

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.91;004.04

Рылеев
Евгений Константинович

Автоматизированная информационная система документооборота в
стоматологических учреждениях здравоохранения

АВТОРЕФЕРАТ

на соискание академической степени
магистра технических наук

по специальности 1-40 80 05 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Научный руководитель
Таборовец В. В.
к.т.н., доцент

Минск 2019

КРАТКОЕ ВВЕДЕНИЕ

Заинтересовавшись внутренним устройством стоматологий и выявив недостатки бумажного ведения документооборота, был произведен поиск подобных систем для стоматологий. Однако, такой поиск не дал действительно успешных результатов, так как он приводил или к информации о таких системах для Российской Федерации, что явно не подходит для документооборота в Республике Беларусь, либо к невероятно перегруженным функциональностью системам.

В связи с этим было решено разработать систему для автоматизации документооборота, а для удобства в использовании представить её в виде веб-приложения.

Предметной областью диссертации стали: документооборот, учет и здравоохранение. Целью – разработка инструмента для автоматизации документооборота в стоматологических учреждениях при выдвинутой гипотезе, что полученная информационная система упростит ведение документооборота в стоматологиях, а также станет наиболее подходящей системой для использования в малых учреждениях здравоохранения.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Цель работы состоит в проектировании и разработке автоматизированной информационной системы документооборота в малых стоматологических учреждениях здравоохранения.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследовать документооборот в стоматологическом учреждении.
2. Разработать алгоритмы автоматизации работы системы.
3. Спроектировать архитектуру автоматизированной информационной системы документооборота.
4. Реализовать автоматизированную информационную систему документооборота на основе результатов проектирования.
5. Провести тестирование полученной автоматизированной информационной системы документооборота.

Объектом исследования являются автоматизированные системы документооборот, учет и здравоохранение.

Предметом исследования является программное обеспечения автоматизации документооборота в стоматологических учреждениях здравоохранения.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии с научно-техническим заданием и планом работ кафедры «Программное обеспечение информационных технологий» по теме «Разработка моделей, методов, алгоритмов, повышающих показатели проектирования, внедрения и эксплуатации программных средств для перспективных платформ обработки информации, решения интеллектуальных задач, работы с большими массивами данных и внедрение в современные обучающие комплексы» (ГБ № 16-2004, № ГР 20163588, научный руководитель НИР – Н. В. Лапицкая).

Личный вклад соискателя

Результаты, приведенные в диссертации: анализ, проектирование, разработка программного средства, – получены соискателем лично. Вклад научного руководителя В.В. Таборовца, заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались и обсуждались на 54-й научной конференции аспирантов, магистрантов и студентов, проводимых в БГУИР.

Опубликованность результатов диссертации

По теме работы было опубликовано 2 печатных работы. Одна работа была опубликована в научно-методическом журнале «Проблемы современной науки и образования»; одна работа опубликована в виде тезисов в материалах 54-й научной конференции аспирантов, магистрантов и студентов, проводимых в БГУИР.

Структура и объем диссертации

Диссертация состоит из введения, общей характеристики работы, четырех глав, заключения, списка использованных источников, списка публикаций автора и приложений. В первой главе представлен анализ предметной области, выявлены основные существующие проблемы в рамках тематики исследования, показаны направления их решения. Вторая глава посвящена разработке архитектуры информационной системы и алгоритмов для автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения. В третьей главе предложена практическая реализация автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения. В четвертой главе проведено тестирование системы, как модульное, интеграционное так и нагрузочное с использованием JMeter.

Общий объем работы составляет 67 страниц, из которых основного текста 53 страницы, 22 рисунка, 1 таблица, список использованных источников из 33 наименований и 2 приложения на 8 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и указаны основные направления исследования, показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов, обозначена практическая ценность работы.

В **первой главе** проведен обзор существующих систем и предметной области. Определены недостатки SAP и 1С систем. Проведен обзор аналога, для которого были выявлены плюсы и недостатки по сравнению с вышеуказанными системами.

Был проведен анализ организации документооборота в стоматологических учреждениях здравоохранения. Все стоматологические услуги делятся на шесть основных видов: терапия, эндодонтическое, пародонтическое, хирургическое, ортопедическое лечения. Было определено, что как правило, большинство стоматологических клиник предоставляют терапевтические услуги и некоторые и остальных перечисленных. Также определились основные документы и роли в стоматологии.

