

Тольятти 2018

АННОТАЦИЯ

Объект исследования: предобработка данных по пациенту и отправка их клиенту.

Предмет исследования: система управления контентом для медицинского приложения.

Цель работы: разработка системы управления контентом для медицинского приложения.

Новизна работы заключается в разработке системы управления контентом для передачи данных по пациентам мобильному клиенту и обеспечению мобильного доступа к данным пациентов врачам.

Для достижения цели работы необходимо решить следующие задачи:

1. Проанализировать существующую обработку данных в учреждениях здравоохранения.
2. Выбрать архитектуру сервер-приложения, обеспечивающей необходимые показатели эффективности при передаче и предобработке данных по пациенту.
3. Разработать новую информационную технологию, существенно улучшающую процесс передачи и предобработки данных по пациентам мобильному клиенту.
4. Реализовать проект разработанного программного модуля с использованием современных подходов при конструировании сервер-приложений.

Отчет состоит из введения, трех глав и заключения.

В аналитической части рассказано, о существующей технологии взаимодействия с данными по пациентам в учреждениях здравоохранения и описываются выявленные требования.

Описана разработка диаграммы прецедентов, выбор средств, для реализации, разработка архитектуры системы управления контентом.

Описываются алгоритмы реализации технологии, и анализируются ранняя версия приложения и существующая.

Выпускная квалификационная работа выполнена на 50 страницах, состоит из введения, трех глав, заключения и списка литературы, состоящего из 30 литературных источников. Также работа содержит 7 рисунков, 9 таблиц.

Результатом работы является разработанная система управления контентом реализующая функции передачи и предобработки данных по пациентам мобильному клиенту, а так же имеющую возможность расширения, все это позволяет снизить нагрузку на мобильный–клиент и время, затрачиваемое на передачу данных по пациентам.

ABSTRACT

The title of the graduation work is «Analysis of software architectures for a content management system in a medical application »

The object of the graduation work is preprocessing patient data and sending them to the client.

The subject of the graduation work is the content management system for a mobile application.

The aim of the work is to give some information about the development of the content management system for a mobile application.

The graduation work describes in details the development of a new system for a mobile physician access to patient data.

The graduation work consists of an explanatory note on 50 pages, including 7 figures, 9 tables, the list of 30 references including 7 foreign sources.

We first discuss the relevance of this graduation work, evaluate the current state of the doctors' work with the patients' data, describe how manual and automated systems of monitoring and processing patients' data in hospitals work using the example of Togliatti municipal clinical hospital No. 5 and develop the requirements for the new architecture.

Next we describe the development of the case diagram use, the choice of means for implementation, the development of the architecture of the content management system. To select the means for the implementation, a comparative analysis is performed using an interpolating cubic spline.

Finally, we present the results of comparing the first application without the content management system and the new system.

Overall, the results suggest that the developed content management system implementing the functions of transmitting and preprocessing the patient data to the mobile client, and having the possibility of expansion allows reducing the load on the mobile client and the time spent on transferring data to patients.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 АНАЛИЗ ОБРАБОТКИ ДАННЫХ В УЧРЕЖДЕНИЯХ ЗДРАВООХРАНЕНИЯ.....	8
1.1 Анализ информационных систем в учреждениях здравоохранения.....	8
1.2 Анализ технологии доступа к информации в системе учреждений здравоохранения.....	17
1.3 Выработка требований к новой технологии по доступу к информации по пациенту	21
2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ДЛЯ МЕДИЦИНСКОГО ПРИЛОЖЕНИЯ	28
2.1 Разработка новой архитектуры системы управления контентом для медицинского приложения.....	28
2.2 Проектирование архитектуры системы управления контентом	29
2.3 Проектирование реляционной модели данных для обеспечения хранения данных о пациентах	31
3 КОНСТРУИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ	34
3.1 Выбор языка программирования и дополнительных фреймворков.....	34
3.2 Реализация первой версии системы с использованием библиотеки DICOM	42
3.3 Реализация системы управления контентом с использованием фреймворка Spring с надстройкой Spring Boot	43
ЗАКЛЮЧЕНИЕ	45
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	47

ВВЕДЕНИЕ

Мобильные устройства и приложения имеют большой вес в современном мире, их развитие и внедрение в различные сферы деятельности активно растет.

Активное использование мобильных приложений для развития малого бизнеса, среднего бизнеса, путем мобильного контроля бумаг, составления графиков и анализа данных, хранение любой информации и иметь доступ к ней в любой момент времени, является отличным примером использования мобильных технологий.

В настоящее время проблема эффективного и быстрого доступа к данным, касается и больниц. Врачам приходится работать с большим количеством бумаг и снимков, имея в использовании стационарное рабочее место, которое хоть и предоставляет большой спектр возможности, но ограничивает врача в передвижении и снижает продуктивность. Мобильное приложение способно избавить работников клиник от полной привязки к стационарному компьютеру и поиск информации по пациенту в большом количестве документов. Но хранение большого объема информации без предобработки на мобильном устройстве не рационально, т.к. это замедлит работу самого приложения и эффективность мобильной копии рабочего места снизиться.

Актуальность работы заключается в том, что создание системы управления контентом для медицинского приложения позволяет снять с приложения необходимость постоянного содержания данных по пациенту на мобильном устройстве и при необходимости вытаскивание необходимых данных за считанные секунды. Объект исследования: предобработка данных по пациенту и отправка их клиенту.

Предмет исследования: система управления контентом для медицинского приложения.

Цель работы: разработка системы управления контентом для медицинского приложения.

Задачи работы:

1. Проанализировать существующую обработку данных в учреждениях здравоохранения.
2. Выбрать архитектуру сервер-приложения, обеспечивающей необходимые показатели эффективности при передаче и предобработке данных по пациенту.
3. Разработать новую информационную технологию, существенно улучшающую процесс передачи и предобработки данных по пациентам мобильному клиенту.
4. Реализовать проект разработанного программного модуля с использованием современных подходов при конструировании сервер-приложений.

Выпускная квалификационная работа состоит из введения, трех глав и заключения.

Введение кратко описывает актуальность исследования, ставится цель, определяется объект и предмет выпускной квалификационной работы, выявляются задачи, которые необходимо выполнить для достижения цели.

В первой главе, анализируются информационные системы в учреждениях здравоохранения, технологии доступа к информации в системе учреждений здравоохранения, требования к новой технологии по доступу к информации по пациенту.

Во второй главе описывается разработка системы управления контентом для медицинского приложения, выбор языка и фреймворков.

В третьей главе описывается реализация и анализ программного блока.

В заключении подводятся итоги проделанной работы, приводятся выводы, достигнутые в ходе выполнения выпускной квалификационной работы.

1 АНАЛИЗ ОБРАБОТКИ ДАННЫХ В УЧРЕЖДЕНИЯХ ЗДРАВООХРАНЕНИЯ

1.1 Анализ информационных систем в учреждениях здравоохранения

В учреждениях здравоохранения каждый день обрабатывается большой объем данных, для контроля этих данных были разработаны различные виды систем, которые можно разделить на ручные и автоматизированные. Соответственно анализ информационных систем в учреждениях здравоохранения был проведен на примере ТГКБ №5 и его работе с ручными системами и автоматизированными.

Клиническая больница №5(ТГКБ №5) – является одной из самых крупных региональных учреждений здравоохранения Российской Федерации. Стационарный комплекс строился в 70-80-е годы и изначально являлся Медсанчастью.

На данный момент больница располагает 2408 коек, где ежегодно получают стационарную медицинскую помощь более 80 тысяч человек и 200 тысяч получают амбулаторную помощь, рождаются 6000 детей, проводится более 1 миллиона исследований в год. В составе больницы имеются службы такие как: хирургии, терапии, онкологии, гинекологии, межрайонный перинатальный центр, межрайонный травматологический центр 1 уровня, городской центр эндопротезирования крупных суставов, инфекционная, детская, диагностический центр, городской СПИД центр.

Так же в больнице функционируют 48 коечных отделений, 5 приемных, 10 реанимационных и анестезиологических, 5 операционных блоков, 17 параклинических отделений, 2 поликлинических, 5 вспомогательных служб. Круглосуточно работают два компьютерных томографа. В больнице имеется два МРТ-томографа [7].

Как писалось ранее виды систем способных контролировать данные в учреждениях здравоохранения можно подразделить на ручные и автоматизированные. Под ручными системами подразумевается

регистратура, хоть постепенно и идет создание электронной регистратуры, но до сих пор не все учреждения поддерживают данную функцию. Устройство регистратуры стандартизировано во всех учреждениях страны, исключением не стала и ТГКБ №5. Регистратура формирует и распределяет потоки пациентов, тем самым налаживая и систематизируя постоянный прирост данных по пациентам. Углубляясь в основные задачи регистратуры, выделяется всего 3 вида:

- регистратура организует предварительную и неотложную запись пациентов на прием, как в учреждениях здравоохранения, так и по телефону;
- регистратура контролирует и распределяет интенсивный поток пациентов с целью равномерного распределения нагрузки на врачей и распределения по видам оказываемых услуг;
- проведение в короткие сроки подбора и доставки медицинской документации в кабинеты врачей, а так же ведение и хранение картотеки учреждения здравоохранения [18].

