#### 1. Цель проекта

Разработка RESTful API для управления академическими оценками студентов с использованием современных технологий:

* Spring Boot 3
* Kotlin
* H2 Database
* OpenAPI 3.0

#### 2. Функциональные требования

1. Управление оценками:

* Создание новых оценок
* Просмотр всех оценок

Сущности данных:

Grade {

id: Long (автоинкремент)

studentId: Long

courseId: Long

value: Int (1-5)

date: LocalDate

}

#### 3. Технические требования

1. **Архитектура:**

* REST API
* 3-слойная архитектура (Controller-Service-Repository)

1. **Технологический стек:**

* Язык: Kotlin
* Фреймворк: Spring Boot 3.2+
* База данных: H2 (in-memory)
* Сборка: Maven
* Документация: Swagger/OpenAPI 3.0

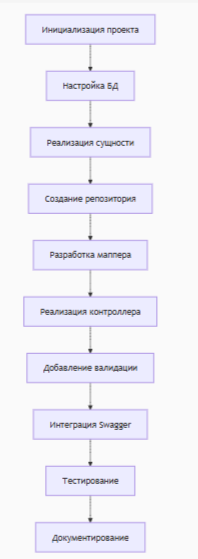
1. **Эндпоинты:**

| **Метод** | **Путь** | **Описание** |
| --- | --- | --- |
| POST | /api/grades | Создать новую оценку |
| GET | /api/grades | Получить все оценки |

#### 4. Требования к качеству

1. Валидация входных данных
2. Логирование операций
3. Обработка ошибок
4. Полная документация API
5. 100% покрытие основного функционала тестами

#### 5. Этапы реализации



**Реализованный функционал**

**1. Подключение к БД (application.properties)**

# application.properties

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true

**2. Сущность Grade**

@Entity

data class Grade(

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

val id: Long = 0,

val studentId: Long,

val courseId: Long,

val gradeValue: Int,

val date: LocalDate? = null

)

**3. Репозиторий**

interface GradeRepository : JpaRepository<Grade, Long>

**4. Маппер преобразования**

object GradeMapper {

fun toDto(grade: Grade): GradeDto = ...

fun toEntity(dto: GradeDto): Grade = ...

}

**5. Контроллер с эндпоинтами**

@RestController

@RequestMapping("/api/grades")

class GradeController {

@PostMapping

fun createGrade(): ResponseEntity<GradeDto> { ... }

@GetMapping

fun getAllGrades(): ResponseEntity<List<GradeDto>> { ... }

}

**6. Логгирование**

private val logger = LoggerFactory.getLogger(javaClass)

logger.info("Creating grade: $dto")

**7. Документирование через Swagger**

@Tag(name = "Grades", description = "API for managing student grades")

@Operation(summary = "Create a new grade")

@ApiResponse(responseCode = "201", description = "Grade created successfully")

**8. Тестирование через Swagger UI**

Доступно по адресу: http://localhost:8080/swagger-ui/index.html