**Pizza Project Functionality**

The pizza project functionality would include the ability to create, read, update, and delete (CRUD) records for both pizzas and cafes. The functionality would be secured by requiring a username of "admin" to access these features.

**Pizza CRUD**: Users with the "admin" username would be able to create new pizza records by providing information such as the pizza's name, size, key\_ingredients, cafe\_id. They would also be able to view, update, and delete existing pizza records.

**Cafe CRUD**: Users with the "admin" username would be able to create new cafe records by providing information such as the cafe's name, location, and phone. They would also be able to view, update, and delete existing cafe records.

**POST, DELETE and PUT requests**: Users with the "admin" username would be able to create new resources by sending a POST request, delete existing resources by sending a DELETE request, and update existing resources by sending a PUT request.

GET is working for all.

**Security**: The application would be secured by requiring a username of "admin" to access the pizza and cafe CRUD functionality, as well as the POST, DELETE and PUT requests. This could be implemented using Spring Security, which would check the user's credentials and restrict access to certain parts of the application based on their role.

The application would also have the ability to retrieve all pizzas, retrieve a specific pizza, retrieve all cafes, retrieve a specific cafe and so on(all of the functions you can find here->https://docs.google.com/document/d/1d634J1H5YcM2TPcX8iCWfT5-SL1RmdlUt3hV3Xrp-F4/edit?usp=sharing)

Cafe to Pizza has bidirectional @OneToMany relationship. This means that one cafe can have multiple pizzas, but each pizza can only belong to one cafe.

**Add programmatically some pizzas and cafes. For this create class InitDataBase.java and add some data.**