

# АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДОКУМЕНТООБОРОТА В СТОМАТОЛОГИЧЕСКИХ УЧРЕЖДЕНИЯХ ЗДРАВООХРАНЕНИЯ

Таборовец В.В.<sup>1</sup>, Рылеев Е.К.<sup>2</sup> Email: Taborovets17139@scientifictext.ru

<sup>1</sup>Таборовец Вячеслав Васильевич – кандидат технических наук, доцент;

<sup>2</sup>Рылеев Евгений Константинович – магистрант,

кафедра программного обеспечения информационных технологий, факультет компьютерных сетей и систем,  
Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь

**Аннотация:** статья посвящена проектированию автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения. Перечислены основные функции системы, сравнены монолитная и микросервисная архитектура, сделаны выводы по сравнению и выбор архитектуры. Результаты исследований использованы при создании автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения.

**Ключевые слова:** монолитная и микросервисная архитектуры, информационная система, документооборот, автоматизация, расширяемость. Результаты исследований использованы при создании автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения.

## AUTOMATED INFORMATION SYSTEM OF DOCUMENT CIRCULATION IN DENTAL INSTITUTIONS

Taborovets V.V.<sup>1</sup>, Ryleyeu E.K.<sup>2</sup>

<sup>1</sup>Taborovets Vjacheslav Vasilyevich – PhD in Techniques, Associate Professor;

<sup>2</sup>Ryleyeu Evgeny Konstantinovich – Master's degree,

DEPARTMENT SOFTWARE FOR INFORMATION TECHNOLOGIES, FACULTY OF COMPUTER SYSTEMS AND  
NETWORKS,  
BELARUSIAN STATE UNIVERSITY OF INFORMATICS AND RADIOELECTRONICS,  
MINSK, REPUBLIC OF BELARUS

**Abstract:** the article is devoted to the design of automated information system of document circulation in dental institution. The main functions of the system are listed, the monolithic and microservice architecture are compared, conclusions are made in comparison and the choice of architecture. The research results were used to create an automated information system for document flow in dental health care institutions.

**Keywords:** monolithic and microservice architecture, information system, workflow, automation, extensibility. The research results were used to create an automated information system for document flow in dental health care institutions.

УДК 004.91;004.04

В современном мире, любое действие, приводящее к контакту с обществом, не обходится без сопутствующего документа, будь то поход к врачу (здесь в качестве документов выступают карточки, талоны, рецепты и т.д.), поход в магазин (чек) или поход в кино (билет).

Поэтому, во всех современных предприятиях накапливается большое количество бумаг, которое необходимо хранить, а также часто производить поиск какой-либо информации по ним. Но при достаточно крупных объемах фирмы, времени их существования, становится затруднительным оба этих действия, т.к. место в шкафах и кабинетах ограничено, а поиск по тоннам бумаг можно производить вечно.

Не обошло стороной данное явление и стоматологии. Здесь необходимо хранить информацию о: клиентах, т.е. их медицинские карты; материалах, используемых в работе; проделанных работах. Так же необходимо вести ежедневный учет посещений и проведенных работ для каждого стоматолога, вести журнал посещений, отчитываться перед надзирающими инстанциями о потраченных материалах, выплаченных зарплатах и т.п.

В связи с этим стали разрабатываться различные базы данных и системы управления базами данных для хранения и управления информацией, а также системы автоматизации документооборота для удобного доступа к информации, хранящейся в базах данных.

Такие системы автоматизации документооборота стали внедряться повсеместно и существенно упростили не только ведение документооборота, но и его хранение, просмотр, поиск нужной информации. Таким образом, программные средства автоматизации документа оборота не только упрощают все связанное с документами, но и уменьшают используемое место на хранение всех этих бумаг. Также, при помощи таких систем намного проще строить отчеты за различные периоды и передавать их контролирующим организациям.

Вдобавок ко всему, такие программные средства можно интегрировать со средствами онлайн-записи клиентов, что еще больше уменьшит работу регистраторов и бухгалтеров, так как им не придется вписывать эту информацию вручную.

