WayMatcher

WayMatcher

- **Projekt:** WayMatcher – Fahrgemeinschafts-Applikation
- **Angebot an:** Hr. Rudigier & Hr. Wonnebauer
- **Auftraggeber:** Hr. Pausch
- **Autor des Dokuments:** Hr. Rudigier & Hr. Wonnebauer

---

1. Einführung und Zielbestimmung

Im Folgenden werden die Anforderungen für das Projekt und die aktuelle Ausgangssituation beschrieben.

1.1. Beschreibung des Unternehmens/Organisation

WayMatcher ist ein Startup-Unternehmen, das sich auf die Entwicklung innovativer und nachhaltiger Mobilitätslösungen spezialisiert hat.
Das Unternehmen verfolgt das Ziel, den Individualverkehr zu reduzieren und die Nutzung von umweltfreundlichen Verkehrsmitteln zu fördern.

1.2. Beschreibung und Hintergründe des Projektpraktikums

Im Rahmen des Projektpraktikums soll die Webanwendung **WayMatcher** entwickelt werden, die eine Plattform zur Vermittlung von Mitfahrgelegenheiten bietet.
Das Projektpraktikum dient dazu, die in der Berufsschule erworbenen Kenntnisse in der Praxis anzuwenden und Erfahrungen in der Softwareentwicklung zu sammeln.

---

2. Produktübersicht und Einsatz

2.1. Aktuelle Situation

Derzeit existiert keine zentrale und benutzerfreundliche Plattform, die die Vermittlung von Mitfahrgelegenheiten im regionalen Bereich effizient unterstützt.
WayMatcher soll diese Lücke schließen und eine einfache und komfortable Möglichkeit bieten, Mitfahrgelegenheiten anzubieten oder zu finden.

2.2. Beschreibung des Soll-Konzepts

WayMatcher soll eine Webanwendung sein, die es Nutzern ermöglicht, Reisetermine mit detaillierten Informationen zu erstellen und nach passenden Mitfahrgelegenheiten zu suchen.
Die Anwendung soll sowohl für die Desktop- als auch für die mobile Nutzung optimiert sein und eine intuitive Benutzeroberfläche bieten.

2.3. Beschreibung von minimalen Schnittstellen und Funktionen

2.3.1. Relationale Datenbank

- Eine relationale Datenbank (**Azure**) wird zur Speicherung der Anwendungsdaten verwendet.
- Die Datenbank enthält Tabellen für Benutzer, Reisetermine, Nachrichten etc.

2.3.2. Backend

- Das Backend wird mit **C# mittels ASP.NET (MVC)** implementiert.
- Es stellt eine **REST-API** für die Kommunikation mit dem Frontend bereit.
- Das Backend ist verantwortlich für die **Datenverwaltung, die Benutzerauthentifizierung und die Businesslogik** der Anwendung.

2.3.3. Frontend

- Das Frontend wird mit **React über TypeScript** entwickelt.
- Es bietet eine **benutzerfreundliche Oberfläche** zur Interaktion mit der Anwendung.
- Das Frontend kommuniziert mit dem Backend über die **REST-API-Schnittstelle**.

2.3.4. Quellcodeverwaltung

- Die **Quellcodeverwaltung erfolgt mit Git**.
- Ein **Git-Repository wird zur Speicherung und Versionierung** des Quellcodes verwendet.
- Das Repository nutzt **GitHub für Dokumentation und Zusammenarbeit**.

2.4. Funktionale Anforderungen

- **Reisetermine:**
    - Erstellen von Reiseterminen mit Typ (Passagier/Pilot), Zielort, Datum & Uhrzeit (wiederkehrend), Strecke (Start & Ziel mit Radius), Plätze, Inhaber, Preis/km (automatische Berechnung), One-Shot/Seriell, Beschreibung, Fortbewegungsmittel, Status (offen, limitiert, abgebrochen, erledigt).
    - Beitreten und Erstellen von Reiseterminen.
    - Abbrechen von Reiseterminen mit Begründung.
    - Integrierter Chat innerhalb eines Reisetermins.

- **Einladungen zu Reiseterminen:**
    - Versenden von Einladungen mit Reisetermin-Daten und kurzer Nachricht.
    - Pflicht zum Ausfüllen des **Passagierformulars** nach Annahme der Einladung.
    - Möglichkeit für Piloten, das Passagierformular zu akzeptieren oder abzulehnen.
- **Passagierformular:**
    - Zielort, Startort (inkl. Suchradius), Datum & Uhrzeit, Passagiere, Beschreibung.
    - Erstellen von Angeboten an Piloten.
    - Abbrechen des Termins mit Begründung.
- **Benutzerverwaltung:**
    - Registrierung und Login mit E-Mail, Passwort und Benutzername.
    - Profilverwaltung mit Führerschein, Fahrzeugbesitz, Profilbild, Beschreibung, Terminverlauf, offenen Terminen, Statistik und Bewertungen (1-5 Sterne).
    - Login/Registrierung/Bearbeiten/Löschen Funktionen.
    - Liste von Fortbewegungsmitteln.
- **Suche/Filter:**
    - Suche nach Reiseterminen anhand von Orten/Strecke, Uhrzeit/Datum (Hin- und Rückfahrt), Anzahl freier Plätze, bestimmte Benutzer, beigetretene Reisetermine.

