

# Muay Turkish Aviation



Marwin Kohl & Deniz Ugur  
Technische Berufsschule Zürich  
Modul 254: Geschäftsprozesse beschreiben

Dokumentation



# Inhaltsverzeichnis

## Inhalt

EINLEITUNG .....	3
BESCHREIBUNG DES PROZESSES .....	4
EIGENSCHAFTEN UNSERES BUCHUNGSTOOLS.....	4
VERWENDETE TECHNOLOGIEN .....	5
CAMUNDA .....	5
VISUAL STUDIO CODE .....	5
REACT.JS.....	6
VORGEHENSWEISE .....	7
IDEENFINDUNG .....	7
PROJEKTAUFSETZUNG .....	7
REALISIERUNG .....	8
ENDPRODUKT .....	9
FRONTEND .....	10
BPMN .....	12
POOL: USER .....	12
POOL: MUAY TURKISH AVIATION .....	13
WICHTIGE EREIGNISSE UND GATEWAYS .....	13
FORMULAR.....	14
DMN .....	15
DMN-DECISION-TABELLE: "CALCULATE PRICE" .....	15
ENTSCHEIDUNGSLOGIK-BILD (DECISION TABLE) .....	15
FAZIT .....	16
MARWIN'S EIGENSCHÄTZUNG .....	16
DENIZ'S EIGENSCHÄTZUNG.....	16

# Einleitung

Willkommen zur Dokumentation des Projekts "Muay Turkish Aviation". In einer Zeit, in der Reisen immer zugänglicher und häufiger werden, spielt die Buchung von Flugtickets eine entscheidende Rolle. "Muay Turkish Aviation" ist ein innovatives Flugzeugticket-Buchungstool, das darauf abzielt, die Buchungserfahrung für Passagiere zu optimieren und den gesamten Prozess effizienter und benutzerfreundlicher zu gestalten.

Unser Ziel mit "Muay Turkish Aviation" ist es, eine Plattform zu schaffen, die den Anforderungen moderner Reisender gerecht wird. Wir legen besonderen Wert auf eine intuitive Benutzeroberfläche, schnelle Buchungsprozesse und umfassende Suchoptionen, um den idealen Flug für jeden Kunden zu finden. Darüber hinaus bieten wir massgeschneiderte Lösungen für verschiedene Reisebedürfnisse und -vorlieben an, um ein nahtloses und zufriedenstellendes Buchungserlebnis zu gewährleisten.

Diese Dokumentation dient als umfassender Leitfaden durch die Entwicklung, die Funktionen und die Nutzung unseres Buchungstools. Sie richtet sich sowohl an Endbenutzer, die die Vorteile der Plattform nutzen möchten, als auch an Entwickler und Partner, die sich für die technischen Aspekte und die Implementierung interessieren. Wir laden Sie ein, mit uns auf diese Reise zu gehen und die Möglichkeiten von "Muay Turkish Aviation" zu entdecken.

# Beschreibung des Prozesses

Das Ziel unseres Projekts "Muay Turkish Aviation" ist es, Reisenden die Möglichkeit zu bieten, ihre Reisen mühelos und effizient zu planen und zu buchen. In einer global vernetzten Welt, in der Mobilität eine zentrale Rolle spielt, benötigen Reisende ein zuverlässiges und benutzerfreundliches Buchungstool, um ihre Reisepläne problemlos umsetzen zu können.

"Muay Turkish Aviation" wurde entwickelt, um genau diese Bedürfnisse zu erfüllen. Unser Buchungstool ermöglicht es den Nutzern, schnell und unkompliziert Flugtickets zu suchen und zu buchen. Dabei legen wir besonderen Wert auf eine intuitive Benutzeroberfläche, die es auch technisch weniger versierten Personen erlaubt, ihre Flüge problemlos zu buchen.

Unser Tool zielt darauf ab, den gesamten Buchungsprozess zu vereinfachen und zu beschleunigen, sodass unsere Nutzer mehr Zeit haben, sich auf das Wesentliche ihrer Reise zu konzentrieren.

So kann man sich unser Prozess in einem BPMN-Diagramm vorstellen:

## Eigenschaften unseres Buchungstools

Unser Buchungstool "Muay Turkish Aviation" bietet eine Vielzahl von Eigenschaften, die den Buchungsprozess für Reisende erleichtern und verbessern. Die wichtigsten Funktionen im Überblick:

- **Abflugort und Zielort:** Benutzer können den Abflughafen und das gewünschte Reiseziel eingeben. Unser Tool bietet eine umfassende Suchfunktion, die es ermöglicht, aus einer Vielzahl von Flughäfen weltweit zu wählen.
- **Abflug und Rückflug:** Es können sowohl einfache Flüge als auch Hin- und Rückflüge gebucht werden. Das Tool erlaubt die Auswahl von Abflug- und Rückflugdatum, wobei flexible Datumsoptionen zur Verfügung stehen, um den besten Preis zu finden.
- **Anzahl Erwachsene:** Benutzer können die Anzahl der Erwachsenenreisenden angeben, um die Suche und Buchung entsprechend anzupassen.
- **Anzahl Kinder:** Ebenso kann die Anzahl der mitreisenden Kinder angegeben werden, wobei spezielle Kinderpreise und -optionen berücksichtigt werden.

