



Vom Formular zum Dialog: Neue Wege der Auftragserteilung in der Finanzbranche

Forschungsbericht zum Abschluss des
Innosuisse Innovationsprojekts 27130.1 PFES-ES

Autoren:

Adrian Moser, Elke Brucker-Kley, Alexandre de Spindler,
Ninja Leikert-Böhm

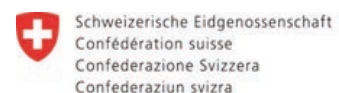
Mit Beiträgen der Wirtschaftspartner:

Marc Mallepell & Raphael Bianchi, Synpulse
Andreas Barattiero, St.Galler Kantonalbank

Wirtschaftspartner:



Gefördert durch:



Kommission für Technologie und Innovation KTI

IMPRESSUM

Herausgeber

ZHAW School of Management and Law
Theaterstrasse 17
8400 Winterthur
Schweiz

Abteilung General Management
Institut für Wirtschaftsinformatik
www.zhaw.ch/iwi

Kontakt, Co-Projektleitung:
Elke Brucker-Kley
elke.brucker-kley@zhaw.ch
Adrian Moser
adrian.moser@zhaw.ch

Juli 2021

<https://doi.org/10.21256/zhaw-2407>

Copyright © 2021,
ZHAW School of Management and Law

Alle Rechte für den Nachdruck und die
Vervielfältigung dieser Arbeit liegen bei der
Abteilung General Management der
ZHAW School of Management and Law.
Die Weitergabe an Dritte bleibt ausgeschlossen

Dank

Das angewandte Forschungsprojekt "Vom Formular zum Dialog: Neue Wege der Auftragserteilung in der Finanzbranche" (Projektnummer: 27130.1 PFES-ES) wurde von der Schweizerischen Agentur für Innovationsförderung Innosuisse gefördert. Die prototypische Lösung, die im Rahmen des Projektes entstanden ist, konnte dank der Banken- und Avaloq-Expertise des Implementierungspartners Synpulse gemeinsam mit Forschenden der Zürcher Hochschule für Angewandte Wissenschaften erfolgreich entworfen, implementiert und validiert werden. Ein grosser Dank geht an die Pilotpartnerin St.Galler Kantonalbank, insbesondere an Herrn Andreas Barattiero, Leiter Finanzverarbeitung, sowie an die beteiligten Mitarbeitenden im Front- und Backoffice, die mit ihren Feedbacks das Verständnis der Forschenden und die Lösung wesentlich vorangebracht haben.

Management Summary

Formulare bestimmen die Kommunikation zwischen Frontoffice und Backoffice in Banken. Für die KuBe ist dies herausfordernd, da das Frontoffice mit dem Kunden in einer dynamischen und wenig formalisierten Interaktionsbeziehung steht, während das Back-Office durch seine Transaktionsorientierung und hochgradig standardisiert und stark formalisiert arbeitet. Die Transformationsleistung erbringt heute der KuBe, indem er die passenden Formulare identifiziert, ausfüllt und weiterleitet. Sehr einfache und komplexe Aufträge durchlaufen denselben formularbasierten Prozess. Design und Unterhalt der Formulare ist aufwendig, da mit jeder Variante einer Dienstleistung, jedem neuen Kanal und jeder neuen regulatorischen Anforderung die Formulare angepasst und erweitert werden müssen. Die Formulare werden umfassender, weisen Abhängigkeiten auf und machen die Auftragserteilung für die KuBe (KuBe) selbst bei simplen Aufträgen kompliziert.

An diesem Punkt setzt die im Rahmen dieses Forschungsprojektes entstandene prototypische Lösung auf. Mit ihr wird das Formular-Paradigma zwischen KuBe und Backoffice durchbrochen und die Grundvoraussetzung für weitere Digitalisierungsschritte geschaffen. An die Stelle der Formulare tritt eine Dialoganwendung, die den KuBe befähigt, in möglichst wenigen Dialogschritten, den relevanten Geschäftsfall auszuwählen, die minimal notwendigen Instruktionen zu hinterlegen und einen Auftrag auszulösen. Die validierten Aufträge werden an die korrekte Stelle im Backoffice disponiert und manuell mutiert. Für zwei pilotierte Auftragsstypen können die Aufträge mittels Robotic Process Automation (RPA) durchgängig automatisiert front-to-back abgewickelt werden.

Die prototypische Lösung wurde im Rahmen eines human-zentrischen Vorgehens vom Forschungspartnerin ZHAW mit dem Hauptumsetzungspartner Synpulse Schweiz AG und der Pilotumsetzungspartnerin St. Galler Kantonalbank in einem iterativen Prozess erarbeitet und getestet. Contextual inquiries mit Backoffice-Mitarbeitenden, KuBe und KuBe-Assistenten unterschiedlicher Profile lieferten die Basis für das Dialogdesign. Die untersuchten Geschäftsfälle konnten auf der Grundlage der mit den KuBe und Assistenten identifizierten Problemszenarien auf granulare Anwendungsfälle heruntergebrochen und prototypisch umgesetzt werden. Die aus den Contextual inquiries abgeleiteten übergeordneten Akzeptanzkriterien (Schnelligkeit, reduzierte Komplexität, Transparenz) und die darauf ausgerichteten Interaktionsmuster wurden bei den Testings mit KuBe und Assistenten validiert.

Die Herausforderung ein konfigurierbares und erweiterbares Produkt zu schaffen, stand im Fokus der zweiten Iteration. Während die erste Iteration auf die Identifikation geeigneter Interaktionsmuster ausgerichtet war, wurde der Prototyp im Rahmen der zweiten Iteration in eine für die Bank konfigurierbare Lösung verwandelt. Im Rahmen eines Enduser-Testings mit dem Backoffice gelang es, eine neue Variante eines bestehenden Anwendungsfalls abzubilden sowie einen neuen Anwendungsfall zu erstellen. Mit der Konfigurierbarkeit der Lösung wurde nicht nur die Grundlage für ein marktfähiges Produkt und somit die Realisierung des Business Cases des Hauptumsetzungspartners geschaffen, sondern auch ein Application Framework für semistrukturierte Dialoganwendungen. Das Framework ist auf andere Domänen übertragbar und ermöglicht Entwurf und Prototyping semistrukturierter Dialoganwendungen im Kontext angewandter Forschung & Entwicklung.

Die abschliessende Iteration konzentrierte sich auf den Proof of Value im Sinne einer durchgängigen Automatisierung. Auch wenn die Relevanz der Lösung für die Zielgruppe kleiner und mittlerer Banken bereits mit der intuitiven und schnellen Erstellung eines vollständigen, korrekt disponierbaren und manuell im Back Office ausführbaren Auftrags gegeben ist, stellt die automatisierte Ausführung des Auftrages mittels Robotic Process Automation (RPA) ein Zusatznutzen dar. Auch mit Blick auf eine mögliche Erweiterung der Lösung in Richtung der Endkunden, die so einfache Aufträge ohne Einbezug des KuBe auslösen könnten, ist diese Durchgängigkeit inklusive Echtzeitfeedback zum ausgeführten Auftrag interessant.

Inhaltsverzeichnis

Dank	3
Management Summary	4
Inhaltsverzeichnis	5
1 Einleitung	6
1.1 Ausgangslage und Motivation	6
1.2 Lösungsidee	7
1.3 Zielsetzung und angestrebter Nutzen	8
1.4 Begriffsklärung und Stand der Forschung	8
2 Forschungsdesign	11
2.1 Basis: Humanzentrische Gestaltung	11
2.2 Vorgehen	11
3 Analyse	13
3.1 Auswahl der Geschäftsfälle	13
3.2 Erhebung der Nutzerperspektive	14
4 Prototyping	16
4.1 Vorgehen	16
4.2 Zielgruppen und Testings	16
4.3 Technische und methodische Grundlagen	17
4.4 Funktionale Architektur	20
4.5 Prototyp I	33
4.6 Prototyp II	39
4.7 Prototyp III	39
4.8 Datenmodell	43
5 Fazit und Ausblick	47
5.1 Zusammenfassung der Ergebnisse	47
5.2 Nächste Schritte: Umsetzungsplanung	48
Literaturverzeichnis	51
Tabellenverzeichnis	53
Abbildungsverzeichnis	54

1 Einleitung

1.1 AUSGANGSLAGE UND MOTIVATION

«Von der Wiege bis zur Bahre: Formulare, Formulare». Dieser Ausruf des österreichischen Volksdichters Johann Nestroy aus dem 19. Jahrhundert gilt bis heute auch für das massiv digitalisierte Bankwesen. Egal ob in Papierform oder digital, Formulare bestimmen den Arbeitsalltag von Kundenberater*innen (KuBe) im Front-Office. Kundenanliegen müssen verstanden, der auszulösende Geschäftsfall erkannt, das richtige Formular gefunden, digital oder physisch ausgefüllt und zur Verarbeitung an die richtige Stelle im Backoffice weitergeleitet werden. Gelingt dies nicht, unvollständig oder fehlerhaft oder treten im Prozessverlauf Rückfragen auf, kommen unstrukturierte E-Mails, Telefonate oder persönliche Gespräche zum Einsatz. Banken haben sich mit dieser Situation arrangiert, Formulare sind Teil der Kultur und Optimierungs- und Digitalisierungsbemühungen konzentrieren sich bis heute auf die Formularbereitstellung und -verarbeitung, ohne das Konzept „Formular“ grundlegend zu hinterfragen.

Zur Illustration der Problematik ein Beispiel: Ein KuBe der St.Galler Kantonalbank ist im Gespräch mit einer Bankkundin, die sich über die Finanzierung der Renovation ihres Einfamilienhauses Gedanken macht. Auch die Option, einen Teil der Renovation, konkret den Anbau eines Wintergartens, mittels Vorbezug der 3. Säule-Gelder zu finanzieren taucht im Gesprächsverlauf auf. Im Idealfall war der KuBe vorinformiert und hat das Formular bereits im Dokumentenmanagementsystem gefunden und vorbereitet. Das komplexe Formular „Antrag Teil-/ Auflösung Konto Sparen 3, Verwendung für Wohneigentum“ umfasst 2 Seiten. Ebenfalls beigezogen werden muss eine Detailliste, die Auskunft gibt, ob eine Finanzierung der Renovation aus der 3. Säule möglich ist oder nicht. So ist zum Beispiel ein beheizter Wintergarten möglich, ein unbeheizter jedoch nicht.

Dieser Geschäftsfall ereignet sich bei der St.Galler Kantonalbank ca. 400-mal pro Jahr, das heisst durchschnittlich pro KuBe ein Mal. Formular und Detailliste zu finden, auszufüllen und weiterzuleiten, fällt den KuBe daher schwer und kostet Zeit. Unter Umständen sind Kundendialog oder Prozess schon fortgeschritten, bis man im ungünstigsten Fall erst bei der Verarbeitung im Backoffice bemerkt, dass der Antrag bestimmte Kriterien nicht erfüllt. Oder es werden weitere relevante Geschäftsfälle offensichtlich, die der KuBe in der Situation mit dem Kunden nicht erkannt und daher nicht ausgelöst hat.

Dieses Beispiel steht stellvertretend für eine Vielzahl schwach strukturierter Kundenanliegen, deren Vielfalt jedoch keine überzeugenden Business Cases für klassische Prozessautomatisierung oder die Entwicklung einzelner Schnittstellen und Anwendungen für Front-End-Eingaben im Kernbankensystem liefern. Im Fall der Sankt Galler Kantonalbank werden einfache und komplexe Aufträge mit mehr als 200 verschiedenen Formularen abgewickelt. Hinzu kommen Aufträge, die in unstrukturierten E-Mails oder telefonisch an das Backoffice gelangen.

Die Schwachpunkte dieses formularzentrischen Prozesses lassen sich wie folgt zusammenfassen:

- Unstrukturierte Kundenanliegen müssen vom KuBe in ausführbare Aufträge übersetzt werden. Diese Formulare folgen der Logik und Begrifflichkeit des Backoffices und nicht der Kundensituation.
- Der KuBe muss die Kundensituation und alle relevanten Informationen vollständig erfassen, um alle relevanten Geschäftsfälle erfolgreich auslösen zu können.
- Einfache und komplexe Aufträge durchlaufen denselben Prozess, das heisst unter Umständen muss auch für eine simple Anweisung ein komplexes und umfassendes Formular ausgefüllt werden.
- Das Auffinden des richtigen Formulars erfordert Zeit und Erfahrung, auch wenn die Formulare in einem Dokumentenmanagementsystem verwaltet werden und über das Kernbankensystem aufrufbar sind. Erschwert wird diese Suche durch unterschiedliches Fachvokabular der KuBe in Abhängigkeit davon, bei welchen Banken sie gelernt/gearbeitet haben.
- Kundenkontakt und Ausfüllen des Formulars können zeitversetzt erfolgen, so dass Rückfragen beim Kunden notwendig werden.

- Es werden unter Umständen unnötige Aufträge zu Geschäftsfällen ausgelöst, die im Back-End bereits proaktiv beziehungsweise automatisch erledigt wurden, weil Regelwerke den Sachverhalt in den Systemen korrekt abbilden.

Banken versuchen diese Defizite bei der formularbasierten Auftragserteilung mit diversen technologischen Mitteln seit langem zu adressieren. Den Stand der Praxis bei der Auftragserteilung in Schweizer Banken beschreiben Branchenkenner¹ mit folgenden Varianten:

- Formulare werden in Dokumentenmanagementsystemen oder einfachen Fileshare-Ablagen bereitgestellt. Häufig sind diese Dokumentenmanagementsysteme mit dem Kernbankensystem integriert, so dass die Formulare über die Oberfläche des Kernbankensystems aufgerufen werden können.
- Für Aufträge mit grossen Volumen kommen Workflow-Tools mit integrierter Dokumentenablage zum Einsatz (z.B. auf der Basis von Sharepoint), die die Standardfälle erfassen, aber nicht das Gros an Spezialfällen.
- Grosse Banken setzen auf Prozessautomatisierung, häufig auf der Grundlage von Business Process Management Plattformen, auf deren Grundlage spezifische Anwendungen für bestimmte formularzentrische Geschäftsfälle (z.B. für das Client Onboarding) mit implementierten Geschäftsregeln entwickelt werden. Diese Lösungen sind Gegenstand grosser Entwicklungs- und Systemintegrationsprojekte. Überzeugende Business Cases für derartige Lösungen ergeben sich nur für grossvolumige Geschäftsfälle.

Alle diese Varianten optimieren formularzentrische Prozesse, ohne jedoch das Konzept „Formular“ zu hinterfragen. Das Potenzial für Dialogsysteme wird primär für Endkundeninteraktion gesehen (UBS, 2016). Für das Anwendungsfeld des Projekts (Auftragserteilung Front-to-Back) kommen sie nicht zum Einsatz.

1.2 LÖSUNGSSIDEE

An diesem Punkt setzt die Lösungsidee dieses angewandten Forschungsprojekts auf. Mit ihr wird das Formular-Paradigma zwischen KuBe und Backoffice durchbrochen und die Grundvoraussetzung für weitere Digitalisierungsschritte geschaffen. An die Stelle der Formulare tritt ein trainierbarer digitaler Assistent, der den KuBe befähigt, in einem Dialog die Kundensituation zu erfassen, relevante Geschäftsfälle zu erkennen und auszuwählen und einen Auftrag auszulösen. Dieser wird automatisch disponiert, das heisst der Assistent entscheidet, welche Aufträge von welcher Backoffice-Funktion manuell bearbeitet werden und welche automatisiert mutiert werden können.

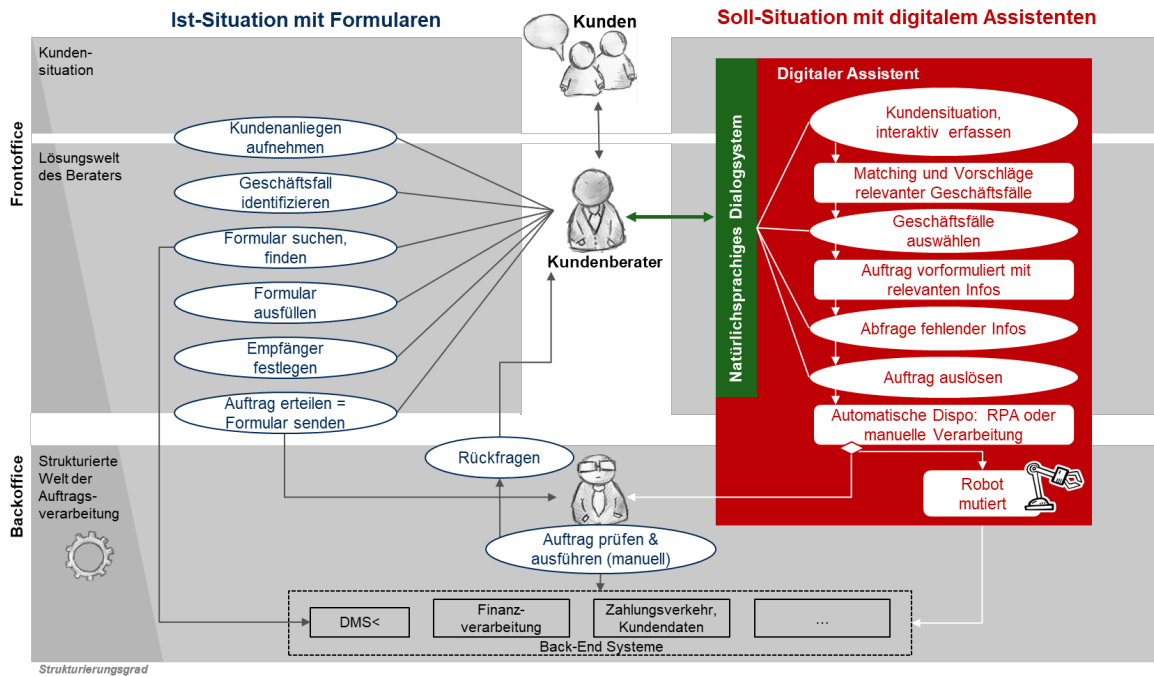
Für die durchgängige Ausführung der Aufträge wird die Lösung um eine Robotic Process Automation (RPA)-Komponente erweitert. RPA ist eine schnelle und kostengünstige Form der Prozessautomatisierung, da sie keine aufwendigen Schnittstellen oder Veränderungen der Back-End-Systeme erfordert, sondern auf der Benutzeroberfläche aufsetzt, indem sie Eingaben eines Benutzers emuliert. RPA wird aktuell primär für hochvolumige batchartige Routineaufgaben eingesetzt. Im Rahmen des Projekts soll das Potenzial von RPA genutzt werden, um ein breites Spektrum an Aufträgen mit geringem Aufwand durchgängig zu automatisieren.

Diese neuartige Kombination aus Dialoganwendung und Robotics an der Schnittstelle zwischen KuBe und Backoffice einzusetzen, ermöglicht aus der Perspektive der Banken eine Prozessinnovationen. Darüber hinaus ist der Assistent mit zunehmender Unterstützungsqualität und Technologiereife potenziell auf Endkunden ausdehnbar, so dass ein Innovationspotenzial für das gesamte System Kunde, KuBe und Backoffice-Verarbeitung gegeben ist.

¹ Die Experten des Hauptumsetzungspartners Synpulse und der Pilotumsetzungspartner Sankt Galler Kantonalbank kennen durch Projekte bzw. Netzwerkkontakte die Praxis in vielen Schweizer Banken aller Grössenklassen.

8 Vom Formular zum Dialog - Forschungsbericht

Abbildung 1 Ist- und Soll-Situation



1.3 ZIELSETZUNG UND ANGESTREBTER NUTZEN

Die Lösungsidee wird im Rahmen eines von der Schweizerischen Agentur für Innovationsförderung (Innosuisse) geförderten angewandten Forschungs- und Entwicklungsprojekts umgesetzt. Abbildung 2 fasst den Gegenstand, die Zielsetzung und das erwartete Ergebnis des Projekts zusammen.

Abbildung 2 Zielsetzung und Projektinhalte



1.4 BEGRIFFSKLÄRUNG UND STAND DER FORSCHUNG

Dialogsysteme

Ziel von Dialogsystemen ist, dass Menschen nicht mit einer Kommandosprache oder Eingabefeldern, sondern frei in einem gesprochenen oder geschriebenen Dialog mit einem Computer interagieren können (Berg, 2013; Chai et

al., 2002). Während Chatbots tendenziell als generelle Konversationspartner zu beliebigen Themen auftauchen (z.B. Apple Siri, Google Now), soll in diesem Projekt ein IT-gestützter Dialog ermöglicht werden, der in einer wohldefinierten Domäne stattfindet. Die Konfigurationskomponente des domänenunabhängigen Produkts ermöglicht die Anpassung auf eine konkrete Bank, das heisst die Spezialisierung auf ihre Geschäftsfälle und Kundensituationen, wodurch ein „restricted domain“-System mit hoher Dialogqualität entstehen kann (Sordoni et al., 2015; Ritter et al., 2011). Als Gradmesser gilt hierfür eine Recall-Rate von >90%, an der natürlichsprachige Schnittstellen im kommerziellen Bereich mehrheitlich scheitern (Pazos et al., 2013).

IT-gestützte Dialoge bestehen typischerweise aus 3 Elementen (Perez-Marin & Pascual-Nieto, 2011): Der Input-Analyser (1) nimmt eine Eingabe entgegen und erkennt darin eine semantisch identifizierte Information, welche er dem Dialog-Manager (2) übergibt. Dabei werden, je nach Ansatz auf unterschiedliche Weise, Wörter und Sätze segmentiert, benannte Entitäten erkannt und syntaktische sowie semantische Analysen durchgeführt. Der Dialog-Manager kennt den bisherigen Verlauf und aktuellen Stand des Dialogs und weiss, welche Informationen bisher bekannt sind und welche fehlen. Mit dieser Information wird der Answer Generator (3) angesteuert, der das nächste Element im Dialog präsentiert. Existierende Lösungen für IT-gestützte Dialoge variieren in der Umsetzung dieser Elemente sowie an den Schnittstellen zum Benutzer (Pazos et al., 2013). Während gesprochene Ein- und Ausgabe im beantragten Projekt nicht als Mehrwert erachtet werden, gibt es in der manuellen Eingabe und visuellen Ausgabe gleichermaßen verschiedene Typen angefangen bei simplen Phraseneingaben, über auswählbare Textelemente (z.B. systemgesteuerte Menüs, Entscheidungsbäumen, Tag Clouds), Frage-Antwort-Interaktionen in Form eines Chats bis hin zu mehr oder weniger intelligenten Dialogen (Berg & Düsterhöft, 2010; Pazos et al., 2013; Dodge et al., 2015). Allgemeingültige Dialogmodellierungs- und -implementierungsstrategien gibt es hierfür nicht (Berg, 2013). Die Kombination der effektivsten Dialogstrukturen ist Gegenstand der ersten Iteration des Forschungsprojekts (Prototyp I), ebenso wie die Ausgestaltung der domänenspezifischen Wissensmodule und deren Ausprägung (deterministisch vs. lernend). Die oben beschriebenen Konfigurationen dieser Elemente soll ohne programmatische Aufwände möglich sein, was eine Trennung der Spezifikation dieser Elemente und deren Umsetzung verlangt. Insbesondere sollen die Daten wie Frage- und Antworttexte unabhängig von der Logik (Algorithmik) spezifiziert und ausgetauscht werden können.

