REQUIREMENT

The requirement specified for the To-do List are the basic CRUD operations like:

* Create new to-do task to the To-Do list.
* View the list of tasks in the To-Do list.
* Modify the tasks in the To-Do list and change to completed if it is completed.
* Delete the tasks we didn’t need them.

DESIGN

The design implementation needed for the To-Do list is

* Set up the project: Create a new Spring Boot project using STS and include the necessary dependencies for H2 database which is used as a database for the project and Thymeleaf as the front end.
* Create the Entity class: Define a new entity class for the To-do items and use JPA annotations to map it to a table in the H2 database. You can use the @Entity and @Table annotations for this purpose.
* Create the repository: Define a interface that extends

JpaRepository<TodoListModel, Long>, which will allow you to perform basic CRUD operations on the To-do items. Spring Boot will automatically generate an implementation of this interface at runtime.

* Create the controller: Define the controller class that maps HTTP requests to CRUD operations on the repository. You can use the @GetMapping, @PostMapping, @PutMapping, and @DeleteMapping annotations for this purpose.
* Create the views: Define the HTML templates for the different pages of the To-do list application using Thymeleaf.

IMPLEMENTATION

Timeline

Description automatically generated

Adding To-do Task

Timeline

Description automatically generated

Updating the To-do Task

A picture containing timeline

Description automatically generated

Deleting To-do Tasks

TodoListModel Class

* It is the Entity class where Long id, String description, Boolean completed, LocalDate createdtime, LocalDate modifiedtime.
* It also contains the getters and setters, and also it contains the @ArgsConstructor and @NoArgsConstructor from the lambok class.
* It is the model for the table in the database.

TodoRepository Class

* Its extends by JpaRepository class and Entity class is mapped along the datatype of the id which is Long to the database.
* Also the query findById() is defined in the class.

TodoController Class

* The class annotated with @Controller to specify that the class is the controller class.
* @GetMapping(“/”) is used to load the index page in the webpage by showing all the values of the database.
* @GetMapping(“/createtodo) is used to load the addTodo html page in the database.
* @PostMapping(/todo) is used take the value given by the html page .
* @GetMapping(“/edit/{id}”) is used to load the Update html page.
* @PostMapping(“/todo/{id}”) is used to update the value given to the update html.
* @DeletedMapping(“/todo/{id}”) is used to delete the values in the database.

TodoService Class

* It is annotated using @Service to show that the class is a service.
* All the operations for the controllers are done in this class.