



# **Klausurersatzleistung**

des Studienganges Informatik  
an der Dualen Hochschule Baden-Württemberg Stuttgart Campus Horb  
am Neckar

in der Vorlesung C# und .NET

von  
**Felix Wochele**

Dozent  
**Dipl.-Ing. (DH) Nils Schmid**

Abgabedatum: 30. Juni 2022

Bearbeitungszeitraum  
Matrikelnummer, Kurs

KW 20 bis KW 26  
6404744, TINF20 (Informatik)

## Inhaltsverzeichnis

Inhaltsverzeichnis .....	II
Abbildungsverzeichnis .....	III
1. Grundlegendes .....	1
2. Starten der Anwendung .....	2
3. Oberfläche und Funktionen .....	4
3.1. Erläuterung der Oberfläche .....	4
3.2. Bestandteile der Oberfläche .....	5
3.3. Beweggründe .....	9
4. Aufbau und Architektur.....	11
4.1. Struktur.....	11
4.2. Vorgehen.....	12
5. Versionsverwaltung.....	13
Anhang 1 .....	IV

## Abbildungsverzeichnis

Abbildung 1: 2Do - App Logo.....	1
Abbildung 2: Eigenschaftsseite der Solution.....	3
Abbildung 3: Hauptfenster der 2Do - App.....	4
Abbildung 4: Listenansicht der Kategorien .....	5
Abbildung 5: Listenansicht der Listen .....	5
Abbildung 6: Ausschnitt Aufgeban - Detailansicht .....	6
Abbildung 7: Auswahl Sortierung.....	7
Abbildung 8: Fenster "Aufgabe hinzufügen" .....	8
Abbildung 9: Fenster "Liste hinzufügen" .....	8
Abbildung 10: Fenster "Aufgabe hinzufügen" .....	8
Abbildung 11: Adobe Phtoshop Gradiationskurve .....	9
Abbildung 12: Auszug aus dem Github Rep .....	13

## 1. Grundlegendes

Als Grundlage des Projektes dient das in der Vorlesung ausgeteilte Dokument „INF2020 Klausurersatzleistung.pdf“. Zu Beginn des Projektes ergab sich der Projektname „2Do – App“. Im Folgenden wird die Anwendung und das Projekt folglich als *2Do – App* bezeichnet.

Im Zuge des angegebenen gestalterischen Freiraums wurde ein Logo für die Anwendung entwickelt. Dieses Logo ist auch in der Anwendung zu finden.



Abbildung 1: 2Do - App Logo

## 2. Starten der Anwendung

Die Anwendung ist mit dem in der Vorlesung verwendeten Programm *Microsoft Visual Studio* entwickelt worden. Um die Anwendung auszuführen, ist auch diese Entwicklungsumgebung zu verwenden.

Die *2Do – App* kann, wenn nicht anders verfügbar, von der Versionsverwaltung heruntergeladen werden. Als Versionsverwaltung wird *GitHub* verwendet. Das Projekt ist unter folgender URL verfügbar.

<https://github.com/FelixWochele/2DO>

Nach erfolgreichem herunterladen oder kopieren des Projektes, können die Dateien mithilfe von Visual Studio geöffnet werden. Visual Studio verwendet zum Öffnen die Solution Datei im Verzeichnis „\2DO\ 2DO.sln“.

Nun können die Projekte innerhalb der Entwicklungsumgebung ausgeführt werden. Es gilt zu beachten, dass die Projekte nicht als eine gemeinsame Instanz gestartet werden. Die ist aufgrund des Aufbaus und der Architektur des Projektes nicht möglich. Um die Projekte in unterschiedlichen Instanzen zu starten, sind folgende Schritte notwendig.

Zu Beginn ist die Projektmappe als Startprojekt festzulegen. Ist dies geschehen muss auf der erscheinenden Eigenschaftenseite der Projektmappe die Auswahl „*Mehrere Startprojekte*“ getroffen werden. Unter diesem Punkt können dann die einzelnen zu startenden Projekte ausgewählt werden. Bei einem normalen Start der Anwendung sind das der *2Do-Service* und der *2Do-Client*.

