

Konzeptbericht

Status	In Arbeit / In Prüfung / Abgeschlossen
Projektname	<Projektname>
Projektleiter	<Projektleiter>
Auftraggeber	<Auftraggeber>
Autoren	<Autor1> <Autor2>
Verteiler	<Verteiler>

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle

Definitionen und Abkürzungen

Begriff / Abkürzung	Bedeutung

Referenzen

Referenz	Titel, Quelle
[1]	
[2]	
[3]	

Inhaltsverzeichnis

1	Zusammenfassung	4
2	Systemanforderungen	4
2.1	Anforderungen an die Funktionalität.....	4
2.2	Anforderungen an die Informationssicherheit und den Datenschutz.....	4
3	Systemarchitektur.....	5
3.1	Gliederung der Lösung	5
3.2	Technische Schnittstellen	7
3.3	Benutzerschnittstelle	8
4	Testkonzept	8

Abbildungsverzeichnis

1 Zusammenfassung

Geben Sie hier eine kurze Zusammenfassung des Inhalts dieses Dokumentes. Wozu dient das Dokument (Zweck) und welche Informationen enthält es?

2 Systemanforderungen

2.1 Anforderungen an die Funktionalität

Führen Sie hier das Produkt Backlog das Sie erstellt haben wieder auf. Dabei müssen Sie auch die Akzeptanztestfälle pro User Story und das Definition Of Done bezüglich aller User Stories ebenfalls festhalten.

Überprüfen Sie im Team, ob die angegebenen Story Point immer noch korrekt sind oder passen Sie diese entsprechend an, falls nicht.

Das Produkt Backlog muss wie immer der Priorität entsprechend sortiert sein.
Aufgaben.

Anforderung / Tätigkeit	Story Points
S: Benutzer können erfasst werden A: Hans Muster, Abteilungsleiter, Marketing, 031 999 99 99 hm@yxz.ch konnte erfasst werden	5

2.2 Anforderungen an die Informationssicherheit und den Datenschutz

Oft wird übersehen, dass neben den funktionalen Anforderungen auch die Informationssicherheit und der Datenschutz berücksichtigt werden müssen. Überlegen Sie hier genau, ob Sie schützenswerte Daten besonders sorgfältig behandeln müssen, ob Sie Vorkehrungen bezüglich Datenverlust oder -veränderung treffen sollen oder ob Sie auf betriebliche Vorgaben Rücksicht nehmen müssen.

Formulieren Sie die entsprechenden Resultate in Form von Anforderungen.

2.3 Fachliche Entitätstypen

Zum Begriff Entität / Entitätstyp:

Eine Entität ist ein „Ding“, welches im Problembereich existiert.

Beispiele:

- Ein Hypothekarvertrag zwischen Herr Sowieso und der Bank Geizhals & Co.
- Der Benutzer-Account des Benutzers Ray Charles bei Facebook

Eine Entität wird einem Entitätstyp zugeordnet. Dieser umfasst alle Entitäten mit den gleichen Attributen.

Beispiele:

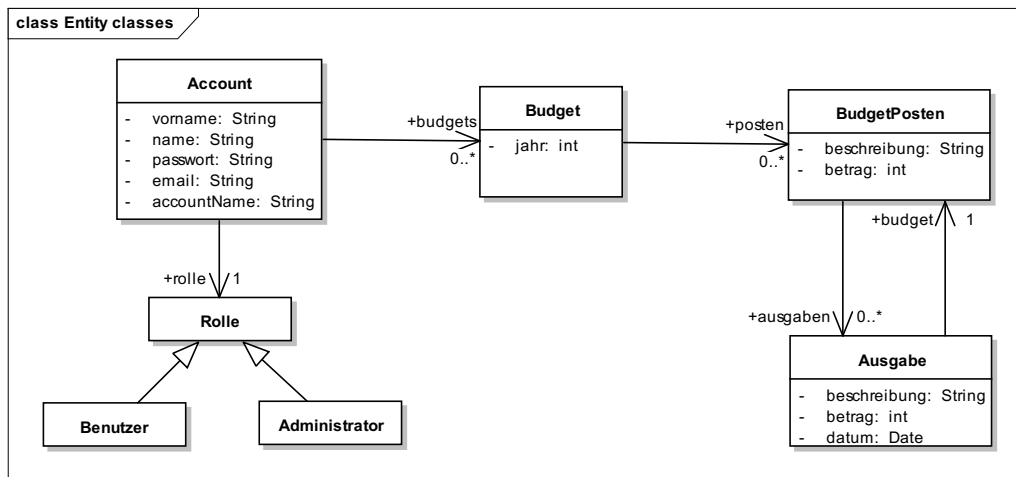
- Hypothekarvertrag. Attribute: Schuldner, Summe, Zins, Laufzeit
- Benutzer-Account. Attribute: Name, Vorname, Account-Name, Passwort, E-Mail