# Verwendete Technologien

## Camunda

Camunda ist eine ausgezeichnete Wahl für die Modellierung von Geschäftsprozessen, insbesondere für komplexe Anwendungen wie ein Flugzeigticket-Buchungstool aus mehreren Gründen:

1. **BPMN-Standard:** Camunda basiert auf dem BPMN-Standard (Business Process Model and Notation), der eine klare und standardisierte Darstellung von Geschäftsprozessen ermöglicht. Dies erleichtert die Modellierung, Kommunikation und Implementierung der Prozesse.
2. **Flexibilität und Anpassbarkeit:** Camunda bietet eine flexible Plattform, die es erlaubt, Geschäftsprozesse genau nach den spezifischen Anforderungen eines Flugzeigticket-Buchungstools zu modellieren. Man kann komplexe Abläufe, Entscheidungslogik und Interaktionen zwischen verschiedenen Systemen einfach abbilden.
3. **Integration:** Camunda lässt sich gut in bestehende IT-Infrastrukturen integrieren. Dies ist besonders wichtig für ein Buchungstool, das möglicherweise mit anderen Systemen wie Zahlungsabwicklung, Flugdatenbanken und Kundeninformationssystemen kommunizieren muss.
4. **Workflow-Management:** Durch die Workflow-Management-Fähigkeiten von Camunda können Geschäftsprozesse automatisiert und überwacht werden. Dies führt zu einer effizienten Durchführung der Buchungsvorgänge und ermöglicht eine schnelle Reaktion auf Änderungen oder Probleme.
5. **Skalierbarkeit und Performance:** Camunda ist darauf ausgelegt, auch bei grossen Mengen von Transaktionen und komplexen Prozessabläufen stabil und performant zu bleiben. Dies ist entscheidend für ein Buchungstool, das potenziell viele gleichzeitige Benutzer hat.

Insgesamt bietet Camunda eine robuste Plattform für die Modellierung, Automatisierung und Überwachung von Geschäftsprozessen, die speziell für anspruchsvolle Anwendungen wie ein Flugzeigticket-Buchungstool geeignet ist.

## Visual Studio Code

Visual Studio Code (VS Code) ist eine leichtgewichtige, aber leistungsstarke integrierte Entwicklungsumgebung (IDE), die von Microsoft entwickelt wurde. Hier sind einige Gründe, warum VS Code eine gute Wahl für die Entwicklung eines Flugzeigticket-Buchungstools sein könnte:

1. **Erweiterbarkeit:** VS Code bietet eine grosse Auswahl an Erweiterungen, die die Funktionalität der IDE erheblich erweitern können. Für die Entwicklung eines Buchungstools könnten Erweiterungen für HTML, CSS, JavaScript/TypeScript, sowie für Frameworks wie React, Angular oder Vue.js nützlich sein.
2. **Sprachunterstützung:** VS Code bietet Unterstützung für eine Vielzahl von Programmiersprachen und Technologien. Dies ist besonders hilfreich, da moderne Webanwendungen oft aus einer Kombination von Frontend- und Backend-Technologien bestehen.
3. **Integrierte Tools:** VS Code verfügt über integrierte Tools für Versionskontrolle (Git), Debugging, und Terminalintegration, die die Entwicklung und Zusammenarbeit erleichtern.

4. **Performance:** VS Code ist für seine Geschwindigkeit und Reaktionsfähigkeit bekannt, selbst bei grossen Codebasen. Dies ist wichtig für die effiziente Entwicklung und Bearbeitung von Code in einem komplexen Projekt wie einem Buchungstool.
5. **Plattformübergreifend:** VS Code ist plattformübergreifend verfügbar (Windows, macOS, Linux), was die Flexibilität für Entwickler erhöht, unabhängig von ihrer Betriebssystemwahl.
6. **Community und Unterstützung:** VS Code hat eine grosse und aktive Community, die sich ständig weiterentwickelt und neue Erweiterungen und Ressourcen bereitstellt. Dies ist besonders hilfreich für die Lösung von Problemen und das Lernen neuer Techniken.

Für ein Flugzeuggticket-Buchungstool, das moderne Webtechnologien und eine benutzerfreundliche Benutzeroberfläche erfordert, bietet Visual Studio Code die notwendigen Werkzeuge und Erweiterungen, um eine effiziente Entwicklung zu ermöglichen und die Anforderungen des Projekts zu erfüllen.

