Оглавление

[1. Введение 4](#_Toc103532227)

[1.1. Обоснование выбора темы 4](#_Toc103532228)

[1.2. Развернутая оценка состояния решаемой задачи 4](#_Toc103532229)

[1.3. Определение актуальности и значимости 4](#_Toc103532230)

[1.4. Определение основной цели работы 4](#_Toc103532231)

[1.5. Планируемые результаты 4](#_Toc103532232)

[2. Анализ и проектирование предметной области предприятия 4](#_Toc103532233)

[2.1. Этапы разработки информационной системы. 4](#_Toc103532234)

[2.1.1. Анализ информационной системы 5](#_Toc103532235)

[2.1.2. Проектирование информационной системы 5](#_Toc103532236)

[2.1.3. Разработка приложения пользователя 5](#_Toc103532237)

[2.1.4. Оформление документации 5](#_Toc103532238)

[2.1.5. Сдача проекта 5](#_Toc103532239)

[**2.2.** **Сведения об использованных при проектировании нормативно-технической документации, стандартов** 9](#_Toc103532240)

[**2.3.** **Описание процесса деятельности разрабатываемой системы;** 10](#_Toc103532241)

[2.3.1. Виды автоматизированной деятельности 10](#_Toc103532242)

[2.3.2. Участники процесса обработки и использования ИС 10](#_Toc103532243)

[2.3.3. Управление информацией 10](#_Toc103532244)

[**2.4.** **Описание процедур обработки информации** 10](#_Toc103532245)

[2.4.1. Подготовка и размещения информации 10](#_Toc103532246)

[2.4.2. Добавление информации 11](#_Toc103532247)

[2.4.3. Редактирование информации 11](#_Toc103532248)

[2.4.4. Администрирование 11](#_Toc103532249)

[**2.5.** **Основные технические решения** 11](#_Toc103532250)

[2.5.1. Описание предметной области (IDEF0) 11](#_Toc103532251)

[2.5.2. Описание информационных потоков (IDEF3) 14](#_Toc103532252)

[2.5.3. Определить содержание функций обработки входной информации при решении задачи](#_Toc103532253)

[2.5.4. Выбор СУБД и обоснование 22](#_Toc103532254)

[2.5.5. Построение и тестирование физической модели 22](#_Toc103532255)

[**2.6.** **Определение требований к клиентскому приложению ИС.** 25](#_Toc103532256)

[2.6.1. Разработка технического задания 25](#_Toc103532257)

[2.6.2. Определить основные функции, реализуемые приложением (создать модели вариантов использования, определить действующих, сценарии использования) 29](#_Toc103532258)

[2.6.3. Сформировать альбом всех форм проектируемого приложения 29](#_Toc103532259)

[2.6.4. Сформировать отчеты приложения, построить макеты отчетов 30](#_Toc103532260)

[3. Разработка приложения пользователя. 32](#_Toc103532261)

[3.1. Разработать архитектуру приложения 32](#_Toc103532262)

[3.2. Составить перечень всех отчетов приложения, построить макеты отчетов; 33](#_Toc103532263)

[3.3. Определить независимые вспомогательные программы, необходимые для обслуживания приложения 37](#_Toc103532264)

[3.4. Построить главную форму приложения 37](#_Toc103532265)

[3.5. Построить формы, обеспечивающие ввод-модификацию данных; 38](#_Toc103532266)

[3.6. Создать набор отчетов для печати выходных данных 41](#_Toc103532267)

[3.7. Создать инструкции по использованию приложения и справочную систему 43](#_Toc103532269)

[4. Заключение 44](#_Toc103532270)

[5. Список литературы 45](#_Toc103532271)

[Приложение А Техническое задание 45](#_Toc103532272)

[Приложение Б Смета расходов на разработку ИС 48](#_Toc103532273)

[Приложение В (ER – диаграмма физической модели данных) 49](#_Toc103532274)

[Приложение Г Альбом форм проектируемого приложения 50](#_Toc103532275)

# Введение

## **Обоснование выбора темы;**

Тема моей курсовой работы «Информационная система регистратуры поликлиники» была выбрана на основе моего желания, т.к. сфера здравоохранения в нашем мире очень важна и значима. И, конечно, эта сфера должна быть наиболее автоматизирована и технологична.

## **Развернутая оценка состояния решаемой задачи;**

На данный момент не во всех больницах существует система автоматизации регистратуры или в общем всей поликлиники. Хотя, в нашей области встраивают систему «БАРС», в которой создаются отчёты, есть аутентификация и связь с главным сервером, но на данный момент она довольно ненадёжна и очень часто ломается, а также имеет очень плохую производительность. Ещё одна проблема то, что некоторые регионы не могут позволить себе такую большую АИС для своих поликлиник.

## **Определение актуальности и значимости;**

В данный момент приоритет автоматизации поликлиники повышен т.к. во время эпидемии небезопасно находиться большому скоплению людей в одном тесном помещении. Также и без эпидемии есть риск заразиться другими заболеваниями. Более того это служит экономией времени работников регистратуры и пациентов что приходят в поликлинику. Ещё при помощи такой АИС можно оперативно регулировать нагрузку на врачей.

## **Определение основной цели работы;**

Основные цели моего курсового проекта это:

1. Создание быстрой и надежной АИС для регистратуры поликлиники.
2. Создание дружелюбного интерфейса для пользователя.
3. Получить новые знания, которые могут понадобиться для таких же работ в будущем, научиться делать интерфейс и программировать.

## **Планируемые результаты;**

Планируемый результат – это, создание рабочей АИС регистратуры поликлиники и прилегающей к ней БД. Также чтобы она имела возможность вывода отчетов для персонала и направлений для пациентов. Сделать понятный интерфейс приложения для работы с отчетами и БД.

# Анализ и проектирование предметной области предприятия;

## Этапы разработки информационной системы

Этапы разработки автоматизированной информационной системы:

### Анализ информационной системы

Первым делом нужно выяснить что делает предприятие, какие услуги оно предоставляет. Также, какие данные входят на обработку и что выходит после обработки, куда эти данные отправляются и что с ним будут.

### Проектирование информационной системы;

На этом этапе определяем вид отчётов предприятия, которые создаёт предприятие. И выбираем СУБД, что подходит по всем нашим желаниям, а именно сколько мы готовы заплатить за СУБД, подходит ли она к данному предприятию, поддерживают ли наше аппаратное обеспечение эту СУБД.

### Разработка приложения пользователя;

В этот этап входит создание приложения от главной формы, до последней версии готовой к использованию. Мы будем разрабатывать все формы для работы с данными, главной формы, форм авторизации и регистрации, форм вывод отчётов.

### Оформление документации;

Приложения:

* Техническое задание
* ER-диаграмма
* Смета расходов на разработку ИС
* Альбом форм проектируемого приложения

В общем всё что указано по ГОСТ ‘у создания ПЗ

### Сдача проекта;

По завершению срока сдачи, необходимо предоставить заказчику (в нашем случае преподавателю): готовое к работе приложение с прилегающей к нему базой данных, документацию, включающую те пункты что я указал выше, а ещё пояснительную записку, презентацию нашего проекта. Сдача моего проекта 24 мая 2022 года.

Ниже на рис. 1 представлена часть диаграммы Ганта с описанием каждой задачи, которые необходимо выполнить, для создания нашего проекта.

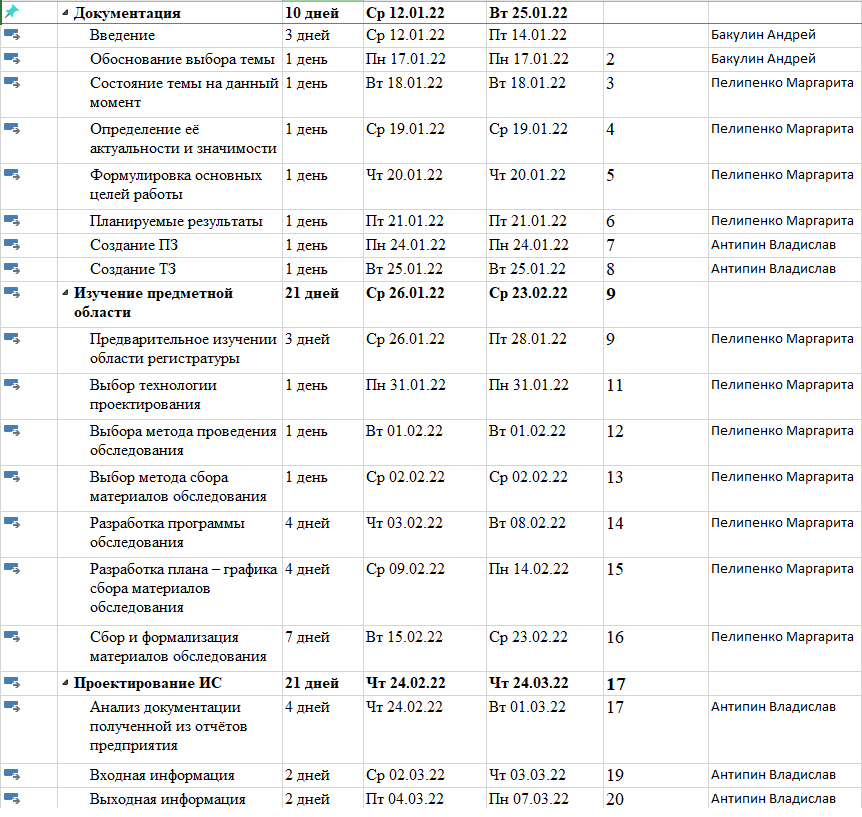


Рис. 1 задачи в диаграмме Ганта.

Для проектирования и кодирования необходимы работники, также для создания локальной сети нужен сервер, рабочие станции и лицензии на программы.

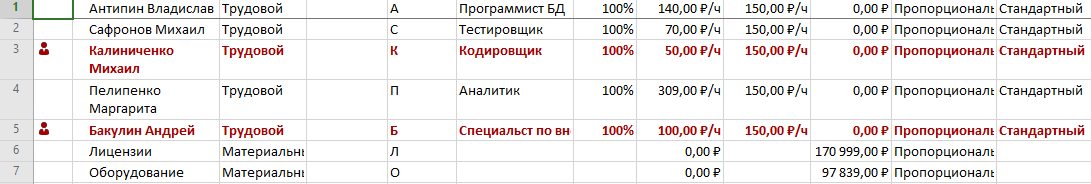


Рис. 2. Лист ресурсов.

## **Сведения об использованных при проектировании нормативно-технической документации, стандартов**

Проектирование моей АИС опиралось на ГОСТ’ы и стандарты. Вот те, что я использовал:

1. ГОСТ Р 54472-2011/ISO/TS 13606-4:2009- ПЕРЕДАЧА ЭЛЕКТРОННЫХ МЕДИЦИНСКИХ КАРТ
2. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.
3. Пояснительная записка к техническому проекту по ГОСТ 34.201-89.
4. Руководство пользователя по ГОСТ 34.201-89 и РД 50-34.698-90;
5. Руководство системного администратора по ГОСТ 19.504-79;

## **Описание процесса деятельности разрабатываемой системы;**

### Виды автоматизированной деятельности;

АИС для регистратуры поликлиники должно иметь возможность:

– обработки данных пациентов;

– обработки данных врачей;

– обработки данных расписания;

– обработки данных о услугах предоставляемых поликлиниках;

– обработки данных о кабинетах;

– вывода списков записанных пациентов на определённый день для врачей;

– вывода талонов для пациентов

Участники процесса обработки и использования АИС;

Главный регистратор – создание отчётов, просмотр всех таблиц, создание новых пользователей.

Регистратор – имеет простые права для изменения таблиц в БД и создания отчётов.

Системный администратор – наделён особыми правами, позволяющими ему делать всё что угодно в программе.

Гость – не имеет прав на изменение таблиц, не имеет своей учётной записи. Все пользователи, вошедшие в программу, считаются гостями до момента авторизации. Может открыть справку к программе и вкладку «Авторы».

Управление информацией;

Регистраторы с доступом к изменению информации могут делать поиск по данным, их также можно изменять, удалять, добавлять, выводить на печать в виде отчёта, сохранять в виде файлов Word или Excel.

