Studienarbeit ERP-Programmierung

„Business-Objekt FlightBooking“

Wintersemester 2018/2019

bei Prof. Dr. Martin Thost

von Thomas Zerr

Inhaltsverzeichnis

[Das Business-Objekt FlightBooking 1](#_Toc530754095)

[„Confirm“ 1](#_Toc530754096)

[„Cancel“ 2](#_Toc530754097)

[„CreateFromData“ 2](#_Toc530754098)

[„GetList“ 2](#_Toc530754099)

[Das Business-Objekt „Flight” und ihre Methode „GetList“ 2](#_Toc530754100)

[Architektur der Software 2](#_Toc530754101)

[„LoginWindow“ 3](#_Toc530754102)

[„FlightBookingMainWindowFactory“ 4](#_Toc530754103)

[„FlightBookingMainWindow“ 4](#_Toc530754104)

[„FlightBookingCreateWindowFactory“ 5](#_Toc530754105)

[„FlightBookingCreateViewModel“ 6](#_Toc530754106)

[„FlightBookingEditWindowFactory“ 7](#_Toc530754107)

[„FlightBookingEditViewModel“ 8](#_Toc530754108)

[„FlightBookingFactory“ 9](#_Toc530754109)

[„FlightBooking“ 9](#_Toc530754110)

[„FlightFactory“ 10](#_Toc530754111)

[Problem mit BapiServiceTransactionCommit 10](#_Toc530754112)

[Handbuch 10](#_Toc530754113)

[Login 10](#_Toc530754114)

[Logout 11](#_Toc530754115)

[Flüge suchen 11](#_Toc530754116)

[Flugbuchung erstellen 12](#_Toc530754117)

[Flugbuchungen suchen 13](#_Toc530754118)

[Flugbuchung bestätigen 14](#_Toc530754119)

[Flugbuchung stornieren 16](#_Toc530754120)

# Das Business-Objekt FlightBooking

Das Business-Objekt FlightBooking wurde für diese Studienarbeit ausgewählt. Es stellt die Buchung für einen Flug dar. Für weitere Informationen siehe in die BAPI Dokumentation dieses Objektes.

## „Confirm“

Durch diese Methode kann eine reservierte Flugbuchung bestätigt werden. Ist die Buchung bereits bestätigt oder storniert ist es nicht möglich sie zu bestätigen.

Für die Implementierung des Aufrufes dieser Methode siehe [„FlightBooking“](FlightBooking#_). Für eine Anleitung, um eine Flugbuchung zu bestätigen, siehe [„Flugbuchung bestätigen“](#_Flugbuchung_bestätigen).

## „Cancel“

Durch diese Methode kann ein reservierte oder bestätigte Flugbuchung storniert werden.

Für die Implementierung des Aufrufes dieser Methode siehe [„FlightBooking“](FlightBooking#_). Für eine Anleitung, um eine Flugbuchung zu stornieren, siehe [„Flugbuchung stornieren“](#_Flugbuchung_stornieren).

## „CreateFromData“

Durch diese Methode kann eine neue Flugbuchung zu einem Flug erstellt werden. Diese neue Flugbuchung kann als Reservierung oder als bestätigte Flugbuchung erstellt werden.

Für die Implementierung des Aufrufes dieser Methode siehe [„FlightBookingFactory“](FlightBookingFactory#_). Für eine Anleitung, um eine Flugbuchung zu erstellen, siehe [„Flugbuchung erstellen“](#_Flugbuchung_erstellen).

## „GetList“

Durch diese Methode kann nach bereits angelegten Flugbuchungen gesucht werden.

Für die Implementierung des Aufrufes dieser Methode siehe [„FlightBookingFactory“](FlightBookingFactory#_). Für eine Anleitung, um nach Flugbuchungen zu suchen, siehe [„Flugbuchung suchen“](#_Flugbuchung_suchen).

