

Abgabe 2: Konzeptuelle Modellierung

Third Party Marketplace

„Spirit Animal“

Jonas Speiser

a12012545@unet.univie.ac.at

Inhalt

A: Use Case Beschreibung	3
B2C Use Cases.....	3
Registrieren und Einloggen.....	3
Tier inserieren (B2C)	4
Lieblingstiere auswählen	5
Tier kaufen (B2C)	6
Widerrufen des Kaufs (B2C)	8
B2B Use Cases.....	10
Tier inserieren (B2B)	10
Kauf von Premiumfunktionen (B2B)	11
Tier kaufen (B2B)	12
Widerrufen des Kaufs (B2B)	13
Nicht umgesetzte Use Cases.....	15
„Date“ vereinbaren (B2C)	15
Futter u. Spielzeugpaket kaufen (B2C)	15
Futter- u. Spielzeugpakete inserieren (B2B).....	15
Statistiken abrufen (B2B).....	15
B: Datenmodell	16
C: Definition der EDI Sprache für B2B Use Case Prozesse und B2B Schnittstellenbeschreibung.....	17
C.1 EDI Sprache.....	17
Inserat erstellen.....	17
Inserat ändern	19
Premiumfunktionen kaufen	21
Tier kaufen.....	22
Widerruf	23
C.2 B2B Schnittstelle.....	24
/api/inserate	24
/api/kaeufe	25
C.3 Protokollsequenzen.....	26
Tier inserieren.....	26
Kauf von Premiumfunktionen	27
Tier kaufen.....	28
Widerrufen des Kaufs	28
C.4 GraphQL-Schnittstelle	29

A: Use Case Beschreibung

Aus Gründen des Projektumfangs (vier Use Cases) konzentriere ich mich in Abgabe 2 und 3 auf die wichtigsten Kernanwendungsfälle. Um das Geschäftsmodell wie in Abgabe 1 beschrieben vollständig zu implementieren, wären weitere Use Cases von Nöten (siehe Abschnitt „Nicht umgesetzte Use Cases“).

B2C Use Cases

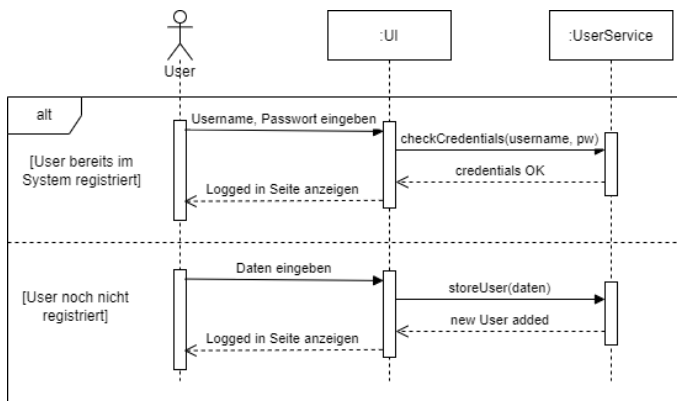
Registrieren und Einloggen

Egal ob Geschäftskunde oder Privatkunde: Die erstmalige Registrierung eines User-Accounts muss über das Frontend stattfinden. Daher ist dieser Use Case nur unter B2C gelistet.

Damit ein User sinnvoll mit dem System interagieren kann muss dieser eingeloggt sein (B2C Use Cases) bzw. sich über einen API-Token authentifizieren können (B2B Use Cases). Damit die jeweiligen Credentials hierfür im System hinterlegt sind, muss der Nutzer sich einmalig im System registrieren. Fortan kann er sich mit Username und Passwort (B2C) bzw. mit dem API-Token (B2B) am System anmelden.

Für einen Login gibt der Nutzer auf der Startseite Username und Passwort ein und bestätigt mit einem Klick auf „Login“. Bei einem Fehler (z.B. falsche Credentials) wird er um erneute Eingabe gebeten. Bei einem erfolgreichen Login landet er auf einer Übersichtseite.

Für eine Registrierung gibt der Nutzer Username, Passwort und persönliche Daten an und klickt auf „Registrieren“. Ist der Username bereits vergeben, erscheint eine Fehlermeldung und der User wird zur Eingabe eines anderen Usernames aufgefordert. Gewerbliche User geben zusätzliche ihren Firmennamen an. Bei Erfolg wird der User automatisch eingeloggt.



Tier inserieren (B2C)

Für diesen Use Case muss der User eingeloggt sein.

Der User erstellt ein Inserat, indem er auf der Übersichtsseite auf „Inserieren“ klickt und die sich nun öffnende Maske mit den Details des Inserats befüllt. Sollten beim Klick auf „Bestätigen“ wichtige Daten fehlen, wird der User zur Eingabe dieser Daten aufgefordert. Nach einem erfolgreichen Klick auf „Bestätigen“ wird dieses Inserat nun online gestellt und dem User wird das fertige Inserat angezeigt.

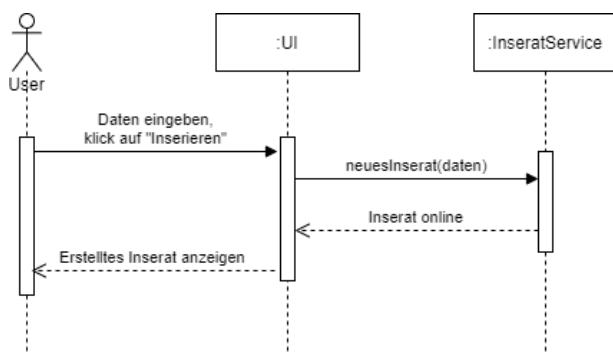
Möchte der User ein bestehendes Inserat ändern oder löschen, wählt er auf der Übersichtsseite den Punkt „Meine Inserate“. Auf dieser Seite werden ihm alle eigens erstellten Inserate angezeigt. Klickt der User nun auf eines dieser Inserate, bekommt er die Inseratsdetails sowie die Schaltflächen „Bearbeiten“ und „Löschen“ angezeigt.

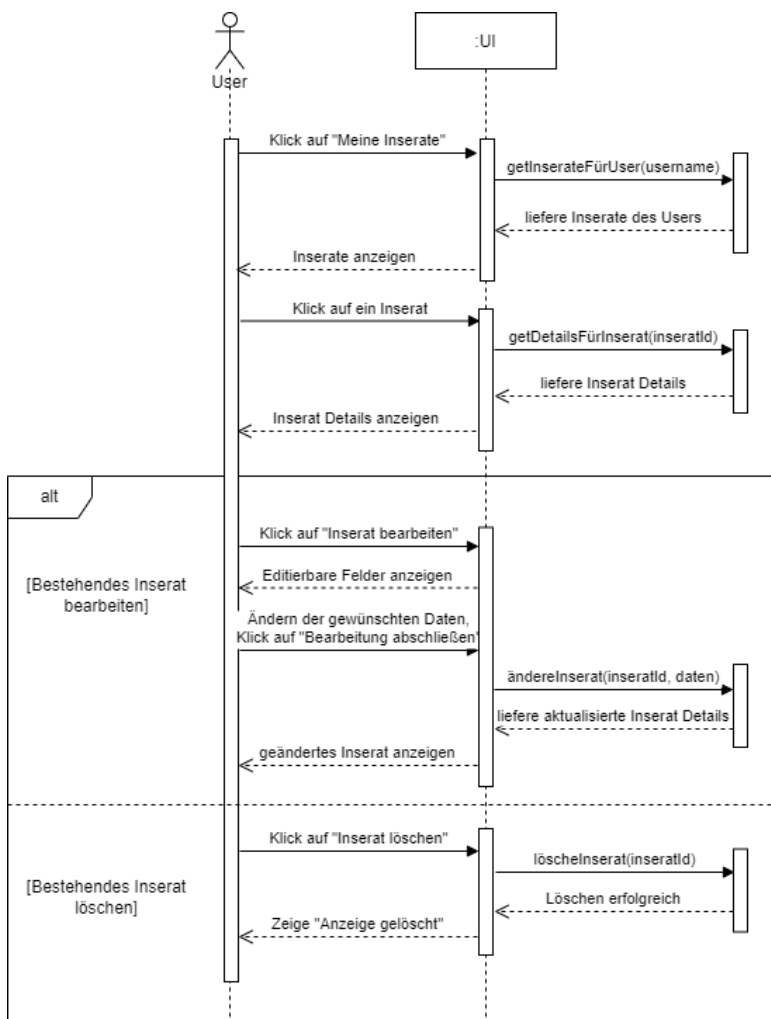
Bei einem Klick auf „Ändern“ wird das Inserat mit editierbaren Felder angezeigt. Diese können nun angepasst werden. Mit einem Klick auf „Bearbeitung abschließen“ werden die Änderungen an das System übermittelt und dem User das geänderte Inserat angezeigt. Im Fehlerfall wird eine entsprechende Fehlermeldung ausgegeben und auf die Seite „Meine Inserate“ weitergeleitet.