Главный регистратор может создавать только талоны, остальные данные он может просматривать и выводить любые отчеты.

Системный администратор может делать всё что могут остальные участники управления информацией.

## **Описание процедур обработки информации;**

### Подготовка и размещения информации;

Все необходимые данные должны хранится в БД IB. Для построения наиболее правильной АИС необходимо выделить сервер, место под него и соединить все рабочие станции регистраторов с сервером.

### Добавление информации;

Добавление записей происходит через приложение, путём динамического создания команды на вставку (INSERT). Первичный ключ в таблице заполняется автоматически при помощи генератора, это сделано для того, чтобы было невозможно написать две одинаковые записи. Благодаря ограничений на полях ввода невозможно написать неправильные данные в БД.

### Редактирование информации;

Чтобы изменить уже существующую запись используется команда для изменения данных (UPDATE).

Чтобы удалить уже существующую запись используется команда для удаления данных (DELETE).

### Администрирование;

В моей БД есть 1 роль, регистраторы, т.к. все остальные пользователи встречаются лишь раз, а делать дли них роль, не имеет смысла. Пользователи без ролей:

1. Администратор.
2. Главный регистратор

## **Основные технические решения**

### Описание предметной области (IDEF0);

Изучив несколько регистратур поликлиник необходимо было выделить несколько основных характеристик:

* Деятельность регистратуры
* Оргштатная структура
* Входные данные
* Выходные данные
* Контроль управления данными
* С помощью чего могут обрабатывать полученные данные

Деятельность регистратуры поликлиники:

1. Выдача талонов с направлениями.
2. Добавление новых пациентов.
3. Создание списка пациентов, записанных на сегодняшний день для врача.

Основные задачи регистратуры поликлиники – принимать всех пришедших пациентов, ознакамливать их с информацией приема врачей и строго придерживаться методички общения с пациентами. Также обязанность регистратуры записывать всех пациентов, создавать запись в БД и создавать карту амбулаторного больного. Основная задача регистратуры - это запись пациентов к врачам, по телефону или прямо в отделении. Кроме всего этого регистратура составляет список пациентов в день записи для врачей, к которым они записаны.

Оргштатная структура - схема, в которой иерархически показаны работники регистратуры поликлиники (Рис.3.).

Численность сотрудников:

1. Главный регистратор.
2. Системный администратор.
3. Регистраторы.

**Организационная структура**:

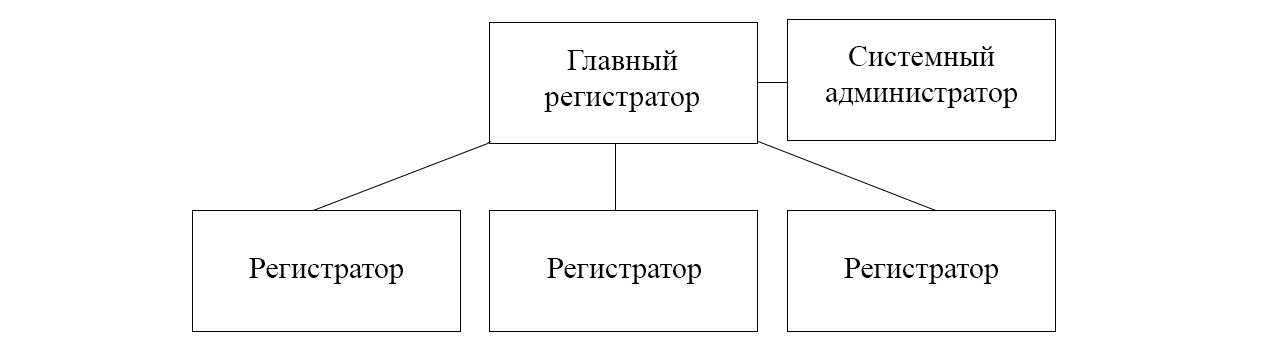


Рис. 3 Оргштатная структура.

IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. На Рис. 4 изображено*:*

1. Входные данные – это информация, над которой мы будем работать. В нашем случае это документы и информация о пациентах
2. Управление – это правила по которым мы будем работать с данными. В нашем случае это законодательство РФ, распорядок работы регистратуры
3. Механизм – это те, кто будет работать над данным или то, что будет работать над ними. В нашем случае это персонал регистратуры и БД.
4. Выходные данные – это то, что мы получили в итоге. В нашем случае это отчёты и талоны.

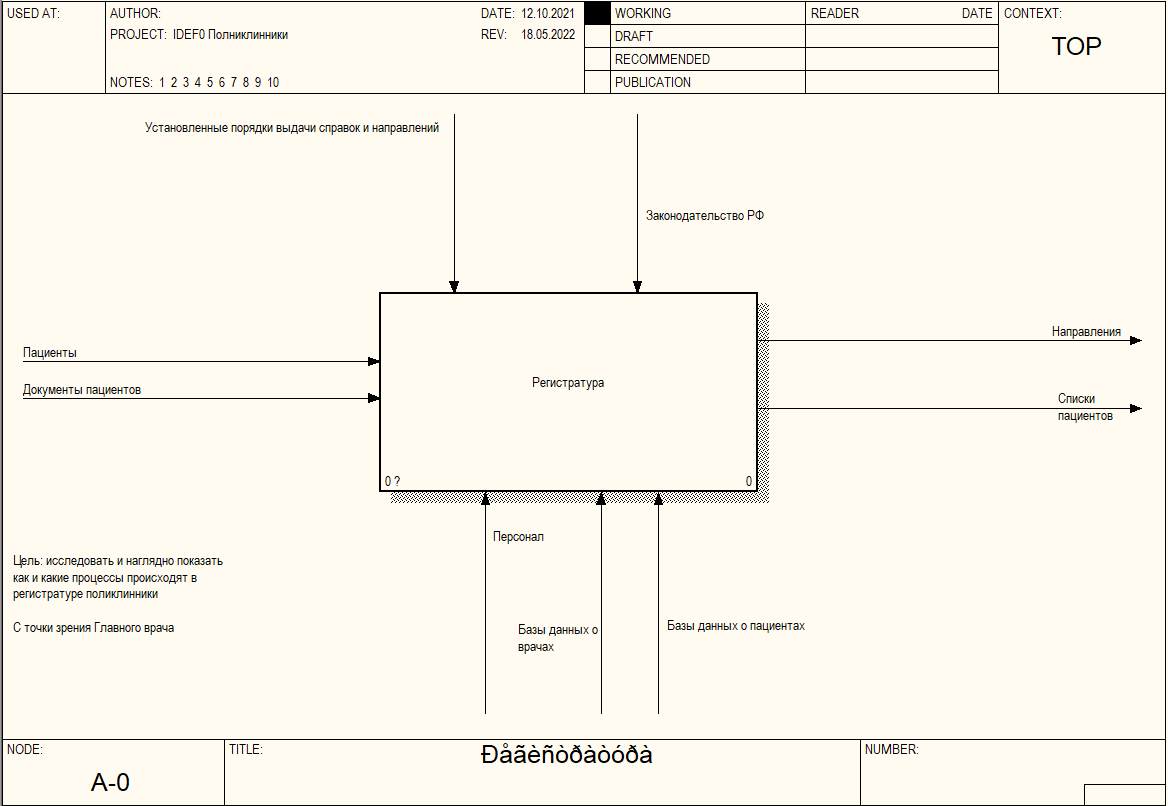


Рис. 4 Первый уровень IDEF0.

Декомпозиция:

1. Выбор услуги – регистраторы узнают у пациента какая услуга ему нужна, просят назвать своё ФИО и берут документы для записи.
2. Обработка данных пациента и данных для записи ко врачам выбор времени записи -Здесь регистраторы ищут время для пациента на запись к врачу, проверяют есть ли пациент в БД, если нет, то предлагают записаться. Всем этим управляет законодательство РФ, чтобы работа с данными пациента велась правильно.
3. Печать направления/Списка пациентов- если пациент не передумал или не захотел записываться в поликлинику, то мы выводим на печать талон для записи, с которым пациенту необходимо обратиться к врачу в назначенное время в талоне. Или если уже кончилось время на запись к врачу, то печатается список записанных пациентов и передаётся врачу. Пациент уходит.

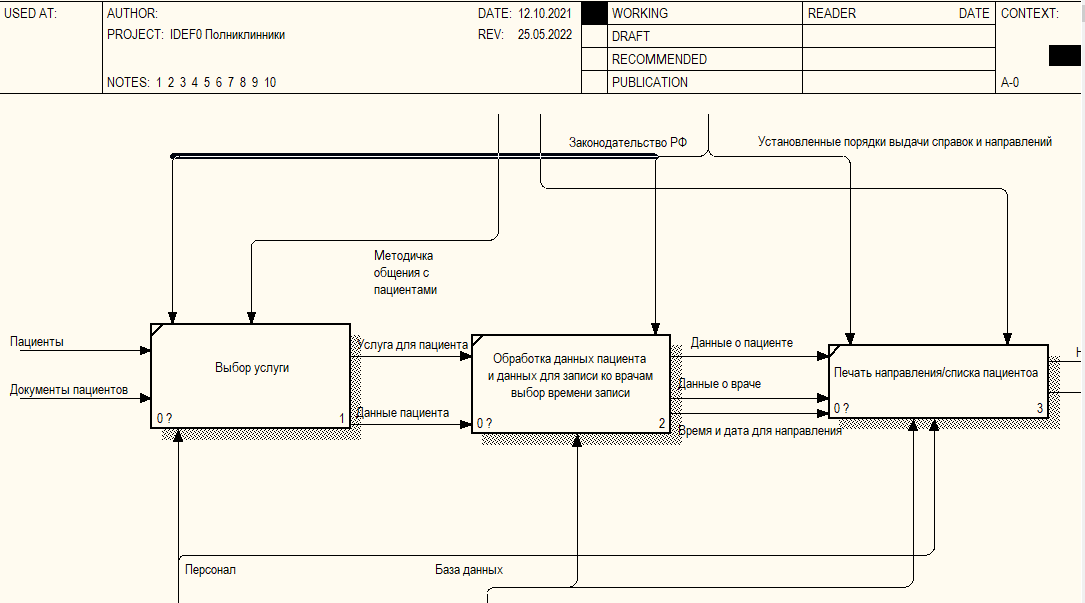


Рис. 5 Декомпозиция IDEF0.

### Описание информационных потоков (IDEF3);

На Рис. 6 отображён процесс выдачи направления. Пациент проходит через живую очередь или звонит в регистратуру. Мы выясняем приписан ли к нам пациент. Если да, то мы выбираем услугу для пациента. Позже мы ищем время для приёма. Если время подходит, то записываем. Если не подходит, пациент уходит. Если пациент не приписан к нам, то мы предлагаем записаться. Если пациент не хочет приписываться, то он уходит. Если пациент хочет приписаться, то мы продолжаем с момента выбора временя записи к врачу.

Пациент уходит с записью.