# Das Business-Objekt „Flight” und ihre Methode „GetList“

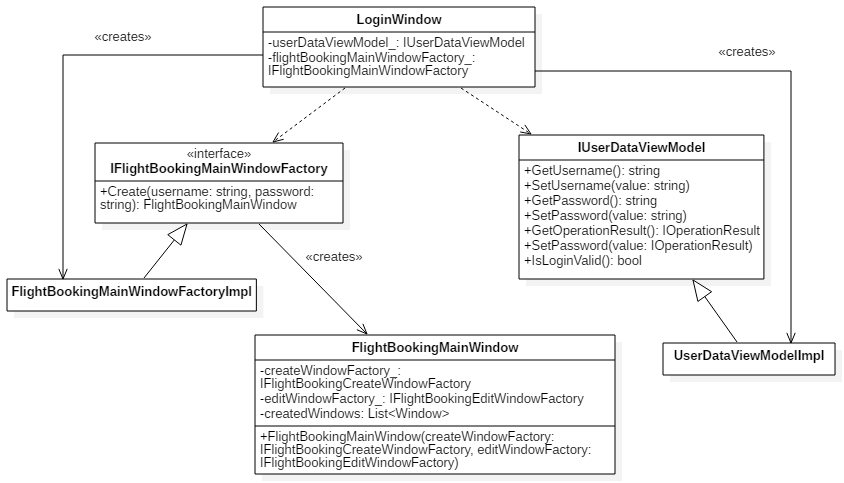
Jede Flugbuchung gehört zu genau einem Flug. Daher muss beim Erstellen einer Flugbuchung den entsprechenden Flug mit angeben. Um die dazu benötigten Informationen des Fluges zu beschaffen, wird die „GetList“-Methode des Business-Objekts „Flight“ verwendet. Da bei dieser Studienarbeit der Fokus auf das Business-Objekt „FlightBooking“ liegt, wurden nicht alle Suchparameter von „GetList“ (des Objektes „Flight“) verwendet.

Für die Implementierung des Aufrufes dieser Methode siehe [„FlightFactory“](FlightFactory#_). Für eine Anleitung, um nach Flügen zu suchen, siehe [„Flüge suchen“](#_Flüge_suchen).

# Architektur der Software

Bei dieser Software handelt es sich um eine WPF-Anwendung in C#. Die Software ist nach dem MVVM-Architekturmuster designend. Die wichtigsten Klassen und Vorgänge werden in den anschließenden Unterkapiteln erläutert.

## „LoginWindow“



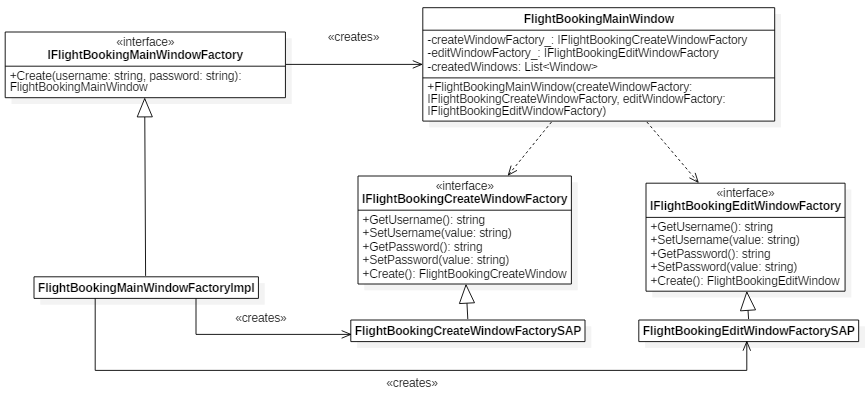
Beim Starten der Anwendung wird das „LoginWindow“ initialisiert. Aufgabe dieses Fensters ist es, die Login-Daten des Benutzers entgegen zu nehmen und den Login-Vorgang zu starten.

Beim Login-Vorgang muss überprüft werden, ob die angegebenen Login-Daten gültig sind. Dafür werden die Login-Daten, in den Eigenschaften „Username“ und „Password“ eines „IUserDataViewModels“ gespeichert und mit der Methode „IsLoginValid“ überprüft. Bei der Implementierung von „UserDataViewModelImpl“ wird lediglich überprüft, ob der „Username“ und das „Password“ nicht leer sind.

Sind die Login-Daten gültig, wird als nächstes ein „FlightBookingMainWindow“ erzeugt. Dazu wird die „Create“-Schnittstelle einer „IFlightBookingMainWindowFactory“ verwendet. Sobald dieses Fenster erzeugt wurde, wird es angezeigt und das „LoginWindow“ schließt sich.

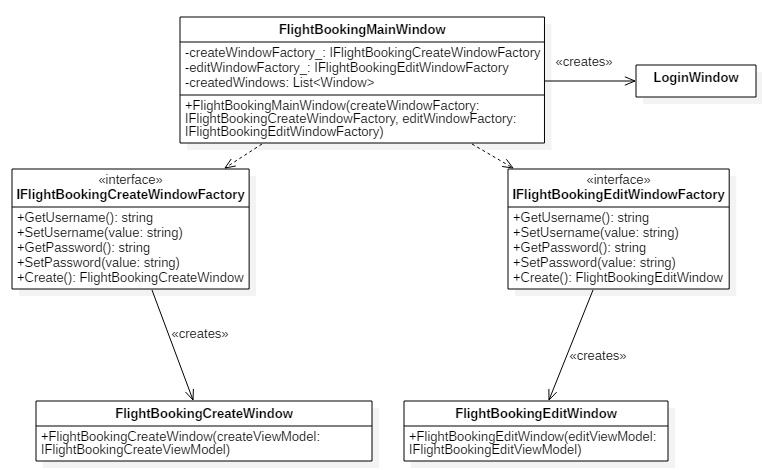
Die konkreten „IFlightBookingMainWindowFactory“ und „IUserDataViewModels“ werden vom „LoginWindow“ selbst erzeugt.

