**SWDE – Software-Entwicklung**

Applikationsdokumentation

Gruppe [02]:

1. [Soto Irene]
2. [Zbinden Christian]
3. [Lys Robin]
4. [Vadakumpadan Godwin]

Gruppensprecher: [Zbinden Christian]

Version: x.x

Datum: xx.xx.xxxx

**Inhaltsverzeichnis**

[1 Einführung 3](#_Toc101052318)

[2 Anforderungen 4](#_Toc101052319)

[3 Architektur 5](#_Toc101052320)

[3.1 Module / Komponenten 5](#_Toc101052321)

[3.2 Schichten und Module 5](#_Toc101052322)

[4 Datenstrukturen 6](#_Toc101052323)

[5 User Interface 7](#_Toc101052324)

[6 Interfaces 8](#_Toc101052325)

[7 Testen 9](#_Toc101052326)

[8 Verteilung (Deployment) 10](#_Toc101052327)

[9 Konfigurationsangaben 11](#_Toc101052328)

[10 Gemachte Erfahrungen (lessons learned) 12](#_Toc101052329)

[11 Tutorials 13](#_Toc101052330)

# Einführung

Kurze Beschreibung, um was es sich in diesem Dokument handelt.

# Anforderungen

Beschreibung von Anforderungen in Form von User Stories. Sollten die User Stories in einem anderen Dokument (z.B. Excel-Tabelle etc.) erfasst sein, kann hier auf dieses Dokument verwiesen werden (das zusätzliche Dokument oder Tabelle muss auch abgegeben werden).

# Architektur

## Module / Komponenten

Hier sollen die wesentlichen Module / Komponenten mit deren Zuständigkeiten und Beziehungen zu anderen Modulen / Komponenten mit Hilfe eines passenden Diagramms angegeben werden[[1]](#footnote-1). Daraus sollen sowohl die von einem Modul offerierten Schnittstellen (*provided interfaces*) als auch die von einem Modul benötigten Schnittstellen (*required interfaces*) ersichtlich sein.

## Schichten und Module

Hier sollen die Zusammenhänge zwischen Schichten und Modulen / Komponenten passend dargestellt werden. Daraus soll die Zugehörigkeit von Modulen / Komponenten zu den entsprechenden Schichten der Applikation ersichtlich sein.

# Datenstrukturen

Hier sollen die Klassen (als Klassendiagramm) aufgeführt werden, die als Entities[[2]](#footnote-2) verwendet werden (und nur diese Klassen). Aus dem Diagramm sollen die Beziehungen unter den einzelnen Entity-Klassen ersichtlich werden. Um das Diagramm übersichtlich zu halten kann auf die Abbildung von Konstruktoren und set- und get-Methoden verzichtet werden: Die Angabe von Attributen (Instanzvariablen) reicht aus.

# User Interface

In diesem Kapitel soll das User Interface in einer passenden Form abgebildet werden. Daraus sollen die unterschiedliche UIs und einzelnen Sichten (Views) sowie die Navigation zwischen einzelnen Sichten ersichtlich sein. Die Einzelnen Sichten (Views) können mit Hilfe von Tools abgebildet werden, jedoch sind auch einfache Skizzen denkbar, die mit einem Bleistift von Hand erstellt wurden. Wichtig ist einzig, dass die abgebildeten Sachverhalte klar und nachvollziehbar sind.

# Interfaces

In diesem Kapitel werden die Business-Interfaces aufgeführt und kurz beschreiben. Daraus soll ersichtlich sein, wozu das Interface benötigt wird und welche Funktionalitäten das Interface vorgibt. Die Abbildung von einzelnen Interfaces soll in UML Form erfolgen (kein Java-Code verlangt).

# Testen

In diesem Kapitel soll das Testkonzept beschrieben und auf die im Code erstellten Tests hingewiesen werden. Daraus soll ersichtlich werden, wie und was getestet wurde (Unit Tests, Integrationstests).

# Verteilung (Deployment)

In diesem Kapitel soll die Verteilung von einzelnen Komponenten abgebildet werden. Daraus soll ersichtlich sein, welche Komponente bzw. welches Modul auf welchem System im produktiven Einsatz ausgeführt wird und wie die Kommunikation (inklusive das Protokoll) zwischen einzelnen Komponenten / Modulen erfolgt[[3]](#footnote-3).

# Konfigurationsangaben

In diesem Kapitel müssen alle nötigen Angaben enthalten sein, die ein Administrator für die Installation und Inbetriebnahme der Applikation benötigt. Insbesondere ist es wichtig, dass die Initialdaten nicht fehlen: Benutzername und Kennwörter, die am Anfang nötig sind, Angaben zu DBMS und Datenbank, Angaben von IP-Adressen und Ports, die in den einzelnen **config**-Dateien evtl. angepasst werden müssen usw. Gehen Sie davon aus, dass der Administrator die Applikation nicht kennt und bei der Installation und Inbetriebnahme der Applikation auf die Angaben von Ihnen angewiesen ist[[4]](#footnote-4).

# Gemachte Erfahrungen (lessons learned)

In diesem Kapitel können die gemachten Erfahrungen, Höhen und Tiefen adressiert werden. Des Weiteren kann man angeben, was gut und was weniger gut gelaufen ist und was man im Nachhinein anders machen würde.

# Tutorials

* [Component Diagram Tutorial](https://online.visual-paradigm.com/diagrams/tutorials/component-diagram-tutorial/)[[5]](#footnote-5)
* [Deployment Diagram Tutorial](https://online.visual-paradigm.com/diagrams/tutorials/deployment-diagram-tutorial/)[[6]](#footnote-6)

1. Hier könnte beispielsweise ein Komponentendiagramm sinnvoll verwendet werden. Eine einfache Erläuterung kann an der folgenden Adresse gefunden werden: <https://online.visual-paradigm.com/diagrams/tutorials/component-diagram-tutorial/> [↑](#footnote-ref-1)
2. Klassen, deren Instanzen in der Datenbank verwaltet werden. [↑](#footnote-ref-2)
3. Hier könnte beispielsweise ein Verteilungsdiagramm sinnvoll verwendet werden. Eine einfache Erläuterung kann an der folgenden Adresse gefunden werden: <https://online.visual-paradigm.com/diagrams/tutorials/deployment-diagram-tutorial/> [↑](#footnote-ref-3)
4. Oder noch besser: Versuchen Sie die generierten Artefakte selbst auf einem neuen System zu installieren und dann werden Sie schnell sehen, welche Informationen / Angaben noch fehlen und was noch nicht klar angegeben ist. [↑](#footnote-ref-4)
5. <https://online.visual-paradigm.com/diagrams/tutorials/component-diagram-tutorial/> [↑](#footnote-ref-5)
6. <https://online.visual-paradigm.com/diagrams/tutorials/deployment-diagram-tutorial/> [↑](#footnote-ref-6)