Ski-Service .nET-APP

Projektdokumentation

Fokko Vos & Robin Ruf

IBZ 2023

Inhaltsverzeichnis

[1 Versionsverlauf 1](#_Toc152068762)

[2 Einleitung 1](#_Toc152068763)

[3 Informieren 2](#_Toc152068764)

[3.1 Ausgangssituation 2](#_Toc152068765)

[3.2 Anforderungsanalyse 2](#_Toc152068766)

[3.2.1 Funktionale Anforderungen 2](#_Toc152068767)

[3.2.2 Nicht-Funktionale Anforderungen 2](#_Toc152068768)

[3.3 Technische Anforderungen 3](#_Toc152068769)

[4 Planen 3](#_Toc152068770)

[4.1 Zeitplan 3](#_Toc152068771)

[4.2 Systemarchitekturentwurf 4](#_Toc152068772)

[5 Entscheiden 5](#_Toc152068773)

[5.1 Technologie und Testgeräte 5](#_Toc152068774)

[5.2 Test-Strategie 5](#_Toc152068775)

[6 Realisieren 6](#_Toc152068776)

[7 Kontrollieren 6](#_Toc152068777)

[8 Auswerten 6](#_Toc152068778)

[9 Anhänge 7](#_Toc152068779)

[9.1 Quellen 7](#_Toc152068780)

[9.2 Abbildungsverzeichnis 7](#_Toc152068781)

[9.3 Tabellenverzeichnis 7](#_Toc152068782)

# Versionsverlauf

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Autor | Bemerkung |
| 1.0 | 28. November 2023 | Fokko Vos & Robin Ruf | Initial Version |
|  |  |  |  |

1 - Versionsverlauf

# Einleitung

Willkommen zur Dokumentation des Projekts „Ski-Service.NET App“ der Firma Jetstream-Service. In einer Welt, in der digitale Technologien zunehmend an Bedeutung gewinnen, ist die Anpassung und Optimierung von Geschäftsprozessen durch innovative Lösungen ein Schlüsselfaktor für den Unternehmenserfolg. Die Ski-Service.NET App ist eine Antwort auf die Notwendigkeit, den Workflow und die Datenverwaltung im Bereich der Ski-Service-Aufträge effizienter zu gestalten.

Diese Dokumentation bietet einen umfassenden Einblick in das Projekt, das darauf abzielt, eine benutzerfreundliche und für unterschiedliche Endgeräte optimierte Anwendung zu entwickeln. Unser Ziel ist es, eine intuitive Benutzeroberfläche zu schaffen, die den spezifischen Anforderungen und Arbeitsbedingungen einer Skiservice-Werkstatt gerecht wird.

In dieser Dokumentation werden alle Phasen des Projekts abgedeckt, von der Konzeption bis zur technischen Umsetzung, einschließlich der Überlegungen zu den verwendeten Technologien und der Integration bestehender Systeme. Sie dient als Leitfaden und Informationsquelle für alle Projektbeteiligten und Stakeholder und gibt einen klaren Überblick über die Ziele, Strategien und erwarteten Ergebnisse des Projekts.

# Informieren

## Ausgangssituation

**Unternehmen**  
Jetstream-Service, ein Unternehmen, das sich auf Skiservicearbeiten spezialisiert hat.

**Aktueller Stand**  
Das Unternehmen hat neue Touchscreen-fähige Hardware angeschafft (Tablets, Surface) und stellt diese den Mitarbeitern für die Datenpflege der Ski-Service-Aufträge zur Verfügung.

**Ziel**  
Die einfache, intuitive und aufgabenangemessene Bedienung der Benutzeroberfläche, unter Berücksichtigung der Arbeit mit Handschuhen.

**Integration**  
Erweiterung der existierenden Datenbasis zur Online-Anmeldung und der Service-App mit der neuen GUI-Lösung für Tablets und Handys.

## Anforderungsanalyse

### Funktionale Anforderungen

* Sicheres Login-System für Mitarbeiter mit Benutzername und Passwort.
* Verwaltung von Serviceaufträgen
  + Anzeige aller anstehenden Ski-Serviceaufträge in einer Liste.
  + Möglichkeit zur Filterung und Suche innerhalb der Serviceauftragsliste.
  + Fähigkeit zur Mutation (Statusänderung, Notizen hinzufügen, Löschen) bestehender Serviceaufträge.
  + Unterschiedliche Statusoptionen für Aufträge: Offen, In-Arbeit, Abgeschlossen.
  + Zuordnung und Verwaltung von Prioritäten bei Serviceaufträgen.
  + Ergänzung von Kundeninformationen (Name, E-Mail, Telefon) bei Bedarf.
  + Eindeutige Zuordnung einer Dienstleistung pro Serviceauftrag aus einer vorgegebenen Liste (z.B. Kleiner Service, Großer Service, Rennski-Service).

