ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

# «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

**Факультет программной инженерии и компьютерной техники**

Образовательная программа: Системное и прикладное программное обеспечение

Направление подготовки (специальность): 09.03.04 Программная инженерия

**Отчет по производственной, технологической (проектно-технологической) практике**

Тема задания: Разработка системы напоминаний для пациентов, находящихся на диспансерном наблюдении.

Обучающийся: Яковлев Григорий Алексеевич, P34111

Руководитель практики от профильной организации: Вахитов Линар Маратович, Владелец Продукта (Product Manager) ООО «Ланит-Терком»

Руководитель практики от университета: Маркина Татьяна Анатольевна, Доцент факультета программной инженерии и компьютерной техники

Санкт-Петербург

2024 г

СОДЕРЖАНИЕ

[ХАРАКТЕРИСТИКА ОРГАНИЗАЦИИ 3](#_Toc164861093)

[ВВЕДЕНИЕ 4](#_Toc164861094)

[1 Изучение документации и аналитика 5](#_Toc164861095)

[2 Создание нового микросервиса 6](#_Toc164861096)

[3 Разработка модели данных для аудита 7](#_Toc164861097)

[4 Реализация алгоритмов для работы с аудитом 9](#_Toc164861098)

[5 Адаптация существующих алгоритмов под новые требования 10](#_Toc164861099)

[6 Реализация обработчика напоминаний 11](#_Toc164861100)

[7 Проверка, устранение ошибок и дополнительная оптимизация 12](#_Toc164861101)

[ЗАКЛЮЧЕНИE 13](#_Toc164861102)

[СПИСОК ИСПОЛЬЗУЮЕМЫХ ИСТОЧНИКОВ 14](#_Toc164861103)

[ПРИЛОЖЕНИЕ А 15](#_Toc164861104)

# ХАРАКТЕРИСТИКА ОРГАНИЗАЦИИ

ООО «Ланит-Терком» — это ведущая российская ИТ-компания с многолетним опытом в инновационных технологических решениях и разработке программного обеспечения. Компания работает на рынке с 1991 года и является частью группы компаний ЛАНИТ. Среди клиентов компании как отечественные, так и зарубежные предприятия.

Компания реализует крупные и технологически сложные проекты, значительно улучшающие качество жизни людей. Среди них разработка социально значимых и медицинских информационных систем, а также образовательных платформ и проектов по цифровизации производственных и бизнес-процессов.

Основные направления деятельности «Ланит-Терком» включают разработку и поддержку программного обеспечения, цифровую трансформацию, реинжиниринг сложных систем, а также создание ИТ-решений для нефтегазовой отрасли, медицинского и образовательного секторов, используя передовые технологии.

# ВВЕДЕНИЕ

Тема практики: Разработка системы напоминаний для пациентов, находящихся на диспансерном наблюдении.

Цель работы: Повышение соблюдения режима лечения и профилактических мероприятий, для пациентов, находящихся на диспансерном наблюдении.

Задачи:

1. Разработка сервиса напоминаний
2. Интеграция с внешними системами

Этапы выполнения:

1. Изучение документации и аналитика
2. Создание нового микросервиса
3. Разработка модели данных для аудита
4. Реализация алгоритмов для работы с аудитом
5. Адаптация существующих алгоритмов под новые требования
6. Реализация обработчика напоминаний
7. Проверка, устранение ошибок и дополнительная оптимизация

# 1 Изучение документации и аналитика

Приступая к этапу, я начал с ознакомления с текущими процессами и архитектурой системы ЕМИАС.Обзвоны, которая взаимодействует с внешней системой оповещений. Моя задача заключалась в анализе требований, а также в формулировании вопросов для уточнения неясных моментов.

На странице с доработками в системе "Confluence" я выяснил, что процесс обзвона включает в себя напоминание о записях на консультации, лабораторные исследования (ЛИ) и инструментальные исследования (ИИ). Основываясь на предоставленной документации, стало понятно, что ЕМИАС.Обзвоны подписан на топики сообщений, из которых получает данные о записях и передает их в систему оповещения.

В доработке требовалось улучшить интеграцию системы с новыми потребностями клинического учреждения в части уведомления пациентов с активной программой диспансерного наблюдения (ПДН) о предстоящих медицинских исследованиях и консультациях. Я понял, что важно настроить систему таким образом, чтобы она могла ежедневно получать и актуализировать данные о записях пациентов с ПДН и отправлять их в ЕМИАС.Обзвоны для создания задач на обзвон.

