

fleXtime

Alexander Steinacher
Camillo Zannitti
Daniel Besmer

21. Januar 2015

Inhaltsverzeichnis

1	Einleitung	2
2	Projektidee fleXtime	2
3	Informationsmodellierung	3
3.1	Problemstellung	3
3.2	ER-Diagramm	3
4	Installation	3
4.1	Anforderungen	3
4.2	Installation	5
4.2.1	Datenbank	5
4.2.2	Web-Interface	5
5	Applikation	5
5.1	Login	5
5.2	Calendar	6
5.3	Absences	6
5.4	Team	6
5.5	Reports	6
5.6	Manage	6
5.7	Settings	7
5.8	Admin	7
6	Lessons Learned	7

1 Einleitung

Bei der Diskussion bezüglich einer Projektidee haben sich einige Punkte heraus kristallisiert, die ein Projekt neben den bestehenden Anforderungen im Rahmen dieser Vorlesung erfüllen sollte. Dabei standen zwei wichtige Anforderungen im Mittelpunkt. Erstens sollte die Web-Applikation einen praktischen Nutzen mit sich bringen, so dass sie auch nach Abschluss der Vorlesung noch weiter entwickelt und genutzt werden kann und zweitens eine möglichst spannende Herausforderung darstellen, um die Motivation und das Engagement zu fördern. Des Weiteren war man der Ansicht, dass die Daten in der Datenbank, falls möglich, von den Anwendern generiert und nicht durch einen Webcrawler oder von einer bestehenden Datenbank abgegriffen werden.

Mit den gegebenen Anforderungen ergaben sich dann mehrere mögliche Projektideen, wie beispielsweise das Überwachen bzw. Aufzeichnen von bestimmten Ereignissen (Anzahl Toiletten-Spühlvorgänge im Informatikdepartement, Besucher in der Universitäts Bibliothek, etc.), mittels über mehrere Standorte verteilte Raspberry Pi's oder die Erstellung eines ansprechenden Web-Interfaces für die Erfassung von Arbeitsstunden, welches als Ablösung einer bestehenden Lösung mit einem Excel-Sheet dienen sollte.

Da sich bei der Idee bezüglich des Aufzeichnens von Daten einen nicht zu unterschätzender Zeitaufwand in die Entwicklung der Hardware eingeschlichen hätte, hat man sich für die vielleicht zunächst etwas weniger interessante Idee zur Erfassung von Arbeitsstunden entschieden. Da zudem ein Gruppenmitglied die Applikation anschliessend verwenden könnte, schien die Idee perfekt.

2 Projektidee fleXtime

Bei fleXtime handelt es sich wie bereits erwähnt um eine Web-Applikation zur Erfassung von Arbeitsstunden. fleXtime sollte ein bestehendes Excel-Sheet zur Zeiterfassung ersetzen und den Prozess vereinfachen. Die Verwendung von Excel für solch eine Aufgabe scheint zu Beginn womöglich geeignet, doch bei näherer Betrachtung zeigen sich folgende Probleme:

1. Verwendung von Excel Makro's
2. Unübersichtlich und nicht Benutzerfreundlich
3. Keine Plattformunabhängigkeit

4. Keine Standortunabhängigkeit
5. Verwendung von proprietärer Software
6. Mangelnde **fleX**ibilität

Die Verwendung einer plattformunabhängigen Web-Applikation scheint daher einleuchtend. Eine zentrale Verwaltung dieser Daten in einer Datenbank, anstelle von einzelnen Dateien in Ordnern, bietet zudem die Möglichkeit den entstehenden Datensatz für eine zukünftige Anwendung flexibler zu verwalten, wobei in erster Linie das Tool eher für den Nutzer gedacht ist und nicht zur Überwachung der Mitarbeiter.

3 Informationsmodellierung

3.1 Problemstellung

Ein Mitarbeiter sollte die Möglichkeit haben seine geleisteten Arbeitsstunden durch Angabe von Anfangs- und Endzeit zu erfassen. Pro Tag können dabei mehrere dieser Intervalle entstehen (Mittagspause, etc). Zudem soll es möglich sein, sich gegebenenfalls Notizen zu einem bestimmten Tag zu machen. Des Weiteren soll die Eingabe von Absenzen (Krankheit, Ferien, etc.) realisierbar sein. Diese können von einem Teamleiter entweder bewilligt oder abgelehnt werden. Durch Angabe eines Beschäftigungsgrads und eines Zeitplans werden Soll- und Ist-Arbeitszeit angezeigt. Eine Team-Ansicht soll allen Mitarbeitern einen Überblick verschaffen, wer zurzeit abwesend (Ferien, Mutterschaftsurlaub, etc.) ist. Die erfassten Daten sollen durch informative Grafiken für den Benutzer repräsentiert werden, um allfällige Trends (geleistete Überstunden nehmen zu/ab?, späterer Arbeitsbeginn am Montagmorgen?, etc.) zu erkennen.