Первые две задачи являются средством постоянного потока пациентов и помогают централизованно и систематизировано вести учет за прибывающими пациентами, а также вносить их данные в базы данных для дальнейшего использования. Это является первым плюсом регистратур, централизованное и систематизированное управление интенсивным потоком данных по пациентам, с учетом загруженности врачей и видами оказываемых услуг. Третий пункт является одной из проблем регистратуры. Медицинская документация, в числе которых и медицинские карты пациентов, хранится в бумажном виде на стеллажах, так чтобы можно было легко найти и выдать необходимые данные, но в реальности количество данных зачастую настолько большое, что сотрудники не могут вести весь учет такого количества данных. Из-за этого происходят случаи утраты медицинской документации с последующими проблемами, как у сотрудников, так и у пациентов. Данный минус является основным в ручной информационной системе.

В настоящий момент организация работы в регистратуре происходит следующим образом:

- работа организуется по централизованной системе и исходит от участково-территориального принципа обслуживания населения;
- руководит всей работой заведующий, назначенный главным врачом учреждения здравоохранения;
- организует прием больных медицинский регистратор, данный сотрудник первым встречается с больным, беседует и в случае необходимости помогает с разбором организации приема. Так же медицинский регистратор разбирается с сортировкой больных по медицинским показаниям, т.к. это позволяет решать вопросы регулирования потока больных;
- для регулирования большого потока больных совершенствуются существующие формы работы регистратуры;
- рациональная организация приема новых пациентов, для сокращения времени ожидания на прием к врачам. Данная организация задает ритм работы всех подразделений учреждения здравоохранения.

В настоящее время, чтобы попасть на прием к врачу в ТГКБ №5 необходимо записаться при помощи одного из методов.

Первый, это с помощью талонной системы. Посетитель получает в регистратуре талоны с указанием порядкового номера очереди, фамилии врача, номера кабинета и времени явки на прием, после чего отправляется на прием [13].

Второй метод, это самозапись на данный момент она выглядит следующим образом в ТГКБ №5. Посетитель авторизуется на сайте электронной регистратуры Министерства здравоохранения Самарской области, выбирает город, нужное медицинское учреждение, специалиста, дату и время, после этого автоматически формируется талон и посетителю остается лишь прийти к выбранному времени.

Стоит оговориться, что существует и третий метод, комбинированный, который включает в себя и самозапись и талонную систему.

Как было сказано ранее регистратура хранит медицинскую документацию и включает в себя следующие основные принципы:

- все амбулаторные карты расставлены на стеллажах по участкам, внутри них идет еще большее распределение – по улицам, домам и квартирам, данная система позволяет организовать быстрое обслуживание больных;

- все амбулаторные карты маркируются разными цветами, в соответствии с терапевтическим участком;

- система хранения медицинских карт с номерами — каждому участку определяется количество номеров равное числу лиц, проживающих на данной территории, при этом поиск одной карты может составлять всего несколько секунд.

Плюсом такого хранения медицинской документации является грамотная структурированность, позволяющая быстро находить документы, выдавать и вести учет.

Минусом же является то, что в реальных условиях, когда поступает большой поток посетителей, документы постоянно уходят к врачу посредством передачи их посетителям, поскольку количество посетителей гораздо превышает число сотрудников регистратуры, однако это не самая большая проблема. Основной проблемой является передача документов на дом посетителям, к примеру, когда необходимо амбулаторная карта в другом медицинском учреждении. Из-за такого постоянного хождения медицинской документации происходит утеря данной документации самим посетителем или же по вине сотрудников медицинского учреждения.

Как писалось ранее, существует также и автоматизированная информационная система в учреждениях здравоохранения. Данный вид информационных систем включает в себя целый ряд взаимосвязанных элементов таких как медицинское оборудование, автоматизированное

рабочее место (АРМ), медицинская информационная система (МИС), медицинский сервер. Так же хоть электронная регистратура и была описана в ручной информационной системе под методом самозапись, но сама по себе электронная регистратура является автоматизированной.

Ниже на рисунке 1, представлена диаграмма развертывания, объединяющая в структуру получения и хранения данных по пациентам, взаимосвязанные элементы перечисленные ранее.

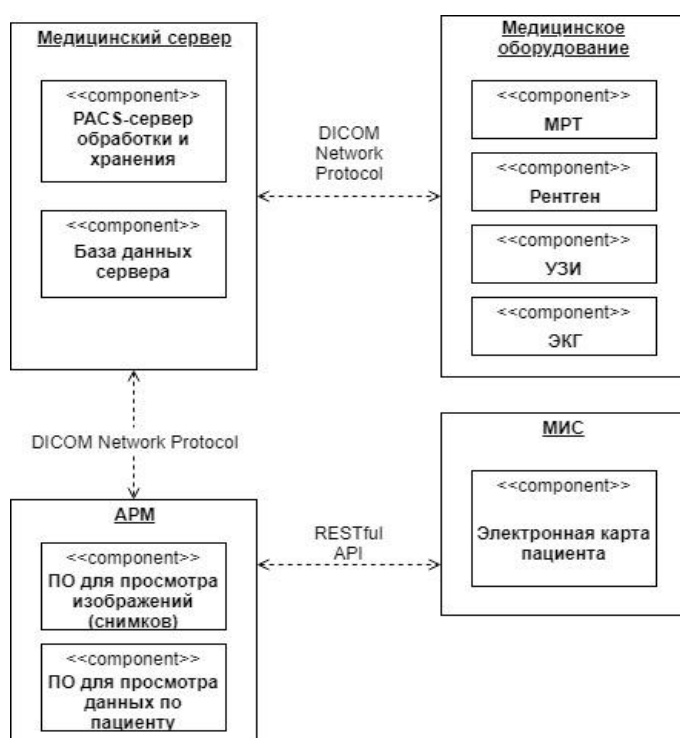


Рисунок 1 – Диаграмма развертывания для учреждений здравоохранений

Из диаграммы мы видим, что врач выполняет обследование пациента при помощи медицинского оборудования такого, как МРТ, рентген, УЗИ и т.д. Данное оборудование при диагностике пациента упаковывает всю информацию в DICOM-файл. DICOM (англ. Digital Imaging and Communications in Medicine) – это стандарт создания, хранения, передачи и визуализации медицинских изображений и документов обследованных пациентов. Он опирается на ISO – стандарт OSI и поддерживается основными производителями медицинского оборудования и медицинского программного обеспечения. Данный стандарт позволяет создавать, хранить, передавать и печатать отдельные кадры изображения, серии кадров, информацию о пациенте, исследовании, оборудовании, учреждениях, медицинском персонале, производящем обследование, и т. д [9].

Медицинское оборудование передает DICOM-файл медицинскому серверу по DICOM Network Protocol, сетевому протоколу, предназначенному для передачи DICOM-файлов и управления DICOM-командами по сетям с поддержкой TCP/IP. После передачи, медицинские сервера «парсят» содержащуюся в нем информацию и кладут в базу данных или при помощи PACS'а архивируется и передается на АРМ.

АРМ является очень важной частью в данной информационной системе, поскольку именно данное стационарное автоматизированное рабочее место врача является инструментом, для обработки и просмотра данных по пациентам.

Автоматизированное рабочее место (АРМ) - это совокупность средств, реализованных на базе персонального компьютера, для решения задач в определенной предметной области.

Другое определение трактует АРМ как аппаратно-программный комплекс, предназначенный для автоматизации технологических процессов данной специальности.

В медицине и здравоохранении АРМ подразделяют по функциональным возможностям на следующие категории:

- административно-организационные (например, АРМ главного врача больницы, заведующего отделением, старшей медицинской сестры и др.);
- технологические (например, АРМ врача-рентгенолога, врача функциональной диагностики и др.);
- интегрированные, т. е. объединяющие функции первых двух в разных комбинациях (например, АРМ главного рентгенолога города) [6].

Одним из больших преимуществ АРМ является его модульность, под этим подразумевается, что при необходимости можно подключить определенный модуль с дополнительными функциональными возможностями. Для наглядности пример будет предоставлен из статьи обобщающая "3D моделирование изображений компьютерной томографии (КТ) и магнитно-резонансной томографии (МРТ) сколиоза позвоночника" [29]. Были выбраны пациенты со сколиозом в возрасте от 10 до 25 лет. Пациентов ознакомили с условиями исследования. Такие оценки исходного состояния, как возраст, пол, рост, вес и дата завершения лечения, записываются для всех пациентов, участвующих в исследовании. Информация, полученная от пациентов, хранится конфиденциально и никогда не будет общедоступной.