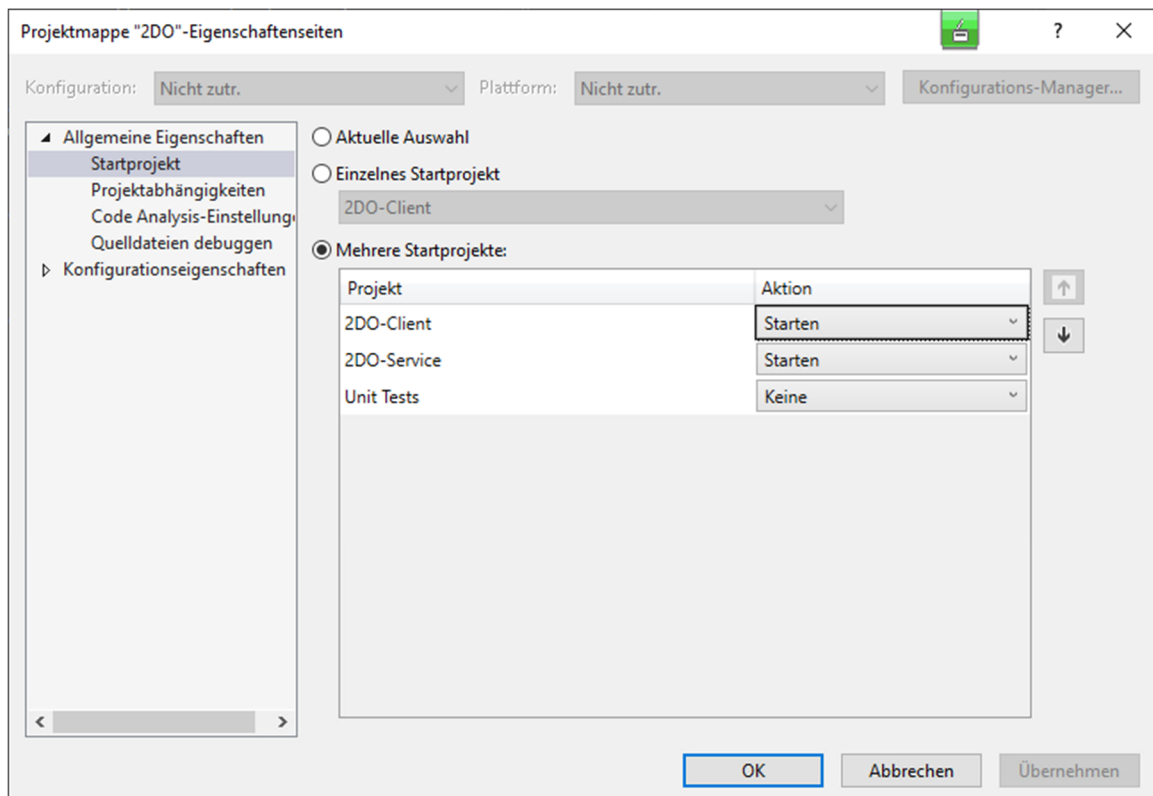


Abbildung 2: Eigenschaftenseite der Solution

Im Verzeichnis des Projekts sind unter anderem auch Beispiellisten. Da die verwendete SQLite Datenbank vom Kunde stammt und keine Änderungen erfolgen durften, ist sie zu Beginn leer.

Die Beispiele können zur Anschauung importiert und bearbeitet werden. Siehe hierzu *3. Oberfläche und Funktionen*.

(Beigefügte Beispiellisten: Einkaufen.xml, Kinofilme.xml, Serien.xml, DHBW-ToDo.xml, Geschäft-ToDo.xml)

### 3. Oberfläche und Funktionen

Die Oberfläche ist eines der wichtigsten Kernelemente der *2Do - App* Anwendung. Der Nutzer kann nur über die Oberfläche mit der Anwendung interagieren. Damit ist die Oberfläche die einzige Schnittstelle für einen Nutzer und ist deshalb umso wichtiger.

#### 3.1. Erläuterung der Oberfläche

Nach dem Ausführen der *2Do - App* wird dem Nutzer ein Splashscreen angezeigt. Hat die Anwendung vollständigen geladen, erscheint das Hauptfenster der *2Do - App*. Folgen wird das Hauptfenster dargestellt.