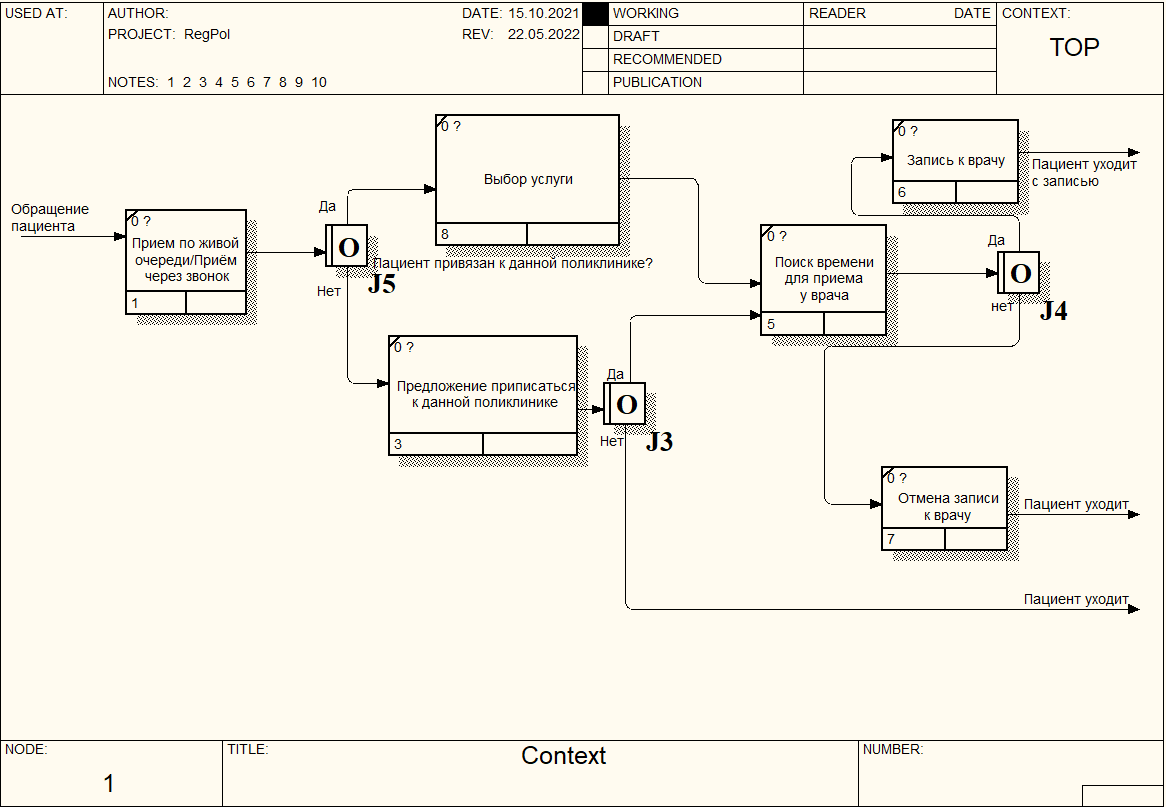


Рис. 6. IDEF3.

### Определить содержание функций обработки входной информации при решении задачи;

Определение происходит в самой программе, где в полях можно писать только данные, которые должны быть в БД (в поля имени нельзя написать цифры или специальные символы). Также невозможно оставить пустым поле, которое обязательно для заполнения. Но не только в программе происходит проверка, проверка происходит и в самой БД где невозможно написать в поля пустые значения, где это запрещено. И тоже самое что невозможно написать отличный тип данных от типа данных в поле.

ER диаграмма и описание физической модели АИС;

Построение модели данных:

* На основе документации формируемой предприятием проектируем журналы операций.
* На основе данных в документах делаем сущности-справочники для БД

На Рис.7 изображён талон пациентов, от которого мы будем отталкиваться для создания журнала операций

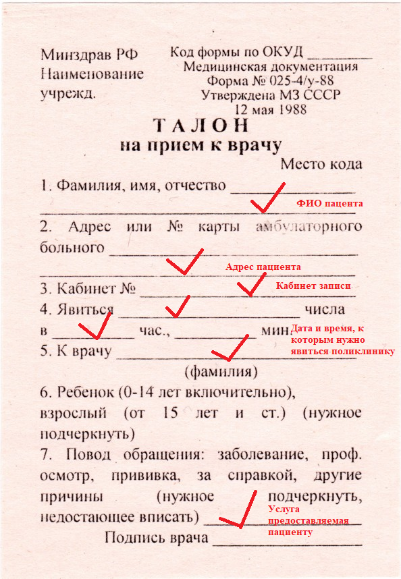


Рис.7. Талон записи пациента

При изучении предметной области были выделены поля, которые будут использоваться для построения БД и ER-Диаграммы, но сначала необходимо создать концептуальную модель данных Таб.1, Таб.2.

Спецификация:

Таб. 1. Концептуальная модель.

| **Наименование атрибута** | **Назначение атрибута** |
| --- | --- |
| FName | Фамилия пациента |
| Name | Имя пациента |
| OtchName | Отчество пациента |
| Tel | Телефон |
| Date\_Born | День рождения пациента |
| Pol | Пол пациента |
| OMS | ОМС Пациента |
| Email | Электронная почта пациента |
| Med\_usl | Медицинская услуга |
| FName\_Doc | Фамилия доктора |
| Name\_Doc | Имя доктора |
| OtchName\_Doc | Отчество доктора |
| TimeZap | Время записи в талоне |
| DAYS\_START | День начала работы |
| DAYS\_END | День окончания работы |
| Hours\_START | Часы начала работы |
| Hours\_END | Часы окончания работы |
| ID\_Cart | Номер амбулаторной карты |
| Street | Улица проживания пациента |
| Apart | Номер квартиры пациента |
| BUILDING | Номер дома пациента |
| DataTALON | Время создания талона |
| Name\_cab | Название кабинета |
| DateZap | Дата записи в талоне |

Таб. 2. Поля без нормализации.

| **Наименование атрибута** | **Назначение атрибута** | **Тип данных** |
| --- | --- | --- |
| FName | Фамилия пациента | Varchar[30] |
| Name | Имя пациента | Varchar[30] |
| OtchName | Отчество пациента | Varchar[30] |
| Tel | Телефон | Varchar[11] |
| Date\_Born | День рождения пациента | Date |
| Pol | Пол пациента | Varchar[1] |
| OMS | ОМС Пациента | Varchar[16] |
| Email | Электронная почта пациента | Varchar[30] |
| Med\_usl | Медицинская услуга | Varchar[50] |
| FName\_Doc | Фамилия доктора | Varchar[30] |
| Name\_Doc | Имя доктора | Varchar[30] |
| OtchName\_Doc | Отчество доктора | Varchar[30] |
| Cab | ID Кабинета | INTEGER |
| TimeZap | Время записи в талоне | TIME |
| ID\_Pacient | ID пациента | INTEGER |
| ID\_ Med\_usl | ID медицинской услуги | INTEGER |
| ID\_Doctor | ID Врача | INTEGER |
| ID\_RASP | ID расписания | INTEGER |
| DAYS\_START | День начала работы | Varchar[11] |
| DAYS\_END | День окончания работы | Varchar[11] |
| Hours\_START | Часы начала работы | TIME |
| Hours\_END | Часы окончания работы | TIME |
| ID\_Cart | Номер амбулаторной карты | Varchar[5] |
| Street | Улица проживания пациента | Varchar[30] |
| Apart | Номер квартиры пациента | Varchar[10] |
| BUILDING | Номер дома пациента | Varchar[10] |
| DataTALON | Время создания талона | TIMESTAMP |
| ID\_talon | ID талона | INTEGER |
| Name\_cab | Название кабинета | Varchar[40] |
| DateZap | Дата записи в талоне | Date |

**Нормализация БД**

**Нормализация** – это метод проектирования базы данных, который позволяет привести базу данных к минимальной избыточности.

**Цель нормализации**: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении кортежей (строк таблицы).

**Первая нормальная форма (1НФ)**- Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

**Вторая нормальная форма (2НФ)**- Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от первичного Ключа (ПК).

**Третья нормальная форма (3НФ)**- Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут не транзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Как итог я выделил все необходимые сущности для построения БД и составил: типы сущностей, типы связей и атрибуты сущностей Таб.3, Таб.4, Таб.5.

Таб. 3. Сущности

| **Имя сущности** | **Назначение** |
| --- | --- |
| CABINET | Кабинеты в поликлинике |
| DOCTOR | Врачи поликлиники |
| PACIENT | Пациенты приписанные к поликлинике |
| TALON | Талоны созданные для пациентов |
| RASP | Расписание, по которому работают врачи |
| USLUGA | Услуги предоставляемые в поликлинике |

**Сведение о типах связей:**

Таб. 4. Связи

|  |  |  |  |
| --- | --- | --- | --- |
| Тип сущности | Тип связи | Тип сущности | Кардинальность |
| PACIENT | Указывается в | TALON | 1:M |
| CABINET | Указывается в | TALON | 1:M |
| USLUGA | Указывается в | CABINET | 1:M |
| DOCTOR | Указывается в | CABINET | 1:M |
| RASP | Указывается в | DOCTOR | 1:M |

**Сведения об атрибутах сущностей:**

Таб. 5. Спецификация модели данных.

| **Имя сущности** | **Поля** | **Описание** | **Тип данных** | **Ограничение** | **Возможность NULL** |
| --- | --- | --- | --- | --- | --- |
| PACIENT | ID\_PACIENT | Идентификатор  таблицы | INTEGER | PK | NO |
| FNAME | Фамилия пациента | Varchar[30] |  | NO |
| NAME | Имя пациента | Varchar[30] |  | NO |
| OTCHNAME | Отчество пациента | Varchar[30] |  | NO |
| TEL | Телефон пациента | Varchar[11] |  | NO |
| DATE\_BORN | Дата рождения пациента | DATE |  | NO |
| POL | Пол пациента | Varchar[1] |  | NO |
| OMS | ОМС пациента | Varchar[16] |  | NO |
| EMAIL | Электронная почта пациента | Varchar[30] |  | YES |
| STREET | Улица проживания пациента | Varchar[30] |  | NO |
| BUILDING | Номер дома пациента | Varchar[10] |  | NO |
| APART | Квартира пациента | Varchar[8] |  | NO |
| ID\_CART | Номер амбулаторной карты | Varchar[5] |  | NO |
| RASP | DAY\_START | День начала работы | Varchar[11] |  | NO |
| DAY\_END | День окончания работы | Varchar[11] |  | NO |
| HOURS\_START | Часы начала работы | TIME |  | NO |
| HOURS\_END | Часы окончания работы | TIME |  | NO |
| ID\_RASP | Идентификатор  таблицы | INTEGER | PK | NO |
| USLUGA | ID\_MED\_USL | Идентификатор  таблицы | INTEGER | PK | NO |
| MED\_USL | Название услуги | Varchar[50] |  | NO |
| DOCTOR | ID\_DOCTOR | Идентификатор  таблицы | INTEGER | PK | NO |
| FNAME\_DOC | Фамилия врача | Varchar[30] |  | NO |
| NAME\_DOC | Имя врача | Varchar[30] |  | NO |
| OTCHNAME\_DOC | Отчество врача | Varchar[30] |  | NO |
| ID\_RASP | Подключение таблицы расписания | INTEGER | FK | YES |
| CABINET | CAB | Идентификатор  таблицы | INTEGER | PK | NO |
| NAME\_CAB | Название кабинета | Varchar[40] |  | NO |
| ID\_DOCTOR | Подключение таблицы с врачами | INTEGER | FK | YES |
| ID\_MED\_USL | Подключение таблицы С услугами | INTEGER | FK | YES |
| TALON | ID\_TALON | Идентификатор  таблицы | INTEGER | PK | NO |
| DATEZAP | Дата записи | DATE |  | NO |
| TIMEZAP | Время записи | TIME |  | NO |
| DATATALON | Дата создания талона | TIMESTAMP |  | NO |
| ID\_PACIENT | Подключение таблицы с пациентами | INTEGER | FK | YES |
| CAB | Подключение таблицы с кабинетами | INTEGER | FK | YES |

Модель данных: на Рис. 8 графически показаны взаимодействия данных, их разделение по таблицам. Здесь видно сколько таблиц получилось в итоге.

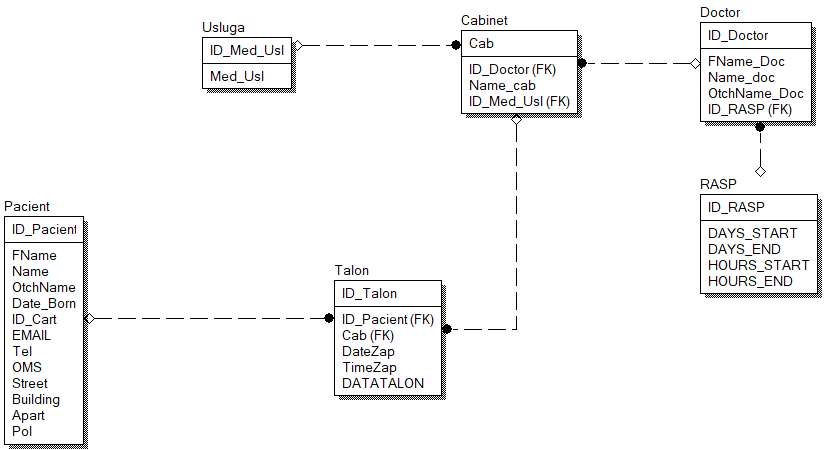


Рис.8. Модель данных

В итоге мы получили модель данных в 3-ей нормальной форме.

