# PlanMyTrip

Pflichtenheft

Projekt:

Version:

Datum:

Auftraggeber:

Autor/Entwickler:

Website-Applikation „PlanMyTrip“

1.1

16.11.2020

IBB

Mario Phillipp

# Historie

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Änderung** | | | **Geänderte**  **Kapitel** | **Beschreibung der**  **Änderung** | **Autor** | **Zustand** |
| **Nr.** | **Datum** | **Version** |
| 1 | 16.11.2020 | 1.0 | Alle | Initiale Produkterstellung | Mario Phillipp | Vorgelegt |
| 2 | 12.03.2021 | 1.1 | Alle | Aktualisiert | Mario Phillipp | Vorgelegt |

# Verteiler

|  |  |  |  |
| --- | --- | --- | --- |
| **Rolle** | **Name** | **Erreichbarkeit** | **Bemerkungen** |
| Projektleiter | Mario Phillipp | N.N. |  |
| Ansprechpartner | Thomas Meyer | N.N. |  |

Inhaltsverzeichnis

[PlanMyTrip 1](#_Toc65750685)

[Historie 2](#_Toc65750686)

[Verteiler 2](#_Toc65750687)

[1. Zielbestimmung 4](#_Toc65750688)

[1.1 Musskriterien 4](#_Toc65750689)

[1.2 Wunschkriterien 4](#_Toc65750690)

[1.3 Abgrenzungskriterien 4](#_Toc65750691)

[2. Produkteinsatz 5](#_Toc65750692)

[2.1 Anwendungsbereiche 5](#_Toc65750693)

[2.2 Zielgruppen 5](#_Toc65750694)

[3. Produktübersicht 6](#_Toc65750695)

[4. Produktfunktionen 6](#_Toc65750696)

[4.1 Geschäftsprozesse 6](#_Toc65750697)

[5. Produktdaten 10](#_Toc65750698)

[5.1 Traveller Daten 10](#_Toc65750699)

[5.2 Location Daten 10](#_Toc65750700)

[5.3 PointOfInterest 10](#_Toc65750701)

[6. Qualitätsanforderungen 10](#_Toc65750702)

[7. Technische Produktumgebung 11](#_Toc65750703)

[7.1 Software 11](#_Toc65750704)

[7.2 Hardware 11](#_Toc65750705)

[8. Testszenarien 11](#_Toc65750706)

# 1. Zielbestimmung

## Musskriterien

* Für die Erzeugen der dynamischen Teile der Webseite kommt in der Programmierung ausschließlich PHP und JavaScript zum Einsatz.
* Zur Implementierung der REST-API Abfrage mit Hilfe von Triposo.com kommt ausschließlich Javascript zum Einsatz.
* Der (nicht angemeldete) User kann zunächst nach einer bestimmten Destination (Location), welche als Country, Region, City oder National Park eingeteilt wird, suchen. Mit Hilfe der einmaligen Location\_id kann im Anschluss auf die jeweiligen PointsOfInterest (nachfolgend als POI‘s bezeichnet) zugegriffen werden.
* Weiterführende Links zu Location und POI‘s werden im Webbrowser dargestellt (Wikitravel, Wikipedia)
* Ein GoogleMaps Modal wird zu den jeweiligen POI‘s zur Verfügung gestellt
* Mit Hilfe von PHP PDO gibt es die Möglichkeit sich als Traveller(User) zu registrieren und Daten in der Datenbank MariaDB (MySQL) zu speichern.
* Login mit Email und Password möglich, durch Abgleich mit Datenbank
* Das Produkt soll im Quelltext gut kommentiert werden. Die Programmierung soll eine gute Erweiterbarkeit und Anpassbarkeit erlauben. Die Programmierung soll zu diesem Zweck sehr gut dokumentiert werden.
* Der angemeldeten Traveller kann seine Suchergebnisse in der Datenbank Location abspeichern
* Pro gespeicherter Location gibt es die Möglichkeit zugehörige PointsOfInterest in Datenbank zu speichern
* Traveller kann auf der Seite „MySavedTrips“ individuelle gespeicherte Locations und POI‘s anzeigen lassen
* Zur Übersichtlichkeit wird erst eine Liste der gespeicherten Locations angezeigt, welche durch Klick aufgeklappt werden
* Traveller kann Inhalte updaten, Notizen hinzufügen und eigene Locations sowie zugehörige POI’S in Datenbank abspeichern
* Zu den POI‘s ist es möglich per Klick ein eingebettetes GoogleMaps Modal zu öffnen, und wieder geschlossen werden kann, um sich einen schnellen Überblick zu verschaffen
* Passwörter werden im Backend automatisch mit dem neuesten Standard verschlüsselt

