



---

# SKI-SERVICE.NET APPLIKATION

---

Für die Jetstream-Service AG

Dokumentation zur Projektarbeit  
ICT Modul 322

Abgabedatum: Basel, 06.01.2024

**Prüfungsbewerber:**

Bobby Bilali, Mahir Gönen



06.01.2024

# Inhalt

1	Versionsverzeichnis .....	3
2	Executive Summary .....	4
3	Ausgangslage .....	5
4	Anforderungen .....	6
5	Zeitplanung .....	7
6	Informieren .....	7
6.1	Einleitung .....	7
6.2	Vorgabe/Anforderungen .....	8
6.3	Zusätzliche Anforderung .....	8
6.3.1	Kommentarfunktion für Aufträge .....	8
6.3.2	Flexibilität bei der Auftragsbearbeitung .....	8
6.3.3	Sicherheitsmechanismus beim Login .....	8
6.3.4	Umgang mit gelöschten Aufträgen .....	8
7	Planen .....	9
7.1	Entwurf eines Anwendungsdesign .....	9
7.2	Organisation .....	9
7.2.1	Technologieauswahl .....	9
7.2.2	GitHub .....	10
7.2.3	Trello-Board .....	10
7.3	Planung des Mockups .....	11
7.4	Entwurf für das Frontend Adminpanel .....	11
8	Entscheiden .....	13
8.1	Backendentscheidung .....	13
8.2	Mockups vergleichen und zusammenführen .....	13
8.3	WPF oder .NET MAUI .....	13
9	Realisieren .....	14
9.1	WPF-Entwicklung im MVVM-Pattern .....	14
9.2	Listendarstellung-Entwicklung .....	14
9.3	Backend-Anbindung .....	15
10	Kontrollieren .....	16
10.1	Testplan .....	16
10.2	Testprojekt implementieren .....	17
11	Auswerten .....	18
12	Fazit .....	19
12.1	Persönliches Fazit - Bobby Bilali .....	19
12.2	Persönliches Fazit – Mahir Gönen .....	19

Anhänge.....	20
I.    Glossar .....	20
II.   Weitere Dokumente .....	21
III.  Versionsverzeichnis der Anwendung/Pakete .....	22
13    Quellenverzeichnis .....	23
14    Tabellenverzeichnis.....	23
15    Abbildungsverzeichnis .....	23

## 1 Versionsverzeichnis

Version	Autor	Datum	Änderung
<b>1.0</b>	Bobby Bilali	28.12.2023	Erstellung des Dokuments
<b>1.1</b>	Mahir Gönen	29.12.2023	Ergänzung des Dokuments
<b>1.2</b>	Bobby Bilali	29.12.2023	Ergänzung des Dokuments
<b>1.3</b>	Mahir Gönen	30.12.2023	Rechtschreibung und Grammatik Kontrolle
<b>1.4</b>	Bobby Bilali	04.01.2024	Fertigstellung des Dokuments

## 2 Executive Summary

In diesem Projekt wird eine Anwendung entwickelt, um den Prozess der Ski-Service Verwaltung zu verbessern. Die "Ski-Service.NET App" zielt darauf ab, eine nahtlose Integration von Benutzerinteraktion und Systemfunktionalität zu bieten, um eine optimale Nutzererfahrung für Ski-Service Mitarbeiter zu schaffen. Die Anwendung ist für die Nutzung in Werkstätten ausgelegt, mit besonderem Augenmerk auf eine einfache Bedienung gelegt.

Die Implementierung folgt strengen Designprinzipien und technischen Richtlinien, wobei das MVVM-Design Pattern und moderne Web-API REST-Services zum Einsatz kommen. Diese Technologien gewährleisten nicht nur eine hohe Benutzerfreundlichkeit, sondern auch eine robuste und sichere Plattform für die Verwaltung und Überwachung von Ski-Service Aufträgen.

Zu den Hauptfunktionen der App gehören Benutzeranmeldung, Anzeige und Filterung von Aufträgen sowie die Aktualisierung von Auftragsstatus, alles mit einer effizient gestalteten Schnittstelle.

Das Projekt wird im Rahmen der IPERKA-Methodik durchgeführt, was eine systematische und phasenorientierte Entwicklung ermöglicht. Dieser Ansatz hilft, die Anforderungen und Bedürfnisse der Nutzer in jeder Phase des Projekts zu verstehen und umzusetzen. Die Entwicklung der "Ski-Service.NET App" stellt einen bedeutenden Schritt in der Digitalisierung von Ski-Service Prozessen dar und zielt darauf ab, die Effizienz und Mitarbeiterzufriedenheit in diesem Bereich erheblich zu steigern.

Durch die detaillierte Berücksichtigung von Benutzerinteraktion und Schnittstellengestaltung bietet dieses Projekt einen umfassenden Ansatz zur Modernisierung des Ski-Service-Managements und setzt neue Standards in der Branche.

### 3 Ausgangslage

Das Projekt "Ski-Service.NET App" hat seinen Ursprung in einem vorangegangenen Modul, in dem eine WebAPI mit einem dazugehörigen Webclient entwickelt wurde. Diese diente als Management-Tool für Ski-Service Prozesse. Um die Anwendung robuster und zugänglicher für die Mitarbeiter zu machen, wurde mithilfe .NET WPF entwickelt. Dieser Schritt berücksichtigte sowohl die ISO-Normen als auch die Best Practices in Softwareentwicklung und Design.

In der Anfangsphase lag der Fokus auf der Erstellung von Mockups. Diese Entwurfsphase war entscheidend, um unterschiedliche Konzepte und Benutzeroberflächen zu erkunden. Ziel war es, durch Iteration und Feedback mehrere Entwurfsoptionen zu entwickeln und daraus ein finales, optimiertes Mockup auszuwählen. Dieses finale Design diente als Blaupause für die weitere Entwicklung in .NET WPF, wobei besonderes Augenmerk auf Benutzerfreundlichkeit und Einhaltung der Normen gelegt wurde.

