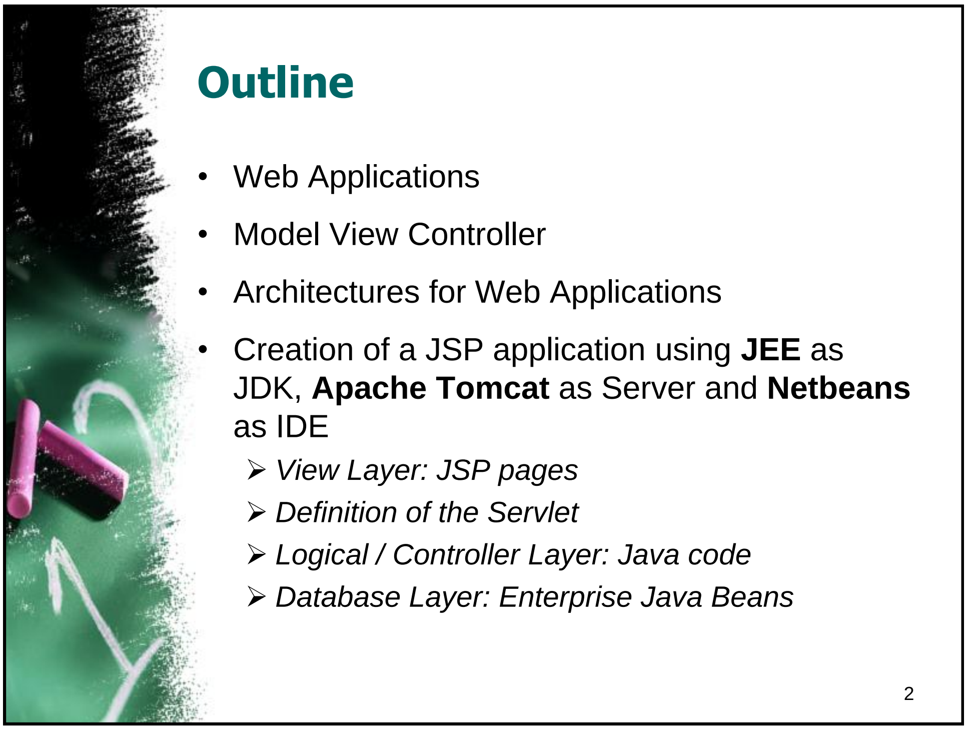
CSI 2132 Lab 8

Web Programming
JSP

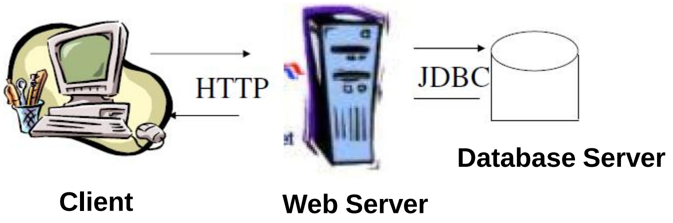


Outline

- Web Applications
- Model View Controller
- Architectures for Web Applications
- Creation of a JSP application using **JEE** as JDK, **Apache Tomcat** as Server and **Netbeans** as IDE
  - ➢ *View Layer: JSP pages*
  - ➢ *Definition of the Servlet*
  - ➢ *Logical / Controller Layer: Java code*
  - ➢ *Database Layer: Enterprise Java Beans*

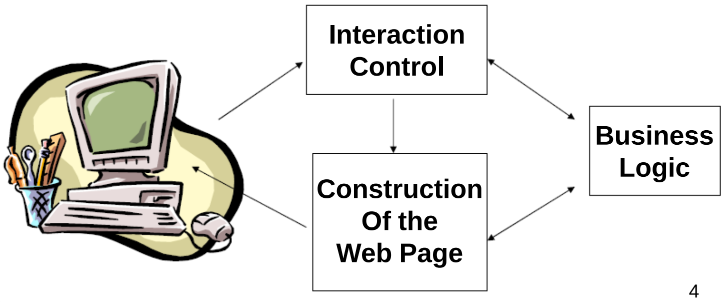
Web Applications

- To add logic (program code) inside a web server, Sun Microsystems extended the Java language with the introduction of the **API Java Servlets**.



Model View Controller

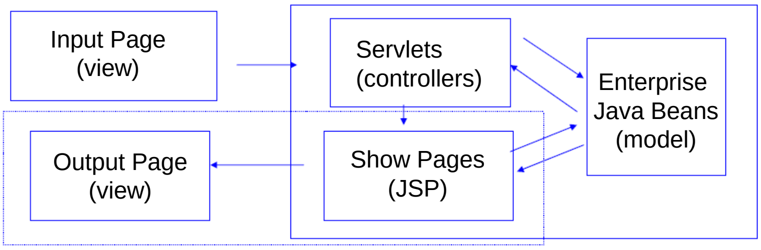
- Web applications are usually build following the three layers model.
- According to this model three separate and perfectly defined process or modules are executed in different platforms



Model View Controller

- An architecture that separates the application components:

  - ***Model:*** *Represents the data*
  - ***View:*** *User Interface*
  - ***Controller:*** *Handles the user input and the application flow. It is implemented as a Java Servlet.*



Input Page
(view)

Servlets
(controllers)

