

Projektbericht: Erstellung einer Webanwendung zur Verbesserung des Workflows des Erstellens von Benutzerhandbüchern

Auszubildender

Oliver Herrmann

Geb.: 23.09.1992

GNS mbH

Tel.: +49 201 109 - 1523

Email: oliver.herrmann@gns.de

Ausbildungsberuf: Fachinformatiker Anwendungsentwicklung

Ausbildungsbetrieb

GNS mbH

Frohnhauser Str. 67

45127 Essen

7. Oktober 2015, Essen

Inhaltsverzeichnis

Abbildungsverzeichnis	4
Tabellenverzeichnis	5
Abkürzungsverzeichnis	6
1. Einleitung	7
1.1. Projektumfeld	7
1.2. Projektziel	7
1.3. Projektbegründung	7
1.4. Projektschnittstellen	8
1.5. Projektabgrenzung	8
2. Projektplanung	9
2.1. Projektphasen	9
2.2. Abweichung vom Projektantrag	9
2.3. Ressourcenplanung	9
2.4. Entwicklungsprozess	10
3. Analysephase	10
3.1. Ist-Analyse	10
3.2. Wirtschaftlichkeitsanalyse	10
3.2.1. Nutzen des Projekts	11
3.2.2. Kosten des Projekts	12
3.2.3. Wirtschaftlichkeit des Projekts	13
3.3. Risikoanalyse	13
3.4. "Make or Buy" Entscheidung	14
3.5. Projektkosten	14
3.6. Amortisationsdauer	14
3.7. Nutzwertanalyse	14
3.8. Anwendungsfälle	15
3.9. Qualitätsanforderungen	15
3.10. Lastenheft / Fachkonzept	16
3.11. Zwischenstand	16
4. Entwurfsphase	16
4.1. Zielplattform	16
4.2. Architekturdesign	17
4.2.1. Serverarchitektur	18
4.2.2. Clientarchitektur	18
4.3. Entwurf der Benutzeroberfläche	19
4.3.1. Visuelles Konzept	19
4.3.2. Prototyp	20

4.4. Datenmodell	20
4.5. Geschäftslogik	20
4.6. Maßnahmen zur Qualitätssicherung / Testkonzept	20
4.7. Pflichtenheft / Datenverarbeitungskonzept	20
4.8. Zwischenstand	20
5. Implementierungsphase	20
5.1. Implementierung der Datenstrukturen	20
5.2. Implementierung des Backends	20
5.3. Implementierung des Frontends	20
5.4. Implementierung der Tests	20
5.5. Zwischenstand	20
6. Abnahmephase	20
6.1. Zwischenstand	21
7. Einführungsphase	21
7.1. Zwischenstand	21
8. Dokumentation	21
8.1. Zwischenstand	21
9. Fazit	21
9.1. Soll- / Ist-Vergleich	21
9.2. Lesson-Learned	21
9.3. Ausblick	21
Anhang	22
A. Projektphasen	22
B. Risikoanalyse	23
C. "Make or Buy"-Bewertung	25
D. Kostenübersicht nach Projektphasen	26
E. Nutzwertanalyse	27
F. Use-Case-Diagramme	28
G. Qualitätsanforderungen (Auszug aus Pflichtenheft)	30
H. Lastenheft (Auszug)	31
I. Oberflächenprototyp	33

Abbildungsverzeichnis

1.	Use-Case "Wiki konvertieren"	28
2.	Use-Case "Wiki bearbeiten"	29
3.	Prototyp - Layout für Desktop-Browser	33
4.	Prototyp - Layout für Mobile-Browser	34

Tabellenverzeichnis

1.	Projektphasen	9
2.	Projektphasen	9
3.	Stundensätze	11
4.	"Make or BuyBewertung	14
5.	Ergebnis der Nutzwertanalyse	15
6.	Qualitätsanforderungen an die Anwendung - Auszug	16
7.	Zwischenstand nach der Analysephase	16
8.	Projektphasen detailliert	22
9.	Risikoanalyse detailliert	24
10.	Detaillierte "Make or BuyBewertung	25
11.	Kosten unterteilt nach Projektphasen	26
12.	Detaillierte Nutzwertanalyse bezüglich der Backend-Architektur	27
13.	Detaillierte Nutzwertanalyse bezüglich der Front-Architektur	27
14.	Qualitätsanforderungen an die Anwendung (Pflichtenheft)	30

Abkürzungsverzeichnis

CVS	Concurrent Version System
GNS	Gesellschaft für Nuklear-Service mbH
Hosting	Bereitstellen einer Web-Anwendung auf einem Web-Server
KIS	Abteilung der GNS für Softwareentwicklung

1. Einleitung

1.1. Projektumfeld

Die GNS (Gesellschaft für Nuklear-Service) mbH ist ein mittelständisches Unternehmen, das 1977 gegründet wurde und aktuell etwa 550 Mitarbeiter beschäftigt. Die Firmenzentrale befindet sich in Essen. Die GNS betreibt des weiteren Standorte in Duisburg, Mülheim an der Ruhr und Karlsruhe. Das Hauptgeschäftsfeld der GNS ist die Entwicklung sowie der Vertrieb von Behältern zur Lagerung und zum Transport von radioaktivem Material, als auch die Beladung solcher Behältnisse.

Ebenfalls zum Geschäftsbereich der GNS zugehörig ist der Ver- und Betrieb von Softwarelösungen. Hier ist beispielhaft das AVK (Abfallfluss- Verfolgungs- und Produktkontroll-System) zu nennen, welches 1991 entwickelt wurde und heute in allen deutschen Kernkraftwerken verwendet wird. Daneben werden von der GNS eine Vielzahl von stark spezialisierten Softwarelösungen betrieben, wovon viele als digitale Dienstleistung für Kunden (z.B. Kernkraftwerke und Zwischenlager) bereitgestellt werden.

Auftraggeber des Projekts "Erstellung einer Webanwendung zur Verbesserung des Workflows des Erstellens von Benutzerhandbüchern" ist die GNS, in Person Friedrich Bauriedel, Leiter der Abteilung KIS.

1.2. Projektziel

Ziel des Projekts ist, den regelmäßig anfallenden Arbeitsschritt "Erstellen eines Benutzerhandbuch aus einem Wiki" für die Mitarbeiter der GNS zu vereinfachen und zu beschleunigen. Ebenfalls soll der Workflow normiert und zentralisiert werden, sodass der Aufwand für vorlaufende Arbeiten minimiert wird.

1.3. Projektbegründung

Wie in 1.1 erwähnt betreibt die GNS eine Vielzahl an hoch spezialisierten Softwarelösungen für diverse Kunden. Aufgrund der Komplexität vieler dieser Anwendungen ist es nötig den Benutzern ein Handbuch zur Verfügung zu stellen. Zu diesem Zweck betreibt die GNS betriebsinterne Wikis, in denen Inhalte bezüglich der korrekten Bedienung einer Anwendung von Mitarbeitern der GNS gepflegt werden.

