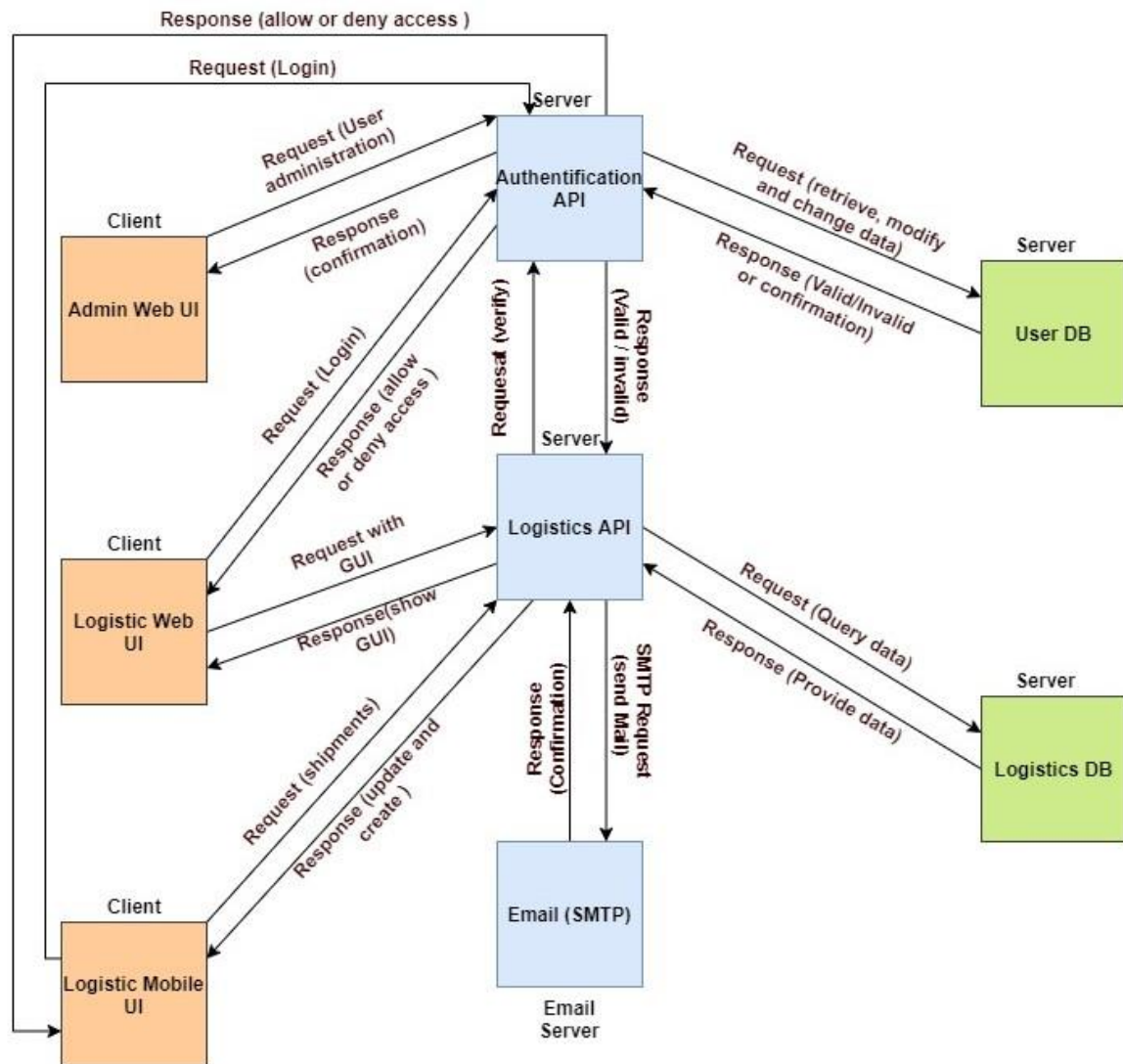


1. Aufgabe 1 – Architektur

Das Diagramm, das die Beschreibung der Aufgabenstellung darstellt.



2. Aufgabe 2 -SQL und JDBC

2.1 Werkzeugunterstützte Datenbankabfrage

- `CREATE TABLE IF NOT EXISTS hints (hint_id INTEGER , hint_text VARCHAR(213));`
- `INSERT INTO hints VALUES (42 , 'Der Typ dieses Attributs ist eine Zeichenfolge. Deren maximal erlaubte Länge ergibt sich aus der Summe von 171 und der ID dieses Hinweises.');`
- `SELECT * FROM hints WHERE hint_text LIKE '%Freitag%';`
next_hint_id = 13
- `SELECT hint_text FROM hints WHERE hint_id = 13;`
Der entsprechende hint_text ist : Ha, Du bist a Kaepsele!

