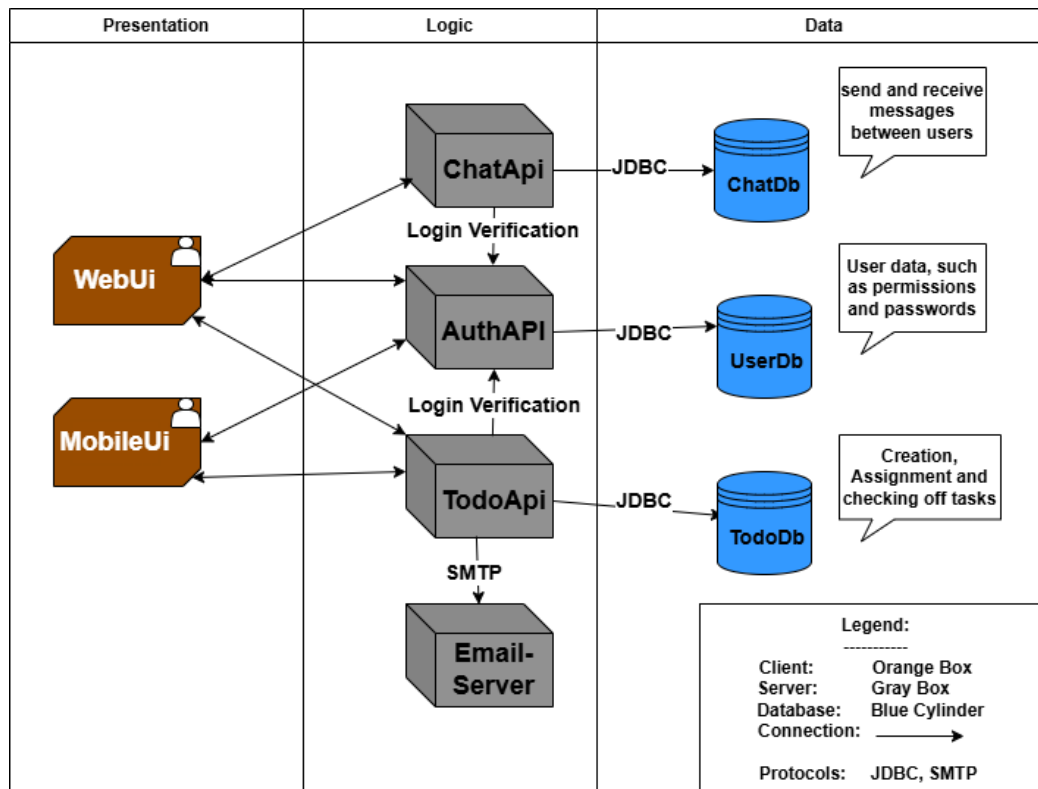


## Aufgabe 1

- (a) Folgendes Diagramm veranschaulicht die im Text beschriebenen Top-Level-Komponenten:



## Aufgabe 2

### 2.1 Werkzeugunterstützte SQL-Befehle

- (a)
- ```

1 CREATE TABLE IF NOT EXISTS todos(
2     id INTEGER PRIMARY KEY,
3     title VARCHAR(256),
4     description TEXT
5 );

```
- (b)
- ```

1 INSERT INTO todos(id, title, description)
2 VALUES(1, 'Dekorieren',
3 'Es ist nun endlich so weit! Mit dem 01. November wird es Zeit, zügig die
4 Weihnachtsdekorationen auspacken.'
5 );

```
- (c)
- ```

1 SELECT todos.description FROM todos WHERE description LIKE '%Weihnacht%';

```

|   | description ▾                                                                                              |
|---|------------------------------------------------------------------------------------------------------------|
| 1 | Es ist nun endlich so weit! Mit dem 01. November wird es Zeit, zügig die Weihnachtsdekorationen auspacken. |
| 2 | Bald sollte ich Weihnachtsplaetzchen backen.                                                               |