Robotic Process Automation (RPA)

Robotic Process Automation ist Teil der Lösungsarchitektur (vgl. Abbildung 1) und spielt für den Business Case des Innovationsprojektes eine Rolle, da diese Komponente die durchgängig automatisierte Verarbeitung von bestimmten Aufträgen ohne manuelle Eingriffe im Back-Office ermöglichen soll.

Robotik Process Automation ist ein Technologiekonzept, das seit etwa 2010 entstanden ist und dessen Ursprünge im Bereich der Makros und insbesondere des Screen Scrapings, das heisst dem automatisierten Auslesen von Computerbildschirmhalten liegen (IEEE Std. 2755-2017, 2017). RPA bedeutet, dass spezifische Technologien und Methoden zur Anwendung kommen, die auf Software und Algorithmen basieren, mit dem Ziel, fehleranfällige menschliche Aufgaben mit repetitivem Charakter zu automatisieren (Ivančić et al., 2019). RPA-Tools übernehmen die Datenübertragung zwischen mehreren Informationssystemen, indem sie einem konfigurierten Workflow folgend mit diesen über deren Graphical User Interfaces (GUI) interagieren (Scheppeler & Weber, 2020).

Zum Zeitpunkt der Projektstarts 2017 generierte RPA zunehmend Interesse und es existieren bereits kommerziell verfügbare Produkte (Le Clair et al., 2017). Forschungsseitig fand RPA jedoch kaum statt. Das Potenzial für die Automatisierung stark strukturierter Prozesse wurde zum damaligen Zeitpunkt nur in wenigen Fallstudien nachgewiesen (Allweyer, 2016; Lacity et al., 2015; Willcocks et al., 2015). Seither wuchs sowohl das Interesse der Wirtschaft als auch der Forschung für RPA. 2018 erschienen rund viermal so viele wissenschaftliche Publikationen zum Thema RPA wie 2017 (Ivančić et al., 2019). 2018 wuchs der weltweite Markt für RPA-Software um 63% (Gartner Research, 2019). RPAs können mächtige Business Process Management Solutions (BPMS) nicht ersetzen, gewinnen aber für begrenzte Anwendungszwecke zunehmend an Attraktivität.

Scheppeler & Weber (2020) untersuchen die Nutzenversprechen und Risiken des Einsatzes von RPA, welche in Tabelle 1 zusammengefasst sind.

10 Vom Formular zum Dialog - Forschungsbericht

Tabelle 1 Nutzenversprechen und Risiken des RPA Einsatzes in Anlehnung an (Scheppeler & Weber, 2020)

Nutzenversprechen von RPA	Risiken von RPA
Effizienzsteigerungen durch: <ul style="list-style-type: none">- Kostenersparnisse- Höhere Mitarbeitendenzufriedenheit und Wertschöpfung, da Mitarbeitende höherqualifizierte Aufgaben übernehmen können	Angst vor Arbeitsplatzverlust / Personalabbau Kosten für Umschulung und Weiterbildung
Geringere Abhängigkeit der Fachabteilungen von der IT-Abteilung, da RPAs konfigurierbar sind und geringe Programmierkenntnisse erforderlich	Gefahr wachsender Komplexität der IT-Architektur Dezentrale Compliance- und Sicherheitsrisiken
Hohe, gleichbleibende Qualität, damit bessere Einhaltung von Compliance-Anforderungen	Aufbau und Betrieb einer kontrollierten RPA Infrastruktur bedarf Ressourcen der IT.
Reduktion von Durchlaufzeiten	Investitionen in die Ausbildung der Fachabteilungen für die Konfiguration von RPA Bots.
Kostengünstig im Vergleich zu BPMS aufgrund der nicht-invasiven Integration der beteiligten Systeme über deren UIs. Keine Veränderung an vorhandenen Systemen / keine technischen Schnittstellen notwendig	Potentiale für grundlegende organisatorische Prozessoptimierungen wie bei BPMS-Projekten werden bei RPA-Projekten mit opportunistischem Charakter in der Regel nicht genutzt

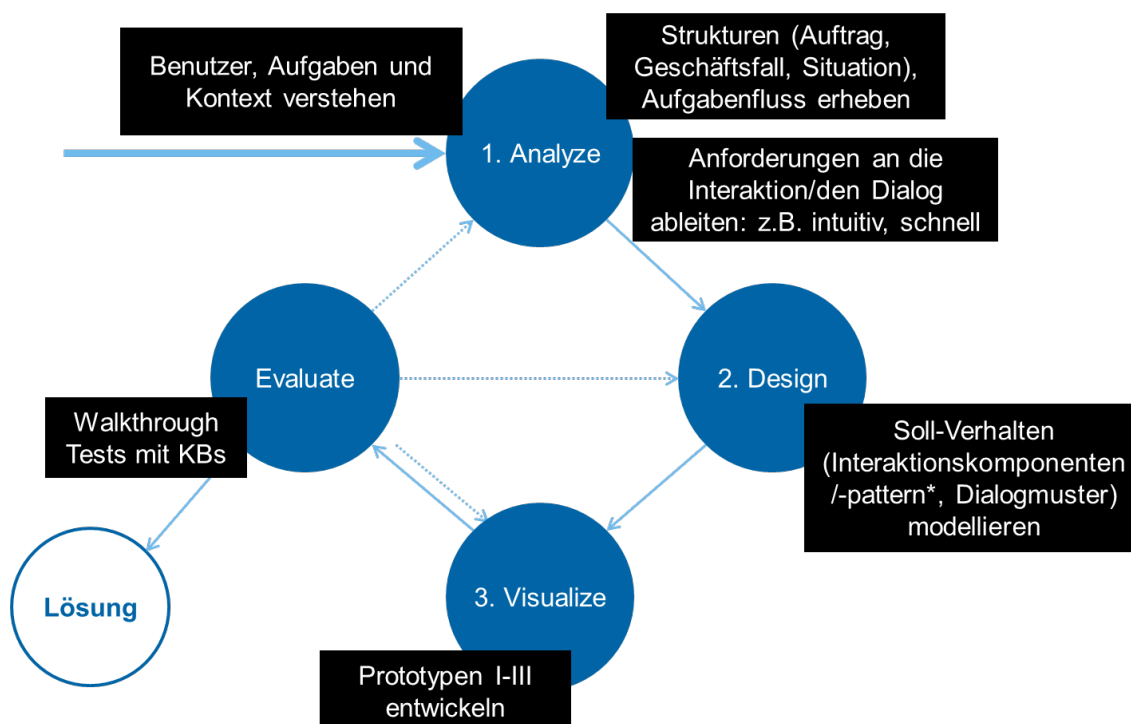
2 Forschungsdesign

2.1 BASIS: HUMANZENTRISCHE GESTALTUNG

Das als Projektziel formulierte Dialogsystem soll in Gestalt eines digitalen Assistenten die Arbeitsweise der Bank-KuBe bei der Auftragserteilung grundlegend verändern. Um sicherzustellen, dass die damit angestrebte Innovation wirksam und erfolgreich stattfinden kann, werden die KuBe, als primäre Nutzer des Systems, mit ihren Aufgaben, Zielen und Eigenschaften in den Mittelpunkt des Entwicklungsprozesses gestellt. Eine wesentliche Forschungsherausforderung liegt darin, in einem iterativen Designprozess herauszufinden, welche Interaktionsformen mit dem digitalen Assistenten für welche Aufgaben und Ziele der KuBe geeignet sind. Das Projekt folgt dementsprechend einem humanzentrischen Vorgehen, das in Anlehnung an den ISO-Standard 9241-210 folgende Charakteristiken aufweist (DIN-ISO, 2011; Giacomini, 2014):

- Die Gestaltung basiert auf einem umfassenden Verständnis der Nutzer, der Aufgaben und des Arbeitskontexts.
- Nutzer werden frühzeitig in die Gestaltung und Entwicklung einbezogen.
- Designergebnissen werden fortlaufend, iterativ und auf der Basis von Evaluationen durch die Nutzer verfeinert und angepasst.

Abbildung 3: Humanzentrisches Vorgehen: Elemente und ihre Anwendung im Projekt



2.2 VORGEHEN

Der Projektplan setzt das humanzentrische Vorgehen um. Entwickelt werden sollte in 4 Phasen ein Prototyp (I-IV), dessen Funktionalität sukzessive erweitert und mit einer wachsenden Zahl von User Stories mit den KuBe des Pilotumsetzungspartners (St.Galler Kantonalbank) im Rahmen des Forschungsprojekts evaluiert werden sollte. Als Grundlage für die Projektplanung diente die hypothetische Lösungsarchitektur (siehe Kapitel 1.2). Abbildung 4 stellt

den Bezug zu den funktionalen Elementen der Lösungsarchitektur her und fasst die wesentlichen Inhalte und Ziele für die vier geplanten Iterationen des Prototypen zusammen. Da im Projektverlauf die Erweiterbarkeit und Trainierbarkeit bereits ab Prototyp II sukzessive in die Lösung integriert wurde, reduzierte sich die Anzahl der Iterationen und daraus resultierenden Prototypen auf drei statt vier. Aus den 5 Geschäftsfällen kristallisierten sich im Rahmen der contextual inquiries (Kapitel 3.2) 10 granularere User Stories heraus, die in Form von Kundenanliegen im operativen Alltag der KuBe effektiv stattfinden, so dass bereits im Prototyp I mit 10 praktischen User Stories statt 5 theoretischen Geschäftsfällen gearbeitet wurde (siehe Kapitel 4 Prototyping)

Abbildung 4: Geplantes Vorgehen: Inhalte und Ziele der Prototyper

<p>1. Prototyp I (Juni 2018) umfasst die Komponenten A, B und wird mit mind. 5 Geschäftsfällen u. mind. 2 Kundenberatern für mind. 2 alternative Dialogstrukturen getestet. Effektivste Dialogstrukturen für die Erfassung der Situation und die Auswahl der Geschäftsfälle sind festgelegt.</p>	<p style="writing-mode: vertical-rl; transform: rotate(180deg);">Natürlichsprachiges Dialogsystem</p>	<p>A Matching Situation - Geschäftsfall</p>
<p>2. Prototyp II (September 2018) umfasst die Komponenten A, B, C und wird mit min. 15 Geschäftsfällen u. mind. 2 Kundenberatern getestet =Proof-of-Use/Value (Voraussetzung für die Weiterführung des Projekts): Für mind. 60% der Fälle wird ein relevanter Geschäftsfall identifiziert, ein Auftrag vollständig formuliert und korrekt disponiert. Mind. 1 Kandidat für die Mutation via Robot ist identifiziert.</p>		<p>+</p> <p>C Auftragsformulierung und -erteilung</p>
<p>3. Prototyp III (Dezember 2018) umfasst die Komponenten A, B, C, D und wird mit mind. 15 Geschäftsfällen getestet. Für mehr als 60% der getesteten Fälle wird ein relevanter Geschäftsfall identifiziert und ein Auftrag vollständig formuliert und korrekt disponiert. Für mind. 1 Geschäftsfalltyp wird ein Auftrag durchgängig via RPA ausgeführt.</p>		<p>+</p> <p>D Dispo und Integration RPA</p>
<p>4. Finaler Prototyp IV (Januar 2019) umfasst alle Komponenten inkl. der Konfigurations- und Trainingskomponente E und wird mit mind. 16 Geschäftsfällen getestet. Für > 90% der Fälle wird ein relevanter Geschäftsfall identifiziert und ein Auftrag vollständig formuliert und korrekt disponiert. Für mind. 1 Geschäftsfalltyp wird ein Auftrag durchgängig mit RPA-Robot ausgeführt. Die Anpassbarkeit des Produkts ist für mind. 1 weiteren Geschäftsfall validiert.</p>		<p>+</p> <p>E Erweiterbarkeit, Trainierbarkeit</p>

3 Analyse

Ziel der Analysephase des humanzentrischen Vorgehens war, die Perspektive der KuBe im Front-Office von Beginn an in das Lösungsdesign einzubeziehen, aktuelle Arbeitsmuster bei der Auftragserteilung und Erwartungen an eine dialogorientierte Lösung zu erheben. Aber auch die Perspektive der Back-Office Mitarbeitenden, die die Aufträge entgegennehmen und typische Fehler und Probleme besser kennen als jeder andere, wurde erfasst.

3.1 AUSWAHL DER GESCHÄFTSFÄLLE

Im Rahmen des Kickoffs wurden mit dem Wirtschaftspartner Synpulse und der Pilotumsetzungsbank St.Galler Kantonalbank (SGKB) relevante Geschäftsfälle für das Prototyping identifiziert und nach folgenden Kriterien evaluiert:

Vielfalt der Geschäftsfälle, d.h. möglichst breites Spektrum in Bezug auf:

- Kundensituationen, die diese Geschäftsfälle auslösen
- Auftragsherkunft (Formular, E-Mail, Telefon ...)
- Komplexität
- Volumen
- Ausführende Systeme
- Integration Umsysteme (für die Situationserfassung & Auftragsformulierung)

Vollständiges Automatisierungspotenzial:

- Mindestens drei Geschäftsfälle mit vermutetem Potenzial für eine durchgängige Automatisierung mittels RPA.

Hohe Relevanz des KuBe

- Geschäftsfälle, bei denen die Auftragsqualität in hohem Masse KB-abhängig ist.

Attraktivität für die Bank und die KuBe:

- Geschäftsfälle, die aus Sicht der Pilotumsetzungsbank als Showcase prädestiniert sind, weil sie im Front- und/oder Backoffice bei der Auftragsverarbeitung mit grossen Aufwänden, Fehlerquoten und Frustrationen verbunden sind.

Auf der Grundlage dieser Kriterien und einer statistischen Auswertung der Aufträge in einem Team der Backoffice Operations wurden 5 grosse Geschäftsfälle selektiert, für die im Projektverlauf weitere Varianten identifiziert und Spezialisierungen in Form von User Stories formuliert wurden. Abbildung 5 listet die 5 Geschäftsfälle auf und die Anzahl Formulare, die für diesen Geschäftsfall existieren. Die Angaben, wie häufig diese Geschäftsfälle das Front-Office beschäftigen, stammen aus den contextual inquiries mit den 8 KuBe und Assistenzen von KuBe. Versandinstruktionen und Depotstammmutationen beschäftigen demnach alle Profile am häufigsten. Die Teilauflösung der Säule 3a ist wie zu erwarten für Privatkunden-KuBe relevanter.

Abbildung 5: Die projektrelevanten Geschäftsfälle und ihre Häufigkeit

Geschäftsfall	Forms	Interview-Partner Front-Office & Geschäftsfallhäufigkeit								
		P.Z. KuBe Privatkunden	L.P. Assistent. Privatkunden	F.E. Assistent. Privatkunden Ausenstelle	T.H. KuBe Private Banking	R.G. Assistent. Private Banking	P.G. Assistent. Private Banking	P.K. Assistent. eV Desk	A.S. Assistent. eV Desk	
1 Mutation Versandinstruktionen	4 Formulare	3	3	2	1	1	3	3	4	
2 (Teil)auflösung Konto Sparen 3	2 Formulare	3	3	2	1	1	1	1	1	
3 Depotstamm-Mutationen	1 Formular	4	2	3	2	3	4	3	4	
4 Mutation Todesfall (aka Selig-Prozess)	2 Formulare	3	2	2	2	4	2	1	1	
5 Zahlungsverkehr Recherchen	1 Formular	1	0	2	0	1	1	2	1	

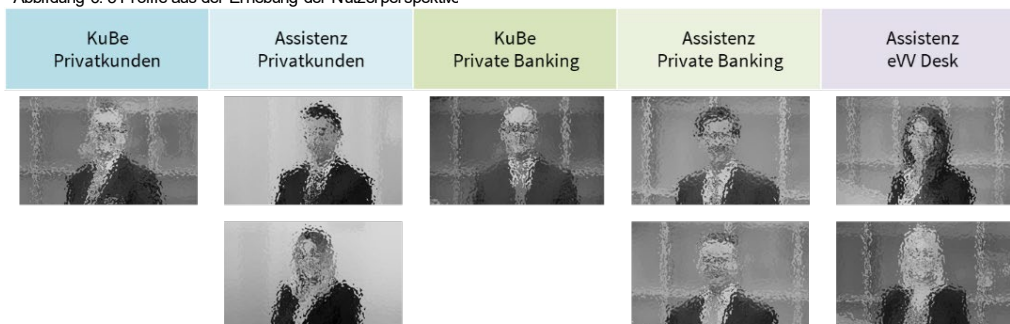
Wie häufig hast Du mit den Geschäftsfällen zu tun?
 0=nie, 1=selten, 2=mehrmals im Monat/regelmässig;
 3=täglich/häufig, 4=mehrmals täglich/sehr häufig
 (erhoben zum Auftakt der Contextual inquiries)

3.2 ERHEBUNG DER NUTZERPERSPEKTIVE

Wie in Kapitel 2.1 beschrieben basierte das Projekt auf den Prinzipien der humanzentrischen Systemgestaltung. Die Perspektive der Endnutzer wurde initial im Rahmen von contextual inquiries und fortlaufend im Rahmen des Prototyping (siehe Kapitel 4) erhoben.

Contextual inquiries stammen aus dem Methodenkreis des humanzentrischen Designs. Nutzer aus der Zielgruppe eines Systems werden bei der Ausführung ihrer Aufgaben in ihrer Arbeitsumgebung und mit den bestehenden Werkzeugen und Systemen begleitet und dabei situativ befragt, womit, wie und warum bestimmte Arbeitsschritte stattfinden (Holtzblatt & Beyer, 2016). Contextual inquiries fanden in der St.Galler Kantonalbank im Front-Office aber auch mit 4 Kadermitarbeitenden der Back-Office Operations statt, die die Perspektive der Auftragsverarbeitung für die 5 identifizierten Geschäftsfälle einbrachten. Obwohl die KuBe und deren Assistenzen die primären Endnutzer des Dialogsystems darstellen, war die Perspektive der Auftragsverarbeitung wesentlich, da die Mitarbeitenden im Backoffice ein sehr viel umfassenderes Bild der typischen Fehler und Probleme mit Aufträgen, die von den KuBe zu ihnen gelangen, haben. Die 10 Contextual inquiries mit KuBe und deren Assistenzen führte zu 5 Profilen (Abbildung 6). 8 Front-Office-Mitarbeitende aus diesen 5 Profilen und 1 Mitarbeitender aus dem Back-Office begleiteten als Testuser das Prototyping, das im nachfolgenden Kapitel beschreiben wird.

Abbildung 6: 5 Profile aus der Erhebung der Nutzerperspektive

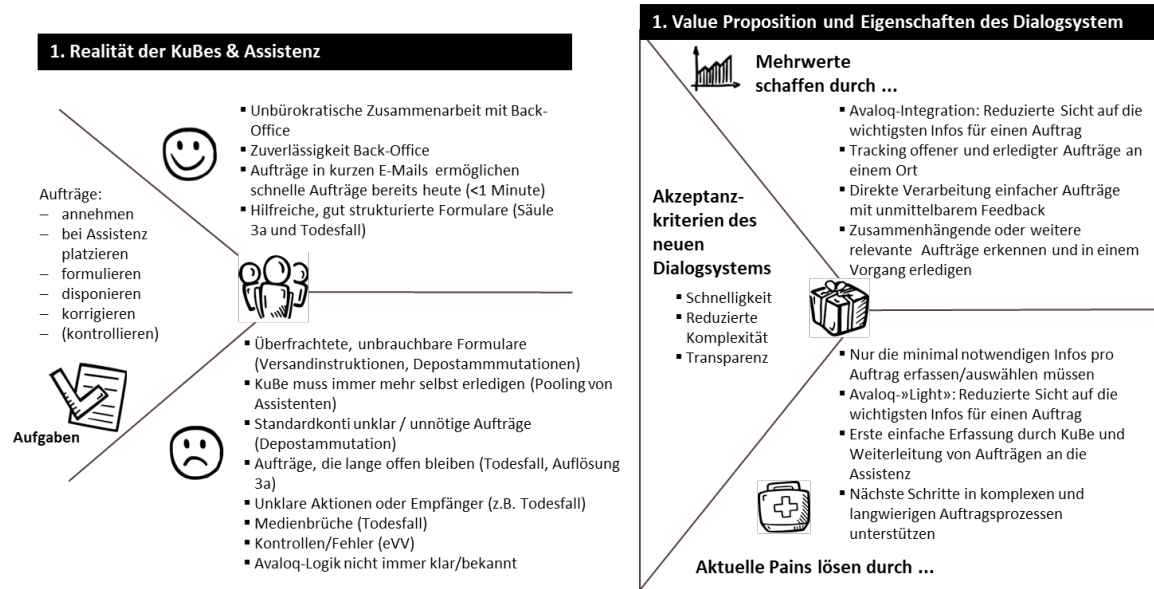


+ Contextual Inquiries mit 4 Back-office Operations Kadermitarbeitenden = Perspektive «Auftragsverarbeitung»

Der Customer Value Proposition Canvas in Abbildung 7 (in Anlehnung an die Notation von Osterwalder / (Strategizer, o. J.) fasst die Erkenntnisse aus den contextual inquiries mit den involvierten Mitarbeitenden der St.Galler Kantonalbank im Back- und Front-Office zusammen und formuliert das Nutzenversprechen für die neue Lösung.

Abbildung 7: Customer Value Proposition Canvas für das Dialogsystem

Essenz: Die KuBe und Assistenzen haben hohe Ansprüche an die Schnelligkeit. Mehrwerte können nur über reduzierte Komplexität, Transparenz und intelligente Unterstützung erreicht werden.



4 Prototyping

4.1 VORGEHEN

Im Rahmen des Projektes wurde ein Prototyp entwickelt und dessen Funktionalität iterativ validiert und erweitert. Der Prototyp wurde im Projektkontext und in diesem Forschungsbericht mit dem Arbeitstitel „OrderButler“ bezeichnet, da er für die Zielgruppe im Front-Office einer Bank, die KuBe und deren Assistenzen, als hilfsbereiter, unauffälliger «Butler» dient, mit dem Kundenanliegen in Aufträge für das Back-Office erfasst werden können.

Ursprünglich waren vier Iterationen für den Prototypen geplant (Abbildung 4, Kapitel 2.2.). Die Funktionalitäten von Prototyp IV, konkret die Erweiterbarkeit und Trainierbarkeit, wurden jedoch bereits mit Beginn der Arbeiten am Prototypen II entwickelt und in der dritten Iteration weiterentwickelt, so dass die vierte Iteration nicht mehr notwendig war (Abbildung 8).

Abbildung 8 Angepasstes Vorgehen für das Prototyping auf der Basis von User Stories

- Prototyp I (Juni 2018)** umfasst die Komponenten A, B, C und wird mit mind. 3 User Stories mit mind. 6 Kundenberatern mit hybriden Dialogstrukturen getestet. Effektivste Dialogstrukturen für die die Auswahl der Geschäftsfälle und Formulierung des Auftrages sind festgelegt.
- Prototyp II (September 2018)** umfasst die Komponenten A, B, C, E und wird mit mit ca. 10 User Stories mit mind. 6 Kundenberatern und mind. 1 Backoffice-Mitarbeitenden getestet =Proof-of-Use/Value (Voraussetzung für die Weiterführung des Projekts): Für mind. 60% der Fälle wird ein ausführbarer Auftrag vollständig formuliert und korrekt disponiert. Mind. 1 Kandidat für die Mutation via Robot ist identifiziert.
- Prototyp III (Januar 2019)** umfasst die Komponenten A, B, C, D, E und wird mit ca. 10 User Stories mit mind.3 Kundenberatern und mind. 1 Backoffice-Mitarbeitenden. Für mehr als 60% der getesteten Fälle ein ausführbarer Auftrag vollständig formuliert und korrekt disponiert. Für mind. 1 Geschäftsfalltyp wird ein Auftrag durchgängig via RPA ausgeführt.



