

Gruppe 2

Pascal Brunner (brunnpa7)
Maximilian König (koenimax)
Martin Ponbauer (ponbamar)
Aurel Schwitter (schwiaur)
Lucca Willi (willilu1)

Aufgabe 14 – Kapitel 1, 2 und 3 der Architekturdokumentation erstellen

Aufgabe 1 - Einführung und Ziele

Dokumentieren Sie gemäss dem Fallbeispiel Auktionsplattform:

- Aufgabenstellung,
- Qualitätsziele,
- eine Kurzfassung der architekturelevanten Anforderungen (insb. die nichtfunktionalen),
- Stakeholder.

Die Grundlagen können Sie aus Ihrer Anforderungsdokumentation übernehmen (Copy-Paste).
Erweitern Sie ggf. die nichtfunktionalen Anforderungen.

Aufgabe 2 – Randbedingungen

Erstellen Sie die Leitplanken für die Entwurfsentscheidungen hinsichtlich:

- Infrastruktur: Container (Docker, Kubernetes, OpenShift, Rancher)
- Technologie und Frameworks (FrontEnd Angular, Backend Spring Boot)
- Gesetzliche Grundlagen (Datenschutz1, Kreditkarten2)
- Weitere Randbedingungen

Aufgabe 3 - Kontextabgrenzung

In welchem fachlichen und/oder technischen Umfeld soll das System arbeiten? Nehmen Sie dazu das untenstehende Kontextdiagramm (Core-System) und analysieren Sie wie die Schnittstellen von externen Payment Organisationen aufgebaut sind. Sie können dazu die Dokumentation von DataTrans <https://docs.datatrans.ch/docs/getting-started-home> und Saferpay <https://www.six-payment-services.com/en/site/e-commerce/solutionspaymentsolution.html> (Developer <https://www.six-payment-services.com/en/site/e-commerce-developer.html>) vergleichen (für DataTrans existiert ein ZHAW Testaccount mit Testkreditkarten). Der verwendete Paymentprozessor soll parametrierbar sein.

- Welche Kreditkartenprozesse müssen unterstützt werden?
- Möchten Sie Customer Initiated3 oder Merchant Initiated Payments unterstützen?
- Welche zusätzlichen architekturelevanten Anforderungen ergeben sich durch diese Analyse?

Einführung und Ziele

Aufgabenstellung

Das Start-up-Unternehmen „Cool Software Solutions“ ist im Bereich E-Commerce-Softwarelösungen tätig und hat seit einiger Zeit erfolgreich eine Online-Shop-Software entwickelt und vermarktet. Der Online-Shop ist schon an einige Kunden verkauft worden und in Betrieb. Die Online-Shop-Software kann als Produkt auf eigener Plattform (On-Premises) oder als Software-as-a-Service (SaaS) aus der Cloud betrieben werden. Das Start-up plant nun ihre E-Commerce-Suite, um ein neues Modul für Online-Auktionen zu erweitern. Ein Kunde der Online-Shop-Software - der Grosshändler „Big Sales“ - interessiert sich schon für dieses neue Modul. Der Grosshändler will mit einer Online-Auktionsplattform den Verkauf von Restposten, Sonderposten und defekter Ware effizienter und rascher durchführen können. Heute müssen die Kunden - Wiederverkäufer, Einzelhändler etc. - für den Verkauf von Lagerrestbeständen per Mailing angefragt und der Preis jeweils individuell ausgehandelt werden. Es wurde mit dem Grosshändler vereinbart, dass eine erste Version des neuen Moduls für Online-Auktionen mit ihm zusammen entwickelt wird (Pilotkunde). Nach erfolgreicher Realisierung und erstem Betrieb der Online-Auktionsplattform beim Grosshändler, soll dieses Modul dann die E-Commerce-Suite des Start-ups ergänzen und zusätzlich zum Online-Shop auch anderen Kunden angeboten werden können.

