CRUD Framework lab.

Suppose you need to make a relatively simple CRUD (Create, Read, Update, Delete) application that manages employees. The application has the following requirements:

- You can add new employees, update existing employees, remove employees, view an employee, search employees and get a list of all employees.
- You should be able to undo/redo the add, update and delete action
- All employees are stored in the database
- You want to log all add, update and delete actions in a logfile

Just when you want to start with the design of this application you get a request from another customer to write a CRUD ProductService application that manages products.

This application has the following requirements:

- You can add, update, remove and view a product.
- You can search products.
- You should be able to undo/redo the add, update and delete action
- All products are stored in an XML file
- You need to send an email to the sales manager whenever a product is added or deleted.
- You want to log all CRUD actions in a logfile
- You should be able to print an overview of all available products.

Because you need to implement 2 similar CRUD applications you decide to design a general CRUDService framework so that you can reuse this framework for both CRUD applications. This CRUDService framework should be very flexible so that we can reuse it for other CRUD-like applications with similar requirements.

Draw in one class diagram the ProductService application using the general CRUDService framework. In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework. Make sure that the framework contains all the general implementation so that the classes outside the framework are as simple as possible.

Make sure you add all necessary UML elements (attributes, methods, multiplicity, etc) to communicate the important parts of your design.