
Projektdokumentation

Entwicklung eines Systems zur personalisierten Empfehlung von Ausflugszielen in der Schweiz

Modul	431 Aufträge im IT-Umfeld selbständig durchführen
Eingereicht von	J. Vetsch A. Herbst O. Kolomiets M. Saez
Projektthema	Personalisierte Empfehlung von Ausflugszielen
Eingereicht bei	Thomas Jäggi
Datum	18. Oktober 2025

Dokumentinformation

Organisation	
Projektleiter	Jonas Vetsch
Stv. Projektleiter	Adrian Herbst
Projektmitglieder	Adrian Herbst Oleksandr Kolomiets Marco Saez

Inhaltsverzeichnis

1	Zusammenfassung / Abstract.....	5
2	Einleitung.....	6
3	Initialisierungsphase	7
3.1	Ausgangslage.....	7
3.2	Ziele.....	7
3.3	Ist-Soll-Analyse.....	9
3.4	Anforderungen an das Projekt	9
3.5	Lösungsvarianten	11
3.6	Zeitplan und Projektstrukturplan.....	15
3.7	Organisation	15
3.8	Pendenzenliste	17
4	Konzeptphase.....	19
4.1	Zielgruppe (Nutzerskizze).....	19
4.2	Layout (Aussenskizze).....	21
4.3	Bauplan / Installation (Innenskizze).....	22
4.4	Funktionalitäten (Ablaufskizze).....	26
4.5	Sachmittel (Materialliste).....	27
4.6	Testfälle	28
5	Realisierungsphase	32
5.1	Systemdesign und Schnittstellen	32
5.2	Systemdokumentation	35
5.3	Tests.....	40
6	Reflexion.....	43
6.1	Jonas Vetsch (Projektleitung)	43
6.2	Adrian Herbst (Mitarbeiter / Stv. Projektleitung)	45
6.3	Marco Saez (Mitarbeiter)	46
6.4	Oleksandr Kolomiets (Mitarbeiter).....	48
7	Fazit.....	49
8	Anhang: Arbeitsjournal Realisierungsphase.....	51
8.1	Jonas Vetsch	51
8.2	Adrian Herbst.....	56
8.3	Marco Saez	58
8.4	Oleksandr Kolomiets	60

Vorwort

Dieses Projekt wurde im Rahmen des Moduls 431 „Aufträge im IT-Umfeld selbständig durchführen“ an der Gewerblich-Industriellen Berufsfachschule Bern (gibb) durchgeführt.

Im Zentrum des Moduls stand nicht die technische Entwicklung eines komplexen Produkts, sondern die methodisch korrekte Anwendung der HERMES-Projektmanagementmethode.

Unser vierköpfiges Team – bestehend aus Jonas Vetsch, Adrian Herbst, Marco Saez und Oleksandr Kolomiiets – entwickelte im Verlauf des Herbstquartals 2025 ein System zur personalisierten Empfehlung von Ausflugszielen in der Schweiz. Das Projekt wurde vollständig nach dem klassischen HERMES-Vorgehensmodell umgesetzt, wobei jede Phase klar definiert, geplant und dokumentiert wurde. Die Phasen sind Initialisierung, Konzept, Realisierung und Einführung.

Die vorliegende Projektdokumentation fasst sämtliche Arbeitsschritte, Entscheidungen und Resultate zusammen. Sie zeigt auf, wie wir innerhalb eines begrenzten Rahmens ein IT-Projekt strukturiert, nachvollziehbar und praxisnah umsetzen konnten.

Ein besonderer Fokus liegt dabei auf der methodischen Qualität: Alle Entscheidungen wurden im Sinne der HERMES-Methode fachlich und vorgehensbezogen begründet, um den gesamten Projektverlauf transparent und prüfbar zu machen.

1 Zusammenfassung / Abstract

Diese Projektdokumentation beschreibt Planung, Entwicklung und Einführung eines Systems (Programm bzw. Algorithmus) zur personalisierten Empfehlung von Ausflugszielen in der Schweiz und schliesst damit eine Lücke gegenüber kategorienzentrierten Plattformen.

Das System wurde in C# als interaktives Konsolenprogramm umgesetzt und nutzt eine Empfehlungslogik auf Basis dreier Spannungsfelder (kulturelle Komponente, körperliche Aktivität, Indoor / Outdoor), die zur Analyse des Nutzerbedürfnisses verwendet werden.

Das Projekt wurde konsequent nach HERMES durchgeführt: Im abschliessenden Test erfüllte das Produkt die Anforderung an die Zufriedenheit und zudem erfüllen wir auch die übrigen definierten Ziel- und Nutzenvorgaben vollständig.

2 Einleitung

Die Idee zu diesem Projekt entstand zu Beginn des Moduls 431 im Rahmen der Themenfindung für ein IT-Projekt, das einerseits realisierbar, andererseits methodisch anspruchsvoll sein sollte.

Bei der Analyse bestehender Plattformen zur Ausflugsplanung in der Schweiz (beispielsweise myswitzerland.com, ausflugsziele.ch oder SBB Freizeitideen) zeigte sich ein gemeinsames Muster: Alle Lösungen verfügen über umfangreiche Datenbanken und ansprechendes Bildmaterial, verlangen aber ein manuelles Durchsuchen nach Kategorien und Filtern. Eine echte Personalisierung der Empfehlungen findet kaum statt.

Aus dieser Beobachtung entstand unsere Projektidee, ein System auf der dritten Personalisierungsstufe zu entwickeln, das nicht Kategorien, sondern die individuellen Bedürfnisse der Nutzer ins Zentrum stellt. Durch die Abfrage weniger, aber aussagekräftiger Spannungsfelder wollen wir passende Ausflugsziele vorschlagen, ohne dass die Nutzer*innen bereits wissen müssen, wonach sie suchen.

Das Projekt wurde gemäss der klassischen HERMES-Vorgehensweise umgesetzt.

- In der Initialisierungsphase wurden Ausgangslage, Ziele und Anforderungen analysiert sowie verschiedene Lösungsvarianten evaluiert.
- In der Konzeptphase folgte die Definition der Systemarchitektur, der Abläufe und Testfälle.
- In der Realisierungsphase wurde der Prototyp in C# programmiert und getestet.
- Schliesslich erfolgte in der Einführungsphase die praktische Erprobung mit Testpersonen und die Auswertung der Zufriedenheit.

Mit unserem Projekt zeigen wir, wie ein kleines Team durch strukturiertes Vorgehen nach HERMES ein funktionsfähiges IT-System entwickeln kann, das einen klaren praktischen Nutzen bietet und gleichzeitig die methodischen Anforderungen eines professionellen Projektmanagements erfüllt.

3 Initialisierungsphase

Die Initialisierungsphase ist der erste Teil des Projektmanagement nach HERMES. Es geht in erster Linie darum, das Projekt abzugrenzen und gewisse Rahmenbedingungen zu setzen. Dazu gehört es, die Organisation und Rollen zu definieren, eine Planung zu erstellen, sowie die Machbarkeit zu analysieren. Ist dies alles solide gemacht, wird in unserem Modul das Projekt für die Konzeptphase sowie alle weiteren Phasen freigegeben.

3.1 Ausgangslage

Im Schweizer Tourismusmarkt existieren bereits zahlreiche Plattformen zur Suche nach Ausflugszielen (z. B. ausflugsziele.ch, myschweiz.com, SBB-Freizeitideen). Diese Angebote verfügen über umfangreiche Datenbanken, hochwertiges Bildmaterial und sehr detaillierte Berichte. Sie sind jedoch überwiegend kategorienzentriert aufgebaut: Nutzer müssen sich manuell durch viele Filter und Kategorien klicken und bereits relativ genau wissen, wonach sie suchen. Eine echte Personalisierung findet höchstens in sehr grundlegender Form statt (z. B. Standortangabe, Zielgruppe oder Distanz).

Unsere Analyse hat ein Drei-Stufen-Modell der Personalisierung identifiziert:

Stufe	Personalisierung	Merkmale	Was steht im Zentrum?	Eigenleistung des Nutzers
1	Keine	reine Kategoriensuche	Kategorien	Hoch!
2	Leicht	Suchfunktionen mit zusätzlichen Kriterien zur Zielgruppe	Kategorien	Mittel bis hoch
3	Stark	Empfehlungen basieren auf Nutzerbedürfnissen statt Kategorien	Nutzer	gering

Alle untersuchten Plattformen bewegen sich heute auf Stufe 1 oder 2. Eine Lösung auf Stufe 3 konnten wir nicht identifizieren. Dies steht im Kontrast zum globalen Trend einer zunehmenden starken Personalisierung von Online-Dienstleistungen.

Daraus ergibt sich ein erhebliches Potenzial: Mit einem System zur stark personalisierten Empfehlung von Ausflugszielen in der Schweiz könnte eine Marktlücke geschlossen werden.

Vor diesem Hintergrund hat uns Schweiz Tourismus (fiktiv!) beauftragt, ein innovatives Empfehlungssystem zu entwickeln, das nicht Kategorien, sondern Nutzerbedürfnisse ins Zentrum stellt und dadurch relevantere, personalisierte Vorschläge generiert.

3.2 Ziele

Nr.	Ziel	Beschreibung
1	Personalisierte Vorschläge	Bis zum 5.11.25 steht ein System zur Verfügung, das mit Nutzereingaben die Bedürfnisse (Präferenzen) des Nutzers abfragt, verarbeitet und daraus passende Ausflugsziele empfiehlt.

2	Innovative Empfehlungslogik	Bis zum 5.11.25 ist eine Empfehlungslogik implementiert, die mindestens drei Spannungsfelder (z. B. Aktivitätsintensität, Sozialform, Geistige Reize) berücksichtigt und damit die Eingaben der Nutzer systematisch in personalisierte Ausflugsvorschläge übersetzt.
3	Benutzerinteraktion	Das System läuft per 5.11.25 über die Kommandozeile als einfache, interaktive Benutzerschnittstelle, in der der Nutzer in max. 5 Schritten seine Präferenzen angeben kann.
4	Prototyp im Projektumfang	Das System steht bis zum 5.11.25 als funktionierender Prototyp als Konsolenprogramm bereit.
5	Zufriedenheit prüfen	Bis Start der Einführungsphase wird das System mit mindestens 5 Nutzern getestet, die auf Papier ihre Zufriedenheit mit den empfohlenen Reisezielen von 1-10 bewerten. Dabei soll der Durchschnitt der Wertungen mindestens 6.0 betragen.

3.3 Ist-Soll-Analyse

3.3.1 Analyse des Ist-Zustands

Im Schweizer Tourismusmarkt stehen Nutzer vor einer Vielzahl von Plattformen zur Suche nach Ausflugszielen. Diese zeichnen sich durch grosse Datenbanken, qualitatives Bildmaterial und detailreiche Berichte aus. Dennoch zeigen alle untersuchten Lösungen strukturelle Schwächen: Sie sind stark kategorienzentriert, verlangen hohes manuelles Filtern und setzen voraus, dass Nutzer schon relativ klare Vorstellungen zu ihrem geplanten Ausflug mitbringen. Personalisierung findet lediglich in rudimentärer Form statt – etwa über Standortfilter, Distanzangaben oder leichte Zielgruppeneingrenzung.

Eine echte, auf Bedürfnisse zugeschnittene Empfehlung existiert bisher nicht. Damit bleibt das Nutzungserlebnis aufwendig und für viele potenzielle Anwender unattraktiv, insbesondere für jene, die spontan Inspiration suchen oder nur wenig Zeit investieren wollen.

3.3.2 Ableitung des Soll-Zustands

Aus dieser Analyse ergibt sich eine klare Lücke: Ein System, das die individuellen Bedürfnisse ins Zentrum stellt und zudem den Suchaufwand deutlich reduziert. Ziel ist es, die Suche nutzerzentriert zu gestalten, sodass personalisierte Empfehlungen nicht mehr über Kategorien (orientiert sich am Ausflugsziel), sondern über die Nutzerbedürfnisse (orientiert sich am Nutzer) zum Beispiel über Spannungsfelder wie Aktivitätsintensität, Sozialform oder geistige Reize abgeleitet werden.

Ein solches Tool sollte den Nutzer in wenigen Schritten abholen, anstatt ihn mit komplexen Filtern zu konfrontieren. Dadurch entsteht ein erheblicher Mehrwert – sowohl für die Nutzerfreundlichkeit als auch für die Qualität der Ergebnisse. Gleichzeitig eröffnet dies für den Auftraggeber (fiktiv: Schweiz Tourismus) die Chance, ein innovatives, marktunterscheidendes System bereitzustellen, das von einem globalen Trend zur stärkeren Personalisierung gestützt wird.

3.4 Anforderungen an das Projekt

3.4.1 Anforderungen aus den gesetzten Zielen (funktional)

Nr.	Typ	Anforderung	Bezug
F1	funktional	Das System muss Eingaben des Nutzers zu mindestens drei Spannungsfeldern (z. B. Aktivitätsintensität, Sozialform, Geistige Reize) erfassen können.	Ziel 1, 2
F2	funktional	Das System muss anhand dieser Eingaben passende Ausflugsziele aus einer gespeicherten Liste auswählen.	Ziel 1
F3	funktional	Das System muss die Ergebnisse als konkrete Ausflugsziele ausgeben (mind. 1–3 Vorschläge).	Ziel 1
F4	funktional	Falls keine passenden Ergebnisse gefunden werden, muss das System eine entsprechende Rückmeldung geben.	Ziel 1
F5	funktional	Die Benutzerschnittstelle muss interaktiv über die Kommandozeile laufen und die Eingabe in max. 5 Schritten ermöglichen.	Ziel 3

3.4.2 Anforderungen bezüglich der Qualität (nicht-funktional)

Nr.	Typ	Anforderung	Bezug
NF1	nicht-funktional	Die Lösung soll als funktionierender Prototyp in Form eines Konsolenprogramms vorliegen.	Ziel 4
NF2	nicht-funktional	Die Umsetzung muss mit den Kenntnissen von Lernenden im Modul realisierbar sein.	alle Ziele
NF3	nicht-funktional	Die Bedienung muss intuitiv und in max. 2 Minuten möglich sein.	Nutzen 1
NF4	nicht-funktional	Das System darf keine unklaren oder widersprüchlichen Ergebnisse anzeigen.	Nutzen 2
NF5	nicht-funktional	Die Verarbeitung soll deterministisch erfolgen (gleiche Eingaben = gleiche Ergebnisse).	Nutzen 2
NF6	nicht-funktional	Für den Prototyp werden ca. 20–40 Ausflugsziele hinterlegt, um die Empfehlungslogik demonstrieren zu können.	Ziel 1, 2
NF7	nicht-funktional	Abnahmekriterium: Die Projektabnahme gilt als erfüllt, wenn mindestens 5 Papierfragebögen erfasst wurden und Durchschnittswert dieser Fragebögen ≥ 6.0 erreicht wird.	Ziel 5

3.4.3 Anforderungen an die Validierung

Nr.	Typ	Anforderung	Bezug
V1	Validierung/Prozess	Die Zufriedenheit der Nutzer wird per Papierfragebogen nach der Nutzung erhoben (Skala 1–10).	Ziel 5
V2	Validierung/Prozess	Es werden mindestens 5 Testpersonen befragt; die Erhebung erfolgt während der Einführungsphase.	Ziel 5
V3	Validierung/Prozess	Die Auswertung (Durchschnitt, ggf. Median) erfolgt manuell (Papier/Excel) und wird im Abnahmeprotokoll dokumentiert.	Ziel 5

3.5 Lösungsvarianten

Wir haben die folgenden drei Lösungsvarianten analysiert.

Variante	Beschreibung
Command-line Tool	<p>Wir könnten unser System zur personalisierten Empfehlung von Reisezielen in der Schweiz als Command Line Interface entwickeln.</p> <p>Der Nutzer würde mit unserem System über die Kommando-Zeile interagieren. In der Kommandozeile können wir den Nutzer zu seinen Bedürfnissen befragen und so die notwendigen Daten einholen, die unser System braucht, um die Empfehlungen auf den Nutzer zu personalisieren.</p> <p>Unser System würde in C# geschrieben werden, da wir diese Sprache aktuell an der Berufsschule lernen und damit schon einige simple Kommandozeilenprogramme geschrieben haben. Das System könnte als Exe-Datei dem Auftraggeber übergeben werden, damit dieser es ebenfalls erleben und testen kann.</p>
Website	<p>Unser System könnte als Webseite entwickelt werden. Auf der Webseite könnten wir den Nutzer zu seinen Bedürfnissen befragen und so die notwendigen Daten einholen.</p> <p>Die Logik würde dann in JavaScript oder TypeScript geschrieben werden und das Interface zum Beispiel mit React.</p> <p>Dazu benötigen wir einen Webserver und eine Domain, um das System zu hosten.</p>
Fragebogen auf Papier	<p>Unser System könnte als Fragebogen auf Papier entwickelt werden. Der Nutzer würde seine Bedürfnisse im Fragebogen angeben, ähnlich wie bei einem Persönlichkeitsfragebogen.</p> <p>Danach könnte der Nutzer seine Ergebnisse aus dem Bedürfnisfragebogen in eine ausgereifte Serie von Tabellen übertragen und die entsprechenden Empfehlungen ablesen.</p> <p>Das System könnte von Nutzern entweder physisch auf Papier oder digital als einem PDF genutzt und getestet werden.</p>

3.5.1 Lösungsauswahl

Kriterien	Comandline-Tool			Webseite		Fragebogen auf Papier		
	G	E	T	E	T	E	T	
Kosteneffizienz	5	5	25	3	15	2	10	Es entstehen zusätzlich kosten für Druck, Papier und Versand. Hinzu kommen die Arbeitsstunden.
Zeitungfang	4	4	16	2	8	3	12	Wir können es schnell im Word, ohne viel programmieren erstellen. Jedoch braucht die Verteilung viel Zeit.
Benutzerfreundlichkeit	5	4	20	5	25	2	10	Einfaches Ankreuzen auf Papier, jedoch ist die Zugänglichkeit und die Auswertung nicht praktisch.
Kenntnisse vorhanden	3	5	15	3	9	4	12	Wir kennen uns gut mit Word aus, Layout, Bilder sowie der Fragebogen zu erstellen ist kein Problem.
Machbarkeit ist gewährleistet	4	4	16	3	12	3	12	Die Machbarkeit ist sehr hoch, jedoch könnten auch hier Probleme beim Versand oder beim Druck entstehen.
Ressourcenfreundlich	3	5	15	4	12	3	9	Nicht Ressourcenfreundlich, da es viel Papier und Tinte braucht. Hinzu kommt der Transport.
Total Punkte			107		81		65	

3.5.2 Begründung der gewählten Lösungsvariante

Nach der Analyse mehrerer Lösungsvarianten (Command-Line-Tool, Webseite, Fragebogen auf Papier) hat sich unser Projektteam entschieden, die Umsetzung als Command-Line-Programm in C# zu realisieren. Diese Entscheidung basiert auf einer Abwägung zentraler Kriterien wie Kosten, Zeitaufwand, Benutzerfreundlichkeit, vorhandene Kenntnisse, Machbarkeit und Ressourcenschonung. In der Gesamtbewertung erreichte diese Variante mit 107 Punkten den mit Abstand höchsten Wert und erwies sich damit als klarer Favorit gegenüber den Alternativen.