Bei einem Klick auf „Löschen“ wird die Anzeige aus dem System gelöscht und dem User die Nachricht „Anzeige erfolgreich gelöscht“ angezeigt. Im Fehlerfall wird eine entsprechende Fehlermeldung ausgegeben. In beiden Fällen wird der User auf die Seite „Meine Inserate“ weitergeleitet.

Vorbedingung: Der User muss eingeloggt sein

Inserat erstellen:



Bestehendes Inserat ändern oder löschen:

Lieblingstiere auswählen

Das Alleinstellungsmerkmal der Plattform ist es, dass dem potenziellen Käufer statt einem klassischen Webshop eine Oberfläche vergleichbar derer berühmter Dating Apps geboten wird. So bekommt der Käufer, abhängig von seinen Präferenzen, die „Profilbilder“ einiger inserierter Tiere angezeigt und kann durch ein „Like“ oder „Dislike“ festlegen, welche Tiere ihm besonders gut gefallen.

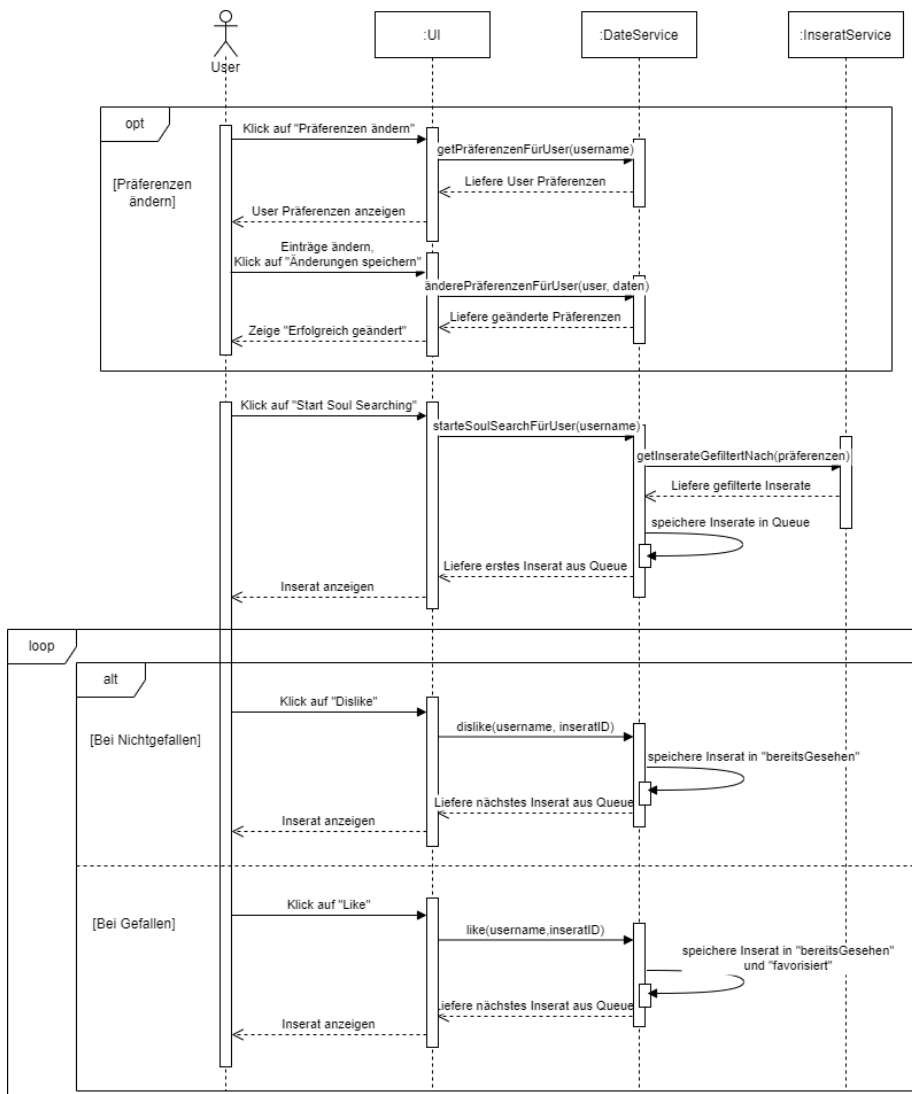
Für diesen Use Case muss der User eingeloggt sein. Anschließend klickt der User auf der Übersichtsseite auf „Soul Searching“.

Hat der User noch keine Präferenzen definiert, oder möchte diese anpassen, klickt er zunächst auf „Präferenzen ändern“. Er bekommt eine Seite mit den aktuell eingestellten Präferenzen angezeigt, welche er verändern kann. Durch einen Klick auf „Änderungen speichern“ werden diese Präferenzen im System abgelegt und der User erhält die Nachricht „Erfolgreich geändert“. Im Fehlerfall wird eine entsprechende Fehlermeldung ausgegeben und die Seite „Soul Searching“ weitergeleitet.

Ist der User mit seinen Präferenzen zufrieden, kann mit einem Klick auf „Find your Spirit Animal!“ der Auswahlprozess gestartet werden. Der User bekommt nun ein Inserat, welches seinen Präferenzen entspricht, angezeigt und kann hier entweder auf „Like“ oder „Dislike“ klicken. „Liked“ der User ein Inserat, wird es seinen Favoriten („My Darlings“) hinzugefügt und das nächste Inserat gezeigt. Klickt der

User auf „Dislike“ wird direkt das nächste Inserat gezeigt. Dieser Vorgang wird oft wiederholt, bis keine ungesehenen, den Präferenzen entsprechenden Inserate mehr vorhanden sind oder der User eigenständig den Vorgang beendet. Die Reihenfolge in der die Inserate angezeigt werden ist nicht ganz zufällig: Anbieter können ihre Inserate durch Zahlung einer geringen Gebühr besser positionieren lassen (siehe B2B Use Case „Kauf von Premiumfunktionen“). Im Fehlerfall wird eine entsprechende Fehlermeldung ausgegeben und die Seite „Soul Searching“ weitergeleitet.

Vorbedingung: Der User muss eingeloggt sein, Klick auf "Soul searching"



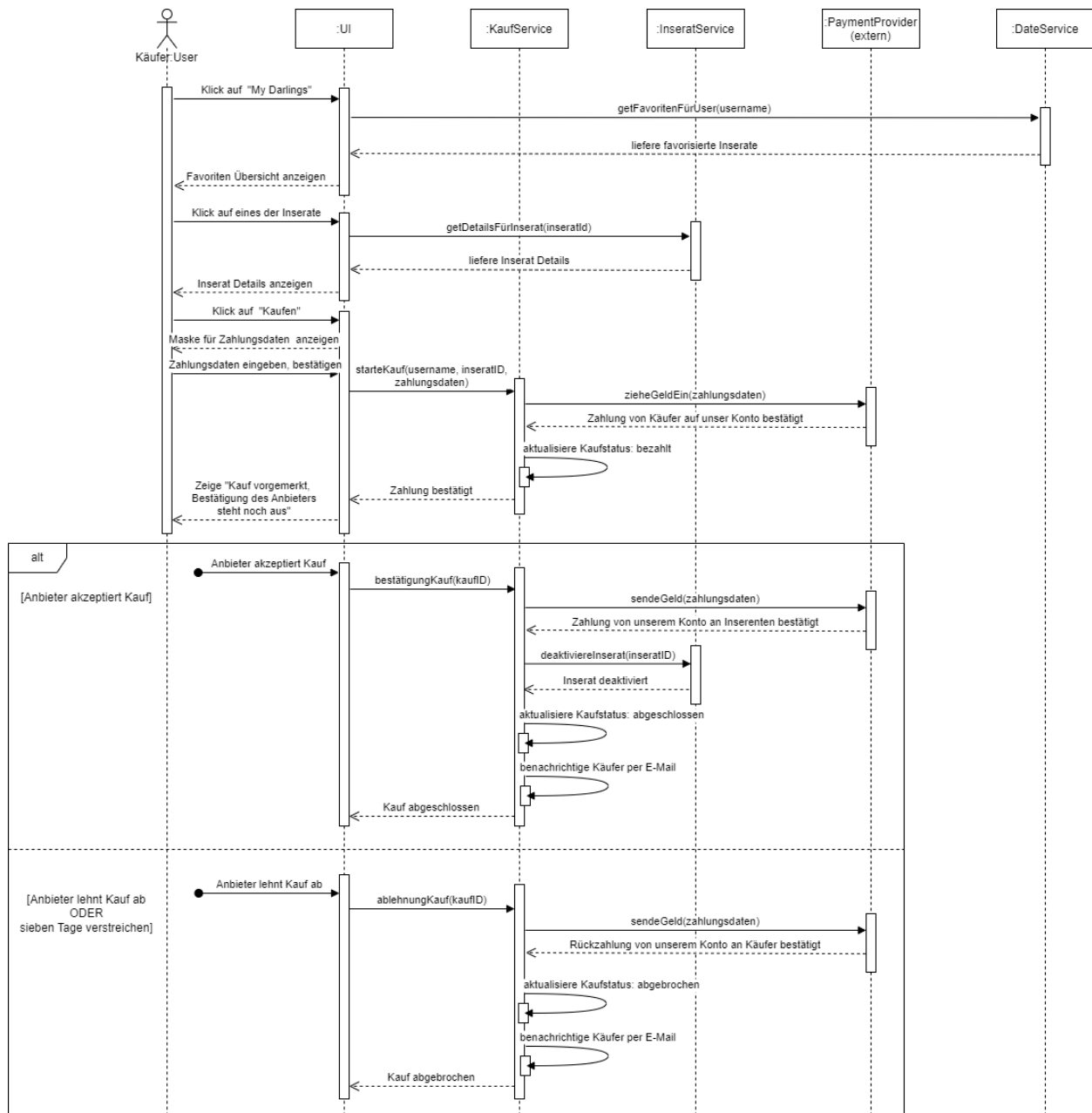
Tier kaufen (B2C)

Für diesen Use Case muss der User eingeloggt sein. Er muss zudem bereits Inserate favorisiert haben (siehe Use Case „Lieblingstiere auswählen“).

