

Application Development Frameworks

Assignment 1

This project is designed for you to apply all that you have learned so far to a project.

You are not allowed to use Spring Boot. You are not allowed to use JPA.

Specification

A database holds household information

- the Eircode of the house
- the address of the house
- details about the occupants of the house

The data about the occupants:

- Name
- Age
- Occupation
 - "scholar" for school-going children
 - "pre-school" for pre-school children

The application should allow a user to **(using a menu based system)**:

- Search for a household by Eircode, listing the details of the people in the household
- Add a household, along with its occupant(s)
- Add a new person and assign that person to a household
- Move a person from one household to another
- Delete a household, along with its occupants
- Delete a person
- Display some statistics*
 - the average age of householders
 - the number of students in the households
 - the number of OAPs (aged 65+) in the household

No security is needed.

You must write 4 unit tests,

Refer to the rubric for expected standards.

* Use SQL queries here - don't just retrieve lists and do the work in Java as I have seen done in the past - SQL has queries for these.

Penalties

Penalties for late submission are applied as per section 4.4.2 of the Exam Regulations (see <http://www.mycit.ie/examregulations> (Links to an external site.)).

If you have issues that you think I should be made aware of which may cause you to be late submitting work, please contact me directly using Inbox in Canvas.

Do not plagiarize – don't copy code, don't provide code, don't work too closely with someone else on the project. Trying to be clever by taking someone else's code and renaming classes and variables is not a good strategy and will likely result in a zero grade or worse.

Advice

- Set up your Spring project first, adding all of your dependencies.
- Create a Spring Beans Configuration File, adding beans for a data source and a JdbcTemplate, injecting the data source into the latter. Also add the boilerplate code for h2 (embedded database) and a web server so that you can view the embedded database through a browser.
- Design the database.
- Create the schema for the database.
- Run the program and view the database.
- Create data for the database in SQL.
- Run the program and view the database.
- Create any classes you might need (they usually map to each table in the database).
- Create a `RowMapper()` for each class and/or row from a resultset.
- Create one or more repositories/data access objects for queries through the database.
 - Autowire the `JdbcTemplate` bean into this class. Run your queries through this bean.
 - Test the DAO's methods in the `main()` method e.g. if you have written a `findAll()` method for a particular table e.g. `findAll()` for households, get the `main()` method to extract the DAO bean, and use that bean to call the method, using a loop with `System.out.print()` statements to display the details. This is sufficient as proof-of-concept for yourself that your database queries work and you can extract data from the database through the `JdbcTemplate` bean.
- Add a service layer (interface and implementation) which may include business rules e.g. in a `save()` method you might make sure that the Eircode is unique
- Add more methods to the service layer and implement these methods in the DAO layer.
- Don't forget your unit tests - why not create them as you go along.