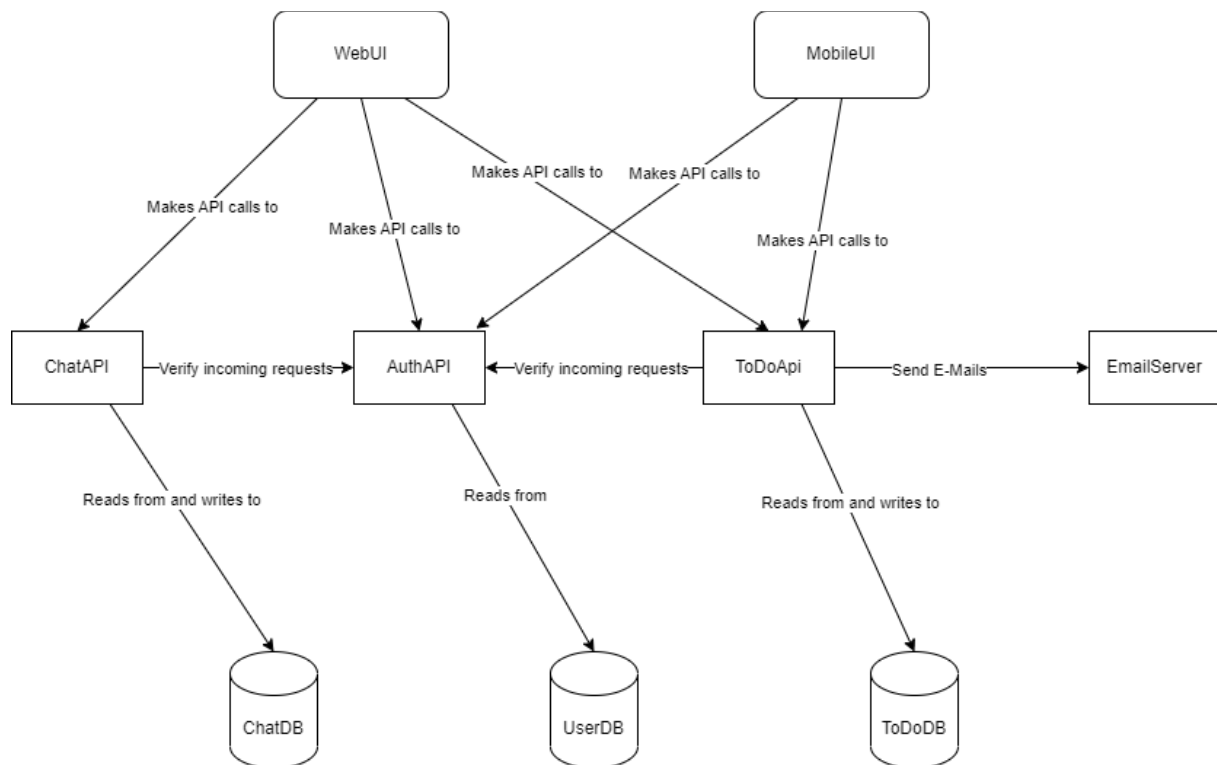


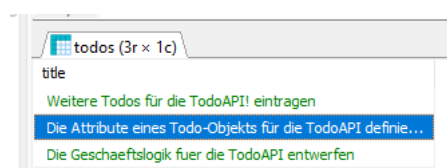
## Programmentwicklung ÜB01

1.



2.1

- CREATE TABLE IF NOT EXISTS todos(  
id INTEGER PRIMARY KEY,  
title VARCHAR(100) NOT NULL DEFAULT 'New todo',  
description VARCHAR(500)  
);
- INSERT INTO todos  
(ID)  
VALUES  
(1);
- SELECT title FROM todos  
WHERE title LIKE '% TodoAPI%';