Möchte der User ein Tier kaufen, klickt er auf der Übersichtsseite zunächst auf den Bereich „My Darlings“. Dort bekommt er sämtliche von ihm favorisierte Inserate angezeigt. Klickt der User nun auf eines dieser Inserate, bekommt er dessen Details und eine Schaltfläche „Kaufen“ angezeigt. Mit einem Klick auf „Kaufen“ öffnet sich eine Maske, in welcher der User seine Zahlungsdaten eingibt. Mit einem Klick

auf „Bestätigen“ wird der Kauf im System eingetragen, der Kaufbetrag wird vom Konto des Users eingezogen und der User bekommt die Nachricht „Kauf vorgemerkt, Bestätigung des Anbieters steht noch aus“. Bestätigt der Anbieter den Kauf, wird der Kaufbetrag auf das Zahlungsmittel des Anbieters weitergeleitet und der User wird per E-Mail benachrichtigt. Lehnt der Anbieter den Kauf ab oder reagiert sieben Tage nicht, wird dem User das eingezogene Geld auf sein Zahlungsmittel rückerstattet und er wird per E-Mail darüber informiert.

Vorbedingung: Der User muss eingeloggt sein



Widerrufen des Kaufs (B2C)

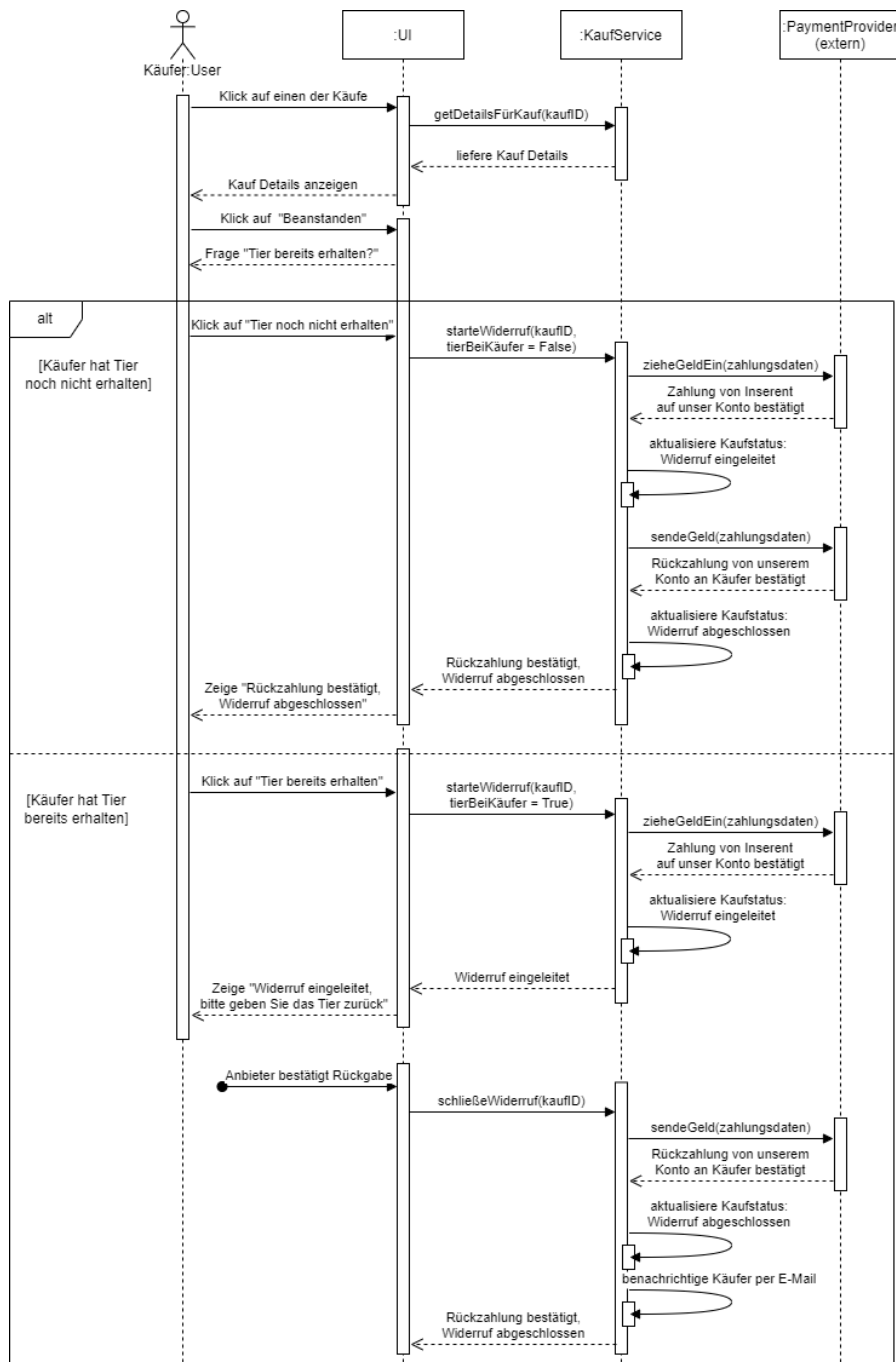
Für diesen Use Case muss der User eingeloggt sein und auf der Übersichtsseite auf „Meine Käufe“ klicken. Er muss zudem bereits mindestens einen Kauf getätigt haben (siehe Use Case „Tier kaufen“).

Möchte der User einen getätigten Kauf widerrufen, klickt er auf diesen Kauf. Er klickt in den sich öffnenden Kaufdetails nun auf "Beanstanden". Darauf öffnet sich eine Auswahlmaske, wo er auswählen kann, ob er das gekaufte Tier bereits erhalten hat, oder ob sich dieses noch beim Anbieter befindet. Klickt er auf „Tier noch nicht erhalten“, wird im System ein Widerruf für den fraglichen Kauf gestartet. Der Kaufbetrag wird vom Zahlungsmittel des Anbieters eingezogen und auf das Zahlungsmittel des Users weitergeleitet. Damit ist der Widerruf abgeschlossen und dem User wird die Nachricht „Rückzahlung bestätigt, Widerruf abgeschlossen“ angezeigt.

Klickt der User auf „Tier bereits erhalten“ wird ebenfalls ein Widerruf im System ausgelöst und der Kaufbetrag vom Zahlungsmittel des Anbieters eingezogen. Dem User wird nun die Nachricht „Widerruf eingeleitet, bitte geben Sie das Tier zurück“ angezeigt. Sobald der Anbieter die Rückgabe des Tieres bestätigt, wird das Geld auf das Zahlungsmittel des Users weitergeleitet, dieser per E-Mail benachrichtigt und der Widerruf im System geschlossen.

Ein User kann nicht die Käufe anderer User widerrufen.

Vorbedingung: Der User muss eingeloggt sein, Klick auf „Meine Käufe“



B2B Use Cases

Für alle B2B Use Cases gilt die folgende Vorbedingung: Die gezeigten API-Calls sind nur authentifiziert mit entsprechendem Token möglich. Der Token wird bei der Registrierung angelegt und ausgegeben. Die Registrierung ist nur über das Frontend möglich (siehe Use Case „Registrieren und Einloggen“).

Tier inserieren (B2B)

Inserate können mittels eines POST-Request mit den Inseratsdetails als payload erstellt werden. Der Endpoint sowie das Schema für den payload sind in Abschnitt C.2 bzw. C.1 dokumentiert. Sollte der Request nicht korrekt sein oder wichtige Daten fehlen, antwortet der Server mit einem Code 400 und der User muss erneut ein Request stellen. Ist das Erstellen der Inserate erfolgreich, erhält der User eine Response mit den erstellten Inseraten als payload.