## 1.2 Wunschkriterien

* Funktion um gesamte Liste in „MySavedLocations“ aufzuklappen
* Die Prozesse sollen threadsicher sein, damit Inkonsistenzen der Datenbank und im Dateisystem verhindert werden.
* Entwicklung eines mobilen Layouts zur Darstellung auf Smartphones

## 1.3 Abgrenzungskriterien

* Es wird kein Buchungssystem, Ticketkauf oder ähnliches in Verbindung mit POI’s angeboten
* Layout und Website ist auf Abfragen von Reisedaten und der Übersicht der gespeicherten Reisedaten beschränkt
* Es wird explizit kein Shopsystem für reiserelevante Daten zur Verfügung gestellt

# 2. Produkteinsatz

Das Produkt dient zur Information, Recherche durch weiterführende Links und Archivierung von Reisezielen. Mit Hilfe der gekoppelten Abfrage von Locations und dessen POI‘s soll ein schneller Überblick erzeugt werden.

Es soll im Internet öffentlich zugänglich sein und jedem User Zugriff erlauben.

Der User soll motiviert werden sich zu registrieren um eigene Datenabfragen zur individuellen Reiseplanung zu generieren.

Der User soll möglichst schnell eine auf Ihn zugeschnittene Reise erstellen können. Das Abfragen, Speichern und Aufrufen der Daten soll daher simpel gehalten werden um einen strukturierten Überblick zu erhalten.

## 2.1 Anwendungsbereiche

Das Produkt wird ausschließlich zur Reiseplanung und Informationsbeschaffung verwendet.

## 2.2 Zielgruppen

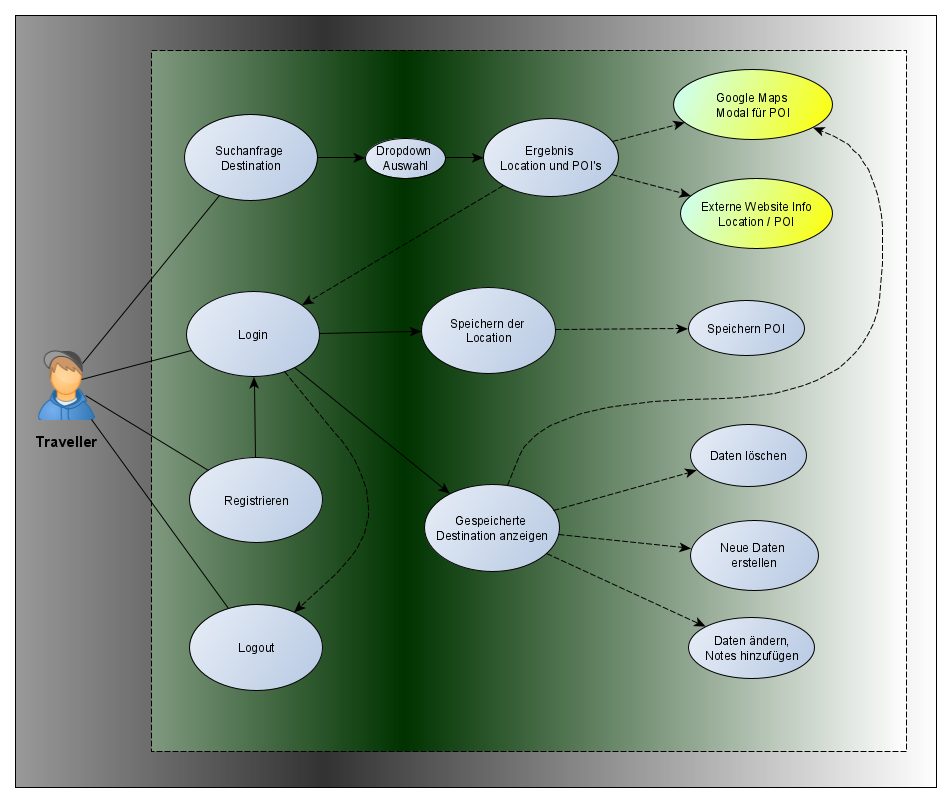
Reisefreudige Internet-User auf der Suche nach Individueller Reiseplanung und maximaler Flexibilität u.a.