3.5.2.1 *Passung zum Projektauftrag und -umfang*

Der Auftraggeber (fiktiv: Schweiz Tourismus) erwartet im Rahmen dieses Projekts einen funktionierenden Prototyp, der die innovative Empfehlungslogik demonstriert. Ziel ist ausdrücklich nicht, eine komplette kommerzielle Plattform mit umfangreicher Datenbank, Bildmaterial oder Webhosting zu entwickeln. Das Command-Line-Tool erfüllt diese Vorgabe optimal: Es reduziert die Lösung auf ihre Kernelemente – Empfehlungslogik und Nutzerinteraktion – und vermeidet unnötigen Aufwand für Design, Hosting oder Papierorganisation. Damit bleibt das Projekt im vorgegebenen Rahmen des Moduls realistisch und gleichzeitig methodisch hochinteressant und anspruchsvoll.

3.5.2.2 *Effizienz und Kosten*

Eine Weblösung würde zusätzliche Infrastruktur erfordern, etwa Webserver, Domain und Kenntnisse in neuen Technologien. Dies hätte nicht nur Zeit-, sondern auch Kostenaufwand bedeutet. Ein Papierfragebogen wiederum hätte Materialkosten (Druck, Papier, Versand) nach sich gezogen und wäre aufwendig in der Verteilung und Auswertung. Demgegenüber entstehen beim Command-Line-Tool keine Zusatzkosten – es genügt die Nutzung unserer bestehenden Schul-Laptops und kostenloser Software wie Visual Studio. Somit ist diese Variante die klar kosteneffizienteste Lösung.

3.5.2.3 *Zeitlicher Rahmen und Realisierbarkeit*

Der Projektzeitplan ist eng bemessen. Insbesondere zwischen dem Konzeptabschluss (03.09.2025) und der Realisierung (14.10.2025) stehen nur wenige Wochen zur Verfügung. Eine Weblösung hätte uns durch fehlende Vorkenntnisse in HTML/JavaScript sowie zusätzlichen Designaufwand zeitlich überfordert. Der Papierfragebogen wäre zwar schnell erstellt, hätte jedoch viel Aufwand in der Distribution und Erhebung der Ergebnisse bedeutet. Das Command-Line-Tool konzentriert sich dagegen auf die Logik und kann innerhalb der geplanten Realisierungsphase mit unseren Kenntnissen umgesetzt werden. Damit ist die zeitliche Machbarkeit am besten gewährleistet.

3.5.2.4 *Kenntnisse und Lernzielorientierung*

Ein zentrales Kriterium bei der Lösungswahl war die pädagogische Ausrichtung des Moduls. Im Modul geht es nicht primär darum, neue Programmiersprachen oder Webframeworks zu erlernen, sondern um das korrekte Anwenden und Dokumentieren der HERMES-Projektmanagementmethode. Deshalb haben wir eine Lösungsvariante gesucht, die den Kern der Aufgabe – die Umsetzung einer Empfehlungslogik – sichtbar macht, ohne uns durch unnötige technische Komplexität abzulenken.

Das Command-Line-Tool erfüllt diese Bedingung ideal: Es erlaubt uns, die Logik der Personalisierung klar und überprüfbar zu implementieren, gleichzeitig aber den technischen Rahmen

so schlank wie möglich zu halten. Wir müssen uns nicht in zusätzliche Technologien wie JavaScript, Webserver oder Hosting einarbeiten, die mit den Lernzielen dieses Moduls wenig zu tun haben und vor allem Zeit beanspruchen würden. Auch eine Papierlösung wäre zwar einfach umzusetzen, würde aber die wesentliche IT-Komponente völlig ausklammern.

Durch das Command-Line-Tool können wir also ein funktionierendes Produkt entwickeln, das als Prototyp die Funktionsweise des Empfehlungssystems demonstriert – und zugleich bleibt uns ausreichend Zeit, die methodischen Aspekte (Projektphasen, Rollen, Ergebnisse, Meilensteine, Dokumentationen) konsequent nach HERMES umzusetzen. Damit ist diese Variante diejenige, die pädagogisch am besten zum Modul passt, weil sie die Balance zwischen inhaltlicher Machbarkeit und methodischem Fokus herstellt.

3.5.2.5 Benutzerfreundlichkeit und Testbarkeit

Obwohl eine Kommandozeile optisch weniger ansprechend wirkt als eine Website, ist sie für unser Ziel vollkommen ausreichend. Der Nutzer kann seine Präferenzen in maximal fünf Schritten eingeben und erhält direkt konkrete Empfehlungen. Für die Abnahme ist definiert, dass mindestens fünf Nutzerfragebögen mit einer durchschnittlichen Zufriedenheit von ≥ 6.0 erfasst werden müssen. Auch dieser Validierungsschritt ist mit einem Konsolenprogramm leicht umsetzbar, da die Ergebnisse klar und deterministisch angezeigt werden und anschließend einfach mit Papierfragebögen bewertet werden können.

3.5.2.6 Ressourcenschonung und Nachhaltigkeit

Eine Papierlösung hätte den Nachteil, dass viel Material verbraucht wird (Papier, Druckertinte, Transport). Eine Weblösung bräuchte dauerhafte Infrastruktur (Server), die für ein Schulprojekt überdimensioniert wäre. Das Command-Line-Tool benötigt lediglich Strom und unsere Laptops – es ist damit die ressourcenschonendste Variante und entspricht gleichzeitig den heutigen Erwartungen an nachhaltige Projektarbeit.

3.5.2.7 Methodische Begründung im Sinne von HERMES

Nach HERMES wird die Wahl der Lösungsvariante fachlich und vorgehensbezogen begründet. Mit dem von uns gewählten Ansatz erfüllen wir beide Anforderungen:

- **Fachlich:** Das Tool konzentriert sich auf die Kernfrage der Empfehlungssysteme – wie aus Nutzerpräferenzen passende Ziele berechnet werden können.
- **Vorgehensbezogen:** Die Umsetzung als Prototyp in C# ist für ein kleines Team mit beschränkten Ressourcen realistisch und führt zu einem prüfbareren Ergebnis im Sinne der Ziel- und Nutzenvorgaben.

Damit ist das Command-Line-Tool diejenige Variante, die unter Abwägung aller Kriterien am besten geeignet ist, die Projektziele methodisch sauber und innerhalb des Projektumfangs zu erreichen.

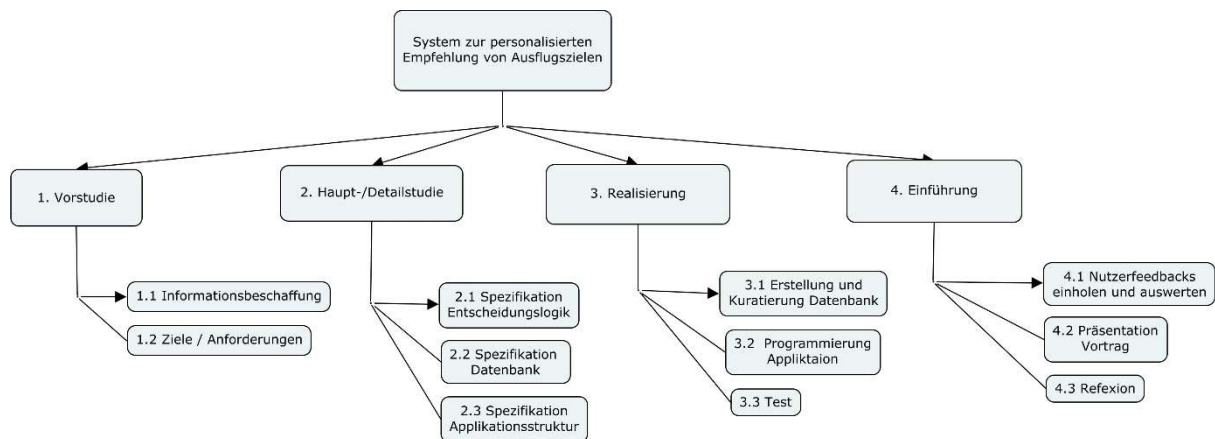
3.6 Zeitplan und Projektstrukturplan

3.6.1 Zeitplan

Unser Zeitplan bildet die Grundlage für die Steuerung und Überwachung des Projekts. Er zeigt die Abfolge der einzelnen Projektphasen gemäss der HERMES-Methode (Initialisierung, Konzept, Realisierung, Einführung und Abschluss) in chronologischer Ordnung. Innerhalb der Phasen sind die zentralen Arbeitspakete mit ihren jeweiligen Terminen sowie den dazugehörigen Meilensteinen aufgeführt. Dadurch wird sichergestellt, dass die Abhängigkeiten zwischen den Aufgaben klar erkennbar sind und die Projektfortschritte kontinuierlich überprüft werden können.

Unser Projektzeitplan liegt in einer separaten Excel-Tabelle vor, welche unsere zeitliche Planung strukturiert darstellt und zusammen mit der vorliegenden Projektdokumentation eingereicht wird.

3.6.2 Projektstrukturplan



3.7 Organisation

3.7.1 Kick-Off-Meeting

Das Kick-Off-Meeting markierte den offiziellen Start unseres Projekts und diente dazu, alle Teammitglieder auf einen gemeinsamen Stand zu bringen. Im Rahmen des Treffens wurden verschiedene Projektideen gesammelt und diskutiert. Zur Sprache kamen unterschiedliche Ansätze, die wir im Rahmen des Moduls umsetzen könnten. In dieser Phase war auch die Idee eines Systems zur Empfehlung von Ausflugszielen bereits dabei.

Durch den Austausch im Team konnten die Stärken und Schwächen der einzelnen Ideen abgewogen werden. Dabei zeigte sich, dass die Empfehlung von Ausflugszielen ein besonders spannendes und gleichzeitig realisierbares Projekt darstellt: einerseits, weil es eine klare Marktlücke gibt, andererseits, weil die Umsetzung in einem überschaubaren Prototypen gut in den gegebenen Rahmen passt.

Ein weiterer Schwerpunkt des Treffens war die Organisation der Zusammenarbeit. Die Rollen im Team wurden verteilt, die Kommunikationswege festgelegt und der Umgang mit Dokumentation und offenen Punkten vereinbart. Damit wurde sichergestellt, dass alle Mitglieder ein gemeinsames Verständnis für Projektidee, Ziele und Vorgehen haben.

Das Kick-Off-Meeting schuf somit Klarheit, eine gute Basis für das Engagement aller Mitarbeiter*innen und die Grundlage für die strukturierte Umsetzung des Projekts.

3.7.2 Projektorganisation

Organigramm	<pre> graph TD AG["Auftraggeber: Thomas Jäggli Gibb"] --- A["Ausschluss: Thomas Jäggli Gibb"] AG --- C["Coach: Thomas Jäggli Roman Berger"] AG --- PL["Projektleiter: Jonas Vetsch"] PL --- SV["Stv. Projektleiter: Adrian Herbst"] SV --- M1["Mitarbeiter: Marco Saez"] SV --- M2["Mitarbeiter: Oleksandr Kolomiets"] </pre> <table border="1"> <thead> <tr> <th>Person</th> <th>Name</th> <th>E-Mail</th> <th>Telefon</th> </tr> </thead> <tbody> <tr> <td>Projektleiter</td> <td>Jonas Vetsch</td> <td>jve161514@stud.gibb.ch</td> <td>+41 79 890 88 66</td> </tr> <tr> <td>Stv. Projektleiter</td> <td>Adrian Herbst</td> <td>ahe158459@stud.gibb.ch</td> <td>+41 79 859 41 19</td> </tr> <tr> <td>Mitarbeiter</td> <td>Oleksandr Kolomiets</td> <td>oko161366@stud.gibb.ch</td> <td>+41 79 835 38 91</td> </tr> <tr> <td>Mitarbeiter</td> <td>Marco Saez</td> <td>msa163277@stud.gibb.ch</td> <td>+41 79 397 99 60</td> </tr> </tbody> </table>	Person	Name	E-Mail	Telefon	Projektleiter	Jonas Vetsch	jve161514@stud.gibb.ch	+41 79 890 88 66	Stv. Projektleiter	Adrian Herbst	ahe158459@stud.gibb.ch	+41 79 859 41 19	Mitarbeiter	Oleksandr Kolomiets	oko161366@stud.gibb.ch	+41 79 835 38 91	Mitarbeiter	Marco Saez	msa163277@stud.gibb.ch	+41 79 397 99 60
Person	Name	E-Mail	Telefon																		
Projektleiter	Jonas Vetsch	jve161514@stud.gibb.ch	+41 79 890 88 66																		
Stv. Projektleiter	Adrian Herbst	ahe158459@stud.gibb.ch	+41 79 859 41 19																		
Mitarbeiter	Oleksandr Kolomiets	oko161366@stud.gibb.ch	+41 79 835 38 91																		
Mitarbeiter	Marco Saez	msa163277@stud.gibb.ch	+41 79 397 99 60																		
Auftraggeber	Der Auftraggeber ist Thomas Jäggli und die Gewerblich-Industrielle Berufsfachschule Bern (gibb).																				
Projektleiter	Der Projektleiter ist Jonas Vetsch. Jonas Vetsch übernimmt die Koordination mit dem Auftraggeber, sowie die Organisation des Teams.																				
Stv Projektleiter	Der Stellvertretende Projektleiter ist Adrian Herbst. Adrian Herbst vertritt Jonas Vetsch, sollte dieser abwesend sein. Ansonsten hat er dieselbe Funktion wie ein Projektmitglied.																				
Projektmitglieder	Die Projektmitglieder sind Adrian Herbst, Oleksandr Kolomiets und Marco Saez. Sie übernehmen diverse Aufgaben, welche zur Erfüllung des Projekts benötigt werden.																				
Projektausschuss	Der Projektausschuss ist Thomas Jäggli und die Gewerblich-Industrielle Berufsfachschule Bern (gibb).																				

3.7.3 Rollen

Unser **Projektleiter** ist Jonas, mit seiner breiten Lebenserfahrung ist er von uns allen dafür am besten geeignet. Jonas ist ein organisierter Mensch und arbeitet exakt. Er ist ambitioniert und hat einen guten Zugang zum ganzen Team auf einer menschlichen Ebene. Jonas meldete sich freiwillig, da er einen praktischen Erfahrungsgewinn im Leiten des Projekts sieht, welcher ihm auch im Berufsleben weiterhelfen wird.

Adrian ist der **Stellvertreter** von Jonas. Sollte dieser Abwesend sein übernimmt er die Verantwortung. Adrian ist dafür geeignet, da er auch Erfahrungen an der Universität sammeln durfte, exakt arbeitet und auf menschlicher Ebene gut mit dem Team umgehen kann. Im Normalfall arbeitet er als normaler Mitarbeiter.

Marco, Oleksandr und Adrian sind **Mitarbeiter**. Sie bilden den Grundstein unserer Arbeit, da sie von Jonas gesetzte Aufträge erfüllen und Deadlines gut einhalten. Die Aufträge erfüllen sie exakt, zielorientiert und zu verlässlich, was eine gute Zusammenarbeit erst möglich macht.

3.8 Pendenzenliste

3.8.1 Phase Initialisierung (bis 29.08.2025)

- Stakeholder-Liste erstellen (Projektteam, Auftraggeber, Lehrpersonen)
- Soll-/Ist-Analyse erstellen
- Nutzen/Ziele/Anforderungen definieren
- Lösungsvarianten analysieren und bewerten
- Entscheid „Lösungsvariante“ dokumentieren
- Projektmanagementplan grob entwerfen (Aufwand, Zeitplan, Rollen)
- Projektantrag fertigstellen und einreichen
- Vorstudie fertigstellen und einreichen

3.8.2 Phase Konzept (bis 03.09.2025)

- Situationsanalysen detaillieren (technisch, organisatorisch)
- Lösungsanforderungen präzisieren (inkl. 3 Spannungsfelder)
- Systemkonzept/Architektur definieren
- Wireframe / grober Ablauf (CLI-Interaktion) erstellen
- Entscheid „Lösungsarchitektur“ treffen und dokumentieren

3.8.3 Phase Realisierung (16.09.–14.10.2025)

- Prototyp (Commandline-Tool in C#) entwickeln
- Datenbank mit 20–40 Ausflugszielen erstellen, kuratieren und anbinden
- Empfehlungslogik implementieren (Spannungsfelder → Matching)
- Tests vorbereiten (Testkonzept, Testfälle definieren)
- Interne Tests durchführen und dokumentieren (Testprotokoll)
- Produktdokumentation verfassen (Codebeschreibung, Architektur)
- Vorabnahme im Projektteam durchführen

3.8.4 Phase Einführung (21.–28.10.2025)

- Einführungskonzept konkret umsetzen (Fragebögen drucken, Testablauf planen)
- Nutzertests mit mind. 5 Personen durchführen
- Papierfragebögen einsammeln und auswerten (Excel, Durchschnitt ≥ 6.0)

- Ergebnisse im Protokoll dokumentieren
- Präsentation/Vortrag vorbereiten und durchführen (21. oder 28.10.2025)

3.8.5 Phase Abschluss (bis 05.11.2025)

- Reflexion / Realisierungsbericht erstellen
- Projekterfahrungen dokumentieren (Lernpunkte, Verbesserungen)
- Projektschlussbeurteilung erstellen
- Projektorganisation formell auflösen
- Projektabschluss dokumentieren und abgeben

4 Konzeptphase

4.1 Zielgruppe (Nutzerskizze)

Das System zur Empfehlung von Ausflugszielen richtet sich an Einzelpersonen sowie Gruppen, die entweder in der Schweiz wohnen oder sich vorübergehend dort aufhalten. Dies können beispielsweise Tagestouristen aus dem Inland, Wochentouristen aus dem Ausland, Geschäftsreisende oder Austauschstudierende sein.