Die Entwicklung des finalen Mockups war ein iterativer Prozess, der eng mit den spezifischen Anforderungen des Ski-Service-Managements verknüpft war. Durch die Anwendung von .NET WPF konnten wir eine Anwendung erstellen, die nicht nur technisch robust ist, sondern auch eine intuitive und effiziente Schnittstelle für die Nutzer bietet.

## 4 Anforderungen

Das Projekt "Ski-Service.NET App" stellt spezifische Anforderungen an das Backend-System, um eine effiziente und sichere Verwaltung der Skiservice-Aufträge zu gewährleisten. Diese beinhalten:

- **Login-Funktionalität:** Implementierung eines sicheren Login-Dialogs mit Passwort für autorisierten Mitarbeiterzugang. Dies gewährleistet, dass nur befugte Personen Datenänderungen vornehmen können.
- **Listendarstellung:** Die Serviceaufträge sollten in einer übersichtlichen Listendarstellung dargestellt werden.
- **Filtern der Listeneinträge:** Es sollte möglich sein, dass man als Mitarbeiter durch ein Suchfeld nach Eigenschaften suchen kann.
- **Serviceaufträge Management:** Ein systematisches Management der Serviceaufträge, das es Mitarbeitern ermöglicht, Aufträge zu erfassen, ihren Status zu ändern (Offen, In Arbeit, Abgeschlossen) und sie bei Bedarf zu löschen.
- **EN ISO 9241-110 berücksichtigen:** Die Dialoggestaltung muss nach der Normierung bestimmt und gestaltet werden.
- **MVVM Design Pattern:** Die Projektstruktur muss nach dem MVVM Design Pattern strukturiert sein.
- **Optionale Erweiterungen:** Zusatzfunktionen wie Kommentarfunktion für Aufträge, umfassende Auftragsmodifikationen und ein automatisches Sperrsystem für das Login nach wiederholten Fehlanmeldungen.

Diese Anforderungen sollen sicherstellen, dass die "Ski-Service.NET App" nicht nur funktional und benutzerfreundlich ist, sondern auch höchsten Sicherheits- und Qualitätsstandards entspricht.

## 5 Zeitplanung

Das Projekt wird in mehreren Phasen durchgeführt und es wurden folgender Zeitplan festgelegt (totale Stunden von beiden Entwicklern):

Projektphase	Geplante Zeit
Informieren	3h
Planung, Entwurf, Entscheidung	15h
Realisierung	40h
Abnahme	4h
Dokumentation	10h
Gesamt	72h

*Tabelle 1: Grobe Planung*

Dieses Projekt findet innerhalb von knapp zwei Wochen statt.

## 6 Informieren

In dieser Phase des IPERKA-Modells liegt der Schwerpunkt auf der gründlichen Informationsbeschaffung. Ziel ist es, ein detailliertes Verständnis für die spezifischen Bedürfnisse, Anforderungen und Herausforderungen bei der Entwicklung des Windows Presentation Forms für Jetstream-Service zu erlangen. Diese Phase beinhaltet die Analyse für das Erkennen der Bedürfnisse der Nutzer und Mitarbeiter, das Verständnis für die technischen Anforderungen sowie die Berücksichtigung aller weiteren Faktoren, die für den erfolgreichen Abschluss der Applikation entscheidend sind.

### 6.1 Einleitung

In der Einleitung für das Projekt "Ski-Service.NET App" wird der Kontext und die Relevanz des Vorhabens erläutert. Hier wird aufgezeigt, wie die Einführung der bei Jetstream-Service die Entwicklung einer angepassten Benutzeroberfläche notwendig macht. Das Hauptziel ist die Steigerung der Effizienz in Werkstatt und Administration durch eine benutzerfreundliche und intuitive App. Die Implementierung des Web-API REST-Services ist ein weiterer Kernaspekt. Die Einleitung betont die Wichtigkeit der Modernisierung und Digitalisierung für das Unternehmen und stellt den Bezug zur IPERKA-Methodik her, um die strukturierte und zielorientierte Vorgehensweise des Projekts zu unterstreichen.



## 6.2 Vorgabe/Anforderungen

Die Hauptziele des Ski-Service.NET-Projekts umfassen die Entwicklung und Implementierung von Schlüsselementen, die für ein effizientes und sicheres Auftragsmanagement erforderlich sind:

- Login-Funktionalität: Ein sicherer Login-Dialog mit Passwort ist erforderlich, um den Mitarbeitern autorisierten Zugang zu den Daten zu gewähren.
- Mitarbeiter können Serviceaufträge einsehen, den Status ändern (Offen, In Arbeit, Abgeschlossen) und bei Bedarf Aufträge löschen (z.B. Stornierungen).
- Sortierungsfunktionen

Die Dokumentation folgt dem IPERKA-Modell und ist in entsprechende Abschnitte unterteilt, die die Informationsbeschaffung, Planung, Entscheidung, Realisierung, Kontrolle und Auswertung des Projekts abdecken. Zudem enthält sie ein Versionsverzeichnis, um die Aktualität und Relevanz der bereitgestellten Informationen sicherzustellen.

## 6.3 Zusätzliche Anforderung

Im Laufe der Informationsphase wurden neben den grundlegenden Anforderungen weitere spezifische Bedürfnisse und Wünsche evaluiert, die das Projekt in einer optimierten Form prägen könnten.

### 6.3.1 Kommentarfunktion für Aufträge

Um die Kommunikation und Dokumentation zu verbessern, soll die Möglichkeit für Mitarbeiter geschaffen werden, zu jedem Auftrag individuelle Kommentare oder Notizen hinzuzufügen. Diese Funktion erleichtert den Informationsaustausch und bietet eine Plattform für interne Anmerkungen und Details, die für die Bearbeitung des Auftrags relevant sind.

### 6.3.2 Flexibilität bei der Auftragsbearbeitung

Ein wichtiger Aspekt ist die Fähigkeit, sämtliche Datenfelder eines Auftrags bei Bedarf ändern zu können. Diese Flexibilität ermöglicht es den Mitarbeitern, auf Änderungen in den Kundenanforderungen oder interne Updates reagieren zu können, was die Effizienz und Kundenzufriedenheit erhöht.