## React.js

React.js ist eine JavaScript-Bibliothek, die speziell für den Bau von Benutzeroberflächen entwickelt wurde. Hier sind einige Gründe, warum es eine gute Wahl für ein Flugzeuggticket-Buchungstool als Frontend-Framework ist:

1. **Komponentenbasierte Architektur:** React.js basiert auf einer komponentenbasierten Architektur, was bedeutet, dass die Benutzeroberfläche in wiederverwendbare, unabhängige Komponenten unterteilt wird. Dies ist ideal für komplexe Anwendungen wie ein Buchungstool, da verschiedene Teile der Benutzeroberfläche (wie Suche, Ergebnisliste, Buchungsformular) separat entwickelt und gewartet werden können.
2. **Virtual DOM:** React verwendet einen Virtual DOM, der Änderungen effizient verwaltet und nur die tatsächlich geänderten Teile der Benutzeroberfläche neu rendert. Das ist besonders wichtig für Anwendungen mit dynamischen Inhalten wie ein Buchungstool, das regelmässig aktualisiert wird, wenn neue Flüge verfügbar werden oder Buchungen durchgeführt werden.
3. **Einfache Integration:** React kann gut mit anderen JavaScript-Bibliotheken und -Frameworks sowie mit Backend-Technologien integriert werden. Das ermöglicht eine flexible Architektur, bei der React für die Benutzeroberfläche verantwortlich ist, während andere Technologien für die Datenverarbeitung und -speicherung verwendet werden können.
4. **Reaktive Updates:** Durch den Einsatz von Zustandsmanagement (z.B. mit Hooks oder Redux) können Änderungen im Zustand der Anwendung effizient behandelt werden. Das ist besonders nützlich für ein Buchungstool, um den Buchungsprozess nahtlos zu verwalten und dem Benutzer Echtzeit-Updates zu bieten.
5. **Grosse Community und Ökosystem:** React wird von einer grossen Entwicklergemeinschaft unterstützt, was bedeutet, dass es eine Vielzahl von vorgefertigten Komponenten, Bibliotheken und Tools gibt, die die Entwicklung beschleunigen können. Für ein komplexes Projekt wie ein Buchungstool ist diese Unterstützung von unschätzbarem Wert.

Insgesamt ist React.js eine ausgezeichnete Wahl für die Entwicklung eines Flugzeuggticket-Buchungstools aufgrund seiner Leistungsfähigkeit, Flexibilität und der breiten Unterstützung durch die Entwicklergemeinschaft.

# Vorgehensweise

Hier dokumentieren wir den Entwicklungsprozess unseres Projekts für das Flugzeugticketbuchungstool "Muay Turkish Aviation", von der initialen Ideenfindung bis zur Fertigstellung des Endprodukts. Dabei beleuchten wir unsere Herangehensweise bei der Bewältigung von Hindernissen und Schwierigkeiten sowie die Lösungen, die wir auf dem Weg gefunden haben.

## Ideenfindung

Die Idee für unser Flugticketbuchungstool-Projekt "Muay Turkish Aviation" entspringt einer Vielzahl persönlicher Interessen und Beobachtungen über die Bedürfnisse moderner Reisender. Wir haben erkannt, dass das Reisen eine universelle Leidenschaft ist, die Menschen aus allen Lebensbereichen und Kulturen verbindet. Ob für geschäftliche Verpflichtungen, Familienbesuche oder persönliche Abenteuer – die Möglichkeit zu reisen ist für viele ein wesentlicher Bestandteil des Lebens.

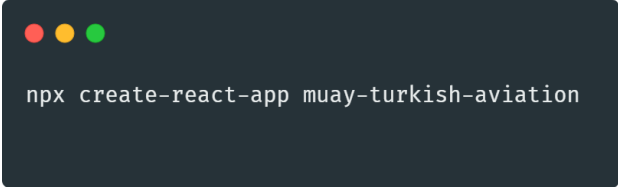
Ein weiterer wichtiger Motivationsfaktor war unser Bestreben, eine tolle Schulnote zu erzielen. Mit einem Projekt, das noch niemand an unserer Schule durchgeführt hat, wollten wir uns von den anderen Projekten abheben und eine einzigartige Leistung hervorbringen. Die Umsetzung eines Flugticketbuchungstools bietet uns die Möglichkeit, sowohl technische als auch kreative Herausforderungen zu meistern und damit unser Können unter Beweis zu stellen.

## Projektaufsetzung

Da wir Visual Studio Code verwenden ist das Aufsetzen eines React.js Projektes ganz einfach. Dazu sollte man zunächst in das gewünschte Verzeichnis navigieren, in dem das Projekt gespeichert werden soll. Dies kann durch Öffnen von Visual Studio Code und Auswahl des Menüs "File" und dann "Open Folder..." erfolgen. Anschliessend kann man das Terminal in Visual Studio Code öffnen, indem man auf "Terminal" in der oberen Menüleiste klickt und dann "New Terminal" auswählt.

```
PS C:\Users\marwin.kohl\TBZLocal\m245\muay-turkish-aviation>
```

