

2022

Jetstream-Service

M322 PROJEKTARBEIT
RAPHAEL HUG

[FIRMENNAME] | [Firmenadresse]

Inhalt

1. Informieren.....	2
1.1. Anforderungen	2
1.2. Was muss ich alles beachten.....	2
1.3. Was muss ich mich informieren	3
1.3.1. Standards.....	3
1.3.2. Binding.....	3
1.3.3. Darstellung	3
2. Planen.....	3
2.1. PSP.....	3
2.2. Zeitmanagement	4
2.3. Wireframe	4
3. Entscheiden	6
3.1. Design.....	6
3.2. Aufbau	6
4. Realisieren	7
4.1. View	7
4.2. Models.....	7
4.3. ViewModel	8
4.4. Logik.....	8
5. Kontrollieren.....	8
5.1. Testen	8
5.2. Checkliste.....	9
6. Auswertung	9
6.1. Fazit	9

1. Informieren

1.1. Anforderungen

Die Firma Jetstream-Service hat neue Touchscreen-fähige Hardware angeschafft (Tablet, Surface) und stellt diese den Mitarbeitern in der Werkstatt und Administration für die Datenpflege der Ski-Service Aufträge zur Verfügung. Die Mitarbeiter können damit standortungebunden in der Werkstatt oder bei der An- oder Rückgabe der Skier die Stammdaten eines Auftrages über handliche mobile Tablets oder Handys mutieren. Grösstmögliches Augenmerk wird dabei auf eine einfache, intuitive und aufgabenangemessene Bedienung gelegt. Die jeweils erforderliche Anzahl Mausklicks oder TippEingaben zur Datenpflege der Aufträge sind dabei entscheidende Bewertungsfaktoren. In der Werkstatt wird aus Sicherheitsgründen mit Handschuhen gearbeitet, auch dieser Punkt gilt es folglich bei der Gestaltung der Benutzeroberfläche zu berücksichtigen. Die bereits existierende Datenbasis zur Online-Anmeldung und der Service-App soll bestehen bleiben und mit der neuen GUI-Lösung für die Tablet- bzw. Handy erweitert werden. In der Hauptsaison sind bis zu 10 Mitarbeiter mit der Durchführung der Serverarbeiten beschäftigt. Diese sollen einen autorisierten passwortgeschützten Zugang zu den anstehenden Aufträgen erhalten und diese zur Abarbeitung übernehmen und ändern können.

1.2. Was muss ich alles beachten

Anforderungen

A1	Login Dialog mit Passwort für den autorisierten Zugang der Mitarbeiter
A2	Erfasste Serviceaufträge müssen übersichtlich angezeigt sein (Listendarstellung)
A3	Die Listenanzeige der Serviceaufträge ist nach Priorität zu sortieren
A4	Listeneinträge können gefiltert werden (Suche von Einträgen)
A5	Mitarbeiter können eine Statusänderung eines Auftrages vornehmen
A6	Die .NET-Anwendung muss nach dem MVVM (Model, View, ViewModel) Design Pattern strukturiert sein.
A7	Datenaustausch per Web-API, REST-Service (PHP/ASP.NET)
A8	Intuitive u. Aufgabenangemessene Bedienung
A9	Grundsätze der Dialoggestaltung aus der Norm EN ISO 9241-110 berücksichtig
A10	Ganzes Projektmanagement nach IPERKA dokumentiert

Zusätzliche Anforderungen

A1	Dashboard mit Statistikdaten usw.
A2	Die Mitarbeiter können zu einem Auftrag einen Freitext bzw. Kommentar hinterlegen
A3	Ein Auftrag kann mit sämtlichen Datenfeldern geändert werden
A4	Das Login des Mitarbeiters wird nach drei nachfolgenden Falschanmeldungen automatisch gesperrt
A5	Personalisierte Auftragsliste des eingeloggtten Mitarbeiters. Der Mitarbeiter kann sich zusätzlich zur gesamten Auftragsliste nur die von ihm übernommenen Aufträge ansehen.
A6	Eingeloggte Mitarbeiter können neue Benutzer (Login) eröffnen.
A7	Eingeloggte Mitarbeiter können ein gesperrtes Login zurücksetzen.
A8	Gelöscht Aufträge werden nicht aus der Datenbank entfernt, sondern nur als gelöscht markiert.

1.3. Was muss ich mich informieren

1.3.1. Standards

Alles Standards der Norm EN ISO 9241-110 Lesen und anwenden.

