

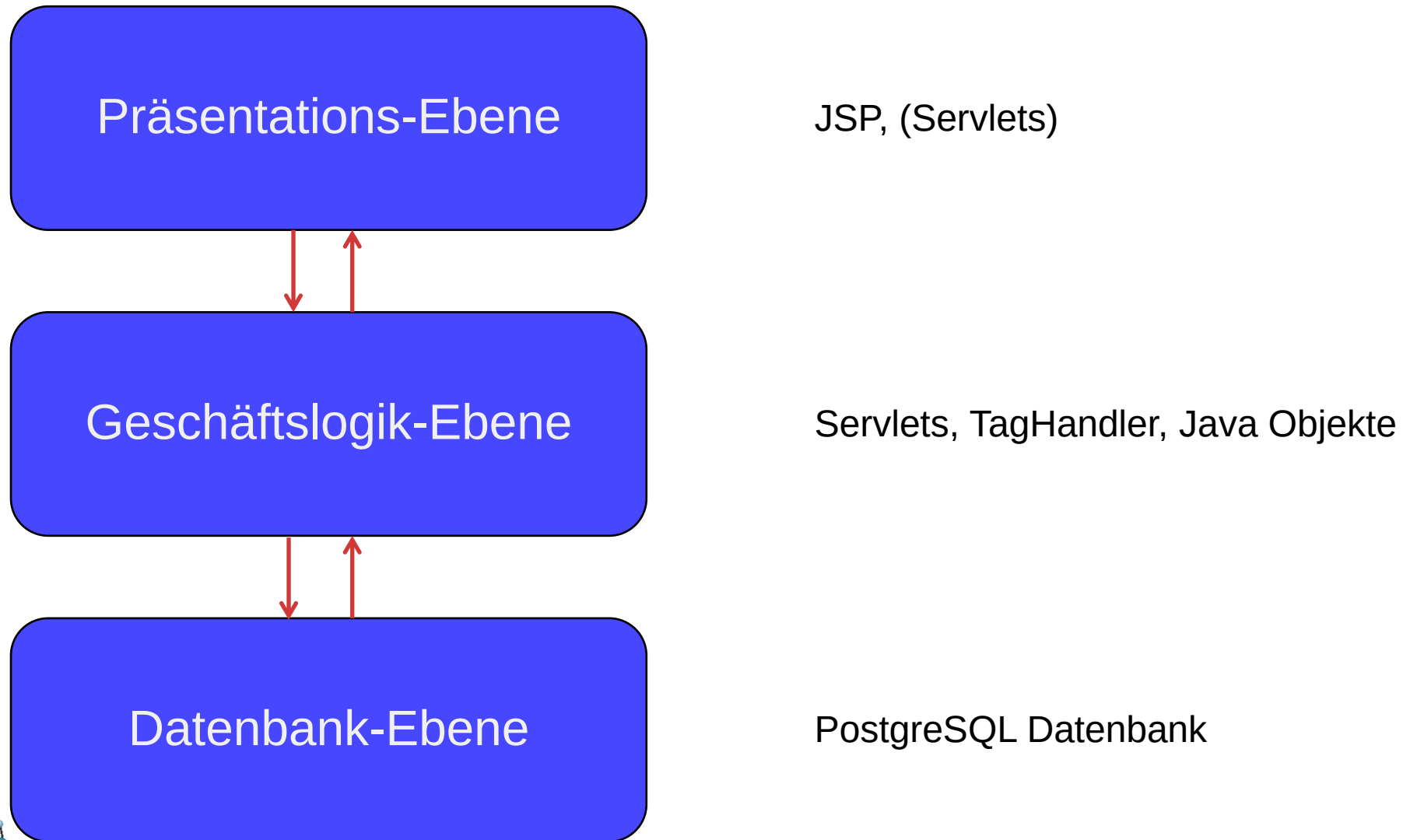
Praktikum DBAE / WI

Übersicht für Testat 3

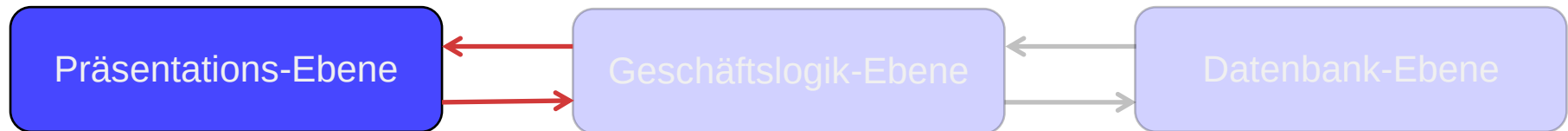
Pascal Reuss, M.Sc.

