

Projektdokumentation
IT-Projekt

Hochschule der Medien Stuttgart
Fachbereich Information und Kommunikation
Studiengang Wirtschaftsinformatik und digitale Medien
(4. Semester)
Prof. Dr. Peter Thies
Prof. Dr. Christian Rathke
Sommersemester 2015

Vorgelegt von: Florin Geier (27974)
Christian Mähler (27831)
Benjamin Schwab (27949)
Fabian Tschullik (27937)
Manuel Voelker (27950)

Stuttgart, 08.07.2015

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis.....	III
Abkürzungsverzeichnis	IV
1 Anforderungsspezifikation	1
1.1 Rahmenbedingungen	1
1.2 Beschreibung des Zielsystems	2
1.3 Entwicklerwerkzeuge	6
2 Meilensteine	6
2.1 Meilenstein 1.....	6
2.2 Meilenstein 2.....	6
2.2.1 Use-Case-Diagramme	7
2.3 Meilenstein 3.....	12
2.3.1 Klassendiagramme	13
2.3.2 Entity-Relationship-Modell	18
2.3.3 Verteilung der Aufgaben und Vorgehen im Projekt.....	19
2.4 Meilenstein 4.....	20
2.5 Meilenstein 5.....	20
3 Installationsanleitung	20
4 Benutzerhandbuch	23
4.1 Editor	23
4.1.1 Allgemeines.....	23
4.1.2 Bauteil	23
4.1.3 Baugruppe.....	25
4.1.4 Enderzeugnis	26
4.2 Report Generator.....	26
4.2.1 Allgemeines.....	26
4.2.2 Strukturstückliste	27
4.2.3 Materialbedarf	27

5	Fazit.....	28
	Literatur- und Quellenverzeichnis.....	29

Abbildungsverzeichnis

Abbildung 1: Beschreibung des Zielsystems.....	3
Abbildung 2: Viewer-Übersicht.....	7
Abbildung 3: Stückliste verwalten - Textuelle Beschreibung.....	7
Abbildung 3: Bauteil verwalten - Diagramm	8
Abbildung 4: Bauteil verwalten - Textuelle Beschreibung	8
Abbildung 5: Baugruppe verwalten - Diagramm.....	9
Abbildung 6: Baugruppe verwalten - Textuelle Beschreibung.....	9
Abbildung 7: Enderzeugnis verwalten - Diagramm	10
Abbildung 8: Enderzeugnis verwalten - Textuelle Beschreibung	10
Abbildung 11: Report-Generator - Diagramm	11
Abbildung 12: Report Generator - Textuelle Beschreibung 1.....	11
Abbildung 13: Report-Generator - Textuelle Beschreibung 2	12
Abbildung 14: Applikationslogik.....	13
Abbildung 15: It04gwtServiceImpl – DbConnection	14
Abbildung 16: It04gwtServiceImpl - Business Objekte.....	15
Abbildung 17: It04gwtServiceImpl - Synchrones Interface – Report.....	16
Abbildung 18: ERM	18
Abbildung 19: Google Cloud - Projekt erstellen	20
Abbildung 20: Google Cloud - Instanz erstellen	21
Abbildung 21: Google Cloud - Datenbank anlegen	21
Abbildung 22: Google Cloud - Berechtigungen	22
Abbildung 23: GitHub - Neues Repository	22
Abbildung 24: Startbildschirm Editor	23
Abbildung 25: Untermenü Bauteil.....	24
Abbildung 26: Eingabemaske Bauteil.....	24
Abbildung 27: Bauteil erfolgreich erstellt	25
Abbildung 28: Report Generator Startbildschirm.....	27

Abkürzungsverzeichnis

ERM	Entity-Relationship-Modell
GWT	Google Web Toolkit
ID	Identifikator
SDK	Software Development Kit

1 Anforderungsspezifikation

Ziel der Lehrveranstaltung IT-Projekt ist eine Vertiefung der in den vorangegangenen Semestern vermittelten Grundlagen zur Software-Entwicklung und zu Datenbanken. Im Zuge dessen sollen im Rahmen der Veranstaltungen einerseits weitere Techniken vermittelt werden. Andererseits sollen diese Techniken von allen Studierenden im Kontext eines Projekts angewendet werden. Da Software stets ein Ergebnis von Teams ist, sollen auch in diesen Veranstaltungen die Projektarbeiten im Gruppenkontext durchgeführt werden.

1.1 Rahmenbedingungen

Die Aufgabe ist in Gruppen zu je 5-6 Personen zu bearbeiten. Die Aufgabe ist so dimensioniert, dass sie auch bei Ausscheiden von Teammitgliedern durch 4 Personen erfolgreich bearbeitet werden kann.

Die Zusammensetzung der Gruppen wird den Studierenden überlassen. Die Gruppen müssen ihre Zusammensetzung spätestens eine Woche nach Bekanntgabe der Aufgabenstellung den Dozenten verbindlich über das Moodle-System mitteilen. Ansonsten erfolgt eine Zuteilung durch die Dozenten. Weiterhin müssen alle Teilnehmer spätestens zu diesem Zeitpunkt in dem bekanntgegebenen Moodle-Kurs registriert sein, um die Prüfungsvorleistung in diesem Semester erbringen zu können.

Im Projektverlauf muss jede Gruppe Teilaufgaben zusammen mit den dafür verantwortlichen Gruppenmitgliedern definieren. Zum Zeitpunkt des dritten Meilensteins (s. Tabelle 1 am Ende dieses Dokuments) müssen die für die einzelnen Teilaufgaben verantwortlichen Gruppenmitglieder genannt werden.

Die Beurteilung der praktischen Arbeit einer Gruppe erfolgt durch die o.g. Dozenten für die Gruppe als Ganzes, d.h. jedes Teammitglied erhält dieselbe Bewertung. Dies hat zur Folge, dass die gesamte Gruppe wie im beruflichen Alltag einem kollektiven Erfolgszwang gegenüber steht. Die Dozenten behalten sich eine individuelle Beurteilung von Teilnehmern in Sonderfällen vor.

Zum erfolgreichen Bestehen der Praktischen Arbeit muss jeder der in der Tabelle 1 am Ende dieser Ausführungen dargestellten Meilensteine erbracht werden. Die Note der Prüfungsleistung wird in einer individuellen mündlichen Prüfung von 20 Minuten Dauer ermittelt. Die für die erfolgreiche Bearbeitung des Projekts nötigen Kenntnisse bauen auf den in den Vorlesungen vermittelten Inhalten auf und müssen falls erforderlich selbständig erweitert werden. Zur Unterstützung der Projektteams werden anstelle der organisierten Übungsstunden Sprechstunden bei den Dozenten abgehalten, die individuell von den Projektteams gebucht oder von den Dozenten

den Projektteams zugewiesen werden. Ansonsten können die Übungsstunden in eigener Regie für Teambesprechungen, die Dokumentenerstellung und zur Programmierung verwendet werden.

1.2 Beschreibung des Zielsystems

Jede Gruppe hat die Aufgabe, ein verteiltes System zum **Management von Strukturstücklisten**, wie sie z.B. in der Produktionsplanung und –steuerung Anwendung finden, zu realisieren. Stücklisten stellen die Struktur von Baugruppen dar. Eine Baugruppe ist ein spezielles Bauteil, das aus weiteren Baugruppen und Bauteilen besteht, zu denen jeweils deren Anzahl in der übergeordneten Baugruppe anzugeben ist. Bauteile und Baugruppen können in beliebig vielen Stücklisten verwendet werden. Vergleichen Sie dies z.B. mit der Verwendung eines bestimmten Schraubentyps oder eines bestimmten Motors in PKWs. Ein Enderzeugnis ist eine spezielle Baugruppe. Das System soll das Erstellen, Verwalten und Anzeigen von Stücklisten sowie deren Elemente ermöglichen. Für eine solche Funktionsweise müssen dem System die Begriffe Stückliste, Baugruppe, Bauteil, Enderzeugnis bekannt sein. Objekte der vorgenannten Begriffe (Klassen) sollen in dem Zielsystem jederzeit angelegt, editiert und gelöscht werden können. Außerdem soll zu jedem Objekt dieser Klassen die Identität des letzten Bearbeiters sowie der Zeitpunkt der letzten Bearbeitung gespeichert werden und jederzeit nachvollziehbar sein. Für jedes Element kann eine Reihe von Attributen festgelegt werden. Dies sind für Bauteile mindestens: Teilnummer, Bezeichnung und textuelle Beschreibung des Bauteils, Materialbezeichnung (z.B. Eisen, Stahl, Messing). Jeder Stückliste soll eine Listennummer, ein Name sowie das Datum der Erstellung (dieses ggf. automatisch) zuordenbar sein. Baugruppen besitzen ebenso mindestens eine Nummer, einen Namen sowie ein Datum der letzten Änderung (auch dieses ggf. automatisch). Enderzeugnisse sind speziell gekennzeichnete Baugruppen. Gemeinsame Attribute werden dabei vorteilhaft in einer gemeinsamen Basisklasse definiert. Wie zu Beginn dieses Abschnitts erwähnt, soll es sich bei dem Zielsystem um ein verteiltes System handeln, auf das von mehreren Benutzern simultan zugegriffen werden kann. Im Zuge dessen soll eine dreischichtige Architektur¹ realisiert werden (vgl. Abbildung 1). Die unterste Schicht wird die Datenbankschicht sein. Diese Schicht sorgt für die so genannte *Persistenz*² der mit dem System gehandhabten Daten. Hier soll mit Hilfe von *Google Cloud SQL* bzw. *mySQL* eine relationale Datenbank entworfen und implementiert werden, die sämtliche Daten des Systems aufnehmen kann.

