**About**

The application is a dashboard which will help us manage projects in a remote working environment. We have built the application to be scalable and performance oriented. The application can currently handle up to 100000 simultaneous users provided it is deployed on a server of comparable size.

It will consist of a name, a short name, and a short description of what we are trying to achieve in that project.

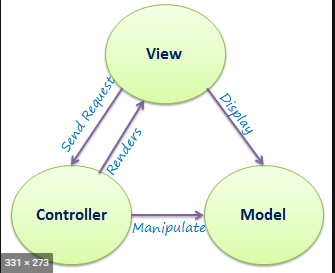
After creating the project, we should be able to edit, archive and un-archive the project.

We can create tasks in the form of cards in the dashboard.

It requires Java 8 and MySQL database.

**CollApp Architecture (Design Pattern) -**

In order to build collapp we have chosen the Model View Controller(MVC) architecture design pattern. MVC architecture pattern separates the application into three main logical components: the model, the view and the controller. Each of these components are built to handle specific development aspects of the application.



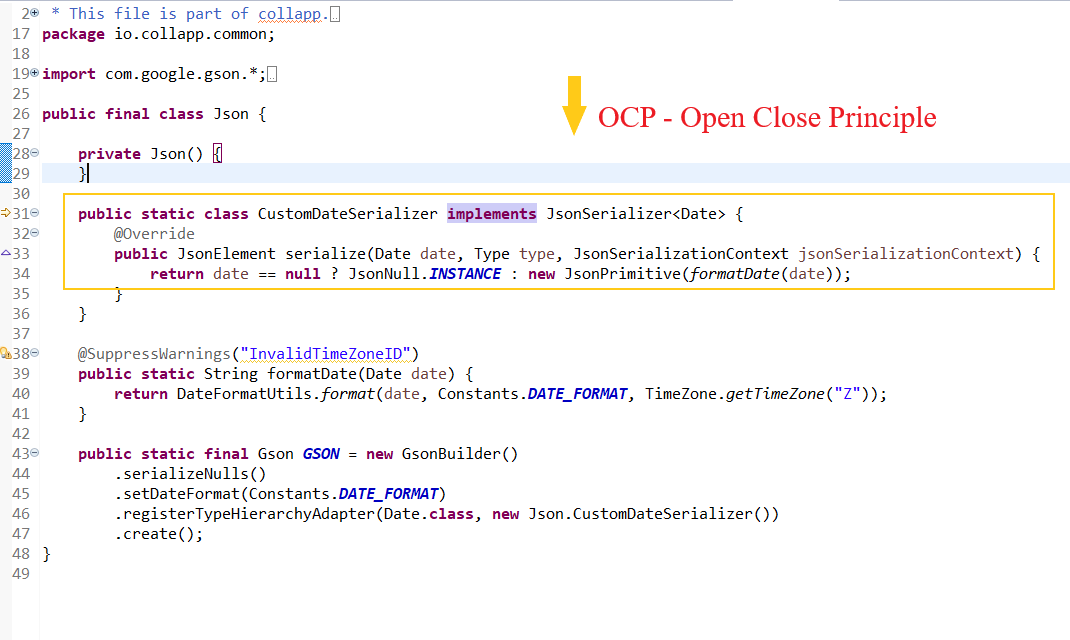
We will now be explaining each of the component of the MVC architecture in relation to our application.  
The Model – For the model we have mainly used MySQL to store and manage data. In order to interact and manage the database we have used Java and Kotlin.

The View – This is the frontend part of the application which the user sees and interacts with. In order to build the Graphical user interface(GUI) we have mainly used AngularJS framework of javascript. Along with AngularJS we have used bootstrap for design management.

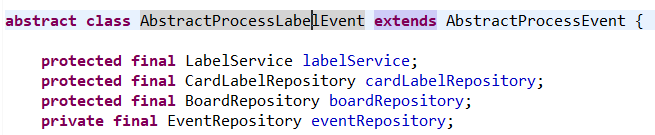
The Controller – This is the backend part of the application which manages how the data from the frontend is sent to the Model and from the model back to the frontend. It also takes care of processing data and this part of the application is written mainly using java and kotlin.

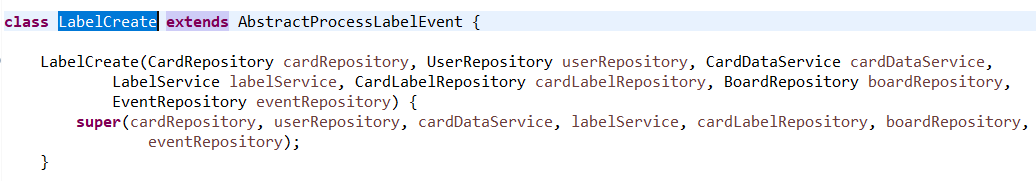
**Design Principles**

* **SRP: Single Responsibility Principle** - We have used interfaces and classes separately when required to stop frequent change of class and each class is defined for one purpose only
* **OCP: Open Close Principle** - Software entities like classes, modules and functions are open to extension but closed for modification.

