

<DM-ID-App>

Anforderungs- und Designspezifikation, 1.0.1

Status: in progress Datum: 8. Oktober 2020

**Dokumentenverwaltung**

### Dokument-Historie

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Status** | **Datum** | **Änderungsgrund** |
| 0.0.1 | in progress | 30.09.2020 | Erstellung des Dokuments |
| 0.0.2 | in progress | 02.10.2020 | Klassendiagramm, Aktivitätsdiagramm, Use-Case Diagramm |
| 0.0.3 | in progress | 08.10.2020 | Dokumentation, Abbildungsverzeichnis, Literaturverzeichnis |
| 1.0.1 | in progress | 26.11.2020 | Ergänzung der Sins |

### Dokument wurde mit folgenden Tools erstellt:

### Textverarbeitungsprogramm: Word

* **Erstellung der Diagramme** (Use-Case Diagramm, Klassendiagramme, Aktivitätsdiagramm): app.diagrams.net

### Version Control System (VCS): Git

### Statuslegende:

* not started
* in progress
* waiting
* deferred
* completed

**Inhaltsverzeichnis**

1. [Einleitung 4](#_TOC_250021)
   1. [Zweck des Dokuments 4](#_TOC_250020)
   2. [Gültigkeit des Dokuments 4](#_TOC_250019)
   3. [Begriffsbestimmungen und Abkürzungen 4](#_TOC_250018)
2. [Allgemeine Beschreibung des gewünschten Systems 5](#_TOC_250017)
   1. [Zweck des gewünschten Systems 5](#_TOC_250016)
   2. [Überblick über die geforderte Funktionalität 5](#_TOC_250015)
   3. [Allgemeine Einschränkungen 6](#_TOC_250014)
   4. [Vorgaben zu Hardware und Software 6](#_TOC_250013)
   5. [Anforderungsquellen / Zielgruppen 7](#_TOC_250012)
3. [Detaillierte Beschreibung der Anforderungen (Leistungsmerkmale) 8](#_TOC_250011)
   1. [Lieferumfang 8](#_TOC_250010)
   2. [Geforderte Funktionen des Produkts 8](#_TOC_250009)
   3. [Struktur und Verhalten des System (Klassendiagram) 9](#_TOC_250008)
   4. [Qualitätsanforderungen / sonstige entwicklerorientierte Anforderungen 11](#_TOC_250007)
4. [Vorgaben des Auftraggebers an die Projektabwicklung 12](#_TOC_250006)
   1. [Anforderungen an die Realisierung 12](#_TOC_250005)
   2. [Abnahmebedingungen 12](#_TOC_250004)
   3. [Lieferbedingungen 12](#_TOC_250003)
5. [Verpflichtungen des Auftraggebers 13](#_TOC_250002)
6. [Literaturverweise 14](#_TOC_250001)
7. [Abbildungsverzeichnis 14](#_TOC_250000)
8. Anhang 14

# Einleitung

## Zweck des Dokuments

Dieses Dokument dient zur Beschreibung einer Client/Server Applikation zum Dokumentenmanagement. Es beinhaltet die Funktionsweise, den Aufbau, die funktionalen und nicht-funktionale Anforderungen der Software, welche durch Screenshots verbildlicht werden. Dieses Dokument beabsichtigt die beschriebene Software für Interessenten der Client/Server Architektur.

## Gültigkeit des Dokuments

Die Vorlage für dieses Dokument liefert die Hochschule Furtwangen, somit basiert das Dokument auf IEEE Software Requirements Specification Template (SRS). Das Dokument beschreibt den Aufbau und die Funktionalität eines Basis-Dokumentenmanagementsystems.

## Begriffsbestimmungen und Abkürzungen

Im Laufe der Entwicklung wird dieser Abschnitt ergänzt.