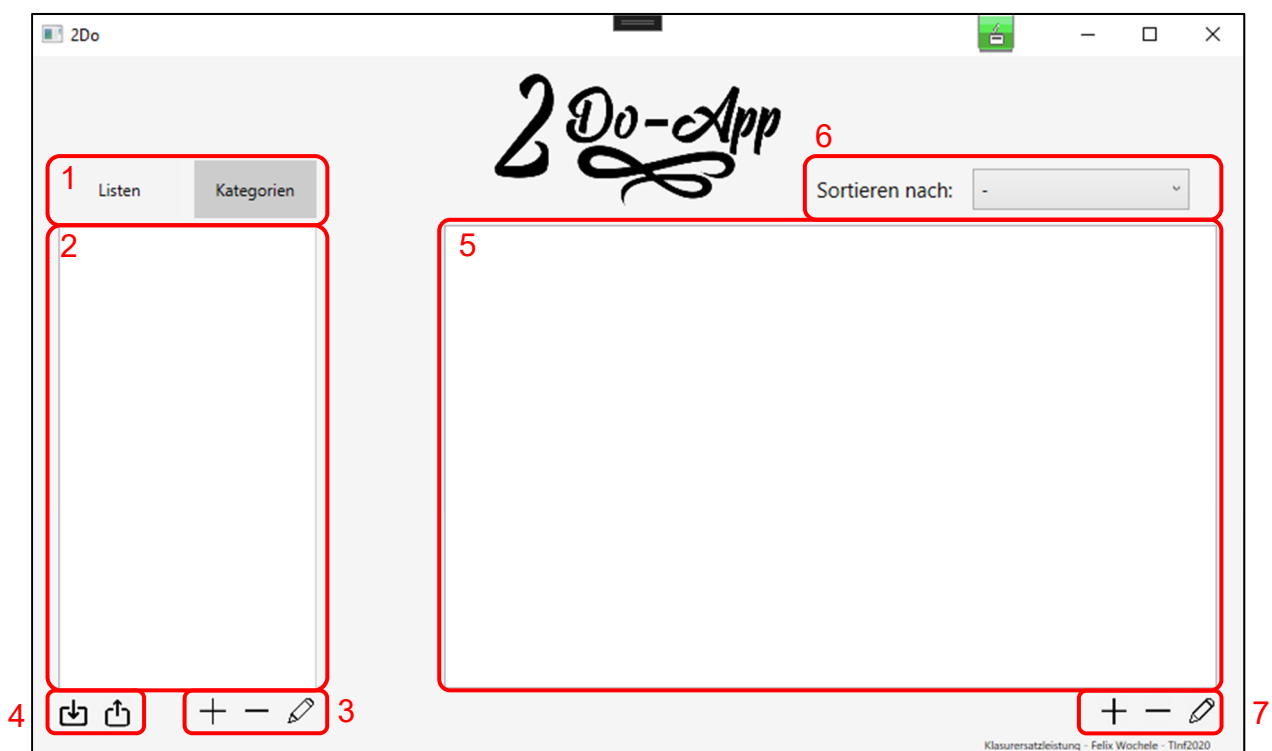


Abbildung 3: Hauptfenster der 2Do - App

Das Hauptfenster ist wie zu erkennen in zwei große Bereiche eingeteilt. Auf der linken Seite befindet sich die Auswahl der Listen und Kategorien. Auf der rechten Seite befindet sich die Detailansicht der ausgewählten Liste oder Kategorie. In dieser werden die einzelnen untergeordneten Aufgaben angezeigt. Folgen wird näher auf die Bestandteile der Oberfläche eingegangen.

### 3.2. Bestandteile der Oberfläche

- 1) Das sind die Buttons für die Wahl zwischen Listen und Kategorien. Durch einen Mausklick auf den jeweiligen Button werden in der darunterliegenden Anzeigefläche (2) die einzelnen Listen oder Kategorien aufgezeigt. Erkennbar ist, dass standardmäßig bereits die Listenanzeige ausgewählt ist. Daher ist nur der Button *Kategorien* auswählbar. Dies wird zusätzlich durch den grauen Hintergrund des Buttons angezeigt. Bei Auswahl der Kategorien ist der *Listen* Button grau hinterlegt.
- 2) Hier werden jeweils die Listen oder Kategorien aufgelistet. Es ist möglich, durch einen Mausklick auf eine Liste oder Kategorie diese auszuwählen. Die genauere Auflistung der einzelnen Einträge ist in danach 5 zu erkennen.