### 6.3.3 Sicherheitsmechanismus beim Login

Zur Erhöhung der Sicherheit wird ein Mechanismus implementiert, der das Login eines Mitarbeiters nach drei aufeinanderfolgenden Fehlanmeldungen automatisch sperrt. Diese Funktion dient dem Schutz sensibler Daten und verringert das Risiko unautorisierten Zugriffs.

### 6.3.4 Umgang mit gelöschten Aufträgen

Anstatt gelöschte Aufträge komplett aus der Datenbank zu entfernen, werden diese lediglich als gelöscht markiert. Dieser Ansatz sorgt dafür, dass alle Daten für zukünftige Analysen und Berichte erhalten bleiben, während die aktuelle Datenansicht für die Mitarbeiter übersichtlich und relevant bleibt.

## 7 Planen

Die Planungsphase stellt einen entscheidenden Abschnitt dar. Nachdem in der Informationsphase alle erforderlichen Daten gesammelt und ein klares Bild der Anforderungen und Rahmenbedingungen geschaffen wurde, gilt es nun, diese Erkenntnisse in einen strukturierten und realisierbaren Plan zu übertragen.

### 7.1 Entwurf eines Anwendungsdesign

Im Rahmen der Entwicklung für die Firma Jetstreamski-Service liegt ein besonderer Fokus auf die Anbindung der Applikation zum Backend. Dazu wurde ein Anwendungsdesign erstellt, um die Backendverbindung einfacher darzustellen.

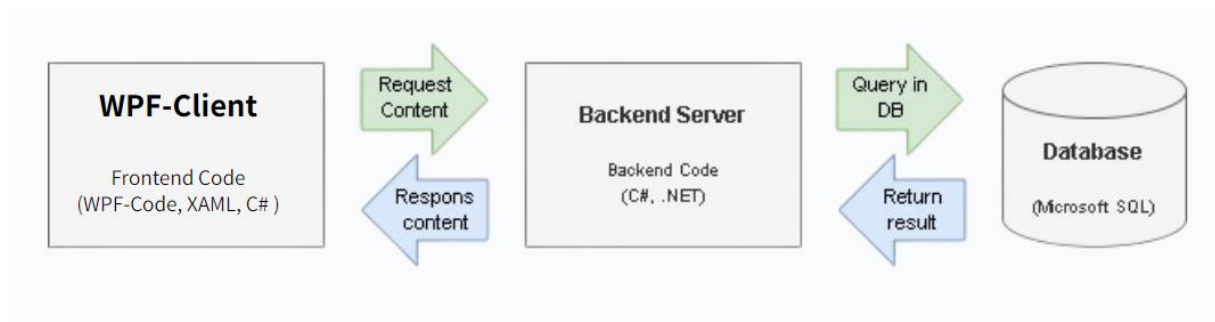


Abbildung 1 Anwendungsdesign

### 7.2 Organisation

Um die Architektur und das Zusammenspiel der Komponenten des Backend-Systems optimal zu gestalten, wurde eine sorgfältige Auswahl an Technologien und Methoden getroffen.

#### 7.2.1 Technologieauswahl

Für das Projekt "Ski-Service.NET App" wurde eine Reihe spezifischer Technologien und Bibliotheken ausgewählt, um eine effiziente und benutzerfreundliche Anwendung zu entwickeln. Der Einsatz von .NET 8.0 WPF ermöglicht den Zugriff auf fortgeschrittene UI-Komponenten wie Dockpanel, TreeView und DataGrids, die für die Gestaltung einer intuitiven Benutzeroberfläche unerlässlich sind. Die Nutzung von NuGet-Paketen wie Fluent.Ribbon unterstützt das Design eines Ribbon-Layouts, das Benutzern vertraut ist und Ähnlichkeiten zu Microsoft Word aufweist. Gu.Wpf.Adorners vereinfacht die Integration von Placeholdern und Watermarks. Zudem wurden Microsoft.Extensions.Http für die Handhabung von HTTP-Anfragen und weitere Extensions genutzt, um eine reibungslose und effektive Kommunikation mit dem Backend-Service zu gewährleisten. Diese Technologieauswahl trägt entscheidend zur Realisierung einer modernen, ansprechenden und leistungsfähigen Anwendung bei.

#### 7.2.1.1 Anwendungen und NuGet-Pakete

Alle verwendeten Anwendungen und NuGet-Pakete werden hier aufgelistet und beschrieben. Um Komplikationen zu verhindern sind die Versionierungen und vollständigen Beschreibung auf [hier](#) dokumentiert.

Die WebAPI läuft unter dem Framework WPF.NET8.0

Folgende NuGet-Pakete werden benötigt für das Backend:

- Fluent.Ribbon
- Gu.Wpf.Adorners
- Microsoft.Extensions.Configuration
- Microsoft.Extensions.Configuration.Binder
- Microsoft.Extensions.Configuration.FileExtensions
- Microsoft.Extensions.Configuration.Json
- Microsoft.DependencyInjection
- Microsoft.Extensions.Http
- Newtonsoft.Json
- RestSharp

Da, dass Testprojekt ein eigenständiges xUnit-Projekt ist kommen folgende NuGet-Pakete für das Testprojekt dazu:

- Moq
- Xunit
- Microsoft.Extensions.Configuration.Abstractions

#### 7.2.2 GitHub

Mithilfe von GitHub wurde das komplette Projekt durch Versionsverläufe dokumentiert. Durch ein Git-Repository kann der Entwickler in der Realisierungsphase die komplette Programmierarbeit einfacher und unabhängig des Standortes bewältigen. Durch die Commits, die der Entwickler bei allen Änderungen macht, bringt dies eine grosse Transparenz in das Projekt zwischen dem Projektteam und Kunden.

#### 7.2.3 Trello-Board

Durch den Einsatz von einem Trello-Board wird die Organisation zwischen den Entwicklern weiter gefördert um allen Anforderungen, die in dieser Phase bestimmt werden, gerecht zu werden. Es gibt verschiedene Abteilungen, die farblich gekennzeichnet werden.

