

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра информатики

УТВЕРЖДАЮ

Заведующий кафедрой

 А.И. Рубан
подпись инициалы, фамилия

«21» июня 2016 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

231000.62 – Программная инженерия

код – наименование направления

Автоматизированное рабочее место врача стоматолога.

Руководитель


подпись, дата должность, ученая степень

М.И. Никитина

инициалы, фамилия

Выпускник


подпись, дата

Д.И. Хомусяк

инициалы, фамилия

Нормоконтролер


подпись, дата

О.А. Антамошкин

инициалы, фамилия

Красноярск 2016

Студенту Хомусяку Денису Игоревичу

Группа: КИ12-18 Направление подготовки: 23100.62,
Программная инженерия.

Тема выпускной квалификационной работы: «Автоматизированное рабочее место врача стоматолога».

Утверждена приказом по университету № 6145/с от 10.05.2016.

Руководитель ВКР: М. И. Никитина кандидат технических наук.
Исходные данные для ВКР: автоматизация деятельности врача стоматолога.

Перечень разделов ВКР:

- введение;
- описание предметной области;
- обзор существующих систем автоматизации рабочего места врача стоматолога;
- техническое задание;
- описание и обоснование выбора инструмента для разработки программного приложения;
- моделирование и проектирование программного приложения;
- заключение.

Перечень графического или иллюстративного материала с указанием основных чертежей, плакатов, слайдов: презентационные слайды PowerPoint.

Руководитель ВКР



М. И. Никитина

(подпись)

Задание принял к исполнению



Д.И. Хомусяк

(подпись)

«__» _____ 2016 г.

ОТЗЫВ РУКОВОДИТЕЛЯ

на выпускную квалификационную работу студента группы КИ12-18Б

Хомусяка Дениса Игоревича

на тему: «Автоматизированное рабочее место врача стоматолога»

В задачи выпускной квалификационной работы входили проектирование и реализация программного приложения, позволяющего автоматизировать рабочее место врача стоматолога. Программное приложение является модулем информационной системы «Стационар», эксплуатируемой в здравоохранении Красноярского края.

Хомусяк Д.И. изучил используемую в здравоохранении края Единую систему ведения справочников и классификаторов (ЕСВС), медицинскую информационную систему (МИС) «Стационар», технологию разработки приложений в трехзвенной архитектуре. Выполнил проектирование и разработку программного приложения, состоящего из двух частей: веб-сервиса, предоставляющего данные из справочников, и клиентского приложения на стороне МИС «Стационар», являющегося автоматизированным рабочим местом врача стоматолога.

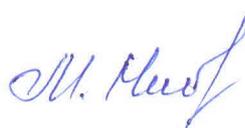
Выполненная работа имеет практическое применение и рекомендована к внедрению в медицинских учреждениях Красноярского края.

Хомусяк Д.И. выполнил задание на выпускную квалификационную работу полностью, показал себя квалифицированным специалистом, способным самостоятельно решать сложные задачи.

Считаю, что выпускная квалификационная работа Хомусяка Д.И. заслуживает оценки "отлично", а ее автор заслуживает присвоения квалификации «бакалавр» по направлению «Программная инженерия».

Доцент кафедры «Информатика»

ФГАОУ ВПО «СФУ», к.т.н.



М.И. Никитина

СОДЕРЖАНИЕ

1	ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1	ВВЕДЕНИЕ В ПРЕДМЕТНУЮ ОБЛАСТЬ.....	10
1.2	Требования предъявляемые к системам, которые автоматизируют работу врача стоматолога.....	11
1.3	Обзор существующих решений	11
2	ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	14
2.1	НАИМЕНОВАНИЕ И ОСНОВАНИЕ ДЛЯ СОЗДАНИЯ.....	14
2.1.1	Полное наименование.....	14
2.1.2	Основание для создания	14
2.2.	Назначение и цели создания системы.....	14
2.2.1	Назначение разрабатываемой системы.....	14
2.2.2	Цели создания системы	15
2.3.	Характеристика объекта информатизации	15
2.3.1	Краткие сведения об объекте информатизации.....	15
2.4.	Требования к системе	16
2.5.	Требования к структуре и функционированию системы.....	17
2.5.1	Перечень модулей, их назначение и основные характеристики.....	17
2.5.2	Требования к характеристикам взаимосвязей создаваемой системы со смежными системами	17
2.6.	Требования к модулям системы	17
2.6.1	Требования к модулю «Список пациентов».....	17
2.6.1.1	Функция «Список пациентов на прием»	18
2.6.1.2	Функция «Поиск пациентов по указанной дате».....	18
2.6.2	Требования к модулю «Подробная информация о приеме».....	18
2.6.2.1	Функция «Информация о пациенте».....	19

2.6.2.2 Функция «Список ранее зарегистрированных случаев обслуживания»	19
2.6.2.3 Функция «Список посещений»	19
2.6.2.4 Функция «Добавление результатов осмотра»	19
2.6.2.5 Функция «Создание нового случая обслуживания»	20
2.6.2.6 Функция «Внесение изменений в случай обслуживания»	20
2.6.2.7 Функция «Создание новых посещений к врачу»	21
2.7.Требование к архитектуре системы	22
2.8.Требования к интерфейсу системы	22
2.9.Требования к надежности системы	23
2.10.Требования к техническому обеспечению	24
3 ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА ИНСТРУМЕНТА ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	24
3.1 Требования к инструментам серверной части приложения	24
3.1.1 Система управления базами данных	24
3.1.2 Языки программирования	29
3.1.3 Веб– сервер	32
3.2 Требования к клиентской части приложения.	34
МОДЕЛИРОВАНИЕ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	34
4.1 Архитектура проектируемого приложения	34
4.3 Построение диаграммы вариантов использования.	36
4.3.1 Диаграммы вариантов использования	36
4.3.2 Диаграмма вариантов использования для модуля «Список пациентов»	36
4.3.3 Диаграмма вариантов использования для модуля «Подробная информация о приеме»	38
4.4 Описание информационной базы	39
4.4.1 Описание структуры базы данных части «Список пациентов»	39

4.4.2 Описание структуры базы данных части «Подробная информация о приеме»	40
4.5 Разработка логической структуры системы	44
4.5.1 Диаграмма классов модуля «Список пациентов»	45
4.5.2 Диаграмма классов модуля «Подробной информации о приеме».....	46
4.6. Вывод.....	48
5 ОПИСАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.	49
5.1 Вход в систему.....	49
5.2 Модуль «Список пациентов»	50
5.3 Модуль «Подробная информация о приеме»	51
ЗАКЛЮЧЕНИЕ	54
СПИСОК СОКРАЩЕНИЙ.....	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	56

РЕФЕРАТ

Выпускная квалификационная работа по теме «Автоматизированное рабочее место врача стоматолога» содержит 53 страницы текстового документа, 7 использованных источников, 20 иллюстраций.

МЕДИЦИНА, ПОЛИКЛИНИКА, СТОМАТОЛОГИЯ, ПАЦИЕНТ, ТЕХНИЧЕСКОЕ ЗАДАНИЕ, ТРЕБОВАНИЯ, ПРОЕКТИРОВАНИЕ, МОДЕЛИРОВАНИЕ, ОТЧЕТ, АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО, ПРОГРАММНОЕ ПРИЛОЖЕНИЕ.

Целью работы является разработка программного обеспечения для автоматизации работы врача стоматолога.

Исходя из поставленной цели, в работе приведены: анализ работы врача стоматолога, обзор существующих систем автоматизации работы врача стоматолога, анализ технологий для автоматизации работы стоматолога, техническое задание, моделирование бизнес-процессов, диаграммы вариантов использования, описание базы данных, диаграммы классов.

В специальной части работы приведено описание разработанного программного обеспечения.

В результате разработано программное обеспечение, автоматизирующее работу врача стоматолога, которое написано с использованием технологий Yii PHP Framework, ExtJS Framework.

Введение

Современные медицинские организации производят и накапливают огромные объемы данных. От того, насколько эффективно эта информация используется врачами, руководителями, управляющими органами, зависит качество медицинской помощи, общий уровень жизни населения.

Говоря о региональной информатизации здравоохранения, крайне важно оценить состояние текущих крупных региональных проектов, ресурсный потенциал, необходимый для их реализации, состояние телекоммуникаций. При этом необходимо иметь в виду, что сегодня акцент постепенно смещается в сторону реализации проектов информатизации медицинской деятельности. То есть приоритеты начали меняться в сторону таких направлений, как автоматизация рабочего места врача, создание единой электронной медицинской карты, распространение интеллектуальных систем поддержки принятия административных решений, применение и широкое распространение новейших медицинских технологий, развитие телемедицины [1].

Наверное, нет такой области деятельности человека, где не приходилось бы иметь дело с заполнением документов и бланков. Стоматология - не исключение. При этом вся информация бывает чрезвычайно важна: например, результаты первичного осмотра пациента, заносимые в его личную карточку, на поиск которых при последующем приеме может тратить свое время врач[2].

Автоматизация оформления медицинских документов в лечебно-профилактических учреждениях является актуальной задачей. Оформлением пациента занимается работник регистратуры, который получает от пациента всю необходимую информацию. Эту информацию (такую как фамилия, имя, отчество и др.) использует врач стоматолог.

Кроме того, трудоемкой задачей является поиск истории случаев обслуживания, истории посещений пациента с их результатами. Амбулаторную карту пациента необходимо вести в электронном виде. Таким образом, разработка программного обеспечения для автоматизации деятельности врача стоматолога поликлиники является актуальной.

Цели и задачи

Целью данной работы является разработка и реализация программного обеспечения для автоматизации рабочего места врача стоматолога поликлиники. Программное обеспечение должно иметь функции ввода, накопления и предоставления всей необходимой информации о пациентах, записанных на прием к врачу, а так же информацию о приеме пациента. Разработка автоматизированного рабочего места врача стоматолога поликлиники очень важна, так как разработанный программный продукт упростит работу стоматологам поликлиники в части формирования медицинских документов и поиска сведений о случаях лечения пациента.

Достижение поставленной цели предполагает решение следующих задач:

- изучение бизнес-процессов стоматологической поликлиники;
- обзор существующих медицинских информационных систем для автоматизации деятельности врача стоматолога;
- формирование требований к программному приложению;
- разработка и реализация базы данных программного приложения;
- разработка и реализация программного приложения в клиент-серверной технологии;
- тестирование программного приложения;
- разработка сопроводительной документации.

