Covid 19 Tracker

Project Report

Submitted to:

Dr. Constantinos Constantinides

Submitted By:

Krutik Gevariya (40232386)

Shivam Patel (40226428)

Github UserName

krutik2377

shivampatel304

Date :- 14 / 11 / 2022

SOEN 6441 Advanced Programming Practices

Faculty of Engineering and Computer Science

Department of Computer Science and Software Engineering

Fall 22 Term

Contents

1. Introduction

The project illustrates a functional knowledge of java servlet, which falls under the category of advanced JAVA Topics. “ Covid 19 Tracker ” is a web-based program that we created. This Project’s main objective is to study various architecture , different coding standards with respect to JAVA language , design patterns , different refactoring strategies as well as testing strategies on the application.

The Project uses Covid 19 data, in order to perform such tasks like Create , Read , Update , Delete, Simply means CRUD Covid 19 Information. Covid 19 tracker, first show the data which is fetched from API. Further, User can add new Country by entering different fields data and also can Edit as well as Delete the Covid 19 data which is showing in the application by edit and delete button. Here project is organized with all of the above given functionalities.

2. Tools & Technologies

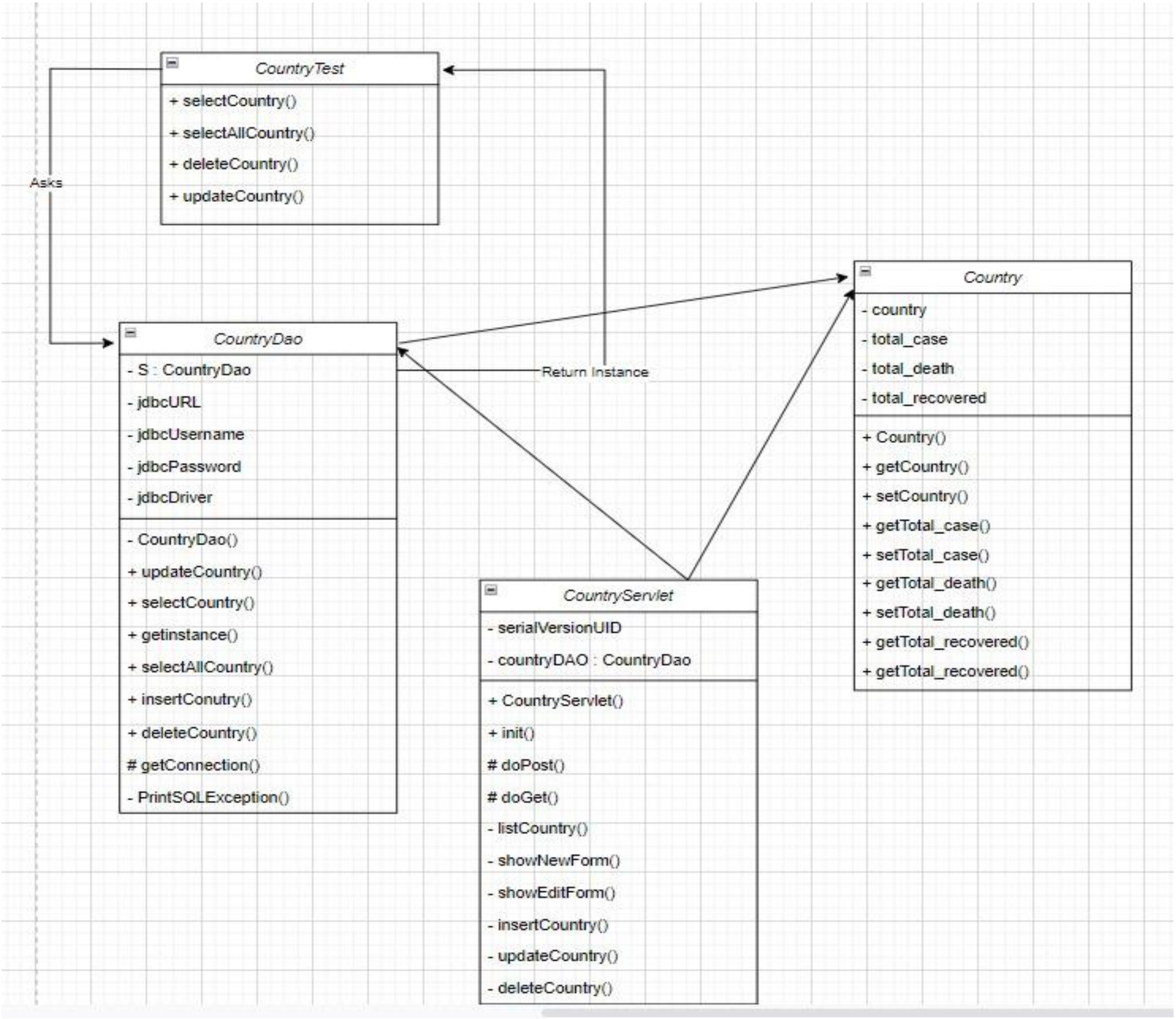
[1] Technologies

- Front End
