: Учеб. для вузов. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. — 320 с.
4. ГОСТ 19.701-90 Схемы алгоритмов, программ данных и систем.
5. SQL (типы данных): таблица URL: <http://fb.ru/article/282533/sql-tipyi-dannyih-tablitsa> (Дата обращения: 15.04.2019)
6. Записки программиста: Генерируем документацию на doxygen URL: <https://eax.me/doxygen/> (Дата обращения: 20.04.2019)
7. Основы эксплуатации автоматизированных систем обработки информации и управления. Краткий курс: учеб. пособие / В. М. Постников. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2013. — 177, [2] с.: ил.
8. П.Лузанов, Е.Рогов «PostgreSQL для начинающих», 1-е издание. 2016 год,
9. Qt
URL: <https://doc.qt.io/qt-5/qtassistant-index.html/> (Дата обращения: 22.05.2019)
10. Страуструп Бьерн. Программирование. Принципы и практика с использованием C++ 2-е издание, 2016г.
11. Саммерфилд Марк, Бланшет Жасмин Qt4. Программирование GUI на C++ 2007г.

ПРИЛОЖЕНИЕ А

Техническое задание

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э.Баумана)**

Утверждаю

_____ Виноградова М.В.

«__» _____ 2019 г.

«Автоматизированная система учета и анализа телефонных контактов»

техническое задание

(вид документа)

писчая бумага

(вид носителя)

5

(количество листов)

Исполнитель:

студент группы РТ5-81

Житенев В.Г.

«__» _____ 2019 г.

Москва, 2019 г.

СОДЕРЖАНИЕ

1	Наименование	85
2	Основание для разработки	85
3	Исполнитель.....	85
4	Назначение и цель разработки	85
5	Содержание работы.....	79
5.1	Задачи, подлежащие решению.....	79
5.2	Требования к функциональности программного изделия	86
5.3	Требования к разрабатываемым компонентам	87
5.4	Требования к составу программного обеспечения.....	80

1 Наименование

Автоматизированная система учета и анализа телефонных контактов.
Шифр: «Записная книжка 2.0»

2 Основание для разработки

Основанием для разработки является задание на ВКР, выданное кафедрой ИУ5, подписанное руководителем ВКР и утвержденное заведующим кафедрой ИУ5 МГТУ им. Н.Э. Баумана.

3 Исполнитель

Студент МГТУ им. Н.Э. Баумана группы РТ5-81 Житенев Владислав Германович.

4 Назначение и цель разработки

Программный продукт предназначен для Анализа информации о телефонных записных книжках с целью выявления связей первого и второго порядка и повышения эффективности работы пользователей.

5 Содержание работы

5.1 Задачи, подлежащие решению

- исследование предметной области;
- выбор и обоснование критериев качества
- сравнение аналогов и прототипов
- разработка структуры и архитектур системы
- разработка инфологической и даталогической модели базы данных
- разработка алгоритмов обработки информации
- разработка алгоритма-анализа записных книг;
- разработка алгоритма устранения дубликатов;
- разработка алгоритмов импорта и экспорта данных;
- выбор средства программной реализации;
- программная реализация;
- разработка графа диалога и экранных форм
- разработка документации.

5.2 Требования к функциональности программного изделия

АСУА «Записная книжка 2.0» должна удовлетворять следующим требованиям к функциональным характеристикам:

2.4.7 Перечень процессов, подлежащих автоматизации

- добавление, редактирование и удаление записных книг;
- добавление, редактирование и удаление телефонов;
- добавление, редактирование и удаление контактов;

- отображение данных;
- анализ записных книг;
- быстрый поиск по номеру записной книги;
- расширенный поиск по реквизитам;
- формирование справок по результатам поиска и анализа;
- вывод информации о служебных телефонах на экран;
- экспорт данных в файл;
- шифрование экспортируемых данных;
- импортирование данных из файла;

5.3 Требования к разрабатываемым компонентам

Программное обеспечение для взаимодействия с базой данных.

Программный продукт ориентирован для пользователей, которые не обладают техническими знаниями в области баз данных. При разработке необходимо сделать упор на удобность и понятность интерфейса.

5.4 Требования к составу программного обеспечения

Система должна работать на операционной системе семейства Microsoft Windows (XP, 7, 8 ,10) или семейства Linux

ПРИЛОЖЕНИЕ Б
Программа и методика испытаний

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э.Баумана)**

Утверждаю

_____ Виноградова М.В.

«__» _____ 2019 г.

«Автоматизированная система учета и анализа телефонных контактов»

программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

6

(количество листов)

Исполнитель:

студент группы РТ5-81

Житенев В.Г...

«__» _____ 2019 г.

Москва, 2019 г.

СОДЕРЖАНИЕ

1	Объект испытаний.....	91
2	Цель испытаний.....	91
3	Требования к объекту испытаний.....	91
4	Требования к программной документации	91
5	Средства и порядок испытаний	92
6	Методика испытаний	92

1 Объект испытаний

Объектом испытания является программный продукт, предназначенный для учета и анализа данных базы.

2 Цель испытаний

Целью проведения испытаний является доказательство работоспособности таких функций программного продукта, как проверка связи с базой данных, внесение изменений в таблицы БД, поиск данных, экспорт, импорт и анализ данных.