### Выбор СУБД и обоснование

На выбор есть несколько СУБД для БД:

1. **Microsoft SQL Server**
2. MySQL
3. Interbase

Мной была выбрана СУБД InterBase т.к. она довольно проста в управлении и использовании. Также она идеально подходит к Delphi из-за того, что они имеют одного разработчика.

### Построение и тестирование физической модели;

За пример возьмём таблицу(сущность) пациент, на котором я покажу код создания таблицы.

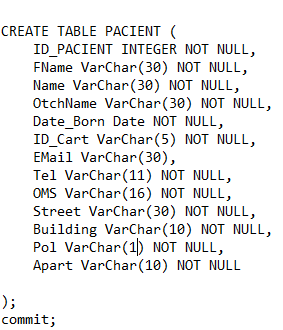


Рис.9. SQL код создания таблицы пациентов

Как мы создали таблицы их надо соединить связями и назначить первичными ключами нужные нам поля. Т.е. ниже мы устанавливаем реляционные связи между таблицами ( Рис.10.)

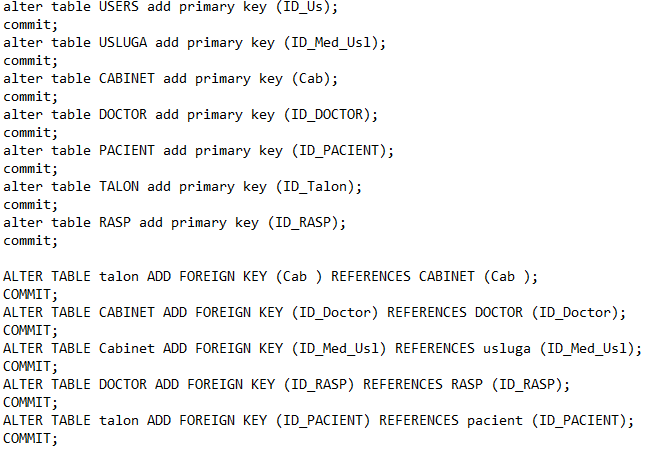


Рис.10. код SQL создания первичных ключей и реляционных связей связями таблиц

Кроме всего этого, чтобы все первичные ключи отличались друг от друга чтобы удобнее было в будущем добавлять данные мы создадим генераторы для каждого первичного ключа   
Рис. 11.

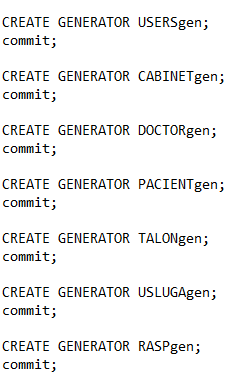


Рис.11. код создания генераторов

Чтобы проверить работу приложения нам необходимо заполнить её данными, на Рис. 12 будет пример заполнения таблицы пациентов.

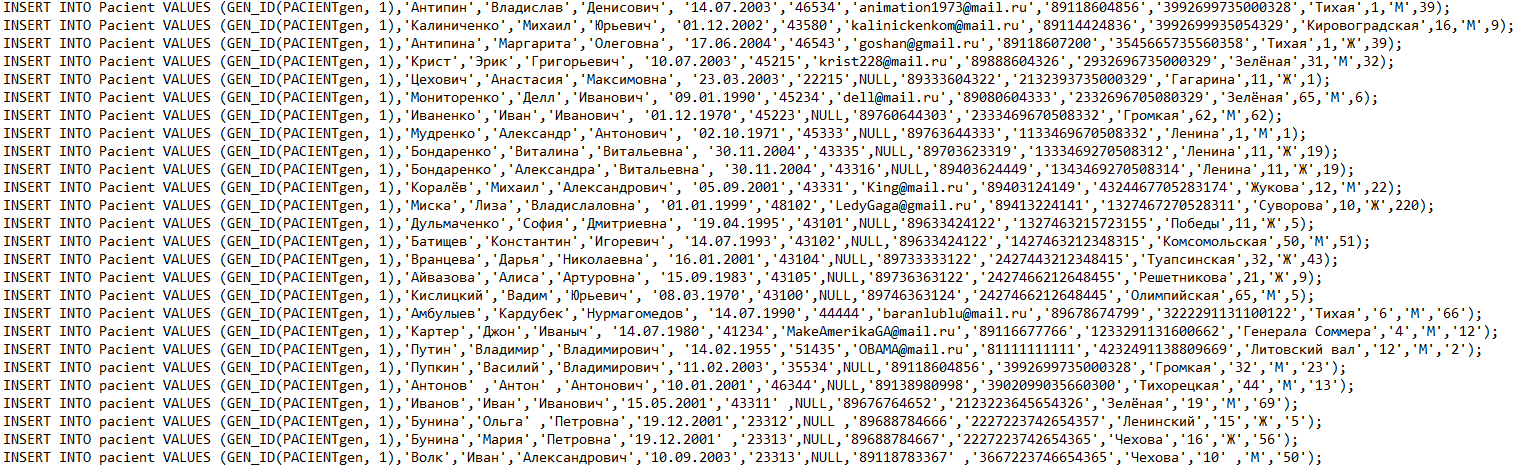


Рис.12. заполнение таблицы пациентов.

**Тестирование БД.**

Для теста созданной БД необходим написать запрос (Рис. 13), который покажет какие-либо данные, если они правильные, то БД функционирует правильно. Я специально возьму запрос для талона, т.к. он затрагивает абсолютно все таблицы:

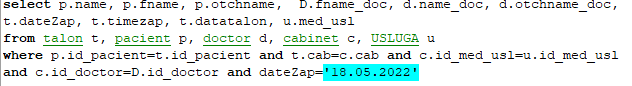


Рис.13. Запрос

Как видно (Рис. 14) вывелись все пациенты, записанные на 18 мая 2022 года. Следовательно, мы можем приступать к следующему этапу

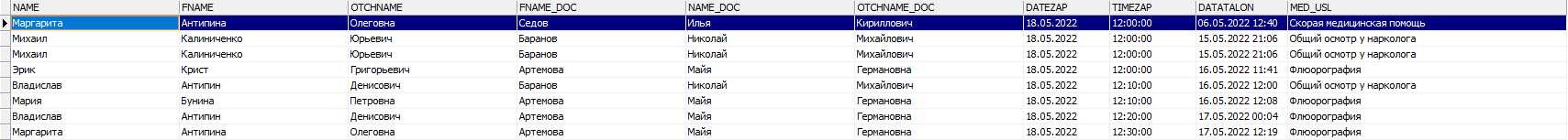


Рис.14. Результат запроса

## **Определение требований к клиентскому приложению ИС.**

### Разработка технического задания

Функции, которыми будет обладать приложение:

– обработка данных пациентов;

– обработка данных врачей;

– обработка данных расписания;

– обработка данных о услугах предоставляемых поликлиниках;

– обработка данных о кабинетах;

– вывода списков записанных пациентов на определённый день для врачей и его печать;

– вывода талонов для пациентов и его печать.

Для более подробного технического задания, смотри приложение А.

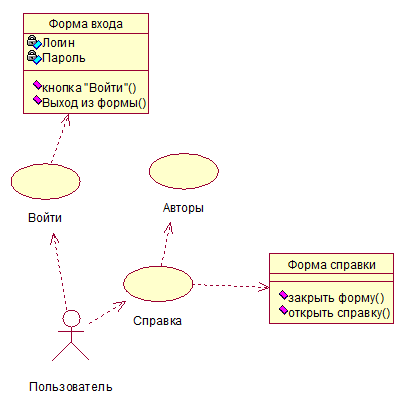


Рис.15. Диаграмма вариантов использования для обычного пользователя, до входа

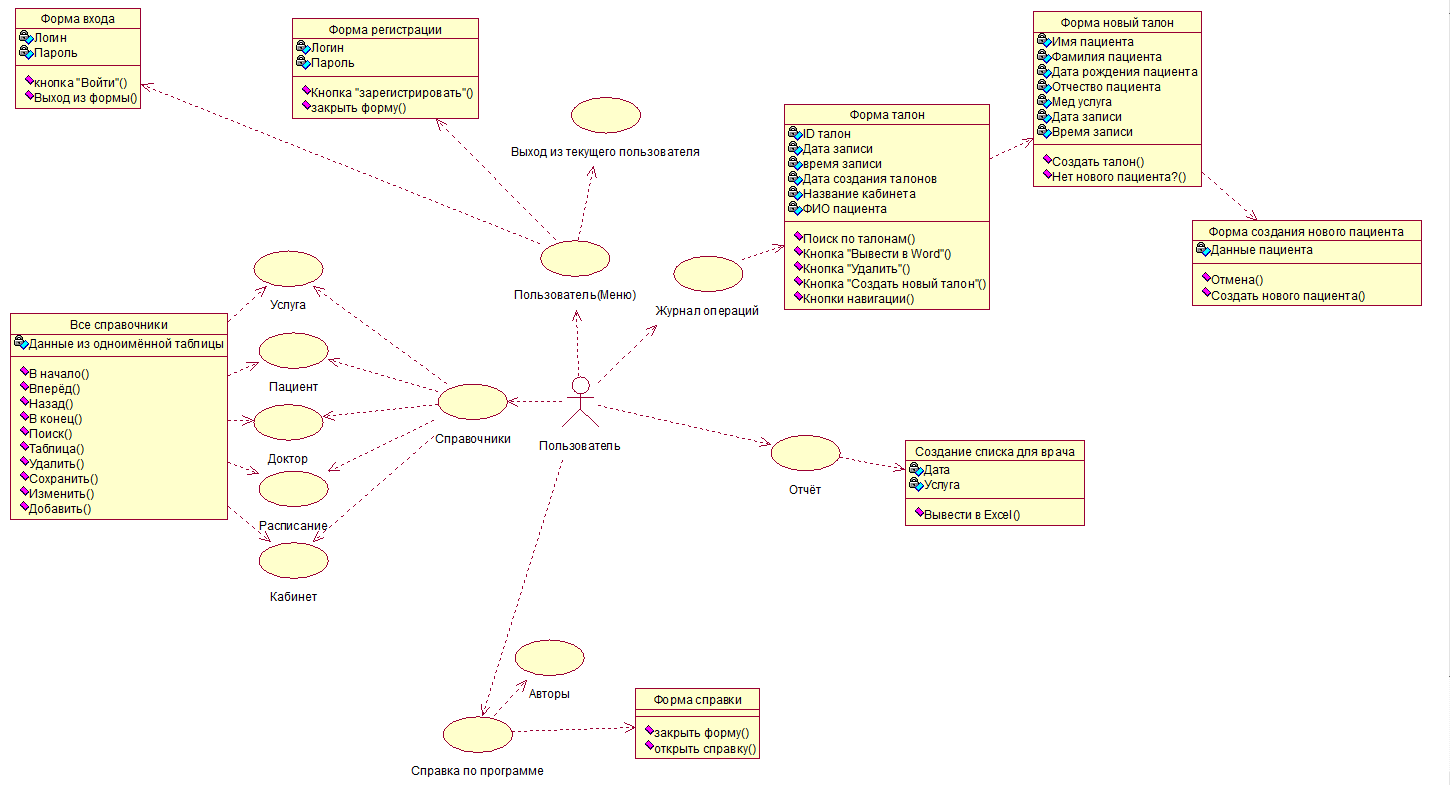


Рис.16. Диаграмма вариантов использования для пользователя, который уже вошёл

Кнопка «Войти»-кнопка для входа после заполнения данных

Кнопка «вперёд» - переходим на следующая запись, изменяет данные таблицы, записанные в полях справочника

Кнопка «В конец» - переходим на последнюю запись, изменяет данные таблицы, записанные в полях справочника

Кнопка «В начало» - переходим на первую запись, изменяет данные таблицы, записанные в полях справочника

Кнопка «Назад» - переходим на предыдущую запись, изменяет данные таблицы, записанные в полях справочника

Кнопка «открыть справку» - открывают Help- справку формата CHM

Кнопка «Вывести в Word» - выводит выбранные данные в виде отчёта в формате docx при помощи шаблона

Кнопка «Вывести в excel» - выводит выбранные данные в виде отчёта в формате Xlsx при помощи шаблона

Кнопка «Создать нового пациент» - читает введённые данные и делает из них запрос на вставку в БД

Кнопка «Зарегистрировать»-кнопка для регистрации нового пользователя после заполнения данных.