* Backpacker
* Individual-Touristen
* Familien

Der Anspruch ist es, sowohl für (Kurzzeit)-Reisende einen Überblick zur Einteilung der Reisezeit und der jeweiligen Strukturierung der Destinationen und Sehenswürdigkeiten zu schaffen, als auch eine ToDo Liste für den User für zukünftige Reisen zu erstellen und durch eigene Notizen zu optimieren.

# 3. Produktübersicht

Use-Case-Diagramm:



# 4. Produktfunktionen

## 4.1 Geschäftsprozesse

Akteur ist, solange nicht eingeloggt bzw. nicht registriert, der User. Im eingeloggten Zustand wird er als Traveller bezeichnet.

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F10/ (/LF10/) Starten der Suchanfrage Location |
| **Kategorie** | Primär |
| **Vorbedingung** | Website wurde erfolgreich geladen |
| **Nachbedingung Erfolg** | Suchanfrage wird angezeigt und in Html dargestellt |
| **Nachbedingung Fehlschlag** | Ausgabe einer Aufforderung, ‚Location nicht vorhanden‘ |
| **Auslösendes Ereignis** | Traveller tippt Wert in Suchmaske ein |
| **Beschreibung** | Beim eintippen öffnet sich ein Dropdown mit vorausgewählten Ergebnissen, die mit Klick in Html übernommen werden.  Weiterführende Information zur Location wird angezeigt |
| **Erweiterung** | Siehe /F20/ |
| **Alternativen** | - |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F20/ (/LF20/) Erweiterte Suchanfrage POI‘s |
| **Kategorie** | Primär |
| **Vorbedingung** | Anzeige ausgewählter Location in Webseite |
| **Nachbedingung Erfolg** | Bis zu 20 Treffer der POI‘s werden angezeigt |
| **Nachbedingung Fehlschlag** | Keine POI‘s für gewählte Location vorhanden |
| **Auslösendes Ereignis** | Traveller wählt eingetippte Location aus Dropdown Menü |
| **Beschreibung** | Anzeige der 20 am besten bewerteten POI‘s  mit weiterführender Information,  zugehörig zur zuvor ausgewählten Location. |
| **Erweiterung** | Siehe /F30/ |
| **Alternativen** |  |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F30/ (/LF30/) Ein User kann ein eigenes Konto für das Spiel erstellen. |
| **Kategorie** | Primär |
| **Vorbedingung** | User klickt auf den Button „Register“ |
| **Nachbedingung Erfolg** | Email und Passwort entsprichen den Anforderungen. Siehe auch /F50/ |
| **Nachbedingung Fehlschlag** | Kurze Fehlermeldung an den User. |
| **Auslösendes Ereignis** | User erstellt ein Konto |
| **Beschreibung** | Der User gibt FirstName, LastName, Email und Passwort in ein Formular ein und registriert sich mit „Save“ |
| **Erweiterung** | - |
| **Alternativen** | Ein nicht eingeloggter Traveller kann lediglich Suchanfragen /F20/ /F30/ durchführen aber nicht speichern |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F40/ (/LF40/) Ein User kann sich mit seinem Konto auf der Seite einloggen. |
| **Kategorie** | Primär |
| **Vorbedingung** | Webseite wurde erfolgreich geladen, ein Konto wurde angelegt (siehe auch /F30/) |
| **Nachbedingung Erfolg** | Traveller kann Abfragen in Datenbank speichern und einsehen |
| **Nachbedingung Fehlschlag** | Kurze Fehlermeldung an den Traveller |
| **Auslösendes Ereignis** | Der Traveller loggt sich durch Eingabe von Email und Password in sein Konto ein. |
| **Beschreibung** | Nach der Eingabe der Login-Daten in das Formular werden diese mit der DB abgeglichen, im Erfolgsfall wird die ID des Users ermittelt und zurückgegeben. |
| **Erweiterung** | - |
| **Alternativen** | Ein nicht eingeloggter Traveller kann lediglich Suchanfragen /F20/ /F30/ durchführen aber nicht speichern |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F50/ (/LF50/) Ein Traveller kann eigene Daten speichern |
| **Kategorie** | Primär |
| **Vorbedingung** | Webseite wurde erfolgreich geladen. Der Traveller hat ein Konto und ist eingeloggt. |
| **Nachbedingung Erfolg** | Eigene Inhalte werden in Traveller Tabellensystem angezeigt |
| **Nachbedingung Fehlschlag** | Kurze Fehlermeldung an den User |
| **Auslösendes Ereignis** | Der angemeldete Traveller übermittelt Daten an die Datenbank |
| **Beschreibung** | Traveller öffnet die Create Maske durch klick auf den entsprechenden Button, erstellt eigene Location / POI und speichert diese in der Datenbank |
| **Erweiterung** | - |
| **Alternativen** | Ergänzen vorhandener Daten /F60/ |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F60/ (/LF60/) Ein Traveller kann vorhandene Tabellen updaten bzw. ergänzen |
| **Kategorie** | Primär |
| **Vorbedingung** | Webseite wurde erfolgreich geladen. Der User hat ein Konto und ist eingeloggt. |
| **Nachbedingung Erfolg** | Inhalte werden in Traveller Tabellensystem angezeigt |
| **Nachbedingung Fehlschlag** | Kurze Fehlermeldung an den User |
| **Auslösendes Ereignis** | Der angemeldete Traveller übermittelt Daten an die Datenbank |
| **Beschreibung** | Traveller öffnet die Update Maske durch klick des Update-Buttons.  Traveller greift auf vorhandene gespeicherte Location und/oder POI zu und kann diese verändern |
| **Erweiterung** | - |
| **Alternativen** | Erstellen neuer Daten /F50/ |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F70/ (/LF70/) Der User kann sich ausloggen |
| **Kategorie** | Primär |
| **Vorbedingung** | Der Traveller ist eingeloggt. |
| **Nachbedingung Erfolg** | Der Traveller wird auf die Startseite weitergeleitet, die Login Maske erscheint wieder |
| **Nachbedingung Fehlschlag** | - |
| **Auslösendes Ereignis** | Der Traveller klickt auf den entsprechenden Button der Navigationsleiste. |
| **Beschreibung** | Ein entsprechender Button ermöglicht dem Traveller, sich unmittelbar auszuloggen. Eine Rückfrage ist nicht erforderlich. |
| **Erweiterung** | - |
| **Alternativen** | - |

