

## ОГЛАВЛЕНИЕ

Введение.....	3
1 Теоретическая часть квалификационной работы .....	5
1.1 О медицинских информационных системах .....	5
1.2 Обзор существующих медицинских информационных систем...	5
1.3 Предпосылки разработки модуля «Статистика» .....	8
1.4 Обзор средств разработки .....	10
1.4.1 Языки программирования для разработки.....	10
1.4.2 Библиотеки элементов интерфейса .....	13
1.4.3 Система управления базами данных .....	15
1.5 Выбор средств разработки .....	16
1.6 Структура базы данных МИС САМСОН .....	17
2 Практическая часть квалификационной работы .....	35
2.1 Техническое задание.....	35
2.2 Структура модуля .....	35
2.3 Интерфейс и функциональные возможности.....	37
2.4 Расчеты экономической эффективности использования модуля	
47	
Заключение .....	52
Список литературы .....	53
Приложение А .....	55
Приложение Б.....	60

## ВВЕДЕНИЕ

Современные медицинские организации накапливают огромные объемы данных. От того, насколько эффективно используется эта информация, зависит качество медицинской помощи. Подавляющее число медицинских учреждений разных форм собственности, занимающихся лечением пациентов, внедряют информационные медицинские системы (далее МИС) в соответствии с приказом Министерства здравоохранения Российской Федерации от 28 апреля 2011 года № 364 «Об утверждении концепции создания единой государственной информационной системы в сфере здравоохранения»

Основная задача МИС – это выполнение функции сбора, хранения, обработки, выдачи информации. Любая информационная система должна обеспечить однократный ввод и многократное использование первичной информации. Повсеместное внедрение медицинских информационных систем в практику организаций здравоохранения Российской Федерации обеспечивает высокую актуальность данной работы.

Таким образом, объектом исследования работы является применение медицинских информационных систем в здравоохранении.

Предмет исследования – медицинская информационная система «Самсон», используемая в ФГБУ «РНЦ «ВТО» им. акад Г.А. Илизарова».

Цель дипломного проекта: разработка модуля «Статистика» комплекса САМСОН для последующего использования в РНЦ ВТО им. Илизарова.

При разработке модуля «Статистика» были поставлены следующие задачи:

- проанализировать предметную область;
- провести анализ требований и определить необходимую функциональность разрабатываемого модуля;
- определить программные средства разработки;
- спроектировать и разработать модуль.

Для решения поставленных задач были изучены наиболее распространенные медицинские информационные системы, проанализированы возможности этих систем для вывода статистической информации. Выявлены проблемы и недочеты информационной системы Самсон в части извлечения статистических данных и невозможность выбора необходимых полей для отображения информации о группах пациентов.

# **1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

## **1.1 О медицинских информационных системах**

Медицинская информационная система (МИС) – система автоматизации документооборота для лечебно-профилактических учреждений, в которой объединены система поддержки принятия медицинских решений, электронные медицинские карты о пациентах, данные медицинских исследований в цифровой форме, данные мониторинга состояния пациента с медицинских приборов, средства общения между сотрудниками, финансовая и административная информация. [3]

Медицинская информационная система должна обеспечивать так же вывод информации о группах пациентов, эта информация необходима для анализа работы отделений медицинского Центра и общей работы лечебно-профилактического учреждения (ЛПУ).

Внедрение МИС так же скрывает в себе проблемы, связанные со стрессом для персонала, не привыкшему к такой системе, и как следствие возможный саботаж. Такие решения приводят к замедлению работы при обслуживании пациентов.

Однако с внедрением системы может возрасти уровень сервиса, в том числе доступный объем справочной информации для пациента.

## **1.2 Обзор существующих медицинских информационных систем**

Единая государственная информационная система в сфере здравоохранения министерства здравоохранения российской федерации. Подсистема мониторинга реализации государственного задания по оказанию высокотехнологичной медицинской помощи.

Областью применения информационно-аналитической системы Министерства здравоохранения Российской Федерации являются процессы сбора, обработки и анализа данных в сфере здравоохранения.

Подсистема мониторинга реализации государственного задания по оказанию высокотехнологичной медицинской помощи предназначена для автоматизации процессов сбора, обработки и анализа данных об оказании ВМП гражданам Российской Федерации.

Так как цель данной дипломной работы «Разработка модуля «Статистика»», то обзор будет осуществлен по возможностям других систем вывода статистики.

Система представляет следующие способы для вывода статистической информации:

- выбор колонок для отображения (Рисунок 1);

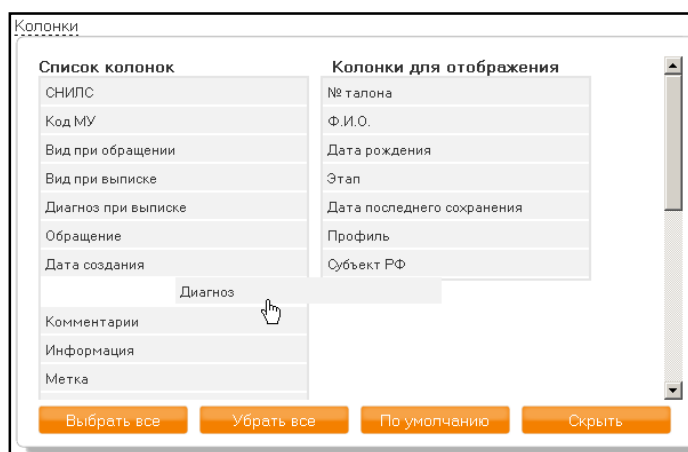


Рисунок 1 – Выбор колонок для отображения

- возможность расширенного поиска представляющая поиск по полям, отображенным на рисунке (Рисунок 2).

Профиль: Все профили | Группа: Не выбрано | Возрастные группы: Все

Субъект РФ: Российская Федерация | Пол: Все | Год: 2016

Фамилия | Имя | Отчество | СНИЛС | Номер страхового полиса

☐ Дата планируемой госпитализации с 07.06.2016 по 14.06.2016 ☐ В листе ожидания

☐ Только талоны с меткой

☐ Дата обращения в МО 07.06.2016 по 14.06.2016 Реабилитация: Не указано

☐ Дата выписки с 07.06.2016 по 14.06.2016 ☐ Дата операции с 07.06.2016 по 14.06.2016

☐ Дата создания талона с 07.06.2016 по 14.06.2016 ☐ Дата принятия решения с 14.06.2016 по 14.06.2016

Код категории льготы: Не выбрано | Социальная группа: Все

☐ Дата последнего изменения с 07.06.2016 по 14.06.2016 ☐ Дата рождения с 09.06.2016 по 14.06.2016

☐ Талоны, просроченные более чем на 10 дней

Вид при обращении: Не выбрано | Вид при выписке: Не выбрано

Результат госпитализации: Житель: Обращение за ВМП Ответственное лицо Госпитализирован в отделение

Все | Все | Все | |

**Скрыть**

Рисунок 2 – Расширенный поиск

Информационная система «Медведь». Российская платформа «Медведь» автоматизирует работу органов управления здравоохранением регионального масштаба и предоставляет электронные услуги населению.

На сайте разработчика медицинской информационной системы «Медведь» указаны следующие достоинства:

- «Сбор и хранение персональных данных;
- Статистическая отчетность по утвержденным государственным формам отчетности;
- Нерегламентированная аналитическая отчетность на основе произвольных запросов клиента;
- Представление ключевых параметров системы здравоохранения на рабочем месте руководителя для наглядного визуального анализа.» [4]

Медицинская информационная система «Медведь» представляет возможность вывода статистической информации в виде заранее подготовленных отчетных форм. Такой подход не всегда удобен, так как

исключает возможность получения определенной информации из медицинской системы. Система сбора отчетности МИС «Медведь» может быть внедрена одним из двух способов. Способы отличаются наличием в лечебно-профилактическом учреждении собственной системы. Диаграмма внедрения представлена на рисунке (Рисунок 3).



Рисунок 3 – «Медведь» внедрение

### 1.3 Предпосылки разработки модуля «Статистика»

Приложение к программе «Самсон» позволит решить многочисленные проблемы, связанные с учетом и анализом состава больных: пролеченных в стационаре, больных, находящихся на лечении в стационаре, а также амбулаторных пациентов. Рассмотрим перечисленные проблемы подробнее:

#### 1. Ведение базы данных «Нозологический и функциональный каталог».

С 1970 года в архиве Центра Илизарова существует информационно-поисковая система «Нозологический и функциональный каталог». До 1998 года она существовала в виде картотеки, затем была создана электронная версия. В нее вносятся сведения из истории болезни по каждому этапу лечения больного. (Ф.И.О. больного, возраст, пол, регион проживания,

отделение, дата выписки, заключительный клинический диагноз, код по международной классификации болезней 10-го пересмотра (МКБ-10), название операции, архивный номер). Далее каталог дополняется информационными систематическими сведениями: нозологическая форма заболевания, сегмент, локализация и т.д.

Нозологический каталог используется в целях подбора пациентов по заданным параметрам для научно-исследовательской работы ФГБУ РНЦ «ВТО» им. акад. Г.А. Илизарова: подготовки докторских и кандидатских диссертаций, монографий, научных статей. Используется он и в клинической работе: для подготовки аналитических отчетов о работе стационара.

Информация из историй болезни пролеченных пациентов переносится в ИПС в ручном режиме. В связи с тем, что объем информации достаточно большой, это занимает значительное количество времени, не исключает возможные ошибки и неточности.

## 2. Заполнение «Журнала учета стационарных больных».

Журнал учета больных, поступивших на стационарное лечение в профильное отделение, в настоящее время заполняется в ручном режиме старшей медицинской сестрой отделения. Вся информация, необходимая для ведения журнала вносится в МИС «Самсон» на этапе оформления истории болезни.

3. Группировка данных по произвольно заданным параметрам для составления оперативных отчетов о работе клинических отделений и стационара в целом, проведения анализа работы учреждения.

В течение предыдущих лет для полноценного анализа работы клиники заполнялась база данных «Медицинская статистика», которая содержала информацию из «Статистической карты выбывшего больного» по тридцати одному пункту. С активным внедрением МИС «Самсон» дублирование информации оказалось излишне затратным по времени. Однако обнаружилось, что «Самсон» формирует только заранее созданные отчеты без возможности редактирования схемы запроса. Извлечь необходимую



информацию оказалось невозможным. В качестве примера, центр Илизарова, в отличие от медицинских учреждений муниципального здравоохранения, принимает на лечение пациентов из большинства субъектов Российской Федерации и иностранных больных. Поэтому, довольно часто необходимо получать информацию о пролеченных больных в разрезе регионов с указанием прочих статистических данных, например, источника финансирования лечения, продолжительности лечения, половозрастного состава и т.д.