Danach kann man anschliessend mit dem folgenden Befehl ein React.js-Projekt aufsetzen und beginnen:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The command `npx create-react-app muay-turkish-aviation` is entered in the terminal.

```
npx create-react-app muay-turkish-aviation
```

```

PS C:\Users\warwin.kohl\182Local\m245\muay-turkish-aviation> npx create-react-app muay-turkish-aviation
Need to install the following packages:
create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm WARN deprecated uid-number@0.0.6: This package is no longer supported.
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated fstream@1.0.12: This package is no longer supported.
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

Creating a new React app in C:\Users\warwin.kohl\182Local\m245\muay-turkish-aviation\muay-turkish-aviation.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1482 packages in 1m

262 packages are looking for funding
  run `npm fund` for details

Installing template dependencies using npm...

added 63 packages, and changed 1 package in 11s

262 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

removed 1 package, and audited 1545 packages in 3s

262 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Success! Created muay-turkish-aviation at C:\Users\warwin.kohl\182Local\m245\muay-turkish-aviation\muay-turkish-aviation
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd muay-turkish-aviation
  npm start

Happy hacking!
PS C:\Users\warwin.kohl\182Local\m245\muay-turkish-aviation>

```

Dieser Befehl erstellt ein neues React.js-Projekt mit dem Namen "muay-turkish-aviation" und installiert alle notwendigen Abhängigkeiten automatisch. Zum Starten des Projekts benötigt man den folgenden Befehl:



## Realisierung

Für die Realisierung unseres Projekts haben wir Camunda und React verwendet. Zu Beginn hatten wir Schwierigkeiten, da uns das Wissen fehlte, wie man über die API effektiv mit Camunda kommuniziert. Anfangs war es viel Experimentieren, doch im Laufe der Zeit haben wir immer mehr Kenntnisse gewonnen und unsere Fähigkeiten verbessert. Die Realisierung des Projekts hat viel Zeit in Anspruch genommen, da wir beide zum ersten Mal mit diesen Technologien gearbeitet haben. Trotz der anfänglichen Herausforderungen haben wir wertvolle Erfahrungen gesammelt und unser Verständnis sowohl für Camunda als auch für React erheblich erweitert.



## Endprodukt

Unser Projekt wurde erfolgreich abgeschlossen. Wir haben unsere Visionen erfolgreich umgesetzt und eine innovative Webseite entwickelt, über die Nutzer Flugtickets buchen können. Besonders hervorzuheben ist das Design, das nicht nur ästhetisch ansprechend ist, sondern auch eine benutzerfreundliche Navigation bietet.


Ein herausragendes Merkmal unserer Webseite ist die Integration mit Camunda, die es ermöglicht, dass das Buchen eines Tickets automatisch einen Prozess in Camunda auslöst. Diese nahtlose Integration vereinfacht nicht nur den Buchungsprozess für die Nutzer, sondern ermöglicht es uns auch, interne Abläufe effizient zu verwalten und zu optimieren.

Wir sind stolz darauf, ein Produkt geschaffen zu haben, das nicht nur technologisch fortschrittlich ist, sondern auch den Bedürfnissen und Erwartungen unserer Zielgruppe gerecht wird. Diese Webseite repräsentiert unser Engagement für Qualität und Innovation im Bereich der Online-Flugbuchungen.

Wir haben eine Webseite erstellt mit mehreren Seiten:

### Login

Die Loginseite unserer Webseite für Flugbuchungen ist das Tor zu einem personalisierten und sicheren Nutzungserlebnis. Mit einem klaren und modernen Design lädt sie Benutzer ein, sich einfach und intuitiv anzumelden, um ihre Reisepläne zu verwalten.