3.2 ER-Diagramm

Das erstellte ER-Diagramm zur Problemstellung ist in Abb. 1 gegeben.

4 Installation

4.1 Anforderungen

Für die Verwendung von fleXtime ist ein Host mit einer MySQL-Datenbank (Version ≥ 5.6) und ein Web-Server mit Apache (Version ≥ 2.2) und PHP

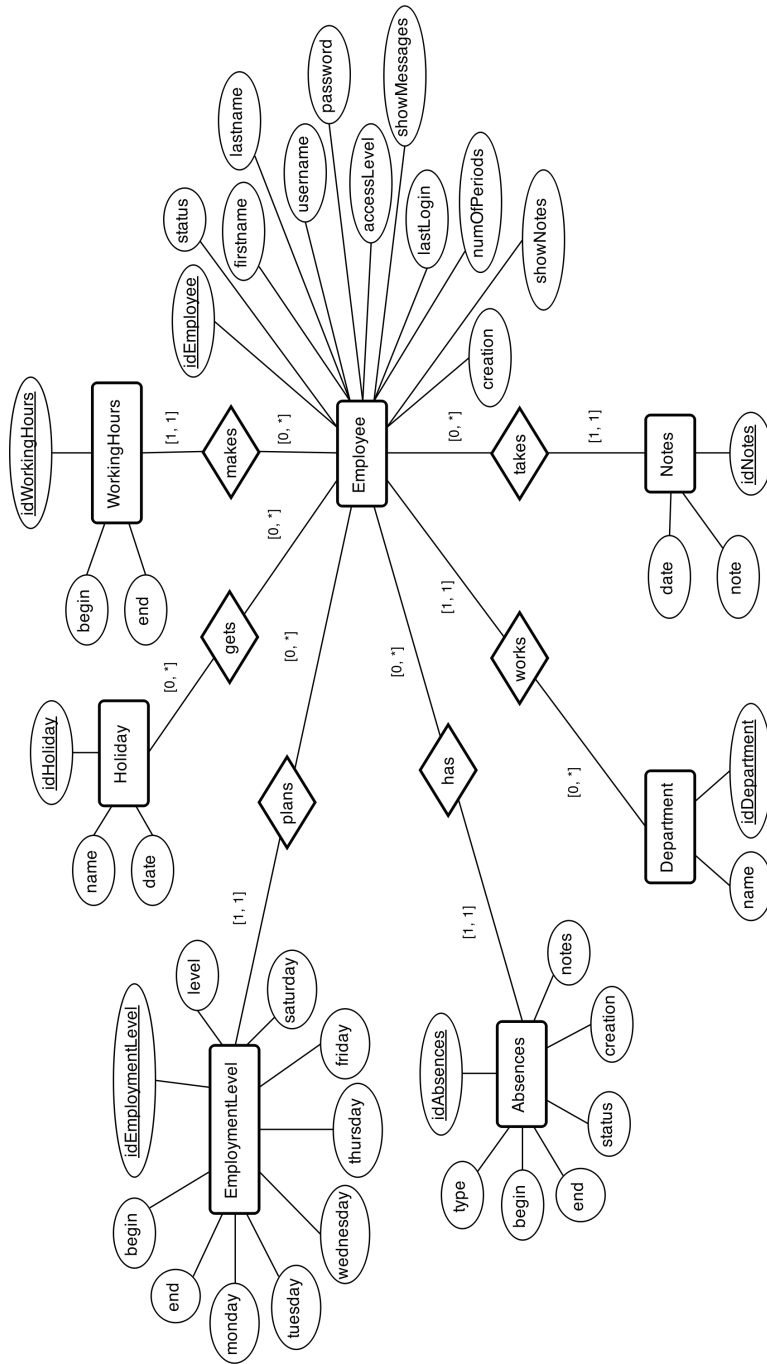


Abbildung 1: fleXtime ER-Diagramm

(Version ≥ 5.4) nötig. Als Client dürfte ein beliebiger Browser in aktueller Version dienen (getestet wurde unter Google Chrome).

4.2 Installation

Die Installation von flextime gliedert sich in zwei Schritte. Zu Beginn wird eine Datenbank mit entsprechenden Schemas erstellt und anschliessend kann das Web-Interface installiert werden.

4.2.1 Datenbank

Zur Erstellung der Datenbank sollte das SQL-Skript `install.sql` ausgeführt werden. Dabei werden alle Relationen gelegt und die Trigger und Funktionen definiert. Zudem wird ein erster User mit Administrator-Rechten (username: *admin*, password: *admin*) erstellt. Beispiel-Daten können durch die Ausführung des Skripts `data.sql` erstellt werden. Zur Verwendung der Datenbank durch das Web-Interface sollte zudem ein neuer Datenbank-Benutzer (z.B. flextime) mit entsprechenden Rechten erstellt werden.