1 Описание предметной области

1.1 Введение в предметную область

Бизнес-процессы стоматологической поликлиники достаточно сложны и включают работу регистратуры, врача стоматолога, лаборатории, заведующего и другие. В данной работе рассматривается деятельность врача стоматолога. В функции врача стоматолога входит:

- Проводить амбулаторный прием по графику, утвержденному администрацией поликлиники, регулируя поток посетителей путем рационального распределения повторных больных;
- Проводить профилактические осмотры и санацию полости рта Среди больных, находящихся на диспансерном наблюдении в поликлинике;
- Направлять больных, при наличии показаний, на дополнительные виды исследования (лабораторные, рентгенологические, функциональные и т. д.).

Обязательным элементом работы врача стоматолога является запись результатов осмотра пациента.

Для использования медицинской и общей документации в практической и научной работе в поликлинике должен быть создан медицинский архив. В медицинском архиве производятся прием, учет, классификация, хранение и выдача документальных материалов как персоналу поликлиники, так и больным в случае изменения их места жительства или запросов различных организаций. В архиве хранятся вся документация о работе поликлиники, в том числе приказы по поликлинике, годовые отчеты, личные дела уволенных, а также вся медицинская документация (в основном медицинские карты амбулаторных больных). Медицинские карты амбулаторных больных поступают в архив из регистратуры в том случае, если больной в течение 3 лет не обращался в поликлинику. Кроме того, в действующий архив можно передавать

«утолщенные» и принявшие небрежный вид медицинские карты, после того как взамен их будут заведены новые с копированием из старых карт уточненных диагнозов и краткого эпикриза.

Значительное место в работе стоматолога занимает оформление медицинских документов: справок, направлений и др. .

1.2 Требования предъявляемые к системам, которые автоматизируют работу врача стоматолога

Основными функциями автоматизированного рабочего места врача стоматолога поликлиники являются:

- Возможность работы с очередью пациентов, записанных к врачу;
- Возможность просмотра истории случаев обслуживания записанного пациента;
- Возможность просмотра информации об предыдущих посещениях пациента;
- Возможность повторной записи пациента на прием к врачу;
- Возможность записи результатов осмотра пациента.

1.3 Обзор существующих решений

В настоящее время существуют автоматизированные информационные системы, осуществляющие решение задачи автоматизации деятельности врача стоматолога поликлиники.

Рассмотрим некоторые разработки, информация о которых доступна в Интернете.

1.3.1 Информационная система «IDent» АРМ «Стоматологии»

Компьютерная программа для управления стоматологической

клиникой, предназначена для оптимизации работы персонала, сокращения временных и денежных издержек на обслуживание клиентов. Разработана компанией «Мой софт» в 2010 году [3].

В данной системе реализованы следующие функции:

- Составление расписания;
- Добавление результатов осмотра с оказанными услугами;
- Добавление пациентов;
- Формирование отчетных форм.

1.3.2 Информационная система «Dentist+»

Удобное веб-приложение, созданное для руководителей стоматологических клиник, врачей-стоматологов и администраторов. Программа позволяет работать с расписанием и базой пациентов, вносить все данные о лечении, рассчитывать прибыль от каждого сотрудника, анализировать работу всей клиники[4].

В данном приложении реализованы следующие возможности:

- Создание графика работы, записи пациентов;
- Создана система напоминаний о задачах;
- История болезни, зубная формула, данные пациента;
- Внедрена бухгалтерия;
- Складской учет;

1.3.3 Информационная система «iStom»

Эта программа для управления стоматологической клиникой или сетью клиник. Это современный и мощный комплекс охватывающий все основные бизнес процессы в стоматологии при масштабируемости от одного кресла до сети клиник. Разработано в ООО «ПРОМЕД» Башкортостан г. Уфа[5].

- Предоставляет возможность работы с расписанием;
- Выписка направления и запись пациента;
- Вести прием пациентов по списку записанных;
- Контролировать очередь по профилю;
- Выполнять поиск по регистру прикрепленного населения;
- Просматривать журнал направлений;
- Работать с Электронной медицинской картой пациента;
- Работать с картами диспансерных больных;
- Получать данные о заключениях ВК;
- Выполнять поиск участков по адресу;
- Встроена система SMS оповещения.

1.3.4 Выводы

Все рассмотренные системы решают сходные задачи с поставленной в настоящей работе, однако интеграция их с информационной системой «Стационар», используемой в Красноярском крае для автоматизации деятельности районных больниц, частных медицинских организаций невозможна вследствие несовместимости платформ и схем данных. Информационная система «Стационар» разрабатывается для работы в web-технологии с размещением центральной базы данных на серверных мощностях ККМИАЦ. Такая архитектура системы позволяет сократить расходы небольших медицинских организаций на сопровождение. Кроме того, надо сказать, системы, предоставляемые сторонними разработчиками, не являются гибкими и адаптируемыми к региональным регламентам информационного обмена. Например, при изменении форм талонов или отчетов, либо при каких-нибудь других изменениях в процессе работы стоматологии требуется значительное время на согласование и доработку.

Поэтому было решено разрабатывать модуль АРМ «Врача стоматолога» для информационной системы «Стационар».

2 Техническое задание

2.1 Наименование и основание для создания

2.1.1 Полное наименование

Полное наименование: Автоматизированное рабочее место врача стоматолога.

2.1.2 Основание для создания

Разработка ведется на основании «Распоряжения об утверждении тем выпускных квалификационных работ» утвержденного заведующим кафедрой «Информатика». Тема выпускной квалификационной работы: «АРМ врача стоматолога».

2.2. Назначение и цели создания системы

2.2.1 Назначение разрабатываемой системы

АРМ врача стоматолога предназначен для автоматизации деятельности врача стоматолога поликлиники, в частности следующих рабочих процессов:

- Управление очередью к врачу;
- Просмотра случаев обращения в поликлинику;
- Просмотра истории посещений с их результатами;
- Изменение диагноза случая;
- Добавление результатов осмотра посещения.

АРМ «Стоматолога» планируется к использованию в медицинских

учреждениях районного уровня в Красноярском крае.

2.2.2 Цели создания системы

Основными целями создания АРМ «Стоматолога» является:

- замещение существующих информационных систем, которые морально устарели и не отвечают современным требованиям, предъявляемым к информационным системам;
- повышение эффективности исполнения процессов, перечисленных выше, путем сокращения непроизводительных и дублирующих операций, операций, выполняемых «вручную», оптимизация информационного взаимодействия участников процессов;
- повышение качества оказания медицинской помощи на основе совершенствования информационно–технического обеспечения деятельности врача стоматолога поликлиники.

2.3. Характеристика объекта информатизации

2.3.1 Краткие сведения об объекте информатизации

Объектом информатизации является деятельность врача стоматолога в поликлиник .

Данное подразделение поликлиники является объектом автоматизации.

В функции врача стоматолога входит:

- организация предварительной и неотложной записи больных на прием к врачу при приеме пациента;
- проводить профилактические осмотры;
- направлять больных, при наличии показаний, на дополнительные виды исследования;

2.4. Требования к системе

Разрабатываемое программное приложение должно учитывать следующие основные принципы:

- **Открытость.** Система должна использовать общедоступные и специфицированные решения, протоколы и интерфейсы, что будет обеспечивать интеграцию создаваемых решений, как между собой, так и с внешними информационными системами.
- **Модульность.** Система должна быть построена с использованием модульной архитектуры, подразумевающей реализацию основных функций в качестве отдельных модулей, обеспечивающих возможность их независимой модификации. Сбой в работе одного из модулей не должен приводить к полному прекращению функционирования программного приложения в целом.
- **Управляемость и конфигурирование.** Механизмы управления должны обеспечивать управление создаваемой Системой на всех уровнях ее архитектуры: на уровне инфраструктуры, на функциональном уровне, на уровне представления данных.
- **Системность.** Все взаимосвязанные компоненты создаваемого программного приложения должны использовать единую методологию и отвечать единым принципам взаимодействия, надежности и управления.
- **Единство графического представления.** При проектировании и разработке пользовательских интерфейсов должны использоваться общие принципы графического представления информации и организации доступа пользователей к функциональным возможностям и сервисам создаваемого программного приложения.

2.5. Требования к структуре и функционированию системы

2.5.1 Перечень модулей, их назначение и основные характеристики

В состав АРМ «Стоматолога» должны входить следующие модули:

- модуль «Список пациентов»;
- модуль «Подробная информация о приеме»;

Модуль «Список пациентов» на прием предназначен для просмотра очереди пациентов, которые записаны к врачу, а так же он включает поиск записанных пациентов по дате.

Модуль «Подробная информация о приеме» включает в себя всю информацию, которая нужна для просмотра информации о пациенте, а так же более подробной информации приема и внесение результатов осмотра пациента.

2.5.2 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами

В АРМ «Стоматолога» должны использоваться единые справочники и классификаторы системы здравоохранения Красноярского края.

Программное приложение так же связано с системой АРМ «Регистратура» т.к. запись пациентов осуществляется через регистратуру.

2.6. Требования к модулям системы

2.6.1 Требования к модулю «Список пациентов»

Модуль «Список пациентов» на прием предназначен для просмотра очереди пациентов, которые записаны к врачу, а так же он включает поиск записанных пациентов по дате.

Модуль список пациентов должен предоставить пользователю следующие возможности:

- список пациентов на прием;
- поиск пациентов по указанной дате;
- выбор пациента;

2.6.1.1 Функция «Список пациентов на прием»

Данная функция отображает список пациентов на прием к врачу на текущий день.

2.6.1.2 Функция «Поиск пациентов по указанной дате»

При реализации данной функции должна быть предусмотрена возможность указать следующие данные, по которым будет выполняться поиск:

- Дата посещения.

2.6.2 Требования к модулю «Подробная информация о приеме»

Модуль «Подробная информация» о приеме включает в себя всю информацию, которая нужна для просмотра информации о пациенте, а так же более подробной информации приема и внесение результатов осмотра пациента.

Модуль должен предоставить пользователю следующие возможности:

- Информация о пациенте;
- Список ранее зарегистрированных случаев обслуживания;
- Список посещений;
- Добавление результатов осмотра;

- Создание нового случая обслуживания;
- Внесение изменений в случаи обслуживания;
- Создание новых посещений к врачу.

2.6.2.1 Функция «Информация о пациенте»

Функция отображает подробную информацию о пациенте, такую как: ФИО, дату рождения и пол.

2.6.2.2 Функция «Список ранее зарегистрированных случаев обслуживания»

Функция отображает ранее зарегистрированные случаи обслуживания пациента в поликлиники с подробной информацией о случае АПП.

2.6.2.3 Функция «Список посещений»

Функция «список посещений» зависит от выбора случая обслуживания, т.к. случай обслуживания имеет несколько посещений, которые хранят остальную информацию, такую, как результаты осмотра, врач к которому был записал пациент по этому случаю и др.

2.6.2.4 Функция «Добавление результатов осмотра»

Для добавления результатов осмотра, относящихся к посещениям необходимо заполнить следующие поля:

- Результат осмотра.