2012 wurde von der GNS der sogenannte WikiConverter entwickelt. Dies ist ein Tool, welches die Inhalte eines MediaWikis extrahiert und diese in ein PDF- bzw. ein HTML-Dokument konvertiert. Ziel der Entwicklung war es, die mithilfe des WikiConverters erzeugten Dateien als interaktive Hilfe in die jeweiligen Anwendungen einzubetten bzw. als Benutzerhandbuch den Kunden bereitstellen zu können.

Allerdings wurde bei der Entwicklung des WikiConverters wenig Wert auf eine intuitive Benutzerführung gelegt, was dazu führte, dass die Bedienung von diesem äußerst komplex ist.

Durch die in den letzten Jahren gesteigerte Anzahl der von der GNS betriebenen Softwarelösungen, sowie die stetige Verkürzung der Release-Intervalle dieser, ist der Arbeitsschritt "Erstellen von Benutzerhandbüchern" immer mehr in den Vordergrund gerückt, was schließlich dazu führte, dass eine Überarbeitung von diesem unumgänglich wurde.

Zu diesem Zweck wurde von der Abteilung KIS Mitte 2015 das Projekt "Erstellung einer Webanwendung zur Verbesserung des Workflows des Erstellens von Benutzerhandbüchern" initiiert, welches zum Ziel hat, die Erstellung von Benutzerhandbüchern zu vereinfachen und zu zentralisieren.

Im Rahmen des Projekts wird eine Web-Anwendung entwickelt, welche die Kernfunktionalität des WikiConverters ummantelt und eine einheitliche, graphische Benutzerschnittstelle zu dieser anbietet. So wird die Komplexität des WikiConverters vor den Benutzern verborgen, um diesen eine einfachere und intuitivere Benutzung zu ermöglichen.

Als messbares Ziel des Projekts ist eine Reduzierung der durchschnittlichen Arbeitszeit für den Workflow "Erstellen eines Benutzerhandbuchs" um mindestens 25% veranschlagt.

1.4. Projektschnittstellen

Die technischen und sozialen Schnittstellen des Projekts sind ausschließlich betriebsinterner Natur.

Technisch betrachtet greift das Projekt die bestehende Anwendung "Wikiconverter" auf und adaptiert dessen Kernfunktionalität.

Aus projekt-spezifischer Sicht sind die Schnittstellen rein intern, da sowohl Auftraggeber als auch Endnutzer der Anwendung Mitarbeiter bzw. Abteilungen der GNS sind.

Die Projektabschließende Abnahme erfolgt gegenüber dem Auftraggeber, hier die Abteilung KIS der GNS.

1.5. Projektabgrenzung

Ausdrücklich nicht Teil des Projekts ist das Hosting der im Rahmen des Projekts entwickelten Anwendung sowie alle in Verbindung mit dem Hosting stehenden administrativen Maßnahmen. Diese werden von nicht direkt am Projekt beteiligten Mitarbeitern der GNS durchgeführt.

2. Projektplanung

2.1. Projektphasen

Projekphase	Geplante Zeit
Anforderungsanalyse	7h
Projektplanung	4h
Software-Architektur	14h
Implementierung	26h
Projektabschluss	3h
Sonstiges	16h
Gesamt	70h

Tabelle 1: Projektphasen

Eine detaillierte Übersicht der einzelnen Phasen findet sich im Anhang A auf Seite 22

2.2. Abweichung vom Projektantrag

Abweichend vom Projektantrag werden das Pflichten- sowie das Lastenheft nur in Auszügen im Projektbericht aufgeführt. Dies geschieht aus dem Grund den Umfang des Projektbericht nicht übermäßig zu gestalten.

2.3. Ressourcenplanung

Im Folgenden werden alle für die Durchführung des Projekts bezogenen Ressourcen aufgelistet. Diese werden durch den Auftraggeber, hier die GNS, bereitgestellt.

Art der Resource	Beschreibung
Sachliche Resource	
Projektbüro	
Computer	
IDE	Eclipse for JEE Developers
Demilitarisierte Zone	zu Testzwecken
Personelle Resource	
Projektleiter	Oliver Herrmann
Projektmitarbeiter	Oliver Herrmann

Tabelle 2: Projektphasen

2.4. Entwicklungsprozess

Grundlage für den Entwicklungsprozess bildet das iterative Wasserfallmodell¹. Dies wurde vor allem gewählt, da es gut mit Projekten geringen bis mittleren Umfangs harmonisiert. Andere Modelle wie z.B. Scrum oder Extreme-Programming wurden in erster Linie nicht gewählt, da diese die Projektarbeit in einem Team erfordern.

3. Analysephase

3.1. Ist-Analyse

Der jetzige Stand bezüglich des Workflows "Erstellen eines Benutzerhandbuch" sieht vor, dass der ausführende Mitarbeiter den WikiConverter aus dem internen CVS auscheckt, im Quellcodeangaben zu dem zu konvertierenden Wiki (URL, Hauptseite) ergänzt und den Quellcode anschließend kompiliert und ausführt. Das Ergebnis der Ausführung sind die generierten PDF- und HTML-Dateien.

Das Projekt "Erstellung einer Webanwendung zur Verbesserung des Workflows des Erstellens von Benutzerhandbüchern" sieht vor, den Workflow wie folgt zu definieren:

- Der Benutzer loggt sich online in der Anwendung ein.
- Der Benutzer wählt ein existierendes Wiki aus oder pflegt ein neues ein. (Nötige Angaben sind URL und Hauptseite des Wikis)
- Der Benutzer kann über die graphische Oberfläche das Konvertieren starten.
- Nach erfolgreicher Konvertierung kann der Benutzer die generierten Dateien herunterladen.

Außerdem werden aus Konvertierungsvorgängen erstellte Dateien dauerhaft gespeichert. Diese können ebenfalls vom Benutzer heruntergeladen werden, falls eine neue Konvertierung nicht benötigt wird.

3.2. Wirtschaftlichkeitsanalyse

Im folgenden wird die Wirtschaftlichkeit des Projekts betrachtet. Da das Projekt keinen direkten Gewinn erwirtschaftet sondern der monetäre Vorteil durch die Erleichterung der Arbeit der betroffenen Mitarbeiter erzielt wird, wird zuerst der Nutzen des Projekts messbar festgehalten. Dieser wird dann den Kosten des Projekts gegenübergestellt um eine qualifizierte Aussage über die Wirtschaftlichkeit des Projekts zu machen.