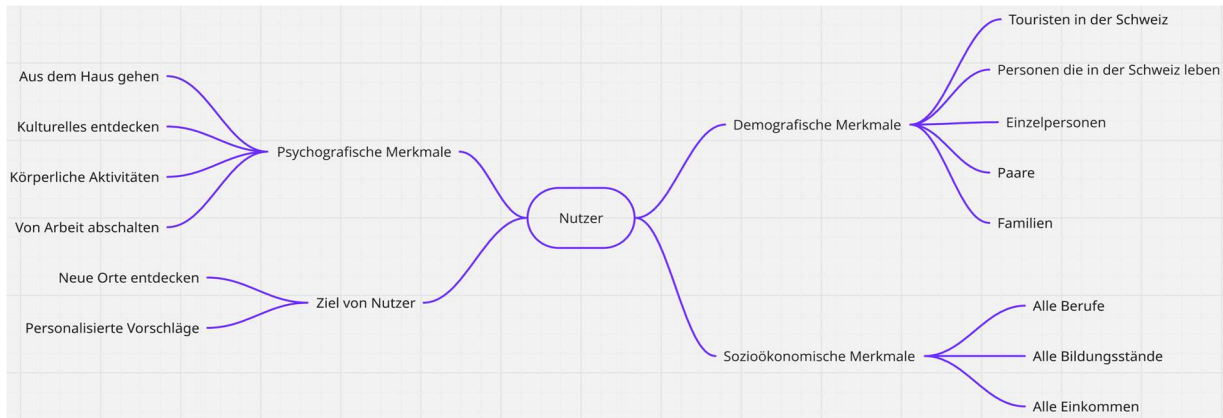


Abbildung 1 Im Rahmen der Analyse der Zielgruppe ist dieses Mindmap entstanden.

Unser System bietet dem Nutzer personalisierte Inspiration für Ausflüge und Freizeitaktivitäten: Ziel ist es, den Nutzern neue Erlebnisse zu ermöglichen, die seinen Interessen und Vorlieben entsprechen.

Die Zielgruppe zeichnet sich durch ein generelles Interesse daran aus, ihre Freizeit aktiv zu gestalten, Neues zu entdecken und dem Alltag für eine gewisse Zeit zu entfliehen. Dabei spielt es keine Rolle, ob es sich um Naturfreunde, Kulturliebhaber, Sportbegeisterte oder Erholungssuchende handelt. Das System soll bewusst so konzipiert sein, dass es für eine breite Zielgruppe geeignet ist. Es ist unabhängig von Bildungsgrad, Beruf oder Einkommen nutzbar und spricht Menschen aus allen gesellschaftlichen Schichten an. Die Vielfalt der empfohlenen Ausflugsziele, welche von kostenlosen Wanderungen bis hin zu kostenpflichtigen Museumsbesuchen reicht, trägt zur inklusiven Nutzung bei.

Einzigste Voraussetzung für die Nutzung des Systems ist die Bereitschaft, sich auf neue Erfahrungen einzulassen und Interesse daran zu zeigen, unbekannte Orte zu entdecken. Damit die Empfehlungen möglichst passgenau auf die persönlichen Bedürfnisse zugeschnitten werden können, sollten die Nutzer offen dafür sein, einige Fragen zu ihren Interessen zu beantworten. Dazu gehören beispielsweise Angaben zum bevorzugten Aktivitätsniveau (1 = wenig aktiv bis 10 = sehr aktiv), sowie zum kulturellen Interesse. Auch wird die Präferenz zu Indoor- oder Outdooraktivität erfragt. Auf Basis dieser Informationen kann das System individuell abgestimmte Vorschläge generieren, die ein hohes Mass an Relevanz und Zufriedenheit für den Nutzer und sein aktuelles, persönliches Bedürfnis versprechen.

Um die individuellen Bedürfnisse unserer Nutzer tiefgreifender verstehen zu können, haben wir aus unseren bisher gewonnenen Erkenntnissen die folgenden vier Personas definiert. Die Personendaten sind dabei fiktiv gewählt.

Die vier Personas des Systems zur personalisierten Empfehlung von Ausflugszielen

Personendaten		Freizeitverhalten	Ziele und Bedürfnisse
1. Familienvater			
Name	Markus Schneider	<ul style="list-style-type: none"> • Unternehmungen mit der Familie am Wochenende • Wert auf Abwechslung und Natur 	<ul style="list-style-type: none"> • Kinderfreundliche Ausflugsziele in der Nähe • Kombination aus Aktivität und Erholung • Geringer Organisationsaufwand
Alter	45		
Wohnort	Bern		
Familienstand	Verheiratet, zwei Kinder (8 und 12 Jahre)		
Beruf	IT-Projektleiter		
Einkommen	Mittel-Hoch		
2. Person im Ruhestand			
Name	Urs Huber	<ul style="list-style-type: none"> • Besuch von Museen • Stadtführungen • historische Orte 	<ul style="list-style-type: none"> • Kulturell wertvolle Ausflüge • Barrierefreie Orte und geringe Aktivität • Zeitlich flexible Angebote unter der Woche
Alter	68		
Wohnort	Lugano		
Familienstand	Verwitwet		
Beruf	Pensionierter Lehrer		
Einkommen	Durchschnittliche Rente		
3. Austauschstudentin			
Name	Paula Silva	<ul style="list-style-type: none"> • Mit Freunden unterwegs • Liebt Wanderungen und spontane Trips 	<ul style="list-style-type: none"> • Günstige oder kostenlose Ausflugsziele • Fokus auf Natur und Abenteuer • Orientierung in einer neuen Umgebung
Alter	23		
Wohnort	Zürich (Austauschstudentin aus Portugal)		
Familienstand	Ledig		
Studiengang	Sozialwissenschaften		
Einkommen	Gering		
4. Ein Paar ohne Kinder			
Namen	Nicole Moser Daniel Meyer	<ul style="list-style-type: none"> • Wochenendausflüge • Food- & Kulturinteresse • Natur als Ausgleich zum Job 	<ul style="list-style-type: none"> • Inspiration für spontane Kurztrips • Kombination aus Kulinarik, Natur und Kultur • Empfehlungen abseits der Touristenmassen
Alter	33 und 35		
Wohnort	Lausanne		
Familienstand	In Beziehung, kinderlos		
Berufe	Marketingmanagerin Landschaftsgärtner		
Einkommen	Mittel-Hoch		

4.2 Layout (Aussenskizze)

Für Schweiz Tourismus erstellen wir ein Prototyp, wo Benutzer bequem ein Ausflugsziel in der Schweiz finden können. Dieser Prototyp schreiben wir als Konsolenprogramm, daher sieht er etwas rudimentär aus. Sollte das Projekt weitergezogen werden und ein Programm für den End-User geschrieben werden, wird dies ein schön designten Interface haben.

Die folgende Aussenskizze veranschaulicht den grundlegenden Ablauf unseres Programms aus Nutzersicht.

```

Microsoft Visual Studio-Debu x + v
-----
1.) Guten Tag!
Ich bin ein Programm, welches dir hilft, dein optimales Ausflugsziel in der Schweiz zu finden!
Dafür habe ich dir drei Fragen, welche du zwischen 1 und 10 beantworten musst!
Lass und doch direkt loslegen!
-----
2.) Kulturell: Wie wichtig ist dir die Kulturelle Komponente des Ausflugsziels ?
(1 = unwichtig | 10 = sehr wichtig)
|10|
-----
3.) Aktiv: Wie sehr möchtest du körperlich aktiv sein?
(1 = wenig aktiv | 10 = sehr aktiv)
|0|
-----
4.) In-/Outdoor: Möchtest du am Ausflugsziel lieber drinnen oder draussen sein?
(1 = nur indoor | 5 = egal | 10 = nur Outdoor)
|10|
-----
5.) Vielen Dank für deine Eingabe!
Auf dich passt am besten das Ausflugsziel Bern: Kino im Kocher!
-----
6.) Gefällt dir das Ausflugsziel nicht? Dann gib ein belibigen Buchstaben ein, um noch mehr Vorschläge zu erhalten
-----
    
```

Abbildung 2 Die Aussenskizze

Nr.	Beschreibung
1	Das Programm begrüsst den Nutzer und erklärt ihm, wie es zu benutzen ist.
2	Erstes Spannungsfeld Kulturell, Nutzer gibt an wie wichtig ihm Kultur ist.
3	Zweites Spannungsfeld aktiv, Nutzer gibt an wie körperlich aktiv er sein will.
4	Drittes Spannungsfeld In-/Outdoor, Nutzer gibt an, ob er lieber draussen oder drinnen ist.
5	Ausgabe, das Programm gibt dem Nutzer das passendste Ausflugsziel.
6	Das Programm bietet zusätzliche Ausflugsziele, sollte der Nutzer mit dem Vorschlag nicht zufrieden sein.

4.3 Bauplan / Installation (Innenskizze)

4.3.1 Innenskizze

Zunächst werden die möglichen Ausflugsziele (für unser Projektraum ca. 40 Ausflugsziele) in einer Datenbanktabelle gespeichert. Jedes Ziel wird durch die drei Attribute *howCulture*, *howActive* und *howOutdoor* beschrieben, die Werte zwischen 1 und 10 annehmen können.

Aus dieser Datenbank werden die Daten in das C# Programm importiert, sodass jedes Ausflugsziel in Form eines Place-Objekts vorliegt. Diese Klasse speichert die genannten Attribute und dient später als Grundlage für den Empfehlungsalgorithmus.

Parallel dazu gibt es die Klasse User, welche die Eingaben des Benutzers aufnimmt. Hier beantwortet der Benutzer die drei Fragen zu seinen Vorlieben in Bezug auf Kultur, Aktivität und Indoor/Outdoor. Neben den eigentlichen vom Benutzer eingegebenen Werten werden zusätzlich Koeffizienten berechnet. Standardmässig beträgt dieser Koeffizient 1,0. Falls der Benutzer jedoch eine besonders hohe (≥ 9) oder besonders niedrige (≤ 2) Bewertung abgibt, wird die Frage als für ihn besonders wichtig gewertet und damit der Koeffizient auf 1,5 gesetzt.

Sobald die Eingaben vorliegen, tritt der Empfehlungsalgorithmus in Kraft. Für jedes Place-Objekt wird nun der Wert *howUnsuitable* berechnet. Dieser ergibt sich aus der Differenz zwischen den Benutzerangaben und den Werten des jeweiligen Ortes, wobei für jede der drei Fragen die Differenz noch mit dem Koeffizienten (KF) gewichtet wird:

$$Place.\text{howUnsuitable} =$$

$$\begin{aligned} & | User.\text{howCulture} - Place.\text{howCulture} | \times User.\text{howCultureKF} + \\ & | User.\text{howActive} - Place.\text{howActive} | \times User.\text{howActiveKF} + \\ & | User.\text{howOutdoor} - Place.\text{howOutdoor} | \times User.\text{howOutdoorKF} \end{aligned}$$

Nachdem für alle Orte der jeweilige *howUnsuitable*-Wert berechnet wurde, werden die Ergebnisse miteinander verglichen. Derjenige Ort, der den niedrigsten *howUnsuitable*-Wert aufweist, ist der passendste Vorschlag für den Benutzer. Somit wird gewährleistet, dass die Empfehlung genau auf die Präferenzen des Nutzers abgestimmt ist.

Nachfolgend findet sich eine schematische Darstellung des Algorithmus, zusammen mit einer Erklärung zur Darstellung.

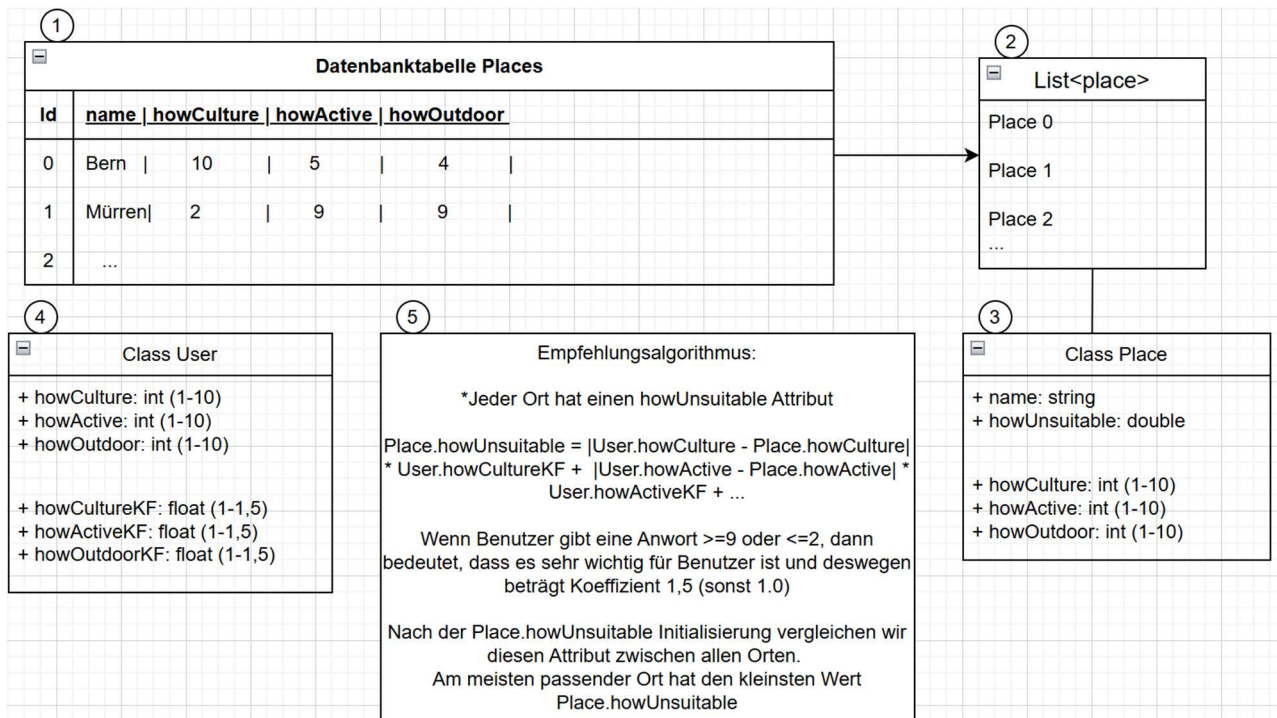


Abbildung 3 Innenskizze: Schematische Darstellung des Algorithmus

Nr.	Elementname	Beschreibung
1	Datenbankta- belle	Visuelle Darstellung unserer Datenbankstruktur. Jede Entität ist durch drei Attribute beschrieben – <i>howCulture</i> , <i>howActive</i> , <i>howOutdoor</i> . Diese haben einen Wertebereich von 1 bis 10.
2	Liste von Orten	Datenstruktur im Programm, in welche die Orte aus der Datenbank importiert werden.
3	Class Place	Zeigt die Daten, die ein Ort-Objekt ausmachen. Mit diesen Werten wird unser Empfehlungsalgorithmus arbeiten.
4	Class User	Zeigt die Daten, die vom Nutzer in der Applikation eingegeben werden. Die Koeffizienten werden standardmässig mit dem Wert 1.0 initialisiert. Gibt der Benutzer für ein Spannungsfeld eine Antwort ≥ 9 oder ≤ 2 , so werten wir dies als besonders wichtig oder besonders unwichtig. Dadurch wird der Koeffizient auf 1,5 erhöht. Damit sind alle Information vorhanden, die der Empfehlungsalgorithmus benötigt.
5	Empfehlungs- algorithmus	Jeder Ort hat ein Attribut <i>howUnsuitable</i> . Dieses Attribut ist die Summe von Differenzen zwischen den Nutzereingaben und den aus der Datenbank importierten Werten zu den Orten. Jede Differenz wird noch mit dem entsprechenden Koeffizienten multipliziert. Nach der Initialisierung von <i>Place.howUnsuitable</i> vergleichen wir alle Orte nach diesem Attribut. Der passendste Ort hat den kleinsten Wert <i>Place.howUnsuitable</i> .

4.3.2 Visualisierungshilfe für den Algorithmus (VAPEAS)

Im Anschluss an die Entwicklung unseres Empfehlungsalgorithmus kam in der Projektgruppe das Bedürfnis auf, den Algorithmus zu testen. Dafür haben wir mit Microsoft Excel ein «Programm» entwickelt. Diese Visualisierungshilfe für unseren Algorithmus nennen wir «VAPEAS» (Visualisierungshilfe für den Algorithmus zur personalisierten Empfehlung von Ausflugszielen in der Schweiz).

VAPEAS können Sie, werter Leser, selbst nutzen und damit die Einzelheiten unseres Algorithmus erleben. Verglichen mit dem Endprodukt unserer Arbeit gibt Ihnen VAPEAS zwar nur sehr eingeschränkte Empfehlungen, lässt Sie dafür aber viel genauer die Funktionsweise unseres Algorithmus erleben. Die Exceldatei liegt dieser Projektdokumentation bei.

4.3.2.1 Die Funktion von VAPEAS

Im VAPEAS gibt der Nutzer in der Zeile «Nutzereingabe» bei jedem der drei Spannungsfeldern (howCulture, howActive, howOutdoor) eine Zahl von 1-10 ein. Dadurch wird festgelegt, wie viel Kulturanteil und wie viel körperliche Aktivität der Nutzer bei seinem Ausflugsziel möchte. Zudem wissen wir dadurch, ob der Nutzer am Ausflugsziel lieber draussen oder drinnen (oder eine Mischform) sein möchte.

VAPEAS hat drei Ausflugsziele standardmässig verbaut: Interlaken Selfiemuseum, Bern Foxtrail und Adelboden Wanderung Engstligenalp. Diese drei werden sofort ausgewertet, sobald der Nutzer seine Bedürfnisse eingegeben hat. Im Beispiel, das nachfolgend abgebildet ist, passt das Ziel Interlaken Selfiemuseum am besten zum Nutzer. Dies ist an der grünen Farbe im Bereich «Rating» und der tiefen Zahl dieses Ratings erkennbar.

Das Rating wird nach unserem Algorithmus berechnet. Der Algorithmus wird im Kapitel 4.3 Bauplan / Installation (Innenskizze) genauer beschrieben. Das Resultat des Algorithmus ist eine Zahl, die die Passung eines jeden Ausflugsziels zum abgefragten Nutzerbedürfnis beschreibt. Passt das Ziel gut, so ist der Wert klein. Passt das Ziel nicht so gut, so ist der Wert gross. Je kleiner der Wert also, desto besser passt das Ziel zum eingegebenen Nutzerbedürfnis. Bei einer exakten Übereinstimmung von Nutzerbedürfnis und Ausflugsziel ist der Wert des Ratings 0.

