Online Katalog

Konzeptbericht (Applikationsentwicklung)

| Auftraggeber | Silvia Sager, Beat Walter |
| --- | --- |
| Projektleiter | Maximilian Seifert |
| Autor | Nicole Sager  Maximilian Seifert |
| Klassifizierung | Vertraulich |
| Status | In Arbeit |
|  |  |

Änderungsverzeichnis

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
| 25.02.2016 | 1.0 |  | Nicole Sager  Maximilian Seifert |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

[1 Zusammenfassung 2](#_Toc444156551)

[2 Systemanforderungen 2](#_Toc444156552)

[2.1 Fachliche Entitätstypen 2](#_Toc444156553)

[2.2 Anwendungsfälle / Product-Backlog 3](#_Toc444156554)

[3 Benutzerschnittstelle 3](#_Toc444156555)

[4 Systemarchitektur 3](#_Toc444156556)

[4.1 Gliederung der Lösung 3](#_Toc444156557)

[4.2 Schnittstellen 6](#_Toc444156558)

[5 Qualitätssicherung 7](#_Toc444156559)

[6 Projektplanung 7](#_Toc444156560)

# Zusammenfassung

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

# Systemanforderungen

Hier analysieren Sie den Problembereich. Unter Problembereich, oder englisch problem domain versteht man im weitesten Sinn das Geschäftsumfeld, in welchem eine Anwendung betrieben wird, z.B. das Bankgeschäft oder die Steuerverwaltung.

## Fachliche Entitätstypen

Zum Begriff Entität / Entitätstyp:

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

Beispiele:

* *Ein Reservationsanfrage von Herr Muster und an die Eigentümerin.*
* *Der Benutzer-Account des Benutzers Ray Charles*
* *Ein Artikel der von der Eigentümerin erstellt wurde*

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

Beispiele:

* *Reservationsanfrage. Attribute: Artikelname, Summe, Text, Datum & Uhrzeit,*
* *Benutzer-Account. Attribute: Name, Vorname, Account-Name, Passwort, E-Mail*
* *Artikel. Attribute: Titel, Preis, Künstler,*

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.

## Anwendungsfälle / Product-Backlog

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.