Grundlage der Analyse sind folgende Stundensätze²:

¹http://www.pi.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_PI/LV__SE_RE/R_01_Wasserfallmodell__Ausarbeitung__Schwaiger.pdf abgerufen am 05.10.2015

²Beruhend auf Schätzungen bzw. betriebsinternen Angaben

Art der Beschäftigung	Stundensatz
Fachinformatiker	30 €
Auszubildender	7 €
Gemeinkosten	10 €

Tabelle 3: Stundensätze

Hieraus ergibt sich ein *effektiver* Stundensatz von 30 € für die Arbeit eines Fachinformatikers, bzw. 17 € für die Arbeits eines Auszubildenden (hier Projektmitarbeiter).

3.2.1. Nutzen des Projekts

Der Nutzen des Projekts manifestiert sich primär darin, dass der regelmäßig auftretende Workflow "Erstellen eines Benutzerhandbuchs" beschleunigt wird. Da die exakte Bestimmung der durchschnittlich für diesen Workflow anfallenden Zeit von vielen, auch personellen, Faktoren abhängt, wird er für die Zwecke der Wirtschaftlichkeitsanalyse konservativ abgeschätzt. Diese Schätzung wird fortan als Grundlage für weitere Berechnungen herangezogen, um so ein Worst-Case Ergebnis ausdrücken zu können. Der aktuelle durchschnittliche zeitliche Aufwand für den Workflow ergibt sich aus

- Einarbeitung in den Workflow und den WikiConverter im Allgemeinen
- Auschecken der Quellen
- Anpassungen am Quellcode
- Kompilieren der Quellen
- Ausführen der Anwendung
- Sichern der Ergebnisse

und beträgt nach Schätzungen **8 Minuten**.

Der neue, vom Projekt angestrebte Workflow setzt sich dagegen wie folgt zusammen

Alternative 1

- Login
- Konvertierung anstoßen
- herunterladen der Ergebnisse

Geschätzte Dauer: 5 Minute.

Alternative 2

- Login
- Auswählen eines bereits existierenden Ergebnisses
- herunterladen der Ergebnisse

Geschätzte Dauer: 2 Minuten.

Hier ist anzumerken, dass durch die zentrale Speicherung der konvertierten Ergebnisse dem Nutzer ermöglicht wird, den zeitintensiven Schritt der Konvertierung auszulassen, wenn ein adäquates Ergebnis bereits existiert. Dies ist z.B. dann der Fall, wenn die Konvertierung kürzlich durch einen dritten Mitarbeiter vorgenommen wurde und seit dem keine relevanten Änderungen am zu Grunde liegenden Wiki auftraten.

Konservativ abgeschätzt wird davon ausgegangen, dass in 80% der Fälle Alternative 1 und in den übrigen 20% Alternative 2 eintreten wird.

$$T_{\emptyset} = 0.8 * 5m + 0.2 * 2m = 4.4m \quad (1)$$

Aus Formel 1 ergibt sich eine durchschnittliche Dauer des Workflows von **4,4 Minuten**.

$$T_{\Delta} = \frac{(8m - 4,4m)}{8m} = 45\% \quad (2)$$

Dies entspricht nach Formel 2 einer Einsparung von **45%** und verspricht ein Einhalten des in Kapitel 1.3 ausgesprochenen Ziels, die durchschnittliche Arbeitszeit des Workflows um mindestens mindestens 25% zu reduzieren.

$$K_{wchtl} = 7,2m * \frac{1h}{60m} * \frac{40\text{€}}{h} = 4,8\text{€} \quad (3)$$

Unter der Annahme, dass der Workflow durchschnittlich zweimal pro Woche durchgeführt wird ergibt sich daraus eine Zeitersparnis von 7,2m pro Woche, was nach Formel 3 einer wöchentlichen monetären Ersparnis von **4,8 €** entspricht.

3.2.2. Kosten des Projekts

Die Kosten des Projekts werden allein durch die Kosten der Entwicklung bestimmt. Es werden keine nicht bereits vorhandenen Ressourcen für das Projekt verwendet.

$$K_{ges} = 70h * \frac{17\text{€}}{h} = 1190\text{€} \quad (4)$$

Bei einem kalkulierten Projektumfang von 70 Stunden ergibt sich aus Formel 4 ein Kostenumfang von **1190 €**, unter der Voraussetzung, dass als Projektmitarbeiter ausschließlich Auszubildende eingesetzt werden.

3.2.3. Wirtschaftlichkeit des Projekts

Zieht man die Ergebnisse aus Kapitel 3.2.1 und 3.2.2 heran und betrachtet diese relativ zueinander, lässt sich feststellen, dass sich das Projekt in etwa 5 Jahren amortisiert hat.

3.3. Risikoanalyse

Im folgenden wird eine Risikoanalyse und -bewertung durchgeführt. Eine detaillierte Übersicht der Risikobewertung findet sich im Anhang B auf Seite 23.

Zusammenfassend lässt sich das Ergebnis der Risikoanalyse in drei Teilbereiche gruppieren.

Gruppe 1 Unvermeidbare Risiken

Hierzu zählen z.B. krankheitsbedingte personelle Ausfälle. Diese stellen ein unvermeidbares Risiko dar. Eine Strategie zur Vermeidung ist hier nicht vorgesehen.

Gruppe 2 Bedingt vermeidbare Risiken

Hierzu zählen vor allem Risiken aus Fehlern der Planung. Diese Risiken sind bedingt vermeidbar. Dem Auftreten solcher Risiken wird insbesondere durch die im Qualitätssicherungskonzept vorgesehenen Zwischenstandsanalysen entgegengewirkt.

Gruppe 3 Leicht vermeidbare Risiken

Dies sind in erster Linie Risiken die in der Entwicklung auftreten können. Das Eintreten dieser Risiken wird in erster Linie durch die Erfahrung und intensive fachliche Vorbereitung der Projektmitarbeiter vermieden. Des weiteren hilft das im Qualitätssicherungskonzept beinhaltete Testkonzept hier für eine frühzeitige Erkennung von Konsequenzen dieser Risikogruppe.

Die möglichen Konsequenzen lassen sich wiederum in zwei Bereiche gruppieren.

Gruppe A Zeitliche Konsequenzen

Dies sind Risikofolgen, welche in erster Linie eine Ausdehnung des Projektzeitraums bzw. eine Verzögerung des Projektabschlusses mit sich ziehen. Da das Projekt allerdings für rein betriebsinterne Zwecke konzipiert ist und keinen direkten Gewinn erwirtschaftet beschränken sich die negativen Auswirkungen dieser Gruppe auf ein Minimum. Dies hat zur Folge, dass Zeitliche Konsequenzen, sofern sie nicht in sehr hoher Quantität auftreten, nicht zum Abbruch des Projekts führen.

Gruppe B Finanzielle Konsequenzen

Dies sind Risikofolgen, welche primär eine Erhöhung des monetären Aufwands des Projekts mit sich ziehen. Erzeugt werden diese Konsequenzen vor allem durch Verzögerungen im Projektablauf.