In VAPEAS können zu den vorgegebenen drei Reisezielen noch drei weitere, eigene Ziele definiert werden. Diese werden dann ebenfalls ausgewertet, sobald der Nutzer sie mit einer Zahl für die Attribute howCulture, howActive und howOutdoor charakterisiert.

Visualisierungshilfe für den Algorithmus (zur personalisierten Empfehlung von Ausflugszielen in der Schweiz), kurz: VAPEAS					
	Variablen				Rating
	name [string]	howCulture [int]	howActive [int]	howOutdoor [int]	howUnsuitable [int]
Nutzereingabe		9	5	6	
Koeffizient >= 9 ? 1.5 : 1		<i>erhöht (1.5)</i>	<i>normal (1)</i>	<i>normal (1)</i>	
<i>Daten Ausflugsziel aus DB</i>	Interlaken Selfiemuseum	5	5	1	
Differenz zur Nutzereingabe		4	0	5	
Koeffizient		1.5	1	1	11
<i>Daten Ausflugsziel aus DB</i>	Bern Foxtrail	3	8	10	
Differenz zur Nutzereingabe		6	3	4	
Koeffizient		1.5	1	1	16
<i>Daten Ausflugsziel aus DB</i>	Adelboden Wanderung Engstligenalp	3	8	9	
Differenz zur Nutzereingabe		6	3	3	
Koeffizient		1.5	1	1	15
<i>Imaginäres Test-Ausflugsziel</i>	Per Zufall genau der Nutzereingabe	9	5	6	
Differenz zur Nutzereingabe	entsprechend	0	0	0	
Koeffizient		1.5	1	1	0
<i>Imaginäres Test-Ausflugsziel</i>	Extreme Abweichung von der Nutzereingabe	1	1	1	
Differenz zur Nutzereingabe		8	4	5	
Koeffizient		1.5	1	1	21
<i>Dein Test-Ausflugsziel</i>	Erfinde dein eigenes Ausflugsziel				
Differenz zur Nutzereingabe	und teste mit eigenen Werten!				
Koeffizient		1.5	1	1	
<i>Dein Test-Ausflugsziel</i>	Erfinde dein eigenes Ausflugsziel				
Differenz zur Nutzereingabe	und teste mit eigenen Werten!				
Koeffizient		1.5	1	1	
<i>Dein Test-Ausflugsziel</i>	Erfinde dein eigenes Ausflugsziel				
Differenz zur Nutzereingabe	und teste mit eigenen Werten!				
Koeffizient		1.5	1	1	

Abbildung 4 VAPEAS: Visualisierungshilfe für unseren Algorithmus

4.3.2.2 Reflexion zu VAPEAS

Wie beschrieben war das Ziel von VAPEAS, dass sich die Projektmitglieder ein Bild von der Funktionalität des Algorithmus machen konnten. Damit wollten wir sehen, ob der Algorithmus sinnvolle Ergebnisse liefert und ob wir «aus dem Bauchgefühl heraus» mit den Ergebnissen des Algorithmus einverstanden sind. Am besten konnte dies getestet werden, indem Projektmitglieder eigene, tatsächliche Ausflugsbedürfnisse eingaben und dazu drei eigene Ziele einpfligten. Dies führte zu sehr akkuraten Empfehlungen.

Die Entwicklung und Nutzung von VAPEAS war für die Projektmitglieder eine sehr gute Erfahrung. Dank VAPEAS haben wir ein starkes Vertrauen in den von uns entwickelten Algorithmus und sind sehr zuversichtlich, dass auch andere Nutzer sehr gute personalisierte Empfehlungen von unserem System bekommen werden.

4.4 Funktionalitäten (Ablaufskizze)

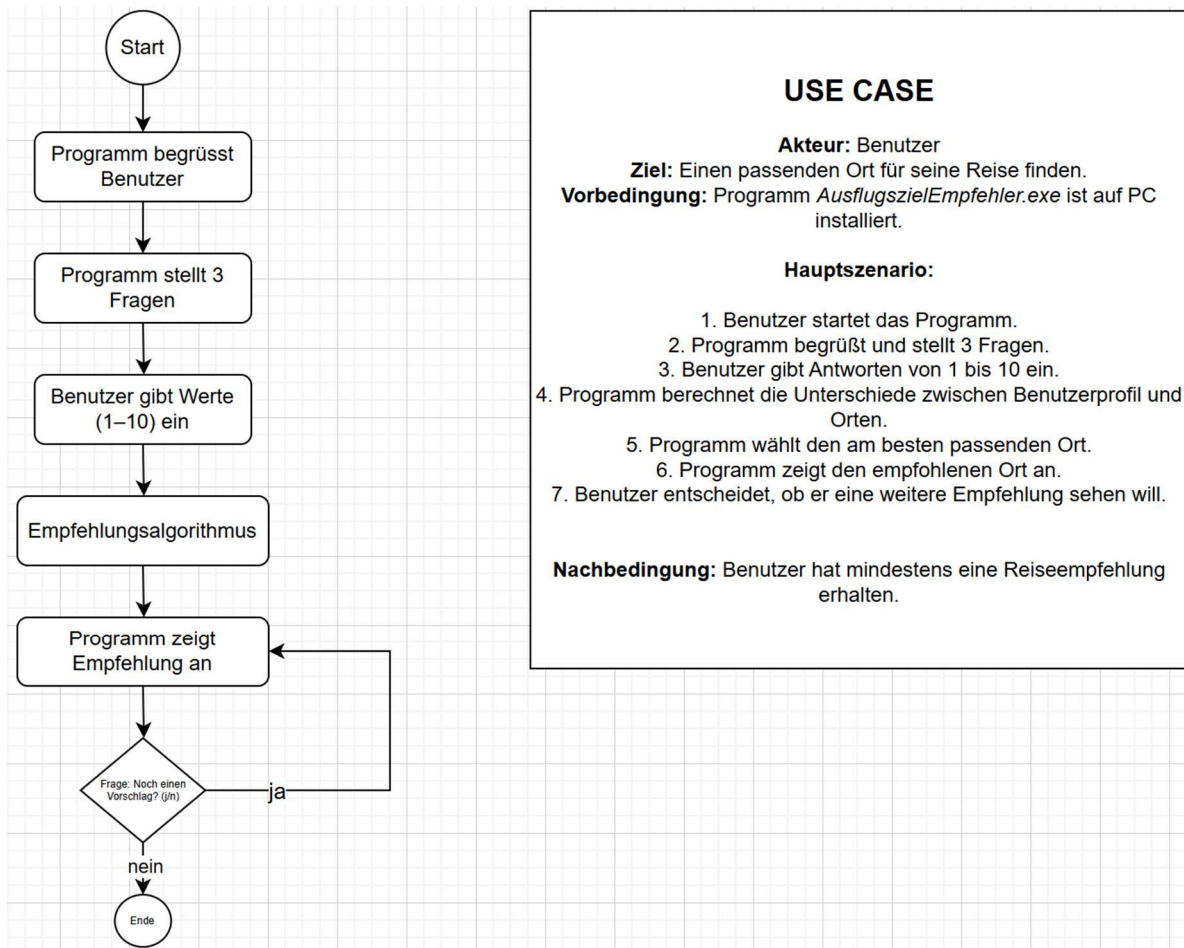


Abbildung 5 Ablaufskizze und Use Case

Der Benutzer startet das Programm (.exe-Datei) auf seinem PC, um ein geeignetes Reiseziel zu finden. Nachdem das Programm gestartet ist, wird der Benutzer begrüßt und ihm werden drei Fragen gestellt. Die Fragen dienen dazu, die Präferenzen des Benutzers in Bezug auf sein Ausflugsziel zu erfassen.

Der Benutzer beantwortet diese Fragen, indem er für jede einen Wert zwischen 1 und 10 eingibt. Auf Grundlage dieser Eingaben führt das Programm einen Empfehlungsalgorithmus aus. Dabei werden die Angaben des Benutzers mit den hinterlegten Eigenschaften der möglichen Ausflugsziele verglichen.

Anschliessend berechnet das Programm das Ausflugsziel, das am besten zum Nutzer passt. Das Programm präsentiert dieses als Empfehlung auf dem Bildschirm. Nach der ersten Empfehlung hat der Benutzer die Möglichkeit, das Ausflugsziel mit der zweitbesten, drittbesten etc. Passung anzuzeigen. Diese hat das Programm bereits berechnet, da es zum Auffinden des am besten passenden Ziels bereits alle verglichen hat.

Der Use Case endet, sobald der Benutzer keine weiteren Empfehlungen mehr sehen möchte. Als Nachbedingung hat der Benutzer mindestens eine konkrete Reiseempfehlung erhalten.

4.5 Sachmittel (Materialliste)

Für die Umsetzung des Projekts steht jedem Teammitglied ein eigener Laptop zur Verfügung. Eingesetzt werden zwei Lenovo ThinkPad E14 Gen 6, ein Asus Zenbook 14 sowie ein Lenovo ThinkPad P15 Gen 1. Alle Geräte laufen unter Windows 11 und gewährleisten damit eine einheitliche Arbeitsumgebung.

Zur Entwicklung des Prototyps wird **Visual Studio Community Edition 2022** als zentrale Entwicklungsumgebung genutzt. Für die Verwaltung und Bearbeitung der hinterlegten Ausflugsziele kommt **DB-Browser for SQLite** zum Einsatz. Ergänzend wird das **Microsoft 365 Paket** verwendet, insbesondere für die Dokumentation, Zusammenarbeit und Kommunikation innerhalb des Teams.

Für die Visualisierung und Modellierung greifen wir auf zwei bewährte Tools zurück: **Miro-Mindmaps** und **cmap Tools** zur Erstellung von Mindmaps sowie **Draw.io** zur Anfertigung von Flussdiagrammen. Diese Werkzeuge unterstützen die strukturierte Darstellung von Ideen, Entscheidungslogiken und Programmabläufen.

Mit dieser Kombination aus Hardware und Software ist eine stabile, effiziente und methodisch saubere Durchführung des Projekts gewährleistet.

Alle nachfolgend aufgelisteten Materialien sind Stand 11.09.2025 - 17.51 Uhr vorhanden und einsatzbereit.

Hardware	
Lenovo ThinkPad E14 Gen 6	Adrian Herbst
Lenovo ThinkPad E14 Gen 6	Jonas Vetsch
Asus Zenbook 14	Oleksandr Kolomiiets
Lenovo ThinkPad P15 Gen 1	Marco Saez
Software	
Betriebssystem	Windows 11
Entwicklungswerkzeuge	Visual Studio Community Edition 2022
Datenbank	DB-Browser for SQLite
Lizenzierte Software	Microsoft 365 Paket
Visualisierungssoftware	Miro-Mindmaps, cmap Tools
	Draw.io

4.6 Testfälle

In der vorliegenden Tabelle werden die Testanforderungen für das System zur personalisierten Empfehlung von Ausflugszielen in der Schweiz dargestellt. Diese beziehen sich jeweils auf die funktionalen und nicht-funktionalen Anforderungen des Systems zur Empfehlung von Ausflugszielen. Die Testanforderungen bilden die wesentlichen Merkmale und Qualitätskriterien, die für die erfolgreiche Umsetzung und Validierung des Systems erforderlich sind.

4.6.1 Funktionale Anforderungen

Nr.	Anforderung	Testanforderung
F1	Spannungsfelder	<p>Das System muss Eingaben des Nutzers zu mindestens drei Spannungsfeldern erfassen können. Wir haben verschiedene Optionen geprüft und schliesslich folgende drei Spannungsfelder als effektivste Lösung identifiziert.</p> <p>Kultur erleben: Dieses Spannungsfeld beschreibt, wie wichtig dem Nutzer die kulturelle Komponente am Ausflugsziel ist. Ist es für ein Nutzer beispielsweise besonders wichtig, dass er am Ausflugsziel Kultur erlebt, so gibt er «10» an. Somit werden eher Ziele wie Museen (bei denen Kulturgut im Vordergrund steht) empfohlen. Eine 1 ist ein nicht kultureller Ausflug, wie zum Beispiel Wasserskifahren.</p> <p>Körperlich aktiv sein: Beim Spannungsfeld aktiv geht es darum, wie viel Wert man auf körperliche Aktivität setzt. Ist es einem Benutzer wichtig wählt er die 10, dann gibt es eine Ausgabe wie Klettern oder schwere Wanderungen aus. Wird eine 1 gewählt handelt es sich um gemütliche Ausflüge ohne körperliche Aktivität zum Beispiel ein Open-Air Kino.</p> <p>Drinne oder draussen sein: Hiermit erfahren wir, ob der Nutzer am Ausflugsziel lieber drinnen oder draussen sein möchte. Eine 10 bedeutet, dass man sehr gerne draussen unterwegs ist. Es kommen dann Vorschläge wie eine Wanderung, ein Seilpark oder Open-Air Kino.</p> <p>Zudem fügen wir hier noch die Funktion ein, dass bei der Eingabe von «5» das Spannungsfeld nicht ausgewertet wird. Damit kann der Nutzer angeben, dass er bezüglich Indoor oder Outdoor keine Präferenz hat.</p>
F2	Eingaben	<p>Durch die Eingabe entstehen Variablen, welche wir mit der Bewertung unserer Ausflugsziele abgleichen. Unseren Ausflugszielen geben wir mit den gleichen Bewertungskriterien eine Wertung. Durch die Abgleichung der beiden Bewertungen findet der Algorithmus das passendste Ausflugsziel.</p> <p>Kulturell: Wie wichtig ist dir die kulturelle Komponente des Ausflugsziels? (1 = unwichtig / 10 = sehr wichtig)</p> <p>Aktiv: Wie sehr möchtest du körperlich aktiv sein?</p>

		<p>(1, wenig aktiv / 10, sehr aktiv) In-/Outdoor: Möchtest du am Ausflugsziel lieber drinnen oder draussen sein? (1, nur indoor / 5, egal / 10, nur Outdoor)</p>
F3	Konkrete Ausgabe	<p>Spezifische Aktivität: Das System muss die Ergebnisse als konkrete Ausflugsziele ausgeben. Sind mehrere Ziele per Zufall genau gleich passend, so wird eine kleine Anzahl (1-3) davon ausgegeben. Der Nutzer wird in diesem Fall informiert, dass es noch weitere Ziele gibt, die gleich passend sind. Zudem wird angegeben, wie viele weitere Ziele noch gleich passend sind. Der Nutzer kann sich diese dann anzeigen lassen.</p>
F4	Sonderfälle	<p>Als Gruppe erarbeiten wir folgende Sonderfälle, auf welche unser Programm vorbereitet sein soll.</p> <p>Keine Ausgabe: Unser Programm sollte nie kein Ergebnis herausgeben. Sollte dies aus programmiertechnischen Gründen trotzdem geschehen, ist es uns wichtig, dass dies auch angezeigt wird. Zum Beispiel «Hoppla, zu deiner Eingabe haben wir leider kein Ausflugsziel gefunden, versuche es doch noch einmal!»</p> <p>Falls Nutzer unzufrieden: Ist der Nutzer unzufrieden, hat er die Option, dass weitere Ausflugsziele angezeigt werden. Mit einer Eingabe wird das nächstpassende Ergebnis ausgegeben. Dies dient der Erfüllung der Nutzerzufriedenheit.</p> <p>In-/Outdoor = 5: Dabei handelt es sich um einen Sonderfall, wenn bei In-/Outdoor (howOutdoor) eine 5 angegeben wird. In diesem Fall wird dieses Spannungsfeld nicht gewertet. Da so ein Bewertungspunkt wegfällt, kann es sein, dass mehr als nur ein Ausflugsziel vorgeschlagen wird.</p> <p>Mehrere Antworten: Dies trifft ein, wenn mehrere Ausflugsziele gleich passend sind. In dem Fall soll unser Programm per Zufall eines auswählen und ausgeben. Es wird dem Nutzer angezeigt, dass es noch mehr passende Ausflugsziele gibt. Diese kann der Nutzer sich bei Interesse anzeigen lassen.</p>
F5	Benutzerschnittstelle	<p>Interaktive Eingabe: Wir haben eine interaktive Benutzerschnittstelle, welche über die Kommandozeile läuft. Der Nutzer beantwortet drei kurze Fragen bezüglich der Spannungsfelder mit einer Bewertung von 1-10. Als Resultat bekommt er ein Ausflugsziel, welches zu seinen Ansprüchen passt. Mit der Nutzereingabe in drei Schritten übertreffen wir sogar unsere Anforderung von maximal fünf Schritten.</p>

4.6.2 Nicht-funktionale Anforderungen (Qualität)

Nr.	Anforderung	Testanforderung
NF1	Konsolenprogramm	Wir realisieren unsere Prototypen als Konsolenprogramm. Als Programmiersprache haben wir C# gewählt. Durch unser Vorwissen in C# schätzen wir die Umsetzung als realistisch. Um dies jedoch zu garantieren, fangen wir frühzeitig mit Programmieren an. Zudem werden wir unvorhergesehene "Knacknüsse" beim Programmieren im Selbststudium zu bearbeiten und lösen.
NF2	Machbarkeit	Die Umsetzung muss mit unseren Kenntnissen im Rahmen des Modules realisierbar sein. Unser Projekt ist ein gut gewähltes, kleines Projekt. Ziel dieses Moduls ist nicht, dass programmieren und tüfteln an einem riesigen IT-Projekt. Wir wollen möglichst viel über das Management drum herum lernen. Daher haben wir unser Produkt klar eingegrenzt und realisierbar gestaltet.
NF3	Intuitiv / max. 2 Minuten	Unser Programm muss intuitiv sein, das heisst es soll ohne Erklärung vom durchschnittlichen Nutzer verwendbar sein. Dies erfüllen wir mit einem simplen ausführbaren Konsolenprogramm, welches die Eingaben entgegennimmt und auswertet. Zudem sollten die Eingabe und Auswertung nicht länger als zwei Minuten dauern. Daher haben wir uns auf drei Spannungsfelder begrenzt, dies entlastet sowohl den Nutzer als auch die Datenbank.
NF4	Keine unklaren Resultate / Widersprüche	Um zu verhindern, dass unser Programm unklare Ausflugsziele vorschlägt, müssen wir die Ausflugsziele sinnvoll und logisch mit den Spannungsfeldern vergleichen. Dies ist auch etwas Interpretation, da zum Beispiel körperlich aktiv nicht für jeden gleich ist. Um dies zu erfüllen, bewerten alle vier Personen die Ausflugsziele individuell und danach werten wir dies aus.
NF5	Deterministisch: gleiche Eingabe = gleiche Ausgabe	Unsere Ausflugsziele haben klare Werte zugeteilt, zum Beispiel ein kultureller Kino Film (Kultur: 10, Aktiv: 0 In-/Outdoor: 1). Da unser Programm die nächstbeste Lösung wählt, wird dies bei gleichen Eingaben immer dieselbe Ausgabe garantieren.
NF6	ca. 40 Ausflugsziele	Um die Empfehlungslogik zu demonstrieren haben wir uns für rund 40 Ausflugsziele entschieden. Wir werden via Internet diverse Ziele recherchieren und diese unserer Datenbank hinzufügen. Dabei ist wichtig, viele verschiedene Ausflugsziele zu finden.

