DNP exam A21

VIA University College
Software Technology Engineering
Written examination in DNP
(3 hours)

You are allowed to use any IDE (Visual Studio, VS Code, Rider, etc) and to browse the Internet for information (**however**, no live communication, e.g. chats, or contact with any other people, no uploading to public git repository). This is an individual exam!

When finished, hand in your **entire solution in a zip file** to WiseFlow.

In general, you should remember to follow the conventions, theory, and best practices taught in class.

In this exam you will create a system to manage children's toys at a kindergarten.

Part 1 (15%)

1.a – Initial setup

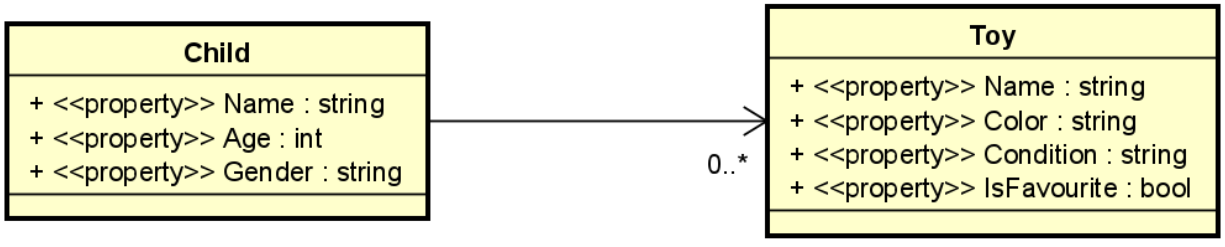
Create a new empty solution, the name of which must contain your student number, eg: DNPexam-123456.

Inside your new solution, create a new Web API project. Call this project WebAPI.

1.b – Domain classes: Child and Toy

Pick an appropriate place to create the domain classes, defined as in the diagram below.

Notice the <<property> indicates a property.



1.c – Data annotations

Add data annotations to the model classes to provide the following constraints:

Child:

- Name: Required, and maximum length is 50
- Age: Required, and must be between 3 and 6 (both inclusive)
- Gender: Required

Toy:

- Name: Required, maximum length is 20

You must decide on a primary key for the Child, and a primary key for the Toy.

1.d – Initial database

Create a database context class, call it KinderGartenContext, put it somewhere appropriately.

Set it up as needed, so that it can provide access Children and Toys.

Use Entity Framework Core to create a migration, and then an SQLite database.

Part 2 (25%)

2.a – Add Child

The following functionality should either just be implemented in the KinderGartenContext, or a repository class. Which of the two approaches you choose will not affect your grade. **All future methods for interacting with the database should be placed in a similar manner.**

You must now create a method for adding a new Child to the database.

2.b – Web API Child Controller

Create a new controller for Children, in an appropriate directory.

Inject either your repository or context, so the controller can persist and retrieve children to/from the database.

2.c – Add Child Endpoint

Create an endpoint for adding a new Child.

2.d – Blazor

Inside your solution, create a new Blazor server project, call it Blazor.

2.e – Create Child page

Create a new page, where you can create and add a new Child to the system. Remember validation.

Part 3 - Create Toy (25%)

1. Create a new page, where you can create a new Toy.
2. You must be able to select which Child, the Toy should be associated with. This should be done by retrieving a list of Child-IDs from the Web API, so the user has a selection of only valid Child-IDs. Create the appropriate endpoint.
3. The request to make a Toymust be a POST to a URI with this format: https://localhost:5001/Toy/owner/{Child-ID}
   Where Child-ID is the value identifying a specific Child.
4. The Toy should be stored in the database, with a connection to its owning Child.

Part 4 – Display (20%)

4.a – Table view

Create a new page to display children and toys.

The view should display the data in the following structure:

Anna is 4 years old Girl. Their toys include:

Toy Name	Color	Is Favourite?
Doggo	brown	True

Jens is 3 years old Boy. Their toys include:

Toy Name	Color	Is Favourite?
Spiderman	red	True

Peter is 5 years old Boy. Their toys include:

Toy Name	Color	Is Favourite?
Donatello	green	False
Leonardo	green	True

4.b – Data retrieval

The data must be retrieved from the web api, create the necessary endpoints.

Similarly create the necessary methods to interact with the database.

Part 5 – Delete toys (10%)

Implement the functionality to delete a Toy.

The endpoint URI must be like this: https://localhost/Child/Toys/{toyId}

The view must reflect the changes, when a toy is deleted.

Part 6 – Filter (5%)

On the page displaying the children and their toys implement the functionality to filter the data by the following two criteria:

1. The favourite status of toys, i.e. we should be able to see
    a. all toys
    b. only favourite toys
    c. only non-favourite toys
2. Gender of the children

We must be able to apply both filters simultaneously, e.g. if we wish to see the favourite toys of all the boys.