А теперь диаграмма Use Case становится основанием для создания форм.

### Сформировать альбом всех форм проектируемого приложения;

Для примера справочников возьмём справочник пациентов (Рис. 17), т.к. он самый большой. Я не буду показывать все справочники т.к. они все одинаковые по функционалу, а меняются только данные.

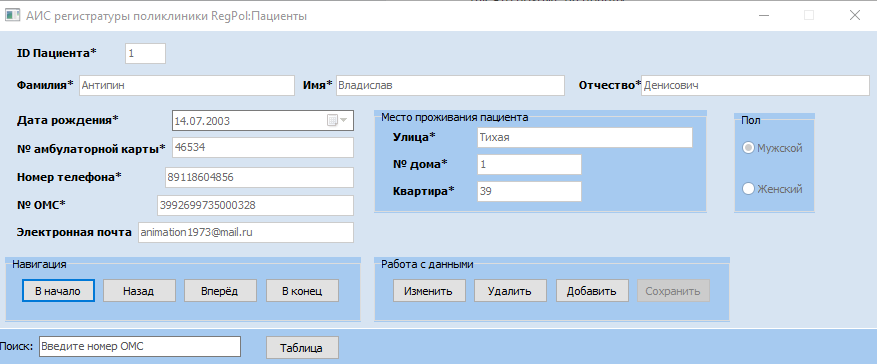


Рис. 17. Справочник пациент.

Здесь мы можем наблюдать: кнопки навигации, а именно

* В начало – к первой записи
* Назад – на запись назад
* Вперёд – на запись вперёд
* В конец – на последнюю запись

Кнопки работы с данными, а именно:

* Изменить – позволяет изменить выбранную запись
* Удалить – удаляет выбранную запись
* Добавить – даёт возможность сделать новую запись
* Сохранить – сохраняет внесённые изменения

Поле поиска, в котором нужно писать данные указанного поля и будет происходит поиск, а также кнопка таблица, которая при нажатии показывает таблицу со всеми записями данного справочника.

Остальные формы приложения находятся в приложении Г.

### Сформировать отчеты приложения, построить макеты отчетов.

Шаблон талона для пациентов (Рис.18).  
На нём есть все необходимые поля пациента, чтобы он не забыл, что от него требуется.

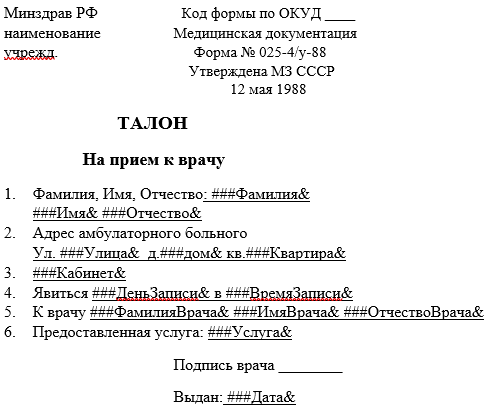


Рис.18. Шаблон талона для пациентов

Ниже (Рис.19) шаблон списка записанных пациентов на какой-либо день для врача в Excel.

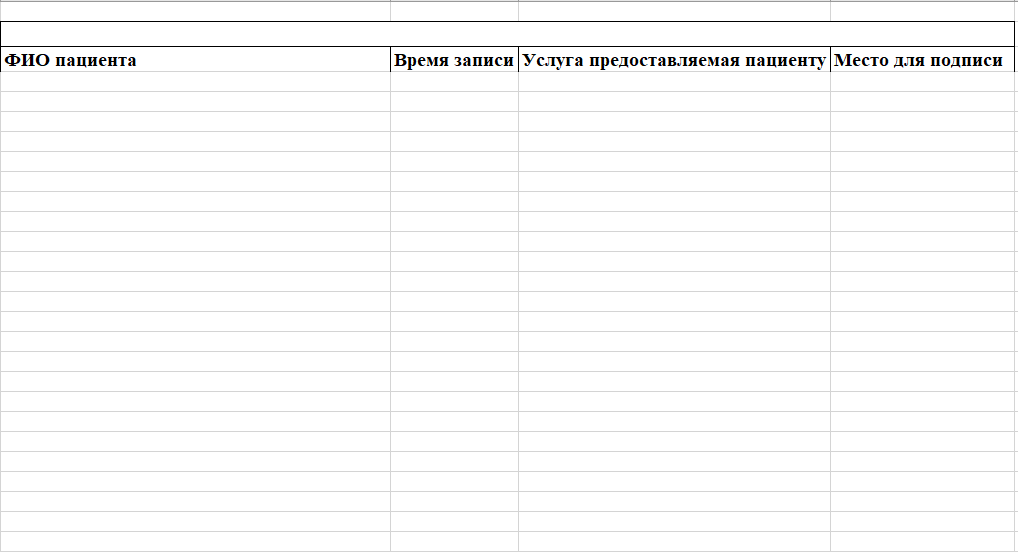


Рис.19. Шаблон списка пациентов.

# Разработка приложения пользователя.

## Разработать архитектуру приложения;

Архитектура приложения имеет такой вид. В ней присутствуют DLL библиотеки для оптимизации. Архитектура приложена ниже (Рис.20). DLL используется потому что это сохраняет ОЗУ, т.к. память на них затрачивается только когда они открываются.

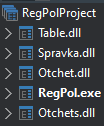


Рис.20. Архитектура приложения

Приложение имеет архитектуру Клиент-сервер, а именно толстый клиент. Суть в том, что БД лежит на сервере, а копии программы на компьютерах регистраторов и они работают с БД со своих компьютеров.

## Составить перечень всех отчетов приложения, построить макеты отчетов;

В моём курсовом есть 2 отчёта. Первый это талон для пациентов, второй это список пациентов для врача. В талоне мы указываем данные того, кто получил талон, дату выдачи, дату и время записи, врача к которому явиться.

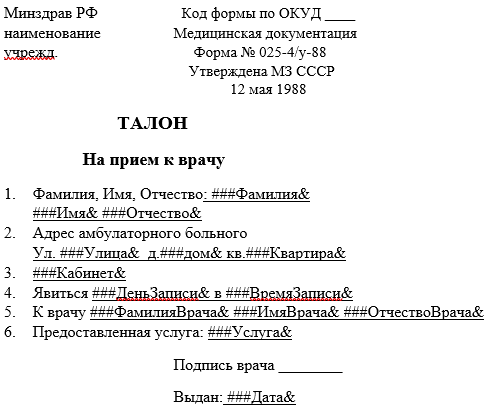


Рис.21. Шаблон талона для пациентов.



Рис.22. Список пациентов.

Список пациентов необходим для врача, чтобы он знал сколько пациентов ему принимать сегодня. Этот список печатается в случае окончания времени для записи или в день окончания записи. Врач принимает и расписывается в последнем поле. Рис. 22.

## Определить независимые вспомогательные программы, необходимые для обслуживания приложения;

* **Microsoft Office**– Пакет офисных приложений.
* **InterBase SQL Server** — Реляционная система управления базами данных, разрабатывающаяся компанией Embarcadero.

## Построить главную форму приложения;

Главная форма представляет из себя пустое окно с меню для работы

Вкладки:

Пользователь:

* Войти – Войти в свою учётную запись и подключение к БД
* Выйти – Выйти из своей учётной записи
* Регистрация – создать новую учётную запись

Справочники:

* Расписание – окно для редактирования и просмотра данных в таблице RASP
* Услуга– окно для редактирования и просмотра данных в таблице USLUGA
* Кабинет– окно для редактирования и просмотра данных в таблице CABINET
* Пациент– окно для редактирования и просмотра данных в таблице PACIENT
* Доктор– окно для редактирования и просмотра данных в таблице DOCTOR

Журнал операций

* Талон – окно для редактирования и просмотра данных в таблице TALON, а также  
  в отдельное окно для вывода отчёта на печать и файл Word.

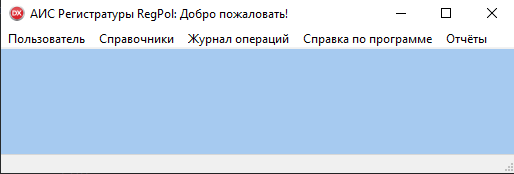


Рис.23. Главная форма

## Построить формы, обеспечивающие ввод-модификацию данных;

Как пример для формы ввода-модификации данных я буду использовать форму справочника услуг. На этой форме регистратор может смотреть какие услуги предоставляются в поликлинике. Он может добавить новые или удалить старые. Для модификации он должен нажать на кнопку «изменить», а потом менять поле названия услуги как требуется, а потом нажать кнопку «сохранить». Чтобы удалить запись любой услуги, нужно нажать кнопку «Удалить». А чтобы вставить новые данные необходимо нажать кнопку «добавить» вписать новые данные в поле названия услуги и для сохранения изменений нажать кнопку «Сохранить»

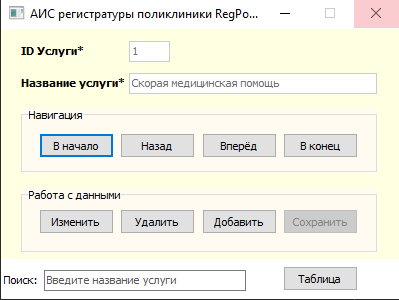


Рис.24. Справочник «Услуга»

При нажатии на кнопку «Изменить»/«Добавить» блокируются другие кнопки, но разблокируется кнопка «Сохранить», а кнопка «Изменить»/«Добавить» стала кнопкой «Отмена», также разблокируются поля с данными, кроме поля ID.

Код кнопок редактирования:

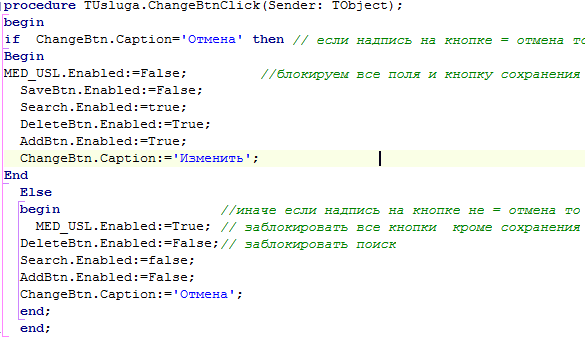


Рис. 25. Код кнопки «Изменить»

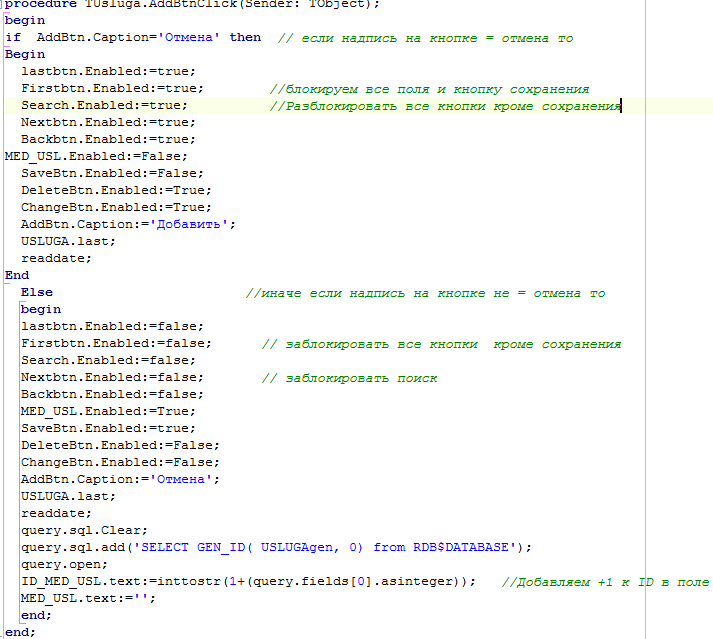


Рис. 26. Код кнопки «Добавить»

Здесь и в кнопке «Сохранить» я использовал специальную процедуру для сохранения и изменения данных.