**Welcome to Muay Turkish Aviation**

Email

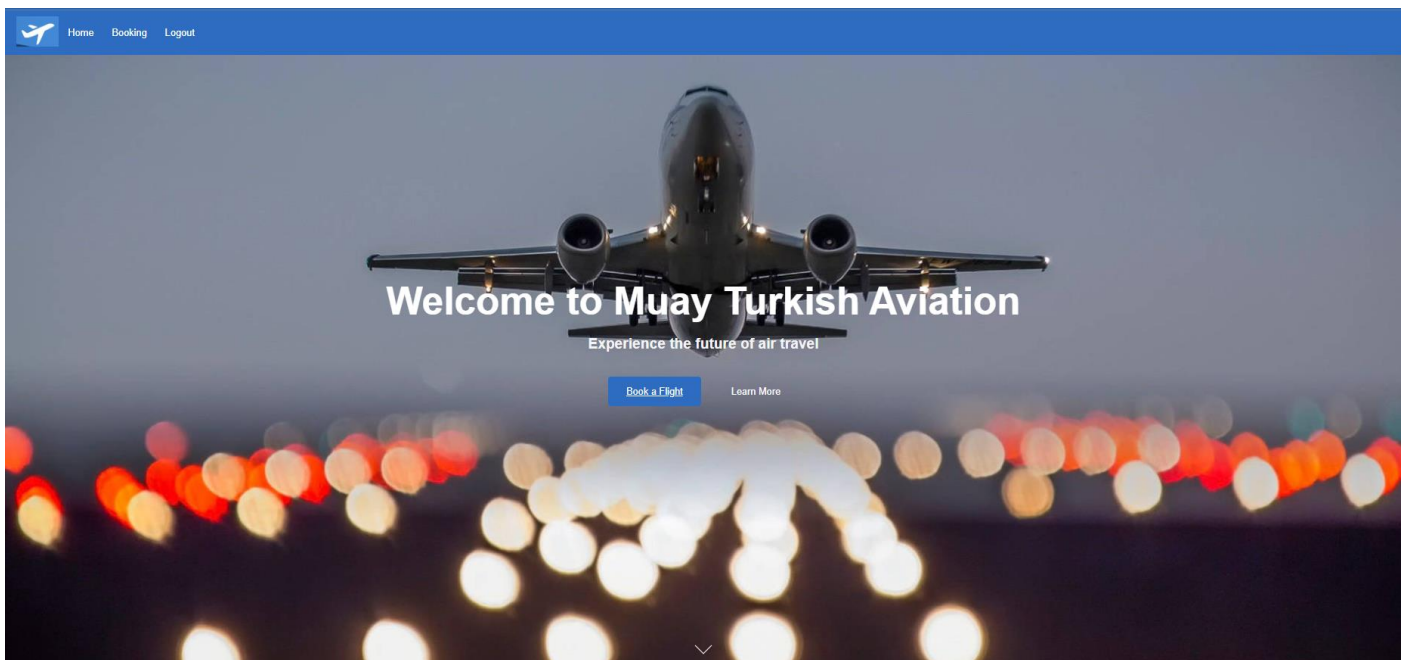
Password

Login

# Frontend

## Homepage

Von unserer Homepage aus kann man verschiedene Aktionen ausüben. Unter anderem kann man mehr Informationen über uns sammeln, sowie direkt einen Flug nach Wahl buchen. Die Preise sind auch auf der Webseite, damit man ohne viel Suchen die direkten Preise sieht.



### Our Features

#### Modern Fleet

Experience state-of-the-art aircraft with unparalleled comfort and safety.

#### Premium Service

Enjoy top-notch service from check-in to landing.

#### Global Destinations

Fly to over 100 destinations worldwide with ease.

### About Us

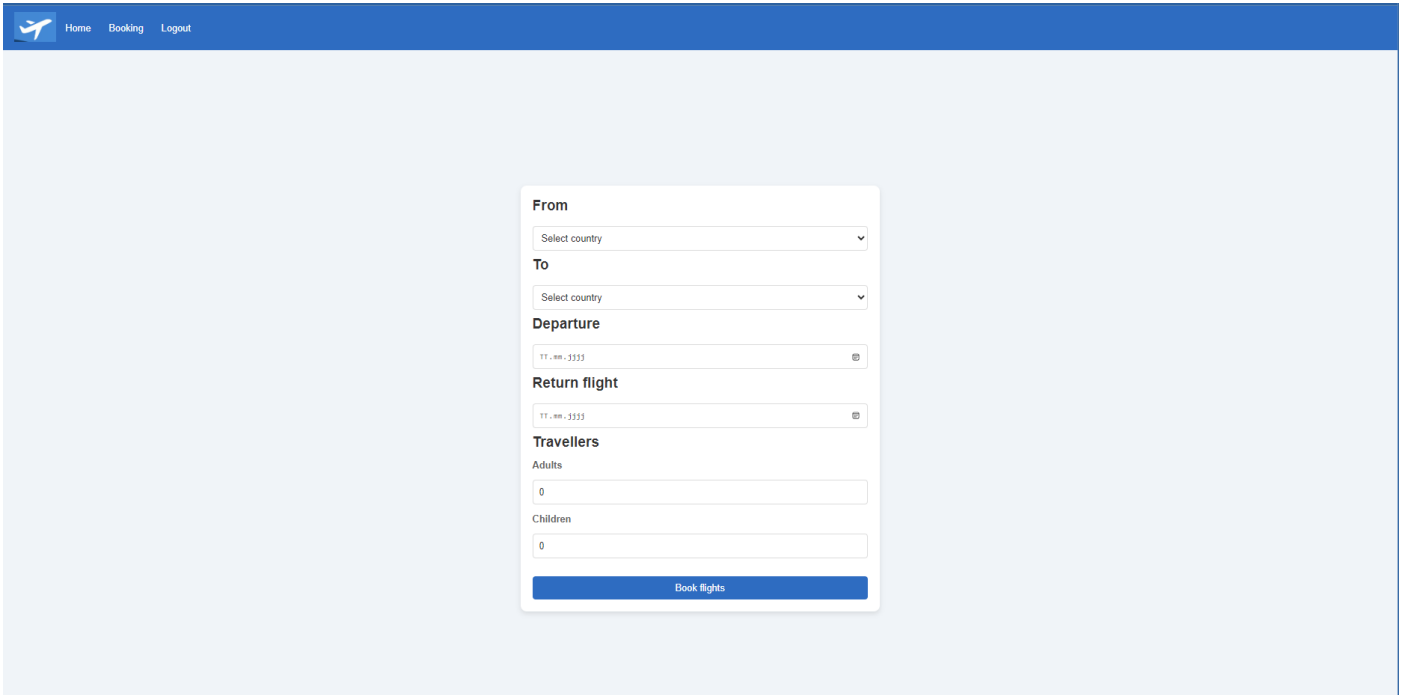
Muay Turkish Aviation is committed to providing the best air travel experience with a focus on innovation, safety, and customer satisfaction.