NF7	Abnahmekriterium	Unser Abnahmekriterium gilt als erfüllt, wenn mindestens fünf Papierfragebögen erfasst wurden und die Bewertung mindestens ≥ 6 ist. Sollte dies nicht der Fall sein, nehmen wir noch mehr Feedback auf und implementieren dies. Danach testen wir erneut, jedoch mit anderen Personen.
-----	-------------------------	---

4.6.3 Anforderungen an die Validierung

Nr.	Anforderung	Testanforderung
V1	Zufriedenheit	Nach dem Ausfüllen des Prototyps auf Papier, verlangen wir noch ein Feedback von unseren Testpersonen. Sie sollen ihre Zufriedenheit in einer Skala von 1 bis 10 angeben und uns Feedback geben. Dadurch können wir gewährleisten, dass unser Endprodukt funktionell und qualitativ ist.
V2	Anzahl Test Personen	Es werden mindestens fünf Testpersonen befragt. Diese sollten von verschiedenem Alters- und Sozialgruppen kommen. Wir haben bereits genug Personen angefragt, welche bereit sind unser Programm zu testen. Wichtig ist, dass wir mehr als fünf Personen angefragt haben, da oft jemand einen Rückzieher macht.
V3	Auswertung	Für die Auswertung werden wir den Durchschnitt nehmen. Den Median werden wir uns auch anschauen, um zu sehen, wo dieser im Vergleich zum Durchschnitt liegt. Dies wird manuell erfolgen und uns klar zeigen, wo noch Verbesserungsbedarf besteht.

5 Realisierungsphase

5.1 Systemdesign und Schnittstellen

Das entwickelte System zur personalisierten Empfehlung von Ausflugszielen basiert auf einer klar strukturierten Architektur. Im Zentrum stehen drei Klassen, die in C# implementiert wurden: *User*, *Place* und *Program*. Ergänzt wird die Systemarchitektur durch eine SQLite-Datenbank, in welcher die Ausflugsziele gespeichert sind.

5.1.1 Klassenstruktur

Die nachfolgende Darstellung zeigt die grundlegenden Klassen und deren Eigenschaften sowie Methoden:

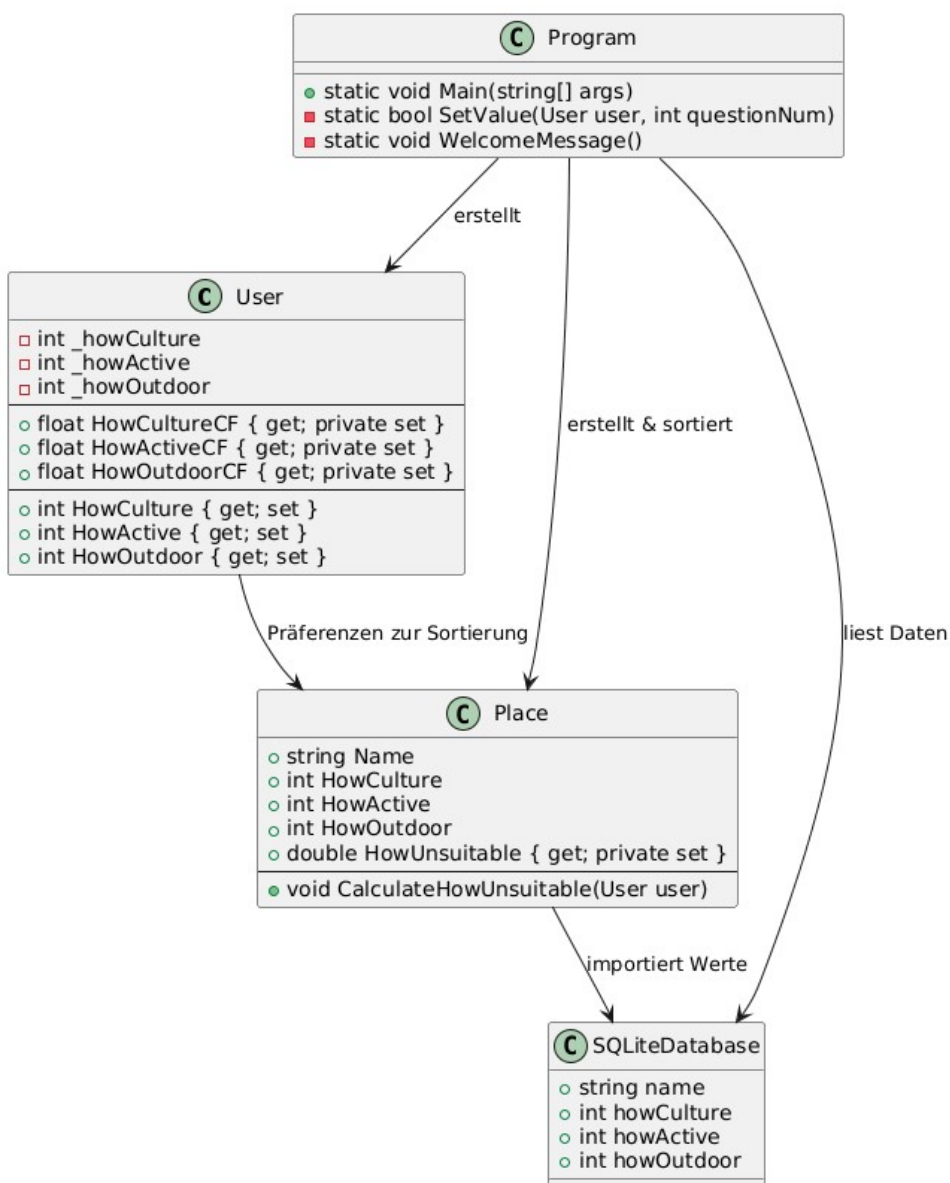


Abbildung 6 UML-Diagramm der Klassenstruktur

User

- Enthält die Präferenzen des Nutzers (*HowCulture*, *HowActive*, *HowOutdoor*).
- Verwaltet intern die Gewichtungskoeffizienten (CFs), welche bei extremen Eingaben automatisch angepasst werden.
- Die Präferenzen können nur über Properties verändert werden, um Konsistenz zu gewährleisten.

Place

- Repräsentiert ein einzelnes Ausflugsziel aus der Datenbank.
- Wird in der Klasse *Program* Teil der Liste aller Ausflugsziele.
- Speichert die Eigenschaften Kultur, Aktivität und Indoor/Outdoor eines Ausflugsziels
- Enthält die Methode *CalculateHowUnsuitable(User user)*, welche den Kern des Empfehlungsalgorithmus bildet.

Program

- Verbindet die Komponenten des Systems.
- Verantwortlich für den Start des Programms, das Einlesen der Datenbank, die Benutzerinteraktion über die Kommandozeile sowie die Auswertung und Ausgabe der Ergebnisse.
- Enthält zudem Hilfsmethoden wie *SetValue()* zur Eingabevalidierung und -verarbeitung und *WelcomeMessage()* zur Begrüßung und Instruktion des Nutzers.

5.1.2 Schnittstellen

Benutzerinterface	Das System verfügt über eine textbasierte Kommandozeilen-Schnittstelle. Über diese werden drei Fragen gestellt, welche der Nutzer mit Werten zwischen 1 und 10 beantwortet.
Datenbankzugriff	Der Zugriff auf die SQLite-Datenbank erfolgt direkt über <i>Microsoft.Data.Sqlite</i> . Die Datenbank liegt im selben Ordner wie das Programm: Eine externe Konfiguration oder Netzwerkverbindung ist nicht erforderlich.
Ausgabe	Nach Auswertung wird das passendste Ausflugsziel direkt in der Konsole ausgegeben. Auf Wunsch des Nutzers können sukzessive weitere Vorschläge angezeigt werden.

Damit ist das System als geschlossenes, eigenständiges Produkt aufgebaut: Eingaben des Nutzers werden erfasst, mit den gespeicherten Ausflugszielen verglichen und die Ergebnisse unmittelbar präsentiert.

5.1.3 Datenbankschnittstelle

Die Daten zu den Ausflugszielen sind in einer SQLite-Datenbank (*ziele.db*) abgelegt. Jede Entität ist durch die folgenden Attribute beschrieben:

Attribut	Datentyp	Beschreibung
<i>name</i>	Text	Name des Ausflugsziels
<i>howCulture</i>	Integer	Kulturelle Relevanz (Skala 1–10)
<i>howActive</i>	Integer	Körperliche Aktivität (Skala 1–10)
<i>howOutdoor</i>	Integer	Indoor-/Outdoor-Anteil (Skala 1–10)

Beim Programmstart wird die Datenbank geöffnet und alle enthaltenen Ausflugsziele in eine Liste von *Place*-Objekten eingelesen. Damit ist die Datenbank vollständig in das Programm integriert und benötigt zur Laufzeit keine manuelle Anpassung.

5.1.4 Kompilierung Lieferobjekt

Das entwickelte System wurde in C# mit Visual Studio 2022 erstellt. Für die Auslieferung wurde der Quellcode im Release-Modus kompiliert, sodass eine lauffähige Version des Programms vorliegt.

Die Kompilierung erfolgte mit folgendem Befehl:

```
dotnet publish -c Release -r win-x64 --self-contained true -p:PublishSingle-File=true -p:IncludeNativeLibrariesForSelfExtract=true
```

Das Ergebnis der Kompilierung liegt im Ordner:

```
bin\Release\net8.0\win-x64\publish\
```

Darin befindet sich die Datei *Ausflugsziel Empfehler.exe*, die als Haupt-Lieferobjekt ausgeliefert wird. Zusätzlich liegt die *ziele.db* als separate SQLite-Datenbank im gleichen Ordner. Diese wird vom Programm automatisch geladen und enthält die hinterlegten Ausflugsziele.

Die Datei *.pdb* (Debug-Symbole) ist nur für Entwickler relevant und wird nicht mit ausgeliefert.

Damit besteht das finale Lieferobjekt aus:

- *Ausflugsziel Empfehler.exe* (ausführbares Programm)
- *ziele.db* (Datenbank mit Ausflugszielen)

Die Ausführung erfolgt durch einen einfachen Doppelklick auf die *.exe*-Datei.

5.2 Systemdokumentation

5.2.1 Konfiguration

Das System liegt als kompiliertes Konsolenprogramm (.exe) vor. Es ist keine zusätzliche Installation notwendig. Das Programm ist lauffähig auf jedem Rechner mit Windows 11 und einer Standard-Kommandozeilenumgebung.

Installation

- Der Benutzer lädt den Programmordner (umfasst eine .exe-Datei und die Datenbank der Ausflugsziele) herunter.
- Die Datenbank muss nicht separat heruntergeladen werden, da sich diese bereits im Programmordner befindet.

Start

- Mit einem Doppelklick auf die .exe-Datei wird die Konsole geöffnet.
- Das Programm startet automatisch und begrüsst den Benutzer.

Benutzung

- Der Benutzer beantwortet nacheinander drei Fragen zu seinen Präferenzen (Kultur, Aktivität, Indoor/Outdoor).
- Die Eingaben erfolgen durch Zahlenwerte von 1 bis 10.
- Nach Abschluss der Eingaben präsentiert das Programm das am besten passende Ausflugsziel.
- Durch Drücken der Enter-Taste können weitere Vorschläge angezeigt werden, bis alle Ausflugsziele ausgeschöpft sind.
- Anschliessend wird das Programm mit einem weiteren Enter beendet.

Voraussetzungen

- Betriebssystem: Windows 11
- Speicherplatzbedarf: ca. 50 MB
- Keine zusätzlichen Software-Abhängigkeiten, da alle benötigten Bibliotheken in der ausführbaren Datei enthalten sind.

Das System ist somit out-of-the-box nutzbar und benötigt keinerlei Konfigurationsaufwand durch den Benutzer oder Administratoren.

5.2.2 Benutzerhandbuch

5.2.2.1 Zweck

Das Benutzerhandbuch dient als Anleitung für Anwenderinnen und Anwender, die den „Ausflugsziel-Empfehler“ testen oder nutzen möchten. Es beschreibt die Bedienung, Eingabe- und Ausgabefunktionen sowie den Umgang mit typischen Fehlermeldungen.

5.2.2.2 Start des Programms

Öffne den Programmordner und starte die Datei «Ausflugsziel Empfehler.exe».

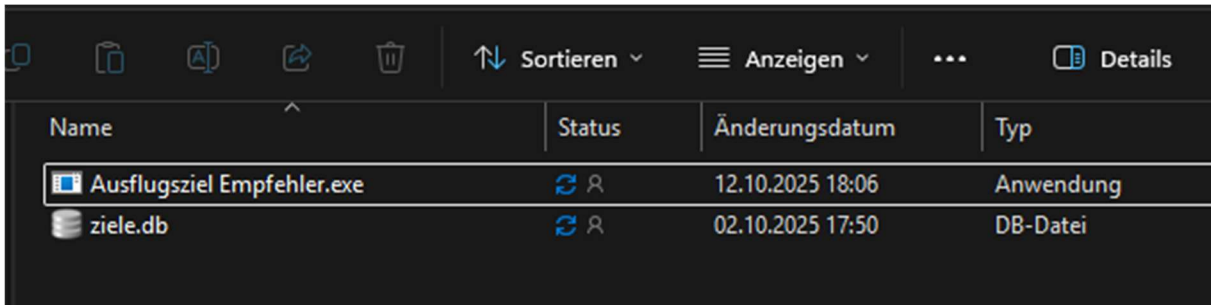


Abbildung 7 Screenshot Programmordner im File Explorer

Es öffnet sich ein Konsolenfenster und das Programm startet von selbst. Das Programm kann nun normal verwendet werden, es sind keine weiteren Konfigurationen nötig.

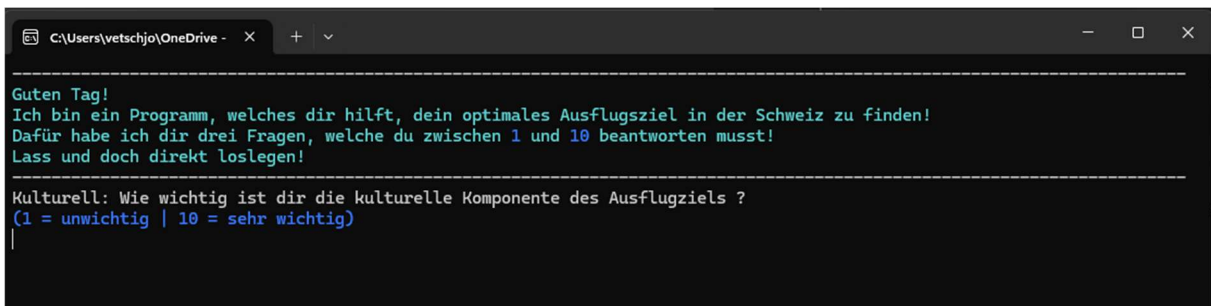


Abbildung 8 Screenshot bei Programmstart

5.2.2.3 Ablauf der Nutzung

Das Programm führt dich in drei Schritten durch eine kurze Befragung zu deinen Bedürfnissen.

Die Eingabe erfolgt jeweils durch Zahlen (1–10) mit anschliessendem Drücken der Eingabetaste «Enter».

	Frage	Bedeutung der Eingaben
1	Kultur-Anteil des Ausflugsziels	1 = Ich möchte möglichst keine Kultur erleben. 5 = Es darf ein kulturelles Ausflugsziel sein. Kultur soll aber nicht im Zentrum stehen. 10 = Die kulturelle Komponente des Ausflugsziels ist mir enorm wichtig.

2	Körperliche Aktivität	1 = Ich möchte mich sehr wenig bewegen. 5 = Ich möchte eine mittlere körperliche Aktivität haben. 10 = Ich möchte sehr stark körperlich aktiv sein.
3	Indoor / Outdoor	1 = Am Ausflugsziel möchte ich so viel wie möglich drinnen sein. 5 = Ich habe keine Präferenz. 10 = Ich möchte möglichst viel draussen sein.

5.2.2.4 Beispielhafter Programmdurchlauf

```

C:\Users\vetschjo\OneDrive - x + v
-----
Guten Tag!
Ich bin ein Programm, welches dir hilft, dein optimales Ausflugsziel in der Schweiz zu finden!
Dafür habe ich dir drei Fragen, welche du zwischen 1 und 10 beantworten musst!
Lass und doch direkt loslegen!
-----
Kulturell: Wie wichtig ist dir die kulturelle Komponente des Ausflugsziels ?
(1 = unwichtig | 10 = sehr wichtig)
8
-----
Aktiv: Wie sehr möchtest du körperlich aktiv sein?
(1 = wenig aktiv | 10 = sehr aktiv)
8
-----
In-/Outdoor: Möchtest du am Ausflugsziel lieber drinnen oder draussen sein?
(1 = nur indoor | 5 = egal | 10 = nur Outdoor)
3
-----
Vielen Dank für deine Eingabe!
Das für dich passendste Ausflugsziel ist Sundlaunen, St. Beatus Höhlen!
-----
Drücke Enter, um einen weiteren auf dich passenden Vorschlag zu bekommen.
|
    
```

Abbildung 9 Beispielhafter Programmdurchlauf

Nach Beantwortung der drei Fragen zu Kultur, körperlicher Aktivität und Indoor / Outdoor wird das Ausflugsziel aus der Datenbank vorgeschlagen, das am besten zu deinem angegebenen Bedürfnis passt. In diesem Beispiel sind es die St. Beatus Höhlen.