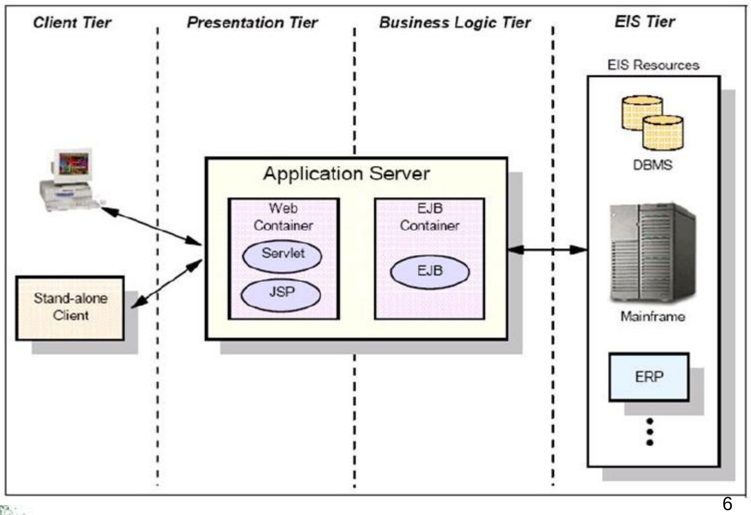
Enterprise
Java Beans
(model)

Output Page
(view)

Show Pages
(JSP)

Architectures for Web Applications

Model View Controller

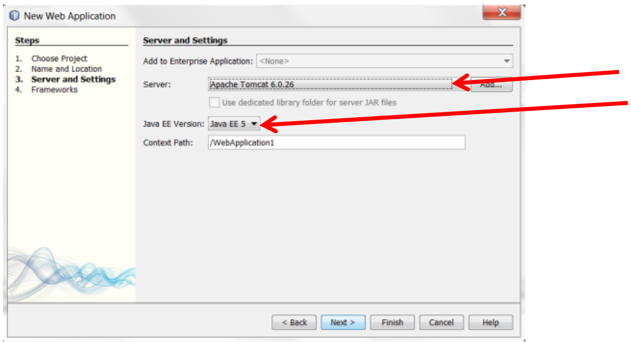


Architectures for Web Applications

- Model
  - ➢ *Encapsulates the business domain. It is completely independent from the Controller and the View.*
- View:
  - ➢ *Presentation of the model, it can access to the the model, but never change it state. It can be notified when there is a change in the model state.*
- Controller:
  - ➢ *Responds to the client request, executing the appropriate action and creating the pertinent model.*

Web Project in Netbeans

- File → New Project…
  - ➢ *Java Web*
    - Web Application
  - ➢ *Give it a name*
  - ➢ *Select proper Server settings*



