

ASSIGNMENT A1

=====

1. Objective

The objective of this assignment is to allow students to become familiar with layers architectural pattern, repository and services.

2. Application Description

Use C# Web.API/ JAVA SWING (optional) to design and implement an application for the National Theater of Cluj. The application should have two types of users (a cashier user represented and an administrator) which must provide a username and a password to use the application.

The administrator user can perform the following operations:

- CRUD on cashiers' information.
- CRUD on the list of shows that are performed at the theater. Keep track of the Genre (Opera, Ballet), Title, Distribution list (a long string is enough), Date of the show and the Number of tickets per show.
- He/she can export all the tickets that were sold for a certain show (either in a csv or xml file).

The cashier can perform the following operations:

- Sell tickets to a show. A ticket should hold information about the seat row and seat number.
- The system should notify the cashier that the number of tickets per show was not exceeded when trying to book a new seat.
- A cashier can see all the tickets that were sold for a show, cancel a reservation or edit the reserved seat.

3. Application Constraints

- The data will be stored in a database. Make sure the database schema is designed to avoid redundancy.
- Use the Layers architectural pattern to organize your application. Use the following layers: Presentation (API) / Services for BL / Repositories
- The application should use Dependency Injection for separation of layers
- One of the APIs should be protected against invalid data / violations by providing **server-side validations**

- Use **factory method** for export to csv or xml.

4. Requirements

- Create the analysis and design document (see the template).
- Implement and test the application.
- Your application should adhere to clean code standards discussed (class design, coding standards, SOLID principles), no duplicate code

5. Deliverables

- GIT link with:
 1. Analysis and design document.
 2. Source files.
 3. SQL script for creating and populating the database with initial values (this can be also done using a seeding method)

6. Deadline – 2 weeks

7. Biography

- **Factory method** - <https://refactoring.guru/design-patterns/factory-method>
- **Asp.net core** - <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/?view=aspnetcore-6.0&tabs=windows>

Asp.net core

<https://app.pluralsight.com/library/courses/implementing-restful-aspdotnet-web-api/table-of-contents>

Spring Boot:

<https://javabrainsthatthink.com/courses/take/springboot-quickstart/multimedia/2784009-adding-a-rest-controller>