ASTANA IT UNIVERSITY

Educational Institution Management

Project topic: Educational Institution Management
Student: Yerbolatov Damir Group: 2501
Course: Object-Oriented Programming (Java)
Instructor: [Arailym Orazova] Date: 9.02.2026

PROJECT OBJECTIVE

Objective of the Project

The main objective of this project is to create a Java application that demonstrates:

- object-oriented programming principles,
- work with data collections,
- REST API with JSON,
- clean code and good architecture.

Speech:

The project shows how a real backend system can be designed and implemented using Java.

PROJECT REQUIREMENTS

Main Requirements

- Use OOP principles
- Work with collections (data pool)
- Handle exceptions correctly
- Create REST API with JSON

- Demonstrate SOLID principles
- Use design patterns
- Show the project using Postman

Student: Yerbolatov Damir

PROJECT ENTITIES (BASICS)

Main Entities

- Institution
- Student
- Teacher

Each entity contains:

- attributes (fields),
- constructors,
- getters and setters,
- overridden toString(),
- overridden equals() and hashCode() methods.

PACKAGE STRUCTURE

- • domain
- dto
- repository
- service
- controller
- config

Student: Yerbolatov Damir

OOP PRINCIPLES

Object-Oriented Programming Principles

- Encapsulation
- All fields are private and accessed using getters and setters.
- Abstraction
- Business logic is implemented using interfaces and service layers.
- Polymorphism
- The program works with objects through interface references.
- Inheritance
- Inheritance is used only where it is logically correct.
- In other cases, composition is preferred.

Speech:
This approach makes the code flexible and easy to maintain.

Student: Yerbolatov Damir

DATA POOL (COLLECTIONS)

In-Memory Data Processing

The project uses Java collections to store and process data:

- List and Map collections,
- searching students by name,
- filtering data,
- sorting data using Streams and Lambda expressions.

Speech:
Collections allow fast and simple data processing inside the application.

Student: Yerbolatov Damir

EXCEPTION HANDLING

Error and Exception Handling

The project includes:

- custom exceptions (for example: NotFoundException, ValidationException),
- validation of input data,
- safe handling of incorrect requests.

Speech:

The application does not crash and always returns clear error messages.

REST API (JSON)

REST API

- •The system provides REST API endpoints.
- Data is transferred in JSON format.
- CRUD operations are available for all main entities.

Examples:

- • GET /api/students
- POST /api/students
- PUT /api/students/{id}
- DELETE /api/students/{id}

Speech:

The API is tested and demonstrated using Postman.

Student: Yerbolatov Damir

DESIGN PATTERNS

DESIGN PATTERNS

- Builder Pattern
- Used to create complex objects step by step.
- Factory Pattern
- Used to centralize object creation logic.

JAVA LANGUAGE FEATURES

- Java Features
- Generics for type-safe collections,
- Lambda expressions,
- Stream API for filtering and sorting,
- Default or static methods in interfaces.

Speech:

Design patterns improve code structure and readability.

Student: Yerbolatov Damir