Der Zusammenhang zwischen Entität und Entitätstyp ist derselbe wie zwischen Objekt und Klasse in der objektorientierten Programmierung oder zwischen Zeile und Tabelle in relationalen Datenbanken.

Was soll nun hier gezeigt werden?

Die Struktur des Problembereichs, soweit sie für das Projekt relevant ist: Welche fachlichen Entitätstypen existieren und wie stehen diese miteinander in Beziehung? Stellen Sie diese Entitätstypen und Beziehungen als UML-Klassendiagramm dar. Jeder Entitätstyp wird als eine Klasse modelliert. Die Klassen enthalten nur Attribute und keine Operationen. Beschreiben Sie wo nötig jeden Entitätstyp kurz.

Beispiel:



Sehr oft werden die hier identifizierten Entitätstypen später als Tabellen in einer Datenbank realisiert.

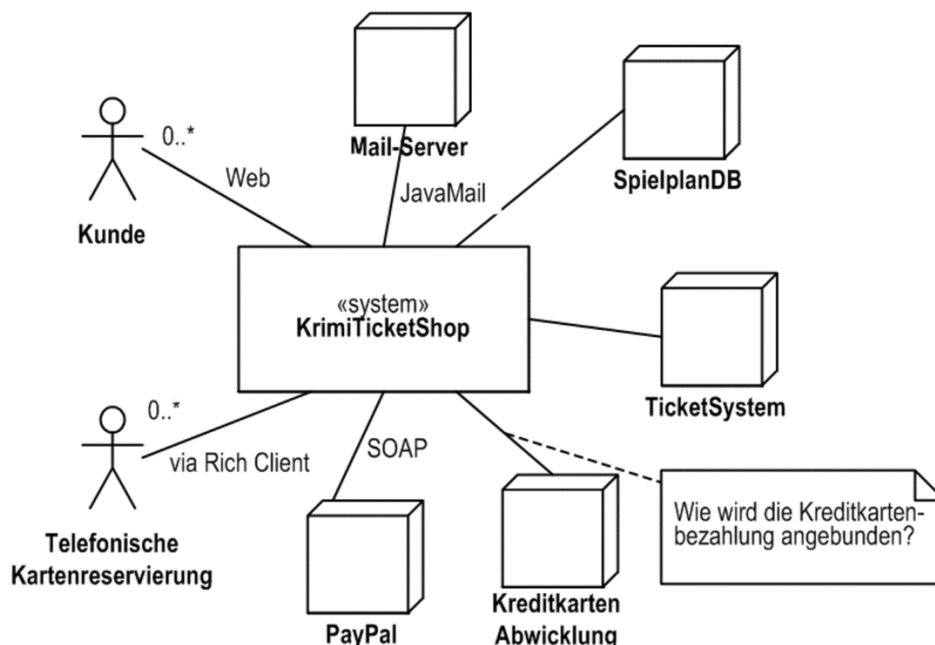
3 Systemarchitektur

In der Initialisierung haben Sie Lösungsvarianten ausgearbeitet und einen Variantenentscheid getroffen. Die Architektur der gewählten Lösungsvariante wird nun am besten soweit verfeinert, dass Ihr System komplett in einzelne Module und Schnittstellen zergliedert wird (Theorielektion). Beschreiben Sie sämtliche Elemente (Module und Schnittstelle) bis ins Detail.

Diese Aufgabe kann je nach System sehr anspruchsvoll sein. Lassen Sie sich im Zweifelsfall von Ihrer Lehrperson beraten!