1.3.2. Binding

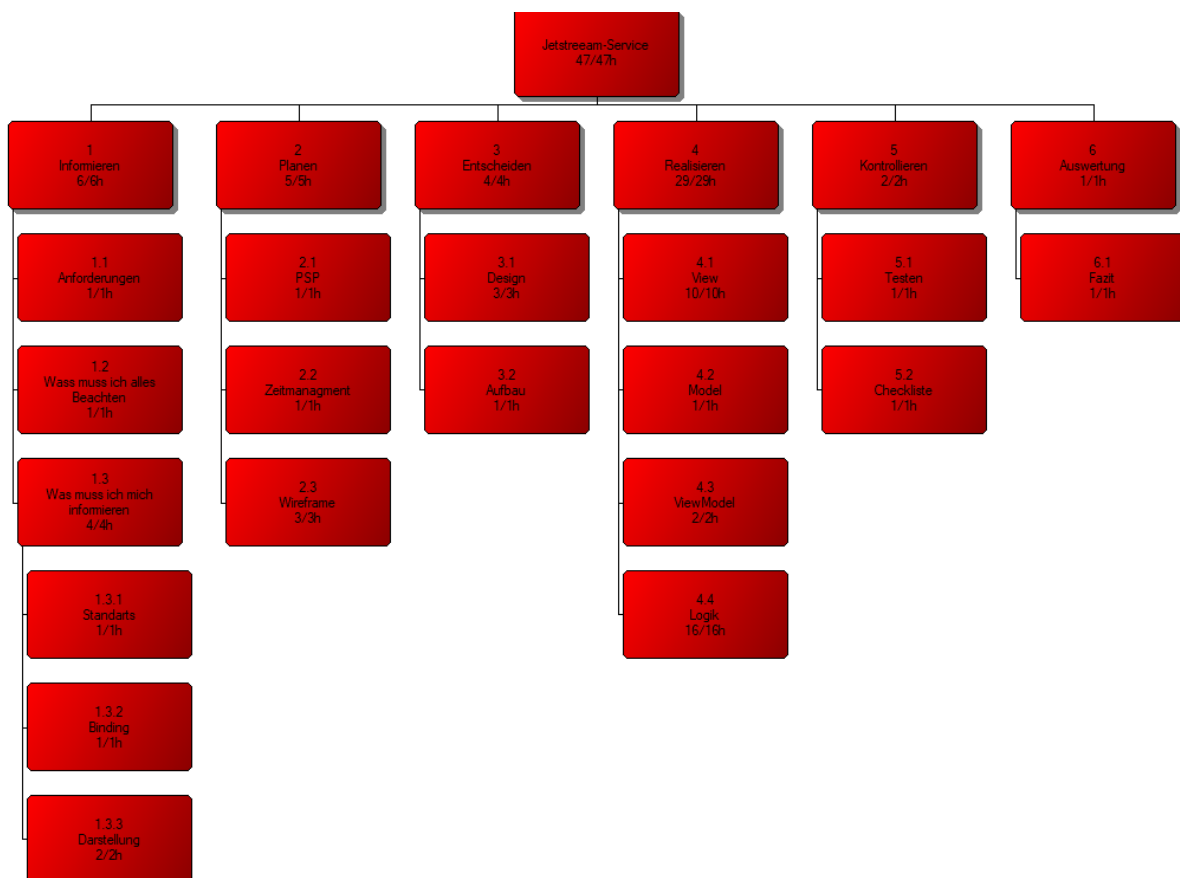
Command und Data Binding Verstehen, das haben wir in der Schule gelernt.
Kein Code Behind machen alles in Prinzip MVVM zu machen.