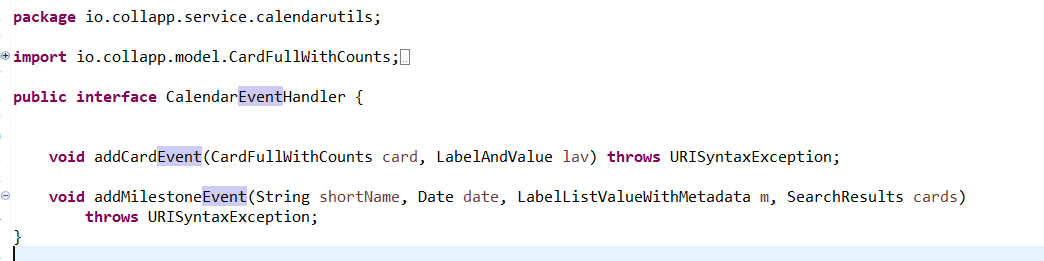


* **LSP: Liskov Substitution Principle** - Derived classes must accept anything that the base class could accept. It basically defines the Inheritance principle.

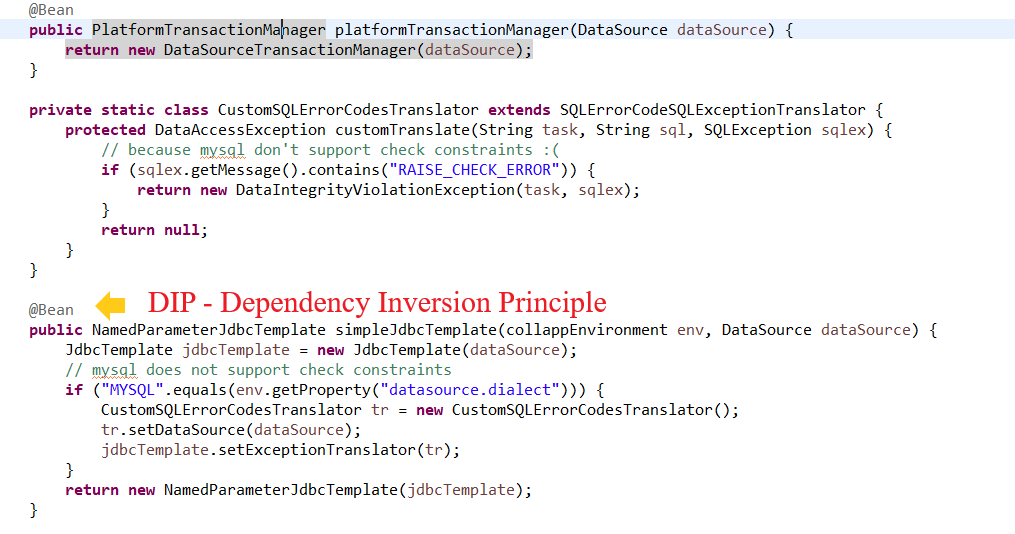




* **ISP: Interface Segregation Principle** - Implemented in any class which handles events.



* **DIP: Dependency Inversion Principle** - Injecting a dependency without changing the class internally. When we write a class and define functions within it then we will have dependency on the class to access the functions. We are using @bean which comes with spring framework which is basically like an injection without changing the class.





**How to Run**

Import the project folder into your Eclipse workspace.

Right-click on the directory in the project workspace window and select Run As…-> Maven build…

A screenshot of a computer

Description automatically generated

In the goals field, enter “maven clean” without the quotes select skip test, and then click on Apply and Run.

A screenshot of a computer

Description automatically generated

Once the build process is complete, navigate to Maven build… once again.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

This time, type in “jetty:run” in the goals field and Run it. Now check the skip tests option.

A screenshot of a cell phone

Description automatically generated

Once this is completed successfully, open a web browser at “localhost:8080”

A screenshot of a computer screen

Description automatically generated

This will bring you to the setup page as shown here:

Click on continue:

A screenshot of a computer

Description automatically generated

Choose the “Password” login type and continue:

A screenshot of a cell phone

Description automatically generated

Here, you can create your first user. This setup process runs only if there are no users currently setup on the system.

A screenshot of a computer screen

Description automatically generated

Click on continue, then verify the details and Activate the account.

A screenshot of a cell phone

Description automatically generated

This will bring you to the login page where you may login using the user credentials you have created.

A screenshot of a cell phone

Description automatically generated

Refer to the “COLLAPP.pptx” presentation file to view how to navigate and use the various features of CollApp