3.1 Systemübersicht

Hier zeigen Sie den Systemkontext. Sie zeigen auf, in welchem Umfeld sich Ihre Anwendung befindet, mit welchen Systemen und Geräten wie und mit welchem Protokoll kommuniziert wird. Bsp.



3.2 Gliederung der Lösung

Hier zeigen Sie aus welchen Elementen Ihre Lösung besteht. Elemente können sein:

- Schichten/Layers.

Schichten gliedern Ihre Lösung horizontal. Eine Schicht bietet eine Schnittstelle der darüber liegenden Schicht an. Sie verwendet die Schnittstelle der darunterliegenden Schicht. Häufig verwendete Schichten sind

Persistenzschicht (DB-Layer), Applikationsschicht (Business-Logic), Präsentationsschicht (GUI, UI), siehe auch Three-Tier-Architektur bei Web-Applikationen.

- Pakete.

Ein Paket dient zur Kapselung von Funktionalität, welche einen Zusammenhang hat. Beispiel: Benutzerverwaltung, Administration. Pakete können Klassen oder Module enthalten. Pakete können auch weitere Pakete enthalten. Wenn Sie spezielle Libraries verwenden, können Sie auch diese als Paket darstellen.

- Klassen.

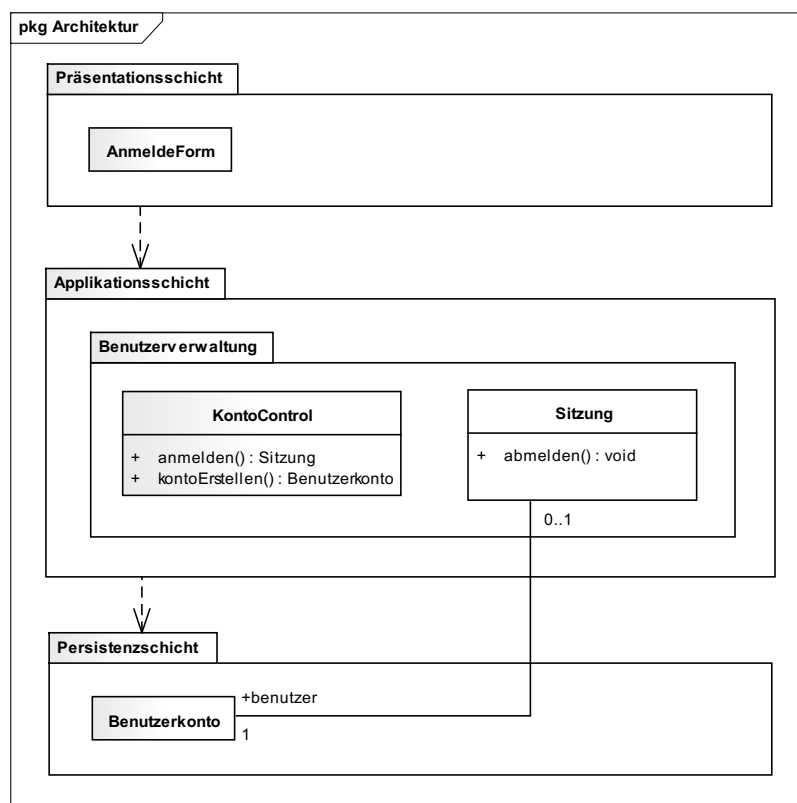
Zu den Entitätstypen kommen jetzt weitere Klassen hinzu. Dies sind beispielsweise Controller-Klassen. Sie kapseln die Applikationslogik. Ausserdem kommen hier auch Klassen hinzu, welche Schnittstellen kapseln (GUI, Proxys, Stubs für Web-Services)

- Module.