Für jeden Prototypen wurde bei der Planung des Projektes eine angestrebte Anzahl von Geschäftsfällen definiert. Bereits mit **Prototyp I** stellte sich aber heraus, dass die Abgrenzung eines Geschäftsfalles schwierig ist. Als Beispiel möge die Auszahlung aus der dritten Säule (Sparen 3) dienen, welche beispielweise bei Pensionierung aber auch beim Erwerb oder der Renovation von Wohneigentum stattfinden kann. Handelt es sich nun um einen oder zwei Geschäftsfälle? Es wurde daher entschieden, statt eine grosse Anzahl von Geschäftsfällen nur oberflächlich abzubilden, mit granularen User Stories zu arbeiten, welche möglichst viele Varianten von Geschäftsfällen aus dem Alltag der KuBe abdecken. Zudem wurde die Realisierung der Erweiterbarkeit (ursprünglich vorgesehen für **Prototyp IV**) bereits mit **Prototyp II** in einer ersten Version getestet, so dass weitere Geschäftsfälle auch bereits ohne Programmierung erstellt werden konnten.

4.2 ZIELGRUPPEN UND TESTINGS

4.2.1 Zielgruppen

Die primäre Zielgruppe des OrderButlers sind die KuBe einer Bank. Diese sollen mit dem OrderButler Aufträge erfassen können, welche bis anhin auf Papier oder über ein einfaches elektronisches Formular erfasst werden. Eine weitere Zielgruppe sind die Mitarbeitenden im Backoffice, welche die Aufträge (sofern nicht automatisch ausführbar) abarbeiten aber auch bestehende Geschäftsfälle anpassen sowie neue Geschäftsfälle erstellen können. Eine mögliche zukünftige Zielgruppe des OrderButler sind Endkunden der Bank, die einfache Aufträge über das Online-Banking oder eine andere Kundenschnittstelle direkt mithilfe des intuitiven OrderButlers formulieren könnten. Diese Zielgruppe wurde im Projektkontext noch nicht berücksichtigt.





Dies Ausrichtung auf KuBe und deren Assistenzen bedeutet, dass der OrderButler nach kurzer Schulung intuitiv sowohl von regelmässigen als auch von gelegentlichen Nutzern im Front-Office bedienbar sein soll. Wie im Value Proposition Canvas in Kapitel 3.2 formuliert (Abbildung 7) liegt der Fokus bei der Gestaltung des OrderButlers auf den identifizierten Akzeptanzkriterien aus Nutzersicht:

- Schnelligkeit bei der Erfassung und Weiterleitung des Auftrages, um den KuBe einen Mehrwert gegenüber unstrukturierten E-Mails und der aktuellen Formularpraxis zu bieten.
- Reduzierte Sicht auf ausschliesslich auftragsrelevante Informationen, um die Komplexität des Kernbankensystems zu verbergen.
- Transparenz, um den Status eines Auftrages nachvollziehen zu können.

4.2.2 Testings auf der Grundlage von Szenarien

Die Erprobung der drei Iterationen des Prototypen mit den KuBe fanden in Form von szenariobasierten Usability-Tests statt (Möller, 2017). Dabei wurden den KuBe typische Kundenanliegen in Form kleiner Szenarien vorgelegt, die sie mithilfe des Prototypen in Aufträge formulieren sollten (Beispiele in Abbildung 9). Die Bildschirmbewegungen der Sessions und die Sprache der Probanden wurden aufgezeichnet und die KuBe wurden gebeten, ihre Überlegungen während des Testlaufs laut auszusprechen (think-aloud Methode) (Someren et al., 1994). Direkt im Anschluss an die Testläufe wurden die KuBe einzeln befragt und die Feedbacks in einem Feedback-Grid dokumentiert. Die Sitzungsaufzeichnungen und die Feedbacks wurden gesamthaft ausgewertet und in Form von Entwicklungsaufgaben (issues) für die nächste Iteration formuliert und priorisiert (4.3.4). Die Anzahl der Szenarien nahm von Iteration zu Iteration zu, um ein breiteres Spektrum an Funktionalität und die Zielsetzungen pro Iterationen validieren zu können. Die vordefinierten Zielwerte für erfolgreiche Testläufe (Abbildung 8) stellten dabei kein Problem dar, da es allen Probanden gelang, die Aufgabenstellungen mithilfe des Prototypen erfolgreich zu erledigen.

Abbildung 9: Beispiele von Usability-Testszenarien

<p>Szenario #1 Vermögensausweis</p> <p>Eine Frau Sterk (oder Stark?) hat sich telefonisch gemeldet. Sie möchte gerne den Vermögensausweis häufiger erhalten, am besten alle 3 Monate. Es geht um das Depot 1000.0001.1000. Es wird bereits ein Vermögensausweis versendet, aber nicht so häufig wie gewünscht.</p> 	<p>Szenario #2 Steuerausweis</p> <p>Sie erhalten eine E-Mail von Herrn Eddard Stark. Er möchte zusätzlich einen Steuerausweis an seine Adresse in St. Gallen (Teufenerstrasse 26) erhalten. Auf Nachfrage erklärt er, in Zukunft den Steuerausweis an seine Adresse in Bern und in St. Gallen erhalten zu wollen.</p> 
<p>Szenario #3 Verbindungskonti</p> <p>Herr Eddard Stark ruft an. Er hat in seinem Depot eine Wertschrift in USD. Diese Erträge fließen zur Zeit auf ein CHF-Konto. Er möchte diese in Zukunft auf sein bestehendes USD-Konto fließen lassen. Da er gerade unterwegs ist, kann er leider weder seine Depot-Nr. noch die Nummer des USD-Kontos nennen. Da er aber nur ein Depot hat, sollte es klar sein um welches es geht.</p> 	<p>Szenario #4 Sparen 3</p> <p>Herr Eddard Stark wird pensioniert da er das AHV-Alter erreicht hat. Er möchte sich alles Geld aus seinem Sparen 3 Konto 1000.0001.2001 auszahlen lassen. Das Geld soll auf das Konto 1000.0001.2002 ausbezahlt werden. Die Wertschriften sollten so bald wie möglich verkauft werden und das Sparen 3 Konto baldmöglichst saldiert werden.</p> 

4.3 TECHNISCHE UND METHODISCHE GRUNDLAGEN

In diesem Kapitel wird die Konzeption und die Entwicklung des OrderButlers aus der Perspektive des Requirements Engineerings und des Software Engineerings im Kontext des Prototypings beschrieben. Zunächst wird ein Überblick die verwendeten Endgeräte gegeben, es wird eine Systemarchitektur vorgeschlagen und begründet und in einem weiteren Schritt werden die für das Prototyping eingesetzten Technologien und agilen Methoden vorgestellt.

4.3.1 Endgeräte

Um eine potentielle spätere Integration in den Avaloq Workspace zu ermöglichen, wurde mit dem Umsetzungspartner Synpulse entschieden, keine Desktop-Anwendung, sondern eine Webanwendung zu erstellen. Installation

und Wartung beim Endbenutzer fallen damit weg, es ist nur eine Internet-Verbindung sowie ein zeitgemässer Webbrowser erforderlich.

Die Webanwendung wird primär auf Tablets und Desktop-PC optimiert, da zur Erfassung von Text und Zahlenwerten eine echte Tastatur sinnvoll erscheint. Auf kleineren Bildschirmen wie Mobiltelefonen soll die Software lauffähig, aber nicht speziell für solche Endgeräte optimiert sein.

4.3.2 Systemarchitektur²

Die Webanwendung basiert auf einer klassischen Client-/Server-Lösung, bei der ein Webbrowser den Client darstellt. Die Daten werden über das http bzw. https-Protokoll übertragen und im Browser mittels HTML und Javascript dargestellt.

Serverseitig gibt es verschiedene Technologien, die die Anfragen des Clients verarbeiten und Antworten zurücksenden. Allen Technologien gemeinsam ist die Notwendigkeit, die Daten persistent zu speichern. Dazu kommen verschiedene Datenbanktypen in Frage. Für den OrderButler wird eine relationale Datenbank verwendet, da es sich beispielsweise bei der Konfiguration von Geschäftsfällen, Interaktionsschritten sowie Interaktionselementen um klassische relationale Daten handelt und der Einsatz eines anderen Datenbanktyps (NoSQL, Key/Value) wenig sinnvoll erscheint.

4.3.3 Eingesetzte Technologien

Bei allen eingesetzten Technologien und Software-Bibliotheken stand die Wahl einer Open-Source Lösung im Vordergrund, um das Projektbudget zugunsten der Entwicklungsleistung zu entlasten. Im Rahmen einer anschliessenden Produktentwicklung kann von diesem Grundsatz abgewichen werden.

Die Tabelle 2 gibt einen Überblick über die wichtigsten Technologien und Software-Bibliotheken, auf welchen der OrderButler aufbaut:

Tabelle 2 Verwendete Technologien

BEREICH	TECHNOLOGIE	VERWENDUNG	LIZENZ
Client	HTML JavaScript TypeScript Angular Path Framework	GUI GUI GUI GUI GUI	- - Apache 2 License MIT License LGPL
Server	Java Spark Guice Gson Lucene Deeplearning4J	Server REST-Services Dependency Injection JSON Volltextsuche Machine Learning	GNU General Public License Apache 2 License Apache 2 License Apache 2 License Apache 2 License Apache 2 License
Datenbank	JPA Hibernate Postgresql, H2	Datenbank-Verbindung Datenbank-Verbindung Datenbank	GNU General Public License LGPL Diverse

4.3.3.1 Client

Das Frontend des OrderButlers basiert auf HTML und JavaScript. Als Basis wird das von der ZHAW entwickelte Path Framework verwendet. Path selbst verwendet Angular zur Darstellung von Webseiten. Angular bietet folgende Vorteile:

- grosse Community und weite Verbreitung
- wird durch eine grosse Firma (Google) unterstützt und weiterentwickelt
- qualitativ hochstehende Entwicklung in TypeScript möglich (inklusive Kompilierung nach JavaScript)

² Hinweis: Die St.Galler Kantonalbank fungierte lediglich als Pilotumsetzungspartnerin für den Prototypen. Im Rahmen des Forschungsprojekts fanden jedoch in keiner Weise Integrationen mit oder Zugriffe auf Systeme der St.Galler Kantonalbank statt. Die verschiedenen Iterationen des Prototypen und der notwendigen Demodatenbanken wurden ausschliesslich auf Seiten der Wirtschaftspartnerin Synpulse und der Forschungspartnerin ZHAW erstellt und betrieben.

Bereits der erste Prototyp des OrderButlers wurde mit Path und Angular entwickelt und danach wurde das Framework aufgrund der guten Erfahrungen beibehalten.

4.3.3.2 Server

Java

Im professionellen Serverbereich konkurrenzieren hauptsächlich Java, Node sowie .NET um die Vorherrschaft. Für die Entwicklung des OrderButlers wurde Java gewählt, hauptsächlich um die Java-Erfahrung auf Seiten des Forschungspartners und des Umsetzungspartners auszunutzen. Zudem ist die Kernbankensoftware Avaloq zu einem grossen Teil aus Java aufgebaut, so dass sich bei einer späteren Integration des OrderButlers in Avaloq Vorteile ergeben.

Guice

Auf eine klassische JEE (Java Enterprise) Implementation wurde verzichtet und stattdessen das kompaktere Google Guice verwendet. Dieses bietet ebenfalls die benötigten Services wie Dependency Injection an, aber ohne die erhöhte Komplexität von JEE.

Spark und Gson

Für die REST-Services wird das Spark-Framework verwendet. Damit können sehr einfach REST-Services definiert und in Java entwickelt werden. Für die Konvertierung der Java-Objekte in das JSON-Format und zurück wird die Gson-Bibliothek von Google verwendet.

Lucene

Bei den ersten Tests wurde festgestellt, dass die Suche und das Auffinden von Geschäftspartnern, Personen und weiteren Business-Objekten ein zentraler und wichtiger Teil des OrderButlers ist. Um die Suche zu optimieren und für eine maximale Performance wird eine Volltextsuche angeboten, welche mit der Programm-Bibliothek Lucene implementiert ist.

Deeplearning4J

Der OrderButler unterstützt die KuBe beim Erstellen von Aufträgen. Ein Auftrag wird aufgrund eines Geschäftsfalles erstellt. Die Auswahl dieses Geschäftsfalles gestaltet sich nicht einfach, da im Endausbau sehr viele Geschäftsfälle möglich sein werden, und die Liste auch durch das Backoffice jederzeit erweitert werden kann.

Um die Auswahl eines Geschäftsfalles zu vereinfachen, wurde mit Deeplearning4J exemplarisch ein Machine Learning Algorithmus eingesetzt, um dem Benutzer nach Auswahl eines Geschäftspartners passende Geschäftsfälle vorzuschlagen.

4.3.3.3 Datenbank

Als Schnittstelle zwischen Java und der Datenbank wurde JPA (Java Persistence API) mit der Implementierung Hibernate gewählt. Einerseits wird die Programmierung der Datenbankabfragen stark vereinfacht, andererseits kann die Datenbank mit kleinem Aufwand ausgetauscht werden. So wurde für den Entwicklungstest die kompakte Datenbank H2, für die Funktions- und Integrationstests aber eine leistungsfähigere Postgres-Datenbank verwendet. Beim Einsatz mit Avaloq kann beispielsweise auch eine Oracle-Datenbank verwendet werden.

4.3.4 Methoden und Werkzeuge

Für das Requirements Engineering, Software Engineering und Testing wurde eine agile Vorgehensweise gewählt. Für dieses kompakte und humanzentrische Softwareprojekt mit wenigen, direkt involvierten Partnern erwies sich ein agiles Vorgehen als sehr effizient und effektiv.

Das GUI wurde nach einer kurzen Mockup-Phase direkt in Path, HTML und Angular entwickelt und für Feedbacks passwortgeschützt online gestellt. In mehreren Runden wurde das GUI danach verfeinert. Bereits vor Definition und Fertigstellung des gesamten GUI konnten gewisse Teile serverseitig implementiert werden.

Für das Testing wurde ein Testprotokoll verwendet und die einzelnen, nicht erfüllten Punkte bzw. fehlende Features als Issues in Jira erfasst. Auf ein statisches Requirements-Dokument wurde verzichtet, als Referenz dienten

vielmehr das entwickelte GUI, das Testprotokoll sowie die dynamische Dokumentation einzelner Issues, deren Lösungen diskutiert und in Jira definiert wurden.

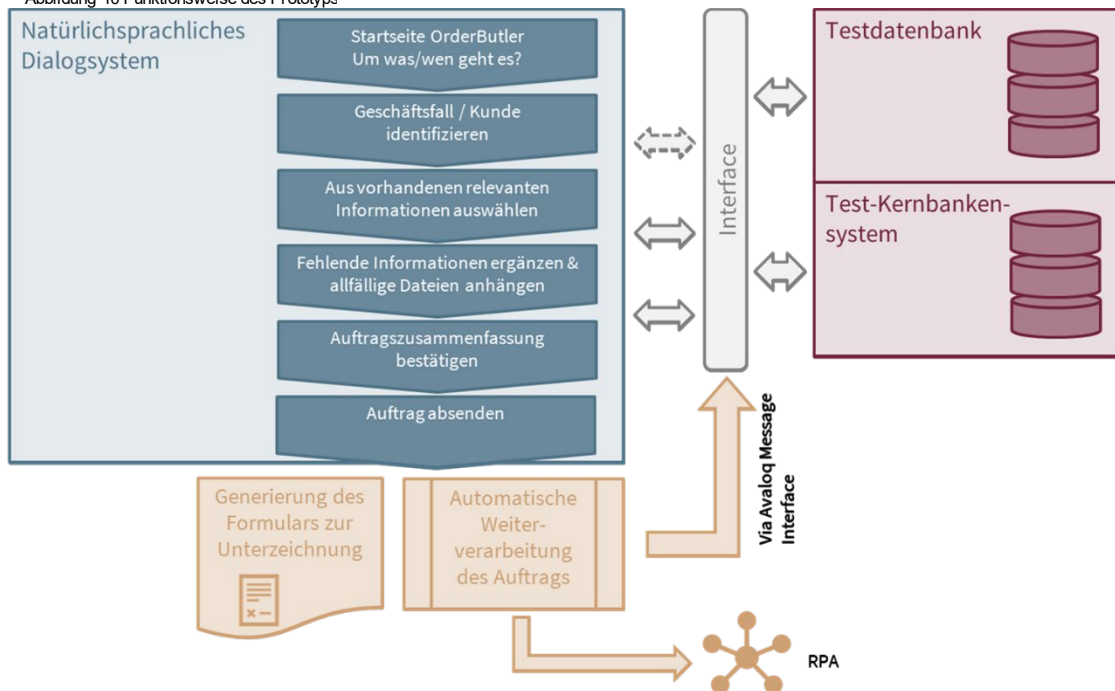
Ebenfalls zu einer agilen Vorgehensweise gehören das automatische Ausliefern (Continuous Delivery) sowie automatisches Testing (Continuous Integration). Zu diesem Zweck wurde ein Dienst des Cloud-Anbieters Heroku genutzt.

4.4 FUNKTIONALE ARCHITEKTUR

4.4.1 Übersicht OrderButler

Der Einstieg in den OrderButler beginnt mit einem einzigen Freitext-Eingabefeld. Als nächster Schritt werden der Geschäftsfall und der Kunde identifiziert. Vorhandene Informationen werden genutzt und der Benutzer wird in Interaktionsschritten aufgefordert, die fehlenden Informationen zu liefern. Der fertige Auftrag wird vom Benutzer zur Abarbeitung freigegeben.

Abbildung 10 Funktionsweise des Prototyps



Die Abarbeitung geschieht entweder automatisch (durch ein RPA-Tool oder über eine Avaloq API) oder manuell durch Backoffice-Mitarbeitende.

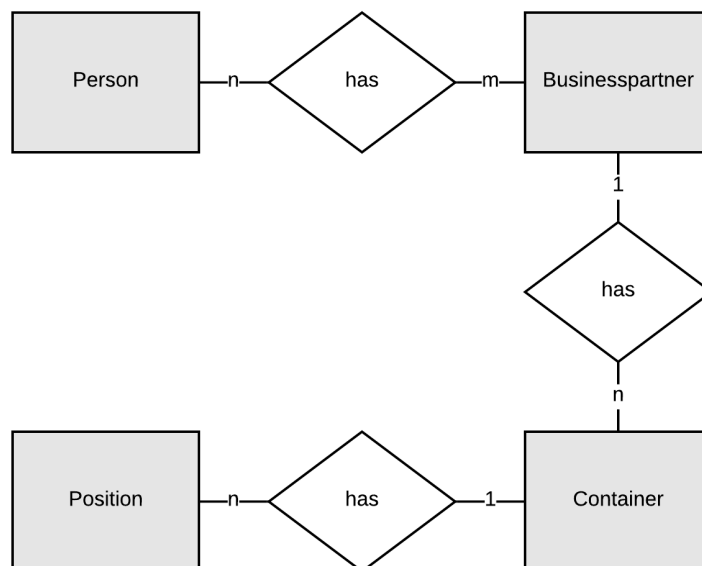
Geschäftsfälle können durch ausgewählte KuBe oder Backoffice-Mitarbeitende angepasst und erstellt werden.

4.4.2 Avaloq-Objekte im OrderButler

Zum Verständnis des OrderButlers gehören minimale Kenntnisse des Kernbankensystems, welches im Hintergrund verwendet werden. Für den Prototypen bei der Pilotpartnerin St.Galler Kantonalbank war dies die Kernbankensoftware Avaloq.

Im OrderButler werden insbesondere die vier Avaloq-Entitätstypen **Businesspartner**, **Person**, **Container** und **Position** verwendet. Die grundlegenden Beziehungen zwischen diesen vier Typen ist in Abbildung 11 skizziert. Positionen entsprechen beispielsweise Konten oder Wertschriften, welche in einem Container gruppiert werden. Jeder Container gehört zu einem Businesspartner. Die realen Personen sind diesem Businesspartner zugeordnet.

Abbildung 11 Vereinfachtes Datenmodell wichtiger Avaloq-Entitätstypen

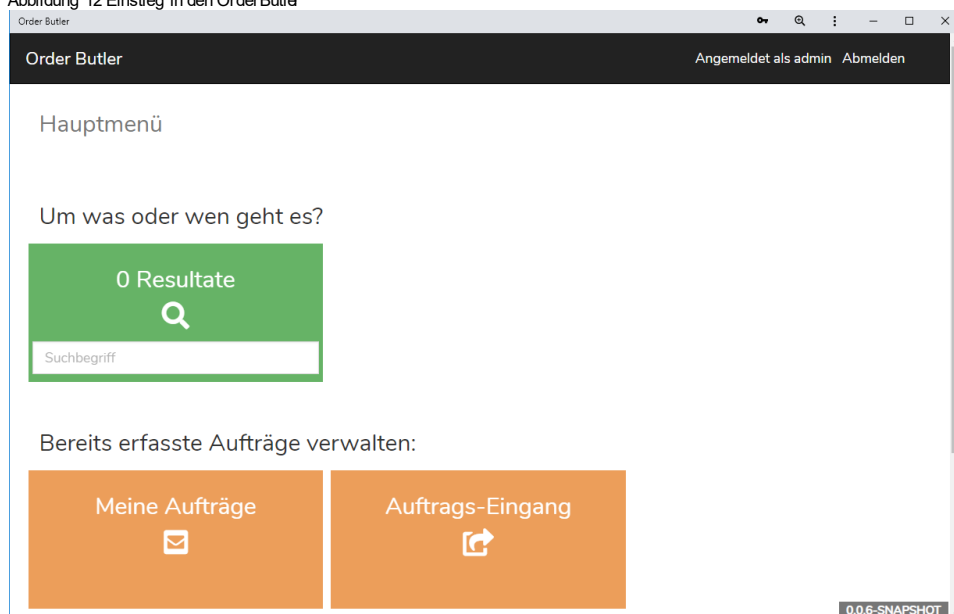


Die Nutzer des Order Butlers müssen mit dieser Avaloq-Logik nicht vertraut sein. Der OrderButler verbirgt diese Komplexität und ermöglicht eine intuitive Erstellung eines Auftrages, wie im nachfolgenden Kapitel beschrieben.

4.4.3 Auftragserstellung

Eine Auftragserstellung beginnt immer mit dem Eingabefeld «Um was oder wen geht es?» (Abbildung 12)

Abbildung 12 Einstieg in den OrderButler



Im OrderButler kann ein Geschäftsfall jeweils einem der vier Avaloq-Objekte Businesspartner, Person, Container und Position zugeordnet werden. Beispielsweise würde der Geschäftsfall «Auszahlung» einem Konto (in Avaloq Position genannt) zugeordnet werden, eine Namensänderung wird hingegen dem Avaloq-Objekt Person zugeordnet.

