

# Software Requirements Specification SRS

## STM32 Knowledge Base

Version 0.02

**Dokument-Historie**

<b>Version</b>	<b>Status</b>	<b>Datum</b>	<b>Verantwortlicher</b>	<b>Änderungsgrund</b>
<b>0.01</b>	Revised	22/03/2017	Max Pessl Erika Wood	1. Entwurf
<b>0.02</b>	In Bearbeitung	27/03/2017	Max Pessl Erika Wood	Diverse Anpassungen nach Feedback

## **Abkürzungen**

KB	Knowledge Base
ILV	Integrierte Lehrveranstaltung
MAD	Mobile App Development
TDD	Test Driven Development

**Inhaltsverzeichnis**

<b>1. EINLEITUNG</b>	<b>1</b>
1.1. ZWECK	1
1.1.1. <i>Hintergründe und Ziele des Projekts</i>	1
1.1.2. <i>organisatorische Einbettung</i>	1
1.1.3. <i>technische, wirtschaftliche, organisatorische, ergonomische Ziele</i>	1
1.2. PRODUKTUMFANG	1
1.2.1. <i>Musskriterien</i>	1
1.2.2. <i>Wunschkriterien</i>	1
1.2.3. <i>Abgrenzungskriterien</i>	1
1.2.4. <i>Bewertungskriterien für die Beurteilung der Zielsetzung</i>	1
1.2.5. <i>Kostenrahmen</i>	1
1.3. DEFINITIONEN, AKRONYME, ABKÜRZUNGEN	1
1.4. REFERENZEN	1
1.5. ÜBERSICHT	1
<b>2. ALLGEMEINE ÜBERSICHT</b>	<b>2</b>
2.1. BESCHREIBUNG DER AUSGANGSSITUATION (IST-ZUSTAND)	2
2.1.1. <i>Organisation, Arbeitsabläufe</i>	2
2.1.2. <i>technischer Prozess</i>	2
2.1.3. <i>Datendarstellung und Mengengerüst</i>	2
2.2. PRODUKTEINSATZ	2
2.2.1. <i>Anwendungsbereiche</i>	2
2.2.2. <i>Zielgruppen, Qualifikationsniveau</i>	2
2.2.3. <i>Betriebsbedingungen</i>	2
2.2.4. <i>allgemeine Restriktionen</i>	2
2.2.5. <i>Annahmen und Abhängigkeiten</i>	2
2.3. PRODUKTUMFELD	2
2.3.1. <i>Systemschnittstelle</i>	2
2.3.2. <i>Benutzerschnittstelle</i>	2
2.3.3. <i>Hardwareschnittstelle</i>	2
2.3.4. <i>Softwareschnittstelle</i>	2
2.3.5. <i>Kommunikationsschnittstelle</i>	2
2.3.6. <i>Speicherbeschränkungen</i>	2
2.3.7. <i>Operationen</i>	2
2.3.8. <i>Anpassung von standortspezifischen Anforderungen</i>	2
2.4. PRODUKTFUNKTIONALITÄT	3
2.5. BENUTZERCHARAKTERISTIK	3
2.6. RANDBEDINGUNGEN	3
2.7. ANNAHMEN UND ABHÄNGIGKEITEN	3
2.8. VERZÖGERUNGEN	3
<b>3. FUNKTIONALE ANFORDERUNGEN</b>	<b>3</b>
<b>4. NICHT FUNKTIONALE ANFORDERUNGEN</b>	<b>3</b>
4.1. ANFORDERUNGEN AN DIE DATENHALTUNG	3
4.1.1. <i>allgemeine Beschreibung der Daten</i>	3
4.1.2. <i>Archivierung</i>	3
4.2. ANFORDERUNGEN AN DIE BENUTZEROBERFLÄCHE	3
4.2.1. <i>allgemeine Anforderungen an die Oberfläche</i>	3
4.2.2. <i>Berechtigungen</i>	3
4.2.3. <i>individuelle Anpassung der Oberfläche</i>	3
4.2.4. <i>Bildschirmlayout</i>	3
4.2.5. <i>Prüfungen: feldbezogene und feldübergreifende</i>	3
4.2.6. <i>Drucklayout, Tastaturbelegung</i>	3
4.2.7. <i>Dialogstruktur, Dialogabläufe</i>	3
4.2.8. <i>Hilfesystem</i>	3
4.3. LEISTUNGSANFORDERUNGEN	4
4.3.1. <i>zeitbezogene oder umfangsbezogene Produktleistungen</i>	4
4.3.2. <i>Performance-Daten, Dialogantwortzeiten</i>	4
4.3.3. <i>Maximaler und durchschnittlicher Datenumfang bzw. Datendurchsatz</i>	4