## 7.3 Planung des Mockups

Die Planung des Mockups für die "Ski-Service.NET App" ist ein entscheidender Schritt im Designprozess. In dieser Phase werden die visuellen und funktionellen Aspekte der Benutzeroberfläche konkretisiert. Es geht darum, ein vorläufiges Design zu erstellen, das die grundlegenden Funktionen und das Layout der Anwendung darstellt. Wichtig ist dabei, die Nutzererfahrung zu berücksichtigen und sicherzustellen, dass das Mockup intuitiv und benutzerfreundlich ist. Die Planung beinhaltet die Auswahl Icons und die Anordnung der Benutzeroberflächenelemente. Es wurde geplant, dass von beiden aktiven Entwicklern jeweils ein Mockup erstellt werden sollte und daraus ein finales entsteht.

## 7.4 Entwurf für das Frontend Adminpanel

Da, dass Projekt sich auf das Frontend und Benutzerschnittstellen konzentriert, wurde ein sehr ausführliches Mockup erstellt. Diese ist auch im PDF im GitHub erhalten und ist klickbar.

Um ein kleiner Einblick zu geben, wie das Mockup aussieht wird im Dokument einige Screenshots eingefügt. Es war uns wichtig, dass im Dashboard nur essenziell wichtige Eigenschaften eines Auftrages steht. Somit wurde ein Master-Detailansicht erstellt.



Abbildung 3 Login-Dialog

	Auftrags-Nr.	Nachname	Vorname	Service	Enddatum	Status	Priorität
Alle Aufträge	100001	Schmidt	Lisa	Kleiner Service	25.12.2023	Offen	Tief
	100002	Wagner	Michael	Rennski Service	26.12.2023	Offen	Standard
Offene Aufträge	100003	Müller	Sabine	Grosser Service	27.12.2023	Offen	Express
Kleiner Service	100004	Becker	Andreas	Fell zuschneiden	25.12.2023	Offen	Tief
Grosser Service	100005	Weber	Julia	Bindungen montieren und einstellen	28.12.2023	Offen	Standard
Rennski Service	100006	Fischer	Markus	Kleiner Service	26.12.2023	Offen	Standard
Bindungen montieren und einstellen	100007	Keller	Petra	Heisswachsen	27.12.2023	In Arbeit	Standard
Fell zuschneiden	100008	Braun	Thomas	Kleiner Service	25.12.2023	Abgeschlossen	Tief
Heisswachsen	100009	Meier	Sandra	Rennski Service	28.12.2023	Offen	Standard
In Arbeit	100010	Lehmann	Marco	Bindungen montieren und einstellen	26.12.2023	Abgeschlossen	Tief
Kleiner Service							
Grosser Service							
Rennski Service							
Bindungen montieren und einstellen							
Fell zuschneiden							
Heisswachsen							
abgeschlossene Aufträge							

