

<TITEL DES PRAKTIKUMS>

<TITEL DES TEILPROJEKTES>

Software-Entwicklungspraktikum (SEP)
Sommersemester 2012

G r o b e n t w u r f



Auftraggeber
Technische Universität Braunschweig
<Name des Instituts>
<Name des Institutsleiters>
<Straße und Hausnummer>
<Postleitzahl und Ort>

Betreuer: <Name>

Auftragnehmer <überzählige Zeilen löschen>

Name	E-Mail-Adresse
<Name des Teilnehmers>	<E-Mail-Adresse>
<Name des Teilnehmers>	<E-Mail-Adresse>
<Name des Teilnehmers>	<E-Mail-Adresse>
...	...

Braunschweig, <Erstellungsdatum>

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
...

Status: „in Bearbeitung“ oder „abgenommen“ Kommentar: hier eintragen, was geändert bzw. ergänzt wurde

Hinweis zum Template: Dieses Template enthält Hinweise, die alle kursiv geschrieben sind. Alles Kursivgeschriebene ist selbstverständlich bei Abgabe zu entfernen sind. Angaben in <...> sind mit dem entsprechendem Text zu füllen. Überzählige Kapitel, d.h. Kapitel, die nicht bearbeitet werden müssen, da sie nicht der Aufgabenstellung entsprechen, bitte entfernen.

Aufgabe des Grobentwurfs: Aufgabe dieses Dokumentes ist es, die Architektur des Systems zu beschreiben und die daraus resultierenden Pakete durch die Definition von Schnittstellen zu Komponenten auszubauen.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Projektdetails	5
2	Analyse der Produktfunktionen	6
2.1	Analyse von Funktion F100 (Anmeldung)	6
2.2	Analyse von Funktion F101 (Anmeldung über LDAP)	6
2.3	Analyse von Funktion F102 (Anbindung an die Universitätsbibliothek)	7
2.4	Analyse von Funktion F103 (Mailtexte ändern)	7
2.5	Analyse von Qualitätsmerkmal /Q10/ (Sicherheit der Anmeldung)	7
3	Resultierende Softwarearchitektur	8
3.1	Komponentenspezifikation	8
3.2	Schnittstellenspezifikation	8
3.3	Protokolle für die Benutzung der Komponenten	8
4	Verteilungsentwurf	10

Abbildungsverzeichnis

Abbildung 1: Aktivitätsdiagramm Projektdetails 5

Abbildung 2: Verteilung von <ID> 7

Abbildung 3: Sequenzdiagramm für <ID> 7

Abbildung 4: Komponentendiagramm 8

Abbildung 5: Protokollstatechart für jede Komponente 9

Abbildung 6: Verteilungsdiagramm 10

1 Einleitung

Hier ist die Arbeitsweise des Systems anhand von Aktivitätsdiagrammen darzustellen und kurz zu erläutern.

1.1 Projektdetails

Die Details der Aufgabenstellung bitte anhand von Aktivitätsdiagramme darstellen und kurz verbal beschreiben.

2 Analyse der Produktfunktionen

Im folgenden werden die Funktionen und Qualitätsmerkmale des System auf ihre Modulzugehörigkeit hin geprüft und eingeordnet. Dabei wird auf das Basis-Konstrukt von Django aufgebaut und die Funktionen bzw. Qualitätsmerkmale werden in unterschiedliche Kategorien eingeteilt.

Hier zuerst die Kategorien in die, Die einzelnen Funktionen und Qualitätsmerkmale eingeordnet werden.

- Build in Django
- Model einer Django-App
- Klassen bzw. Objectmethode
- Template
- View im Backend
- View im Frontend

2.1 Analyse von Funktion F100 (Anmeldung)

Diese Funktion ist eine Build-In-Funktion von Django. Innerhalb von Django werden die Benutzer durch eine eigenständige Klasse dargestellt. Diese Klasse muss erweitert werden um die zusätzlichen Daten der Benutzer zu speichern. Da Django eine Datenbank zugrunde liegt müssen diese weiteren Daten in einer Extra Klasse mit Referenz auf die Benutzer gesetzt werden.

2.2 Analyse von Funktion F101 (Anmeldung über LDAP)

Diese Funktion muss in einer extra App dargestellt werden. Die LDAP-Anmeldung muss dabei die normale Anmeldung erweitern und im View evtl. durch eine Checkbox zur Verfügung gestellt werden.

2.3 Analyse von Funktion F102 (Anbindung an die Universitätsbibliothek)

2.4 Analyse von Funktion F103 (Mailtexte ändern)

Die Mailtexte sind Bestandteil einer veränderbaren aber nicht großen Tabelle an Basisinformationen wie auch andere Einstellungen für die Software. Um die Informationen bereit zu stellen muss eine neue Klasse erzeugt werden, das alle Informationen für die App bereit stellt.

2.5 Analyse von Qualitätsmerkmal /Q10/ (Sicherheit der Anmeldung)

Die Verschlüsselung der Anmeldung wird von Django bereit gestellt und ist daher bereits eine Build-In-Funktion.

3 Resultierende Softwarearchitektur

Dieser Abschnitt hat die Aufgabe, einen Überblick über die zu entwickelnden Komponenten und Subsysteme zu liefern.

3.1 Komponentenspezifikation

In diesem Abschnitt wird die aus der Analyse der Produktfunktionen (Kapitel 2) resultierende Komponentenstruktur zunächst überblickartig durch ein Komponentendiagramm beschrieben. Die Bezeichnungen und Anzahl der Komponenten muss natürlich konsistent sein mit der in Kapitel 2!

3.2 Schnittstellenspezifikation

Im Folgenden werden die einzelnen Schnittstellen der Komponenten aus der Komponentenspezifikation näher erläutert, d.h. die von Ihnen zur Verfügung gestellten Operationen werden dokumentiert. Die Tabelle ist dabei um so viele Zeilen zu erweitern, wie es Schnittstellen im Komponentendiagramm gibt. In der innen liegenden Aufteilung ist für jede Operation einer Schnittstelle eine Zeile einzufügen. Reine Set- und Get-Aufrufe brauchen nicht aufgeführt zu werden (sollten auch möglichst nicht komponentenübergreifend auftauchen).

Schnittstelle	Aufgabenbeschreibung
<Schnittstellen – ID>:	Operation Beschreibung
<Bezeichnung>	<Signatur der Operation> <Aufgabenbeschreibung der Operation>

3.3 Protokolle für die Benutzung der Komponenten

In diesem Abschnitt wird mit Hilfe von Protokoll-Statecharts die korrekte Verwendung der zu entwickelnden Komponenten dokumentiert. Dies ist insbesondere für diejenigen Komponenten notwendig, für die eine Wiederverwendung möglich erscheint oder sogar bereits geplant ist.

Begründen Sie für welche Komponenten eine Wiederverwendung sinnvoll erscheint und für welche nicht!

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

Fügen Sie so viele Statechartdiagramme ein, wie sie Komponenten gefunden haben.

4 Verteilungsentwurf

Sollte es sich bei dem Produkt um eine verteilte Anwendung handeln, so wird diese in diesem Abschnitt dokumentiert. Die Verteilung der Komponenten auf die beliebigen Knoten wird durch das folgende Verteilungsdiagramm beschrieben.

Eigenes Verteilungsdiagramm einsetzen!