## „FlightBookingMainWindowFactory“



Um ein „FlightBookingMainWindow“ zu erzeugen, muss man es mit den Abhängigkeiten „IFlightBookingCreateWindowFactory“ und „IFlightBookingEditWindowFactory“ versorgen. Dementsprechend muss die „FlightBookingMainWindowFactoryImpl“ diese Abhängigkeiten erzeugen. Die konkreten Klassen dieser Abhängigkeiten werden aktuell über eine Einstellung in der „app.config“-Datei festgelegt. Aktuell ist „SAP“ der einzig gültige Wert für diese Option, dementsprechend können nur die konkreten Klassen „FlightBookingCreateWindowFactorySAP“ und „FlightBookingEditWindowFactorySAP“ instanziiert werden.

## „FlightBookingMainWindow“



Im „FlightBookingMainWindow“ kann der Benutzer drei Vorgänge starten:

1. Logout
2. Ein „FlightBookingCreateWindow“ erstellen und öffnen
3. Ein „FlightBookingEditWindow“ erstellen und öffnen

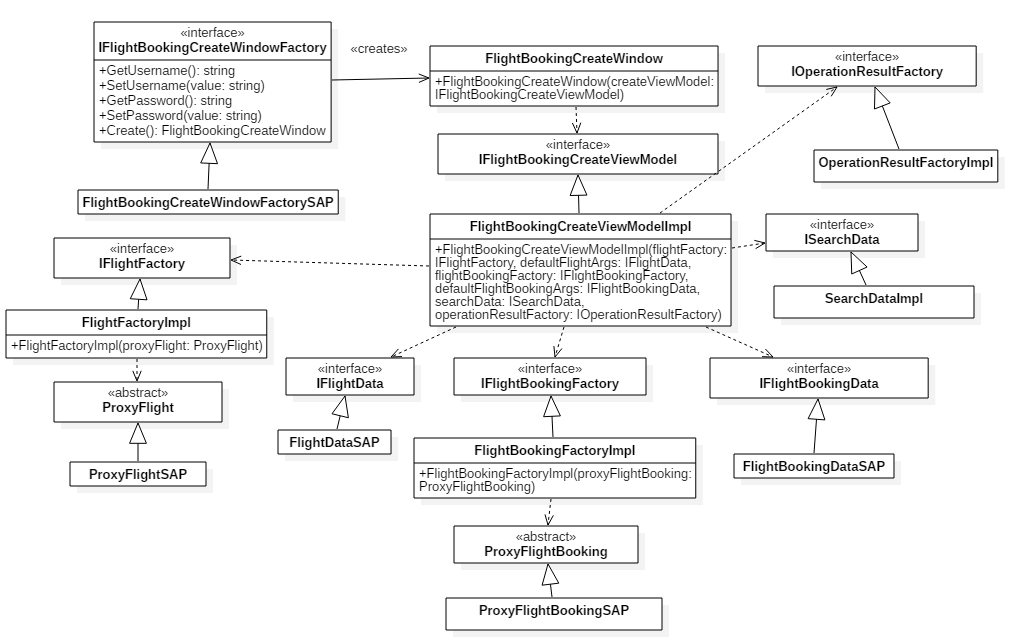
Beim Logout erstellt das „FlightBookingMainWindow“ einen neuen „LoginWindow“ und schließt sich selbst.

Beim zweiten und dritten Vorgang wird über die „Create“-Schnittstelle der jeweiligen Fabrik, dass entsprechende Window-Objekt initialisiert und vom „FlightBookingMainWindow“ geöffnet.

Alle dadurch geöffneten Fenster, werden vom „FlightBookingMainWindow“ in einer Liste verwaltet. Wird „FlightBookingMainWindow“ geschlossen, so werden alle Window-Objekte in dieser Liste ebenfalls geschlossen.

