



**School of
Engineering**

InIT Institut für angewandte
Informationstechnologie

Projektarbeit (Informatik)

Bootstrap A Web App

Autoren

Reto Canova
Philipp Schürch

Hauptbetreuung

Mark Cieliebak

Datum

19.12.2014

Zusammenfassung

Aufstrebende Web-Entwickler besitzen oft gute Ideen für neue Web-Applikationen. Viele von Ihnen gründen Startup-Unternehmen und wollen ihre Idee mithilfe ihres Unternehmens im Web realisieren. Die Arbeit, die jedoch getan werden muss, bevor ein Entwickler anfangen kann seine eigentliche Applikation zu programmieren ist heutzutage sehr umfangreich.

Jede Web-Applikation besitzt diverse Komponenten, die vom Entwickler zuerst installiert und konfiguriert werden müssen, was den Entwickler enorm viel Zeit kostet, bevor er produktiven Code für seine Applikation schreiben kann.

Diese Arbeit stellt ein Konzept vor, welches als Ziel hat, den Initialaufwand für Entwickler zu minimieren. Anstatt mehrere Tage damit zu verbringen die Entwicklungsumgebung einzurichten, die richtige Front- und Backend Sprache auszuwählen und alles so zu konfigurieren, dass die einzelnen Komponenten untereinander kompatibel sind, soll ein Technologie-Stack definiert werden, dem es dem Entwickler erlaubt, direkt mit dem Coden seines Projektes anzufangen.

In einer ersten Phase wurden verschiedene Typen von Web-Applikationen und Benutzergruppen analysiert, um daraus auf die notwendigen Komponenten einer Web App schliessen zu können. Anschliessend wurden die Komponenten ausgewählt, welche für das schnelle entwickeln eines Prototypen benötigt werden. Die zweite Phase bestand darin, dass Software von unterschiedlichen Herstellern verglichen wurde um daraus zu evaluieren, welche Software sich am besten für die genannte Problemstellung eignen würde.

In einem letzten Schritt wurde eine Virtual Machine mit den auserwählten Tools aufgesetzt, die es erlauben sollte, eine Web-Applikation direkt aus dieser virtuellen Maschine ohne irgendwelchen Initialaufwand entwickeln zu können. Mithilfe des ausgewählten Technologie-Stacks wurde ein Prototyp von der Beispielapplikation „myMoneyPenny“ entwickelt, welcher die elementaren Funktionen einer Web-Applikation beinhaltet.

Abstract

Up-and-coming Web-Developers often have good ideas for new Web-Applications. Many of them realise their idea by founding their own company. The work that has to be done before a developer can actually start programming his application is pretty extensive. Every Web-Application consists of different components that have to be installed and configured first, which can cost the developer a lot of time before he can start writing productive code for his application.

This paper presents a concept, which tries to minimize the initial work a developer has to do. Instead of investing several days into finding the right front- and backend language, set up their development environment and configure the single components to work together, a technology stack is defined which allows the developer to directly start writing code for his application.

In a first phase different types of web-applications were analyzed to assess its necessary components. Subsequently a subset of these components which are necessary for developing a quick prototype was chosen. The second phase consisted of comparing software solutions of different manufacturers to evaluate which of them are suitable for the aforementioned problem.

The last step consisted of setting up a virtual machine with the chosen set of tools. This should allow the developer to start the design process directly, thus reducing the initial expense. With the help of the mentioned technology-stack a prototype of the application „myMoneyPenny“ was developed, which contains the elemental functionalities of a web-application.

Einleitung

Ausgangsaufgabenstellung

Junge Ingenieure haben oft spannende Ideen für neue und innovative Produkte und aus diesen Ideen sind schon viele erfolgreiche Firmen entstanden. Dabei geht es sehr oft um Web- und Mobile-Applikationen.

Uns ist aufgefallen, dass der initiale Aufwand für solche Apps sehr hoch ist: man muss eine Sprache für Frontend und Backend auswählen, IDE und CI-Server einrichten etc. Mit diesen Vorarbeiten kann man locker mehrere Tage verbringen, ohne eine einzige Zeile produktiven Code zu schreiben.

Die spannende Frage ist deswegen, ob man für ein "normales" Web-Startup eine vollständige IT-Umgebung out-of-the-box zur Verfügung stellen kann so dass man sich von Anfang an vollständig auf das Produkt konzentrieren kann. Was dabei "normal" ist, welche Tools es braucht, und in welcher Form man solch eine Umgebung bereitstellen kann, soll in dieser Arbeit untersucht werden. Die Aufgabe besteht aus zwei Teilen:

1. Aufbau einer IT-Umgebung, die für ein "typisches" Web-Startup alle benötigten Technologien bereitstellt. Dies umfasst Programmiersprache(n), Frameworks, Code Guidelines, etc. Für all diese Themen gibt es etablierte Technologien das Ziel ist hier eine sinnvolle Auswahl zu treffen und diese in geeigneter Form (z.B. eine Virtuelle Maschine) bereitzustellen.
2. Anwendung anhand einer Beispiel-Applikation. Sie evaluieren am Beispiel von einem konkreten Produkt, wie gut ihre IT-Umgebung funktioniert. Dazu entwickeln Sie einen ersten Prototyp von "myMoneypenny": eine Web/Mobile-App, die Führungspersonen in ihren täglichen Aufgaben als Vorgesetzter unterstützen soll. Ein externer Wirtschaftspartner wird anschliessend aus ihrem Prototyp ein kommerzielles Produkt machen.

Vorwort

Web-Applikationen sind heute ein fester Bestandteil des Web 2.0 und entwickeln sich laufend weiter. Die Webseite www.infoworld.com ist die Weiterführung der Zeitschrift InfoWorld (früher: The Intelligent Machines Journal), nachdem ihre letzte Printausgabe im Jahre 2007 erschien. Die Webseite bezeichnet sich selber als „*Information Technology Online Media Business*“ und veröffentlicht Analysen und Berichte von erfahrenen Journalisten und Leuten aus der IT-Wirtschaft, welche sich an professionelle IT-Ingenieure und technisch versierte Leute aus dem Wirtschaftssektor richten.

In einem Artikel vom Januar 2014 [1] werden auf dieser Webseite die Trends der Web-Entwicklung im kommenden Jahr 2014 vorgestellt. Angesagt laut InfoWorld sind im Jahre 2014 diverse Frameworks, Mobile Web-Applikationen und Single-Page Web-Applikationen, während Webseiten und native Mobile-Apps an Popularität verlieren.

Entwickler, welche in absehbarer Zukunft eine Web-Applikation programmieren wollen stehen vor einer sehr grossen Auswahl an Tools und Frameworks, welche designt wurden, um die Entwicklung einer solchen Applikation zu vereinfachen. Jedoch haben selbst erfahrene Entwickler zum Teil Mühe für sich das passende Tool zu finden oder benötigen viel Zeit, um all diese Tools zu installieren.

Diese Arbeit richtet sich an ebensolche Entwickler und fungiert als Erfahrungsbericht von zwei Studenten, die eine solche Web-Applikation programmiert haben. Sie hat als Ziel einen Technologie-Stack zu definieren, der Entwicklern helfen soll den Aufwand für das Entwickeln ihrer Web-Applikation zu reduzieren.

