* @Spring MVC:

→ The MVC design pattern that consists of three modules:

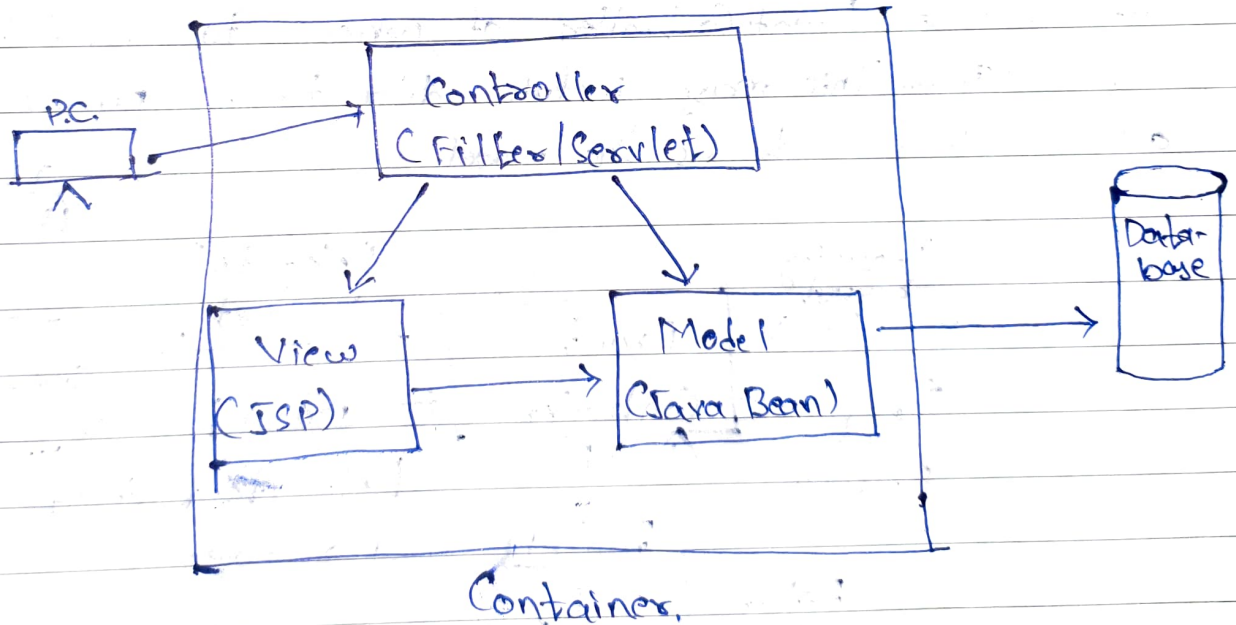   (i) model

      (ii) view & controller

      (iii) Controller.

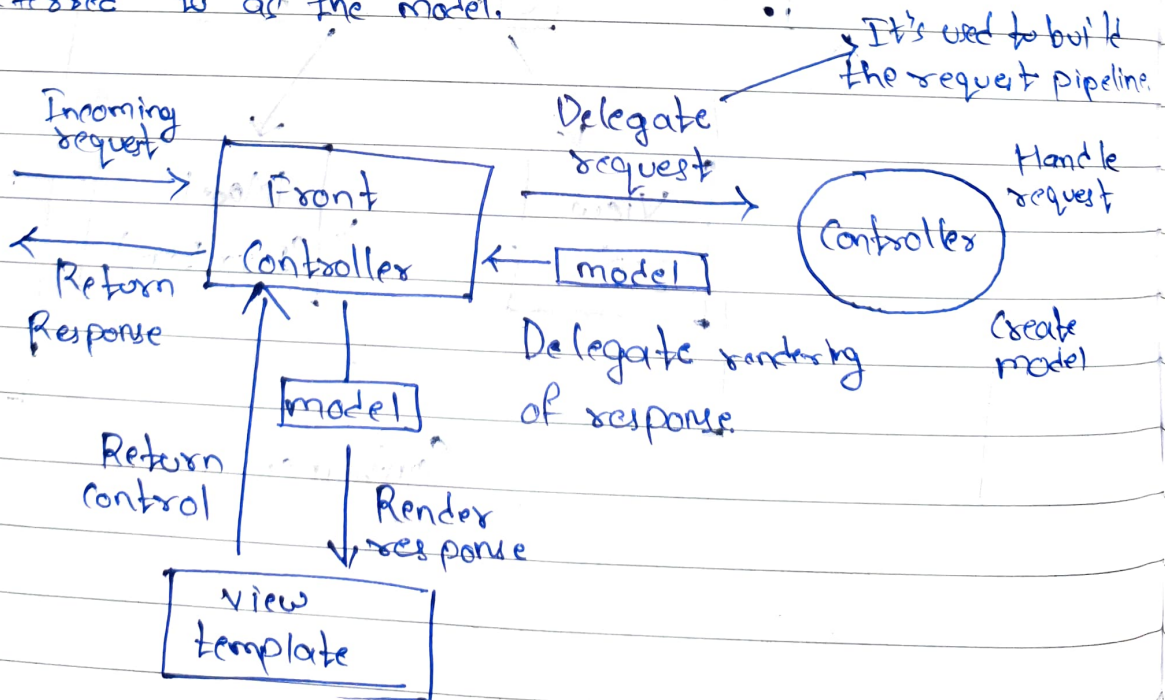(i) Model: The model represents the state(data) & business Logic for of the application.

(ii) View: The view is responsible for displaying data i.e. it represents the presentation.

(iii) Controller: The controller acts as an interface between the view & model. It intercepts all the requests to both a view or to the model.

Container.

* **Front Controller in Spring:**

→ The Spring Web model-view-controller (MVC) framework is designed around a Dispatcher Servlet that dispatches request to handlers with configurable handler mappings.

→ The DispatcherServlet follows the Front Controller pattern, providing a single point of entry into an application.

→ Wildcarded URLs all go through a single controller that delegates to sub controllers to get at the core application beyond (Model).

→ The response from the sub controller to the Front Controller informs the · Front Controller of where to· dispatch to next i.e. a View. This may include data to be passed to the view, also referred to as the model.

> It's used to build the request pipeline.

Incoming request → **Front Controller** ← Return Response

Delegate request → **Controller** Handle request

Front Controller ← model

Return control

model Delegate rendering of response Create model

Render response

View template

Servlet engine. (e.g. Tomcat).

maven project / web app creating
open others
Select maven proj.
catalog internal

classmate
Date _____
Page _____

* __Objectives__ of MVC:

→ A Java web application development framework based on the model-view-controller pattern.

→ Originally needed because the Java EE MVC arch., called Model 2, which used vanilla servlets/JSP, had many short comings.

Short comings: (i) in model 1, the ~~model~~ MVC they could all talk to each other. That way cleaned up to represent what we know as MVC model 2.
~~(ii) With model 2,~~

→ __Goals__ of MVC: (i) ~~simply~~ Simplify the task of creating web application.

(ii) Encourage good architecture by providing a strong MVC framework with clear separation of roles.
(i.e. controller, model, view)

(iii) Integrate ~~eas~~ easily with the rest of Spring.