3 Требования к объекту испытаний

Требования к объекту испытаний содержатся в документе «Техническое задание», представленном в приложении А.

4 Требования к программной документации

Во время проведения испытания должны быть представлены следующие документы:

- Техническое задание;
- Расчётно-пояснительная записка.

5 Средства и порядок испытаний

Для проведения испытаний необходима минимум одна рабочая станция с операционной системой Windows и экземпляром тестируемого программного продукта. Так же понадобится наличие локальной базы данных или доступ к базе, расположенной на сервере. Подключение к БД через программу обеспечивается за счёт драйвера QPSQL.

6 Методика испытаний

Методика испытаний показана в таблице Б1.

Таблица Б1

Методика испытаний

№	Проверяемая функция	Выполняемые действия	Результат
1.	Запуск программы	Запустить файл ZK.exe	Открытие главной формы
2.	Загрузка привязанных телефонов	Выбрать записную книгу из списка	Появление списка телефонов, привязанных к ЗК, в соответствующей таблице
3.	Загрузка привязанных контактов	Выбрать записную телефон из списка	Появление списка контактов, привязанных к телефону, в соответствующей таблице

№	Проверяемая функция	Выполняемые действия	Результат
4.	Быстрый поиск	Ввести номер ЗК и нажать поиск	Загрузка результатов поиска в таблицу ЗК на главной форме
5.	Настройки	Открыть форму настроек выбрать колонки отображаемых ЗК нажать кнопку сохранить	Загрузка таблицы ЗК с учетом измененных параметров
6.	Редактирование	Выбрать ЗК из списка Нажать кнопку «Редактировать» Изменить редактируемые данные Нажать кнопку завершить редактирование	Если новые номера не совпадают с уже имеющимися, то сохранение данных в базе, перезагрузка таблицы на главной странице, открытие главной формы Если новые номера совпадают с уже имеющимися, то вывод сообщения

№	Проверяемая функция	Выполняемые действия	Результат
7.	Добавление	<p>Нажать кнопку «Добавить ЗК»</p> <p>Заполнить поля требуемой информацией</p> <p>Нажать кнопку добавить</p>	<p>Если новые номера не совпадают с уже имеющимися, то сохранение данных в базе, перезагрузка таблицы на главной странице, открытие главной формы.</p> <p>Если новые номера совпадают с уже имеющимися, то вывод сообщения</p>
8.	Расширенный поиск	<p>Нажать кнопку «Расширенный поиск ЗК»</p> <p>Ввод реквизитов</p> <p>Нажать кнопку «Выполнить поиск»</p>	<p>Если результаты получены, то вывод результатов на таблицу ЗК главной формы</p>
9.	Удаление данных из БД	<p>Выбрать ЗК из списка</p> <p>Нажать кнопку «Редактировать»</p> <p>поиском. Нажать кнопку «Выполнить».</p>	<p>Удаление строки из таблицы ЗК, затем из базы данных.</p>

№	Проверяемая функция	Выполняемые действия	Результат
10.	Анализ	Нажать кнопку «Анализ» Выбрать параметры анализа Выполнить анализ	Формирование справки с результатами анализа Вывод справки
11.	Экспорт	Нажать кнопку «Экспорт» Выбрать путь сохранения файла Выбрать экспортируемые данные Нажать на кнопку «Выполнить экспорт»	Сохранение файла с экспортированными данными по выбранному пути
12.	Импорт	Нажать кнопку «Импорт» Выбрать путь файла для импорта данных Нажать на кнопку «Выполнить Импорт»	Обновление таблицы ЗК на главной форме
13.	Служебные телефоны	Нажать кнопку «Служебные телефоны»	Открытие вкладки со списком служебных телефонов

ПРИЛОЖЕНИЕ В

Графическая часть

Перечень графических материалов:

1. Взаимодействие элементов предметной области
2. Инфологическая модель;
3. Схема взаимодействия с источниками данных
4. Схема взаимодействия рабочих станций «Записной книжки 2.0»
5. Структурная схема
6. Поиск первого варианта связей контактов
7. Поиск второго варианта связей контактов
8. Поиск третьего варианта связей контактов
9. Блок-схема работы системы анализа