## **1.4 Обзор средств разработки**

### **1.4.1 Языки программирования для разработки**

Выбор языка разработки не такая уж и простая задача, так как язык программирования должен отвечать требованиям производительности, безопасности, скорости разработки на языке, распространенности, документированности и ещё большому количеству требований разработчика.

В связи с этим для разработки было отобрано несколько языков: C#, Ruby, Java, Javascript.

C# — ОО (Объектно-ориентированный) язык программирования разработанный в компании Microsoft. C# имеет статическую, строгую типизацию. К плюсам данного языка программирования можно отнести:

- встроенную поддержку .NET — графического фреймворка позволяющего создавать кроссплатформенные приложения (не зависящие от операционной системы);
- поддержку событийно-ретинированного программирования;
- унифицированную систему типизации.

К недостаткам можно отнести слабую поддержку ОС отличных от Windows, довольно сложный синтаксис, относительно невысокую производительность.

Java — объектно-ориентированный язык с статической типизацией и кроссплатформенной реализацией. Код на Java транслируется в специальной байт-код и выполняется виртуальной машиной, поэтому язык является интерпретируемым и может выполняться на любой платформе. К преимуществам языка можно отнести автоматического управление памятью, богатый набор средств фильтрации исключительных ситуаций, наличие классов позволяющий выполнять различные HTTP запросы и обрабатывать ответы, встроенную поддержку многопоточности. Так же Java позволяет разрабатывать приложения не только для компьютеров, но и для мобильных устройств. Благодаря этому вокруг языка сформировано большое сообщество разработчиков, которые помогают делать этот язык лучше. С GUI фреймворками как и у всех кроссплатформенных языков существуют определенные проблемы, тут и распространенный Qt, и wxWidgets, и используемый реже AWT на основе которого реализовано множество других фреймворков графического интерфейса для Java.

К недостаткам Java можно отнести не очень быструю скорость выполнения программы из-за отказа от арифметики указателей, неявных преобразований типов с потерей точности. Так же в Java полностью отсутствует возможность управления памятью, объект будет удален только тогда, когда на него не будет существовать указателей и, интерпретатор запустит очередную операцию освобождения памяти.

Ruby — интерпретируемый язык программирования высокого уровня. Ruby, это тщательно сбалансированный язык. Его создатель Юкури Мацумото, объединил части своих любимых языков, для того чтобы сформировать новый язык, в котором парадигмы функционального и императивного программирования будут сбалансированы. Ruby получил свою популярность благодаря фреймворку Ruby on Rails который реализует среду разработки WEB-приложений. Преимущества Ruby:

- кроссплатформенный;

- может внедряться в HTML разметку (как PHP);
- реализует «чистую» концепцию объектно-ориентированного программирования;
- независимая от операционной системы реализация многопоточности;
- итераторы;
- расширяется за счет библиотек, написанных на C или Ruby;
- встроенный отладчик;
- огромное сообщество разработчиков;
- централизованная система управления пакетами для языка.

Недостатки:

- низкая производительность;
- слабая русскоязычное описание нюансов.

JavaScript – прототипно-ориентированный сценарный язык программирования. Контролируется нотацией ECMAScript. JavaScript обычно используется для выполнения сценариев внутри браузера, однако только этим использование языка не ограничивается. Так, например, node.js разрешает использовать JavaScript на сервере, позволяя обрабатывать входящие запросы пользователя и работать с базой данных. Также появился фреймворк для построения десктопных графических приложений на JavaScript который называется Electron. Достоинства языка:

- массовость;
- динамическая типизация;
- поддерживается огромным количеством устройств, телефоны, планшеты, компьютеры, всем что имеет браузер;
- применение во многих областях.

Недостатки:

- необходимость обеспечения кроссбраузерности;

- система наследования вызывает трудности после перехода с ООП языка.

### **1.4.2 Библиотеки элементов интерфейса**

При разработке приложений сталкиваются с необходимостью отображения графического интерфейса для пользователя. Так как проблему, связанную с отображением пользовательского интерфейса, приходится решать часто и это занимает много времени, то для того что бы ускорить процесс разработки в большинстве проектов используются библиотеки элементов интерфейса.

В каждой системе, имеющей графический интерфейс, присутствует набор стандартных простых элементов интерфейса с возможностью работы с ними. Такие простые элементы называются «виджетами», они могут быть вложены друг в друга что позволяет строить сложные пользовательские интерфейсы. Библиотеки элементов интерфейса осуществляют «высокоуровневый» доступ к элементам интерфейса операционной системы, позволяя не заботиться о размещении и компоновке элементов друг с другом. Также большинство библиотек имеют независимую от операционной системы реализацию, что позволяет программисту разрабатывать приложение не под определённую операционную систему, а под все распространенные сразу.

Нам необходимо чтобы приложение работало на любой операционной системе, поэтому будут рассматриваться только кроссплатформенные библиотеки графического интерфейса.

Qt – свободный, кроссплатформенный инструмент для построения графического интерфейса на C++, также есть «привязки» к другим языкам программирования таким как: Python, Ruby, Java, PHP и другим.

Отличительной особенностью Qt является использование системы сигналов и слотов, которые реализуют шаблон проектирования, называемый

«наблюдателем». Концепция заключается в том, что компонент может посылать сигнал о событии (выделение текста, нажатия кнопки и др.), а другие компоненты могут принимать эти сигналы с помощью специальных функций называемых слотами.

Ещё один плюс использования Qt является наличие качественной документации. Статьи документации имеют множество качественных примеров, благодаря которым значительно упрощается понимание материала. Исходный код библиотеки Qt подробно комментирован, хорошо отформатирован, и легко читается, что также упрощает изучение данного фреймворка.

Другая библиотека графического интерфейса это FLTK (Fast, Light Toolkit). Она также является кроссплатформенной и имеет свободную лицензию. Изначально библиотека разрабатывалась для поддержки 3D графики, поэтому имеет встроенный интерфейс к OpenGL, но хорошо подходит и для построения обычного пользовательского интерфейса. Библиотека использует независимые от операционной системы модель отображения, поэтому приложения выглядят в операционной системе неестественно. Также к недостаткам можно отнести малое число виджетов по сравнению с другими графическими библиотеками. Данный фреймворк плохо документирован из-за чего сложен в изучении.

GTK+ — популярный кроссплатформенный фреймворк, имеет простой в использовании API. GTK+ написан на Си, но является объектно-ориентированным. Причиной выбора языка Си было желание строить интерфейсы для других языков программирования. Благодаря этому под GTK+ можно писать на большом количестве языков программирования (Erlang, Haskell, Perl, PHP, Lua, Lisp, Ruby и др.). При использовании GTK+ графический интерфейс может конфигурироваться пользователем или программистом, при этом осуществляется настройка не только цветов и шрифтов, но и способы отображения различных элементов. К недостаткам данной графической библиотеки можно отнести слабую документацию и

скудное количество, и качество примеров. В русском сегменте программистов про GTK мало кто знает, поэтому на русском документации почти нет.

### **1.4.3 Система управления базами данных**

База данных (БД) – набор предназначенных для совместного использования логически связанных данных. База данных определяется однократно, а затем используется одновременно несколькими пользователями.

Система управления базами данных (СУБД) – это совокупность программных и лингвистических средств благодаря которым пользователи могут осуществлять управление созданием и использованием баз данных.

СУБД по способу доступа разделяются на:

- файл-серверные;
- клиент-серверные.

«Клиент-серверная СУБД позволяет обмениваться клиенту и серверу минимально необходимыми объёмами информации. При этом основная вычислительная нагрузка ложится на сервер. Клиент может выполнять функции предварительной обработки перед передачей информации серверу, но в основном его функции заключаются в организации доступа пользователя к серверу.» [1]

В большинстве случаев клиент-серверная СУБД гораздо менее требовательна к пропускной способности компьютерной сети, чем файл-серверная СУБД, особенно при выполнении операции поиска в базе данных по заданным пользователем параметрам, т.к. для поиска нет необходимости получать на клиент весь массив данных: клиент передаёт параметры запроса серверу, а сервер производит поиск по полученному запросу в локальной базе данных. Результат выполнения запроса, который обычно на несколько

порядков меньше по объёму, чем весь массив данных, возвращается клиенту, который обеспечивает отображение результата пользователю.

## 1.5 Выбор средств разработки

Для разработки приложения «Статистика» был выбран скриптовый язык программирования Ruby. Ruby является открытым программным обеспечением и распространяется свободно под лицензией GNU General Public License. Этот язык программирования имеет большое сообщество разработчиков и поэтому в нем имеется множество готовых решений, благодаря которым процесс разработки значительно увеличивается.

Для построения графического интерфейса был выбран фреймворк Qt. Qt отлично показывает себя на распространенных платформах таких как Windows, OS X и графической среде Gnome.

Для доступа к базам данных используется библиотека ActiveRecord. Данная библиотека реализует шаблон проектирования Active Record. Принцип работы AR: пусть в базе данных существует некоторая таблица. Для этой таблицы создается специальный класс AR, который является представлением таблицы как класса в программе. Таким образом, каждый экземпляр данного класса соответствует одной записи таблицы, при чтении полей объектом считывается значение записи таблицы базы данных. Благодаря реализации Active Record в Ruby объявление двух классов и связей между ними будет выглядеть так:

```
class Client < ActiveRecord::Base
  has_many :clientAddress
end

class ClientAddress < ActiveRecord::Base
  belongs_to :client
end
```

И это всё. Это работает без единой строки конфигурации. В данном примере предполагается что в базе данных есть таблицы Client и ClientAddress. В ClientAddress есть внешний ключ с именем client\_id. Если соблюдать данное соглашение, то все будет работать само.

Как видно из примера использование Active Record значительно упрощает работу с базой данных и соответственно скорость разработки приложения. Программисту больше не нужно заботиться о реализации доступа к базам данных.

## **1.6 Структура базы данных МИС САМСОН**

Медицинская информационная система «САМСОН» использует для хранения три базы данных: s11, mes, kladr.

База данных s11 содержит информацию о пролеченных пациентах, их адреса, документы, связи с другими пациентами, этапы лечения, действия, совершаемые над пациентом и другие данные. Также база s11 содержит информацию о персонале учреждения, структуру учреждения, заключенные контракты с другими учреждениями, патенты, и огромное количество справочников.

База s11 содержит 350 таблиц. В тексте данной работы структура показана не будет, но она содержится в файле s11\_strict.pdf который приложен к данной работе (Приложение Б). Основные таблицы приведены на рисунке (Рисунок 4).

Таблица Client содержит информацию о пациентах, которые когда-либо обращались в ЛПУ.

Таблица OrgStructure содержит информацию о подразделениях, участках и отделениях внутри учреждения.

Таблица Event содержит события, связанные с обращениями пациента (плановое поступление, экстренное поступление, стоматология, проф. осмотр, обращение в регистратуру, запись на прием и.др.).



Таблица Action содержит действия, происходящие после обращения пациента (забор крови, перевод в другое отделение, рентгенография, различные операции и анестезии, и многое другое).

