

Dokumentation

1.0 Einleitung und Projektidee

Die Idee war ein Programm zu schreiben, das es dem User ermöglicht anzugeben welche Zutaten er zum Kochen grade im Haus hat beziehungsweise welche Zutaten er zum Kochen verwenden möchte.

Es sollte möglich sein sich eine Beilage (Reis, Kartoffeln oder Nudeln) und Gemüsesorten auszusuchen. Nach dem Auswählen der entsprechenden Zutaten sollte das Programm Rezepte, die zu den angegebenen Zutaten passen, vorschlagen. Zunächst sollte lediglich der Name des Rezepts angezeigt werden. Anschließend sollte es möglich sein ein Rezept auszuwählen und sich die gesamte Zutatenliste anzeigen zu lassen.

2.0 Definitionen, Hintergrundinformationen und Begründungen

Die Idee eine Datenbank zu programmieren entstand durch ein reges Interesse an dem Modul Datenbanken. Da es erfahrungsgemäß oft schwierig ist Rezepte zu finden, die zu bereits vorhandenen Zutaten passen, stand schnell fest, dass sich die Datenbank damit befassen sollte.

Um dem User die Auswahl der Zutaten zu ermöglichen, sollte es eine GUI geben, in der die Zutaten angeklickt werden können. Mithilfe der GUI sollte dem User außerdem das Anzeigen der Rezepte und der Zutatenlisten ermöglicht werden. Die möglichen Rezepte sollten schon in einer Datenbank hinterlegt und dort mit den Zutaten verknüpft sein.

3.0 Projektplan und Projektaufbau

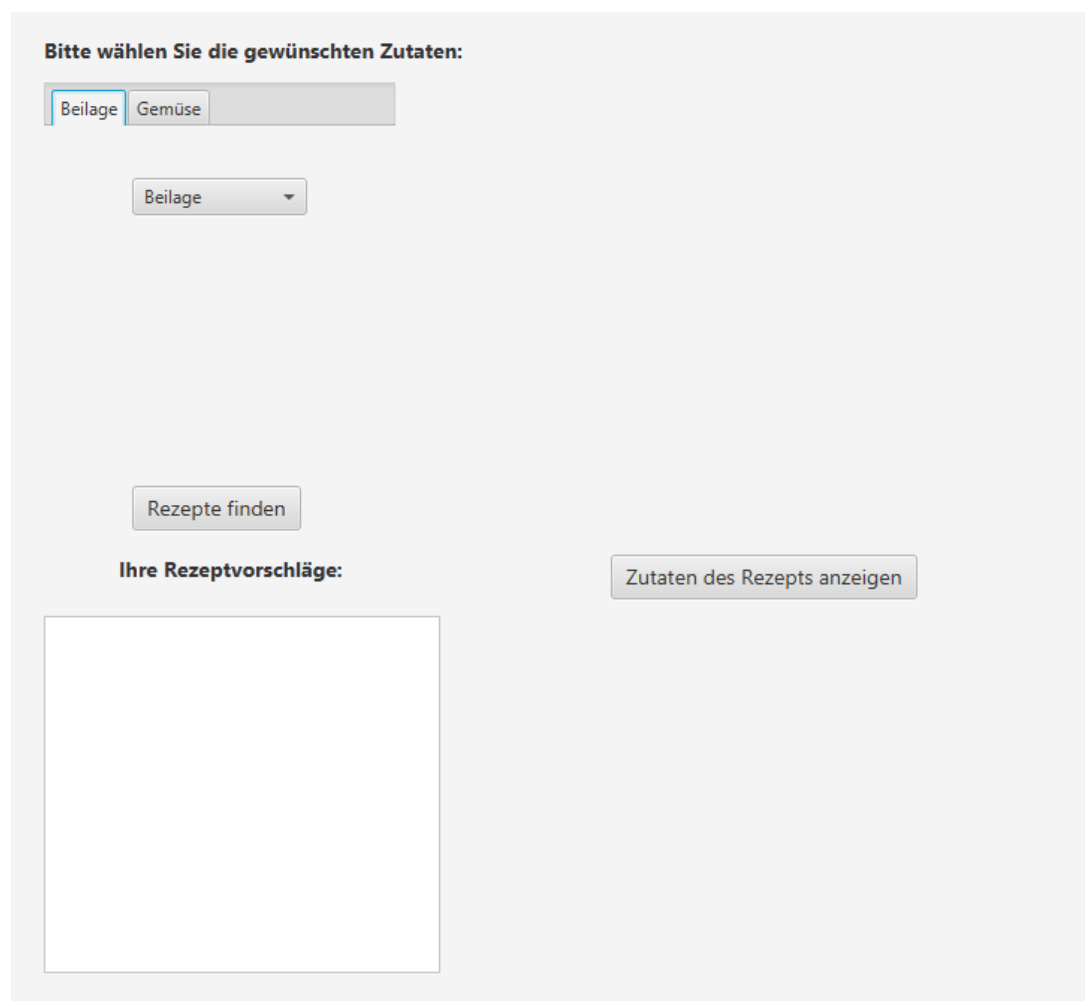
3.1 Beschreibung der Werkzeuge und Hilfsmittel

