

# Tour-Planer

## Wintersemester 2021

Ersteller: Wolfgang Witschel

[if19b126@technikum-wien.at](mailto:if19b126@technikum-wien.at)

<https://github.com/GoldiFirstOfHisName/TourPlanner>

---

### 1. Oktober 2021

Die Angabe wurde verfügbar gemacht und ich habe mit der Erstellung des grundsätzlichen User Interfaces begonnen.

- Das Farbschema wurde gewählt um meine Augen bei der Arbeit zu schonen, da ich sehr oft erst spät in der Nacht nach der Arbeit anfangen konnte an dem Projekt zu arbeiten. Dadurch ergaben sich diverse Vor und Nachteile
  - Manuelle Farb, Komposition und Code Suche.
  - Schlechterer Kontrast für Farbenblinde.
  - Viel gelernt über WPF XAML properties, und ihre Anwendung/Missbrauch.
- Das Layout wurde leicht umstrukturiert, um mehr Platz zum anzeigen von Listviews.gridviews zu ermöglichen.
  - Einfacher lesbar.
  - Moderneres Design, mehr Whitespace ist angenehmer für die Augen

(Im falle keiner genaueren Angabe kann davon ausgegangen werden das ich bis zum nächsten eintrag mit der Arbeit des davorigen beschäftigt war)

### 9. Oktober 2021

Nachdem ich das User interface abgeschlossen hatte, entschloss ich mich an das aufteilen nach MVVM zu machen, dazu erstellte ich 3 Views(user controls) und blendete diese für die 3 Haupt widgets ein.

- Ich habe mich aktiv für user controls entschieden, da mir Pages mehr probleme entgegen geworfen haben, zwar werden die Views hier nicht wieder benutzt wie es User Controls sollten, aber der Code wirkt viel sauberer
- DataBinding stellte sich als ein kleines problem heraus, das aber dank eines Service containers schnell und einfach gelöst wurde. Zu diesem Zeitpunkt wusste ich noch nicht wie viel ich in diesen Container reinstopfen würde. Außerdem benutze ich hier auch einen Singleton, damit hab ich direkt mein design pattern abgehackt. Ich rufe alle Services als Singelton auf, damit ich die selbe instanz ohne probleme für alle ViewModel benützen kann. Es vereinfacht die benutzung einfach enorm.

## 19.Oktober 2021

Nach einer unfreiwilligen Arbeitspause habe ich mich daran gemacht meine wiederbenutzbare UI Komponente zu definieren, ich präsentiere, einen Button.

```
<!--Main Style for Button-->
<Style x:Key="MainStyleButton" TargetType="Button">
  <Setter Property="Background" Value="#FE6584"/>
  <Setter Property="Cursor" Value="Hand"/>
  <Setter Property="Foreground" Value="White"/>
  <Setter Property="Margin" Value="5,3,5,3"/>

  <Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Background" Value="#FF4C70"/>
    </Trigger>
  </Style.Triggers>
</Style>
```

```
<!--Template for Main Button-->
<ControlTemplate x:Key="MainTemplateButton" TargetType="Button">
  <Border Width="auto"
          Height="auto"
          CornerRadius="12"
          Background="{TemplateBinding Background}">

    <ContentPresenter VerticalAlignment="Center"
                      HorizontalAlignment="Center"/>

  </Border>
</ControlTemplate>
```

---

Sehr einfallsreich ich weis, aber in Not frisst der Teufel Fliegen. Dies macht den Code nicht nur wesentlich einfacher lesbar, es sorgt auch dafür das ich mit den Farben experimentieren konnte ohne viele einzelne Werte zu manipulieren.

## 21. Oktober 2021

An dieser Stelle habe ich angefangen das Projekt in Layer zu trennen, nach einiger recherche und hilfe von Studienkollegen bin ich dazu gekommen eine gute Trennung durch zu führen, und mit hilfe des Service-Containers konnte ich alle Komponenten initialisieren ohne das Layering zu verletzen. An dieser Stelle habe ich mir auch eine Grobe aufteilung für die Layer Methoden überlegt, auch wenn ich diese später nach Verlangen geändert habe.

- UI
  - Sehr einfache Funktionen die keine Berechnungen ausführen, aber die zuweisung von Variablen oder der aufruf von internen Methoden sollte hier erledigt werden. Z.b. Das verschieben des Fensters durch das halten der oberen Randleiste
- BL
  - Dieser Layer hatte als hauptaufgabe als Rechenzentrum und Übersetzer zu fungieren, die Methoden die sich hier befinden haben entweder vital Informationen von Layer zu Layer weiter getragen, oder Daten ergänzt. Z.b. Das hinzufügen einer tour beginnt im UI wo die TourDaten erfasst werden, dies werden dem BL übergeben, der daraufhin klargestellt das die Tour hinzugefügt werden soll, und übergibt dann den erweiterten block an Information an den DAL
- DAL
  - Der DAL sollte grundlegend nur Daten ein oder aus speisen, er darf nicht selber Rechnen oder Neue Daten kreieren. Z.b. Ein Tour-Update muss alle Tours aus der Datenbank fetchen, der DAL darf alle Daten sammeln,