New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

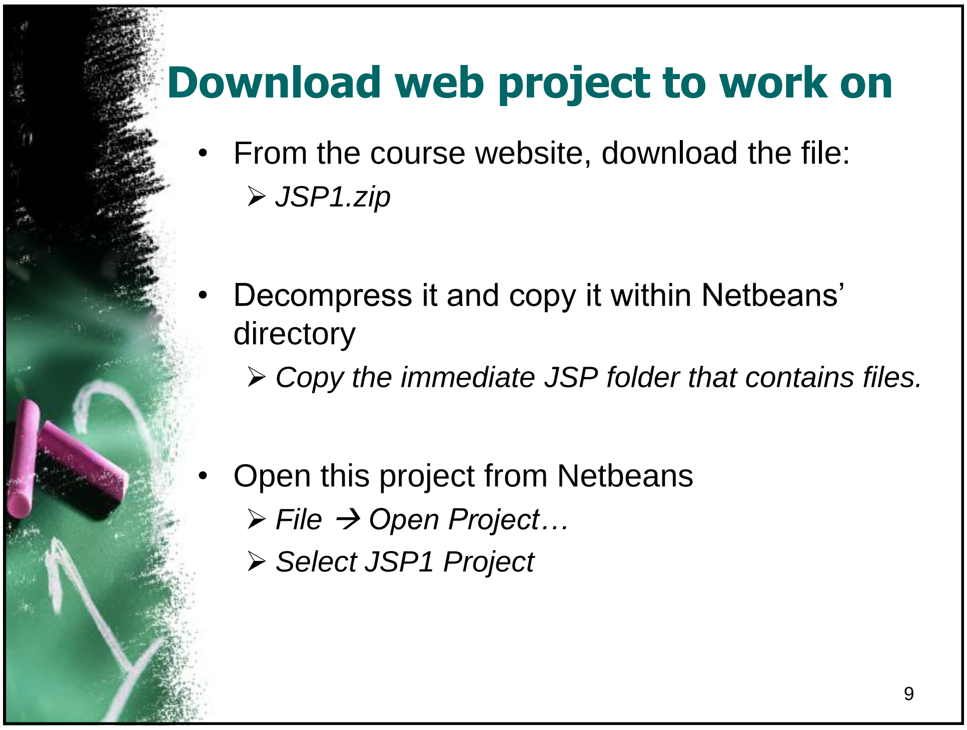
Server: Apache Tomcat 6.0.26

□ Use dedicated library folder for server JAR files

Java EE Version: Java EE 5

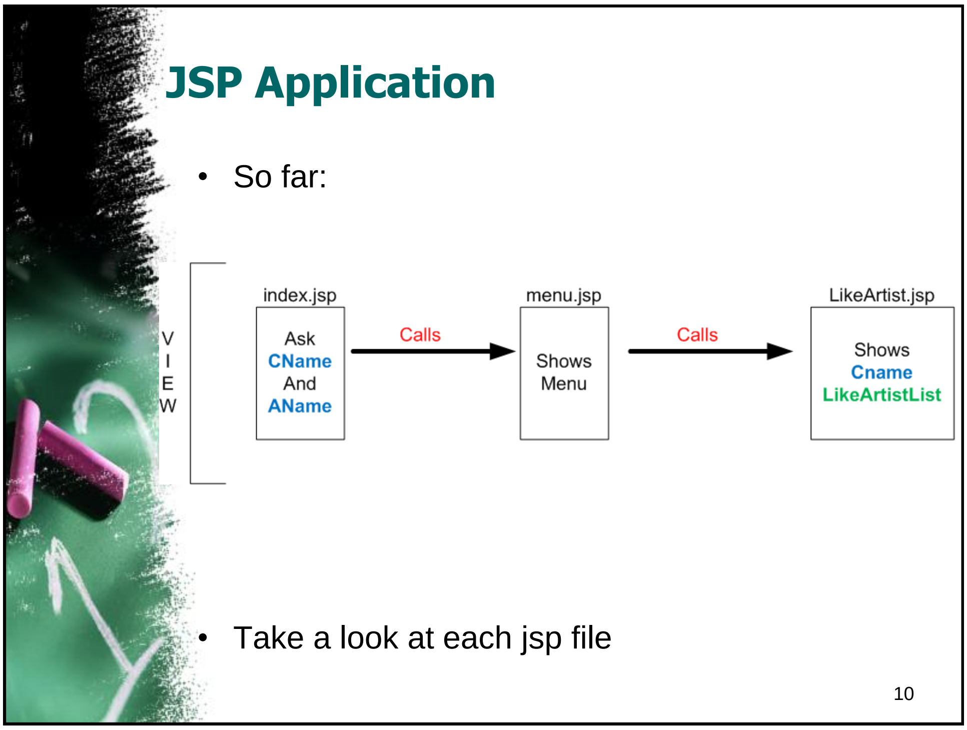
Context Path: /WebApplication1

  - ➢ *Don’t select any Framework*



Download web project to work on

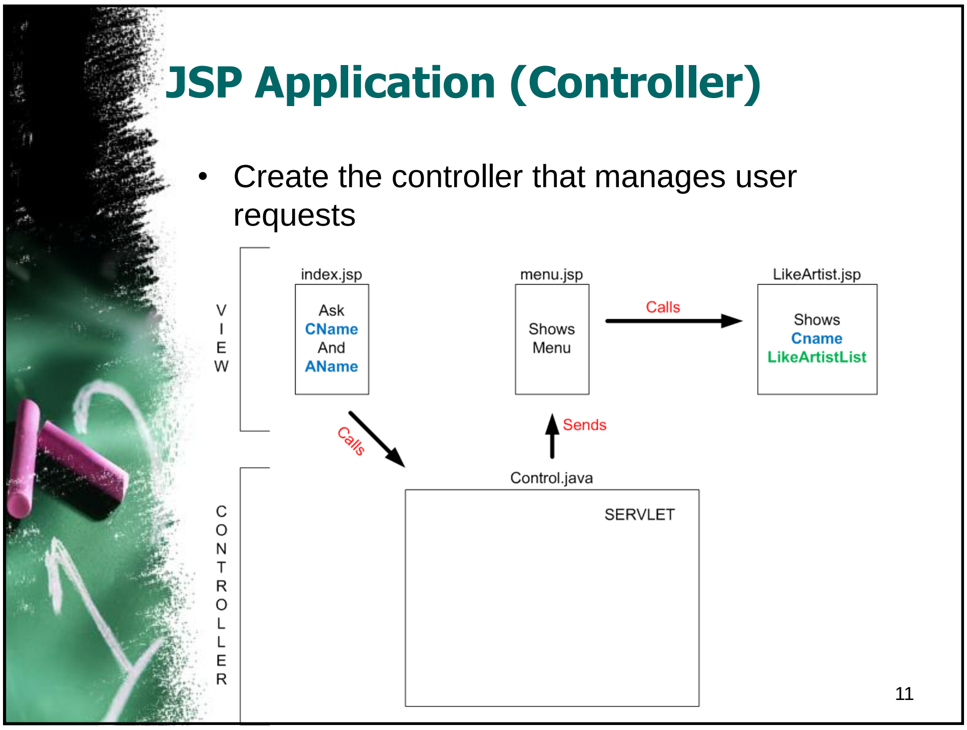
- From the course website, download the file:
  - *JSP1.zip*
- Decompress it and copy it within Netbeans' directory
  - *Copy the immediate JSP folder that contains files.*
- Open this project from Netbeans
  - *File → Open Project…*
  - *Select JSP1 Project*



JSP Application

- So far:

- Take a look at each jsp file



JSP Application (Controller)

