Pflichtenheft Projektarbeiten-Tool

7. April 2017

Zusammenfassung

Für das Fach Systemadministration soll eine Webplattform entwickelt werden, die dazu dient, den Abgabeprozess für die Dokumentation der Projekte zu verwalten.

1 Einleitung

Im Fach Systemadministration müssen die Studierenden jedes Semester in Gruppen Projekte umsetzen. Dabei sieht die Organisation der Projekte verschiedene Meilensteine vor, zu denen die Studierenden Dokumente einreichen müssen: Projektskizze, Pflichtenheft, Beta-Version der Dokumentation, finale Dokumentation. Nach jeder Abgabe erfolgt eine Feedback-Runde durch andere Gruppen, die nach dem Zufallsprinzip bestimmt werden.
Aktuell wird der Kurs über die E-Learning Plattform der Hochschule, Moodle, verwaltet. Die weitere Kommunikation zu den Abgaben findet per Email statt. Da die bisherige Lösung die Anforderungen des mehrstufigen Abgabeprozesses nicht vollständig erfüllt, wird ein administrativer Mehraufwand generiert.
Im Rahmen des Projektes Projektarbeiten-Tool soll eine webbasierte Anwendung entwickelt werden, die den Abgabeprozess besser unterstützt, als die bestehende Lösung.

1.1 Ziel

Es soll eine Anwendung entwickelt werden, die die bestehende Lösung ablöst. Die Anwendung soll den Abgabeprozess für die Studierenden und den Dozenten klar strukturiert darstellen. Die Abgaben zu den einzelnen Meilensteinen sowie das dazugehörige Feedback sollen über die Anwendung erfolgen. Die Studierenden sollen automatisierte Erinnerungs-Emails zu den Abgabeterminen erhalten.

2 Grundbegriffe

3 Probleme der aktuellen Umsetzung

- Der Abgabeprozess muss derzeit manuell überwacht werden:
  - Zu welchen Terminen müssen die jeweiligen Dokumente abgegeben werden?
  - Welche Dokumente hat eine Gruppe schon eingereicht? War die Abgabe termingerecht?
  - Benachrichtigungen/Terminerinnerungen per Email müssen manuell versendet werden
- Feedbacksystem entspricht nicht den Anforderungen:

  - Feedback ist nicht anonym
  - Zuteilung der Feedback-Gruppen erfolgt nicht automatisiert
  - Feedback erfolgt als Fließtext; es fehlt ein standardisiertes Formular
  - Feedback kann ggf. auf Grund unterschiedlicher Abgabe-Formate nicht einheitlich zusammengefasst werden und wird dadurch unübersichtlich
- Bisher ist keine Zuordnung der abgegebenen Dokumentationen zu einem Semester möglich
- Aktuell müssen die abgegebenen Dokumente am Ende des Semesters von Hand gesichert werden

4 Anforderungsanalyse

4.1 Allgemein

Muss - Ziel Das verwendete Betriebssystem muss Linux sein.
Auf das System muss über das Netzwerk der Hochschule zugegriffen werden können.
Der Zugriff auf das System muss per Webbrowser(HTTP,HTTPS) und SSH möglich sein.
Das System soll Webbasiert sein.
Das System muss dem Dozenten eine Möglichkeit zur Konfiguration bieten: Meilensteine (Was muss wann abgegeben werden), e-Mail Benachrichtigung(Text Vorlagen, Erinnerungstermine), User (Wer? und Gruppen)
Am Ende des Semesters müssen die abgegebenen Dokumente zur Archivierung zur Verfügung gestellt werden. (Versand per E-Mail/ Download per Webinterface)

Kann - Ziel Auf das System kann über das Internet zugegriffen werden.

4.2 Benutzerverwaltung

Muss - Ziel Benutzer müssen sich zur Nutzung des Systems authentifizieren können.
Einzelne Studenten müssen zu Projektgruppen zusammengefasst werden können.
Das Webinterface muss 2 Berechtigungsstufen und Ansichten haben: Dozent und Student. Der Dozent hat Rechte zur Konfiguration.

Kann - Ziel Eine Authentifikation gegenüber dem LDAP - Server der Hochschule ist wünschenswert.

4.3 Feedback

Muss - Ziel Über das Webinterface muss vom Studierenden Feedback zu den Dokumentationen gegeben werden können.
Das Feedback zu den abgegebenen Dokumenten muss anonym sein. D.h. das Feedback darf den Verfassern nicht zuordenbar sein.
Das Feedback muss den abgegebenen Dokumenten zuordenbar sein.
Um das Feedback zu geben müssen Formulare zur Verfügung gestellt werden.

Kann - Ziel Die Formulare können über die Ansicht des Dozenten konfiguriert werden.

4.4 e-Mailing

Muss - Ziel Das System soll e-Mails automatisiert an die Studierenden versenden: Infos zu Deadlines.
Die e-Mails sollen aus verschiedenen Vorlagen erzeugt werden.