Встретившейся проблемой было, как мне показалось, несоответствие некоторых функциональных требований технической документации касающееся оркестрации процесса обзвона. Мне пришлось уточнять требования, поскольку в документации не было четкости в описании процессов взаимодействия с ЕМИАС.ДН.Уведомления, который уведомляет об изменениях в ПДН и передает нам данные о записях.

Также мною было обнаружено, что сценарии работы с обзвонами не всегда корректно отражают изменения статусов записей, что может привести к повторным, ненужным звонкам пациентам. Для решения этой проблемы я предложил доработать механизмы отмены вызовов, связанных с переносом или отменой записей так, чтобы задачи на обзвон создавались только после получения определенного сообщения, что на практике должно повысить точность и актуальность данных для обзвонов.

Таким образом, начальный этап изучения документации и аналитики позволил мне не только уточнить требования к системе, но и предложить конкретные улучшения, которые впоследствии будут успешно внедрены для оптимизации процесса обзвонов в рамках ЕМИАС. Документация на странице была обновлена аналитиками, а я приступил к следующему этапу.

# 2 Создание нового микросервиса

На этом этапе мне было необходимо реализовать каркас сервиса, чтобы в будущем он мог считывать данные о записях пациентов с ПДН, инициировать задачи на обзвон этих пациентов, посредством механизма создания асинхронных задач.

Первым делом я создал модуль нового микросервиса в репозитории и создал "pom.xml" файл, в котором далее указал требуемые зависимости для работы микросервиса.

Далее я сконфигурировал некоторые библиотеки в соответствии с тем, какая ожидается функциональность от них. Одной из таких библиотек была "com.fasterxml.jackson.Core" с помощью которой объекты Java преобразуются в формат JSON для их отправки по сети:

1. Необходимо было отключить функцию, которая вызывала бы исключение при обнаружении непредвиденных свойств во время десериализации JSON в объект Java. Иначе, если коллеги захотят добавить новые данные в сообщение, которых мы не ожидаем, выполнение бы завершилось с ошибкой.
2. Отключить сериализацию даты и времени как метки времени (timestamp). Вместо этого, даты и времена будут сериализованы в формате строки. Так и только так при отправке сообщения в топики, исключения вызвано не будет.

Код предоставлен в Приложении А.

Проблем на пути разработки у меня не возникало, все изменения были внесены без значительных препятствий. Чтобы обеспечить правильность работы сервиса, я добавил тестовый класс, который в будущем будет запускать модульные тесты и не позволит загрузить изменения на сервер при наличии ошибок.

По завершении разработки я зафиксировал код в удаленном репозитории, откуда он был развернут инженерами в тестовой среде. Никаких ошибок обнаружено не было. Сервис был успешно внедрен и можно было приступать к последующей разработке.

# 3 Разработка модели данных для аудита

На этом этапе я работал над созданием модели данных, которая включает в себя необходимые сущности и репозитории для работы с аудитом обзвонов с напоминанием о записи. Для этого требовалось создать таблицу аудита в базе данных, реализовать JPA-entity и репозиторий.

Сперва я начал работать над реализацией структуры таблицы, в которой хранились бы все необходимые данные для аудита обзвонов. Атрибуты таблицы были определены на основе требований к функциональности системы, а также необходимости отслеживать статус обзвона и сопутствующую информацию. Вот ключевые атрибуты, которые я перенес в таблицу:

* Уникальный идентификатор записи (первичный ключ)
* Статус передачи задания на звонок
* Статус совершения звонка во внешней системе
* Инициатор отправки напоминания
* Идентификатор записи на прием
* Время начала и окончания записи
* Информация о пациенте
* Время создания и обновления записи

Создав таблицу в базе данных и добавив ее в миграции, следующим шагом было разработать JPA-entity для работы с этой таблицей, а также репозиторий, обеспечивающий CRUD-операции (создание, чтение, обновление, удаление) с данными.

Сущности (JPA-entity) позволяют манипулировать данными в программном коде, как с объектами. Сущность включает в себя поля, соответствующие колонкам таблицы, а также методы для получения и установки значений этих полей, что обеспечивает типизированный и безопасный доступ к данным таблицы.

Процесс написания сущности у меня занял много времени, несмотря на то что я занимался как будто бы обычным переносом названий и типов из существующей таблицы. Там было много системных полей и в результате класс сущности вышел на 300 строк.