Abbildung 2 Dashboard Listeneinträge

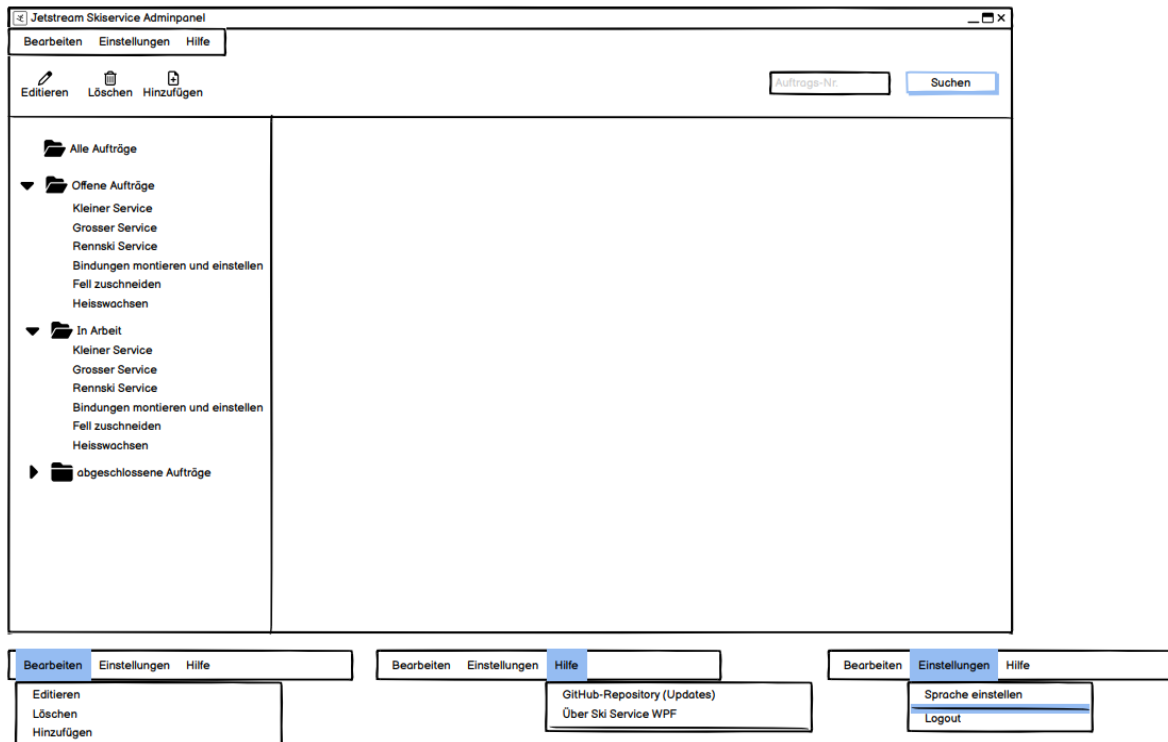


Abbildung 5 Dashboard mit Menüband v2.0

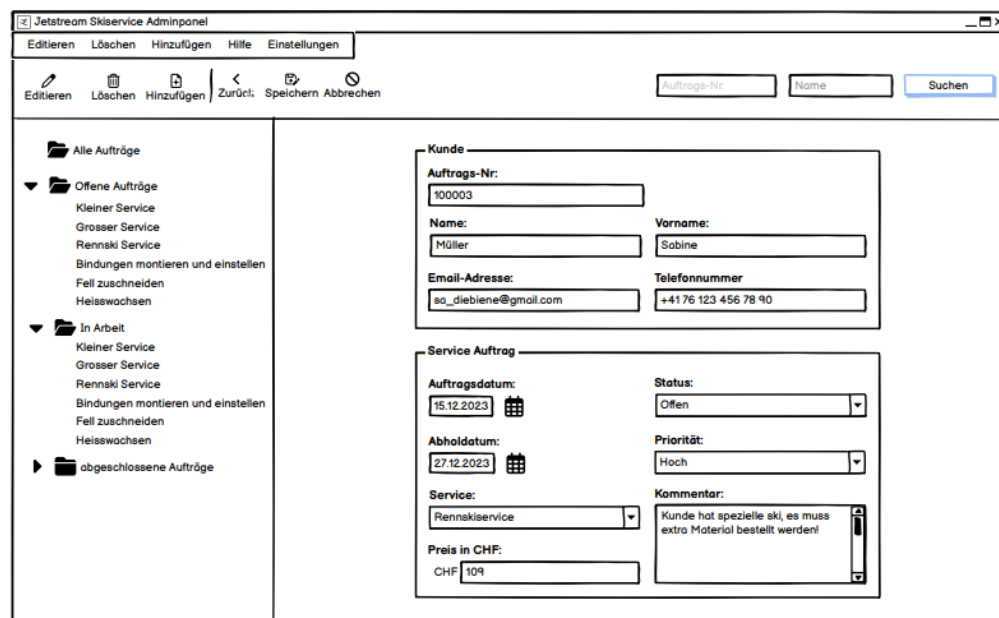


Abbildung 4 Dashboard "Editieren"-Ansicht

## 8 Entscheiden

### 8.1 Backendentscheidung

Die Entscheidung für ein gemeinsames Backend im Projekt "Ski-Service.NET App" ist ein wichtiger Schritt zur Gewährleistung von Konsistenz und Effizienz in der Entwicklung. Im letzten Projekt entwickelte jeder Entwickler eigenständige Lösungen, was zu Inkonsistenzen und Integrationsproblemen führen konnte. Durch die Einigung auf ein einheitliches Backend wird eine gemeinsame Basis für alle Entwicklungsarbeiten geschaffen. Dies erleichtert die Zusammenarbeit, fördert die Wiederverwendbarkeit von Code und verringert das Risiko von Kompatibilitätsproblemen. Zudem ermöglicht es eine effizientere Nutzung von Ressourcen und eine strukturierte Herangehensweise, was für die Umsetzung eines kohärenten und leistungsfähigen Systems unerlässlich ist.

### 8.2 Mockups vergleichen und zusammenführen

In diesem Schritt des Projekts "Ski-Service.NET App" werden die verschiedenen Mockups der Entwickler verglichen und zusammengeführt. Da individuelle Ansichten über Benutzerfreundlichkeit variieren, erstellt jeder Entwickler ein eigenes Mockup. Anschliessend erfolgt ein Austausch von Feedback, um die Stärken und Schwächen jedes Designs zu identifizieren. Ziel ist es, ein finales Mockup zu entwickeln, das die besten Elemente aus jedem Vorschlag integriert und somit ein Höchstmass an Benutzerfreundlichkeit gewährleistet. Dieser Prozess fördert die Zusammenarbeit und ermöglicht es, ein umfassendes und gut durchdachtes Design für die Anwendung zu kreieren.

### 8.3 WPF oder .NET MAUI

Bei der Wahl zwischen WPF und MAUI für das Projekt entschied man sich für WPF. Diese Entscheidung beruhte auf den vorhandenen Erfahrungen der Entwickler mit WPF sowie den umfangreichen UI-Elementen, die WPF bietet. Im Vergleich zu MAUI bietet WPF fortgeschrittene Elemente wie ListView, DataGrid und Ribbon, die für das Projekt notwendig waren und in MAUI nur mit grösserem Aufwand umsetzbar wären. Das Mockup, das sich stark auf diese Elemente stützte, spielte ebenfalls eine entscheidende Rolle bei der Entscheidung für WPF.

## 9 Realisieren

Dieser Abschnitt dokumentiert die Umsetzungsphase des Projekts, in der die geplanten Konzepte und Entscheidungen in die Praxis umgesetzt werden.

### 9.1 WPF-Entwicklung im MVVM-Pattern

Bei der Realisierung der "Ski-Service.NET App" unter Anwendung des MVVM-Prinzips wurde eine Herausforderung deutlich:

Die Balance zwischen der Einhaltung des Prinzips und der praktischen Umsetzbarkeit. Die initiale Strukturierung folgte einer standardisierten MVVM-Ordnerstruktur, um Klarheit und Organisation zu gewährleisten. Die Integration verschiedener User Controls wie `DashboardViewUserControl`, `EditViewUserControl`, `ListViewUserControl` und `LoginView` ins `MainWindow`, mit Ausnahme des `DeleteWindows`, und die Steuerung durch Dependency Injection in `App.xaml.cs`, stellten den Kern des Designs dar. Trotz der sorgfältigen Planung führte die Unerfahrenheit im Umgang mit MVVM zu Schwierigkeiten, besonders bei der Umsetzung komplexer Bindungen und der Auslagerung von Operationen. Einige Elemente, wie das `LoginView`, hätten möglicherweise effektiver als eigenständige Fenster realisiert werden können, um die Komplexität zu reduzieren und die Einhaltung des MVVM-Prinzips zu erleichtern.