4.3.4.	<i>Genauigkeit von Berechnungen</i>	4
4.4.	ANFORDERUNG FÜR INBETRIEBNAHME UND EINSATZ	4
4.4.1.	<i>Sicherheitsziele</i>	4
4.4.2.	<i>Betriebssicherheit</i>	4
4.4.3.	<i>Installationsprozedur</i>	4
4.4.4.	<i>Pilot- bzw. Probetrieb</i>	4
4.4.5.	<i>Fehlerreaktion, Garantie, Service, »Wiederanlauf«</i>	4
4.4.6.	<i>Schulungen</i>	4
4.5.	QUALITÄTSANFORDERUNGEN	4
4.5.1.	<i>Qualitätsmerkmale</i>	4
4.5.2.	<i>Qualitätssicherung</i>	4
4.5.3.	<i>Qualitätsnachweis</i>	4
4.5.4.	<i>Offenlegung der Qualitätskontrollpläne</i>	4
4.5.5.	<i>Berichte, Protokolle zum Nachweis des Vorgehens gemäß der Qualitätskontrollpläne</i>	4
4.6.	ANFORDERUNG AN DIE ENTWICKLUNG	5
4.6.1.	<i>Entwurfsrestriktionen</i>	5
4.6.2.	<i>Entwicklungs-Umgebung</i>	5
4.6.3.	<i>Projekt-Organisation</i>	5
4.6.4.	<i>Projektplanung</i>	5
4.6.5.	<i>Projektüberwachung</i>	5
4.6.6.	<i>Projektsteuerung</i>	5
4.6.7.	<i>Konfigurationsmanagement</i>	5
4.6.8.	<i>Änderungsmanagement</i>	5
4.6.9.	<i>Testanforderungen</i>	5
4.6.10.	<i>Reviews, Refactoring</i>	5
5.	ANHANG	6
5.1.	GLOSSAR, AKRONYME UND ABKÜRZUNGEN	6
5.2.	DATENKATALOG	6
5.3.	DIALOGMASKEN	6
5.4.	DRUCKMASKEN	6
5.5.	GLOBALE TESTSZENARIEN/TESTFÄLLE	6
5.6.	ANZUWENDENDE DOKUMENTE	6
5.7.	DOKUMENTATIONSANFORDERUNGEN	6
5.8.	LISTE DER SOFTWARELIEFERUNGEN	6
5.9.	BEISTELLUNGEN DURCH DEN AUFTRAGGEBER	6
5.10.	PROJEKTORGANIGRAMM	6
5.11.	PROJEKTSTRUKTURPLAN	6
5.12.	HAUPT-TERMINDATEN	6
6.	INDEX	6
7.	ANHANG	7
7.1.	GLOSSAR	7

# 1. Einleitung

## 1.1. Zweck

### 1.1.1. Hintergründe und Ziele des Projekts

Im Zuge der integrierten Lehrveranstaltung (ILV) Mobile App Development (MAD) wurde als Aufgabe eine Projektgruppenarbeit aufgetragen. Ziel ist es, eine App zu entwickeln, welche die erlernten Inhalte der ILV abdeckt.

