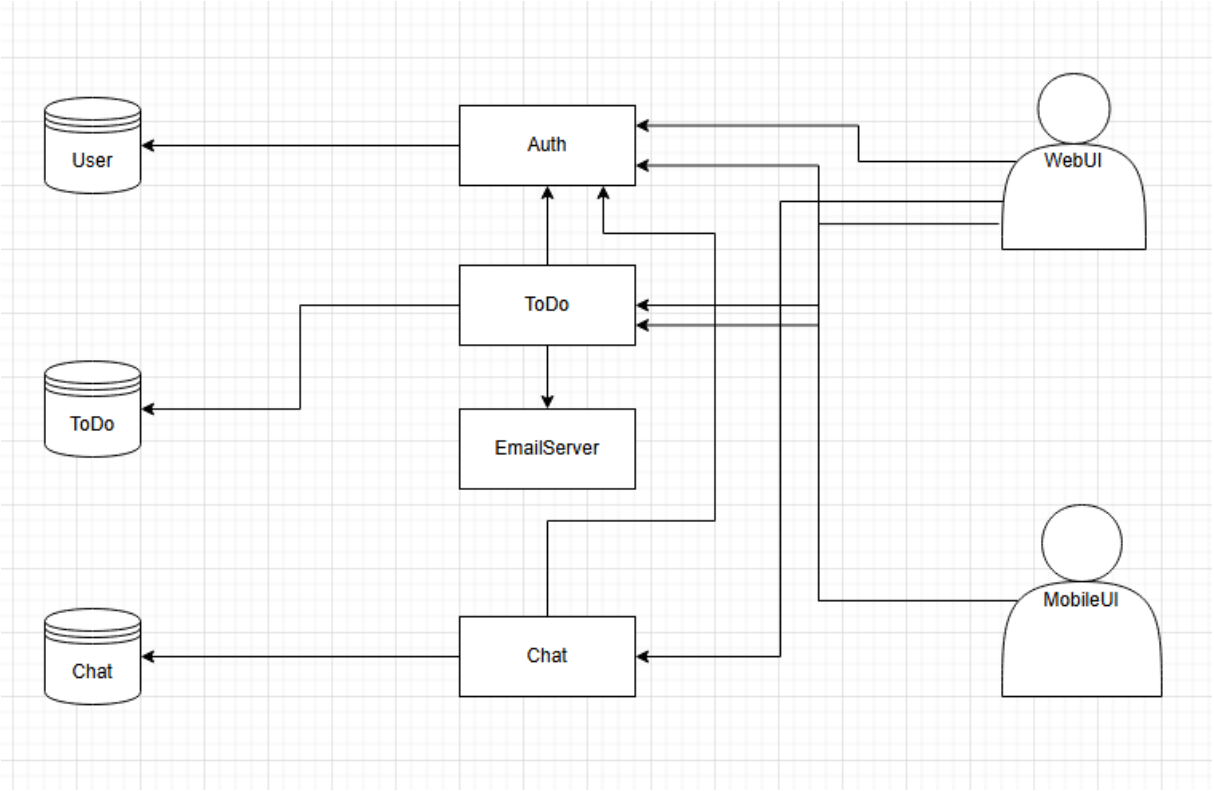
1.



2.

a)

```
CREATE TABLE IF NOT EXISTS todos (
  Id INTEGER PRIMARY KEY,
  title VARCHAR(255),
  description TEXT
);
```

b)

```
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)

```
SELECT description
FROM todos
WHERE description LIKE '%Weihnacht%';
```

Ausgabe:

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

a)

Main:

```
package org.example;

import java.sql.SQLException;

//TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or
// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
public class Main {
   public static void main(String[] args) {

      LetterQuery bla = new LetterQuery();

      bla.connect();
      try {
         bla.buildWord();
      } catch (SQLException e) {
         throw new RuntimeException(e);
      }
      bla.closeConnection();
```

```
    }
}
```

Letter Klasse:

```
package org.example;

import com.j256.ormlite.field.DatabaseField;
import com.j256.ormlite.table.DatabaseTable;

@DatabaseTable(tableName = "letters")
public class Letter {

    @DatabaseField(id = true)
    private Integer id;

    @DatabaseField(columnName = "letter")
    private String letter;

    //Get function for ID
    public Integer getId() {
        return this.id;

    }

    //Set function for ID
    public void setId(Integer id) {
        this.id = id;
    }

    //Get function for Letter
    public String getLetter() {
        return this.letter;
    }

    //Set function for Letter
    public void setLetter(String letter) {
        this.letter = letter;
    }

    public Letter() {}

}
```

Klasse LetterQuery:

```
package org.example;
```

```
import com.j256.ormlite.dao.Dao;
import com.j256.ormlite.dao.DaoManager;
import com.j256.ormlite.jdbc.JdbcConnectionSource;
import com.j256.ormlite.support.ConnectionSource;

import java.sql.SQLException;
import java.util.List;

public class LetterQuery {

    ConnectionSource connectionSource;
    Dao<Letter, Integer> letterDao;

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

    public void connect (){
      try {
        connectionSource = new JdbcConnectionSource("jdbc:mariadb://bilbao.informatik.uni-
stuttgart.de/pe2-db-a1", "pe2-nutzer", "esJLtFm6ksCT4mCyOS");
        letterDao = DaoManager.createDao(connectionSource, Letter.class);

      } catch (Exception e) {
        throw new RuntimeException(e);

      }
    }

    //Build word and print it
    public void buildWord() throws SQLException {
      StringBuilder word = new StringBuilder();
      for (int id : arrayIndexes) {
        Letter letter = letterDao.queryForId(id);
        if (letter != null) {
          word.append(letter.getLetter());
        }
      }
      System.out.println("Lösungswort: " + word.toString());
    }

    //close connection
    public void closeConnection() {
      try {
        connectionSource.close();
      } catch (Exception e) {
        throw new RuntimeException(e);
```

```
        }

      }
}
```

Lösungswort: EntwickLUnGPrOgrAMMII

b)

IDs für 'v':

52

78

IDs für 'b':

9

32

58

IDs für 't':

50

76

Hinzugefügte Operation in der Klasse LetterQuery:

```
//Find letter and print IDs
public void findLetterIds(String letter) throws SQLException {
   System.out.println("IDs für '" + letter + " :");

   List<Letter> ids = letterDao.queryForEq("letter", letter);

   for (Letter l : ids) {
      System.out.println(l.getId());
   }
}
```

Main Klasse:

```
public class Main {
   public static void main(String[] args) {
```

```
        LetterQuery bla = new LetterQuery();

        bla.connect();
        try {
            bla.findLetterIds("v");
            bla.findLetterIds("b");
            bla.findLetterIds("t");
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
        bla.closeConnection();

    }
}
```

3)

a)

```
<!doctype html><html lang="en"><head><title>HTTP Status 400 – Bad Request</title><style type="text/css">body {font-family:Tahoma,Arial,sans-serif;

} h1, h2, h3, b {color:white;background-color:#525D76;

} h1 {font-size: 22px;

} h2 {font-size: 16px;

} h3 {font-size: 14px;

} p {font-size: 12px;

} a {color:black;

} .line {height: 1px;background-color:#525D76;border:none;

}</style></head><body><h1>HTTP Status 400 – Bad Request</h1></body></html>
```

b)