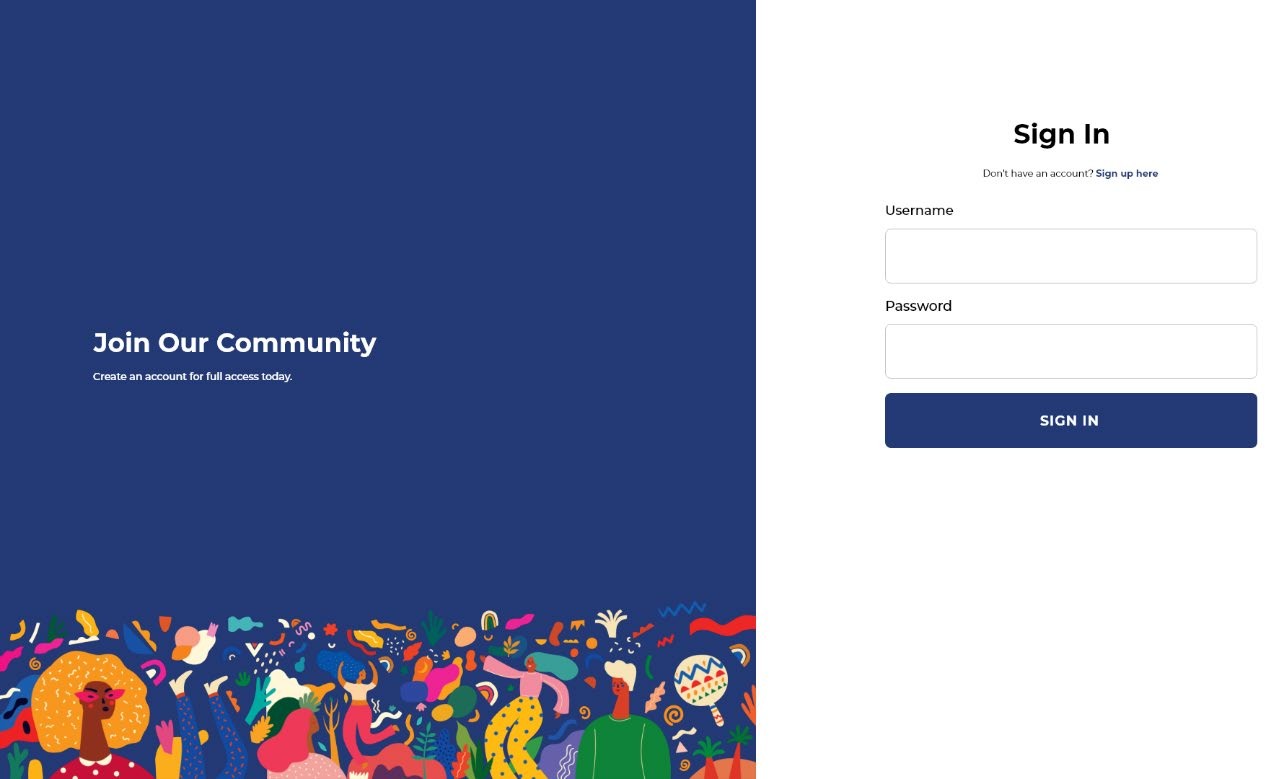
|  |  |
| --- | --- |
| **Abkürzungen** | **Bedeutung** |
| DG | Daniel Gasser |
| IK | Irem Koldere |
| SRS | Software Requirements Specification Template |
| VCS | Version Control System |
| HTTP | Hypertext Transfer Protocol |

# Allgemeine Beschreibung des gewünschten Systems

## Zweck des gewünschten Systems

DM-ID-App ist eine Client/Server Applikation zum Dokumentenmanagement. Diese Applikation soll Benutzern ermöglichen Dokumente hoch- und herunterzuladen, welche nur einen Datentyp unterstützt (TXT) und das Auflisten, sowie Löschen dieser auch ermöglichen wird.