После завершения исследования состояния пациентов, были получены DICOM-файлы с КТ и МРТ. Данные DICOM-файлы использовались в программном обеспечении InVesalius версии 3.0. InVesalius - это программное обеспечение с открытым исходным кодом для реконструкции компьютерной томографии и магнитно-резонансных изображений. Оно создает 3D-визуализацию изображений, основанную на последовательности DICOM-файлов. В InVesalius была импортирована папка с DICOM-файлами, и была выбрана интересующая область. Свойства маски, свойства поверхности и пороговые свойства настраиваются в соответствии с потребностями. После установки свойств создается 3D-поверхность и экспортируется в формате STL-файла. После создания 3D-поверхности он

импортируется в программное обеспечение Magics версии 18. Magics - это программный пакет для подготовки данных и редактор STL. Файл, экспортированный из InVesalius, импортируется в Magics.

Импортированный файл STL редактируется в Magics. Оболочки, отверстия, треугольники, наложения, сшивание и т. д. редактируются, чтобы получить окончательную трехмерную модель человеческих позвонков.

Такие методы, как метод Кэчмарза (Kaczmarz), используются в области реконструкции изображений из проекций. Это итеративный алгоритм решения систем линейных уравнений. Его также называют ART. Он обозначается формулой:

$$x^{k+1} = x^k + \lambda_k \frac{b_i - (a_i, x^k)}{\|a_i\|^2} a_i, \quad (1)$$

где a_i – i -ый столбец матрицы A ,

b_i – i -ая компонента вектора b ,

λ_k – параметр релаксации.

Импортированный файл в Magics первоначально выглядел как поврежденная, шумная 3D-модель. Края и кривые не были хорошо видны. После обработки в Magics и устранения нескольких проблем получается четкая трехмерная модель человеческого позвоночника с уменьшенным искажением. Получившаяся модель намного лучше предыдущей. Данная трехмерная модель, при включении тканевых свойств, может быть использована для определения механических свойств сколиотических позвонков с использованием анализа конечных элементов. Существуют и другие программные продукты, такие как ITKSnap, которые могут преобразовывать DICOM-файлы в 3D-модель. Авторы статьи не получили лучшего результата, когда пытались создать 3D-модель с использованием программного обеспечения ITKSnap, поскольку изображения DICOM содержат много шумов.

Выше приведенный пример наглядно показывает, что используя АРМ можно внедрить отображение 3Д моделей, поскольку в данное рабочее место

можно встраивать различные программные средства, имея необходимый метод, есть возможность моделирования полноценных моделей из, например, снимков МРТ, что позволит более качественно вынести диагноз пациенту, чем и пользуются в ТГКБ №5.

АРМ, как диагностическое устройство подходит как никто лучше для работы с данными пациентов, его плюсы, заключающиеся в возможности расширения функциональности и полном оборудовании для просмотра, а также диагностики данных по пациентам. Но главным минусом АРМ, является его стационарность, из-за этого врач вынужден либо находится за рабочим местом, либо брать данные по пациентам с собой, к примеру, рентгеновские снимки. Но врач не всегда может взять все снимки с собой, поскольку существующее исследование МРТ может содержать в одной серии более чем 400 снимков.

Последним элементом в автоматизированной информационной системе является МИС. Медицинская информационная система (МИС) – является системой для автоматизации документооборота для лечебно-профилактических учреждений. Данная система объединяет в себе поддержки принятия медицинских решений, электронные медицинские карты о пациентах, данные о медицинских исследованиях, данные отслеживания состояния пациента с медицинских приборов, средства общения между сотрудниками, финансовая и административная информация [12]. МИС классифицируют по направлению деятельности медицинского учреждения:

- для стационаров;
- для поликлиник и амбулаторий;
- для стоматологических клиник;
- для санаториев.

В случае ТГКБ №5 МИС используется для поликлиник и амбулаторий и содержит в себе электронные медицинские карты о пациентах.

Из достоинств автоматизированных информационных систем можно выделить:

- строгая структурированность;
- возможность расширения функционала АРМ.

Недостаток автоматизированных информационных систем это:

- полная стационарность, привязывающая врача к своему АРМ.

Для устранения данного недостатка было принято решение разработать систему мобильного доступа к данным по пациентам.

1.2 Анализ технологии доступа к информации в системе учреждений здравоохранения.

Для доступа информации в системе учреждений здравоохранения используется АРМ, а так же МИС. Не смотря на то, что ранее было описано, как работает АРМ и то, что она состоит из модулей, не было написано, как именно получает доступ к информации врач с помощью АРМ на данный момент.

В настоящее время в ТГКБ №5 для просмотра и обработки снимков по пациентам используются следующие программные средства:

1. Philips DICOM Viewer (работает не только на устройствах Philips).
2. eFilm for Workstations.
3. OsiriX (только для macOS).
4. RadiAnt.

Рассмотрим две основные программы для работы со снимками пациентов, это Philips DICOM Viewer и eFilm for Workstations.

eFilm for Workstations предназначено для работы с электронными снимками формата DICOM и широко используется учреждениями здравоохранения, поскольку является лидером среди аналогов и обладает обширным функционалом, включающим в себя широкий набор инструментов для работы с изображениями [13]. eFilm позволяет масштабировать, измерять расстояния, измерять выделенную площадь,

располагать несколько снимков в одном окне, изменять яркость и контрастность снимка, для выявления темных пятен. Так же вся необходимая информация по пациенту отображена по бокам самого снимка, поскольку DICOM файл несет в себе данные по пациенту.

Philips DICOM Viewer – это программа для просмотра снимков формата DICOM. Основные функции данной программы представлены ниже:

1. Поддержка формата DICOM. Работа как с локальными сериями снимков, так и получение снимков с удалённого сервера.
2. Быстрая навигация по серии снимков. Возможность свободного перемещения по серии, используя колёсико манипулятора типа “мышь”.
3. Возможность обработки изображений. Изменение яркости и контрастности, применение различных фильтров.
4. Манипуляций с изображениями. Возможность изменения масштаба.
5. Поддержка метаданных. Отображение информации о серии и пациенте.

Построим диаграмму прецедентов для выявления проблем существующей системы.

Выделим актёров системы:

1. Лечащий врач.
2. Врач, проводящий исследования.
3. Программист.
4. Инженер.

Выявим прецеденты системы:

1. Проведение исследования.
2. Просмотр сведений по пациенту.
3. Обслуживание системы.

На рисунке 2 представлены актеры и прецеденты в виде диаграммы прецедентов.



Рисунок 2 – Диаграмма прецедентов получения медицинских данных на АРМ

Опишем спецификации прецедентов в таблицах 1-3:

Таблица 1 – Прецедент: Проведение исследования

Прецедент: Проведение исследования
ID: 1
Краткое описание: Проведение исследования пациента при помощи медицинского оборудования
Главные актеры: Врач, проводящий исследования
Второстепенные актеры: Нет
Предусловия: Прецедент начинается по инициативе пользователя
Основной поток: 1. Пользователь проводит исследование пациента при помощи медицинского оборудования. 2. Данные исследований перемещаются из устройства в базу данных информационной системы.
Постусловия: Исследование проведено.
Альтернативные потоки: Нет

Таблица 2 – Прецедент: Просмотр сведений по пациенту

Прецедент: Просмотр сведений по пациенту
ID: 2
Краткое описание: Проведение врачом анализа исследования с целью установления диагноза
Главные актеры: Лечащий врач
Второстепенные актеры: Нет
Предусловия: Прецедент начинается по инициативе пользователя
Основной поток: 1. Поиск нужного исследования. 2. Если исследование не найдено, то выход из прецедента. 3. Если исследование найдено: 1. Провести анализ исследования. 2. Сделать заключение.
Постусловия: Врач получил данные исследования.
Альтернативные потоки: Нет

Таблица 3 – Прецедент: Обслуживание системы

Прецедент: Обслуживание системы
ID: 3
Краткое описание: Проведение своевременного технического обслуживания системы
Главные актеры: Программист, Инженер
Второстепенные актеры: Нет
Предусловия: Прецедент начинается по регламенту или по состоянию системы

Продолжение таблицы 3

Основной поток:
1. Поиск неисправности.
2. Если неисправность найдена:
1. Проведение ремонта
3. Если неисправности не найдены:
1. Выполняется выход из прецедента.
Постусловия: Обслуживание системы проведено.
Альтернативные потоки: Нет

На основании спецификации претендентов можно выделить следующие недостатки:

1. Жесткая привязка к стационарному автоматизированному рабочему месту.

2. Использование формата DICOM для просмотра снимков пациента увеличивает количество затрачиваемого времени по сравнению, например, с JPEG, но стоит выделить, что на сильных аппаратных средствах данное замедление будет не критическим, но ощутимым поскольку вес таких изображений выше в 5-10 раз.