Im OrderButler kann daher über den Geschäftsfall sowie die vier Avaloq-Objekte Businesspartner, Person, Container und Position eingestiegen werden. Je nach Einstieg fragt danach der OrderButler die notwendigen Daten ab,

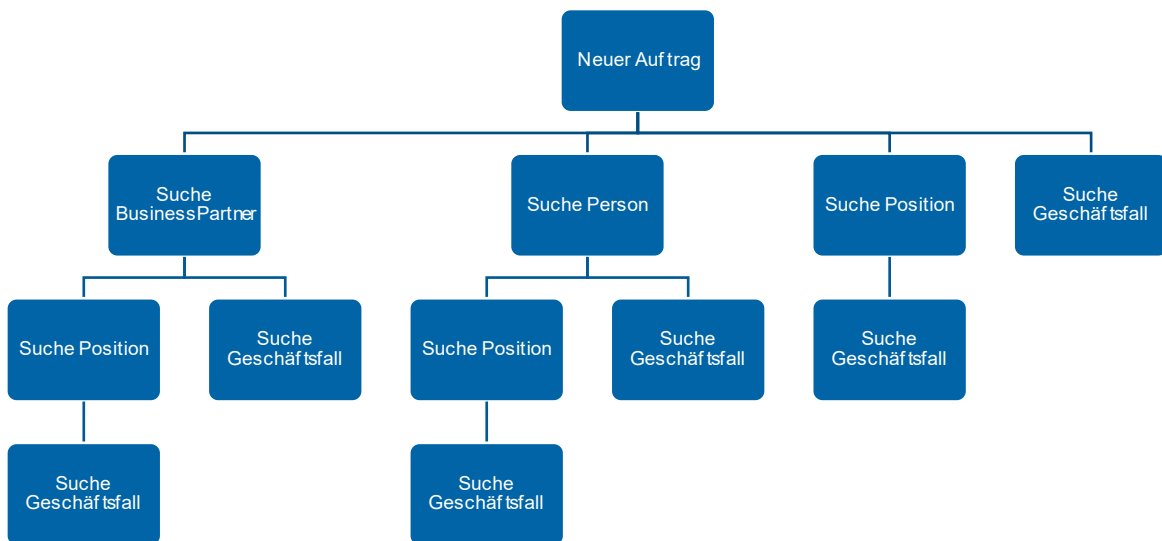
bis das Tupel «Geschäftsfall» - «Avaloq-Objekt» definiert ist. In Abbildung 13 sind einige mögliche Pfade dargestellt.

Beispielsweise kann der Benutzer im Suchfeld nach dem Nachnamen «Kobler»³ suchen. Es folgt die Resultatanzeige gemäss Abbildung 14. Abhängig vom Ranking, welches von der Lucene Volltextsuche errechnet wurde, wird eine Liste von Resultaten dargestellt. Diese Liste zeigt im Beispiel Container, Businesspartner und Personen an, welche Ähnlichkeiten zum Namen «Kobler» aufweisen. Nun gibt es für den Benutzer verschiedene Möglichkeiten fortzufahren. Wenn beispielsweise der Businesspartner erkannt und ausgewählt wird, so werden in einem nächsten Schritt weitere vom Businesspartner abhängige Objekte angezeigt (Abbildung 15) oder es kann ein Geschäftsfall gewählt werden, welcher auf der Ebene «Business-partner» operiert (Abbildung 16).

Wir hingegen in der ersten Resultatliste (Abbildung 14) der Suche nach dem Nachnamen «Kobler» direkt ein Container identifiziert, so schlägt der OrderButler direkt Geschäftsfälle vor, welche auf der Ebene «Container» operieren (Abbildung 17).

Wenn sowohl das Avaloq-Objekt als auch der Geschäftsfall definiert sind, beginnt die eigentliche Datenerfassung (siehe Auftragsbearbeitung).

Abbildung 13: Beispiele für Einstiegspfade



³ Alle Kundennamen in diesem Dokument und im erstellten Prototypen sind fiktiv und entstammen einer Demo-Datenbank des Umsetzungspartners Synpulse, die keinerlei Bezug zu den Systemen oder realen Kundendaten der Pilotpartnerin St.Galler Kantonalbank oder einer anderen Bank hat.

Abbildung 14 Suchresultate für den Namen "Kobler"

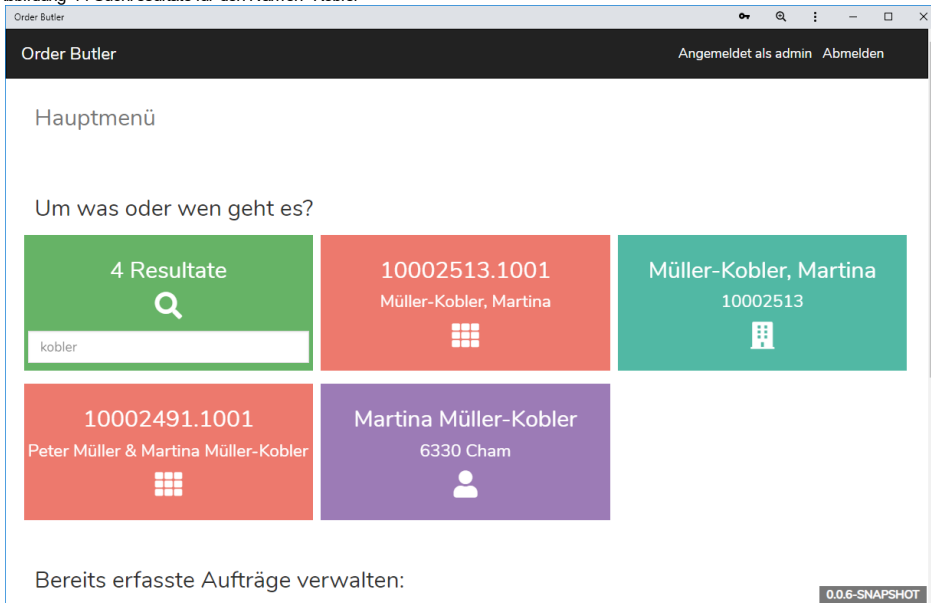


Abbildung 15 Vom gewählten Businesspartner abhängige Avaloq-Objekte



Abbildung 16 Passende Geschäftsfälle zu Businesspartner

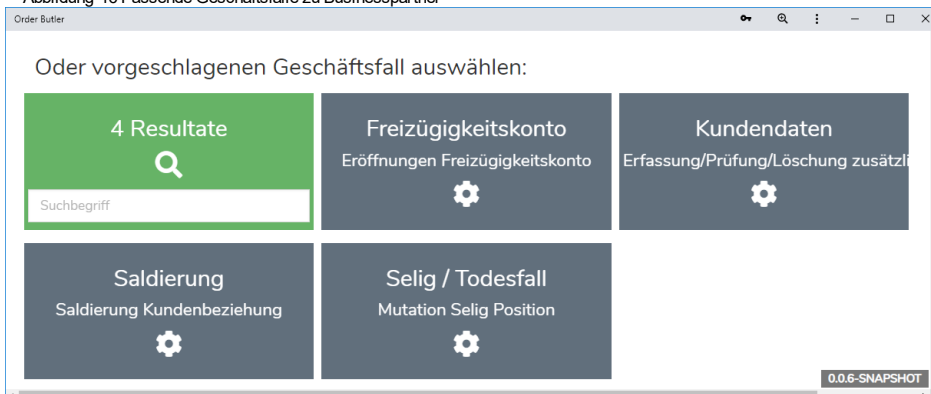
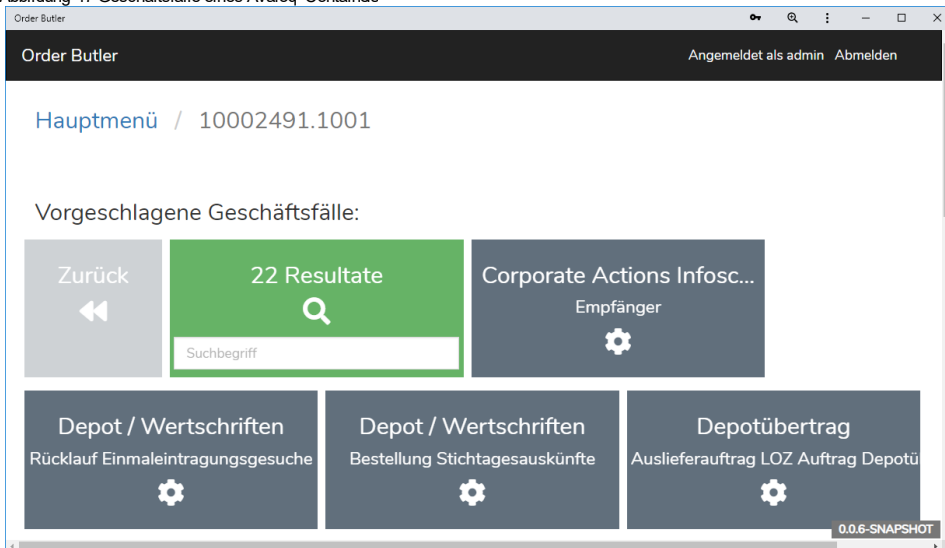


Abbildung 17 Geschäftsfälle eines Avaloq-Containers



4.4.4 Auftragsbearbeitung

Die Auftragsbearbeitung ist grundsätzlich vollständig abhängig vom gewählten Geschäftsfall. Dieser Geschäftsfall definiert einen oder mehrere Interaktionsschritte, die zur Datenerfassung notwendig sind. Es ist möglich, verschiedene Varianten eines Geschäftsfalles abzubilden. Je nach gewählter Variante erscheinen zusätzliche Interaktionsschritte.

Ein Interaktionsschritt kann entweder durch Klicken/Touch beantwortet werden (Abbildung 18 Interaktionsschritt mit Antwort-Buttons) oder es muss eine Eingabe über die Tastatur gemacht werden (Abbildung 19 Interaktionsschritt mit Eingabefeld). Zu den Eingaben zählt auch das Hochladen einer oder mehrere Dateien in den OrderButler (Abbildung 20).

Als letzter Schritt vor dem Versenden des Auftrages erscheint eine Zusammenfassung (Abbildung 21). Wird der Auftrag versandt, so kann er je nach Prozess durch das Backoffice manuell weiterbearbeitet oder über einen RPA-Roboter oder eine Avaloq API automatisch ausgeführt werden.

Abbildung 18 Interaktionsschritt mit Antwort-Buttons

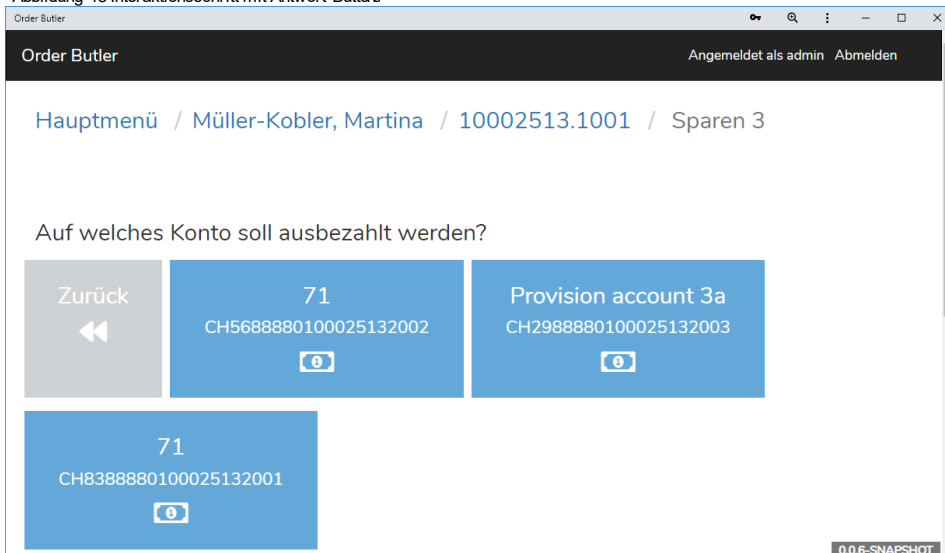


Abbildung 19 Interaktionsschritt mit Eingabefeld

Order Butler

Welcher Betrag soll ausbezahlt werden?

Betrag

Weiter

0.0.6-SNAPSHOT

Abbildung 20 Hochladen mehrerer Dateien in den OrderButler

Order Butler

Bitte Daten vervollständigen um den Sparen 3-Auftrag abzuschliessen:

Datei-Anhänge

File Name	Size	
NegativeSettings.png	26.5 kB	Remove
NegativeSettings2.png	23.1 kB	Remove
NegativeSettings3.png	15.2 kB	Remove

[Choose Files](#) No file chosen

Weiter

0.0.6-SNAPSHOT

Abbildung 21 Zusammenfassung des Auftrages vor dem Versenden

Order Butler

Angemeldet als admin [Abmelden](#)

[Hauptmenü](#) / [Müller-Kobler, Martina](#) / [10002513.1001](#) / [Sparen 3](#)
 / [Provision account 3a](#) / [Interaktions-Element](#) / [Interaktions-Element](#)

Ihr Auftrag 326 ist bereit zum Versand an das Backoffice.

[Auftrag absenden](#) [Abbrechen](#)

Auftragsübersicht:

Beschreibung	Wert
Auftragsnummer	326
Geschäftsfall	Sparen 3 Auszahlung WEF
Ersteller	User Admin

0.0.6-SNAPSHOT

4.4.5 Konfiguration eines Geschäftsfalles

Ein Administrator des OrderButlers kann neue Geschäftsfälle erfassen oder bestehende Geschäftsfälle anpassen. Dieser Vorgang wird Konfiguration eines Geschäftsfalles genannt. Ein Geschäftsfall besteht im Wesentlichen aus

Interaktionsschritten sowie den darin enthaltenen Interaktionselementen. Der Zusammenhang ist in Abbildung 22 ersichtlich.

Abbildung 22 Zusammenhang zwischen Geschäftsfall, Interaktionsschritt und Interaktionselement



Ein Geschäftsfall wird mit der Geschäftsfall-Maske erstellt oder angepasst (Abbildung 23). Name und Beschreibung können frei gewählt werden und erscheinen in der Anzeige des KuBe. Als erster Interaktionsschritt kann ein bestehender Schritt gewählt werden oder es kann über den Button «Neu» sofort ein neuer Schritt angelegt werden.

Als Geschäftsobjekt-Typ kann eines der vier Avaloq-Objekte Businesspartner, Person, Container oder Position ausgewählt werden. Dies definiert den möglichen Einstieg in den OrderButler, welcher in Abschnitt Auftragserstellung beschrieben ist. Die Bearbeitungsgruppe definiert den Personenkreis, welcher im Backoffice diesen Auftrag bearbeiten wird (bei manueller Bearbeitung eines Auftrages).

Abbildung 23 Maske "Geschäftsfall"

Sobald der Geschäftsfall erstellt ist kann dieser weiter konfiguriert werden (Abbildung 24). Unter Stichworten können weitere, alternative Stichworte definiert werden (Abbildung 25), unter denen der Geschäftsfall mittels Volltextsuche ebenfalls gefunden wird.

Die eigentliche Konfiguration des Ablaufs besteht aus der Konfiguration mindestens einen oder idealerweise auch mehrerer Konfigurationsschritte. Mit Klick auf den Button «Interaktions-Schritte» wird die Liste der Interaktionsschritte pro Geschäftsfall angezeigt (Abbildung 26).

Für jeden Interaktionsschritt kann die Liste seiner Elemente (Abbildung 27) angezeigt werden. Jeder Interaktionsschritt wird aufgrund der Liste seiner Elemente vom OrderButler dynamisch aufgebaut und dem KuBe beim Ausführen des Geschäftsfalles präsentiert. Dieser Zusammenhang ist in Abbildung 28 ersichtlich. Eine Beschreibung der möglichen Typen von Elementen findet sich in Tabelle 3.