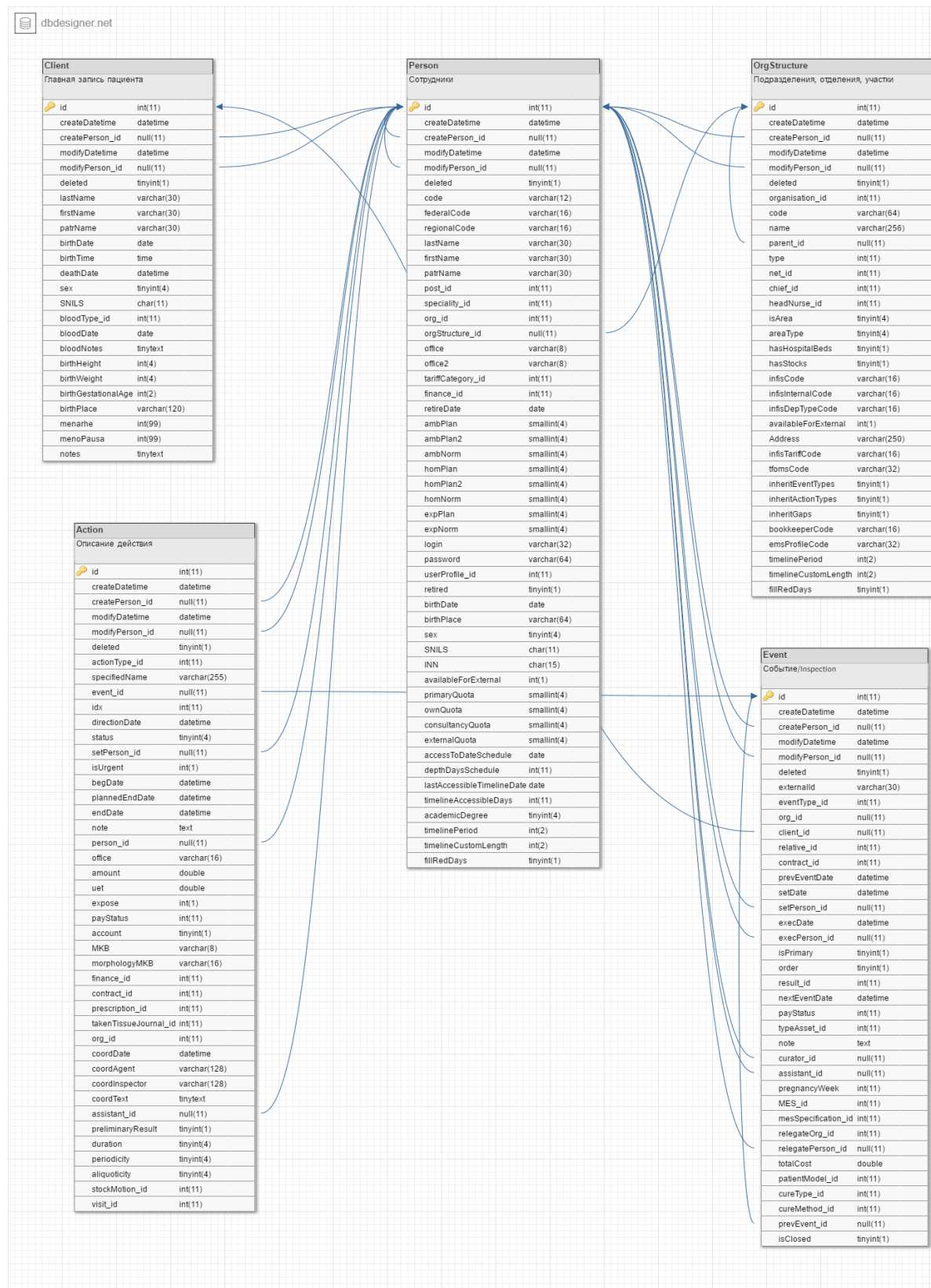


Рисунок 4 – Схема данных s11

Для хранения справочника адресов используется база данных KLADR, состоящая из 14 таблиц: ALTNames, DOMA, eisHouse, FLAT, infisAREA, infisREGION, infisSTREET, infisSTREETYP, infisToEis, KLADR, OKATO, SOCRBASE, STREET, tmpOKATO. Таблицы хранятся в СУБД MyISAM, в которой нет поддержки внешних ключей, поэтому на схеме данных нет связей (Рисунок 5). Связи в данной таблице организуются только логически, по полю CODE, которое содержит строку с кодом, например, AABBBBBBBBCCCCCCCCDDDDDE. Где AA – код региона, BBBBBBBB – код населённого пункта, CCCCCCCC – код улицы, DDDD – код домов, E – незначащая цифра, защищающая поле NAME

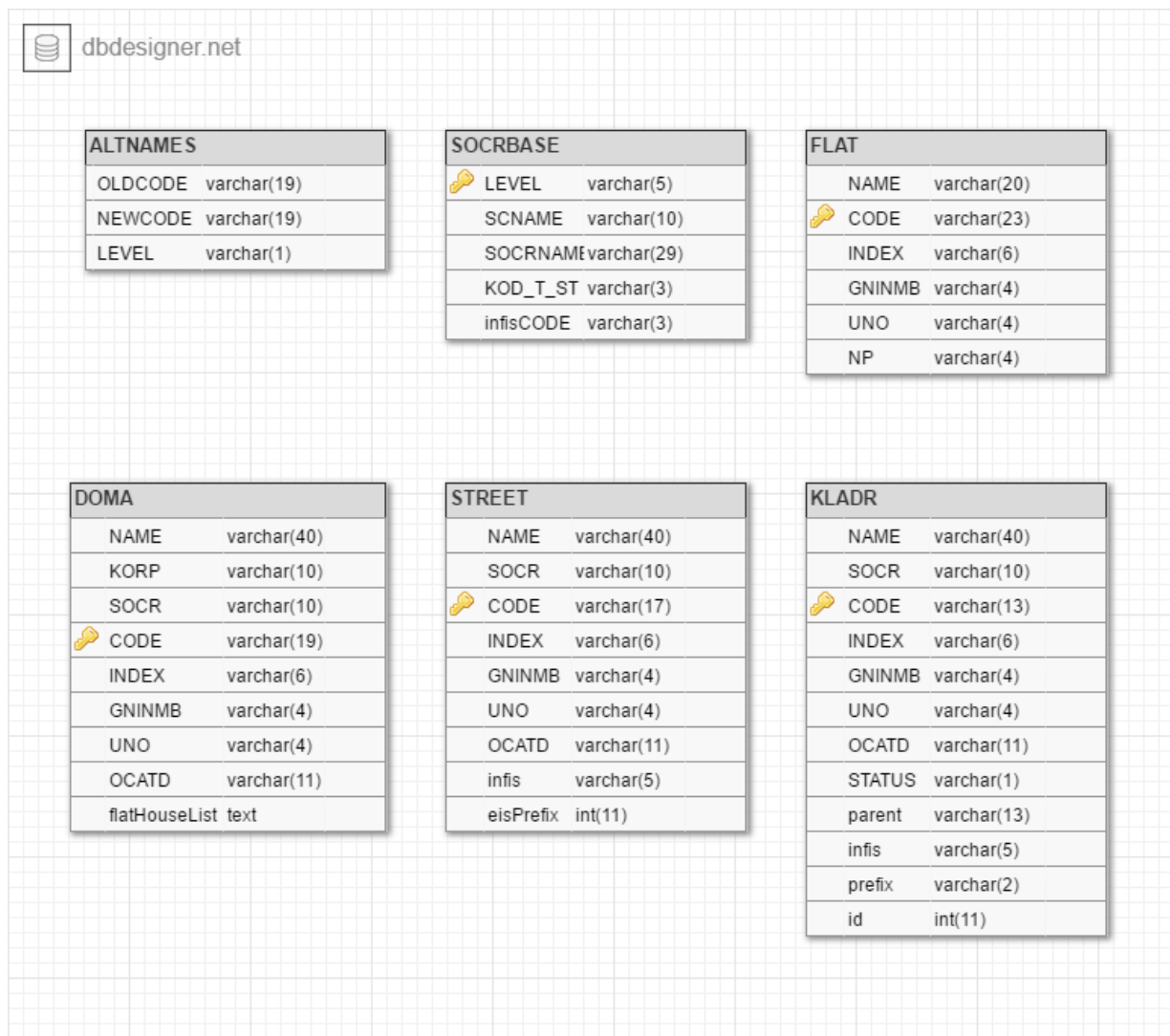


Рисунок 5 – Схема данных KLADR

Таблица ALT NAMES (Таблица 1) содержит сведения о изменении кодов.

Таблица 1 – kladr.ALT NAMES

Столбец	Тип	Комментарии
OLDCODE	varchar(19)	Старый код
NEWCODE	varchar(19)	Новый код
LEVEL	varchar(1)	Уровень объекта

Таблица DOMA (Таблица 2) содержит записи с объектами номеров домов, улиц городов и населенных пунктов.

Таблица 2 – kladr.DOMA

Столбец	Тип	Комментарии
NAME	varchar(40)	Номера домов, владений (в виде списка и/или диапазонов)
KORP	varchar(10)	Корпус дома
SOCR	varchar(10)	Сокращенное наименование типа объекта
CODE	varchar(19)	Код
INDEX	varchar(6)	Почтовый индекс
GNINMB	varchar(4)	Код ИФНС (ИМНС)
UNO	varchar(4)	Код территориального участка ИФНС
OCATD	varchar(11)	Код ОКАТО
flatHouseList	text	Список домов

Таблица FLAT (Таблица 3) содержит номера квартир.

Таблица 3 – kladr.FLAT

Столбец	Тип	Комментарии
NAME	varchar(20)	Номера квартир (в виде списка и/или диапазонов)
CODE	varchar(23)	Код
INDEX	varchar(6)	Почтовый индекс
GNINMB	varchar(4)	Код ИФНС (ИМНС)
UNO	varchar(4)	Код территориального участка ИФНС
NP	varchar(4)	Номер подъезда дома

Таблица KLADR (Таблица 4) содержит районы, города, поселки городского типа, сельсоветы, сельские населенные пункты.

Таблица 4 – kladr.KLADR

Столбец	Тип	Комментарии
NAME	varchar(40)	Название населенного пункта
SOCR	varchar(10)	Сокращение
CODE	varchar(13)	
INDEX	varchar(6)	
GNINMB	varchar(4)	Номер налоговой
UNO	varchar(4)	код участка
OCATD	varchar(11)	Код по ОКАТО
STATUS	varchar(1)	Статус местности
parent	varchar(13)	ид родителя
infis	varchar(5)	
prefix	varchar(2)	номер региона

id	int(11)	
----	---------	--

Таблица SOCRBASE (Таблица 5) содержит записи с краткими именами типов адресных объектов.