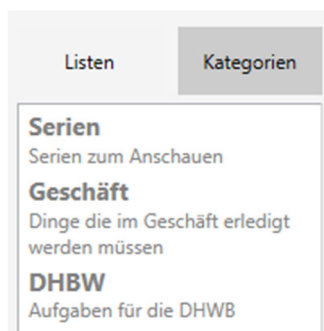


Abbildung 5: Listenansicht der Listen

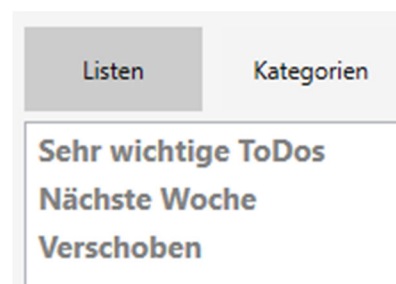


Abbildung 4: Listenansicht der Kategorien

- 3) Das sind die Bearbeitungsmöglichkeiten für die einzelnen in 2 angezeigten Einträge. Dabei gibt es drei Möglichkeiten.
  - hinzufügen (+)  
Dieser Button ist immer verfügbar. Bei betätigen des Buttons wird dem Nutzer ein neues Fenster angezeigt, in welchem er eine neue List oder Kategorie hinzufügen kann.
  - entfernen (-)  
Dieser Button ist nur verfügbar, solange eine Liste oder Kategorie ausgewählt ist. Ist keine Liste oder Kategorie ausgewählt, so ist dieser Button deaktiviert.



Zu beachten ist, dass beim Löschen einer List auch alle Aufgaben dieser Liste gelöscht werden. Dagegen kann das Löschen einer Kategorie nur durchgeführt werden, wenn alle Aufgaben aus der Kategorie entfernt wurden.

- bearbeiten (Stift)

Dieser Button ist wie der entfernen Button, nur bei aktiver Auswahl einer Liste oder einer Kategorie verfügbar. Mit einem Mausklick auf diesen Button erscheint wie beim Hinzufügen einer Aufgabe oder Kategorie ein zusätzliches Fenster. Der einzige Unterschied ist, dass die vorhandenen Daten in diesem Fenster schon eingetragen sind und nun bearbeitet werden können.

- 4) Hier finden sich die Import- und Exportfunktionen. Durch den linken Button können Listen importiert und durch den rechten Button exportiert werden. Nach betätigen des jeweiligen Button öffnet sich ein neues Dialogfenster, in welchem eine Datei ausgewählt oder gespeichert werden kann.

Beim Export wird eine *.xml* Datei mit den Informationen erzeugt. Importiert werden können nur die erzeugten *.xml* Dateien. Alternativ auch anderweitig erzeugte *.xml* Dateien, jedoch müssen diese strikt dem vorgegebenen Schema entsprechen. Das *.xml* Schema einer beigelegten Beispielliste ist im *Anhang 1* zu sehen.

Es wird bei jedem Export die ganze Liste mit den jeweiligen Einträgen Exportiert. Die exportierte Liste ist nach dem Export weiterhin verfügbar.

- 5) Dieses Fenster ist für die Detailansicht der ausgewählten Liste oder Kategorie. Dabei werden die einzelnen Aufgaben aufgelistet. Neben dem Titel der Aufgabe sind auch weitere Details zu sehen.

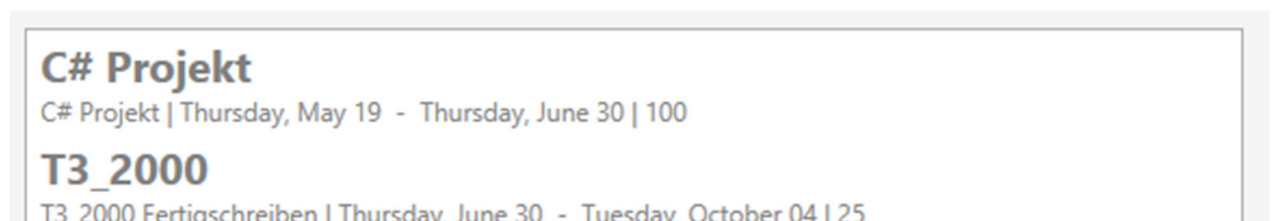


Abbildung 6: Ausschnitt Aufgabe - Detailansicht

6) Hier kann die Sortierung der Auflistung bearbeitet werden. Standardmäßig werden neue oder bearbeitete Aufgaben unten an der Liste angehängt. Es gibt wie in der Aufgabenstellung gefordert, folgenden Sortierungsmöglichkeiten.

- Anlagedatum
- Alphabetisch
- Fälligkeitsdatum
- Priorität
- Wichtigste

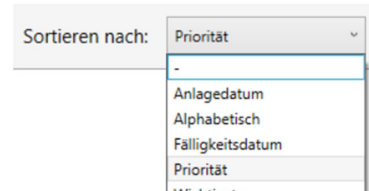


Abbildung 7: Auswahl Sortierung

Nach jedem Wechsel der Kategorie wird die Sortierung wieder zurückgesetzt (vgl. 3.2 Beweggründe).