- Create the controller that manages user requests

VIEW

index.jsp

Ask CName And AName

menu.jsp

Shows Menu

Calls

LikeArtist.jsp

Shows Cname LikeArtistList

Calls

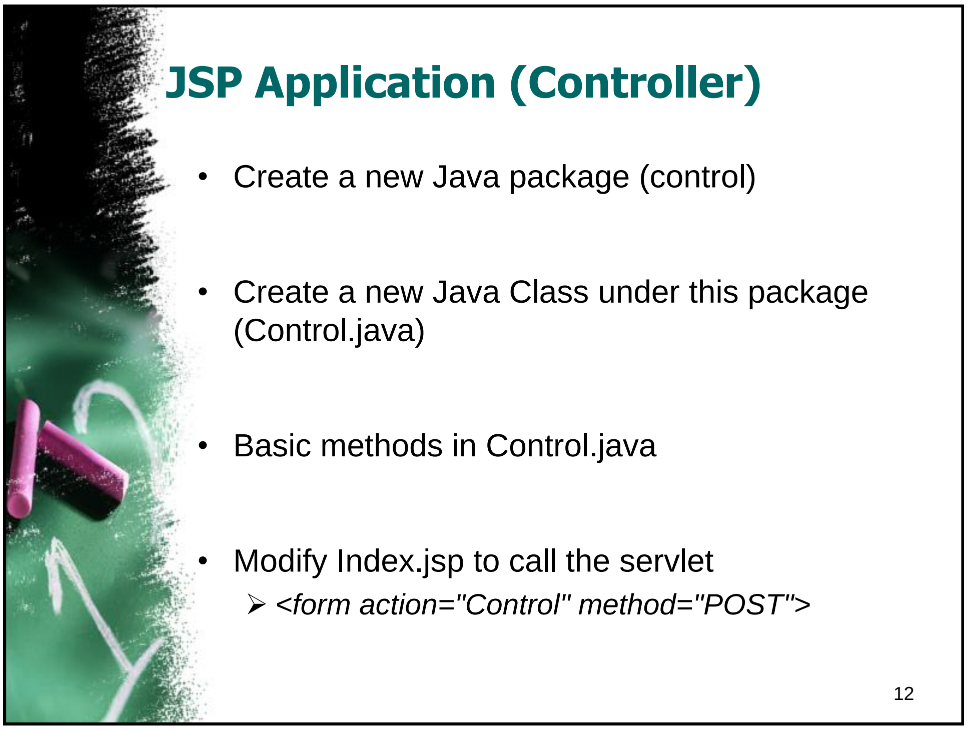
Sends

CONTROLLER

Control.java

SERVLET

11



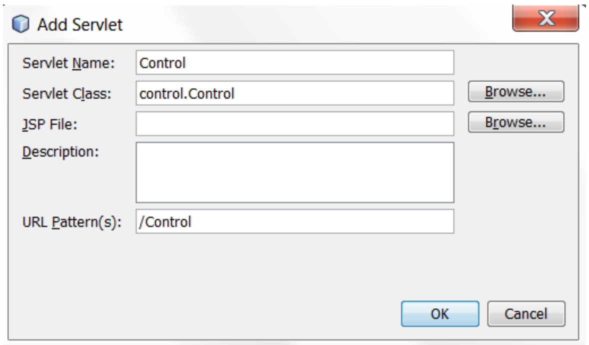
JSP Application (Controller)

- Create a new Java package (control)
- Create a new Java Class under this package (Control.java)
- Basic methods in Control.java
- Modify Index.jsp to call the servlet
  - *<form action="Control" method="POST">*

12

JSP Application (Controller)

- Modify the web.xml file to use control as servlet
  - ➢ *WEB-INF → Web.xml*
    - Servlets tab → Add Servlet Element



Add Servlet

Servlet Name: Control

Servlet Class: control.Control

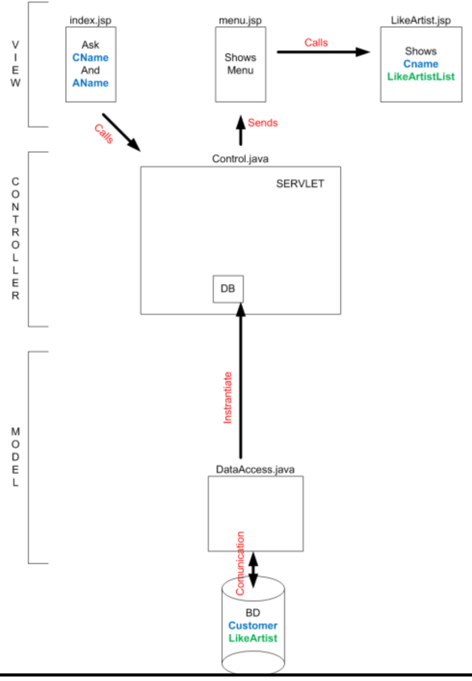
JSP File:

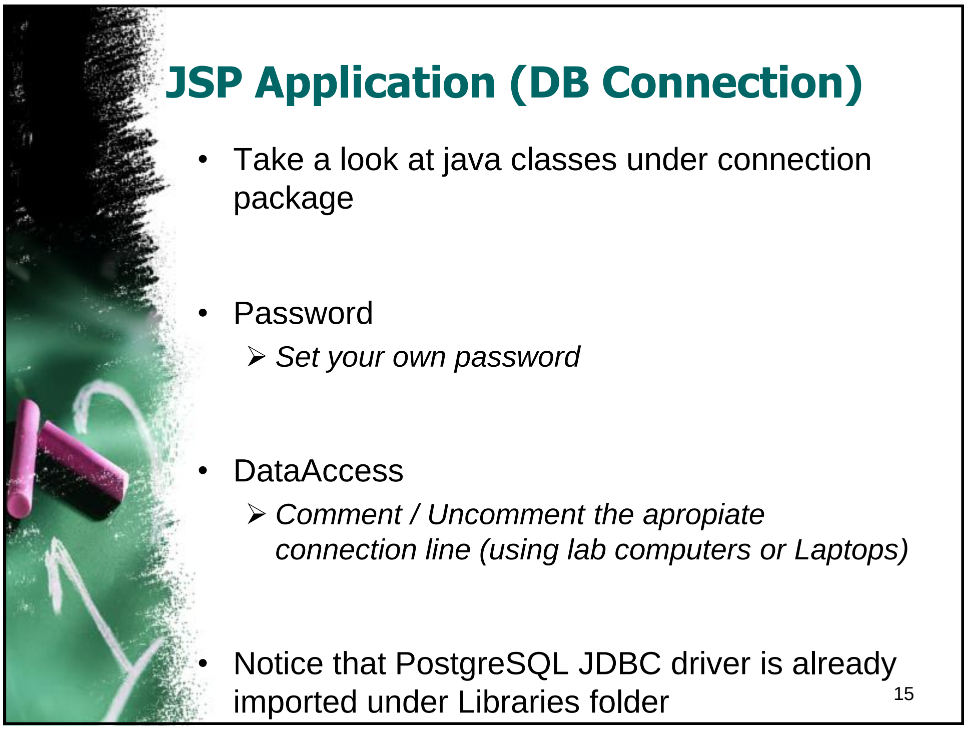
Description:

