

# Projektdokumentation Python: ToDo-Liste

## 1. Projekt Beschreibung

Ziel des Projekts ist die Erstellung einer ToDo-Listen-App, welche den Alltag und die Planung bei der Arbeit aber auch im privaten Bereich erleichtern soll. Durch die Anwendung kann man Termine, Tätigkeiten und vieles mehr sammeln und sich gebündelt anzeigen lassen. So kann man schnell einen Überblick über die ausstehende Arbeit erlangen und den Tag besser und konstruktiver nutzen. Dabei soll die ToDo-App eine angenehme Nutzeroberfläche bieten, welche wenig bis gar keine Eingewöhnungszeit zur Nutzung fordert. Es soll installiert und direkt losgelegt werden können.

Funktionen/Anforderungen, welche im Scope sind:

- Neue Einträge müssen erstellbar sein
- Einträge können mit einer Beschreibung versehen werden
- Einträge können mit einer Priorität eingestuft werden
- Einträge können gelöscht werden
- Man kann ein Fälligkeitsdatum oder ein Zeitraum für ein Eintrag/Ereignis bestimmen
- Punkte der ToDo-Liste können vom Ersteller auf „erledigt“ gesetzt werden
- Verschiedene Sortiermöglichkeiten
  - Nach Datum
  - Alpha-Numerisch (Titelabhängig)
  - Nach Priorität
- Alle ToDo-Einträge werden abgespeichert in einer Datenbank
- Benutzersystem mit Registrierung und Login

Nicht im Scope:

- Anbindung an Online-Dienste
  - Datenbank auf einem Server
  - Benutzerverwaltung auf einem Server
- Kategorisierung der Einträge
- Bearbeiten von ToDo's

## 2. Zielplattform

Als Hauptzielplattform dient das Betriebssystem macOS. Während alle Funktionen der ToDo-Liste sowohl unter macOS als auch Windows laufen, entstehen Probleme bei der Benutzeroberfläche. Hier setzt das Python Modul Tkinter nicht auf eine einheitliche Darstellung, sondern bedient sich an Betriebssystem individuellen Widgets. In den nachfolgenden Screenshots ist die GUI auf Basis von macOS zu sehen.

## 3. Module, Klassen und Objekte

### 3.1 Modul welcome

Die ToDo-Liste ist mit dem Modul „welcome“ zu starten. Der Benutzer kann sich bei Erstbenutzung für eine Registrierung entscheiden oder sich über den Button „Login“ Button bei einem bestehenden Konto anmelden. In beiden Fällen erfolgt eine Weiterleitung zu den Modulen „login“ oder „register“.

### 3.2 Module register und login

Neue Benutzer registrieren sich mit Vor- und Nachnamen (Nachname optional), eindeutigem Benutzernamen und einem Passwort. Bei Erfolg wird ein neuer Benutzer in der Datenbank angelegt. Sollten bis auf den freiwilligen Nachnamen nicht alle Daten angegeben werden, wird der Benutzer über eine Fehlermeldung darüber informiert. Meldet sich ein Nutzer bei seinem Account mit Benutzernamen und Passwort an, erfolgt eine Überprüfung, ob der Benutzer existiert und das eingegebene Passwort korrekt ist. Auch hier wird gegebenenfalls eine Fehlermeldung angestoßen.

### 3.3 Modul mainPage

Mit dem Aufruf der MainPage werden Parameter für die Sortierung, der Anzeige aktiver und nicht aktiver Einträge und der User-ID zur eindeutigen Zuordnung der Aufgaben zu den Benutzern übergeben. Bei Registrierung wird der Parameter User-ID zu Beginn von der Datenbank ausgelesen. Links sind in einer Spalte alle ToDo's aus einer Datenbanktabelle aufgelistet, welche durch Klicken auf den jeweiligen Button aufgerufen und rechts angezeigt werden. Alle Details einer Aufgabe werden angezeigt und die Aufgabe kann als erledigt gesetzt, wodurch sie nach Update eines „aktiv“ Wertes in der Datenbank nicht mehr angezeigt wird. Einträge vollständig zu löschen ist auch möglich, in diesem Fall wird auch der Eintrag in der Datenbank vollständig gelöscht. Über das Menüband können die ToDo's sortiert werden oder auch bereits abgeschlossene Aufgaben wieder eingeblendet werden. Jedes Mal wird die Hauptseite mit den neuen Parametern erneut aufgerufen. Zusätzlich können durch das Menüband neue ToDo's angelegt werden. In einem neuen Fenster können alle Angaben für eine Aufgabe eingegeben werden. Für einen erfolgreichen Insert in die Datenbank muss mindestens die Überschrift angegeben sein.