Abbildung 24 Weitere Konfiguration eines Geschäftsfalles

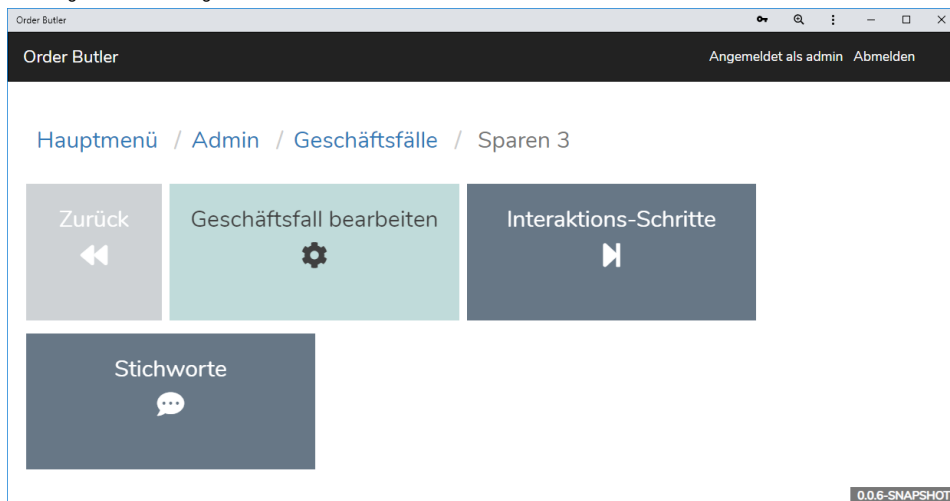


Abbildung 25 Alternative Stichwort für die Volltextsuche

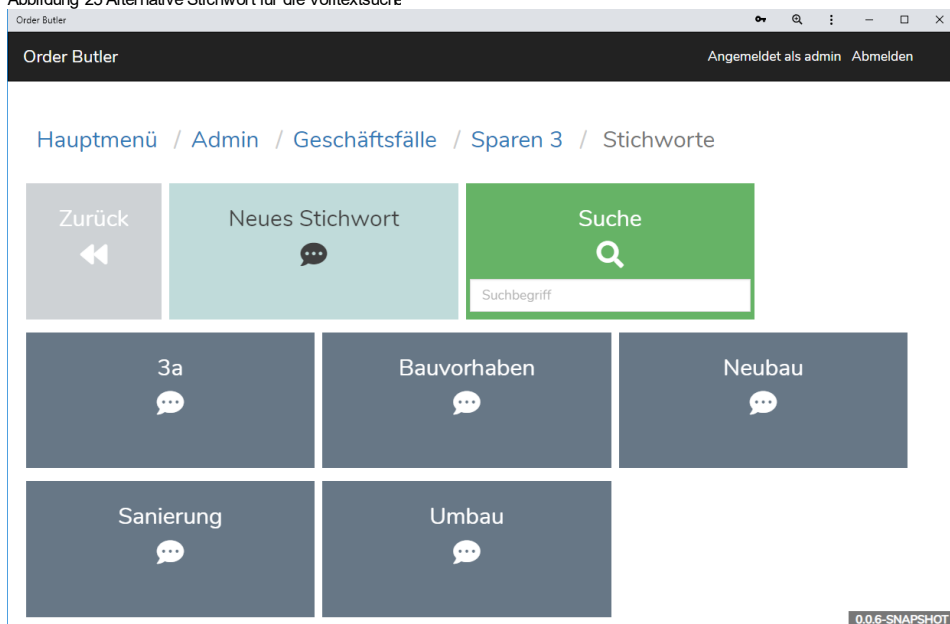


Abbildung 26 Von einem Geschäftsfall verwendete Interaktionsschritte

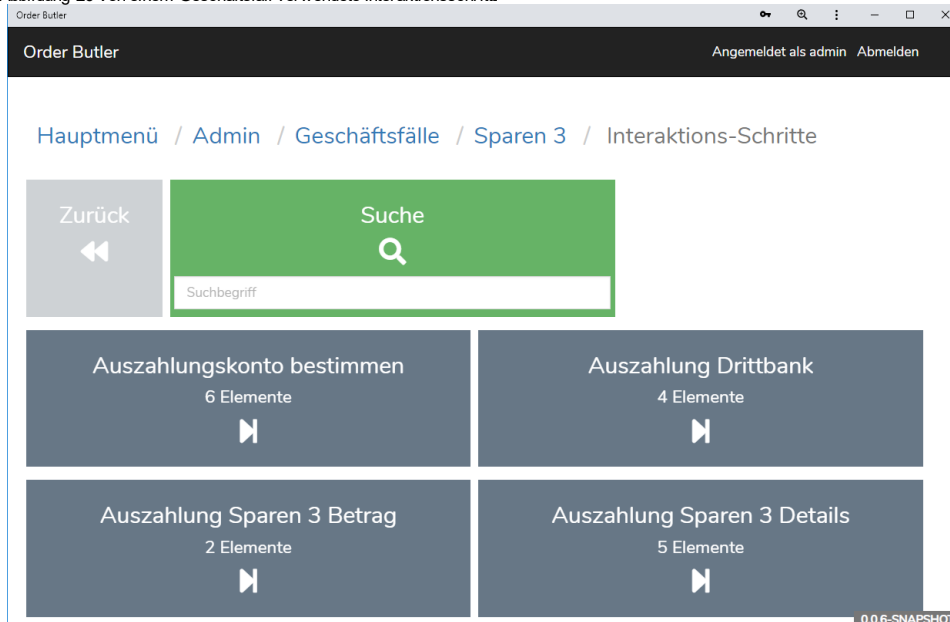


Abbildung 27 Liste der Elemente eines Interaktionsschrittes

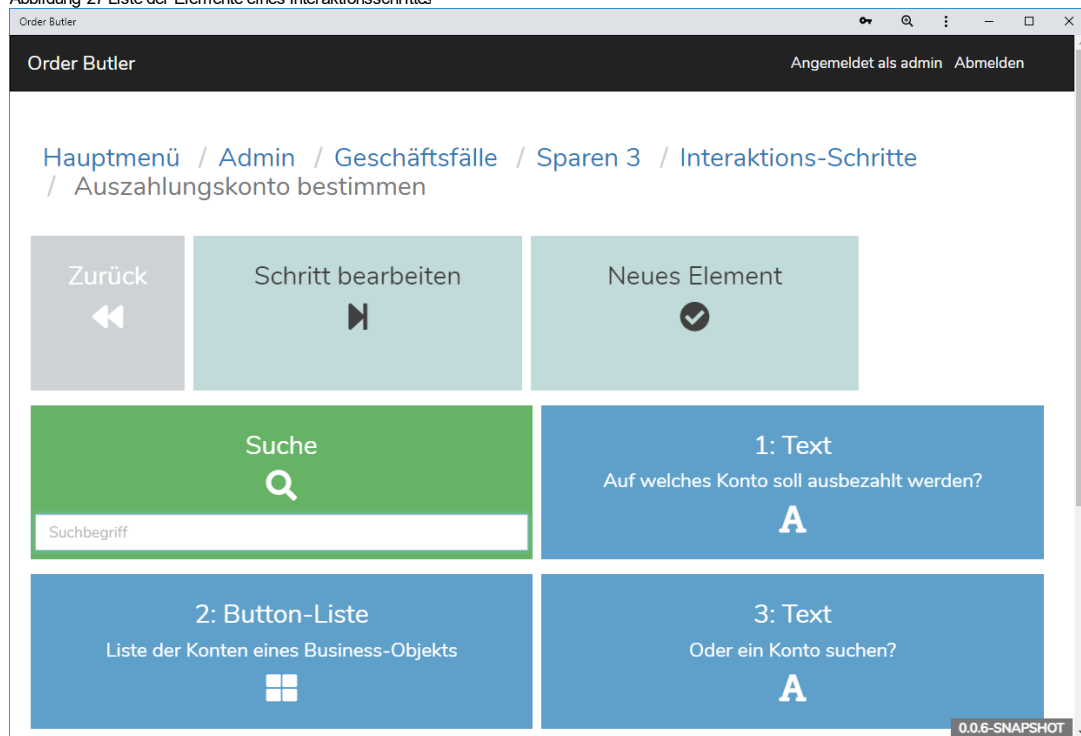
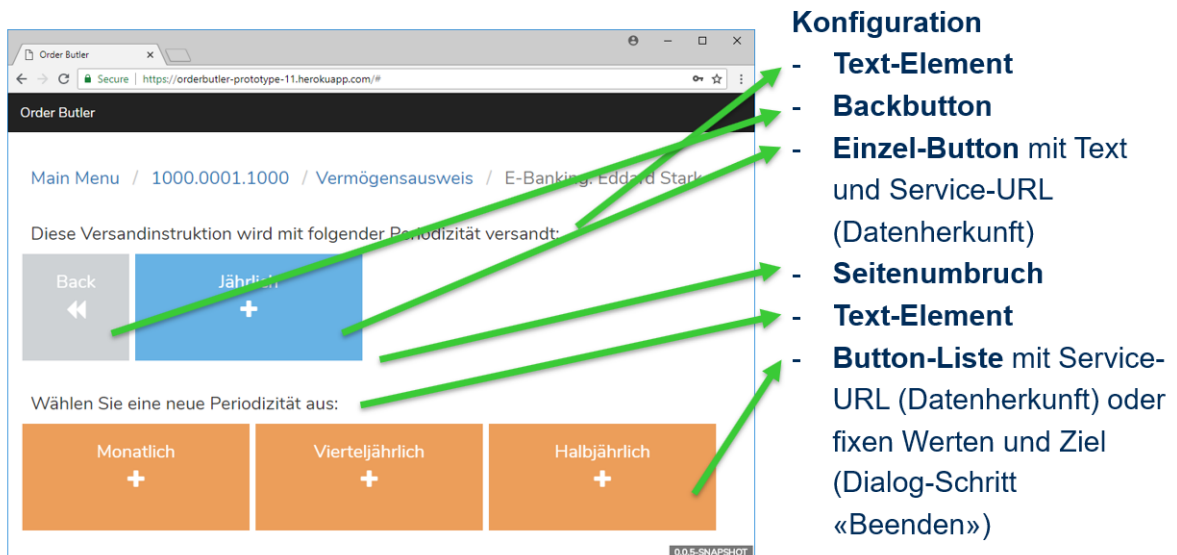
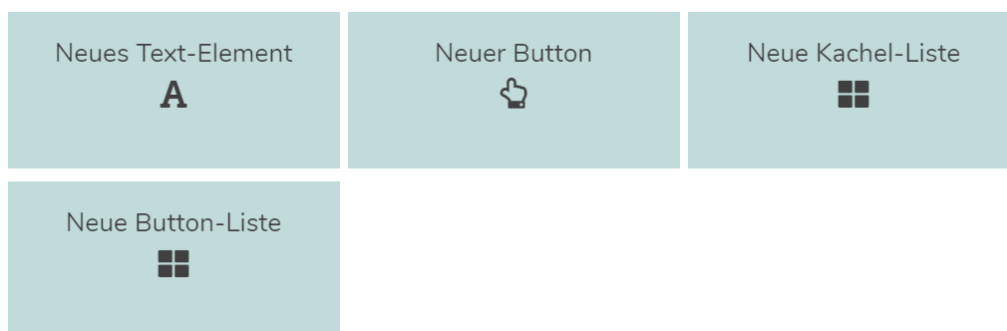


Abbildung 28 Zusammenhang zwischen Interaktionselementen und Ansicht des KuBe



Die aktuelle Version des OrderButlers verfügt über neun verschiedene Interaktionselemente (Abbildung 29).

Abbildung 29 Verschiedene Interaktions-Elemente
Text, Kacheln und Buttons



Eingabefelder



Tabelle 3 Beschreibung der Interaktions-Elemente

INTERAKTIONS-ELEMENT	BESCHREIBUNG
Text-Element	Ein statisches Text-Element. Wird typischerweise dazu verwendet, dem Benutzer eine Frage zu stellen.
Button	Ein einzelner Button, der auf einen nächsten Interaktionsschritt verweisen kann.
Kachel-Liste	Eine dynamische Liste von Kacheln, die aufgrund von Avaloq-Daten angezeigt wird, beispielsweise eine Liste von Positionen.
Button-Liste	Ebenfalls eine dynamische Liste analog zur Kachel-Liste, aber es kann ein Element ausgewählt werden und auf einen nächsten Interaktionsschritt verweisen.
Datum-Eingabefeld	Der Benutzer kann ein Datum erfassen.
Zahlen-Eingabefeld	Der Benutzer kann eine Zahl erfassen. Es besteht die Möglichkeit Minimum und Maximum sowie die Anzahl Nachkommastellen festzulegen.
Radio-Buttons	Ein Liste von Radio-Buttons, von denen der Benutzer genau einen Wert auswählen kann.
Text-Eingabefeld	Der Benutzer kann einen Text erfassen.
Datei-Upload	Der Benutzer kann eine oder mehrere Dateien über Drag & Drop beziehungsweise die Dateiauswahl hochladen.

Die Maske zur Erfassung eines Interaktions-Elementes (Abbildung 30) sieht je nach Typ des Interaktions-Elementes unterschiedlich aus.

Abbildung 30 Maske zur Erfassung eines Interaktions-Elementes

The dialog box 'Button-Liste mit Server-Daten' contains the following fields and controls:

- Interaktions-Schritt:** A list of buttons: 'Auszahlungskonto b...', 'Detail...', 'Nächster Interaktions-Schritt', 'Auszahlung Sparen £', and 'Detail...'.
- Reihenfolge:** A text input field containing the value '4'.
- Export-Schlüssel:** A text input field containing the value 'Account'.
- Daten-URL:** A text input field containing 'Alle Positionen' and a 'Detail...' button.
- Suche sichtbar:** Radio buttons for 'Ja' and 'Nein', with 'Nein' selected.
- Suche notwendig:** Radio buttons for 'Ja' and 'Nein', with 'Ja' selected.
- Wert-Beschreibung:** A text input field containing 'Auszahlungskonto'.
- Buttons:** 'Löschen', 'Abbrechen', and 'OK'.

Tabelle 4 Eigenschaften eines Interaktions-Elementes

FELD	BESCHREIBUNG
Interaktions-Schritt	Ein Interaktions-Element gehört immer genau zu einem Interaktions-Schritt.
Nächster Interaktions-Schritt	Wenn ein Interaktions-Element vom Endbenutzer bedient werden kann, so wird hier der nächste Interaktions-Schritt definiert.
Reihenfolge	Die Reihenfolge der Elemente auf einer Seite wird durch ganze Zahlen definiert. Die Position eines Elementes kann hier als ganze Zahl definiert werden.
Export-Schlüssel	Wenn ein Auftrag für den RPA-Roboter exportiert wird, so wird als Feld-Bezeichnung dieser technische Wert verwendet.
Daten-URL	Definiert die Daten, welche von Avaloq abgerufen werden.
Suche sichtbar	Definiert, ob ein Suchfeld angezeigt wird. Empfehlenswert wenn grosse Datenmengen erwartet werden.
Suche notwendig	Definiert, ob bei der Anzeige sofort Resultate angezeigt werden oder ob der Benutzer zuerst eine Suche ausführen muss. Empfehlenswert bei grossen Datenmengen.
Wert-Beschreibung	Die Beschreibung des Wertes, welche in der Zusammenfassung des Auftrages verwendet wird.

4.4.6 Benutzerverwaltung

Der OrderButler besitzt eine einfache Benutzerverwaltung. Dies reicht für einen Prototypen aus, im produktiven Betrieb wird sinnvollerweise ein bestehendes Verzeichnis (z.B. LDAP, Exchange oder Avaloq) verwendet.

Die Eigenschaften eines Benutzers werden in der dazugehörigen Maske (Abbildung 31) erfasst. Als Benutzerkennung wird die E-Mail-Adresse verwendet, das Passwort wird verschlüsselt auf der Datenbank abgelegt.

Über die Rollen kann einem Benutzer die Rolle «Administrator» zugewiesen werden. Dies erlaubt dem Benutzer das Konfigurieren von Geschäftsfällen.

Jeder Benutzer kann einer oder mehreren Bearbeitungsgruppen zugeordnet werden. Diese sind im aktuellen Stand des Prototypen nicht im UI ersichtlich und können nur über die Datenbank administriert werden. Eine Bearbeitungsgruppe ist eine Gruppe von Backoffice-Mitarbeitenden, welche einen bestimmte Art von Geschäftsfällen abarbeitet.

Abbildung 31 Maske "Benutzer"

Benutzer

Nachname: Moser Vorname: Adrian

E-Mail: mosa@zhaw.ch

Passwort:

Passwort wiederholen:

Erstellungsdatum: 01.03.2019 Gültig bis: []

Kommentar: []

Löschen Abbrechen OK

4.4.7 Machine Learning

Die Geschäftsfall-Liste nach Auswahl beispielsweise eines Businesspartners kann lange und unübersichtlich werden (siehe Abbildung 32). Mit Hilfe von Machine Learning wird versucht, diese Liste sinnvoll anzuordnen und in Frage kommende Geschäftsfälle am Anfang aufzulisten.

Diese Funktionalität befindet sich jedoch im Aufbau und ist daher standardmässig abgeschaltet.

Einschalten lässt sich die Funktion im Abschnitt "Learning" in der Administrationssicht (Abbildung 33). Hier können auch Testdaten verwaltet werden. Sobald die Testdaten erfasst sind, kann das Training gestartet werden und der OrderButler zeigt Accuracy, Precision und Recall an.

Abbildung 32 Liste der Geschäftsfälle eines Businesspartners

Order Butler

Angemeldet als admin Abmelden

Hauptmenü / 10002513.1001

Vorgeschlagene Geschäftsfälle:

Zurück 22 Resultate Corporate Actions Infosc... Depot / Wertschriften

Empfänger Rücklauf Einmaleintragungsgesuche

Suchbegriff

Depot / Wertschriften Depotübertrag Depotübertrag

Bestellung Stichtagesauskünfte Auslieferungsauftrag LOZ Auftrag Depotü Auftrag Depotübertrag

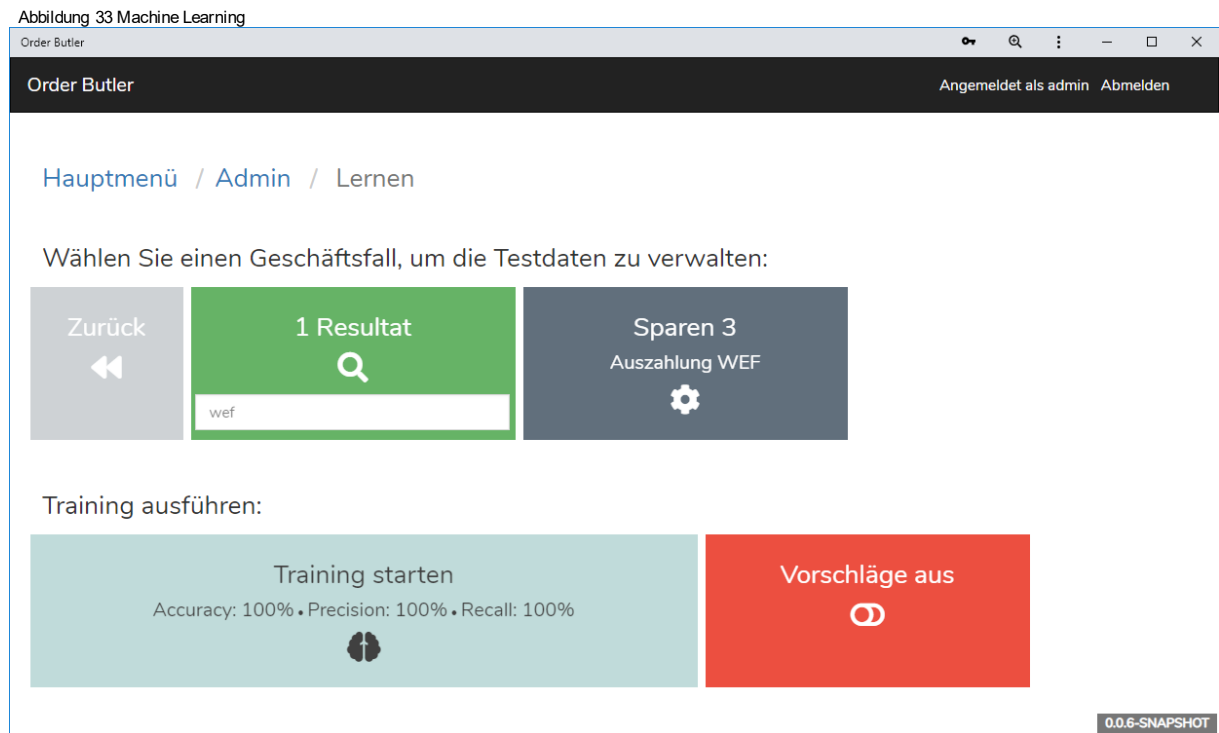
Erfassung Versandinstruk... Erfassungen Einstandspr... Eröffnung Valoren

Erfassung Versandinstruktionen Erfassungen Einstandspreise Anfragen/Korrekturen Valor

Freizügigkeitskonto Quellen- und Verrechnun... Rückerstattungen Gebühr...

0.0.6-SNAPSHOT

Über den Button «Vorschläge aus» können die Vorschläge für Geschäftsfälle aktiviert werden. Diese Implementation von Machine Learning ist als exemplarisch zu betrachten, für einen ersten Einsatz müssten Testdaten in genügender Zahl zur Verfügung stehen, damit sich der Nutzen optimieren liesse.



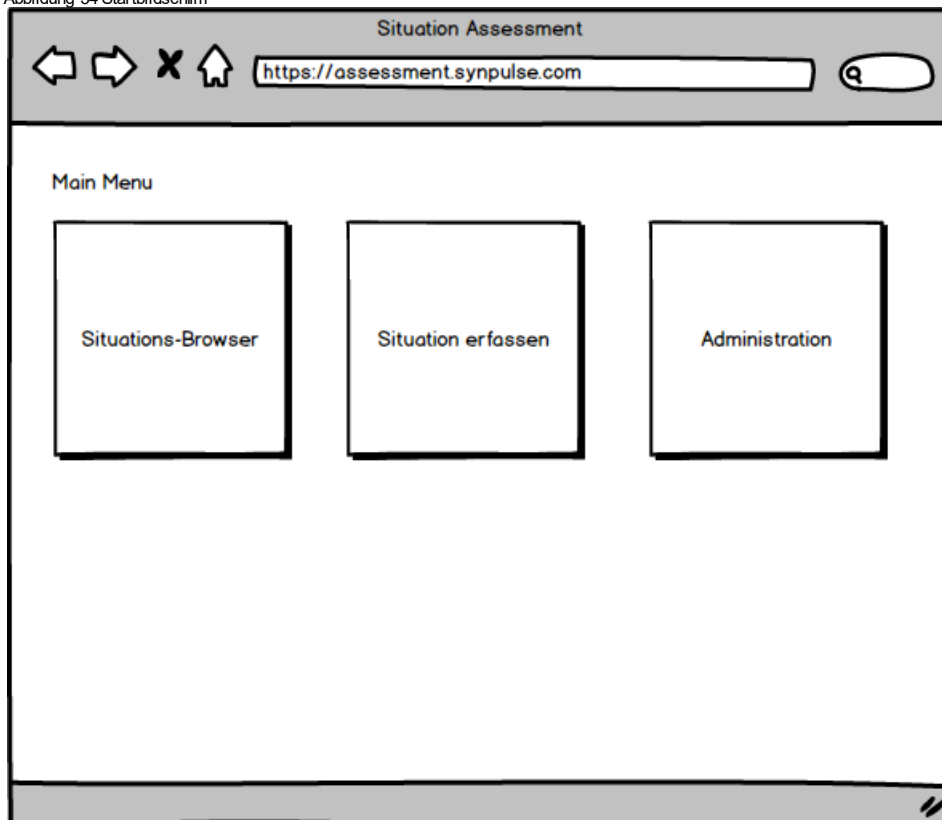
4.5 PROTOTYP I

Ziel des Prototypen I war es, effektive Dialogstrukturen für die Auswahl der Geschäftsfälle und die Formulierung des Auftrages zu identifizieren. Der Prototyp I diente als Artefakt, um unterschiedliche Strukturierungsgrade von Dialogen mit den KuBe für verschiedene Aktivitäten und Ziele der Auftragserteilung erproben und beobachten zu können.

4.5.1 Erste Mockups

Für den Prototyp I wurde mit dem Entwurf von Mockups gestartet. Dabei stand zuerst die Idee im Vordergrund, die Situation des Bankkunden zu erfassen (Abbildung 34 Startbildschirm und Abbildung 35 Erfassen einer Situation).

Abbildung 34 Startbildschirm



Danach wurden dem Benutzer für diese Kundensituation passende Geschäftsfälle präsentiert (Abbildung 36 Auswahl eines Geschäftsfalles). Um ein Feedback zu den für den Auftrag wesentlichen Informationen zu erhalten, wurde anschliessend eine reduzierte Eingabemaske in Anlehnung an die papierbasierten Formulare präsentiert (Abbildung 37 Ausfüllen des digitalen Formulars).

Bereits bei den ersten Treffen mit KuBe der Bank wurde auf der Grundlage der Mockups jedoch klar, dass die Auswahl eines Geschäftsfalles in einer konkreten Kundensituation typischerweise keine Herausforderung darstellt. Vielmehr ist den KuBe in der Regel klar, welcher Geschäftsfall in welcher Situation in Frage kommt.

Somit verlagerte sich der Fokus des Dialogsystems von der Analyse der Situation hin zum Erfassen der relevanten Daten. Auch hier war schnell klar, dass es nicht einfach reicht bestehende Formulare digital abzubilden und somit die Schwierigkeiten beim Ausfüllen des Papierformulars auf die elektronische Variante zu übertragen.

Abbildung 35 Erfassen einer Situation

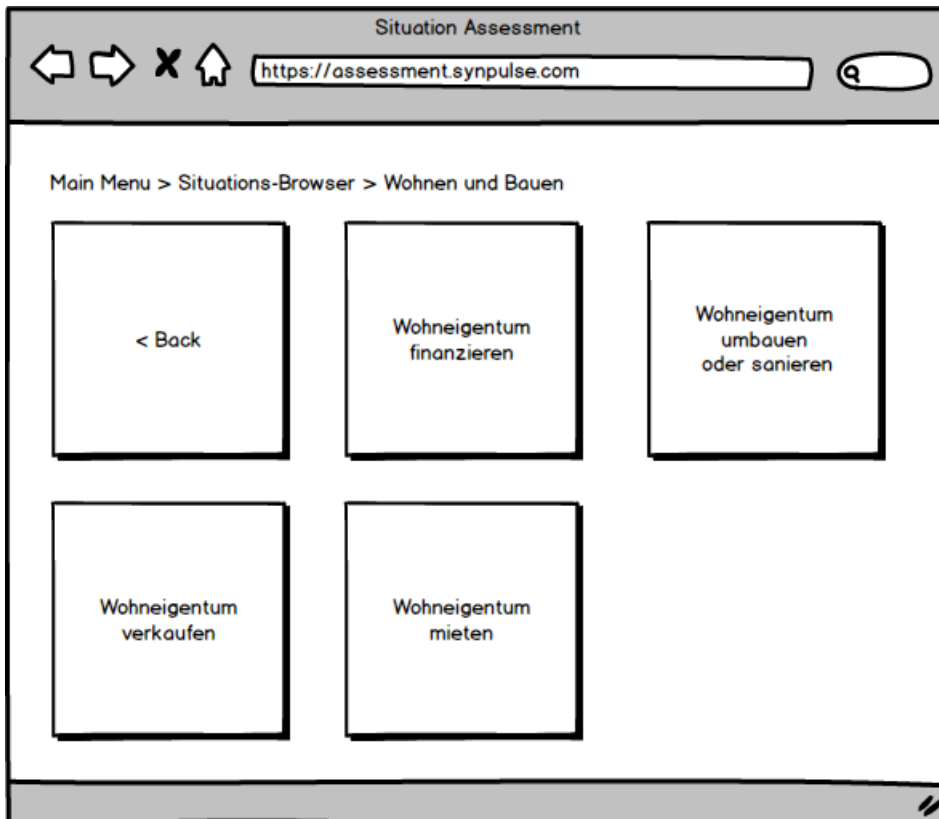


Abbildung 36 Auswahl eines Geschäftsfalles

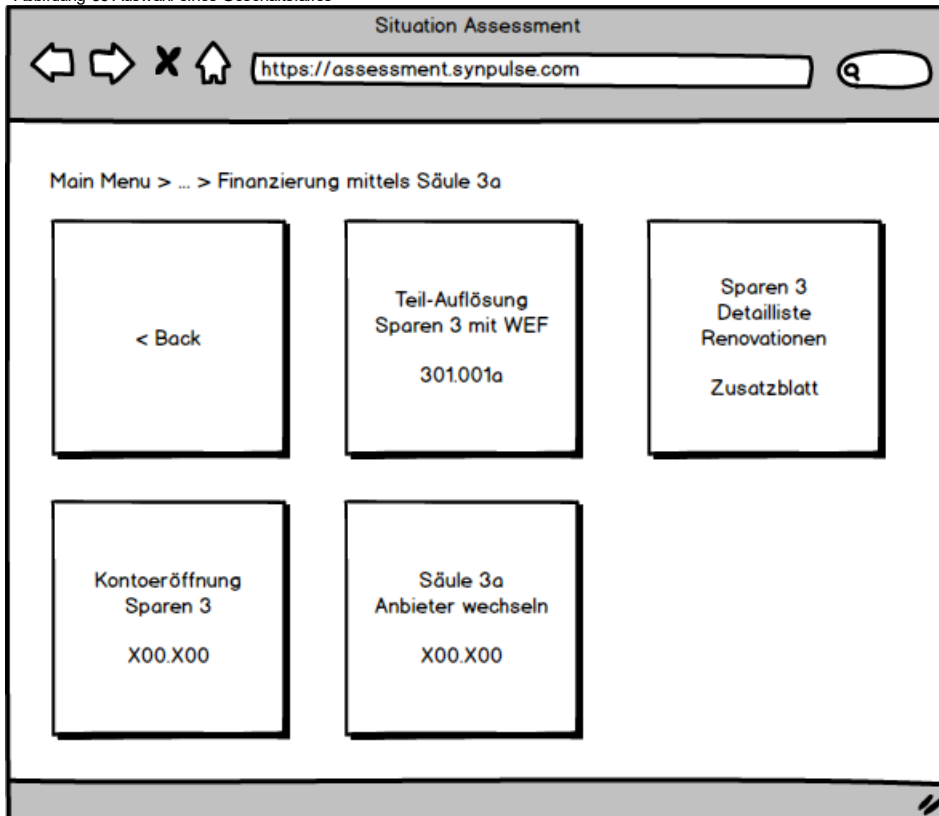


Abbildung 37 Ausfüllen des digitalen Formulars

4.5.2 Funktionsweise des Prototyp I

In Prototyp I wurde versucht einige der Formulare mittels eines möglichst einfachen User Interfaces abzubilden (Abbildung 39 Startbildschirm Prototyp I). Dabei wurde auf ein echtes Backend verzichtet, d.h. es wurden manuell einige Testdatensätze erzeugt und im Prototyp verwendet. Nur einige wenige Interaktionen im Prototyp waren funktional, d.h. der Benutzer musste sich genau an die Vorgaben aus den User Stories halten, mit abweichenden Daten war der Prototyp I nicht funktionsfähig.

