Оглавление

[Введение 4](#_Toc452387946)

[1 Теоретическая часть квалификационной работы 5](#_Toc452387947)

[1.1 Об медицинских информационных системах 5](#_Toc452387948)

[1.2 Обзор существующих медицинских информационных систем 6](#_Toc452387949)

[1.3 Предпосылки разработки модуля "Статистика" 7](#_Toc452387950)

[1.4 Обзор средств разработки 8](#_Toc452387951)

[1.4.1 Язык программирования для разработки 8](#_Toc452387952)

[1.4.2 Библиотеки элементов интерфейса 8](#_Toc452387953)

[1.4.3 Система управления базами данных 10](#_Toc452387954)

[1.5 Выбор средств разработки 11](#_Toc452387955)

[1.6 Структура базы данных МИС САМСОН 12](#_Toc452387956)

[2 Практическая часть квалификационной работы 13](#_Toc452387957)

[2.1 Техническое задание 13](#_Toc452387958)

[2.2 Структура модуля 14](#_Toc452387959)

[2.3 Интерфейс и функциональные возможности 16](#_Toc452387960)

[Заключение 17](#_Toc452387961)

[Список литературы 18](#_Toc452387962)

[Приложение А. Техническое задание 19](#_Toc452387963)

# Введение

На данный момент у всех больших компаний занимающихся лечением пациентов есть свои медицинские системы, которые выполняют различные функции сбора, хранения, обработки, выдачи информации, и подчеркивают серьезность действий компании.

Цель дипломного проекта: разработка модуля «Статистика» комплекса САМСОН для 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

# Теоретическая часть квалификационной работы

## Об медицинских информационных системах

В современном мире большинство медицинских учреждений, заинтересованных в эффективном обслуживании пациентов. Поэтому, очень популярными становятся медицинские информационные системы, предлагающие сбор, хранение и обработку информации, связанную с пациентами клиники.

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.[[1]](#footnote-1)

## Обзор существующих медицинских информационных систем

## Предпосылки разработки модуля "Статистика"

## Обзор средств разработки

### Язык 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 кандидатов: C#, Ruby, Python, Java, PHP, Javascript.

C# -- ООП язык программирования разработанный в компании Microsoft. C# имеет статическую, строгую типизацию. К плюсом данного языка программирования можно отнести встроенную поддержку .NET фреймворка позволяющего создавать кроссплатформенные приложения (не зависящие от операционной системы). К недостаткам можно отнести слабую поддержку ОС отличных от Windows,

### Библиотеки элементов интерфейса

При разработке приложений мы сталкиваемся 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 заботится о размещении и компоновкой 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 что бы 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 использование 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 Cи 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.

### Система управления базами данных

## Выбор средств разработки

## Структура базы данных МИС САМСОН

# Практическая часть квалификационной работы

## 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.

Назначение системы: Данный проект создается для предпринимателей, 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 А.

## 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 (скоро это изменится, надеюсь)

./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 дополнении 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.

## Интерфейс и функциональные возможности

# Заключение

# Список литературы

Bodnar, J. (29 05 2016 г.). *Ruby Qt tutorial*. Получено из ZetCode, tutorials for programmers: http://zetcode.com/gui/rubyqt/

Gerasimov. (29 05 2016 г.). *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*. Получено из САМСОН - Официальный сайт проекта: http://samson-rus.com/

*MySQL.RU .:. Одобрено лучшими российскими программистами*. (29 05 2016 г.). Получено из MySQL.RU: http://www.mysql.ru/docs/man/

*Ruby — Викиучебник*. (05 29 2016 г.). Получено из Викиучебник: https://ru.wikibooks.org/wiki/Ruby

*Rusrails: Интерфейс запросов Active Record*. (29 5 2016 г.). Получено из Rusrails: Ruby on Rails по-русски: http://rusrails.ru/active-record-query-interface

*Разработка/Руководства/Введение в Qt4 на Ruby — KDE TechBase*. (29 05 2016 г.). Получено из KDE TechBase: https://techbase.kde.org/Development/Tutorials/Qt4\_Ruby\_Tutorial/ru

Фултон, Х. (2014). *Программирование на языке Ruby.* Litres.

Шлее, М. (2012). *Qt 4.8. Профессиональное программирование на C++.* БХВ-Петербург.

# Приложение А. Техническое задание

НЕЗАБУДЬ ВОТКНУТЬ ТЗ СЮДА

1. https://ru.wikipedia.org/wiki/Медицинская\_информационная\_система [↑](#footnote-ref-1)