Aufgrund bereits gesammelter Erfahrungen im Bereich Datenbankprogrammierung wurden als Programmiersprache Java und als Entwicklungsumgebung NetBeans IDE gewählt. Außerdem wurde

die GUI mit dem Tool JavaFx Scene Builder erstellt. Die Datenbank wurde mit dem Tool SQLite erstellt und gefüllt. Für die Verknüpfung des Java-Codes mit der SQLite-Datenbank wurde der SQLite JDBC Driver verwendet. Es handelte sich hierbei um die Datei „sqlite-jdbc-3.8.11.2.jar“.

3.2 Beschreibung der einzelnen Elemente und Funktionen

3.2.1 GUI



Bitte wählen Sie die gewünschten Zutaten:

Beilage

Gemüse

☐ Tomaten

☐ Kidneybohnen

☐ Paprika

☐ Brokkoli

☐ Spinat

Rezepte finden

Ihre Rezeptvorschläge:

Zutaten des Rezepts anzeigen

Bei dem Text „Bitte wählen Sie die gewünschten Zutaten“ handelt es sich um ein Label, das den User darauf hinweist was zu tun ist.

Der Kasten, in dem sich die Reiter Beilage und Gemüse befinden, ist ein TabPane, das es ermöglicht eben solche Reiter anzuzeigen.

Bei den Reitern Beilage und Gemüse handelt es sich um Tabs. Jeder Tab ist mit einem sogenannten AnchorPane verbunden. In diesem befinden sich jeweils die wählbaren Zutaten.

Das Kästchen, mit dem die Beilage ausgewählt wird, ist eine ComboBox. Durch Anklicken der ComboBox werden, in diesem Fall, die auswählbaren Beilagen angezeigt. Bei einer ComboBox kann lediglich eine Option gewählt werden.

Im Gegensatz dazu handelt es sich bei den Gemüsesorten um CheckBoxen. Jede CheckBox kann unabhängig von den anderen ausgewählt werden.

„Rezepte finden“ und „Zutaten des Rezepts anzeigen“ sind Buttons, also Knöpfe, die gedrückt beziehungsweise angeklickt werden können.

Unter dem Button „Zutaten des Rezepts anzeigen“ befinden sich 12 Labels, die erstmal nicht sichtbar sind, da sie keinen Text enthalten. Im weiteren Verlauf können dort die gesamten Zutaten eines Rezepts angezeigt werden.

Auch der Text „Ihre Rezeptvorschläge:“ ist wieder ein Label.

Der Kasten unter diesem Text ist eine ListView. Diese ist anfangs leer, wird aber später mit den Namen der Rezepte gefüllt.

3.2.2 Funktionen

In den Zeilen 7 – 23 werden die Pakete und Klassen aus einem Paket importiert. „Durch den Import eines Pakets oder einer Klasse werden alle Vereinbarungen sichtbar gemacht, die von ihrer Zugriffsklasse her in anderen Paketen sichtbar sein dürfen“ (dpunkt Verlag für digitale Technologie Heidelberg, 2002).

In Zeile 29 wird die Klasse FXMLDocumentController erstellt:

```
public class FXMLDocumentController implements Initializable {  
  
    Connection connection = null;  
  
    [...]  
  
}
```

Diese beinhaltet alle weiteren, für das Projekt benötigten, Initialisierungen und Funktionen, welche fast alle mit der Annotation @FXML versehen sind. „Dies bewirkt, dass das private Objekt wie üblich zwar nicht von anderen Klassen, jedoch von der Markupsprache aus angesprochen werden kann“ (Czeschla - javabeginners.de, 2016).