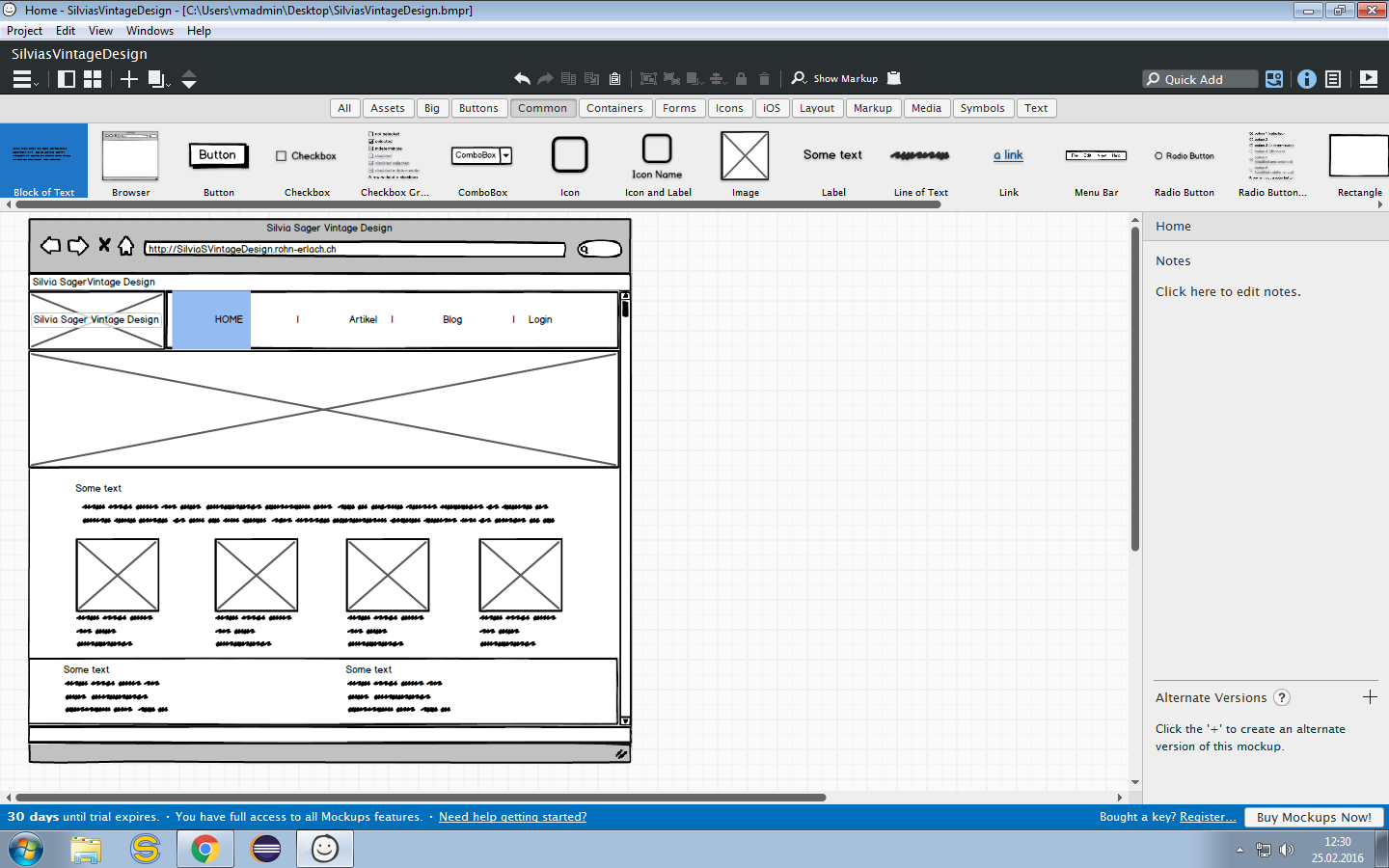
Überprüfen Sie im Team, ob die angegebenen Story Point immer noch korrekt sind oder passen Sie diese entsprechend an, falls nicht.

