

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

|  |  |
| --- | --- |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**  **Департамент математического и компьютерного моделирования** | |
|  |

Отчёт по \_\_\_\_\_\_\_\_ работе

**«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»**

|  |  |
| --- | --- |
| Работа защищена  с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Регистрационный номер \_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г. | Студент(ы) группы № Б9122-02.03.01сцт  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Поповкин А. А.  (подпись)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\*г.  Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (ФИО)  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\*г. |

г. Владивосток

2024

Оглавление

[1. Введение 2](#_Toc169973758)

[1.1. Актуальность 2](#_Toc169973759)

[1.2. Цели и задачи 2](#_Toc169973760)

[1.3. Используемые технологии 2](#_Toc169973761)

[2. Основная работа 4](#_Toc169973762)

[2.1. База данных 4](#_Toc169973763)

[2.1.1. Введение 4](#_Toc169973764)

[2.1.2. Модели 4](#_Toc169973765)

[2.1.3. DTO 4](#_Toc169973766)

[2.1.4. Миграции 4](#_Toc169973767)

[2.2. Взаимодействие с базой данных 4](#_Toc169973768)

[2.2.1. Введение 4](#_Toc169973769)

[2.2.2. Репозиторий 4](#_Toc169973770)

[2.2.3. Сервис 4](#_Toc169973771)

[2.2.4. Исключения 4](#_Toc169973772)

[2.3. Интернет-запросы 4](#_Toc169973773)

[2.3.1. Введение 4](#_Toc169973774)

[2.3.2. Контроллер 4](#_Toc169973775)

[2.3.3. Mapper 5](#_Toc169973776)

[3. Вывод 5](#_Toc169973777)

[4. Источники 6](#_Toc169973778)

# Введение

## Актуальность

В современном программировании работа над приложением или сайтом делится на две составляющие: **backend** и **frontend**.

**Frontend** разработка отвечает за визуальную составляющую программы: за расположение кнопок, за красивые иконки и т. д.

**Backend** в свою очередь отвечает за всё остальное. Создание бизнес-логики, управление базами данных, обработка интернет-запросов пользователей к серверу, аутентификация пользователей, интеграция с другими сервисами и многое другое. От того, как backend-разработчики опишут работу сервера, от оптимизации запросов к базам данных зависит безопасность и скорость работы всего приложения или сайта.

Хорошо структурированный и чистый код облегчает поддержку и дальнейшее развитие приложения, снижает вероятность возникновения багов и значительно упрощает тестирование.

Использование современных фреймворков и библиотек, таких как Spring для Java, помогает упростить и ускорить процесс разработки, при этом повысив качество конечного продукта.

## Цели и задачи

В рамках курсовой работы было необходимо реализовать приложение на языке программирования Java для создания списков задач для различных пользователей, с системой управления базами данных (СУБД) PostgreSQL.

Для достижения поставленной цели было необходимо решить следующие задачи:

1. Планирование структуры базы данных.
2. Изучение типов веб-запросов
3. Чтение документации к новым инструментам.
4. Изучение методологии написания приложений.

В функции приложения входило:

1. Добавление, чтение, изменение данных, удаление пользователей.
2. Добавление, чтение, изменение данных, удаление заметок.
3. Создание системы аутентификации для пользователей.
4. Обработка веб-запросов к приложению.
5. Обработка исключений.
6. Исполнение написанных миграций.