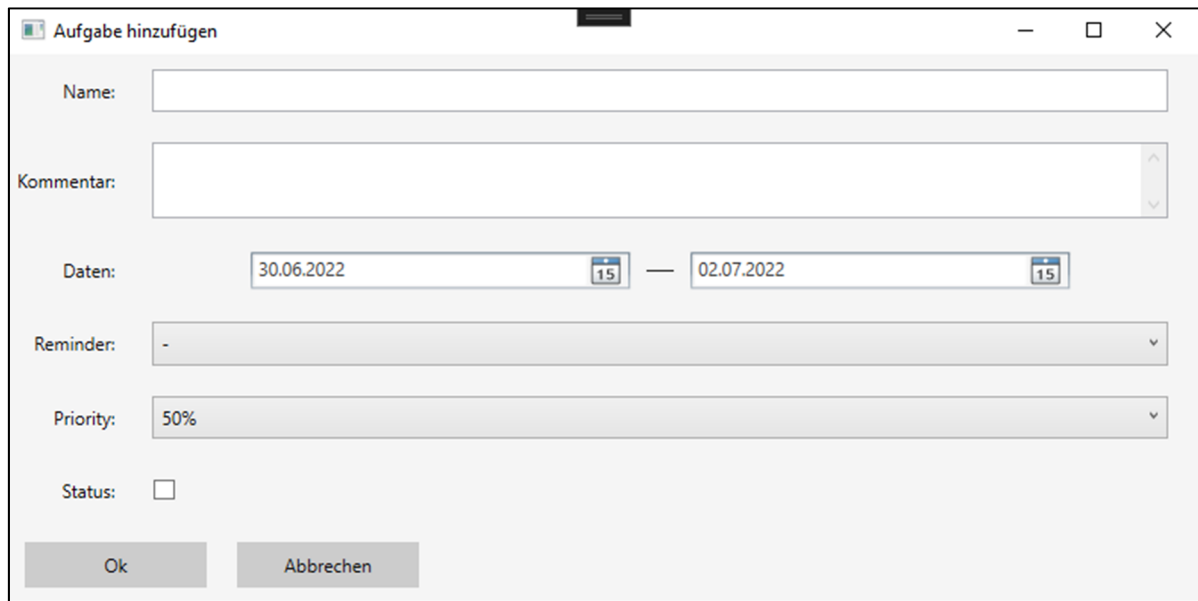
7) Es handelt sich um die gleichen Bedienelemente wie bei 2), nur sind diese auf die einzelnen Aufgaben bezogen.

Über das plus Symbol lassen sich, bei Auswahl einer Liste, einzelne Aufgaben erzeugen. Diese sind danach in der ausgewählten Liste enthalten. Wird bei der Auswahl einer Kategorie das plus Symbol betätigt, so besteht die Möglichkeit, aus einer Auflistung aller Aufgaben, eine Aufgabe in die Kategorie aufzunehmen.

Beim Entfernen einer Aufgabe aus einer Liste verschwindet diese auch aus den Kategorien. Wird eine Aufgabe aus einer Kategorie entfernt, so bleibt diese aber in der Liste vorhanden.

Editierbar sind die Aufgaben in jeder Auswahl. Sowohl bei ausgewählter Liste als auch bei einer ausgewählten Kategorie.

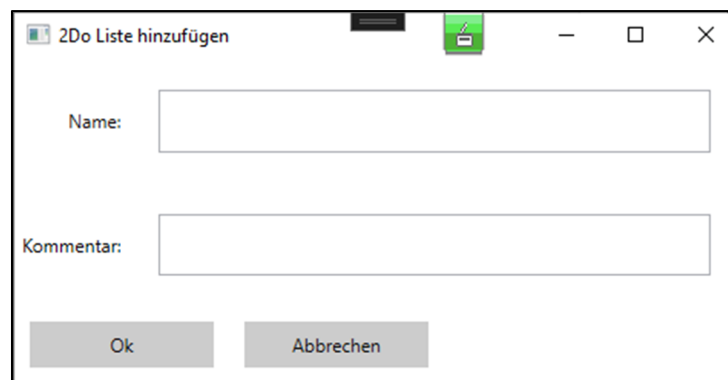
Folgend sind noch die erwähnten Fenster zum Erstellen und Editieren der einzelnen Komponenten dargestellt. Diese verfügen über Eingabefelder für die in der Aufgabenstellung geforderten Eigenschaften.