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

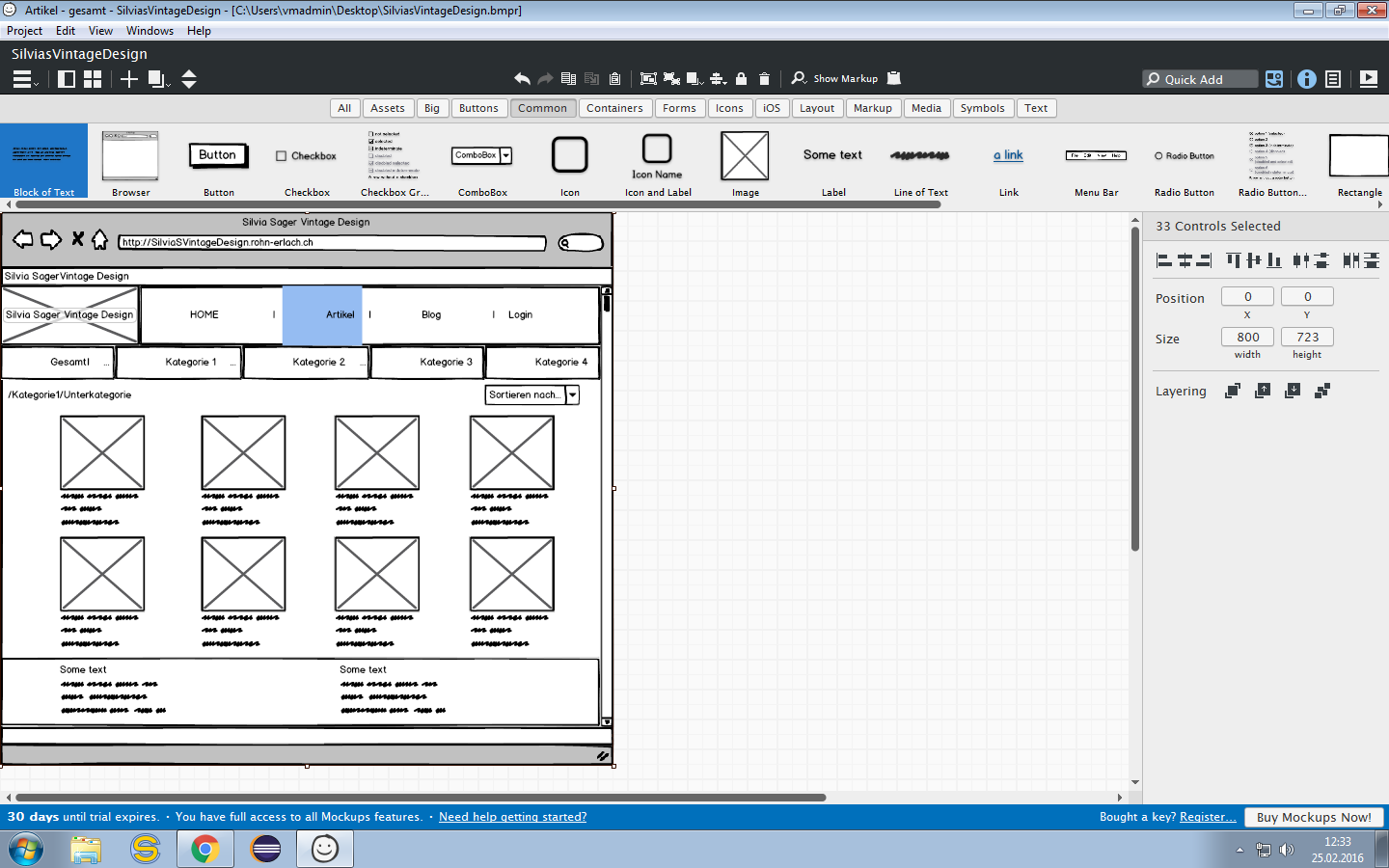
# Benutzerschnittstelle

In diesem Teil des Konzeptberichts sehen sie die ungefähre Seiten-Strucktur

