

DönerWo Dokumentation

Einführung in die Software:

DönerWo soll das Erreichen einer Döneria mit qualitativ hochwertigen Kebabs erleichtern. Aus der Sicht des Benutzers ist das nächste gute Döner nur ein Knopfdruck entfernt. Ausgehend vom Standort des Users wird die nächste Döneria aus unserer Datenbank ausgesucht und angezeigt.

Komponenten:

MongoDB Datenbank:

Eine Datenbank bei MongoDB wurde angelegt, um das persistente Speichern und Abrufen von Objekten zu ermöglichen. Dies zusammen mit der nächsten Komponente bildet den Mechanismus, wie wir unsere Datenquelle ins Projekt integriert haben.

ExcelManager:

Diese Komponente liest Döner Standorte aus einer Excel-Datei aus und generiert dazu Objekte, die anschließend in der MongoDB Datenbank gespeichert werden. Die Daten in der Excel-Datei werden beim derzeitigen Stand manuell ergänzt. Eine Alternative dazu wird näher erläutert.

GUI:

Für die grafische Darstellung unserer Applikation wird Java Swing verwendet. Wir haben uns für eine minimalistische, altbekannte Darstellungsweise entschieden.

OpenStreetMapUtils:

Hier werden Adressen in Koordinaten umgewandelt, damit diese dann in einer anderen Komponente bezüglich der Entfernung verglichen werden können. Dabei erfolgt ein "Data Crawling" von der Google Maps Seite.

Entfernungsrechnung:

Die Entfernung vom derzeitigen Standort zu allen anderen Adressen erfolgt mit dem trigonometrischen Ansatz. Dabei wird hauptsächlich die Entfernung nach Luftlinie berechnet, welche dann zum geordneten Anzeigen der möglichen Döner Optionen verwendet wird.

Probleme:

Maps API:

Ursprünglich wurde eine Lösung erdacht wo die Standorte von einer Google Maps API ermittelt werden. Hierbei hätte man eine Adresse übermittelt und die nächsten Ergebnisse wären dann als rückgabe übergeben werden. Leider ergaben unsere Recherchen, dass eine solche API anzubinden entweder zu aufwendig oder zu teuer gewesen wäre. Beim derzeitigen Stand werden die Daten Manuell übermittelt

JavaFX:

Leider sind wir an der Integration von einer ausgefalleneren grafischen Darstellung gescheitert. Hierbei handelte es sich um den Versuch JavaFX in unserer SpringBoot Anwendung anzubinden.

Dies lag unter anderem daran, dass die Dependency und Bean injections sich als überaus komplizierter Prozess erwiesen, weswegen nach langem Versuchen wir uns doch für Java Spring entschieden haben.

FH Wlan:

Die Wartezeit zum Ansprechen von der MongoDB Datenbank über das von der FH Campus Wien bereitgestellte Netz war zu lang, was zum Scheitern des Ansprechens der Datenbank führte. Dieses Problem sind wir umgangen, indem wir anstatt einen Hotspot verwendet haben.