На основе вышеперечисленного, были выявлены следующие требования к системе:

1. Система должна поддерживать документооборот основных видов документов в стоматологический учреждениях здравоохранения.
2. Система должна предоставлять возможность ведения журнала посещения.
3. Система должна предоставлять возможность ведения учета материалов.
4. Система должна предоставлять возможность генерации документов.
5. Система должна предоставлять возможность задавать набор предоставляемых услуг.
6. Система должна предоставлять возможность администрирования персонала.

Далее проведено сравнение монолитной и микросервисной архитектур, в результате чего было выявлено что микросервисная архитектура обладает целым рядом преимуществ, которые можно использовать в системе. Независимость развёртки и независимость технического стека, позволит реализовать систему команде разработчиков независимо друг от друга и постоянно расширять функциональность.

Вторая глава посвящена проектированию системы. В данном разделе поэтапно рассмотрено проектирование системы. Разработаны основные алгоритмы автоматизации работы системы, определены основные функции системы, спроектирована архитектура системы и баз данных.

В качестве основных были определены следующие алгоритмов автоматизации: добавление записи об оказании услуги, алгоритм добавления новой услуги и алгоритм создание документов. Основные функции дублируют требования к системе.

Во время проектирования архитектуры были определены основные компоненты системы.

Архитектура информационной системы представляет собой несколько микросервисов, каждый из которых выполняет определенную задачу. Так же основным связующим звеном данной системы является пользовательский интерфейс позволяющий взаимодействовать со всеми микросервисами данной информационной системы. Все данные хранятся в реляционной базе данных, в нашем случае MySQL и, для ускорения поиска, дублируются в нереляционном хранилище предоставляющем полнотекстовый поиск ElasticSearch.

Данная информационная система разворачивается в облачном сервисе AWS компании Amazon. [1]

Все микросервисы представляют собой docker-контейнеры, которые развертываются с помощью сервиса ECS (Elastic Container Service).

Сообщение между всеми микросервисами происходит с помощью REST API и очередей сообщений, под управлением сервиса уведомлений представленными в AWS сервисами SNS (simple notification service) и SQS (simple queue service).

Пользовательский интерфейс храниться в хранилище данных S3 (simple storage service), который предоставляет возможность хранения в качестве статического контента и вместе с этим генерирует единую точку входа для доступа к интерфейсу.

Реляционные хранилища данных также разворачиваются в облачном сервисе AWS с помощью сервиса RDS (relational database service). Данный сервис предоставляет возможность создания реплик хранилища для ускорения операций параллельного чтения. ElasticSearch разворачивается в рамках виртуальной машины EC2 с помощью AWS ElasticSearch service. [10]

Таким образом структура информационной системы представляет собой:

1) Микросервисы, каждый из которых реализует ту или иную часть системы. Каждый микросервис имеет одну задачу, будь то ведение учета материалов, работа с клиентами, ведение медицинских карт и так далее. В связи с этим мы имеем следующие микросервисы:

а) Микросервис для работы с клиентами. Оперирует всей информацией связанной с клиентами.

б) Микросервис для работы с медицинскими картами. Предоставляет возможность хранения, модификации и чтения медицинских записей клиента в рамках одной организации.

в) Микросервис для работы с материалами. Предоставляет возможность ведения учета материалов.

г) Микросервис для работы с путевыми листами. Предоставляет возможность создания и заполнения путевых листов.

д) Микросервис для работы с предоставляемыми услугами. Позволяет добавлять, модифицировать, удалять и вести статистику по предоставляемым услугам.

е) Микросервис для работы с персоналом. Предоставляет возможность управления персоналом, такие как добавление новых сотрудников в систему, удаление уволившихся сотрудников из системы, предоставление доступа к частям системы.

ж) Микросервис для генерации документов. Предоставляет возможность на основе данных хранящихся в системе сгенерировать разные документы, такие как медицинские карты, путевые листы и т.п., в виде электронных документов в форматах docx, pdf, excel.

2) Пользовательский интерфейс, который предоставляет возможность «общения» с микросервисами представленными выше.

3) Реляционные базы данных. Предоставляют возможность хранения данных для каждого микросервиса. Каждый микросервис имеет свою базу данных.

4) Хранилище данных для полнотекстового поиска. Предоставляют возможность более быстрого поиска данных, являются частичной репликой данных и реляционных баз данных.

