

Hochschule Bochum
Fachbereich Elektrotechnik und Informatik

Arbeitsthese

Online Projektzeiterfassung und
Rechnungsverwaltung mit Ruby On Rails

Autor: Daniel Pietzsch

Datum: 17. Januar 2008

Firma: innoQ Deutschland GmbH

1. Vorwort	3
2. Ausgangssituation und Zielsetzung	4
2.1. <i>Worum geht es?</i>	4
2.2. <i>Ist-Zustand</i>	4
2.3. <i>Soll-Zustand</i>	4
2.4. <i>Gegenstand der Diplomarbeit</i>	5
3. Zielbestimmung	6
3.1. <i>Musskriterien</i>	6
3.2. <i>Wunschkriterien</i>	6
3.3. <i>Abgrenzungskriterien</i>	6
4. Systemumgebung	7
4.1. <i>Clients</i>	7
4.2. <i>Server</i>	8

1. Vorwort

Dies ist ein Arbeitstheze zu meiner bevorstehenden Diplomarbeit. Sie beschreibt die grobe Funktionsweise der zu implementierenden Anwendung sowie deren Umfeld und Motivation. Sie soll einen ersten Überblick geben und allen beteiligten Personen - wie z.B. Prüfern und Mitarbeitern - eine erste Diskussionsgrundlage bieten.

Für die hier erwähnten, eingesetzten Technologien sowie die genaue Systemumgebung sind Änderungen vorbehalten. Dieses Dokument ist in keiner Weise als Pflichtenheft o.Ä. zu verstehen.

2. Ausgangssituation und Zielsetzung

2.1. Worum geht es?

Die Grundidee und -motivation der Anwendung besteht darin, den Arbeitsablauf der Projektzeiterfassung der einzelnen Mitarbeiter sowie die darauf basierende Rechnungserstellung für Kunden der innoQ Deutschland GmbH¹ zu vereinfachen und homogener zu gestalten.

Dabei soll eine zentrale, webbasierte Anwendung in der Sprache Ruby² mit Hilfe des Web-Frameworks Ruby On Rails³ implementiert werden, die alle notwendigen Daten in einer Datenbank speichert und die notwendige Funktionalität bereit stellt.

2.2. Ist-Zustand

Z.Zt. werden die Projektzeiten der einzelnen Mitarbeitern, welche an verschiedenen Projekten bei verschiedenen Kunden arbeiten, mittels Email an jene Mitarbeiterin geschickt, welche auf dieser Basis die Rechnungen für die Kunden erstellt.

Diese Zeiten werden dann für jeden Mitarbeiter manuell in ein Lotus Notes Dokument kopiert, welches seinerseits wieder eine kleine Excel-Tabelle enthält, über welche dann die Zeiten und Leistungen zusammengerechnet werden, so dass man auf einen monatlichen Gesamtbetrag kommt. In dem Lotus Notes Dokument selber stehen sämtliche weitere Daten, wie z.B. Anschrift, Rechnungstext, Bankverbindung usw. Als Vorlage für eine Rechnung wird meistens die Rechnung vom Vormonat verwendet - sofern vorhanden.

Im Anschluss wird die Rechnung ausgedruckt und per Post an den Kunden verschickt. Der Prozess muss für jede Rechnung in dieser Form wiederholt werden.

2.3. Soll-Zustand

Ziel der Anwendung ist es, sämtliche bisherige Software und Technologien durch eine einzige, webbasierte Anwendung zu ersetzen.

¹ <http://www.innoq.com>

² <http://www.ruby-lang.org/de/>

³ <http://www.rubyonrails.org/>

Die Mitarbeiter tragen ihre Arbeitszeiten für ein Kundenprojekt direkt in der Anwendung ein. Dies kann von einem beliebigen Ort mit Internetzugang aus geschehen, und die einzige Softwarevoraussetzung soll ein aktueller Webbrowser sein. Die eingetragenen Zeiten können dann direkt für die Rechnungserstellung genutzt werden, d.h. sie werden bei Bedarf automatisch, richtig formatiert in die entsprechende Rechnung eingetragen. Die Rechnung wird auch sonst auf Grund vorhandener Stammdaten, wie z.B. Kunden- und/oder Projektdaten, erstellt, so dass nur kleinere Anpassungen notwendig sind, um die Rechnung zu erstellen.

Es soll aber auch nach wie vor die Möglichkeit gegeben sein eine Rechnung manuell erstellen zu können.

Auch für die Rechnungserstellung soll die einzige Softwarevoraussetzung ein aktueller Webbrowser sein.

2.4. Gegenstand der Diplomarbeit

Gegenstand der Diplomarbeit ist es, das Projekt zu planen und zu implementieren.

Bei der Planung wird festgestellt, welche Anforderungen die Anwendung erfüllen muss, und es wird ein grobes Modell der Datenbankstruktur festgelegt.

Bei der Implementierung steht der Umgang mit dem Framework Ruby On Rails sowie der Programmiersprache Ruby im Vordergrund. Es kommen aber auch weitere Technologien zum Einsatz, wie z.B. JavaScript⁴, CSS⁵ oder HTML⁶ usw.