4.5 Abgabeprozess

Muss - Ziel Die abgegebenen Dokumente müssen gespeichert und den Projektgruppen zugeordnet werden können.
Abgegebende Dokumente müssen, für das Feedback, gleichmäsig an Gruppen verteilt werden.
Studenten müssen ihre Dokumente hochladen können.
Studenten müssen das Feedback zu ihren abgegebenen Dokumenten einsehen können.
Der Dozent muss sehen können, wie der Abgabestatus der einzelnen Projektgruppen ist.
Der Dozent muss die Meilensteine sehen können.
Die Studenten müssen die Meilensteine und ihren Abgabestatus sehen können.
Die Studenten dürfen nur den eigenen Abgabestatus, nicht den der anderen Projektgruppen sehen können.

5 Lösungsvorschläge

5.1 Allgemein

Prinzipiell muss das System aus folgenden Komponenten bestehen: Betriebssystem, Webserver, Datenbankserver, Mailserver. Abbildung 1 zeigt einen ersten Entwurf der Komponenten.

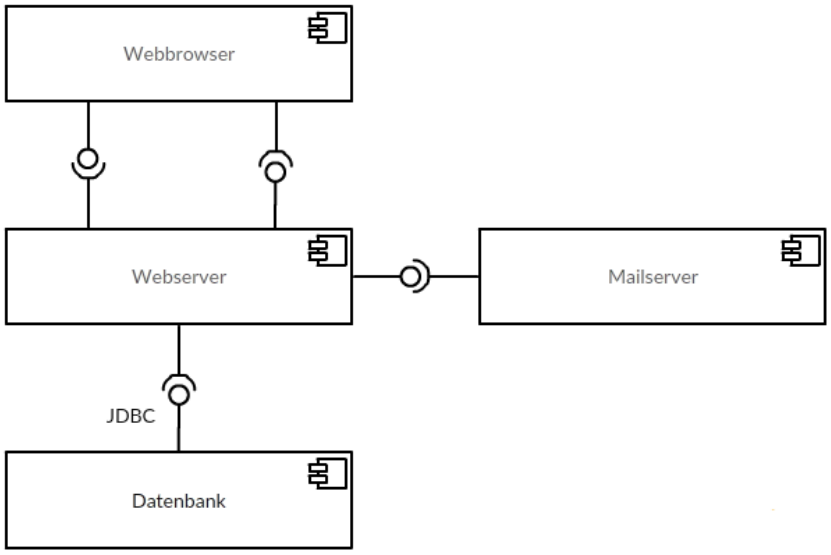


Abbildung 1: Erster Entwurf des Komponentendiagramms

Auswahl der Linux Distribution Das Linux System muss als Distribution verfügbar sein. Programm sollten über eine Packetverwaltung installierbar sein. Müssen also nicht grundsätzlich selbst kompiliert werden.
Das System benötigt keine graphische Benutzeroberfläche.
Sicherheitsupdates sollten regelmäßig verfügbar sein.
Sicherheitsupdates sollten über mehrere Jahre hinweg erscheinen.
Die Linux Distribution sollte eine zum Projektzeitraum aktuelle Kernelversion verwenden. (04.04.2017: stable 4.10.8, longterm 4.9.20, longterm 4.4.59)
Die Linux Distribution sollte kostenlos verfügbar sein.

Red-Hat, SUSE Enterprise Die Distributionen Red-Hat, Marktführer im Serverbereich, und SUSE Enterprise sind kostenpflichtig.

Ubuntu Desktop Frei verfügbare Desktop Version von Ubuntu. Basiert auf Debian. Viele desktopspezifische Anwendungen, die im Server Bereich nicht benötigt werden und zu unnötigem Overhead und unter Umständen zu Sicherheitslücken führen.
Support lifespan: 3 Jahre
Kernel: 4.10

Ubuntu Server LTS Serverversion der Ubuntu Distribution. Keine desktopspezifischen Anwendungen vorinstalliert.
Support lifespan: 5 Jahre
Kernel 4.4 (Stand 04.04.2017)

Debian Bekannte Linux Distribution. Sehr ausführliche Sicherheitsinformationen verfügbar.
Kernel 3.16 (Stand 04.04.2017)

CentOS Als Offene Alternative zu Red-Hat gibt es CentOS, das vom selben Hersteller stammt und als Testplattform für Red-Hat Updates dient.
End-Of-Life: Version 7.7: 20.06.2024 (Stand 04.04.2017)
Kernel: 3.10

Die Entscheidung ist auf Ubuntu Server LTS gefallen. Entscheidend war der lange Support, ein aktueller Kernel, und den Profit durch Debian als Basis.

5.2 Datenhaltung

Speicherung der Daten Zum Einen müssen Daten über die Konfiguration und den Abgabeprozess gespeichert werden. Hierfür bietet sich eine Datenbank an. Zum Anderen müssen die abgegebenen Dokumente gespeichert werden. Es gibt zwei Möglichkeiten um die Dokumente zu speichern. Entweder auf einem Fileserver, oder in einer Datenbank.