## 2.2

a)

```
public void runDemonstration() {
    // creates connection to a MariaDB installed on localhost

    try {

        connectionSource = new JdbcConnectionSource("jdbc:mariadb://bilbao.informatik.uni-stuttgart.de/pe2-db-a1", username: "pe2-nutzer", password: "esJltFm6ksCT4mCy0S");
        // TODO: instantiate the Database Access Object (DAO) for handling DB operations using the DaoManager
        this.letterDao = DaoManager.createDao(this.connectionSource, Letters.class);

        int[] arrayIndexes = { 72, 48, 71, 37, 48, 11, 69, 5, 20, 44, 50, 13, 17, 33, 41, 68, 77, 44, 29, 65, 65 };

        String word="";
        String letter="";

        for (int arrayIndex: arrayIndexes) {

            letter= letterDao.queryForId(arrayIndex).getLetter();

            word += letter;
        }
        System.out.println(word);

        // closes connection to the database
        this.closeConnectionToDB();

    } catch (SQLException exception) {
        this.logSQLException(exception);
    }

}
```

b)

```
public void runDemonstration() {
    // creates connection to a MariaDB installed on localhost

    try {

        connectionSource = new JdbcConnectionSource("jdbc:mariadb://bilbao.informatik.uni-stuttgart.de/pe2-db-a1", username: "pe2-nutzer", password: "esJltFm6ksCT4mCy0S");
        // TODO: instantiate the Database Access Object (DAO) for handling DB operations using the DaoManager
        this.letterDao = DaoManager.createDao(this.connectionSource, Letters.class);

        List<Letters> indexW;
        indexW=letterDao.queryForEq("@ "letter", @ "w");
        System.out.println(indexW);

        List<Letters> indexA;
        indexA=letterDao.queryForEq("@ "letter", @ "a");
        System.out.println(indexA);

        List<Letters> indexS;
        indexS=letterDao.queryForEq("@ "letter", @ "s");
        System.out.println(indexS);

    }
```

c)

```
public void runDemonstration() {
    // creates connection to a MariaDB installed on localhost

    try {

        connectionSource = new JdbcConnectionSource( url "jdbc:mariadb://bilbao.informatik.uni-stuttgart.de/pe2-db-a1", username: "pe2-nutzer", password: "esJltFm6ksCT4mCy0S");
        // TODO: instantiate the Database Access Object (DAO) for handling DB operations using the DaoManager
        this.letterDao = DaoManager.createDao(this.connectionSource, Letters.class);

        List <Letters> IndexList = letterDao.queryForAll();
        IndexList.toArray();
        int sum = 0;
        int i=0;
        for (Letters letters : IndexList) {
            sum= sum+ letters.getId();
            i++;
        }
        int median = sum / i;
        System.out.println(sum);
        System.out.println(median);
        {

        }

        // closes connection to the database
        this.closeConnectionToDB();

    } catch (SQLException exception) {
        this.logSQLException(exception);
    }

}
```

3.

a)

```
{
  "categories": [
    "science"
  ],
  "created_at": "2020-01-05 13:42:19.576875",
  "icon_url": "https://assets.chucknorris.host/img/avatar/chuck-norris.png",
  "id": "qszz44pvr3iv7kzgwoedza",
  "updated_at": "2020-01-05 13:42:19.576875",
  "url": "https://api.chucknorris.io/jokes/qszz44pvr3iv7kzgwoedza",
  "value": "Chuck Norris can do a roundhouse kick faster than the speed of light. This means that if you turn on a light switch, you will be dead before the lightbulb turns on."
}
```

b)

```

{
  "args": {},
  "data": "-----204350700025308920854499\r\nContent-Disposition: form-data; name=\"pe2ws22\"\r\n\r\nThis is a test.\r\n\r\n-----204350700025308920854499--\r\n",
  "files": {},
  "form": {},
  "headers": {
    "x-forwarded-proto": "https",
    "x-forwarded-port": "443",
    "host": "postman-echo.com",
    "x-amzn-trace-id": "Root=1-6377e9e5-81eac3d755d26b4325a560cc",
    "content-length": "177",
    "content-type": "application/json",
    "user-agent": "PostmanRuntime/7.29.2",
    "accept": "*/*",
    "postman-token": "2cefdb5e-f3b5-4bc3-aea0-a701aae6bc2d",
    "accept-encoding": "gzip, deflate, br",
    "cookie": "sails.sid=s%3AfrGk9W7DG_ZAWb-TJDu0TDAPoivxt-r3.t4SEWgqfo9tKS1xySR7swoOvZ8UMsNXQ7FCSu82rljM"
  },
  "json": null,
  "url": "https://postman-echo.com/post"
}
```

c)

Get all Games: GET /games/\$category&\$title&\$is18=true

Create a single game: POST /games

Update a single game: PUT /games/\$id

Get a single game GET /games/\$id

Delete a single game DELETE /games/\$id