INHALTSVERZEICHNIS

BOOTSTRAP A WEB APP	7
BEREITS EXISTIERENDE TECHNOLOGIE-STACKS	7
DREI BENUTZERGRUPPEN EINER WEB-APP	8
BENUTZER 1: DER AUFSTEIGER	8
BENUTZER 2: DER AUSSTEIGER	9
BENUTZER 3: DER AKADEMIKER	9
FAZIT DER BENUTZERSTECKBRIEFE	10
VERSCHIEDENE TYPEN VON WEB-APPLIKATIONEN	10
ELEMENTE EINER WEBAPPLIKATION	11
PLANUNG DER ENTWICKLUNG	12
AUFSETZEN DER SOFTWARE	13
DIE WAHL DER ENTWICKLUNGSUMGEBUNG	13
LOKAL VS. CLOUD	13
NETBEANS VS. ECLIPSE	13
GIT VS. SVN	13
CLOUD9	14
UNSERE WAHL DER ENTWICKLUNGSUMGEBUNG	14
ANNO 2014: WELCHE TECHNOLOGIEN LIEGEND BEI STARTUPS IM TREND	15
FRONTENDTECHNOLOGIEN	16
WELCHES FRONT-END FRAMEWORK FÜR UNS?	17
BOOTSTRAP	17
FOUNDATION	17
FRONTEND FAZIT	17
BACKENDTECHNOLOGIEN	18
JAVA SERVER FACES	19
NODEJS	19
RUBY ON RAILS	19
UNSERE BACKEND WAHL	19
DATENBANKTECHNOLOGIEN	20
WELCHE DATENBANK IST DIE RICHTIGE?	20
MYSQL VS. POSTGRESQL	20
MONGODB	21
UNSERE DATENBANK WAHL	21
JERSEY	22
RESTLET	22
API FRAMEWORK FAZIT	22
UNSERE (VIRTUELLE) ENTWICKLERMASCHINE	23
GRUNDSPEZIFIKATIONEN DER VIRTUELLEN MASCHINE	23
INSTALLATION DES AUSGEWÄHLTEN TECH-STACKS	23
DAS REFERENZPROJEKT MYMONEYPENNY	24
PROJEKTSTRUKTUR	24
FRONTEND	25
DATENBANK	26
USER MANAGEMENT	27
SECURITY	28
FRAGEBOGEN	29
SCHLUSSWORT & AUSBLICK	30
ABBILDUNGSVERZEICHNIS	31
QUELLENVERZEICHNIS	32

Bootstrap A Web App

Bereits existierende Technologie-Stacks

Schon bereits heute existieren vorgefertigte Software-Zusammenstellungen, die beim Entwickeln von Web-Applikationen helfen sollen. Um einen ersten Eindruck von den benötigten Komponenten für das Entwickeln einer Web-Applikation zu bekommen, haben wir verschiedene Software-Bundles analysiert, die eine Möglichkeit bieten, den Konfigurationsaufwand für das Entwickeln in einem lokalen Rahmen möglichst gering zu halten.

XAMPP

Ein bekanntes Entwicklertoolset für Webapplikationen ist XAMPP. Es ist eine leicht zu installierende Apache Distribution und beinhaltet alles, was für das Entwickeln von Webapplikationen von Nöten ist. Ausserdem bietet es ein Angebot von nachträglich zu installierenden Zusatzmodulen. XAMPP überzeugt mit einer sehr einfachen Einrichtung (simple Installer-Runtime) und einer entsprechenden Vorkonfiguration, die ein schnelles Einsteigen in die Entwicklung von PHP oder Java-basierten Webapplikationen erlaubt. Auf Knopfdruck kann der MySQL oder Apache Service gestartet werden und man kann mit dem Entwickeln beginnen.

Bitnami

Bei der Suche nach bereits vorhandenen Tech-Stacks für die Entwicklung von Webapplikationen stießen wir auf einen Anbieter namens „Bitnami“, welcher sich zur Aufgabe gemacht hat, über Cloud-Hosting in sich geschlossene, teils kostenlose Software-Bundles zu vertreiben. Diese können entweder als Installer, vorinstalliert in einer Virtual Machine oder als Cloud Image bezogen werden. Die Cloud Images können zurzeit entweder in der Amazon Elastic Compute Cloud oder in Microsofts Windows Azure Plattform installiert werden.

Erwähnenswert ist hierbei das „DevPack“, eine Kompilation aus verschiedenen Runtimes und Frameworks um mit dem entwickeln von Webapplikationen zu beginnen. Darunter fallen PHP, Java, Ruby On Rails und Django sowie Apache Tomcat als Webserver und MySQL oder PostgreSQL als Datenbank-Backend.

Drei Benutzergruppen einer Web-App

Um eine bessere Übersicht über die Anforderungen einer Web-App zu geben wurden folgende 3 Benutzerkonzepte aufgestellt:

Benutzer 1: Der Aufsteiger

Altersbereich	20-25 Jahre
Ausbildung	Studium / Lehre
Internetaffinität	Sehr hoch, quasi mit dem Netz aufgewachsen, versiert im Umgang damit und kennt keine Berührungängste. Starker Googler, findet daher benötigte Informationen beinahe immer selbständig.
Finanzielle Mittel	Meist gering, vom seinem Alter abhängig. Fokussiert sich beim Umsetzen eher auf kostenlose / open-source Varianten von professioneller Software da ihm ein generell kleines Budget zur Verfügung steht.
Einflussfaktoren	Ist sehr abhängig von seinem Bekanntenkreis und hat ein eher freundschaftliches Verhältnis zu Teammitgliedern.
Motivation	Hat eine tolle Idee für eine Webplattform oder einen eShop und möchte diese gerne mit Freunden lancieren.

Das erste Benutzerprofil entspricht wohl am ehesten unserem eigenen und war darum auch recht schnell erstellt.

Dieser Benutzer besitzt viel Zeit die er in die Entwicklung seiner Idee stecken kann aber hat nur sehr beschränkte finanzielle Mittel weil er meistens noch nicht in einem Unternehmen arbeitet, welches ihm das für die Entwicklung mit kostenpflichtiger Software etc. benötigte Budget stellen könnte. Er ist technisch noch nicht auf dem Zenit seiner Fähigkeiten angelangt aber kennt sich gut genug aus, um seine Applikation, meistens Plattformen oder Online-Shops, selbstständig entwickeln zu können.

Benutzer 2: Der Aussteiger

Altersbereich	30+ Jahre
Ausbildung / Know-How	Starke, jedoch sehr auf seinen Bereich fokussierte Kenntnisse aus dem professionellen IT-Umfeld. Kein Allrounder
Internetaffinität	Mittel bis hoch, kommt nicht unbedingt aus einer Web-basierten Umgebung, kennt sich jedoch mit dem Web genug gut aus um darin zurechtzukommen.
Finanzielle Mittel	Hoch. Kann sich auf den Ausstieg finanziell vorbereiten.
Einflussfaktoren	Hat meistens eine Familie und trägt ungern Risiken.
Motivation	Fühlt sich entweder berufen etwas Eigenes auf die Beine zu stellen oder sieht eine einmalige Gelegenheit sich von seinem Angestelltenverhältnis zu lösen.

Der Aussteiger trägt ungern viel Verantwortung, da seine Familie und dessen finanzielle Sicherheit für ihn oftmals im Vordergrund steht. Er ist zwar meistens auf einen Ausstieg vorbereitet, muss jedoch schnell wieder Anschluss finden, da er über ein limitiertes Budget verfügt und ist darum abhängig von der Fertigstellungsgeschwindigkeit seiner Applikation. An den Aussteiger können die höchsten technischen Ansprüche gestellt werden, da dieser schon einige Arbeitserfahrung mit sich bringt und die Web-Applikation nicht sein erstes Projekt sein wird.

Benutzer 3: Der Akademiker

Altersbereich	30+ Jahre
Ausbildung	mindestens einen Master In Computer Science
Internetaffinität	Sehr hoch, starke Recherchier- und Programmierkenntnisse
Finanzielle Mittel	Hoch, Möglichkeit Teile des Projekts als Forschungsprojekt durchzuführen
Einflussfaktoren	-
Motivation	Resultate aus Forschungsprojekten wirtschaftlich umsetzen

Der Akademiker ist erfahren im Umgang mit dem Web. Er steht unter mässigem Zeitdruck, da er zum Schluss seines Projekts meistens eine vollumfängliche Applikation abliefern muss, jedoch bei bestimmten Entscheidungen Risiken eingehen kann. Er kann sich allerdings Teile seiner Arbeit von seinem jeweiligen Institut als Forschungsprojekt finanzieren lassen und hat dadurch wenig finanziellen Druck. Der technische Anspruch an einen Akademiker ist in vielen Fällen sehr hoch, seine Arbeitsgebiete fallen meistens in den Bereich von serverseitigen Analyse-Tools, Big Data Applikationen und Machine-Learning Projekte.