Da das Projektbudget ausschließlich für die Beschäftigung der Projektmitarbeiter aufgewendet wird, ergibt sich dessen Betrag aus der geplanten Projektdauer. Dies hat zur Folge, dass finanzielle Konsequenzen direkt mit zeitlichen Konsequenzen gleichgesetzt werden können und daher, bei gemäßigten Auftreten, nicht zum Projektabbruch führen.

3.4. "Make or Buy" Entscheidung

Im Folgenden wird abgewägt, ob sich das Projekt bezüglich Nutzen und Wirtschaftlichkeit gegenüber bereits existierenden Lösungen behauptet.

Alternative	Nutzen	Wirtschaftlichkeit
MediaWiki Erweiterung	⊖ ⊖ ⊖	⊖ ⊖ ⊖ ⊖
Erstellung der Anwendung durch Dienstleister	⊕ ⊕	⊖ ⊖ ⊖
Eigenständige Erstellung der Anwendung	⊕ ⊕ ⊕	⊖ ⊖

Tabelle 4: "Make or Buy" Bewertung

Eine detaillierte Übersicht der "Make or Buy"-Bewertung findet sich im Anhang C auf Seite 25.

Als Ergebnis der "Make or Buy"-Analyse lässt sich festlegen, dass eine eigenständige Entwicklung der Anwendung aus wirtschaftlicher und nutzenorientierter Sicht die beste Alternative darstellt.

3.5. Projektkosten

Wie in Kapitel 3.2.2 bereits erläutert setzen sich die Kosten ausschließlich aus den Mitarbeiterkosten der Projektmitarbeiter zusammen. Da für das Projekt nur ein Mitarbeiter veranschlagt ist, sind die Kosten äquivalent zu der kalkulierten Dauer des Projekts.

$$K_{ges} = T_{ges} * K_{projmitarbeiter} = 70h * \frac{17€}{1h} = 1190€ \quad (5)$$

Aus Formel 5 wird ersichtlich, dass sich die Gesamtkosten des Projekts auf **1190 €** belaufen.

Eine detaillierte, nach Phasen unterteilte Übersicht der Kosten findet sich im Anhang D auf Seite 26

3.6. Amortisationsdauer

Siehe Kosten und Nutzen

3.7. Nutzwertanalyse

Für die entwicklungsrelevanten Bereiche "Backend" und "Frontend" werden jeweils Nutzwertanalysen durchgeführt, um im Vorhinein die optimale Architektur zu bestimmen.

Architekturbereich	beste Alternative
Backend	JEE-Stack
Frontend	HorCrux

Tabelle 5: Ergebnis der Nutzwertanalyse

Die zugrunde liegenden Nutzwertanalysen befinden sich im Anhang E auf Seite 27.

3.8. Anwendungsfälle

Die Anwendung deckt alle für den Workflow "Erstellen eines Handbuchs" relevanten Anwendungsfälle ab. Auch vor- und nachbereitende Anwendungsfälle werden von der Anwendung abgedeckt.

Es folgt eine Auflistung aller relevanten Anwendungsfälle.

1. vorgelagerte Anwendungsfälle
 - a) Anlegen eines Benutzers
 - b) Login durch einen Benutzer
 - c) Anlegen eines neuen Wikis
2. Hauptanwendungsfälle
 - a) Konvertieren eines Wikis
 - b) Herunterladen eines Konvertierungsergebnisses
 - c) Bearbeiten eines Wikis
3. nachgelagerte Anwendungsfälle
 - a) Löschen eines Benutzers
 - b) Löschen eines Wikis
 - c) Löschen eines Konvertierungsergebnisses

Zu den Anwendungsfällen 2a und 2c finden sich im Anhang F auf Seite 28 Use-Case-Diagramme, welche die Anwendungsfälle detailliert darstellen.

3.9. Qualitätsanforderungen

Die Qualitätsanforderungen der Anwendung sind hinlänglich im dem dem Projekt zugehörigen Pflichtenheft beschrieben.

Hier wird ein gekürzter Auszug aus diesem dargestellt.

Qualitätsmerkmal	sehr gut	gut	normal	nicht relevant
Funktionalität		X		
Zuverlässigkeit			X	
Benutzbarkeit		X		
Effizienz			X	
Wartbarkeit			X	
Übertragbarkeit		X		

Tabelle 6: Qualitätsanforderungen an die Anwendung - Auszug

Eine detaillierte Übersicht der Qualitätsanforderungen findet sich im Anhang G auf Seite 30.

3.10. Lastenheft / Fachkonzept

Das Grobkonzept der Anwendung ist wie folgt beschrieben:

”Die Anwendung ist eine Webapplikation, welche es dem Benutzer erlaubt den Inhalt eines Wikis in eine PDF- bzw. HTML-Datei zu konvertieren und diese anschließend herunterladen. Dazu kann der Benutzer selbst neue Wikis einpflegen. Gespeicherte Wikis sowie konvertierte Inhalte können mit anderen Nutzern geteilt werden.”

Eine detaillierte Übersicht über die Zielbestimmungen, den Produkteinsatz und die Produktübersicht findet sich in Anhang H auf Seite 31.

3.11. Zwischenstand

Teilphase	Geplant	Tatsächlich	Differenz
Sichtung des Lastenhefts	1h	1h	
Erstellen des Pflichtenhefts	2h	3h	+ 1h
Wirtschaftlichkeitsanalyse	1h	2h	+ 1h
Risikoanalyse	1h	1h	
”Make or Buy” Analyse	1h	1h	
Anwendungsfälle definieren	1h	1h	
Gesamt	7h	9h	+ 2h

Tabelle 7: Zwischenstand nach der Analysephase

4. Entwurfsphase

4.1. Zielplattform

Im folgenden wird dargelegt, aus welchen Gründen die gewählte Zielplattform anvisiert wird.

Programmiersprache JAVA, TypeScript

Das Backend der Anwendung wird mit der Programmiersprache Java in Version 7 umgesetzt. Dies ist primär mit der bereits vorhandenen Erfahrung der Projektmitarbeiter begründet. Außerdem ist die bereits existierende Ziellaufzeitumgebung (s. 4.1 für Java-Anwendungen) ausgelegt. Für die Frontend-Programmierung wird TypeScript in Version 1.6 verwendet. Dies ist damit begründet, dass wie in Kapitel 3.7 beschrieben der Einsatz eines JavaScript-Frameworks anvisiert wird. Des weiteren bietet TypeScript im Gegensatz zu JavaScript umfassendere statische Typprüfungen. Dies erleichtert die Programmierung und verhindert spätere Laufzeitfehler.

Application-Server Tomcat

Als Application-Server wird Tomcat in Version 7 eingesetzt. Dies liegt vor allem an dem leichtgewichtigen Umfang und den damit verbundenen geringen Installations- und Administrationsaufwand des Application-Servers.