Die Funktionsweise des Prototyps I wird anhand des Testfalles „Sparen 3“ erläutert.

Abbildung 38: Szenario Sparen 3

Szenario Sparen 3

Herr Eddard Stark wird pensioniert da er das AHV-Alter erreicht hat. Er möchte sich alles Geld aus seinem Sparen 3 Konto 1000.0001.2001 auszahlen lassen. Das Geld soll auf das Konto 1000.0001.2002 ausbezahlt werden. Die Wertschriften sollten so bald wie möglich verkauft werden und das Sparen 3 Konto soll anschliessend saldiert werden.



Quelle: pexels.com – CC0

Gestartet wird mit dem Einstiegsbildschirm (Abbildung 39 Startbildschirm Prototyp I). Bereits hier kann die Aufgabe in verschiedenen Varianten absolviert werden. Es wäre möglich direkt nach dem Geschäftsfall (Sparen 3), dem Konto oder der betroffenen Person zu suchen.

Im Beispiel wird der Name der betroffenen Person eingegeben (Eddard Stark) und es folgen weitere Informationen zu dieser Person (Abbildung 40 Szenario Sparen 3: Auswahl von Positionen oder Containern der gewählten Person). Im konkreten Beispiel wurde ein Avaloq-Container sowie drei Positionen in Avaloq gefunden. Im Prototyp I werden in diesem Schritt aber keine echten Avaloq-Daten verwendet, es handelt sich um ein sehr begrenztes Set von hardkodierten Daten.

Gemäss Testfall (Abbildung 38: Szenario Sparen 3) wird nun das Konto 1000.0001.2001 als betroffenes Konto identifiziert und ausgewählt. Da die Daten hardkodiert sind, wird die Suche nach einer anderen Kontonummer nicht unterstützt und der Anwender muss sich genau im Rahmen des vorgegebenen Geschäftsfalles bewegen. Im nächsten Schritt (Abbildung 41 Szenario Sparen 3: Vorgeschlagene Geschäftsfälle) kann der passende Geschäftsfall ausgewählt werden, in diesem Fall „Sparen 3 – Auszahlung Pensionierung“.

Nun sind die betroffene Person (Eddard Stark) sowie der gewünschte Geschäftsfall identifiziert (Sparen 3 – Auszahlung Pensionierung). Damit beginnt die eigentliche Abarbeitung des Geschäftsfalles.

Abbildung 39 Startbildschirm Prototyp I

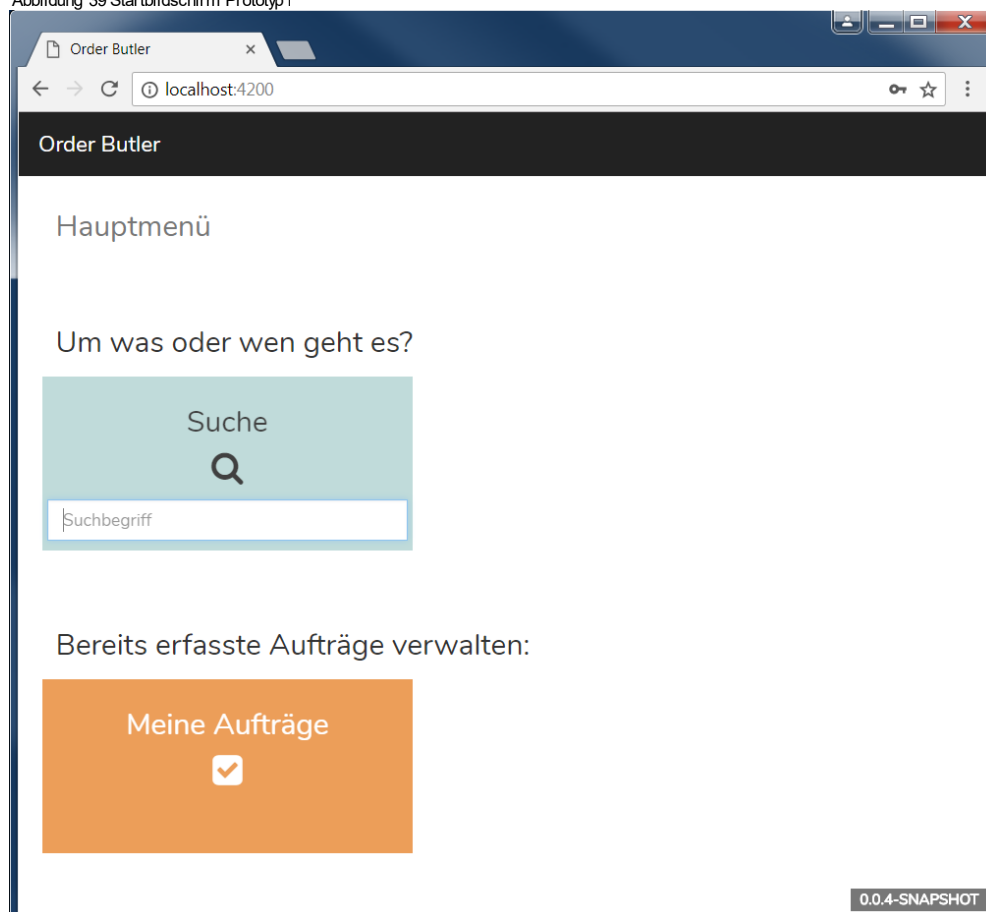


Abbildung 40 Szenario Sparen 3: Auswahl von Positionen oder Containern der gewählten Person

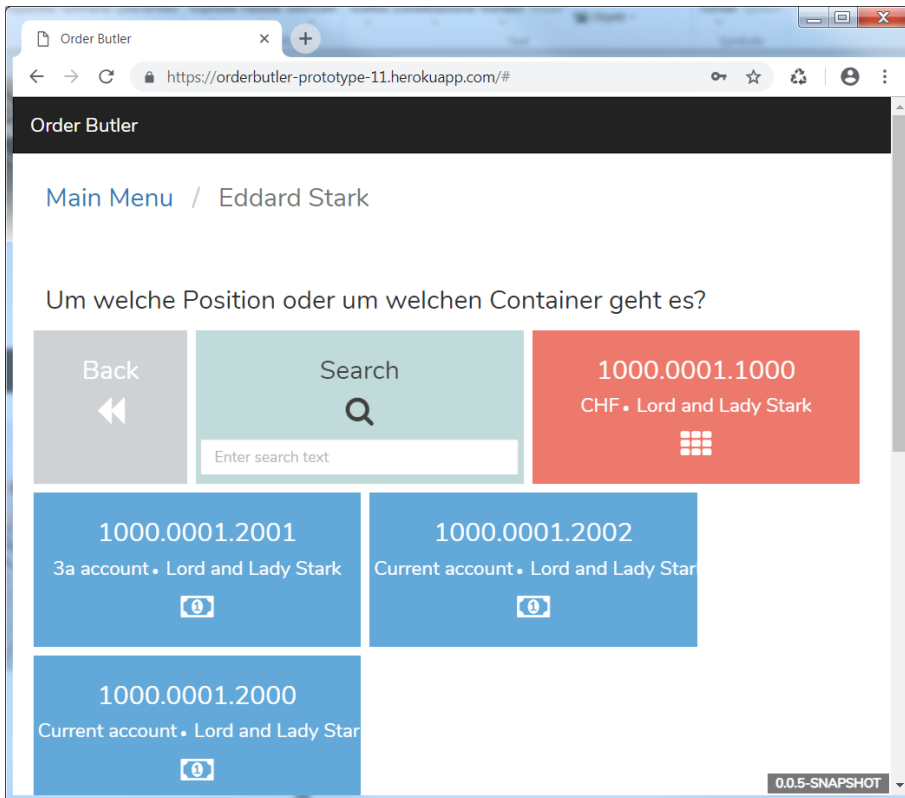


Abbildung 41 Szenario Sparen 3: Vorgeschlagene Geschäftsfälle

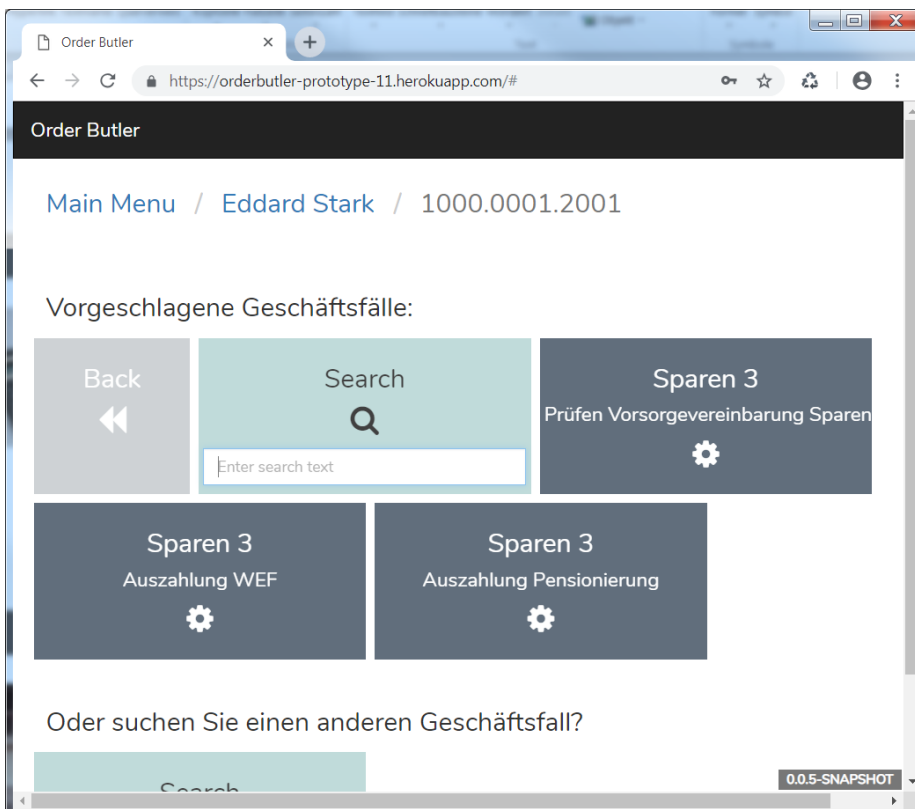
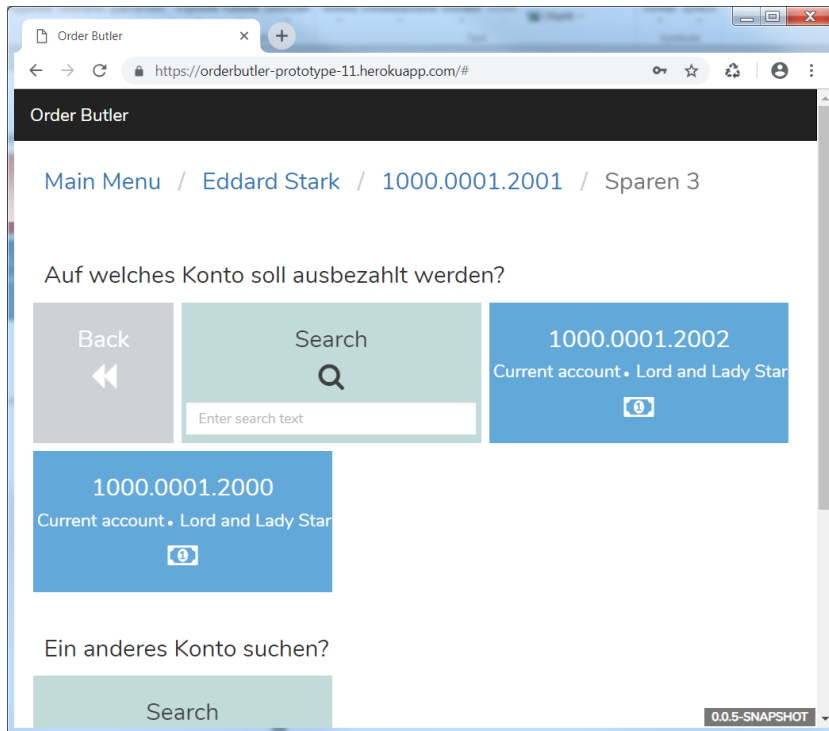


Abbildung 42: Szenario 3: Auswahl des Auszahlungskontos



Nun wird das Konto abgefragt, auf welches ausbezahlt werden soll. Nach Wahl des Kontos 1000.0001.2002 wird der Benutzer auf die letzte Interaktionssseite weitergeleitet (Abbildung 43 Szenario Sparen 3: Vervollständigen des Auftrages). Hier wurde der Formularstil beibehalten, um die letzten Daten abzufragen. Die Idee war, den Vergleich zwischen verschiedenen Interaktionsformen zu ermöglichen (Kacheln vs. klassische Formularelemente).

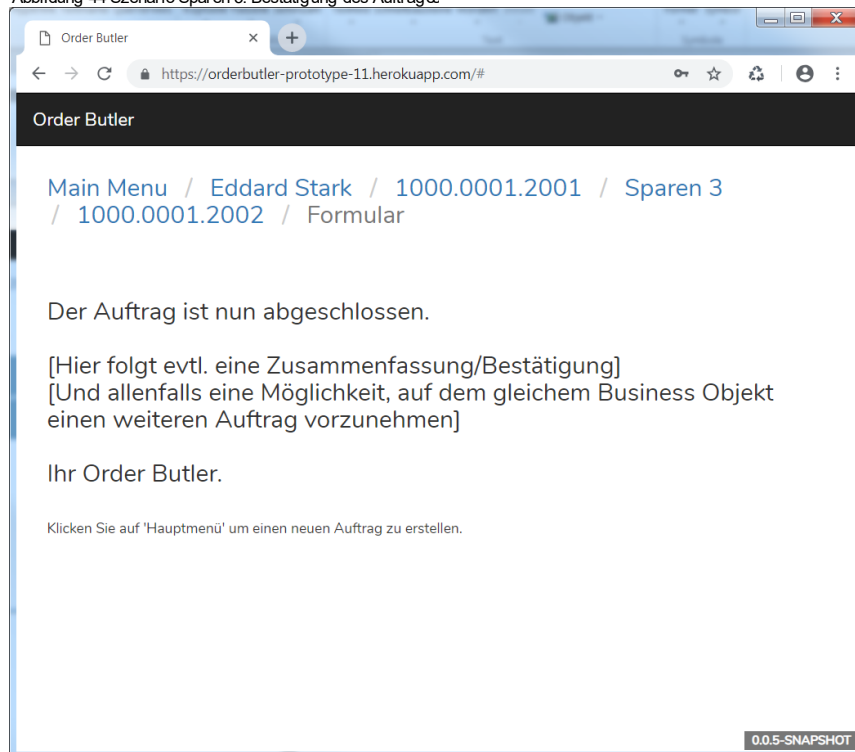
Nach Ausfüllen der restlichen Daten ist der Auftrag abgeschlossen und dem Benutzer wird eine Bestätigungsseite angezeigt (Abbildung 44 Szenario Sparen 3: Bestätigung des Auftrages).

Abbildung 43 Szenario Sparen 3: Vervollständigen des Auftrages



Bei Prototyp I werden aber keine Aufträge gespeichert, da die Anbindung an ein echtes Backend mit Datenbank noch nicht gegeben ist.

Abbildung 44 Szenario Sparen 3: Bestätigung des Auftrages



4.6 PROTOTYP II

4.6.1 Erweiterbarkeit, Konfigurierbarkeit

In Prototyp II wurde einerseits der Prototyp I funktionsfähig gemacht und andererseits eine erste Version der Erweiterbarkeit und Konfigurierbarkeit umgesetzt.

Folgende Punkte wurden umgesetzt:

- Verwendung von erweiterten Avaloq-Testdaten: Praktisch alle hardkodierte Testdaten aus Prototyp I wurden auf Testdaten umgestellt, welche durch den OrderButler aus einer MongoDB gelesen werden. Diese MongoDB stellt einen Export der Avaloq-Testdatenbank von Synpulse dar.
- Indexierung der Daten: Wo immer möglich wurde bei Suchen ein mit Apache Lucene aufgebauter Volltextindex verwendet. So konnte der Zugriff auf die Avaloq-Testdaten in angemessener Zeit erfolgen. Als zusätzlicher Nutzen für den Benutzer wird auch Ähnlichkeitssuche unterstützt (beispielweise die Suche nach Herrn Stark ergibt auch Resultate mit dem Namen Stark).
- Sämtliche Abläufe und Interaktionen in Prototyp I waren hardkodiert. Für Prototyp II wurde ein Datenmodell aufgebaut, welches sich im Wesentlichen bereits am finalen Prototyp orientierte (siehe Abschnitt 4.7). Die bestehenden Abläufe und Interaktionen wurden in dieses Datenmodell migriert und neu programmiert.
- Es konnte bereits ein einfaches User Interface zur Verwaltung des erwähnten Datenmodells erstellt werden. Damit konnten bereits Geschäftsfälle erstellt, mit Interaktionsschritten und Interaktionselementen befüllt werden und diese neuen Geschäftsfälle durchgespielt werden.

4.7 PROTOTYP III

Im Fokus der Entwicklungsarbeiten und Testings von Prototyp III standen die Elemente «Durchgängigkeit & Lernfähigkeit» sowie «Machine Learning».

4.7.1 Durchgängigkeit (RPA)

Das Tool «OrderButler» fokussiert primär auf das vollständige und strukturierte Erfassen von Aufträgen. Um die Wirtschaftlichkeit des Einsatzes eines solchen Tools wesentlich zu erhöhen, stand die Validierung der Durchgängigkeit im Fokus des letzten Prototypen. Unter Durchgängigkeit verstehen wir die Möglichkeit, die formulierten Aufträge an richtiger Stelle zu disponieren oder gar automatisch und vollständig, z.B. durch Mutation von Stammdaten in einem Kernbankensystem, abzuarbeiten.

Die automatische Disponierung und ein hoher Grad an Automatisierung bringen zwei wesentliche Vorteile mit sich:

- Effizienzsteigerung: Durch die korrekte Disposition und die automatische Abarbeitung sinkt die Anzahl der manuellen Schritte und Eingriffe durch das Bankpersonal wesentlich. Zum einen können dadurch Ressourcen eingespart oder anderweitig eingesetzt werden, zum anderen können Aufträge schnell oder gar zeitgesteuert abgewickelt werden.
- Qualitätssteigerung: Während der Erfassung von Aufträgen können durch die Komponente «Mensch» zusätzliche Fehler in der Erfassung, Disponierung und Abwicklung eines Auftrags auftreten. Für die Korrektur dieser Fehler muss in der Regel der Empfänger beim Auftraggeber (in diesem Fall der KuBe) nachfragen, was die Effizienz der Abwicklung ebenfalls senkt. Zusätzlich ermöglicht ein Set mit Regeln zur Disponierung, jeden Auftrag an den richtigen Empfänger zu senden oder an ein Drittsystem zur automatischen Verarbeitung zu übergeben.

Die Anbindung des OrderButlers an Drittsysteme zur Herstellung der Durchgängigkeit kann in verschiedenster Weise umgesetzt werden, und hängt in erster Linie von den betroffenen Prozessen innerhalb der Bank sowie den involvierten IT-Systemen ab. Im Rahmen dieses Projekts wurde die Anbindung des OrderButler an das Kernbankensystem Avaloq geprüft und der Fokus auf die automatische Abwicklung eines Auftrags gelegt.

Grundsätzlich kann das Tool über mehrere Schnittstellen an Avaloq angebunden werden, welche beide Vor- und Nachteile aufweisen (siehe Tabelle 5):

- Direkte Anbindung durch Nutzung von Avaloq Open APIs oder AMI Web Services
- Nutzung eines Roboters zur Robotergesteuerten Prozessautomatisierung (RPA)

Tabelle 5: Vor- und Nachteile der Anbindung des OrderButlers an das Kernbankensystems Avaloq mittels RPA oder AMI Web Services/Avaloq Open APIs

Primäre Anbindung des OrderButlers	Vorteile	Nachteile
Avaloq Open APIs oder AMI Web Services	<ul style="list-style-type: none"> – Direkte Anbindung an das Kernbankensystem – Kann eine hohe Anzahl von Aufträgen parallelisiert verarbeiten – Schnittstelle durch Avaloq gewartet, abhängig von der genutzten Anbindungsart (Open APIs oder AMI Webservices) 	<ul style="list-style-type: none"> – Einsatz nur für Geschäftsfälle sinnvoll, die in Avaloq abgearbeitet werden sollen – Gesamter Avaloq Stack notwendig – Ev. muss die Schnittstelle beim Erweitern der Geschäftsfälle angepasst werden – Spezifisches Avaloq API/AMI Know-how notwendig
RPA	<ul style="list-style-type: none"> – Eignet sich als Triage: Die Aufträge können im Kernbankensystem erfasst oder an anderer Stelle (z.B. an einem Empfänger gesendet) disponiert werden 	<ul style="list-style-type: none"> – Zusätzlich muss eine RPA Instanz installiert und entsprechende Lizenzkosten durch den Kunden getragen werden – Beim Erweitern der abgedeckten Geschäftsfälle muss auch der Roboter angepasst werden – Die Bank benötigt zum Unterhalt und Erweitern spezifisches RPA Know-how

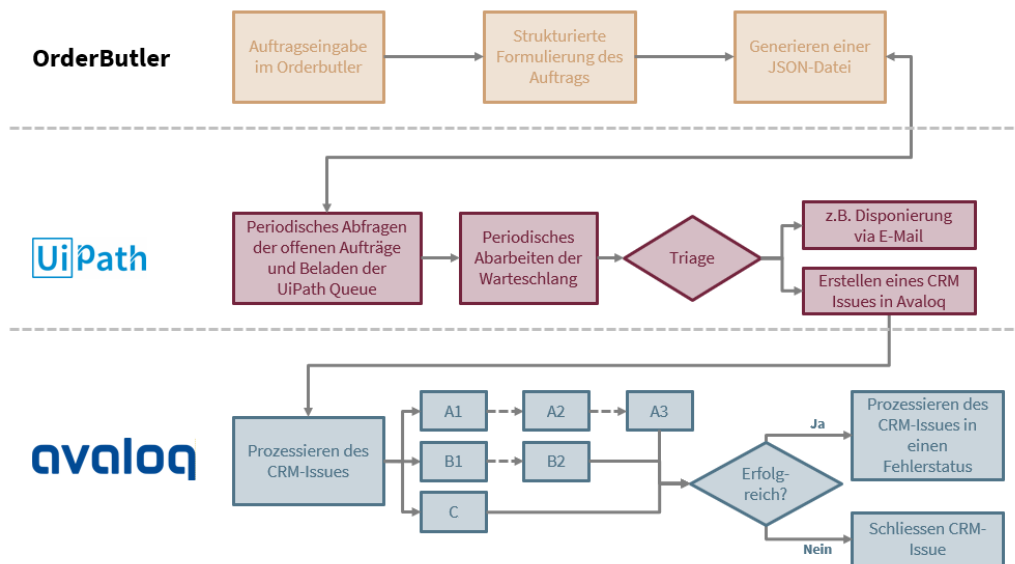
Im Projekt wurde entschieden, die Durchgängigkeit für den letzten Prototypen mit Hilfe eines RPAs zu realisieren. Insbesondere der Vorteil, den Roboter als Triage einzusetzen und komplexe Aufträge mittels E-Mail an einen entsprechenden Empfänger zu versenden, haben den Ausschlag gegeben. Zusätzlich bietet der Roboter den Vorteil, dass der Orderbutler auch an andere Kernbankensystem als Avaloq angebunden werden kann.