Fazit der Benutzersteckbriefe

Die gezeigten Benutzersteckbriefe lassen erkennen, dass auffällig häufig eine Situation vorherrscht, bei der es wichtig ist, dass ein erster Prototyp möglichst schnell ins Web gelangt. Basierend auf dieser Erkenntnis scheint der Fokus auf das schnelle Entwickeln eines Prototypen angebracht.

Verschiedene Typen von Web-Applikationen

Das World Wide Web bietet eine unzählige Menge an Plattformen und Applikationen. Für diese Arbeit wurden vier verschiedene Typen von Web-Applikationen kurz analysiert und zusammengetragen, um die gemeinsamen Elemente herauszukristallisieren, welche anschliessend in einer Grafik dargestellt werden.

Typ 1: Medienhosting-Applikationen

Dieser Typ von Applikation bietet Usern eine Plattform um eigene Inhalte hochzuladen, von anderen Usern hochgeladene Medien abzurufen und oftmals auch zu bewerten. Solche Plattformen zeichnen sich durch Ihre grossen Userzahlen und ihrem dementsprechenden Traffic aus. Die bekanntesten Vertreter dieser Kategorie sind **Youtube, MyVideo, DailyMotion, Snapfish, Flickr** oder **Imgur**.

Typ 2: Soziale Netzwerke

Social Networks sind eines der neueren Phänomene des Web 2.0, welche sich hoher Beliebtheit erfreuen. Die Möglichkeit, Lebensinhalte untereinander austauschen zu können, stösst auf viel Bedarf bei Benutzern von solchen Plattformen. Beispiele für solche Applikationen sind **Facebook, Twitter, Myspace** und in einer Form auch **Xing** oder **LinkedIn**.

Typ 3: Webshops

Onlineshops finden heute ihren Nutzen in nahezu jeder Wirtschaftsbranche. Das Einkaufen im World Wide Web ist dank einfachen Bezahlungsmöglichkeiten und der Zeitgewinnung durch den Benutzer heutzutage so gefragt wie noch nie. Beinahe jeder moderne Onlineshop basiert auf einer Web-Applikation, die bekanntesten darunter sind **Amazon** oder **Zalando**.

Typ 4: Desktop Software als Web-Applikationen

Dies ist die am schnellsten wachsende Gruppe von Web-Applikationen. Der Fakt, nur noch einen Web-Browser zu benötigen, um auf Software zurückgreifen zu können, finden bei vielen Computerusern hohen Anklang. Die Software muss nicht mehr gekauft werden und man ist nicht an seine Arbeitsmaschine gebunden, sondern kann immer und überall an seinen Projekten arbeiten. Die Bandbreite solcher Applikationen reicht von Mail-Clients über Bürosoftware bis hin zu Computerspielen, die im Browser gespielt werden können. Beispiele für diesen Typ sind **Google Mail, Google Docs, Office365** oder **Mint**.

Elemente einer Webapplikation

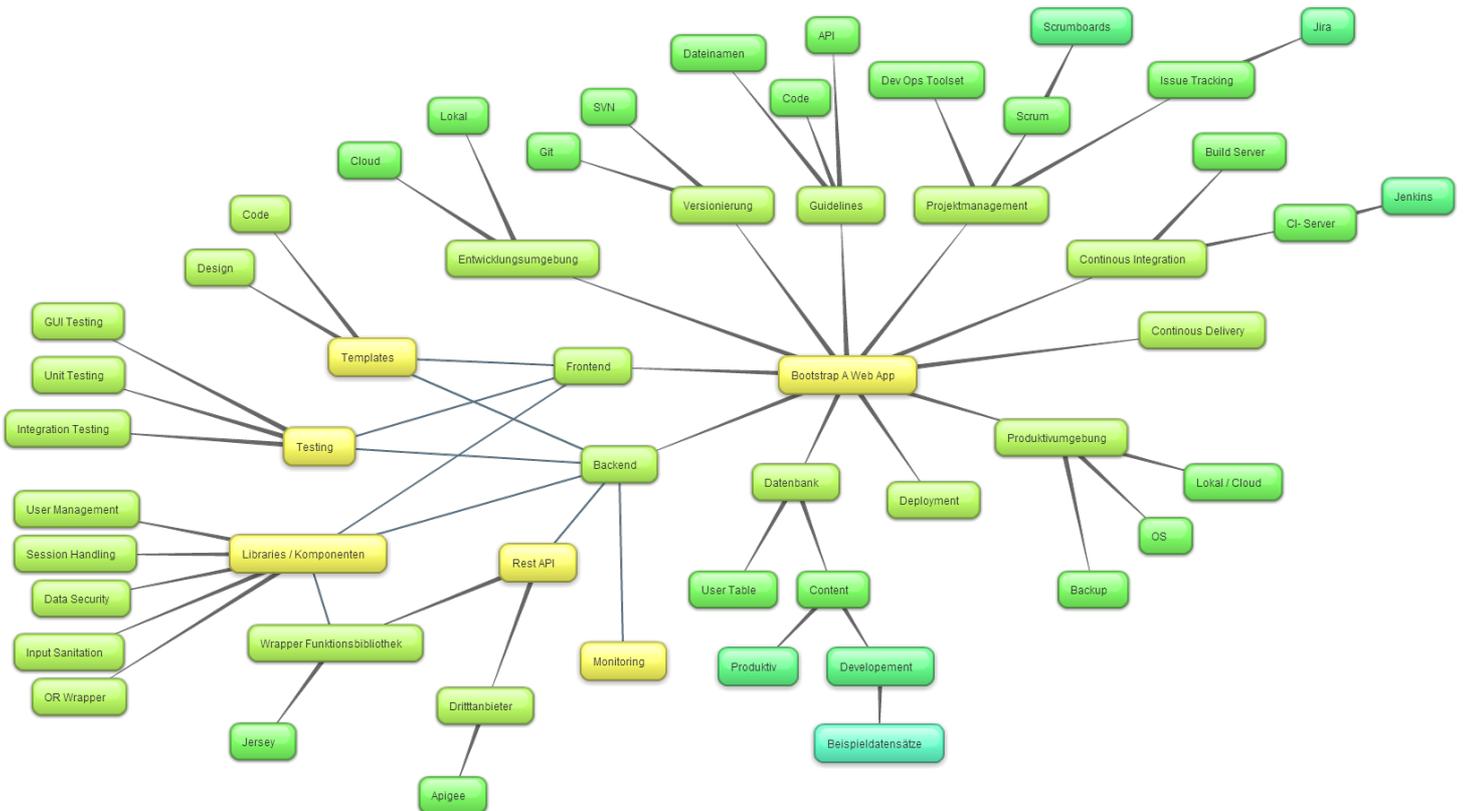


Abbildung 1: Elemente einer Webapplikation

Basierend auf der Auswertung der Funktionalitäten der oben aufgeführten Web-Applikationen ergibt sich folgendes Mindmap, welches Komponenten, die einen Grossteil aller Web-Applikationen gemeinsam haben, aufführt.

Nachfolgend folgt ein Vergleich von Technologien, die für das Entwickeln eines frühen Prototyps benötigt werden, nämlich Frontend- Backend- und Datenbanktechnologien, im obigen Mindmap in der linken, unteren Hälfte aufgeführt. Diese Komponenten bilden den Kern einer modernen Web-Applikation.

Planung der Entwicklung

Sobald man spezifischere Informationen zur Applikation vorliegen hat beginnt die Planung. Es gibt ein paar grundlegende Fragen, mit denen sich jeder Entwickler auseinandersetzen sollte:

- Wie gross wird der Traffic ab einer bestimmten Usermenge? Müssen pro User nur wenige Daten transferiert werden oder können einzelne User einen grossen Datendurchsatz bewirken?