Таблица 5 – kladr.SOCRBASE

Столбец	Тип	Комментарии
LEVEL	varchar(5)	Уровень объекта данного типа
SCNAME	varchar(10)	Сокращенное наименование типа объекта
SOCRNAME	varchar(29)	Полное наименование типа объекта
KOD_T_ST	varchar(3)	Код типа объекта
infisCODE	varchar(3)	

Таблица STREET (Таблица 6) содержит улицы городов и населенных пунктов.

Таблица 6 – kladr.STREET

Столбец	Тип	Комментарии
NAME	varchar(40)	Наименование
SOCR	varchar(10)	Сокращенное наименование типа объекта
CODE	varchar(17)	
INDEX	varchar(6)	Почтовый индекс
GNINMB	varchar(4)	Код ИФНС
UNO	varchar(4)	Код территориального участка ИФНС
OCATD	varchar(11)	Код ОКАТО
infis	varchar(5)	
eisPrefix	int(11)	

База данных MES содержит значения медико-экономических стандартов. Таблицы базы данных стандартов медицинской помощи: MES, MES\_bloodPreparation, MES\_equipment, MES\_ksg, MES\_limitedBySexAge, MES\_medicament, MES\_mkb, MES\_nutrient, MES\_service, MES\_visit, ModelDescription, mrbBloodPreparation, mrbBloodPreparationType, mrbEquipment, mrbEquipmentGroup, mrbMedicament, mrbMedicamentDosageForm, mrbMedicamentGroup, mrbMESGroup, mrbModelAgeGroup, mrbModelAidCase, mrbModelAidPurpose, mrbModelCategory, mrbModelContinuation, mrbModelDiseaseClass, mrbModelExpectedResult, mrbModelInstitutionType, mrbModelSertificationRequirement, mrbModelStateBadness, mrbNutrient, mrbNutrientGroup, mrbService, mrbServiceGroup, mrbService\_Contents, mrbSpeciality, mrbVisitType. Структура Рисунок 6

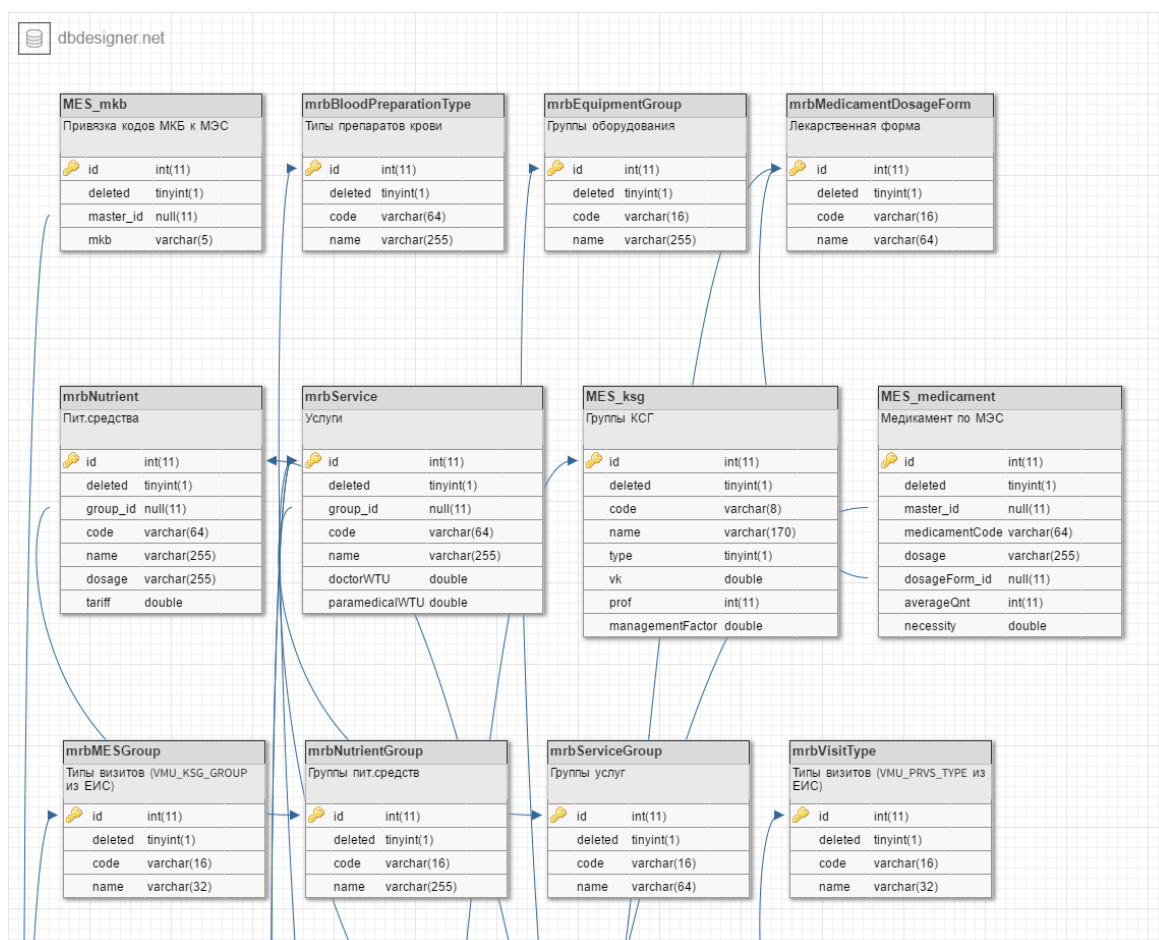


Рисунок 6 – Схема данных MES

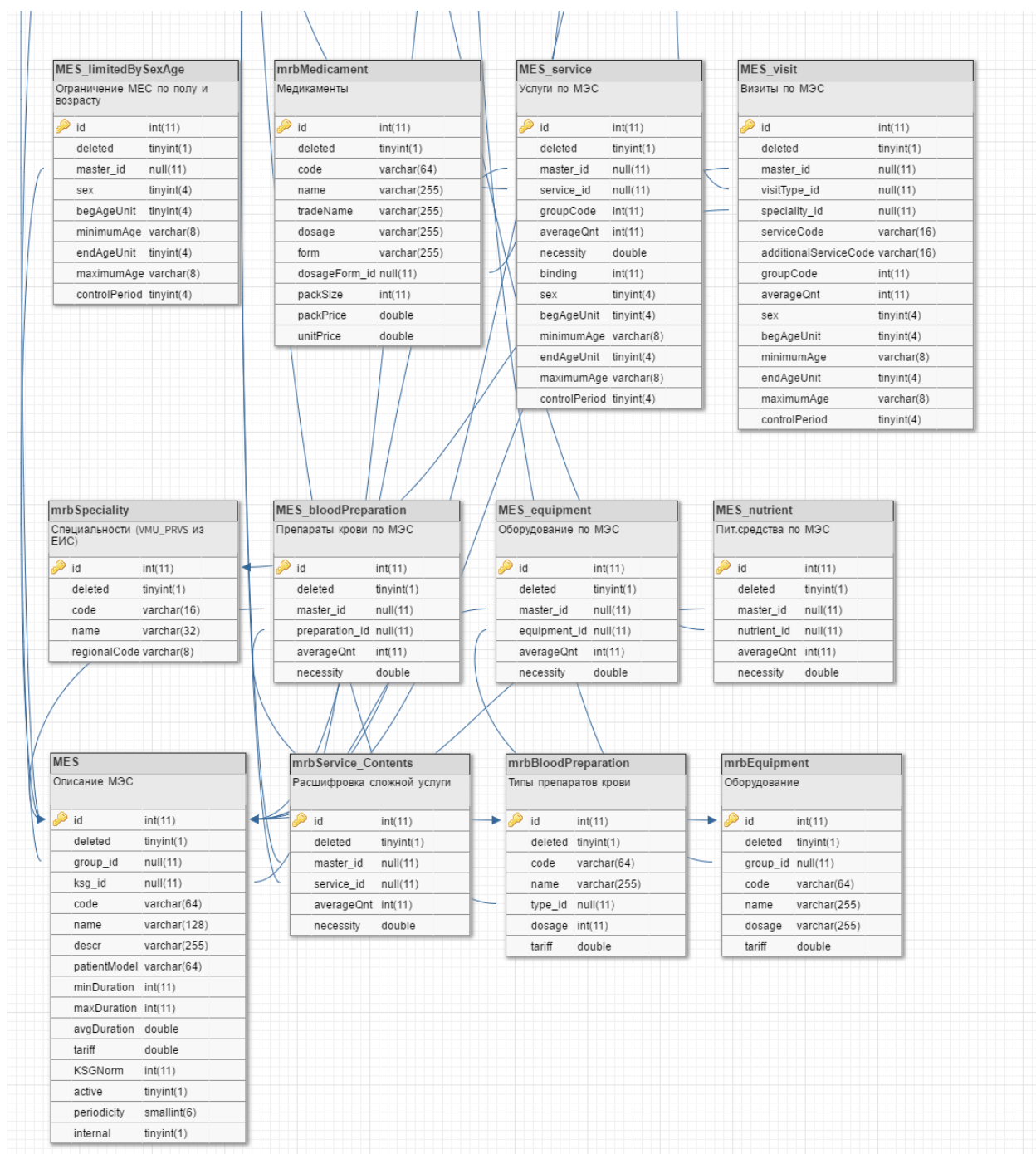


Рисунок 7 – Схема данных MES (продолжение)

В таблице MES (Таблица 7) хранятся описания МЭС.

Таблица 7 – mes.MES

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	

group_id	int(11)	{ mrbMESGroup }
ksg_id	int(11)	Группа КСГ
code	varchar(64)	Код
name	varchar(128)	Наименование
descr	varchar(255)	Описание
patientModel	varchar(64)	Модель пациента
minDuration	int(11)	Минимальная длительность
maxDuration	int(11)	Максимальная длительность
avgDuration	double	Средняя длительность
tariff	double	Тариф
KSGNorm	int(11)	Норматив визитов
active	tinyint(1)	
periodicity	smallint(6)	Периодичность в днях
internal	tinyint(1)	является внутренним

В таблице MES\_bloodPreparation (Таблица 8) хранится описание препаратов крови по МЭС

Таблица 8 – mes.MES\_bloodPreparation

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
master_id	int(11)	Применяется для МЭС {MES}
preparation_id	int(11)	Препарат {mrbBloodPreparation}
averageQnt	int(11)	среднее число препарата на курс лечения



necessity	double	частота назначения препарата в рамках стандарта
-----------	--------	---

В таблице MES\_equipment (Таблица 9) содержится описание оборудования по МЭС

Таблица 9 – mes.MES\_equipment

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
master_id	int(11)	Применяется для МЭС {MES}
equipment_id	int(11)	Оборудование {mrbEquipment}
averageQnt	int(11)	среднее число единиц препарата на курс лечения
necessity	double	частота назначения препарата в рамках стандарта

В таблице MES\_ksg (Таблица 10) содержится описание клинико-статистических групп.