In den Zeilen 33 – 70 werden alle benötigten ListViews, ComboBoxen, CheckBoxen und Labels initialisiert.

Die Funktion `Rezepte_anzeigen` wird in den Zeilen 73 – 107 erstellt:

@FXML

```
private void Rezepte_anzeigen(ActionEvent event) {  
    try {  
        Statement statement = connection.createStatement();  
        statement.setQueryTimeout(30);  
        String queryString = "SELECT Name_Rezept FROM enthaelt WHERE  
1=1 ";  
        if (combobox_Beilage.getSelectionModel().getSelectedIndex()>=0) {  
            queryString = queryString+" AND Name_Zutat_1 =  
            '"+combobox_Beilage.getSelectionModel().getSelectedItem()+  
            "';    }  
        if (checkbox_Kidneybohnen.isSelected()) {  
            queryString = queryString+" AND Name_Zutat_2 =  
            '"+checkbox_Kidneybohnen.getText()+"';    }  
        if (checkbox_Paprika.isSelected()) {  
            queryString = queryString+" AND Name_Zutat_3 =  
            '"+checkbox_Paprika.getText()+"';    }  
        if (checkbox_Spinat.isSelected()) {  
            queryString = queryString+" AND Name_Zutat_4 =  
            '"+checkbox_Spinat.getText()+"';    }  
        if (checkbox_Brokkoli.isSelected()) {  
            queryString = queryString+" AND Name_Zutat_5 =  
            '"+checkbox_Brokkoli.getText()+"';    }  
        if (checkbox_Tomaten.isSelected()) {  
            queryString = queryString+" AND Name_Zutat_6 =  
            '"+checkbox_Tomaten.getText()+"';    }  
        ResultSet rs = statement.executeQuery(queryString);
```

```

        Rezepte_Liste.getItems().clear();

        while(rs.next()) {

            Rezepte_Liste.getItems().add(rs.getString("Name_Rezept")); }

    } catch(SQLException e) {

        System.err.println(e.getMessage());    }

    aktualisieren();

}

```

Diese Funktion ist mit dem Button „Rezepte finden“ verknüpft und beinhaltet die Abfrage welche Beilage und welches Gemüse gewählt wurde. Außerdem veranlasst die Funktion beim Anklicken des Buttons, dass die Namen der Rezepte, in denen die gewählten Zutaten vorkommen, in der ListView angezeigt werden.

In den Zeilen 109 – 133 wird die Funktion Zutaten_anzeigen erstellt:

@FXML

```

public void Zutaten_anzeigen(ActionEvent event) {

    try {

        Statement statement = connection.createStatement();

        statement.setQueryTimeout(30);

        if (Rezepte_Liste.getSelectionModel().getSelectedItem() != null) {

            ResultSet rs = statement.executeQuery("select * FROM
            Rezepte          WHERE          Name_Rezept          =
            '"+Rezepte_Liste.getSelectionModel().getSelectedItem()+"'");

            Zutat1_Label.setText(rs.getString("Zutat_1"));

            Zutat2_Label.setText(rs.getString("Zutat_2"));

            Zutat3_Label.setText(rs.getString("Zutat_3"));

            Zutat4_Label.setText(rs.getString("Zutat_4"));

            Zutat5_Label.setText(rs.getString("Zutat_5"));

            Zutat6_Label.setText(rs.getString("Zutat_6"));

            Zutat7_Label.setText(rs.getString("Zutat_7"));

```

```

        Zutat8_Label.setText(rs.getString("Zutat_8"));

        Zutat9_Label.setText(rs.getString("Zutat_9"));

        Zutat10_Label.setText(rs.getString("Zutat_10"));

        Zutat11_Label.setText(rs.getString("Zutat_11"));

        Zutat12_Label.setText(rs.getString("Zutat_12"));
    }

    } catch(SQLException e) {

        System.err.println(e.getMessage());    }

}

```

Diese Funktion ist mit dem Button „Zutaten des Rezepts anzeigen“ verknüpft und veranlasst beim Anklicken des Buttons, dass alle Zutaten eines Rezepts in den 12 nicht sichtbaren Labels angezeigt werden, vorausgesetzt, dass in der ListView ein Rezept ausgewählt wurde.