Von Benutzern generierte grosse Datendurchsätze benötigen viel Bandbreite, was mit hohen Kosten verbunden ist. Jedoch darf sich eine Web-Applikation bei ihrer Benutzung nicht langsam anfühlen, was genug Bandbreite pro Benutzer voraussetzt.

- Welche Art von User-Daten speichert die Applikation?

Jede Web-Applikation verlangt eine gewisse Anzahl persönliche Informationen, weswegen entsprechende Vorkehrungen getroffen werden müssen. Das open-source Projekt „OWASP“ [2], was für „Open Web Application Security Project“ steht, hat sich mittlerweile zu einer Sicherheitsnorm für Web-Applikationen entwickelt und wird von vielen Web-Apps eingesetzt.

- Hält die Applikation das Datenschutzgesetz ein? Gibt es rechtlich gesehen andere kritische Punkte die bei der Ausführung der Applikation beachtet werden sollten?

Applikationen welche mit kritischen Userdaten arbeiten, müssen gesetzliche Auflagen zum Datenschutzgesetz einhalten. Falls eine Web-App solche Daten speichert, sollte folgender Satz auf der Homepage aufgeführt sein: „Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden.“ [3]

- Welche Technologie eignet sich am besten um die Applikation umzusetzen? Bietet diese Technologie eventuell bereits Lösungen für Probleme an, von denen man weiss, dass man sie haben wird?

Man sollte sich über bereits existierende ähnliche Applikationen und deren Technologiewahl informieren. Frameworks bieten für Bereiche wie Templating bereits Lösungen an.

Aus diesem Fragekatalog sollte sich die Wahl zur richtigen Software ergeben. Viele Entwickler programmieren in der Sprache mit der sie schon am meisten gearbeitet haben, dies kann allerdings ebenfalls eine Fehlerquelle sein. Man sollte darum sicherstellen, dass die Sprache gewählt wird, die die Aufgabenstellung der App am besten bewältigen kann.

Falls einem ein besonderes Feature einer anderen App positiv in Erinnerung geblieben ist, hat dies meistens einen guten Grund. Diese bestimmte Funktionalität hat den User überzeugt; er verbindet sein positives Erlebnis, nämlich das Benutzen der App, mit dem einwandfreien Funktionieren dieses Features, welches schnell zum Merkmal einer Plattform werden kann.

Aufsetzen der Software

Viele Frameworks enthalten zum Beispiel ganze Software Packages inklusive Datenbank, vorgefertigten Plugins, Komponenten, Designs und Scripts.

Eine Sache, die man als Entwickler immer in Hinterkopf behalten sollte ist: Je mehr Arbeit einem durch Hilfssoftware oder Frameworks abgenommen wird, umso mehr kann man sich auf das Wesentliche konzentrieren.

Die Wahl der Entwicklungsumgebung

Mittlerweile gibt es eine Vielzahl von guten bis sehr guten kostenlosen IDEs. Die beiden wohl grössten Vertreter dieser Kategorie sind NetBeans und Eclipse, welche mit einem breiten Support an Sprachen glänzen und ebenfalls leicht über ein internes Plugin-System um fehlende Funktionalität erweitert werden können.

Speziell bei Web-Applikationen bieten sich jedoch noch eine neue Art des Entwickelns an. Anbieter wie Cloud9 bieten die Möglichkeit einer eigenen Entwicklungsumgebung welche über eine Cloud-Lösung gehostet wird und machen somit Versionisierungssoftware wie SVN oder GIT unnötig, wenn es darum geht den Projektstand über mehrere Teammitglieder zu synchronisieren.

Lokal vs. Cloud

Offensichtlich sind Cloud-IDEs ein recht junges Angebot, bieten sich aber speziell für sehr modulare bzw. gute kapselbare Projekte wie Web-Applikationen an. Der hauptsächliche Nachteil ist die Einarbeitungszeit der Entwickler. Netbeans und Eclipse haben den Vorteil, dass sie extrem verbreitet sind und somit eine grosse Chance besteht, dass ein Entwickler schon mal in Kontakt mit ihnen gekommen ist. Hierbei ist der Geschwindigkeitsvorteil ausschlaggebend den eine bereits bekannte Umgebung nach sich zieht.

Netbeans vs. Eclipse

Nach unseren Recherchen ist der Feature-technische Unterschied zwischen beiden mittlerweile sehr gering. Netbeans setzt bei seiner Projektorganisation auf ein ähnliches System wie Visual Studio und abstrahiert Projekte in eigene Strukturen, wobei bei Eclipse direkt auf dem Filesystem gearbeitet wird. Für beide existieren Unmengen an Zusatzmodulen, um die Umgebung an persönliche Bedürfnisse anzupassen.

Schlussendlich sind auch hier wieder beide Applikationen leistungsmässig so nahe beieinander, dass es wieder eine Frage der eigenen Präferenz wird.

Git vs. SVN

Bei den Versionskontrolltools wurde hauptsächlich der Vergleich zwischen den beiden verbreitetsten Vertretern, Git und Subversion (SVN), gesucht. Hier besteht jedoch ein grosser Unterschied zwischen den Grundkonzepten beider Technologien und ein damit einhergehender Mehraufwand in der Einrichtung.

Bootstrap A Web App

Subversion ist alteingesessen und die wohl denkbar einfachste Art, Versionskontrolle in einem Projekt zu betreiben. Das Prinzip hierbei ist das Einbinden eines sogenannten Repository-Servers, welcher die unterschiedlichen Projektversionen verwaltet, sowie bei asynchronen Änderungen Buch führt, um ein einfaches Zusammenführen von Dokumenten zu erlauben.

Git ist das „neuere“ von beiden Tools und setzt auf einen dezentralen Vorgang der Versionskontrolle. Im Gegensatz zu SVN befindet sich jeweils ein lokales Repository auf dem jeweiligen Entwicklerrechner, was dem Entwickler erlaubt auch ohne Verbindung zum Hauptrepository auf Funktionen wie Version-Rollbacks zurückzugreifen. Nachteil hier ist jedoch ein entsprechender Mehraufwand in der Einrichtung als auch bei der Benutzung. Der Gebrauch ist nur zu empfehlen, wenn alle beteiligten Entwickler wissen, wie sie das Tool zu gebrauchen haben. Bei unseren persönlichen Projekten in denen wir Git benutzten, hatten wir mitunter extremen Aufwand bei dem Reparieren von Folgeschäden, die durch den Mangel an Wissen von Teammitgliedern und dem dadurch resultierenden, fehlerhaften Gebrauch entstanden.

Cloud9

Bei den Cloud-IDEs haben wir speziell Cloud9 [4] unter die Lupe genommen. Diese bietet die Möglichkeit nach Anmeldung auf ihrer Homepage auf einer IDE zu arbeiten, die auf einer virtuellen Maschine läuft. Der zu brauchenden Technologien für die Entwicklung sind vorgegeben und lassen sich nicht ändern, es können als „Free User“ ebenfalls keine eigenen Technologien installiert werden. Da wir diese Möglichkeit jedoch erst spät im Arbeitsverlauf aufgefunden haben, konnten wir kein abschliessendes Fazit über die Benutzbarkeit von Cloud9 ziehen. Aufgefallen sind uns jedoch der Fokus auf NodeJS und PHP und der ein wenig mager wirkende Funktionsumfang der Entwicklerumgebung.

Unsere Wahl der Entwicklungsumgebung

Wir haben uns für die Entwicklungsumgebung Netbeans mit SVN als Versionisierungssoftware entschieden, da diese uns bereits bekannt sind und wir somit keine Zeit mit der Einarbeitung in andere Software verlieren. Da das Entwickeln in der Cloud einen konstanten Zugang zum Internet benötigt und wir beide Pendler sind, welche manchmal auch unterwegs arbeiten wollen, wo ein Internetzugang nicht immer garantiert ist, haben wir uns gegen eine Cloud-IDE entschieden.