Datenbank MySQL

Als Datenbanksystem wird MySQL verwendet. Dies ist in erster Linie mit der bereits vorhandenen Erfahrung der Projektmitarbeiter begründet.

Server

Die spätere Laufzeitumgebung inklusive der Installation der Anwendung in dieser ist, wie in Kapitel 1.5 beschrieben, zwar nicht Teil des Projekts, der Vollständigkeit halber ist hier allerdings in Kürze die vom Kunden bereitgestellte Server-Umgebung beschrieben.

- Server mit ausreichendem leistungsfähigem Intel / AMD Prozessor
- ausreichend Arbeitsspeicher
- Debian Linux Betriebssystem
- MySQL Datenbankserver
- Tomcat Application-Server

4.2. Architekturdesign

Im folgenden wird dargelegt, aus welchen Gründen die jeweilige Softwarearchitektur des Back- und Frontends gewählt wurde. Eine detaillierte Übersicht über die Wahl Architekturspezifischer Frameworks findet dagegen sich in Kapitel 3.7.

4.2.1. Serverarchitektur

Als Grundlage der Serverarchitektur wird das REST-Prinzip³ in Verbindung mit JSON als Datenformat gewählt.

Daraus ergeben sich folgende Vorteile:

Separation of Concerns Der Server hat klar definierte und abgegrenzte Aufgabengebiete. Dazu zählen das Verwalten (CRUD⁴) von persistenten Daten. Sowohl von solchen, die den Application-Lifecycle überleben (Datenbankeinträge), als auch von flüchtigen, wie etwa Session-Daten. Außerdem ist der Server für die Zugriffsbeschränkung auf die Daten zuständig.

Nicht zu den Aufgaben des Servers gehören dagegen z.B. Oberflächen-Logik oder das halten des generellen Status einer Session.

Applikationsübergreifende Schnittstelle Aufgrund des des REST-Prinzips ist leicht möglich, Anwendungsdaten gegenüber dritten, anwendungsfremden Diensten bereitzustellen, um so den Nutzen der Anwendung langfristig zu sichern.

Datenformat Durch die Wahl von JSON als Datenformat entfällt zum einen eine clientseitige Konvertierung, da JSON in der JavaScript-Umgebung eines Browsers ein natives Datenformat darstellt. Zum anderen kann bei der serverseitigen Konvertierung der Daten auf vorhandene Bibliotheken, beispielsweise Jackson⁵ zurückgegriffen werden.

Allgemein Die Wahl der Architektur weist in Verbindung mit der Programmiersprache Java den Vorteil auf, dass der JEE-Stack bereits Standardkomponenten für diese Anwendungsbereiche wie etwa JAX-RS⁶ aufweist.

4.2.2. Clientarchitektur

Clientseitig wird eine auf dem Flux⁷ Konzept basierende Model-View-Architektur verwendet.

Diese weist folgende Vorteile auf:

³<http://www.infoq.com/articles/designing-restful-http-apps-roth>, abgerufen am 07.10.2015

⁴Create, Update, Read, Delete

⁵<http://wiki.fasterxml.com/JacksonHome>, abgerufen am 07.10.2015

⁶<https://jcp.org/en/jsr/detail?id=339>, abgerufen am 07.10.2015

⁷<https://facebook.github.io/flux/docs/overview.html>, abgerufen am 07.10.2015

Trennung von Daten und Darstellung Durch die strikte Trennung von Daten (repräsentiert durch zustandslose Stores) und deren Darstellung (zustandsbehaftete Komponenten) werden Fehler in der Geschäfts- und Oberflächenlogik vermieden. Der unidirektionale Datenfluß der Flux-Architektur verhindert außerdem konkurrierende Zugriffe auf Daten.

Übersichtlichkeit Die starke Modularisierung (Aufteilung in Stores, Actions und Komponenten) erhöht die Übersichtlichkeit und erleichtert langfristig die Analyse bzw. das Refactoring des Frontends.

Erweiterbarkeit Aufgrund der bereits erläuterten Modularisierung und der strikten Objektorientierung des Frontends ist eine leichte Erweiterbarkeit von diesem gewährleistet.

4.3. Entwurf der Benutzeroberfläche

Die Benutzeroberfläche wird mit den Techniken HTML in Version 5 und CSS in Version 3 realisiert. JavaScript in Version 5 wird zur Unterstützung und Verbesserung der User Experience Verwendung finden.

4.3.1. Visuelles Konzept

Die Benutzeroberfläche wird in einzelne Bereiche unterteilt, um dem Benutzer eine durchgehend gleichbleibende Grundstruktur zu bieten und so die Übersichtlichkeit der Anwendung zu maximieren.

Layout Zu diesem Zweck wird die Oberfläche in einen Header, welcher die Navigationsleiste, sowie Angaben zum aktuellen Status beherbergt, sowie einen Hauptbereich, welcher die statusabhängigen Formulare und Daten anzeigt unterteilt. Der Hauptbereich ist wiederum in Spalten und Zeilen (je nach aktuellem Status) unterteilt, wobei die maximale Anzahl dieser Unterbereiche auf üblicherweise zwei bis drei beschränkt ist, um die Übersichtlichkeit zu gewährleisten.

Design Grundlagen Die Benutzeroberfläche wird nach den Grundlagen von Responsive Design⁸ entwickelt, um die Attraktivität und den Nutzwert der Anwendung für verschiedene Endgeräte wie z.B. Smartphones oder Tablets zu gewährleisten.

Visuelle Richtlinien Die Darstellung lehnt sich dabei zum einen an die Richtlinien von Material Design⁹ an, um einen modernen Gesamteindruck zu erzielen. Zum anderen orientiert sich das Farbschema der Oberfläche am Corporate Design des Auftraggebers.

⁸<http://alistapart.com/article/responsive-web-design>, abgerufen am 07.10.2015

⁹<https://www.google.com/design/spec/material-design>, abgerufen am 07.10.2015

4.3.2. Prototyp

Eine bildhafte Darstellung eines funktionslosen Prototypen, welcher die hier aufgeführten Visuellen Konzepte verdeutlichen soll findet sich in Anhang I auf Seite 33.