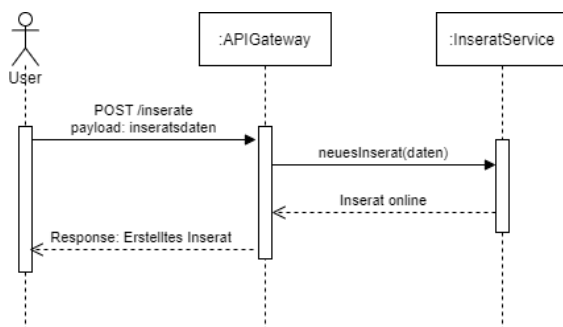
Sollen bestehende Inserate geändert oder gelöscht werden, muss der User zunächst die inseratIDs für die betreffenden Inserate kennen. Sind diese unbekannt, können mit einem GET-Request auf den in C.2 definierten Endpoint `/api/inserate?username={username}` die inseratIDs sämtlicher Inserate des Users ermittelt werden.

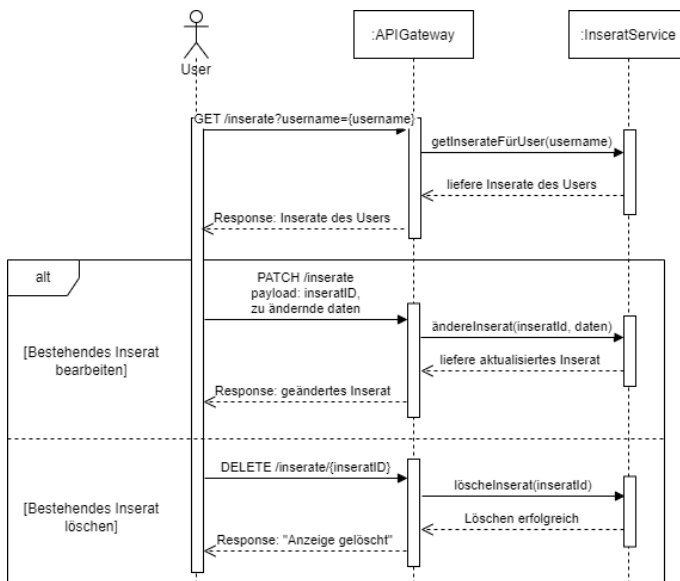
Mit einem PATCH-Request auf den Endpoint `/api/inserate` können nun eigene Inserate verändert werden. Hierzu werden im payload die inseratIDs der zu ändernden Inserate angegeben, jeweils gefolgt von den zu ändernden Details (siehe C.1). Sollte der Request nicht korrekt sein oder wichtige Daten fehlen, antwortet der Server mit einem Code 400 und der User muss erneut ein Request stellen. Ist das Ändern der Inserate erfolgreich, erhält der User eine Response mit den aktualisierten Inseraten als payload.

Zum Löschen eines eigenen Inserates erfolgt ein DELETE-Request auf den Endpoint (siehe C.2) unter Angabe der inseratID. Bei einem erfolgreichen Löschvorgang enthält die Response die Nachricht „Anzeige gelöscht“. Bei einem Fehler enthält die Response den entsprechenden Fehlercode.

Der User kann nicht die Inserate anderer User ändern oder löschen.

Erstellen von Inseraten:

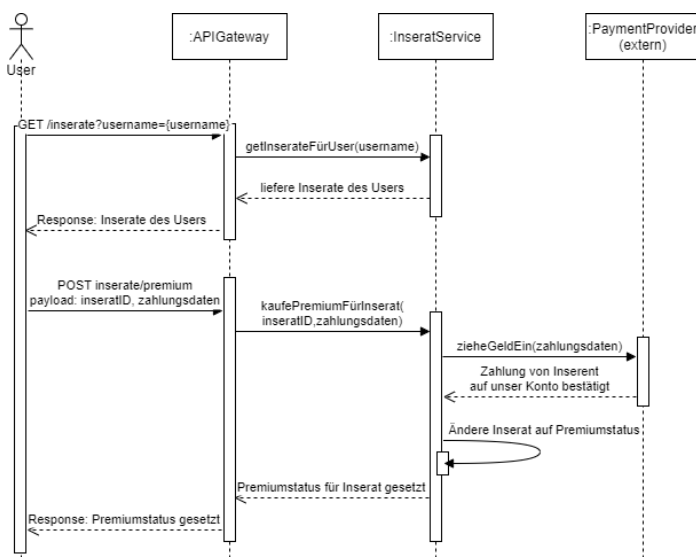


Erstellte Inserate ändern oder löschen:**Kauf von Premiumfunktionen (B2B)**

Um einzelne Inserate stärker hervorzuheben und häufiger bzw. früher bei potenziellen Käufern in der „Soul Search“ erscheinen zu lassen (siehe B2C Use Case „Lieblingstiere auswählen“), kann der User Premiumfunktionen für besagte Inserate kaufen.

Hierfür muss der User zunächst die inseratIDs der entsprechenden Inserate kennen. Sind diese unbekannt, können sie mit einem GET-Request auf den in Endpoint `/api/inserate?username={username}` ermittelt werden.

Nun erfolgt ein POST-Request auf den Endpoint `/api/inserate/premium` mit den inseratIDs sowie den Zahlungsdaten des Users im Payload. Daraufhin wird die Zahlung angestoßen und die entsprechenden Inserate auf Premiumstatus gehoben. Die Response enthält als payload die betroffenen inseratIDs sowie die Nachricht „Premiumstatus gesetzt“. Tritt während der Zahlung ein Problem auf oder entsprach das payload der Request nicht der EDI-Sprache, antwortet der Server mit einem entsprechenden Fehlercode.

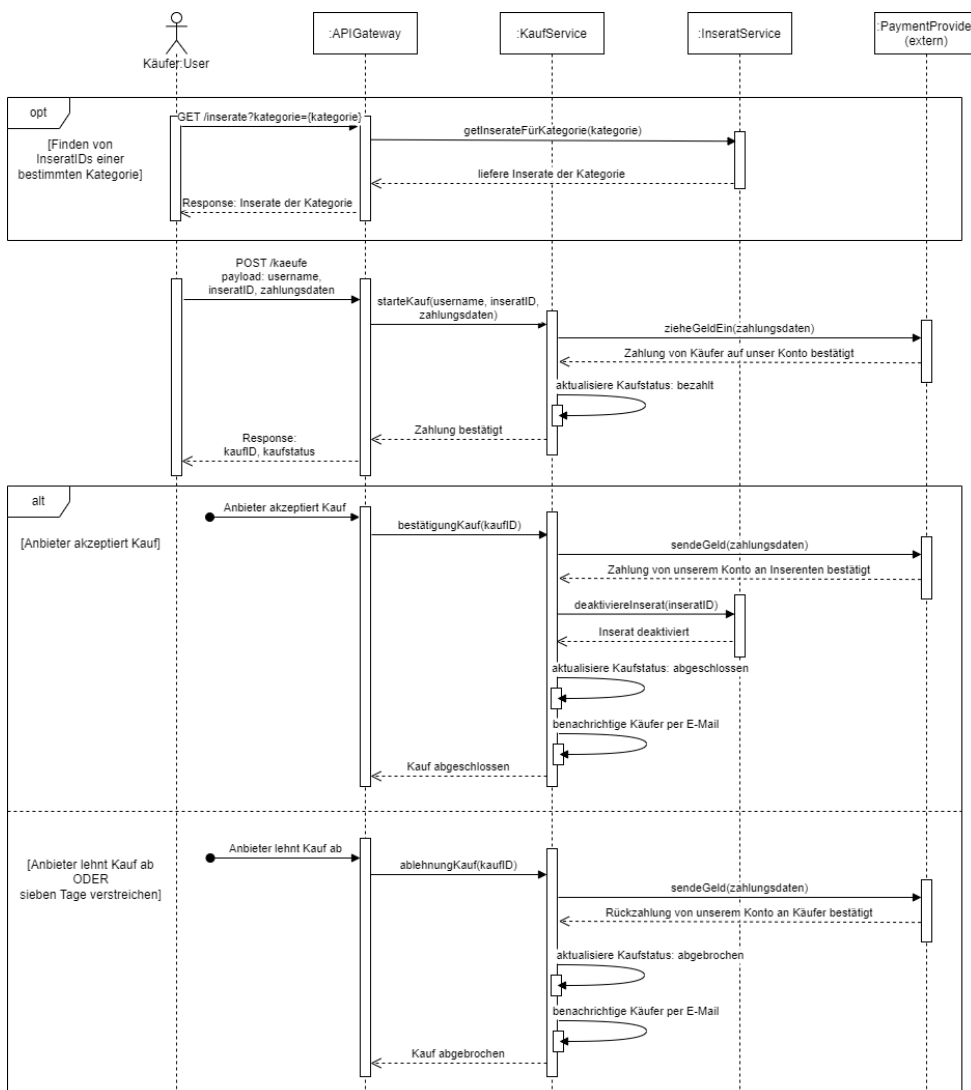


Tier kaufen (B2B)

Möchte ein User Tiere kaufen, muss er zunächst die inseratIDs für die betreffenden Inserate kennen. Sind diese unbekannt, können mit einem GET-Request auf den in C.2 definierten Endpoint `/api/inserate` unter Hinzugabe entsprechender Parameter sämtliche Inserate durchsucht und so die fraglichen inseratIDs ermittelt werden.