5.2.2.5 Weitere Ziele entdecken

Wenn du ein weiteres, auf deine Bedürfnisse passendes Ziel anzeigen möchtest, so drücke nun die Eingabetaste «Enter». Das System hat anhand deiner Bedürfnisse bereits aus allen ihm bekannten Ausflugszielen eine Rangliste erstellt. Diese Rangliste ist nach deinen Bedürfnissen geordnet.

Das oberste Ziel auf der Rangliste wurde dir bereits ausgegeben. Mit dem Drücken der Eingabetaste «Enter» kannst du nun das zweite Ziel auf der Rangliste anzeigen lassen.

```

C:\Users\vetschjo\OneDrive - x + v
-----
Guten Tag!
Ich bin ein Programm, welches dir hilft, dein optimales Ausflugsziel in der Schweiz zu finden!
Dafür habe ich dir drei Fragen, welche du zwischen 1 und 10 beantworten musst!
Lass und doch direkt loslegen!

-----
Kulturell: Wie wichtig ist dir die kulturelle Komponente des Ausflugsziels ?
(1 = unwichtig | 10 = sehr wichtig)
8

-----
Aktiv: Wie sehr möchtest du körperlich aktiv sein?
(1 = wenig aktiv | 10 = sehr aktiv)
8

-----
In-/Outdoor: Möchtest du am Ausflugsziel lieber drinnen oder draussen sein?
(1 = nur indoor | 5 = egal | 10 = nur Outdoor)
3

-----
Vielen Dank für deine Eingabe!
Das für dich passendste Ausflugsziel ist Sundlaunen, St. Beatus Höhlen!

Drücke Enter, um einen weiteren auf dich passenden Vorschlag zu bekommen.

Montreux, Château de Chillon

-----
Drücke Enter, um einen weiteren auf dich passenden Vorschlag zu bekommen.

|
    
```

Abbildung 10 Das Drücken der Eingabetaste zeigt ein weiteres passendes Ausflugsziel

Durch erneutes Drücken der Eingabetaste wird dir das dritte Ziel ausgegeben. Du kannst diesen Vorgang so lange wiederholen, bis du genug Inspiration bekommen hast oder bis dir alle Ziele der Datenbank angezeigt wurden.

```

-----
Drücke Enter, um einen weiteren auf dich passenden Vorschlag zu bekommen.

Weggis, Vierwaldstättersee

-----
Drücke Enter, um einen weiteren auf dich passenden Vorschlag zu bekommen.

Hinwil, Fondue im Hot Tub

-----
Zurzeit haben wir keine weiteren Ausflugsziele.
Drücke Enter, um das Programm zu beenden.

|
    
```

Abbildung 11 Programmzustand, nachdem alle Ziele angezeigt wurden

Nachdem alle Ziele angezeigt wurden, beendet ein weiteres Drücken der Eingabetaste «Enter» das Programm.

5.2.2.6 Fehlermeldungen und Hinweise

Das Programm kann deine Bedürfnisse nur als ganze Zahl von 1-10 verarbeiten. Texteingaben und andere Eingaben werden nicht angenommen. Das Programm informiert dich mit «Bitte gebe eine ganze Zahl zwischen 1 und 10 ein.», solltest du aus Versehen eine ungültige Eingabe tätigen. Zudem wird dir die aktuelle Frage erneut gestellt.

```

C:\Users\vetschjo\OneDrive - x + v
-----
Guten Tag!
Ich bin ein Programm, welches dir hilft, dein optimales Ausflugsziel in der Schweiz zu finden!
Dafür habe ich dir drei Fragen, welche du zwischen 1 und 10 beantworten musst!
Lass und doch direkt loslegen!
-----
Kulturell: Wie wichtig ist dir die kulturelle Komponente des Ausflugsziels ?
(1 = unwichtig | 10 = sehr wichtig)
2
-----
Aktiv: Wie sehr möchtest du körperlich aktiv sein?
(1 = wenig aktiv | 10 = sehr aktiv)
Markus
Bitte gebe eine ganze Zahl zwischen 1 und 10 ein.
Aktiv: Wie sehr möchtest du körperlich aktiv sein?
(1 = wenig aktiv | 10 = sehr aktiv)
|
    
```

Abbildung 12 Programmverhalten bei ungültiger Eingabe

Zudem sind diese zwei Details wichtig zu wissen:

- Alle Ziele wurden angezeigt** «Zurzeit haben wir keine weiteren Ausflugsziele.»
- Abbruch** Du kannst das Programm mit Ctrl + C jederzeit beenden.

5.3 Tests

5.3.1 Funktionale Anforderungen

Testfälle	Durchführung	Auswertung
Spannungsfelder	Nach dem Öffnen der Anwendung bewertet man die drei Spannungsfelder (Kulturell, Aktiv und In-/Outdoor). Danach wird einem ein Ausflugsziel vorgeschlagen.	Die Spannungsfelder sind vorhanden und werden dem Nutzer angezeigt, so dass er dieses Bewerten kann. Zudem haben alle Ausflugsziele der Datenbank zu den drei Spannungsfeldern konkrete werte. Die Testanforderung wurde erfüllt.
Eingaben	Es wird einem erklärt wie die Spannungsfelder von 1-10 zu bewerten sind. Ich als Nutzer kann verschiedenen Eingaben zu den Spannungsfeldern machen.	Der Nutzer kann selbstständig Eingaben machen. Dadurch entsteht ein spezifischer wert, welcher mit den Ausflugszielen verglichen werden kann. Die Testanforderung wurde erfüllt.
Konkrete Ausgabe	Nach der Auswertung meiner Eingabe wird mir das passendste Ausflugsziel vorgeschlagen.	Die Ausgabe ist konkret, da die Spannungsfelder der Eingabe mit den Ausflugszielen der Datenbank verglichen werden. Dadurch wird dem Nutzer das passendste Ausflugsziel vorgeschlagen. Die Testanforderung wurde erfüllt.
Sonderfälle	<p>Keine Ausgabe: Bei falschen Eingaben bleibt das Programm offen und gibt eine Rückmeldung zur falschen Eingabe.</p> <p>Falls Nutzer unzufrieden: Bin ich als Nutzer mit dem Ausflugsziel unzufrieden kann ich per "Enter" ein weiteres Ausflugsziel anzeigen lassen.</p> <p>In-/Outdoor = 5: Falls es mir egal ist, ob die Aktivität Indoor oder Outdoor stattfindet, kann ich eine 5 angeben. Dadurch fällt dieses Spannungsfeld weg</p> <p>Mehrere Antworten: Wenn ich bei In-/Outdoor eine 5 angebe kann es sein, dass mehrere Ausflugsziele gleich passend sind. In diesem Fall wird mir eins Zufällig angegeben, ich habe jedoch die Option, dass via "Enter" mir weitere Ziele angezeigt werden.</p>	<p>Keine Ausgabe: Für diesen Sonderfall gibt die Anwendung dem Nutzer eine Fehlermeldung. Dies Hilft dem Nutzer den Fehler zu beheben.</p> <p>Falls Nutzer unzufrieden: Eine Funktion, um das nächstpassendste Ausflugsziel anzuzeigen ist vorhanden und kann einfach benutzt werden.</p> <p>In-/Outdoor = 5: Bei diesem Spannungsfeld wird dem Nutzer angezeigt, dass er eine 5 als Sonderfall eingeben kann. Gibt der Nutzer eine 5 bei In-/Outdoor ein wird dieses Spannungsfeld nicht gewertet.</p> <p>Mehrere Antworten: Für den Fall, dass mehrere Ausflugsziele genau gleich passend sind, gibt es einen Algorithmus welcher Zufällig eines der Ausflugsziele ausgibt. Der Nutzer hat dann die Option via "Enter" sich weitere Ziele anzuzeigen. Die Sonder-Testfälle sind alle abgedeckt und wurden erfüllt.</p>

Benutzerschnittstelle	Beim Öffnen des Programmes wird die Benutzerschnittstelle, welche über die Kommandozeilen läuft, geöffnet. Hier kann ich drei kurze Fragen beantworten und erhalte dann eine Ausgabe.	Das Programm hat eine interaktive Benutzerschnittstelle, welche optisch simple und einfach zu bedienen ist. In nur drei Schritten wird dem Nutzer ein passendes Ausflugsziel vorgeschlagen. Die Testanforderung wurde erfüllt.
-----------------------	---	---

5.3.2 Nicht-Funktionale Anforderungen

Testfälle	Durchführung	Auswertung
Konsolenprogramm	Wir haben über die Herbstferien unser Projekt als Konsolenprogramm realisiert. Als Programmiersprache haben wir C# benutzt, da wir uns darin am besten auskennen.	Der fertige Prototyp ist ein funktionierendes Programm, welches über die Konsole läuft. Der Code ist wie geplant in C# programmiert. Die Testanforderung wurde erfüllt.
Machbarkeit	Bei der Umsetzung der Realisierung hatten wir nicht viele Probleme. Wir konnten ziemlich ohne grosse "Knacknüsse" die Datenbank erstellen und das Programm schreiben.	Wir haben unser Projektziel gut gewählt und die Machbarkeit war von Anfang an gegeben. Das Projekt konnte wie geplant umgesetzt werden und wir mussten nichts Funktionelles weglassen. Die Testanforderung wurde erfüllt.
Intuitiv / max. 2 Minuten	Nach dem Öffnen des Programmes wird mir in drei Zeilen erklärt, worum es geht. Danach kann ich drei einfache Fragen von 1-10 beantworten und erhalte sofort mein Ausflugsziel.	Das Programm ist sehr intuitiv und es ist kaum möglich etwas Falsch zu verstehen. Zudem ist eine kleine Erklärung vorhanden. Durch die Begrenzung auf drei Spannungsfelder ist eine schnelle Eingabe und Auswertung möglich Die Testanforderung wurde erfüllt.
Keine unklaren Resultate	Wir haben alle die verschiedenen Ausflugsziele nach den Spannungsfeldern bewertet und diese ausgewertet. Dadurch haben wir eine realistische Bewertung der Ausflugsziele, welche klar und logisch sind.	Durch die Bewertung mehrere Personen wird der Risikofaktor, dass etwas nicht logisch bewertet ist, verkleinert. Besonders das Thema "körperlich aktiv" ist sehr individuell, daher war es uns wichtig die Ausflugsziele realistisch und logisch zu bewerten. Die Testanforderung wurde erfüllt.
Deterministisch	Wenn ich zweimal die gleichen Eingaben betätige, erhalte ich zweimal das gleiche Resultat.	Unser Programm vergleicht die Spannungsfelder der Eingabe mit den Ausflugszielen der Datenbank. Dadurch wird die gleiche Eingabe auch immer die gleiche Ausgabe geben. Einzig wenn es mehrere gleichpassende Ausgaben gibt, kann es vorkommen, dass es zwei unterschiedliche Ausgaben gibt, Da diese per Zufall gewählt werden.

		Die Testanforderung wurde erfüllt.
ca. 40 Ausflugsziele	In unserer Datenbank sind 40 Ausflugsziele vorhanden. Sie sind alle interessant und unterschiedlich. Zudem sind sie alle nach den drei Spannungsfeldern bewertet.	Um die Empfehlungslogik zu demonstrieren haben wir 40 Ausflugsziele gesucht, bewertet und in der Datenbank gespeichert. Die Ausflugsziele sind alle verschieden und interessant. Die Testanforderung wurde erfüllt.
Abnahmekriterium	Wir haben die Papierfragebögen von 5 Personen ausfüllen lassen. Diese Personen waren aus unserem Umfeld und waren sehr zufrieden mit unserem Programm. Die Zufriedenheitsbewertung auf einer Skala von 1-10 lag bei durchschnittlich 7.8.	Das Abnahmekriterium von einer Zufriedenheit von mindestens ≥ 6 wurde erfüllt. Daher wissen wir, auch wenn die Anzahl der Rückmeldungen nicht sehr gross ist, dass unser Programm intuitiv und nützlich ist. Die Testanforderung wurde erfüllt.

5.3.3 Anforderungen an die Validierung

Testfälle	Durchführung	Auswertung
Zufriedenheit	Wir haben Testpersonen um Feedback gebeten. Sie beantworteten auf einer Skala von 1-10 wie zufrieden sie waren.	Die Testpersonen waren sehr zufrieden. Vielen gefiel, wie einfach und trotzdem sinnvoll das Programm ist.
Anzahl Test Personen	Die fünf befragten Testpersonen waren alle von verschiedenem Alter und Sozialgruppen. Wir fragten Personen im Alter von 16 bis 53 Jahren.	Wir konnten wie geplant fünf sehr verschiedene Persönlichkeiten mit verschiedenem Alter befragen und deren Feedback auswerten. Es ist uns gelungen, auch wenn es nur fünf Testpersonen waren, fünf sehr individuelle Feedbacks zu erhalten. Die Testanforderung wurde erfüllt.
Auswertung	Bei der Auswertung der Fragebögen erhielten wir folgende Bewertungen: 9, 9, 8, 7, 6 , dadurch ergibt sich ein Durchschnitt von 7.8 und ein Median von 8 .	Die Bewertungen waren äusserst positiv. Besonders gut kam an, wie einfach das Programm zu benutzen ist. Verbesserungspotenzial hat das Programm bei der Optik der Benutzeroberfläche. Die Testanforderung wurde erfüllt.

6 Reflexion

6.1 Jonas Vetsch (Projektleitung)

6.1.1 Welche Arbeiten habe ich für die Projektgruppe übernommen und realisiert?

Zu Beginn unseres Projekts war ich ursprünglich als das technische Teammitglied vorgesehen. Bereits nach kurzer Zeit (ca. 1 Woche) wurde aber deutlich, dass es nicht wirklich eine Leitung gab im Projekt. Da ich strukturierte Arbeitsweisen schätze und «von Natur aus» ein starkes Bedürfnis habe, den Überblick zu behalten, übernahm ich spontan die Initiative: Ich ordnete die anstehenden Aufgaben, erstellte erste Strukturvorschläge und plante die nächsten Schritte. Dies wurde vom Team positiv aufgenommen, sodass es sich als Konsens im Team erwies, dass ich Rolle der Projektleitung übernehmen soll. Das war für mich in Ordnung, ich sah es als eine interessante Herausforderung.

6.1.2 Was ist mir gut gelungen?

Als Projektleiter war ich verantwortlich für die Koordination der Arbeiten, sowie für die Planung gemäss der HERMES-Methode. Ich bereitete jede Phase im Voraus vor, damit unsere Teammeetings effizient ablaufen konnten. Dazu las ich jeweils vor der Lektion die kommenden Anforderungen des Moduls, definierte Arbeitspakete und erarbeitete Vorschläge, die wir im Team besprachen. Dieses Vorgehen hat sich sehr bewährt – wir hatten dadurch in jeder Phase eine klare Richtung, was sich auch in den konstant sehr guten Bewertungen widerspiegelte.

6.1.3 Schwierigkeiten, Probleme, Lösungen und Hilfestellungen

Besonders fordernd war für mich anfangs das Verständnis der HERMES-Methode selbst. Ich kannte zuvor nur agile Vorgehensweisen, bei denen man direkt mit dem Produkt arbeitet. HERMES dagegen verlangt viel konzeptionelle Vorarbeit und umfangreiche Dokumentation. Anfangs empfand ich dies als bürokratisch und wenig produktiv, weil man sehr viel schreibt und vergleichsweise wenig programmiert. Mit der Zeit erkannte ich jedoch den Sinn dahinter: Durch die gründliche Planung wird das eigentliche Coden in der Realisierung erleichtert. Diese Erkenntnis hat mein Verständnis von Projektarbeit nachhaltig erweitert – gute Planung ist ein eigenständiger und zentraler Teil des Projekterfolgs.

Im Laufe des Projekts wurde mir deutlich bewusst, wie stark die Qualität unserer Teammeetings davon abhing, wie gut ich selbst vorbereitet war. Wenn ich die Kapitel des Moduls im Vorhinein nicht gelesen oder noch nicht ganz verstanden hatte, verliefen die Sitzungen oft zäh und unproduktiv. Wir diskutierten dann lange über Vorgehensfragen, verloren uns in Detaildebatten oder merkten erst spät, dass wir Anforderungen übersehen hatten. Sobald ich jedoch das ganze Kapitel im Voraus gründlich durchgearbeitet und mir die methodischen Zusammenhänge angeeignet hatte, konnte ich die Diskussion gezielt steuern, offene Punkte rasch klären und die Gruppe effizient durch die Aufgaben führen. Diese Erfahrung hat mir gezeigt, wie wichtig es als Projektleiter ist, nicht nur den Überblick über den Stand des Projekts zu haben, sondern auch über die theoretischen Grundlagen des Moduls.

Daraus entstand ein zentrales Learning für mich: Als Projektleiter musst du den vollständigen Durchblick über das Vorgehen, die Phasenlogik und die geforderten Ergebnisse haben – du musst derjenige sein, der das Gesamtbild versteht. Das bedeutet nicht, dass du jedes einzelne

Code-Detail kennen oder selbst programmieren musst. Dafür sind die Teammitglieder zuständig, die sich in der Technik vertiefen. Deine Aufgabe ist es, den Ablauf zu meistern, Hindernisse früh zu erkennen und die Brücke zwischen Methode, Ziel und Umsetzung zu schlagen. Diese Erkenntnis hat mir geholfen, meine Rolle klarer zu verstehen und mich stärker auf die Organisation, Planung und Qualitätssicherung zu konzentrieren, statt mich in technischen Details zu verlieren.

6.1.4 Lernprozess

Ich lernte auch, dass Projektleitung nicht bedeutet, alles selbst zu machen, sondern den Rahmen zu schaffen, in dem andere ihren Beitrag leisten können – auch wenn der Arbeitsstil verschieden ist. Ich versuchte, Aufgaben klar zu formulieren, Zwischenziele zu setzen und positives Feedback zu geben, wenn Fortschritte sichtbar wurden. Rückblickend hat sich über das ganze Projekt hinweg eine immer besser werdende positive Stimmung ergeben, insbesondere als die ersten erfolgreichen Phasenbewertungen kamen. Diese Erfahrung hat mir gezeigt, wie wichtig es ist, Motivation im Team nicht vorauszusetzen, sondern aktiv zu gestalten (z.B. durch Zelebrierung von Zwischenerfolgen).

Ein besonderes Highlight war die Idee von VAPEAS, die ich spontan eingebracht habe. Sie zeigte mir, dass auch in einem streng strukturierten Vorgehen nach HERMES Platz für kreative Einfälle bleibt. Das hat mir Freude gemacht, weil ich dadurch beides verbinden konnte – methodische Genauigkeit und eigene Kreativität.