Anno 2014: Welche Technologien liegend bei Startups im Trend

Es gibt heute unzählbar viele Programmiersprachen und Frameworks mit denen eine Web-Applikation entwickelt werden kann.

Nun wäre es interessant zu wissen, welche Technologien von aktuellen Startups benutzt werden. Hier kommt AngelList [5] ins Spiel. AngelList ist laut seiner eigenen Definition [6] eine Plattform für Startups, auf denen diese Investoren oder Arbeitskräfte finden können. Auf AngelList besitzt ein Startup einen Signal Score, welcher einer Bewertung des Unternehmens entsprechen könnte, es ist jedoch nicht klar, aus welchen Faktoren sich dieser Score zusammensetzt oder was er genau repräsentiert. Klar ist jedoch, dass populäre und erfolgreiche Startups über einen höheren Signal Score als die restlichen Startups verfügen. Ausserdem können Unternehmen auf ihrer Profilsseite angeben, mit welchen Technologien sie arbeiten, müssen dies jedoch nicht.

Nachfolgend werden einige Grafiken gezeigt, welche die Benutzung von verschiedenen Technologien von Startups aus dem Jahre 2014 reflektieren sollen. Dabei wird zwischen verschiedenen Qualitätslevel (okay/good/great) unterschieden, alles anhand des Signal Scores der jeweiligen Unternehmen. Alle für diese Arbeit irrelevanten Technologien oder Komponenten eines Startups wurden nicht aufgeführt. Jede Farbe in den nachfolgenden Diagrammen stellt einen Qualitätslevel dar.

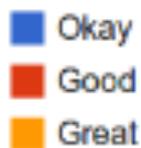


Abbildung 2: Bewertungslegende

Orange stellt also die basierend auf dem Signal Score besten Startups dar, rot die guten und blau die am niedrigsten bewerteten Unternehmen.

Frontendtechnologien

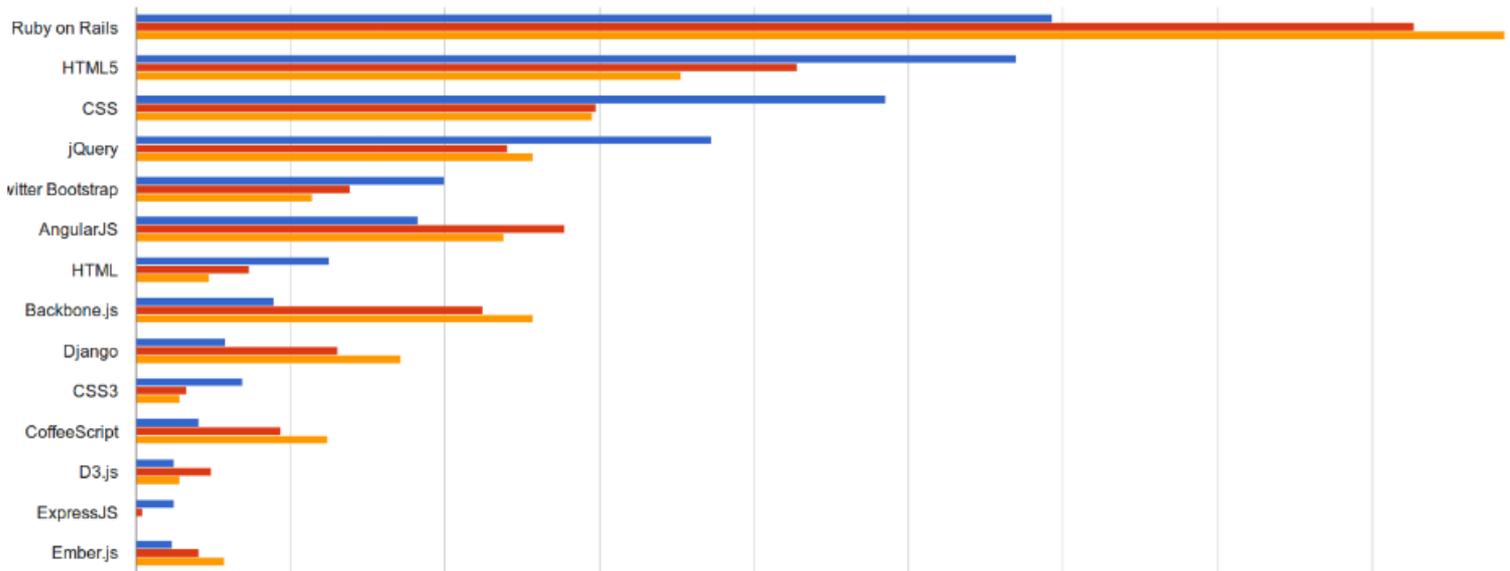


Abbildung 3: Von Startups verwendete Frontendtechnologien

Allen voran dominiert hier Ruby on Rails, gefolgt von HTML5 und CSS, wobei ROR mehrheitlich von erfolgreichen Startups benutzt wird, während weniger erfolgreiche Unternehmen ungefähr gleich stark auf HTML5, CSS und JQuery setzen. Die Beliebtheit von dem Ruby Framework erklärt sich durch die raschen Umsetzungsmöglichkeiten die Ruby bietet und seiner eher konservativen Einstellung gegenüber Konventionen, welche aber Anklang zu finden scheint. Ebenfalls erwähnenswert ist an dieser Stelle die Philosophie [7] von Ruby on Rails, welche auf zwei Prinzipien basiert: „Don't repeat yourself“ & „Convention over Configuration“.

Don't repeat yourself steht dafür, dass jede relevante Information nur an einem Ort eingetragen werden sollte. Dadurch kann verhindert werden, dass Daten inkonsistent werden, weil sie an zwei verschiedenen nicht synchronisierten Orten gelagert wurden.

Convention over Configuration steht für die Konventionen welche Ruby on Rails streng einhält. Dies reicht von der Deklaration von Datentypen bis hin zu der Namensgebung bei Datenbanken und soll helfen, die Komplexität eines Projektes zu mindern.

Mit dieser einfachen Philosophie hat sich Ruby on Rails die eindeutige Vorherrschaft im Bereich Front-end Technologie gesichert.

Interessant ist ebenfalls, dass HTML5 im Ranking vor HTML4 oder niedrigeren Versionen liegt. HTML5 ist mit seinen neuen Elementen und Tags, der Implementation von Geolocation, Scalable Vector Graphics und der WebStorage-Funktion seit einigen Jahren der neue Standard in der Webentwicklung. Die Ursachen dafür liegen hauptsächlich im raschen Wachstum des Mobile Sektors und dem Aufkommen von Responsive Designs mittels CSS3. Web Apps in niedrigeren Versionen als 5.0 zu entwickeln würde in einem Mehraufwand resultieren, da beinahe alle modernen Frameworks auf HTML5 setzen. HTML5 ist ausserdem einfacher zu implementieren und komplett rückwärtskompatibel.

Bootstrap A Web App

Als letztes sticht ins Auge, dass die am besten bewerteten Unternehmen Backbone.js, Django und CoffeeScript etwa doppelt bis dreifach so oft nutzen wie die schlechter bewerteten. Dies hängt mit der Komplexität der jeweiligen Technologien und deren Anforderungen an einen Programmierer zusammen.

Welches Front-End Framework für uns?

Beim Frontend führt heutzutage kein Weg mehr an der Kombination von HTML, CSS und JavaScript vorbei. Hierbei sind für uns Frameworks interessant, welche es uns erlauben, schnell und einfach präsentierbare Ansichten für unsere Webapplikation zu erstellen.