2.2 Programmatische Bankabfrage

a)

```
@DatabaseTable(tableName = "elements")
public class Element {

    @DatabaseField(id = true)
    private int element_id;

    @DatabaseField(canBeNull = false)
    private String element_letter;

    public int getElementID() {
        return this.element_id;
    }

    public String getElementLetter() {
        return this.element_letter;
    }

    public static void main(String[] args) {
        try {
            int[] arrayIndexes = {64, 23, 2, 66, 22, 34, 20, 25, 35, 15, 14, 4, 61, 49, 50, 20,
                                  48, 16, 11, 4};
            String word = "";
            ConnectionSource connectionSource = new JdbcConnectionSource( "jdbc:mariadb://bilbao.informatik" +
                                   ".uni-stuttgart.de/pe2-db-a1", username: "pe2-nutzer", password: "V6GACEKi");
            Dao<Element, Integer> elementDao = DaoManager.createDao(connectionSource, Element.class);
            for (int i = 0; i < arrayIndexes.length; i++) {
                word += elementDao.queryForId(arrayIndexes[i]).getElementLetter();
            }
            System.out.println(word);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Wort = HeyJudEhelpYEstErdAY

b)

```
public static void main(String[] args) {
    try {
        int sumIDs = 0;
        ConnectionSource connectionSource = new JdbcConnectionSource( "jdbc:mariadb://bilbao.informatik" +
                               ".uni-stuttgart.de/pe2-db-a1", username: "pe2-nutzer", password: "V6GACEKi");
        Dao<Element, Integer> elementDao = DaoManager.createDao(connectionSource, Element.class);
        List<Element> elementsLetterU = elementDao.queryForEq( s: "element_letter", o: "U");
        List<Element> elementsLettern = elementDao.queryForEq( s: "element_letter", o: "n");
        List<Element> elementsLetteri = elementDao.queryForEq( s: "element_letter", o: "i");

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Obwohl ich z.B "U" als Parameter übergeben habe, wurden in der Liste Elemente gespeichert, deren `element_letter` "u" und "U" sind. Das gilt für die drei Abfragen mittels der Methode `queryForEq`.

IDs für "U" : 77 , IDs für "u" : 22, 51

IDs für "N" : 70 , IDs für "n" : 44

IDs für "I" : 17, 39 , IDs für "i" : 65

c)

```
public static void main(String[] args) {  
    try {  
        int sumIDs = 0;  
        ConnectionSource connectionSource = new JdbcConnectionSource( url: "jdbc:mariadb://bilbao.informatik" +  
            ".uni-stuttgart.de/pe2-db-a1", username: "pe2-nutzer", password: "V6GACEKi");  
        Dao<Element, Integer> elementDao = DaoManager.createDao(connectionSource, Element.class);  
        List<Element> listOfAllElements = elementDao.queryForAll();  
        for(final Element element : listOfAllElements){  
            sumIDs += element.getElementID();  
        }  
        System.out.println(sumIDs);  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

Summe = 3366 ; Durchschnittswert : 42

3. HTTP und REST

a)

```
1  {  
2      "categories": [  
3          "science"  
4      ],  
5      "created_at": "2020-01-05 13:42:18.823766",  
6      "icon_url": "https://assets.chucknorris.host/img/avatar/chuck-norris.png",  
7      "id": "sbdpxom2tm2-h_wdxncvta",  
8      "updated_at": "2020-01-05 13:42:18.823766",  
9      "url": "https://api.chucknorris.io/jokes/sbdpxom2tm2-h_wdxncvta",  
10     "value": "Chuck Norris eats lightning and shits out thunder."  
11 }
```

b)

```
1  {  
2      "args": {},  
3      "data": {  
4          "key": "pe2ws20",  
5          "purpose": "This is a test."  
6      },  
7      "files": {},  
8      "form": {},  
9      "headers": {  
10         "x-forwarded-proto": "https",  
11         "x-forwarded-port": "443",  
12         "host": "postman-echo.com",  
13         "x-amzn-trace-id": "Root=1-5fbec08e-18bbcd2540b519c44b6a7862",  
14         "content-length": "55",  
15         "content-type": "application/json",  
16         "user-agent": "PostmanRuntime/7.26.8",  
17         "accept": "**/*",  
18         "postman-token": "0915f96a-c191-4642-85dc-1c1d3bb992cf",  
19         "accept-encoding": "gzip, deflate, br"  
20     },  
21     "json": {  
22         "key": "pe2ws20",  
23         "purpose": "This is a test."  
24     },  
25     "url": "https://postman-echo.com/post"  
26 }
```

c)

✚ Ressource zum Abrufen eines Spiels basierend auf dem eindeutigen ID :
[GET /api/games/\\$id](#)

Mit diesem Befehl bzw. dieser HTTP Methode bekommt man das Spiel mit dem vorgegebenen ID aus den Ressourcen games.

✚ Ressource zum Hinzufügen eines Spiels :
[POST /api/games](#)

Mit diesem Befehl bzw. dieser HTTP Methode fügt man ein Spiel den Ressourcen games hinzu. (REQUEST BODY müsste nicht modelliert werden, weswegen die Daten des hinzuzufügenden Spiels fehlen)

✚ Ressource zum Löschen eines existierenden Spiels :
[DELETE /api/games/\\$id](#)

Mit diesem Befehl bzw. dieser HTTP Methode löscht man das Spiel mit dem vorgegebenen ID aus den Ressourcen games.

✚ Ressource zum Abrufen aller alterbeschränkten Spiele :
[GET /api/games?adult=true](#)

Mit diesem Befehl bzw. dieser HTTP Methode bekommt man alle alterbeschränkte Spiele aus den Ressourcen games.

✚ Ressource zum Überschreiben eines existierenden Spiels :
[PUT /api/games/\\$id](#)

Mit diesem Befehl bzw. dieser HTTP Methode überschreibt man ein Spiel aus den Ressourcen games. (REQUEST BODY müsste nicht modelliert werden, weswegen die Daten des hinzuzufügenden Spiels fehlen)