Поле «Результат осмотра» является текстовым.

При нажатии кнопки «Закрыть» модальное окно с формой закроется.

После нажатия кнопки «Добавить» в форме добавления результатов осмотра информация должна сохраняться в БД.

2.6.2.5 Функция «Создание нового случая обслуживания»

Для добавления случая обслуживания необходимо заполнить поля:

- Дата и время начала случая;
- Место(тип) посещения;
- Цель случая АПП обслуживания;
- Срок беременности;
- Номер беременности;

Поле «Место(тип) посещения» является обязательным и заполняется с использованием справочника «Места амбулаторно-поликлинического приема» единой системы ведения справочников здравоохранения Красноярского края (ЕСВС).

Поле «Цель случая АПП обслуживания» является обязательным и заполняется с использованием справочника цели амбулаторно-поликлинических посещений ЕСВС.

При выделении поля с флагом «Беременность» - появится новая вкладка в форме с полями об беременности.

При выделении поля с флагом «Создать текущее посещение» - будет создано новое посещение по созданному случаю обслуживания.

При нажатии кнопки «Создать» создастся случай обслуживания с введенной информацией в форме.

2.6.2.6 Функция «Внесение изменений в случай обслуживания»

Для внесения изменений в случай обслуживания необходимо заполнить поля:

- Источник финансирования;

- Дата и время начала случая;
- Дата и время окончания случая;
- Диагноз(МКБ10);
- Исход;
- Результат.

Поле «Источник финансирования» заполняется с использованием справочника источников финансирования ЕСВС.

Поле «Диагноз(МКБ10)» заполняется с использованием справочника МКБ-10 ЕСВС.

Поле «Исход» заполняется с использованием справочника исходов заболевания ЕСВС.

Поле «Результат» заполняется с использованием справочника результатов обслуживания ЕСВС.

При нажатии кнопки «Сохранить» будет отправлен запрос на сервер с введенной информацией в форме.

2.6.2.7 Функция «Создание новых посещений к врачу»

Для создания новых посещений нужно выбрать случай обслуживания и необходимо ввести следующие данные:

- Дата и время посещения;
- Вид приема;
- Врач;
- Заключительное посещение.

Поле «Вид приема» заполняется с использованием справочника видов амбулаторно-поликлинического приема ЕСВС.

Поле «Врач» заполняется с использованием регистра медицинских работников учреждения.

2.7.Требование к архитектуре системы

Разрабатываемое программное приложение должно учитывать следующие основные принципы:

- **Модульность**

Программное приложение должно быть построено с использованием модульной архитектуры, подразумевающей реализацию основных функций в качестве отдельных модулей, обеспечивающих возможность их независимой модификации. Сбой в работе одного из модулей не должен приводить к полному прекращению функционирования программного приложения в целом.

- **Масштабируемость**

Архитектура программного приложения должна позволять увеличивать производительность подсистем, объемы хранимой и обрабатываемой информации без длительной остановки работы и значительной модификации программного кода программного приложения.

- **Управляемость и конфигурирование**

Механизмы управления должны обеспечивать управление создаваемой Системой на всех уровнях ее архитектуры: на уровне инфраструктуры, на функциональном уровне, на уровне представления данных.

- **Системность**

Все взаимосвязанные компоненты создаваемого программного приложения должны использовать единую методологию и отвечать единым принципам взаимодействия, надежности и управления.

2.8.Требования к интерфейсу системы

Интерфейс пользователя должен удовлетворять следующим требованиям:

- Интерфейс пользователя должен быть выполнен в стиле системы «Стационар»;
- При составлении расписания интерфейс должен предусматривать копирование и вставку путем выделения и переноса интервала расписания на другой день;
- Интерфейс пользователя должен информировать пользователя о вводе не корректных данных и не заполненных полях на формах ввода;
- Данные из справочников должны выбираться из выпадающих списков;
- Редактирование записи должно выполняться двумя способами: нажатием на кнопку «Изменить» панели инструментов или двойным нажатием на запись в списке;
- Должна быть реализована возможность отправки формы на сервер по нажатию на клавишу «Enter»;

2.9.Требования к надежности системы

Программное приложение должно сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска веб– приложения в браузере;
- при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

2.10. Требования к техническому обеспечению

Техническое обеспечение программного приложения должно максимально и наиболее эффективным образом использовать существующие в медицинских учреждениях технические средства.

В состав комплекса должны следующие технические средства:

- Серверы БД;
- Веб сервер;
- ПК пользователей;

3 Описание и обоснование выбора инструмента для разработки программного обеспечения

3.1 Требования к инструментам серверной части приложения

В этой главе будут сформулированы требования к технологической платформе приложения (программной составляющей), и в соответствующих подразделах будет приведен обзор существующих решений и выбор наиболее оптимального варианта.

3.1.1 Система управления базами данных

Система управления базой данных (СУБД) представляет собой совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Для разрабатываемой системой надежность и производительность СУБД является критическим моментом, поскольку приложение предполагает наличие большого количества записей в БД, а также большое количество пользователей, чьи действия так или иначе будут связаны с манипулированием информации из БД.

Основные требования к СУБД:

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных;
- после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными);
- поддержка реляционной модели хранения данных;
- поддержка распределенного режима;
- поддержка клиент–серверной архитектуры взаимодействия;
- наличие внешних утилит для управления СУБД;
- низкая стоимость.

3.1.1.1 Обзор современных СУБД

В мире существует множество СУБД, обзор их всех не входит в рамки этой работы. Рассмотрены будут наиболее популярные СУБД, которые имеют многолетнюю историю разработки и поддержки.

3.1.1.1 Oracle Database 11g Standard Edition

Oracle Database 11g является единственной СУБД, предназначенной для распределенных вычислительных сред (Grid). Выпуская Oracle Database 11g, Oracle предоставляет клиентам возможность управлять корпоративной информацией проще, чем когда-либо прежде, лучше понимать свой бизнес и быстрее внедрять инновации. Oracle Database 11g обеспечивает высочайшие уровни производительности, масштабируемости, доступности, безопасности и удобства управления в сетях распределенных вычислений на базе стандартных серверов и систем хранения. СУБД Oracle Database 11g предназначена для эффективного развертывания на базе различных типов оборудования, от небольших серверов до мощных симметричных многопроцессорных серверных систем и кластеров любых масштабов. Этот продукт предлагает функции автоматизации управления, обеспечивая удобство и эффективность эксплуатации. Уникальные возможности управления всеми данными предприятия – от традиционной бизнес-информации до XML и трехмерных пространственных данных – делает Oracle Database 11g идеальным решением для эффективной реализации OLTP приложений, хранилищ данных и систем управления контентом.

Стоимость решения — от 100 000 рублей.

3.1.1.1.2 Firebird

Firebird (FirebirdSQL) — компактная, кроссплатформенная, свободная система управления базами данных (СУБД), работающая на GNU/Linux, Microsoft Windows и разнообразных Unix платформах.

В качестве преимуществ Firebird можно отметить многоверсионную архитектуру, обеспечивающую параллельную обработку оперативных и аналитических запросов (это возможно потому, что читающие пользователи не блокируют пишущих), компактность (дистрибутив 5Mb), высокую

эффективность и мощную языковую поддержку для хранимых процедур и триггеров.

Среди недостатков: отсутствие кеша результатов запросов, полнотекстовых индексов.

Стоимость — бесплатно.

3.1.1.1.3 MySQL

MySQL — свободная система управления базами данных (СУБД). MySQL является собственностью компании Sun Microsystems, осуществляющей разработку и поддержку приложения. Распространяется под GNU General Public License и под собственной коммерческой лицензией, на выбор. Помимо этого разработчики создают функциональность по заказу лицензионных пользователей, именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Стоимость простой лицензии MySQL Enterprise на год — 600 у.е., или 18 000 рублей.

3.1.1.2 Выбор СУБД

Решение Oracle является неприемлемым с точки зрения стоимости лицензии, а также необходимой аппаратной части для работы этой СУБД.

СУБД FireBird по сравнению с MySQL похожи по функциональности, однако для использования MySQL в коммерческих целях необходима покупка лицензии, что неприемлемо с точки зрения стоимости лицензии. Также MySQL обладает намного меньшим сообществом в Интернет, что означает увеличение времени поиска решений для возможных проблем.

СУБД FireBird в отличие от MySQL позволяет выполнять резервное копирование базы данных «на лету» не останавливая работу пользователей с базой данных.

Основные характеристики Firebird:

- Полная поддержка хранимых процедур и триггеров.
- Транзакции, полностью совместимые с концепцией ACID.
- Ссылочная целостность.
- Версионная архитектура.
- Очень небольшой размер.
- Мощный внутренний язык для написания хранимых процедур и триггеров (PSQL).
- Поддержка внешних пользовательских функций (UDF).
- Firebird практически не требует работы системного администратора или позволяет свести ее к минимуму.
- Почти не требует настройки – использовать СУБД можно сразу же после ее установки.
- Огромное интернет– сообщество пользователей и разработчиков, множество мест, где вы можете получить быструю и бесплатную помощь.
- Десятки специализированных приложений от сторонних разработчиков, включая средства администрирования, репликации, и так далее.
- Безопасная запись данных (careful write) – быстрое восстановление после сбоев, отсутствие необходимости в журналировании транзакций!
- Большое количество средств доступа к базе данных: native/API, драйверы dbExpress, ODBC, OLEDB, .Net provider, JDBC– драйвер, модули для Python, PHP, Perl, и так далее.
- Поддержка большинства распространенных операционных систем, включая Windows, Linux, Solaris, MacOS.
- Инкрементные бэкапы
- Билды для 32– и 64– разрядных ОС
- Полная реализация курсоров в PSQL

- Таблицы мониторинга
- Триггеры на подключение и транзакции
- Временные таблицы
- TraceAPI – узнайте, что делается на сервере

Так же выбор СУБД был сделан исходя из того, что система «Стационар», для которой разрабатывается модуль АРМ «Стоматолога» работает на СУБД Firebird. Поэтому переход на какую– либо другую СУБД невозможен.

3.1.2 Языки программирования

Приложение состоит из двух равноценных частей: серверной и клиентской. Возможно использовать один и тот же язык программирования во всем приложении, однако это является самым не оптимальным вариантом.

Для серверной части будет использоваться язык PHP в частности Yii PHP Framework[7] версии 1.1. PHP (англ. PHP: Hypertext Preprocessor — «PHP: препроцессор гипертекста») — скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб–приложений. В настоящее время поддерживается подавляющим большинством хостинг– провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб–сайтов.