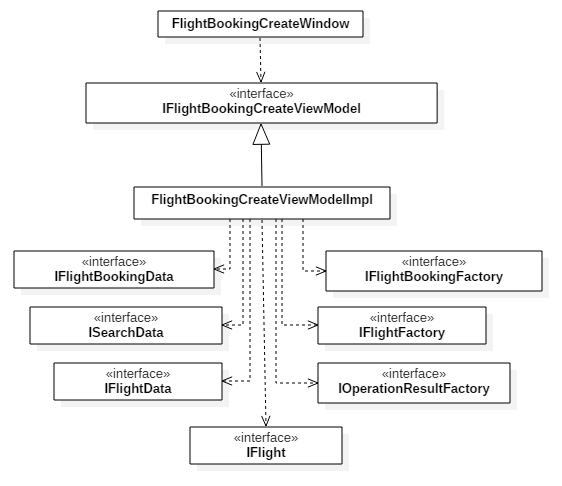
Die Benötigten Objekte der Fabriken, werden dem „FlightBookingMainWindow“ über die Konstruktor Parameter injiziert.

## „FlightBookingCreateWindowFactory“



Um ein „FlightBookingCreateWindow“ zu erzeugen muss man ein „IFlightBookingCreateViewModel“ über den Konstruktor injizieren. Eine Instanz dieses „ViewModels“ wird ebenfalls von dieser Fabrik erzeugt. Die vom „ViewModel“ benötigten Abhängigkeiten werden ebenfalls von der Fabrik erzeugt. Von jeder konkreten Klasse (im Diagramm sichtbar) wird von der „FlightBookingCreateWindowFactorySAP“ eine Instanz erzeugt. Die konkreten Klassen wurde so ausgewählt, damit diese Software mit einem SAP-System über BAPI-Schnittstellen kommuniziert.

## „FlightBookingCreateViewModel“

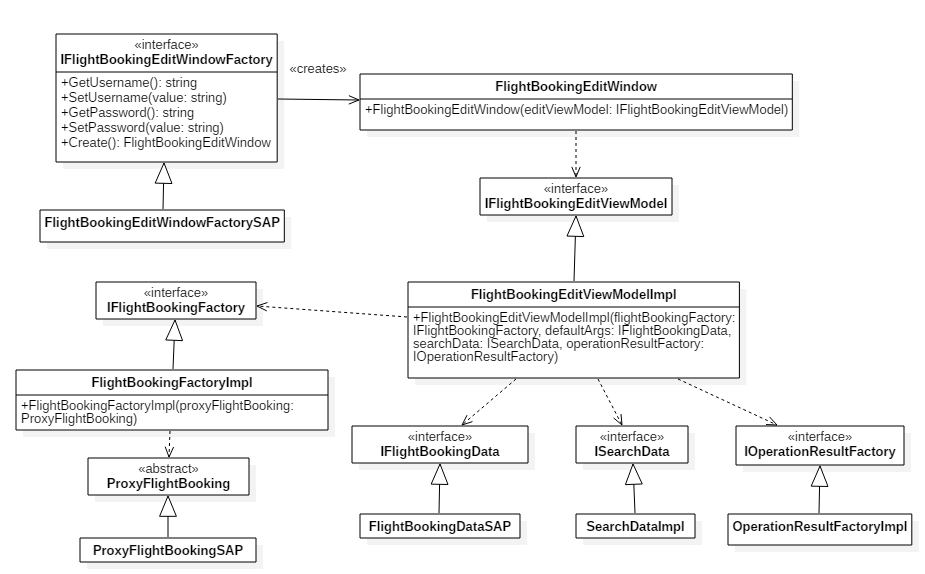


Sämtliche eingaben des Benutzers im „FlightBookingCreateWindow“ sind an Eigenschaften oder Methoden von „IFlightBookingCreateViewModel“ gebunden. Die implementierung „FlightBookingCreateViewModelImpl“ delegiert das Setzen und Abfragen der Eigenschaften an ihre Entsprechende Abhängigkeit. Diese übernehmen dann die Validierung der zu setzenden Werte und werfen eine Exception in Fehlerfälle.

Die „DoFlightSearch“- und „DoCreateFlightBooking“-Methoden delegieren an die Schnittstellen „Retrieve“ von „IFlightFactory“ und an „Create“ von „IFlightBookingFactory“ weiter.

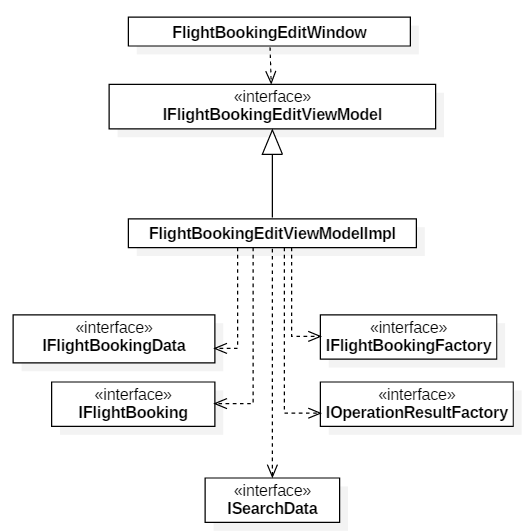
Das „ViewModel“ fängt Exceptions von ihren Abhängigkeiten ab und erstellt dazu eine neue „OperationResult“ und setzt die „OperationResult“-Eigenschaft damit.