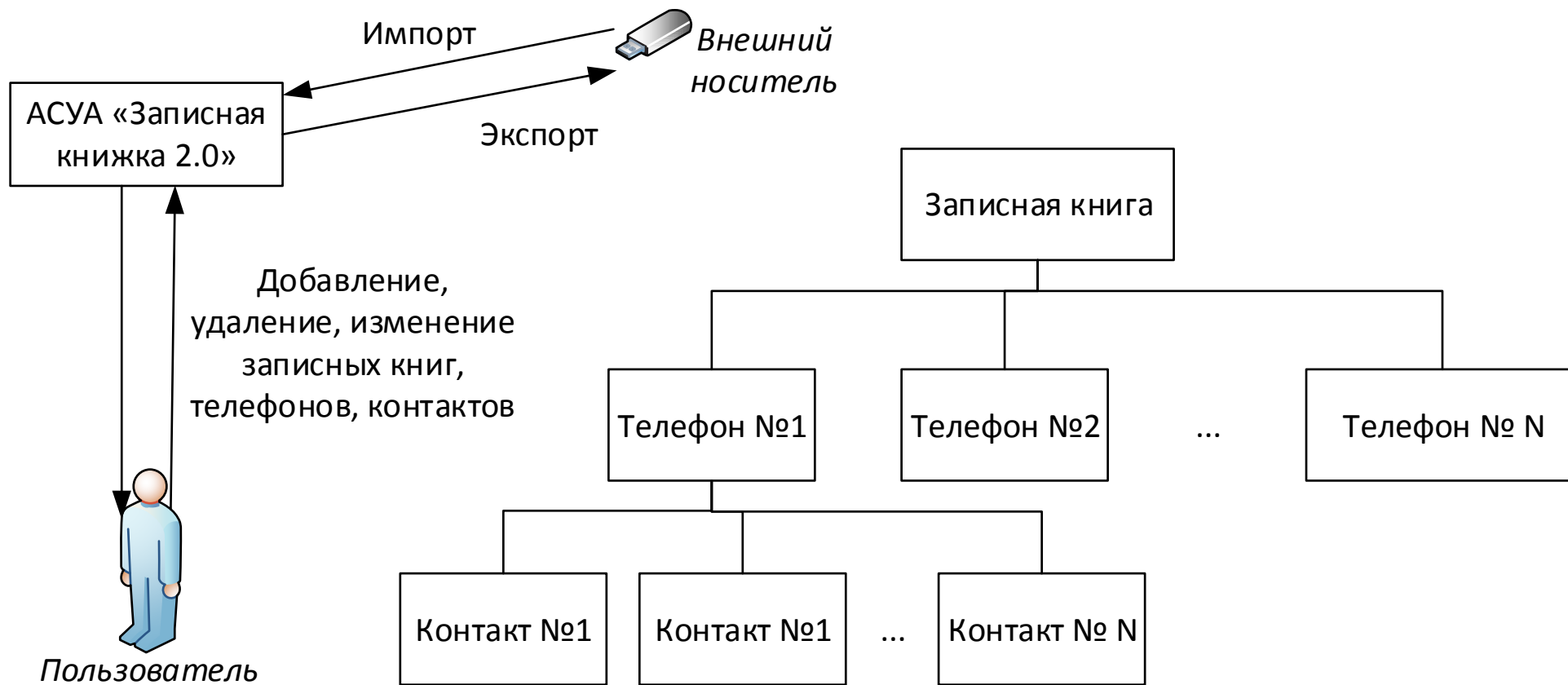


Рисунок В1 Взаимодействие элементов предметной области

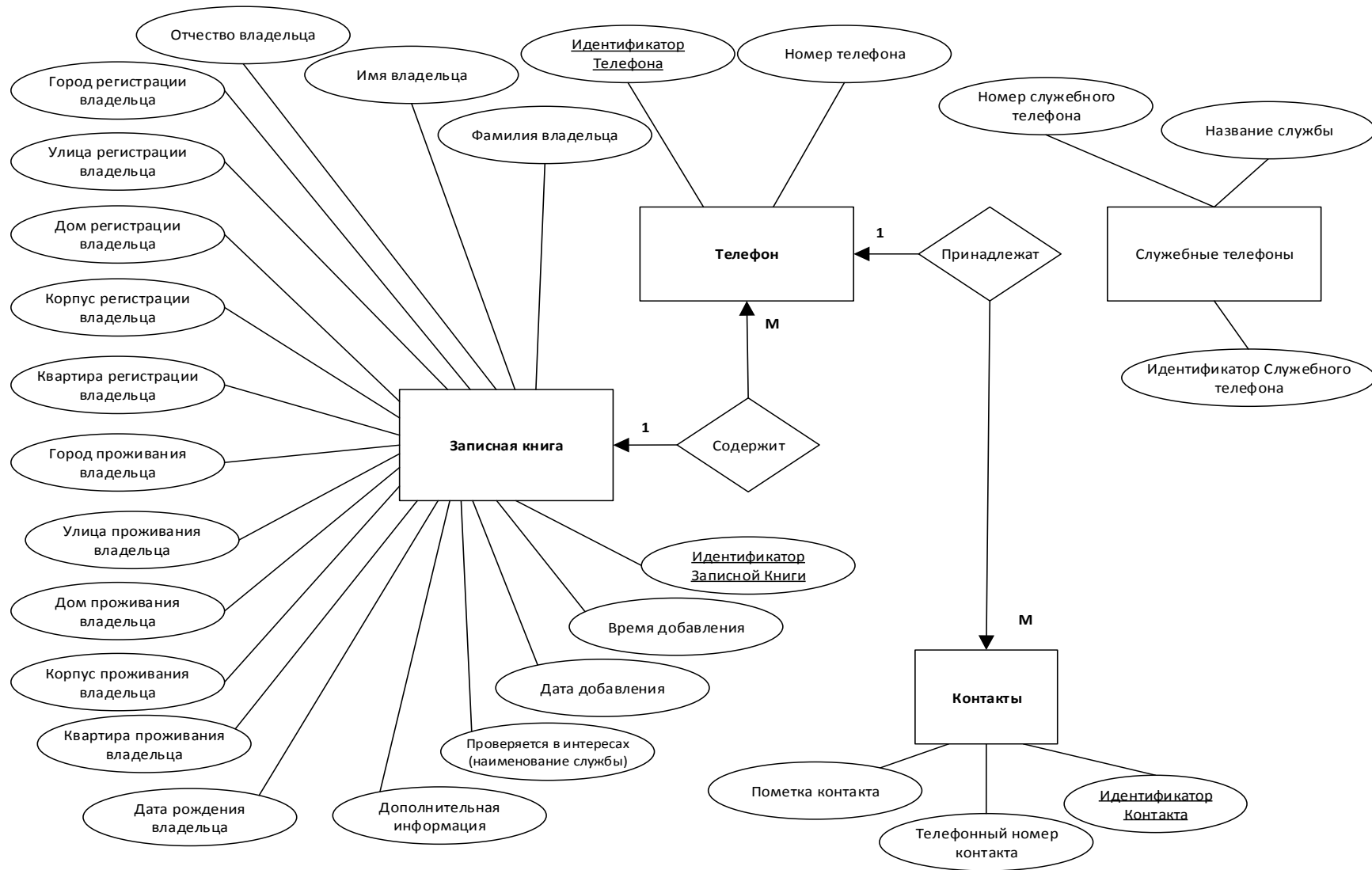


