

EXMDirect — GPT Assistant Collaboration Log

Duration: May 2024 – July 2025

Author: Baldruev Sergei — System Analyst

GPT Assistant Collaboration Log

Context

During the system analysis of the EXMDirect project, GPT (ChatGPT-4) was used to:

Clarify architectural decisions

Simulate realistic endpoints for missing modules (e.g. AntiCheat, Hints)

Generate documentation structure

Draft diagrams (BPMN, Use Case, ER)

Create OpenAPI (Swagger) specs

This file captures key moments and responses that contributed to project structuring.

Example 1: AntiCheatController Design

Prompt:

«Хочу эндпоинты для защиты от списывания: вкладка, таймер, активность — как системный аналитик это описать?»

Generated response:

```
POST /api/anticheat/log
{
  "studentId": 123,
  "examId": 45,
  "eventType": "TAB_SWITCH",
  "timestamp": "2025-07-16T12:00:00"
}
```

Swagger & DTOs: Auto-generated and inserted into Swagger YAML.

Example 2: Use Case Diagram Prompting

Prompt:

«Опиши диаграмму Use Case для платформы тестирования — роли, действия, сценарии.»

Generated Response Summary:

Actors: Student, Teacher, System

Use Cases: Authorize, Create Exam, Generate Test, Pass Test, Review Results, Notify via Telegram

Relationships: «includes» between Generate Test → Randomize Questions, «extends» on AntiCheat Logging

Result: Draft used to build final Use Case diagram in draw.io

Example 3: Endpoint Structure Validation

Prompt:

«Вот список контроллеров и методов — чего не хватает?»

Generated insights:

Missing: /api/hints/by-topic, /api/anticheat/by-student

Suggested grouping: /api/questions/exams → split by subject, add pagination

Recommendation: Add DTOs with validation annotations

Result: Updated controller layout and endpoints reflected in Swagger

Example 4: BPMN Diagram Prompting

Prompt:

«Построй BPMN по процессу “Учитель создаёт тест → Ученик проходит → Проверка и отчёт”»

Response Outline:

Start: Teacher logs in

Task: Creates and publishes exam

Parallel Flow: Students access and complete exam

Intermediate Event: AntiCheat Log event

End: System sends report and grade to teacher

Result: BPMN diagram created in visual tool based on this flow

Example 5: ER Diagram Design

Prompt:

«Составь структуру сущностей для базы данных: ученики, экзамены, попытки, логи»

Generated Entities:

Student (id, name, class)

Exam (id, subject, date, teacherId)

Attempt (id, studentId, examId, score, startedAt, finishedAt)

AntiCheatEvent (id, attemptId, eventType, timestamp)

Result: ER-диаграмма визуализирована по этой модели

Example 6: Swagger Documentation Annotations

Prompt:

«Сгенерируй OpenAPI-аннотации для контроллера HintController с методами GET и POST»

Generated Output:

```
@Operation(summary = "Get hints by topic")
@GetMapping("/api/hints/by-topic")
public List<HintDto> getHints(@RequestParam String topic) {...}
```

```
@Operation(summary = "Add a new hint")
@PostMapping("/api/hints")
public ResponseEntity<?> addHint(@RequestBody HintDto hintDto) {...}
```

Result: Аннотации добавлены в код, YAML обновлён

Diagram Generation

Diagrams were first outlined in Chat:

Use Case: flow between Teacher / Student / System

BPMN: business process for test lifecycle

ER: Student → Exam → Attempt → AntiCheatEvent

All diagrams were created based on descriptions generated collaboratively.

API Structure Assistance

ChatGPT was used to validate completeness of endpoints:

Matched each controller to business need

Identified missing routes

Proposed logical RESTful structure

Summary

Using GPT as an assistant helped:

Speed up idea validation

Avoid structural inconsistencies

Prototype a real system with minimal friction

All prompts were based on real project needs. GPT's role was as a second brain, not a replacement for analysis.

Full Session Log

[Public ChatGPT Share Link](#)