URL Pattern(s): /Control

- Test the application so far

JSP Application (DB Connection)

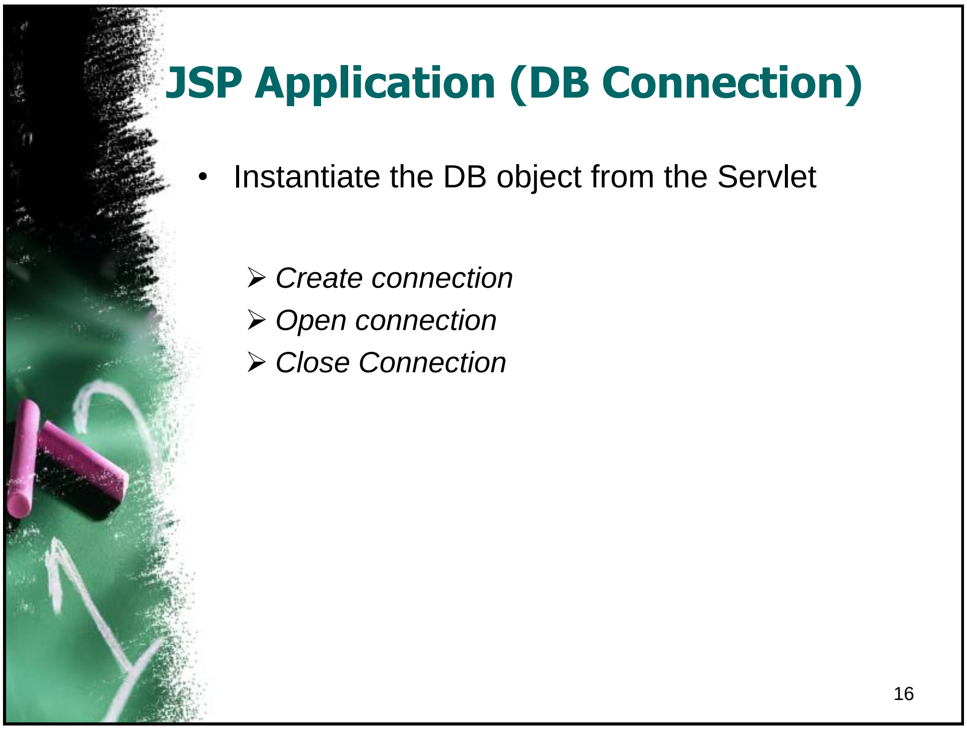




JSP Application (DB Connection)

- Take a look at java classes under connection package
- Password
  - *Set your own password*
- DataAccess
  - *Comment / Uncomment the apropiate connection line (using lab computers or Laptops)*
- Notice that PostgreSQL JDBC driver is already imported under Libraries folder

15

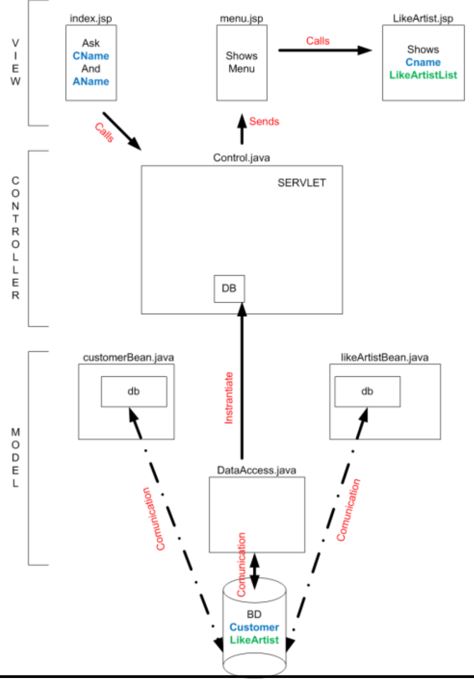


JSP Application (DB Connection)

- Instantiate the DB object from the Servlet
  - *Create connection*
  - *Open connection*
  - *Close Connection*