The dialog box titled 'Aufgabe hinzufügen' contains the following fields and controls:

- Name:** A single-line text input field.
- Kommentar:** A multi-line text area with a vertical scrollbar.
- Daten:** Two date pickers. The first is set to '30.06.2022' and the second to '02.07.2022'. Both have a small calendar icon with the number '15'.
- Reminder:** A dropdown menu currently showing '-'. A small downward arrow is on the right.
- Priority:** A dropdown menu currently showing '50%'. A small downward arrow is on the right.
- Status:** A checkbox that is currently unchecked.
- Buttons:** 'Ok' and 'Abbrechen' buttons at the bottom.

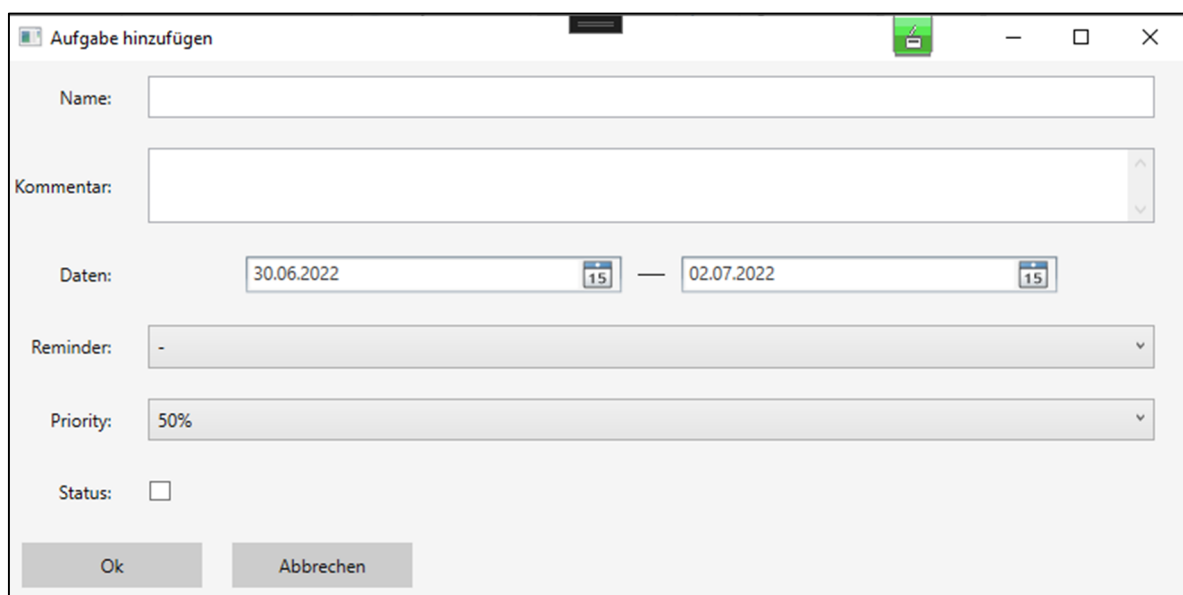
Abbildung 8: Fenster "Aufgabe hinzufügen"



The dialog box titled '2Do Liste hinzufügen' contains the following fields and controls:

- Name:** A single-line text input field.
- Kommentar:** A multi-line text area with a vertical scrollbar.
- Buttons:** 'Ok' and 'Abbrechen' buttons at the bottom.

Abbildung 9: Fenster "Liste hinzufügen"



This is another instance of the 'Aufgabe hinzufügen' dialog box, identical to the one in Abbildung 8, but it includes a small green icon with a white document symbol in the title bar.

Abbildung 10: Fenster "Aufgabe hinzufügen"

### 3.3. Beweggründe

Das Ziel war es, eine möglichst einfach zu bedienende und übersichtliche ToDo – App zu entwickeln. Daher wurde auf die Aufteilung in verschiedene Unterseiten und Tabs verzichtet. Alles ist auf einer kompakten Hauptseite ersichtlich.

Auf der linken Seite die Grobauswahl und auf der rechten Seite die Detailansicht. Dies ist bei vielen modernen Anwendungen so, darunter auch bei Office Outlook, Intelij und Microsoft Visual Studio. Dabei variiert die horizontale Anordnung, aber das Schema bleibt das Selbe.

Beim Hinzufügen von Elementen wurde ein extra Fenster als beste Lösung angesehen. Dadurch kann der Nutzer ggf. zeitgleich einen Blick auf andere Aufgaben werfen und eine neue Aufgabe anlegen.

Die Elemente zum Hinzufügen, Löschen und Bearbeiten sind nahe an der zugehörigen Auswahl angeordnet. Diese Anordnung von Listen und Buttons ist auch bei diversen anderen Anwendungen, wie den Windows 7 Systemeinstellungen oder bei Adobe Photoshop vorzufinden.

