

Local Sharing Software Requirements Specification

Version 1.0

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Revision History

Date	Version	Description	Author
12/Oct/14	1.0	Erste, vorläufige Version	Corinna Hertweck, Johannes Grün, Julia Anken
19/Oct/14		Aktuellstes UseCase-Diagramm eingebunden Hinzufügen der UseCase Verlinkung	Johannes Grün

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	5
1.5	Overview	5
2.	Overall Description	5
3.	Specific Requirements	7
3.1	Functionality	7
3.1.1	Registrieren	7
3.1.2	Login	7
3.1.3	Logout	8
3.1.4	Angebote einstellen	8
3.1.5	Angebote verwalten	8
3.1.6	Angebote suchen	8
3.1.7	Angebote löschen	8
3.1.8	Userverwalten	8
3.1.9	Profilinformationen bearbeiten	8
3.1.10	User bewerten	8
3.1.11	Anfragen an User senden	8
3.1.12	Anfragen beantworten	9
3.2	Usability	9
3.2.1	Bedienung	9
3.2.2	Userinterface Aufbau	9
3.3	Reliability	9
3.3.1	Erreichbarkeit	9
3.3.2	Durchschnittliche Reparaturzeit (MTTR)	9
3.4	Performance	9
3.4.1	Antwortzeit	9
3.4.2	Kapazität	9
3.4.3	Ressourcen Verbrauch	9
3.5	Supportability	10
3.6	Design Constraints	10
3.6.1	Spring MVC	10
3.6.2	Hibernate	10
3.6.3	Git	10
3.7	On-line User Documentation and Help System Requirements	10
3.8	Purchased Components	10
3.9	Interfaces	10
3.9.1	User Interfaces	10
3.9.2	Hardware Interfaces	10
3.9.3	Communications Interfaces	10
3.10	Licensing Requirements	10
3.11	Applicable Standards	10
4.	Supporting Information	10

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Software Requirements Specification

1. Introduction

Im Folgenden soll der Zweck des Dokuments sowie dessen Aufbau erläutert werden. Dazu wird kurz auf das Projekt an sich und seine Use-Cases eingegangen. Zusätzlich enthält die Einleitung Erklärungen der Abkürzungen, Begriffe und Akronyme, die in diesem Dokument genannt werden. Referenzen auf andere Dokumente werden in einem eigenen Verzeichnis unter 1.4 zusammengefasst.

1.1 Purpose

In den Software Requirement Specifications werden die Anforderungen, die an das Projekt gestellt werden, genauer erläutert. Das dient als Grundlage der Programmierung für den Entwickler, da hierin alle Funktionalitäten und Anforderungen, die an das Projekt gestellt werden, erläutert sind.

Außerdem kann anhand der SRS bei Projektabschluss geprüft werden, ob die gewünschten Kriterien wie zu Beginn geplant umgesetzt wurden. Das wiederum kann als Messgröße für den Projekterfolg herangezogen werden.

Dazu wird zum einen eine allgemeine Beschreibung der Einflussfaktoren auf das Produkt geliefert sowie die genauen Anforderungen. Dazu zählen die einzelnen Funktionen, Anforderungen an das Design, die Performance oder auch die Schnittstellen.

1.2 Scope

Die Webanwendung „LocalSharing“ ermöglicht den Austausch oder das Ausleihen von Gegenständen oder Hilfeleistungen unter Usern. Dabei wird auf den Einsatz jedweder Währung, also sowohl real als auch fiktiv, verzichtet. Durch das Bewerten von Usern kann eine Vertrauensbasis vor dem Handel aufgebaut werden.