2.5. Nichtfunktionale Anforderungen

- **Usability:** Intuitive Bedienung, klare Navigation, übersichtliche Gestaltung.
- **Performance:** Schnelle Ladezeiten, Skalierbarkeit.
- **Sicherheit:** Datenschutz, Datensicherheit, sichere Authentifizierung.
- **Zuverlässigkeit:** Stabilität, Fehlertoleranz, Datenintegrität.
- **Kompatibilität:** Browserkompatibilität, Gerätekompatibilität, Betriebssystemkompatibilität.
- **Wartbarkeit:** Modularer Aufbau, Dokumentation, Testbarkeit.

2.6. Technische Grundlagen

- Die **Website soll lokal gehostet werden** können.
- Soll für **Mobile-Geräte** benutzbar sein.
- Es muss dem Benutzer **schnell reagieren**.

2.7. Projektverlauf

- Definition des Projektumfangs und der Ziele.
- Analyse der Anforderungen und Erstellung des Lastenhefts.
- **Kommunikation zwischen den Projektmitgliedern:**
    - Wöchentliche Besprechungen mit Dokumentation.
    - Wöchentliche Planungen (**Kanban Board über GitHub**).
    - Leserechte für den Projektbetreuer.

- Design und Entwicklung der Anwendung (Frontend und Backend).
- Testphase und Fehlerbehebung.
- Deployment und Inbetriebnahme der Anwendung.

---

3. Minimale Qualitätsanforderungen an den Quellcode

- **Sauberer, lesbarer und gut dokumentierter Code.**
- **Einhaltung einheitlicher Code-Konventionen.**
- **Getesteter und fehlerfreier Code.**

---

4. Applikation im Livebetrieb

- **Bereitstellung auf einem lokalen Server.**
- **Skalierbare Infrastruktur zur Unterstützung einer wachsenden Nutzerzahl.**

---

5. Zeitliche Vorgaben und Deadlines

- **Projektstart:** 05.02.2025
- **Lastenheft abgeschlossen:** 07.02.2025
- **Pflichtenheft abgeschlossen:** 21.02.2025
- **Umsetzung:** 24.02.2025 - 18.03.2025
- **Testphase:** 18.03.2025 - 24.03.2025
- **Projektpräsentation:** 25.03.2025
- **Projektabschluss:** 01.04.2025

---

6. Check-Liste

6.1. Hardware:

Status	Hardware
	Notebook
	Virtualisierte Serverinstanzen
	Externe Hardware (falls benötigt)

6.2. Software und Datenbank:

Status	Software/Datenbank
	Funktionierende Serverdienste für Backend nach Anforderung im Pflichtenheft
	Frontend nach Anforderung im Pflichtenheft
	Relationale Datenbank nach Anforderung mit CRUD und referentieller Integrität
	Datenbank: View / Trigger / Funktion / Prozedur (3 davon verwenden)

6.3. Dokumente:

Status	Dokument
	Projektsteckbrief
	Lastenheft
	Pflichtenheft
	Projekthandbuch vollständig mit Wochenberichten
	Technische Dokumentation und Bedienungsanleitung
	Quellcodedokumentation – generiert und digital
	Projektstrukturplan

6.4. Präsentation:

Status	Präsentation
	Mindestens 8 Folien gem. Vorgabe
	Bei einer Gruppe, beide Projektpartner anwesend und gleichmäßige Aufteilung der Präsentation
	Vorführung des Projektes, der Applikation und Erklärungen der techischen Details
	Präsentationsdauer ca. 18-22 Minuten pro Projektmitglied

7. Verpflichtende Erweiterungen

7.1. Datenbank

Status	Erweiterung	Note
	Funktion sinnvoll eingesetzt	1
	Prozedur sinnvoll eingesetzt	2
	View sinnvoll eingesetzt	3

7.2. Dokumente sind für die jeweilige Beurteilung erforderlich

Status	Erweiterung	Note
	vollständiges GANTT-Diagramm (Zeitvorgaben real)	3
	Testkonzept, vollständig dokumentiert und funktional	2
	Kostenplan	3
	Netzplan mit kritischem Pfad	2
	Ressourcenplan	2

Status	Erweiterung	Note
	DevOps oder agiles Board (vollständige Planung)	1
	Sicherheitskonzept (Betrieb, Entwicklung, ...)	1

8. Optionale Erweiterungen

Status	Erweiterung	Note
	Intelligente Pfadfindung der Reisetermine (z.B. Suche in einem Radius am Weg)	1
	Chat zwischen Benutzern	1
	Verwendung von **Designpattern**	2
	Detailierte Benutzerprofile (Bewertungen, Automodell, Führerschein)	2
	Multifaktor Authentifizierung	2
	Github Dokumentation via Markdown und Projektmanagement	2
	Karte hinzufügen (Google Maps, Ansicht der Strecke)	3
	Clouddienste (App-Service)	3
	E-Mail Versand (Passwort zurücksetzen & Benachrichtigungen)	3