  1. Servlet
  2. JSP
- Backend
  1. JDBC
- Database
  1. MySQL

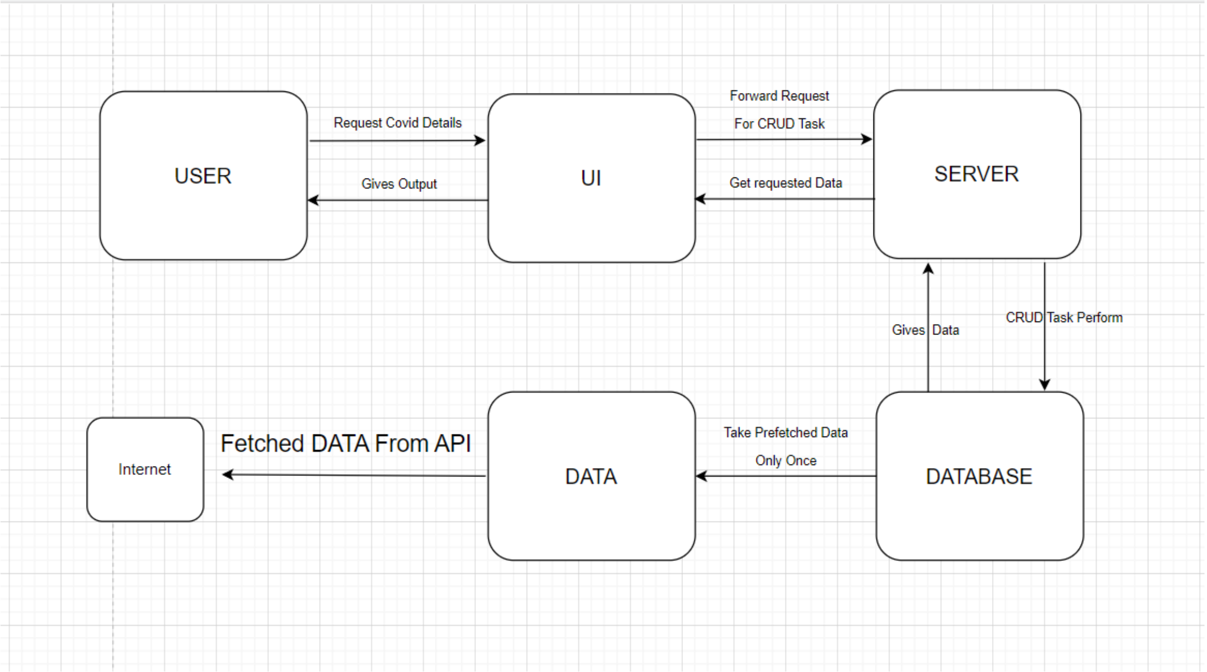
[2] Tools

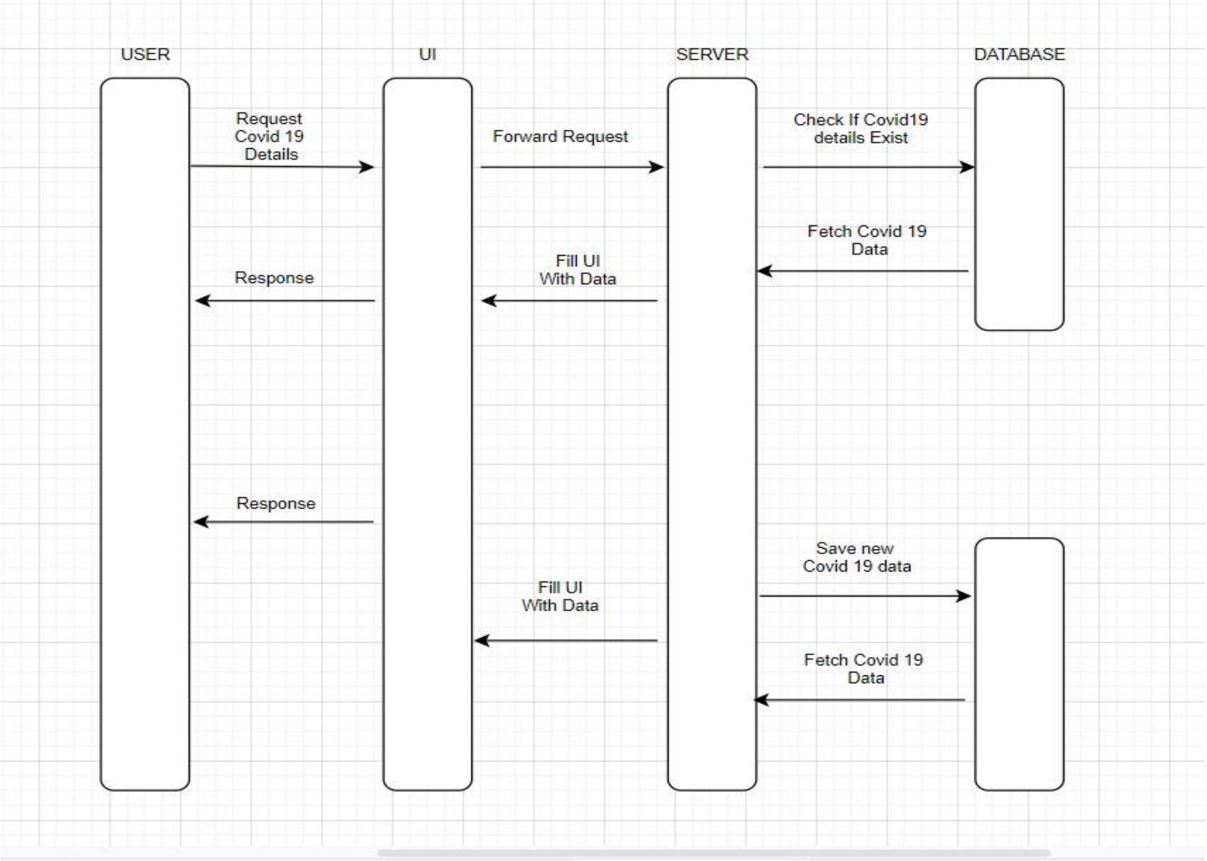
- Junit Testing

3. Class Diagram



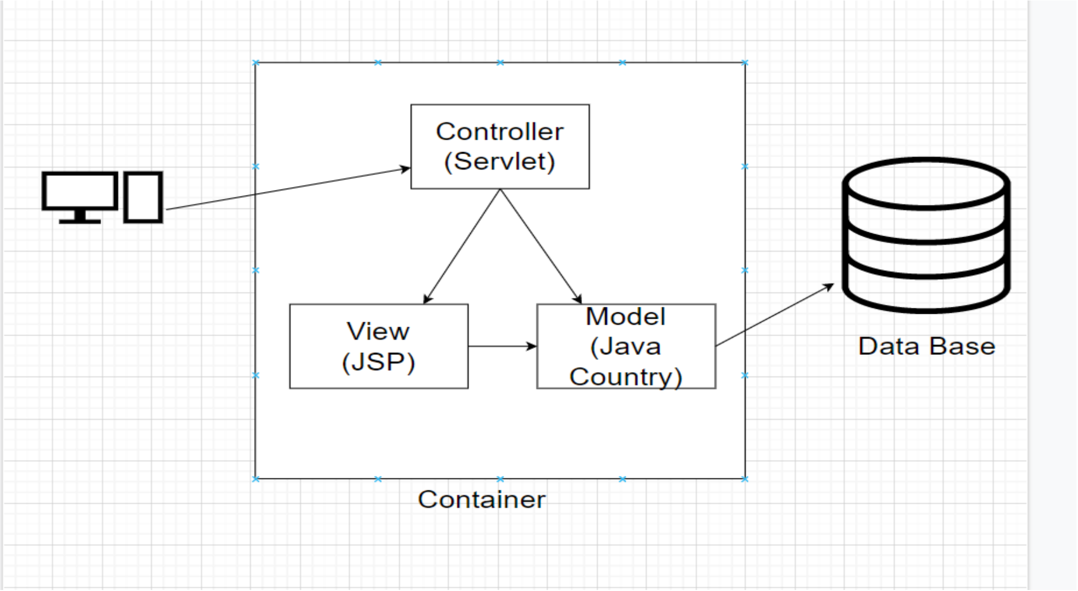
4. Sequence Diagram





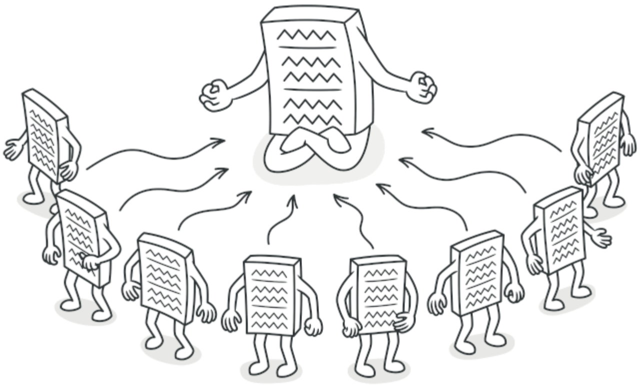
5. Design Patterns

5.1 MVC Design Pattern

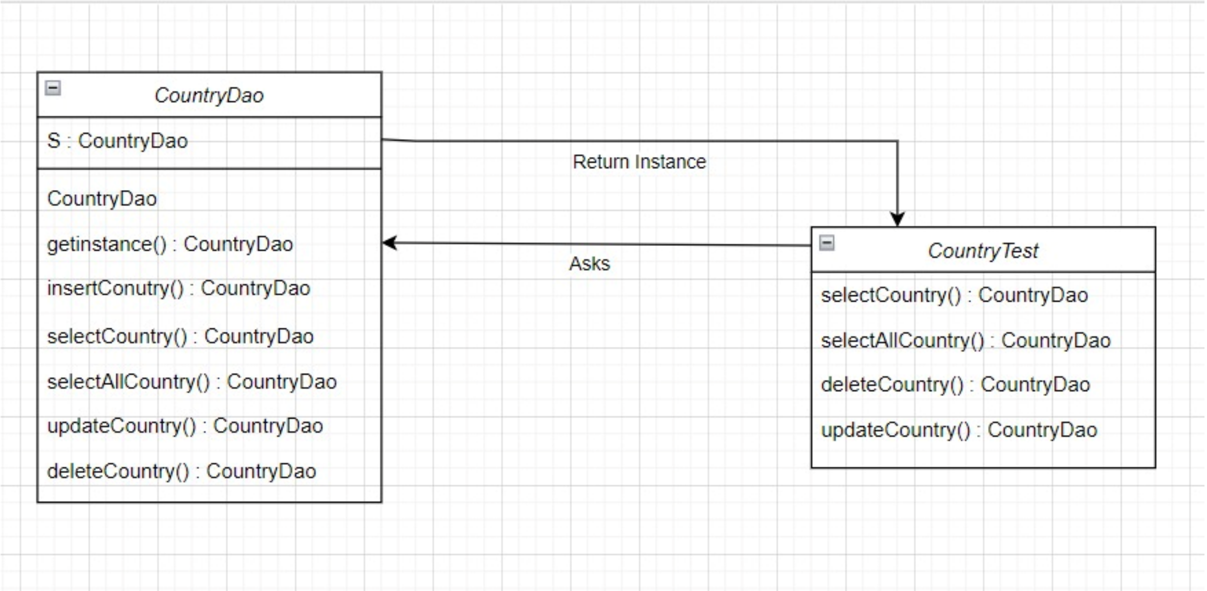


5.2 Singleton Design Pattern

- Singleton Pattern
  Singleton is a creational pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance.



Here How Our Singleton Pattern looks like.


CountryDao
S : CountryDao
CountryDao
getinstance() : CountryDao
insertConutry() : CountryDao
selectCountry() : CountryDao
selectAllCountry() : CountryDao
updateCountry() : CountryDao
deleteCountry() : CountryDao
Return Instance
Asks
CountryTest
selectCountry() : CountryDao
selectAllCountry() : CountryDao
deleteCountry() : CountryDao
updateCountry() : CountryDao

Here we have implemented Singleton pattern so that our constructor of CountryDao is private so that apart from one and only object, no one else can make new instance from outside and we have defined getInstance method into it while one and only instance is created which is null. While CountryTest required instances for testing, at that time it can only access one and only instance which is defined in CountryDao by getInstance method.