⁴ <http://de.wikipedia.org/wiki/JavaScript>

⁵ http://de.wikipedia.org/wiki/Cascading_Style_Sheets

⁶ http://de.wikipedia.org/wiki/Hypertext_Markup_Language

3. Zielbestimmung

3.1. Musskriterien

- Alle Mitarbeiter müssen ihre Arbeitszeiten aus dem jeweiligen Projekt minutengenau eintragen und bearbeiten können
- Rechnungen müssen erstellt werden können
- Rechnungen müssen ausgedruckt werden können
- Sämtlicher Text in den Rechnungen muss manuell veränderbar sein
- Es gibt eine Stammdatenpflege, d.h. Daten wie z.B. Kunden- oder Projektdaten, müssen änderbar und erweiterbar sein
- Man muss eine Rechnung speichern und laden können
- Die Anwendung liefert sinnvolle Vorlagen bei der Rechnungserstellung (z.B. eine modifizierte Rechnung vom Vormonat)
- Die Anwendung trägt die Zeiten eines Mitarbeiters in die Rechnung ein, errechnet den Gesamtbetrag und weist die Umsatzsteuer aus
- Die Währung der Rechnung muss variabel sein
- Ein Rechnung muss in den Sprachen Deutsch und Englisch erstellt werden können
- Die Anwendung läuft unter dem Webbrowser Firefox⁷ ab Version 2 fehlerfrei

3.2. Wunschkriterien

- Die Anwendungen läuft auf weiteren Webbrowsern fehlerfrei
- Email-Benachrichtigungen werden gesendet, sollte ein Mitarbeiter noch keine Arbeitszeiten für den vergangenen Monat eingetragen haben.
- Erzeugung von PDF⁸-Dokumenten der Rechnungen
- Die Anwendung ist RESTful⁹

3.3. Abgrenzungskriterien

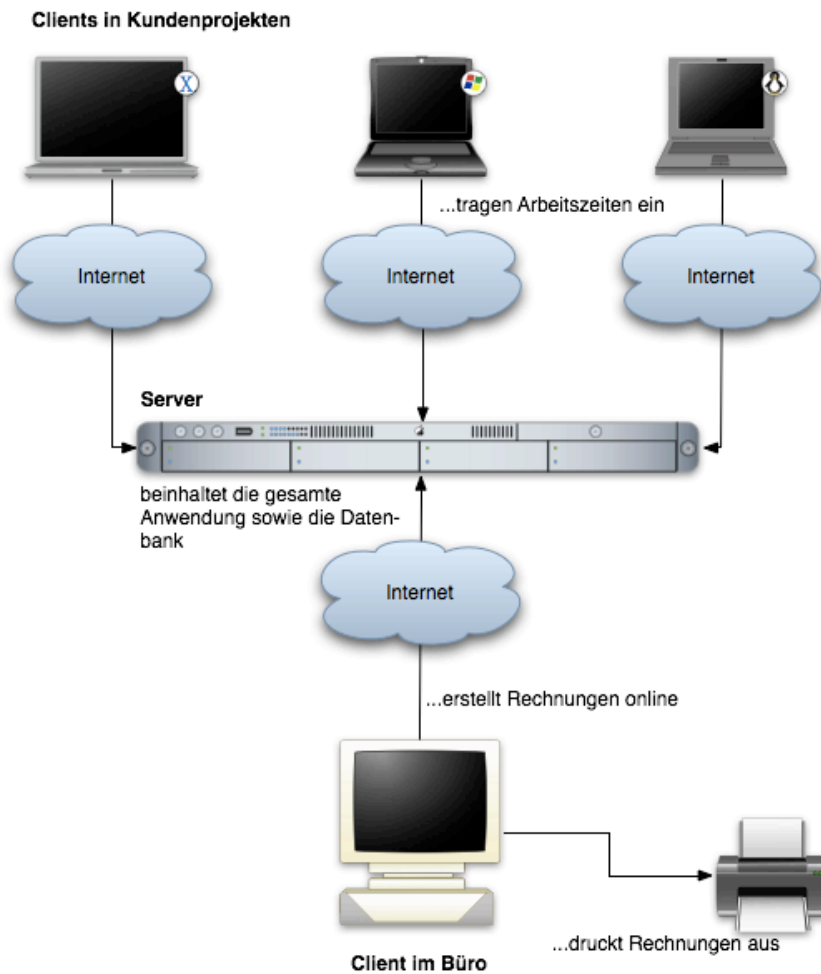
- Die Anwendung ist keine Finanz- und/oder Steuerverwaltung

⁷ <http://www.mozilla-europe.org/de/products/firefox/>

⁸ <http://www.adobe.com/de/products/acrobat/adobepdf.html>

⁹ <http://en.wikipedia.org/wiki/REST>

4. Systemumgebung



4.1. Clients

Sowohl bei den Clients in Kundenprojekten, als auch bei dem Client im Büro wird die Anwendung im Webbrowser ausgeführt. Die Browser müssen jeweils JavaScript aktiviert haben, da die Anwendung z.B. JavaScript-Frameworks wie Prototype¹⁰ und script.aculo.us¹¹ einsetzen wird um die Bedienung ansprechender zu gestalten. So werden z.B. einige Befehle mittels AJAX¹² abgesetzt, um zu vermeiden, dass Webseiten andauernd komplett neu geladen werden müssen.

Die Clients greifen jeweils über eine sichere SSL¹³-Verbindung auf den Server zu.

¹⁰ <http://www.prototypejs.org/>

¹¹ <http://script.aculo.us/>

¹² [http://de.wikipedia.org/wiki/Ajax_\(Programmierung\)](http://de.wikipedia.org/wiki/Ajax_(Programmierung))

¹³ http://de.wikipedia.org/wiki/Transport_Layer_Security

4.2. Server

Auf dem Server laufen der Apache Webserver¹⁴ und der Mongrel¹⁵ Server, welche die Anwendung bereitstellen.

Ferner läuft auf dem Server auch eine Datenbank, welche alle Daten der Anwendung beinhalten wird. Hier wird das relationale Datenbankverwaltungssystem MySQL¹⁶ eingesetzt werden.

¹⁴ <http://httpd.apache.org/>

¹⁵ <http://mongrel.rubyforge.org/>

¹⁶ <http://mysql.com/>