Die Abbildung 45 zeigt schematisch die involvierten Schritte bei der Disposition eines Auftrags. Der OrderButler ist so konzipiert, dass mittels Abfrage einer URL die Liste der offenen Aufträge im JSON-Format zur Verfügung gestellt werden. Die Roboter-Instanz stellt sicher, dass die (?) Liste periodisch geprüft und abgearbeitet wird. Bei Aufträgen, die im Kernbankensystem Avaloq abgearbeitet werden können, öffnet der Roboter jeweils eine Avaloq-Instanz, erstellt ein CRM-Issue mit allen auftragsrelevanten Informationen und prozessiert das CRM-Issue durch Verwenden einer spezifischen Aktion. Bei der Ausführung dieser Aktion wird mittels Avaloq-Scripts sichergestellt, dass die betroffenen Objekte in der korrekten Reihenfolge modifiziert und Transaktionen erstellt werden. Als Beispiel trifft der Auftrag die Saldierung eines bestimmten Kontos: In Avaloq werden folglich zuerst die aufgelaufenen Zines auf das Konto ausbezahlt, anschliessend die Zinsen und der Kontosaldo mit Hilfe einer Zahlung an den Empfänger überwiesen und zum Schluss das Konto geschlossen.

Schlägt das Prozessieren des CRM-Issues fehl (z.B. weil der zu modifizierende Container bereits durch einen offenen Order blockiert ist), kann das CRM-Issue in einen Fehlerstatus geschoben werden. Ein angepasstes Orderbuch im Kernbankensystem erlaubt es, den Mitarbeitenden den Status der Aufträge jederzeit zu prüfen und falls notwendig, die fehlenden Schritte manuell abzarbeiten.

Der Auftrag gilt im OrderButler als erledigt, sobald das CRM-Issue erfasst werden konnte. Dazu ruft der Roboter erneut einen URL auf, die das Schliessen von einzelnen Aufträgen erlaubt.

Abbildung 45: Schematische Darstellung der involvierten Schritte bei der Abwicklung eines Geschäftsfalls mit Hilfe von RPA und dem Kernbankensystem Avaloq



Der Roboter steht bei der besprochenen Implementierung der Durchgängigkeit im Zentrum. Der Vorteil liegt darin, dass bei fehlender Konnektivität zwischen OrderButler, Roboter und Avaloq, die Anfrage an das Tool und das Erfassen der CRM-Issues periodisch durchgeführt wird (z.B. alle 5-10 Minuten). Steht der OrderButler zu Beginn der automatisierten Prozesskette, muss ein entsprechender zusätzlicher Algorithmus im OrderButler programmiert werden. Dieser muss darauf ausgelegt sein, dass Aufträge bei Fehlschlägen zu einem späteren Zeitpunkt manuell oder automatisch erneut platziert werden können.

In Hinblick auf eine vollständige Durchgängigkeit muss neben der Abwicklung der Aufträge ebenfalls sichergestellt werden, dass die Stammdaten wie Kundennamen, Geschäftsbeziehungen und Adressen im OrderButler regelmäßig aktualisiert werden. Theoretisch denkbar ist die jeweilige Abfrage der relevanten Information über entsprechende APIs. Da jedoch nur eine Teilmenge von Daten für den OrderButler verwendet wird, ist die Abfrage direkt in Avaloq zeitintensiv. Deshalb empfiehlt es sich, die relevanten Daten periodisch zu exportieren und für eine schnelle Suche zu indexieren. Der Export wurde dazu mit Hilfe von «File-Reports» erstellt, die in der Tagesendverarbeitung⁴ ausgeführt werden. Die resultierenden Files werden automatisch in eine zusätzliche Datenbank geladen und anschliessend der Index erstellt.

4.7.2 Machine Learning

). Aufgrund nicht vorhandener Testdaten wurde diese Funktion nicht mit Endbenutzern getestet und ist standardmässig im Prototyp abgeschaltet.

⁴ Pilotbankfeedback hat gezeigt, dass eine einmalige Aktualisierung der Daten pro Tag die Anforderungen erfüllt.

4.8 DATENMODELL

In den folgenden Kapiteln ist das Datenmodell des OrderButlers beschrieben. Es handelt sich um UML-Klassendiagramme. Aus Gründen der Übersichtlichkeit wurde das komplette Diagramm in einzelne Bereiche gegliedert.

4.8.1 Datenmodell Geschäftsfälle (BusinessCase)

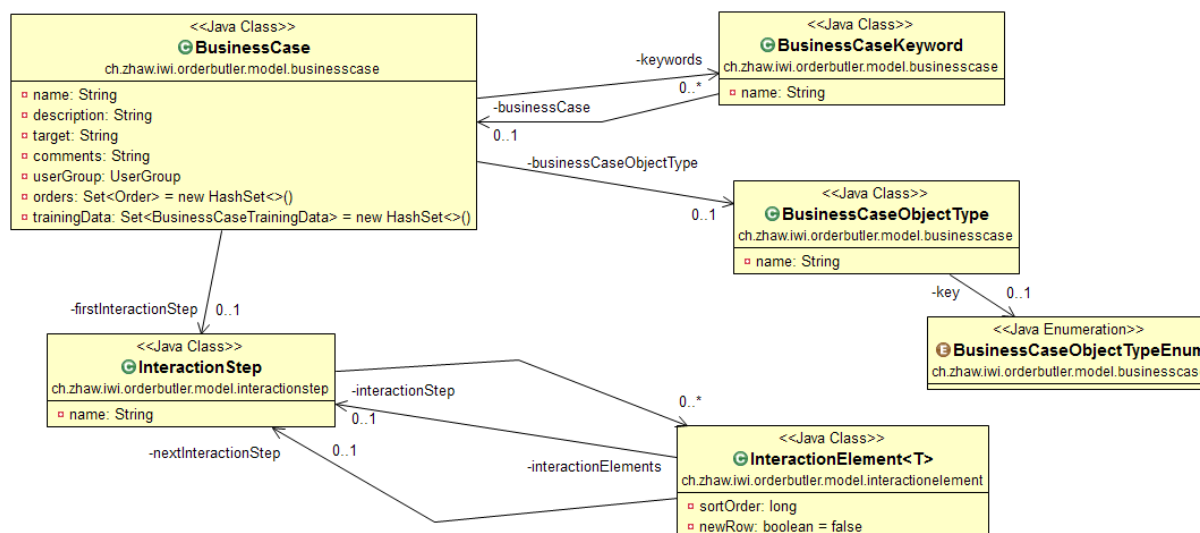


Abbildung 46 Klassendiagramm Geschäftsfälle

Ein **BusinessCase** (Geschäftsfall) ist ein zentrales Element im OrderButler. Die Endbenutzer können Geschäftsfälle abarbeiten und daraus Aufträge erstellen. Ein **BusinessCase** beginnt immer mit einem klar definierten ersten Interaktionsschritt (**InteractionStep**). Dieser wird im Datenmodell **firstInteractionStep** genannt. Jeder Schritt enthält dann eine bestimmte Anzahl von Interaktionselementen (**InteractionElement**). Diese sind im Kapitel 4.4 beschrieben.

Geschäftsfälle können vom Benutzer anhand der Attribute **name** und **description** identifiziert werden. Zusätzlich können weitere Schlüsselwörter hinterlegt werden (**BusinessCaseKeyword**).

Ein BusinessCase ist immer einem Avaloq-Geschäftsobjekt (siehe Kapitel Avaloq-Objekte im OrderButler) zugeordnet. Diese Zuordnung findet über die Klasse **BusinessCaseObjectType** sowie die Aufzählung **BusinessCaseObjectTypeEnum** statt.

4.8.2 Datenmodell Interaktionen (InteractionElement)

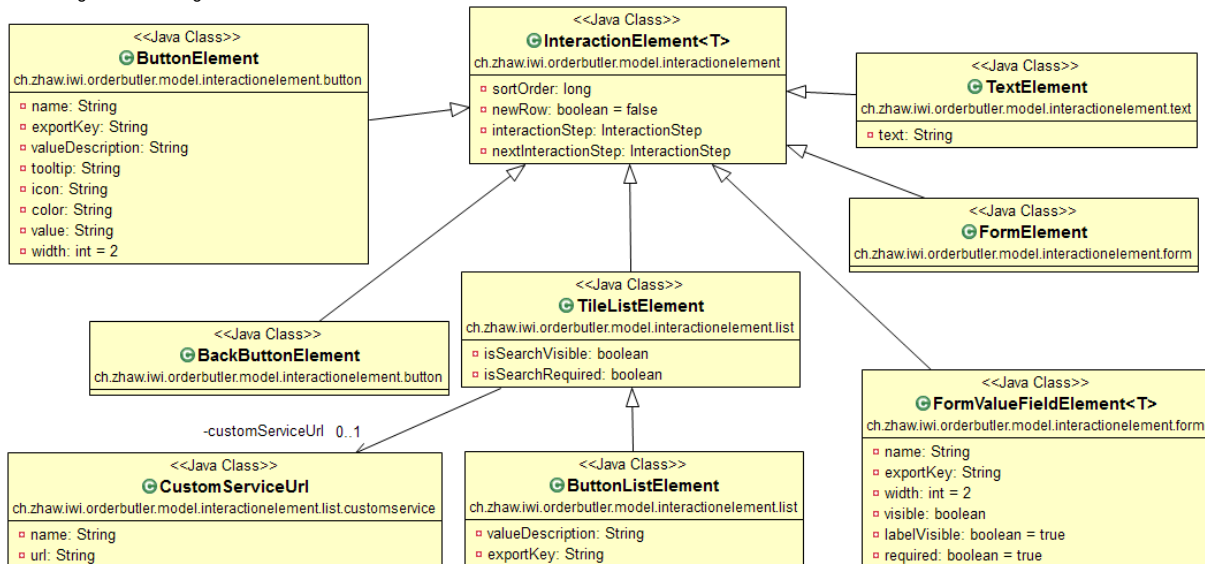
In diesem Abschnitt werden die Interaktionselemente (**InteractionElement**) beschrieben, aus denen ein einzelner Interaktionsschritt (siehe Abschnitt Datenmodell Geschäftsfälle (BusinessCase)) aufgebaut ist.

Ein einfaches Beispiel für ein Interaktionselement ist ein statischer Button (**ButtonElement**). Ebenfalls können statische Text auf der Seite dargestellt werden (**TextElement**). Das **BackButtonElement** kann nicht konfiguriert werden, es wird vom OrderButler automatisch an der passenden Stelle eingefügt um einfach zum vorhergehenden Schritt zu gelangen.

Etwas komplexere Interaktionselemente sind **TileListElement** und **ButtonListElement**. Hier werden Daten dynamisch vom Server geladen und im User Interface als Kacheln bzw. Buttons dargestellt. Der Service, welcher die Daten zur Verfügung stellt, ist in der Klasse **CustomServiceUrl** definiert. Der Benutzer kann beim Konfigurieren eines Interaktionsschrittes aus einer bestimmten, vorgegebenen Liste die passenden Server-Daten auswählen.

Eine dritte Gruppe von Interaktionselementen sind Eingabefelder, welche an Formulare verwenden. Das **FormElement** steht dem Benutzer nicht direkt zur Konfiguration zur Auswahl, wird aber verwendet um Formularelemente (**FormValueFieldElement**) auf der Seite zu gruppieren. Formularelemente sind im Abschnitt Datenmodell Eingabefelder (**FormValueFieldElement**) beschrieben.

Abbildung 47 Klassendiagramm Interaktionen



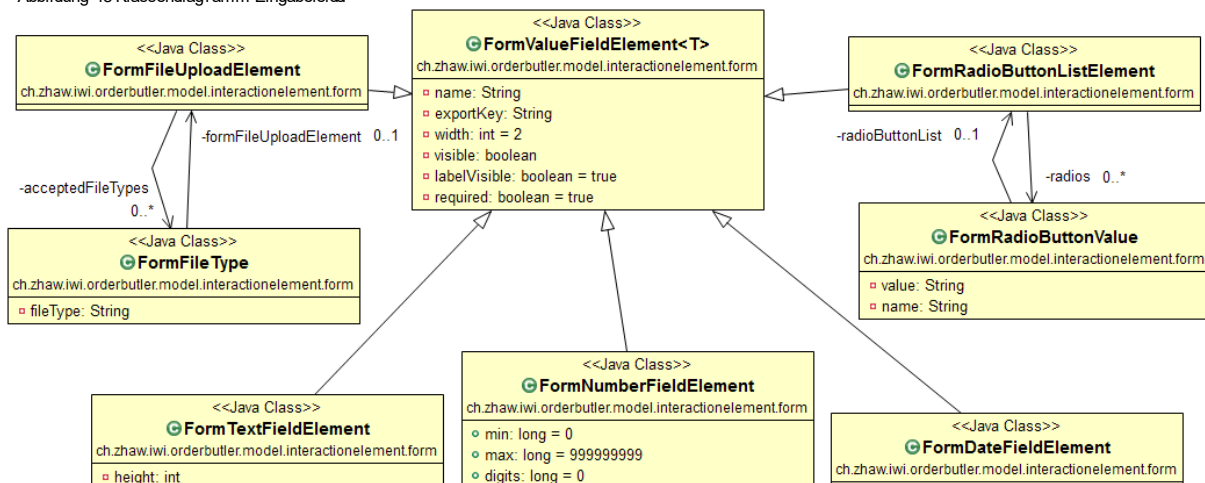
4.8.3 Datenmodell Eingabefelder (FormValueFieldElement)

Eingabefelder (**FormValueFieldElement**) sind eine bestimmte Art von Interaktionselementen (siehe Abschnitt Datenmodell Interaktionen (InteractionElement)). Bei diesen Interaktionen werden Daten über einen Touch-Befehl, einen Mausklick oder eine Tastatureingabe erfasst.

Ein **FormFileUploadElement** erlaubt das Hochladen von einer oder mehreren Dateien. Der Dateityp kann mit der Klasse **FormFileType** definiert werden. So kann beispielsweise dem Benutzer nur das Hochladen von PDF-Dateien erlaubt werden.

Mit dem **FormTextFieldElement** kann der Benutzer in diesem Interaktionsschritt freie Texte erfassen. Ein Textfeld kann einzeilig oder mehrzeilig sein. Das **FormNumberFieldElement** erlaubt die Eingabe von Zahlen. Es kann das Minimum, das Maximum sowie die Anzahl Nachkommastellen definiert werden. Das **FormDateFieldElement** ermöglicht die Auswahl eines Datums.

Abbildung 48 Klassendiagramm Eingabefelder



Das **FormRadioListElement** stellt eine Auswahl von Radio-Buttons dar. Der Benutzer kann genau einen Radio-Button auswählen. Die möglichen Radio-Buttons sind in der Klasse **FormRadioListValue** abgebildet.

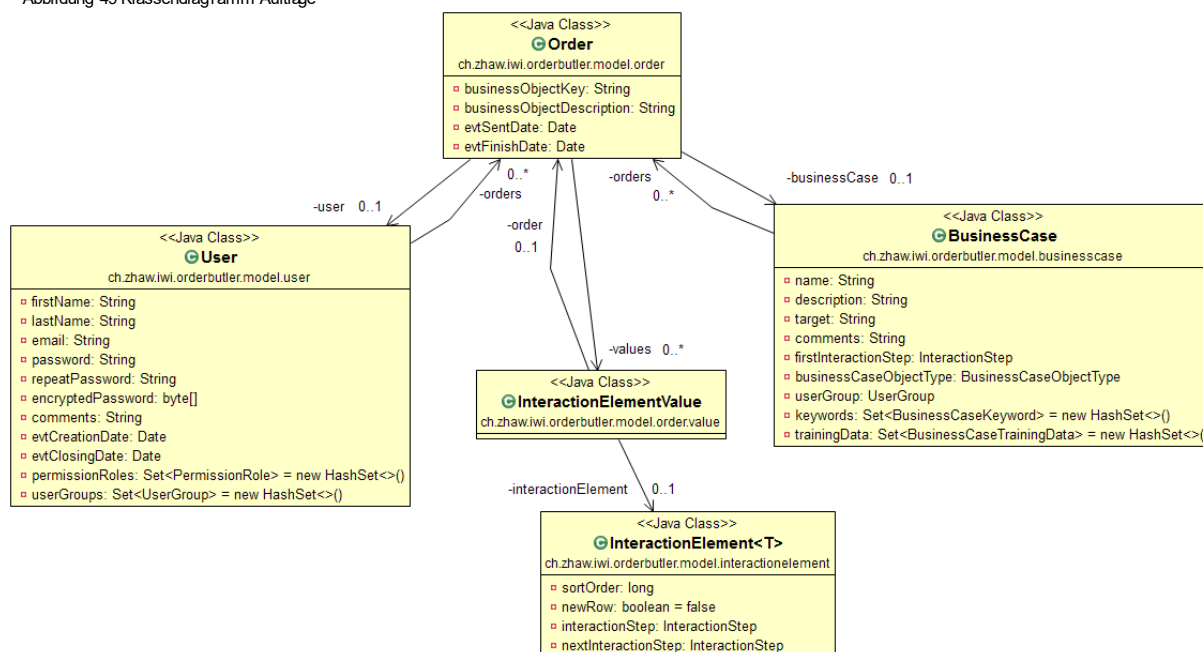
4.8.4 Datenmodell Aufträge (Order)

Aufträge (**Order**) sind die zentralen Datensätze, welche mit dem OrderButler erzeugt werden. Jeder Auftrag wird von einem Benutzer (**User**) erzeugt und bildet genau einen **BusinessCase** ab. Das Avaloq-Objekt, welches in diesem Auftrag bearbeitet wird, ist in den Feldern **businessObjectKey** (für den Export) und **businessObjectDescription** (für das User Interface) abgelegt.

Ein Auftrag ist entweder offen (**evtSentDate** und **evtFinishDate** sind leer), wurde versandt (nur **evtFinishDate** ist leer) oder fertig bearbeitet (**evtSentDate** und **evtFinishDate** sind mit Werten befüllt).

Der Inhalt eines Auftrags besteht aus einer Liste von Eingabewerten (**InteractionElementValue**). Eine Beschreibung dieser Werte findet sich in Abschnitt Datenmodell Eingabewerte (InteractionElementValue). Jeder Eingabewert gehört genau zu einem Auftrag und bezieht sich auf genau ein Interaktionselement.

Abbildung 49 Klassendiagramm Aufträge

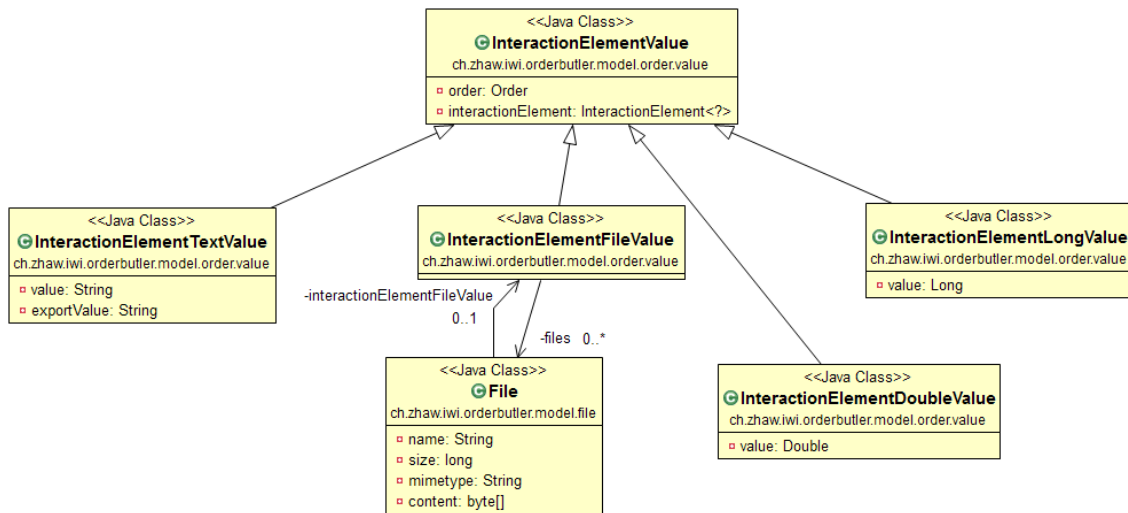


4.8.5 Datenmodell Eingabewerte (InteractionElementValue)

Eingabe-Werte sind grundsätzlich vom Typ des Interaktions-Elementes abhängig. Es gibt die drei Grundvarianten Text (**InteractionElementTextValue**), Ganzzahl (**InteractionElementLongValue**) und Fließkommazahl (**InteractionElementDoubleValue**).

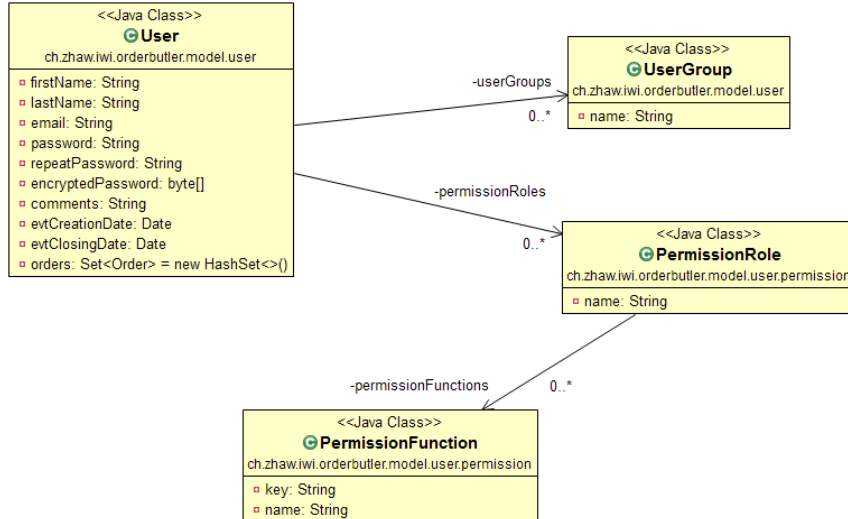
Dateien werden mit der Klasse **InteractionElementFileValue** abgelegt. Diese verweist auf die Klasse **File**, wo die eigentliche Datei im Attribut **content** gespeichert ist.

Abbildung 50 Klassendiagramm Eingabewert



4.8.6 Datenmodell Benutzer (User)

Abbildung 51 Klassendiagramm Benutzer



Für den Prototypen wurde eine einfache Benutzerverwaltung umgesetzt. Die Klasse **User** beinhaltet alle Benutzer mit verschlüsseltem Passwort sowie E-Mail-Adresse als Benutzername. Ein Benutzer kann einer oder mehreren Benutzergruppen (**UserGroup**) zugeordnet werden. Benutzergruppen können definiert werden, um das Ausführen der mit dem OrderButler erstellten Aufträge auf mehrere Personen zu verteilen.