Das Projekt ist in die Subsysteme „Profilverwaltung“, „Angebotsverwaltung“ und „Anfrageverwaltung“ aufgeteilt. Zur Gruppe „Profilverwaltung“ gehören sowohl die Use-Cases zum Anlegen, Verändern oder Löschen eines Profils als auch diejenigen, welche sich auf das Bewerten der User beziehen. In der Gruppe „Angebotsverwaltung“ stehen alle Use-Cases, bei denen es um die Bearbeitung von Angeboten für Gegenstände oder Hilfeleistungen geht. Use-Cases wie „Anfrage für Angebot senden/annehmen/ablehnen“, also alle, die für die Verwaltung von Anfragen gebraucht werden, sind in der letzten Gruppe zu finden.

1.3 Definitions, Acronyms, and Abbreviations

User, unangemeldeter User, Admin, Tauschanfrage, Angebotsanfrage, Bewertung, Bestätigung, Ablehnen, Bearbeiten von Anfragen

User	Ein „User“ ist ein Benutzer der Webanwendung „LocalSharing“, der hierfür einen Account angelegt hat und angemeldet ist.
unangemeldeter User	Ein „unangemeldeter User“ ist jeder Benutzer, der nicht angemeldet ist. Hierzu zählen also sowohl diejenigen, die noch keinen Account haben als auch solche, die zwar ein Profil haben, aber im Moment nicht angemeldet sind.
Admin	Ein „Admin“ ist ein Verwalter der Webanwendung „LocalSharing“. Er hat die Möglichkeit, User sowie Angebote zu löschen.
Tauschen	„Tauschen“ bezeichnet den vorläufig endgültigen Austausch von Waren.
Ausleihen	„Ausleihen“ bezeichnet einen Prozess, bei dem Güter für eine bestimmte zuvor definierte Zeitspanne an einen anderen User übergeben werden.
Angebot	User haben die Möglichkeit mehrere „Angebote“ aufzugeben. Das bedeutet, dass sie einen Gegenstand oder eine Hilfeleistung anbieten.

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Anfrage	Ein User kann eine „Anfrage“ an einen anderen User stellen, um ihn zu bitten, einen Gegenstand zu tauschen oder ihm auszuleihen. Eine Anfrage kann auch für die Erbringung einer Hilfeleistung gestellt werden.
Bestätigung	Anfragen können durch eine „Bestätigung“ angenommen werden.
Ablehnen	Anfragen müssen nicht angenommen werden. User können diese auch ablehnen.
Hilfeleistung	Unter „Hilfeleistung“ wird eine Dienstleistung verstanden, die ein User für einen anderen erbringt wie z.B. Rasenmähen oder bei Computerproblemen behilflich sein.
Bewertung	Eine „Bewertung“ kann von Usern an andere User vergeben werden, sofern diese über „LocalSharing“ Kontakt aufgenommen haben und Güter oder Hilfeleistungen ausgetauscht oder verliehen haben.
Bearbeiten von Anfragen	„Bearbeiten von Anfragen“ meint, dass bereits gesendete Anfragen, solange vom anderen User noch nicht angenommen oder abgelehnt, weiterhin bearbeitet werden können. Bei jeder Änderung erhält der betreffende User eine Benachrichtigung und kann dann nur noch auf das aktuellste Angebot antworten.

1.4 References

1.5 Overview

In den folgenden Punkten der Software Requirements Specifications sollen die zukünftigen Anforderungen an das Produkt genauer erläutert werden. Dazu werden zunächst einige Einflussfaktoren wie Benutzermerkmale beschrieben. Anhand dieser werden dann im 3. Kapitel die spezifischen Anforderungen an das Produkt abgeleitet und genau definiert. Diese bilden für den Entwickler die Grundlage für seine Arbeit.

2. Overall Description

Produktfunktionen

Ein unangemeldeter User hat die Möglichkeit nach Angeboten zu suchen. Dabei soll er sowohl nach bestimmten Gegenständen oder Hilfeleistungen sowie nach Orten suchen können. Die Ansicht einer Karte hilft ihm dabei sich zu orientieren und die für ihn relevanten Angebote einzuschränken. Das soll dabei helfen neue User durch die Einsicht des Angebots zu gewinnen.