Eine der Funktionen, die nicht mit der Annotation @FXML versehen ist, wird in den Zeilen 135 -148 erstellt. Es handelt sich um die Funktion initialize, welche mit der Annotation @Override versehen ist:

```

@Override

public void initialize(URL url, ResourceBundle rb) {

    // TODO

    try {

        // create a database connection

        connection = DriverManager.getConnection("jdbc:sqlite:Rezepte.db");

    }

    catch(SQLException e) {

        // if the error message is "out of memory",

        // it probably means no database file is found

        System.err.println(e.getMessage());    }

    aktualisieren();
}

```

```
}
```

Diese Funktion stellt mit dem SQLite JDBC Driver die Verbindung zur SQLite-Datenbank her.

In den Zeilen 150 – 165 wird außerdem die Funktion aktualisieren erstellt:

```
public void aktualisieren() {  
  
    // ComboBox Beilage füllen  
  
    ObservableList<String>Beilage_combobox=  
        FXCollections.observableArrayList();  
  
    try {  
  
        Statement statement = connection.createStatement();  
  
        statement.setQueryTimeout(30);  
  
        ResultSet rs = statement.executeQuery("select * from Zutaten WHERE  
        ist_Beilage = 1");  
  
        while(rs.next()) {  
  
            Beilage_combobox.add(rs.getString("Zutat_Name"));  
  
        }  
  
        combobox_Beilage.setItems(Beilage_combobox);  
  
    }  
  
    catch(SQLException e) {  
  
        System.err.println(e.getMessage());  
  
    }  
  
}
```

Durch diese Funktion wird die ComboBox mit den Beilagen gefüllt.

4.0 Anleitung zur Ausführung und Benutzung

Zunächst einmal muss das Projekt in NetBeans IDE geöffnet werden. Hierfür öffnet man NetBeans IDE und klickt zuerst auf „File“ und anschließend auf „Open Project“. Dann wird das entsprechende Projekt herausgesucht.

Ist das Projekt geöffnet, muss es ausgeführt werden. Hierfür klickt man auf den grünen Pfeil oder auf „Run“ und anschließend auf „Run Project“.

Dadurch öffnet sich die GUI.

Anschließend wird eine oder keine Beilage und beliebig viele der vorhandenen Gemüsesorten ausgewählt. Durch Anklicken des Buttons „Rezepte finden“ werden alle Rezepte, die zu den Zutaten passen, in der ListView angezeigt.

Durch Anklicken eines Rezeptes und des Buttons „Zutaten des Rezepts anzeigen“ werden alle benötigten Zutaten für das ausgewählte Rezept angezeigt.

5.0 Reflexion

Insgesamt wurde das Ziel erreicht.

Schief gelaufen ist an sich nichts, aber es war zeitlich leider nicht mehr möglich die Datenbank noch zu erweitern.

Die Datenbank mit SQLite zu erstellen und zu füllen war erstaunlich leicht.

Allerdings war es trotz bereits gesammelter Erfahrung im Bereich Datenbankprogrammierung schwer den Code zu schreiben.

Im Enddefekt waren einige der vorher angedachten Funktionen unnötig und wurden durch eine einzige komplexere Funktion ersetzt.

5.1 Fazit

Auch, wenn ich mein Ziel erreicht habe, bin ich nicht 100prozentig zufrieden, da ich gehofft hatte sowohl die GUI als auch die Datenbank erweitern zu können. Dies habe ich allerdings zeitlich nicht mehr geschafft. Außerdem kam ich anfangs mit der Programmierung des Codes nur langsam voran, da ich Schwierigkeiten hatte mich wieder in das Thema einzufinden.

Trotzdem hat mir das Projekt großen Spaß gemacht und ich würde mich jederzeit wieder dazu entscheiden eine Datenbank zu programmieren.

6.0 Literaturverzeichnis

Czeschla - javabeginners.de (2016). Javabeginners - FXML. Verfügbar unter view-source:<https://javabeginners.de/Frameworks/JavaFX/FXML.php> [21.09.2016].

dpunkt Verlag für digitale Technologie Heidelberg (2002). Importieren von Paketen und Klassen. Verfügbar unter https://www.dpunkt.de/java/Die_Sprache_Java/Objektorientierte_Programmierung_mit_Java/31.html [21.09.2016].