Die Projektgruppe aus folgenden Personen zusammen:

- Maximilian Pessl
- Erika Wood

Der Name der App lautet STM32 Knowledge Base (STM32 KB) und umfasst eine Wissensdatenbank rund um den MicroController ( $\mu$ C) STM32 F4xx.

### 1.1.2. Organisatorische Einbettung

Das Projekt dient der Anwendung des erlernten Wissens aus der Vorlesung und Übungseinheit von Mobile App Development

### 1.1.3. Technische, wirtschaftliche, organisatorische Ziele

Die App erleichtert und unterstützt Studierende sowie Lehrende in der Programmierung von Funktionen eines  $\mu$ C.

## 1.2. Produktumfang

### 1.2.1. Musskriterien

Musskriterien umfassen folgende Use Cases:

- Artikel lesen
- Artikel schreiben
- Autor registrieren
- Autor login

### 1.2.2. Wunschkriterien

Wunschkriterien lauten wie folgt:

- Should
  - Artikel versenden
  - Autor kontaktieren
- Could
  - Artikel kommentieren
  - Artikel favorisieren
  - Artikel suchen

### 1.2.3. Abgrenzungskriterien

Die App wird nur für Android-Endgeräte mit mindestens API-Level 15 Ice Cream Sandwich (4.0.3) entwickelt. Die verwendete Entwicklungsumgebung ist Android Studio in der Version ab 2.3.

#### 1.2.4. **Bewertungskriterien für die Beurteilung der Zielsetzung**

Die Abnahme findet im Rahmen einer Präsentation statt. Hierbei werden die umgesetzten Use Cases anhand eines Live Previews vorgestellt und den jeweiligen MoSCoW Kriterien zugeordnet und bewertet.

## 2. Allgemeine Übersicht

### 2.1. *Beschreibung der Ausgangssituation (Ist-Zustand)*

#### 2.1.1. **Organisation**

Zusätzlich zu den Übungseinheiten im Rahmen der ILV finden regelmäßige Meetings à 2-4 Stunden statt.

#### 2.1.2. **Technischer Prozess**

Folgende Vorgehensmodelle kommen zum Einsatz:

- Scrum
  - Erstellung von Product Backlogs
  - Daily Scrum Meetings
- KanBan
  - Gemeinsames Domainwissen
- Evolutionäre Entwicklung
  - Exploratory Prototyping
- xTreme Programming
  - Pair Programming
  - Collective Ownership
  - Continuous Integration (GitHub)
  - On-site Customer
- TestDriven Development (TDD)
  - Start mit einfachsten Sache, die funktionieren kann
  - JUnit Tests fähige Methoden (verifizierbare Rückgabewerte)

#### 2.1.3. **Datendarstellung und Mengengerüst**

- Statisch
  - Bereits während der Entwicklung eingespielte Artikel
  - Es werden zu Anfangs mindestens 3 Artikel zu je 2 Kategorien abzurufen sein
- Dynamisch
  - Neu angelegte Artikel durch Autoren über die App
  - Autoren
- Artikel bestehen aus 4 Teilen
  - Titel
  - Bild
  - Codeblock
  - Textuelle Codebeschreibung

## 2.2. *Produkteinsatz*

### 2.2.1. **Anwendungsbereiche**

Im Zuge der ILVs für technische Studiengänge, welche  $\mu$ C implementieren.

### 2.2.2. **Zielgruppen**

Als Zielgruppe sind einerseits aktive Studenten technischer Studienrichtungen sowie Lehrende, die die App als rasches Nachschlagewerk heranziehen können.

### 2.2.3. **Allgemeine Restriktionen**

Online Betrieb ist nur bei bestehender Internetverbindung möglich:

- Laden von neuen Artikel
- Schreiben von neuen Artikeln
- Autorenregistrierung
- Autoren-Login

Offline Betrieb ermöglicht das Lesen bestehender Artikel.