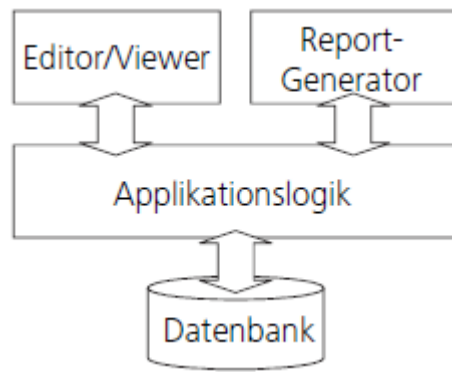


Abbildung 1: Beschreibung des Zielsystems

Die oberste Schicht wird die Präsentationsschicht sein. Sie wird durch *zwei getrennte Clients* mit graphischer Benutzungsschnittstelle (engl.: Graphical User Interface, GUI) realisiert. Der erste Client dient der Interaktion mit den Stücklisten und ihren Elementen. Der zweite Client ist ein einfacher „Report- Generator“, mit dessen Hilfe die durch die Nutzung entstandenen Daten sehr einfach zur Anzeige gebracht werden können. Die Clients werden nachfolgend als Editor/Viewer (Client 1) und als Report-Generator (Client 2) bezeichnet. Es ist darauf zu achten, dass die Clients eine klare Aufteilung von Funktionsbereichen besitzen. Zur Navigation in einer größeren Datenmenge (Editor) bietet sich z.B. die Verwendung einer scrollbaren, listen- oder baumartigen Darstellung an. Zum Erzeugen von Objekten sollte für jede Klasse (z.B. Stückliste, Bauteil, Baugruppe) eine spezifische Interaktionsform existieren, mit deren Hilfe die jeweiligen Objekte erzeugt oder gelöscht werden können. Die Interaktion wird durch Auswahl von Schaltflächen oder Selektion aus Listendarstellungen initiiert. Die Zuordnung von z.B. Bauteilen zu Baugruppen ergibt sich aus der Platzierung und Auswahl der jeweiligen Interaktionselemente. Die bloße Eingabe von Werten in Zahlen- oder Textform, wie sie z.B. von Primärschlüsseln relationaler Datenbanken bekannt sind, ist hier zur Selektion nicht ausreichend. Die Reports sollen in HTML-Form ausgegeben werden. Es ist nicht erforderlich, mit Hilfe der einzelnen Abschnitte eines Reports eine weitere Navigation in der Datenbasis zu ermöglichen. Ein Report ist also eine einmalige, statische Ausgabe. Als Reporting soll das System Strukturstücklisten ausgeben können sowie die Möglichkeit bieten, den Materialbedarf für die Produktion einer einzugebenden Anzahl von selektierbaren Enderzeugnissen zu berechnen und in einer geeigneten, aggregierten Listenform auszugeben. Die mittlere Schicht nimmt die Applikationslogik (auch Anwendungslogik oder Business-Logik genannt) auf. Sie enthält die Algorithmen, Regeln und Strukturen, die

notwendig sind, um die Elemente (Bauteile, Baugruppen, Stücklisten, etc.) und Funktionen (Zuordnen, Erstellen, Löschen, Modifizieren etc.) einer Anwendung beschreiben zu können. Damit erfolgt eine Trennung von den anderen Funktionen des Systems, die sich z.B. mit der äußeren Darstellung oder der internen Abspeicherung der Daten befassen. Für eine Applikationslogik lassen sich somit mehrere Präsentationslogiken und mehrere Speicherlogiken definieren. Die mittlere Schicht stellt den sog. *Applikationsserver* dar. Er soll der Präsentationsschicht mindestens folgende Dienste anbieten:

1. Anlegen, Editieren und Löschen von Baugruppen, Enderzeugnissen und Bauteilen,
2. Zuordnung zwischen Baugruppen, Bauteilen und Enderzeugnissen. Für das Umgehen mit Nutzerdaten (vgl. letzter Bearbeiter) soll auf die Google-Accounts-API zurückgegriffen werden.
3. Abfrage von Strukturstücklisten,
4. Abfrage des Materialbedarfs für eine gegebene Menge von Enderzeugnissen in Form von einfachen Produktionsplänen.

1) + 2) werden von dem Editor/Viewer, 3) + 4) durch den Report-Generator genutzt. Weitere Funktionen der Applikationslogik können bei Bedarf hinzugefügt werden. Es ist in dem gesamten System auf die Einhaltung der referentiellen Integrität zu achten.

Die oben genannte Funktionalität erfordert, dass sich die Applikationslogik in Kontakt mit der Datenbank befindet, um die nötigen Daten aus der Datenbank auslesen bzw. neue Daten in der Datenbank ablegen zu können. Die Datenbankschicht wird demzufolge mit dem *Datenbank-Server* sowie seiner Anbindung an den Applikationsserver (Mapper-Klassen) realisiert, der den Zugriff auf die abgespeicherten Daten als Dienst zur Verfügung stellt. Als Datenbank-Server ist *Google Cloud SQL* zu verwenden. Hierzu müssen die Gruppen mindestens ein Benutzerkonto bei diesem Dienst anlegen. Applikationsserver und Datenbankserver müssen bei der Präsentation vollständig in der Cloud laufen. Die Clients sind auf einem beliebigen Rechner, der auf die Cloud zugreifen kann, vorzuführen.

Die Teilnehmer haben dafür Sorge zu tragen, dass die Präsentation vorab in dem ihnen genannten Raum lauffähig ist. Hierzu bietet sich ein Test an. Der Raum wird den Teilnehmern rechtzeitig vorher bekannt gegeben. Termine für Tests sind rechtzeitig zu vereinbaren.

Die Aufteilung in drei Schichten bietet einige wichtige Vorteile gegenüber z.B. zweischichtigen Ansätzen. Zum einen wird durch die Konzentration sämtlicher Applikationslogik auf die mittlere Schicht eine Entkopplung der Präsentationsschicht von der Datenbank erreicht. Zum anderen können die Präsentations-Clients verändert werden, ohne die Applikationslogik modifizieren zu müssen. Darüber hinaus ist es so möglich, auf derselben Applikationslogik unterschiedliche Clients aufsetzen zu können (z.B. Editor/Viewer und Report-Generator). Dies könnten einerseits Clients bzgl. unterschiedlicher Medien (Web-basiert oder dedizierte Applikationen) oder bzgl. unterschiedlicher Benutzerrollen (z.B. für Reporting oder Systemnutzung) sein. Aus diesem Grund ist unbedingt darauf zu achten, dass die Clients keinerlei Applikationslogik enthalten. Diese ist ausschließlich im Applikationsserver zu finden!

Im Folgenden sollen noch einige weitere, schichtspezifische Anforderungen erläutert werden.

Die Präsentationsschicht soll mit Hilfe des *Google Web Toolkit* (kurz: *GWT*) realisiert werden. Hierbei dürfen keine sog. GUI-Builder verwendet werden!

Ein Teil der Datenbankschicht soll relationale Strukturen (Datenbanktabellen, Tupel, usw.) in Objekte und vice versa transformieren.

Die Applikationsschicht soll Datenstrukturen und Algorithmen realisieren, mit deren Hilfe überhaupt erst das Management von Strukturstücklisten in dem System beschrieben werden kann. Dabei ist in dieser Schicht besonders darauf Wert zu legen, dass Randbedingungen der Anwendung eingehalten werden.

Zudem soll die Applikationsschicht mit der Präsentationsschicht mittels des *GWT Remote Procedure Call* (kurz: *GWT RPC*) kommunizieren. Die Applikationslogik ist auf Basis von *Google App Engine* zu realisieren. Die Kommunikation mit dem Datenbankmanagementsystem erfolgt gemäß der Vorgaben von *Google Cloud SQL* mittels der *Java Database Connectivity (JDBC)*.

Die Datenbankschicht soll, wie bereits oben erwähnt, mit Hilfe des Datenbank-Management-Systems *Google Cloud SQL* bzw. *MySQL* realisiert werden. Hierzu sind geeignete Entity-Relationship-Modelle in Chen-Notation zu erstellen und in einer Datenbank abzubilden.

Der Text unter Kapitel 1 und die Unterkapitel 1.1 und 1.2 ist vollständig aus der Aufgabenstellung der betreuenden Professoren Peter Thies und Christian Rathke übernommen (vgl. Thies & Rathke Aufgabenstellung IT-Projekt (SS15) v1.pdf).

1.3 Entwicklerwerkzeuge

Das Projekt wurde unter Zuhilfenahme folgender Werkzeuge realisiert:

Eclipse + UML-Plugin Papyrus	für Source-Code, Use-Case-Diagramme und Klassendiagramme.
yEd Graph Editor	für das Entity-Relationship-Model.
Google Web Toolkit (GWT)	für die Präsentationsschicht.
Google App Engine	für die Applikationslogik.
Google Cloud SQL	für die Datenbank.
Microsoft Word	für die Projektdokumentation.
Facebook, Skype, WhatsApp	für die Teilnehmer-Kommunikation.
GitHub/ EGit-Plugin	für die verteilte Versionsverwaltung.

2 Meilensteine

2.1 Meilenstein 1

Meilenstein 1 bestand darin, dass sich alle Teilnehmer im Moodle-Kurs IT-Projekt anmelden und die Gruppenbildung bis zum 29.03.2015 abgegeben wurde.

2.2 Meilenstein 2

Die Aufgabe zum Meilenstein 2 beinhaltete die Erstellung der Use-Case-Diagramme. Abgabedatum war der 07.04.2015. Die Realisierung erfolgte mit Hilfe von Papyrus. Die Diagramme bzw. textuellen Beschreibungen wurden nach Feedback zum Meilenstein 2 für diese Projektdokumentation nochmals nachgearbeitet.