Рисунок В2 – Инфологическая модель

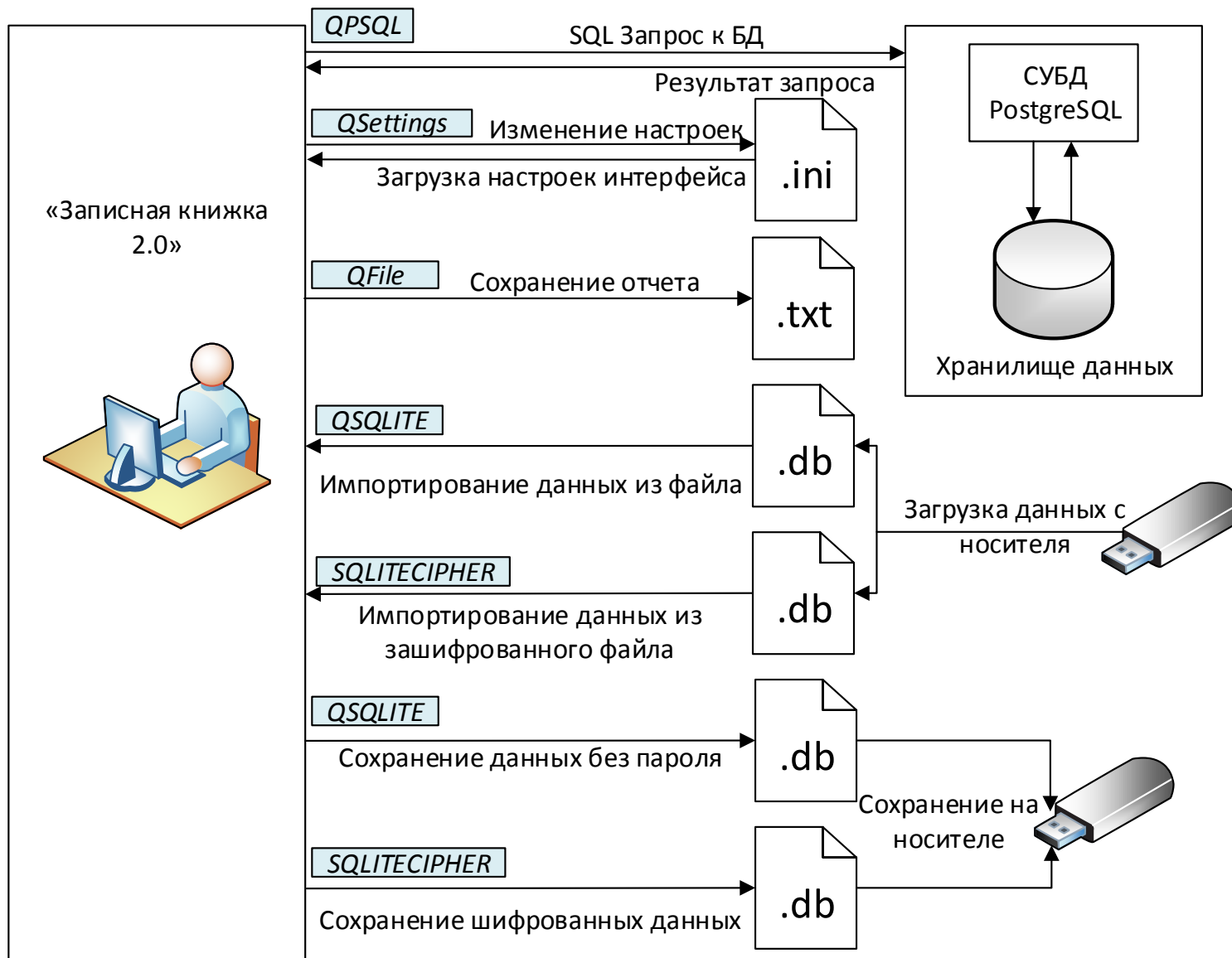


Рисунок В3. Схема взаимодействия с источниками данных.