3. Наличие высокоскоростного сервер хранения данных.

Устранить недостатки существующей системы можно, разработав отдельное приложение для просмотра данных по пациенту с использованием новой системы управления контентом.

1.3 Выработка требований к новой технологии по доступу к информации по пациенту

Для того чтобы составить и проанализировать требования новой технологии возьмем существующую систему классификаций требований FURPS+ IBM, где IBM – это методология ведения программных проектов, относящаяся к спиральной модели, путем итеративного процесса разработки программного обеспечения, созданного компанией Ration Software в 1996 году [14].

Для фиксирования дополнительной информации будет использоваться набор атрибутов требований RUP [1]. Требования, которые были выявлены для новой технологии, представлены в таблице 4.

Таблица 4

№	Требования	Статус	Полезность	Обоснование выбора
Funcrionality – Функциональные требования				
1	Первичный поиск данных по пациенту с критериями поиска (Дата поступления, вид исследования)	Одобренные	Критичное	В системе управления контентом должен быть реализован первичный поиск всех пациентов по дате поступления и виду исследования
2	Поиск конкретного пациента из списка по ФИО	Одобренные	Критичное	В системе управления контентом должен быть реализован поиск конкретного пациента по ФИО
3	Передача списка ссылок серий снимков сразу с первичным поиском пациентов	Одобрено	Критичное	В системе управления контентом должна быть реализована передача списка ссылок серий снимков, поскольку это уменьшает время передачи снимков при дальнейшем запросе серий снимков

Продолжение таблицы 4

4	Передача атрибутов пациента (Пол, дата рождения и тд.)	Одобрено	Критичное	В системе управления контентом должна быть реализована передача атрибутов пациента, поскольку эти данные необходимы врачу для работы с пациентом
5	Передача серий снимков по пациентам	Одобрено	Критичное	В системе управления контентом должна быть реализована потребность в передаче серий снимков по пациентам
6	Обновление поля даты обращения	Одобрено	Важно	Для удаления старых снимков система должна обновлять дату последнего обращения к данным пациента
Usability – Требования к удобству работы				
7	Автономность	Одобрено	Важное	Система должна автономно работать без постоянного вмешательства человека
8	Подробная документация	Одобрено	Важное	Система должна полностью документирована для улучшения и поддержки системы лицами не разрабатывающими с самого начала

Продолжение таблицы 4

Reliability – Требования к надежности				
9	Передача серий снимков без информации по пациентам	Одобрено	Критично	Система управления контентом не должна нарушать закона о персональных данных и при утечке снимков, не передавать конфиденциальную информацию по пациентам
10	Работа при отсутствии данных по пациенту отправка сообщения об этом и корректное продолжение работы	Одобрено	Критичное	Система управления контентом должна продолжать работать при отсутствии данных по пациентам
11	Передача снимков без большого потери качества	Одобрено	Критичное	Система управления контентом должна передавать серии снимков по пациентам без сильной потери качества снимков, чтобы врач мог их

Продолжение таблицы 4

				просмотреть и проанализировать
12	Работа без перебоев в течении рабочего дня и обновление данных в ночное время	Одобрено	Критичное	Система управления контентом должна работать без перебоев в течение рабочего дня, чтобы не вызывать проблемы с просмотром информации по пациентам, особенное в критических случаях и обновляться во время, когда система не используется
Performance – Требования к производительности				
13	Передача серии из 400 снимков не более чем за 1 минуту	Одобрено	Важно	Система управления контентом должна иметь минимальное время отклика для врача
14	Передача 1-го снимка не более чем за 95 мс	Одобрено	Важно	
Supportability – Требования к простоте поддержки				
15	Простое разворачивание системы	Одобрено	Важно	Система управления контентом не

Продолжение таблицы 4

	управления контентом			должна доставлять неудобства в установке
16	Простая настройка без указания местонахождения базы данных и серий снимков по умолчанию, но с возможностью их изменения	Одобрено	Важно	Система не должна занимать большого количества времени при установке, но должна содержать возможность для гибкого использования
17	Простота тестирования при помощи Junit	Одобрено	Важно	Система должна иметь способ тестирования новой функции не занимающего много времени
Implementation requirements – Требования к реализации				
18	Реализация системы с помощью языка программирования Java и фреймворка Spring с надстройкой Spring Boot	Одобрено	Важно	Система должна быть реализована при помощи средств предназначенных для работы с серверной части
Interface requirements – Требование к интерфейсу				
19	Взаимодействие с локальным Conquest сервером	Одобрено	Критически	Взаимодействие системы управление контентом с локальным сервером необходимо для обновления баз данных и вытягивание данных по пациентам из основной базы данных

Продолжение таблицы 4

20	Время ожидания локальной БД не более 5 секунд	Одобрено	Важно	Система должна выдавать ошибку при ожидании отклика от локальной БД, чтобы иметь осведомленность о проблеме
Physical requirements – Физические требования				
21	Система должна быть развернута на компьютерах с размером жесткого диска не менее 30 Гбайт	Предложенные	Важно	Системе необходимо место для серий снимков

Таким образом, на основе анализа было выявлено, что необходимо реализовать 6 функциональных требований и 15 нефункциональных ограничений представленных в таблице.

2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ДЛЯ МЕДИЦИНСКОГО ПРИЛОЖЕНИЯ

2.1 Разработка новой архитектуры системы управления контентом для медицинского приложения

Новая архитектура должна представлять собой трехуровневое приложение и включать в себя клиент (мобильный или браузер), уровень логики (сервер приложение) и уровень хранения данных (SQLite).

Для разработки архитектуры будет использоваться шаблон Model-View-Controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») – это схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- модель (англ. Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать;

- представление, вид (англ. View). Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами;

- контроллер (англ. Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

На рисунке 3 представлена новая разработанная архитектура с использованием Spring.

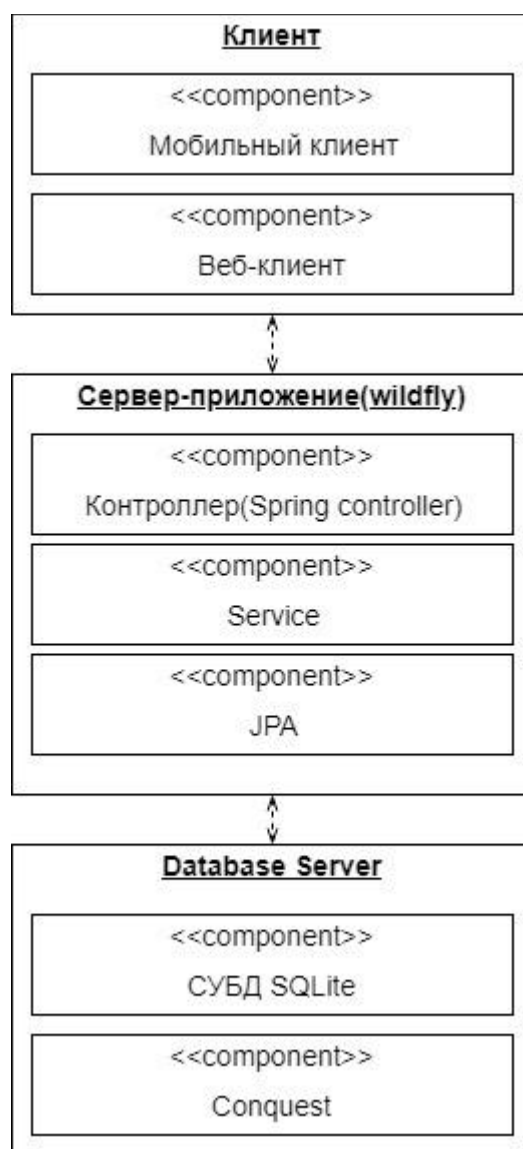


Рисунок 3 – Диаграмма компонентов системы управления контентом

Клиент отвечает за вывод данных по пациентам для врача. Контроллер(Spring controller) выполняет обработку запроса клиента. Service отвечает за бизнес-логику сервера. JPA отвечает за выполнение запросов в базу данных. СУБД SQLite отвечает за хранение данных по пациентам. Conquest – сервер для обращения к базе данных медицинского учреждения.

Результатом разработки получилась архитектура обеспечивающая реализацию всех функциональных требований к новой информационной системе.

2.2 Проектирование архитектуры системы управления контентом

Вся платформа для просмотра медицинских данных разработана при помощи спиральной модели. Данная модель, является классическим

примером эволюционной стратегии конструирования, разработанной Боэмом в 1998 году, и включает в себя как итеративность, так и этапность [11].

На рисунке 4 представлена диаграмма компонентов разработанной платформы.