SNS/SQS сервисы. Предоставляют возможность микросервисам уведомления друг друга об изменении данных. Например при заполнении и сохранении путевого листа отправляется уведомление об использовании материала и о предоставленных услугах.

В результате чего была разработана архитектурная диаграмма системы. Данная диаграмма представлена на рисунке 1.

Далее каждый из сервисов был рассмотрен подробнее и определены их структуры и схемы баз данных.

В конце был произведен выбор средств разработки. Для реализации данного проекта были выбраны следующие средства разработки:

- 1) Язык программирования – Groovy.
- 2) Spring/Spring Boot – для создания REST-сервисов и предоставления внедрения зависимостей.
- 3) Среда разработки – IntelliJ IDEA/WebStorm.
- 4) Система контроля версий – Git.
- 5) Система сборки проекта – Gradle.
- 6) Хранение информации – MySQL, ElasticSearch, S3.
- 7) Angular 2+ – для разработки пользовательского интерфейса.
- 8) AWS – облачная платформа.
- 9) Flyway – для изменения структуры базы данных.

Данный выбор обуславливается относительной простотой использования в связи со знакомством с данными средствами разработки.

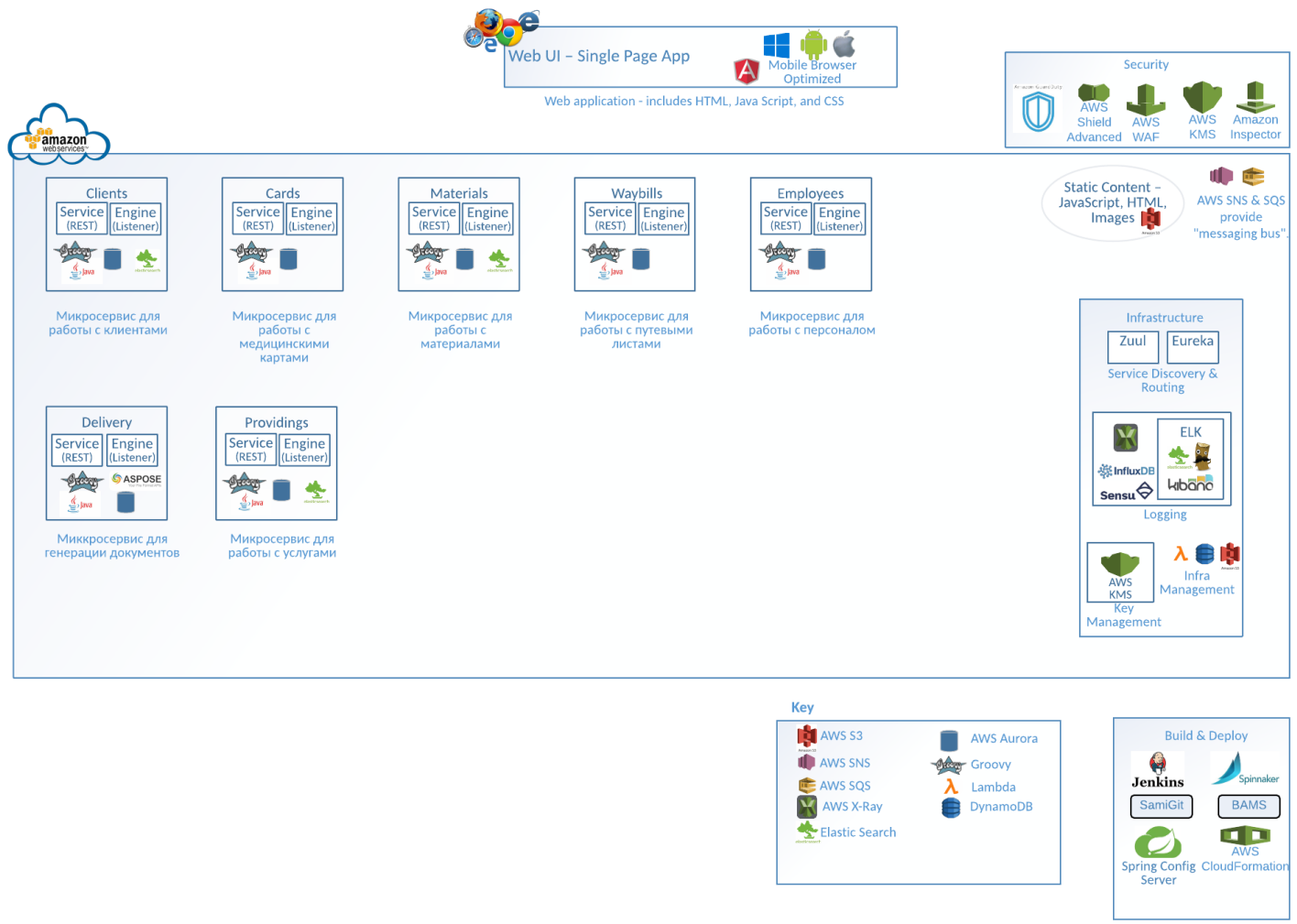


Рисунок 1 - архитектура информационной системы.

В **третьей главе** проиллюстрирован этап разработки системы. Были произведены: настройка RDS, создан DDL-код и данный код был применен к RDS.