Mit einer POST-Request auf den Endpoint `/api/kaeufe` können nun die Käufe getätigt werden. Die payload muss inseratIDs sowie die Zahlungsdaten des Users enthalten. Der Kauf wird somit im System eingetragen und der Kaufbetrag vom Konto des Users eingezogen. Die Response enthält die kaufIDs sowie den jeweiligen Kaufstatus („Kauf vorgemerkt, Bestätigung des Anbieters steht noch aus“). Tritt während der Zahlung ein Problem auf oder entsprach das payload der Request nicht der EDI-Sprache, antwortet der Server mit einem entsprechenden Fehlercode.

Bestätigt der Anbieter den Kauf, wird der Kaufbetrag auf das Zahlungsmittel des Anbieters weitergeleitet und der User wird per E-Mail benachrichtigt. Lehnt der Anbieter den Kauf ab oder reagiert sieben Tage nicht, wird dem User das eingezogene Geld auf sein Zahlungsmittel rückerstattet und er wird per E-Mail darüber informiert.



Widerrufen des Kaufs (B2B)

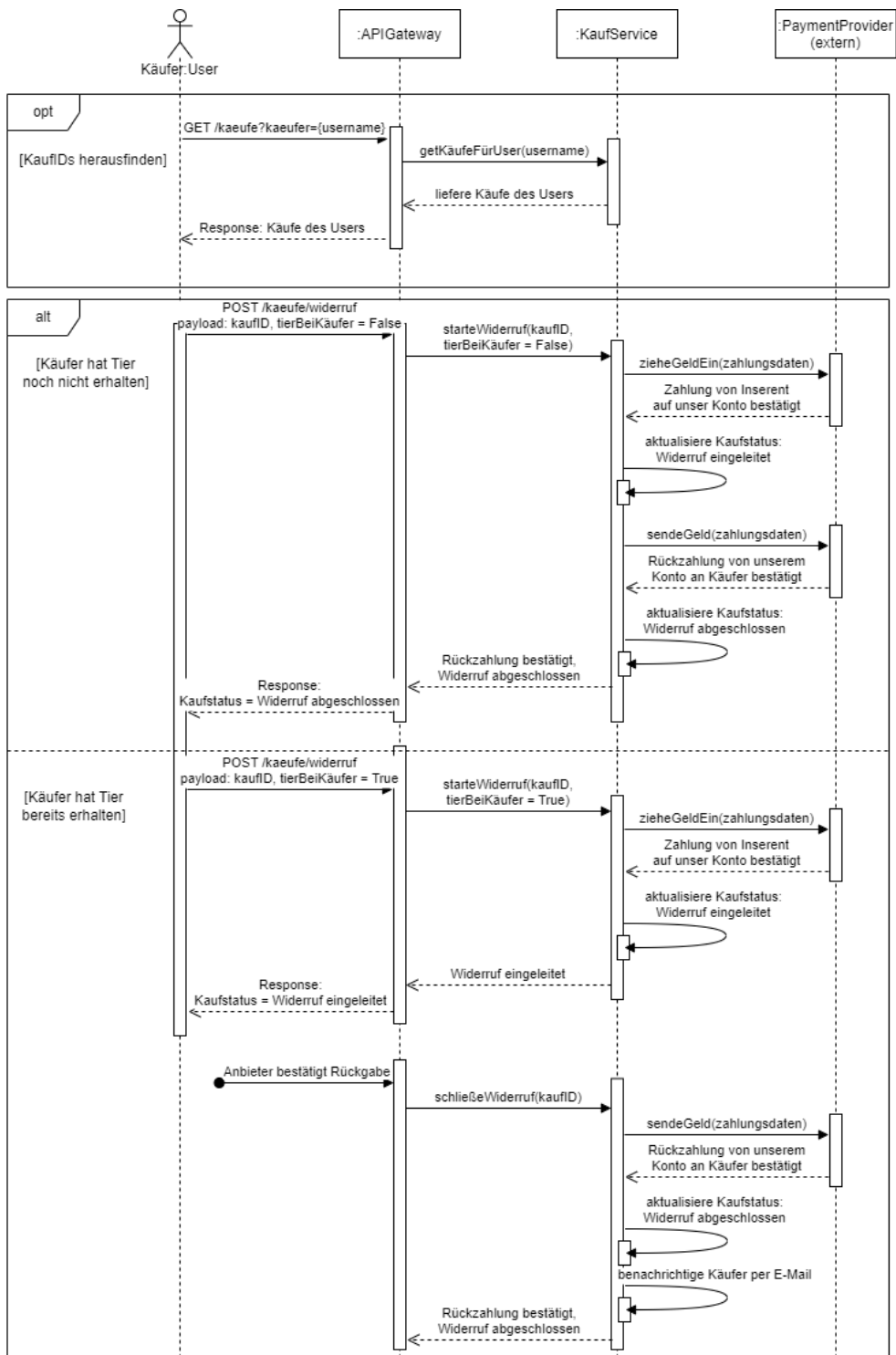
Für diesen Use Case muss der bereits mindestens einen Kauf getätigt haben (siehe Use Case „Tier kaufen“).

Möchte ein User Käufe widerrufen, muss er zunächst die kaufIDs für die betreffenden Käufe kennen. Sind diese unbekannt, können mit einem GET-Request auf den in C.2 definierten Endpoint `/api/kaeufe?username={username}` die kaufIDs sämtlicher Käufe des Users ermittelt werden.

Widerrufe können mit einem POST-Request auf den Endpoint `/api/kaeufe/widerruf` gestartet werden. In der payload ist neben den entsprechenden kaufIDs jeweils anzugeben, ob der User das gekaufte Tier bereits erhalten hat, oder ob sich dieses noch beim Anbieter befindet. Gibt der User an, das Tier noch nicht erhalten zu haben, wird im System ein Widerruf für den fraglichen Kauf gestartet. Der Kaufbetrag wird vom Zahlungsmittel des Anbieters eingezogen und auf das Zahlungsmittel des Users weitergeleitet. Damit ist der Widerruf abgeschlossen durch den in der Response enthaltenen Kaufstatus wiedergegeben wird („Widerruf abgeschlossen“).

Gibt der User im payload des Requests an, das Tier bereits erhalten zu haben, wird ebenfalls ein Widerruf im System ausgelöst und der Kaufbetrag vom Zahlungsmittel des Anbieters eingezogen. Die Response enthält nun den Kaufstatus „Widerruf eingeleitet“ sowie die Nachricht „Bitte geben Sie das Tier an den Anbieter zurück“. Sobald der Anbieter die Rückgabe des Tieres bestätigt, wird das Geld auf das Zahlungsmittel des Users weitergeleitet, dieser per E-Mail benachrichtigt und der Widerruf im System geschlossen. Tritt während des Vorgangs ein Problem auf oder entsprach das payload der Request nicht der EDI-Sprache, antwortet der Server mit einem entsprechenden Fehlercode.

Ein User kann nicht die Käufe anderer User widerrufen.



Nicht umgesetzte Use Cases

Die folgenden Use Cases wären zur vollständigen Implementierung aller in Abgabe 1 beschriebenen Funktionalitäten von Nöten. Sie seien daher an dieser Stelle der Vollständigkeit halber kurz skizziert. Da deren Implementierung den Umfang für diese und die folgende Abgabe (vier Use Cases) sprengen würde, beschränke ich mich fortan auf die obigen Kern Use Cases.

„Date“ vereinbaren (B2C)

Für favorisierte Tiere („My Darlings“, siehe Use Case „Lieblingstiere auswählen“) kann der Kunde per Klick auf das Tier mit dessen Besitzer (Anbieter) kommunizieren und über die Plattform ein „Date“ vereinbaren, zu welchem man das Tier einmal ausführen kann, bevor man sich für einen Kauf entscheidet. Akzeptiert der Anbieter die Datinganfrage, erhält der Käufer entsprechend eine Bestätigung.

Futter u. Spielzeugpaket kaufen (B2C)

Vereinbart der Käufer ein solches Date (siehe Use Case „Date vereinbaren“), kann dieser Futter- und Spielzeugpakete passend zum Treffen erwerben. Diese werden ihm direkt bei der Datevereinbarung vorgeschlagen, sind spezifisch auf Tierart und Rasse des involvierten Tieres zugeschnitten und bieten somit eine bequeme Form für den Nutzer, sich auf das Treffen vorzubereiten. Es handelt sich wiederum um eine Vermittlungsleistung: Der eigentliche Anbieter des Paketes ist jeweils eine Tierhandlung (siehe Use Case „Futter- u. Spielzeugpakete inserieren“). Kommt ein Verkauf zustande, wird von der Plattform eine entsprechende Bestellung an das System des Anbieters gesendet, das Geld vom Käufer eingezogen und an den Anbieter weitergeleitet. Hierbei erhält die Plattform eine Provision.