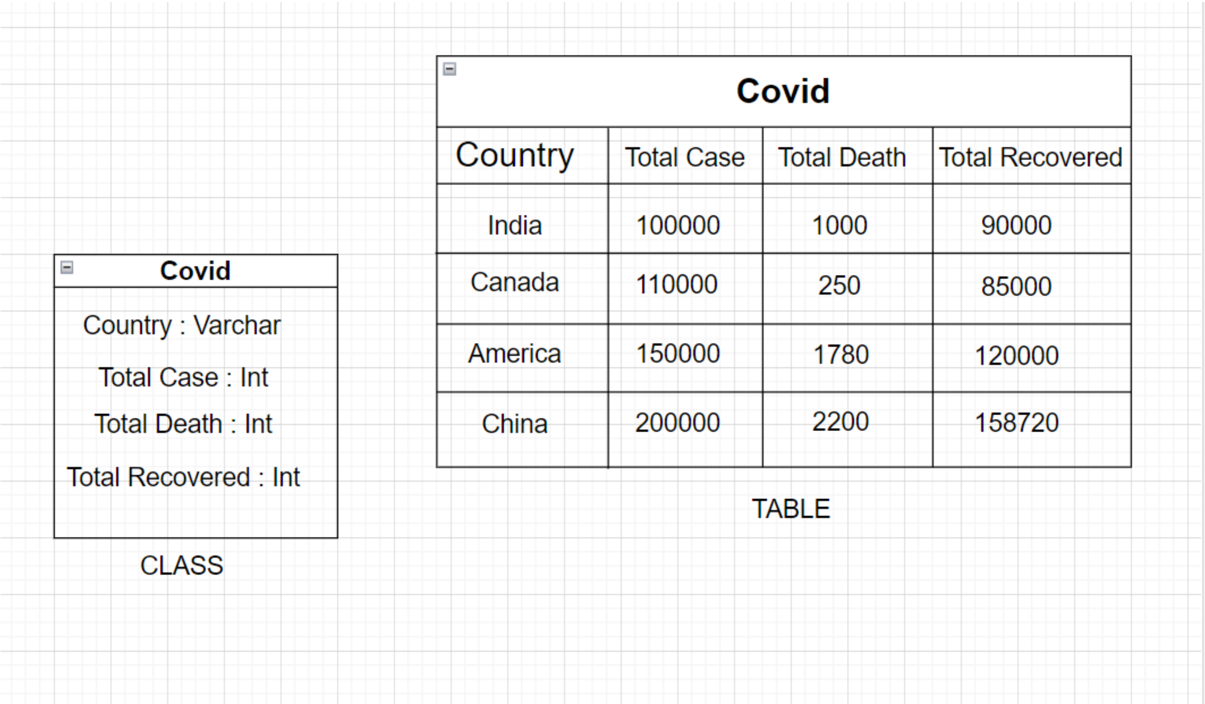
Difference between Singleton instanciation and normal instanciation.

Singleton :- CountryDao dao = CountryDao.getinstance();

Normal :- CountryDao dao = new CountryDao();

We have used Singleton type Instanciation in every file and worked with one and only instance of CountryDao class.

6. Design Patterns



Covid

Country : Varchar

Total Case : Int

Total Death : Int

Total Recovered : Int

CLASS

Covid

Country	Total Case	Total Death	Total Recovered
India	100000	1000	90000
Canada	110000	250	85000
America	150000	1780	120000
China	200000	2200	158720

TABLE

7. Refactoring Strategies

Refactoring, also known as code refactoring in its entirety, is a methodical process of making changes to source code that has already been developed without adding new features or changing how the software in question functions in its core.

You might think of it as a little programming adjustment meant to improve the underlying code's design, implementation, and structure without impairing the program's original usefulness. The readability, maintainability, and extensibility of the software are typically improved as a result.

Steps That we followed for Refactoring

- Reduced the size of the code
- Encapsulating the Fragement which performs cohesive task into it’s own method(Ex. CountryDao.java , CountryServlet.java)
- Replace Constant Variable by Creating own new class (Ex. Constants.java)
- Made the code reusable.
- Extracting two class which performs unrelated tasks (Ex. CountryDao.java , CountryServlet.java)
- Maintain Unidirectional Association for Connection class to maintain security.
- Separate query from modifiers by creating getConnection method.

8. Testing Strategies

For Testing Strategies, We have done unit testing using Juint Framework.

The automated testing method laid out above where the input to a program is modified is quite convenient, but limited nonetheless. Testing larger programs in this way is challenging. One solution to this is unit testing, where small parts of the program are tested in isolation.

Unit testing refers to the testing of individual components in the source code, such as classes and their provided methods. The writing of tests reveals whether each class and method observes or deviates from the guideline of each method and class having a single, clear responsibility. The more responsibility the method has, the more complex the test. If a large application is written in a single method, writing tests for it becomes very challenging, if not impossible. Similarly, if the application is broken into clear classes and methods, then writing tests is straightforward.

We just tested four methods such as selectCountry , selectAllCountry , deleteCountry , UpdateCountry which can test whether for a given parameter the application provides require output or not .

9. References

https://www.javatpoint.com/MVC-in-jsp

https://refactoring.guru/design-patterns/singleton

https://java-programming.mooc.fi/part-6/3-introduction-to-testing

Github Link :- https://github.com/shivampatel304/APP-Project.git