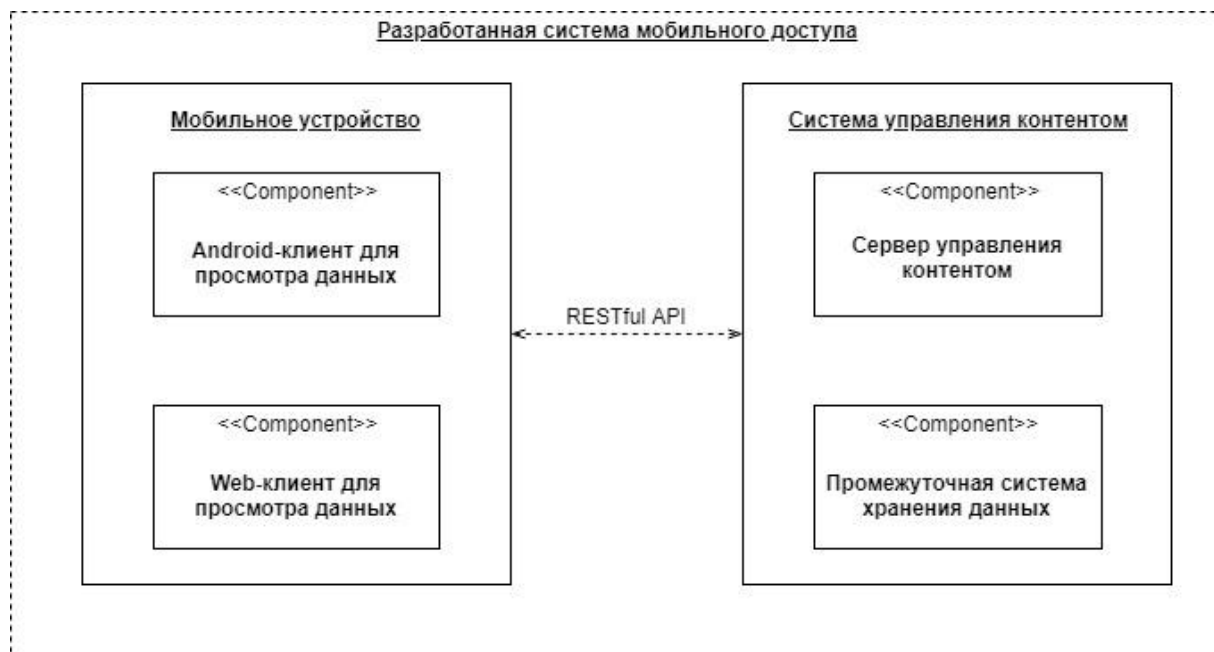


Рисунок 4 – Диаграмма компонентов новой системы

Данная модель показывает отношения между актерами и прецедентами, а так же формулирует требования к функциональному поведению системы. Также выделены функции, которые будет выполнять система.

Ниже на рисунке 5, представлена диаграмма последовательности системы управления характеризующая и отображающая поведение и функционал системы управления контентом.

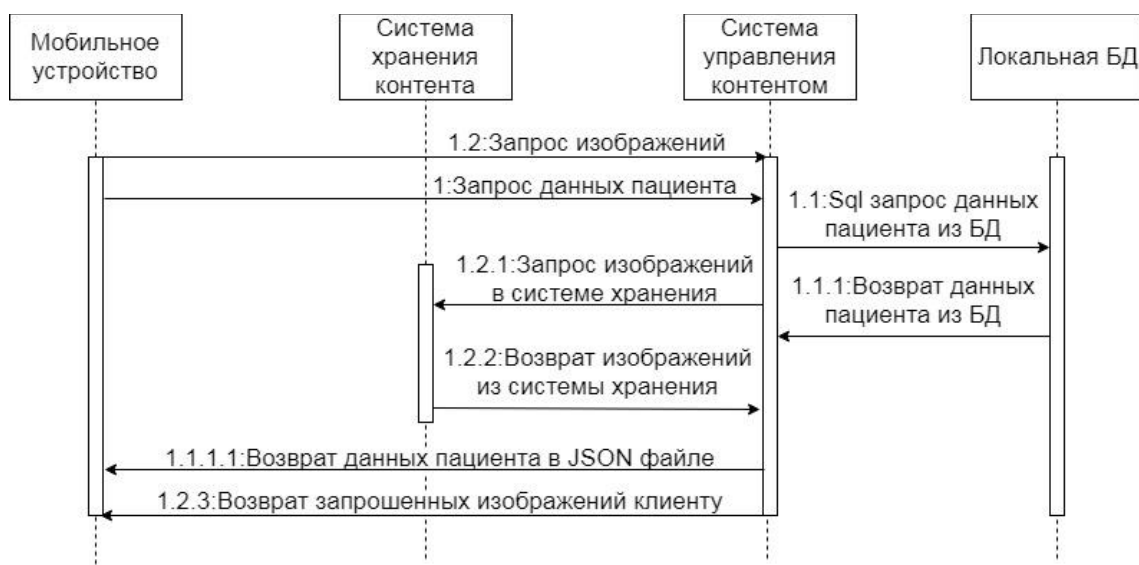


Рисунок 5 – Диаграмма последовательности системы управления контентом

Исходя из выше представленной диаграммы, можно наглядно увидеть всю цепочку работы системы управления контентом и ее взаимодействия с другими элементами системы.

2.3 Проектирование реляционной модели данных для обеспечения хранения данных о пациентах

Разрабатываемая система должна хранить свои данные в базе данных. Один из этапов создания базы данных это проектирование. Проектированием – это процесс в ходе, которого создается схема базы данных. Для этого процесса сначала необходимо разработать модель базы данных, в данном случае, реляционную модель данных [4].

На основании требований выделим имена атрибутов и запишем их в виде множества схемы отношения R:

R {ФИО пациента, Модальность, Пол пациента, День рождения пациента, Идентификатор пациента, День обследования пациента, Номер серии, Описание обследования, Номер снимка, Путь расположения снимка, Идентификатор серии, Идентификатор снимка}.

Полученная схема отношений имеет такие недостатки как:

1. Дублирование информации (избыточность).
2. Потенциальная противоречивость (аномалии обновления).

3. Потенциальная возможность потери сведений (аномалии удаления).

4. Потенциальная возможность не включения информации в базу данных.

Для исключения данных недостатков воспользуемся методом декомпозиции (разбиения). Метод декомпозиции представляет собой процесс последовательной нормализации схем отношений: каждая новая итерация соответствует нормальной форме более высокого порядка и обладает лучшими свойствами по сравнению с предыдущей. При этом число отношений возрастает.

На рисунке 5 представлена полученная модель в виде диаграммы объектов-связей (ER-диаграммы).

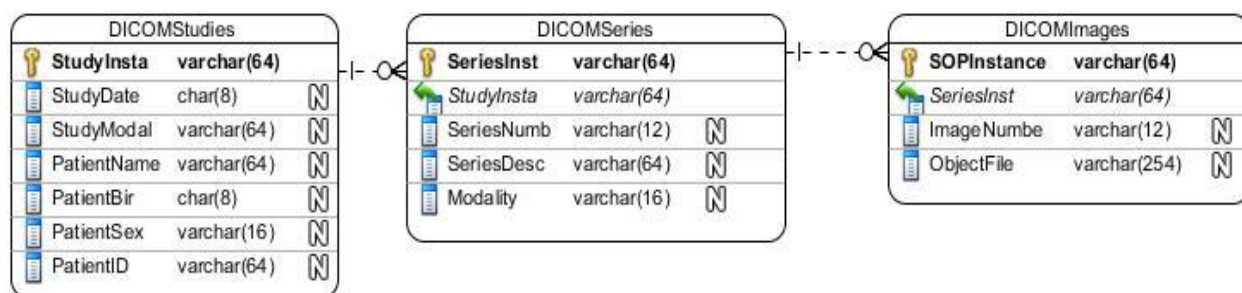


Рисунок 5 – Диаграмма сущность-связь разработанной модели

Таким образом, была разработана реляционная модель базы данных необходимая для хранения данных в разрабатываемом приложении.

2.4 Разработка архитектуры программного модуля системы управления контентом

Для реализации архитектуры, представленной на рисунке 3, разработаем диаграмму пакетов системы управления контентом.

Главный пакет `its.krut.medp` является базовым пакетом для всего приложения. В данном пакете находится пакет `server`, который в свою очередь содержит следующие пакеты, где лежат классы:

— `config` – это пакет, содержащий классы связанные с настройкой подключения к СУБД и безопасности;

- `controller` – это пакет, содержащий классы обеспечивающие связь между клиентом, то есть эти классы принимают запросы клиента, обрабатывают и отправляют обратно ответ;
- `entity` – это пакет, содержащий классы описывающие модели данных;
- `repository` – это пакет, содержащий классы которые взаимодействуют с СУБД. Их обязанность реализовать CRUD операции. CRUD операции – это возможность создания, чтения, обновления и удаления объекта или списка из СУБД;
- `service` – это пакет, содержащий классы, которые описывают бизнес логику приложения;
- `util` – это пакет, содержащий утилитарные классы, например, правильный перевод кириллицы.