Для написания запросов к базе данных будет использоваться SQL в частности Firebird SQL. SQL — формальный не процедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД). SQL основывается на исчислении кортежей.

Для построения пользовательского интерфейса и бизнес– логики будет использоваться JavaScript в частности библиотека ExtJS версии 4.2 [6].

JavaScript – объектно-ориентированный скриптовый язык программирования. JavaScript поддерживается всеми существующими браузерами и является стандартом де-факто для современных интерактивных веб– приложений.

3.1.2.1 Описание Yii PHP Framework

Yii - это высокоэффективный, основанный на компонентной структуре PHP - фреймворк для быстрой разработки крупных веб– приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб– разработки. Название Yii (произносится как Yee или [ji:]) означает простой (easy), эффективный (efficient) и расширяемый (extensible)[7].

3.1.2.1.1 Назначение Yii PHP Framework

Yii - это фреймворк для веб - программирования общего назначения, который может быть использован для разработки практически любых веб– приложений. Благодаря своей легковесности и наличию продвинутых средств кэширования, Yii особенно подходит для разработки приложений с большим потоком трафика, таких как порталы, форумы, системы управления контентом (CMS), системы электронной коммерции и др.

3.1.2.1.2 Возможности Yii PHP Framework

Подобно большинству других PHP - фреймворков, Yii - это MVC фреймворк.

Превосходство Yii над другими фреймворками заключается в эффективности, широких возможностях и качественной документации. Yii изначально спроектирован очень тщательно для соответствия всем требованиям при разработке серьезных веб– приложений. Yii не является ни

побочным продуктом какого– либо проекта, ни сборкой сторонних решений. Он является результатом большого опыта авторов в разработке веб–приложений, а также их исследований наиболее популярных веб–фреймворков и приложений.

Главным плюсом Yii является отличная поддержка ООП, скорость работы и, конечно же, тех поддержка от разработчиков. Фреймворк включает в себя большой набор библиотек, которые помогут вам в создании полноценного веб приложения отвечающего всем современным стандартам (интегрированное использование Ajax, встроенная поддержка интернационализации приложения, простой инструмент работы с базой данных). Тот код который занимал бы у вас 100 строчек чистого php кода может быть сокращен до десяти благодаря встроенным методам фреймворка. Yii является бесплатным программным обеспечением и распространяется под лицензией «new BSD».

3.1.2.2 Описание ExtJS Framework

Ext JS представляет собой библиотеку JavaScript для разработки веб-приложений и пользовательских интерфейсов. И прежде всего Ext JS предназначен для создания RIA-приложений, то есть таких интернет-приложений, которые отличаются богатым и насыщенным интерфейсом.

Ext JS помогает организовывать и управлять элементами на веб-странице и проводить более точную и эффективную их настройку. Причем данный фреймворк позволяет не только эффективно работать с элементами управления, но и с некоторыми графическими элементами, например, с диаграммами.

Возможности, предоставляемые данным фреймворком, позволяют применять к веб– приложениям шаблон MVC, который позволяет отделить логику приложения, и его данные от визуальной части.

Еще одной важной чертой фреймворка является кроссбраузерность. В настоящее время Ext JS поддерживается следующими версиями браузеров:

- Internet Explorer 6+
- Firefox 3.6+ (PC, Mac)
- Safari 4+
- Chrome 10+
- Opera 11+ (PC, Mac)

Кроме того, Ext JS постоянно развивается и совершенствуется.

3.1.3 Веб– сервер

Веб– сервер — это сервер, принимающий HTTP– запросы от клиентов, обычно веб– браузеров, и выдающий им HTTP– ответы, обычно вместе с HTML– страницей, изображением, файлом, медиа– потоком или другими данными. Веб– серверы — основа Всемирной паутины.

Основным требованием для веб– сервера является обеспечение совместной работы всех предыдущих выбранных технологических решений.

3.1.3.1 Обзор современных веб– серверов

В мире существует множество веб-серверов, обзор их всех не входит в рамки этой работы. Рассмотрены будут наиболее популярные решения, которые имеют многолетнюю историю разработки и поддержки. Рассмотрим некоторые из веб– серверов:

- IIS;
- Apache HTTP– сервер.

3.1.3.1.1 Обзор веб– сервера IIS

IIS (Internet Information Services, до версии 5.1 - Internet Information Server) - набор серверов для нескольких служб Интернета от компании Майкрософт. IIS распространяется с операционными системами семейства Windows NT.

Основным компонентом IIS является веб– сервер, который позволяет размещать в Интернете сайты. IIS поддерживает протоколы HTTP,HTTPS, FTP, POP3, SMTP, NNTP.

3.1.3.1.1 Обзор веб– сервера Apache HTTP– сервер

Apache HTTP– сервер - свободный веб-сервер. По статистике на май 2009 года используется в 46% всех веб– серверов, что делает его самым популярным веб– сервером. Версия 2.2.x обладает проверенной годами и миллионами пользователей стабильностью и надежностью. Сервер имеет большое количество модулей для работы со многими серверными технологиями.

3.1.3.2 Выбор веб-сервера

IIS - разработка Microsoft и легко интегрируется в Windows, но не возможно интеграция с ОС основанных на ядре Linux,. Кроме того, для качественной настройки IIS нужно иметь определённый опыт и знания.

Решающий фактор влияющий на выбор веб– сервера это совместимость с операционными системами, которые основаны на ядре Linux, так как выбрана ОС Ubuntu. Исходя из этого веб– сервер IIS не может быть выбран. Поэтому для работы в качестве веб– сервера выбирается Apache HTTP сервер версии 2.2.

3.2 Требования к клиентской части приложения

Клиентская часть должна состоять из следующих компонент:

- Программное обеспечение:
- Одна из операционных систем:
- Windows XP и выше,
- Любой дистрибутив Linux с поддержкой графического режима (Ubuntu 7.04, Red Hat Linux, ALT Linux, PlayStation 3 OS и другие),
- MacOS X и выше;
- Любой интернет проводник (браузер) поддерживающий:
- HTML 1.0,
- JavaScript,
- Доступ к сети Интернет со скоростью не ниже 512 Кбит\сек.

Моделирование и проектирование программного обеспечения

4.1 Архитектура проектируемого приложения

В основе разрабатываемого программного приложения лежит трехзвенная архитектура «клиент-сервер», она реализуется на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере (рисунок 1).

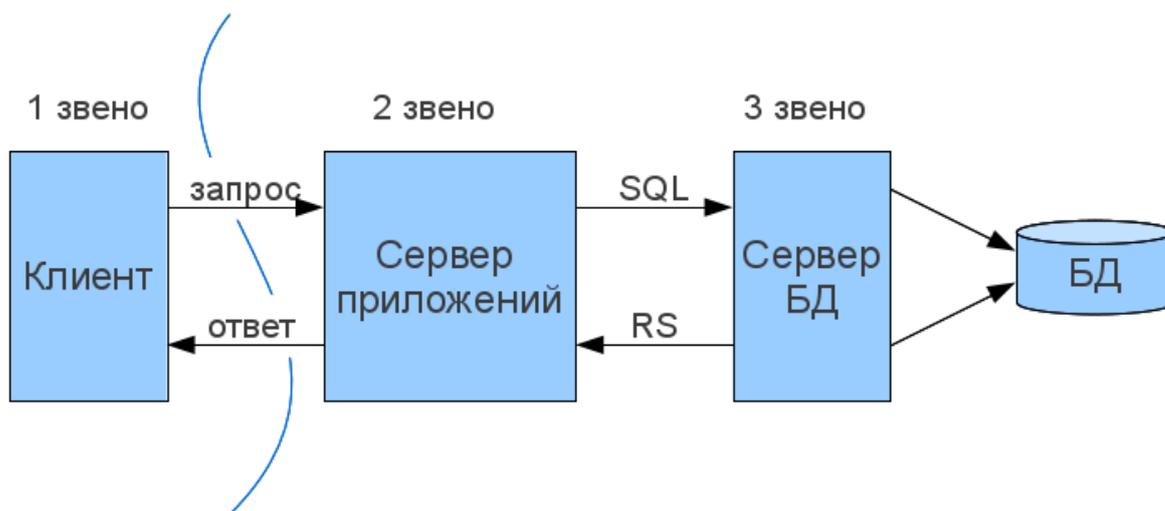


Рисунок 1 – Общая архитектура приложения

Основными достоинствами архитектуры «клиент– сервер» являются:

- Возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы. В частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов.
- Все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.
- Позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т.п.

Основные недостатки:

- В случае использования централизованной системы, неработоспособность основного сервера может сделать неработоспособным все приложение;
- Администрирование данной системы требует квалифицированного профессионала;

- Высокая стоимость оборудования.

4.3 Построение диаграммы вариантов использования

4.3.1 Диаграммы вариантов использования

Диаграммы вариантов использования представляют собой графическое представление взаимодействия пользователя и программного приложения. Каждый вариант использования охватывает некоторую очевидную для пользователей функцию программного приложения и решает некоторую дискретную задачу пользователя. Список всех вариантов использования фактически определяет функциональные требования к системе, с помощью которых может быть сформулировано техническое задание, поэтому диаграмма вариантов использования является необходимым средством при анализе требований, выдвигаемых по отношению к проектируемой системе.

4.3.2 Диаграмма вариантов использования для модуля «Список пациентов»

Для описания функциональности разрабатываемого модуля «Список пациентов» разработана диаграмма вариантов использования, представленная на рисунке 2.



Рисунок 2 – Диаграмма вариантов использования модуля «Запись пациентов»

На диаграмме вариантов использования присутствует один «актант» – пользователь программного приложения.

Врач – пользователи, которые использует систему для выбора и поиска пациентов записанных на прием.

Из рисунка 2 видно, что задачу записи на прием можно представить совокупностью действий:

- Просмотр списка пациентов записанных к врачу на текущий день;
- Поиск пациентов по указанной дате;
- Выбор пациента;
- Страница с информацией о приеме;

4.3.3 Диаграмма вариантов использования для модуля «Подробная информация о приеме»

Для описания функциональности разрабатываемого модуля «Подробная информация о приеме» разработана диаграмма вариантов использования, представленная на рисунке 3.

