

Proposal: Reimplementation einer Webapplikation für Schulen

Pascal Ernst

70302367

Ostfalia Hochschule für angewandte Wissenschaften

Januar 2015

1 Ursprüngliche Implementation

Geschichte Vor einigen Jahren hat die Informatik-AG des Lessing-Gymnasiums Uelzen eine neue Webapplikation erstellt, um den Ablauf der Essensbestellung in der dortigen Cafeteria zu erleichtern.

Entstanden ist das Bargeldlose Bestellsystem für die Legeria, oder kurz "BaBeSK". Es erlaubt es Schülern, online über einen Webbrowser Mahlzeiten bereits im voraus zu bestellen. Jeder Schüler hat eine RFID-Karte bei sich, die ihn identifiziert. Die Cafeteria kann dann für den jeweiligen Schüler die Bestellungen einsehen und herausgeben.

Für dieses System habe Ich über die Zeit weitere Funktionalitäten, in Babesk Module genannt, geschrieben und verbreitet. Darunter befinden sich "KuWaSys", ein Kurswahlsystem, oder auch "ElaWa", ein Modul für Elternsprechtagswahlen. Diese Module werden von 3 weiteren Schulen momentan benutzt.

Lösungen Dieses System benutzt intern PHP 5 & MySQL. Es ist über die Zeit gewachsen, hat viele strukturelle Veränderungen durchlaufen und ist mit der Zeit selber zu einem Framework geworden. In dem Projekt werden keine automatisierten Tests benutzt. Die Sicherheit des Web-Interfaces ist gering und veraltet. So sind zum Beispiel CSRF- (dt.: Webseiten-übergreifende Anfragefälschung) und vom Backend sogar SQL-Injection-Attacken möglich.

Um dagegen vorzugehen, kann Ich:

- Das jetzige Programm refactoren und das manuell gebaute Framework pflegen
- Ein existierendes, weit verbreitetes und gepflegtes PHP-Framework integrieren
- Das Programm von Grund auf neu aufsetzen

Die erste Option fällt für mich klar weg. Im Lauf der Jahre kam es immer wieder zu unerwarteten Problemen mit dem Framework; So war es schwierig, von HTTP auf das verschlüsselte HTTPS-Protokoll umzusteigen, da vom Framework generierte AJAX-Urls von modernen Browsern blockiert wurden. Es gibt inzwischen genug existierende Open-Source Web-Framework-Lösungen, die die vorgesehene Rolle des Babesk-Frameworks besser füllen.

Die Wahl zwischen der zweiten und dritten Option fällt mir schwieriger. Entweder Ich bleibe beim bisher benutzten LAMP-Technologie-Stack (Linux, Apache, MySQL, PHP) und kann Teile des Programmes (Datenbank, Apache-Konfiguration) weiter benutzen, oder wähle einen komplett neuen Stack. Dieser müsste grundlegend besser als der LAMP-Stack sein und mir ermöglichen, schnell auf den jetzigen Stand des Stacks zu kommen.

Im Endeffekt habe Ich mich für Ruby on Rails als Alternativ-Stack entschieden. Es ist sehr weit verbreitet, wird gut unterstützt, integriert automatisierte Tests und ist

bekannt dafür, dass es dem Programmierer eine schnelle Implementierung von Funktionen erlaubt.

2 Ziel

Im Rahmen des IT-Projekts will Ich die grundlegende Funktionalität, die von allen Modulen genutzt werden, implementieren. Dies beinhaltet:

- Benutzer
 - Benutzer sollen sich einloggen können
 - Es können Rollen an Benutzer verteilt werden, um deren Rechte zu bestimmen
 - CRUD (Anlegen, Anzeigen, Verändern, Löschen von Benutzern)
- Schuljahre
 - CRUD
- Schulklassen
 - CRUD
- Frontend
 - Grundlegendes Design
 - Log In nötig
 - Für normale Benutzer (zb Schüler) um sich an Kurse anzumelden, Bestellungen auszuwählen, Account-Einstellungen usw
- Backend
 - Grundlegendes Design
 - Über das Frontend mit ausreichenden Rechten erreichbar
 - Erlaubt, alle Aspekte des Programmes einzustellen
- Tests
 - Aspekte des Systems werden mithilfe von automatisierten Tests (vor allem Integrationstests) vor Regressionen geschützt.

Außerdem will Ich als Beispiel-Modul das Modul Elawa (Elternsprechtagwahlen) auf einen ähnlich weit implementierten Stand wie im Babesk bringen. Das beinhaltet:

- Backend
 - Lehrerverwaltung

- Sprechstundenverwaltung
- Frontend
 - Anmeldung von Schülern an Sprechstunden
 - Ansicht der Sprechstunden für Lehrer

3 Projektplan

Aufgrund einer länger andauernden Erkrankung ist der bisherige Projektplan, der in dem Scrum-Tool eingetragen worden war, hinfällig geworden. Die Benutzerverwaltung, Login und Rechteverteilung sind im Dezember bereits implementiert worden. Das Backend-Interface wurde bereits grundlegend erstellt.

Da das Modul Elawa bereits Mitte Juni benutzt werden soll, will Ich mit dem grundlegenden Programm und dem Modul Ende Mai fertig werden.

Die Sprints sind jeweils in 2 Wochen aufgeteilt. Der grundlegende Plan sieht wie folgt aus:

- Sprint 1608:
 - CRUD Schulklassen
- Sprint 1610:
 - Frontend Design
 - Frontend Benutzereinstellungen (Passwort ändern, Email ändern)
- Sprint 1612:
 - Für das Backend relevante Funktionen des Moduls Elawa
- Sprint 1614:
 - Für das Frontend relevante Funktionen des Moduls Elawa
- Sprint 1616:
 - Besprechung mit dem Kunden; Abschließende Änderungen.

Abschließend zu Sprint 16-8 will Ich mit dem Kunden den Stand der neuen Implementierung besprechen und mit Babesk vergleichen.