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

В первую очередь, перед созданием системы необходимо определиться с ее основной функциональностью и определить ее архитектуру.

Для определения функциональности, было решено ознакомиться с основными ролями и документами в стоматологиях. Итак, стоматология – раздел медицины, занимающийся изучением зубов, их строения и функционирования, их заболеваний, методов их профилактики и лечения, а также болезней полости рта, челюстей и пограничных областей лица и шеи.

Все стоматологические услуги делятся на шесть основных видов: терапия, эндодонтическое, пародонтическое, хирургическое, ортопедическое лечения.

Как правило, большинство стоматологических клиник предоставляют терапевтические услуги и некоторые и остальных перечисленных.

Для большего понимания предметной области, определим каждый из видов услуг:

а) Терапия – самый распространенный вид услуг представляет собой лечение кариеса и его осложнений и реставрацию зубов.

б) Эндодонтическое лечение – это лечение зубных каналов, а также кости, окружающей верхушку корня.

в) Пародонтология – это вид лечения и профилактики околозубных тканей и их патологий. К пародонту относят все органы и ткани, которые размещены вокруг зуба: десна, костная ткань, в которой расположен корень зуба и связочный аппарат зуба [1].

г) Хирургическое лечение – это, как правило, имплантация или удаление зуба.

д) Ортопедическое лечение – это раздел стоматологии, посвященный диагностике и лечению нарушений целостности и функции зубочелюстной системы путем протезирования или установки регулирующих аппаратов [1].

В стоматологических клиниках, как правило, выделяют следующие роли:

а) Регистратор. Основная функция регистратора – запись клиентов на прием, также прием оплаты лечения.

б) Медсестра. Основные функции – обработка инструментов после использования, подготовка материалов, помощь в лечении клиента и ведение путевого листа пациента.

в) Бухгалтер. Основная функция – ведение бухгалтерии: учет материалов, подсчет затрат, прибыли, налоговых отчислений и т.д.

г) Врач. Основная функция – лечение пациентов, а также ведение их медицинских карт.

Как и в любой другой организации в стоматологии есть перечень документов, необходимых для функционирования предприятия. В нашем приложении основными такими документами являются:

а) Медицинская карта (хранит информацию о проведенных лечениях, осмотрах, историю болезней и т.п.).

б) Путевой лист (хранит информацию о проведенном лечении: какие работы были проведены, какие материалы потрачены, какая проблема была и стоимость услуги).

в) Накладная на препараты (хранит информацию о закупленных препаратах).

На основе этих данных, получим следующие требования к информационной системе:

1. Система должна поддерживать документооборот основных видов документов в стоматологический учреждениях здравоохранения.

а) Медицинская карта.

б) Путевой лист.

в) Накладные на материалы.

2. Система должна предоставлять возможность ведения журнала посещения.

3. Система должна предоставлять возможность ведения учета материалов.

4. Система должна предоставлять 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 должна предоставлять возможность задавать набор предоставляемых услуг.
6. Система должна предоставлять возможность администрирования персонала.
  - а) Учет сотрудников в системе.
  - б) Задание ролей сотрудников в системе.
  - в) Предоставление возможности пользования системой сотрудникам учреждения.

Далее необходимо определиться с архитектурой системы, для этого сравним монолитную и микросервисную архитектуры.

Рассмотрим преимущество каждой из подходов, которые предлагает один из самых авторитетных людей в этой сфере проектирования приложений Мартин Фаулер [2].

Монолитная архитектура:

1) Простота. Монолитная архитектура гораздо проще в реализации, управлении и развёртывании. Микросервисы требуют тщательного управления, поскольку они развёртываются на разных серверах и используют API.

2) Согласованность (Consistency). При монолитной архитектуре проще поддерживать согласованность кода, обрабатывать ошибки и т. д. Зато микросервисы могут полностью управляться разными командами с соблюдением разных стандартов.

3) Межмодульный рефакторинг. Единая архитектура облегчает работу в ситуациях, когда несколько модулей должны взаимодействовать между собой или когда мы хотим переместить классы из одного модуля в другой. В случае с микросервисами мы должны очень чётко определять границы модулей.