Таблица 10 – mes.MES\_ksg

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	Отметка удаления записи
code	varchar(8)	Код КСГ
name	varchar(170)	Наименование КСГ
type	tinyint(1)	Тип КСГ 0-Неизвестный,1-Тер.,2-Комб.,3-Хир.,4-Проч.

vk	double	
prof	int(11)	
managementFactor	double	Управленческий коэффициент

В таблице MES\_limitedBySexAge (Таблица 11) содержится ограничения медико-экономических стандартов по полу и возрасту.

Таблица 11 – mes.MES\_limitedBySexAge

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	Отметка удаления записи
master_id	int(11)	МЕС {MES}
sex	tinyint(4)	Пол (0-неопределено, 1-М, 2-Ж)
begAgeUnit	tinyint(4)	Применимо для минимального возраста (0-любой, 1-Д, 2-Н, 3-М, 4-Г)
minimumAge	varchar(8)	Минимальный возраст (yy-mm-dd)
endAgeUnit	tinyint(4)	Применимо для максимального возраста (0-любой, 1-Д, 2-Н, 3-М, 4-Г)
maximumAge	varchar(8)	Максимальный возраст (yy-mm-dd)
controlPeriod	tinyint(4)	Период контроля {0-Текущая дата, 1-Конец текущего года, 2-Конец предыдущего года}

В таблице MES\_medicament (Таблица 12) содержится медикаменты применяемые для определенных МЭС.

Таблица 12 – mes.MES\_medicament

Столбец	Тип	Комментарии
id	int(11)	

deleted	tinyint(1)	
master_id	int(11)	применяется для МЭС {MES}
medicamentCode	varchar(64)	Код
dosage	varchar(255)	дозировка
dosageForm_id	int(11)	Лекарственная форма {mrbMedicamentDosageForm}
averageQnt	int(11)	СЧЕ – среднее число единиц медикамента (таблеток, ампул и т.д.) на курс лечения в стационаре
necessity	double	ЧН – частота назначения медикамента (потребность) в рамках настоящего стандарта

В таблице mes.MES\_mkb (Таблица 13) содержится привязки кодов МКБ к МЭС.

Таблица 13 – mes.MES\_mkb

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
master_id	int(11)	применяется для МЭС {MES}
mkb	varchar(5)	Код диагноза по МКБ

В таблице MES\_nutrient (Таблица 14) содержатся информация о питательных средствах по МЭС.

Таблица 14 – mes.MES\_nutrient

Столбец	Тип	Комментарии
id	int(11)	

deleted	tinyint(1)	
master_id	int(11)	Применяется для МЭС {MES}
nutrient_id	int(11)	Пит.средство {mrbNutrient}
averageQnt	int(11)	СЧЕ – среднее число единиц препарата на курс лечения
necessity	double	ЧН – частота назначения препарата в рамках настоящего стандарта

В таблице mrbNutrient (Таблица 15) содержится информация о питательных средствах.

Таблица 15 – mes.mrbNutrient

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
group_id	int(11)	Группа {mrbNutrientGroup}
code	varchar(64)	Код
name	varchar(255)	Наименование
dosage	varchar(255)	дозировка
tariff	double	Стоимость

В таблице mrbService (Таблица 16) содержатся услуги.

Таблица 16 – mes.mrbService

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
group_id	int(11)	Группа услуг {mrbServiceGroup}

code	varchar(64)	Код
name	varchar(255)	Наименование
doctorWTU	double	УЕТвр – усл. единицы трудозатрат врача
paramedicalWTU	double	УЕТср – усл. единицы трудозатрат сред. перс.

В таблице mrbService\_Contents (Таблица 17) содержатся расшифровки для сложных услуг.

Таблица 17 – mes.mrbService\_Contents

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
master_id	int(11)	Главная услуга {mrbService}
service_id	int(11)	Часть услуги {mrbService}
averageQnt	int(11)	СЧЕ – среднее число единиц на курс лечения
necessity	double	ЧН – частота назначения (потребность) в рамках настоящего стандарта

В таблице MES\_service (Таблица 18) содержатся услуги по МЭС.

Таблица 18 – mes.MES\_service

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
master_id	int(11)	применяется для МЭС {MES}
service_id	int(11)	услуга {mrbService}

groupCode	int(11)	Группировка услуг
averageQnt	int(11)	СК – среднее кол-во предоставления услуги (кратность) пациенту за период госпитализации
necessity	double	ЧП – частота предоставления услуги (потребность) в рамках настоящего стандарта
binding	int(11)	Пр.об. – признак объединения услуг в единую технологическую совокупность
sex	tinyint(4)	Пол (0-не определено, 1-М, 2-Ж)
begAgeUnit	tinyint(4)	Применимо для минимального возраста (0-любой, 1-Д, 2-Н, 3-М, 4-Г)
minimumAge	varchar(8)	Минимальный возраст (yy-mm-dd)
endAgeUnit	tinyint(4)	Применимо для максимального возраста (0-любой, 1-Д, 2-Н, 3-М, 4-Г)
maximumAge	varchar(8)	Максимальный возраст (yy-mm-dd)
controlPeriod	tinyint(4)	Период контроля {0-Текущая дата, 1-Конец текущего года, 2-Конец предыдущего года}

В таблице MES\_visit (Таблица 19) содержатся визиты по МЭС

Таблица 19 – mes.MES\_visit

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
master_id	int(11)	применяется для МЭС {MES}
visitType_id	int(11)	Тип визита {mrbVisitType}

speciality_id	int(11)	Специальность {mrbSpeciality}
serviceCode	varchar(16)	Код услуги
additionalServiceCode	varchar(16)	Дополнительный код услуги (ID_SERVICE_DATA)
groupCode	int(11)	Группировка визитов
averageQnt	int(11)	СК – среднее кол-во визитов (кратность) за период госпитализации
sex	tinyint(4)	Пол (0-не определено, 1-М, 2-Ж)
begAgeUnit	tinyint(4)	Применимо для минимального возраста (0-любой, 1-Д, 2-Н, 3-М, 4-Г)
minimumAge	varchar(8)	Минимальный возраст (yy-mm-dd)
endAgeUnit	tinyint(4)	Применимо для максимального возраста (0-любой, 1-Д, 2-Н, 3-М, 4-Г)
maximumAge	varchar(8)	Максимальный возраст (yy-mm-dd)
controlPeriod	tinyint(4)	Период контроля {0-Текущая дата, 1- Конец текущего года, 2-Конец предыдущего года}

В таблице ModelDescription (Таблица 20) содержится описание моделей.

Таблица 20 – mes.ModelDescription

Столбец	Тип	Комментарии
id	int(11)	
idx	int(11)	индекс для сортировки
name	varchar(64)	название
fieldIdx	int(11)	индекс в split(". ")
tableName	varchar(64)	Название таблицы

В таблице mrbBloodPreparation (Таблица 21) содержатся типы препаратов крови.

Таблица 21 – mes.mrbBloodPreparation

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
code	varchar(64)	Код
name	varchar(255)	название
type_id	int(11)	Тип препарата { mrbBloodPreparationType }
dosage	int(11)	Дозировка
tariff	double	Тариф

В таблице mrbEquipment (Таблица 22) содержится оборудование.

Таблица 22 – mes.mrbEquipment

Столбец	Тип	Комментарии
id	int(11)	
deleted	tinyint(1)	
group_id	int(11)	Группа { mrbEquipmentGroup }
code	varchar(64)	Код
name	varchar(255)	Наименование
dosage	varchar(255)	дозировка
tariff	double	Стоимость

В таблице mrbMedicament (Таблица 23) содержатся информация о медикаментах.

Таблица 23 – mes.mrbMedicament

Столбец	Тип	Комментарии
---------	-----	-------------



id	int(11)	
deleted	tinyint(1)	
code	varchar(64)	Код
name	varchar(255)	МНН – международное непатентованное название медикамента
tradeName	varchar(255)	Торговое наименование лек.средства
dosage	varchar(255)	дозировка
form	varchar(255)	форма выпуска
dosageForm_id	int(11)	Лекарственная форма { mrbMedicamentDosageForm }
packSize	int(11)	Количество единиц в упаковке
packPrice	double	Стоимость упаковки
unitPrice	double	Стоимость единицы

Таблицы mrbMedicamentGroup, mrbModelAgeGroup, mrbModelAidCase, mrbModelAidPurpose, mrbModelCategory, mrbModelContinuation, mrbModelDiseaseClass, mrbModelExpectedResult, mrbModelInstitutionType, mrbModelSertificationRequirement, mrbModelStateBadness, mrbNutrientGroup, mrbServiceGroup, mrbVisitType, mrbMESGroup, mrbEquipmentGroup, mrbBloodPreparationType, mrbMedicamentDosageForm имеют структуру аналогичную MES\_ksg (Таблица 10).

## **2 ПРАКТИЧЕСКАЯ ЧАСТЬ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

### **2.1 Техническое задание**

Наименование системы: Модуль «Статистика» комплекса САМСОН.

Плановые сроки начала и окончания работы: рабочий проект должен быть создан до конца июня 2016. Возможны доработки и исправления, но основной функционал к этому времени должен быть готов.

Источники и порядок финансирования: финансирование данного проекта не предусмотрено. Все расходы берет на себя заказчик.

Порядок оформления и предъявления заказчику результатов работ: работы по разработке модуля «Статистика» комплекса САМСОН идут поэтапно, каждый следующий этап устанавливается после выполненного по факту предыдущего.

Данный проект создается для заведующих отделений и отдела мониторинга высокотехнологичной медицинской помощи, чтобы они могли осуществлять расширенный поиск и проводить анализ результатов внутри медицинской информационной системы САМСОН. В данном проекте будет реализован следующий функционал:

1. Поиск
2. Фильтрация
3. Сортировка
4. Экспорт

Подробное техническое задание находится в приложении (Приложение А).

### **2.2 Структура модуля**

Модели, Контроллеры, Файлы.

Модели.

Client – модель содержит список пациентов.

Event – модель содержит события происходящие с пациентами (запись на прием, стационарное лечение, экстренное поступление и др.).

Action – модель содержит действия выполняемые над пациентами (движение, поступление, выписка, терапия, фотосессия, услуги и др.).

Organisation – модель содержит организации и ЛПУ.

OrgStructure – модель содержит подразделения, отделения, участки организаций.

Kladr – модель содержит классификацию адресов России.

Contract – Договор, содержит код оплаты.

Контроллеры.

Search\_window занимается логикой работы с интерфейсом пользователя. Содержит множество экшенов отвечающих за отображение справочников из базы данных, извлечение из интерфейса пользователя необходимых данных для составления запроса и экспорт таблиц во внешние файлы.