Ich habe im Verlauf des Projekts viele Softskills entwickelt: vorausschauendes Planen, klare Kommunikation, Konfliktlösung und die Fähigkeit, Struktur zu schaffen, wo zunächst Chaos herrscht. Ich habe gelernt, Meetings vorzubereiten statt sie „im Meeting“ rollend zu planen, und dadurch unsere gemeinsame Zeit effizient zu nutzen. Gleichzeitig habe ich verstanden, dass Projektleitung auch bedeutet, Kompromisse einzugehen, Verantwortung zu übernehmen und bei Problemen ruhig zu bleiben.

6.1.5 Reflexion, Fazit, Massnahmen

Ich nehme aus diesem Projekt mit, dass gute Führung immer sowohl Planung als auch Menschlichkeit braucht. In zukünftigen Projekten möchte ich noch stärker darauf achten, Verantwortung gezielt zu teilen und Teammitglieder frühzeitig einzubinden, damit nicht zu viel Koordination an mir hängen bleibt. Insgesamt hat mir die Rolle des Projektleiters grossen Spass gemacht – sie hat mir gezeigt, dass ich Organisation und Leadership sehr gerne übernehme und dass strukturierte Planung kein Selbstzweck ist, sondern die Grundlage für erfolgreiche IT-Projekte bildet.

6.2 Adrian Herbst (Mitarbeiter / Stv. Projektleitung)

6.2.1 Welche Arbeiten habe ich für die Projektgruppe übernommen und realisiert?

Meine Rolle in der Gruppe war Stv. Projektleiter/Mitarbeiter, ich musste nur einmal für Jonas einspringen und als Projektleiter die Aufgabenverteilung koordinieren. Im Normalfall arbeitete ich als Mitarbeiter und erfüllte die mir zugewiesenen Aufträge. Ich kümmerte mich primär um die Testfälle und Anforderungen. Nebenbei übernahm ich kleine Aufträge und half Jonas das Dokument zu überarbeiten und Korrekturen vorzunehmen.

6.2.2 Was ist mir gut gelungen?

Mir gelang gut, die Verschiedenen Testfälle zu definieren, dokumentieren und auszuwerten. Ich bin sehr stolz darauf, dass wir alle Testfälle erfüllen konnten, welche wir uns gesetzt haben. Zudem konnte ich gut die Stellvertretung für unseren Projektleiter übernehmen und die diversen Aufträge verteilen. Des Weiteren bin ich zufrieden, dass ich mich immer gut an die gesetzten Deadlines gehalten habe.

6.2.3 Wenn es Schwierig wurde, wie habe ich die Probleme gelöst?

Einmal gab es etwas Zeitstress, da zwei Bewertungskriterien vergessen wurden. An dem Donnerstag arbeitete ich einige Stunden an dem Problem. Ich habe dann kurz in unseren Gruppenchat geschrieben, dann ist Jonas mir schnell zur Hilfe geeilt und hat noch bis spät in den Abend daran gearbeitet. Ansonsten konnte ich meine Schwierigkeiten oft selbst oder via Google beseitigen.

6.2.4 Wie habe ich mich beholfen? Welche Hilfestellung habe von wem erhalten?

Ich hatte nicht sehr viele Situationen, mit denen ich überfordert war, oft war es Fleissarbeit. Dadurch ist für mich der Wichtigste Faktor die Zeiteinteilung und Motivation. Die Motivation fand ich in diesem grossartigen Gruppenprojekt sehr einfach, wodurch ich mich auch gerne am Wochenende mal länger hingesezt habe. Wenn ich Hilfe brauchte, war Jonas die Ansprechperson. Jonas hat oft auch meine Texte korrekturgelesen und diese noch etwas aufgehübscht.

6.2.5 Reflexion, Fazit, Massnahme

Ich bin sehr zufrieden mit unserer Gruppe sowie der Arbeit. Es hat mir viel Spass gemacht, im Rahmen des Moduls, ein Projekt zu planen und zu realisieren. Wir haben eventuell etwas über das Ziel hinausgeschossen, was den Umfang der Arbeit betrifft, doch war der Lerneffekt sehr gross. Es gibt wenig, was ich in Zukunft anders machen würde, eventuell würde ich gerne mal die Rolle des Projektleiters übernehmen. Im Grossen und Ganzen bin ich sehr stolz auf unsere Gruppe und die Arbeit.

6.3 Marco Saez (Mitarbeiter)

6.3.1 Welche Arbeiten habe ich für die Projektgruppe übernommen und realisiert?

Im Rahmen des Projekts übernahm ich die Rolle eines Mitarbeiters.

In der Konzeptphase war ich unter anderem dafür zuständig, eine Nutzerskizze zu erstellen, um typische Nutzerprofile für unser Projekt zu definieren. Zusätzlich habe ich in dieser Phase eine Materialliste angefertigt, in der alle benötigten Ressourcen zur Umsetzung des Projekts erfasst wurden.

Während der Realisierungsphase war ich hauptsächlich mit der Datenbank und der Recherche zu den Ausflugszielen beschäftigt. Zunächst erstellte ich die Datenbank mithilfe von Jonas' Unterstützung und definierte die nötigen Attribute sowie deren Einschränkungen. Im nächsten Schritt suchte ich passende Ausflugsziele, die ich mithilfe von zuerst gesuchten Kategorien effizienter suchen konnte. Die gesammelten Ausflugsziele wurden zunächst in eine Excel-Tabelle eingetragen. Dort habe ich die Ziele anhand von drei Spannungsfeldern bewertet, was mir gut gelungen ist. Nach der Bewertung gab ich die Daten zur weiteren Bewertung an Jonas weiter. Zum Schluss habe ich die Daten der Ausflugsziele in eine andere Datenbank mit identischer Struktur migriert, da ich Probleme hatte sie in die bestehende Datenbank einzuspeisen.

6.3.2 Was ist mir gut gelungen?

Die Aufgaben der Kategorisierung und des Heraussuchens geeigneter Ausflugsziele habe ich als sehr gelungen empfunden. Auch das systematische Bewerten der Ziele anhand der vorgegebenen Spannungsfelder verlief ohne größere Schwierigkeiten.

6.3.3 Wenn es Schwierig wurde, wie habe ich die Probleme gelöst?

Rückblickend war die Arbeit an der Nutzerskizze eine der schwierigeren Aufgaben, vor allem aufgrund der anfangs unklaren Anforderungen. Dennoch konnte ich durch Eigeninitiative eine geeignete Lösung finden. Ebenso war das Thema Datenmigration aus Excel bzw. CSV-Dateien zu Beginn herausfordernd. Erst nach intensiver Internetrecherche und mehreren Tests konnte ich zumindest eine funktionierende Teil-Lösung umsetzen.

Die Migration der Daten stellte sich allerdings als unerwartete Herausforderung heraus. Ich hatte ursprünglich geplant, die Inhalte direkt in unsere bestehende Datenbank zu importieren. Jedoch funktionierte dieser Ansatz jedoch nicht wie vorgesehen. Nach intensiver Recherche und mehreren Lösungsversuchen entschied ich mich schliesslich dafür, eine neue Datenbank mit identischer Struktur anzulegen und die Daten dort erfolgreich zu migrieren. Auch wenn dieser Umweg nicht Teil des ursprünglichen Plans war, konnte ich auf diese Weise dennoch ein funktionierendes Ergebnis erzielen.

6.3.4 Wie habe ich mich beholfen? Welche Hilfestellung habe von wem erhalten?

In schwierigen Situationen habe ich stets versucht, eigenständig über das Internet Lösungsansätze zu finden. Zusätzlich habe ich bei Bedarf auch auf die Unterstützung meines Teams zurückgegriffen, was sich als sehr hilfreich erwiesen hat.

6.3.5 Reflexion, Fazit, Massnahme

Das Projekt zeigte, wie hilfreich eine strukturierte Vorgehensweise. Die definierte Rollenverteilung sowie die gute Koordination durch unseren Projektleiter erleichterten die Zusammenarbeit im Team. Die Anwendung des HERMES-Modells unterstützte uns dabei, den Überblick über Aufgaben, Ergebnisse und Verantwortlichkeiten zu behalten. Insgesamt konnte ich aus dem Projekt wertvolle Erfahrungen in praxisnahes Arbeiten nach einem professionellen Projektmanagementstandard sammeln.

6.4 Oleksandr Kolomiets (Mitarbeiter)

Zu Beginn habe ich die Rolle des Projektleiters übernommen, die ich später an Jonas abgegeben habe, um mich stärker auf die inhaltliche Mitarbeit zu konzentrieren und weil es Jonas besser passt. Deshalb kann ich sagen, dass ich während des Projekts eine Rolle des Mitarbeiters hatte. Weiter habe ich alle Aufgaben erledigt, die Jonas mir zugewiesen hat.

Gemeinsam mit der Gruppe diskutierte ich verschiedene Ideen und wir entschieden uns für ein passendes Projektthema – Ausflugszielempfeher. In der Initialisierungsphase war ich an der Erstellung des Zeitplans sowie am Projektantrag beteiligt.

In der Konzeptphase entwickelte ich den Bauplan des geplanten Programms, dokumentierte diesen und erstellte zusätzlich Wireframes, eine Ablaufskizze sowie Use Cases. Zusätzlich habe ich einen Algorithmus entwickelt, der ein passendes Ausflugsziel auswählt. Die Auswahl erfolgt dabei auf Grundlage der Daten, die wir vom Nutzer erhalten. So wird sichergestellt, dass die vorgeschlagenen Ziele möglichst gut zu den individuellen Bedürfnissen und Interessen passen.

In der Realisierungsphase habe ich die Programmierung alleine übernommen. Dabei habe ich mich an den Bauplan und die Dokumentation aus der Konzeptphase gehalten. Nachdem ich die erste Version fertiggestellt hatte, gab ich das Programm an mein Team weiter, damit es getestet werden konnte. Während der Tests fanden meine Teamkollegen eine Abweichung im Ergebnis. Danach habe ich die Ursache im Code gefunden und das Problem erfolgreich behoben. In der Präsentation wurde ich für die Erklärung des Projekts-Algorithmus und Projekt-Demo verantwortlich.

Es war neu für mich, das alles zu machen, weil es meine erste Erfahrung ist. Natürlich hatte ich am Anfang Schwierigkeiten mit Deutsch, aber mit der Zeit hat mein Gehirn dazu gewöhnt. Wenn ich etwas nicht verstanden habe, konnte ich immer für Hilfe bei meinen Teammitglieder fragen. Jetzt kann ich sagen, dass ich gesehen und verstanden habe, wie echte Projekte man machen kann. Ich habe erfahren und verstanden, welche Projektetappen es gibt und wofür man sie braucht, obwohl am Anfang ich gedacht habe, dass es zu viel Dokumentation gebraucht wird. Ich bin sicher, dass diese Kenntnisse mir im Berufsleben als Informatiker weiterhelfen.

7 Fazit

Im Rahmen der Testphase konnten wir sämtliche funktionalen, nicht-funktionalen und Validierungsanforderungen erfolgreich überprüfen und erfüllen. Diese Resultate zeigen deutlich, dass das entwickelte System die gesetzten Projektziele in vollem Umfang erreicht hat. Insbesondere die einfache Bedienbarkeit, die Nachvollziehbarkeit der Empfehlungen sowie die Zufriedenheit der Testpersonen bestätigen, dass das System methodisch sauber umgesetzt wurde und einen echten Nutzen bietet.

Ein zentraler Erfolg ist die Funktionsweise der Empfehlungslogik. Der implementierte Algorithmus vergleicht die Nutzerangaben zu den Spannungsfeldern mit den hinterlegten Werten der Ausflugsziele. Dabei wird automatisch das Ziel mit der höchsten Übereinstimmung vorgeschlagen. Diese Logik funktioniert zuverlässig und deterministisch: Bei identischen Eingaben werden auch identische Ergebnisse erzeugt. Das zeigt, dass unser Algorithmus korrekt arbeitet und stabil ist. Auch Sonderfälle wie mehrere gleichwertige Resultate oder ungültige Eingaben wurden berücksichtigt. Hier überzeugt die Anwendung durch klare Rückmeldungen, welche die Benutzerfreundlichkeit erhöhen und Fehler verhindern.

Die Tests der nicht-funktionalen Anforderungen belegen, dass unser System intuitiv, effizient und praxistauglich ist. Besonders positiv fiel auf, dass Testpersonen ohne Vorkenntnisse die Anwendung vollständig durchlaufen konnten. Der Aufbau über die Kommandozeile ist zwar visuell schlicht, erfüllt jedoch den Zweck dieses Projekts ideal: Die Konzentration liegt auf der Logik und Benutzerführung, nicht auf grafischem Design. Das Programm beweist, dass die Kernidee – ein personalisiertes Empfehlungssystem für Ausflugsziele – technisch umsetzbar und methodisch schlüssig ist.

Bei den Validierungstests erzielten wir ein sehr gutes Ergebnis: Fünf Testpersonen bewerteten das System durchschnittlich mit 7.8 von 10 Punkten, was das definierte Abnahmekriterium (≥ 6.0) deutlich übertrifft. Besonders geschätzt wurde die einfache Bedienung. Einige Testpersonen lobten zudem die Idee einer individuellen Ausflugsempfehlung als praktisch und innovativ. Kritikpunkte betrafen hauptsächlich das einfache Erscheinungsbild der Konsole – eine Rückmeldung, die wir nachvollziehen können, da das Projekt bewusst auf Design verzichtete, um den Fokus auf Funktionalität zu legen. Dieses Feedback liefert wertvolle Hinweise für mögliche Weiterentwicklungen, beispielsweise in Richtung einer grafischen Benutzeroberfläche oder einer Web-Version.

Aus methodischer Sicht hat sich gezeigt, dass die Planung und Umsetzung nach der HERMES-Methode sehr hilfreich waren. Durch die klare Gliederung in Phasen (Initialisierung, Konzept, Realisierung, Einführung) konnten wir unsere Arbeit strukturiert aufbauen und Fortschritte systematisch dokumentieren. Besonders die Definition von funktionalen und nicht-funktionalen Anforderungen sowie die konsequente Durchführung der Tests sorgten dafür, dass das Projekt überprüfbar blieb und alle Teammitglieder jederzeit wussten, worauf sich die Arbeit konzentriert.

Auch im Team selbst war die Zusammenarbeit ein zentraler Erfolgsfaktor. Die Aufgabenteilung war klar, und der Austausch zwischen Programmierung, Test und Dokumentation verliefen zu unser aller Zufriedenheit.

Zusammenfassend lässt sich festhalten, dass das Projektziel vollständig erreicht wurde. Der Prototyp liefert personalisierte, nachvollziehbare Empfehlungen und bietet damit einen echten

Mehrwert gegenüber bestehenden Plattformen. Gleichzeitig ist die Anwendung einfach, zuverlässig und methodisch nachvollziehbar. Verbesserungspotenzial besteht insbesondere im Bereich der Benutzeroberfläche und dem Umfang der Datengrundlage (z. B. durch mehr manuellen Einträgen in der Datenbank oder Einbindung externer Datenquellen). Diese Aspekte könnten in einem möglichen Folgeprojekt weiterentwickelt werden. Insgesamt kann das Projekt „Personalisierte Empfehlung von Ausflugszielen“ als voller Erfolg gewertet werden – sowohl inhaltlich als auch methodisch.

8 Anhang: Arbeitsjournal Realisierungsphase

8.1 Jonas Vetsch

KW	Zeit	Kürzel	Zielsetzung / Tätigkeit	Erfüllungsgrad Positives / Negatives	Reflexion/Fazit Lessons learnt
37 9.9.	0:15	JV	Datenbank anlegen Ich habe Marco gezeigt, wie man im DB-Browser (SQLite) eine Datenbank anlegt und die Attribute korrekt einschränkt. Dies ist die Grundlage, damit Marco nun in der Realisierungsphase seine Arbeiten durchführen kann.	Positiv: Wissensweitergabe im Team hat gut funktioniert. Dadurch kann die Arbeit effizient aufgeteilt werden. Negativ: Es hat etwas Zeit gebraucht, die Einschränkungen der Attribute (Wertebereich 1–10) korrekt im DB-Browser umzusetzen, da dies zunächst nicht ganz intuitiv war.	Es lohnt sich, frühzeitig im Team das Wissen zu teilen, statt alles selbst zu machen. So kommen alle besser voran. Klare Attribut-Einschränkungen (Constraints) sind entscheidend, damit die Datenbank konsistent bleibt. Teamarbeit bedeutet auch Schulung – wenn jeder dasselbe Grundverständnis hat, spart das im Projektverlauf viel Zeit.
38 16.9.	0:15	JV	Team-Koordination Team-Koordination der Realisierungsphase sicherstellen. Da ich krank war, hat mein Stellvertreter Adrian diese Aufgabe für mich übernommen und die Arbeiten der Realisierungsphase im Team verteilt.	Positiv: Dank Adrian war die Team-Koordination trotzdem gewährleistet. Negativ: Dadurch fehlte mir etwas der Überblick, aber Adrian hat die Aufteilung gut dokumentiert, so konnte ich mich einlesen.	Es ist wichtig, im Projekt eine klare Stellvertretung zu haben. Projektleitung bedeutet auch, Verantwortung abgeben und Vertrauen ins Team haben zu können.
40 30.9.	0:45	JV	Einlesen in Projekt-Code	Positiv: Die Integration der Datenbank hat problemlos funktioniert.	Es ist wichtig, dass nicht nur ein Projektmitglied den Code versteht. Einerseits aus Redundanzgründen

			Mich in den Code von Oleksandr einlesen und die Datenbank (von Marco entwickelt) in das C# Projekt von Oleksandr integrieren.	Oleksandr hat während der Entwicklung seines Codes eine Beispieldatenbank derselben Struktur.	aber auch damit man Feedback geben kann: Vier Augen sehen mehr.
40 30.9.	0:30	JV	Build-/Packaging-Test Testen, ob das C#-Projekt sich als ausführbare Datei (.exe) ausliefern lässt. Durchführung von <i>dotnet publish</i> (Release, win-x64) mit Varianten: normales Release, self-contained Single-File und SingleFile inkl. nativer Bibliotheken. Kontrolle des publish-Ordners und Ausführungstest auf Zielrechner.	Positiv: Single-File-Publish erzeugt eine lauffähige .exe; das Programm startet per Doppelklick und läuft auf einem Windows-Rechner ohne vorherige .NET-Installation. Warnungen beim Kompilieren (z. B. Non-Nullable-Properties, Console.ReadLine Null-Hinweis) sind aufgetreten, aber nicht kritisch für die Ausführung. Negativ: Die Datenbankdatei (<i>ziele.db</i>) verbleibt als separate Ressource und muss mitgeliefert werden.	Für eine einfache Abgabe ist das Single-File-Publish die praktikabelste Lösung: Endbenutzer kann das Programm mit einem simplen Doppelklick ausführen. Daten (DB) und ausführbare Logik sind unterschiedliche Verantwortungsbereiche. Die DB separat zu liefern, erlaubt spätere Datenupdates ohne Neuberechnung des Builds. Warnungen im Build ernst nehmen, aber pragmatisch priorisieren: funktionale Warnungen zuerst beheben; kosmetische/Style-Warnungen können ggf. nach Abgabe bereinigt werden.
40 01.10.	0:50	JV	Verbesserung Lesbarkeit des Quellcodes Ich habe die Methode <i>SetValue()</i> leicht umstrukturiert. Dadurch konnte eine if-Bedingung entfallen, was die Verschachtelung reduziert. Gleichzeitig habe ich damit drei Zeilen wiederholter Code eingespart.	Positiv: Der Code ist schlanker und einfacher nachvollziehbar, unnötige Wiederholungen wurden entfernt. Die Konsolentexte sind klarer formuliert und für den Nutzer verständlicher.	Kleine strukturelle Verbesserungen steigern die Lesbarkeit und reduzieren künftigen Wartungsaufwand. Auch Konsolenausgaben gehören zur Codequalität, da sie das Nutzererlebnis prägen.