Настройка RDS производится с использованием сервиса CloudFormation от AWS. AWS CloudFormation – это сервис, который помогает моделировать и настраивать ресурсы веб-сервисов Amazon, чтобы оптимизировать управление ресурсами и уделять больше внимания приложениям, которые работают в AWS. Для этого создается шаблон, который описывает все необходимые ресурсы AWS (например, экземпляры Amazon EC2 или Amazon RDS DB), а AWS CloudFormation предоставляет и настраивает данные ресурсы. Данный сервис предоставляет возможность избавиться от необходимости создания и настройки каждого ресурса отдельно.

Далее были изложены основные этапы разработки серверной и клиентской части, в частности описаны разработка трех уровней (уровень сообщения между приложением и базой данных, уровень, сервисов и уровень контроллеров) серверной части, был описан метод аутентификации и защиты системы. При описании клиентской части была описана архитектура клиентского приложения. При разработке клиентской части с использованием Angular 7 приложение имеет вид «матрешки»: есть главный компонент (app) в который помещаются другие компоненты, в которые в свою очередь помещаются третьи и так далее. Таким образом шаблон приложения имеет вид как на рисунке 2.

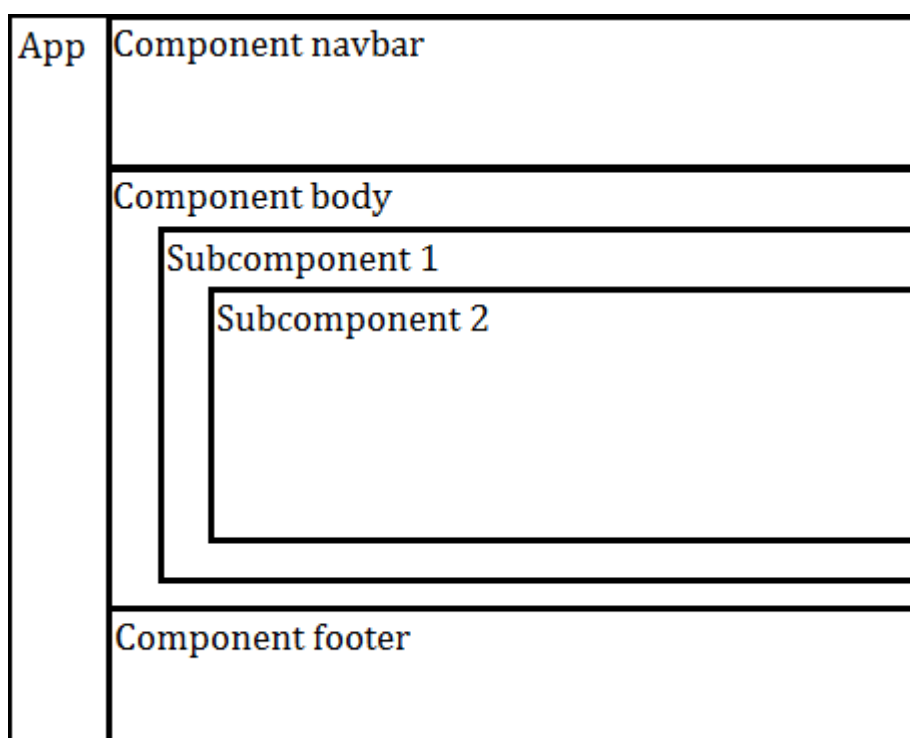


Рисунок 2 – шаблон построения клиентской части.