Разработанная диаграмма пакетов представлена на рисунке 6.

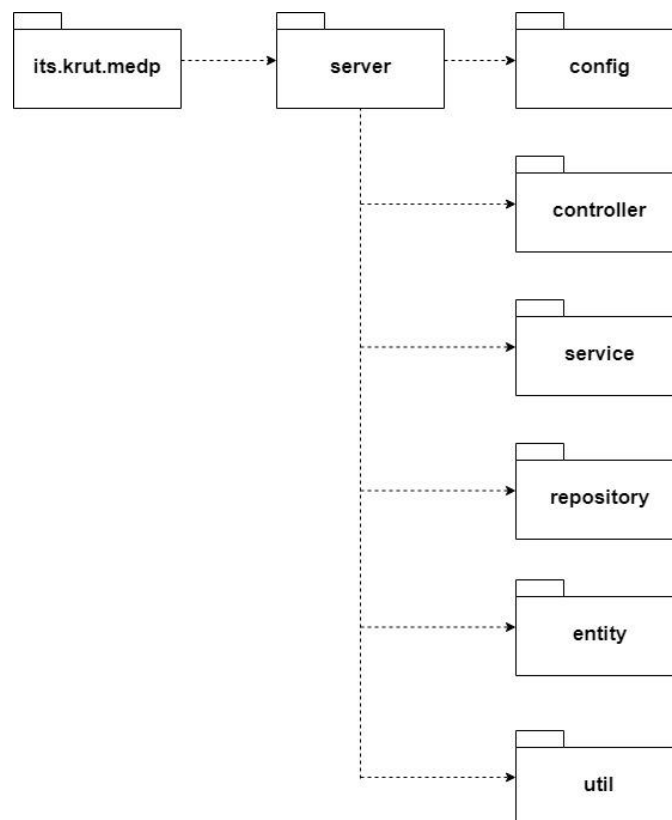


Рисунок 6 – Диаграмма пакетов системы управления контентом

Таким образом, разработана архитектура по шаблону MVC, диаграмма компонентов которой изображена на рисунке 3.

3 КОНСТРУИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

3.1 Выбор языка программирования и дополнительных фреймворков

Перед конструированием системы необходимо определиться с выбором языка программирования. Рассмотрим три языка применяемых в разработке веб-приложений: C#[16], PYTHON[20], JAVA[5].

Введем критерии оценки:

1. Наличие опыта (10 баллов) – это наличие знаний и навыков работы на данном языке программирования;
2. Направленность на веб-приложения (5 баллов) – это наличие инструментов для разработки веб-приложения;
3. Масштабируемость (5 баллов) – в будущем система должна расширяться, чтобы с системой могли работать и другие службы больницы;
4. Кроссплатформенность (5 баллов);
5. Наличие объектно-ориентированной парадигмы программирования (5 баллов).

Сравнительный анализ представлен в таблице 5.

Таблица 5 – Сравнительный анализ языков программирования

Критерии оценки	Java	C#	Python
Наличие опыта	8	2	5
Направленность на веб-приложения	5	1	3
Масштабируемость	5	3	3
Кроссплатформенность	5	4	4
Наличие объектно-ориентированной парадигмы программирования	5	5	5
Итого:	28	15	20

На основе проведенного анализа был выбран объектно-ориентированный язык программирования Java. Исходя из требований, система должна быть реализована в виде веб-приложения. Рассмотрим два

фреймворка для разработки веб-приложения на языке Java: J2EE [2], JAVA Spring [3].

Введем критерии оценки:

1. Наличие опыта (10 баллов) – наличие знаний и навыков разработки веб-приложения на данном языке.
2. Компонентность (5 баллов) – компонентные фреймворки позволяют быстро разрабатывать приложения, используя готовые компоненты.
3. Персистенстность (5 баллов) – разрабатываемый программный модуль должен сохранять своё состояние.
4. Производительность (5 баллов) – система должна иметь минимальную по времени реакцию для ответа на действия пользователя.

Сравнительный анализ представлен в таблице 6.

Таблица 6 – Сравнительный анализ Фреймворков

Критерии оценки	J2EE	Spring
Наличие опыта	5	9
Персистенстность	4	5
Компонентность	5	5
Производительность	4	5
Итого:	18	24

Несмотря на проведенный сравнительный, анализ выделим основные достоинства и недостатки фреймворков.

Достоинства JavaEE:

1. Технология, основанная на спецификации. Это означает, что спецификация EJB оговаривает практически все аспекты реализации, что в свою очередь значит высокую взаимозаменяемость между решениями от различных производителей, что в идеале должно дать разработчику возможность выбора наиболее подходящей платформы для своего проекта.
2. Интеграция с платформой J2EE. Спектр технологий, поддерживаемых в рамках J2EE, позволяет решить любую задачу, связанную с обработкой информации на стороне сервера.

3. Свободный доступ к сложным системам управления ресурсами. Это означает, что целый спектр решений для доступа и управления ресурсами, включающий в себя подсистемы управления транзакциями, безопасностью, временем жизни компонентов, и т.д., не нужно реализовывать самостоятельно.

4. Большое и энергичное сообщество разработчиков и развитый рынок решений. За все время существования JavaEE вокруг него создалось крепкое комьюнити, что означает высокую стабильность и перспективность данной технологии, т.к. большинство IT компаний поддерживают и работают с JavaEE.

Недостатки JavaEE:

1. Большая по объему и сложная спецификация.

2. Увеличение времени разработки. В данном случае, имеется в виду большее время на разработку решения с использованием EJB по сравнению с написанием обычного java-кода, а не с использованием другой платформы поддержки распределенных вычислений.

3. Повышенная сложность по сравнению с обычным java-кодом. Даже на первый взгляд EJB-код достаточно сложен, так для реализации каждого session bean, Вы должны написать 3-4 класса на каждый entity bean, кроме того для снижения трафика Вам может понадобиться реализовать дополнительные объекты-значения (value objects).

4. Потенциальная угроза переусложнения решения. Этот недостаток тесно связан со сложностью спецификации. Имеется в виду, если нет понимания в достаточной степени идеологию EJB, то высока вероятность неудачи в эффективном использовании и скорее всего переусложнение проекта.

5. Так же стоит отметить одним из недостатков громоздкий xml файл для конфигурации и EJB бинов [21].

Достоинства Spring с надстройкой Spring Boot:

1. Первым достоинством можно отметить настройку и конфигурирование Spring Boot проекта. Основной и первоначальный класс наследуется от `SpringBootServletInitializer`, данный класс инициализирует конфигурацию (Подобие `web.xml`) для того чтобы веб сервер понимал, что это веб приложение. Так же в данный пункт входит `application.properties`, это настройщик, который заменяет настройку проекта в `xml`.

2. Использование Фреймворков спринга для устранения шаблонного кода, например, `spring-data-jpa`, который сгенерирует код для доступа к БД. Мы пишем интерфейс, наследуемся от `JpaRepository`(Интерфейс) или `CrudRepository`, и спринг, когда собирается, видя эти интерфейсы, генерирует `java` класс, для доступа к БД.

3. Есть возможность построения приложения на `java` коде, не используя `xml`. Это означает, что, подключая библиотеку спринг MVC, мы можем использовать аннотации вместо описания компонента(бина) в `xml`.

4. На данный момент Spring используют много, поскольку его изучение и использование проще.

5. Простое тестирование `junit` (Анотация `@Test`. Сравнение ожидания и реально полученных результатов). Поскольку это обыкновенный `Java` класс, даже без использования аннотации существует возможность тестирования.

6. Одним из ключевых достоинств является возможность использования обычных `JavaEE` бинов. Например, те же `EJB`.

Основной недостаток Spring со Spring Boot'ом – это относительно новая технология, по этой причине практических примеров мало, а большинство литературы на английском языке, что делает его изучение более затруднительным.

Несмотря на все выше приведенные данные необходимо понять, насколько эффективны оба фреймворка для системы управления контентом. Для этого было написано два стандартных запроса на передачу данных по пациенту и их серий. После чего было проведено тестирование и замерено

время, за которое была совершена передача серий от 100 снимков до 400 при помощи каждого фреймворка. Условиями тестирования были:

1. Формат: jpg;
2. Размер 1-го снимка: 55кб;
3. Скорость передачи данных сети: 1 Мб/сек;
4. Размер серий: от 100 до 1500 с шагом 100.

В качестве критерия оценки выступало время отображения серии полностью.

Ниже в таблице 7 приведены результаты для JavaEE, а в таблице 8 приведены результаты для Spring.