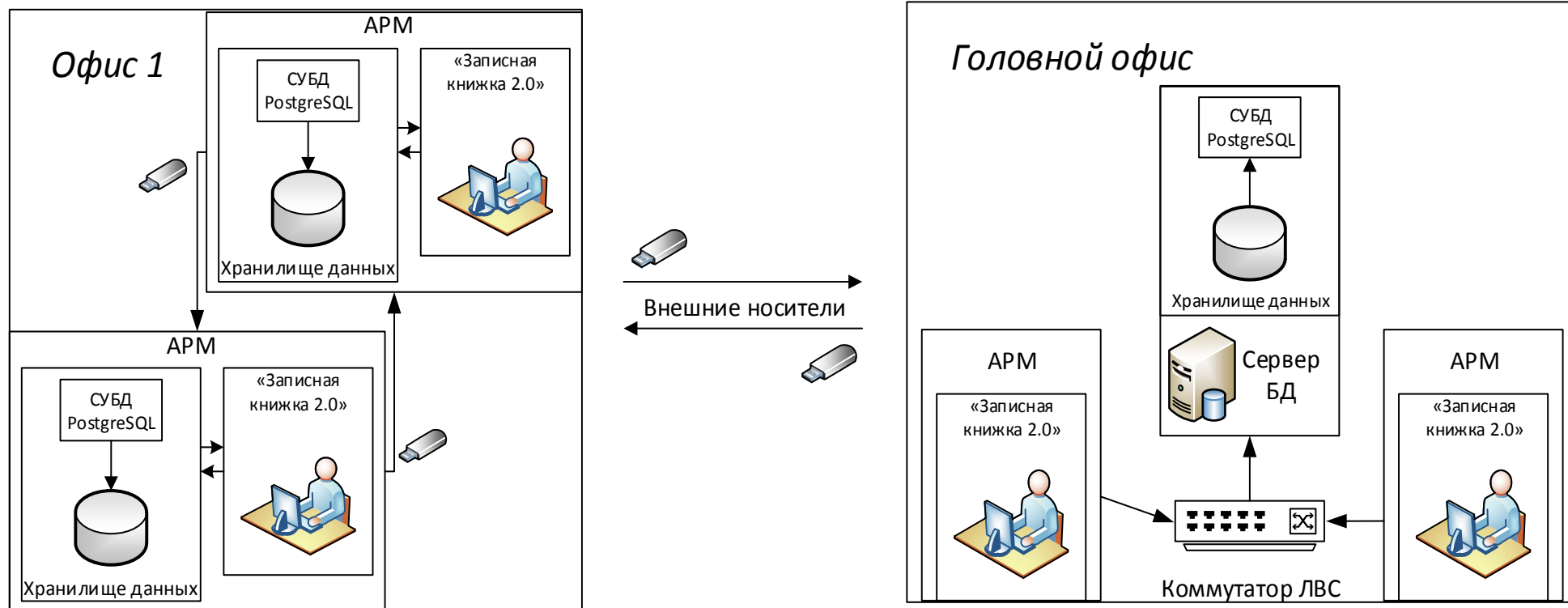


Рисунок В4. Схема взаимодействия рабочих станций «Записной книжки 2.0»