В **четвертой главе** были показаны методы и результаты тестирования системы. Для тестирования данного приложения были использованы следующие фреймворки: тестирование серверной части – JUnit, Mockito, Cucumber, тестирование клиентской части: Jasmin, Karma.

Тестирование серверной и клиентской части заключается в написании unit-тестов, то есть тестов отдельных частей приложения и функциональных тестов, выполняющихся вручную.

Каждый unit-тест представляет собой тестирование какой-то определенной функциональности, обособлено от ее окружения. Так, тестирование контроллеров заключается в вызове точки входа, за которую отвечает каждый метод данного контроллера, подстановке результата, получающегося от сервиса и проверке правильности вернувшегося результата от контроллера.

Таким образом, каждый тест делится на 3 части:

- а) подготовка результатов;
- б) вызов точки входа;
- в) проверка вернувшегося результата.

Аналогична ситуация с сервисами и репозиториями, только вместо вызова точки входа происходит вызов метода класса.

В случае тестирования клиентской части, единственным отличием является используемый язык программирования, в остальном же, тестирование проходит по тому же сценарию – подготовка входных данных, вызов метода, проверка вернувшихся результатов.

Также было произведено нагрузочное тестирование имитирующее десятерых пользователей.

ЗАКЛЮЧЕНИЕ

Основные результаты научной диссертации

1. Предложена архитектура системы для решения задач автоматизации документооборота в стоматологических учреждениях здравоохранения. Данная архитектура представляет собой набор микросервисов, баз данных (реляционных и документарных) и клиентской части. Сообщение между микросервисами и клиентской части осуществляются путем выполнения HTTP-запросов, сообщение между микросервисами и базами данных осуществленно при помощи SSL-соединения.

2. Разработана информационная система документооборота в стоматологических учреждениях здравоохранения. Система была реализована с использованием микросервисной архитектуры. Она способна производить манипуляцию с основными документами в стоматологиях, такими как: карта клиента, регистрационный журнал, путевые листы, накладные по материалам.

1. Система поддерживает документооборот основных видов документов в стоматологический учреждениях здравоохранения.

- а) Медицинская карта.
 - б) Путевой лист.
 - в) Накладные на материалы.
2. Система предоставляет возможность ведения журнала посещения.
 3. Система предоставляет возможность ведения учета материалов.
 4. Система предоставляет возможность генерации документов.
 - а) Журнал посещения за период (день, месяц, квартал полгода, год).
 - б) Медицинская карта клиента.
 - в) Путевой лист.
 - г) Упрощенный документ расходов материалов.
 5. Система предоставляет возможность задавать набор предоставляемых услуг.
 6. Система предоставляет возможность администрирования персонала.
 - а) Учет сотрудников в системе.
 - б) Задание ролей сотрудников в системе.
 - в) Предоставление возможности пользования системой сотрудникам учреждения.

Для реализации данного проекта были использованы следующие средства разработки:

1. Язык программирования – Groovy.
2. Spring/Spring Boot – для создания REST-сервисов и предоставления внедрения зависимостей.
3. Среда разработки – IntelliJ IDEA/WebStorm.
4. Система контроля версий – Git.
5. Система сборки проекта – Gradle.
6. Хранение информации – MySQL, ElasticSearch, S3.
7. Angular 7 – для разработки пользовательского интерфейса.
8. AWS – облачная платформа.
9. Flyway – для изменения структуры базы данных.

Рекомендации по практическому использованию результатов

1. Полученные результаты формируют теоретическую и практическую базу для разработки автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения. Они могут быть использованы для модернизации и дальнейшего развития существующих систем.

2. Результаты работы могут быть использованы в стоматологических учреждениях здравоохранения для осуществления документооборота и ведения регистрационного журнала клиентов.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1-А. Таборовец В. В., Рылеев Е. К. AWS сервисы при разработке программного обеспечения электронного документооборота / Е. К. Рылеев // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов (Минск, 23 – 27 апреля 2018 г.). – Минск: БГУИР, 2018.

2-А. Таборовец В. В., Рылеев Е. К. Автоматизированная информационная система документооборота в стоматологических учреждениях здравоохранения / Таборовец В. В., Рылеев Е. К. // Проблемы современной науки и образования 2019, 63-67.