Qualitätsziele

Qualitätsziel	Motivation und Erläuterung
Verständliche Architektur (Analysierbarkeit)	Eine passende und einfache Architektur erleichtert den Einstieg und verbessert die Analysierbarkeit des gesamten Codes.
Lose Kopplung (Änderbarkeit)	Der Code wird in Komponenten aufgeteilt und mittels Interfaces verbunden. So erreicht man ein lose-gekoppeltes System, bei welchem die Komponenten einfach ausgetauscht werden können.
Standards verwenden (Interoperabilität)	Die Auktionsplattform soll mit Standard Technologien erstellt werden.
Ansprechendes Design (Attraktivität)	Die Auktionsplattform soll ein Ansprechendes und Modernes Design besitzen, damit die Kunden zum Kaufen und Verkaufen angeregt werden.
Schneller Kauf (Effizienz)	Die Auktionsplattform soll die Kaufabwicklung schnell durchführen können.

Nichtfunktionale Anforderungen

Anforderung	Motivation und Erläuterung
Usability	Die Bedienung der Applikation soll ohne Instruktionen oder Handbüchern möglich sein.
Performance	Die Applikation soll auf Anfragen innert maximal 500ms Antworten liefern.
Zuverlässigkeit	Die Applikation soll bei Abstürzen von Hardware / Software weiterlaufen können.
Skalierbarkeit	Die Applikation soll automatisch skalierbar sein, um Kosten zu sparen sowie Zugriffsspitzen standhalten zu können.

Stakeholder

Name	In Kontakt mit	Rolle
Geschäftsleitung	Product Owner	Steuerungsausschuss
Product Owner	Alle	Schnittstelle zum Kunden, Projekt Gesamtverantwortlicher
Scrum Master	Product Owner, Lead-Entwickler, Entwickler	Verantwortlich für die erfolgreiche Entwicklung und Einhaltung der Scrum-Prinzipien
Business Analyst	Alle	Aufnahme und Formulierung der Anforderung
Lead Entwickler	Product Owner, Business Analyst	Verantwortlich für die Architektur der Lösung, technische Entscheide
Entwickler	Lead Entwickler, Product Owner	Entwicklung der Lösung
Grosshändler (Big Sales)	Projektleiter	Kunde / Abnehmer der Auktionsplattform
Auktionsteilnehmer	Grosshändler	Kunde des Kunden

Randbedingungen

Technische Randbedingungen

Randbedingung	Erläuterungen, Hintergrund
Einfache Hardwareausstattung	Der Aufruf der Webseite sollte von möglichst vielen Geräten möglich sein. Darunter zählen zum Beispiel auch schwächere mobile Geräte.
Betrieb des Backends	Das Hosting erfolgt mittels Docker und einem self-hosted Kubernetes Cluster.
Implementierung des Backends	Vorgegeben durch die vorgehenden Projekte wird Java mit Spring Boot verwenden.
Implementierung des Frontends	Vorgegeben durch andere Projekte wird Angular 11 verwendet.

Organisatorische Randbedingungen

Randbedingung	Erläuterungen, Hintergrund
Team	Pascal Brunner (brunnpa7) Maximilian König (koenimax) Martin Ponbauer (ponbamar) Aurel Schwitter (schwiaur) Lucca Willi (willilu1)
Zeitplan	Start des Projekts in Kalenderwoche 47, Abschluss vor Ende Jahr 2020.
Vorgehensmodell	Entwicklung risikogetrieben, iterativ und inkrementell. Zur Dokumentation der Architektur kommt das Template «arc42» zum Einsatz. Eine Architekturdokumentation gegliedert nach dieser Vorlage ist zentrales Projektergebnis.
Entwicklungswerkzeuge	Entwurf mit Stift und Papier, ergänzend Figma für Prototyping. Arbeitsergebnisse zur Architekturdokumentation gesammelt im Word Dokumenten. Erstellung der Java-Quelltexte in IntelliJ.
Konfigurations- und Versionsverwaltung	Git bei GitHub.
Testwerkzeuge und -prozesse	JUnit sowohl für inhaltliche Richtigkeit als auch für Integrationstests und die Einhaltung von Effizienzvorgaben.