Raum A08b Spl.
Email: reusspa@uni-hildesheim.de

Architekturebenen

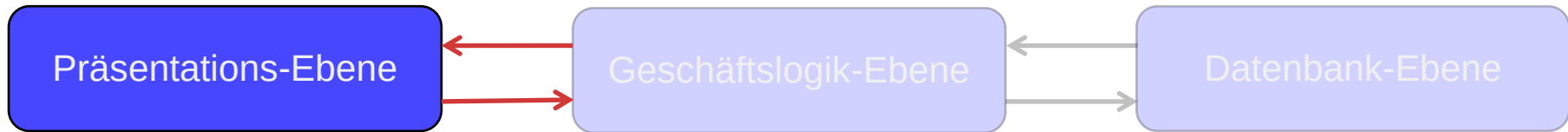


Präsentations-Ebene



- JSP mit Formular
- Mehrere Eingabefelder
- Daten werden an Servlet gesendet
- Namen der Eingabefelder = Parameternamen in Servlet

Präsentations-Ebene



```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1," pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <title>Eingabemaske Testat 3</title>
</head>
<body>
  <form method="post" action="ServletSearch" >
    <label>Liedtitel:</label><input type="text" name="liedtitel" ><br>
    <label>Mediumtitel:</label><input type="text" name="mediumtitel" ><br>
    <input type="submit" value="Suche starten" >
  </form>
</body>
</html>
```

Geschäftslogik-Ebene



- Servlet zum Verarbeiten der Anfrage
- Java Klasse für die DB Verbindung
- Java Klasse für das SQL Statement
 - Fallüberprüfung für mehrere Parameter
- TagHandler (Java Klasse) für eigenen Tag

Geschäftslogik-Ebene



```
public void doPost(HttpServletRequest req, HttpServletResponse res) throws  
ServletException, IOException {
```

```
//Parameter auslesen
```

```
String liedtitel = req.getParameter("liedtitel");
```

```
String mediumtitel = req.getParameter("mediumtitel");
```

```
//Neues SearchObject (Java Klasse für SQL Statement)
```

```
Search searchObject = new Search();
```

```
//Methode aufrufen um Suche durchführen
```

```
searchObject.doSearch(liedtitel, mediumtitel);
```

```
//Methode aufrufen, um das Ergebnis anzuzeigen
```

```
PrintWriter out = res.getWriter();
```

```
out.println(searchObject.displayResult());
```

```
out.close();
```

**Servlet –
doPost Methode**

Geschäftslogik-Ebene



Class DBVerbindung {

```
private Connection con;  
private String driver = "org.postgresql.Driver";           // DB Treiber  
private String DB_SERVER = "localhost:5432";               // DB Server Adresse  
private String DB_NAME = „dbae_gruppe_00“;                // DB Name  
private String user = „dbae_gruppe_00“;                    // DB User  
private String password = „password“;                      // DB Passwort  
private String url = "jdbc:postgresql://" + DB_SERVER + "/" + DB_NAME;
```

DBVerbindung

Geschäftslogik-Ebene

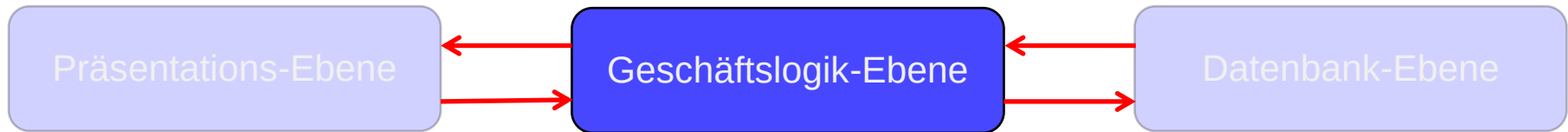


```
//Constructor
public DBVerbindung() {
    try {
        Class.forName(driver);
        this.con = DriverManager.getConnection(url, user, password);
    } catch (SQLException se) {
        system.out.println(se.getMessage());
    } catch (ClassNotFoundException ce) {
        system.out.println(ce.getMessage());
    }
}

//Methode um Verbindung zu bekommen
public Connection getConnection() {
    return this.con;
}
```