Es folgen Funktionalitäten bei bestimmten Aktivitäten des Travellers / Users

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F80/ (/LF80/) Der User kann zu bereits gespeicherten oder in Suchanfrage dargestellten POI’S ein GoogleMaps Modal aufrufen |
| **Kategorie** | Sekundär |
| **Vorbedingung** | POI vorhanden |
| **Nachbedingung Erfolg** | Modal wird geladen |
| **Nachbedingung Fehlschlag** | - |
| **Auslösendes Ereignis** | Der User klickt auf den „show on Map“ Link |
| **Beschreibung** | Der User klickt auf den Link, danach öffnet sich die eingebettete GoogleMaps der geklickten POI und überlagert die aktuelle Website. Nach erneutem Klick ausserhalb der Map-Feldes, schließt diese sich wieder |
| **Erweiterung** | Siehe /F20/ F60/ |
| **Alternativen** | Eine eigene Map speichern siehe /F60/ |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F90/ (/LF90/) Status-Message beim Speichern von Location / POI |
| **Kategorie** | Sekundär |
| **Vorbedingung** | Suchanfrage erfolgreich |
| **Nachbedingung Erfolg** | Message wird als Html Text dargestellt |
| **Nachbedingung Fehlschlag** | - |
| **Auslösendes Ereignis** | Der Traveller klickt auf den „Save“ Button |
| **Beschreibung** | Durch klicken auf den Button wird eine Message als Html dargestellt („Erfolgreich“ oder „bereits Vorhanden) und verschwindet nach 3 Sekunden wieder. Anschließend wird der Button inaktiv und kann nicht wieder gedrückt werden |
| **Erweiterung** | Siehe /F20/ |
| **Alternativen** | - |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F100/ (/LF100/) Popup Bestätigung beim Löschen von Daten |
| **Kategorie** | Sekundär |
| **Vorbedingung** | Traveller eingeloggt und Daten vorhanden siehe /F40/ /F50/ |
| **Nachbedingung Erfolg** | Daten aus Datenbank entfernt |
| **Nachbedingung Fehlschlag** | - |
| **Auslösendes Ereignis** | Traveller klickt auf den Delete-Button |
| **Beschreibung** | Beim Klick auf den Delete Button öffnet sich ein Confirm-Popup, mit einem Hinweis dass aktuelle Daten gelöscht werden. Im Falle des Löschens einer Location gibt es den zusätzlichen Hinweis dass alle zugehörigen POI’s ebenfalls gelöscht werden. |
| **Erweiterung** | - |
| **Alternativen** | Klick auf die Option „Cancel“ |