## 2.2 Programmatische Datenbankabfrage

Für Aufgabenteil a),b) sowie c) befindet sich der Code im Projekt innerhalb des Packages task2 in der Klasse DBManager.java:

<https://github.tik.uni-stuttgart.de/iste-pe2-2024/repo008/blob/master/code/src/exercise1/task2/DBManager.java>

- (a) Das gesuchte Wort lautet: EntwicklUnGPrOgrAMMII
- (b) IDs für 'V': 52, 78  
IDs für 'b': 9, 32, 58  
IDs für 't': 50,76
- (c) Aus dem Java-Programm wird ersichtlich, dass die Summe der IDs 4167 ergibt. Da es 82 Einträge gibt, beträgt der Durchschnittswert  $\frac{4167}{82} \approx 50.817074$ .

## Aufgabe 3

Für die Aufgabenteile wurde mit IntelliJ Ultimate gearbeitet. Die Dateien sind alternativ auch im Repository zu finden:

<https://github.tik.uni-stuttgart.de/iste-pe2-2024/repo008/tree/master/code/src/exercise1/task3>

- (a) auch zu finden unter RandomJoke.http  
Request:

---

```
1 GET https://api.chucknorris.io/jokes/random?category=history
```

---

---

Response:

---

```
1 {
2   "categories": [
3     "history"
4   ],
5   "created_at": "2020-01-05 13:42:19.324003",
6   "icon_url": "https://api.chucknorris.io/img/avatar/chuck-norris.png",
7   "id": "3qr6hlt6rouy2t2jtlzzw",
8   "updated_at": "2020-01-05 13:42:19.324003",
9   "url": "https://api.chucknorris.io/jokes/3qr6hlt6rouy2t2jtlzzw",
10  "value": "The Great Wall of China was originally created to keep Chuck Norris out. It
          failed miserably."
11 }
```

---

---

(b) auch zu finden unter PostRequest.http

Request:

```

1 POST https://postman-echo.com/post
2 Content-Type: application/json
3
4 {
5   "key": "pe2ws23",
6   "purpose": "This is a test."
7 }

```

Response:

```

1 {
2   "args": {},
3   "data": {
4     "key": "pe2ws23",
5     "purpose": "This is a test."
6   },
7   "files": {},
8   "form": {},
9   "headers": {
10    "host": "postman-echo.com",
11    "x-request-start": "t=1730283371.570",
12    "connection": "close",
13    "content-length": "54",
14    "x-forwarded-proto": "https",
15    "x-forwarded-port": "443",
16    "x-amzn-trace-id": "Root=1-6722076b-69e6d1fa41a34d8c68f62fc9",
17    "content-type": "application/json",
18    "user-agent": "IntelliJ HTTP Client/IntelliJ IDEA 2024.2.3",
19    "accept-encoding": "br, deflate, gzip, x-gzip",
20    "accept": "/*/*",
21    "cookie": "sails.sid=s%3AqE-WqVDNdg_Z-npiEKhmNea2bRTLot__
    FlsKk8i0lehgwKwWsilsS5oTCo5kkRSyZ%2BVPEuDz6uY"
22  },
23  "json": {
24    "key": "pe2ws23",
25    "purpose": "This is a test."
26  },
27  "url": "https://postman-echo.com/post"
28 }

```

(c) Folgende Ressourcen werden definiert:

| CRUD-Operation | HTTP-Method | Path                                                                                                   | Description                                                                             |
|----------------|-------------|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| Create         | POST        | /dvds                                                                                                  | allows to add a new DVD                                                                 |
| Read           | GET         | by ID: /dvds/{id}<br>with filter:<br>/dvds?category={category}<br>&title={title}<br>&isAdult={boolean} | returns the DVD with given ID<br>returns all DVDs that fulfill the<br>given constraints |
| Update         | PUT         | /dvds/{id}                                                                                             | updates the DVD with ID                                                                 |
| Delete         | DELETE      | /dvds/{id}                                                                                             | deletes the DVD with ID                                                                 |