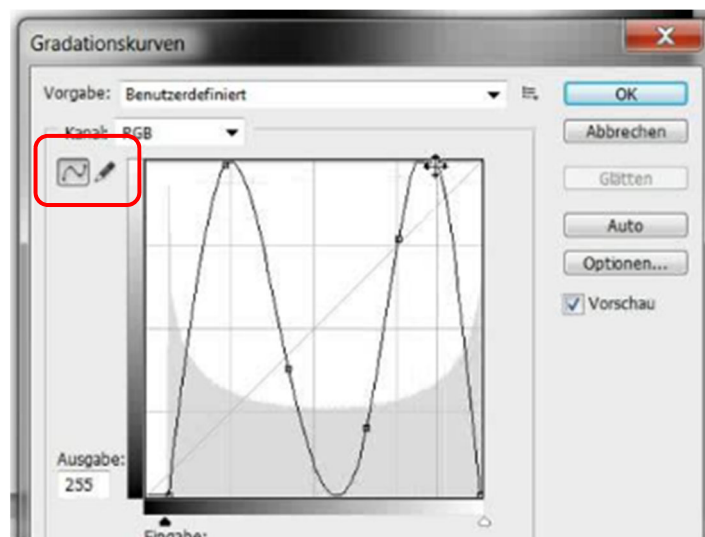


Abbildung 11: Adobe Phtoshop Gradiationskurve

Es wurde auch bewusst auf den Doppelklick zum Bearbeiten der Listenelemente verzichtet, da ohne Doppelklick das Auswählen und Durchsuchen der Listen einfacher gestaltet wird.

Wie in 3.1 *Erläuterung der Oberfläche* erwähnt, werden neue Listenelemente oder zuletzt bearbeitete Listenelemente unten angehängt. Das Anhängen neuer oder bearbeiteter Listenelemente soll erreichen, dass immer die am längsten vorhandenen oder nicht bearbeitete oben ist. Das bringt den Vorteil, dass Aufgaben nicht vergessen werden und dadurch dem Nutzer ins Gedächtnis gebracht.

## 4. Aufbau und Architektur

### 4.1. Struktur

Das Projekt ist wie vorgegeben in einer MVVMC Architektur erstellt worden. Darüber hinaus wurden weitere Unterordner verwendet um die Übersicht im Projekt zu gewährleisten. Der Ordneraufbau ist wie folgt strukturiert:

#### **Client:**

- Controller
  - AddTaskWindowController
  - Mapping
  - Submodels
- Framework
- Service
- ViewModels
  - AddWindow
  - Submodels
- Views
  - AddTaskWindowViews
  - Assets
  - Submodels

#### **Service:**

- Assets
- Database
  - Data
  - Maps
- NHibernate

Die weiteren Ordnerstrukturen im Client, neben dem übliche MVVMC Model, dienen zur Abgrenzung der zusätzlichen Fenster zu dem Hauptfenster.

Alle in der Aufgabenstellung angegebenen Pakete wurden verwendet und keine weiteren eigenen Pakete installiert.

## 4.2. Vorgehen

Beim Vorgehen wurde als erstes die MVVMC Ordnerstruktur erstellt. Vom Client ausgehend fand die Autofac Konfiguration statt. Das ermöglichte das grundlegende Ausführen der Anwendung.

Im Folgenden wurde der WCF Service erstellt und eine Verbindung mit dem Client ermöglicht. Im Service ist anschließend die Verbindung mit der Datenbank eingerichtet worden und es wurde sichergestellt, dass der Client einen Zugriff auf die Datenbank mithilfe des Service hat.

Nach einem Funktionstest der bisherigen Elemente und der Kommunikation ist mit der Implementierung der Oberfläche begonnen worden.

Die Oberfläche wurde von der Hauptseite aus entwickelt und nach und nach wurden einzelne Funktionen ergänzt.

Abschließend sind die import und export Funktionen von Listen ermöglicht worden und es wurde die Sortierung realisiert.

Designanpassungen fanden als letztes statt. Hierbei war jedoch essentiell, dass im Voraus schon ein Konzept entworfen wurde, welches bei der Implementierung der Oberfläche berücksichtigt werden konnte. So mussten abschließend nur minimale Änderungen vorgenommen werden.

## 5. Versionsverwaltung

Im Projekt wurde zum strukturierten Arbeiten eine Versionsverwaltung eingesetzt. Dabei fiel die Wahl auf GitHub (Vgl. 1. *Starten der Anwendung*). Dies erleichterte die Arbeit immens. Es war möglich von verschiedenen Arbeitsplätzen mit der gleichen Version zu Arbeiten. Aber auch das Testen gestaltete sich einfacher. So konnten lokale arbeitsplatzbezogenen Einstellungen die an das Projekt gebunden sind ausgeschlossen werden.