|  |  |
| --- | --- |
| **(ID) Geschäftsprozess** | /F110/ (/LF110/) Html-Message bei verändern der Datensätze |
| **Kategorie** | Sekundär |
| **Vorbedingung** | -User klickt auf den Create-Button um Account zu erstellen.  -Traveller klickt auf den Update- oder Create-Button in der Datenliste bzw. seines Accounts |
| **Nachbedingung Erfolg** | Html-Message erscheint überhalb der Eingabemaske |
| **Nachbedingung Fehlschlag** | - |
| **Auslösendes Ereignis** | Der User/Traveller klickt auf den Save-Button in der Eingabemaske |
| **Beschreibung** | In der Eingabemaske von User/Travellers, der Location oder POI wird nach eventueller Änderung / Erstellung der Daten der Save-Button gedrückt. Nach Prüfung der Felder erscheint eine entsprechende Meldung als Html-Text |
| **Erweiterung** | Beim erfolgreichen Speichern erfolgt nach 2 Sekunden ein Redirect auf die vorherige Seite. |
| **Alternativen** | Abbrechen mit dem Back-Button |

# 5. Produktdaten

## 5.1 Traveller Daten

|  |
| --- |
| /D10/ (/LD10/) Traveller |
| firstName  lastName  email  password |

## 5.2 Location Daten

|  |
| --- |
| /D20/ (/LD20/) Location (max. 50.000) |
| idTraveller  location  locationKey  classification  country  region  intro  travelLink  notes |

## 5.3 PointOfInterest

|  |
| --- |
| /D30/ (/LD30/) PointsOfInterest (max. 50.000) |
| idLocation  poiName  locationKey  attraction  intro  infoLink  poiMap  notes |

# 6. Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Produktqualität:** | **sehr gut** | **gut** | **normal** | **Nicht relevant** |
| Funktionalität |  | **X** |  |  |
| Zuverlässigkeit |  |  | **X** |  |
| Design / Layout |  |  | **X** |  |
| Änderbarkeit |  | **X** |  |  |
| Dokumentation | **X** |  |  |  |

# 7. Technische Produktumgebung

Das Produkt ist eine PHP-basierte Webseite. Die Kommunikation per API zum Datenanbieter [triposo.com](https://www.triposo.com/api/), funktionierender Webserver und PHP 8.0 werden zur Ausführung vorausgesetzt.

## 7.1 Software

Ausführbar ist die Software nach Installation auf dem Server über jeden modernen, HTML5-fähigen Browser.

## 7.2 Hardware

Client: Bildschirm mit mindestens 1024x768 Pixeln.

# 8. Testszenarien

Während und nach der Produktentwicklung werden verschiedene Tests durchgeführt, um die Funktionalität der Software zu gewährleisten.

|  |  |  |
| --- | --- | --- |
|  | - | Einen neuen Traveller erstellen, Prozess abbrechen |
|  | - | Fetch API Anfrage durch Dropdown Auswahl und Click um Ergebnisse anzuzeigen |
|  | - | Speichern von Location und POI mit Status-Meldung angezeigt im Submit Button |
|  | - | Auf- und zuklappen der gespeicherten Location Liste, einzeln und gesamt |
|  | - | Eigene Datensätze (Location, POI) speichern und/oder vorhandene Datensätze updaten bzw. löschen |
|  | - | Embedded Google Maps Modal bei Klick auf Link (Suchanfrage und gespeicherte POI's) |