2.2.1 Use-Case-Diagramme

2.2.1.1 Stückliste verwalten

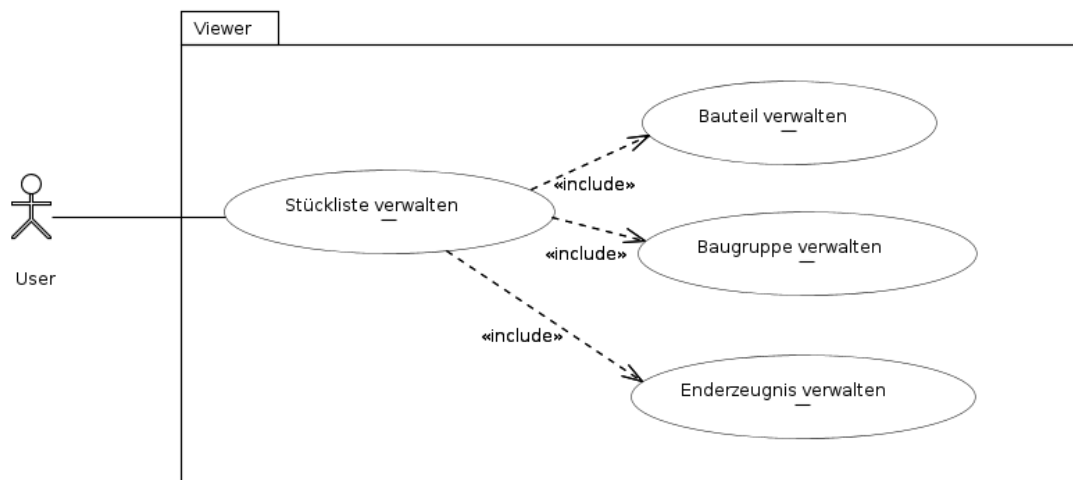


Abbildung 2: Viewer-Übersicht

<p>Name Stückliste verwalten</p> <p>Ziel Der User möchte eine Stückliste anlegen, löschen oder bearbeiten.</p> <p>Vorbedingungen - Der User hat Zugang zum System.</p> <p>Nachbedingungen -Stückliste wurde angelegt, gelöscht oder bearbeitet. -Stückliste ist jetzt im System hinterlegt oder existiert nicht mehr im System.</p> <p>Nachbedingungen im Sonderfall - dem User wird durch eine Bildschirmmeldung angezeigt, welches Objekt mit der entsprechenden Stückliste noch verknüpft ist und erhält die Option, das verknüpfte Objekt zu löschen.</p> <p>Akteure User</p>	<p>Normalablauf 1.) Der User ruft das Strukturstücklisten Management System auf. 2.) Das System hält dem User die vier Optionen bereit, eine Stückliste anzulegen, anzuzeigen, es zu löschen oder zu bearbeiten. 3.) Der User wählt eine der vier Optionen und verwaltet die Stückliste.</p> <p>Sonderfälle 2a) - das Anlegen wurde dem User verweigert, da die Stückliste bereits existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - die Bearbeitung wurde verweigert, da die Stückliste im System nicht existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - das Löschen wurde verweigert, da diese Stückliste noch mit anderen Objekten verknüpft ist (referentielle Integrität). Der User bekommt eine Bildschirmmeldung angezeigt.</p>
---	---

Abbildung 3: Stückliste verwalten - Textuelle Beschreibung

2.2.1.2 Viewer / Bauteil verwalten

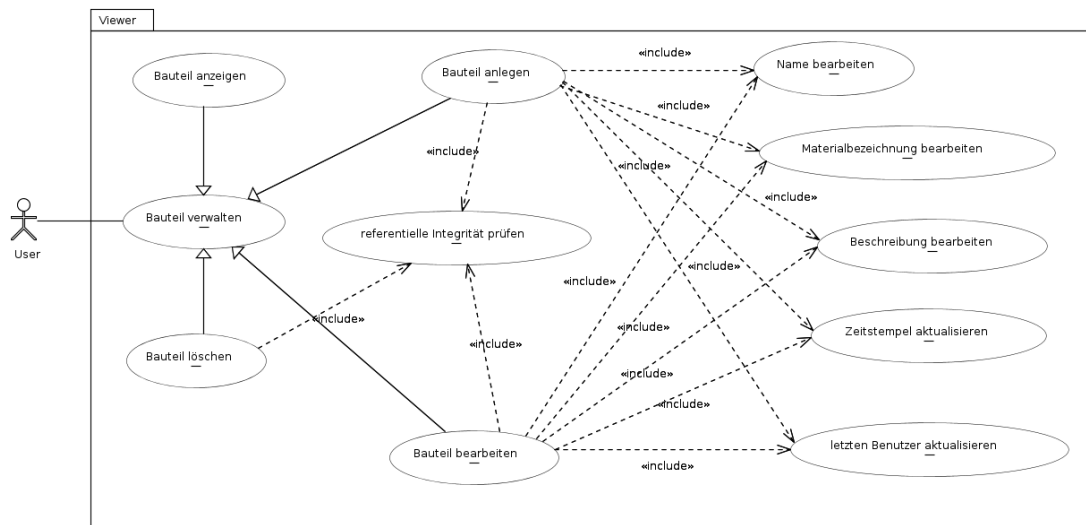


Abbildung 4: Bauteil verwalten - Diagramm

<p>Name Bauteil verwalten</p>	<p>Normalablauf 1.) Der User ruft das Strukturstücklisten Management System auf. 2.) Das System hält dem User die vier Optionen bereit, ein Bauteil anzulegen, anzuzeigen, es zu löschen oder zu bearbeiten. 3.) Der User wählt eine der vier Optionen und verwaltet das Bauteil.</p>
<p>Ziel Der User möchte ein Bauteil anlegen, anzeigen, löschen oder bearbeiten.</p>	<p>Sonderfälle 2a) - das Anlegen wurde dem User verweigert, da das Bauteil bereits existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - die Bearbeitung oder das Anzeigen wurde verweigert, da das Bauteil im System nicht existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - das Löschen wurde verweigert, da dieses Bauteil noch mit anderen Objekten verknüpft ist (referentielle Integrität). Der User bekommt eine Bildschirmmeldung angezeigt.</p>
<p>Vorbedingungen - Der User hat Zugang zum System.</p>	
<p>Nachbedingungen - Bauteil wurde angelegt, angezeigt, gelöscht oder bearbeitet. - Bauteil ist jetzt im System hinterlegt und kann für nachfolgende Operationen der Baugruppen und Enderzeugnisse verwendet werden oder existiert nicht mehr im System.</p>	
<p>Nachbedingungen im Sonderfall - Dem User wird durch eine Bildschirmmeldung angezeigt, welches Objekt mit dem entsprechenden Bauteil noch verknüpft ist und erhält die Option, das verknüpfte Objekt zu löschen.</p>	
<p>Akteure User</p>	

Abbildung 5: Bauteil verwalten - Textuelle Beschreibung

2.2.1.3 Viewer / Baugruppe verwalten

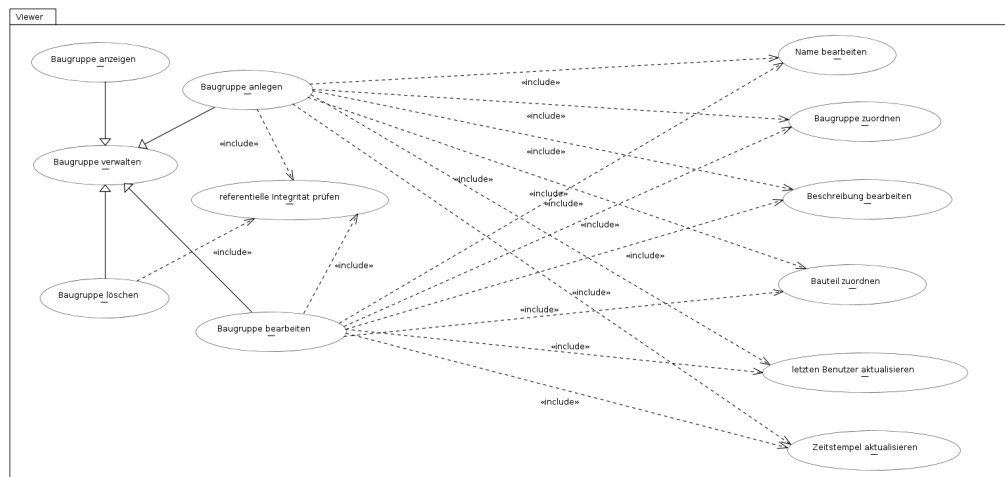


Abbildung 6: Baugruppe verwalten - Diagramm

<p>Name Baugruppe verwalten</p> <hr/> <p>Ziel Der User möchte eine Baugruppe anlegen, anzeigen, löschen oder bearbeiten.</p> <hr/> <p>Vorbedingungen - Der User hat Zugang zum System.</p> <hr/> <p>Nachbedingungen - Baugruppe wurde angelegt, angezeigt, gelöscht oder bearbeitet. - Baugruppe ist jetzt im System hinterlegt und kann für nachfolgende Operationen der Enderzeugnisse verwendet werden oder existiert nicht mehr im System.</p> <hr/> <p>Nachbedingungen im Sonderfall - dem User wird durch eine Bildschirmmeldung angezeigt, welches Objekt mit der entsprechenden Baugruppe noch verknüpft ist und erhält die Option, das verknüpfte Objekt zu löschen.</p> <hr/> <p>Akteure User</p>	<p>Normalablauf 1.) Der User ruft das Strukturstücklisten Management System auf. 2.) Das System hält dem User die vier Optionen bereit, eine Baugruppe anzulegen, anzuzeigen, es zu löschen oder zu bearbeiten. 3.) Der User wählt eine der vier Optionen und verwaltet die Baugruppe.</p> <hr/> <p>Sonderfälle 2a) - das Anlegen wurde dem User verweigert, da die Baugruppe bereits existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - die Bearbeitung oder das Anzeigen wurde verweigert, da die Baugruppe im System nicht existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - das Löschen wurde verweigert, da diese Baugruppe noch mit anderen Objekten verknüpft ist (referentielle Integrität). Der User bekommt eine Bildschirmmeldung angezeigt.</p>
---	--