## Используемые технологии

Для выполнения поставленных задач были использованы различные технологии:

1. **Фреймворк Maven**[3]
   1. Инструмент для автоматической сборки проектов на Java.
2. **Фреймворк Spring**[4]
   1. Набор инструментов, предлагающий различные модули для выполнения задач, таких как доступ к данным, безопасность, транзакции и обмен сообщениями.
3. **Библиотека Hibernate**[5]
   1. Библиотека, автоматизирующая процесс перевода кода в команды для СУБД.
4. **Библиотека Lombok**[6]
   1. Библиотека, позволяющая сократить шаблонный код Java в хорошо читаемые анотации.
5. **Система контейнеров Docker**[1]
   1. Платформа, которая предназначена для разработки, развёртывания и запуска приложений в контейнерах.
   2. Контейнеры были использованы для изолированного запуска СУБД PostgreSQL на виртуальном сервере, что позволило с удобством проводить манипуляции с базой данных.
6. **СУБД PostgreSQL**[2]
   1. Серверная система управления базами данных.
7. **Библиотека Flyway**[7]
   1. Библиотека для автоматического исполнения миграций при запуске приложения.
8. **Интегрированная среда разработки (IDE) IntelliJ IDEA Ultimate**[8]
   1. Удобная среда разработки со встроенной поддержкой **Spring** и **Maven**.

# Основная работа

## База данных

### Введение

В данной подглаве будут описаны созданные модели, а также будет рассказано про процесс миграций и объекты для передачи данных.

### Модели

#### Введение

В рамках задачи было создано две таблицы в базе данных: **User**, отражающей пользователя и Nodes, отражающей записи.

Библиотека hibernate даёт возможность описывать объекты базы данных как сущности, что позволяет автоматизировать перевод класса в базу данных.

Использование библиотеки Lombok позволило сильно сократить количество строк кода с помощью аннотаций, использующих шаблоны для инициализации класса с разным набором параметров, а также для описания методов получения/назначения значений из классов.

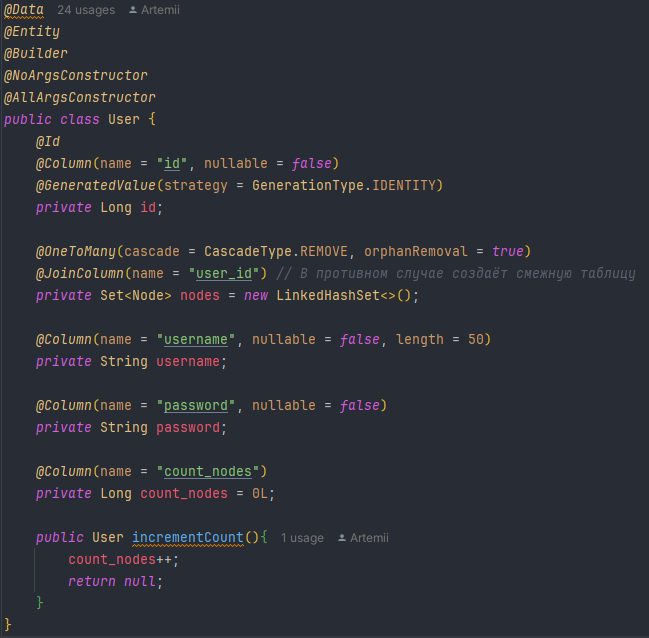
#### User

User имеет следующие поля:

1. **id**
   1. Поле для идентификации пользователя
2. **username**
   1. Имя пользователя
3. **password** 
   1. Пароль
4. **count\_nodes**
   1. Количество записей

Также имеет переменную **nodes**, реализующую OneToMany связь с таблицей Node.

Функция **incrementCount** увеличивает значение count\_nodes на 1.

