

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	3
4	Klassendiagramme	4
5	Sequenzdiagramme	7
6	Glossar	8

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 strukturelle Übersicht des zu entwickelnden Systems wird mittels Komponentendiagrammen modelliert. Auf jedes Diagramm muss eine textuelle Beschreibung (Fließtext mit Umbrüchen / Absätzen oder Tabelle) folgen, in der die Aufgaben der Subkomponenten beschrieben werden.

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

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.

Klassenname	Aufgabe
User	<ul style="list-style-type: none"> • Speicherung der Nutzerdaten, sowie einer künstlichen Datenbank-ID
Composition	<ul style="list-style-type: none"> • Speicherung der Kompositionen als Graphen, durch Speicherung der Knoten und Kanten • Speicherung des Urhebers und der Nutzenden mit Zugriffs- bzw. Bearbeitungsrechten • Konvertermethode, zum Erstellen eines reduzierten Objektes
CompositionNode	<ul style="list-style-type: none"> • Speicherung eines Dienstes, für dessen Verwendung im Kompositionsgraphen • Dazu Speicherung einer Position und Ausdehnung des Knotens • Konvertermethode, zum Erstellen eines reduzierten Objektes
CompositionEdge	<ul style="list-style-type: none"> • Speicherung einer gerichteten Kompositionskante als Paar von Kompositionsknoten • Bestimmung der Kompatibilität der, an der Kante beteiligten Knoten • Konvertermethode, zum Erstellen eines reduzierten Objektes
Service	<ul style="list-style-type: none"> • Speicherung der Dienst-Eigenschaften • Speicherung der zum Dienst gelisteten Tags • Speicherung je einer Liste der passenden Ein- bzw. Ausgabeformate
Format	<ul style="list-style-type: none"> • Speicherung von Name, Version und Kompatibilitätsgrad

Klassenname	Aufgabe
SendComp	<ul style="list-style-type: none">• Speicherung der wichtigsten Daten mit möglichst wenig Referenzen zum Verschieken der Objekte als JSON über REST
ReducedEdge	<ul style="list-style-type: none">• Speicherung der wichtigsten Daten mit möglichst wenig Referenzen zum Verschieken der Objekte als JSON über REST
ReducedNode	<ul style="list-style-type: none">• Speicherung der wichtigsten Daten mit möglichst wenig Referenzen zum Verschieken der Objekte als JSON über REST
ListComp	<ul style="list-style-type: none">• Speicherung der elementarsten Daten für die Listenansicht aller Kompositionen

Tabelle 4.2: Klassen des Models *Fortsetzung*

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.