Entscheidung Datenbank oder Fileserver In Datenbanken müssen Dateien in der Regel in einem Binärformat abgelegt werden. Beispielsweise bieten MySQL und Oracle den Datentyp BLOB (Binary Large Object), unter Postgressql wird ein Binarystream verwendet. Vorteile gegenüber dem Dateisystem sind: Transaktionen, Referenzielle Integrität (Referenzen in der Datenbank auf Dateien im Fileserver), einfaches Verwalten größerer Datenmengen (Indexing).

Da eine Datenbank unabhängig vom Speichern der Dokumente benötigt wird, generiert ein zusätzlicher Fileserver einen Mehraufwand, da dieser ebenfalls administriert werden muss.

Auswahl des Datenbanksystems Es soll ein Relationales Datenbanksystem eingesetzt werden, das Open Source verfügbar ist.
Es muss für Ubuntu Server LTS verfügbar sein.
Zur Auswahl stehen zwei prominente Lösungen, MySQL und PostgreSQL. Der Vergleich betrachtet nur die Unterschiede, hauptsächlich den Zugriff auf die Systeme.

	MySQL	**PostgreSQL**
Typ	Relational	Relational
SQL	Ja	Ja
APIs	ADO.NET, JDBC, ODBC	native C Library, streaming API, ADO.NET, JDBC, ODBC
Unterstützte Programmiersprachen	C, C#, C++, Java, JavaScript(Node.js), PHP, Python, Ruby	C, C++, Java, Python
Server-seitige Scripts	Ja	benutzerdefinierte Funktionen

Tabelle 1: Vergleich MySQL, PostgreSQL. Quelle(db-engines.com/de/system/MySQL%3BPostgreSQL)

Die Auswahl ist auf MySQL gefallen, da MySQL durch die Menge der unterstützten Programmiersprachen flexibler ist.

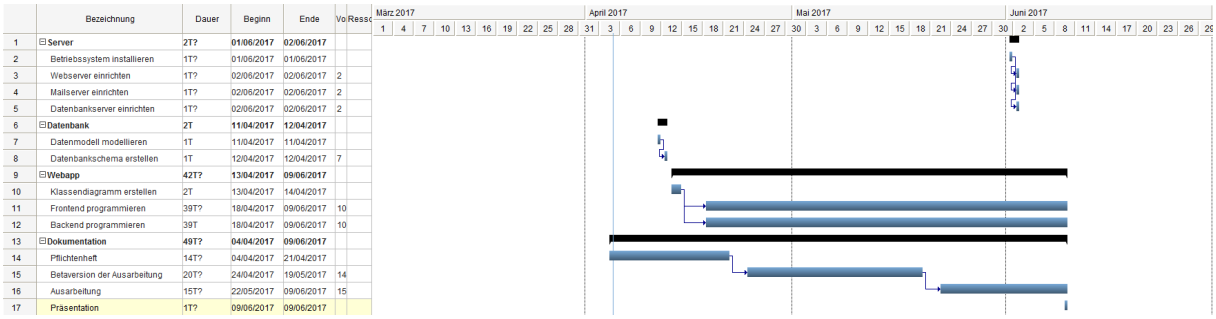
5.3 Webserver

Als Webserver soll Apache zum Einsatz kommen. Dieser bietet umfangreiche Module zur Unterstützung von Websockets, Java Servlets etc.

5.4 Mailserver

Es müssen nur e-Mails versendet werden. Hier bietet sich postfix als MTA an.

6 Zeitplan



	Bezeichnung	Dauer	Beginn	Ende	Vo	Ressc
1	Server	2T?	01/06/2017	02/06/2017		
2	Betriebssystem installieren	1T?	01/06/2017	01/06/2017		
3	Webserver einrichten	1T?	02/06/2017	02/06/2017	2	
4	Mailserver einrichten	1T?	02/06/2017	02/06/2017	2	
5	Datenbankserver einrichten	1T?	02/06/2017	02/06/2017	2	
6	Datenbank	2T	11/04/2017	12/04/2017		
7	Datenmodell modellieren	1T	11/04/2017	11/04/2017		
8	Datenbankschema erstellen	1T	12/04/2017	12/04/2017	7	
9	Webapp	42T?	13/04/2017	09/06/2017		
10	Klassendiagramm erstellen	2T	13/04/2017	14/04/2017		
11	Frontend programmieren	39T?	18/04/2017	09/06/2017	10	
12	Backend programmieren	39T	18/04/2017	09/06/2017	10	
13	Dokumentation	49T?	04/04/2017	09/06/2017		
14	Pflichtenheft	14T?	04/04/2017	21/04/2017		
15	Betaversion der Ausarbeitung	20T?	24/04/2017	19/05/2017	14	
16	Ausarbeitung	15T?	22/05/2017	09/06/2017	15	
17	Präsentation	1T?	09/06/2017	09/06/2017		

Abbildung 2: Gantt Diagramm zur Zeitplanung