1.3.3. Darstellung

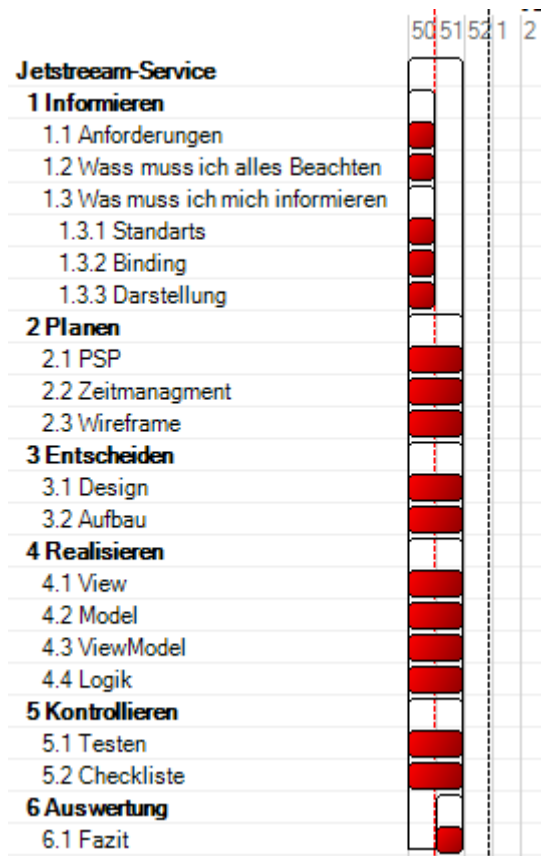
Für die Darstellung habe mehrere Programme angeschaut.
Auch mein Arzt hat mir sein Programm gezeigt, wie es dargestellt ist.

2. Planen

2.1. PSP



2.2. Zeitmanagement



2.3. Wireframe

Ich habe zwei Wireframes gestaltet Diese Sind unterschiedlich gestaltet

Ski-Service

Bitte Auswählen

ID/Status

Eingabe id/status

Achtung in der Schlüssellist ist nur 15min Gültig!

Einloggen

Benutzernamen

Passwort

Key

Eingabe

Name

Email

Phone

Erstellt

Priorität

Service

Status

Fertig

Ausgabe

Name (job title)	Age	Nickname	Employee
Giacomo Gulizzoni Founder & CEO	40	Peldi	<input type="radio"/>
Marco Botton Tuttofare	38		<input checked="" type="checkbox"/>
Mariah MacLachlan Better Half	41	Potato	<input type="checkbox"/>
Valerie Liberty Head Chef	39	Val	<input checked="" type="checkbox"/>
Data Grid Docs			<input type="checkbox"/>

Dieses Layout ist alles auf einer Seite kein anderes Fenster Wo aufgeht

Dieses Layout hat mehre neu Fenster wo Aufgehen.

JetStream-Service

Status Code Login

New Edit View Delete Refresh Suchen

Name	Email	Nickname	Employee
Giacomo Guizzoni Founder & CEO	40	Peldi	<input type="radio"/>
Marco Botton Tutorale	38		<input checked="" type="checkbox"/>
Mariah Maclochan Bettler Hall	41	Patata	<input checked="" type="checkbox"/>
Valerie Liberty Head Chef	3	Val	<input checked="" type="checkbox"/>
Data Grid Docs			<input type="checkbox"/>