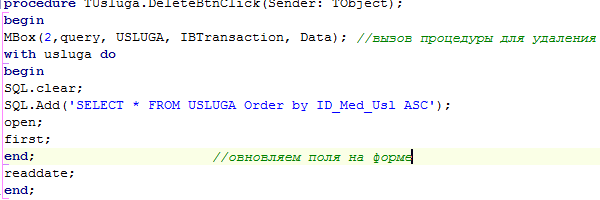


Рис. 27. Код кнопки «Удалить»

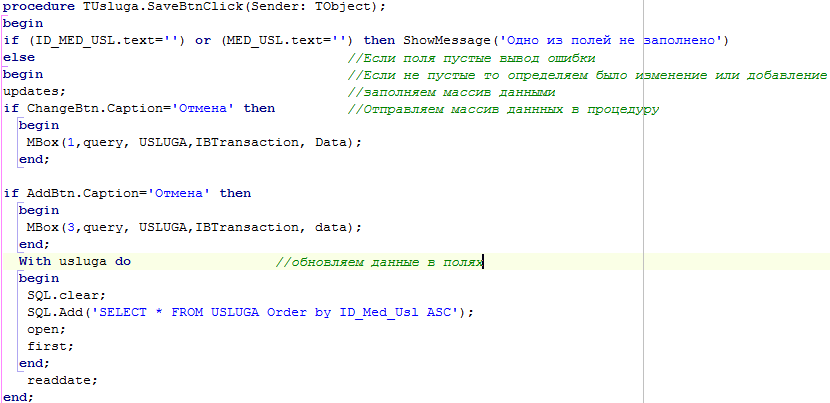


Рис. 28. Код кнопки «Сохранить»

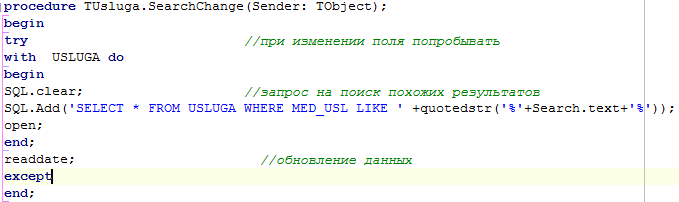
На форме есть поле поиска. Его код (Рис.29.) 

Рис. 29. Код поля поиска

Здесь мы динамически создаём запрос при помощи данных, которые мы передаём в процедуру, где она определяет, что с ними делать по номеру переданной нами переменной, где:

1. Изменить
2. Удалить
3. Добавить

На Рис. 30 представлена часть процедуры на UPDATE, остальные 2 части аналогичны.

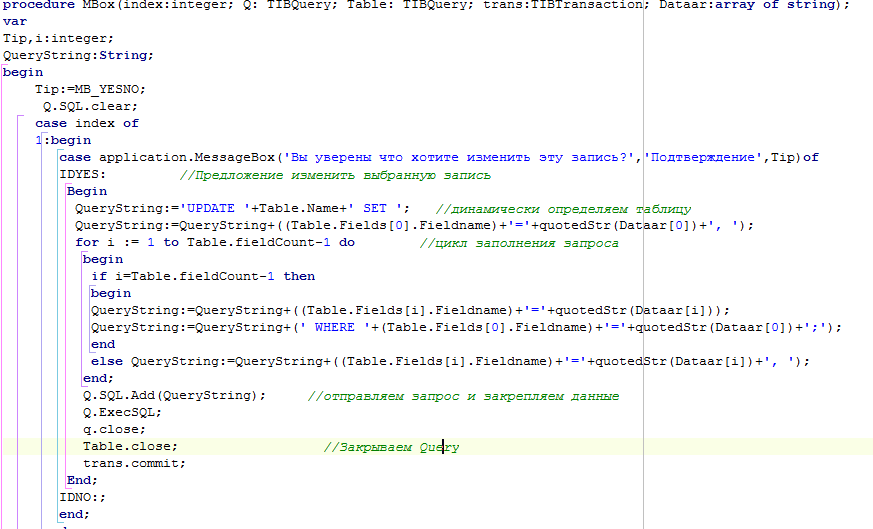


Рис. 30. Код процедуры удаления-изменения-добавить

## Создать набор отчетов для печати выходных данных;

Форма Талон (Рис.31.) и форма нового талона (Рис.32.) необходимы для просмотра уже созданных талонов и создания новых талонов соответственно. Чтобы создать новый талон регистратор должен зайти на форму «новый Талон», через форму «Талон» с помощью кнопки «Создать новый талон». Далее регистратор должен выбрать услугу, дату и время записи для пациента и написать данные пациента для проверки на наличие записи пациента в поликлиники. В случае отсутствия пациенту предлагается приписаться к нашей поликлинике. Иначе талон создаётся по нажатию кнопки «Создать талон»

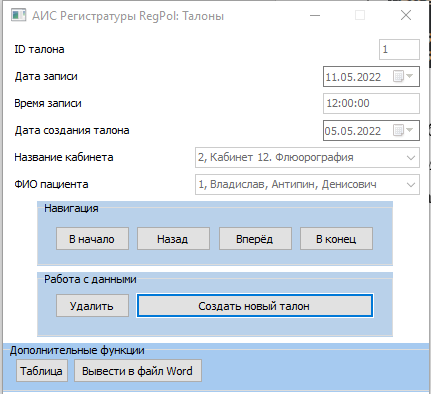


Рис.31 форма «Талон»

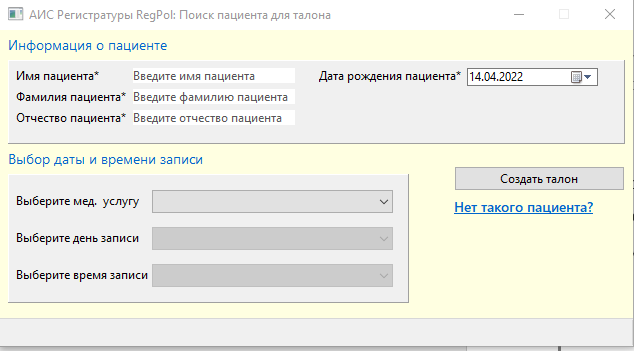


Рис.32. форма «Новый Талон»

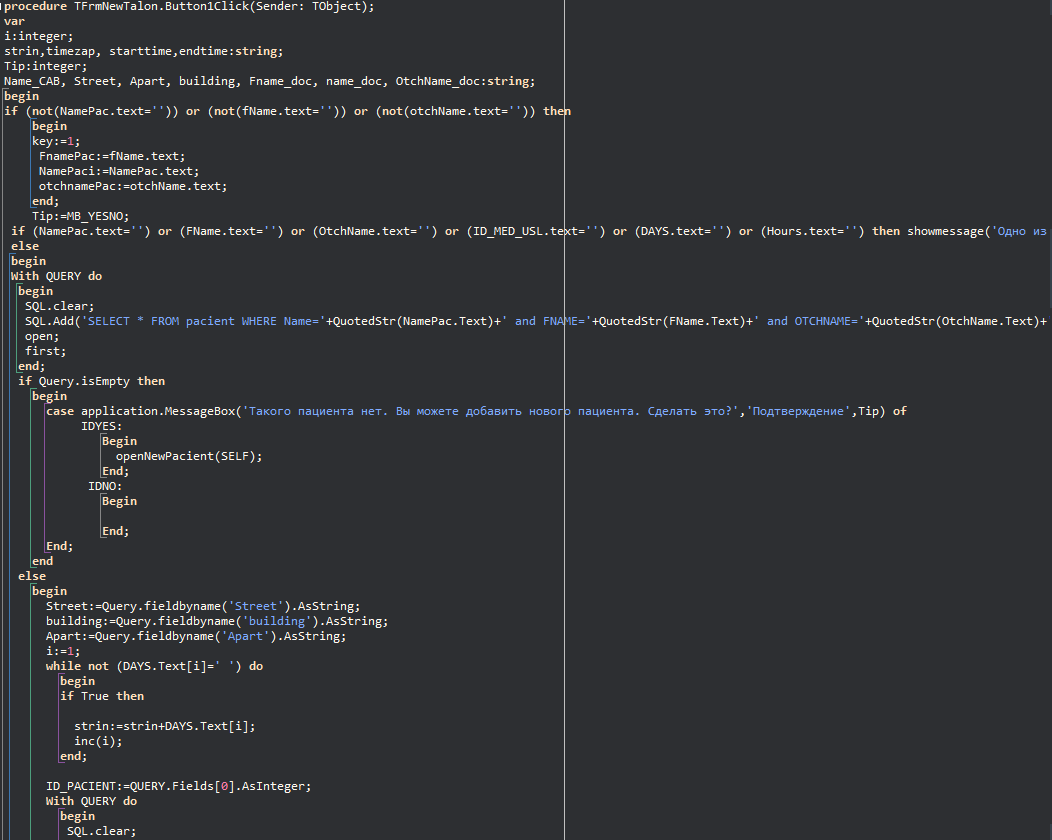


Рис.33. Код формирования талона.

На Рис. 33 отрывок кода происходит проверка на наличие пациента, в случае его отсутствия, отклоняется создание талона. Иначе мы начинаем считывать данные из заполненных полей и выпадающих списков. Далее мы при помощи запросов к БД проверяем наличие такой же записи, в случае существования таковой мы отменяем талон. Иначе программа открывает файл-шаблон и заполняет его введёнными данными. После заполнения шаблона нам предлагают сохранить или вывести сразу на печать. Или, и то, и другое. После чего мы обновляем выпадающие списки на форме чтобы невозможно было сделать новые записи на то же время что мы уже указывали.

Также создать этот же талон можно и на предыдущей форме «Талон», где нам нужно просто выбрать ту запись, которую хотим и нажать кнопку «Вывести файл в Word», далее произойдет та же процедура что и выше.

Второй отчёт Рис.34, который мы получаем для врача. Он представляет собой список записанных пациентов на сегодняшний день. На форме необходимо выбрать дату и услугу, после чего мы увидим в таблице список всех пациентов, пришедших в этот день на эту услугу. Чтобы вывести отчёт нужно нажать кнопку «Вывести список в Excel»

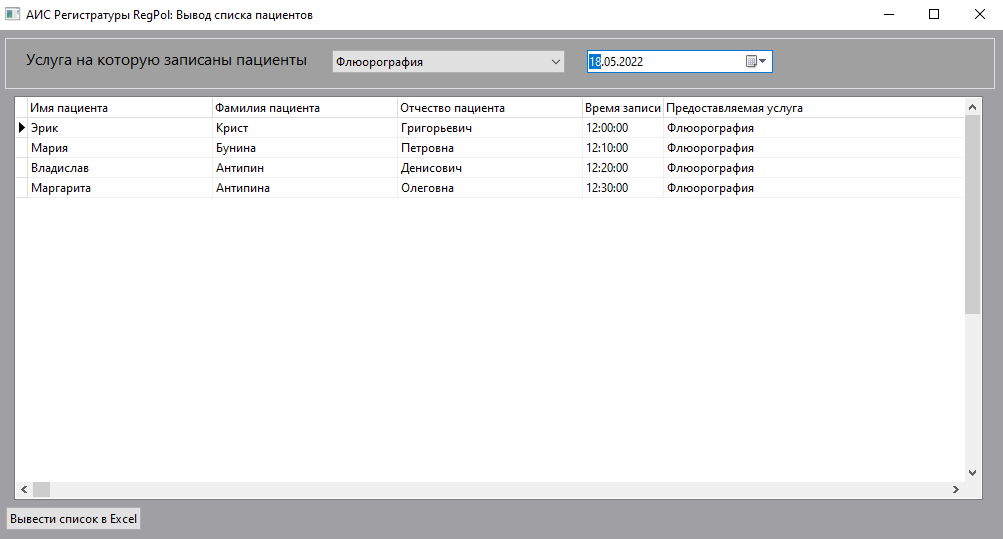


Рис.34. Форма отчёта-списка.