### Price

Adults 150.-  
Children 120.-

## Buchungsformular

Das Buchungsformular bei unserem Flugticketbuchungstool erfüllt mehrere wichtige Funktionen, die das Buchen eines Fluges für unsere Nutzer einfach und effizient gestalten. Sobald man alle Daten eingetragen hat, kann man einen Flug buchen. Nachdem man den «Book flights» button gedrückt hat, wird ein Prozess in Camunda gestartet. Der Preis wird dann mittels DMN berechnet und in der Tasklist in Camunda ausgegeben.



The screenshot shows a web application interface for booking flights. At the top, there is a blue navigation bar with a logo and links for 'Home', 'Booking', and 'Logout'. The main content area is light blue and contains a white form titled 'From'. The form has the following sections: 'From' and 'To' (each with a 'Select country' dropdown), 'Departure' (with a date/time input field showing '11.06.2025'), 'Return flight' (with a date/time input field showing '11.06.2025'), and 'Travellers' (with 'Adults' and 'Children' input fields, both showing '0'). At the bottom of the form is a blue button labeled 'Book flights'.

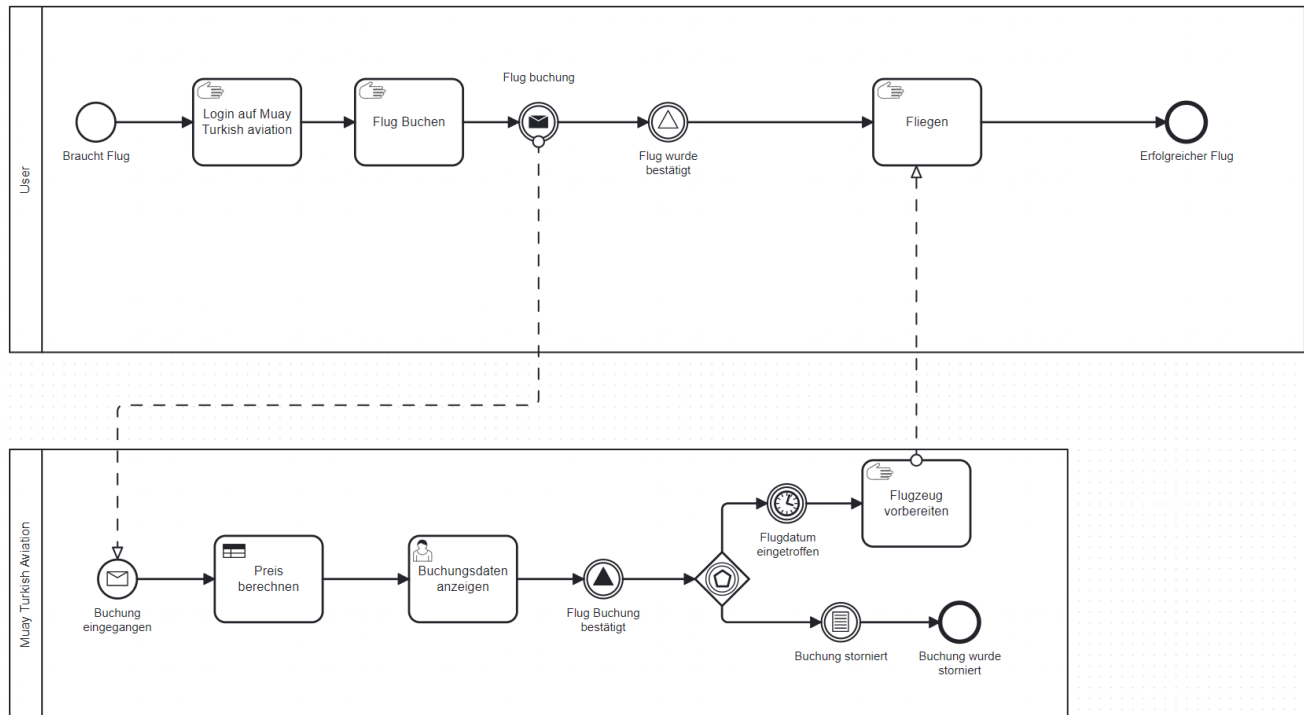
Das hier ist die Logik von dem API call, um das Frontend mit Camunda zu verbinden. Das FormData Objekt wird von den Input Felder im Form zusammengestellt. Danach werden alle daten an die API geschickt. Zuvor wird alles noch zu einem String konvertiert, da wir herausgefunden haben, die API einen String erwartet. Anfangs hatten wir grosse Probleme

```
const formData = {
  variables: {
    fldFrom: { value: countryFrom, type: "String" },
    fldTo: { value: countryTo, type: "String" },
    fldDeparture: { value: departure, type: "String" },
    fldReturnFlight: {
      value: returnFlight,
      type: "String",
    },
  },
  fldAdults: { value: adults, type: "Integer" },
  fldChildren: { value: children, type: "Integer" },
},
};

try {
  const response = await fetch(apiUrl, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(formData),
  });
}
```