4.4. Datenmodell

4.5. Geschäftslogik

4.6. Maßnahmen zur Qualitätssicherung / Testkonzept

4.7. Pflichtenheft / Datenverarbeitungskonzept

4.8. Zwischenstand

5. Implementierungsphase

Hier erfolgen Details über die Implementierung

5.1. Implementierung der Datenstrukturen

5.2. Implementierung des Backends

5.3. Implementierung des Frontends

5.4. Implementierung der Tests

5.5. Zwischenstand

6. Abnahmephase

test

6.1. Zwischenstand

7. Einführungsphase

7.1. Zwischenstand

8. Dokumentation

8.1. Zwischenstand

9. Fazit

9.1. Soll- / Ist-Vergleich

9.2. Lesson-Learned

9.3. Ausblick

Anhang

A. Projektphasen

Projektphase	Geplante Zeit
Anforderungsanalyse	Gesamt: 7h
Sichtung des Lastenhefts	1h
Erstellen des Pflichtenhefts	2h
Wirtschaftlichkeitsanalyse	1h
Risikoanalyse	1h
"Make or Buy" Analyse	1h
Anwendungsfälle definieren	1h
Projektplanung	Gesamt: 4h
Erstellen des Projektplans	2h
Erstellen des Qualitätssicherungskonzepts	2h
Software-Architektur	Gesamt: 14h
Spezifizieren der Architektur-Grundlagen	8h
Erstellen des Datenmodells	4h
Erstellen des Testkonzepts	2h
Implementierung	Gesamt: 26h
Erstellen der Datenbanktabellen	2h
Entwicklung des Backends	7h
Entwicklung des Frontends	9h
Dokumentation des Quellcodes	2h
Testing	2h
Soll-Ist-Vergleich	2h
Puffer für Korrekturen	2h
Projektabschluss	Gesamt: 3h
Anwenderschulung	2h
Abnahme	1h
Sonstiges	Gesamt: 16h
Projektdokumentation	12h
Allgemeiner Puffer	4h
Gesamt	70h

Tabelle 8: Projektphasen detailliert

B. Risikoanalyse

Die folgende Tabelle listet alle für das Projekt als relevant betrachteten Risiken auf. Ermittlungsgrundlage dieser Risiken ist neben der persönlichen Erfahrung der Projektbeteiligten eine interne Mitarbeiterbefragung

Risiko	Ursache	Maßnahme	Eintrittswahr. nach der Maßnahme	Auswirkung nach der Maßnahme
Terminrisiken				
Nicht einhalten des Abnahmetermins	uneingeplante Hindernisse	keine	mittel	Start der Anwendungsnutzung verzögert sich
Technische Risiken				
Anwendung ist auf Zielplattform nicht lauffähig	Entwicklung gegen falsche Plattform	Anpassung der Anwendung	mittel	Zustätzlicher Zeitbedarf für Anpassungen
Personelle Risiken				
Ausfall des Projektleiters	Krankheit	keine	gering	Projekt kommt zum erliegen
Ausfall eines Projektmitarbeiters	Krankheit	keine	gering	Projekt kommt zum erliegen
Planungsrisiken				
Unterschätzung der Dauer einzelner Phasen	mangelnde Erfahrung	nachtrgäliche Planung	mittel	Verzögerung des Projektabschluss
Auslassung relevanter Phasen	mangelnde Erfahrung	nachträgliche Planung	gering	Verzögerung des Projektabschluss
Risiken der Analyse und Konzeption				
Fehlerhafte Ist-Analyse	ungenauere Analyse	Nachbesserung d. Analyse	gering	iterativer Rücksprung im Entwicklungsprozess
Fehlerhafte Soll-Analyse	fehlerhafte Spezifikation	Nachbesserung d. Analyse	gering	iterativer Rücksprung im Entwicklungsprozess

Risiko	Ursache	Maßnahme	Eintrittswahr. nach der Maßnahme	Auswirkung nach der Maßnahme
Fehlerhafte Architektur / Konzept	ungenau Spezifikation	Neukonzeption	gering	iterativer Rücksprung im Entwicklungsprozess
Realisierungsrisiken				
Unerwartete Entwicklungsprobleme	nicht bestimmbar	teilweise Neukonzeption	mittel	verlängte Entwicklungszeit
Betreuungs- und Wartungsrisiken				
Breaking-Changes	MediaWiki Update	Anpassung der Geschäftslogik	hoch	keine

Tabelle 9: Risikoanalyse detailliert

C. "Make or Buy"-Bewertung

Grundlage der im Folgenden betrachteten Alternativen ist eine umfängliche Recherche bezüglich der Kernfrage "Welche Lösungen zur Erstellung von Handbüchern aus Wikis existieren bereits?". Betrachtet werden hier nur die stärksten Ergebnisse dieser Recherche, um die Übersichtlichkeit der Analyse zu bewahren.

Alternative	Nutzen	Wirtschaftlichkeit
MediaWiki Erweiterung		
einfache Installation		⊕
Installation für jedes Wiki nötig		⊖
Workflow umständlich	⊖	⊖
Ergebnis bedarf händische Anpassung	⊖⊖	⊖⊖
Komptibilitätsprobleme mit MediaWiki		⊖
Erstellung der Anwendung durch Dienstleister		
spezialisierte Anwendung	⊕⊕	⊖ ⊖ ⊖
Wartung durch Dienstleister nötig		⊖⊖
Risikoausslagerung		⊕⊕
Eigenständige Erstellung der Anwendung		
spezialisierte Anwendung	⊕⊕	⊖⊖
Wartung durch eigene Mitarbeiter	⊕	⊕
Tragen von Risiken		⊖

Tabelle 10: Detaillierte "Make or BuyBewertung

D. Kostenübersicht nach Projektphasen

Grundlage der Folgenden Kostenübersicht sind die in Anhang A aufgeführten Phasen des Projekts. Als Rechnungsgrundlage wird der in Kapitel 3.2 ermittelte Stundensatz von 17 € herangezogen.

Phase	Kosten
Anforderungsanalyse	119 €
Projektplanung	68 €
Software-Architektur	238 €
Implementierung	442 €
Projektabschluss	51 €
Sonstiges	272 €
Gesamt	1190 €

Tabelle 11: Kosten unterteilt nach Projektphasen

E. Nutzwertanalyse

E.1. Backend-Architektur

Im Folgenden wird über eine Nutzwertanalyse die beste Backend-Architektur ausgewählt. Die hier betrachteten Alternativen sind die stärksten Ergebnisse einer vorangegangenen Recherche.

Eigenschaft	Gewichtung	JEE-Stack	GNS-Framework	SrpingMV
REST-API integriert	3	5	0	4
Trennung von Back- und Frontend	4	5	0	4
Dokumentation	2	3	1	5
Testbarkeit	2	2	1	3
Refactoring (ggf. durch Dritte)	3	3	4	2
Gesamt	14	18	6	18
Nutzwert		3,86	1,14	3,57

Tabelle 12: Detaillierte Nutzwertanalyse bezüglich der Backend-Architektur

E.2. Frontend-Architektur

Im Folgenden wird über eine Nutzwertanalyse die beste Frontend-Architektur ausgewählt. Die hier betrachteten Alternativen sind die stärksten Ergebnisse einer vorangegangenen Recherche.