На Рис.35 мы открываем шаблон и через 2 цикла делаем обводку для таблицы и записываем туда данные, после чего на предлагают распечатать то, что получилось. И мы закрываем файл.

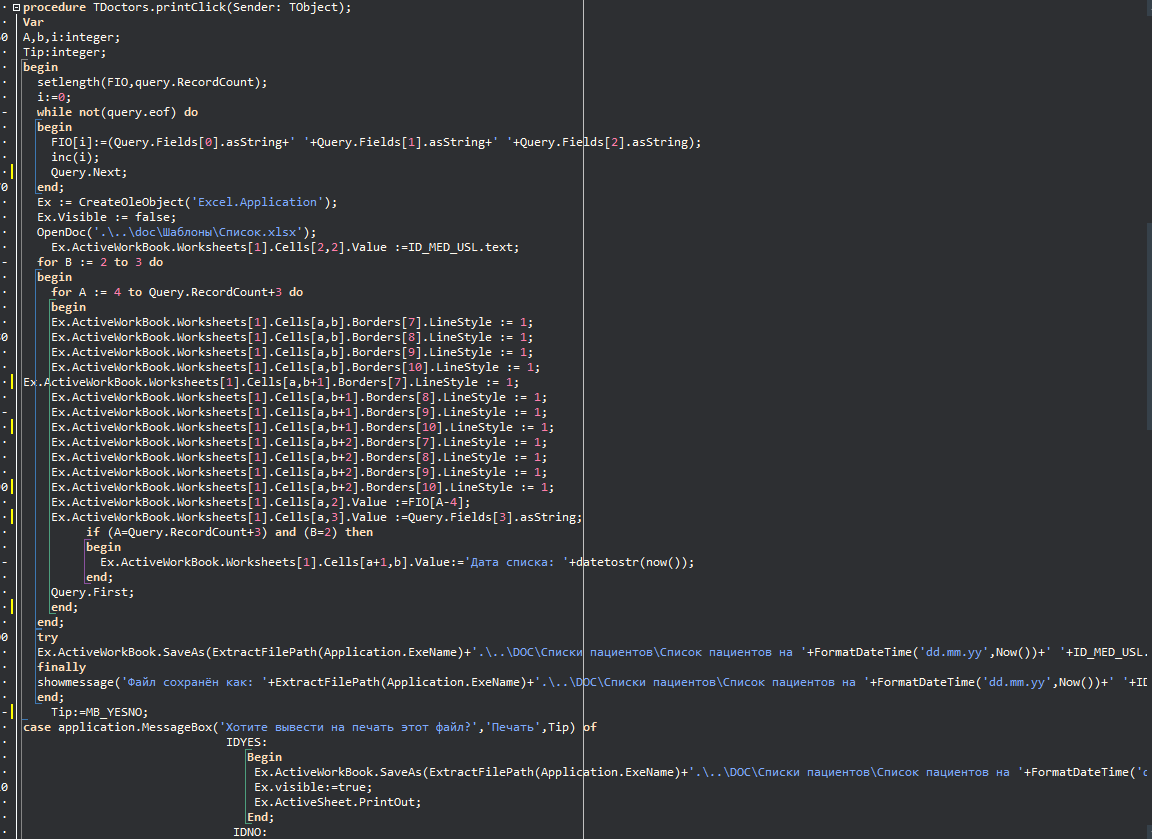


Рис.35. Код формирования списка.

## Создать инструкции по использованию приложения и справочную систему;

Для того чтобы пользователь знал хоть какие-то основы для работы с программой необходимо написать справку с подробным описанием действий и к чему эти действия приводят.  
Справка написана в программе Win CHM и сохранена в справочном формате CHM в папке “Help” с проектом. Внешний вид справки на Рис.36.

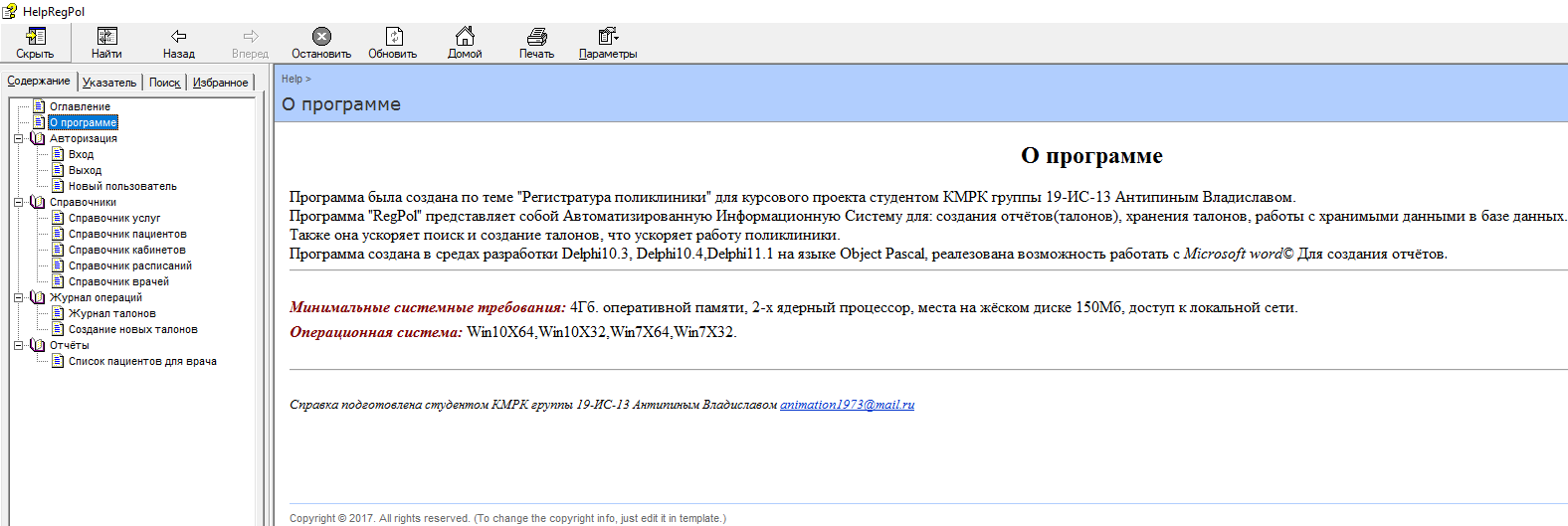


Рис.36. Справка.

# Заключение

В итоге моих работ я изучил предметную область регистратуры поликлиники и создал с нуля приложение «АИС RegPol» для регистратуры поликлиники для ускорения работы регистратуры.

Для того чтобы приложение функционировало я создал БД, с которой работает приложение, изменяя данные в нём. Приложение просто в управлении и не имеет сложной структуры, в следствии и интерфейс дружелюбен к новому пользователю, но, в случае если что-то не ясно всегда можно заглянуть в справку, в которой всё описано.

Приложение соответствует всем требованиям, оно работает, имеет защиту, архитектуру, подключение к БД, пояснительную справку, выводит отчёты для врачей и пациентов, имеет всю документацию, а значит оно готово к эксплуатации.

# Список литературы

В. И. Грекул, Проектирование информационных систем Проектирование информационных систем 2-е изд. исп. — М.: Интернет-Университет

Информационных Технологий «Интуит», 2016.  
Вендров А.М. Проектирование программного обеспечения экономических информационных систем – М: Финансы и статистика, 2002.

Сайт Управление здравоохранением вкладки главная>>Статьи>>региональный опыт>> Организация работы регистратуры мед. Организаций, оказывающих первичную медико-санитарную помощь взрослому населению. 2017.

# Приложение А

1. Введение

В наше время необходима автоматизация сложных процессов, которые проходят на любых предприятиях, чтобы повысить производительность этих предприятий. Поэтому часто необходимо создавать АИС, которые будут нажатием пары кнопок будут решать нам сотни операций и выдавать необходимый нам результат за секунду.

1.1. Наименование программы

Наименование программы: АИС "RegPol"

* 1. Назначение и область применения

Программа должна создавать и печатать направления для пациентов, также создавать и выводить списки пациентов для врачей. Также иметь возможность работать с данными в БД.

1. Требования к программе

К программе предъявлены следующие требования:

• быть высоко производительной

• быть отказоустойчивой

• иметь дружелюбный интерфейс для оператора

• не быть ресурсоемкой

• быть совместимой с программами и ОС

• иметь меры защиту важных элементов

* 1. Требования к функциональным характеристикам Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

– обработка данных пациентов;

– обработка данных врачей;

– обработка данных расписания;

– обработка данных о услугах предоставляемых поликлиниках;

– обработка данных о кабинетах;

– вывода списков записанных пациентов на определённый день для врачей и его печать;

– вывода талонов для пациентов и его печать.

3. Условия эксплуатации

3.1. Требования к квалификации и численности персонала

Минимум персонала, требуемого для работы программы, обязан составлять не менее 2 штатных единиц:

* системный администратор
* Регистратор (оператор)

3.2. Требования к составу и параметрам технических средств

3.2.1. В состав технических средств должен входить IВМ-совместимый персональный компьютер (ПЭВМ), выполняющий роль сервера, включающий в себя:

3.2.1.1. процессор Pentium-2.2GHz, не менее;

3.2.1.2. оперативную память объемом, 4 Гбайт, не менее;

3.2.1.3. свободного пространства на жестком диске, 250 Гигабайт, не менее;

3.2.1.4. операционную систему Windows 2016 Server или Windows 10 (2021Н); 3.2.1.5. Видеокарта с объемом видеопамяти 32 Мб или процессор с видеочипом, не менее.

3.3. Требования к информационной и программной совместимости

3.3.1. Требования к информационным структурам и методам решения

База данных работает под управлением InterBase2020. Используется много поточный доступ к базе данных.

3.3.2. Требования к исходным кодам и языкам программирования

Использовался язык Object Pascal в среде разработки Delphi версии 10.4, 10.3, 11.1.

3.3.3. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 2016 Server или Windows 10 (2021Н) и Microsoft SQL Server 2020

* + 1. Требования к защите информации и программ

Необходим пароль при входе в программу, который будет предотвращать несанкционированный доступ к БД поликлиники.

4. Требования к программной документации

4.1. Предварительный состав программной документации

Состав программной документации должен включать в себя:

1. техническое задание;

2. руководство оператора;

3. Пояснительная записка;

5. Стадии и этапы разработки

5.1. Стадии разработки

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

5.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания. На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы. На стадии внедрения должен быть выполнен этап разработки подготовка и передача программы

5.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4.определение стадий, этапов и сроков разработки программы и документации на неё;

5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы. На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение и методики испытаний;

2. проведение приемо-сдаточных испытаний;

3. корректировка программы и программной документации по результатам испытаний.

6. Порядок контроля и приемки

6.1. Виды испытаний

Приемо-сдаточные испытания должны проводиться на объекте Заказчика 24.05.2022г.

1) Руководство пользователя по ГОСТ 34.201-89 и РД 50-34.698-90;

2) Пояснительная записка к техническому проекту по ГОСТ 34.201-89;

3) Руководство системного программиста по ГОСТ 19.503-79;

4) Руководство программиста по ГОСТ 19.504-79;

5) Методика испытаний по ГОСТ 34.603-92.

# Приложение Б

Таб. №5 перечень оборудования

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Цена за шт.** | **Кол-во** | **Цена за всё** |
| Компьютеры-станции | 30000руб. | 6 | 180000руб. |
| Сервер | 37000руб. | 1 | 37000руб. |
| Маршрутизатор | 5289Руб. | 1 | 5289Руб. |
| Коммутатор | 1342Руб. | 1 | 1342Руб. |
| Провода RJ-45 | 100руб. | 50м | 5000руб. |
| Windows 10 корпоративная | 10000руб. | 6 | 60000руб. |
| Windows Server 2017 | 100000руб. | 1 | 100000руб. |
| Монтажные работы | 5000руб. | 1 раз | 5000руб. |
| Программист БД | 140,00 ₽/ч | 224ч | 31360 руб. |
| Тестировщик | 70,00 ₽/ч | 40ч | 2800 руб. |
| Кодировщик | 50,00 ₽/ч | 200ч | 10000 руб. |
| Аналитик | 309,00 ₽/ч | 64ч | 19776 руб. |
| Лицензии на программы разработки | 0,00 ₽ | 3 | 170 999руб. |

Итого:628566руб.