Таблица 7 – результаты тестирования JavaEE

Количество снимков:	Время отображения в секундах:
100	11.2
200	22.4
300	33.6
400	44.6
500	56.1
600	67.15
700	78.3
800	89.5
900	100.8
1000	111.8
1100	123.2
1200	134.3
1300	145.5
1400	156.8
1500	168

Таблица 8 – результаты тестирования Spring

Количество снимков:	Время отображения в секундах:
100	9.5
200	19
300	28.5
400	38.2
500	47.4
600	57.1
700	66.5
800	76.3
900	85.5
1000	95.1
1100	104.15
1200	114
1300	123.2
1400	133
1500	142.7

После того, как были получены статистические данные, проведем интерполяцию с помощью кубического сплайна, это делается для того, чтобы построить график и спрогнозировать при помощи линии тренда величину, с которой будет расти время, зависящее от количества снимков.

Кубический сплайн – это один из видов интерполяции сплайнами.

Интерполяция – является способом нахождения промежуточных значений некоторой величины по имеющемуся набору данных. Восстановление той или иной функции f на заданном отрезке $[a, b]$ по таблице чисел (x_i, f_i) , $i = 0, \dots, n$, где $f_i = f(x_i)$, а точки x_i образуют упорядоченную последовательность $a = x_0 < \dots < x_n = b$, является типичной задачей интерполяции.

Интерполяционный кубический сплайн – это функция, которая:

1. проходит через все заданные n точек исходных данных;
2. на каждом отрезке между соседними точками является кубическим полиномом вида:

$$S_i = a_i + b_i * x - x_i + c_i * x - x_i^2 + d_i * (x - x_i)^3 \quad (1)$$

3. непрерывна вместе со своими первой и второй производной.

Где a_i, b_i, c_i, d_i – коэффициенты сплайна.

Для нахождения коэффициентов сплайна их необходимо рассчитать по следующим формулам.

$$a_i = f_i \quad (2)$$

$$\delta_1 = - \frac{h_2}{2 * h_1 + h_2}, \quad (3)$$

$$\lambda_1 = \frac{3 * l_2 - l_1}{2 * h_1 + h_2}, \quad (4)$$

$$\delta_{i-1} = \frac{-h_i}{2 * h_{i-1} + 2 * h_i + h_{i-1} * \delta_{i-2}}, i = 3, \dots, N, \quad (5)$$

$$\lambda_{i-1} = \frac{3 * l_i - 3 * l_{i-1} - h_{i-1} * \lambda_{i-2}}{2 * h_{i-1} + 2 * h_i + h_{i-1} * \delta_{i-2}}, i = 3, \dots, N, \quad (6)$$

$$c_{i-1} = \delta_{i-1} * c_i + \lambda_{i-1}, i = N, N - 1, \dots, 2, \quad (7)$$

$$c_0 = 0, \quad (8)$$

$$c_n = 0, \quad (9)$$

$$b_i = l_i + \frac{2 * c_i * h_i + h_i * c_{i-1}}{3}, i = 1, \dots, N, \quad (10)$$

$$d_i = \frac{c_i - c_{i-1}}{3 * h_i}, i = 1, \dots, N. \quad (11)$$

Используя данные формулы и пакет прикладных программ MATLAB, выполним интерполяцию с помощью кубического сплайна, используя статистические данные из таблиц 7 и 8.

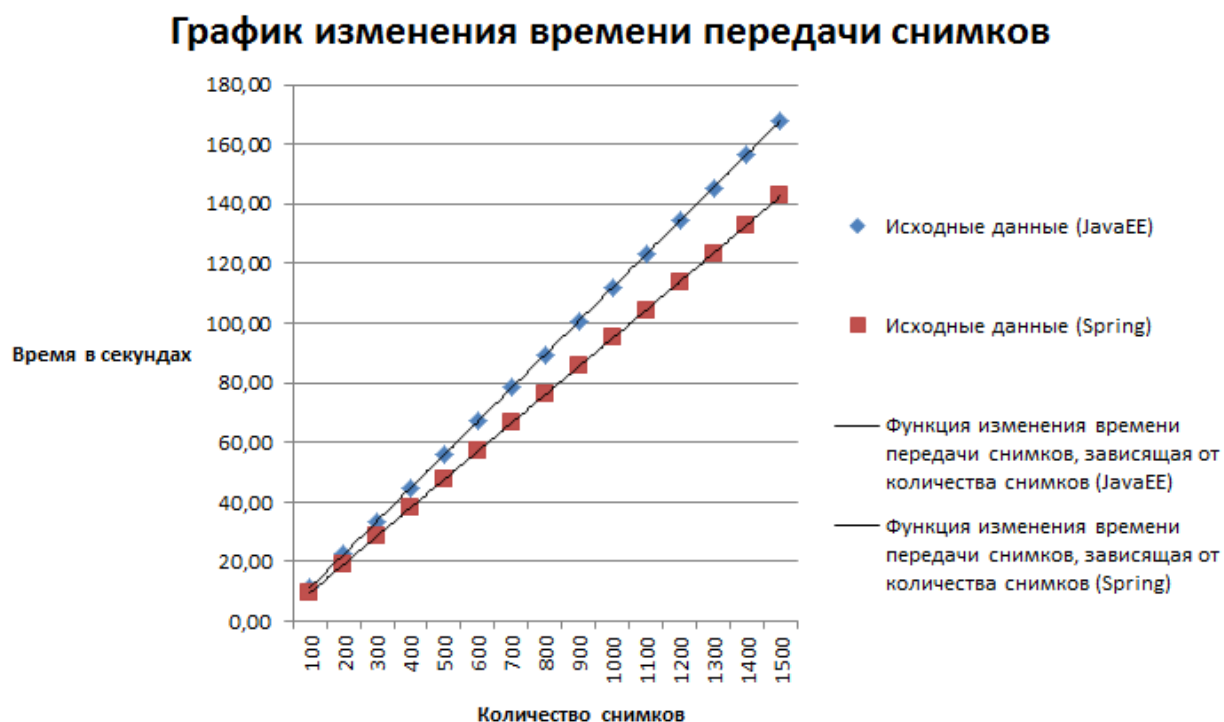


Рисунок 7 – Графики кубических сплайнов

На рисунке 7 построены два кубических сплайна, точки, отмеченные кругами, принадлежат сплайну, построенному на основе данных из таблицы 7, а второй сплайн на основе данных из 8-й таблицы.

Как можно заметить из рисунка 7, графики обоих сплайнов схожи, оба из них представляют из себя растущие тенденции и, исходя лишь из данного графика, невозможно определить точно какой из двух фреймворков лучше и насколько.

Как можно заметить из графика, представленного на рисунке 7, тренд, представленный кубическим, позволяет оценить динамику роста времени при увеличении количества снимков в серии. Можно заметить, что динамика роста графика основанного на данных таблицы 8, ниже, а, следовательно, увеличение времени с увеличением количества снимков в серии ниже. Основываясь на выше перечисленных недостатках и достоинствах, а так же на проведенном сравнительном анализе, было принято решение использовать Spring с надстройкой Spring Boot.

3.2 Реализация первой версии системы с использованием библиотеки DICOM

Изначально это было приложение без дополнительной системы управления контентом, оно предполагало отображение обезличенных снимков пациентов в формате JPEG.

Принцип работы данного приложения был следующим, вводился уникальный идентификатор пациента, который присваивается базой данных, после вводится необходимая серия снимков у данного пациента и в конце отображение снимков. В данном приложении было несколько проблем, одна из которых, это загрузка всех серий снимков пациента, но отображение лишь необходимой, а вторая вытекающая из первой, это время отображения. Под временем отображения подразумевается время с начала загрузки серии и до полного отображения серии. Из-за разного количества снимков, их качества и объема занимаемой памяти, в сериях с разных диагностических аппаратов, то для определения производительности и коэффициента полезного действия были выбраны снимки, сделанные с МРТ. Поскольку количество снимков с МРТ в среднем их число от 20 до 500, то для чистоты исследования были выбраны именно снимки с МРТ. К примеру, количество рентгеновских снимков может равняться двум или одному, но из-за объема занимаемой памяти время, требуемое на отображения снимков, равнялось времени отображения 100 снимков МРТ и т.к. количество снимков мало, то это не позволяет сделать сравнение и анализ версий систем.

Ниже приведена таблица 9 отображающая количество снимков и время, требуемое на их отображение самым первым приложением без использования системы управления контентом и с библиотекой DicomDroid.

Таблица 9 – Время отображения от 100 до 400 снимков

Количество снимков	100	200	300	400
Время отображения снимков	337.7секунд	600 секунд	944.3 секунд	1313.1 секунд

Можно заметить, что время, требуемое для отображения серии из 400 снимков большое и неприемлемо для приложения, которое предназначено для оперативного предоставления мобильного доступа к данным по пациентам.

3.3 Реализация системы управления контентом с использованием фреймворка Spring с надстройкой Spring Boot

Spring Framework – это универсальный фреймворк с открытым исходным кодом, он обеспечивает решение многих задач, с которыми сталкиваются разработчики и организации, создающие информационную систему. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. Spring не всецело связан с платформой Java Enterprise, несмотря на его масштабную интеграцию с ней, что является важной причиной его популярности.