## 2.3. *Produktumfeld*

### 2.3.1. **Systemschnittstelle**

Externe Datenbank Anbindung an Raspberry Pi (MySQL Datenbank)

### 2.3.2. **Benutzerschnittstelle**

Android App

### 2.3.3. **Hardwareschnittstelle**

Android Smartphone

### 2.3.4. **Softwareschnittstelle**

- Email versenden bei folgenden Use Cases
  - Artikel versenden
  - Autor kontaktieren

### 2.3.5. **Kommunikationsschnittstelle**

### 2.3.6. **Speicherbeschränkungen**

150 MB für die App



### 3. Funktionale Anforderungen

Use Case	Autor registrieren
Kurzbeschreibung	Ein Anwender kann sich für die STM32 KB (Knowledge Base) registrieren
Vorbedingung	
Nachbedingung	Autor ist registriert
Fehlersituationen	
Systemzustand im Fehlerfall	
Akteure	<ul style="list-style-type: none"> <li>- Autor, primär, aktiv, menschlich</li> <li>- System, primär, passiv</li> </ul>
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer öffnet die App und wählt registrieren (Sign up) aus.</li> <li>2. Benutzer befüllt folgende Eingabefelder <ul style="list-style-type: none"> <li>- Vorname und Nachname</li> <li>- Email</li> <li>- Password (mind 8 Zeichen, Buchstaben/Zahlen)</li> </ul> </li> <li>3. Benutzer muss die Allgemeinen Geschäftsbedingungen akzeptieren, um fortfahren zu können. <ol style="list-style-type: none"> <li>3.a Hinweis beinhaltet Information, dass der Autor erst nach erfolgter Freischaltung berechtigt ist, Artikel zu verfassen</li> </ol> </li> <li>4. Benutzer muss 4 Fragen beantworten, um die Registrierung beantragen zu können <ul style="list-style-type: none"> <li>- Die Fragen werden aus 20 vordefinierten Fragen zufällig ausgewählt</li> <li>- Die Fragen umfassen das Thema µC Programmierung, welches in der Vorlesung besprochen wird</li> <li>- Die Fragen werden von der App ausgewertet</li> <li>- Wenn 3 von 4 Fragen korrekt sind, wird der Status Autor in der DB gesetzt</li> </ul> </li> <li>5. Toastmessage an den Benutzer – entweder „Freischaltung erfolgreich“ oder „Erforderliche Anzahl korrekter Antworten nicht erreicht“</li> </ol>
Alternativabläufe	<ul style="list-style-type: none"> <li>- Benutzer bricht die Registrierung ab und verwendet die App als Leser (lesender Benutzer)</li> <li>- Registrierung nicht möglich: Toastmessage an Benutzer „Registrierung derzeit nicht möglich“</li> </ul>
Trigger	App Button „Sign up“ am Start Screen

Use Case	Autor login
Kurzbeschreibung	Ablaufbeschreibung für den Login eines Autors
Vorbedingung	Benutzer wurde durch Administrator freigeschalten
Nachbedingung	Autor eingeloggt Benutzer sieht Autoren View (Edit in Menüleiste)
Fehlersituationen	
Systemzustand im Fehlerfall	
Akteure	<ul style="list-style-type: none"> <li>- Autor, primär, aktiv, menschlich</li> <li>- System, primär, passiv</li> </ul>
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer öffnet die App und klickt den Sign-in Button</li> <li>2. Der Benutzer gibt Username / Passwort Kombination ein</li> <li>3. Das System checkt die Eingaben <ul style="list-style-type: none"> <li>- Die App übermittelt die eingegebenen Daten an die Server Komponente</li> <li>- Diese prüft die Kombination, bestehend aus Username und Passwort mit der persistent hinterlegten Username und Passwort Kombination</li> <li>- Bei positiver Überprüfung meldet die Server Komponente folgende Parameter: <ul style="list-style-type: none"> <li>- OK &lt;Username&gt; Author Yes</li> </ul> </li> </ul> </li> <li>4. Der Benutzer wird auf den Autoren View weitergeleitet</li> </ol>
Alternativabläufe	Falsche Benutzereingabe: <ul style="list-style-type: none"> <li>- Toastmessage „Username oder Passwort nicht korrekt“</li> <li>- Parameter der Server Komponenten Antwort: <ul style="list-style-type: none"> <li>Keine definiert</li> </ul> </li> <li>- Button für Passwort anfordern</li> </ul>
Trigger	APP `Login View`

