MVC Architecture in Javaweb

Objectives

+ Understand MVC Architecture pattern

+ Apply MVC Architecture pattern in Java Web Application

What is MVC

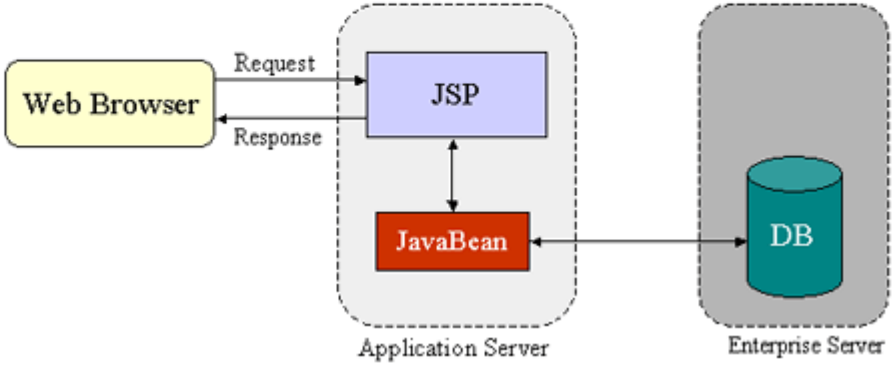
- Model–view–controller (usually known as MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements.
- MVC proposes the construction of three distinct components. One side for the representation of information, and on the other hand for user interaction
  - Model : The central component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic, and rules of the application.
  - View : Any representation of information such as a chart, diagram, or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
  - Controller: Accepts input and converts it to commands for the model or view.

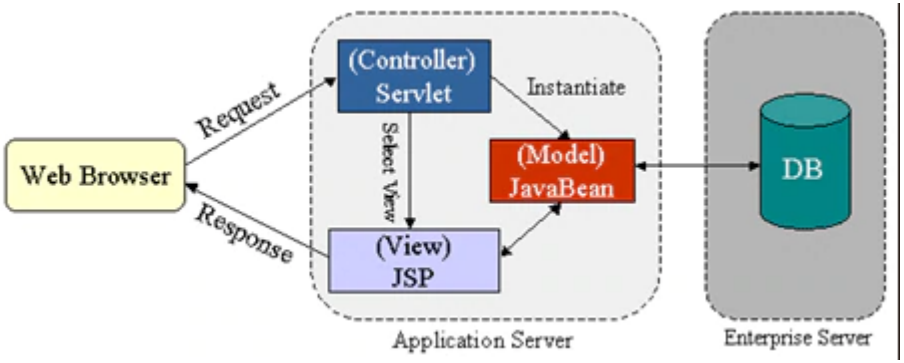
MVC Architecture

- The JSP specification presents two approaches for building web applications using JSP pages

- The JSP page is responsible for processing requests and sending back replies to clients

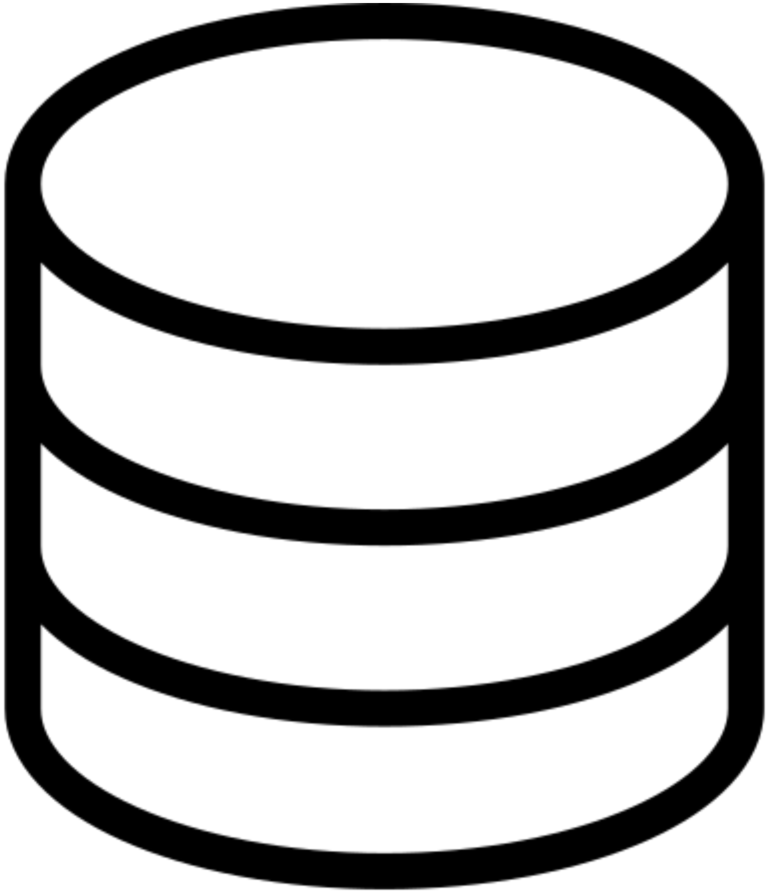
- Integrates the use of both servlets and JSP pages
- JSP pages are used for the presentation layer, and servlets for processing tasks
- This model promotes the use of the Model View Controller (MVC) architectural style design pattern