## Überblick über die geforderte Funktionalität



*Abbildung 1: Login Maske*

Ein Benutzer soll am Client Dokument hochladen, herunterladen, auflisten und löschen können. Die Anmeldung am Client erfolgt durch die Eingabe eines Benutzernamen und Kennwort. Die Rechteverwaltung wird nicht implementiert, daher werden beispielhafte Benutzer erstellt und mit ihnen die App simuliert. Die Kommunikation zwischen dem Server und Client erfolgt über HTTP. Das heißt die Rolle des Clients wird der Webbrowser einnehmen, der Dokumente fordern und präsentieren wird, und der Server als Webserver agieren. Das gemeinsame Protokoll zwischen Webserver und Browser ist dann eben HTTP (Hypertext Transfer Protocol).[1](#_bookmark1)

Damit der Server weiß, welche Aktionen er durchführen muss, verwendet HTTP Methoden, um dem Server mitzuteilen, wenn der Client eine Anfrage schickt. Diese werden auch HTTP- Verben genannt, welche Operationen ausführen und bei jedem Abschluss Erfolg oder Misserfolg zurückmelden. Hierbei werden die meist verbreiteten verwenden:

* + - GET
    - POST
    - PUT
    - DELETE[2](#_bookmark2)

1 Vgl. Schiedermeier 2013: S.446

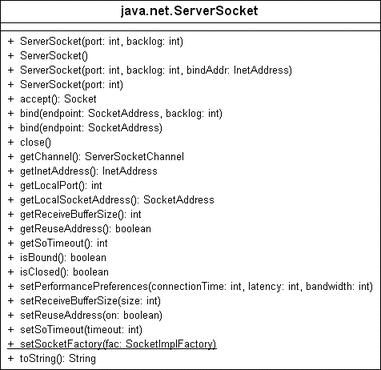
2 Vgl. Hackernoon (2020) online

Und Außerdem werden der Server und Client auf unterschiedlichen Hosts laufen.

*Funktionen:*

* + - *Login*
    - *Upload*
    - *Download*
    - *List*

*Die Methoden, die für die Client/Server Kommunikation in Java nötig sind:*



*Abbildung 2: Methoden in JAVA* [3](#_bookmark3)

## Allgemeine Einschränkungen

Die Client/Server Applikation wird lediglich unter Linux Ubuntu laufen, die Laufzeitumgebung OpenJDK 14 läuft unter der neusten Version, die Rechteverwaltung wird nicht implementiert und der Server und Client werden auf unterschiedlichen Hosts laufen.

## Vorgaben zu Hardware und Software

### Hardware:

Die Client/Server Applikation wird über die Software GIT zur gemeinsamen Versionskontrolle unter MacBook Pro und Lenovo Thinkpad erstellt.

### Software:

Die Dokumentenmanagement Applikation wird unter Linux Ubuntu 20.04 (LTS 64-Bit) laufen, die Implementierung findet im Programmierwerkzeug Eclipse/Intellij in der Sprache Java mit der Laufzeitumgebung OpenJDK 14 statt.

Für die Erstellung der Grafiken, ER- und Use Case Diagramme wird das Tool app.diagrams.net verwendet.

Für die Kommunikation wird die Bibliothek com.sun verwendet.

3 Vgl. Ullenbloom (2020) online

## Anforderungsquellen / Zielgruppen

* + - Typische IT-Studenten, die Interesse an der Entwicklung an Client/Server Modellen haben.
    - Auftraggeber des Projektes: Dipl. -Ing. Manuel Koschuch, Bakk. techn.
    - Personen, die ein Dokumentenmanagementsystem benötigen/verwenden möchten

# Detaillierte Beschreibung der Anforderungen (Leistungsmerkmale)

## Lieferumfang

Laufend werden Entwicklungen in Git geupdated.

## Geforderte Funktionen des Produkts

Das Produkt wird aus funktionaler Sicht anhand von Anwendungsfällen (Use Cases) beschrieben. Die Anwendungsfälle werden anhand eines Diagrammes dargestellt. Dabei werden die verschiedenen Anwendungsfälle mittels <<include>> (Enthältbeziehung) mit einem anderen Anwendungsfall eingebunden:

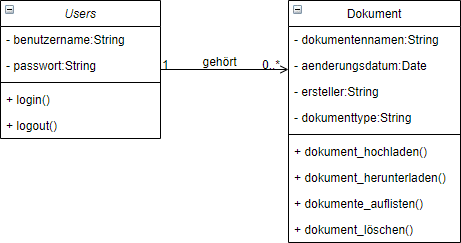
### Use-Case Diagramm

*Abbildung 3: Use-Case Diagramm*

|  |  |
| --- | --- |
| **Bezeichnung** | *Dokumentenmanagementsystem* |
| **Zusammenfassung** | *Bei erfolgreichem Login kann der Users Dokumente hochladen, herunterladen, auflisten und löschen* |
| **Akteure** | *User, Server, Dokument* |
| **Vorbedingung** | Erfolgreiche Anmeldung |
| **Ablaufbeschreibung** | User meldet sich mittels Username und Password an, dabei werden seine/ihre Dokumente aufgelistet. |
| **Verwendungen**  **(Include- Beziehungen)** | Dokument hochladen, Dokument herunterladen, Dokument löschen |
| **Nachbedingung** | Erfolgreiches Herunterladen von Dokumenten, erfolgreiches Hochladen von Dokumenten, erfolgreiches Löschen von Dokumenten |
| **Fehlschlag** | Ausgabe einer Fehlermeldung |

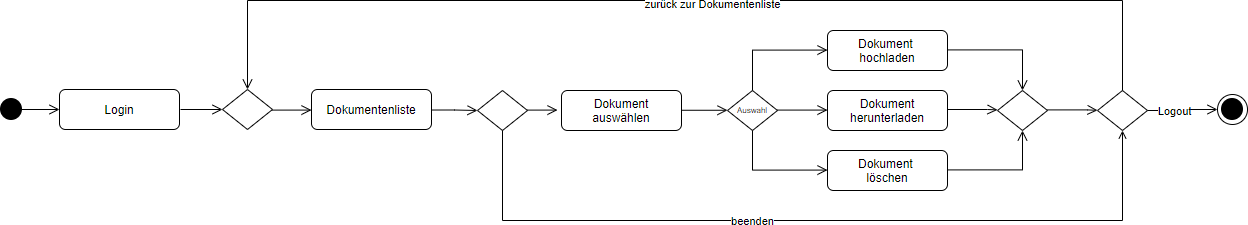
## Struktur und Verhalten des System (Klassendiagram)

### Klassendiagramm



*Abbildung 4: Klassendiagramm*

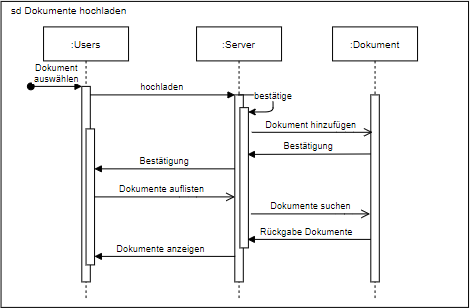
### Aktivitätsdiagramm



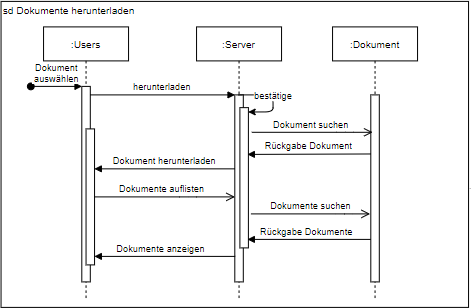
*Abbildung 5: Aktivitätsdiagramm*

### Sequenzdiagramme

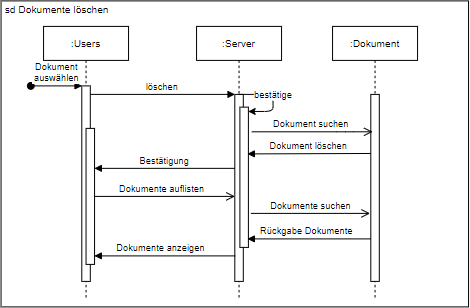
*Abbildung 6: Sequenzdiagramm 1*



*Abbildung 7: Sequenzdiagramm 2*



*Abbildung 8: Sequenzdiagramm 3*



*Abbildung 9: Sequenzdiagramm 4*

## Qualitätsanforderungen / sonstige entwicklerorientierte Anforderungen

DM-ID-App hat keine speziellen Qualitätsanforderungen und somit auch keine Sicherheitsanforderungen.