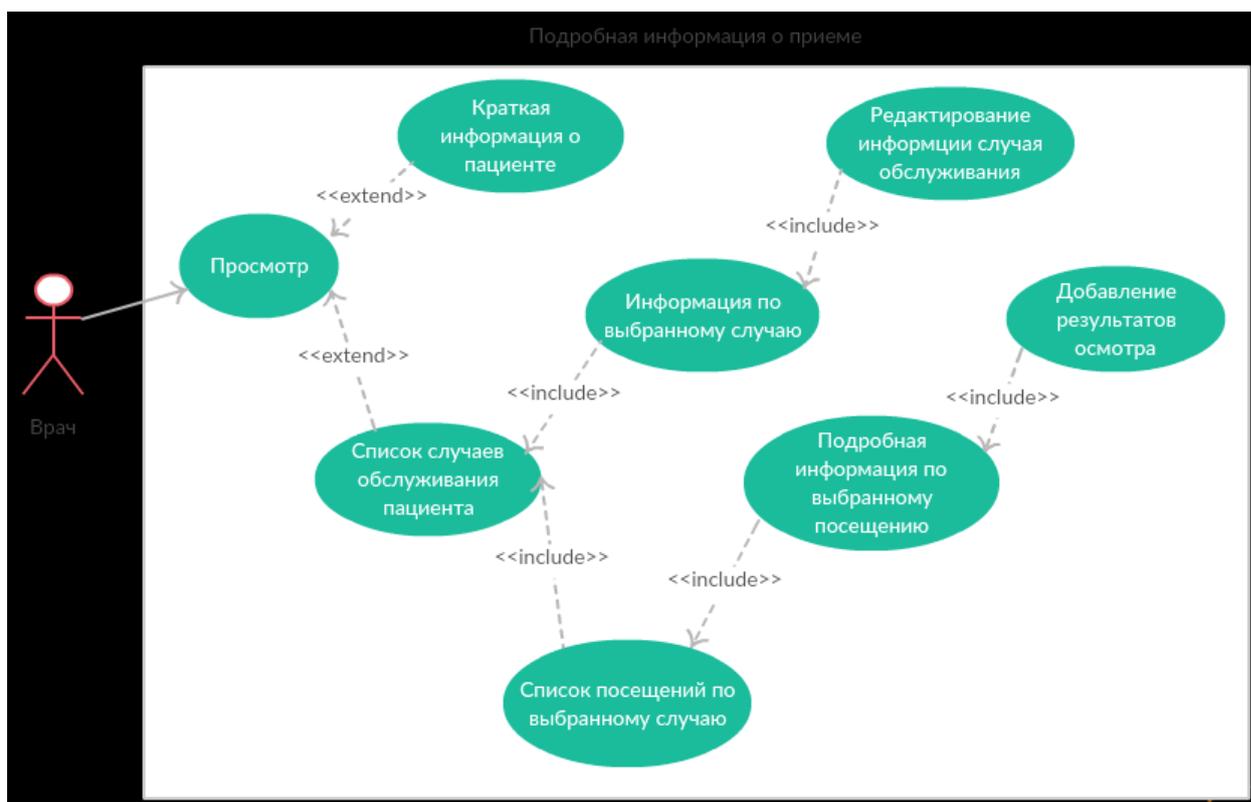


Рисунок 3 – Диаграмма вариантов использования модуля «Подробной информации о приеме»

На диаграмме вариантов использования присутствуют один «актант» – внешний объект, с которым взаимодействует система. Актант «Врач» – пользователь, который использует систему для работы с пациентами.

Из рисунка 3 видно, что пользователю с ролью «Врач» доступны действия:

- Просмотра списка случаев обслуживания пациента;
- Просмотра информации по случаю обслуживания;
- Просмотра краткой информации о пациенте;

- Просмотр списка посещений по выбранному случаю;
- Подробная информация по выбранному посещению;
- Создание нового посещения;
- Создание нового случая;
- Добавление результатов осмотра к посещению;
- Редактирование информации по случаю обслуживания.

4.4 Описание информационной базы

В результате изучения предметной области определена информационная база, представляющая собой следующие части:

- Список пациентов;
- Подробная информация о приеме.

4.4.1 Описание структуры базы данных части «Список пациентов»

Часть «Список пациентов» состоит из логического раздела:

- Список пациентов на прием;

Раздел «список пациентов на прием» состоит из таблицы: RR_VISIT (Посещения пациентов), PACIENT (Список пациентов).

ER-диаграмма раздела представлена на рисунке 4.

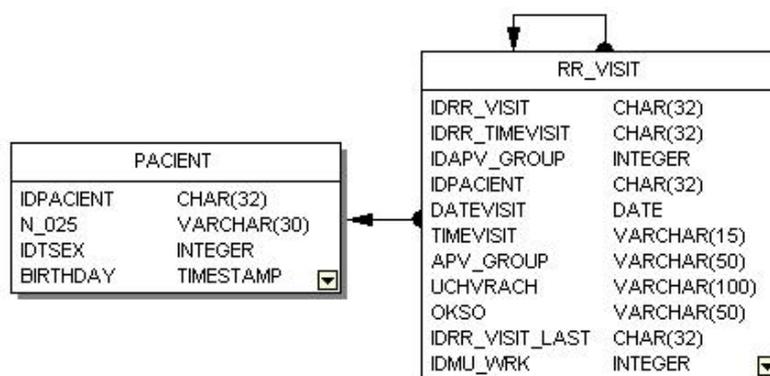


Рисунок 4 – ER-диаграмма раздела список пациентов на прием

4.4.2 Описание структуры базы данных части «Подробная информация о приеме»

Часть «Подробная информация о приеме» состоит из логических разделов:

- Информация о пациенте;
- Случаи обслуживания;
- Информация о случае обслуживания;
- Посещения по случаю обслуживания;
- Информация о посещении;
- Запись случая;
- Запись посещения;
- Запись результатов осмотра.

Раздел «информация о пациенте» состоит из таблицы: PACIENT (Пациенты), HLIST_PASPORT (Паспортная информация пациента).

ER-диаграмма раздела «информация о пациенте» представлена на рисунке 5.

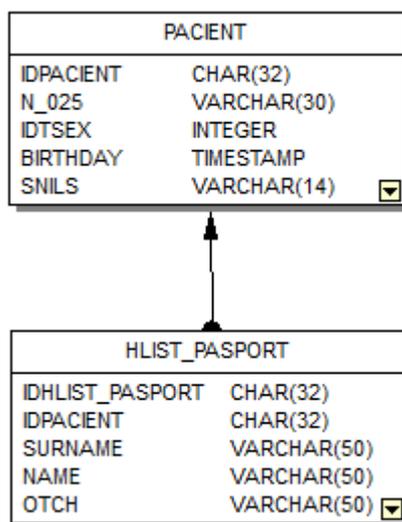


Рисунок 5 – ER-диаграмма раздела информации о пациенте

Раздел «случаи обслуживания» состоит из трех таблиц: LIST(Случаи обслуживания), FFOMS_V006(Классификатор условий оказания медицинской помощи), PACIENT(Список пациентов).

ER-диаграмма представлена на рисунке 6.

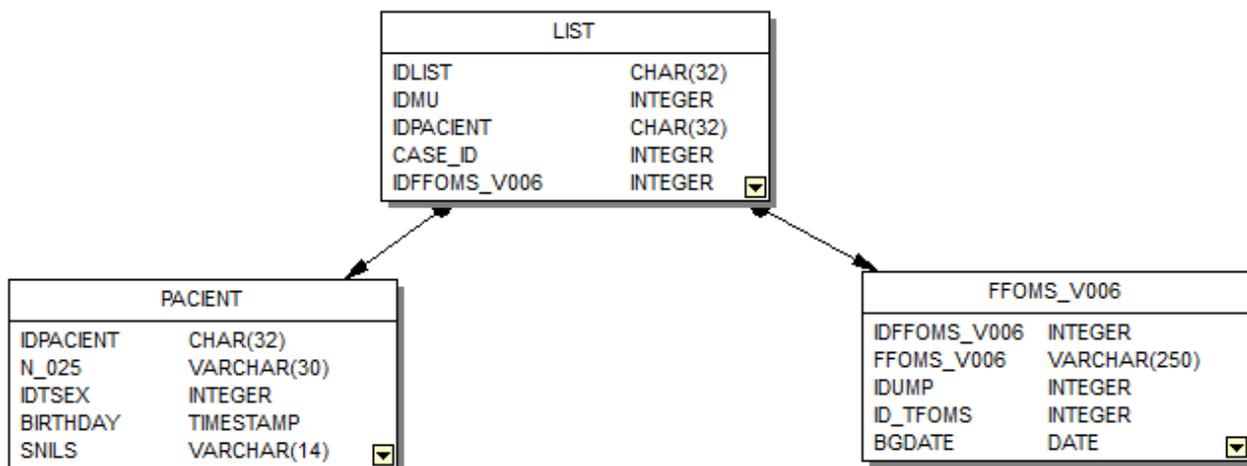


Рисунок 6 – ER-диаграмма случаев обслуживания

Раздел «посещения по случаю обслуживания» состоит из шести таблиц: LIST, LIST_APP(Случаи АПП) VISIT(Посещения), KIND(Справочник видов амбулаторно-поликлинического приема), VRACH(Медработники. Паспортная часть), POSTNCARDS(Должности врачейные).

ER-диаграмма представлена на рисунке 7.

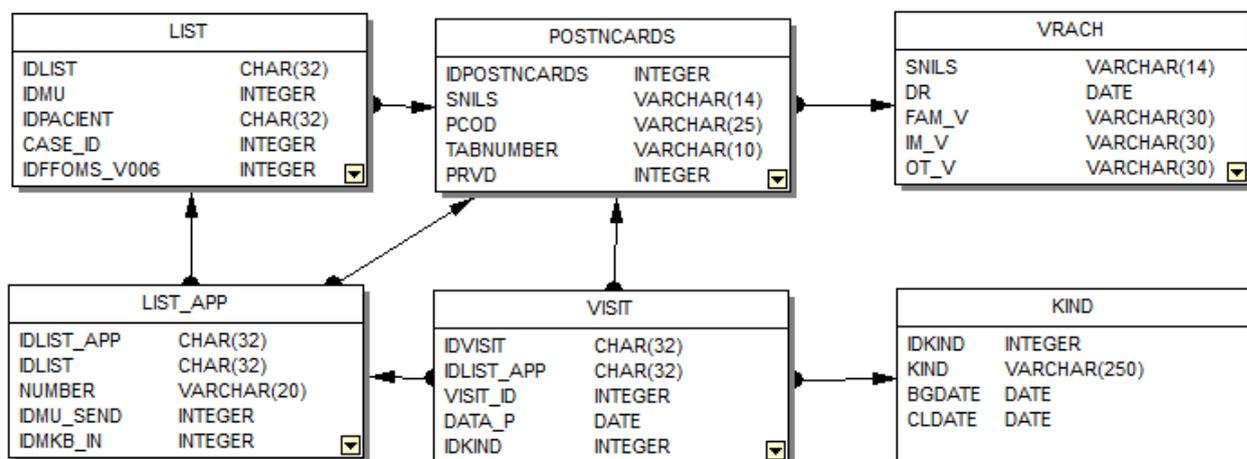


Рисунок 7 – ER-диаграмма посещений по случаю обслуживания

Раздел «информация о посещениях» состоит из семи таблиц: LIST_APP, KIND, VISIT, POST_N(Справочник должностей), DNT_VISIT_RESULTS(Результаты осмотра посещения), VRACH, POSTNCARDS.