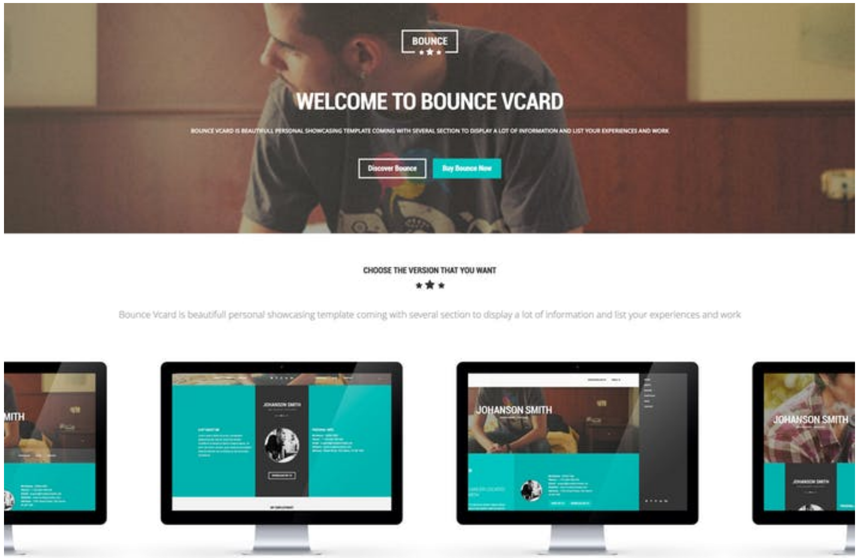
The Model

- It is the specific representation of the information with which the system operates. Logic ensures the integrity of data and allows to derive it



The View

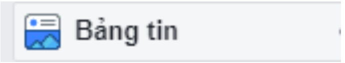
Represents the model in a suitable format to interact and access the data, usually call "User Interface" (GUI Java, HTML, XML)

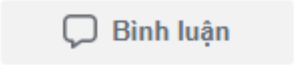


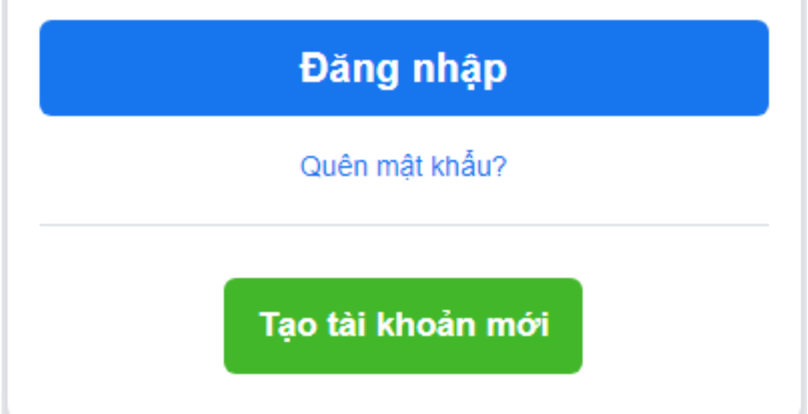
Bounce Vcard is beautifull personal showcasing template coming with several section to display a lot of information and list your experiences and work

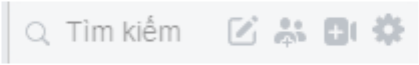
The Controller

- It is the link between the view and the model, is responsible for receiving and responding to events, typically user actions invokes changes on the model and probably in the view