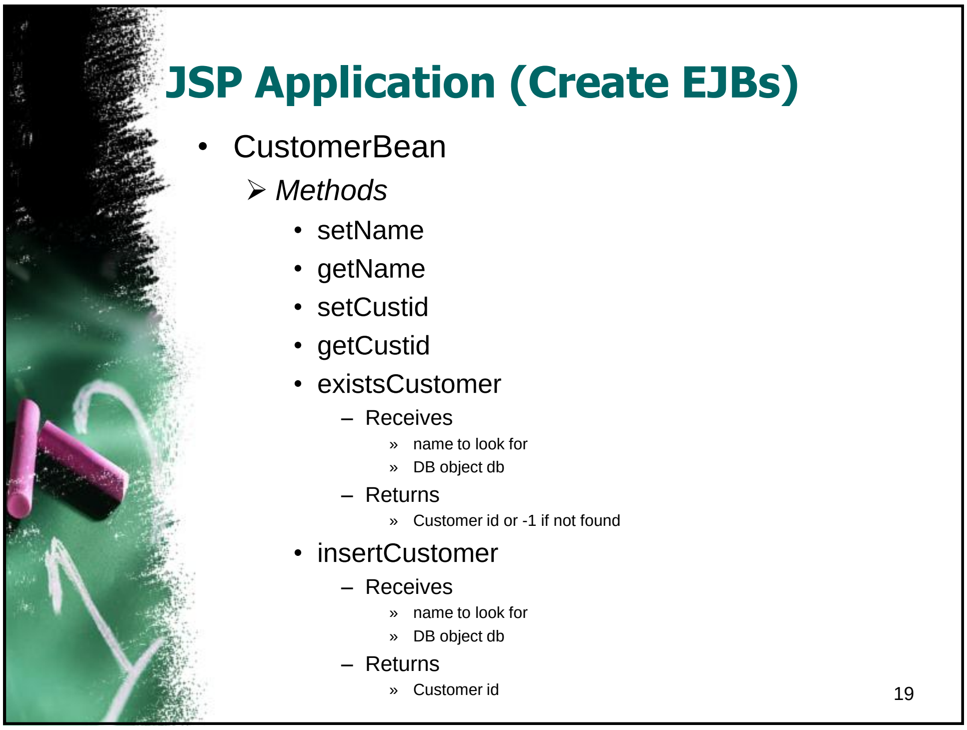
16

JSP Application (EJBs)



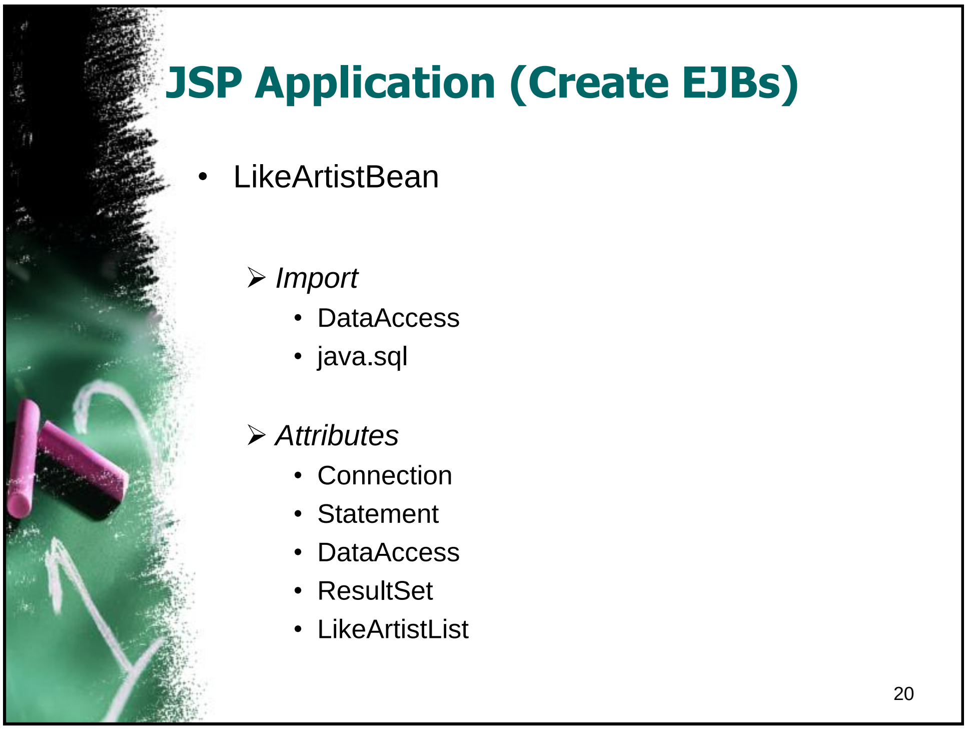
JSP Application (Create EJBs)

- CustomerBean
  - *Import*
    - DataAccess
    - java.sql
  - *Attributes*
    - Connection
    - Statement
    - ResultSet
    - Customer Name
    - Customer id



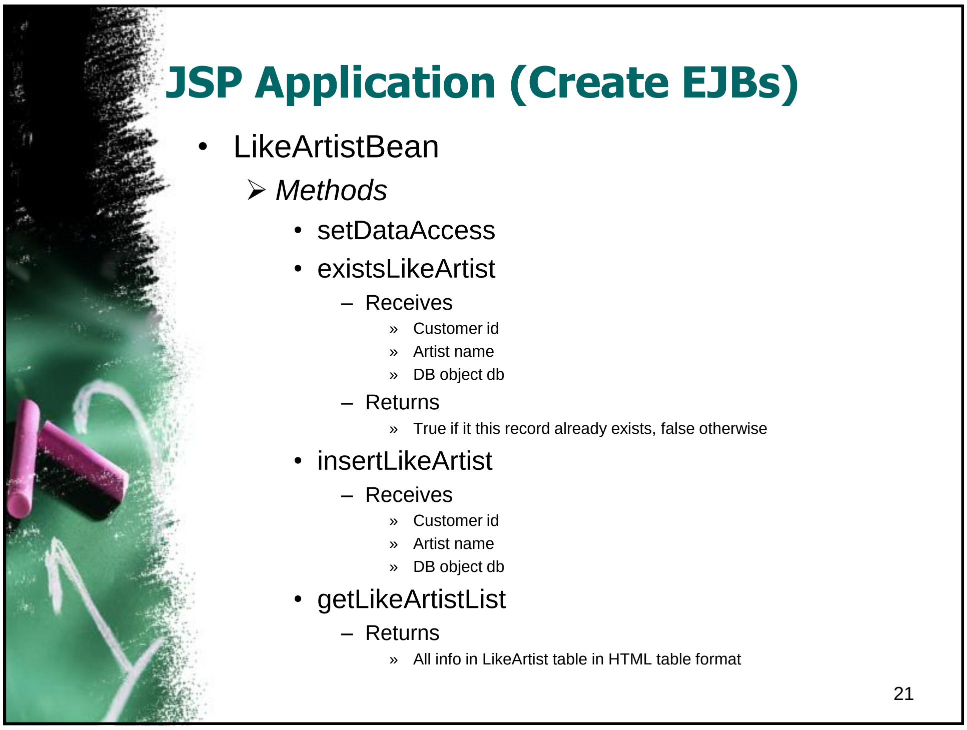
JSP Application (Create EJBs)