S11ActiveRecord реализует интерфейс доступа к базе данных для приложения.

Файлы.

./database/S11\_settings.rb – содержит настройки доступа к БД.

./database/S11.rb – содержит код который осуществляет подключение объектных моделей бд к приложению.

./database/tables/\*.rb – файлы содержат объектные модели, необходимые для доступа к таблицам базы данных.

./module/adv\_qt.rb – расширяет класс Qt для более простой работы с фреймвоком.

./module/crutch.rb – содержит исправления необходимые для верного построения запросов к базе данных.

./module/export.rb – описывает класс для экспорта данных из приложения в внешний мир.

./Application.rb –занимается инициализацией и отображением главного окна приложения.

`./main.rbw` –подключает все модули приложения и запускает приложение.

`./Search_window.rb` – занимается обработкой пользовательского интерфейса.

`./ui_search_window.rb` – пользовательский интерфейс.

В дополнение к этому для нормальной работы приложения необходимы `gem`'ы (внешние модули):

«`qtbindings`» – модуль, осуществляющий «привязку» Qt к ruby;

«`activerecord`» – соединяет классы таблиц реляционных баз данных, чтобы упростить создание и использование объектов, данные которых требуют хранения в базе данных. Объекты Active Record содержат сохраненные данные и описывают поведение, для работы с этими данными;

«`squeel`» – оптимизирует `activerecord` позволяя писать запросы меньшим количеством строк.

## **2.3 Интерфейс и функциональные возможности**

Модулем «Статистика» может пользоваться любой пользователь, имеющий на своём компьютере приложение «Самсон-виста», так как модуль «Статистика» получает доступ к базе данных из настроек «САМСОН». Общий вид приложения представлен на рисунке (Рисунок 8).

Статистика									
Результаты запроса									
	Фамилия	Имя	Отчество	Дата рождения	Пол	СНИЛС	Отделение	Дата начала события	Дата окончания события
1	Кба	Диниза	Хайсаровна	2004-05-31	Ж	92562005554	11 травнаторг...	2016-01-21 14:3...	2016-01-25 16:29:36 UTC
2	Аба	Эльвира	Владимировна	1965-12-14	Ж	82764933328	7 травнаторг...	2016-01-18 13:3...	
3	Бба	Дмитрий	Николаевич	1982-09-14	М	83050363417	1 травнаторг...	2016-01-13 08:2...	
4	Гба	Нина	Семёновна	1940-09-01	Ж	87519344945	2 травнаторг...	2016-01-21 12:1...	
5	Гба	Николай	Андреевич	1970-04-03	М	81552939857	10 травнаторг...	2016-01-20 13:2...	
6	Гба	Раиса	Алексеевна	1945-08-02	Ж	81405459770	10 травнаторг...	2016-01-21 10:3...	
7	Гба	Анна	Робертовна	1975-03-31	Ж	92995341541	10 травнаторг...	2016-01-22 13:1...	
8	Еба	Светлана	Александровна	1981-06-13	Ж	83736923527	15 травнаторг...	2016-01-15 09:0...	
9	Еба	Любовь	Ивановна	1944-10-16	Ж	82678366712	10 травнаторг...	2016-01-15 11:0...	
10	Иба	Алина	Дмитриевна	2008-09-16	Ж	94865999678	9 травнаторг...	2016-01-12 13:1...	2016-01-18 14:24:07 UTC
11	Иба	Людмила	Леонидовна	1974-03-02	Ж	83736923527	3 травнаторг...	2016-01-11 11:3...	2016-01-19 10:46:59 UTC
12	Кба	Павел	Давыдович	1950-11-12	М	85974919196	7 травнаторг...	2016-01-21 10:0...	
13	Кба	Зинаида	Георгиевна	1951-04-02	Ж	87261857287	12 травнаторг...	2016-01-25 09:3...	
14	Кба	Алена	Сергеевна	1998-07-24	Ж	92050572033	14 травнаторг...	2016-01-18 15:3...	
15	Кба	Надежда	Ильинична	1948-12-15	Ж	82839144992	5 травнаторг...	2016-01-22 13:1...	
16	Кба	Людмила	Юрьевна	1973-02-11	Ж	92777423638	10 травнаторг...	2016-01-22 09:4...	
17	Кба	Евгений	Юрьевич	1983-02-07	М	90109606696	8 травнаторг...	2016-01-25 09:5...	
18	Кба	Вера	Васильевна	1939-10-26	Ж	85585969426	16 травнаторг...	2016-01-25 10:2...	
19	Лба	Леонид	Александрович	1969-04-18	М	86795636064	8 травнаторг...	2016-01-11 14:5...	2016-01-22 11:07:24 UTC
20	Мба	Нина	Андреевна	1951-08-05	Ж	96908505416	1 травнаторг...	2016-01-03 13:0...	

Рисунок 8 – Общий вид модуля «Статистика»

При запуске приложения пользователю необходимо ввести свои учетные данные соответствующие учетным данным из системы «САМСОН» (Рисунок 9).

Вход

?

×

Имя

админ

Пароль

OK

Close

Рисунок 9 – Авторизация

При вводе неверных данных будет выдано соответствующее сообщение (Рисунок 10)

Ошибка

×

Неверное имя пользователя или пароль.

OK

Рисунок 10 – Ошибка авторизации

В приложении есть возможность задавать необходимые пользователю столбцы (Рисунок 11). При выборе нужного столбца он отобразится после нажатия кнопки «Применить» находящейся в нижней правой части экрана.

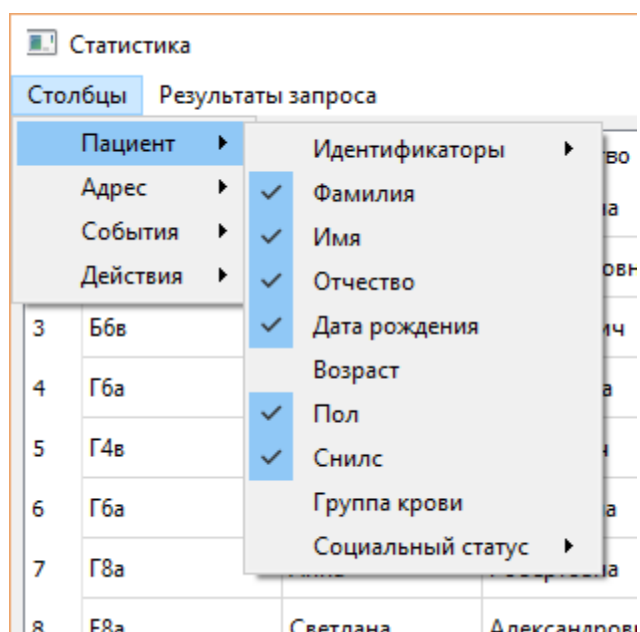


Рисунок 11 – Выбор столбцов для отображения

После нажатия кнопки «Применить» выполнится запрос к базе данных, результат запроса будет отображен в главной таблице окна, а в строке состояния появится надпись об общем количестве записей, которые вернул запрос (Рисунок 12).

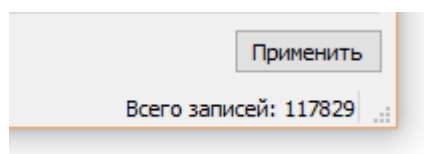


Рисунок 12 – Общее число записей запроса

Сортировка пациентов производится при нажатии на заголовок столбца который нужно отсортировать.

Также у пользователя есть возможность экспортировать таблицу в файл для последующего открытия в его любимом редакторе (Рисунок 13).

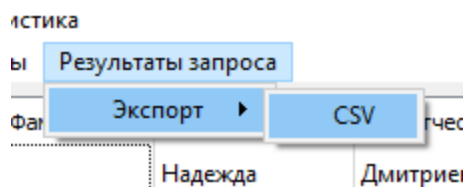


Рисунок 13 – Экспорт

Фильтрация пациентов производится при выборе фильтров в правой части окна.

При установке галочки в контроле «Код» становится доступна фильтрация по кодам пациента (Рисунок 14). У пациента может быть несколько кодов: его уникальный идентификатор в базе данных «САМСОН», фото номер, архивный номер и идентификаторы других систем учета пациентов.

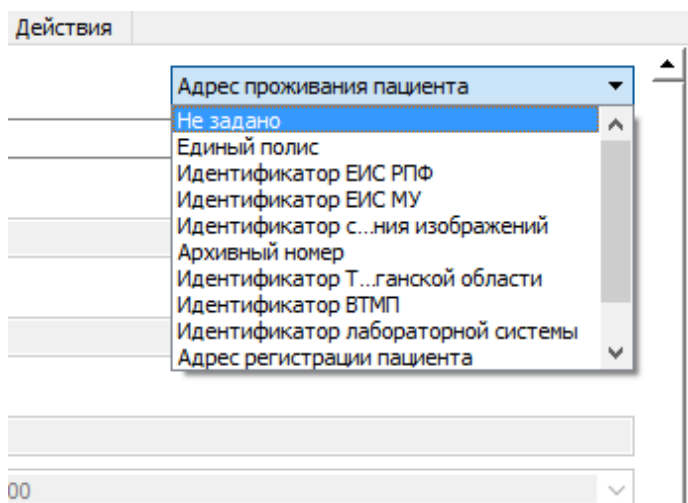


Рисунок 14 – Коды пациента

Фильтрация доступна и по всем основным значениям (Рисунок 15): имени, фамилии, отчеству, дате рождения, возрасту, полу, СНИЛС.

☐ Фамилия  
☐ Имя  
☐ Отчество  
☐ Дата рождения 01.01.2000  
☐ по 01.01.2000  
☐ Возраст 0  
☐ по 150  
☐ Пол ☒ М ☐ Ж

Рисунок 15 – Основные фильтры

Если у пациента внесены документы в базу, то его можно найти по ним (Рисунок 16).

Можно найти пациентов по определенной области, городу, району, улице, дому (Рисунок 17). Существует возможность выбора адреса регистрации или проживания.

☒ Документ  
 Не задано  
 Паспорт РФ  
 Заграничный паспорт  
 Свидетельство о рождении  
 Удостоверение офицера  
 Справка об освобождении  
 Паспорт Морфлота РФ  
 Военный билет  
 Дипломатический паспорт РФ  
 Иностранный паспорт

Рисунок 16 – Фильтрация по документам

☒ Адрес ☒ Регистрации ☐ Проживания  
 Курганская обл  
 Курган г  
 Калинина ул  
 Дом Корп Кв

Рисунок 17 – Фильтр по адресам



Вкладка обращение позволяет фильтровать пациентов по их обращениям в ЛПУ (Рисунок 18).

Контроль «Тип» при первой установке галочки подгружает список типов событий (Рисунок 19).

Также на вкладке обращение можно указать даты назначения, выполнения и следующей явки (Рисунок 20).