### Nicht-Funktionale Anforderungen

* Starke Authentifizierung und Autorisierung, um den Zugang zu sensiblen Daten zu sichern.
* Schnelle Verarbeitung von Anfragen, um eine effiziente Abwicklung der Serviceaufträge zu gewährleisten.
* Einfache, intuitive und aufgabenangemessene Benutzeroberfläche, die auch mit Handschuhen bedienbar ist.
* Stabile und fehlerresistente Anwendungsleistung, insbesondere während der Hochsaison.

## Technische Anforderungen

**UX und UI Design**  
Erstellung von Mockups, die eine einfache und aufgabenangemessene Bedienung ermöglichen, unter besonderer Berücksichtigung der Bedienbarkeit mit Handschuhen.

**Integration und Kompatibilität**  
Sicherstellung, dass das neue User Interface reibungslos mit dem bereits existierenden Backend-System integriert wird. Dies beinhaltet die Anbindung an die bestehenden Web-APIs und Datenstrukturen. Überprüfung und ggf. Anpassung der Schnittstellen, um eine effiziente Kommunikation zwischen dem neuen Frontend und dem existierenden Backend zu gewährleisten.

**Test und Qualitätssicherung**  
Durchführung umfassender Tests, um die Funktionalität und Leistung der neuen UI-Komponenten sowie deren Integration mit dem Backend zu überprüfen.

# Planen

## Zeitplan

Der Aufwand in Stunden bezieht sich auf die von allen Projektmitgliedern gemeinsam aufgewendete Zeit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Beschreibung** | **SOLL-Zeit (h)** | **IST-Zeit (h)** |
| **1** | **Informieren** | **6** |  |
| 1.1 | Situationsanalyse | 2 | 2 |
| 1.2 | Erfassung der Nutzerbedürfnisse | 3 |  |
| 1.3 | Erfassung der Schulischen anforderungen | 1 |  |
| **2** | **Planen** | **12.5** |  |
| 2.1 | Erstellung eines Zeitplans | 1.5 | 1.5 |
| 2.2 | Entwurf der Systemarchitektur | 1 | 1 |
| 2.3 | Mockups für die Benutzeroberfläche | 10 |  |
| **3** | **Entscheiden** | **2.5** |  |
| 3.1 | Technologienwahl | 0.5 | 1 |
| 3.2 | Test-Strategie | 2 | 2 |
| **4** | **Realisieren** | **24.5** |  |
| 4.1 | Git-Repository Einrichten | 0.5 | 0.5 |
| 4.2 | Verbindung zum Backend aufbauen | 1 |  |
| 4.3 | Entwicklung der Benutzeroberfläche | 15 |  |
| 4.4 | Entwicklung der nötigen Tests | 8 |  |
| **5** | **Kontrollieren** | **3** |  |
| 5.1 | Test-Strategie ausführen | 1 |  |
| 5.2 | Anforderungen mit dem Produkt abgleichen | 2 |  |
| **6** | **Auswerten** | **6** |  |
| 6.1 | Finalisierung der Dokumentation | 2 |  |
| 6.2 | Lessons-Learned Identifizieren | 2 |  |
| 6.3 | Präsentation vorbereiten | 2 |  |
| **Gesamt** | | **54.5** |  |

2 - Zeitplan

## Systemarchitekturentwurf

1 - Systemarchitekturentwurf

Das System, das wir für die "Ski-Service.NET App" entwerfen, ist eine Weiterentwicklung der bestehenden Drei-Schichten-Architektur und integriert neue Komponenten für erweiterte Funktionalität und Benutzerfreundlichkeit. Es folgt eine Beschreibung der aktualisierten Systemarchitektur basierend auf dem visuellen Entwurf.

**Web Browser (Frontend UI)**  
Die Präsentationsschicht, zugänglich über einen Web Browser, bleibt die primäre Schnittstelle für Benutzerinteraktionen

**Web API (ASP.NET API)**  
Die Geschäftslogikschicht wird durch eine ASP.NET API repräsentiert, die das Rückgrat unserer Anwendung bildet. Sie verarbeitet Anfragen, führt Geschäftsregeln aus und handhabt die Datenkommunikation mit der Datenbank.

**Datenbank (MSSQL)**  
Die Datenpersistenzschicht basiert weiterhin auf Microsoft SQL Server, welcher die notwendigen Daten speichert und bereitstellt. Diese Schicht ist für hohe Leistung und Transaktionsintegrität optimiert und ermöglicht schnelle Datenabfragen.