Beenden

Anmeldung

Username

Password

Senden Abbrechen

New User

Name Priorität Bitte Auswählen

Email Service Bitte Auswählen

Phone Status Offen

Bemerkungen

Senden Abbrechen

Edit

ID

Name Priorität Bitte Auswählen

Email Service Bitte Auswählen

Phone Status Bitte Auswählen

Erstellt Fertig

Bemerkungen

Senden Abbrechen

Detail

ID

Name Priorität

Email Service

Phone Status

Erstellt Fertig

Bemerkungen

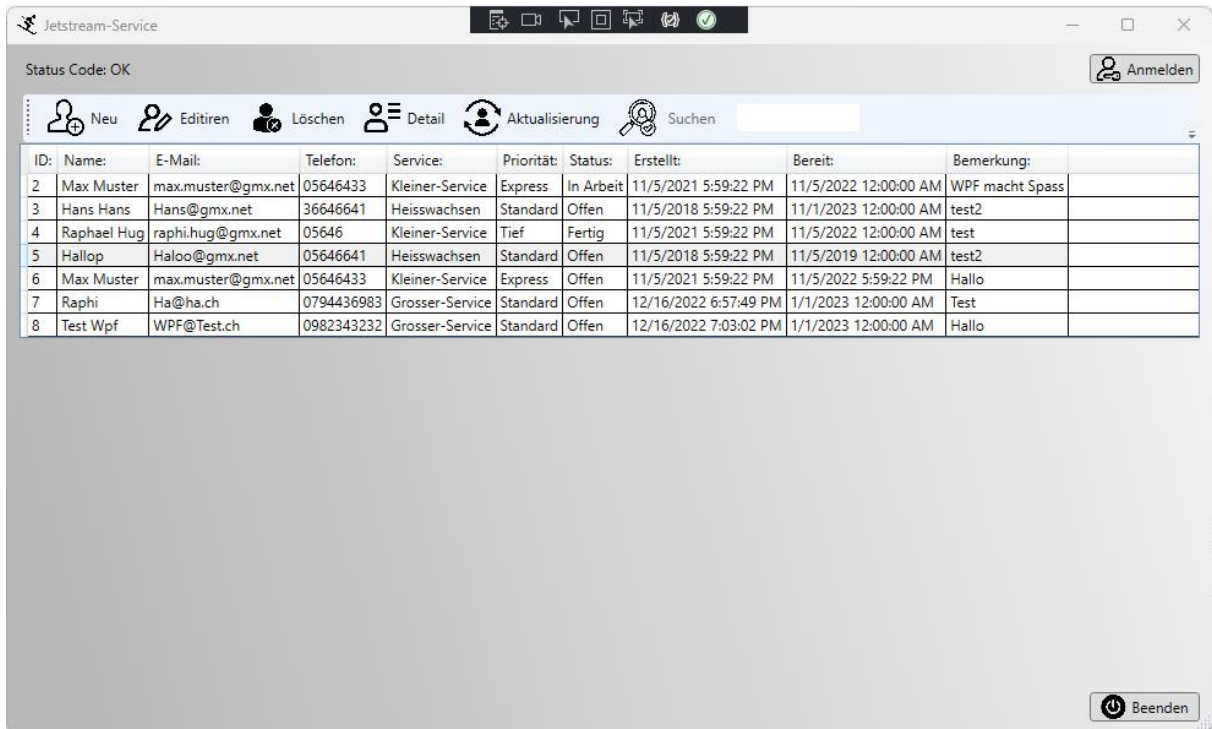
Senden Abbrechen

3. Entscheiden

3.1. Design

Ich habe mich für das Design mit mehreren Fenstern entschieden habe viele Leute gefragt und alle sagten dieses Design ist am klarer Strukturiert.

Mit WPF



3.2. Aufbau

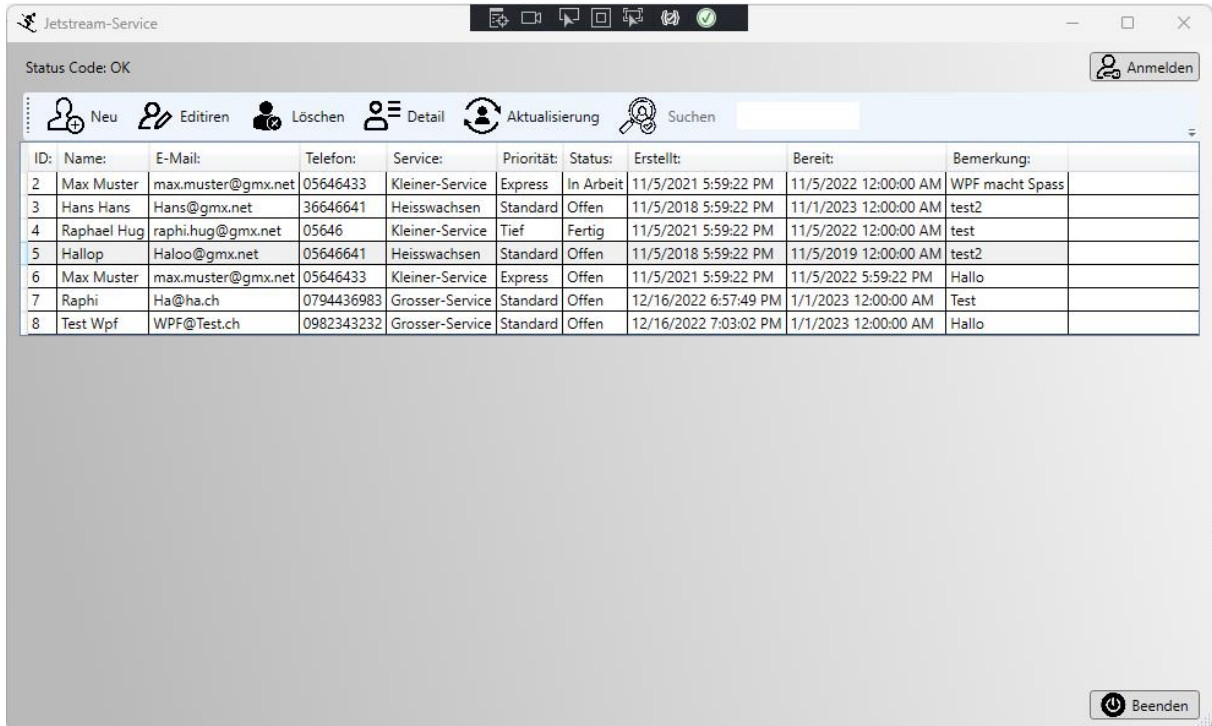
Also ich dachte mir zuerst das Design festzulegen.