Mit der Registrierung oder Anmeldung bei „LocalSharing“ hat der User dann die Möglichkeit zu anderen Usern Kontakt aufzunehmen. Er kann weiterhin alles tun, was auch ein unangemeldeter User kann, doch bei Ansicht eines Angebots hat er die Möglichkeit dem User, der dieses Angebot aufgegeben hat, eine Anfrage zum Tausch oder Verleih zu stellen. Dabei spezifiziert er weitere Rahmenbedingungen für die Anfrage, wie z.B. Dauer bei Ausleihen.

Der andere User hat verschiedene Möglichkeiten mit dem Angebot umzugehen. Er kann es annehmen, ablehnen oder selbst neue Rahmenbedingungen setzen und somit ein Gegenangebot stellen. Damit können User untereinander verhandeln.

Das Produkt soll vor allem dazu dienen User, die relativ nahe beieinander wohnen in Kontakt zu bringen, sodass diese sich gegenseitig aushelfen können. Viel basiert also auf dem Willen zu helfen. Um anderen zunächst unbekannten Usern aber tatsächlich zu helfen, muss eine gewisse Vertrauensbasis bestehen. Aus diesem Grund sollen die User in einem gewissen Maß abgesichert sein. Die Möglichkeit andere User nach einem Handel zu bewerten ist hierfür die wichtigste Grundlage. Ebenfalls sollte eine Verifizierung der Person über E-Mail-Adresse oder Telefonnummer durch „LocalSharing“ durchgeführt werden, damit die Identitäten der User nicht vollständig anonym bleiben, was es gegebenenfalls einfacher macht, diese für beschädigte oder nicht zurückgegebene Ware zur Rechenschaft zu ziehen.

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Benutzermerkmale

Potentielle User der Webseite hatten eventuell davor schon Erfahrungen mit Webseiten wie craigslist.com oder airbnb.com. Generell soll dies für die Nutzung von „LocalSharing“ aber nicht erforderlich sein. Ein gewisses Maß an Vertrautheit zum Internet besteht aber. Das Besuchen von Webseiten und die Erstellung von Profilen für diese ist für potentielle Nutzer von „LocalSharing“ kein Neuland.

Wenn die User die Webseite besuchen ist ihr vordergründiges Ziel nicht diese selbst. Sie ist für sie nur ein Mittel zum Zweck, nämlich den Kontakt zu anderen Usern herzustellen. Dementsprechend muss sie schlicht gehalten werden und sollte einfach zu bedienen sein. Zudem sollte sie performant sein, sodass der User die Nutzung nicht als Belastung oder zeitaufwendig empfindet.

Da „LocalSharing“ davon lebt, dass viele User angemeldet sind, ist es genauso wichtig, potentielle User nicht durch zu viele und zu komplexe Funktionen abzuschrecken, die eine gewisse Einarbeitungszeit voraussetzen würden. Stattdessen soll die Webanwendung von jedem möglichst intuitiv bedienbar sein.

Das Design soll modern gehalten werden. Dazu gehört in erster Linie ein ansprechendes Design, das alle Inhalte der Seite übersichtlich und großzügig abbildet. Das soll es neuen Usern ermöglichen, sich schnell ohne weitere Hilfestellungen auf der Webseite zurechtzufinden und damit dazu beitragen, dass neue Mitglieder gewonnen werden.

Der Dienst soll den Usern rund um die Uhr zur Verfügung stehen, allerdings hat dies nicht die höchste Priorität.