<b>Use Case</b>	<b>Artikel lesen</b>
Kurzbeschreibung	Benutzer können Artikel lesen
Vorbedingung	Artikel wird von DB eingespielt und am Display dargestellt oder Artikel wurde bereits lokal für den Offline Betrieb gespeichert
Nachbedingung	-
Fehlersituationen	
Systemzustand im Fehlerfall	
Akteure	<ul style="list-style-type: none"> <li>- Benutzer, primär, aktiv, menschlich</li> <li>- System, primär, passiv</li> </ul>
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer clicked beim Start Screen auf den „Start reading button“</li> <li>2. Article Screen erscheint</li> <li>3. Benutzer wählt aus dem Spinner Menü gewünschtes Modell</li> <li>4. Die jeweiligen Module zu dem ausgewählten Modell werden angezeigt</li> <li>5. Benutzer clicked auf den Headline link des gewünschten Moduls</li> <li>6. Artikel reading screen erscheint mit dem entsprechenden Artikel</li> </ol>
Alternativabläufe	Benutzer clicked Back/Next-Button oder ArtikelAuswahl Button
Trigger	Button click auf „Start reading“

Use Case	Artikel schreiben
Kurzbeschreibung	Autor kann einen neuen Artikel schreiben oder einen bestehenden Artikel verändern
Vorbedingung	-Benutzer besitzt Autorenberechtigung
Nachbedingung	Änderungen wurden gespeichert
Fehlersituationen	
Systemzustand im Fehlerfall	
Akteure	<ul style="list-style-type: none"> <li>- Autor, primär, aktiv, menschlich</li> <li>- System, primär, passiv</li> </ul>
Standardablauf	<ol style="list-style-type: none"> <li>1. Autoren Screen anzeigen</li> <li>2. Autor clicked auf Edit/Create in der Menüleiste</li> <li>3. Edit öffnet die EditArticle View mit dem aktuellen Artikel</li> <li>4. Create führt zum CreateArticleView Screen</li> <li>5. Änderungen können unter dem Menüpunkt File offline/lokal gespeichert werden.</li> <li>6. Mit „Submit“ werden die Änderungen/der Artikel in die DB eingespielt <ul style="list-style-type: none"> <li>- Die App ruft die Server Komponente über die bestehende Internet Verbindung auf und übergibt die Parameter, die einen Artikel definieren: <ul style="list-style-type: none"> <li>- Title</li> <li>- Articletext</li> <li>- Code</li> <li>- Picture</li> </ul> </li> <li>- Die übermittelten Parameter werden persistent mit einem Zeitstempel und einem NewArticle Attribut hinterlegt</li> <li>- Die Server Komponente liefert eine positive Antwort</li> </ul> </li> <li>7. Das System signalisiert dem Autor, dass der Artikel erfolgreich übermittelt worden ist</li> </ol>
Alternativabläufe	<ul style="list-style-type: none"> <li>- Autor clicked „Back/Next“ Button oder wählt einen anderen Artikel mit „Artikel-Auswahl“</li> <li>- Toastmessage „Änderungen speichern“, falls etwaige Änderungen noch nicht gespeichert wurden</li> </ul>
Trigger	Autor clicked auf Edit/Create

## 4. Nicht Funktionale Anforderungen

### 4.1. *Anforderungen an die Datenhaltung*

#### 4.1.1. **Allgemeine Beschreibung der Daten**

Es werden den Use Case entsprechend spezifische Daten für die Server Komponente übermittelt. Antworten der Server Komponente werden mit OK bzw Yes oder No an die App übermittelt.