Aufgrund Komplexität und Zeitgründen wurden einige Funktionen, die im Mockup geplant waren im späteren Stadium des Projekts gekürzt, um trotzdem die Kernfunktionen korrekt zu implementieren. Das Hauptproblem war es, dass die verschiedenen Ansichten in der Anwendung komplexer und grösser wurden, die im Anfangsstadium nicht unbedingt ersichtlich waren. Es wurde entschieden, dass das Menü Band trotzdem gelassen wird mit den Tabs «Bearbeiten», «Einstellung» und «Hilfe» und deren Untermenüs, weil es zum Teil der Benutzerfreundlichkeit gehört. Es gibt aber noch keine tatsächlichen Funktionen für das Menü Band.

Das `TreeView` wurde gekürzt, weil eine komplette Sortierung nach Services schon komplett in der Liste implementiert wurde.

Es wurde auch Sprachressourcen geplant, die aufgrund zeitlicher Rahmen nicht implementiert wurde auch das «Hinzufügen» Button wurde nicht implementiert, weil es eine eigenständige Ansicht wäre, die wieder komplexe Bindungen beinhaltet.

### 9.2 Listendarstellung-Entwicklung

Bei der Entwicklung der Listendarstellung für die "Ski-Service.NET App" wurde ein `DataGrid` genutzt, um die Anforderung einer Listenansicht zu erfüllen. Inspiriert von bekannten Microsoft Produkten, wurde ein sortierbarer Header implementiert, entsprechend den im Mockup festgelegten Prinzipien. Die Herausforderung lag in der Nutzung von XAML `UserControl.Resources` und `Templates`, um Pfeile und den Header zu stylen. Obwohl angestrebt, konnte die Implementierung eines standardmässig angezeigten Pfeils beim Enddatum nicht erfolgreich umgesetzt werden, was die Komplexität dieser Aufgabe unterstreicht.

### 9.3 Backend-Anbindung

Für die Backend-Anbindung der "Ski-Service.NET App" wurde eine Kombination aus Microsoft Extensions und Konfigurationen in der appsettings.json-Datei genutzt. Die appsettings.json enthielt die notwendigen HTTP-Routen, und für jede Operation im BackendService wurden entsprechende Verbindungen eingerichtet. Diese Methode ermöglichte eine effiziente und strukturierte Herangehensweise, um alle erforderlichen Backend-Operationen zu integrieren und zu managen, was für das reibungslose Funktionieren der App entscheidend ist.



## 10 Kontrollieren

Die Kontrollieren Phase ist entscheidend, um sicherzustellen, dass das entwickelte System den Anforderungen und Erwartungen entspricht. Ein wichtiger Bestandteil dieser Phase ist die Implementierung eines Testprojekts.

### 10.1 Testplan

Dieser Testplan ist für das Testprojekt entwickelt worden.

#### **1. Testziele**

Sicherstellen, dass alle Funktionalitäten korrekt umgesetzt wurden.

Überprüfung der Integration zwischen Controllern, Services und Datenmodellen.

#### **2. Testumfang**

Tests für EmployeeController, PriorityController, RegistrationController und StatusController.

#### **3. Teststrategie und -ansatz**

Unit-Tests für individuelle Funktionen in jedem Controller.

Mocking von Services und Datenmodellen mit Moq.

Verwendung von FluentAssertions für Testergebnisse.

#### **4. Testumgebung**

Die Tests werden in einer standardisierten Entwicklungsumgebung unter .NET durchgeführt.

#### **5. Testdaten**

Erstellung von Testdaten für Mitarbeiter, Prioritäten, Registrierungen und Status.

#### **6. Testfälle und -szenarien**

Jede Testdatei (EmployeeControllerTests.cs, etc.) enthält spezifische Testfälle für die entsprechenden Controller-Funktionen.

#### **7. Risikomanagement**

Identifikation potenzieller Risiken bei der Testdurchführung und Entwicklung von Strategien zur Risikominderung.

#### **8. Kriterien für den Abschluss der Tests**

Alle Testfälle müssen erfolgreich durchlaufen werden.

Keine kritischen Fehler dürfen in den Systemkomponenten verbleiben.

## 10.2 Testprojekt implementieren

Für die Ski-Service.NET wurde ein umfassendes Testprojekt implementiert, das auf den festgelegten Testplan abgestimmt ist. Die Implementierung des Testprojekts umfasste folgende Schlüsselemente:

**Testziele:** Hauptziel war die Sicherstellung der korrekten Funktionsweise aller Knöpfe und Dialogverhalten. Die meisten Tests sind in visueller Ansicht und deshalb wurde es von den Entwicklern durchgeführt, da ein xUnit-Test solche Sachen nicht direkt testen kann.

**Testumfang:** Aufgrund zeitlicher Knappheit wurden nur zwei Unittest erstellt.

**Teststrategie und -ansatz:** Es wurde ein xUnit-Testprojekt erstellt, um HTTP-Requests und einen um Benutzerinteraktionen zu testen.

**Testumgebung:** Die Tests wurden in einer standardisierten .NET-Entwicklungsumgebung durchgeführt, um konsistente und zuverlässige Ergebnisse zu gewährleisten.

**Testdaten:** Testdaten wurden speziell für die Evaluierung verschiedener Funktionen der Anwendung generiert, darunter Benutzerinteraktionen und Datenverarbeitung.

Für die HTTP-Request-Tests wurden Mock-Daten verwendet, die repräsentative Szenarien und Antworten des Backends simulierten.

**Testfälle und -szenarien:** Die Testfälle umfassten die Überprüfung der korrekten Funktion von HTTP-Requests, einschliesslich GET- und POST-Requests.

Es wurden Szenarien getestet, die den Abruf von Registrierungsdaten, Statusinformationen und die Authentifizierung von Benutzern umfassten.

Spezifische Testszenarien beinhalteten die Überprüfung der korrekten Reaktion des Systems auf gültige und ungültige Anfragen, das Handling von Fehlern und die korrekte Datenverarbeitung.

**Risikomanagement:** Mögliche Risiken im Testprozess wurden identifiziert, und es wurden Strategien entwickelt, um diese Risiken zu minimieren. Es wurden kritische Fehler dadurch ausgeschlossen.