Spring Boot – это надстройка, позволяющая не только избавиться от конфигурирования через xml, но и создание полноценных Spring приложений, встроенный Tomcat или Jetty (не требуется установки WAR файлов), обеспечивает 'начальные' POMs для упрощения Maven конфигурации, автоматическая конфигурация Spring когда это возможно, а также обеспечивает такими возможностями, как метрики, мониторинг состояниями и расширенная конфигурация [28].

Чтобы проанализировать изменения внесенные с внедрением системы управления контентом сравним время за которое отображает данные по пациентам первая версия программы и существующая на данный момент новая система с системой управления контентом написанная при помощи

фреймворка Spring с надстройкой Spring Boot. Данные по времени на отображение данных в новой системе указаны в таблице 8, а данные по времени в первой версии приложения в таблице 9. Анализируя представленные данные и, за основу взяв максимальное представленное время, можно сделать вывод, что новая система с использованием системы управления контентом выполняет отображение данных по пациентам в 32 раза, что является значительной разницей между существующей новой системой и первой версией приложения.

Разработанная новая система управления контентом для медицинского приложения позволила ускорить процесс предоставления данных по пациентам врачу и на данный момент удовлетворяет всем функциональным возможностям и требованиям.

ЗАКЛЮЧЕНИЕ

В ходе проведения работы была исследована существующая система обработки данных по пациентам в учреждениях здравоохранения на примере ТГКБ №5. Так же были выделены следующие недостатки существующей системы:

1. В реальных условиях, когда поступает большой поток посетителей, существует не малая вероятность утери документов.

2. Полная стационарность, привязывающая врача к своему АРМ.

На основе анализа были выделены 6 функциональных требований, наиболее важные из них:

1. Передача серий снимков по пациентам.

2. Первичный поиск данных по пациенту с критериями поиска (Дата поступления, вид исследования).

3. Поиск конкретного пациента из списка по ФИО.

Была разработана диаграмма прецедентов новой технологии. Был проведен сравнительный анализ двух фреймворков путем сравнения достоинств и построения кубического сплайна, с дальнейшей оценкой графиков, с помощью линии тренда. В результате сравнительного анализа были выделены следующие средства реализации: язык программирования – Java, фреймворк – Spring с надстройкой Spring Boot. Была разработана программная архитектура, включающая в себя 7 пакетов: server, config, controller, service, entity, repository, util. Так же пакеты представлены в виде диаграммы пакетов. Для классов входящих в пакет server была разработана диаграмма классов.

Была разработана система управления контентом и ее диаграмма последовательности для определения поведения системы. Так же было проведено сравнение между первоначальной версией и текущей, для оценки результатов разработки.

Таким образом, была достигнута цель ВКР, была разработана система управления контентом передающая данные по пациентам мобильному клиенту, с возможностью расширения и предобработкой данных. Так же данная система уже тестируется в медицинском учреждении ТГКБ №5.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно- ориентированный анализ и проектирование / Д. Арлоу. – 2-е изд. (пер. с англ.) – СПб: Символ-Плюс, 2014. – 624 с.
2. Гонсалвес Э. Изучаем Java EE 7: учеб. пособие / Э. Гонсалвес – СПб.: Питер, 2014. – 640 с.
3. Уоллс К. Spring в действии: учеб. пособие / К. Уоллс – М.: ДМК Пресс, 2015. – 754 с.
4. Шефер К., Хо К., Харроп Р. Spring 4 для профессионалов: учеб. пособие / К. Шефер, К. Хо. Р. Харроп – Киев: Вильямс, 2017. – 752 с.
5. Эккмель Б. Философия Java: учеб. пособие / Б. Эккмель – СПб.: Питер, 2014. – 640 с.

Электронные ресурсы

6. Автоматизированное рабочее место врача-специалиста [Электронный ресурс]. – Режим доступа: <http://medic.studio/tehnologii-meditisine-informatsionnyie/avtomatizirovannoe-rabochee-mesto-vracha-58981.html> (дата обращения: 08.03.2018).
7. ГБУЗ СО «Тольяттинская городская клиническая больница №5» - Наша больница [Электронный ресурс]. – Режим доступа: <http://tgkb5.ru/index.php/about> (дата обращения: 01.04.2018).
8. Концепция: Требования [Электронный ресурс]. – Режим доступа: http://dit.isuct.ru/Publish_RUP/core.base_rup/guidances/concepts/requirements_62E28784.html#Interface%20Requirement (дата обращения: 04.04.2018).
9. Курс-АС1 – DICOM Стандарт [Электронный ресурс]. – Режим доступа: <http://www.course-as.ru/dicomdoc.html> (дата обращения: 25.03.2018).
10. Линия тренда – ее виды и способы построения [Электронный ресурс]. – Режим доступа: <http://stock-list.ru/trend-line.html> (дата обращения 15.05.2018)

11. Лекция 5: Жизненный цикл программных систем [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/3632/874/lecture/14297?page=5> (дата обращения: 05.04.2018).
12. МЕДИЦИНСКИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ: СОСТОЯНИЕ, УРОВЕНЬ ИСПОЛЬЗОВАНИЯ И ТЕНДЕНЦИИ [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/meditsinskie-informatsionnye-sistemy-sostoyanie-uroven-ispolzovaniya-i-tendentsii> (дата обращения: 03.10.2018).
13. Программа eFilm: инструкция по использованию [Электронный ресурс]. – Режим доступа: <https://www.mrtspb.ru/articles/programma-efilm-instrukciya-po-ispolzovaniyu/> (дата обращения: 01.04.2018).
14. Порядок организации работы регистратуры поликлиники [Электронный ресурс]. – Режим доступа: http://minzdrav.kcrb/Pages/or_rabory_registratury.aspx (дата обращения: 20.03.2018).
15. Переход от каскадной разработки к итеративной [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/fset.asp?Url=/rational/transform.htm> (дата обращения: 03.04.2018).
16. Руководство по языку C# [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp> (дата обращения: 15.05.2018)
17. Что такое тенденция и ее основные характеристики [Электронный ресурс]. – Режим доступа: <http://stock-list.ru/trend.html> (дата обращения: 17.05.2018)
18. 65. Регистратура поликлиники, структура, задачи [Электронный ресурс]. – Режим доступа: <http://uchenie.net/65-registratura-polikliniki-struktura-zadachi/> (дата обращения: 20.03.2018).

19. EJB-ить или не EJB-ить, вот в чем вопрос [Электронный ресурс]. – Режим доступа: http://www.javaportal.ru/java/articles/toejbornottoejb.html#_Тoc532991434 (дата обращения 20.05.2018)

Литература на иностранном языке

20. About python [Электронный ресурс]. – Режим доступа: <https://www.python.org/about> (дата обращения 30.03.2018)

21. Achilleus A. J2EE Architecture - A Review [Text]/ A. Achilleus // International Journal of Scientific & Technology Research, India (<http://www.ijstr.org/final-print/july2017/J2ee-Architecture-A-Review.pdf>)

22. Bucea-Manea-Tonis R., Bucea-Manea-Tonis R. How to Design a Web Survey Using Spring Boot with MYSQL: A Romanian Network Case Study [Text]/ R. Bucea-Manea-Tonis, R. Bucea-Manea-Tonis // Annals of Spiru Haret University (<http://anale.spiruharet.ro/index.php/economics/article/view/1725/pdf>)

23. Capturing Architectural Requirements [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/developerworks/rational/library/4706.html> (дата обращения: 04.04.2018)

24. Documentation – WildFly 10 [Электронный ресурс]. – Режим доступа: https://docs.jboss.org/author/display/WFLY10/Documentation?_sscc=t (дата обращения 05.03.2018)

25. Feng L. The Current Role of Image Compression Standards in Medical Imaging [Text]/ L. Feng // College of Electronic Information and Optical Engineering, Nankai University, Haihe Education Park, 38 Tongyan Road, Jinnan District, Tianjin 300350, China (<http://www.mdpi.com/2078-2489/8/4/131/htm>)

26. Java EE Documentation [Электронный ресурс]. – Режим доступа: <http://www.oracle.com/technetwork/java/javaee/documentation/index.html> (дата обращения 01.03.2018)

27. RadiAnt DICOM Viewer [Электронный ресурс]. – Режим доступа: <https://www.radiantviewer.com/en/> (дата обращения 16.05.2018)

28. Spring Boot [Электронный ресурс]. – Режим доступа: <https://projects.spring.io/spring-boot> (дата обращения 18.05.2018)
29. The Current Role of Image Compression Standards in Medical Imaging [Электронный ресурс]. – Режим доступа: <http://www.mdpi.com/2078-2489/8/4/131/htm> (дата обращения: 09.12.2017).
30. Spring Framework [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-framework> (дата обращения 03.04.2018)