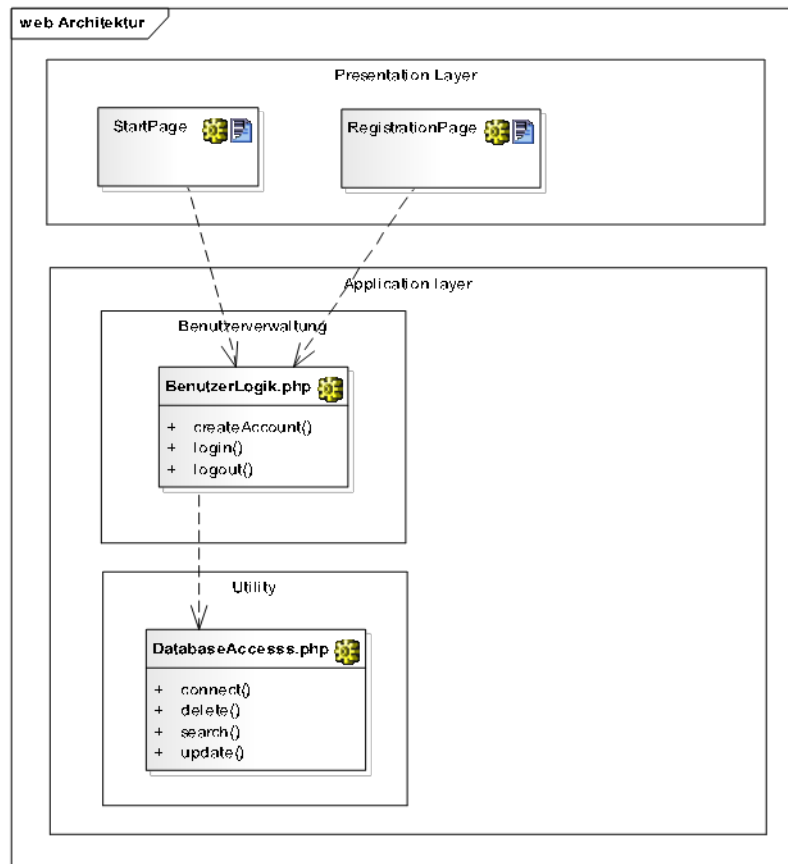
Wenn Sie nicht objektorientiert entwickeln verwenden Sie Module. Ein Modul enthält dann Sammlungen von Prozeduren. Einzelne dieser Prozeduren bilden die Schnittstelle des Moduls. Ein Skript (z.B. PHP-Skript) kann als Modul dargestellt werden.

Wenn Sie objektorientiert vorgehen illustrieren Sie die Architektur mittels UML-Paket- und Klassendiagrammen.

Wenn Sie nicht objektorientiert vorgehen, verwenden Sie Blockdiagramme zur Illustration. Blöcke, welche ein Modul darstellen enthalten eine Liste der Prozeduren des Moduls. Zeichnen Sie zwischen den Blöcken die Abhängigkeiten ein (gestrichelter Pfeil ausgehend vom abhängigen Block).



Beispiel 1: Dreischichtenarchitektur mit weiterer Unterteilung innerhalb der Applikations-schicht (objektorientiert).



Beispiel 2: Zweischichtenarchitektur mit weiterer Unterteilung der Applikationsschicht (prozedural/modular).

Wichtig: Für jedes Element der Architektur muss klar sein, was dessen Zweck ist. Geht dies nicht unmittelbar aus dem Diagramm hervor, so müssen Sie das betreffende Element im Text genauer beschreiben. Im Minimum muss der Text die groben Eckpunkte der Architektur beschreiben.

Beispiel:

Die Applikation ist nach dem klassischen Three-Tier-Model aufgebaut. Die Präsentations-schicht enthält alle Views der Applikation. Diese werden mit JSF und Managed Beans realisiert.

In der Applikationsschicht ist die Geschäftslogik implementiert. Dazu werden JEE Stateless Session Beans verwendet.

Die Persistenzschicht enthält die JEE Entity Beans und die Persistenz-Managerklassen.

3.3 Technische Schnittstellen

Hier beschreiben Sie diejenigen Schnittstellen, welche auf Grund der vorangehenden Abschnitte noch nicht genügend definiert sind. Dies betrifft in der Regel zwei Kategorien von Schnittstellen:

- a. Systeminterne Schnittstellen zwischen verteilten Teilsystemen. In der Regel handelt es sich um die Spezifikation von Kommunikationsprotokollen auf dem OSI-Layer 7. Dazu gehören eine Beschreibung der einzelnen Meldungen, welche über die Schnittstelle übertragen werden (Form/Syntax und Bedeutung der Meldung), sowie einzuhaltende Meldungsabfolgen. Sie müssen genügend genau beschreiben, um die Schnittstelle dann auch implementieren zu können.