Abbildung 7: Baugruppe verwalten - Textuelle Beschreibung

2.2.1.4 Viewer / Enderzeugnis verwalten

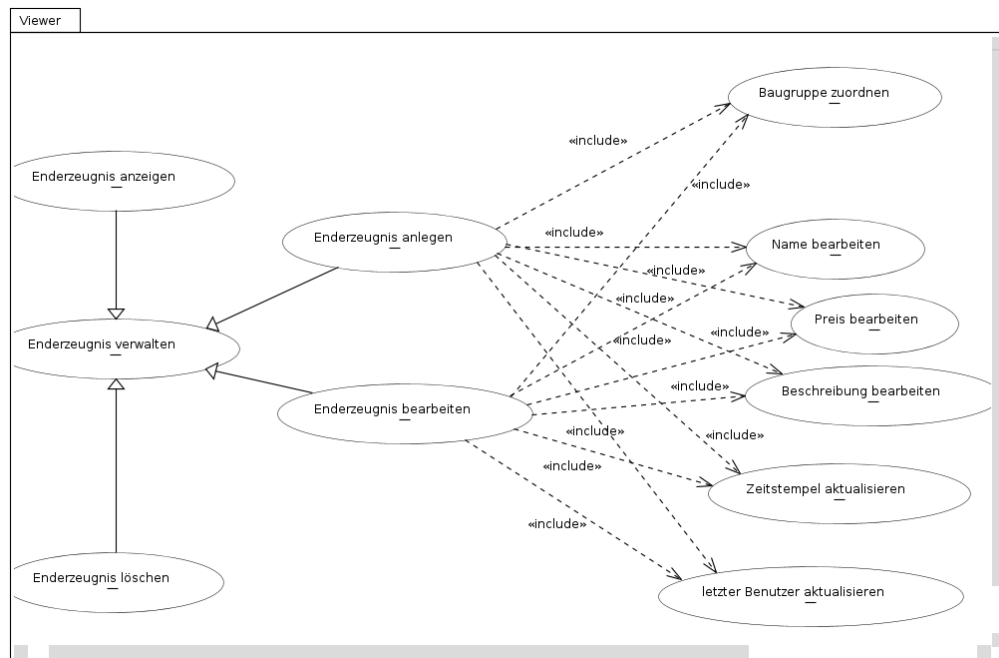


Abbildung 8: Enderzeugnis verwalten - Diagramm

<p>Name Enderzeugnis verwalten</p> <p>Ziel Der User möchte ein Enderzeugnis anlegen, anzeigen, löschen oder bearbeiten.</p> <p>Vorbedingungen - Der User hat Zugang zum System.</p> <p>Nachbedingungen - Enderzeugnis wurde angelegt, angezeigt, gelöscht oder bearbeitet. - Enderzeugnis ist jetzt im System hinterlegt oder existiert nicht mehr im System.</p> <p>Nachbedingungen im Sonderfall - dem User wird durch eine Bildschirmmeldung angezeigt, welches Objekt mit dem entsprechenden Enderzeugnis noch verknüpft ist und erhält die Option, das verknüpfte Objekt zu löschen.</p> <p>Akteure User</p>	<p>Normalablauf</p> <ol style="list-style-type: none"> 1.) Der User ruft das Strukturstücklisten Management System auf. 2.) Das System hält dem User die vier Optionen bereit, ein Enderzeugnis anzulegen, anzuzeigen, es zu löschen oder zu bearbeiten. 3.) Der User wählt eine der vier Optionen und verwaltet das Enderzeugnis. <p>Sonderfälle</p> <p>2a)</p> <ul style="list-style-type: none"> - das Anlegen wurde dem User verweigert, da das Enderzeugnis bereits existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - die Bearbeitung oder das Anzeigen wurde verweigert, da das Enderzeugnis im System nicht existiert. Der User bekommt eine Bildschirmmeldung angezeigt. - das Löschen wurde verweigert, da dieses Enderzeugnis noch mit anderen Objekten verknüpft ist (referentielle Integrität). Der User bekommt eine Bildschirmmeldung angezeigt.
---	--

Abbildung 9: Enderzeugnis verwalten - Textuelle Beschreibung

2.2.1.5 Report-Generator

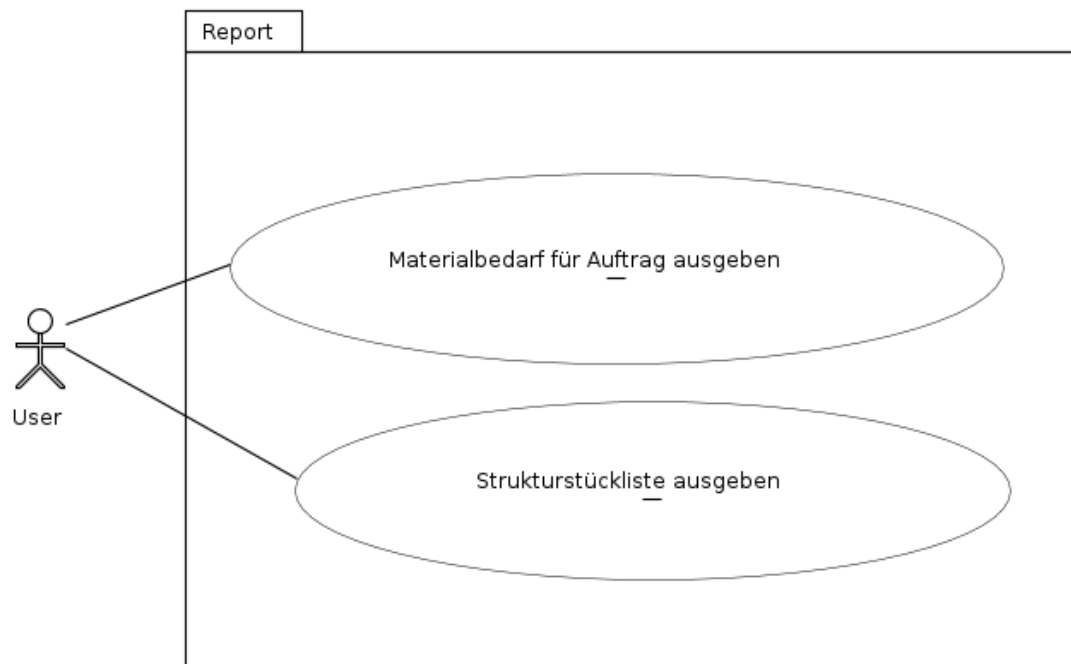


Abbildung 10: Report-Generator - Diagramm

<p>Name Strukturstückliste ausgeben</p> <p>Ziel Der User möchte einen Report in tabellarischer Form über die Strukturstückliste eines Enderzeugnis oder Baugruppe erstellen.</p> <p>Vorbedingungen - Der User hat Zugang zum System.</p> <p>Nachbedingungen - Der tabellarische Strukturstücklisten-Report wurde erstellt und ausgegeben und kann nun vom User betrachtet werden.</p> <p>Nachbedingungen im Sonderfall - dem User wird durch eine Bildschirmmeldung angezeigt, dass das Enderzeugnis nicht vorhanden ist.</p> <p>Akteure User</p>	<p>Normalablauf</p> <ol style="list-style-type: none"> 1.) Der User ruft das Strukturstücklisten Management System auf. 2.) Das System hält ihm die Option bereit, einen Strukturstücklisten-Report für ein bestimmtes Enderzeugnis oder Baugruppe zu erstellen und auszugeben. 3.) User wählt das Enderzeugnis oder Baugruppe aus. 4.) User wählt die Option Strukturstücklisten-Report erstellen. <p>Sonderfälle</p> <p>4a) Die Reporterstellung wurde verweigert, da das Enderzeugnis nicht im System vorhanden ist. Der User bekommt eine Bildschirmmeldung angezeigt.</p>
---	--

Abbildung 11: Report Generator - Textuelle Beschreibung 1

<p>Name Materialbedarf für Auftrag ausgeben</p>	<p>Normalablauf 1.) Der User ruft das Strukturstücklisten Management System auf. 2.) Das System hält die Option bereit, ein Enderzeugnis auszuwählen und die gewünschte Anzahl zu produzierender Enderzeugnisse anzugeben. 3.) Der User gibt das gewünschte Enderzeugnis an. 4.) Der User macht gibt die gewünschte Anzahl an. 5.) Der User wählt die Option Materialbedarf berechnen.</p>
<p>Ziel Der User möchte den Materialbedarf für die Produktion einer einzugebenden Anzahl von selektierbaren Enderzeugnissen berechnen und in einer geeigneten, aggregierten Listenform ausgeben.</p>	
<p>Vorbedingungen - Der User hat Zugang zum System.</p>	
<p>Nachbedingungen - Die Materialbedarfsliste wurde erstellt und ausgegeben und kann nun vom User betrachtet werden.</p>	
<p>Nachbedingungen im Sonderfall - dem User wird durch eine Bildschirmmeldung angezeigt, dass das Enderzeugnis nicht vorhanden ist.</p>	<p>Sonderfälle 4a) Die Ausgabe des Materialbedarfs wurde verweigert, da das Enderzeugnis nicht im System vorhanden ist. Der User bekommt eine Bildschirmmeldung angezeigt.</p>
<p>Akteure User</p>	

Abbildung 12: Report-Generator - Textuelle Beschreibung 2

2.3 Meilenstein 3

Zum Erreichen des 3. Meilensteins wurden die UML-Klassendiagramme und ein Entity-Relationship-Modell (ERM) erstellt, sowie das Projekt in Teilaufgaben aufgeteilt und den Teammitgliedern zugewiesen. Die Diagramme bzw. textuellen Beschreibungen wurden nach Feedback zum Meilenstein 3 und durch Erkenntnisse des weiteren Projektverlaufes für diese Projektdokumentation nochmals nachgearbeitet.

2.3.1 Klassendiagramme

2.3.1.1 Applikationslogik

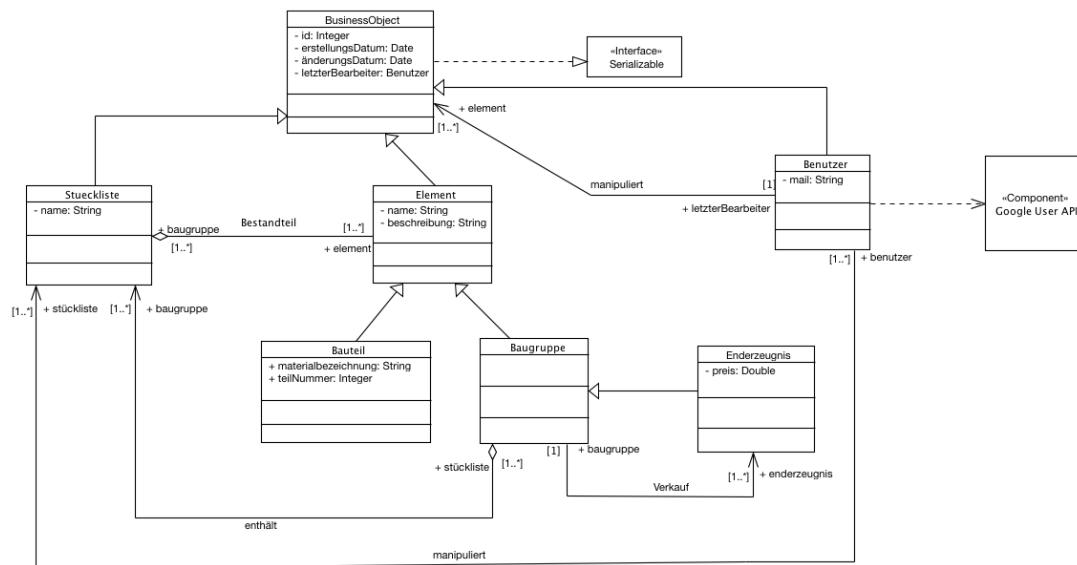


Abbildung 13: Applikationslogik

Erläuterung Klassendiagramm - Applikationslogik

An der Spitze der Vererbungshierarchie steht die Superklasse „BusinessObject“. Somit erben alle Subklassen die Attribute und Methoden von „BusinessObject“ und von dem implementierten Marker-Interface „Serializable“. Dieses Interface ist für die Übertragung der erzeugten Objekte der Klasse „BusinessObject“ zwischen Client und Server zuständig. Die Klassen „Element“ und „Benutzer“ sind Subklassen von „BusinessObject“. Des Weiteren erben die Klassen: „Bauteil“, „Baugruppe“ und „Enderzeugnis“ von „Element“.