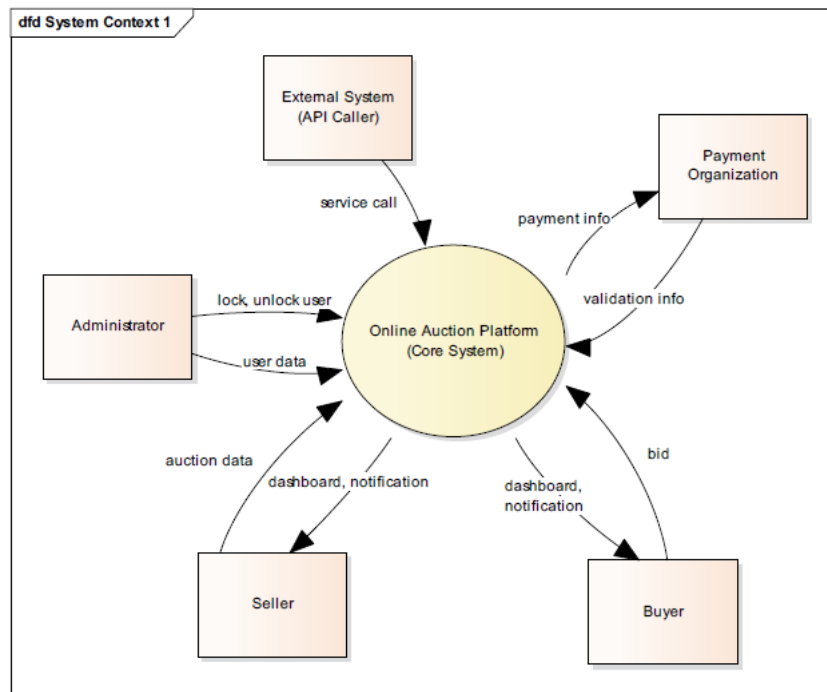
Konventionen

Konvention	Erläuterungen, Hintergrund
Architekturdokumentation	Terminologie und Gliederung nach dem deutschen arc42-Template in der Version 7.0
Kodierrichtlinien für Java	Java Coding Conventions von Microsoft, geprüft mit Hilfe von SonarQube und bei lokaler Entwicklung mit Unterstützung des SonarLint-Plugins
Sprache (Deutsch vs. Englisch)	Benennung von Dingen (Komponenten, Schnittstellen) in Diagrammen und Texten innerhalb dieser (deutschen) arc42-Architekturdokumentation in Deutsch. Verwendung englischer Bezeichner für Klassen, Methoden etc. im Java-Quelltext (es sei denn, die Java-Kodierrichtlinien stehen dem im Wege).

Datenschutz	Datenschutz gemäss Bundesgesetz über den Datenschutz (siehe dazu DSG)
Kreditkarten / Payment-Provider	Der Zahlungsprozess sollte gemäss Industriestandards durchgeführt werden. Dabei sollte der PCI DSS (Payment Card Industry Data Security Standard) Standard verwendet werden.

Kontextabgrenzung

Fachlicher Kontext



Menschlicher Benutzer (Administrator, Seller, Buyer)

Die Auktionsplattform basiert auf dem System, dass es Käufer und Verkäufer gibt. Ein weiterer Benutzer ist der Administrator, welcher die Benutzer verwalten und mutieren kann.

Technische Systeme (External System, Payment Organization)

Das System kommuniziert mit verschiedenen Fremdsystemen, darunter Beispielsweise ein externes Loginsystem sowie ein Bezahlungssystem einer Zahlungsorganisation.

Technischer- oder Verteilungskontext

API (External System)

Externe Systeme sind im Grundsatz technischer Natur und interagieren mit dem System über von uns angegebenen Schnittstellen (API).

Datatrans Payment (Payment Organizations)

Für die Zahlungsart werden „Customer Initiated Payments“ verwendet – der Kunde klickt einen „Bezahlen“-Knopf und leitet damit Zahlung ein.

Für die Zahlung wird der Benutzer auf die Seite von Datatrans / Saferplay umgeleitet, wo er seine Zahlungsdaten eingibt. In der Datenbank sollen keine Kreditkartendaten gespeichert werden.