- b. Externe Schnittstellen. Dies betrifft einerseits Export- / Import-Schnittstellen für Daten und andererseits Kommunikationsschnittstellen zu externen Systemen. Auch hier gilt: Die Schnittstellen müssen genügend genau spezifiziert sein.

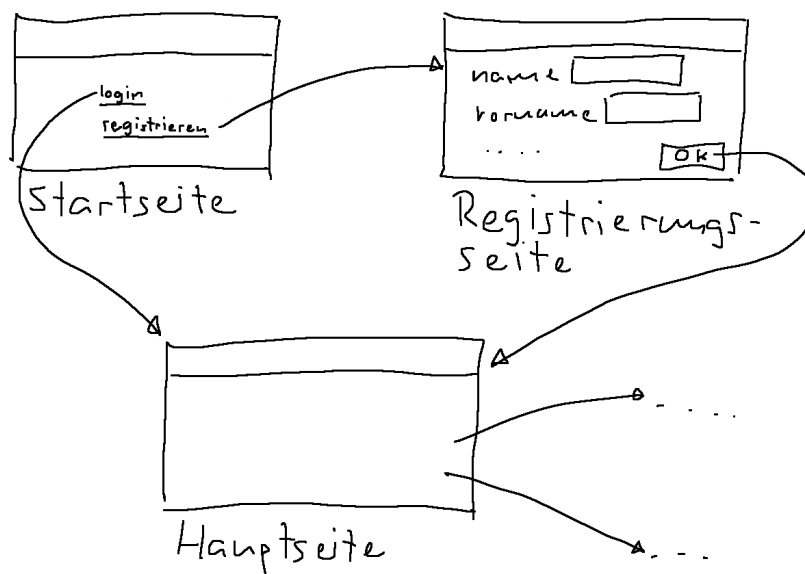
Beispiele:

- In einem netzwerkfähigen Spiel müssen Sie die Meldungen auf der Schnittstelle zwischen den Peers bestimmen und festlegen, welche Inhalte und Auswirkungen jede Meldung hat.
- Bei einem Web-Service müssen Sie die Meldungen zwischen dem Client und dem Server definieren. Dazu verwendet man z.B. die XML-basierte Sprache WSDL. Oft kann diese Beschreibung auch aus anderen Programmiersprachen generiert werden.
- Für den Import von Daten müssen Sie ein Format definieren, welches die Syntax der Importdatei und die Bedeutung der einzelnen Datenfelder genau beschreibt.

3.4 Benutzerschnittstelle

Skizzieren Sie hier die Benutzerschnittstelle mit Ihren Navigationspfaden. Es geht dabei nicht um eine grafisch und designmässig ausgereifte Darstellung, sondern um den logischen Aufbau. Anhand dieser Skizze müssen die Abläufe in den User Storys nachvollzogen werden können. Deshalb müssen alle dort erwähnten Views, Seiten, Dialoge oder Masken hier erscheinen. Die Skizze kann von Hand erstellt und eingescannt werden. Sie können auch mit einem Tool wie z.B. Balsamiq arbeiten.

Beispiel:



4 Testkonzept

Bereits jetzt wird mit der Planung der Tests (welche am Ende der Realisierungsphase stattfinden werden) begonnen.

Überlegen Sie sich eine Teststrategie (z.B. Testen aufgrund der einzelnen Anforderungen / Testen jedes Moduls und jeder Schnittstelle / Testen anhand von User Stories usw.).

Erstellen Sie hier eine erste Tabelle mit den Testfällen für den System- und Abnahmetest. Diese werden Sie in den nächsten Phasen ausbauen und genau spezifizieren. Im Moment geht es darum, die Testfälle zu identifizieren. Nehmen Sie als Ausgangspunkt Ihre User Stories. Als Faustregel gilt: Ein Testfall wird mehrere User Stories aufs Mal abdecken.

Beispiel:

Abgedeckte User Stories	Beschreibung
Neues Benutzerkonto erstellen	Ein Testbenutzer, welcher noch kein Konto besitzt, registriert sich beim System und kann sich anschliessend anmelden.
...	
...	