## „FlightBookingEditWindowFactory“



Die Factory für das „FlightBookingEditWindow“ verhält sich analog zu dem der „FlightBookingCreateWindowFactory“.

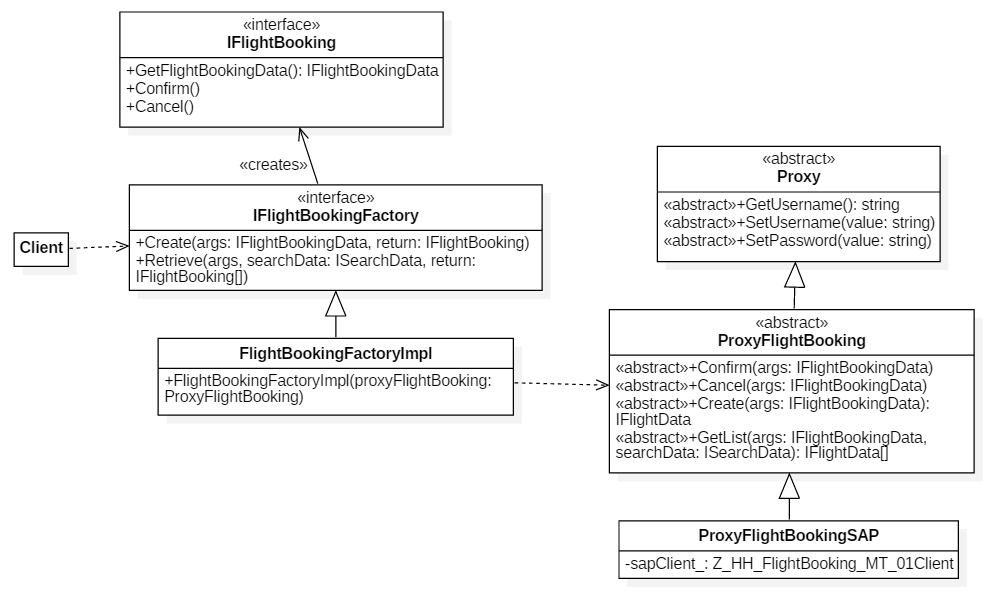
## „FlightBookingEditViewModel“



Das „FlightBookingEditViewModelImpl“ verhält sich analog zu dem „FlightBookingCreateViewModelImpl“.

Die „DoFlightBookingSearch“-, „DoConfirmFlightBooking“- und „DoCancelFlightBooking“-Methoden delegieren jeweils an die Schnittstellen „Retrieve“ von „IFlightBookingFactory“, an „Confirm“ und „Cancel“ von „IFlightBooking“ weiter.

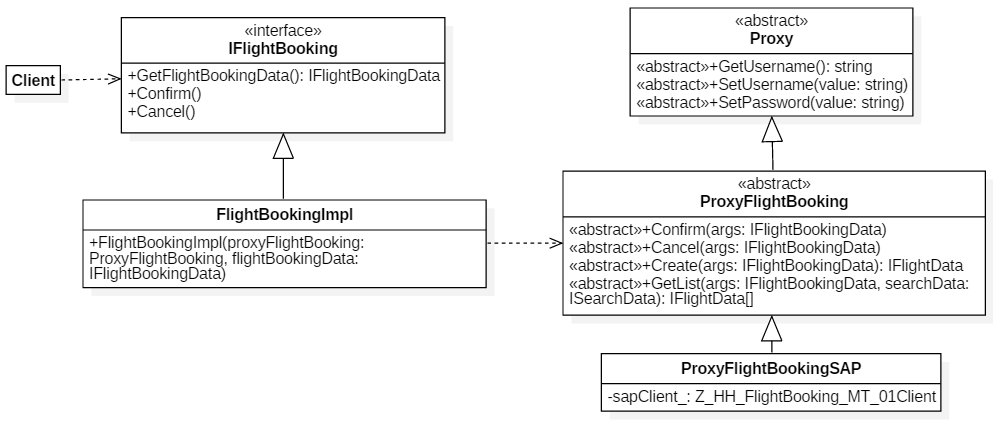
## „FlightBookingFactory“



Die „Create“ und „Retrieve“ Schnittstelle von der „IFlightBookingFactory“ sind dafür da, um neue Flugbuchungen zu erstellen und bereits erstellte Flugbuchungen zu suchen und aus den gefundenen Daten die Objekte zu erstellen.