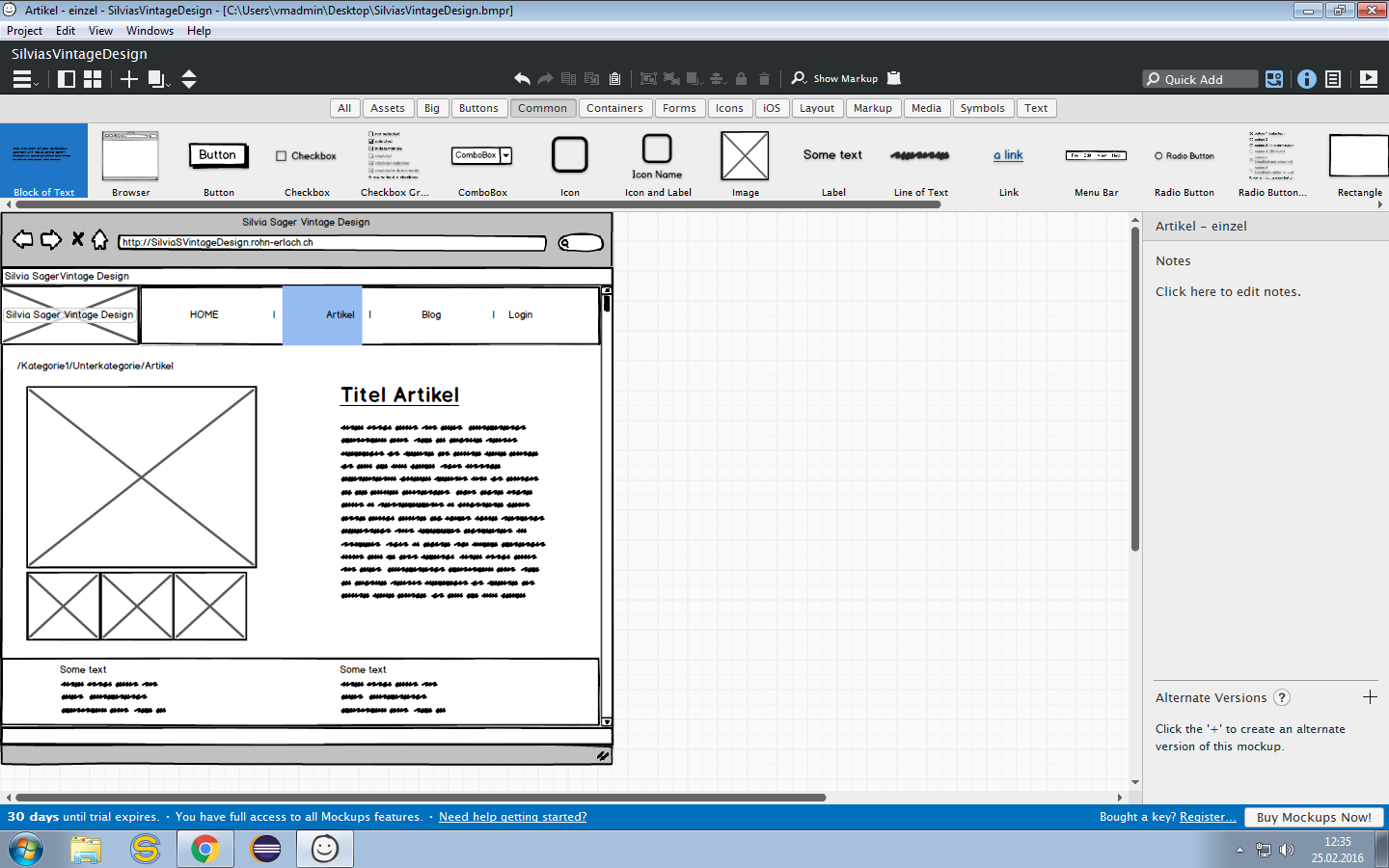
Wenn man als erstes auf die Domain geht wird man zur Home Seite gegracht



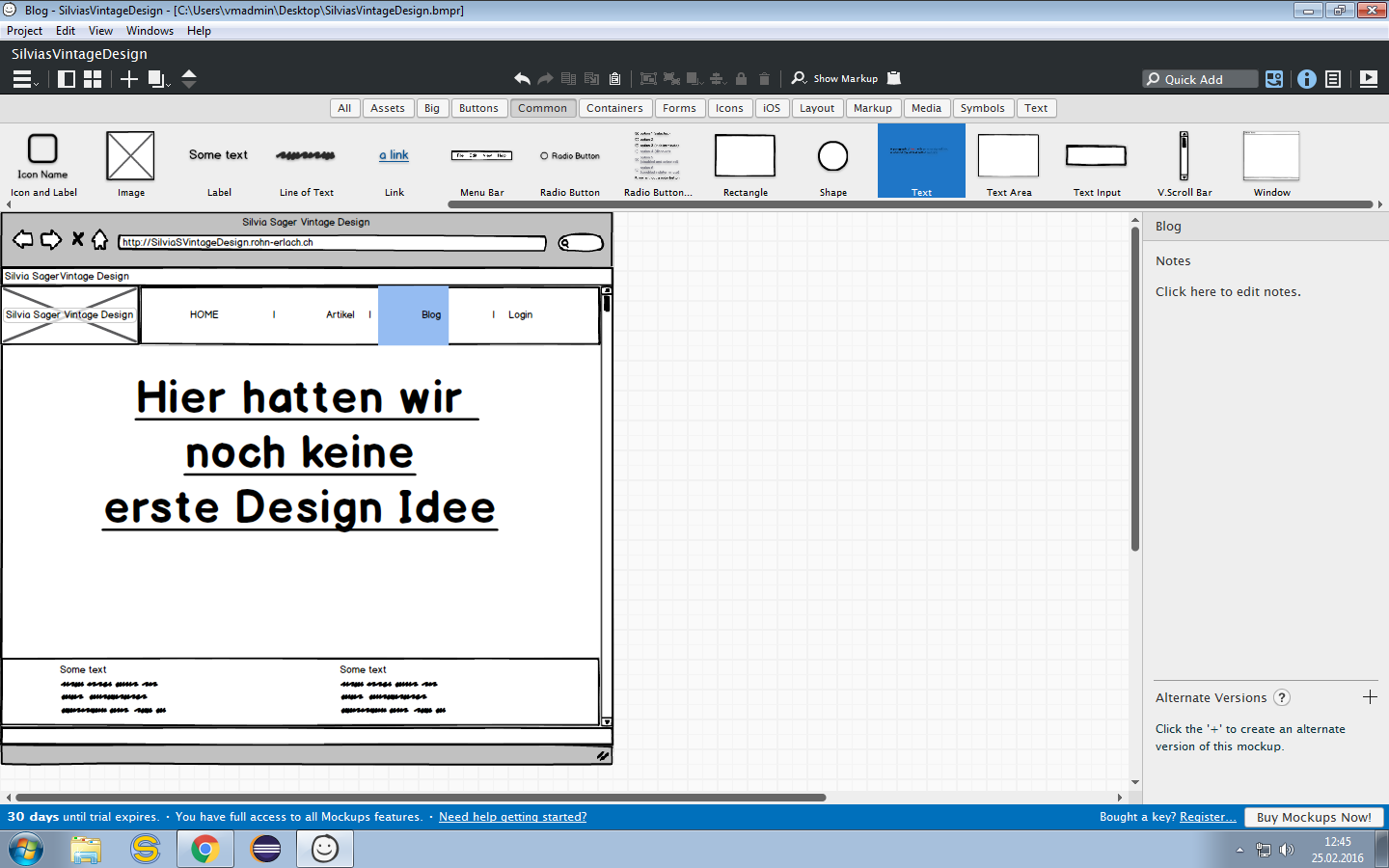
Von dort aus kann der Benutzer sich nun Entweder alle Artikel ansehen,