После установки галочки в контроле «Подразделение» из базы данных будут загружены все подразделения и построена древовидная структура (Рисунок 21)

Статистика

Столбцы Результаты запроса

	Имя	Отчество	Дата рождения	Пол	СНИЛС	Отде.
4	Нина	Семёновна	1940-09-01	Ж	87519344945	2 травма
5	Николай	Андреевич	1970-04-03	М	81552939857	10 травма
6	Раиса	Алексеевна	1945-08-02	Ж	81405459770	10 травма
7	Анна	Робертовна	1975-03-31	Ж	92995341541	10 травма
8	Светлана	Александровна	1981-06-13	Ж	83736923527	15 травма
9	Любовь	Ивановна	1944-10-16	Ж	82678366712	10 травма
10	Алина	Дмитриевна	2008-09-16	Ж	94865999678	9 травма
11	Людмила	Леонидовна	1974-03-02	Ж		3 травма
12	Павел	Давыдович	1950-11-12	М	85974919196	7 травма
13	Зинаида	Георгиевна	1951-04-02	Ж	87261857287	12 травма
14	Алена	Сергеевна	1998-07-24	Ж	92050572033	14 травма
15	Надежда	Ильинична	1948-12-15	Ж	82839144992	5 травма
16	Людмила	Юрьевна	1973-02-11	Ж	92777423638	10 травма
17	Евгений	Юрьевич	1983-02-07	М	90109606696	8 травма
18	Вера	Васильевна	1939-10-26	Ж	85585969426	16 травма
19	Леонид	Александрович	1969-04-18	М	86795636064	8 травма
20	Нина	Андреевна	1951-08-05	Ж	96908505416	1 травма
21	Константин	Евгеньевич	2000-12-15	М	97575099519	14 травма
22	Вячеслав	Евгеньевич	1944-08-15	М	85434896080	16 травма
23	Данил	Венерович	2000-02-29	М	95619857490	17 травма
24	Мария	Валерьевна	1988-01-17	Ж	87206091617	10 травма
25	Владимир	Владимирович	1977-02-22	М	86883712489	15 травма
26	Светлана	Петровна	1963-05-18	Ж	81479961102	15 травма
27	Александр	Васильевич	1973-02-28	М	81423335764	3 травма
28	Галина	Анатольевна	1952-04-08	Ж	80934544984	8 травма
29	Ксения	Евгеньевна	2001-07-22	Ж	90830794686	6 травма
30	Алла	Ивановна	1952-02-18	Ж	86608054642	1 травма
31	Юлия	Викторовна	1975-10-08	Ж	83733156677	12 травма
32	Александр	Сергеевич	2000-09-23	М	94925805702	17 травма
33	Татьяна	Ивановна	1960-01-13	Ж	82086541556	15 травма
34	София	Рославовна	2005-07-28	Ж	02377288342	18 травма

Картотека Обращение Действия

☒ Тип

Стационарное лечение

☒ Дата назначения

01.01.2016

01.01.2017

☐ Не выполнено

☐ Дата выполнения

01.01.2000

01.01.2000

☐ Дата следующей явки

01.01.2000

01.01.2000

☐ Подразделение

☐ Специальность

☐ Врач

☐ Тип обращения

☐ Порядок обращения

☐ Состоит на ДН

☐ ЛПУ

☐ Любое ЛПУ кроме базового

☐ Направитель

☐ Код МЭС

☐ Результат

☐ Коды диагнозов по МКБ

☐ Предварительные ☐ Сопутствующие

01.01.2000

Сбросить Применить

Рисунок 18 – Вкладка обращение

☒ Тип

Стационарное лечение

Плановое поступление

Стоматология

Проф. осмотр

Протезно-ортопедический

Регистратура

Экстренный

Запись на прием

Стационарное лечение

Амбулаторный прием

Лаборатория

01.01.2000

Рисунок 19 – Типы обращений

☒ Дата назначения

01.01.2016

01.01.2017

☐ Не выполнено

☐ Дата выполнения

01.01.2000

01.01.2000

☐ Дата следующей явки

01.01.2000

01.01.2000

Рисунок 20 – Обращение даты

☒ Подразделение

УЗИ

функц

проф

▼ УЗИ

УЗИ гл.корпус

УЗИ КДО

КДЛ

МРТ

маркет

гнойн 01

гнойн 02

Рисунок 21 – Обращение, подразделение

Вкладка действия содержит фильтры для поиска действий совершаемых над пациентом

Тип действия отвечает за выбор действия над пациентом, основная группировка типов действия это: «Статус», «Лечение», «Диагностика» и «Прочие мероприятия». После выбора одного из типов из базы данных загружаются более подробные фильтры (Рисунок 22).

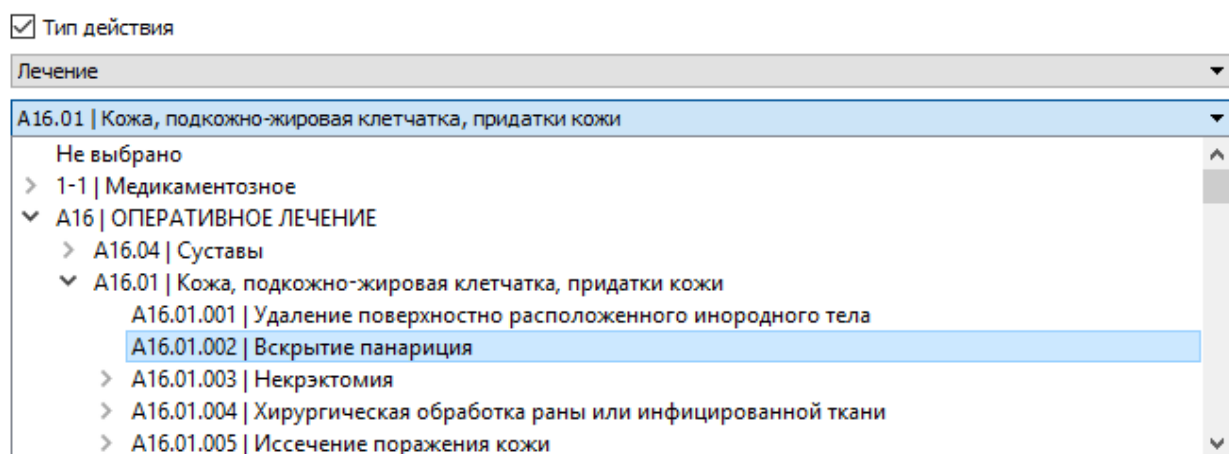


Рисунок 22 – Действия, типы

Подразделение назначившего, специальность назначившего и назначил на вкладке действий позволяет находить всех пациентов у определенных врачей.

Коды диагнозов по МКБ позволяют фильтровать результаты поиска по кодам диагнозов.

Кнопка «Сброс» выключает все фильтры и устанавливает их значения по умолчанию.

В нижней части таблицы есть возможность выбора количества отображения строк (Рисунок 23).

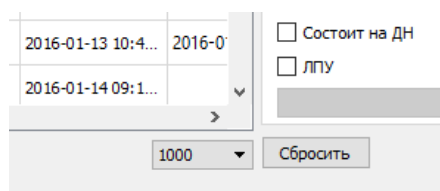


Рисунок 23 – Сколько строк нужно отобразить

Если в результате запроса строк для отображения больше чем указано в настройках, то активируется панель с нумерацией страниц, которая позволяет либо переходить на прошлую или следующую страницы, либо выбрать произвольную (Рисунок 24).

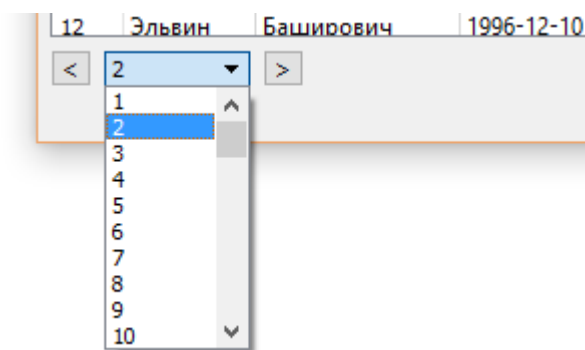


Рисунок 24 – Постраничная навигация

У пользователя есть возможность поменять размер таблицы, сделать её больше или меньше за счет уменьшения области фильтров. Таблицу с результатами или область фильтров можно скрыть совсем если что-то из них в данный момент ненужно (Рисунок 25).

При закрытии приложения настройки отображения сохраняются в файл настроек находящийся в папке с приложением. Настройками отображения являются размещение и размер окна, флаг о полноэкранном режиме, настройки, связанные с отображением страниц и отображаемые столбцы. Во время открытия приложения сохраненные настройки считываются из файла и применяются к пользовательскому интерфейсу.

Рисунок 25 – Скрытие таблицы результатов

## 2.4 Расчеты экономической эффективности использования модуля

При разработке модуля вывода статистики информационной системы «САМСОН» были задействованы два человека: студент, пишущий диплом и руководитель дипломного проекта.

Студент в проекте отвечает за анализ предметной области, разработку внутренних алгоритмов модуля, проектирование интерфейса и отладку программы. Руководитель занимается контролем хода работ и консультирует студента при необходимости. Этапы разработки приведены в таблице (Таблица 24).

Трудоемкость работ определялась с учетом срока окончания работ, объемом выполняемых функций. Выбор комплекса работ по разработке проекта производится в соответствии с ГОСТ 19.102-77 «Единая система программной документации», устанавливающего стадии разработки.

Таблица 24 – Этапы разработки

Стадия разработки	Длительность работ (в днях)		
	минимум	максимум	ожд.
Техническое задание			
Обоснование необходимости разработки программы			
Постановка задачи	1	2	1
Сбор исходных материалов	2	3	2,4
Выбор и обоснование критериев эффективности качества разрабатываемой программы	11	2	7,4
Обоснование необходимости проведения научно-исследовательских работ	1	1	1
Научно-исследовательские работы			
Определение структуры входных и выходных данных.	3	4	3,4
Предварительный выбор методов решения задач.	10	15	12
Обоснование целесообразности применения ранее разработанных программ	2	3	2,4
Определение требований к техническим средствам.	1	2	1,4
Обоснование принципиальной возможности решения	1	1	1

поставленной задачи			
Разработка и утверждение технического задания			
Определение требований к программе.	3	5	3,8
Разработка технико-экономического обоснования разработки программы.	1	2	1,4
Определение стадий, этапов и сроков разработки программы и документации на неё.	1	2	1,4
Выбор языков программирования.	1	2	1,4
Определение необходимости проведения научно-исследовательских работ на последующих стадиях.	1	2	1,4
Согласование и утверждение технического задания.	1	2	1,4
Эскизный проект			
Разработка эскизного проекта			
Предварительная разработка структуры входных и выходных данных.	7	10	8,2
Уточнение методов решения задачи.	2	3	2,4
Разработка общего описания алгоритма решения задачи	10	12	11
Разработка технико-экономического обоснования.	2	3	2,4
Утверждение эскизного проекта			
Разработка пояснительной записки.	1	2	1,4
Согласование и утверждение эскизного проекта.	1	2	1,4
Рабочий проект			
Разработка программы			
Программирование и отладка программы.	34	42	37



Разработка программной документации			
Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.	2	3	2,4
Испытания программы			
Разработка, согласование и утверждение порядка и методики испытаний.	1	2	1,4
Проведение предварительных государственных, межведомственных, приёмо-сдаточных и других видов испытаний.	2	3	2,4
Корректировка программы и программной документации по результатам испытаний.	2	2	2

Продолжительность работ определяется по формуле 1:

$$T_0 = (3 * T_{min} + 2 * T_{max})/5 \quad (1)$$

$T_0$  — ожидаемая продолжительность работ;

$T_{min}$  и  $T_{max}$  — соответственно наименьшая и наибольшая по мнению эксперта длительность работы.