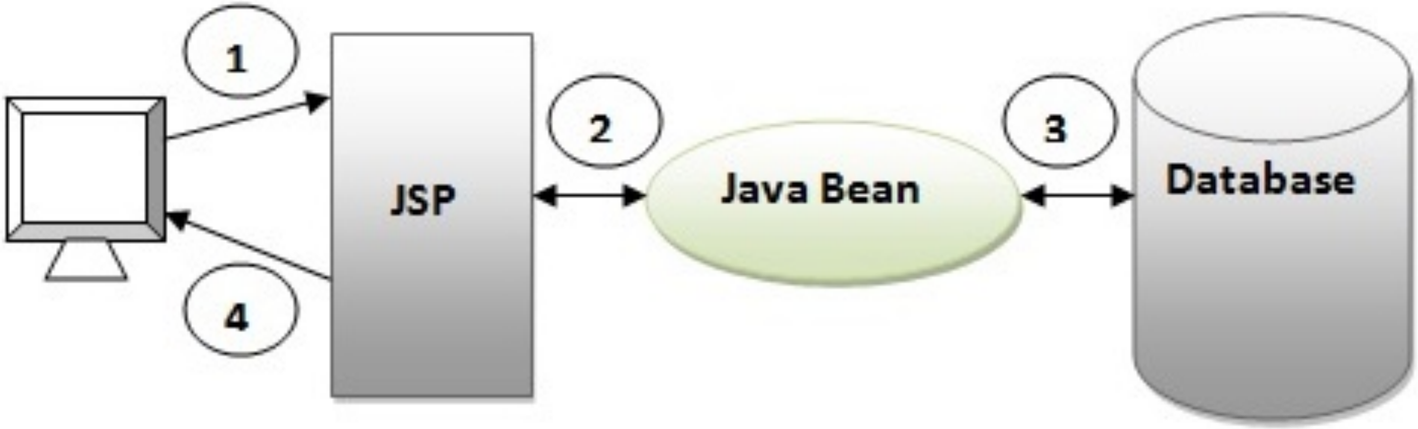






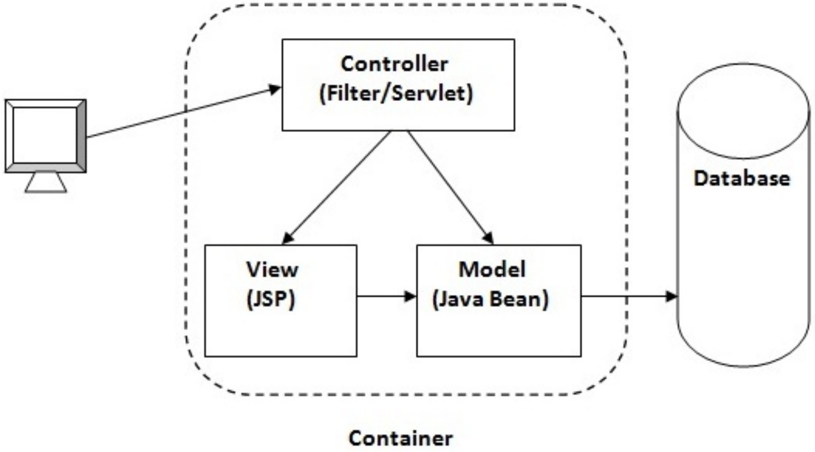
MVC1 and MVC2

- MVC1:
  - Servlet and JSP are the main technology to develop the web application
  - Pros: Easy and Quick to develop web application
  - Cons:
    - Navigation control is decentralized
    - Time consuming for custom JSP tag
    - Hard to extend



MVC1 and MVC2

- MVC2:
  - Base on MVC design pattern which consist of three modules model, view, and controller
  - Pros
    - Navigation control is centralized
    - Easy to maintain
    - Easy to extend
    - Easy to test
    - Better separation of concerns
  - Cons
    - We need to write the controller code self. If we change the controller code, we need to recompile the class and redeploy the application.