Eigenschaft	Gewichtung	JSF	Angular	kein Framework	HorCrux ¹⁰
Bootstrap-Aufwand	2	3	3	5	2
Modularität	4	4	3	1	5
Erweiterbarkeit	5	3	4	0	5
Dokumentation	2	3	5	4	1
User-Experience	4	3	5	2	5
Kompatibilität	2	4	2	5	2
Gesamt	19	20	22	17	20
Nutzwert		3,32	3,79	2,11	3,95

Tabelle 13: Detaillierte Nutzwertanalyse bezüglich der Front-Architektur

¹⁰Ein von Oliver Herrmann entwickeltes TypeScript Framework, welches u.a. Web Components und Flux-Architektur beinhaltet.

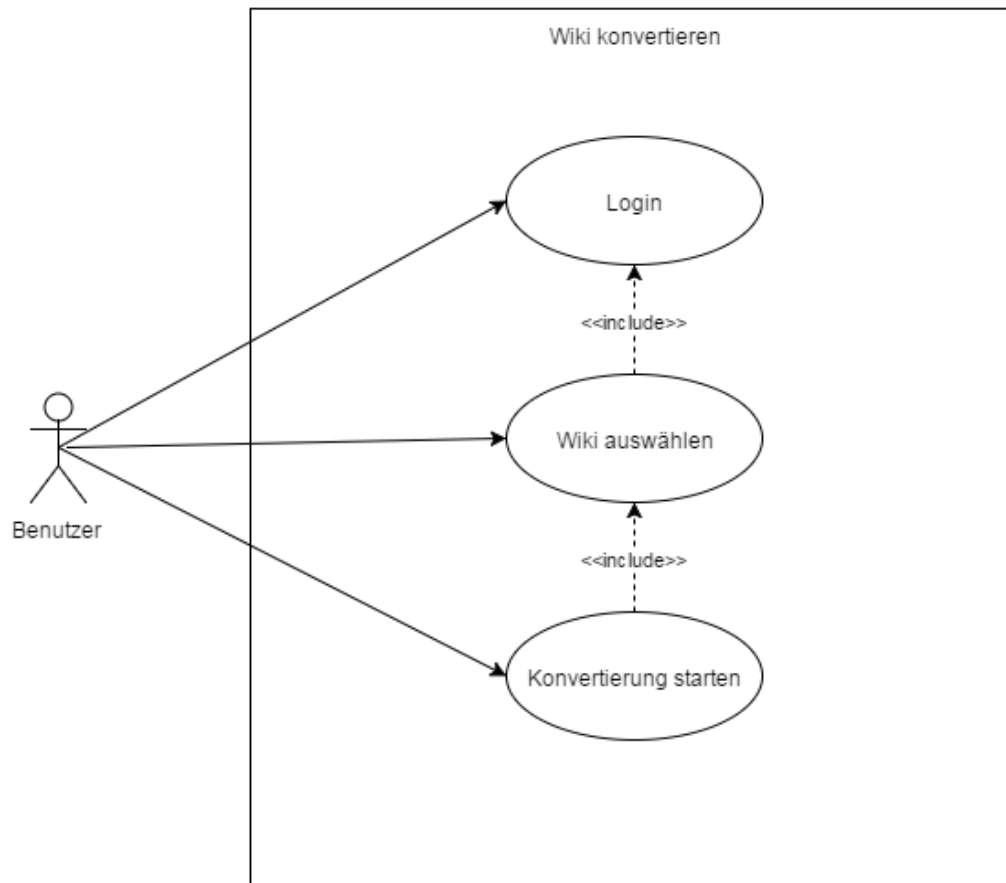


Abbildung 1: Use-Case "Wiki konvertieren"

F. Use-Case-Diagramme

F.1. Wiki konvertieren

F.2. Wiki bearbeiten

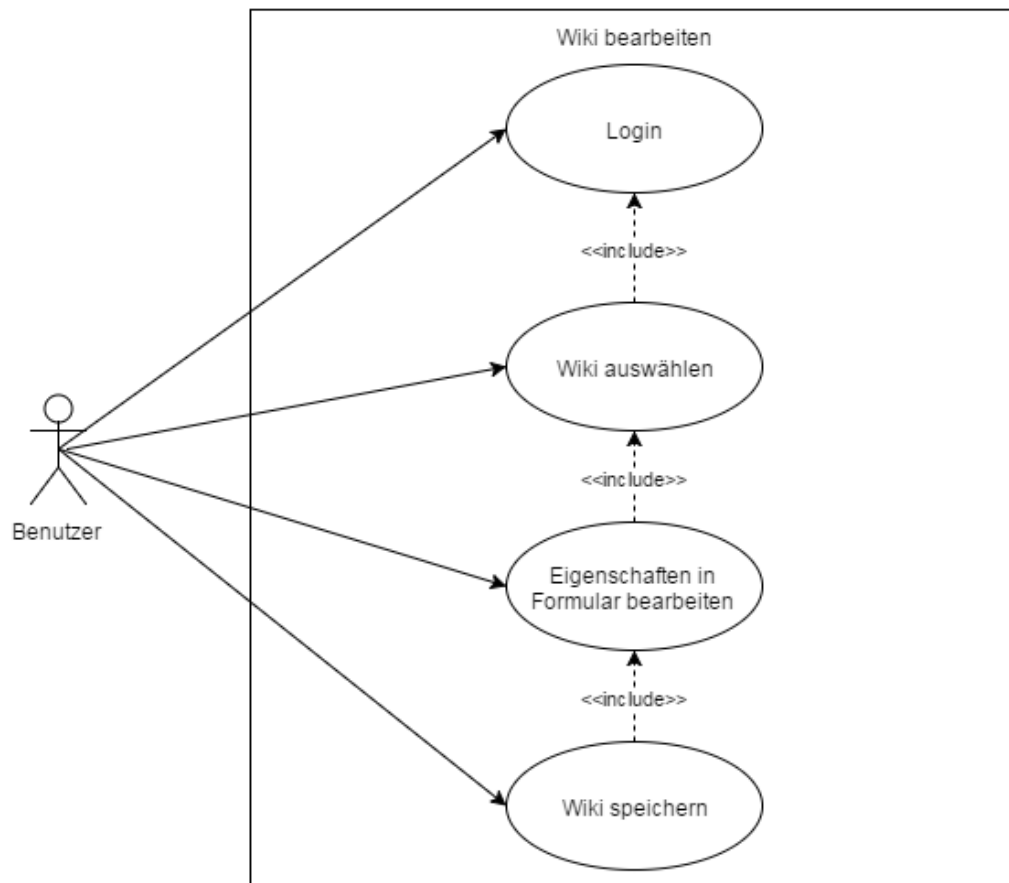


Abbildung 2: Use-Case "Wiki bearbeiten"