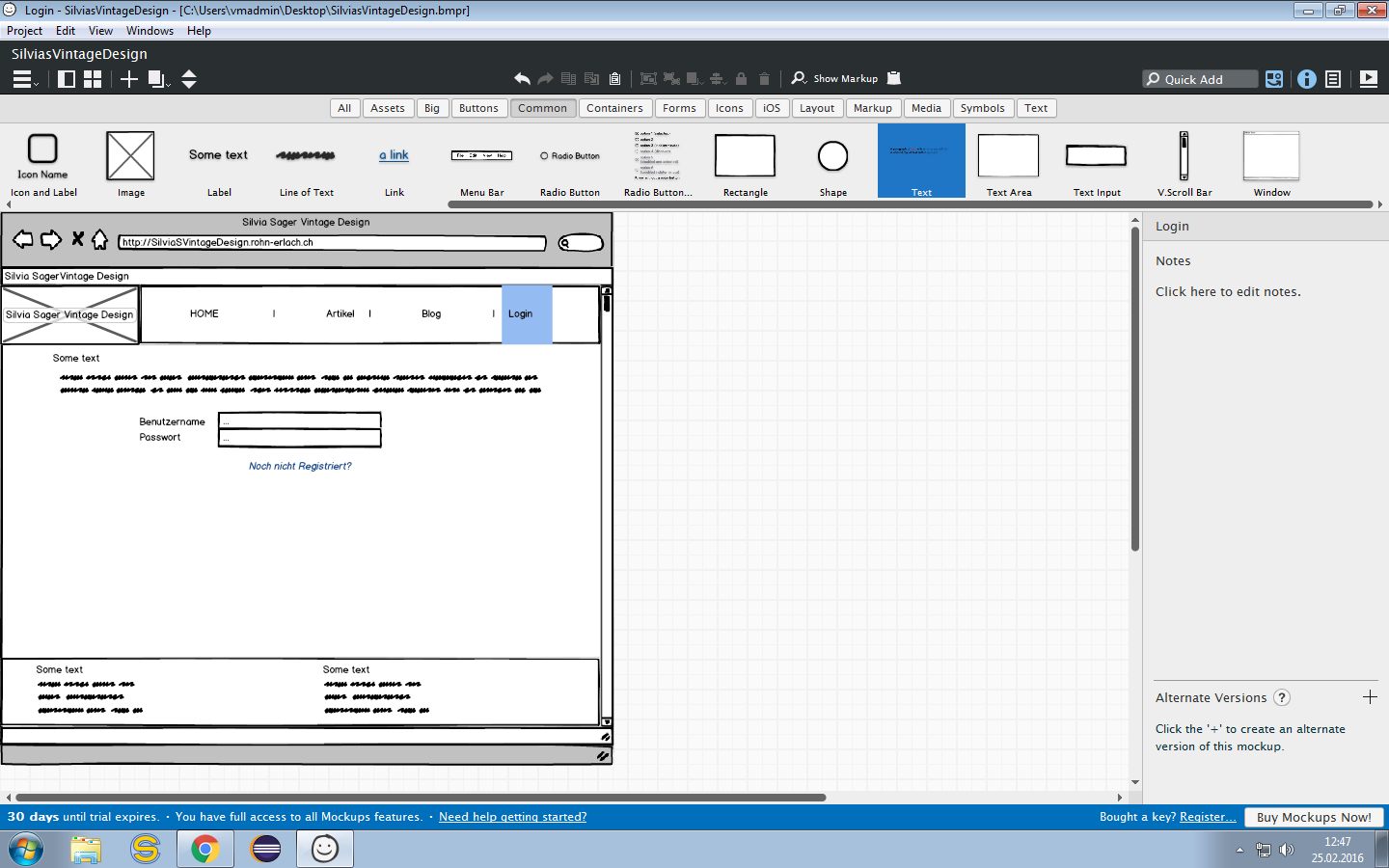
Und von dieser Seite kann er dann zur Einzel Ansicht eines Artikels kommen.