Wie bereits erwähnt ist ein nicht zu vernachlässigender Punkt in der modernen Webentwicklung das Konzept des „Responsive Design“, welches die Anpassung des Layouts auf unterschiedliche Geräteauflösungen predigt. Grid-Systeme wie Bootstrap und Foundation unterstützen dieses Konzept von Haus aus, was der Projektgeschwindigkeit zugute kommt.

Bootstrap

Bootstrap ist aus dem heutigen Web nicht mehr wegzudenken. Gleichzeitig erntet das Framework vermehrt Kritik, zu „unflexibel“ zu sein. Bootstrap eignet sich jedoch nach wie vor sehr gut bei Web-Projekten, in welchen präsentationstechnisches Herausstechen eher zweitrangig ist. Die riesige Auswahl an Templates, die man im Netz finden kann, macht einen ersten Wurf noch einfacher.

Rein technisch gestaltet sich Installation sowie das Einbinden sehr einfach, da Bootstrap nichts weiter als eine Kombination von CSS und JavaScript-Dateien ist.

Foundation

Foundation ist ein weiteres verbreitetes Frontend-Framework, welches ebenfalls Responsive-Design anbietet und zu den schnellsten UI-Frameworks gehört, ohne dabei so restriktiv im Design zu sein wie andere Frameworks. Foundation besitzt über ein robustes Gridsystem, das wohl Geschmackssache bleibt und bietet ebenfalls „fastclick“ an, was das Bedienen einer Web-Applikation auf mobilen Geräten verbessert. Weiter zur Geschwindigkeit trägt die „GPU-Acceleration“ bei, welche flüssige Animationen auf Basis von CSS3 anbieten kann oder die Integration von SASS.

Frontend Fazit

Die Wahl unseres Frontends fiel auf Bootstrap, hauptsächlich weil uns die Templates bei Bootstrap eher ansprachen und unseren Bedürfnissen entsprachen. Bootstrap genießt im Web einen guten Ruf und bietet sich für Arbeiten unsere geradezu an.

Backendtechnologien

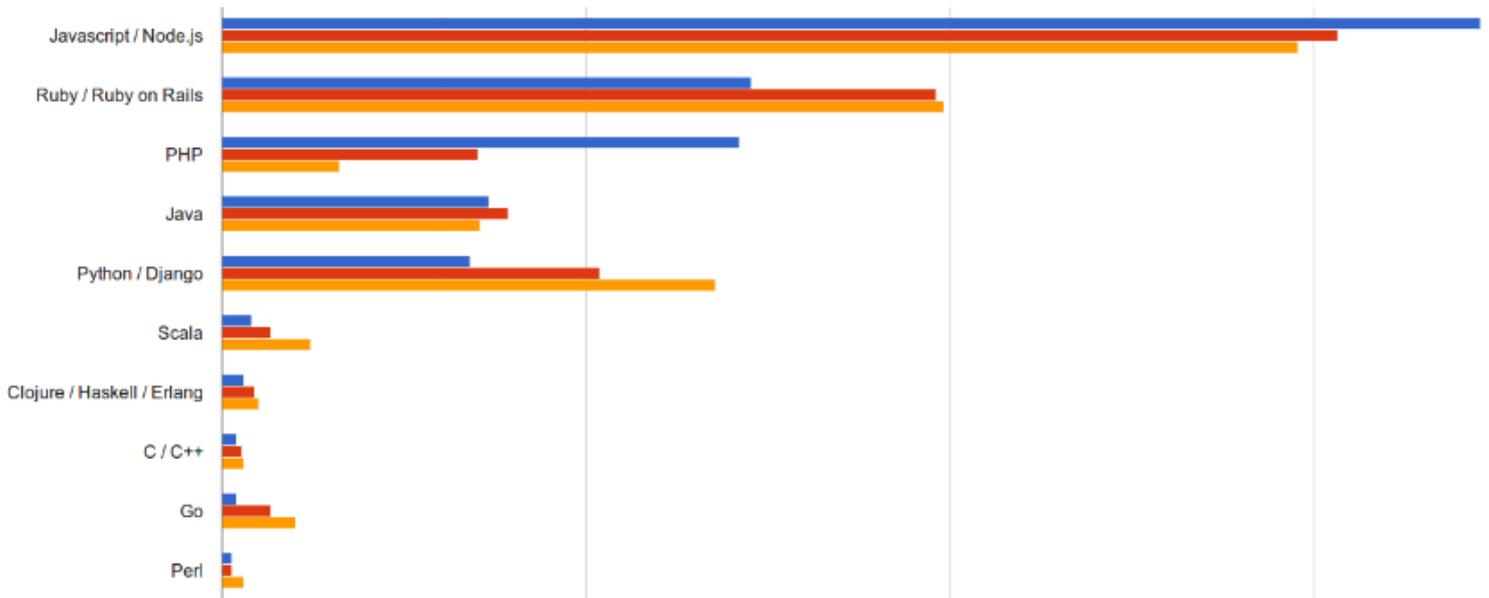


Abbildung 4: Von Startups verwendete Backendtechnologien

Was einem sofort auffällt, ist die Dominanz von Javascript und Node Js, welche jedoch von mehr Startups mit einem niedrigen Signal Score verwendet werden. Nichtsdestotrotz sind beide Sprachen oder eine Kombination aus beiden mehr als doppelt so verbreitet wie Ruby oder Ruby on Rails, welche an zweiter Stelle rangieren. Diese starke Verbreitung von Javascript findet ihre Berechtigung in den vergangenen paar Jahren, in denen beinahe jede Homepage auf der Welt ausgiebigen Gebrauch von Javascript machte. Dass man im Internet für beinahe jedes erdenkliche Problem eine bereits vorgefertigte Javascript-Funktion finden kann kommt der Sprache ebenfalls zugute.

Ebenfalls bemerkenswert ist der Unterschied bei der Benutzung der Sprachen Python/Django und PHP. Laut Angellist besteht ein Zusammenhang zwischen dem Signal Score eines Unternehmens und seiner Benutzung von PHP oder Python.

Wie man auf der Grafik sehen kann, scheinen die Spitzenreiter der Startups im Vergleich mit den schlechter bewerteten Startups viel öfters Python/Django zu benutzen während weniger erfolgreiche Startups öfters auf PHP setzen.

Nach unserer Erfahrung könnte dies daran liegen, dass PHP eine sehr einfache zu lernende Programmiersprache ist, mit der man schnell Ergebnisse erzielen kann und da viele einsteigerfreundliche Frameworks wie Joomla oder Wordpress immer noch PHP als ihre Skriptsprache benutzen, fällt die Verbreitung doch ziemlich hoch aus. Jedoch bringt PHP auch viele kleine Probleme mit sich. Wenn ein Startup also qualitativ hochwertiger ausfallen sollte, verzichtet der erfahrene Programmierer meistens auf PHP.

PHP erlaubt die dynamische Gestaltung von HTML Seiten, eine Sache die modernere Sprachen jedoch besser und sauberer erledigen als PHP. Wie man in dem Bild sieht, befinden sich Python oder Ruby trotz der starken Verbreitung von PHP im Ranking auf höheren Plätzen als PHP.

Java Server Faces

Java Server Faces oder kurz JSF, ist die eine Web-Technologie, die auf den Einsatz von Java und sogenannten Managed Beans im Backend basiert. Die verbreitete Natur von Java und der daraus resultierenden großen Anzahl an verschiedenen Funktionsbibliotheken macht diese Technologie vor allem für Entwickler interessant, welche bereits Erfahrungen im Java-Umfeld mitbringen können. JSF dient hierbei als Wrapper von Java-Funktionalität in eine Webseite.

Ein weiterer Vorteil ist der Zugriff auf das extrem mächtige JUnit-Testing Framework, um Komponententests durchführen zu können. Ausserdem gibt es mit Jenkins einen mächtigen CI-Server, um automatische Tests zu realisieren.