# Vorgaben des Auftraggebers an die Projektabwicklung

## Anforderungen an die Realisierung

DM-ID-App wird in Java implementiert und erfordert die Installation von Java für Drittbenutzer. Außerdem verlangt die Ausführung das Betriebssystem Linux-Ubuntu, da Befehle in Linux zur Programmausführung implementiert werden.

## Abnahmebedingungen

Das Projekt ist in jeweilige Abgaben (siehe Punkt 4.3) gegliedert. Diese müssen zeitgerecht zum Fälligkeitsdatum abgegeben werden und werden jeweils in der Anforderung des Auftraggebers erstellt. Die Vorlage des vorliegenden Designdokuments wurde vom Auftraggeber zur Verfügung gestellt, die Unit Tests werden anhand von White-Box-Tests durchgeführt, die Implementierung erfolgt wie in Punkt 2.2 in Eclipse und Analyse finden im Rahmen der Lehrveranstaltung statt.

## Lieferbedingungen

|  |  |
| --- | --- |
| **Abgaben** | **Fälligkeitsdatum** |
| Designdokument | 08.10.2020 |
| Unit Test | 05.11.2020 |
| Implementierung | 26.11.2020 |
| Analyse + POC | 21.12.2020 |

# Verpflichtungen des Auftraggebers

Im Rahmen der Lehrveranstaltung Sichere Softwareentwicklung Übung wird dieses Projekt durchgeführt. Laufend werden Unterlagen bereitgestellt und bei Anfragen kann der Auftraggeber unter der Mail-Adresse [manuel.koschuch@fh-campuswien.ac.at](mailto:manuel.koschuch@fh-campuswien.ac.at) kontaktiert werden.

# Literaturverweise

**Hackernoon** (2020): HTTP Made Easy: Understanding the Web Client-Server Communication, bezogen unter [https://hackernoon.com/http-made-easy-understanding-the-](https://hackernoon.com/http-made-easy-understanding-the-web-client-server-communication-yz783vg3) [web-client-server-communication-yz783vg3](https://hackernoon.com/http-made-easy-understanding-the-web-client-server-communication-yz783vg3)), Zugriff am 05.10. 2020

**Schiedermeier**, R. (2013): Programmieren mit Java II, München: Pearson

**Ullenbloom**, C. (2020): Java ist auch eine Insel, bezogen unter [http://openbook.rheinwerk-](http://openbook.rheinwerk-verlag.de/javainsel9/javainsel_21_007.htm#mj1ba27dc5fdf53f527163767f188e1d2e) [verlag.de/javainsel9/javainsel\_21\_007.htm#mj1ba27dc5fdf53f527163767f188e1d2e](http://openbook.rheinwerk-verlag.de/javainsel9/javainsel_21_007.htm#mj1ba27dc5fdf53f527163767f188e1d2e), Zugriff am 08.10.2020

# Abbildungsverzeichnis

[Abbildung 1: Login Maske 5](#_bookmark0)

[Abbildung 2: Methoden in JAVA 6](file://Users/kolderir/Downloads/SoftwareSpezifikation__3_.docx#_Toc53087154)

[Abbildung 3: Use Case Diagramm 8](#_bookmark4)

[Abbildung 4: Klassendiagramm 9](#_bookmark5)

[Abbildung 5: Aktivitätsdiagramm 9](file://Users/kolderir/Downloads/SoftwareSpezifikation__3_.docx#_Toc53087157)

[Abbildung 6: Sequenzdiagramm 1 10](#_bookmark6)

[Abbildung 7: Sequenzdiagramm 2 10](file://Users/kolderir/Downloads/SoftwareSpezifikation__3_.docx#_Toc53087159)

[Abbildung 8: Sequenzdiagramm 3 11](#_bookmark7)

[Abbildung 9: Sequenzdiagramm 4 11](#_bookmark8)

# Anhang

Dieser Bereich wird während der Entwicklung ergänzt.