Суммарные затраты на проектирование и разработку приложения определяются суммой всех затраченных средств, а именно затраты на использование машинного времени и расходы на материалы. Оклад разработчиков учитывать в данном проекте не будем.

Так как разработанный программный продукт должен быть создан на ПК, к суммарным затратам на разработку необходимо добавить и использование машинного времени, начисляемое по формуле (**Ошибка! Источник ссылки не найден.**).

$$M_B = t_{MB} S_{MB} \quad (2)$$

где,

$t_{MB}$  — машинное время компьютера, необходимое для разработки программного продукта;

$S_{\text{МВ}}$  – стоимость 1 часа машинного времени.

Рассчитаем стоимость 1 часа использования ПК, без учета стоимости покупки по формуле 3:

$$S_{\text{МВ}} = V * A * (C / 1000) \quad (3)$$

Где,

$V$  – вольты;

$A$  – амперы;

$C$  – стоимость 1 кВт-ч

$$S_{\text{МВ}} = 19 * 4.74 * (3.78 / 1000) = 0.34 \text{ руб}$$

Учитывая время на проектирование приложения, представленное в таблице (Таблица 24) и стоимость разработку программного продукта на компьютере за один час, расходы будут равны  $115 * 8 * 0.34 = 312.8$  руб.

Приведем расчет затрат на материалы используя данные представленные в таблице (Таблица 25).

Таблица 25 – Затраты на материалы

Наименование	Количество шт.	Стоимость руб./ед. изм.	Общая стоимость (в рублях)
Печать черно- белых листов	70	3	210
CD-R диск	1	40	40
Итого			250

Итого сумма затрат на разработку приложения будет составлять 312.8 руб.

Эксплуатационные расходы по созданному модулю отсутствуют, поэтому общая стоимость внедрения и разработки будет равна  $312.8 + 250 = 562.8$  руб.

## **ЗАКЛЮЧЕНИЕ**

В ходе работы над проектом удалось достичь всех поставленных целей. Была проанализирована медицинская информационная система «САМСОН», выявлены проблемы связанные с выводом информации из неё. На основе анализа этих проблем был разработан модуль «Статистика», который позволяет пользователю выбирать необходимые для отображения «Колонки» и устанавливать параметры поиска.

Благодаря разработанному модулю упрощено составление свободных отчетов по работе клинических отделений и стационара в целом, проведения анализа работы учреждения. Упрощена возможность переноса пациентов в архивную базу данных «Нозологический и функциональный каталог». Исключена необходимость заполнения «Журнала учета стационарных больных».

Интерфейс приложения получился простой и удобный для пользователя. Уровень подготовки пользователей должен соответствовать начальной компьютерной грамотности, предполагающей наличие умений и навыков работы с ПК.

## СПИСОК ЛИТЕРАТУРЫ

1. Клиент-серверная СУБД. [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/Клиент-серверная\\_СУБД](https://ru.wikipedia.org/wiki/Клиент-серверная_СУБД). (дата обращения 20 Июнь 2016).
2. М. Шлее, Qt 4.8. Профессиональное программирование на C++, БХВ-Петербург, 2012. - 894с
3. Медицинская информационная система. [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/Медицинская\\_информационная\\_система](https://ru.wikipedia.org/wiki/Медицинская_информационная_система). (дата обращения 20 Июнь 2016).
4. Назначение системы. [Электронный ресурс]. URL: <http://mis-region.ru/about/naznachenie-sistemy/>. (дата обращения 20 Июнь 2016).
5. Разработка/Руководства/Введение в Qt4 на Ruby — KDE TechBase. [Электронный ресурс]. URL: [https://techbase.kde.org/Development/Tutorials/Qt4\\_Ruby\\_Tutorial/ru](https://techbase.kde.org/Development/Tutorials/Qt4_Ruby_Tutorial/ru). (дата обращения 20 Июнь 2016).
6. Х. Фултон, Программирование на языке Ruby, Litres, 2014. - 688с
7. Gerasimov, САМСОН – Официальный сайт проекта , САМСОН-ВИСТА , Электронная регистратура , Электронная история болезни , Медстатистика , Типовая МИС , МИС , Автоматизация ЛПУ , самозапись, Стационар , поликлиника , больница, Запись на прием к врачу – Главная. [Электронный ресурс]. URL: <http://samson-rus.com/>. (дата обращения 20 Июнь 2016).
8. J. Bodnar, Ruby Qt tutorial. [Электронный ресурс]. URL: <http://zetcode.com/gui/rubyqt/>. (дата обращения 20 Июнь 2016).
9. MySQL.RU ∴. Одобрено лучшими российскими программистами. [Электронный ресурс]. URL: <http://www.mysql.ru/docs/man/>. (дата обращения 20 Июнь 2016).
10. Ruby — Викиучебник. [Электронный ресурс]. URL: <https://ru.wikibooks.org/wiki/Ruby>. (дата обращения 20 Июнь 2016).

11.Rusrails: Интерфейс запросов Active Record. [Электронный ресурс]. URL: <http://rusrails.ru/active-record-query-interface>. (дата обращения 20 Июнь 2016).

**Техническое задание**

**1. Общие сведения**

**1.1. Наименование системы**

**1.1.1. Полное наименование системы**

Модуль «Статистика» комплекса САМСОН.

**1.1.2. Краткое наименование системы**

Модуль «Статистика».

**1.2. Наименование организаций – Заказчика и Разработчика**

**1.2.1. Заказчик**

Заказчик: ФГБУ «РНЦ «ВТО» им. акад Г.А. Илизарова».

Представитель заказчика:

Адрес фактический: г. Курган, ул. М. Ульяновой, 6.

Телефон:

**1.2.2. Разработчик**

Разработчик: Поленков К.П.

Адрес фактический: г. Курган, 6-а мкр. 8-69

Телефон: +7 (963) 00 24 303

**1.3. Плановые сроки начала и окончания работы**

Рабочий проект должен быть создан до конца июня 2016. Возможны доработки и исправления, но основной функционал к этому времени должен быть готов.

**1.4. Источники и порядок финансирования**

Финансирование данного проекта не предусмотрено. Все расходы берет на себя заказчик.

**1.5. Порядок оформления и предъявления заказчику результатов работ**

Работы по разработке модуля «Статистика» комплекса САМСОН идут поэтапно, каждый следующий этап устанавливается после выполненного по факту предыдущего.

**2. Назначение и цели создания системы**

## **2.1. Назначение системы**

Данный проект создается для заведующих отделений, и отдела мониторинга высокотехнологичной медицинской помощи, чтобы они могли осуществлять расширенный поиск и проводить анализ результатов внутри медицинской информационной системы САМСОН.

В данном проекте будет реализован следующий функционал:

- Поиск
- Фильтрация
- Сортировка
- Экспорт

## **2.2. Цели создания системы**

Разработка модуля «Статистика» комплекса САМСОН.

## **3. Требования к системе**

### **3.1. Требования к системе в целом**

#### **3.1.1. Требования к структуре и функционированию системы**

Структура системы предусматривает следующие разделы:

- Главное меню приложения
- Таблица для вывода результатов
- Строка состояния
- Вкладки для выбора фильтров

В главном меню будет содержаться несколько меню. Меню «Столбцы» содержит столбцы для отображения в таблице, при необходимости столбцы можно показывать или прятать. Меню «Результаты запроса» содержит функции экспорта.

Таблица для вывода результатов будет отображать результаты работы программы.

В строке состояния будет отображаться информация, связанная с работой программы и общее количество записей по текущему фильтру.

Вкладки для выбора фильтров будут содержать фильтры, благодаря которым можно сузить результат выборки для анализа.

#### **3.1.2. Требования к эргономике и технической эстетике**

Модуль «Статистика» должен обеспечивать удобный для пользователя интерфейс, отвечающий следующим требованиям.

В части внешнего оформления: должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя.

### **3.1.3. Требования к защите информации от несанкционированного доступа**

Авторизация в системе будет осуществляться с помощью логина и пароля.

### **3.1.4. Требования по сохранности информации при авариях**

В системе «САМСОН» уже обеспечено резервное копирование. При разработке будет использоваться система управления версиями GIT.

## **3.2. Требования к видам обеспечения**

### **3.2.1. Требования к математическому обеспечению**

Не предъявляются.

### **3.2.2. Требования к информационному обеспечению**

Требования по применению систем управления базами данных.

Для реализации подсистемы хранения данных должна использоваться СУБД MySQL.

Требования к процедуре придания юридической силы документам, продуцируемым техническими средствами системы.

Требования не предъявляются.

### **3.2.3. Требования к лингвистическому обеспечению**

При реализации системы должны применяться следующие языки высокого уровня: Ruby версии 2.2, Qt с версией 4.8.6.\*

Должны выполняться следующие требования к кодированию и декодированию данных:

UTF-8.

Для реализации алгоритмов манипулирования данными необходимо использовать стандартный язык запроса к данным SQL.

## **4. Состав и содержание работ по созданию системы**



Работа по созданию системы выполняются в три этапа:

Проектирование. Разработка эскизного проекта. Разработка технического проекта. Продолжительность выполнения 1 этапа составляет 2 месяца.

Разработка рабочей документации. Адаптация программы. Продолжительность выполнения 2 этапа составит 1 месяц.

Ввод Системы в эксплуатацию. На исполнения данного этапа потребуется 2 месяца.

## **5. Стадии и этапы разработки**

### **5.1. Стадии разработки**

Разработка должна быть проведена в три стадии:

1. Разработка технического задания
2. Проектирование и разработка
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 методик испытаний;
2. проведение приемо-сдаточных испытаний;
3. корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

## **6. Требования к документированию**

Состав программной документации должен включать в себя:

- техническое задание;
- руководство пользователя.

### **Список файлов**

Папка «Текст/\*» содержит файлы текста дипломной работы;

Папка «Программа/\*» содержит файлы программы.