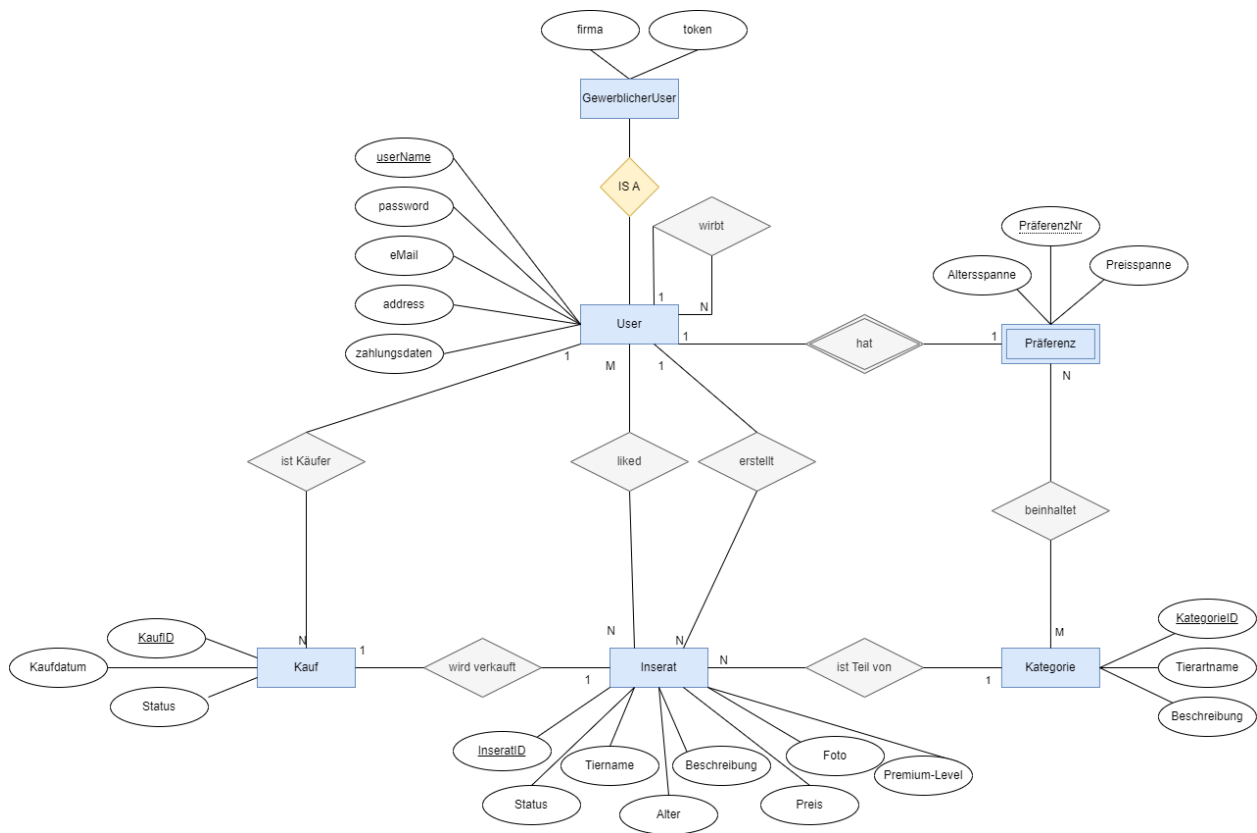
Futter- u. Spielzeugpakete inserieren (B2B)

Das Einstellen, Ändern und Löschen der Artikel für Futter- und Spielzeugpakete kann über eine API erfolgen. Hierüber können Händler ihre Artikel mit Angabe der geeigneten Tierart und verfügbaren Stückzahl einstellen. Spirit Animal stellt aus den angebotenen Artikeln jeweils automatisiert Pakete zusammen, welche Käufern wie in Use Case „Futter u. Spielzeugpaket kaufen“ beschrieben angeboten werden.

Statistiken abrufen (B2B)

Des Weiteren können Anbieter über die API Statistiken abrufen. So erhalten sie jederzeit Auskunft über die beliebtesten Tiere auf der Plattform sowie ihre eigenen Verkaufserfolge.

B: Datenmodell



Ein User wird mit den Daten angelegt, welche bei der Registrierung im System angegeben werden. Gewerbliche User („B2B-User“) geben zusätzlich ihren Firmennamen an und erhalten einen Token, um sich an der B2B-Schnittstelle authentifizieren zu können. Über unser Kunden-werben-Kunden Programm kann ein User angeben, dass er von einem anderen User geworben wurde.

Ein User kann nun Inserate erstellen. Zusätzlich zu den vom User angegebenen Informationen werden vom System noch der aktuelle Status und das Premium-Level des Inserates festgehalten.

Ein User kann seine Präferenzen festlegen. Diese werden verwendet, um die Inserate für den Use Case „Lieblingstiere auswählen“ vorzufiltern bevor sie dem User präsentiert werden. Aus diesem Grund beinhalten Präferenzen auch dieselben im System festgelegten Kategorien wie sie auch die Inserate verwenden. Außerdem kann ein User bestehende Inserate liken (Use Case „Lieblingstiere auswählen“), wodurch diese Inserate später in seinen Favoriten auftauchen.

Ein Kauf beinhaltet immer das Inserat, welches verkauft wurde, sowie einen User als Käufer. Der Anbieter kann über die Inserat-Entität ermittelt werden. Zusätzlich werden vom System das Kaufdatum sowie der aktuelle Status des Kaufvorgangs festgehalten.

C: Definition der EDI Sprache für B2B Use Case Prozesse und B2B Schnittstellenbeschreibung

C.1 EDI Sprache

B2B-Kunden können sich über eine Art Extranet zu unserem System verbinden. Diese Kommunikation zwischen den Systemen findet über eine Schnittstelle (REST-API) statt, welche in Absatz C.2 näher beschrieben wird. Für die erfolgreiche Kommunikation über diese Schnittstelle muss ein gewisses Schema (EDI-Sprache) eingehalten werden, welches im Folgenden definiert wird. Die EDI-Sprache des Systems ist basiert auf der JSON-Schema Spezifikation. Diese Schemata definieren den Aufbau von Requests und dienen somit als Dokumentation für Kunden, welche unsere B2B-Schnittstelle nutzen wollen. Requests, die das entsprechende Schema nicht erfüllen werden mit einem Code 400 (Bad Request) abgelehnt und nicht an das Backend weitergeleitet.

Inserat erstellen

Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "spirit_animal_edi_language",
  "title": "Tier inserieren",
  "description": "Einstellen von Tier-Inseraten ueber die SpiritAnimal-API",
  "type": "object",
  "properties": {
    "token": {
      "description": "Der Token zur Authentifizierung des API-Calls",
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "daten": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "tiername": {
            "description": "Der Name des inserierten Tieres",
            "type": "string"
          },
          "kategorie": {
            "description": "Die Kategorie des inserierten Tieres (z.B. 'HUNDE' oder 'SPANNENDE EXOTEN')",
            "type": "string",
            "enum": ["HUNDE", "KATZEN", "KANINCHEN", "MEERSCHWEINCHEN", "WEITERE NAGER", "SPANNENDE EXOTEN", "PFERDE"]
          },
          "beschreibung": {
            "description": "Eine kurze textuelle Beschreibung des inserierten Tieres",
            "type": "string"
          },
          "alter": {
            "description": "Das Alter des inserierten Tieres in Jahren",
            "type": "integer"
          },
          "preis": {
            "description": "Der Preis des inserierten Tieres",
            "type": "number"
          },
          "foto_base64": {
            "description": "Ein Foto des inserierten Tieres, base64-encoded",
            "type": "string"
          }
        },
        "required": ["tiername", "kategorie", "beschreibung", "alter", "preis"]
      }
    }
  },
  "required": ["token", "username", "daten"]
}
```

Beispiel Datenaustausch

Request

```
{
  "token": "sadffjineasd",
  "username": "tierliebe_heinz",
  "daten": [
    {
      "tiername": "Shelly",
      "kategorie": "HUNDE",
      "beschreibung": "Shelly ist ein liebenswerter brauner Hund. Chelly mag Teddybären besonders gern.",
      "alter": 4,
      "preis": 199.99,
      "foto_base64": "{base64-encoded image}"
    },
    {
      "tiername": "Rufus",
      "kategorie": "WEITERE NAGER",
      "beschreibung": "Rufus ist unser kuscheligster Nacktmull. Er besticht durch seinen exotischen Charme und seine gemütliche Art.",
      "alter": 25,
      "preis": 24.99
    }
  ]
}
```

Response