**Kriterien für den Abschluss der Tests:** Das Testprojekt wurde als erfolgreich abgeschlossen betrachtet, sobald alle Testfälle erfolgreich durchlaufen wurden und keine kritischen Fehler in den Systemkomponenten verblieben.

**Erfolgreiche Testfälle:** Alle HTTP-Request-Tests wurden erfolgreich durchgeführt, was die korrekte Funktion des Backendes und die angemessene Behandlung von Anfragen bestätigte.

Authentifizierungstests bestätigten die Sicherheit und Zuverlässigkeit des Login-Prozesses.

Tests für die Abfrage von Registrierungs- und Statusdaten verifizierten die korrekte Handhabung und Darstellung dieser Daten in der Anwendung.

Die Ergebnisse dieser Testfälle trugen dazu bei, das Vertrauen in die Funktionalität und Zuverlässigkeit der Anwendung zu stärken. Durch die erfolgreichen Tests wurde die Grundlage für eine stabile und benutzerfreundliche Anwendung gelegt [siehe hier](#).

Das Testprojekt hat sich als wichtig erwiesen, um die Zuverlässigkeit und Stabilität der Anwendung zu gewährleisten. Es trug entscheidend dazu bei, das Vertrauen in die Funktionalität der einzelnen Komponenten zu stärken und die Gesamtqualität des Systems zu sichern.

## 11 Auswerten

Die Auswertungsphase des Projekts "Ski-Service.NET App" ist entscheidend, um die Leistung, Benutzerfreundlichkeit und Effektivität der entwickelten Anwendung zu bewerten. Diese Phase umfasst die Analyse der Benutzerfeedbacks, die Überprüfung der Einhaltung der anfänglichen Anforderungen und die Bewertung der technischen Performance. Dabei werden sowohl qualitative als auch quantitative Daten berücksichtigt, um ein umfassendes Verständnis der Stärken und Verbesserungsbereiche der Anwendung zu erlangen. Diese Informationen sind essenziell, um gegebenenfalls Anpassungen vorzunehmen und die Qualität der Anwendung kontinuierlich zu verbessern.

Insgesamt bot die Auswertungsphase tiefe Einblicke in die Stärken und Schwächen des Projekts. Sie lieferte essenzielle Erkenntnisse, die als Basis für zukünftige Innovationen und Verbesserungen bei Jetstream-Service dienen. Diese Phase war somit ein entscheidender Schritt, um die Qualität und Effizienz des Projekts kontinuierlich zu steigern und die Weichen für zukünftige Erfolge zu stellen.

## 12 Fazit

### 12.1 Persönliches Fazit - Bobby Bilali

Das Ski-Service.NET-Projekt war eine umfassende Lernerfahrung, die mir die Komplexität der Softwareentwicklung vor Augen geführt hat. Besonders die Gestaltung eines ansprechenden GUIs und die gleichzeitige Verwendung von MVVM und Sicherstellung der Funktionalität mit Backend-Verbindungen wie Web-API und REST-Service stellten sich als anspruchsvoller heraus als zunächst angenommen.

Eine der wichtigsten Erkenntnisse für mich war die Bedeutung der frühen Einarbeitung in die Verwendung von Windows, Pages und UserControls in WPF. Dies hätte das Projekt erheblich effizienter und strukturierter gemacht. Leider kam diese Erkenntnis zu einem Zeitpunkt, als umfassende Änderungen nicht mehr möglich waren.

Die Teamarbeit und Koordination innerhalb des Projekts waren ebenfalls bereichernd. Sie unterstrichen die Wichtigkeit klarer Kommunikation und strukturierter Zusammenarbeit. Zudem lernte ich, dass es entscheidend ist, sich auf die Kernanforderungen zu konzentrieren und sich nicht in zusätzlichen Features zu verlieren, die zwar interessant, aber nicht essentiell für das Projektziel sind.

Trotz der Herausforderungen und der Tatsache, dass das Projekt nicht ganz meinen ursprünglichen Erwartungen entsprach, betrachte ich die Erfahrung als äusserst wertvoll. Ich plane, das Projekt in Zukunft neu anzugehen, diesmal mit den gewonnenen Erfahrungen und einem besseren Verständnis der technischen Aspekte. Mit dem Wissen, das ich jetzt habe, bin ich zuversichtlich, dass ich eine verbesserte Version erstellen kann, die nicht nur funktionsfähig, sondern auch optimal strukturiert ist.

### 12.2 Persönliches Fazit – Mahir Gönen

Mein persönliches Fazit zum Projekt "Ski-Service.NET App" ist geprägt von einer intensiven Lernphase und signifikantem persönlichem Wachstum. Trotz anfänglicher Herausforderungen bei der Handhabung mehrerer Views, UserControls und der Umsetzung des MVVM-Patterns, erwies sich das Projekt als sehr lehrreich.

Die Komplexität, bot mir die Möglichkeit, meine Problemlösungskompetenzen zu schärfen. Diese Erfahrungen sind wertvoll für zukünftige Projekte, in denen ich die erlernten Kenntnisse und Fähigkeiten noch besser anwenden kann.

Es wurde viele Funktionen nach mehreren Meetings mit Bobby Bilali gekürzt oder komplett weggelassen. Das hat den Grund, dass einige Funktionen sehr komplex waren und nicht unbedingt in der Anforderung. Da auch die Erfahrung fehlte in einer WPF-Umgebung war mir sehr wichtig, dass das Projekt selbst von den Funktionen gut funktioniert und dass die Benutzeroberfläche ansprechend und nach Norm ist.

Insgesamt wurde das Projekt erfolgreich umgesetzt, aber mit dem Wissen und den Erfahrungen, die ich jetzt habe, könnte ich in einem zukünftigen Projekt sowohl den Code als auch das MVVM-Prinzip effizienter und effektiver gestalten.