Далее я разработал репозиторий. Он, играет роль связующего звена между базой данных и логикой приложения. Это интерфейс, предоставляющий набор стандартизированных методов для выполнения CRUD-операций над объектами JPA-entity. Никаких новых методов мною добавлено не было. Стандартных, которые появляются при наследовании класса “JPARepository”, было достаточно для работоспособности на данный момент.

Также в рамках данного этапа требовалось добавить новое поле "initiator" в таблицу "appointment". Это изменение позволило указать, кто является инициатором отправки напоминания, что важно для аудита и отслеживания событий. Для этого я обновил сущность, добавив новое поле, и внес изменения в код сервиса, чтобы корректно обрабатывать новое поле при создании звонка.

Таким образом, разработка модели данных для аудита включала в себя создание и настройку структуры базы данных, реализацию необходимых сущностей и репозиториев. Всё было выполнено мною в срок и без вопросов, наверное, потому что мне нравится всё что связано с базами данных.

# 4 Реализация алгоритмов для работы с аудитом

На этом этапе моей работы необходимо разработать и интегрировать алгоритмы для управления записями аудита в рамках системы уведомлений. Задача включала создание новых записей, обновление существующих данных, а также реализацию поиска актуальных записей в системе.

Я разработал интерфейс, который описывает основные методы управления записями аудита со следующими ключевыми функциями:

* Создание новой записи аудита: позволяет создать запись, связанную с конкретным приемом и контекстом обзвона.
* Дополнение данных аудита: дополняет запись аудита дополнительными данными о пациенте, такими как информация о возрастной группе, номер телефона.
* Обновление статусов записи: позволяет обновить различные статусы в записи аудита, что важно для отслеживания процесса напоминаний и его исходов.
* Поиск актуальной записи аудита: обеспечивает возможность находить и обновлять актуальные записи по идентификатору записи на прием.

Далее я создал сервисный класс, в котором выполнил наследование методов из интерфейса и приступил непосредственно к реализации.

Я внедрил компонент "CurrentTime" для того, чтобы устанавливать точные временные метки. Использовал его, например, для обновления временных отметок при последнем изменении записей аудита.

Кроме того, мною был разработан класс-преобразователь сущностей, который я использовал для преобразования данных между различными объектами и форматами, упрощая обработку и передачу данных в системе.

По мере разработки я также добавлял модульные тесты, которые проверяли корректность работы каждого метода в различных сценариях.

По итогу этапа я успешно разработал алгоритмы, которые будут использованы в следующих этапах для интеграции доработки.

# 5 Адаптация существующих алгоритмов под новые требования

Приступая к этапу адаптации существующих алгоритмов под новые требования, я начал с детального анализа текущих алгоритмов обработки напоминаний в системе ЕМИАС.Обзвоны. Моя задача заключалась в интеграции новой функциональности, которая должна добавить напоминания пациентам на ПДН о предстоящих медицинских исследованиях и консультациях.

* Адаптация алгоритма получения сообщений ЕСУ о действиях с записью.

Изменил обработку входящих сообщений, включив логику поиска и создания записей аудита. Теперь, если актуальная запись не найдена, система создаёт новую, а при получении действий CANCEL или CHANGE соответственно обновляет статусы или создаёт новую запись на основе существующей.

* Модификация алгоритма обогащения данных.

Внедрил процесс обновления данных аудита при обнаружении существующей записи. Обновление включает в себя дообогащение данными о пациенте, номере телефона, возрастной группе и хэше, что позволяет поддерживать актуальное состояние аудита звонков.

* Обработка напоминаний на ЛИ/ИИ пациентам с ПДН.

Изменил алгоритмы создания, отправки задачи на обзвон, формирования оповещений так, что по окончанию их выполнения обновлялся статус в аудите, в том числе даже в случае исключения.

* Создание новых аудитов при обработке статуса звонка.

Имплементировал логику для обновления статусов аудита звонков с учётом новых статусов звонка и уведомления. Если запись была найдена, статусы обновлялись для отражения последних изменений. В случаях, когда нужно было создать новую запись из-за особых условий, система не только регистрировала новую информацию, но и выдавала предупреждение о необычной ситуации.

В процессе адаптации алгоритмов мне также пришлось обновлять условия проверок модульных тестов, включив в них новые сценарии обработки и взаимодействия данных.