```
{
  "token": "sadffjineasd",
  "username": "tierliebe_heinz",
  "erstellt": [
    {
      "inseratID": "i10023",
      "inseratStatus": "online",
      "tiername": "Shelly",
      "kategorie": "HUNDE",
      "beschreibung": "Shelly ist ein liebenswerter brauner Hund. Chelly mag Teddybären besonders gern.",
      "alter": 4,
      "preis": 199.99,
      "foto": true
    },
    {
      "inseratID": "i10024",
      "inseratStatus": "online",
      "tiername": "Rufus",
      "kategorie": "WEITERE NAGER",
      "beschreibung": "Rufus ist unser kuscheligster Nacktmull. Er besticht durch seinen exotischen Charme und seine gemütliche Art.",
      "alter": 25,
      "preis": 24.99,
      "foto": false
    }
  ]
}
```

Inserat ändern

Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "spirit_animal_api_language",
  "title": "Inserat aendern",
  "description": "Aendern von Tier-Inseraten ueber die SpiritAnimal-API",
  "type": "object",
  "properties": {
    "token": {
      "description": "Der Token zur Authentifizierung des API-Calls",
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "daten": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "inseratID": {
            "description": "Die ID fuer das zu aendernde Inserat",
            "type": "string"
          },
          "tiername": {
            "description": "Der Name des inserierten Tieres",
            "type": "string"
          },
          "kategorie": {
            "description": "Die Kategorie des inserierten Tieres (z.B. 'HUNDE' oder 'SPANNENDE EXOTEN')",
            "type": "string",
            "enum": ["HUNDE", "KATZEN", "KANINCHEN", "MEERSCHWEINCHEN", "WEITERE NAGER", "SPANNENDE EXOTEN", "PFERDE"]
          },
          "beschreibung": {
            "description": "Eine kurze textuelle Beschreibung des inserierten Tieres",
            "type": "string"
          },
          "alter": {
            "description": "Das Alter des inserierten Tieres in Jahren",
            "type": "integer"
          },
          "preis": {
            "description": "Der Preis des inserierten Tieres",
            "type": "number"
          },
          "foto_base64": {
            "description": "Ein Foto des inserierten Tieres, base64-encoded",
            "type": "string"
          }
        }
      }
    },
    "required": ["inseratID"]
  },
  "required": ["token", "username", "daten"]
}
```

Beispiel Datenaustausch

Request

```
{
  "token": "saffjineasd",
  "username": "tierliebe_heinz",
  "daten": [
    {
      "inseratID": "i10023",
      "tiername": "Chilly",
      "beschreibung": "Chilly ist ein liebenswerter brauner Hund. Chilly mag Teddybären besonders gern."
    },
    {
      "inseratID": "i10024",
      "foto_base64": "{base64-encoded image}"
    }
  ]
}
```

Response

```
{
  "token": "saffjineasd",
  "username": "tierliebe_heinz",
  "geaendert": [
    {
      "inseratID": "i10023",
      "inseratStatus": "online",
      "tiername": "Chilly",
      "kategorie": "HUNDE",
      "beschreibung": "Chilly ist ein liebenswerter brauner Hund. Chilly mag Teddybären besonders gern.",
      "alter": 4,
      "preis": 199.99,
      "foto": true
    },
    {
      "inseratID": "i10024",
      "inseratStatus": "online",
      "tiername": "Rufus",
      "kategorie": "WEITERE NAGER",
      "beschreibung": "Rufus ist unser kuscheligster Nacktmull. Er besticht durch seinen exotischen Charme und seine gemütliche Art.",
      "alter": 25,
      "preis": 24.99,
      "foto_base64": true
    }
  ]
}
```

Premiumfunktionen kaufen

Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "spirit_animal_edi_language",
  "title": "Premiumfunktionen kaufen",
  "description": "Kaufen von Premiumfunktionen fuer Tier-Inserate ueber die SpiritAnimal-API",
  "type": "object",
  "properties": {
    "token": {
      "description": "Der Token zur Authentifizierung des API-Calls",
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "inseratID": {
      "description": "Ein Array mit den IDs der Inserate, welche den Premiumstatus erhalten sollen",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "zahlungsdaten": {
      "type": "object",
      "properties": {
        "option": {
          "description": "Die gewuenschte Bezahloption (Paypal, Kreditkarte oder SEPA-Lastschrift)",
          "type": "string",
          "enum": ["PAYPAL", "KREDITKARTE", "SEPA"]
        },
        "paymentHandle": {
          "description": "Je nach Bezahloption Angabe des paypal-handles, der Kreditkartennummer oder der Kontonummer",
          "type": "string"
        }
      },
      "required": ["option", "paymentHandle"]
    }
  },
  "required": ["token", "username", "inseratID", "zahlungsdaten"]
}
```

Beispiel Datenaustausch

Request

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz",
  "inseratID": ["i10023", "i10024"],
  "zahlungsdaten": {
    "option": "PAYPAL",
    "paymentHandle": "payment@heinz.at"
  }
}
```

Response

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz",
  "premiumGesetzt": [
    {
      "inseratID": "i10023",
      "premium": true
    },
    {
      "inseratID": "i10024",
      "premium": true
    }
  ]
}
```

Tier kaufen

Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "spirit_animal_edl_language",
  "title": "Tier kaufen",
  "description": "Kaufen von inserierten Tieren ueber die SpiritAnimal-API",
  "type": "object",
  "properties": {
    "token": {
      "description": "Der Token zur Authentifizierung des API-Calls",
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "inseratID": {
      "description": "Ein Array mit den IDs der Inserate, welche gekauft werden sollen",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "zahlungsdaten": {
      "type": "object",
      "properties": {
        "option": {
          "description": "Die gewuenschte Bezahloption (Paypal, Kreditkarte oder SEPA-Lastschrift)",
          "type": "string",
          "enum": ["PAYPAL", "KREDITKARTE", "SEPA"]
        },
        "paymentHandle": {
          "description": "Je nach Bezahloption Angabe des paypal-handles, der Kreditkartennummer oder der Kontonummer",
          "type": "string"
        }
      },
      "required": ["option", "paymentHandle"]
    }
  },
  "required": ["token", "username", "inseratID", "zahlungsdaten"]
}
```

Beispiel Datenaustausch

Request

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz",
  "inseratID": ["i10010", "i10032"],
  "zahlungsdaten": {
    "option": "PAYPAL",
    "paymentHandle": "payment@heinz.at"
  }
}
```

Response

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz",
  "gekauft": [
    {
      "kaufID": "k1042",
      "inseratID": "i10010",
      "kaufstatus": "BEZAHLT"
    },
    {
      "kaufID": "k1043",
      "inseratID": "i10032",
      "kaufstatus": "BEZAHLT"
    }
  ]
}
```

Widerruf

Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "spirit_animal_edi_language",
  "title": "Widerruf",
  "description": "Widerruf eines Kaufs ueber die SpiritAnimal-API",
  "type": "object",
  "properties": {
    "token": {
      "description": "Der Token zur Authentifizierung des API-Calls",
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "widerruf": {
      "type": "object",
      "properties": {
        "kaufID": {
          "description": "Die ID des Kaufs, welcher widerrufen werden soll",
          "type": "string"
        },
        "tierBeiKaeufer": {
          "description": "Angabe, ob das gekaufte Tier bereits beim Kaeufer angekommen ist",
          "type": "boolean"
        }
      },
      "required": ["kaufID", "tierBeiKaeufer"]
    }
  },
  "required": ["token", "username", "widerruf"]
}
```

Beispiel Datenaustausch

Request

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz",
  "widerruf": [
    {
      "kaufID": "k1043",
      "tierBeiKaeufer": false
    },
    {
      "kaufID": "k1002",
      "tierBeiKaeufer": true
    }
  ]
}
```

Response

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz",
  "widerrufen": [
    {
      "kaufID": "k1043",
      "status": "WIDERRUF ABGESCHLOSSEN"
    },
    {
      "kaufID": "k1002",
      "status": "WIDERRUF EINGELEITET"
    }
  ]
}
```

C.2 B2B Schnittstelle

Bei der Schnittstelle handelt sich um eine REST-API, welche von unseren B2B-Kunden über entsprechende HTTP-Endpoints angesprochen werden kann. Im Folgenden werden die Endpoints und deren Funktionalität beschrieben.

/api/inserate

Inserate dursuchen (GET /api/inserate)

Gibt alle Inserate zurück, die den angegebenen Query-Parametern entsprechen. Gibt alle existierenden Inserate zurück, wenn keine Parameter angegeben werden. Calls auf diesen Endpoint benötigen keine Authentifizierung.

Mögliche URL-Parameter: *username, kategorie, preismin, preismax, altermin, altermax*

Beispiele:

GET /api/inserate?username=dogs_paradise

- liefert alle Inserate, welche von User dogs_paradise erstellt wurden

GET /api/inserate?kategorie=hunde&altermax=5&preismin=150

- liefert alle Inserate der Kategorie Hunde, welche maximal 5 Jahre alt sind und mindestens 150 € kosten.

Inserate erstellen (POST /api/inserate)

Erstellt die Inserate welche im Payload des Requests geliefert werden. Der Payload richtet sich nach dem Schema „Inserat erstellen“ der EDI-Sprache (siehe C.1).

Inserate ändern (PATCH /api/inserate)

Ändert die Inserate welche im Payload des Requests geliefert werden. Der Payload richtet sich nach dem Schema „Inserat ändern“ der EDI-Sprache (siehe C.1).

Inserat löschen (DELETE /api/inserate/{inseratID})

Löscht das Inserat mit ID inseratID. Da ein User nur seine eigenen Inserate löschen kann, muss ein solcher Request authentifiziert erfolgen. In der Payload der Request müssen also Username und API-Token mitgeliefert werden.

Beispiel-Payload:

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz"
}
```

Kauf von Premiumfunktionen (POST /api/inserate/premium)

Ermöglicht das Kaufen von Premiumfunktionen für eigene Inserate. Der Payload richtet sich nach dem Schema „Premiumfunktionen kaufen“ der EDI-Sprache (siehe C.1).