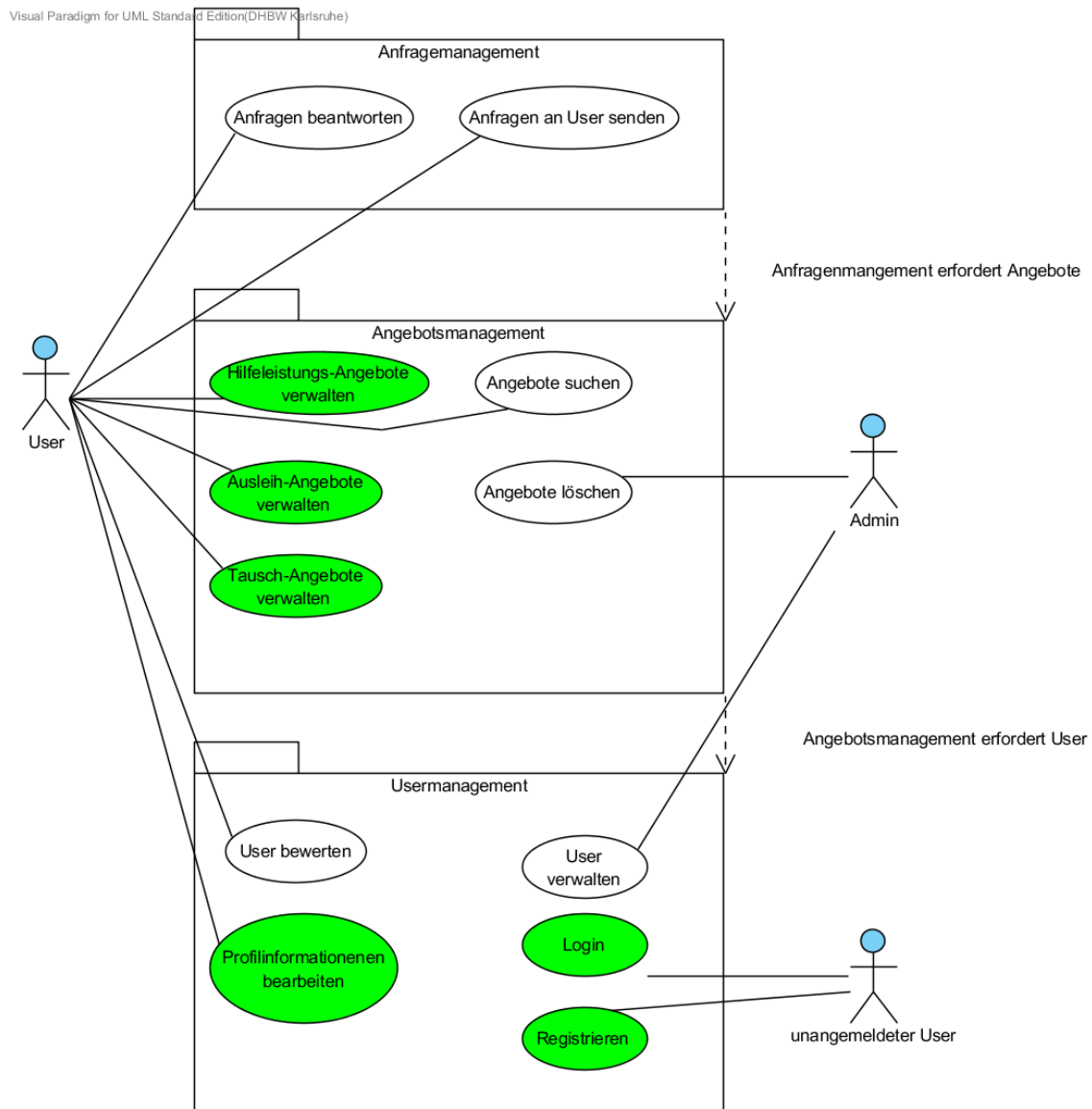
Einschränkungen

Den Entwicklern steht nur eine bestimmte Zeit zur Verfügung. Bis zum Ende des Sommersemesters 2015 soll das Projekt abgeschlossen sein, sodass es zur Beurteilung für das Fach „Softwareengineering“ an der DHBW Karlsruhe vorgelegt werden kann. Die sechs Use-Cases, „Login“, „Profilinformationen verwalten“, „User bewerten“, „Angebote suchen“, „Angebote einstellen“, „Angebote verwalten, sollen bereits bis zum Ende des Wintersemesters 2014/15 implementiert und funktionsfähig sein.