2.3.1.2 It04gwtServiceImpl und It04gwtServiceReportImpl - DbConnection

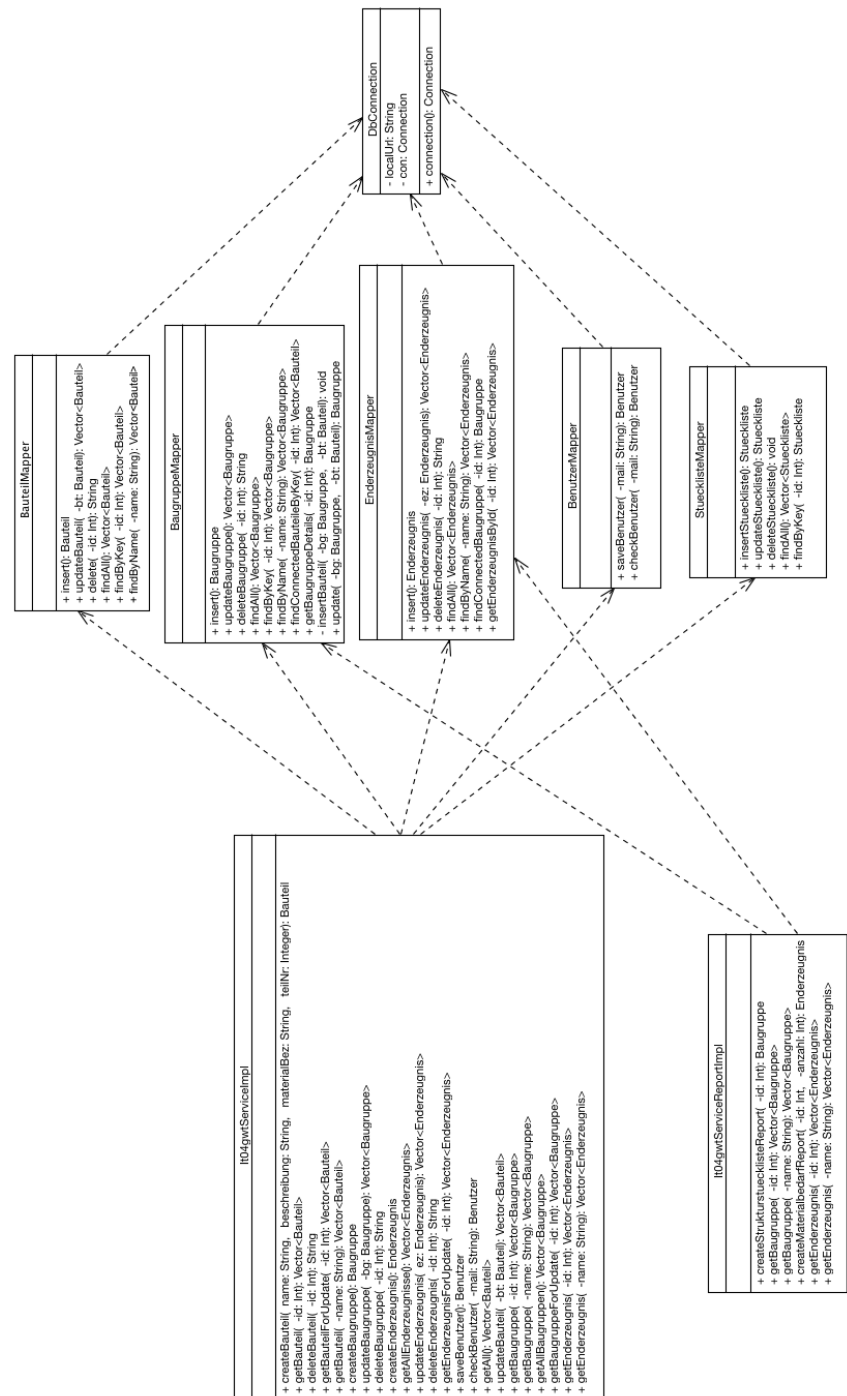


Abbildung 14: It04gwtServiceImpl – DbConnection

Erläuterung Klassendiagramm – It04gwtServiceImpl – DbConnection

Die It04gwtServiceImpl ist durch eine Abhängigkeitsbeziehung mit den Mapperklassen verbunden. Diese sind wiederum, ebenfalls durch eine Abhängigkeitsbeziehung, mit der Klasse DbConnection verbunden.

2.3.1.3 It04gwtServiceImpl – Business Objekte

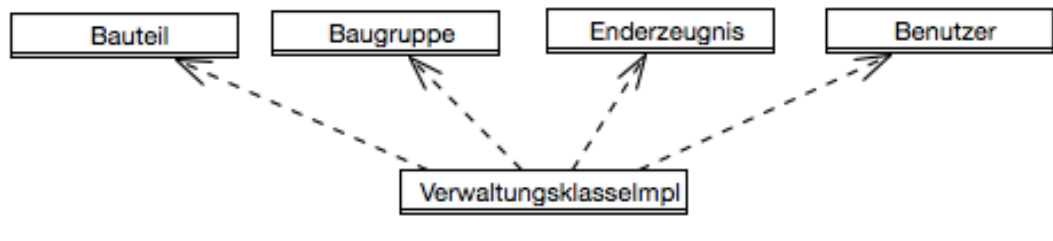


Abbildung 15: It04gwtServiceImpl - Business Objekte

Erläuterung Klassendiagramm It04gwtServiceImpl – Business Objekte

Die Klasse „It04gwtServiceImpl“ ist durch eine Nutzungsbeziehung mit den Klassen „Bauteil“, „Baugruppe“, „Enderzeugnis“ und „Benutzer“ verbunden. Die Verwaltungsklasse benötigt Zugriff auf **die Attribute der Business Objekte und befindet sich somit in einer Abhängigkeitsbeziehung.**

2.3.1.4 It04gwtServiceImpl – Synchrones Interface – Report

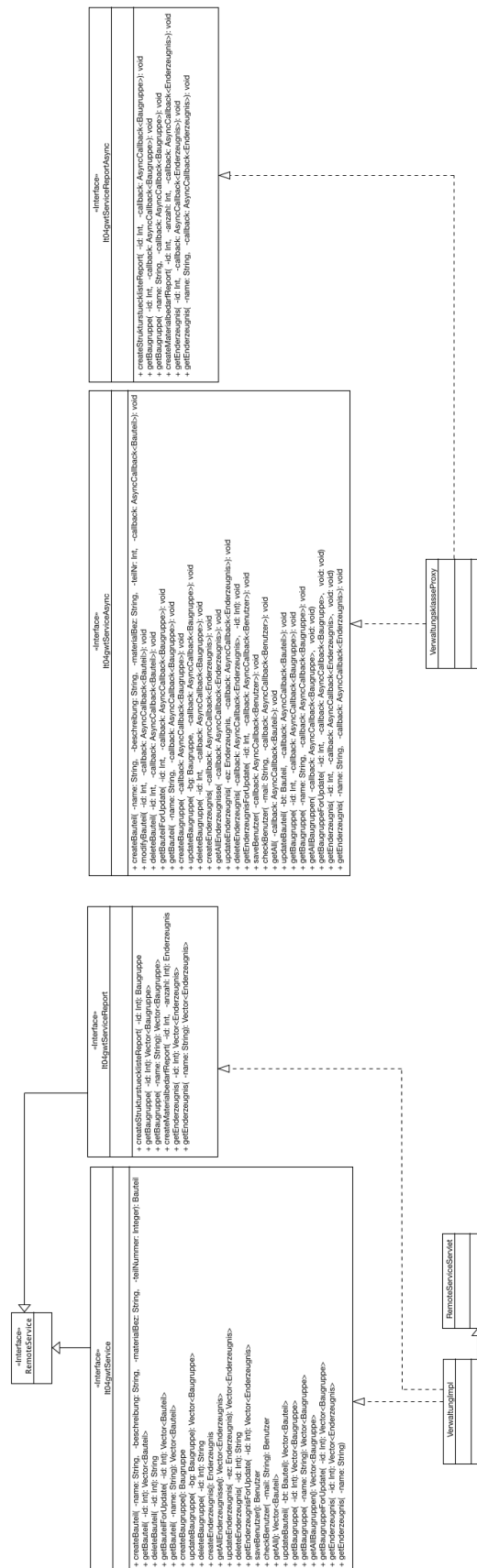


Abbildung 16: It04gwtServiceImpl - Synchrones Interface – Report

Erläuterung It04gwtServiceImpl – Synchrones Interface – Report

It04gwtServiceImpl implementiert den serverseitigen Code und implementiert das synchrone Interface It04gwtService, welche wiederum von dem Interface RemoteService erbt. Gleichzeitig erbt die It04gwtServiceImpl von der Klasse RemoteServiceServlet.

Beziehung Asynchrones Interface – VerwaltungsklasseProxy

VerwaltungsklasseProxy implementiert das asynchrone Interface It04gwtServiceAsync, welches basierend auf It04gwtService definiert wird und eine clientseitige Benutzung erlaubt.

