# YUMMLY 3.0 - README

This paper contains an executive README of the application **YUMMLY 3.0**, created by Group 1 of the course 'Applications in Object-Oriented Programming and Databases'.

## Prerequisites

- **Python version**: python3
- **Python Packages**: check out the file `prerequisits.txt`!

## Application

This section provides an overview over the functionality of the application.

### Functionality

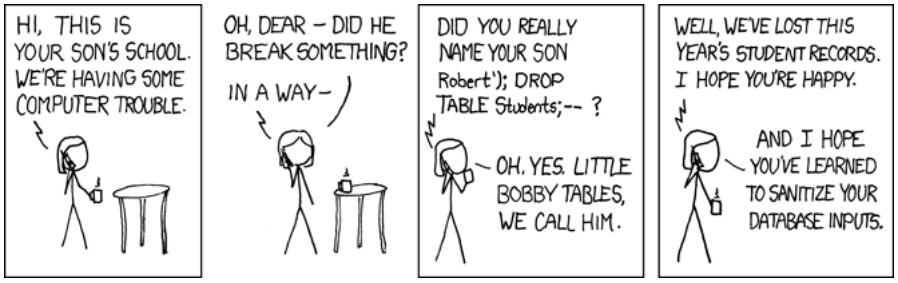
### User Interface

### Under the hood

#### Database

#### APIs

#### Security




## Project

This section describes the project from an organizational perspective. The software development project is in itself not easy to handle, especially for business students who do not have a lot of programming knowledge.

## Organization

To collaborate successfully, we used the version control system [VCS] git. We created a repository on GitHub and thus were able to work remotely and simultaneously on our project without any problem. It can be regarded as a nice side-take-away, that we all learned to handle the kinks of git.

## Challenges

### On the Search

As our main challenge we regard finding a suitable project that fulfills the needs of the course and doesn't completely surpass the scope. After having changed our minds several times, we tried sticking to the things needed, but doing them as clean as possible. In the end we found a use case that not only provided a lot of fun to us food-loving people. The application we built is usable and equipped with a nice front-end and enriched with some more functionality, it could well be used as a web application by a large community of users.

We have long lived in the belief that we are the only ones who offer such a solution. Unfortunately there's already a group of people that is working on a similar idea. They called their application 'Yummly 2.0'. However, we think that our version has much more potential and thus we named it 'Yummly 3.0'.

### Dealing with Databases

MySQL workbench may seem easy and intuitive at first sight and invites the user to start creating tables and building databases before actually knowing what he does. This happened to us as well and reading theory another hour or two would have saved us quite some time.

Luckily we knew from another project that handling locally hosted databases is quite a struggle when working in a team. Therefore we decided to host our database on a server from the beginning on which took some time to set up, but proved to make many things a lot easier.

An additional difficulty we found when accessing the database via a python package. The functions to be written are often nested and finding an error is quite tedious, especially when the error handling (i.e. try and except statements) have already been implemented and the only output is the the 'answer' of the except statement. What may have helped here is a lecture about error handling. Especially in a project of this scope where we are using many different files and nested scripts, a proper lecture about debugging using Pycharm's integrated debugging function would have provided a lot of additional security while fighting errors.

Q: Why do you never ask SQL people to help you move your furniture?
A: They sometimes drop the table.

The Internet

# Appendix