Die Implementierung dieser Fabrik benötigt ein „ProxyFlightBooking“-Objekt. Dieses Objekt wird dafür benötigt, um die Daten einer neuen Flugbuchung zu persistieren bzw. bereits persistierte Daten zu suchen. Dabei kann der Proxy entweder mit einem SAP System über eine BAPI-Schnittstelle kommunizieren oder mit einem DBMS. Das ProxyFlightBookingSAP erledigt dies über die erst genannte Möglichkeit.

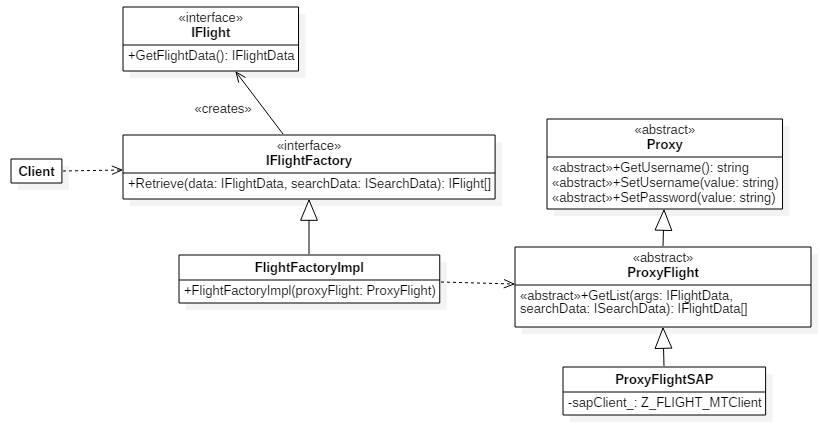
## „FlightBooking“



Mit den „Confirm“ und „Cancel“ Schnittstellen von „IFlightBooking“ kann eine Flugbuchung bestätigt oder storniert werden.

Die Implementierung von „FlightBookingImpl“ ruft dafür jeweils die „Confirm“ und „Cancel“ Schnittstellen von „ProxyFlightBooking“ auf. Zusätzlich aktualisiert es die Werte in der „IFlightBookingData“ Property.

## „FlightFactory“



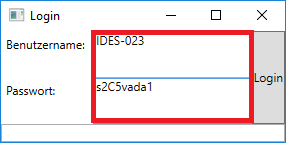
Die Schnittstellen von „IFlightFactory“ funktionieren analog zu den von „IFlightBookingFactory“.

# Problem mit BapiServiceTransactionCommit

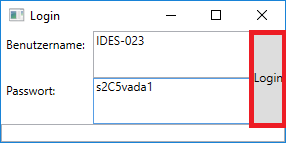
Führt man über eine BAPI-Schnittstelle eine schreibende Methode aus, so werden die Änderungen nicht direkt übernommen und auf die Datenbank geschrieben. Man müsste nach Ausführung der schreibenden Methode die „BapiServiceTransactionCommit“-Methode ausführen. Allerdings wird die „BapiServiceTransactionCommit“-Methode scheinbar nicht ausgeführt, da auch nach Aufruf dieser Methode die Änderungen nicht übernommen sind. Laut BAPI-Dokumentation muss die Antwort der „BapiServiceTransactionCommit“-Methode enthalten, ob sich erfolgreich, fehlerhaft oder ähnliches verlaufen ist, stattdessen ist die Antwort leer. Daher werden die Änderungen durch einen Benutzer vom SAP-System nicht übernommen.

# Handbuch

## Login

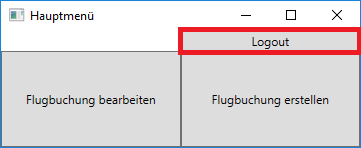


Um sich einloggen zu können, müssen in den rot markierten Textfeldern ein Benutzername und ein Passwort eingegeben sein.



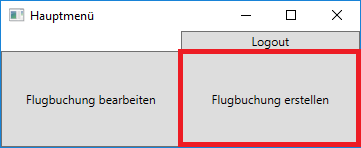
Anschließend muss der rotmarkierte Button betätigt werden.