---

speichern und weiter versenden, aber er darf nicht aus 2 daten eine neue berechnen (dieses Konstraint habe ich mehrmals verletzt)

## 23. Oktober 2021

Jetzt wo ich ein Layout und Layer fertig hatte konnte ich mich an die Datenbank machen, es dauerte nicht zu lange eine Verbindung über die AppSettings.json her zu stellen, und mit ein wenig placeholder Magie konnte ich nach wenigen Tagen arbeit auch schon touren hinzufügen, Löschen und Editieren. Der größte aufwand entstand hier durch das DataBinding, ich bin mir bis jetzt unsicher ob ich es richtig gemacht habe. Die meisten Daten werden problemlos abgerufen, aber die Comboboxen haben mir extremes Kopfzerbrechen bereitet, und funktionieren immer noch nicht genau so wie ich es will.

Allerdings habe ich diese Zeitraum direkt genutzt um mich ausgiebig mit dem auslesen der AppSettings datei zu beschäftigen und ich habe in weiterer Arbeite alle für mich Relevanten Konfigurationen dort hinterlegen können, vielleicht hätte ich noch weitere Werte hinterlegen können, aber keiner schien mir zur Zeitpunkt der erstellung dieses Dokuments relevant genug.

```
"ConnectionStrings": {
  "DB": "Host=localhost;Username=postgres;Password=camel100;Database=postgres"
},
"ApiKey": {
  "Mapquest": "bEZdCy6muY3GGjFew5AyKfi8CGEto1HY"
},
"NLog": {
  "internalLogLevel": "Info",
  "internalLogFile": "c:\\temp\\internal-nlog.txt",
  "extensions": [
    { "assembly": "NLog.Extensions.Logging" }
  ],
  "targets": {
    "file": {
      "type": "File",
      "fileName": "c:\\temp\\nlog-all-${shortdate}.log",
      "layout": "${longdate}|${event-properties:item=EventId_Id}|${uppercase:${level}}|${logger}|${message} ${exception:format=tostring}"
    }
  },
  "rules": [
    {
      "logger": "*",
      "minLevel": "Info",
      "writeTo": "file"
    }
  ]
}
```

---

## 25. Oktober 2021

An diesem Datum ist meine Toilette eingestürzt, nein kein Witz an dieser Stelle.



Ich wünschte ich könnte Sagen das ich trotz dieser Probleme heldenhaft weiter an meinem Projekt gearbeitet habe, aber die Ziegelsteine die es aufgrund der Bauarbeiten in der Wohnung über mir geregnet hat, haben mir die Lust vertrieben, unter jener einbruch gefährdeten Decke zu arbeiten.

Nichts desto trotz habe ich mich mit meinem Arbeitslaptop weiter an die Arbeit gemacht. Da ich aber nur einen Bildschirm mit einer gefühlten 720p Auflösung hatte habe ich in den nächsten Tagen leider

nur PopUps implementiert, diese dienen einfach dem User indem sie den Input aufsammeln den der netter Herr (oder die Frau, wir wollen ja Geschlechtskonform sein) da heute so hingekackt hat. Nachdem ich der Tour class basiswerte zugewiesen habe, ging das auch wunderbar, ohne basiswerte wurden die Felder aufgrund der großen Zuweisung von auto oftmals einfach flach generiert, was wie wir alle wissen, sehr schlecht zu lesen ist.

## 27. Oktober 2021

Obwohl ich es besser wissen sollte habe ich mich wieder an meinen PC gesetzt, nachdem die Bauarbeiter um punkt 07:00 angefangen haben zu Schlagbohren, entschloss ich mich dazu heute etwas "einfaches" um zu setzen, ich wollte einfach nur SQL Injections

---

vermeiden, nur dumm das ich an dieser Stelle früher immer String bereinigungen gemacht habe, das funktionier hier nicht so gut weil ich eine verdammt große Menge an Daten durch die Gegend schicken und nicht alles davon strings waren (jetzt schon weil ich schlauer bin als ich es zuvor war, amen), also hat mir Gott einen Studien-Kollegen geschickt, der mir erklärt hat das meine Dummheit grenzenlos und mein Größenwahn unberechtigt ist. Wie sich herausstellt kann ich einfach alle meine Strings wie eine Nudel in parameter packen und das ist angeblich sicher. Wenns funktioniert, wer bin ich schon den fortschritt durch Technologie zur verweigern, preiset npgsq und so weiter. Lang Lebe microsoft oder wer auch immer, ich bin zu faul zu googeln wer den ersten commit in dem Projekt gemacht hat. Apropro Commit, hier ist mir auch aufgefallen das ich absolut GARNICHTS auf git habe, das ist natürlich absolute Rotze, aber nachdem ich 80% des Projekts offline gemacht hab, hab ich gesagt :“Scheiß drauf, ich erstelle einfach ein git repo und push das fertig ding da hin, wenn die mir die 0,5 Punkte dafür aberkennen, dann hab ich das verdient, war ja mein Fehler, aber die Kommission besteht ja zu 66% aus echt Netten Lektoren die verstehen sicher das ich aufgrund von Panik einen Fehler gemacht habe und werden das schon nicht tun“. Im Nachhinein sehe ich das jene Aussage sehr, sehr, sehr Gewagt und Dumm war.

