

Entwurfsdokumentation

Projektname

-

Softwareprojekt SoSe17

Gruppe X

Logo

Vorname1	Nachname1
Vorname2	Nachname2
Vorname3	Nachname3
Vorname4	Nachname4
Vorname5	Nachname5
Vorname6	Nachname6
Vorname7	Nachname7
Vorname8	Nachname8



4. September 2018

Tipps und Hilfen

Information: Dieses Kapitel und alle folgenden grauen Boxen dienen als Hilfestellungen und sollen im fertigen Dokument nicht enthalten sein.

Zur Versionsverwaltung während des Softwareprojekts muss *Git* genutzt werden. *Git* führt Textdokumente mit unterschiedlichen Zeilenbearbeitungen automatisch zusammen. Wir empfehlen den Einsatz von \LaTeX für alle Textdokumente. Um das Auto-Merging zu unterstützen, sollte nach jedem Satzende eine neue Zeile im Quelltext begonnen werden. Die *.tex*-Datei dieser PDF verdeutlicht dies. Erkennt *Git*, dass eine gleiche Zeile bearbeitet wurde, wird ein Konflikt auftreten. Dieser kann in der entsprechenden Datei von Hand mittels eines Texteditors behoben werden.

Fußnoten¹ werden für Homepages genutzt. Zitierungen können mittels eines *cite*-Befehls gesetzt, z.B. *citep* [1].

Tipps zur UML-Modellierung können im SE-Wiki² nachgelesen werden. Achtet darauf, dass eure Diagramme stets lesbar (Vektor-Grafiken!) und gut strukturiert sind. Oftmals ist es sinnvoll ein bis zwei Sätze zusätzlich für Diagrammelemente zu formulieren. So können Missverständnisse ausgeschlossen werden, was einen Einfluss auf die Korrektur haben kann. Diagramme für unwichtige Tätigkeiten (z.B. Login / Logout, User erstellen / löschen, Passwort ändern etc.) sind nicht erforderlich.

¹<https://www.se.informatik.uni-kiel.de/en>

²<https://git.informatik.uni-kiel.de/ag-se/teaching-public/wikis/home>

So kann eine TODO-Notiz erzeugt werden



So kann eine Placeholder-Grafik beispielsweise in den Text eingefügt werden.

Abbildung 1: Beschreibung

Inhaltsverzeichnis

1	Einleitung	1
1.1	Dokumentaufbau	1
1.2	Zweckbestimmung	1
1.3	Entwicklungsumgebung	1
2	Komponentendiagramme	2
3	Verteilungsdiagramm	4
4	Klassendiagramme	5
5	Sequenzdiagramme	6
6	Glossar	7

Kapitel 1

Einleitung

1.1 Dokumentaufbau

Inhalt und Struktur des vorliegenden Dokuments skizzieren (Fließtext).

1.2 Zweckbestimmung

Zweck des ganzen Systems beschreiben (Fließtext).

1.3 Entwicklungsumgebung

Oftmals treten neue Entwickler einem Projekt bei oder ein Entwicklungs-Rechner muss ersetzt werden. Daher sollen hier nennenswerte und grundlegende Frameworks, Bibliotheken, Tools und Sprachen notiert werden. Tabelle X stellt eine beispielhafte Umsetzung dar. Eine Unterteilung in Komponenten ist sinnvoll.

Software	Version	URL
Java Development Kit	8u144	http://www.oracle.com/technetwork/java/javase/downloads/index.html
Software X	Version X	URL X
Software X	Version X	URL X
Software X	Version X	URL X
Software X	Version X	URL X
Software X	Version X	URL X
Software X	Version X	URL X