# BPMN

Wir haben unser BPMN mit dem Camunda Modeler designt, da wir in diesem Modul bereits ein bisschen Erfahrung mit dieser Technologie.



Das BPMN-Diagramm stellt den Prozess der Buchung und Durchführung eines Fluges dar. Es ist in zwei Pools aufgeteilt: "User" und "Muay Turkish Aviation", die die Interaktion zwischen dem Benutzer und der Fluggesellschaft darstellen.

## Pool: User

**Braucht Flug:** Der Benutzer hat den Bedarf, einen Flug zu buchen.

**Login auf Muay Turkish aviation:** Der Benutzer loggt sich in das Buchungssystem der Fluggesellschaft ein.

**Flug Buchen:** Der Benutzer gibt die Buchungsdaten ein und sendet die Buchung ab.

**Flug wurde bestätigt:** Der Benutzer erhält die Bestätigung, dass der Flug gebucht wurde.

**Fliegen:** Der Benutzer tritt die Reise an.

**Erfolgreicher Flug:** Der Benutzer hat den Flug erfolgreich abgeschlossen.

## Pool: Muay Turkish Aviation

**Buchung eingegangen:** Die Buchungsanfrage des Benutzers wird von der Fluggesellschaft empfangen.

**Preis berechnen:** Die Fluggesellschaft berechnet den Preis für den Flug.

**Buchungsdaten anzeigen:** Die berechneten Buchungsdaten werden angezeigt.

**Flug Buchung bestätigt:** Die Fluggesellschaft bestätigt die Buchung.

**Flugzeug vorbereiten:** Vor dem Flugdatum wird das Flugzeug vorbereitet.

**Flugdatum eingetroffen:** Am Tag des Fluges wird geprüft, ob das Datum eingetroffen ist.

- Wenn ja, wird das Flugzeug für den Flug vorbereitet.
- Wenn die Buchung storniert wurde, endet der Prozess hier mit "Buchung wurde storniert."

## Wichtige Ereignisse und Gateways

**Zwischenereignis (Nachricht):** Die Buchungsbestätigung und die Stornierung werden als Nachrichten zwischen dem Benutzer und der Fluggesellschaft ausgetauscht.

**Gateway:** Die Entscheidungspunkte prüfen, ob der Flug bestätigt oder storniert wurde und ob das Flugdatum erreicht ist.

# Formular

Das folgende Formular wurde in unserem Camunda Modeler verwendet. Es enthält die Felder:

**From** (Abflugort), **To** (Zielort), **Departure** (Abflugdatum), **Return flight** (Rückflugdatum), **Adults** (Anzahl der Erwachsenen) und **Children** (Anzahl der Kinder).

Form Definition	Form Preview
<p>From</p> <input type="text"/>	<p>From</p> <input type="text"/>
<p>To</p> <input type="text"/>	<p>To</p> <input type="text"/>
<p>Departure</p> <input type="text"/>	<p>Departure</p> <input type="text"/>
<p>Return flight</p> <input type="text"/>	<p>Return flight</p> <input type="text"/>
<p>Adults</p> <input type="text"/>	<p>Adults</p> <input type="text"/>
<p>Children</p> <input type="text"/>	<p>Children</p> <input type="text"/>
<p>Price</p> <input type="text"/>	<p>Price</p> <input type="text"/>

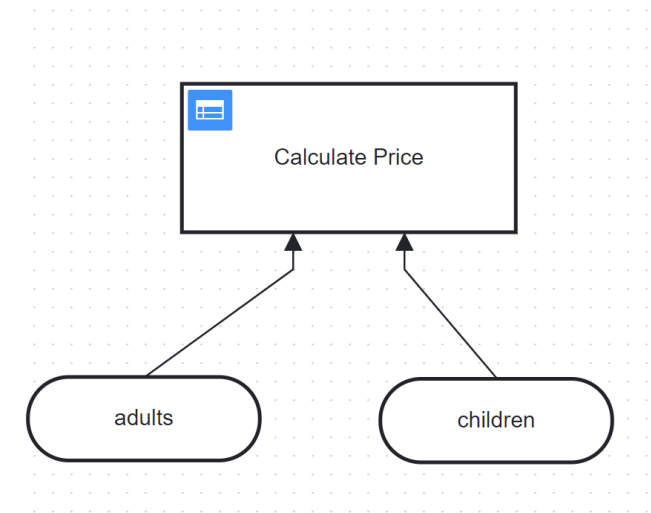
Form Input	Form Output
<pre>1 {}</pre>	<pre>1 { 2   "fldFrom": "", 3   "fldTo": "", 4   "fldDeparture": "", 5   "fldReturnFlight": "", 6   "fldAdults": null, 7   "fldChildren": null, 8   "price": "" 9 }</pre>