ER-диаграмма раздела «информации о посещениях» представлена на рисунке 8.

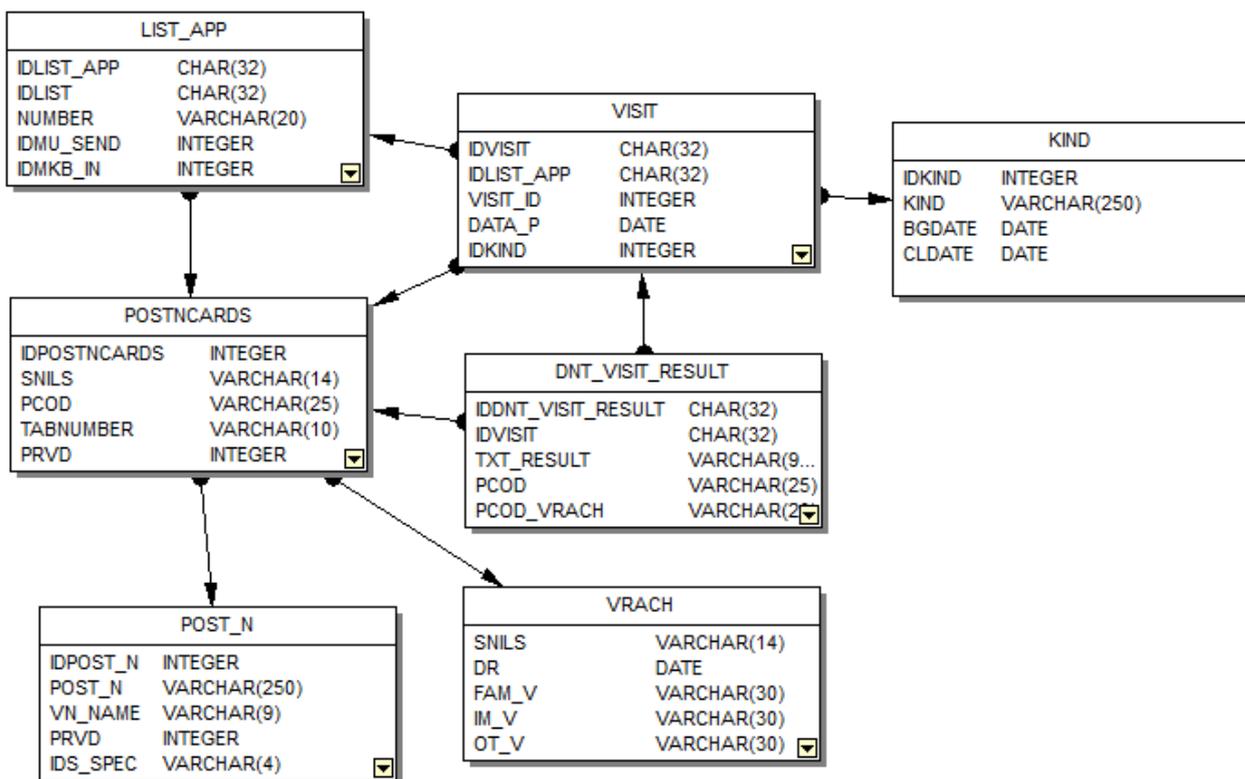


Рисунок 8 – ER-диаграмма информации о посещениях

Раздел «запись случая обслуживания» состоит из 8 таблиц: LIST, LIST_APP, PACIENT, SMAPP(Справочник мест амбулаторно-поликлинического приема), APV(цели амбулаторно-поликлинических посещений), VISIT, KIND, FFOMS_V006.

ER-диаграмма раздела «запись случая обслуживания» представлена на рисунке 9.

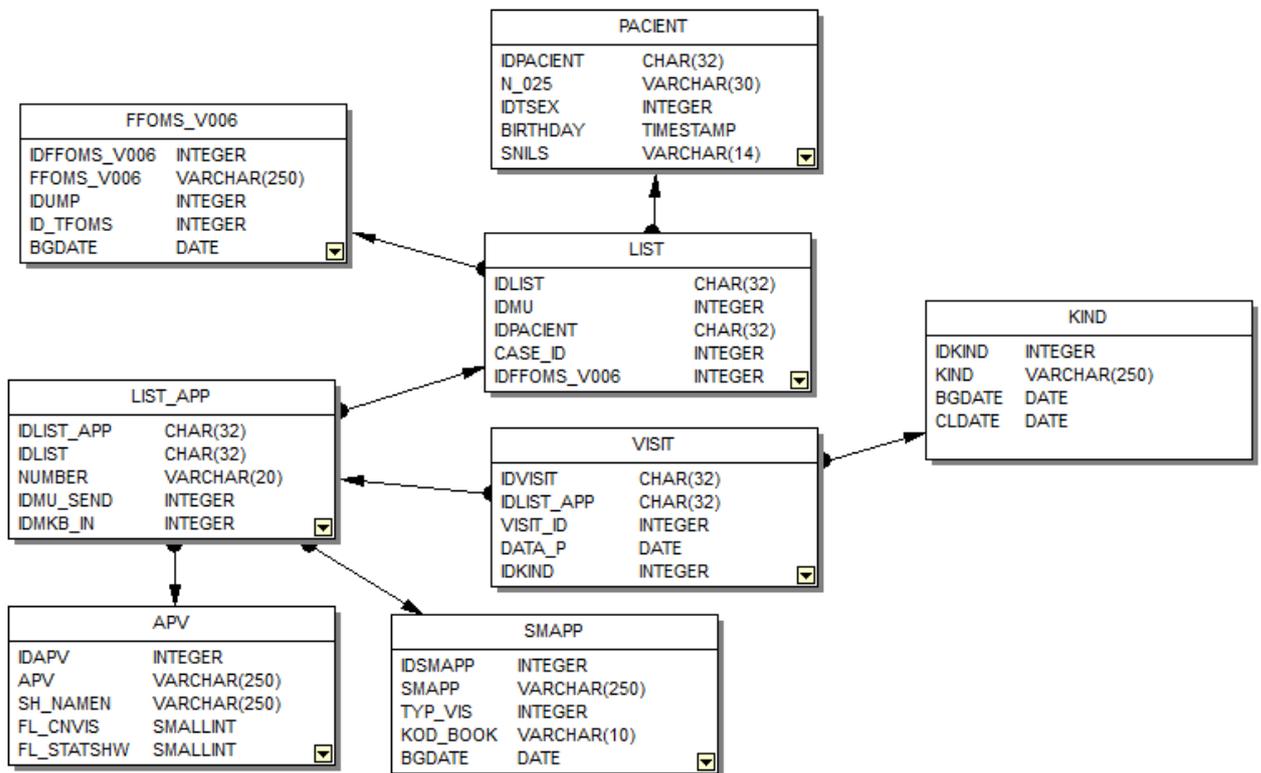


Рисунок 9 – ER-диаграмма записи случая обслуживания

Раздел «запись посещения по случаю обслуживания» состоит из трех таблиц: LIST_APP, KIND, VISIT, VRACH.

ER-диаграмма раздела «запись посещения по случаю обслуживания» представлена на рисунке 10.

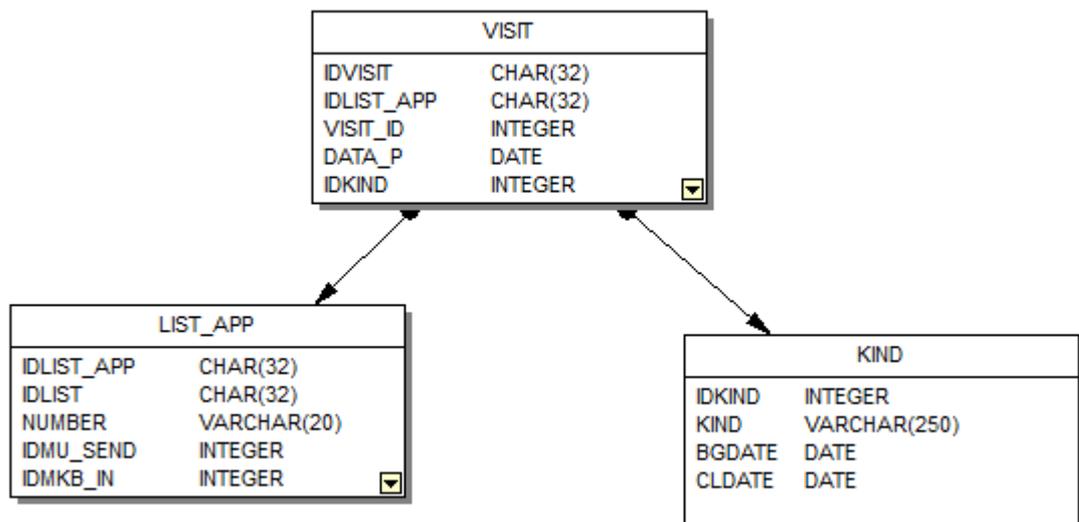


Рисунок 10 – ER-диаграмма записи посещения по случаю обслуживания

Раздел «запись результатов осмотра» состоит из двух таблиц: VISIT, DNT_VISIT_RESULTS.

ER-диаграмма раздела «запись результатов осмотра» представлена на рисунке 11.

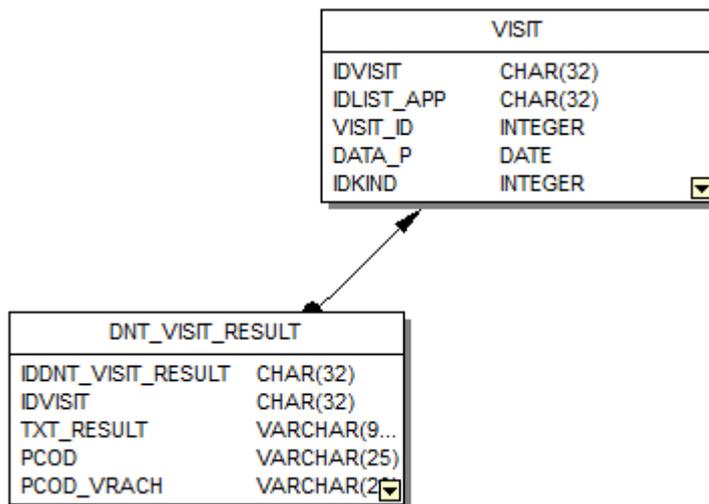


Рисунок 11 – ER-диаграмма записи результатов осмотра

4.5 Разработка логической структуры системы

Для представления статической структуры модели программного приложения необходимо построить диаграмму классов. Диаграмма классов отражает, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и компоненты, а также описывает их внутреннюю структуру и типы отношений.