Tabelle 1.1: Entwicklungsumgebung

Kapitel 2

Komponentendiagramme



Abbildung 2.1: Komponentendiagramm - A

Die Serverkomponente bzw. das Backend beschreibt die internen Vorgänge der Software. Diese Komponente ist über drei verschiedene Schnittstellen, die die unterschiedlichen Benutzerrechte repräsentieren erreichbar. In der Komponente Benutzerverwaltung werden die Daten der Benutzer aus der Datenbank ausgelesen und verarbeitet. Dadurch kann die Benutzerverwaltung eine Authentifizierungsschnittstelle für die anderen Subkomponenten bereit stellen. Zudem kann die Rechteverwaltung manipuliert werden, falls über die *Verwalten*-Schnittstelle zugegriffen wird. Die Komponente Dienstverwaltung liest die Daten der Dienste aus der Datenbank aus und verarbeitet diese. Von außen kann auf die Dienstverwaltung durch die Schnittstellen *Bearbeiten* und *Einsehen* zugegriffen werden. Die Schnittstelle *Bearbeiten* ist nur für Administratoren über die Schnittstelle *Verwalten* zugänglich. Es bei jedem Zugriff die Rechte des zugreifenden Benutzers durch die Schnittstelle *Auth* der Benutzerverwaltung überprüft. Die Komponente Kompositionsverwaltung stellt die benötigten Funktionen für das Bearbeiten und Einsehen von Kompositionen zur Verfügung und bekommt die Daten dafür von den Komponenten Datenbank und Dienstverwaltung. Das Bearbeiten von Kompositionen ist nur über die Schnittstellen *Bearbeiten* und *Verwalten* möglich, das Einsehen auch über das *Einsehen*-Interface. Die Komponente Datenbank speichert alle nötigen Daten für die Komponenten Benutzer-, Dienste- und Kompositionsverwaltung und stellt diese bei Bedarf zur Verfügung. Zusätzlich kann man über die *Einsehen*-Schnittstelle auch direkt auf die Authentifizierung zugreifen um ein initiales registrieren und den Login zu ermöglichen.

Die Komponente Web mit der Subkomponente View Verwaltung ist für die Darstellung in der Webapplikation zuständig. Da über die Weboberfläche sowohl verwaltet, bearbeitet und eingesehen werden können soll, ist die Web-Komponente auch mit all diesen Schnittstellen verbunden. Die

Komponente App mit der Subkomponente View dient zur Darstellung der angefragten Daten in der App. In der App können nur Kompositionen eingesehen werden, daher ist diese nur mit der Einsehen-Schnittstelle verbunden.

Kapitel 3

Verteilungsdiagramm



Abbildung 3.1: Verteilungsdiagramm

Das zukünftige Deployment des Systems wird mittels einem Verteilungsdiagramm modelliert. Weiterhin sollten wichtige oder eventuell undeutliche Zusammenhänge (z.B. warum Schnittstelle X genutzt wird) in einem Fließtext beschrieben werden.

Kapitel 4

Klassendiagramme



Abbildung 4.1: Klassendiagramm - A

Klassenname	Aufgabe
Klasse A	Aufgabe A
Klasse B	Aufgabe B
Klasse C	Aufgabe C
Klasse D	Aufgabe D
Klasse E	Aufgabe E
Klasse F	Aufgabe F
Klasse G	Aufgabe G

Tabelle 4.1: Klassenbeschreibung - A

Teilt eure Klassendiagramme bitte auf und baut **kein** einzelnes riesiges Diagramm. Getter und Setter Methoden müssen hier nicht modelliert werden. Sie sollten aber der klassischen Namenskonvention folgen, um die Nutzung in Sequenzdiagrammen zu ermöglichen.

Auf jedes Diagramm folgt eine Tabelle, in der die Aufgabe **jeder** Klasse beschrieben wird.

Kapitel 5

Sequenzdiagramme



Abbildung 5.1: Sequenzdiagramm - A

Das dynamische Verhalten des Systems wird mittels Sequenzdiagrammen modelliert. Hier müssen wahrscheinlich geräteübergreifende Aufrufe modelliert werden. Findet dafür eine geeignete Notation und nutzt diese durchgehend! Achtet weiterhin darauf, dass die anderen Methoden im Klassendiagramm zu finden sind. Manche Sequenzen erfordern sicherlich eine kurze schriftliche Beschreibung.

Kapitel 6

Glossar

In diesem Glossar können Akronyme und abkürzende Schreibweisen aufgelistet werden. Alle verwendeten Abkürzungen innerhalb des Projekts müssen hier erläutert werden.

Abkürzung	Beschreibung
Abk. A	Beschreibung A
Abk. B	Beschreibung B
Abk. C	Beschreibung C
Abk. D	Beschreibung D
Abk. E	Beschreibung E
Abk. F	Beschreibung F
Abk. G	Beschreibung G

Tabelle 6.1: Glossar

Literaturverzeichnis

- [1] Mary Shaw. Writing good software engineering research papers: minitutorial. In *Proceedings of the 25th International Conference on Software Engineering (ICSE 2003)*, pages 726–736, Washington, DC, USA, 2003. IEEE Computer Society.