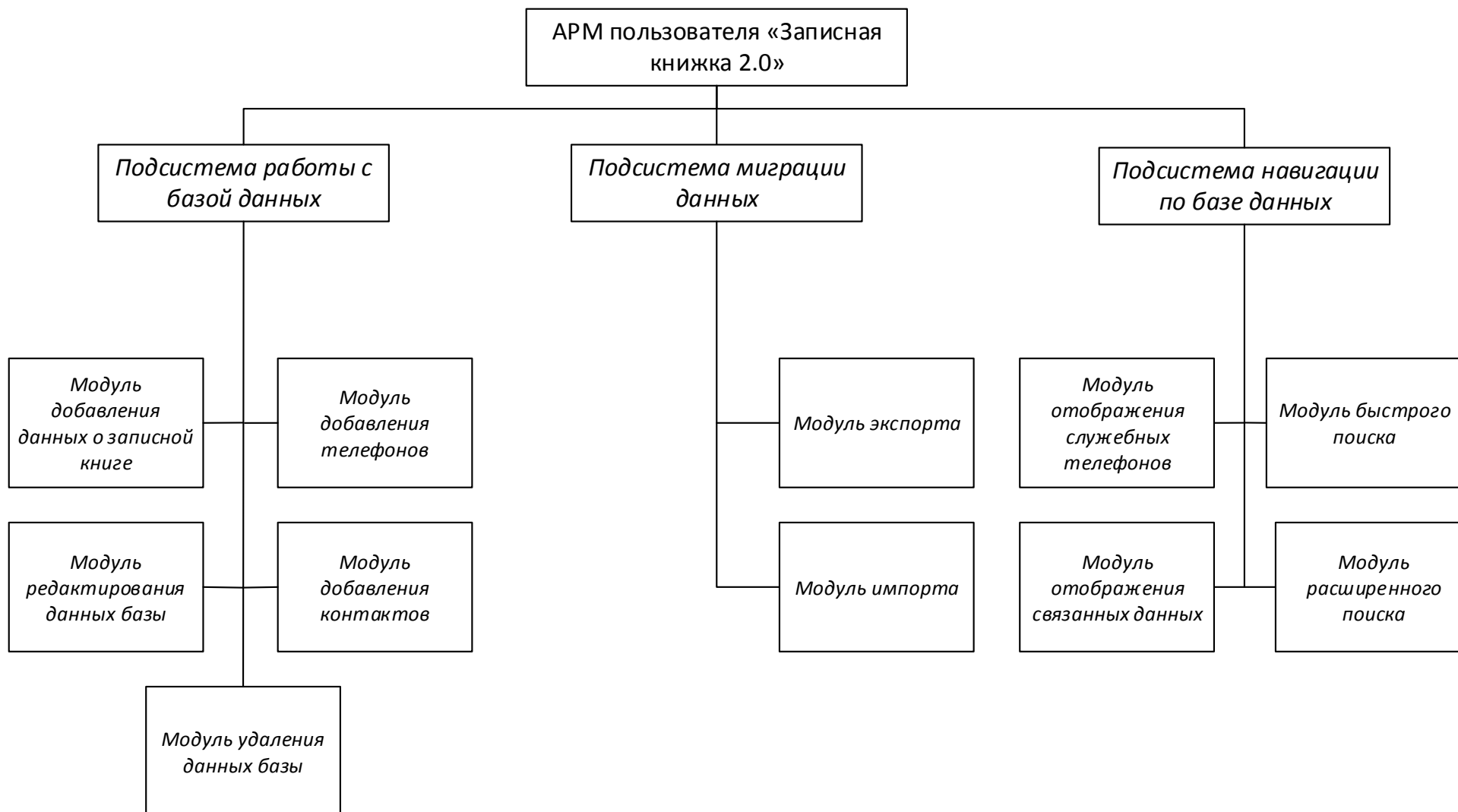


Рисунок В5 – Структурная схема

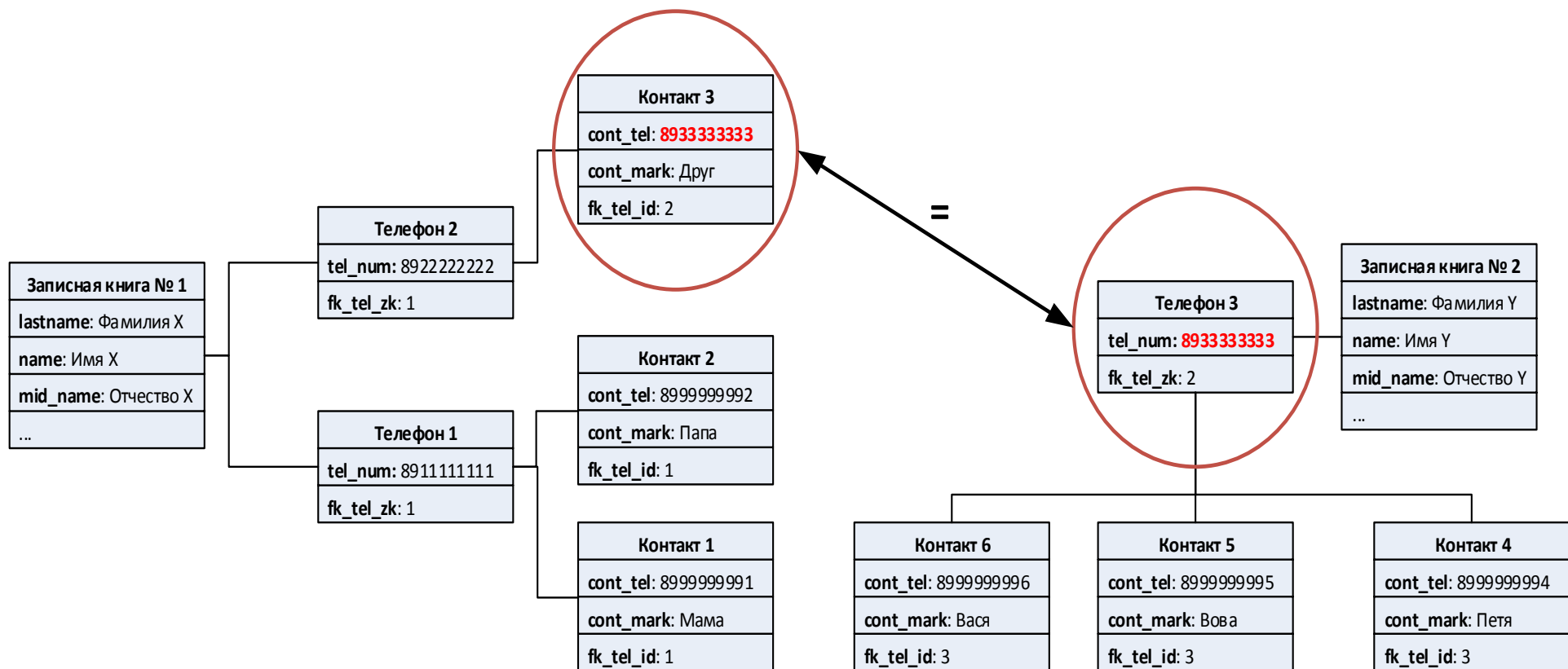


Рисунок В6 Поиск первого варианта связей контактов