Die Parameter für einen Login umfassen:

- Username entspricht text1
- Password entspricht text2

Die Parameter für einen Artikel umfassen:

- Title
- Picture
- Articletext
- Code

Klassendiagramm siehe Anhang 5.4

#### 4.1.2. **Archivierung**

### 4.2. *Anforderungen an die Benutzeroberfläche*

#### 4.2.1. **Allgemeine Anforderungen an die Oberfläche**

- Flache Menüebenen bei Artikelauswahl
- Menüführung mit maximal 3 Ebenen
  - Im ersten Entwurf sind es eine Spinner der die Kategorie definiert
  - Je nach Kategorie werden in einem ScrollView die Artikel angezeigt
- Übersichtliche Gestaltung
- Harmonische Farben

#### 4.2.2. **Berechtigungen**

- Autoren dürfen schreiben
- Jeder Benutzer darf Artikel lesen

#### 4.2.3. **Bildschirmlayout**

Balsamiq Mockups siehe Anhang 5.3

#### 4.2.4. **Hilfesystem**

- Bedienungseinführung der App beim Startscreen sowie Abruf mittels Info-Button

### *4.3. Leistungsanforderungen*

#### **4.3.1. Performance-Daten, Dialogantwortzeiten**

- Server Time-Out auf 30sec
- Toastmessages erhalten den Parameter LENGTH\_SHORT

#### **4.3.2. Maximaler und durchschnittlicher Datenumfang bzw. Datendurchsatz**

- Maximale Artikelgröße wird in der nächsten Version spezifiziert

### *4.4. Anforderung für Inbetriebnahme und Einsatz*

#### **4.4.1. Sicherheitsziele**

Ausführung in nächster Version

### *4.5. Anforderung an die Entwicklung*

#### **4.5.1. Entwicklungs-Umgebung**

Android Studio Version ab 2.3 und Android API Level 15 IceCream Sandwich 4.03

#### **4.5.2. Projekt-Organisation/Planung/Überwachung/Steuerung**

Vorgehensmodelle laut Pkt 2.1.2

#### **4.5.3. Testanforderungen**

JUnit-Testfälle siehe Anhang 5.4- sind noch zu spezifizieren

## 5. Anhang

### 5.1. Datenkatalog

Es wurde für die Server Komponente eine Relationale Datenbank definiert die folgende Tabellen umfasst:

- Users
  - ID
  - Username
  - Password
  - Timestamp
  - IsAuthor

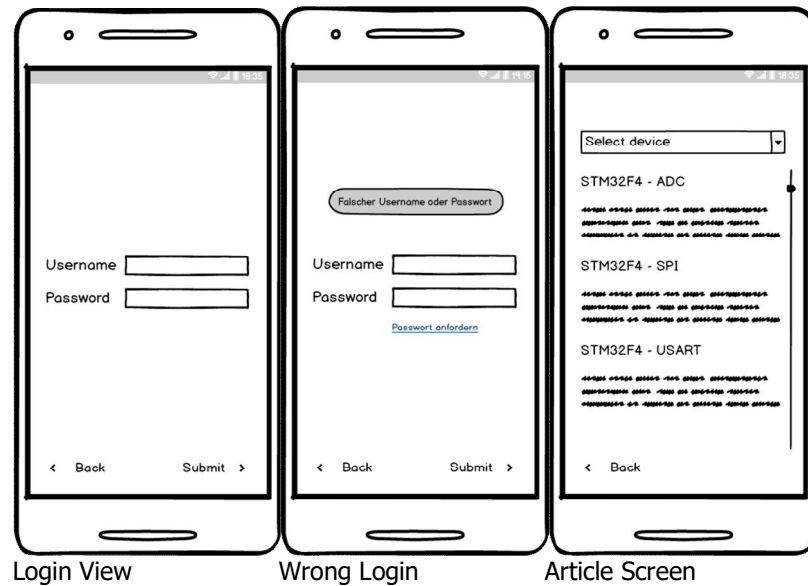
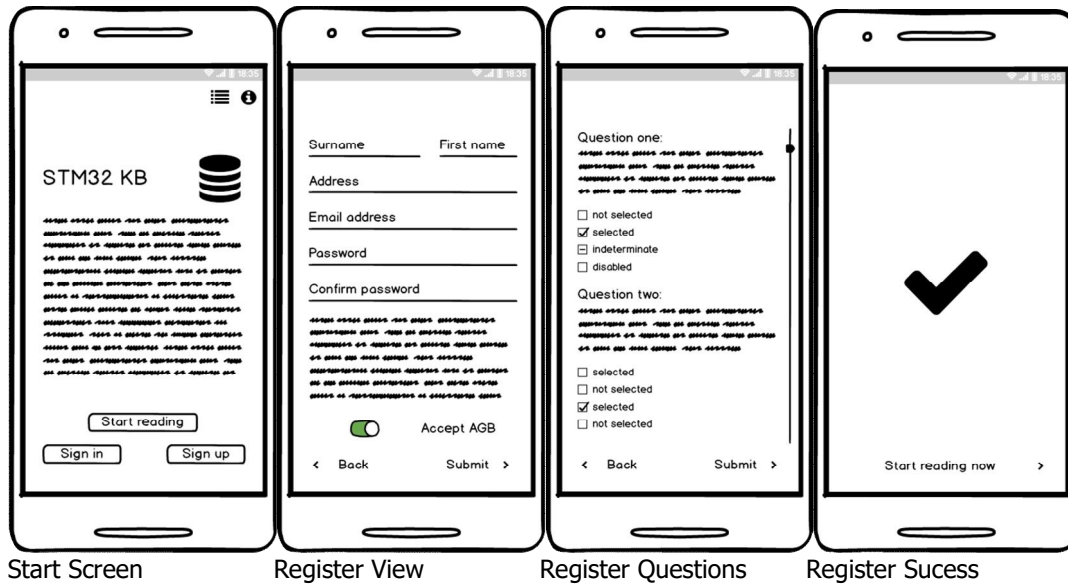
#	Name	Typ	Kollation	Attribute	Null	Standard	Extra
1	<u>ID</u>	int(11)			Nein	kein(e)	AUTO_INCREMENT
2	<u>Username</u>	char(10)	latin1_swedish_ci		Nein	kein(e)	
3	<u>Password</u>	char(32)	latin1_swedish_ci		Nein	kein(e)	
4	<u>Registrationtime</u>	timestamp		on update CURRENT_TIMESTAMP	Nein	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
5	<u>IsAuthor</u>	tinyint(1)			Nein	kein(e)	

- Articles
  - ID
  - Title
  - Articletext
  - Code
  - Picture
  - Publishdate

#	Name	Typ	Kollation	Attribute	Null	Standard	Extra
1	<u>ID</u>	int(11)			Nein	kein(e)	AUTO_INCREMENT
2	<u>Title</u>	char(255)	latin1_swedish_ci		Nein	kein(e)	
3	<u>Articletext</u>	text	latin1_swedish_ci		Nein	kein(e)	
4	<u>Code</u>	tinyint(1)			Nein	kein(e)	
5	<u>Picture</u>	char(100)	latin1_swedish_ci		Nein	kein(e)	
6	<u>Publishdate</u>	timestamp			Nein	CURRENT_TIMESTAMP	