Fortsetzung –
DBVerbindung

Geschäftslogik-Ebene



```
Class SearchObject {
```

```
    private Connection con;  
    private ResultSet result;
```

```
    //Constructor
```

```
    public SearchObject() {  
        this.result = null;  
        //Verbindung holen aus Klasse DBVerbindung  
        this.con = new DBVerbindung().getConnection();  
    }
```

Klasse für
SQL Statement

Geschäftslogik-Ebene



```
//Methode zur Durchführung der Suche
public void doSearch(String liedtitel, String mediumtitel) {
    //Grundlegendes SQL Statement
    String sql =
        „SELECT * FROM lied INNER JOIN medium WHERE true“;

    //Leerer where-String
    String where = “”;

    //Boolean Variablen zur Prüfung ob Parameter gestetzt sind
    boolean paramlied = false;
    boolean parammedium = false;
```

Fortsetzung -
Klasse für
SQL Statement

Geschäftslogik-Ebene



```
//Fall Parameter liedtitel ist gesetzt
if (!liedtitel.equals("") && liedtitel != null) {
    where += „ AND liedtitel iLIKE %?%“;
    paramlied = true;
}
```

```
//Fall Parameter mediumtitel ist gesetzt
if (!mediumtitel.equals("") && mediumtitel != null) {
    where += AND mediumtitel iLIKE %?%“;
    parammedium = true;
}
```

```
//SQL Statement & Where Statement verbinden
String completeSQL = sql + where;
```

```
//Prepared Statement erstellen
PreparedStatement pstmt = this.con.getPreparedStatement(),
pstmt.prepareStatement(completeSQL);
```

Fortsetzung -
Klasse für
SQL Statement

Geschäftslogik-Ebene



```
//Fallunterscheidung
if (paramlied && parammedium) {
    pstmt.setString(1, liedtitel);
    pstmt.setString(2, mediumtitel);

} else if (paramlied && !parammedium) {
    pstmt.setString(1, liedtitel);

} else if (!paramlied && parammedium) {
    pstmt.setString(1, mediumtitel);
}
```

```
//Prepared Statement ausführen
this.result = pstmt.executeQuery();
```

```
}
```

Fortsetzung -
Klasse für
SQL Statement

Geschäftslogik-Ebene



```
//Methode zur Ausgabe des Ergebnisses  
public String displayResult() {
```

```
    while (rs.next()) {  
        //jeden Datensatz in ein Objekt speichern  
        Lied lied = new Lied(rs.get(1), rs.get(2), rs.get(3))  
        //Objekt einer Liste hinzufügen  
        liedliste.add(lied);  
    }
```

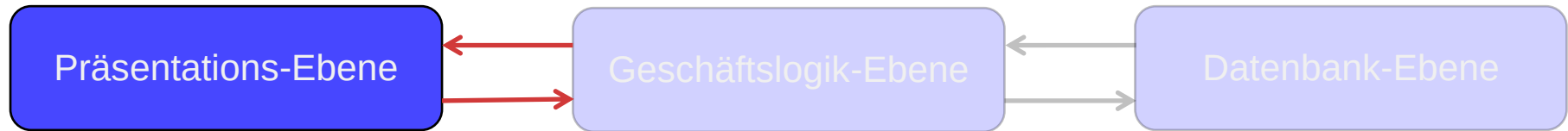
```
}
```

```
// Setzen des Listen-Objekts als Attribut  
request.setAttribute(„liste“, liedliste);
```

```
//weiterleiten auf gewünschte Webseite
```

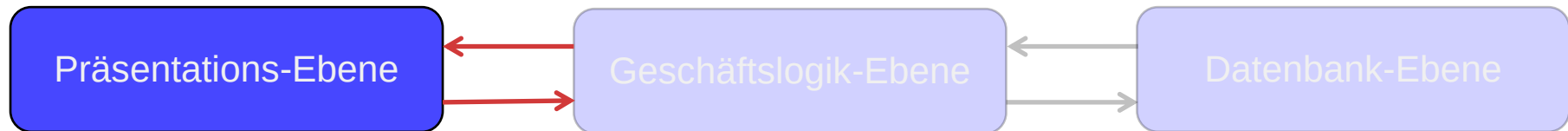
Fortsetzung -
Klasse für
SQL Statement

Präsentations-Ebene



- JSP zur Ausgabe
- Entweder mit eigenen Tag oder
- JSTL

Präsentations-Ebene



```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <title>Eingabemaske Testat 3</title>
</head>
<body>
  //entweder eigener Tag
  <ausgabe liste="liste" />

  //oder JSTL
  <c:foreach var="lied" items=${liedliste}>
    .....
  </c:foreach>
</body>
</html>
```