Каждый модуль разделен на пять типов компонентов:

- модель;
- хранилище;
- пользовательский интерфейс;
- взаимодействие с пользователем;
- взаимодействие с базой данных.

4.5.1 Диаграмма классов модуля «Список пациентов»

Диаграмма классов, отображающая взаимодействие между классами модуля «Список пациентов» представлена на рисунке 12.

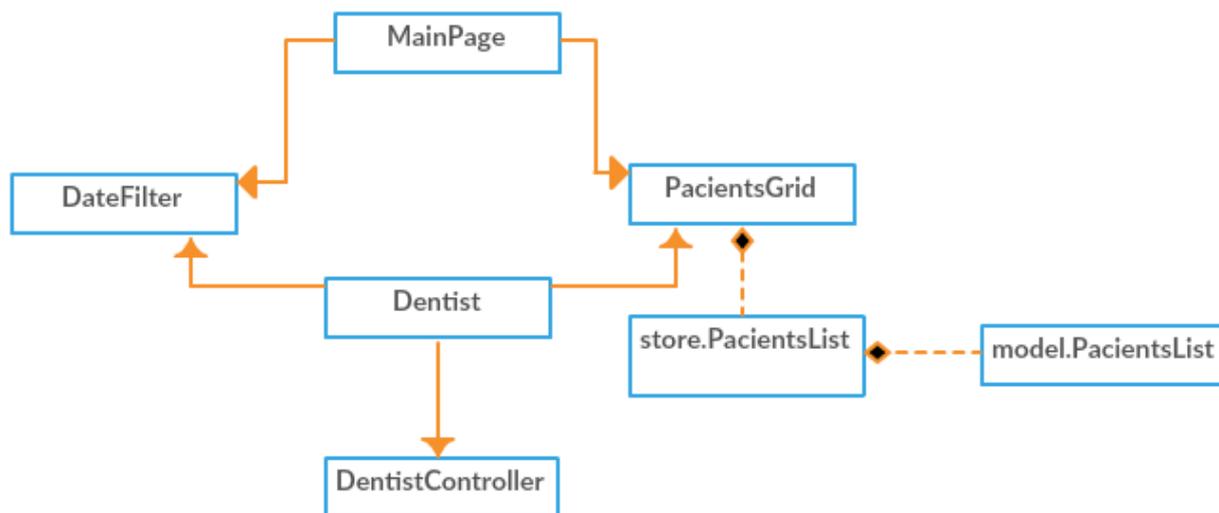


Рисунок 12 – Отношения между классами

Таблица 1 – Описание классов

Тип компонента	Название класса	Описание класса
Модель	model.PacientsList	Модель для хранения данных о списке пациентов
Хранилище	store.PacientsLists	Хранилище для хранения списка пациентов
Пользовательский интерфейс	PacientsGrid	Представление для отображения списка пациентов
	PatientPanel	Представление для отображения панели с информацией о пациенте
	MainPage	Представление для отображения главной страницы модуля «Список пациентов»

Продолжение таблицы 1

	DateFilter	Представление для отображения формы поиска пациентов по дате
Взаимодействие с пользователем	Dentist	Контроллер модуля «Списка пациентов на прием»
Взаимодействие с базой данных	DentistController	Контроллер взаимодействия с базой данных

Окончание таблицы 1.

4.5.2 Диаграмма классов модуля «Подробная информация о приеме»

Диаграмма классов, отображающая взаимодействие между классами модуля «Подробная информация о приеме», представлена на рисунке 13.

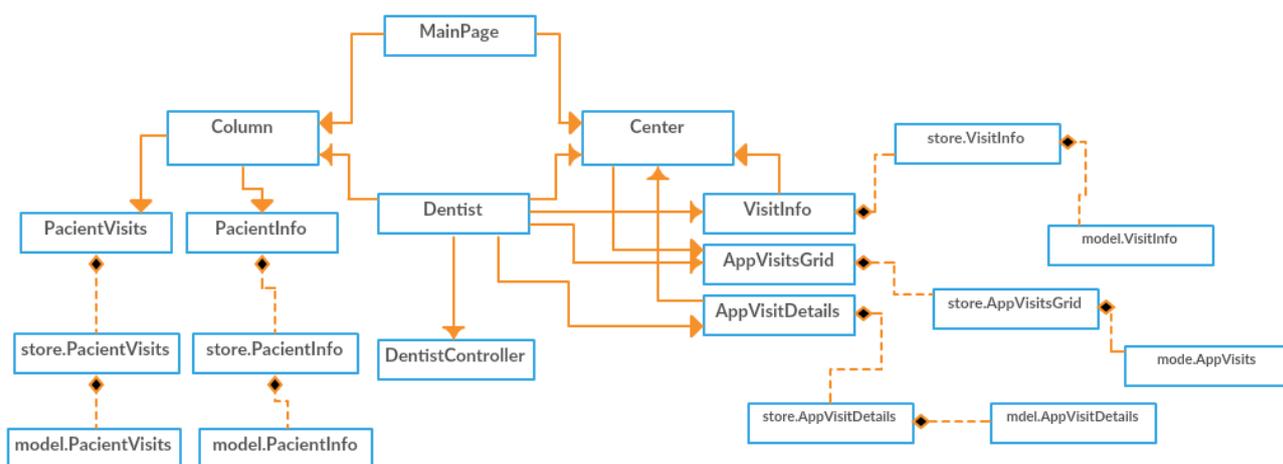


Рисунок 13 – Отношения между классами

Таблица 2 – Описание классов

Тип компонента	Название класса	Описание класса
Модель	model.VisitInfo	Модель для хранения данных о случаи обслуживания
	model.AppVisits	Модель для хранения данных о списке посещений по случаю

Продолжение таблицы 2

	model.AppVisitDetails	Модель для хранения данных о подробной информации посещения по случаю
	model.PacientInfo	Модель для хранения данных о пациенте
	model.PacientVisits	Модель хранения данных о случаях обслуживания
Хранилище	store.PacientInfo	Хранилище для хранения списка пациентов
	store.PacientVisits	Хранилище для хранения списка случаев обслуживания
	store.VisitInfo	Хранилище для хранения информации о случае обслуживания
	store.AppVisitsGrid	Хранилище для хранения списка посещений по случаю обслуживания
	store.AppVisitDetails	Хранилище для хранения детальной информации по посещению
Пользовательский интерфейс	AppVisitsGrid	Представление для отображения списка посещений по случаю обслуживания
	PacientVisits	Представление для отображения -
	PacientInfo	Представление для отображения панели с информацией о пациенте
	AppVisitDetails	Представление для

		отображения подробной информации по посещению
--	--	---

Продолжение таблицы 2

	VisitInfo	Представление для отображения подробной информации о случае обслуживания
	MainPage	Представление для отображения главной страницы модуля «Подробной информации о приеме»
	Center	Представление для отображения центральной панели на странице модуля
	Column	Представление для отображения левой панели на странице модуля
Взаимодействие с пользователем	Dentist	Контроллер модуля «Подробной информации о приеме»
Взаимодействие с базой данных	DentistController	Контроллер взаимодействия с базой данных

Окончание таблицы 2

4.6. Вывод

В данном разделе была проведена работа по проектированию разрабатываемого программного приложения.

Описана функциональность и поведение программного приложения с помощью диаграммы вариантов использования, на которых выделены пользователи разрабатываемого программного приложения, а так же действия, которые они смогут выполнять в системе.

А так же описана схема данных разрабатываемой системы с помощью ER– диаграмм.

5 Описание разработанного программного обеспечения

АРМ «Стоматолога» предназначен для автоматизации деятельности врача стоматолога в поликлинике.

Основными функциями АРМа являются:

- просмотр списка пациентов на прием;
- просмотр случаев обслуживания по пациенту;
- просмотр посещений по случаю обслуживания;
- просмотр результатов осмотра;
- внесение изменений по случаю обслуживания;
- создание нового случая обслуживания;
- создание посещений по случаю обслуживания;
- добавление результатов осмотра.

5.1 Вход в систему

Для входа в программное приложение необходимо запустить браузер и указать в нём адрес ИС «Стационар». Окно входа в систему представлено на рисунке 14.

Авторизация

Пользователь: XXX

Пароль: ...

Роль: Главная мед сестра

Модуль: АРМ Стоматолога

Войти

Рисунок 14 – Вход в систему

В появившемся окне для входа в систему необходимо указать логин пользователя, пароль и роль и модуль.

5.2 Модуль «Список пациентов»

Данный модуль позволяет пользователю программного приложения работать со списком пациентов на прием к врачу. В модуле возможны действия»:

- просмотр списка пациентов;
- поиск по дате;
- выбор пациента.

Пользовательский интерфейс представлен на рисунке 15, а выбора пациента на рисунке 16.

Ачинская межрайонная больница №1 | Рентгенологическое отделение | X:\GL.MED.SESTRA\F:\AmiacR_III_archi... АРМ Стоналога

Список пациентов

13.06.2016

Найти

ФИО	Дата визита	Время визита	Возраст
Аздокина Оризонтл Ныгойрафыш	02.09.2015	09:40 - 10:00	57 лет
Фиреб Тимыз Едонафыш	09.07.2015	08:30 - 08:40	28 лет
Жифрагафо Лыно Фозырьфино	01.06.2016	09:40 - 10:00	71 год
Сюифо Лодорва Кимнотыфино	26.06.2015	11:45 - 12:05	44 года
Ирысолыф Онтлий Улыныфыш	19.06.2015	11:45 - 12:05	35 лет
Зытылгафо Корыно Бодрафино	01.06.2016	09:25 - 09:45	72 года
Ворожаф Фротыныл Тныдыныфыш	10.06.2015	10:20 - 10:40	52 года
Ногафинф Ифимый Фотырыныфыш	24.09.2015	08:30 - 08:40	47 лет
Шогафо Додяно Фозырьфино	10.06.2015	08:00 - 08:10	89 лет
Соголафо Омно Оризонтлафино	26.05.2015	09:20 - 09:40	42 года
Додопыно Ловызо Оризонтлафино	18.06.2015	16:20 - 16:40	82 года
Алригафо Ирысофидо Едонафино	21.05.2015	16:20 - 16:20	93 года
Ондащыннга Фыгдалыя Арихафино	09.07.2015	09:50 - 10:00	6 лет
Калисый Фротыныл Онтлийыфыш	21.05.2015	14:20 - 14:40	54 года
Зонитаф Зыйлон Зофыл ахри	21.05.2015	15:40 - 16:00	47 лет
Эндыном Бобир Родлындыныфыш	21.05.2015	15:00 - 15:20	76 лет
Тимызынга Ловызо Зофырьфино	02.07.2015	15:20 - 15:40	77 лет
Клогарызынг Етоолт Фротынылафыш	01.06.2016	08:20 - 08:40	83 года
Зыталафо Эфидроно Выдалафино	09.06.2015	09:20 - 09:40	53 года
Рогинф Фашыроф Выталафыш	09.06.2015	09:40 - 10:00	76 лет
Фротига Додяно Фротынылафино	26.06.2015	13:05 - 13:25	60 лет
Илагыно Фыгдалыя Фротызофафино	09.06.2015	10:20 - 10:40	9 лет
...