4.2.2 Web-Interface

Zur Installation des Web-Interfaces sollte der Inhalt des Ordners `flextime` in den Web-Root-Folder des Apache-Servers kopiert werden. Dabei ist zu beachten, dass der Ordner eine versteckte `.htaccess` Datei enthält. Anschliessend muss die Datei `flextime/config/config.php` angepasst werden. Dabei sind die Angaben über den zuvor erstellten Datenbank-Benutzer, sowie die Datenbank und den Root-Ordner zu ändern. Zuletzt sollte noch die *RewriteBase* in der Datei `flextime/.htaccess` geändert werden, um das URL-Rewriting zu aktivieren.

5 Applikation

5.1 Login

Die Web-Applikation ist nur durch ein Login-Fenster erreichbar. Beim Login werden die Benutzerrechte (Admin, Manager, User) des Benutzers ermittelt und entsprechende Tabs ein- bzw. ausgeblendet (ein Manager sieht zusätzlich den Manage-Tab und ein Administrator dazu noch den Admin-Tab). Unter jedem Tab wird überprüft ob der Benutzer tatsächlich die benötigten Rechte

verfügt, um ein "Hacken" mittels Angabe des spezifischen URLs zu verhindern. Ist ein Benutzer 30 min nicht aktiv, läuft die Session automatisch ab und er muss sich wieder authentifizieren.

5.2 Calendar

Arbeitszeiten und Notizen können erfasst, verändert und gelöscht werden. Die Angaben werden mittels Klick auf "SSave" bestätigt. Die Datenbank überprüft die Eingaben standartmässig. Nicht zulässig sind überlappende oder negative Zeitintervalle. Bei Angabe eines negativen Intervalls wird der Benutzer durch Einfärben des Eingabefelds gewarnt. Mittels kleiner Navigations-Pfeile kann eine Vor- oder Zurück gesprungen werden. Alternativ kann eine Woche auch mittels Eingabe im URL direkt angewählt werden.

5.3 Absences

Vom Benutzer erfasste Absenzen werden automatisch angezeigt. Zudem hat der Nutzer die Möglichkeit eine bereits erfasste Absenz zu modifizieren bzw. eine Neue zu erstellen. Falls der Benutzer die Absenz löschen möchte, kann er dies im Edit-Dialog bewerkstelligen. Bereits akzeptierte Absenzen nur noch gelöscht und nicht mehr abgeändert werden.

5.4 Team

Diese Ansicht zeigt eine Jahres-Übersicht über die erfassten und bewilligten Absenzen. Ein blauer Balken indiziert jeweils das aktuelle Datum.

5.5 Reports

Die Arbeitsstunden und Absenzen werden in drei unterschiedlichen Views (Week, Month, Year) dargestellt. Entsprechende Diagramme zur besseren Visualisierung der Informationen sind eingebettet.

5.6 Manage

Der Teamleader/Manager hat die Möglichkeit beantragte Absenzen zu avisieren bzw. nicht zu genehmigen. Bereits bearbeitete Absenzen werden ausgeblendet, können aber durch ein Select-Button angezeigt werden.

5.7 Settings

Unter diesem Tab kann der Benutzer grundlegende Einstellungen wie das Anzeigen von Notizen bzw. Anzahl der Intervalle in der Calendar-View vornehmen. Das Ändern des eigenen Passworts und das Erfassen des Beschäftigungsgrads und des Zeitplans kann hier bewerkstelligt werden.

5.8 Admin

Der Administrator kann Feiertage, Benutzer und Departemente erstellen, auslesen, editieren und (löschen). Benutzer werden nicht gelöscht sondern als Deaktiviert vermerkt.

6 Lessons Learned

Nach dem Bearbeiten des Projekts stellt sich zwangsläufig die Frage, was wir dabei alles gelernt haben. Nachdem wir beim Programmierprojekt im Rahmen der Vorlesung CS108 einen recht ausführlichen Projektplan erstellen mussten, der eher geringfügig Anwendung in der Ausführung fand, erschien uns dies als ein bisschen viel Planung. Wahrscheinlich haben wir gerade deswegen bei diesem Projekt zu wenig auf die Planung gesetzt, was manchmal zu kleineren Problemen führte und aufzeigte, wie wichtig eine gute Planung und Kommunikation ist. Eine weitere wichtige Lektion, die wir gelernt haben, ist sicherlich die Bedeutsamkeit der Kommunikation innerhalb der Gruppe. Kommunikationsmittel wie Chats, Emails, etc. stellen sicher eine gute Möglichkeit der Informationsvermittlung dar, doch hat es sich gezeigt, dass der direkte Kontakt oft mehr von Nöten ist. Denn es scheint als sei die unmittelbare Interaktion ein ganz wichtiger Aspekt bei der Entwicklung. Des Weiteren half das Bearbeiten des Projekts massgeblich beim Verinnerlichen der in der Vorlesung besprochenen Inhalte, sei dies nun im Datenbank Teil oder im Web-Technologien Teil.