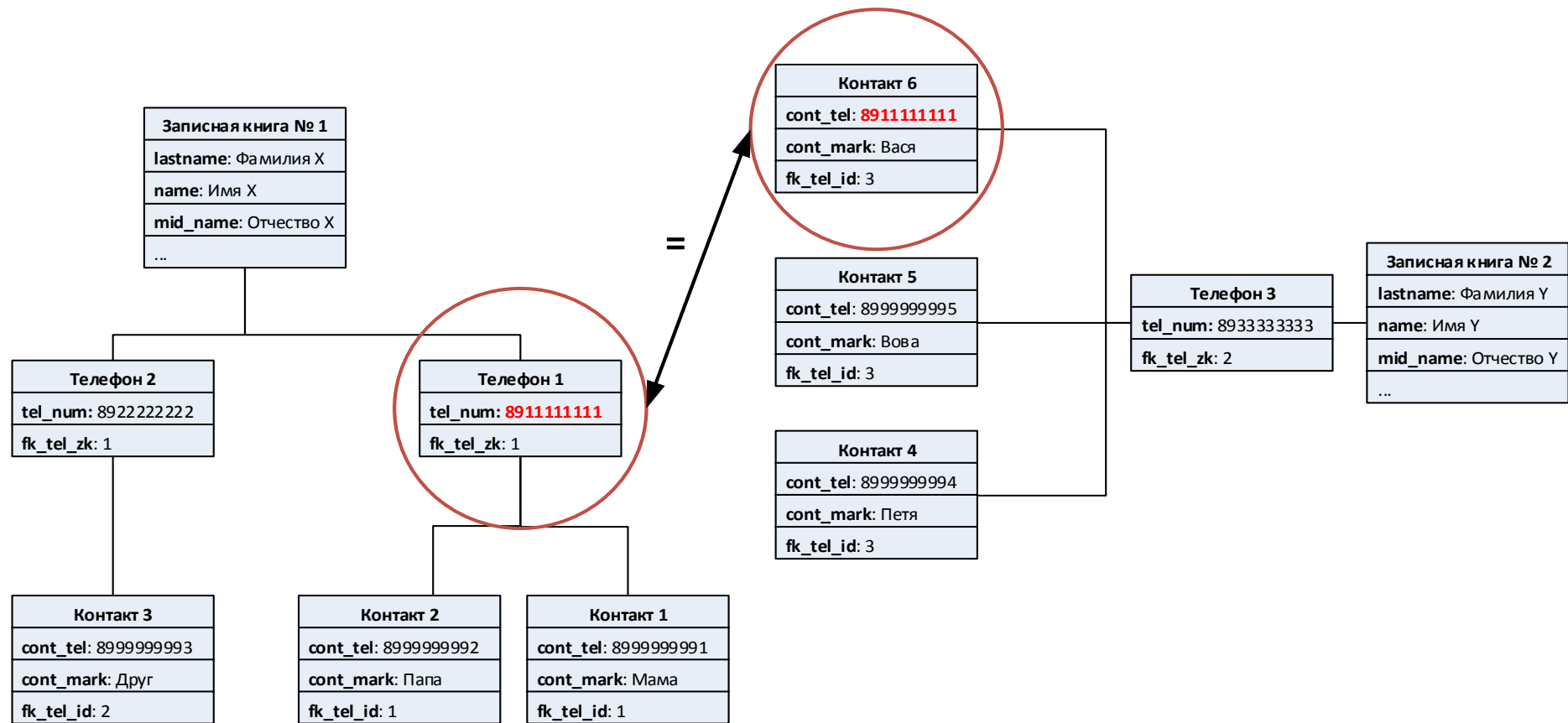


Рисунок В7 Поиск второго варианта связей контактов

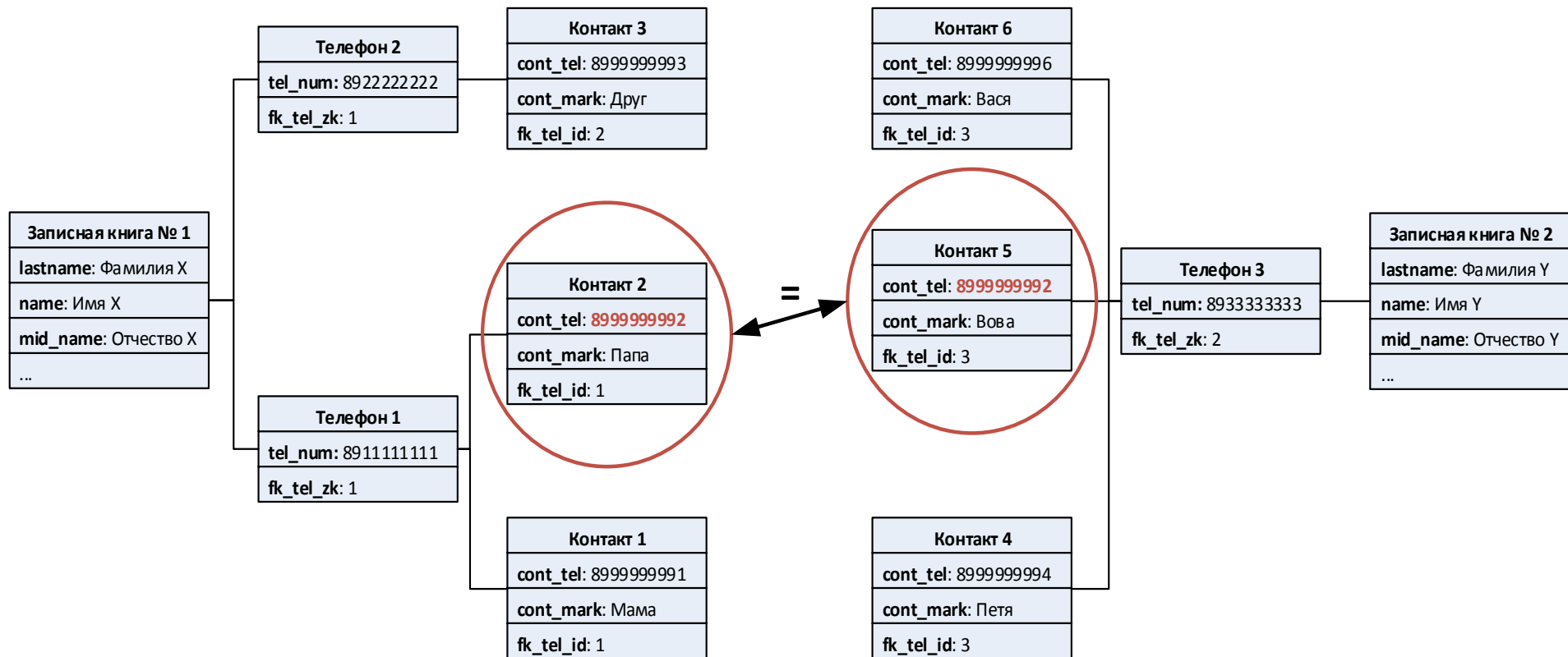