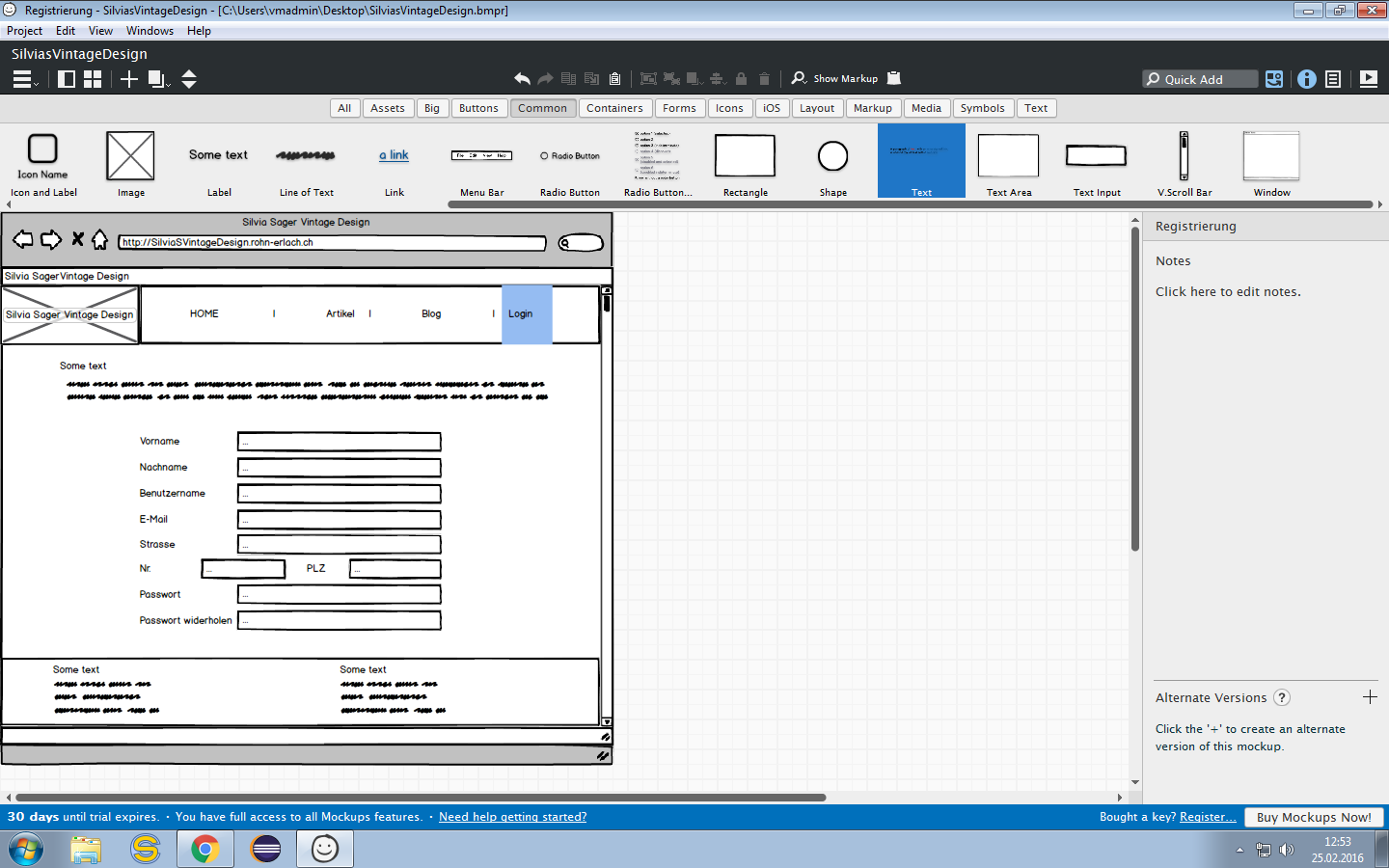
Oder sie können sich den Blog der Eigentümerin ansehen.