			Zusätzlich habe ich kleinere Anpassungen vorgenommen, um die Lesbarkeit insgesamt zu verbessern – insbesondere bei den Fliesstexten in deutscher Sprache (Texte der Konsolenausgaben).	Negativ: Diese Arbeiten hatten keinen direkten Einfluss auf die Funktionalität, sondern dienten rein der Qualitätssicherung. Dadurch musste etwas zusätzliche Zeit investiert werden.	
40 01.10.	0:30	JV	<p>Code Testen</p> <p>Ich habe unser Programm getestet und mehrere Programmdurchläufe durchgeführt. Dabei stellte ich fest, dass die Ausflugsziele zwar korrekt aus der Datenbank eingelesen wurden, jedoch nicht passend zu meinen Präferenzen ausgegeben wurden.</p> <p>Es scheint, dass beim Einlesen der Ausflugsziele aus der Datenbank die Attribute vertauscht in das Place-Objekt gespeichert werden. Dadurch würde die qualitative Bewertung der Ausflugsziele verfälscht. Ich habe diesen Bug Oleksandr gemeldet, damit er die Korrektur übernimmt.</p>	<p>Positiv: Ich konnte ein Fehler frühzeitig entdecken und dem zuständigen Teammitglied zur Korrektur übergeben werden.</p> <p>Negativ: Der Bug hat mich Zeit gekostet, da die Ursache nicht sofort erkennbar war.</p>	<p>Tests sind entscheidend, um auch logische Fehler (nicht nur technische Abstürze) aufzudecken.</p> <p>Es ist wichtig, Beobachtungen klar zu dokumentieren und im Team zu kommunizieren, damit Probleme gezielt gelöst werden können.</p> <p>Ein Fehler im Datenfluss kann die gesamte Logik verfälschen – deshalb müssen Input, Verarbeitung und Output systematisch überprüft werden (vgl. EVA-Prinzip).</p>
40 02.10.	0:45	JV	<p>Code Kommentieren</p> <p>Im Rahmen der Systemdokumentation habe ich den Code mit deutschen Kommentaren versehen.</p> <p>Mein Zeil dabei war, dass eine fremde Person, die noch nichts von unserem Projekt versteht, den Code lesen und verstehen kann. Dies ohne viel Zeit aufzuwenden.</p>	<p>Positiv: Der Quellcode ist nun sehr lesbar und kann auch von einer Person mit nicht so vielen Programmierkenntnissen verstanden werden, vorausgesetzt die Person kennt das Konzept von Klassen / Objekten.</p>	<p>Kommentare im Code wirken zwar manchmal wie «verschwendeter Platz»: Sie machen den Code insgesamt länger. Doch es dient der Lesbarkeit.</p> <p>Ich habe es selbst schon oft erlebt, dass ich eines meiner eigenen (!)</p>

					Projekte nach 6 Monaten Pause wieder hervorgehoben habe und den Code nur noch schwer verstehen konnte. In diesem Fall verliert man sehr viel Zeit, da das Einlesen stark erschwert ist ohne gute Kommentare im Code.
40 2.10.	0:45	JV	Verfassen der Systemdokumentation Nachdem ich den Quellcode verstanden, grob getestet und kommentiert hatte war das Verfassen der System-Dokumentation nicht mehr schwierig: Ich kannte alle Details bereits und musste nur noch meine Gedanken strukturieren und zu Papier bringen.	Positiv: Ich wollte eine neue Art von Diagramm-Erstellung testen. Wenn ich einen UML-Text verfasse, kann ich diesen nachher in ein Diagramm umwandeln (z.B. mit PlantUML). Das hat erstaunlich gut funktioniert.	Eine Systemdokumentation ist wichtig, weil damit dem Projekt fremde Personen einen schnellen Einstieg in den Quellcode und die Anwendung des Programms finden können.
40 2.10.	0:10	JV	Pendenzabgleich und Teamkoordination Nachdem Verfassen der Systemdokumentation habe ich mit dem Team Rücksprache gehalten. Ich habe Adrian informiert, dass er nun mit den Testfällen beginnen kann, da alle vorherigen Schritte vollendet waren.	Positiv: Dank einer guten Kommunikation im Team konnten die Arbeitsschritte in der korrekten Reihenfolge und zu einem sinnvollen Zeitpunkt erledigt werden. Damit ist die Realisierungsphase ohne grosse Zwischenfälle verlaufen.	Eine gute Kommunikation im Team ermöglicht eine genaue Abstimmung der Arbeitsschritte aufeinander. Damit verhindern wir, dass Missverständnisse zu Inkonsistenzen im Arbeitsergebnis führen.
42 14.10.	0:15	JV	Lieferobjekt .exe-Datei finalisieren Nachdem die letzten Feinschliffe am Programmcode gemacht wurden, habe ich die finale Version kompiliert.	Positiv: Der Release-Build ist stabil und startet ohne zusätzliches .NET installiert; das Programm verhält sich wie spezifiziert. Buildprozess reproduzierbar	Ich denke, dass die Reproduzierbarkeit wichtig ist: Ich habe den Build-Befehl deshalb in die Systemdokumentation mit aufgenommen.

			<p>Build im <i>Release-Modus (win-x64)</i>, <i>self-contained Single File</i> via <i>dotnet publish</i> erstellt, Artefakte geprüft (Start per Doppelklick), und das Lieferpaket zusammengestellt (.exe + ziele.db).</p> <p>Testlauf auf Zweitrechner zur Verifikation.</p>	<p>dokumentiert (Befehl + Pfad) in der Projektdokumentation.</p> <p>Negativ: Dateigrösse durch <i>self-contained</i> relativ hoch; <i>ziele.db</i> muss weiterhin separat mitgeliefert werden.</p>	<p>Smoke-Tests nach dem Build (Zweitrechner) vermeiden Überraschungen bei der Abgabe.</p> <p>Trennung von Code (exe) und Daten (DB) ist sinnvoll: Datenupdates sind so ohne Neu-Build möglich.</p>
42 14.10	0:15	JV	<p>Überarbeitung und Abschluss des Arbeitsjournals</p> <p>Alle bisherigen Einträge wurden geprüft, vereinheitlicht und sprachlich nachgeschärft. Dadurch ist mein Journal vollständig und abgabefertig.</p>	<p>Positiv: Das Journal vermittelt nun einen klaren und einheitlichen Überblick über die gesamte Realisierungsphase. Es zeigt den roten Faden und die Lernfortschritte deutlich.</p>	<p>Ein einheitliches Arbeitsjournal ist wichtig. So kann auch lange Zeit später noch nachvollzogen werden, was die Arbeitsschritte waren. Damit kann ich meine eigene erarbeitete Erfahrung später viel einfacher wieder nachvollziehen und für weitere Projekte einsetzen.</p>

8.2 Adrian Herbst

KW	Zeit	Kürzel	Zielsetzung / Tätigkeit	Erfüllungsgrad Positives / Negatives	Reflexion/Fazit Lessons learnt
41 4.10	4:15	AH	Durchführung der Fragebögen und auswerten des Feedbacks	Via Fragebögen habe ich unser Programm getestet und von diversen Personen Feedback eingeholt. Dafür habe ich Screenshots von unserem Konsolenprogramm ausgedruckt und diese den Personen gegeben. Danach habe ich diese ausgewertet und das entsprechende Reiseziel vorgeschlagen.	Durch das Auswerten der Feedbacks habe ich einen guten Eindruck bekommen, wie sich der Average-User verhält. Das Feedback war weitgehend positiv, was mich sehr erfreute. Zudem waren einige Personen am Code interessiert, welchen ich mit ihnen dann anschaute und erklärte.
41 7.10	2:15	AH	Testfälle durchführen und diese Dokumentieren.	Ich habe die verschiedenen Testfälle getestet und dokumentiert. Viele Testfälle waren super umgesetzt und ich konnte schnell und effizient die Testfälle und Anforderungen abarbeiten.	Das Testen und Dokumentieren haben mir sehr Spass gemacht und ich konnte unser grossartiges Programm austesten. Ich verstehe nun, wie wichtig es ist gewisse Testfälle im Voraus zu beschreiben, um durch die Umsetzung dieser ein qualitatives Produkt zu erhalten.
41 7.10	1.45	AH	Dokumentation der Testfälle auswerten und eventuelle Pendenzen festhalten.	Wir haben sowohl die Funktionalen wie auch die Nicht-Funktionalen Anforderungen fast alle erfüllt. Die einzigen nicht erfüllten Anforderungen, sind grob erfüllt, jedoch bei der Programmierung	Beim Auswerten war ich sehr stolz auf unsere Gruppe, da wir so gut wie alle Testfälle abgedeckt und erfüllt haben. Ich bin mit dem Finalen Programm sehr zufrieden und es

				stellten wir fest, dass es mehr Sinn macht diese nicht 1:1 umzusetzen.	freut mich sehr, dass wir es wie geplant umsetzen konnten.
41 12.10	0.30	AH	Verfassen des Arbeitsjournals	Ich habe meine Notizen und Gedanken, welche ich mir beim Bearbeiten der Testfälle gemacht habe in die Dokumentation hinzugefügt.	Ich konnte beim Schreiben des Arbeitsjournal noch einige Dinge Reflektieren. Zudem hilft es mir sehr, im Nachhinein zu sehen, was ich wann gemacht habe.

8.3 Marco Saez

KW	Zeit	Kürzel	Zielsetzung / Tätigkeit	Erfüllungsgrad Positives / Negatives	Reflexion/Fazit Lessons learnt
37 09.09.	0:20	MS	<p>Datenbank anlegen</p> <p>Mir wurde von Jonas gezeigt, wie man im DB-Browser (SQLite) eine Datenbank anlegt und die Attribute korrekt einschränkt.</p> <p>Auch habe ich diese mit drei Beispielzielen befüllt und an Oleksandr weitergegeben, damit er den Aufbau der Datenbank fürs Programmieren hat.</p>	<p>Positiv: Die Instruktion des Erstellens hat gut geklappt, sodass ich mit der Datenbank weiterarbeiten konnte.</p> <p>Negativ: Die Einschränkungen der Attribute (Wertebereich 1–10) war anfangs nicht intuitiv.</p>	<p>Ich habe gelernt, wie eine Datenbank im DB-Browser (SQLite) angelegt wird.</p> <p>Auch sind Attribut-Einschränkungen (Constraints) wichtig, damit die Datenbank konsistent bleibt.</p>
38 20.09.	0:30	MS	<p>Orte Kategorisieren</p> <p>Ich habe verschiedene mögliche Kategorien für Ausflugsziele zusammengesucht. Im Anschluss habe ich diese in einem Mindmap festgehalten.</p>	<p>Positiv: Die Kategorisierung der Ausflugsziele ist sehr nützlich für die weitere Suche der konkreten Ausflugsziele.</p> <p>Negativ: Der Zwischenschritt der Kategorien ist eigentlich nicht notwendig.</p>	<p>Die Kategorien, welche für Ausflugsziele verwendet werden können, sind sehr divers. Einige Kategorien sind mir nicht als Ausflugsmöglichkeit bekannt gewesen.</p>
40 24.09.	1:15	MS	<p>Orte heraussuchen</p> <p>Ich habe basierend auf die vorher erarbeiteten Kategorien verschiedene Ausflugsziele gesucht.</p> <p>Ich habe ausserdem versucht möglichst geografisch verschiedene Orte zu nehmen beziehungsweise darauf zu achten, dass möglichst die ganze Schweiz vertreten ist.</p>	<p>Positiv: Die Datenbank der Ausflugsziele ist aus meiner Sicht sehr divers, sodass es für die meisten ein passendes Ausflugsziel gibt.</p>	<p>Durch die vorherige Kategorisierung wurden die Ausflugsziele mit der Top-down Methode erarbeitet. So konnten die Ausflugsziele organisierter herausgesucht werden.</p>

			<p>Auch habe ich einen Fokus darauf gelegt möglichst diverse Ausflugsziele zu wählen.</p> <p>Die Ausflugsziele wurden in einem Excel aufgelistet und teilweise mit einem Kommentar ergänzt.</p>		
<p>39 25.09.</p>	0:30	MS	<p>Orte bewerten und weitergeben</p> <p>In der Exceldatei habe ich die drei Spannungsfelder Kultur, Aktivität und Indoor/Outdoor bewertet.</p> <p>Im Anschluss habe ich diese an Jonas weitergegeben, damit die Einordnung der Attribute nicht nur subjektiv von einer Person abhängig ist.</p>	<p>Positiv: Die Bewertungen von uns sind weitgehend übereinstimmend. Dies bedeutet, dass unsere Ausflugsziele gut eingeteilt wurden.</p>	<p>Die Bewertung durch mehrere Personen führt zu einer objektiveren Perspektive der Attribute. Dadurch werden die Ergebnisse passender ausgegeben.</p>
<p>40 01.10.</p>	0:30	MS	<p>Excel in Datenbank migriert</p> <p>Aus dem vorherigen Excel habe ich ein Excel mit der Struktur der Datenbank gemacht und dieses als CSV-Datei gespeichert.</p> <p>Anschließend habe ich versucht die Daten aus dem CSV-Dokument in die Datenbank zu migrieren. Da dies nicht funktioniert hat habe ich eine neue Datenbank mit der gleichen Struktur erstellt. In die neue Datenbank konnte die CSV-Datei problemlos migriert werden.</p>	<p>Positiv: Die Migration der Daten in die Datenbank hat zu einer effizienten Übertragung der Daten geführt.</p> <p>Negativ: Ich konnte die CSV-Datei nicht in die bestehende Datenbank migrieren, weshalb ich eine neue Datenbank erstellt habe.</p>	<p>Die Datenübertragung von Excel in eine Datenbank im DB-Browser mit SQLite kann am effizientesten über eine CSV-Datei erreicht werden.</p>
<p>41 10.10.</p>	0:25	MS	<p>Verfassen des Arbeitsjournals</p> <p>Ich habe die einzelnen Arbeitsschritte beschrieben. Die Journaleinträge habe ich nochmals überprüft und schliesse es somit ab.</p>	<p>Positiv: Durch das Journal habe ich eine Übersicht über die Realisierungsphase, sowie über die erlernten Dinge.</p>	<p>Das Lernjournal ist wichtig, um später noch die jeweiligen Schritte der Realisierungsphase zu verstehen und diese für zukünftige Projekte zu nutzen.</p>

8.4 Oleksandr Kolomiets

KW	Zeit	Kürzel	Zielsetzung / Tätigkeit	Erfüllungsgrad Positives / Negatives	Reflexion/Fazit Lessons learnt
39 28.09	0:15	OK	Recherche im Internet – Wie man Daten aus der SQLite Datenbank in C# importiert	Ich konnte viele nützliche Informationen finden und verstehen, wie man mit SQLite in C# arbeitet und welche Bibliothek man dafür benutzen soll.	Die Recherche hat mir geholfen, den Umgang mit Datenbanken in C# besser zu verstehen. Ich habe gelernt, gezielt nach Lösungen zu suchen und Dokumentationen effektiv zu nutzen.
39 28.09.	2:15	OK	Erstellung des Projekts und Programmieren	Das Erstellen des Projekts in Visual Studio verlief ohne Probleme. Ich konnte die Grundstruktur und alle benötigten Dateien anlegen. Beim Schreiben des Codes habe ich darauf geachtet, dass er sauber und übersichtlich ist. Ausserdem habe ich einfache Kommentare hinzugefügt, damit Jonas später ausführlichere Kommentare ergänzen kann.	Ich habe das Projekt nach der Dokumentation erstellt, die wir in der Konzeptphase definiert haben. Dadurch war die Umsetzung klar und effizient. Ich habe gelernt, wie wichtig eine gute Planung für einen reibungslosen Start ist.

<p>40 29.09.</p>	<p>0:30</p>	<p>OK</p>	<p>Testen der Anwendung und Übergabe an das Team</p>	<p>Nach dem Testen habe ich die Anwendung dem Team gezeigt, um Feedback zu erhalten. Jonas hat bemerkt, dass das Ergebnis nicht seinen Erwartungen entsprach. Nach einer kurzen Analyse habe ich den Fehler im SQL-Query gefunden – ich hatte zwei Attribute vertauscht. Nach der Korrektur funktionierte alles wie geplant.</p>	<p>Durch diesen Test habe ich gelernt, wie wichtig gründliches Testen und sauberes Arbeiten mit Datenbanken ist. Auch das Feedback vom Team war hilfreich, um den Code weiter zu verbessern.</p>
<p>40 29.09.</p>	<p>0:20</p>	<p>OK</p>	<p>Verfassen des Arbeitsjournals</p>	<p>Durch das Journal habe ich die Projektarbeit nochmals reflektiert und meine Fortschritte erkannt.</p>	<p>Das Lernjournal unterstützt, die Realisierungsphase besser zu verstehen und die Erfahrungen für kommende Projekte zu verwenden.</p>