Рисунок В8 Поиск третьего варианта связей контактов

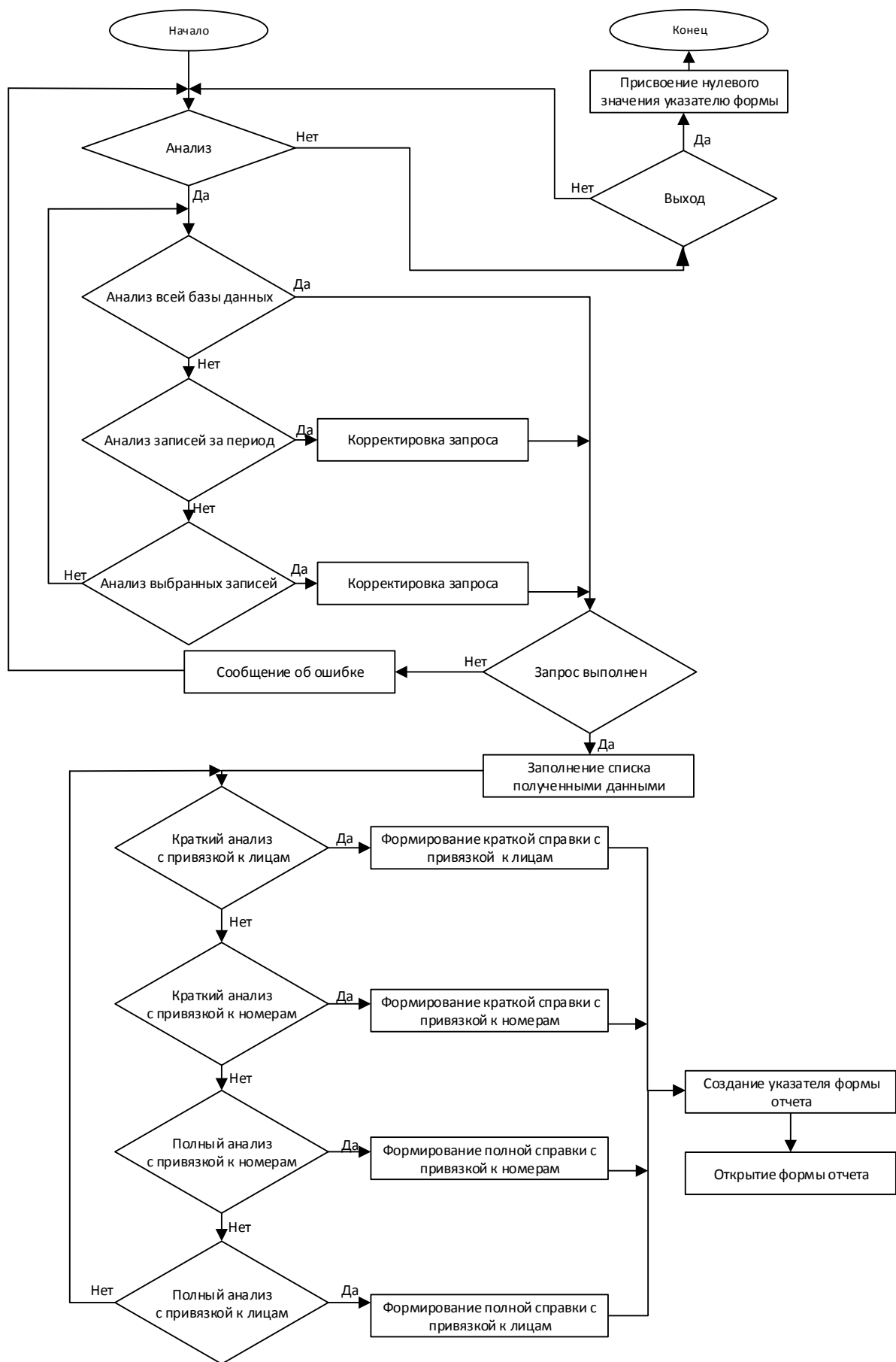


Рисунок В9 Блок-схема работы системы анализа