NodeJS

Die zweite Backend Technologie, welche sich einer immer grösseren Beliebtheit erfreut ist das auf der Chrome-Engine basierende NodeJS Framework. Der grosse Vorteil hier ist, dass das gesamte Framework auf JavaScript basiert. Das macht das Framework flexibel was zusätzlich durch seinen modularen Ansatz unterstrichen wird. NodeJS bietet ausserdem einen eigenen Paketmanager namens NPM (Node Package Manager), über welche einfach zusätzliche Module heruntergeladen werden können.

Für NodeJS gibt es auch schon einige Unit Testing Bibliotheken wie Buster oder Karma. Da es sich wie bereits erwähnt bei NodeJS um JavaScript Code handelt, können auch JavaScript allgemeine Tools wie Jasmine in Betracht gezogen werden.

Ruby On Rails

Wie bereits bei den Frontend-Technologien erwähnt, verdankt Ruby on Rails seinem Erfolg seiner Philosophie und seiner mittlerweile grossen Community. Ein Entwickler, der sich mal in Ruby eingearbeitet hat kann mit Ruby on Rails extrem effizient Applikationen auf hohem Niveau entwickeln. Ruby on Rails hat sich über die letzten Jahre zu einem State-of-the-Art Framework entwickelt und bleibt seinem Ruf als Spitze der Open Source Frameworks treu.

Unsere Backend Wahl

Wir entschieden uns bei der Backendtechnologie für JSF, da wir beide mehr Know-How in Java mitbringen als in anderen Programmiersprachen, Java plattformunabhängig und sehr verbreitet ist und es sich dadurch einfach gestalten sollte die bereits vorhandenen Entwicklungserfahrungen ins Web-Design zu übernehmen, ohne einen grossen Einarbeitungsaufwand zu betreiben.

Datenbanktechnologien

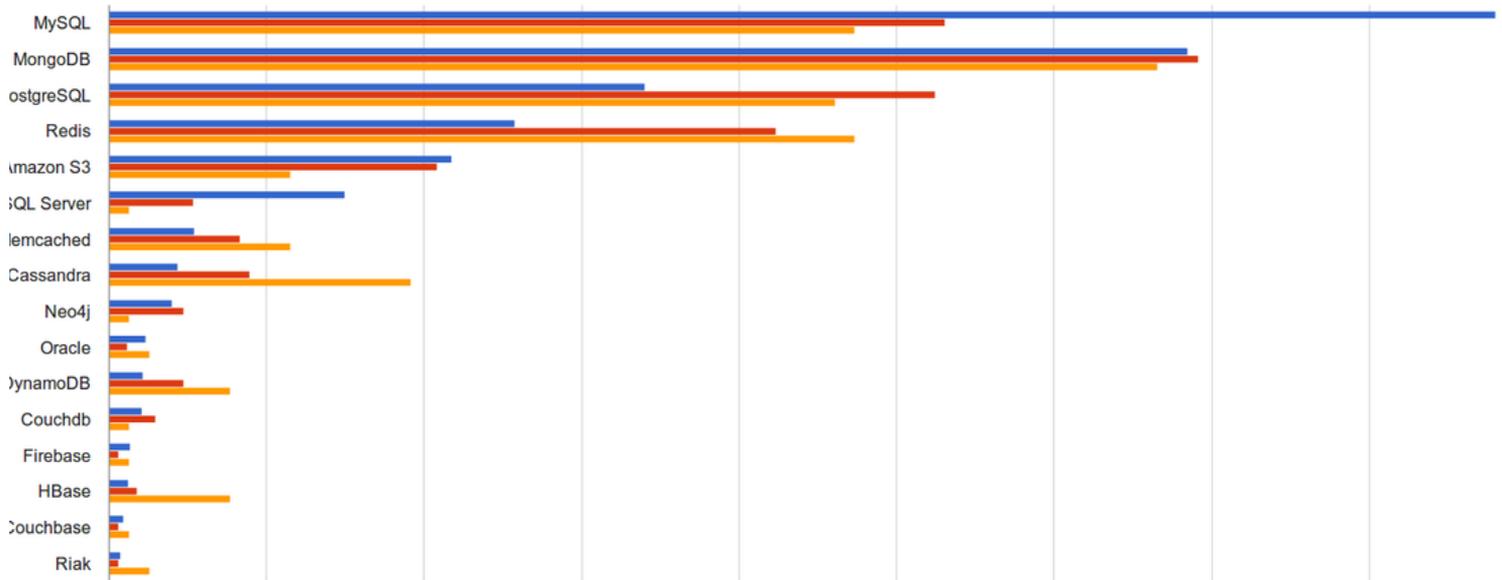


Abbildung 5: Von Startups verwendete Datenbanktechnologien

Auf den ersten Blick fällt einem auf, dass MySQL, MongoDB, PostgreSQL und Redis eindeutig die ersten Plätze besetzen. Während die Verteilung der Qualität der Startups bei MongoDB jedoch beinahe identisch ist und bei PostgreSQL leicht zu den stärkeren Startups neigt, benutzen doppelt so viele weniger populäre Startups MySQL wie die sehr populären.

Dies hängt mit der Zugänglichkeit von MySQL zusammen; ausserdem wird MySQL als fester Bestandteil von vielen modernen, anfängerfreundlichen Frameworks wie Joomla oder Wordpress benötigt und mitinstalliert. Daher kommt die hohe Verbreitung unter den weniger gut bewerteten Unternehmen, falls man davon ausgeht, dass diese von weniger erfahrenen Leuten programmiert werden.

Schliesslich stechen einige Technologien wie Redis, Cassandra oder Memcached hervor, weil sie im Vergleich mit den anderen Technologien von einer verhältnismässig hohen Anzahl von erfolgreichen Startups verwendet werden.

Welche Datenbank ist die Richtige?

Eine erste grundlegende Entscheidung die wir treffen mussten war, ob wir bei der Datenbank den relationalen oder non-relationalen Ansatz verfolgen sollen. Beim herkömmlichen, relationalen Ansatz boten sich uns MySQL und PostgreSQL an, wobei wir bei den non-relationalen vor allem MongoDB untersucht haben.

MySQL vs. PostgreSQL

Die Frage ob lieber MySQL oder PostgreSQL eingesetzt werden soll, ist nach wie vor ein Glaubenskrieg unter den Web-Entwicklern. MySQL gilt hierbei als die „benutzfreundlichere“ Lösung und erfreut sich an einfacher Konfiguration. PostgreSQL gilt als schwerfälliger aber

Bootstrap A Web App

performanter. Außerdem gilt es als gewohnter, wenn man bereits mit einem der grossen Datenbanksysteme Erfahrungen gemacht hat.

Mittlerweile geben sich beide Systeme rein Feature-technisch nicht mehr viel. Somit spielen hauptsächlich benutzerspezifische Vorlieben eine Rolle. Wir persönlich haben mehr Erfahrung mit MySQL über das Benutzen von XAMPP.

MongoDB

Da Webapplikationen meist mit sehr grossen Datenmengen arbeiten, erfreuen sich non-relationale Datenbanken immer grösserer Beliebtheit im Web, da die Datenbankeinträge von solchen Datenbanken stets indiziert werden können. MongoDB ist hierbei einer der grössten Vorreiter der Open-Source Möglichkeiten.

Unsere Datenbank Wahl

Da MySQL als Open Source Software sehr verbreitet und einfach zu verwenden ist, entschieden wir uns es als Datenbanktechnologie zu verwenden. Es existieren im Web viele Tools die bei Verwaltung und Erstellung von Datenbanken mittels MySQL helfen können.

Frameworks für eigene REST-APIs

Ein immer wichtiger werdender Teil einer Webapplikation ist das Bereitstellen einer eigenen API. Das Ziel hierbei ist das Bereitstellen einer einfachen Möglichkeit, um auf applikationsspezifische Daten anzusprechen. Damit erleichtert man das Einbinden von Drittanbietern und fördert die Verbreitung.