Страница 1 из 4

Отображаются записи 1 - 30 из 111

Рисунок 15 – Модуль списка пациентов

Ачинская межрайонная больница №1 | Рентгенологическое отделение | X:\GL.MED.SESTRA\F:\AmiacR_III_archi... АРМ Стоналога

Список пациентов

Фытыфыш Ехаль Онтлийыш

Информация о пациенте

ФИО: **Фытыфыш Ехаль Онтлийыш**
 Дата рождения: 18.05.1992
 Пол: Мужчина

Случай обслуживания

Новый случай

- 09.06.2016 10:40:00 [Амбулаторно]
- 08.06.2016 13:05:00 [Амбулаторно]
- 06.06.2016 12:56:00 [Амбулаторно]
- 06.06.2016 12:54:00 [Амбулаторно]
- 06.06.2016 12:54:00 [Амбулаторно]
- 04.11.2012 02:31:00 [Стационарно]

Информация по случаю обслуживания

Внести изменения

Источник финансирования: Краевой бюджет
 Дата закрытия: 2016-06-09 10:40:00
 Диагноз МКБ10(основной): СМЕРТЬ МАТЕРИ ОТ ПОСЛЕДСТВИЙ ПРЯМЫХ АКУШЕРСКИХ ПРИЧИН Код: O97.X
 Вид травмы: Отсутствует
 Эпикриз: Отсутствует
 Исход: Умер
 Результат: Отсутствует

Визиты пациента

Новое посещение

Дата посещения	Вид приема	Врач
15.04.2014	Основной	Баскова Татьяна Григорьевна
10.04.2014	Основной	Баскова Татьяна Григорьевна

Страница 0 из 0

Посещения для отображения отсутствуют

Информация о посещении

Добавить результаты осмотра

Рисунок 16 – Выбор пациента

5.3 Модуль «Подробная информация о приеме»

Данный модуль позволяет пользователю программного приложения работать с данными пациента. В модуле возможны действия»:

- создание нового случая обслуживания;
- внесение изменений в случай обслуживания;
- создание посещения;

- добавление результатов осмотра.

Пользовательский интерфейс создания нового случая обслуживания представлен на рисунке 17, изменение случая обслуживания на рисунке 18, создание посещения на рисунке 19, а добавление результатов осмотра на рисунке 19.

Создание нового случая

Общая информация | Информация о беременности

Дата и время начала случая: 17.06.2016 00:27

Место(тип) посещения: В поликлинике

Цель случая АПП обслуживания:

Беременность

Создать текущее посещение?: Да

Создать | Заккрыть

Информация о посещении

+ Добавить результаты осмотра

Рисунок 17 – Создание нового случая обслуживания.

Изменение случая обслуживания

Источник финансирования: Добровольное Медицинское Страхования (ДМС)

Дата и время начала случая: 17.06.2016 00:28

Дата и время окончания случая: 17.06.2016 00:28

Диагноз(МКБ10): ДРУГИЕ УТОЧНЕННЫЕ ОБСТОЯТЕЛЬСТВА, НАМЕРЕНИ

Исход: Лечение продолжено

Результат: Улучшение

Сохранить | Заккрыть

Информация о посещении

+ Добавить результаты осмотра

Рисунок 18 – Изменение случая обслуживания.

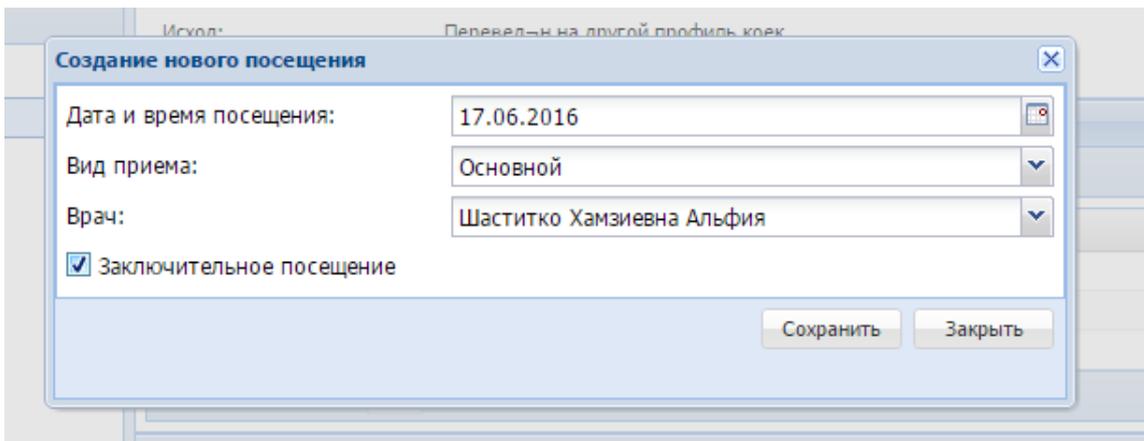


Рисунок 19 – Создание посещения.

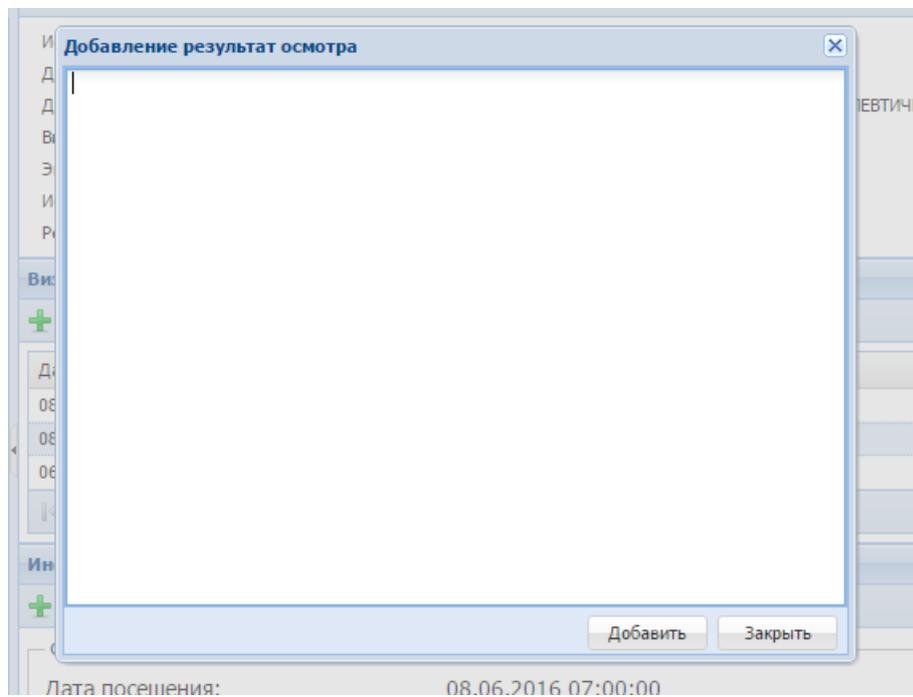


Рисунок 20 – Добавление результатов осмотра.

ЗАКЛЮЧЕНИЕ

Целью данной работы являлась разработка автоматизированного рабочего места врача стоматолога.

В данной работе был проведен анализ работы стоматологии поликлиники, как предметной области. Осуществлен анализ существующих программных продуктов для автоматизации работы стоматологии поликлиники. Были сформированы требования к программному приложению.

Так же в рамках работы были решены следующие задачи:

- Спроектирована структура АРМа;
- Спроектирован пользовательский интерфейс, соответствующий стилю и требованиям системы «Стационар»;
- Проведен анализ схемы базы данных. Введены необходимые сущности, реализованы объекты серверной логики (представления, триггеры);
- Выполнена разработка автоматизированного рабочего места врача стоматолога.

В ходе выполнения работы получены следующие навыки:

- Работа в команде и использование средств коллективной разработки:
 - Система контроля версий – Subversion;
- Использование коллективного кода (ядра):
 - Классы работы взаимодействия серверной и клиентской части приложения;
 - Аутентификация и авторизация пользователя;

Разработанное программное обеспечение планируется к внедрению в медицинские учреждения Красноярского края.

СПИСОК СОКРАЩЕНИЙ

- АПП – амбулаторно-поликлиническая помощь
- АРМ – автоматизированное рабочее место
- ЕСВС – единая система ведения справочников
- ИС – информационная система
- ИТ – информационные технологии
- ККМИАЦ – Красноярский краевой информационно-аналитический центр
- ЛПУ – лечебно-профилактическое учреждение
- МЗ – министерство здравоохранения
- ОМС – обязательное медицинское страхование
- ОС – операционная система
- РМИАС – Региональная медицинская информационно-аналитическая система
- РФ – Российская федерация
- СМО – страховая медицинская организация
- СУБД – система управления базами данных
- ФИО – фамилия имя отчество
- МКБ-10 - Международная классификация болезней 10-го пересмотра (МКБ-10)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. РИАМС «ПроМед». Ссылка на электронный источник:
http://swan.perm.ru/elektronnoe_zdravoohranenie/riams_promed
(Дата обращения 15.01.2016).
2. ИТ в медицине. Ссылка на электронный источник:
<http://www.cnews.ru/reviews/free/publichealth/article/region.shtml>
(Дата обращения 15.03.2016).
3. Информационная система IDent. Ссылка на электронный источник:
<http://dent-it.ru/> (Дата обращения 10.01.2016).
4. Информационная система «Dentist+». Ссылка на электронный источник: <http://www.dentist-plus.com/> (Дата обращения 10.01.2016),
5. Информационная система IStom. Ссылка на электронный источник:
<http://i-stom.ru> (Дата обращения 10.01.2016).
6. ExtJS Sencha Docs. Ссылка на электронный источник:
<http://docs.sencha.com/extjs/4.2.3/> (Дата обращения 10.04.2016).
7. Yii Framework. Ссылка на электронный источник:
<http://www.yiiframework.com/doc/guide/1.1/ru/quickstart.what-is-yii>
(Дата обращения 10.04.2016).