Oh und ich habe Tour und TourLog classes in ein Sub Projekt gegeben damit ich einfach darauf zugreifen kann, ich glaube das sollte MVVM und das Layering nicht verletzen da die Datenstruktur hier ja übergreifend benötigt wird.

## 29. Oktober 2021

Meine Wunderschöne Toilette geht wieder, ihr wisst was das heißt, nicht mehr 15min zu

Mama und Papa Fahren weils um mich herum kein öffentliches Klo ohne Junkies gibt!

Meine Produktivität hat sich sofort um 300% gesteigert, und das ist Statistik!

Ich habe in kürzester Zeit, die Darstellung eines bildes, die einbindung von Mapquest, die Berechnung aller Relevanten Tourdaten UND den Export von Touren angeeignet!

Zugegebenermaßen hab ich sehr viele Fehler gemacht und nicht alle perfekt ausgeglichen, aber alles funktioniert.

- 
- Ich benutze das directions API baue mir aber den Link zum bild selber, einfach eine Liste mit zwischenstopps zu bauen und die Bilder mit pfeilen nacheinander anzeigen zu lassen wäre eine schöner umsetzung gewesen, aber die Zeit wurde Knapp.
  - Ich Exportiere in einem unformatierten Json file, ergo es ist absolut unleserlich, nächstes mal würde ich es formatieren und auch wieder einlesen.
  - Es gibt beim generieren von Reports und exportieren von Tours kein user Feedback sie tauchen einfach auf, am ende der Methoden ein PopUp mit dem Dateipfad aufscheinen zu lassen würde das finden der dateien massiv erleichtern.

## 1. November 2021

Endspurt, ich muss noch Reports generieren und ein Protokoll aus meinem absoluten SchweineStall aus Notizen erstellen.

Überraschender weiße erweist sich ltext7 als extrem einfach zu lernen, die bibliothek ist nicht nur extrem gut dokumentiert, sie macht mir sogar brauchbar autocomplete vorschläge mit meinen eigenen Variablen. Es dauerte keine 4 Stunden um die beiden Verlangten Report Arten zu ermöglichen, das einzige was ich besser hätte machen können wäre einen eigenen button für den All Report zu erstellen, im moment kann man den ALL report nur erstellen wenn man KEINE Tour ausgewählt hat, und sobald man eine Tour auswählt kann man nicht mehr davon weg gehen.

Ich habe außerdem angefangen meine Notizen zu sortieren und Zeitblöcke zusammen zu rechnen, hier ist mir aufgefallen das ich sehr viel, aber nicht unbedingt brauchbar geschrieben habe. Allerdings konnte ich sowohl Zeitraum meiner Änderungen und Arbeit als auch Fortschritte rekonstruieren, nächstes mal sollte ich einfach von anfang an Digitale Notizen führen, das dürfte zu massiven verbesserungen führen.

## 2. November 2021

Ich habe diesen Block bis Heute freigelassen, es ist 16:08, ich habe noch ein wenig meinen Code poliert, (mutex für den Datenbank-Zugriff hinzugefügt, unnötige using direktiven entfernt etc.) und schreibe diese letzten Zeilen mit absoluter Panik, ich weis nicht ob mein

---

Projekt gut genug ist, laut der Tabelle auf moodle sollte ich 32 Punkte bekommen, aber ich und auch der wert Leser weis dass dieser Wert sinken wird. Ich habe Fehler gemacht, ich habe das MVVM pattern verletzt und mein Code ist weit weg von sauber, aber ich glaube ich habe ein akzeptables Projekt abgeliefert, und ich hoffe das ich mit dem hier gelernten das nächsten mal nicht zum dritten Antritt darf.

## **Addendum Unit Tests**

Ich hab irgendwie keine Notizen bezüglich der Tests, dem nach muss ich davon ausgehen, das ich sie in einem Fieberwahn geschrieben habe, sie Testen den core in form der CRUD system von Touren und TourLogs. Da das der zentrale Punkt der Anwendung ist, sollten sie auch am meisten getestet werden um die entwicklung der Anwendung zu erleichtern und zu gewährleisten. Andere Test die ich noch hätte schreiben sollen; Test der Datenbank anwendung, diese werden zwar theoretisch mit abgedeckt, aber ihre exceptions werden nicht gefangen, was sie unnütz macht.

## **Addendum 2**