G. Qualitätsanforderungen (Auszuf aus Pflichtenheft)

Qualitätsmerkmal	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit			X	
Richtigkeit		X		
Interoperabilität	X			
Sicherheit		X		
Ordnungsmäßigkeit			X	
Konformität			X	
Zuverlässigkeit				
Reife			X	
Fehlertoleranz			X	
Wiederherstellbarkeit		X		
Konformität			X	
Benutzbarkeit				
Verständlichkeit	X			
Erlernbarkeit		X		
Bedienbarkeit		X		
Attraktivität			X	
Konformität				X
Effizienz				
Zeitverhalten		X		
Verbrauchsverhalten			X	
Konformität			X	
Wartbarkeit				
Analysierbarkeit			X	
Modifizierbarkeit		X		
Stabilität			X	
Testbarkeit				X
Konformität			X	
Übertragbarkeit				
Anpassbarkeit		X		
Installierbarkeit		X		
Koexistenz	X			
Austauschbarkeit		X		
Konformität				X

Tabelle 14: Qualitätsanforderungen an die Anwendung (Pflichtenheft)

H. Lastenheft (Auszug)

Zielbestimmung

Musskriterien

- Benutzerverwaltung
 - Benutzer anlegen
 - Benutzerdaten editieren
 - Benutzer löschen
- Wikiverwaltung
 - Wiki anlegen
 - Wikidaten editieren
 - Wiki löschen
 - Handbuch aus Wiki generieren
 - Handbuch runterladen

Wunschkriterien

- Handbuchversionen
 - Handbuchhistorie eines Wikis einsehen
 - Handbuch aus Historie löschen
 - Älteres Handbuch runterladen
- Handbuchgenerierung
 - Aktuellen Fortschritt während Generierung anzeigen
 - Erstellung für Wikis sperren, während aktuelle Generierung läuft
- Gruppenverwaltung
 - Gruppen anlegen
 - Gruppen löschen
 - Benutzer Gruppen zuteilen
 - Rechte Gruppen zuteilen

Abgrenzungskriterien

- Single-Sign-On über GNS internes ActiveDirectory
- Anlegen / Konfigurieren der Laufzeitumgebung innerhalb der GNS DMZ

Produkteinsatz

Anwendungsbereiche

Technischer/administrativer Anwendungsbereich.

Zielgruppen

Die Zielgruppe besteht ausschließlich aus Mitarbeitern der GNS mbH. Vornehmlich wird das Produkt durch Angestellte der Abteilung benutzt werden.

Betriebsbedingungen

Das Produkt wird als Webanwendung bereitgestellt. Das Produkt ist über das Internet erreichbar. Es wird eine Up-Time von nahezu 100% angestrebt.

Produktübersicht

Das Produkt ist eine Webapplikation, welche es dem Benutzer erlaubt den Inhalt eines existierenden Mediawikis in eine PDF- respektive HTML-Datei zu konvertieren und diese anschließend herunterzuladen. Hierzu kann der Nutzer selbstständig neue Wikis indizieren um diese für das spätere Konvertieren bereitzustellen. Konvertierte Mediawikis können anderen Benutzern ebenfalls zum download bereitgestellt werden.

I. Oberflächenprototyp



Abbildung 3: Prototyp - Layout für Desktop-Browser

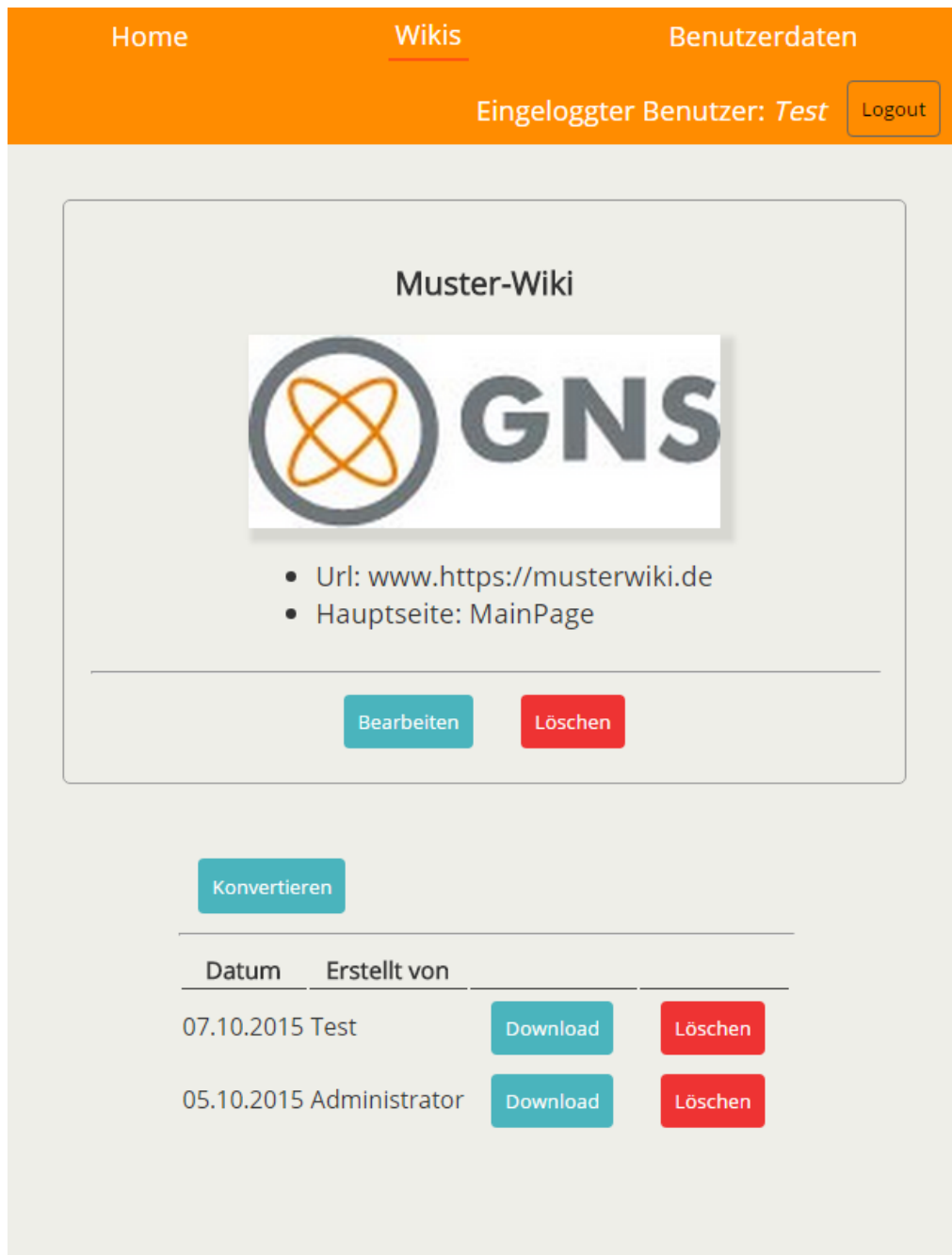


Abbildung 4: Prototyp - Layout für Mobile-Browser