## Logout

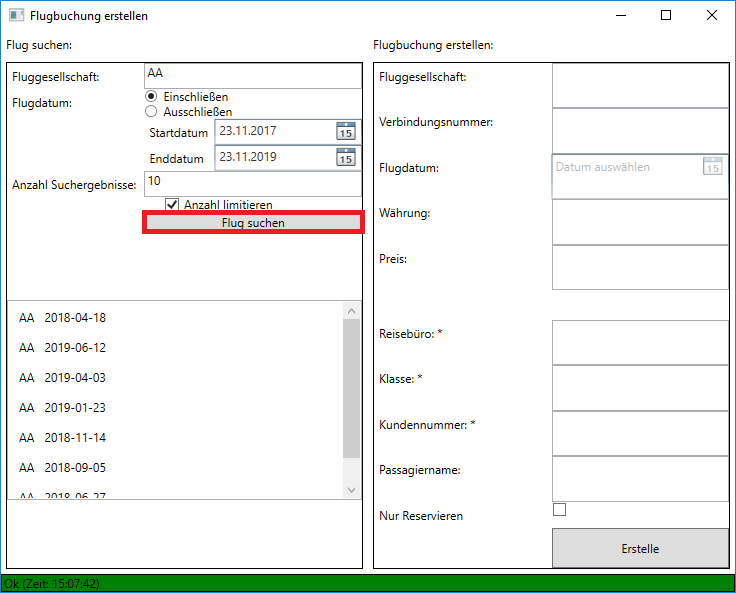


Um sich wieder aus zu loggen, muss der Benutzer im Hauptmenü den rot markierten Button betätigen.

## Flüge suchen



Um nach einem Flug zu suchen muss zunächst der rot markierte Button im Hauptmenü betätigt werden.

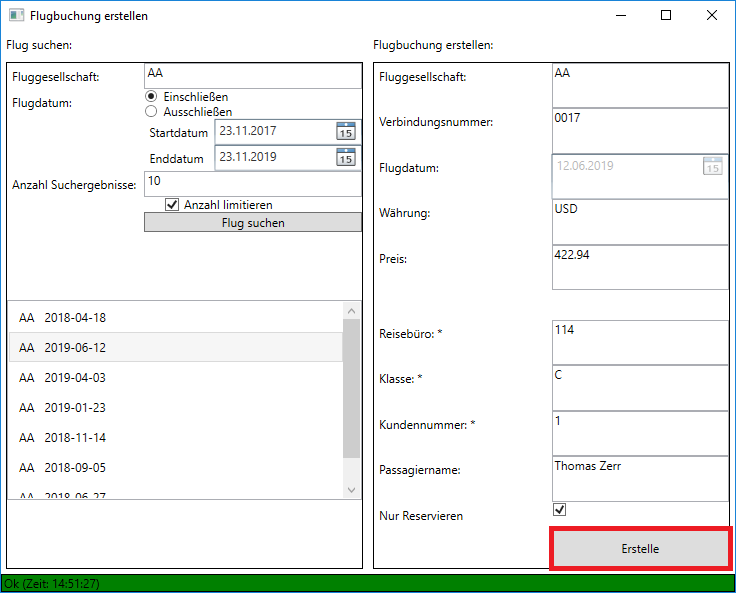


Danach öffnet sich dieses Fenster. In der linken Spalte kann man dann die Suchparameter eingeben und mit dem rot markierten Button die Suche starten.

Für nähere Informationen zu den Suchparametern stehen entsprechende Tooltips bereit, falls man über das entsprechende Label oder Eingabefeld hovered.

## Flugbuchung erstellen

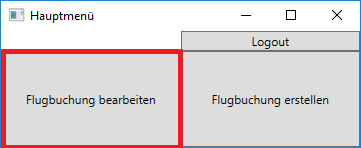
Befolgen Sie zunächst die Schritte, um einen Flug zu suchen und suchen Sie sich eines der Suchergebnisse aus.



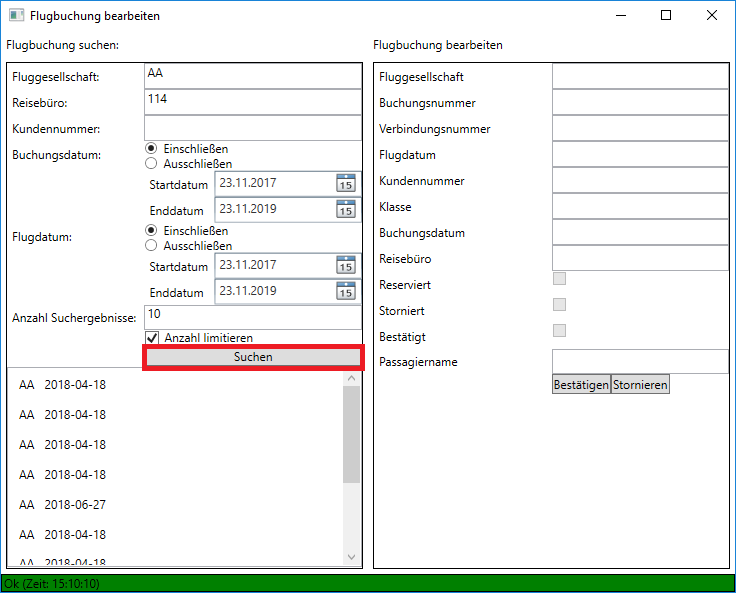
Sobald sie einen Flug ausgewählt haben, erscheinen auf der rechten Seite, die Daten zu diesem Flug. Unter den Daten des Fluges können Sie die Daten der Flugbuchung eingeben. Durch das betätigen des rot markierten Buttons, können Sie die Flugbuchung erstellen.