Jeder Benutzer kann mehrere **PermissionRole** besitzen, welche ihrerseits mehrere **PermissionFunction** gehören. Im Prototyp ist nur die Rolle Administrator definiert, welche die einzige Funktion „Administrations-Sicht anzeigen“ besitzt.

5 Fazit und Ausblick

5.1 ZUSAMMENFASSUNG DER ERGEBNISSE

Die grösste Herausforderung des Projektes bestand darin, die effektivste Dialogstruktur für die Erfassung von Aufträgen aus Perspektive der Zielgruppe im Front-Office einer Bank zu identifizieren. Das human-zentrische Vorgehen auf der Grundlage von drei Prototypen verhinderte eine rein technologische Betrachtung und eröffnete einen Lösungsansatz, der konsequent die Anforderungen nach Geschwindigkeit, Einfachheit und Transparenz (siehe Kapitel 3.2) umsetzte.

User Interface

Das hybride User Interface bestehend aus Kacheln sowie Eingabeelementen hat sich in den End-User-Tests bewährt und stiess auf positives Feedback. Alle Testbenutzer konnten den Prototypen ohne Schulung nach einer kurzen Einführung bedienen. Die contextual inquiries im Vorfeld (auf der Grundlage der Eingabemasken (GUIs) der Fachapplikationen) und bestehenden Formulare) sowie die Iterationen mit verschiedenen Interaktionselementen (Freitext, Kacheln, Listen, Forms) ermöglichten es Stärken, Schwächen und Einsatzfelder der verschiedenen Dialogmuster aus der Perspektive des Users zu positionieren und führten zum hybriden Ansatz (Abbildung 52).

Abbildung 52: Positionierung des hybriden Lösungsansatzes



Im Rahmen der Testings zeigten sich wichtige Erkenntnisse zur Gestaltung des User Interfaces der hybriden Lösung:

- Einheitliche Farben und Begriffe sind wichtig (insbesondere dann wenn die Prozesse inkl. Farben und Texte durch die Benutzer selbst konfiguriert werden)
- Die Umsetzung als Touch-Applikation mittels Kacheln (Verzicht auf rechte Maustaste) war positiv
- Scrolling auf dem Screen sollte vermieden werden (muss bei der Konfiguration eigener Prozesse beachtet werden)

Benutzerführung

Eine Applikation in diesem Umfeld sollte den Benutzer weder zu stark einschränken noch zu viele Freiheiten lassen. Der OrderButler verfolgt hier einen Mittelweg und lässt die Benutzer entscheiden, ob sie beispielsweise mittels Geschäftsfall oder Kunde starten wollen. Besonders positiv erwähnt wurden die Suchmöglichkeiten innerhalb des OrderButlers (Volltextsuche). Diese steht in typischen Kernbankenapplikationen oftmals nicht zur Verfügung.

Anpassung & Erweiterbarkeit

Bei der Erstellung des Datenmodells wurde von Anfang an auf eine spätere Konfigurierbarkeit geachtet. Neue Prozesse lassen sich zum OrderButler hinzufügen und auch anpassen. Dies erfolgt über ein einfaches User Interface mittels Listen und Formularen. Obwohl die Testbenutzer damit zurechtkamen, liessen sich hier bestimmt noch Optimierungen anbringen, z.B. wäre eine Darstellung eines Prozesses und seiner Varianten mittels Flussdiagrammen sinnvoll oder es liesse sich ein grafischer Editor (Drag and Drop) realisieren. Für beides Erweiterungen kann aber das bestehende Datenmodell weiterverwendet werden.

Systemarchitektur

Die Systemarchitektur war für einen Prototypen vollkommen ausreichend und passend für das Avaloq-Umfeld der Pilotbank. Natürlich kann eine produktive Version auf anderen Technologien umgesetzt werden, dabei können die Erkenntnisse des Prototypen (User Interface, Datenmodell, Services) vom Konzept her weiterverwendet werden.

Prototyp vs. produktive Version

Das Ziel des Innovationsprojektes war es, einen funktionsfähigen Prototypen zu erstellen. Dieser umfasst das User Interface, das Datenmodell sowie alle notwendigen Services, um die Aufträge an ein nachfolgendes System weiterzuleiten (RPA und/oder manuelle Verarbeitung).

In einer produktiven Version sind folgende Punkte noch zu definieren:

- Versionierung der Prozesse: Prozesse werden sich im Laufe der Zeit ändern. Deshalb ist es notwendig, bestehende Prozesse zu deaktivieren und eine neue Version freizuschalten
- Testing: Neue Prozesse sollen getestet werden können, bevor sie für alle Benutzer freigeschaltet werden. Dies kann auf einem Test-System geschehen, noch besser wäre aber eine Lösung welche eine Freischaltung eines Prozesses nur für bestimmte, erweiterte Benutzer erlaubt
- Benutzerverwaltung: Die Applikation sollte über bankenübliche Sicherheitsstandards verfügen und die Benutzer und Rechte direkt aus einem zentralen System beziehen
- Archivierung: Mit der Zeit werden im System mehr und mehr Aufträge erfasst. Es muss eine Lösung gefunden werden, diese zu Archivieren

5.2 NÄCHSTE SCHRITTE: UMSETZUNGSPLANUNG

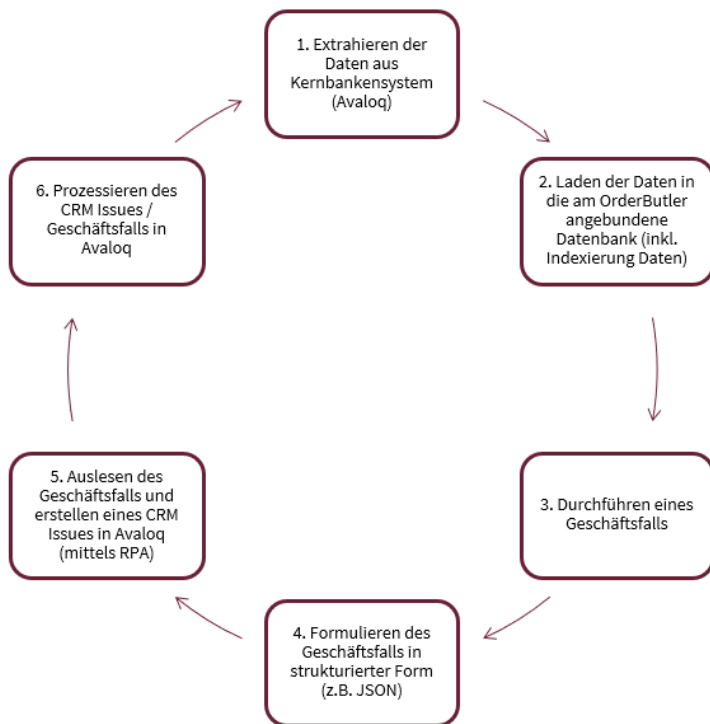
Um den Prototypen zu einem eigenständigen Produkt zu entwickeln und im Markt zu platzieren, hat der Umsetzungspartner Synpulse im Rahmen des Projektabschlusses einen Umsetzungsplan erarbeitet. Um den Prototypen zur Marktreife weiterzuentwickeln, muss die Funktionalität des Prototypen und die involvierte Prozesskette (Abbildung 53):

- Prozesserweiterung E2E. Nebst der Abwicklung eines erfassten Geschäftsfalls muss sichergestellt werden, dass die im Tool vorhanden Daten periodisch aktualisiert werden. Dazu müssen die Daten aus dem Kernbankensystem periodisch extrahiert und in die Datenbank des Tools geladen sowie indiziert werden.
- Geschäftsfallumfang. Der OrderButler umfasst einen minimalen Grundstock an Geschäftsfällen und Automatisierungen (hinsichtlich Disposition und automatischer Abwicklung). Dieser Grundstock muss erweitert und die Anzahl automatischer Prozesse erhöht werden.
- Erweiterung und Flexibilisierung des User Designs. Potentielle Kunden (insbesondere Banken) erwarten, dass das Design eingängig ist und an die UI/UX-Richtlinien der Bank angepasst sind. Dazu sollen bereits

vorhandene GUI weiter verfeinert werden und es muss sichergestellt werden, dass einzelne Komponenten einfach und zeitsparend angepasst werden können.

- Stabilität. Der Prototyp wurde bereits im Rahmen des Projekts mit End-Usern getestet und rapportierte Fehler wurden mehrheitlich behoben. Für ein marktreifes Produkt ist ein systematisches Testen jeglicher Funktionalität und das Lösen aller relevanten Fehler notwendig.

Abbildung 53: Involvierte Prozessschritt



Unter Berücksichtigung der genannten Punkte, die für eine Reifung des Prototypen zu einem Produkt notwendig sind, soll der Prototyp in den Monaten nach Ende des Innovationsprojektes kontinuierlich weiterentwickelt werden (Abbildung 54).

Abbildung 54: Umsetzungsplan zur Entwicklung des Prototypen von einem Produkt

Tätigkeit	Febr.		März					Apr.					Mai					Juni					Juli							
	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
Projektleitung	[Bar chart showing continuous activity from Feb 07 to Jul 30]																													
Anpassung UI/UX (inkl. Trennung)	[Bar chart showing activity from Mar 10 to May 22]																													
Automatische Datenaktualisierung	[Bar chart showing activity from Feb 09 to Mar 13]																													
Erweiterung Geschäftsfälle OrderButler	[Bar chart showing activity from Mar 10 to May 17]																													
Erweiterung der automatischen Verarbeitung	[Bar chart showing activity from Mar 10 to May 17]																													
Testing Orderbutler	[Bar chart showing activity from Jun 23 to Jun 27]																													
Testing E2E (inkl. Integration)	[Bar chart showing activity from Jul 28 to Jul 30]																													
Bugfixing	[Bar chart showing activity from Jun 23 to Jul 30]																													

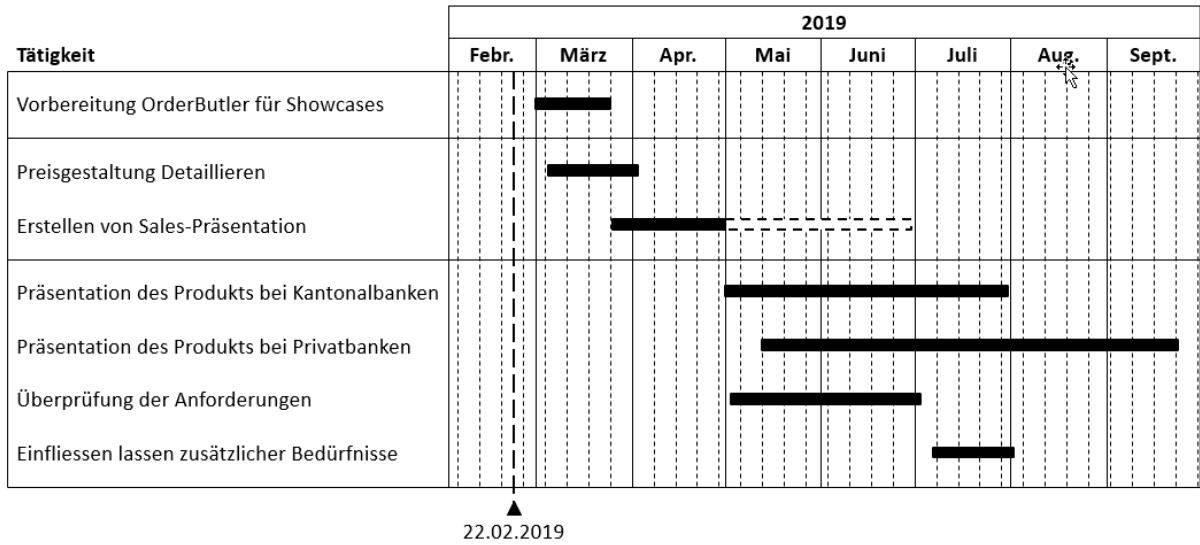
▲
22.02.2019

Ziel des Umsetzungspartners Synpulse ist es Produkte stets «Client Centric» zu entwickeln, das heisst die Bedürfnisse des Kunden im Entwicklungsprozess eines Produktes rechtzeitig zu erkennen und in die Entwicklung einfließen zu lassen. Durch die Zusammenarbeit mit der St. Galler Kantonalbank (SGKB) als Pilotumsetzungspartnerin im Rahmen des Forschungsprojektes konnte die Möglichkeit einer kundennahen Entwicklung bereits genutzt

werden. Als Bestandteil der Markteinführung und Reifung des Produkts ist das Prüfen der Bedürfnisse weiterer potenzieller Kunden geplant.

Dazu soll der Prototyp soweit konfiguriert werden, dass er im Rahmen von Präsentationen bei weiteren Kantonalbanken und anschliessend bei Privatbanken gezeigt werden kann. Abbildung 55 zeigt einen groben Plan zur Präsentation des Prototypen bei potenziellen Kunden in den kommenden Monaten.

Abbildung 55: Markteinführung



Idealerweise kann die Entwicklung des Produkts in Zusammenarbeit mit einem Kunden abgeschlossen werden.

Literaturverzeichnis

- Allweyer, T. (2016). *Robotic Process Automation – Neue Perspektiven für die Prozessautomatisierung*. <http://www.kurze-prozesse.de/2016/11/10/paper-zum-download-jetzt-kommen-die-roboter-und-automatisieren-die-prozesse/>
- Berg, M. (2013). Natürlichsprachlichkeit in Dialogsystemen. *Informatik-Spektrum*, 36(4), 371–381. <https://doi.org/10.1007/s00287-012-0650-3>
- Berg, M., & Düsterhöft, A. (2010, Januar 1). Website Interaction with Text-based Natural Language Dialog Systems. 7. *Wismarer Wirtschaftsinformatiktage*.
- Chai, J., Horvath, V., Nicolov, N., Stys, M., Kambhatla, N., Zadrozny, W., & Melville, P. (2002). Natural Language Assistant: A Dialog System for Online Product Recommendation. *AI Magazine*, 23(2), 63–63. <https://doi.org/10.1609/aimag.v23i2.1641>
- DIN-ISO. (2011). *Ergonomie der Mensch-System-Interaktion—Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010); Deutsche Fassung EN ISO 9241-210:201*. Beuth. <https://www.beuth.de/de/norm/iso-9241-210/127800493>
- Dodge, J., Gane, A., Zhang, X., Bordes, A., Chopra, S., Miller, A., Szlam, A., & Weston, J. (2015). Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. *arXiv:1511.06931 [cs]*. <http://arxiv.org/abs/1511.06931>
- Gartner Research. (2019). Gartner Says Worldwide Robotic Process Automation Software Market Grew 63% in 2018. *Gartner*. <https://www.gartner.com/en/newsroom/press-releases/2019-06-24-gartner-says-worldwide-robotic-process-automation-sof>
- Giacomin, J. (2014). What Is Human Centred Design? *The Design Journal*, 17(4), 606–623. <https://doi.org/10.2752/175630614X14056185480186>
- Holtzblatt, K., & Beyer, H. (2016). *Contextual Design: Design for Life* (2 edition). Morgan Kaufmann.
- IEEE Std. 2755-2017. (2017). *IEEE Guide for Terms and Concepts in Intelligent Process Automation* (Standard IEEE Std. 2755-2017). IEEE Standards Association. <https://standards.ieee.org/standard/2755-2017.html>
- Ivančić, L., Suša Vugec, D., & Bosilj Vukšić, V. (2019). Robotic Process Automation: Systematic Literature Review. In C. Di Ciccio, R. Gabryelczyk, L. García-Bañuelos, T. Hernaus, R. Hull, M. Indihar Štemberger, A. Kö, & M. Staples (Hrsg.), *Business Process Management: Blockchain and Central and Eastern Europe Forum* (S. 280–295). Springer International Publishing.
- Lacity, M., Willcocks, L., & Craig, A. (2015). Robotic process automation at Telefonica O2. *The Outsourcing Unit Working Research Paper Series*, 15/02, 19.
- Le Clair, C., Cullen, A., & King, M. (2017). *The Forrester Wave™: Robotic Process Automation, Q1 2017. The 12 Providers That Matter Most And How They Stack Up* (The Forrester Wave). Forrester Research Inc. <https://www.forrester.com/report/The+Forrester+Wave+Robotic+Process+Automation+Q1+2017/-/RES131182>
- Möller, S. (2017). Usability Engineering. In S. Möller (Hrsg.), *Quality Engineering: Qualität kommunikationstechnischer Systeme* (S. 59–76). Springer. https://doi.org/10.1007/978-3-662-56046-4_4
- Pazos, R. A., González, J. J., Aguirre, M. L., F, J. A. M., & H, H. J. F. (2013). Natural Language Interfaces to Databases: An Analysis of the State of the Art. In O. Castillo, P. Melin, & J. Kacprzyk (Hrsg.), *Recent Advances on Hybrid Intelligent Systems* (S. 463–480). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33021-6_36

- Perez-Marin, D., & Pascual-Nieto, I. (2011). *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices* (New.). Information Science Reference.
- Ritter, A., Cherry, C., & Dolan, W. B. (2011). Data-driven Response Generation in Social Media. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 583–593. <http://dl.acm.org/citation.cfm?id=2145432.2145500>
- Scheppler, B., & Weber, C. (2020). Robotic Process Automation. *Informatik Spektrum*. <https://doi.org/10.1007/s00287-020-01263-6>
- Someren, M. W. van, Barnard, Y. F., & Sandberg, J. A. C. (1994). *The think aloud method: A practical guide to modelling cognitive processes*. Academic Press.
- Sordani, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., & Dolan, B. (2015). A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *arXiv:1506.06714 [cs]*. <http://arxiv.org/abs/1506.06714>
- Strategyzer. (o. J.). *Value Proposition Canvas: A Tool To Understand What Customers Really Want*. Abgerufen 20. Juni 2021, von <https://www.strategyzer.com/blog/value-proposition-canvas-a-tool-to-understand-what-customers-really-want>
- UBS. (2016). The future of banking as a cognitive business. *innovation*. <https://www.ubs.com/magazines/innovation/en/into-the-future/2016/the-future-of-banking-as-a-cognitive-business.html>
- Willcocks, L., Lacity, M., & Craig, A. (2015). *Robotic process automation at Xchanging*.

Tabellenverzeichnis

Tabelle 1 Nutzenversprechen und Risiken des RPA Einsatzes in Anlehnung an (Schepler & Weber, 2020)	10
Tabelle 2 Verwendete Technologien	18
Tabelle 3 Beschreibung der Interaktions-Elemente	29
Tabelle 4 Eigenschaften eines Interaktions-Elementes	30
Tabelle 5: Vor- und Nachteile der Anbindung des OrderButlers an das Kernbankensystems Avaloq mittels RPA oder AMI Web Services/Avaloq Open APIs	41

Abbildungsverzeichnis

Abbildung 1 Ist- und Soll-Situation	8
Abbildung 2 Zielsetzung und Projekthalte	8
Abbildung 3: Humanzentrisches Vorgehen: Elemente und ihre Anwendung im Projekt	11
Abbildung 4: Geplantes Vorgehen: Inhalte und Ziele der Prototypen	12
Abbildung 5: Die projektrelevanten Geschäftsfälle und ihre Häufigkeit	14
Abbildung 6: 5 Profile aus der Erhebung der Nutzerperspektive	14
Abbildung 7: Customer Value Proposition Canvas für das Dialogsystem	15
Abbildung 8 Angepasstes Vorgehen für das Prototyping auf der Basis von User Stories	16
Abbildung 9: Beispiele von Usability-Testszenarios	17
Abbildung 10 Funktionsweise des Prototyps	20
Abbildung 11 Vereinfachtes Datenmodell wichtiger Avaloq-Entitätstypen	21
Abbildung 12 Einstieg in den OrderButler	21
Abbildung 13: Beispiele für Einstiegspfade	22
Abbildung 14 Suchresultate für den Namen "Kobler"	23
Abbildung 15 Vom gewählten Businesspartner abhängige Avaloq-Objekte	23
Abbildung 16 Passende Geschäftsfälle zu Businesspartner	23
Abbildung 17 Geschäftsfälle eines Avaloq-Containers	24
Abbildung 18 Interaktionsschritt mit Antwort-Buttons	24
Abbildung 19 Interaktionsschritt mit Eingabefeld	25
Abbildung 20 Hochladen mehrerer Dateien in den OrderButler	25
Abbildung 21 Zusammenfassung des Auftrages vor dem Versenden	25
Abbildung 22 Zusammenhang zwischen Geschäftsfall, Interaktionsschritt und Interaktionselement	26
Abbildung 23 Maske "Geschäftsfall"	26
Abbildung 24 Weitere Konfiguration eines Geschäftsfalles	27
Abbildung 25 Alternative Stichwort für die Volltextsuche	27
Abbildung 26 Von einem Geschäftsfall verwendete Interaktionsschritte	27
Abbildung 27 Liste der Elemente eines Interaktionsschrittes	28
Abbildung 28 Zusammenhang zwischen Interaktionselementen und Ansicht des KuBe	28
Abbildung 29 Verschiedene Interaktions-Elemente	29
Abbildung 30 Maske zur Erfassung eines Interaktions-Elementes	30
Abbildung 31 Maske "Benutzer"	31
Abbildung 32 Liste der Geschäftsfälle eines Businesspartners	31
Abbildung 33 Machine Learning	32
Abbildung 34 Startbildschirm	33
Abbildung 35 Erfassen einer Situation	34
Abbildung 36 Auswahl eines Geschäftsfalles	34
Abbildung 37 Ausfüllen des digitalen Formulars	35
Abbildung 38: Szenario Sparen 3	35
Abbildung 39 Startbildschirm Prototyp I	36
Abbildung 40 Szenario Sparen 3: Auswahl von Positionen oder Containern der gewählten Person	37
Abbildung 41 Szenario Sparen 3: Vorgeschlagene Geschäftsfälle	37
Abbildung 42: Szenario 3: Auswahl des Auszahlungskontos	38
Abbildung 43 Szenario Sparen 3: Vervollständigen des Auftrages	38
Abbildung 44 Szenario Sparen 3: Bestätigung des Auftrages	39

Abbildung 45: Schematische Darstellung der involvierten Schritte bei der Abwicklung eines Geschäftsfalls mit Hilfe von RPA und dem Kernbankensystem Avaloq	42
Abbildung 46 Klassendiagramm Geschäftsfälle	43
Abbildung 47 Klassendiagramm Interaktionen	44
Abbildung 48 Klassendiagramm Eingabefelder	44
Abbildung 49 Klassendiagramm Aufträge	45
Abbildung 50 Klassendiagramm Eingabewerte	46
Abbildung 51 Klassendiagramm Benutzer	46
Abbildung 52: Positionierung des hybriden Lösungsansatzes	47
Abbildung 53: Involvierte Prozessschritte	49
Abbildung 54: Umsetzungsplan zur Entwicklung des Prototypen von einem Produkt	49
Abbildung 55: Markteinführung	50

School of Management and Law

St.-Georgen-Platz 2
Postfach
8401 Winterthur
Schweiz

www.zhaw.ch/sml



AACSB
ACCREDITED

swissuniversities