**Mobile App (.NET MAUI)**  
Als neue Komponente führen wir eine mobile Anwendung ein, die mit .NET MAUI entwickelt wird. Diese App erweitert die Zugänglichkeit der Systemfunktionen und bietet eine optimierte Benutzererfahrung für mobile Geräte.

Die Netzwerkverbindung zwischen allen Schichten gewährleistet sichere und effiziente Datenübertragungen über HTTPS, wobei JSON als Datenaustauschformat zum Einsatz kommt. Dieses Format unterstützt eine hohe Interoperabilität und erleichtert das Parsen der Daten.

Diese erweiterte Architektur erhält die bewährten Prinzipien der klaren Schichtentrennung und fügt gleichzeitig neue Elemente hinzu, um die Anwendung modern, flexibel und zukunftssicher zu machen. Sie berücksichtigt sowohl die aktuellen Anforderungen als auch die Skalierbarkeit für zukünftige Erweiterungen und ist somit ein solides Fundament für die fortlaufende Entwicklung unserer Projektziele.

# Entscheiden

## Technologie und Testgeräte

Die technologische Basis für das Projekt "Ski-Service.NET App" bildet die Entscheidung für die Nutzung von .NET MAUI und .NET 8, wodurch wir die Vorteile der neuesten Entwicklungen in der Softwaretechnologie voll ausschöpfen. .NET MAUI ermöglicht uns eine plattformübergreifende Entwicklung, während .NET 8 uns Zugriff auf die neuesten Sprachfeatures, Sicherheitsupdates und Performance-Optimierungen bietet.

Für die Entwicklungsumgebung haben wir uns auf Visual Studio festgelegt. Diese IDE stellt uns alle notwendigen Werkzeuge und Emulatoren zur Verfügung, um eine effiziente Entwicklung und ein akkurates Testing zu ermöglichen. Die Emulatoren in Visual Studio erlauben es uns, ein breites Spektrum an Geräten zu simulieren, um so die Kompatibilität und Benutzerfreundlichkeit unserer App sicherzustellen.

Speziell für die Testzwecke haben wir uns entschieden, das "Tablet 420 DPI 8in" als Standard-Testgerät zu verwenden. Dieses Gerät, mit einem Arbeitsspeicher von 1 GB und einer Auflösung von 2200 x 2480 bei 420 DPI, gewährleistet, dass wir die App unter realistischen Bedingungen entwickeln und testen können. Diese Spezifikationen stellen sicher, dass die App performant läuft und die Benutzererfahrung optimiert ist, insbesondere in Bezug auf die Darstellung und Handhabung der Benutzeroberfläche auf Geräten mit hoher Pixel- und Punktdichte (DPI).

## Test-Strategie

Für die "Ski-Service.NET App" wird eine mehrschichtige Teststrategie verfolgt, die sicherstellt, dass sowohl die einzelnen Komponenten als auch die Anwendung als Ganzes die festgelegten Anforderungen und Qualitätsstandards erfüllen. Die Entscheidung für unsere Teststrategie beinhaltet folgende Kernpunkte

**Unit Tests**  
Werden verwendet, um die grundlegenden Bausteine der Anwendung zu prüfen.

**Integrationstests**  
Stellen das korrekte Zusammenspiel der einzelnen Module und Dienste sicher.

**End-to-End-Tests**  
Simulieren den vollständigen Ablauf von wichtigen Benutzerinteraktionen mit der Anwendung.

**UI Tests**  
Konzentrieren sich auf die Überprüfung der Benutzeroberfläche auf korrekte Darstellung und Interaktion.

Zusätzlich zu diesen automatisierten Testverfahren werden manuelle **Usability Tests** während der gesamten Entwicklung durchgeführt. Diese Tests sind entscheidend, um ein benutzerzentriertes Design zu gewährleisten und um sicherzustellen, dass die Anwendung nicht nur funktioniert, sondern auch effizient und angenehm zu bedienen ist. Die Usability Tests werden in verschiedenen Phasen der Entwicklung durchgeführt, um kontinuierliches Feedback in den Entwicklungsprozess einfließen zu lassen und eine hohe Gebrauchstauglichkeit zu sichern.

# Realisieren

# Kontrollieren

# Auswerten

# Anhänge

## Quellen

**Im aktuellen Dokument sind keine Quellen vorhanden.**

## Abbildungsverzeichnis

[1 - Systemarchitekturentwurf 4](file:///C:\Github\ict-322-ski-service-app\files\Dokumentation.docx#_Toc152068783)

## Tabellenverzeichnis

[1 - Versionsverlauf 1](#_Toc152068784)

[2 - Zeitplan 3](#_Toc152068785)