2.3.2 Entity-Relationship-Modell

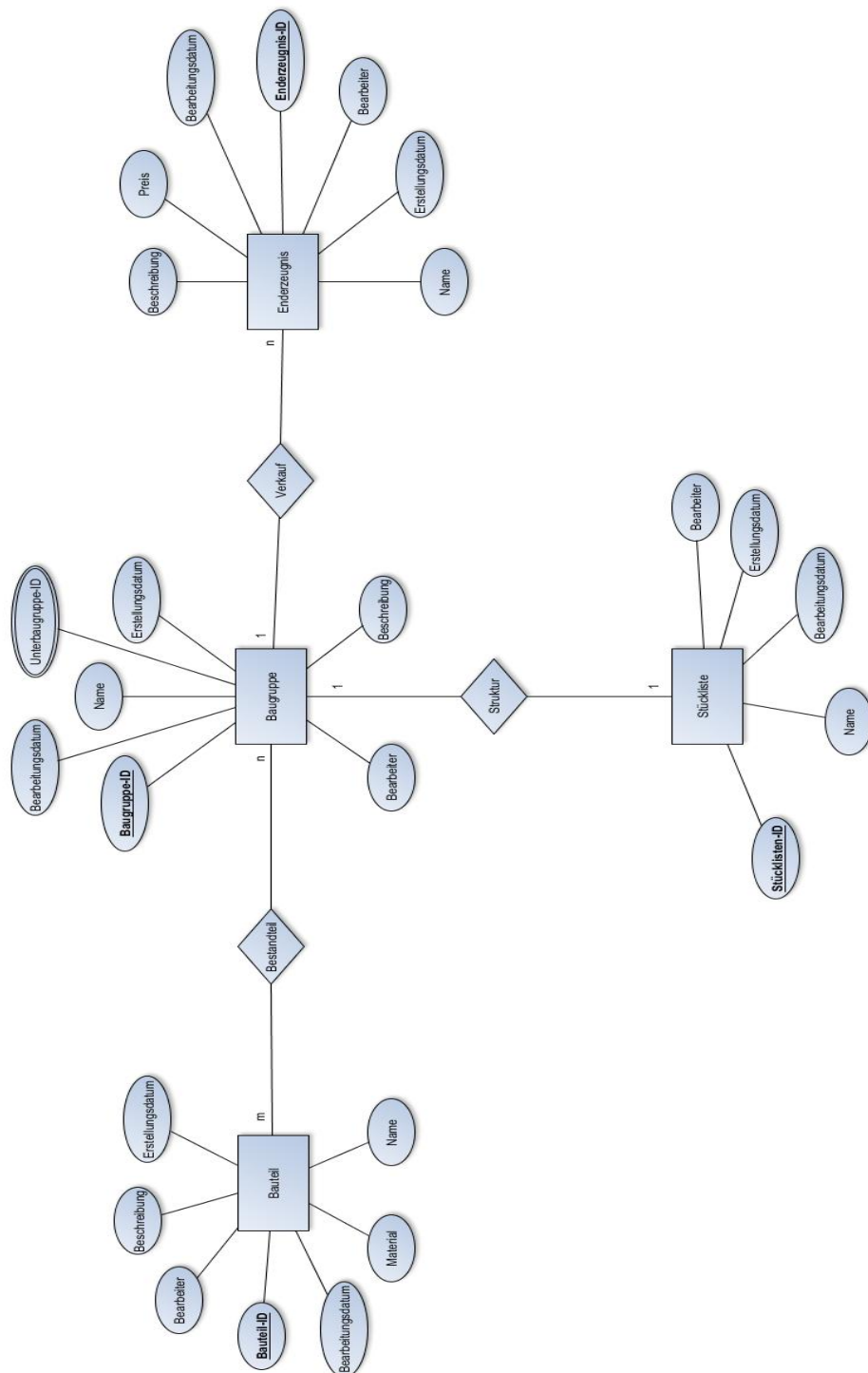


Abbildung 17: ERM

Erläuterung ERM

Das ERM stellt die Beziehungen zwischen den Entitäts-Typen Bauteil, Baugruppe, Enderzeugnis und Stückliste dar. Bei diesen Entitäts-Typen handelt es sich zusätzlich um Stammdaten.

- Ein Bauteil kann Bestandteil in mehreren Baugruppen sein.
- Eine Baugruppe kann aus mehreren Bauteilen bestehen (n:m)
- Der Entitäts-Typ Baugruppe besitzt ein Attribut „Unterbaugruppen-ID“. Dieses Attribut ist ein Fremdschlüssel und verweist auf eine andere Baugruppe (Bestandteil).
- Eine Baugruppe kann als unterschiedliche Enderzeugnisse verkauft werden. (1:n)
- Ein Enderzeugnis ist genau einer verkaufbaren Baugruppe zugeordnet.
- Eine Stückliste bildet die Struktur von genau einer Baugruppe ab. (1:1)

2.3.3 Verteilung der Aufgaben und Vorgehen im Projekt

Um eine grundsätzliche Struktur in der Teamaufteilung sicherzustellen, haben wir uns dazu entschieden, spezialisierte Zuständigkeitsbereiche innerhalb des Teams zuzuweisen.

GUI	Applikationslogik	Datenbanklogik
Fabian Tschullik	Manuel Voelker	Christian Mähler
Benjamin Schwab	Florin Geier	Friedrich Schneider

Unsere Teamstruktur ist angelehnt an das „Demokratische Team“ (vgl. Thies, 2015a, Seite 16), da alle Teammitglieder selbstbestimmt an einem gemeinsamen Team arbeiten, in Entscheidungsfällen aber von einer Person eine Richtung vorgegeben werden kann, um nicht im Entscheidungsprozess feststecken zu bleiben. Als Prozessmodell haben wir uns am Klasse I Prozessmodell Wasserfallmodell orientiert, da dieses vor allem für Einsteiger einfach anzuwenden ist und Rückschritte in vorhergehende Abschnitte möglich sind (vgl. Thies, 2015b, Seite 25). So konnten wir bspw. im Nachhinein die Use-Case und Klassendiagramm nochmals nachbearbeiten. Ab Mitte Juni musste das Projekt zu fünft weitergeführt werden, da Friedrich Schneider aus dem Projekt ausgestiegen ist.

2.4 Meilenstein 4

Der vierte Meilenstein besteht aus der Realisierung sämtlicher System-Bestandteile, Programme und Daten, Source-Code inkl. Dokumentation und dieser Projektdokumentation. Abgabedatum ist der 08. Juli 2015.

2.5 Meilenstein 5

Der fünfte Meilenstein wird zum Ziel haben, das fertige System zu präsentieren, die allgemeine Funktionalität des Systems nachzuweisen. Es werden die Allgemeine Performanz des Systems und die Teilnahme der Teammitglieder beurteilt. Die Präsentation ist auf den 09. Juli 2015 terminiert.

3 Installationsanleitung

1. Das Programmierwerkzeug Eclipse von der Website <https://www.eclipse.org/home/index.php> downloaden auf dem Rechner installieren.
2. Das Papyrus-Plugin für Eclipse installieren.
3. Das eGit-Plugin für Eclipse installieren.
4. Das Google Plugin für Eclipse installieren (Google App Engine Java SDK (Software Development Kit) 1.9.20 und Google Web Toolkit SDK 2.6.0).
5. Ggf. muss für eine korrekte Darstellung von GWT im Browser ein Google Plugin bzw. eine ältere Browserversion installiert werden.
6. Einrichten von Google App Engine und Google Cloud SQL unter Nutzung eines Google-Accounts.

Projekt erstellen unter Vergabe eines Projektnamens und einer Projekt-ID.

<input type="checkbox"/>	Projektname	Projekt-ID	Anfragen ?	Fehler ?	Gebühren ?		
<input type="checkbox"/>	API Project	api-project-1014120177582	0	0	—		
<input type="checkbox"/>	IT04SMS	it04sms	1.550	0	0,00 \$		

[Zum Löschen vorgemerkte Projekte](#)

Abbildung 18: Google Cloud - Projekt erstellen

Anschließend eine Instanz für die Datenbank Nutzung erzeugen unter Vergabe einer Instanz-ID und bei Bedarf Anpassung der erweiterten Optionen, wie bspw. Datenbank-Ebene und Leistung, Region, Tarif oder Sicherungen.

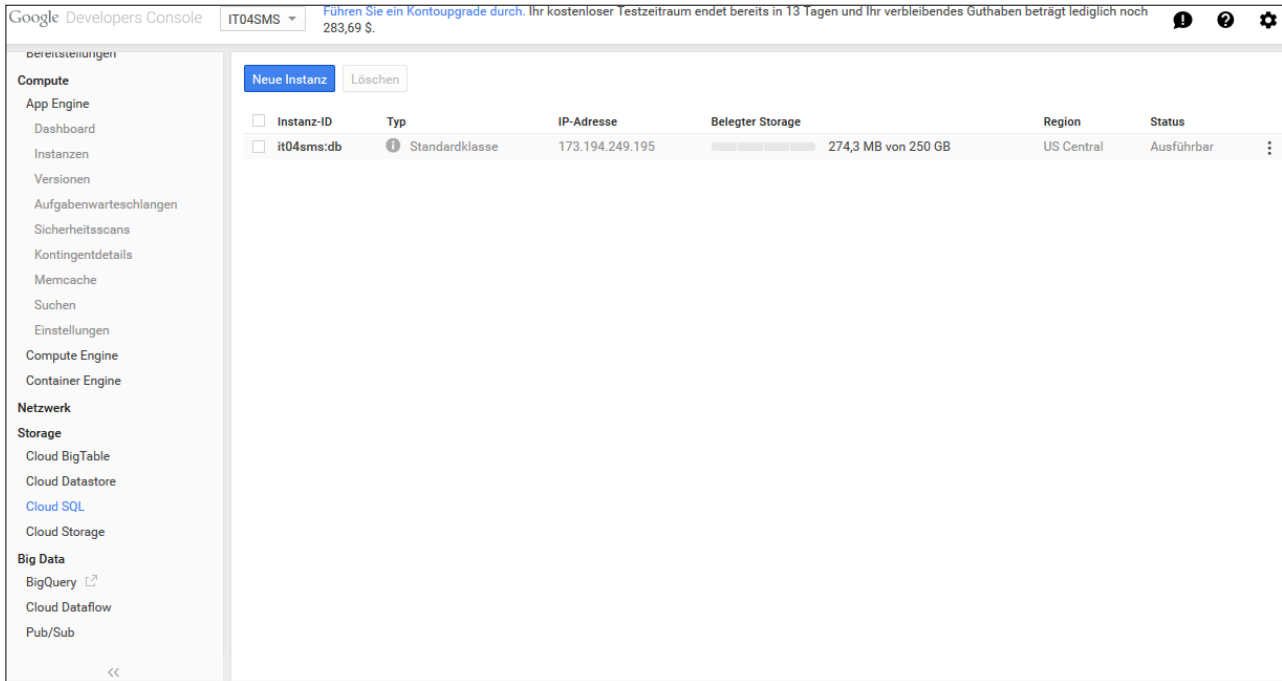


Abbildung 19: Google Cloud - Instanz erstellen

Innerhalb der Instanz kann nun die Datenbank angelegt werden.