/api/kaeufe

Eigene Käufe einsehen (GET /api/kaeufe?kaeuer={username})

Liefert alle Käufe, bei denen der User als Käufer auftritt. Da ein User nur seine eigenen Käufe einsehen kann, muss ein solcher Request authentifiziert erfolgen. In der Payload des Request müssen also Username und API-Token mitgeliefert werden.

Beispiel-Payload:

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz"
}
```

Eigene Verkäufe einsehen (GET /api/kaeufe?anbieter={username})

Liefert alle Käufe, bei denen der User als Anbieter auftritt. Da ein User nur seine eigenen Verkäufe einsehen kann, muss ein solcher Request authentifiziert erfolgen. In der Payload des Request müssen also Username und API-Token mitgeliefert werden.

Beispiel-Payload:

```
{
  "token": "sadjfineasd",
  "username": "tierliebe_heinz"
}
```

Tier kaufen (POST /api/kaeufe)

Ermöglicht das Kaufen von inserierten Tieren. Der Payload richtet sich nach dem Schema „Tier kaufen“ der EDI-Sprache (siehe C.1).

Widerruf (POST /api/kaeufe/widerruf)

Ermöglicht das Widerrufen von getätigten Käufen. Der Payload richtet sich nach dem Schema „Widerruf“ der EDI-Sprache (siehe C.1).

C.3 Protokollsequenzen

Tier inserieren

Schritt	Request/Response	Sender	Empfänger	Methode	Endpoint	Response-Code	Payload
1.1	Request	B2B-User	SpiritAnimal API	POST	/api/inserate	-	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "daten": [{ "tiername": "Shelly", "kategorie": "HUNDE", "beschreibung": "Shelly ist ein liebenswerter brauner Hund.", "alter": 4, "preis": 199.99, "foto_base64": "{base64-encoded image}" }, { "tiername": "Rufus", "kategorie": "WEITERE NAGER", "beschreibung": "Rufus ist unser kuscheligster Nacktmull.", "alter": 25, "preis": 24.99 }] }</pre>
1.2	Response	SpiritAnimal API	B2B-User			200	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "erstellt": [{ "inseratID": "i10023", "inseratStatus": "online", "tiername": "Shelly", "kategorie": "HUNDE", "beschreibung": "Shelly ist ein liebenswerter brauner Hund.", "alter": 4, "preis": 199.99, "foto": true }, { "inseratID": "i10024", "inseratStatus": "online", "tiername": "Rufus", "kategorie": "WEITERE NAGER", "beschreibung": "Rufus ist unser kuscheligster Nacktmull.", "alter": 25, "preis": 24.99, "foto": false }] }</pre>
2.1	Request	B2B-User	SpiritAnimal API	PATCH	/api/inserate	-	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "daten": [{ "inseratID": "i10023", "tiername": "Chilly", "beschreibung": "Chilly ist ein liebenswerter brauner Hund." }, { "inseratID": "i10024", "foto_base64": "{base64-encoded image}" }] }</pre>
2.2	Response	SpiritAnimal API	B2B-User			200	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "geaendert": [{ "inseratID": "i10023", "inseratStatus": "online", "tiername": "Chilly", "kategorie": "HUNDE", "beschreibung": "Chilly ist ein liebenswerter brauner Hund.", "alter": 4, "preis": 199.99, "foto": true }, { "inseratID": "i10024", "inseratStatus": "online", "tiername": "Rufus", "kategorie": "WEITERE NAGER", "beschreibung": "Rufus ist unser kuscheligster Nacktmull.", "alter": 25, "preis": 24.99, "foto_base64": true }] }</pre>

Kauf von Premiumfunktionen

Schritt	Request/Response	Sender	Empfänger	Methode	Endpoint	Response-Code	Payload
1.1	Request	B2B-User	SpiritAnimal API	GET	/api/inserate?username=tierliebe_heinz	-	-
1.2	Response	SpiritAnimal API	B2B-User			200	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "inserate": [{ "inseratID": "i10023", "inseratStatus": "online", "tiername": "Chilly", "kategorie": "HUNDE", "beschreibung": "Chilly ist ein liebenswerter brauner Hund.", "alter": 4, "preis": 199.99, "foto": true }, { "inseratID": "i10024", "inseratStatus": "online", "tiername": "Rufus", "kategorie": "WEITERE NAGER", "beschreibung": "Rufus ist unser kuscheligster Nacktmull.", "alter": 25, "preis": 24.99, "foto_base64": true }] }</pre>
2.1	Request	B2B-User	SpiritAnimal API	POST	/api/inserate/premium	-	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "inseratID": ["i10023", "i10024"], "zahlungsdaten": { "option": "PAYPAL", "paymentHandle": "payment@heinz.at" } }</pre>
2.2	Response	SpiritAnimal API	B2B-User			200	<pre>{ "token": "sadjfineasd", "username": "tierliebe_heinz", "premiumGesetzt": [{ "inseratID": "i10023", "premium": true }, { "inseratID": "i10024", "premium": true }] }</pre>

Tier kaufen

Schritt	Request/Response	Sender	Empfänger	Methode	Endpoint	Response-Code	Payload
1.1	Request	B2B-User	SpiritAnimal API	GET	/api/inserate?kategorie=hunde&altermax=5&preismin=150	-	
1.2	Response	SpiritAnimal API	B2B-User			200	{ "insertate": [{ "insertatID": "i10023", "insertatStatus": "online", "tiername": "Chilly", "kategorie": "HUNDE", "beschreibung": "Chilly ist ein liebenswerter brauner Hund.", "alter": 4, "preis": 199.99, "foto": true }] }
2.1	Request	B2B-User	SpiritAnimal API	POST	/api/kaeufe	-	{ "token": "asdfoijehiu", "username": "dogs_paradise", "insertatID": ["i10023"], "zahlungsdaten": { "option": "PAYPAL", "paymentHandle": "dogs@paradise.at" } }
2.2	Response	SpiritAnimal API	B2B-User			200	{ "token": "asdfoijehiu", "username": "dogs_paradise", "gekauft": [{ "kaufID": "k1044", "insertatID": "i10023", "kaufstatus": "BEZAHLT" }] }

Widerrufen des Kaufs

Schritt	Request/Response	Sender	Empfänger	Methode	Endpoint	Response-Code	Payload
1.1	Request	B2B-User	SpiritAnimal API	GET	/api/kaeufe?kaeuer=dogs_paradise	-	{ "token": "asdfoijehiu", "username": "dogs_paradise" }
1.2	Response	SpiritAnimal API	B2B-User			200	{ "token": "asdfoijehiu", "username": "dogs_paradise", "kaeufe": [{ "kaufID": "k1044", "insertatID": "i10023", "kaufstatus": "BEZAHLT" }] }
2.1	Request	B2B-User	SpiritAnimal API	POST	/api/kaeufe/widerruf	-	{ "token": "asdfoijehiu", "username": "dogs_paradise", "widerruf": [{ "kaufID": "k1044", "tierBeiKaeufer": false }] }
2.2	Response	SpiritAnimal API	B2B-User			200	{ "token": "asdfoijehiu", "username": "dogs_paradise", "widerrufen": [{ "kaufID": "k1044", "status": "WIDERRUF ABGESCHLOSSEN" }] }

C.4 GraphQL-Schnittstelle

```
type User {
  username: String!
}

type Inserat {
  inseratID: ID!
  anbieter: User!
  tiername: String!
  beschreibung: String!
  alter: Int
  preis: Float!
  foto: String
  status: String
}

type Kauf {
  kaufID: ID!
  kaeufer: User!
  inserat: Inserat!
  kaufdatum: Date!
  status: String!
}

type Query {
  # liefert ein Inserat
  getInserat(inseratID: String!): Inserat

  # liefert alle Inserate
  getAlleInserate: [Inserat]

  # liefert alle Kaeufe eines Users
  getKaeufeFuerUser(username: String!): [Kauf]
}

type Mutation {
  createInserat(
    inseratID: String!, username: String!, tiername: String!, beschreibung: String!,
    alter: Int, preis: Float!, foto: String, status: String!
  ): Inserat
}

# Beispiel-Queries
query {
  getAlleInserate {
    inseratID
    tiername
    status
  }
}

query {
  getKaeufeFuerUser(username: dogs_paradise) {
    kaufID
    inserat {
      tiername
    }
  }
}

# Beispiel-Mutation
mutation {
  createInserat(
    username: String!, tiername: String!, beschreibung: String!,
    alter: Int, preis: Float!, foto: String
  ) {
    inseratID
    status
  }
}
```