- CustomerBean
  - ➢ *Methods*
    - setName
    - getName
    - setCustid
    - getCustid
    - existsCustomer
      - – Receives
        - » name to look for
        - » DB object db
      - – Returns
        - » Customer id or -1 if not found
    - insertCustomer
      - – Receives
        - » name to look for
        - » DB object db
      - – Returns
        - » Customer id



JSP Application (Create EJBs)

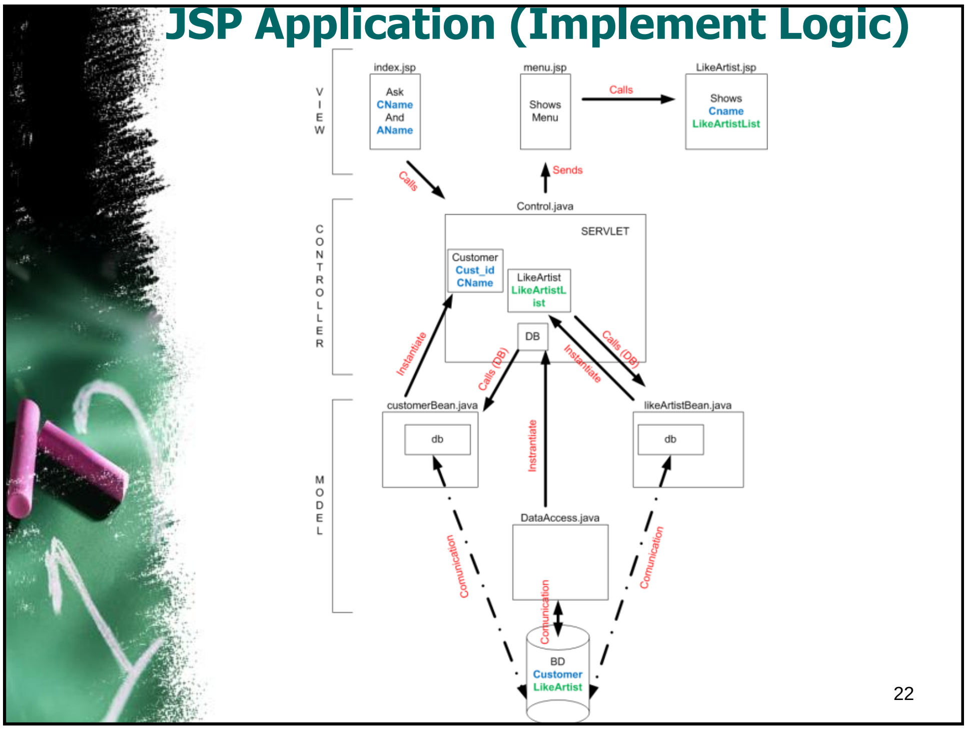
- LikeArtistBean
  - ➢ *Import*
    - DataAccess
    - java.sql
  - ➢ *Attributes*
    - Connection
    - Statement
    - DataAccess
    - ResultSet
    - LikeArtistList

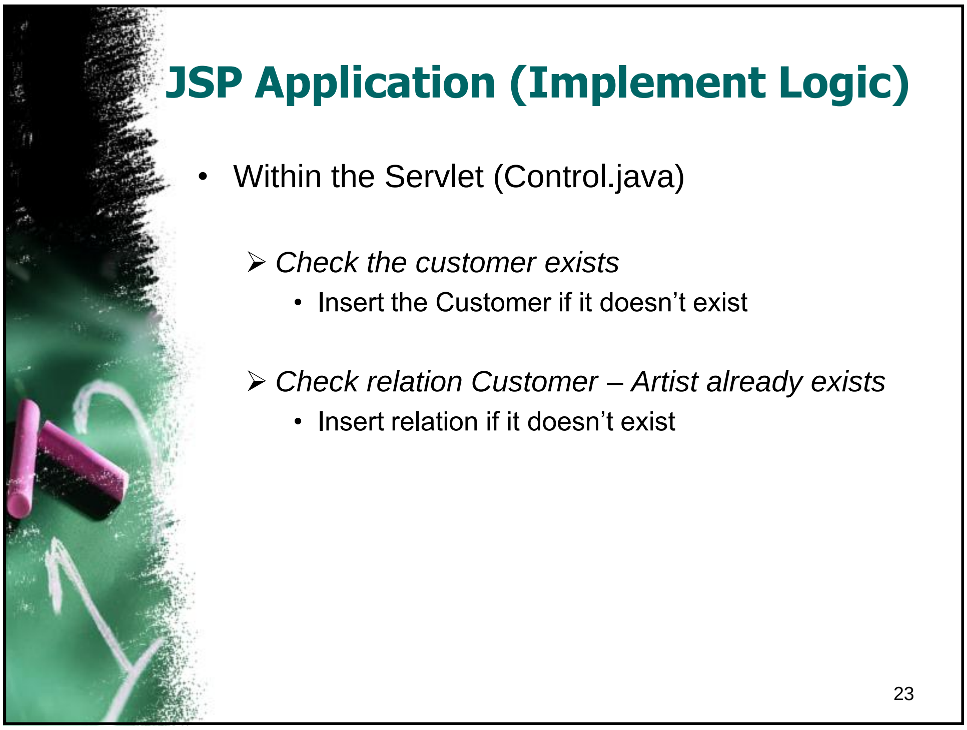


JSP Application (Create EJBs)