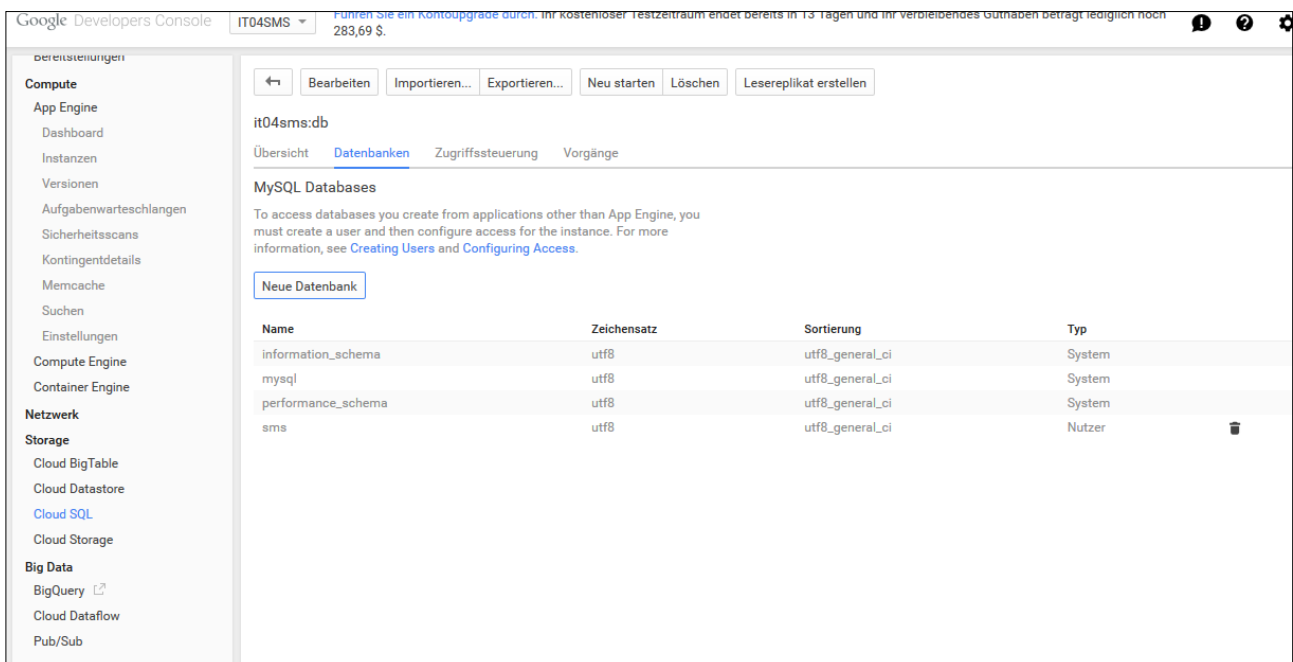


Abbildung 20: Google Cloud - Datenbank anlegen

Zum Schluss werden alle Teammitglieder und Projektbeteiligten über ihre Google E-Mail-Adresse hinzugefügt.

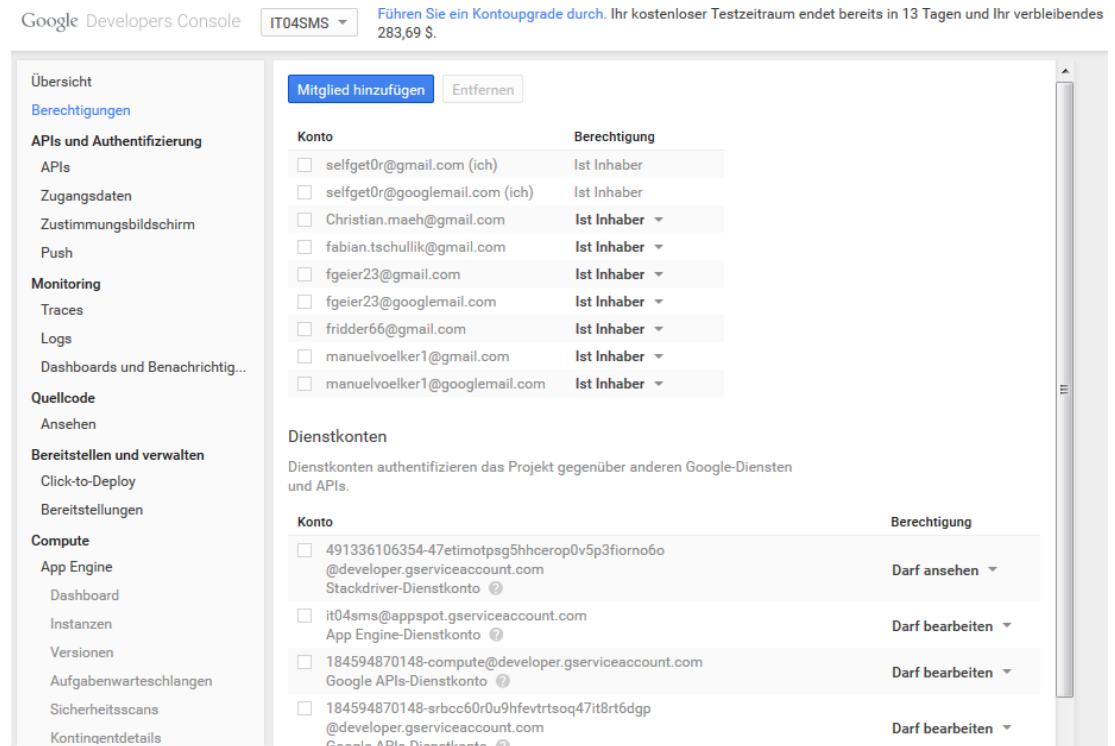


Abbildung 21: Google Cloud - Berechtigungen

- Einrichten von GitHub. Alle Teammitglieder müssen sich hierzu unter Nutzung von Klarnamen jeweils einen Account anlegen. Ein Teammitglied erstellt ein Repository und fügt die anderen Teammitglieder zu diesem Repository hinzu.

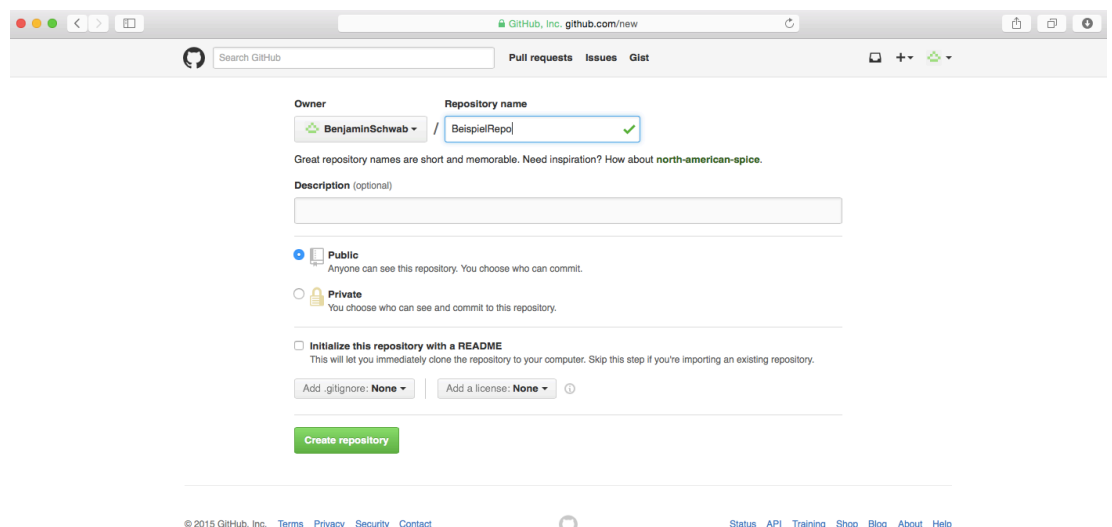


Abbildung 22: GitHub - Neues Repository

4 Benutzerhandbuch

Das Benutzerhandbuch soll dem Benutzer des Systems helfen, dieses bestimmungsgemäß zu verwenden und dient als Orientierungshilfe.

4.1 Editor

4.1.1 Allgemeines

Auf dem Startbildschirm besteht die Möglichkeit, über die Klickflächen „Bauteil“, „Baugruppe“ und „Enderzeugnis“ zur Verwaltung des jeweiligen Objekts zu gelangen. Mit einem Klick auf die Schaltfläche „Impressum“ werden Seiteninformationen nach §5 TMG angezeigt.

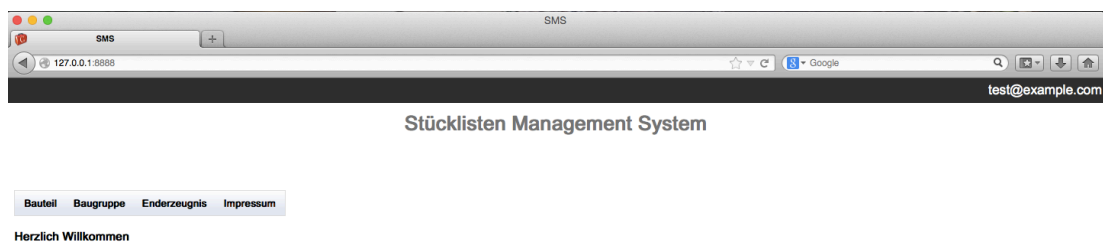


Abbildung 23: Startbildschirm Editor

4.1.2 Bauteil

Mit Klick auf die Schaltfläche „Bauteil“ im Startbildschirm öffnet sich ein Untermenü. Dieses hält die drei Optionen „alle anzeigen“, „anlegen“ und „suchen“ bereit.

4.1.2.1 Anlegen

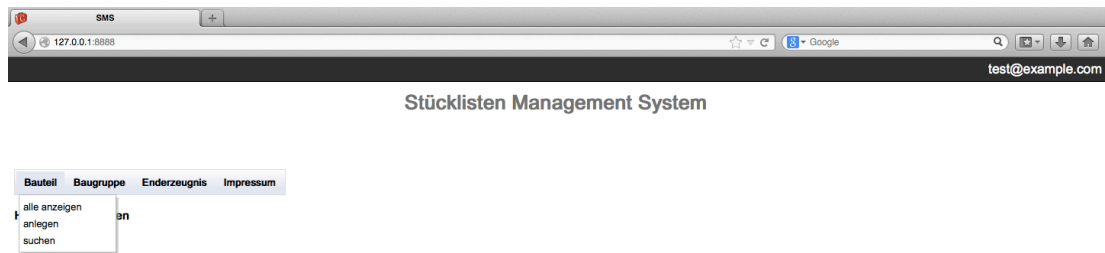


Abbildung 24: Untermenü Bauteil

Mit Klick auf „anlegen“ öffnet sich die Eingabemaske „Bauteil anlegen/ bearbeiten“.