# Приложение В (ER – диаграмма физической модели данных)

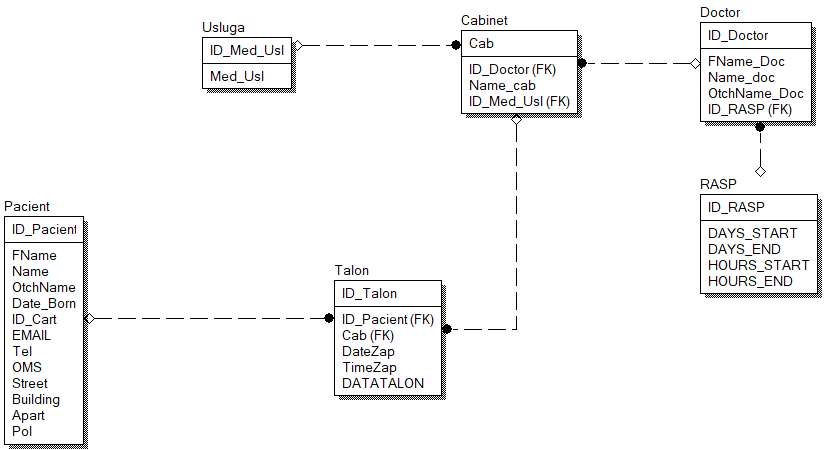


Рис.37. ER-диаграмма.

# Приложение Г

На Рис.38. изображена форма «Входа» мы должны ввести логин и пароль, которые уже есть в БД. Если их там нет, то мы не зайдем. Для теста есть кнопка Ad чтобы не нужно было вспоминать пароль и логин системного администратора.

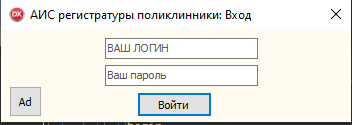


Рис.38. Форма входа.

Форма регистрации (Рис.39). Форма, на которой только системный администратор и главный регистратор может создать нового пользователя

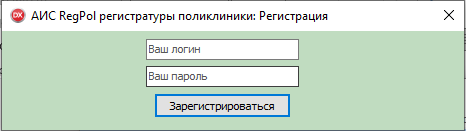


Рис.39. Форма Регистрации.

На Рис.40 изображена форма справочника расписания, на нём, как и на других справочниках, есть возможность изменять данные, просматривать их и делать поиск по данным расписания.

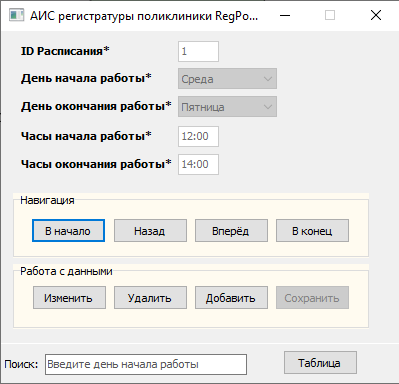


Рис.40. Форма справочника расписания.

Рис.41. изображает форму справочника пациентов, где мы можем добавить новых пациентов.

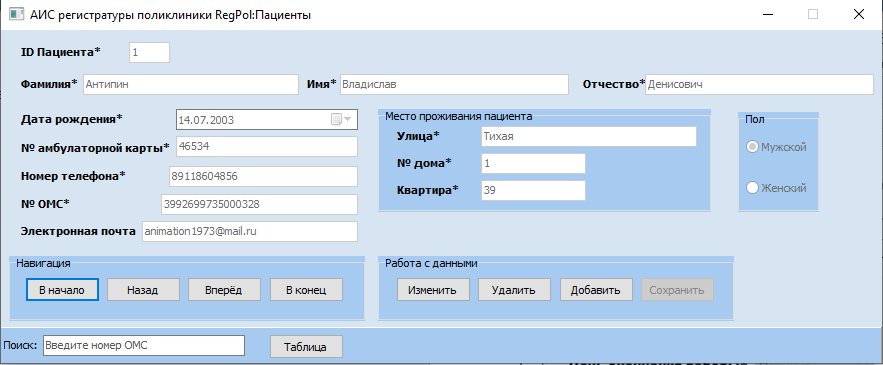


Рис.41. Форма справочника пациентов.

На Рис.42 изображён справочник услуг, где мы можем работать с таблицей Услуг

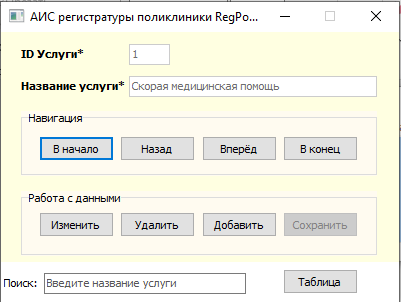


Рис.42. Форма справочника услуг.

Рис.43 изображен справочник выполняющий роль инструмента для работы над таблицей кабинета.

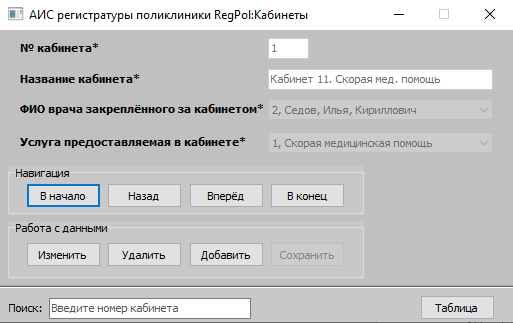


Рис.43. Форма справочника кабинетов.

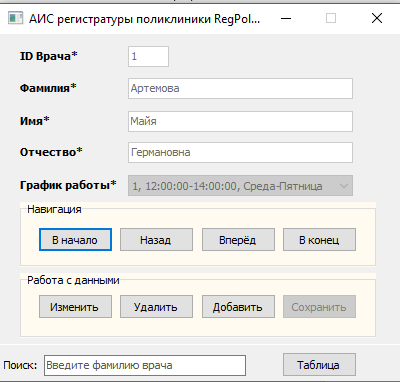


Рис.44. Форма справочника врачей.

На Рис.45 форма имеет сведенья о авторах программы.

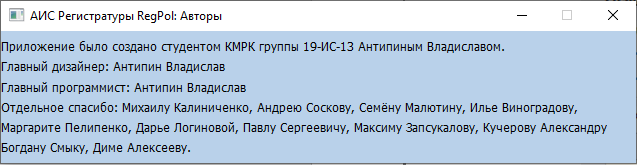


Рис.45. Форма «Авторы».

На Рис.46 форма для справки имеющая возможность открыть справку с руководством пользователя.

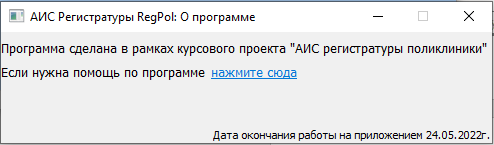


Рис.46. Форма «О программе».

Справочная форма просмотра талонов на Рис. 47

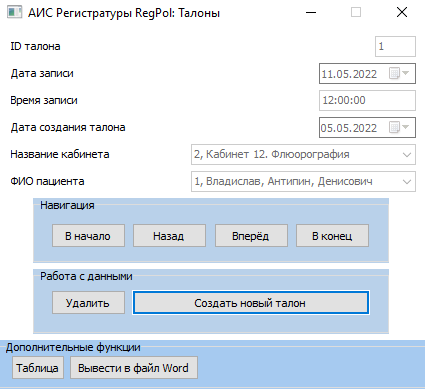


Рис.47. Форма «Талон».

Форма на Рис.48. необходима для формирования новых талонов.

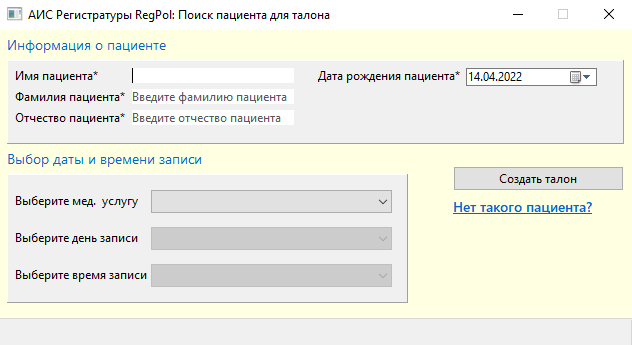


Рис.48. Форма «Новый талон».

Рис.49 показывает форму для просмотра записанных пациентов на какой-либо день и формирования новых списков записанных пациентов на какую-либо дату.

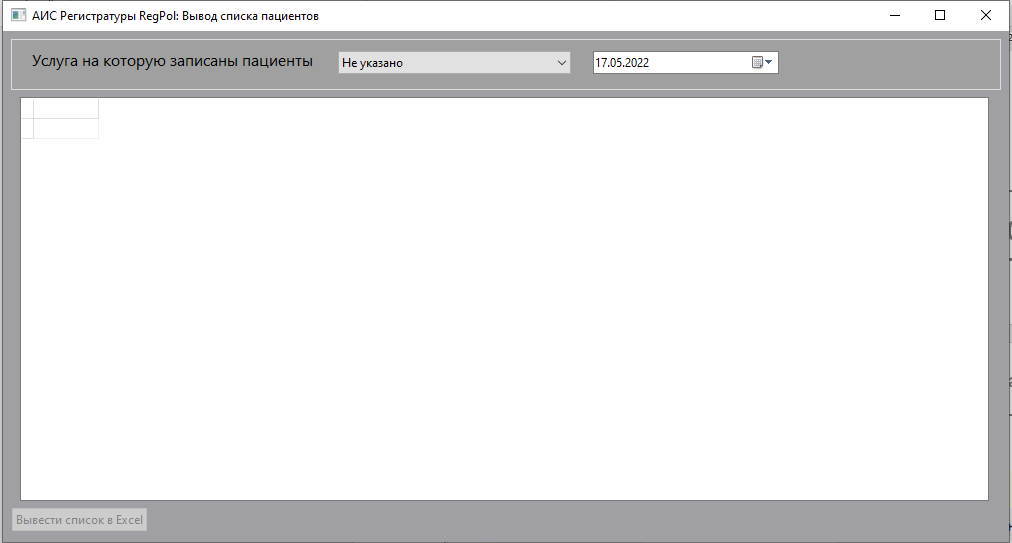


Рис.49. Форма «Вывод списка пациентов».

На Рис.50 видна таблица всех талонов, созданная динамически, с помощью лейблов.

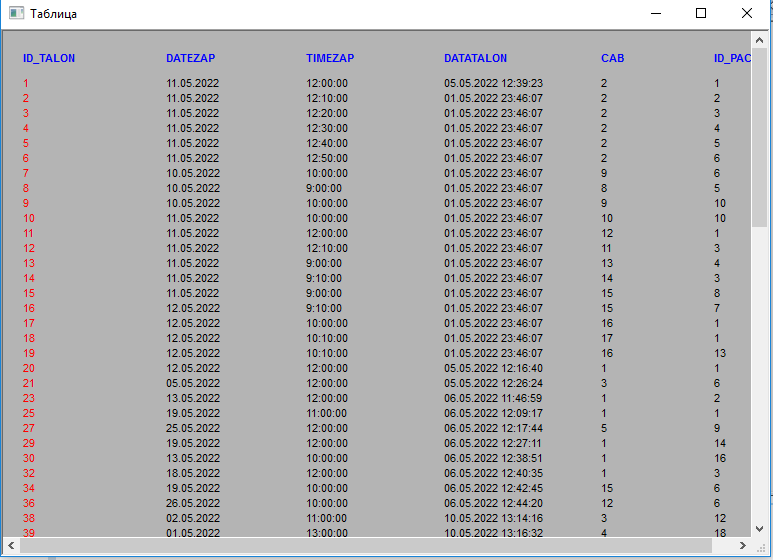


Рис.50. Форма «Таблица талона».