- LikeArtistBean
  - ➢ *Methods*
    - setDataAccess
    - existsLikeArtist
      - – Receives
        - » Customer id
        - » Artist name
        - » DB object db
      - – Returns
        - » True if it this record already exists, false otherwise
    - insertLikeArtist
      - – Receives
        - » Customer id
        - » Artist name
        - » DB object db
    - getLikeArtistList
      - – Returns
        - » All info in LikeArtist table in HTML table format

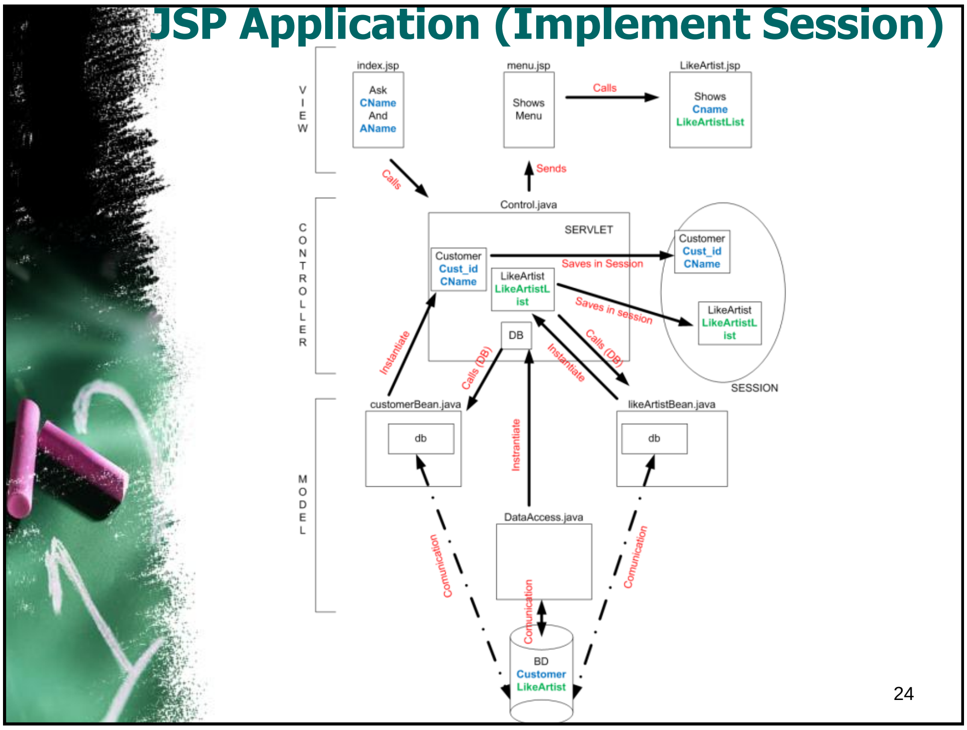
JSP Application (Implement Logic)





JSP Application (Implement Logic)

- Within the Servlet (Control.java)
  - ➢ *Check the customer exists*
    - Insert the Customer if it doesn't exist
  - ➢ *Check relation Customer – Artist already exists*
    - Insert relation if it doesn't exist

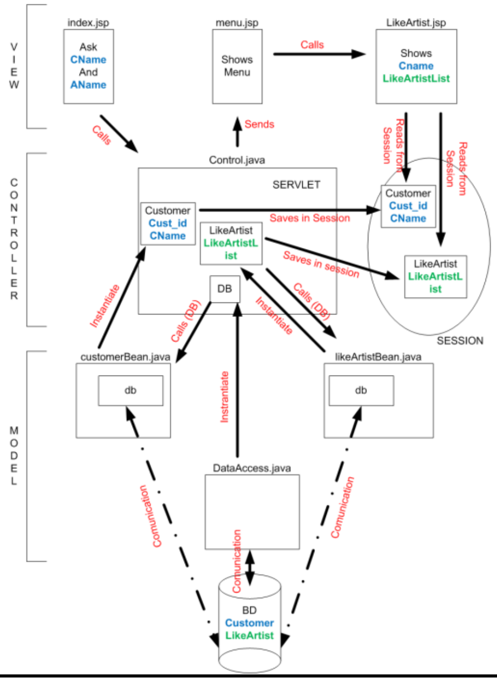


JSP Application (Implement Session)

JSP Application (Implement Session)

- Create Session within our Servlet (Control.java)
- Save EJBs with information within Session
- Check Session is valid in menu.jsp
- Close Session in closesession.jsp

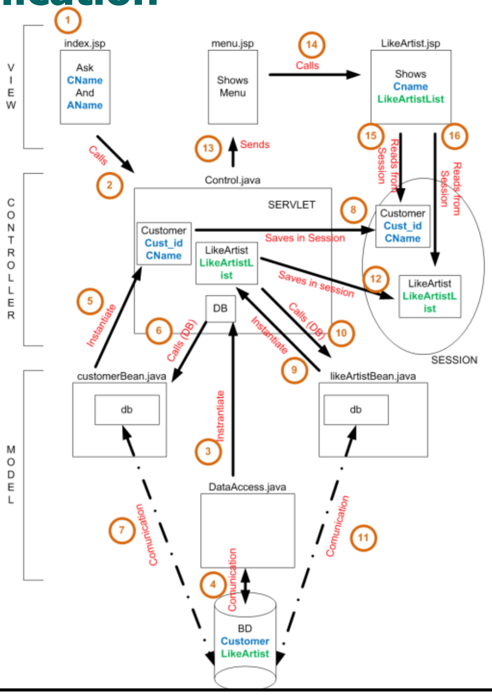
JSP Application (Show LikeArtist tbl)



JSP Application (Show LikeArtist tbl)

- Retrieve information from Session and show LikeArtist table

JSP Application



CSI 2132 Lab 8

Web Programming
JSP