## Anhänge

### I. Glossar

BEGRIFF	BESCHREIBUNG / ERKLÄRUNG
<b>.NET</b>	Ein Software-Framework von Microsoft, das zur Entwicklung und Ausführung von Anwendungen dient, insbesondere für Windows-Umgebungen.
<b>xUnit</b>	Eine Familie von Unit-Testing-Frameworks, die für verschiedene Programmiersprachen verfügbar sind und Entwicklern beim Testen von Code helfen.
<b>HTTP-Requests</b>	Anfragen, die über das Hypertext Transfer Protocol gesendet werden, um Daten zwischen Clients und Servern zu übertragen.
<b>appsettings.json</b>	Eine Konfigurationsdatei in .NET-Anwendungen, die Einstellungen wie Datenbankverbindungen und Anwendungsparameter speichert.
<b>TreeView</b>	Ein Benutzeroberflächenelement, das eine hierarchische Struktur von Informationen oder Optionen in einer Baumstruktur anzeigt.
<b>Mockups</b>	Visuelle Entwürfe oder Skizzen einer Anwendung oder Webseite, die das Layout und die Funktionalitäten zeigen.
<b>Mvvm-Design Pattern</b>	Ein Software-Designmuster, das Model-View-ViewModel bezeichnet, um die Trennung von Logik und Benutzeroberfläche in Anwendungen zu fördern.
<b>Web-API REST-Services</b>	Dienste, die über das Internet zugänglich sind und das Representational State Transfer (REST)-Architekturprinzip verwenden, um mit Anwendungen zu interagieren.

Tabelle 2 Glossar

## II. Weitere Dokumente

Test run finished: 5 Tests (5 Passed, 0 Failed, 0 Skipped) run in 133 ms

Test	Duration	Traits	Error Message
✓ SkiService.Tests (5)	65 ms		
✓ SkiService.Tests (5)	65 ms		
✓ BackendServiceTests (4)	58 ms		
✓ GetRegistrations_ReturnsExpectedData	52 ms		
✓ GetStatuses_ReturnsExpectedData	2 ms		
✓ LoginAsync_ReturnsAuthResponse	3 ms		
✓ UpdateRegistrationAsync_ReturnsTrueOnSuccess	1 ms		
✓ EditViewUserControllerModelTests (1)	7 ms		
✓ EditViewUserControllerModel_InitializesAllPropertiesCorrec...	7 ms		

**Group Summary**  
 SkiService.Tests  
 Tests in group: 5  
 Total Duration: 65 ms  
 Outcomes  
 5 Passed

**Jetstream Service**

Buttons: Show, Edit, Delete, Add, Suchen

Auftrags-Nr.	Nachname	Vorname	Service	Enddatum	Status
123123	Guilizzoni	Giacomo	Rennskiservice	27.12.2023	Offen
232423	Botton	Marco	kleiner Service	26.09.2023	Offen
456456	Maclochia	Marioh	Heisswachsen	21.09.2023	In Bearbeitung
546263	Mofo	Liberty	Bindung mont. und Einste	30.12.2023	Storniert

**Jetstream Service - Kunde**

Kunden-Nr: 123123  
 Name: Guilizzoni, Vorname: Giacomo  
 Email-Adresse: g\_giacomo@gmail.com, Telefonnummer: +4176 123 456 78 90

**Jetstream Service - Service Auftrag**

Auftragsdatum: 15.12.2023, Status: Offen  
 Abholdatum: 27.12.2023, Priorität: Hoch  
 Service: Rennskiservice, Kommentar: Kunde hat spezielle ski, es muss extra Material bestellt werden!  
 Preis in CHF: CHF 109

Abbildung 7 Mockups - Bobby Bilali

**Jetstream Skiservice Adminpanel**

Buttons: Edit, Delete, Add, Help

Buttons: Edit, Delete, Add, Suchen

Auftrags-Nr.	Nachname	Vorname	Service	Enddatum	Status	Priorität
123123	Guilizzoni	Giacomo	Rennskiservice	27.12.2023	Offen	Tief
232423	Botton	Marco	Kleiner Service	28.12.2023	Offen	Standard

**Offene Aufträge**

- Prio: Tief
  - Heisswachsen
  - Fell zuschneiden
  - Rennski Service
  - Standard Service
  - Kleiner Service
- Prio: Standard
- Prio: Express

**In Bearbeitung**

- Prio: Tief
- Prio: Standard
- Prio: Express

**abgeschlossene**

Email	Telefonnummer	Bestelldatum	Enddatum	Preis	Kommentar
guilizzio@gmail.com	+417984445434	10.12.2023	27.12.2023	CHF109	Spezieller Wachs wird benötigt

Abbildung 8 Dashboard Mockup - Mahir Gönen

### III. Versionsverzeichnis der Anwendung/Pakete

Anwendung/Paket	Version
.NET WPF	.NET8.0
Fluent.Ribbon	10.0.4
Gu.Wpf.Adorners	2.1.1
Microsoft.Extensions.Configuration	8.0.0
Microsoft.Extensions.Configuration.Binder	8.0.0
Microsoft.Extensions.Configuration.FileExtensions	8.0.0
Microsoft.Extensions.Configuration.Json	8.0.0
Microsoft.DependencyInjection	8.0.0
Microsoft.Extensions.Http	8.0.0
Newtonsoft.Json	13.0.3
FluentAssertions	6.12.0
Microsoft.Extensions.Configuration.Abstractions	8.0.0
Microsoft.NET.Test.Sdk	17.6.0
Moq	4.20.69
Xunit	2.6.1
xunit.runner.visualstudio	2.4.5
coverlet.collector	3.2.0

## 13 Quellenverzeichnis

Dokumente aus dem OneNote

## 14 Tabellenverzeichnis

Tabelle 1: Grobe Planung.....	7
Tabelle 2 Glossar.....	20

## 15 Abbildungsverzeichnis

Abbildung	
Abbildung 2 Anwendungsdesign .....	9
Abbildung 3 Dashboard Listeneinträge .....	11
Abbildung 4 Login-Dialog .....	11
Abbildung 5 Dashboard "Editieren"-Ansicht .....	12
Abbildung 6 Dashboard mit Menüband v2.0 .....	12
Abbildung 7 Mockups - Bobby Bilali.....	22
Abbildung 8 Dashboard – Mahir Gönen.....	22