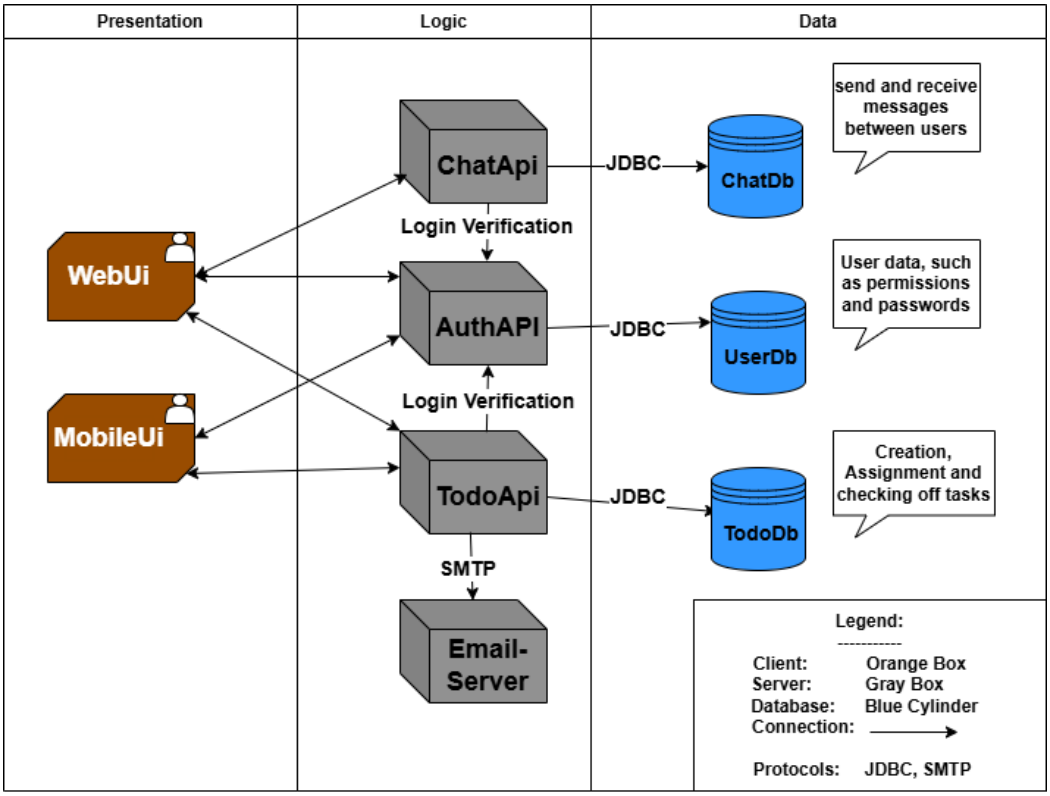
# Aufgabe 1

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




# Aufgabe 2

## 2.1 Werkzeugunterstützte SQL-Befehle

(a)

```sql
CREATE TABLE IF NOT EXISTS todos(
        id INTEGER PRIMARY KEY,
        title VARCHAR(256),
        description TEXT
);
```

(b)

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

(c)

```sql
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 auszupacken. |
| 2 | Bald sollte ich Weihnachtsplaetzchen backen. |

### 2.2 Programmatische Datenbankabfrage

Für Aufgabenteil a),b) sowie c) befindet sich der Code im Projekt innerhalb des Repositorys im code-Ordner unter dem Package 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:

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

Response:

```
{
  "categories": [
    "history"
  ],
  "created_at": "2020-01-05 13:42:19.324003",
  "icon_url": "https://api.chucknorris.io/img/avatar/chuck-norris.png",
  "id": "3qr6hlt6rouy2t2jtlnzzw",
  "updated_at": "2020-01-05 13:42:19.324003",
  "url": "https://api.chucknorris.io/jokes/3qr6hlt6rouy2t2jtlnzzw",
  "value": "The Great Wall of China was originally created to keep Chuck Norris out. It failed miserably."
}
```

(b) auch zu finden unter PostRequest.http
Request:

```
POST https://postman-echo.com/post
Content-Type: application/json

{
  "key": "pe2ws23",
  "purpose": "This is a test."
}
```

Response:


```
{
  "args": {},
  "data": {
    "key": "pe2ws23",
    "purpose": "This is a test."
  },
  "files": {},
  "form": {},
  "headers": {
    "host": "postman-echo.com",
    "x-request-start": "t=1730283371.570",
    "connection": "close",
    "content-length": "54",
    "x-forwarded-proto": "https",
    "x-forwarded-port": "443",
    "x-amzn-trace-id": "Root=1-6722076b-69e6d1fa41a34d8c68f62fc9",
    "content-type": "application/json",
    "user-agent": "IntelliJ HTTP Client/IntelliJ IDEA 2024.2.3",
    "accept-encoding": "br, deflate, gzip, x-gzip",
    "accept": "*/*",
    "cookie": "sails.sid=s%3ArqE-WqVDNdg_Z-npiEKhmNea2bRTLot_.
        FlsKk8i0lehgwKwWsilsS5oTCo5kkRSyZ%2BVPEuDz6uY"
  },
  "json": {
    "key": "pe2ws23",
    "purpose": "This is a test."
  },
  "url": "https://postman-echo.com/post"
}
```


(c) Folgende Ressourcen werden definiert:

| **CRUD-Operation** | HTTP-Method | Path | Description |
|---|---|---|---|
| **C**reate | POST | /dvds | allows to add a new DVD |
| **R**ead | 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 given constraints |
| **U**pdate | PUT | /dvds/{id} | updates the DVD with ID |
| **D**elete | DELETE | /dvds/{id} | deletes the DVD with ID |