## 4. Externe Module und Hilfsmittel

### 4.1 Benutzeroberfläche mit Tkinter

Für die Umsetzung der GUI greift das Projekt auf das Python Modul „tkinter“ zurück. Aus diesem wurden Widgets, unter anderem Labels, Buttons und Text-Boxen, für die Inhaltsdarstellung angewendet. Ausgenommen der Klasse für neue ToDo's werden alle Klassen im gleichen Fenster als unterschiedliche Frames angezeigt. Die Anordnung des Layouts geschieht in der ToDo-List App je nach Anwendungsfall als Grid, einer Gittermäßigen Platzierung der Widgets oder Pack, einer einfachen Unter- und Nebenanordnung der Widgets. Fehlermeldungen an den Benutzer werden mit Messageboxen ausgegeben.

### 4.2 Datenbank SQLite

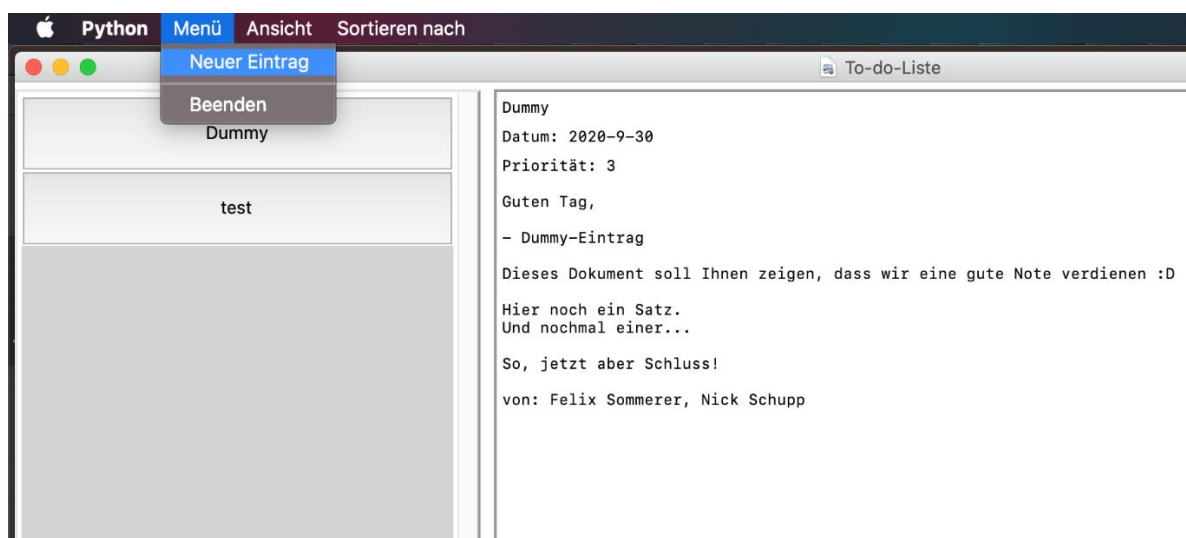
Die ideal integrierte und in Python einfach zu handhabende Datenbank SQLite hilft bei der Verwaltung der Daten. Aufgeteilt in zwei Tabellen für Benutzer und ToDo-Einträge erfolgt eine logische Trennung. Um die Aufgaben in der „MainPage“ den Benutzern richtig zuzuordnen werden die beiden Tabellen dennoch durch einen „JOIN“ miteinander verbunden. Die Anwendung „DB Browser (SQ Lite)“ unterstützt beim Anlegen von Tabellen und beim Erstellen der SQL-Abfragen für das laufende Programm.