Achtung: Mit Klick auf „anlegen“ wird bereits ein Bauteil-Objekt mit einer eindeutigen ID erzeugt. Die Benutzereingabe ergänzt anschließend die benötigten Informationen.

In den drei leeren Textfeldern „Name“, „Beschreibung“ und „Materialbezeichnung“ werden die jeweiligen Eingaben vorgenommen. Mit Klick auf den Button „speichern“ wird das Bauteil gespeichert.

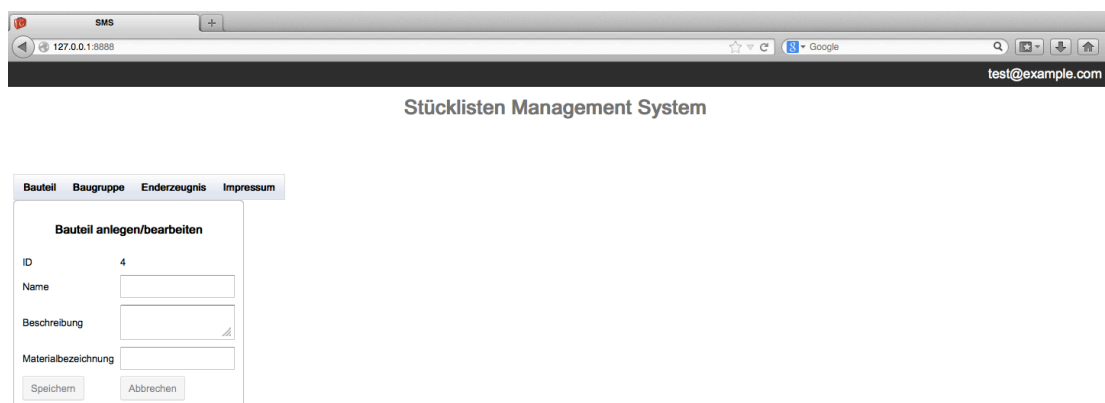


Abbildung 25: Eingabemaske Bauteil

Bei erfolgreichem Vorgang gibt das System den Hinweis „Bauteil wurde erfolgreich gespeichert“ aus und das Bauteil wird in Tabellenform angezeigt.

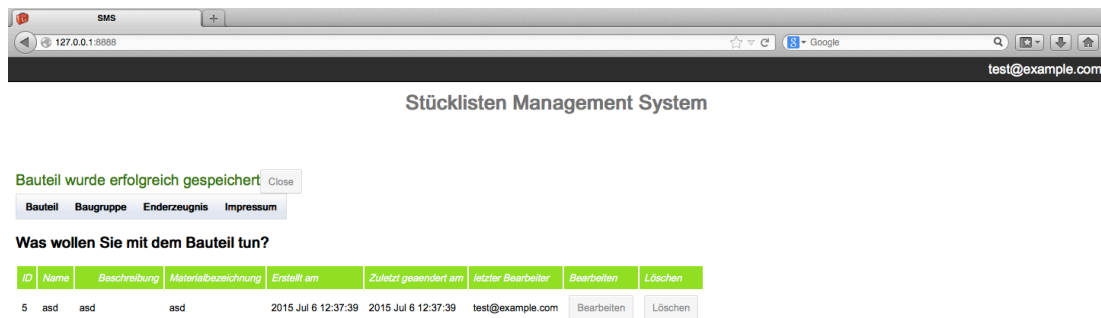


Abbildung 26: Bauteil erfolgreich erstellt

4.1.2.2 Verwalten

Um ein Bauteil zu verwalten, es damit zu bearbeiten oder zu löschen dient die Ansicht aus Abbildung 26. Diese Ansicht kann über drei verschiedene Wege erreicht werden:

1. Die Ansicht wird beim Anlegen eines neuen Bauteils geladen.
2. Die Ansicht wird geladen, wenn man über den in Abbildung 24 gezeigten Untermenü-Punkt „suchen“ ein Bauteil anhand seiner ID oder seines Namens sucht.
3. Die Ansicht wird geladen, wenn man sich über den in Abbildung 24 gezeigten Untermenü-Punkt „alle anzeigen“, alle existierende Bauteile anzeigen lässt. In der angezeigten Tabelle kann anschließend über die jeweiligen Buttons „bearbeiten“ die Ansicht zum Verwalten geladen werden oder „löschen“ das Bauteil entfernt werden.

4.1.3 Baugruppe

Mit Klick auf die Schaltfläche „Baugruppe“ im Startbildschirm öffnet sich ein Untermenü. Dieses hält die drei Optionen „alle anzeigen“, „anlegen“ und „suchen“ bereit.

4.1.3.1 Anlegen

Mit Klick auf „anlegen“ öffnet sich die Eingabemaske „Baugruppe anlegen/bearbeiten“.

Achtung: Mit Klick auf „anlegen“ wird bereits ein Baugruppen-Objekt mit einer eindeutigen ID erzeugt. Die Benutzereingabe ergänzt anschließend die benötigten Informationen.

In den beiden leeren Textfeldern „Name“ und „Beschreibung“ werden die jeweiligen Eingaben vorgenommen. Mit Klick auf den Button „speichern“ wird die Baugruppe gespeichert.

4.1.3.2 Verwalten

Um eine Baugruppe zu verwalten

4.1.4 Enderzeugnis

Mit Klick auf die Schaltfläche „Enderzeugnis“ im Startbildschirm öffnet sich ein Untermenü. Dieses hält die drei Optionen „alle anzeigen“, „anlegen“ und „suchen“ bereit.

4.1.4.1 Anlegen

Mit Klick auf „anlegen“ öffnet sich die Eingabemaske „Baugruppe anlegen/bearbeiten“.

Achtung: Mit Klick auf „anlegen“ wird bereits ein Enderzeugnis-Objekt mit einer eindeutigen ID erzeugt. Die Benutzereingabe ergänzt anschließend die benötigten Informationen.

In den drei leeren Textfeldern „Name“, „Beschreibung“ und „Preis“ werden die jeweiligen Eingaben vorgenommen. Mit Klick auf den Button „speichern“ wird die Baugruppe gespeichert.

4.1.4.2 Verwalten

Um ein Enderzeugnis zu verwalten

4.2 Report Generator

4.2.1 Allgemeines

Der Report Generator dient der Erstellung von HTML Reports. Auf dem Startbildschirm besteht die Möglichkeit, über die Klickflächen „Strukturstückliste“, und „Materialbedarf“ zur Verwaltung des jeweiligen Reports zu gelangen. Mit einem Klick auf die Schaltfläche „Impressum“ werden Seiteninformationen nach §5 TMG angezeigt.

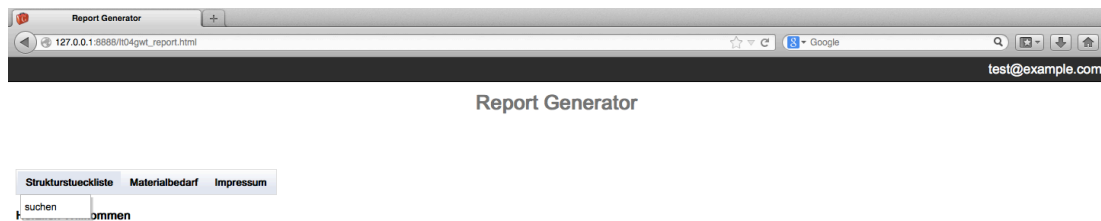


Abbildung 27: Report Generator Startbildschirm

4.2.2 Strukturstückliste

Um sich die Strukturstückliste einer Baugruppe ausgeben zu lassen, ist folgende Vorgehensweise erforderlich:

1. Klick auf „Strukturstueckliste“ im Startbildschirm. Es öffnet sich ein Untermenü mit der Option „suchen“.
2. Klick auf „suchen“. Es öffnet ein Formular mit Eingabefeld, in welche die ID oder der Name einer Baugruppe eingegeben werden kann. Mit Klick auf „suchen“ wird bei erfolgreicher Suche die gewünschte Baugruppe in Tabellenform angezeigt.
3. Klick auf „Erstellen“, um die Strukturstückliste zu erstellen und anzeigen zu lassen.

4.2.3 Materialbedarf

Um sich den Materialbedarf zur Herstellung eines Enderzeugnisses berechnen und ausgeben zu lassen, ist folgende Vorgehensweise erforderlich:

1. Klick auf „Materialbedarf“ im Startbildschirm. Es öffnet sich ein Untermenü mit der Option „suchen“.
2. Klick auf „suchen“. Es öffnet ein Formular mit Eingabefeld, in welche die ID oder der Name eines Enderzeugnisses eingegeben werden kann. Mit Klick auf „suchen“ wird bei erfolgreicher Suche das gewünschte Enderzeugnis in Tabellenform angezeigt.
3. In das Eingabefeld der Spalte „Menge“ wird die gewünschte Anzahl von Enderzeugnissen eingegeben.
4. Klick auf „Berechnen“, um die Materialbedarfsliste zu erstellen und anzeigen zu lassen.

5 Fazit

Im Rahmen der Lehrveranstaltung 335040c IT-Projekt im Modul 335040 Informationssysteme haben wir ein Stücklisten Management System erstellt und hierbei einen typischen Softwareentwicklungsprozess anhand eines praktischen Fallbeispiels realisiert. Wir haben erkannt, welchen Aufwand schon das Erstellen eines Systems mit vermeintlich geringem Funktionsumfang mit sich bringt. Dazu zählt nicht nur der reine Zeitaufwand für die Erstellung des Codes, sondern meint den Gesamtaufwand für das Projekt, sei es die Planung, Erstellung von Diagrammen, Nachbearbeitung, benötigte und eingesetzte Zeit für Teammeetings und Absprachen etc.

Literatur- und Quellenverzeichnis

Thies, P.; Rathke, C. (2015): Aufgabenstellung IT-Projekt (SS15) v1.pdf

Thies, P. (2015a): Vorlesungsskript: Software Engineering – Der Software-Lebenslauf, 05 SW-Lebenslauf-3.pdf

Thies, P. (2015b): Vorlesungsskript: Software Engineering - Der Software-Lebenslauf, 05 SW-Lebenslauf-4.pdf