  
 Рис. 1. Код для таблицы User

#### Node

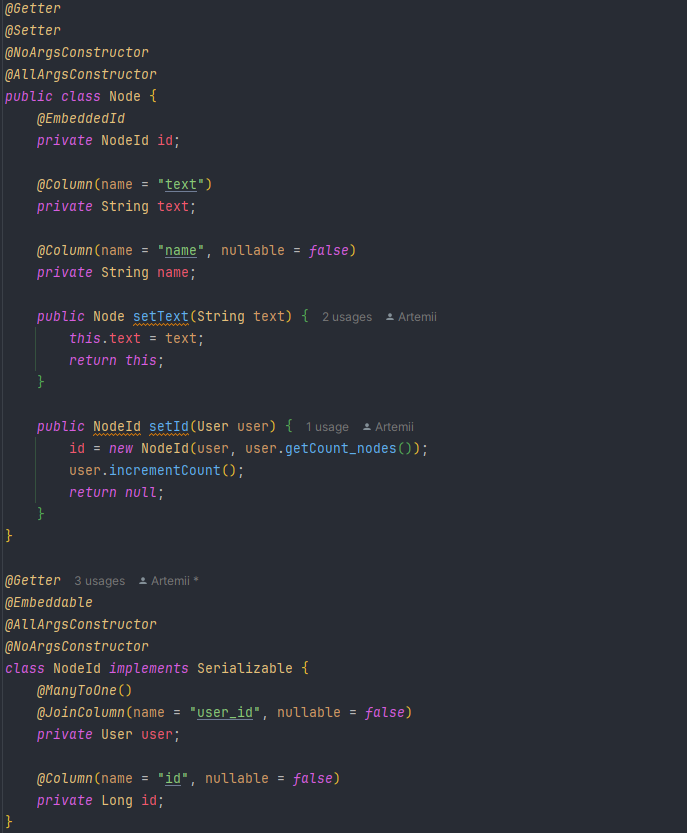
Node имеет следующие поля:

1. **id**
   1. Составной ключ для идентификации записи.  
      Состоит из:
      1. **User**
         1. Пользователь, кому принадлежит запись
      2. **id**
         1. Номер записи относительно пользователя
2. **name**
   1. Название записи
3. **text**
   1. Текст записи

Также имеет переменную **users**, реализующую ManyToOne связь с таблицей User.

Функция **incrementCount** увеличивает значение count\_nodes на 1.

Функция **setId** ставит нужный id, зависащий от количества записей у пользователя, и возвращает запись.



### DTO

### Миграции

## Взаимодействие с базой данных

### Введение

Описание общего концепта работы

### Репозиторий

### Сервис

### Исключения

## Интернет-запросы

### Введение

Общее описание обработки запросов

### Контроллер

#### Введение

#### Пользователи

##### Чтение

##### Аутентификация

##### Добавление

##### Удаление

##### Редактирование

#### Записки

##### Аутентификация

##### Чтение

##### Добавление

##### Удаление

##### Редактирование

### Mapper

Планирование структуры проекта, модульность

4.1 Сущности

Основные методы сущностей

4.2 DTO

4.3 Репозиторий

4.4 Исключения

4.5 Контроллер

4.6 Сервис

4.7 Migration

# Вывод

# Источники

1. Docker - Reference documentation // Docker Documentation [Электронный ресурс]. URL: https://docs.docker.com/reference/ (дата обращения: 18.06.2024).

2. PostgreSQL 16.3 Documentation // PostgreSQL Documentation [Электронный ресурс]. URL: https://www.postgresql.org/docs/16/index.html (дата обращения: 22.06.2024).

3. Maven – Maven Documentation [Электронный ресурс]. URL: https://maven.apache.org/guides/index.html (дата обращения: 22.06.2024).

4. Documentation Overview :: Spring Boot [Электронный ресурс]. URL: https://docs.spring.io/spring-boot/documentation.html (дата обращения: 18.06.2024).

5. Documentation - 6.5 - Hibernate ORM // Hibernate [Электронный ресурс]. URL: https://hibernate.org/orm/documentation/6.5/ (дата обращения: 18.06.2024).

6. Lombok - documentation [Электронный ресурс]. URL: https://projectlombok.org/features/ (дата обращения: 22.06.2024).

7. Flyway CLI and API - Flyway - Product Documentation [Электронный ресурс]. URL: https://documentation.red-gate.com/flyway/flyway-cli-and-api (дата обращения: 18.06.2024).

8. Getting started | IntelliJ IDEA Documentation [Электронный ресурс]. URL: https://www.jetbrains.com/help/idea/getting-started.html (дата обращения: 22.06.2024).