## 5. Fazit des Projekts

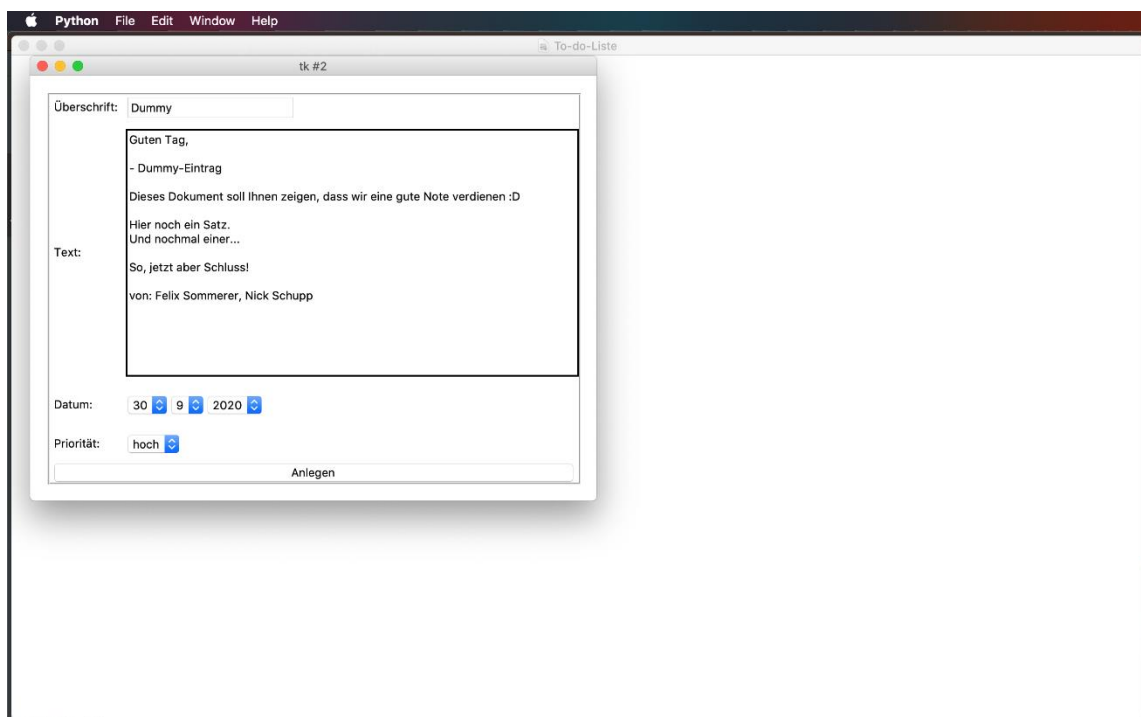
Im Projekt konnte man viele erlernte Python-Grundlagen anwenden. Im Zusammenspiel mit tkinter und SQLite ist ein voll funktionierendes und schönes Projekt entstanden, welches noch auf vielen Wegen erweitert werden kann. Eine Herausforderung hat immer wieder tkinter dargestellt, durch tieferes Einarbeiten in dieses komplexe Modul gab es schlussendlich dennoch eine Lösung für entstandene Probleme.

Lediglich die Darstellung von tkinter auf verschiedenen Betriebssystemen ist wie bereits erwähnt problematisch, dieser Punkt wird bei der Wahl des GUI-Toolkits in zukünftigen Projekten mit einfließen. In diesem Projekt wurden deshalb die Fenster veränderbar gehalten um alle Inhalte einsehen zu können.

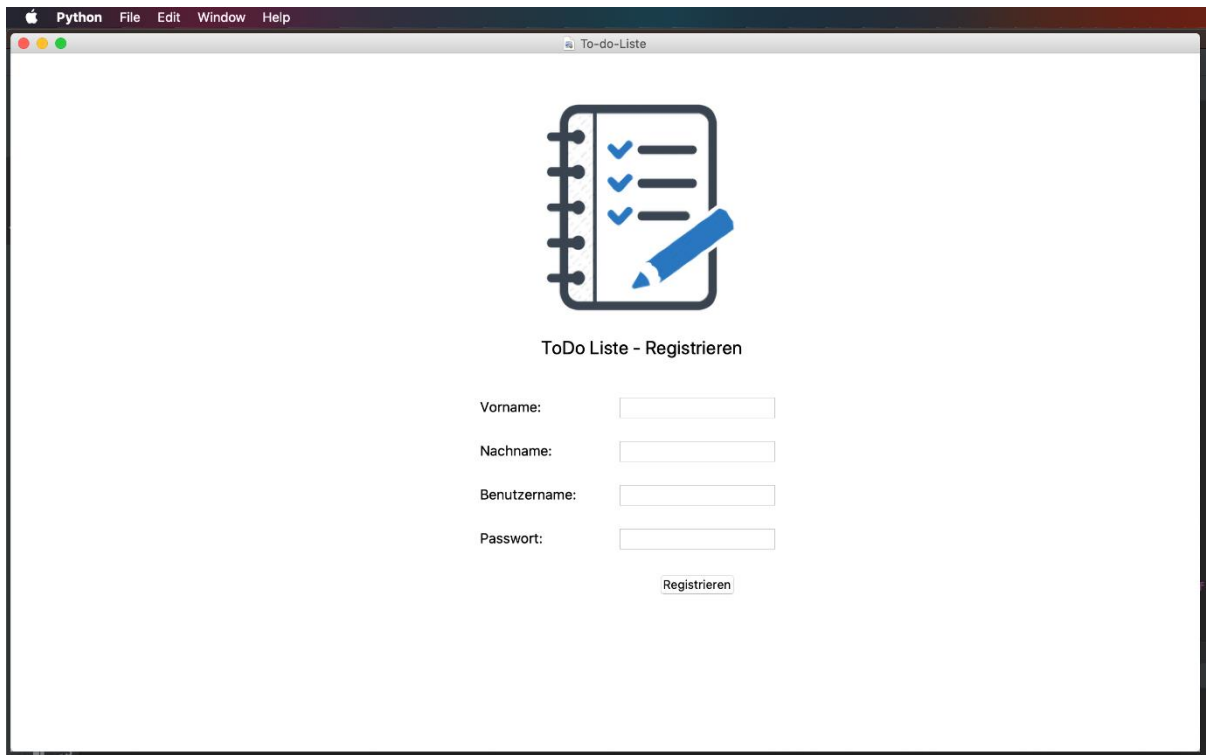
### Referenz Layout bei macOS:



### Menüband




### Neues ToDo anlegen



Python File Edit Window Help

To-do-Liste



ToDo Liste - Registrieren

Vorname:

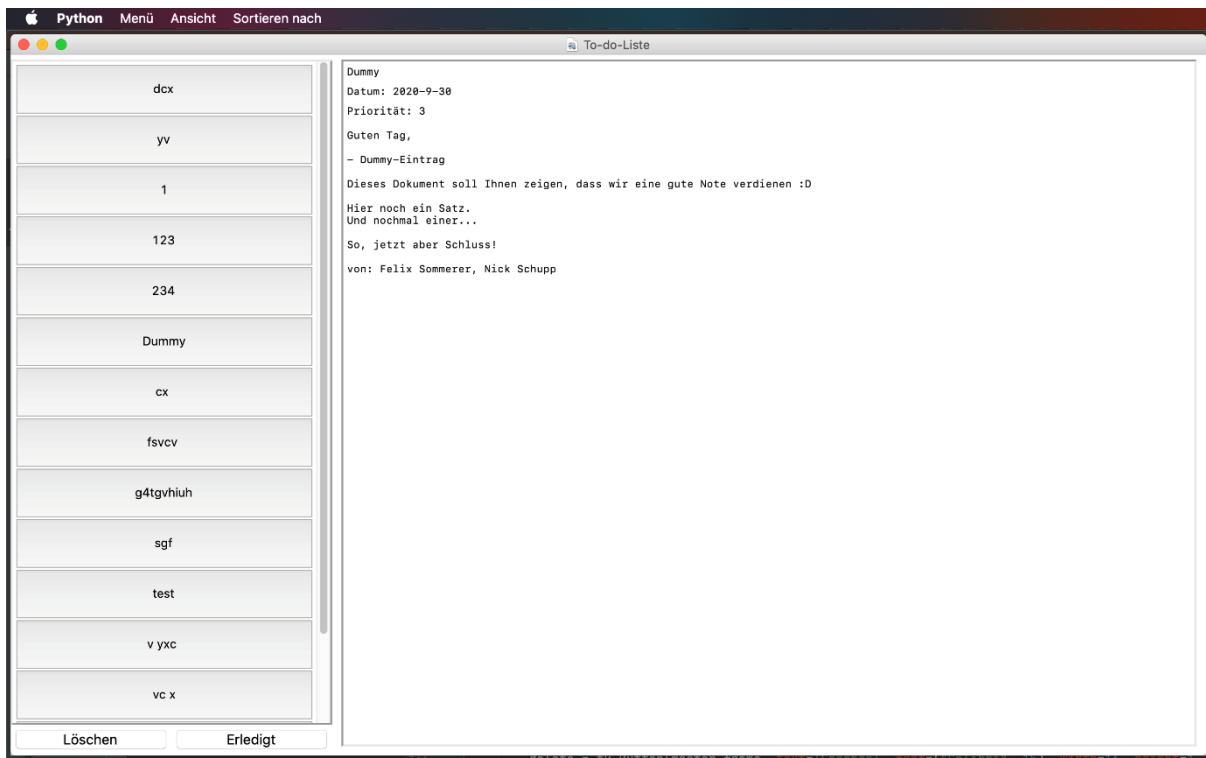
Nachname:

Benutzername:

Passwort:

Registrieren

Registrieren



Python Menü Ansicht Sortieren nach

To-do-Liste

dcx
yv
1
123
234
Dummy
cx
fsvcv
g4tgviuh
sgf
test
v yxc
vc x

Löschen Erledigt

Dummy  
 Datum: 2020-9-30  
 Priorität: 3  
 Guten Tag,  
 - Dummy-Eintrag  
 Dieses Dokument soll Ihnen zeigen, dass wir eine gute Note verdienen :D  
 Hier noch ein Satz.  
 Und nochmal einer...  
 So, jetzt aber Schluss!  
 von: Felix Sommerer, Nick Schupp

`delete = tk.Button(master=frame, text="Löschen", command=lambda: delete_todo(15), width=10, height=1)`

Hauptseite mit ToDos und Detailansicht