Danach kann ich des Models, um Data Binding zu erstellen, aber danach brauchte ich noch das ViewModel und dann kam die Logik.

4. Realisieren

4.1. View

Ich habe zuerst alle Designs erstellt und alles vorbereitet fürs Data Binding
Zuerst habe das MainWindow erstellt dann habe ich den New User, Edit User, Login und Detail User erstellt und die Data/Command Binding Vordefiniert. Mit Allen Icons und Design erstellen hat ziemlich viel zeit gebraucht.



Die Probleme waren die Icons sind in einem Ordner drin.

Aber ich habe den Pfad zum Ordner eingefügt hat es nicht gefunden. Also habe ich beim Probierten alle Icons eingestellt, aber dann hat es die Bilder kopiert in Views das hat mich Gestört.

4.2. Models

Die Models erstellen war leicht und war ganz einfach habe RegistrationsModel wo die Kunden daten Bindet.

Im Authentification habe ich das Login Bindet.

Danach habe ich noch Content dort sind die Status Meldungen Bindet.

AuthenticationResponse wird der JWT-Token Bindet.

4.3. ViewModel

Zuerst habe ViewModelBase erstellt die Klasse vererbt von einem Interface INotifyPropertyChanged.

Danach habe ich MainViewModel Erstellt diese ist vererbt von ViewModelBase.

Dort habe zuerst alle Command Binding gemacht Diese gibt es CanExecute und Execute.

Ich habe alles mal beim CanExecute auf true gesetzt. Und Beim Execute einfach die Show eingebaut
Ohne Grosse Logik

Beim Data Binding habe ich eine ObservableCollection eingabt, dass alle daten ausgelesen werden kann.

Danach habe ich NewViewModel und LoginViewModel eingebaut und habe das gleiche Prinzip gemacht.

4.4. Logik

Ich habe Zuerst ein Settings File gemacht und dort alles eingestellt mit dem URL und API link

Dort Habe ich beim Alle Get, Post, Delete, Update eingebaut.

Get Update und Delete habe ich im MainViewModel gemacht.

Get Habe ich auch beim Login gemacht, aber diese hatte es ein anderen API link.

Beim NewViewModel habe ich ein Post eingebaut.

5. Kontrollieren

5.1. Testen

Ich habe alle Funktionen getestet und es geht alles.

Ich habe New getestet ein neuer Kunde erstellt.

Ändern von Bestehenden Kunden hat funktioniert

Detail hat geklappt man hat alle Infos in einem Fenster."

Löschen hat funktioniert

Aktualisieren und Login hat funktioniert.

5.2. Checkliste

A1	Login Dialog mit Passwort für den autorisierten Zugang der Mitarbeiter	x
A2	Erfasste Serviceaufträge müssen übersichtlich angezeigt sein (Listendarstellung)	x
A3	Die Listenanzeige der Serviceaufträge ist nach Priorität zu sortieren	x
A4	Listeneinträge können gefiltert werden (Suche von Einträgen)	x
A5	Mitarbeiter können eine Statusänderung eines Auftrages vornehmen	x
A6	Die .NET-Anwendung muss nach dem MVVM (Model, View, ViewModel) Design Pattern strukturiert sein.	x
A7	Datenaustausch per Web-API, REST-Service (PHP/ASP.NET)	x
A8	Intuitive u. Aufgabenangemessene Bedienung	x
A9	Grundsätze der Dialoggestaltung aus der Norm EN ISO 9241-110 berücksichtig	x
A10	Ganzes Projektmanagement nach IPERKA dokumentiert	x

6. Auswertung

6.1. Fazit

Ich war leider Krank beim Starten vom Projekt und konnte nichts machen lange Zeit und aber wo ich einigermaßen wieder gesund war habe mich Pressiert das zu erliegen.

Ist sehr spannend Mit dem Binden einfach die Infos über MVVM hat mich weiter gebracht.