Für nähere Informationen zu den Daten der Flugbuchung stehen entsprechende Tooltips bereit, falls man über das entsprechende Label oder Eingabefeld hovered.

## Flugbuchungen suchen



Um nach einer Flugbuchung zu suchen muss zunächst der rot markierte Button im Hauptmenü betätigt werden.

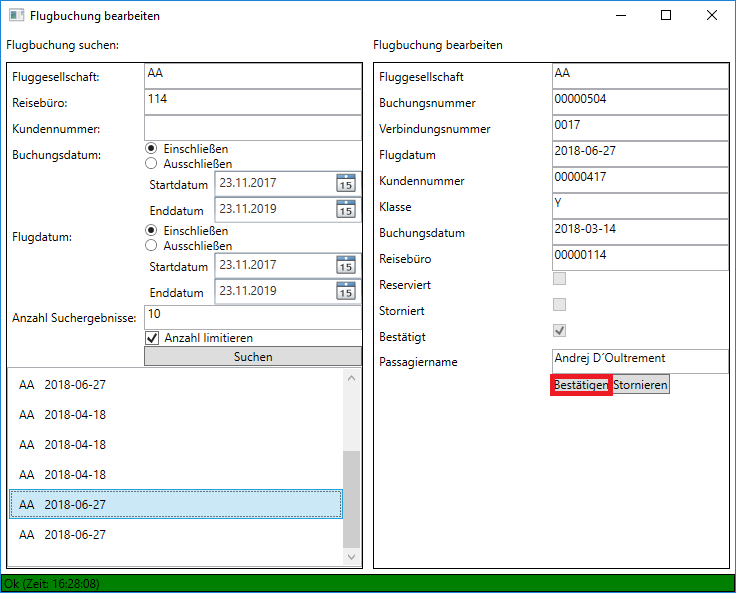


Danach öffnet sich dieses Fenster. In der linken Spalte kann man dann die Suchparameter eingeben und mit dem rot markierten Button die Suche starten.

Für nähere Informationen zu den Suchparametern stehen entsprechende Tooltips bereit, falls man über das entsprechende Label oder Eingabefeld hovered.

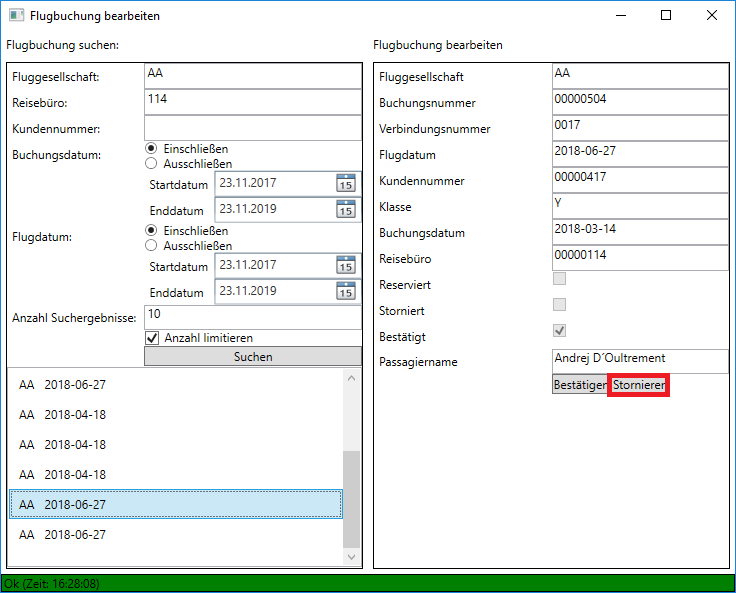
## Flugbuchung bestätigen

Befolgen Sie zunächst die Schritte, um einen Flugbuchung zu suchen und suchen Sie sich eines der Suchergebnisse aus.



Nachdem Sie sich das eine Flugbuchung ausgesucht haben, werden die Daten zu dieser in der rechten Spalte erscheinen. Wenn Sie nun den rot markierten Button betätigen, wird der Flug bestätigt, falls er noch nur reserviert war.

## Flugbuchung stornieren



Das Stornieren funktioniert analog zu dem Bestätigen. Es muss lediglich ein anderer Button gedrückt werden. Dieser ist hier rot markiert. Es kann nur eine bestätigte oder reservierte Flugbuchung storniert werden.