Микросервисы:

1) Частичное развёртывание. Микросервисы позволяют по мере необходимости обновлять приложение по частям. При единой архитектуре нам приходится заново развёртывать приложение целиком, что влечёт за собой куда больше рисков.

2) Доступность. У микросервисов доступность выше: даже если один из них сбоят, это не приводит к сбою всего приложения.

3) Сохранение модульности. Сохранять модульность и инкапсуляцию может быть непросто, несмотря на правила SOLID. Однако микросервисы позволяют гарантировать отсутствие общих состояний (shared state) между модулями.

4) Мультиплатформенность. Микросервисы позволяют использовать разные технологии и языки, в соответствии с вашими задачами.

Эрик Эванс использует более прагматическую оценку выделяет ряд аппаратных и программных преимуществ, которых лишена монолитная архитектура [3].

Аппаратные преимущества:

1) Независимая масштабируемость. При размещении модулей на отдельных серверных узлах мы можем масштабировать их независимо от других модулей.

2) Независимый технический стек. Благодаря распределению модулей по разным серверным узлам и независимому языку взаимодействия мы можем использовать совершенно разные языки программирования, инструменты взаимодействия, мониторинга и хранения данных. Это позволяет выбирать лучшие и наиболее удобные решения, а также экспериментировать с новыми технологиями.

Программные преимущества:

1) Сохранение модульности. И единая, и микросервисная архитектуры позволяют сохранять модульность и инкапсуляцию. Однако это может быть довольно трудной задачей, на решение которой уйдут десятилетия, несмотря на правила SOLID. Зато микросервисы позволяют обеспечивать логическое разделение приложения на модули за счёт явного физического разделения по серверам. Физическая изолированность защищает от нарушения пределов ограниченных контекстов.

2) Независимая эволюция подсистем. Микросервис может развиваться и ломать обратную совместимость, не обременяя себя поддержкой старых версий, так как всегда можно оставить старую версию микросервиса работающей необходимое время.

Таким образом, 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.

В связи с тем, что микросервисная архитектура крайне сложна для самостоятельного развертывания, было решено развертывать данную систему в облачном сервисе компании Амазон, для этого необходимо определиться с инфраструктурой 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.

Все микросервисы представляют собой docker-контейнеры, которые развертываются с помощью сервиса ECS (Elastic Container Service). [4]

Сообщение между всеми микросервисами происходит с помощью REST API и очередей сообщений, под управлением сервиса уведомлений представленными в AWS сервисами SNS (simple notification service) [5] и SQS (simple queue service) [6].

Пользовательский интерфейс хранится 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 [7].

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 [8]. ElasticSearch разворачивается в рамках виртуальной машины EC2 [9] 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. Оперировать 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 данного исследования были определены требования к автоматизированной информационной системе документооборота в стоматологических учреждениях здравоохранения, был выбран тип и была разработана архитектура системы.

Результаты данного исследования были использованы при построении автоматизированной информационной системы документооборота в стоматологических учреждениях здравоохранения.

### *Список литературы / References*

1. Стоматология / Бажанов Н.Н. Москва: Издательство Гэтоар-Мед, 2002. 297 с.
2. Микросервисная архитектура. [Электронный ресурс]. Режим доступа: <https://martinfowler.com/articles/microservices.html/> (дата обращения: 07.04.2019).
3. Микросервисная архитектура. [Электронный ресурс]. Режим доступа: <https://www.slideshare.net/InfoQ/ddd-and-microservices-at-last-some-boundaries/> (дата обращения: 07.04.2019).
4. AWS документация ECS. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/ecs/> (дата обращения: 04.04.2019).
5. AWS документация SNS. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/sns/> (дата обращения: 04.04.2019).
6. AWS документация SQS. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/sqs/> (дата обращения: 04.04.2019).
7. AWS документация S3. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/s3/> (дата обращения: 04.04.2019).
8. AWS документация RDS. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/rds/> (дата обращения: 04.04.2019).
9. AWS документация EC2. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/ec2/> (дата обращения: 04.04.2019).
10. AWS документация Elasticsearch service. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/elasticsearch-service/> (дата обращения: 04.04.2019).