## 5.2. Druckmasken

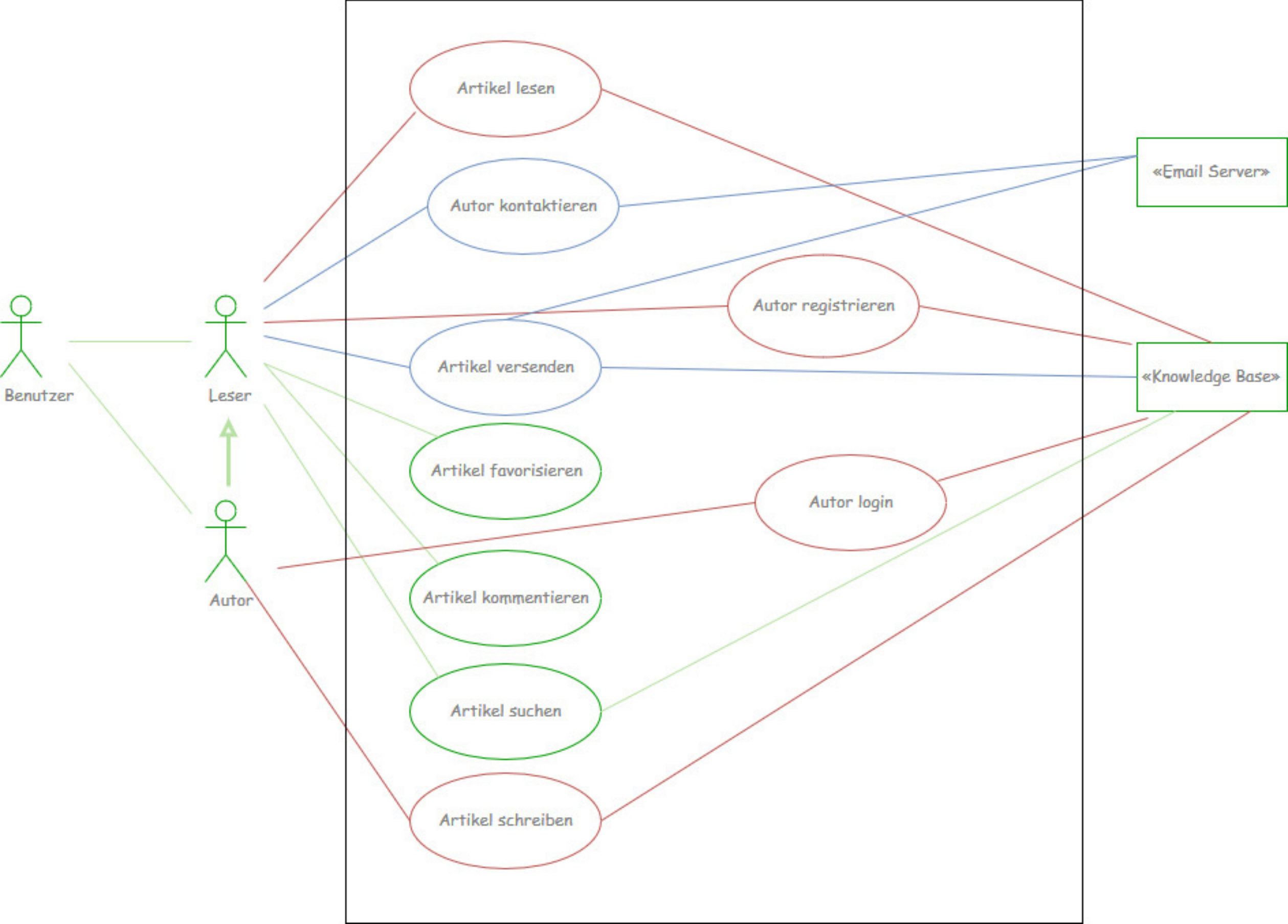
MockUp PDF







### 5.3. *UseCase Diagram*



#### **5.4.   *Globale Testszenarien/Testfälle***