Annahmen und Abhängigkeiten

Als Java-Framework für die Erstellung der Webanwendung „LocalSharing“ wird Spring MVC verwendet. Zur Absicherung dient Spring Security. Als RDBMS wird H2 verwendet. Für den Zugriff auf die Daten soll Hibernate genutzt werden. yEd und Visual Paradigm dienen der Modellierung der Use-Cases und UML-Diagrammen.

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	



3. Specific Requirements

3.1 Functionality

3.1.1 Registrieren

Ein unangemeldeter User soll sich mit Hilfe der Registrierungsfunktion an der Webanwendung registrieren können. Für die Registrierung wird der Vorname, Nachname, Geburtsdatum und eine Email-Adresse benötigt. Für die Anmeldung an der Webanwendung muss er des Weiteren ein Passwort setzen. Ist die Registrierung erfolgreich abgeschlossen, ist der User nun dem System bekannt und er kann sich fortan an am System anmelden (Login).

3.1.2 Login

Ein unangemeldeter User, der am System registriert ist, kann sich mit seiner bei der Registrierung angegebenen Email-Adresse und dem gewählten Passwort an der Webanwendung anmelden. Er ist nun ein User.

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

3.1.3 Logout

Ein User hat die Möglichkeit sich von der Webanwendung abzumelden, sein Status wechselt dann von User zu unangemeldeter User.

3.1.4 Angebote einstellen

Der User hat die Möglichkeit Angebote in die Webanwendung einzustellen. Dazu muss er eine Kategorie angeben zu welcher das Angebot zählt. Des Weiteren muss er angeben, ob er das Angebot tauschen oder ausleihen möchte. Zusätzlich kann ein Bild eingestellt werden und eine nähere Beschreibung hinzugefügt werden.

3.1.5 Angebote verwalten

Der User soll die eigenen eingestellten Angebote verwalten können. Das bedeutet er kann diese bearbeiten und löschen.

3.1.6 Angebote suchen

Ein User soll nach Angebote anderer User suchen können. Dazu wird die beim Einstellen gewählte Kategorie verwendet. Der User soll durch eine Gruppierungsstruktur bzw. hierarchische Anordnung von Kategorien einfach das gewünschte Angebot finden.

3.1.7 Angebote löschen

Ein Admin hat die Möglichkeit einzelne Angebote von Usern zu löschen bzw. Zu bearbeiten (z.B. die Kategorie ändern).

3.1.8 Userverwalten

Ein Admin kann User verwalten, das bedeutet er kann User aus der Webanwendung löschen. Dadurch ändert sich der Status von dem User von „User“ zu „nicht registrierter User“.

3.1.9 Profilinformationen bearbeiten

Ein User kann seine Profilinformationen bearbeiten. Dadurch ist es z.B. möglich seine Email-Adresse zu ändern.

3.1.10 User bewerten

Alle User haben die Möglichkeit andere User zu bewerten. Die Bewertung erscheint dann im Profil des bewerteten Users.

3.1.11 Anfragen an User senden

Ein User kann Anfragen an einen anderen User senden. Dazu wählt er zunächst ein Angebot des entsprechenden Users aus und sendet dann die Anfrage.

Die Anfrage enthält bei einer Verleihung folgende Pflicht-Elemente:

- Zeitraum
- Userinformationen (Link zum Profil)

Die Anfrage enthält bei einem Tausch folgende Pflicht-Elemente:

- Usern Informationen (Link zum Profil)
- Tausch-Angebot

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Zusätzlich enthalten ist:

- Freitext Möglichkeit für Detailfragen zum Angebot.

3.1.12 Anfragen beantworten

Der User soll die an ihn gestellten Anfragen (Tausch/Verleih) angezeigt bekommen und diese beantworten können. Hierbei kann er einer Anfrage zustimmen, ablehnen oder einen Gegenvorschlag machen. Beim Zustimmung muss der User einen Ort und eine Zeit angeben an dem das Angebot übergeben bzw. ausgetauscht werden soll. Dies erscheint bei dem empfangenden User wiederum als Anfrage, die er dann beantworten kann.