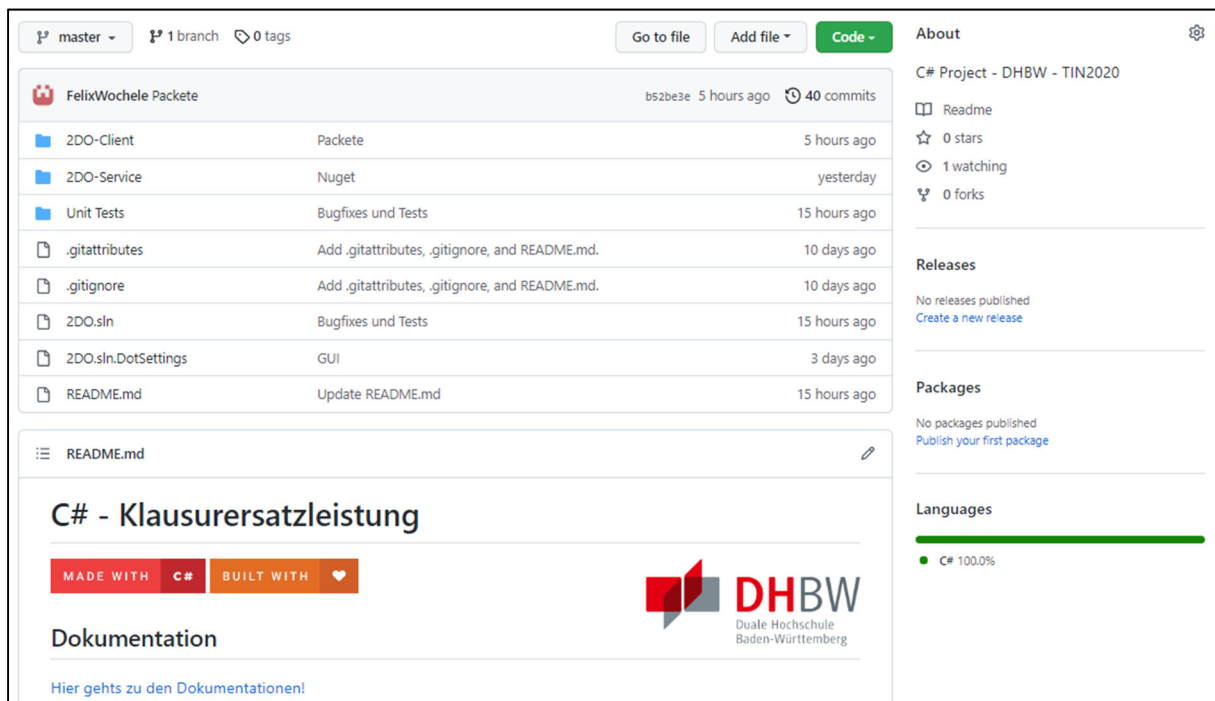


Abbildung 12: Auszug aus dem Github Rep



## Anhang 1

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <XMLExportMap xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3.   <TasList>
4.     <Comment>Beispiel Liste</Comment>
5.     <Description>XML Export</Description>
6.     <ID>4</ID>
7.     <Version>0</Version>
8.   </TasList>
9.   <Tasks>
10.    <Comment>Aufgabe 1</Comment>
11.    <CreationDate>2022-06-30T15:34:59.2082218</CreationDate>
12.    <Description>Kommentar1</Description>
13.    <DueDate>2022-07-02T15:34:59.2082234</DueDate>
14.    <ID>18</ID>
15.    <Priority>50</Priority>
16.    <ReminderMinutes>3</ReminderMinutes>
17.    <State>true</State>
18.    <TasklistID>4</TasklistID>
19.    <Version>0</Version>
20.  </Tasks>
21.  <Tasks>
22.    <Comment>Aufgabe 1</Comment>
23.    <CreationDate>2022-06-30T15:35:11.7581233</CreationDate>
24.    <Description>Kommentar 2</Description>
25.    <DueDate>2022-07-02T15:35:11.7581247</DueDate>
26.    <ID>19</ID>
27.    <Priority>50</Priority>
28.    <ReminderMinutes>2</ReminderMinutes>
29.    <State>false</State>
30.    <TasklistID>4</TasklistID>
31.    <Version>0</Version>
32.  </Tasks>
33. </XMLExportMap>
```