По окончанию этапа все изменения были успешно интегрированы в систему ЕМИАС.Обзвоны. Модульные тесты подтвердили стабильность работы новой функциональности.

# 6 Реализация обработчика напоминаний

На этом этапе передо мной стояла задача реализовать механизм прослушивания топика сообщений, а также разработать алгоритм для их обработки.

Система уже настроена на постоянное прослушивание топика. Я сделал так, что при получении сообщения создается асинхронная задача на получение данных для записи. Для обеспечения возможности отслеживания процесса обработки каждое поступающее сообщение журналируется с указанием уровня INFO.

Далее асинхронная задача должна попадать в отдельный сервис обработки этих сообщений. Алгоритм обработки в системе напоминаний включает несколько этапов, начиная с проверки наличия записи в базе данных и заканчивая обновлением данных и созданием новых задач для дальнейшей обработки. Алгоритм обработки я реализовал так:

1. Проверка наличия записи: если запись по переданному идентификатору не найдена, система выбрасывает техническую ошибку, что предотвращает дальнейшую некорректную обработку
2. Поиск и проверка аудита: система ищет соответствующий аудит по написанному мною ранее алгоритму. Отсутствие аудита также ведёт к выбросу бизнес-ошибки.
3. Обновление статусов аудита: при наличии аудита система обновляет его статусы в соответствии с текущим статусом записи.
4. Обновление записи: система обновляет запись, устанавливая нового инициатора на основе полученных данных, что гарантирует правильное отслеживание источников изменений в записях.
5. Создание асинхронной задачи: на последнем этапе создаётся асинхронная задача для загрузки дополнительных данных для записи, что в дальнейшем будет необходимо для формирования звонка.

Естественно, я написал модульные тесты для обработчика задач, чтобы гарантировать, что каждый аспект алгоритма функционирует корректно. Эта система теперь автоматически обрабатывает и управляет напоминаниями о предстоящих консультациях и исследованиях.

# 7 Проверка, устранение ошибок и дополнительная оптимизация

На завершающем этапе разработки я сосредоточился на всестороннем анализе и тестировании разработанной функциональности. Целью этого этапа было выявление и устранение любых возможных ошибок, а также дополнительная оптимизация кода для улучшения общей производительности системы.

Сосредоточив внимание на модульном тестировании, я проверил каждую функцию системы напоминаний для уверенности в их корректной работе. Всё работало корректно.

Также я провел ручное тестирование ключевых компонентов системы. Этот процесс включал проверку правильности реагирования системы на различные сценарии использования, такие как создание, обновление и отмена напоминаний. Всё работало корректно.

В попытках оптимизации я добавил индекс на таблицу аудитов, чтобы ускорить их поиск в базе данных. Большого прироста производительности это не принесло, однако на дистанции должно принести пользу.

По окончании этого этапа я удостоверился, что система напоминаний работает. Все функции системы были тщательно проверены.

# ЗАКЛЮЧЕНИE

В рамках производственной, технологической (проектно-технологической) практики мною была успешно разработана система напоминаний для пациентов, находящихся на диспансерном наблюдении. Задание, поставленное моим руководителем, а именно разработать сервис и провести интеграцию с внешней системой уведомлений, я выполнил в срок. Были учтены все функциональные и бизнес-требования. В итоге система была успешно внедрена.

Разработка дала мне ценный опыт в областях проектирования программного обеспечения, чтении документации, модификации кода, оптимизации, работы с базами данных.

Не менее полезным для меня также оказался опыт работы под присмотром опытного руководителя. Благодаря его поддержке и руководству я приобрел ценные навыки профессионального общения.

# СПИСОК ИСПОЛЬЗУЮЕМЫХ ИСТОЧНИКОВ

1. Документация для разработки Spring Boot приложений [Электронный ресурс]. – Режим доступа. – <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>, свободный (дата обращения 24.04.2024)
2. Документация PostgreSQL [Электронный ресурс]. – Режим доступа. – <https://www.postgresql.org/docs/>, свободный (дата обращения 24.04.2024)
3. Требования по системе напоминаний для пациентов на ПДН [Электронный ресурс]. – Режим доступа. – <https://wiki.mos.social/>, закрытый (дата обращения 24.04.2024)

# ПРИЛОЖЕНИЕ А

Код-пример конфигурации для преобразования JSON

A screenshot of a computer code

Description automatically generated