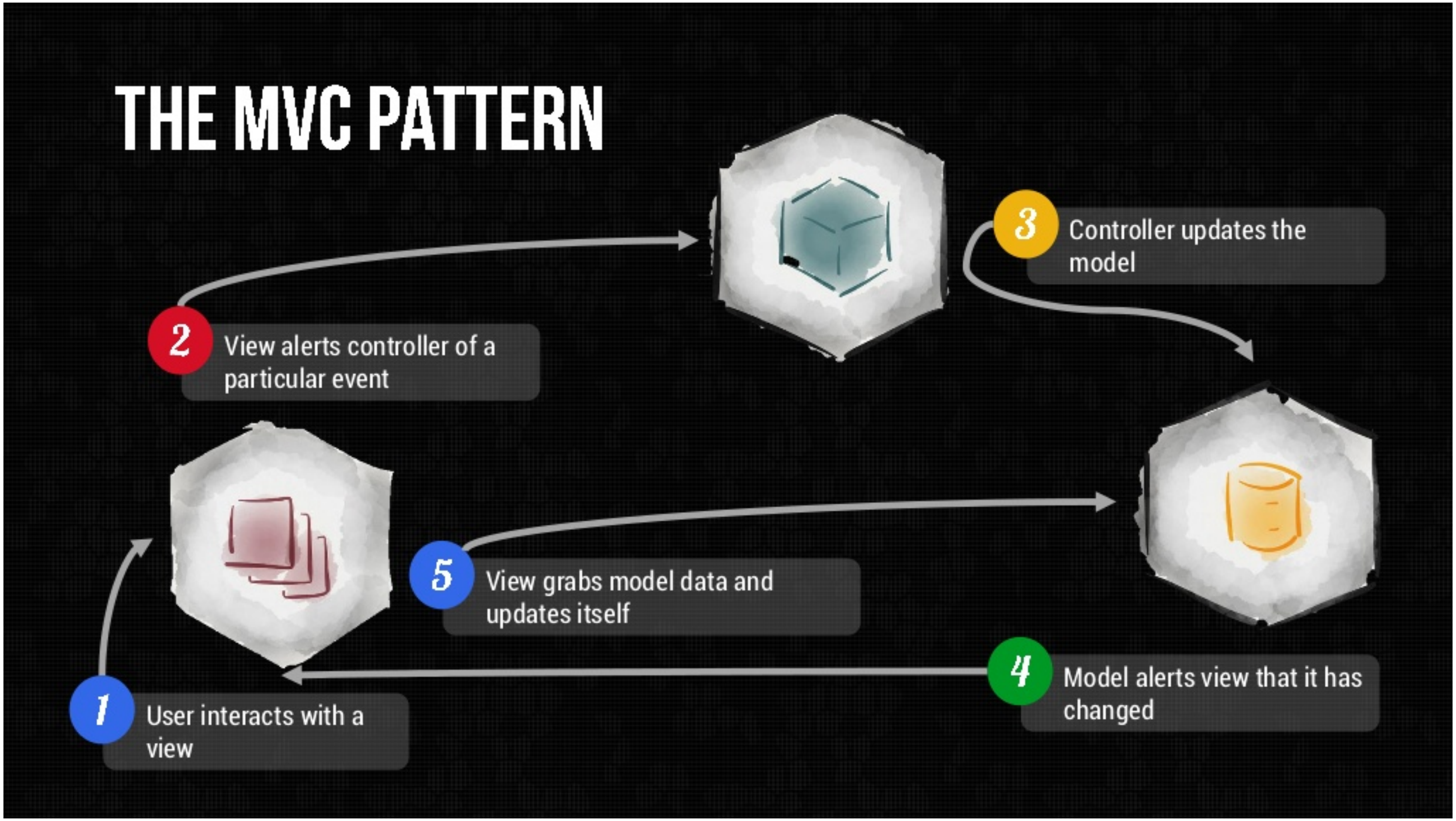
Benefits

- Organization
- Rapid Application Development
- Reusing Code
- Parallel development
- It represent the same information in different ways
- The views and application behavior should reflect the manipulations of the data immediately
- It allow different user interface standards or port it to other environments where the application code should not be affected

The MVC pattern

- Control flow
  1. The user performs an action on the interface
  2. The controller takes the input event
  3. The controller notifies the user action to the model, which may not involve a change of state of the model
  4. It generates a new view. The view takes the data model
  5. The user interface waits for another user interaction, which starts a new cycle

The MVC pattern



THE MVC PATTERN

1 User interacts with a view

2 View alerts controller of a particular event

3 Controller updates the model

4 Model alerts view that it has changed

5 View grabs model data and updates itself

Where it can use?

- It applies to all ty of systems
- And technologies (Java, Python, Ruby, Perl, SmallTalk, …)

MVC in Java

- Model: made by the developer
- View: set of object of classes that inherit from java.awt.Component
- Controller: the event processing thread, which captures and propagates the event to the view and the model. Treatment classes of events sometime as anonymous classes) that implement EventListener type interfaces (ActionListener, MouseListener, WIndowListener,…)

MVC Framework

- Spring: one of the oldest Java Web Frameworks, but as one says: Old but Gold. Up to now, it is still utilized and constantly changing, developing more changes with Java.

- Struts: a free and open source framework for creating simple Java apps. It is mainly used to create the foundation for web apps, and works on the pattern "MVC" - Model-View-Controller.

- JavaServer Faces: supported by Oracle, so it produces a super long and complicated document.

Summary

MVC Introduction

- What is MVC?

MVC Architecture

- The Model
- The View
- The Controller
- MVC1 and MVS2

Benefits

- MVC pattern
- Where is can use?
- MCV in Java

Framework