Aufgrund der Anforderungen der API war es notwendig, alle Angaben in einen einzigen String zu konvertieren. Daher verwenden wir für die Felder Abflugdatum und Rückflugdatum keine Datums-Inputs, da die API diese Formate nicht unterstützt. Diese Anpassung stellt sicher, dass die Daten korrekt übermittelt und von der API verarbeitet werden können, was eine reibungslose Buchungserfahrung ermöglicht.

# DMN

In unserem Projekt haben wir auch eine DMN (Decision Model and Notation) implementiert, die zur Berechnung der Flugticketpreise dient. Basierend auf dem Alter und der Anzahl der Personen können wir den Gesamtpreis ermitteln. Der Preis für ein Ticket eines Erwachsenen beträgt 150 CHF, während der Preis für ein Kinderticket 120 CHF beträgt.

## DMN-Decision-Tabelle: "Calculate Price"



**Inputs:**

**adults:** Anzahl der Erwachsenen (integer)  
**children:** Anzahl der Kinder (integer)

**Decision:**

**Calculate Price:** Berechnet den Gesamtpreis basierend auf der Anzahl der Erwachsenen und Kinder.

## Entscheidungslogik-Bild (Decision Table)

**Hit Policy: First** Die erste zutreffende Regel wird ausgewählt und die nachfolgenden Regeln werden ignoriert

Calculate Price				
		Hit policy: First		
	adults integer	children integer	price integer	Annotations
1	<=0	<=0	0	
2	>0	>0	floor((adults * 150) + (children * 120))	
3	>0	<=0	floor(adults * 150)	
4	<=0	>0	floor(children * 120)	
+	-	-		

Anfangs hatten wir viel Schwierigkeiten mit dem DMN, da wir noch nie mit dieser Technologie gearbeitet haben, sprich nur schon von ihr gehört. Daher mussten wir zuerst die Struktur und den Aufbau dieser Technologie verstehen, bevor wir anfangen konnten zu programmieren. Ein

gutes Hilfsmittel, um DMN zu verstehen, waren die Beispiele, da wir dort bereits eine funktionierende Lösung vor Augen hatten, die wir tiefgründig Analysieren konnten.

## Fazit

Unser Muay Turkish Aviation Flugticketbuchungstool-Projekt hat sich als eine spannende und lehrreiche Erfahrung erwiesen, die weit über die Entwicklung einer einfachen Webanwendung hinausgeht. Mit einem klaren Ziel vor Augen haben wir es geschafft, eine innovative Plattform zu schaffen, die es Reisenden ermöglicht, ihre Flugbuchungen einfach und effizient durchzuführen.

Durch intensive Teamarbeit und die Nutzung modernster Technologien wie React.js und Camunda haben wir nicht nur technische Fähigkeiten verbessert, sondern auch wichtige Einblicke in die Herausforderungen und Chancen der digitalen Reisebuchungsbranche gewonnen. Die Integration von Camunda zur Automatisierung von Prozessen hat gezeigt, wie fortschrittliche Technologie die Effizienz und Benutzererfahrung verbessern kann.

## Marwin's Eigenschätzung

Bevor ich dieses Modul hatte, wusste ich garnicht was ein Geschäftsprozess ist und was das macht. Ich finde durch die Projekte konnte ich auch vieles Lernen. Seit der LB3 weiss ich jetzt, wie ich ein Geschäftsprozess richtig implementiere und wie ich mit einem Geschäftsprozessmodeller wie Camunda umgehen kann. Insgesamt hat mir das Projekt viel Spass gemacht, aber es war teilweise auch sehr mühsam, die Probleme der API in den Griff zu bekommen. Alles in einem finde ich, ist uns das Projekt sehr gelungen.

## Deniz's Eigenschätzung

Vor diesem Modul hatte ich keinerlei Ahnung, was BPMN und Camunda sind. Mir war auch nicht wirklich bewusst, was ein Geschäftsprozess überhaupt ist. Doch jetzt kann ich mit Stolz sagen, dass ich dies alles gelernt und verstanden habe. Ich bin der Überzeugung, dass ich in diesem Modul viele nützliche Fähigkeiten erlernt habe, die mir in meiner zukünftigen Laufbahn als Informatiker von großem Nutzen sein werden. Besonders hilfreich war die praktische Anwendung der theoretischen Kenntnisse in den Projekten. Obwohl es teilweise herausfordernd war, hat mir das Modul insgesamt viel Freude bereitet.