3.2 Usability

3.2.1 Bedienung

Die Bedienung der Webanwendung soll intuitiv und einfach sein. Dadurch ist es für neue User möglich innerhalb kürzester Zeit die Verwendung zu erlernen. Deshalb soll drauf geachtet werden, dass die Webanwendung den modernen Seh- und Suchgewohnheiten (look & feel) entspricht.

3.2.2 Userinterface Aufbau

Beim Design des Userinterfaces soll im Besonderen auf das Verwenden von "sprechenden" Bildern bzw. Icons geachtet werden, diese sollen dem Einsatz von Texten vorgezogen werden. Ist eine Gruppierung bzw. hierarchische Anordnung von Elementen möglich, so ist diese zu wählen, da große Übersichtseiten sehr schnell unübersichtlich werden. So ist es dem User möglich das gewünschte Element schneller zu finden, sowie Elemente, die mit seinem gesuchten in Zusammenhang stehen.

3.3 Reliability

3.3.1 Erreichbarkeit

Die Webanwendung soll wenn möglich 24/7 erreichbar sein. Ausfälle Zwecks Wartungsarbeiten stellen für den Nutzer sofern sie vorher angekündigt wurden keinen Schaden da. Der Anteil von Wartungsarbeiten ist sehr gering, diese treten bei dieser Webanwendung nur bei Umzug auf Leistungsstärkere Hardware also bei Skalierung der Webanwendung auf.

3.3.2 Durchschnittliche Reparaturzeit (MTTR)

Die durchschnittliche Zeit für Reparaturen am System nach einem Fehler soll ca. 5 Stunden betragen.

3.4 Performance

3.4.1 Antwortzeit

Die Antwortzeit der Webanwendung soll maximal 200ms betragen. Der Durchschnitt soll bei 150ms liegen.

3.4.2 Kapazität

Die Verwendung der Webanwendung soll mit bis zu 25 Usern gleichzeitig möglich sein.

3.4.3 Ressourcen Verbrauch

Der Verbrauch von Festplattenspeicher ist als unkritisch einzustufen, die Webanwendung darf daher bis zu 25GB

<Project Name>	Version: 1.0
Software Requirements Specification	Date: 12/Oct/14
<document identifier>	

Speicherplatz verbrauchen. Der Arbeitsspeicher ist hierbei der Flaschenhals, für einen kostengünstigen Betrieb ist es notwendig, dass die Anwendung mit ca. 1GB RAM zurechtkommt.

3.5 Supportability

—to be determined—

3.6 Design Constraints

3.6.1 *Spring MVC*

Zur Realisierung der Webanwendung soll Spring MVC und Spring Security verwendet werden. Dadurch ergibt sich, dass die Server-Logik mit der Programmiersprache Java zu realisieren ist. Client-Seitig wird CSS und JavaScript eingesetzt.

3.6.2 *Hibernate*

Für die Verbindung mit der Persistenten Schicht soll Hibernate verwendet werden.

3.6.3 *Git*

Für die Versionsverwaltung und das Entwickeln im Team wird Git als CVS verwendet.

3.7 On-line User Documentation and Help System Requirements

Dem User soll nach dem Registrieren eine kurze Einführung in die Handhabung gegeben werden. Aber durch die Anforderung der intuitiven und einfachen Bedienung sollte weiterreichende Hilfe nicht notwendig sein.

3.8 Purchased Components

Die Webanwendung verwendet keine eingekauften Software Komponenten.

3.9 Interfaces

3.9.1 *User Interfaces*

—to be determined—

3.9.2 *Hardware Interfaces*

—to be determined—

3.9.3 *Communications Interfaces*

—to be determined—

3.10 Licensing Requirements

—to be determined—

3.11 Applicable Standards

—to be determined—

4. Supporting Information

—to be determined—