Sie können auch falls sie schon ein Benutzerkonto besitzen sich einloggen.



Falls der Benutzer nun noch keinen Account besitzt kann er sich registrieren.



# Systemarchitektur

In der Initialisierung haben Sie Lösungsvarianten ausgearbeitet und einen Variantenentscheid getroffen. Die Architektur der gewählten Lösungsvarianten haben Sie in einem Blockdiagramm grob beschrieben. Hier geht es nun darum die Architektur zu verfeinern.

Zum Beispiel: Wenn Sie eine Web-Applikation als Block beschrieben haben, so geht es jetzt darum die innere Struktur dieser Web-Applikation festzulegen.

## 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 Kontroller-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 Applikationsschicht (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 Präsentationsschicht 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.*

## 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:

1. 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.
2. 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.

# Qualitätssicherung

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, de Testfälle zu identifizieren. Nehmen Sie als Ausgangspunkt Ihre User Stories. Als Faustregel gilt: Eine Testfall wird mehrere User Stories aufs Mal abdecken.

Beispiel:

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Abgedeckter Anwendungsfall oder User Stories** | **Beschreibung** |
| 1 | Neues Benutzerkonto erstellen | Ein Testbenutzer, welcher noch kein Konto besitzt, registriert sich beim System und kann sich anschliessend anmelden. |
| 2 | … |  |
| 3 | … |  |

# Projektplanung

Aktualisieren Sie den Projektplan aus der Initialisierungsphase 🡪 neue Version. Machen Sie hier lediglich einen Verweis auf das aktualisierte Dokument

Beispiel:

*Siehe Projektplan, Version 2.0 vom 18.3.2014*

Im Projektplan selber:

Aktualisieren Sie die tatsächlich aufgelaufenen Aufwände für die erledigten Aktivitäten und die geschätzten Aufwände für die kommenden Aktivitäten. Detaillieren Sie dabei die kommenden Aktivitäten für die nächste Phase 🡪 Iterationen oder Sprints.

Beurteilen Sie auch die Risikosituation erneut. Haben sich an der Risikosituation Änderungen (Verschärfung/Entschärfung, neue Risiken) ergeben? Dann passen Sie die Risikosituation an. Führen Sie wiederum konkrete Massnahmen auf, wie Sie den Risiken in der nächsten Phase begegnen wollen.