Der moderne Standard für Web-API's sind RESTful Services, welche über JSON-Objekte kommunizieren. In Bezug auf JSF sind für uns im Speziellen Java-Bibliotheken interessant, welche es uns einfacher machen, einen solchen Service zu erstellen. Hierbei haben wir uns auf Bibliotheken konzentriert, welche sich an die von Oracle bereitgestellten JAX-RS API halten.

Jersey

Eine sehr verbreitete Funktionsbibliothek ist Jersey. Sie gilt als Referenzimplementation für REST Services in Java. Entsprechend finden sich auch Unmengen an Ressourcen im Web.

RESTlet

RESTlet ist eines der ältesten Java-REST-Frameworks. Es unterstützt die JAX-RS API, ist aber grundsätzlich konzipiert mit der eigenen API-Implementation benutzt zu werden. RESTlet funktioniert auf einem eigenen Webserver und bringt für diesen eigene Client- und Server-Implementationen mit.

API Framework Fazit

Bei uns fiel die Wahl auf Jersey, da es als Referenzimplementation bereits bei Netbeans mitgeliefert wird und dadurch bereits integrierte Tools von der Entwicklungsumgebung bereitgestellt werden.

Unsere (virtuelle) Entwicklermaschine

Um die Ergebnisse dieser Arbeit in der Praxis zu testen, wurde mithilfe von Oracle Virtual Box eine virtuelle Maschine vorbereitet. Das Ziel war es zum einen die Kompatibilität der verschiedenen Technologien zu testen, zum anderen ein Referenzprojekt für unsere gewählte Technologie-Sammlung zu erstellen.

Grundspezifikationen der virtuellen Maschine

Windows 7 32-Bit SP1

Virtual Box von Oracle

Installation des ausgewählten Tech-Stacks

Installiert wurden:

- JDK 8 Update 25
- Netbeans IDE 8.0.2
- Tomcat 8.0.15.0 (integriert in Netbeans)
- MySQL Community Edition 5.6.22

Das Einrichten der einzelnen Elemente verlief grösstenteils problemlos. Ausnahme hierbei war der Tomcat Webserver, da in der installierten Version ein Fehler bei einer Escape-Sequenz in der Catalina.bat Datei vorherrschte.

Das Einbinden des MySQL-Servers in Netbeans ist auch schnell erledigt und ist anschliessend über das „Services“- Tab erreichbar.

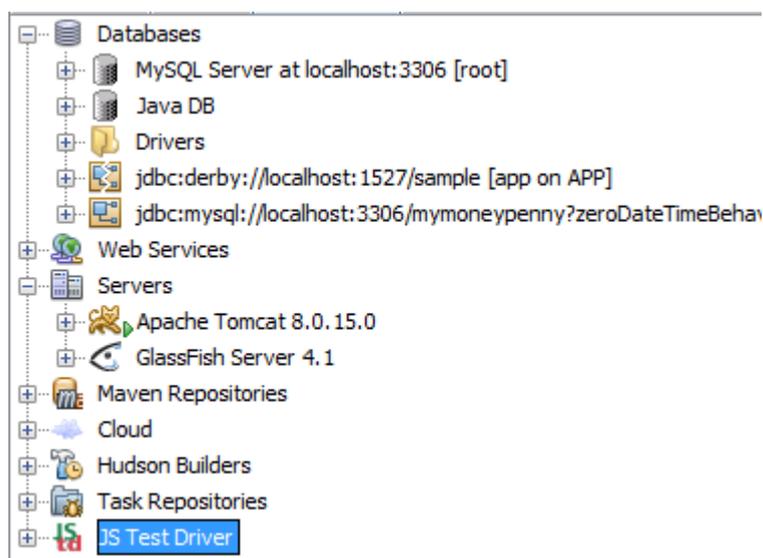


Abbildung 6: Netbeans Service-Tab

Das Referenzprojekt MyMoneyPenny

Projektstruktur

JSF Projekte verlangen eine vordefinierte Projektstruktur. Wichtig hierbei ist die Tatsache, dass alle externen Anfragen keinen Zugriff auf die Ordner WEB-INF haben. Damit bietet sich dieser Ordner für das hinterlegen von applikationsspezifischen Ressourcen und Templates an.

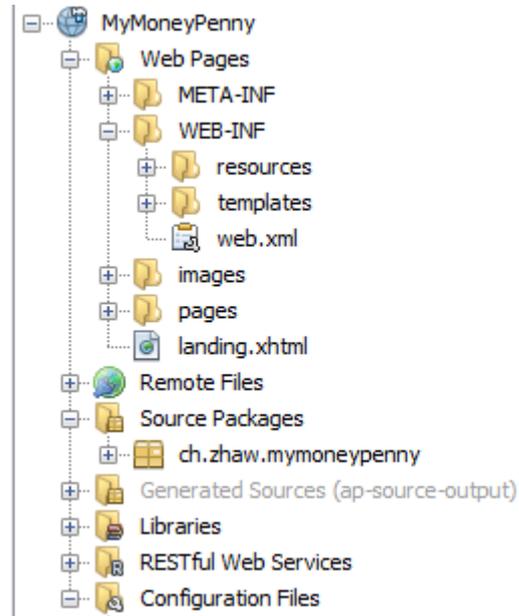


Abbildung 7: MyMoneyPenny Projektstruktur

Frontend

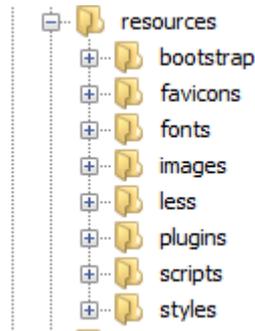


Abbildung 8: Frontend-Ressourcen

Wie definiert wurde für das Frontend Bootstrap eingesetzt, um den Design-Aufwand möglichst gering zu halten. Hierfür entschieden wir uns für ein kostenloses Template von Xiaoying Riley (3rd Wave Media) namens devAid, welches das Einbinden von zuvor nicht geplanten, zusätzlichen JavaScript-Plugins mit sich bringt. Darunter liegt JQuery als einfaches, JavaScript-basiertes Animationsframework.



Abbildung 9: Willkommensseite basierend auf Bootstrap mit angepasstem Template

Datenbank

Für das Projekt wurde eine eigene Datenbank namens „mymonneypenny“ mit folgenden Tabellen und Spalten erstellt:

TABELLE: USER

ID	ID als Schlüssel
NAME	Namen des Benutzers
MAIL	Mailadresse des Benutzers
ROLE	Sicherheitsrolle des Benutzers

TABELLE: QUESTION

ID	ID als Schlüssel
TEXT	Die Frage

TABELLE: ANSWER

FK_QUESTION	Fremdschlüssel der auf die ID von der Question-Tabelle verweist
FK_USER	Fremdschlüssel der auf die ID von der User-Tabelle verweist
ANSWER	Wert der Antwort

Netbeans beinhaltet hierbei zwar ein grafisches Tool für die Bearbeitung von Datenbanken, jedoch empfiehlt es sich das MySQL Workbench Tool zu benutzen. Für das Anbinden der Datenbank an die Applikation empfahl sich der MySQL-JDBC Treiber, sowie die Java Persistence Library (JPA).

User Management

Bei der Benutzerverwaltung bietet JSF eine eigene Lösung an. Innerhalb des Serverkontexts ist es möglich einen sogenannten „Realm“ zu definieren. In diesem Eintrag werden der Datenbank-Pfad und die Tabelle mit den Benutzerinformationen gespeichert. Ebenfall beinhaltet dieser Eintrag die Namen der Spalten mit den Informationen für den Benutzernamen und Passwort, sowie deren Sicherheitsrollen.

Die Frage ob sich dieser Ansatz lohnt, hängt von der Flexibilität der gesuchten Lösung ab. Zum Beispiel ist es so nur schwer möglich, dem Benutzer die Wahl zu lassen, ob er sich mit seinem Benutzernamen oder seiner Email Adresse anmelden möchte. Deswegen wurde hier auf diesen Ansatz verzichtet.

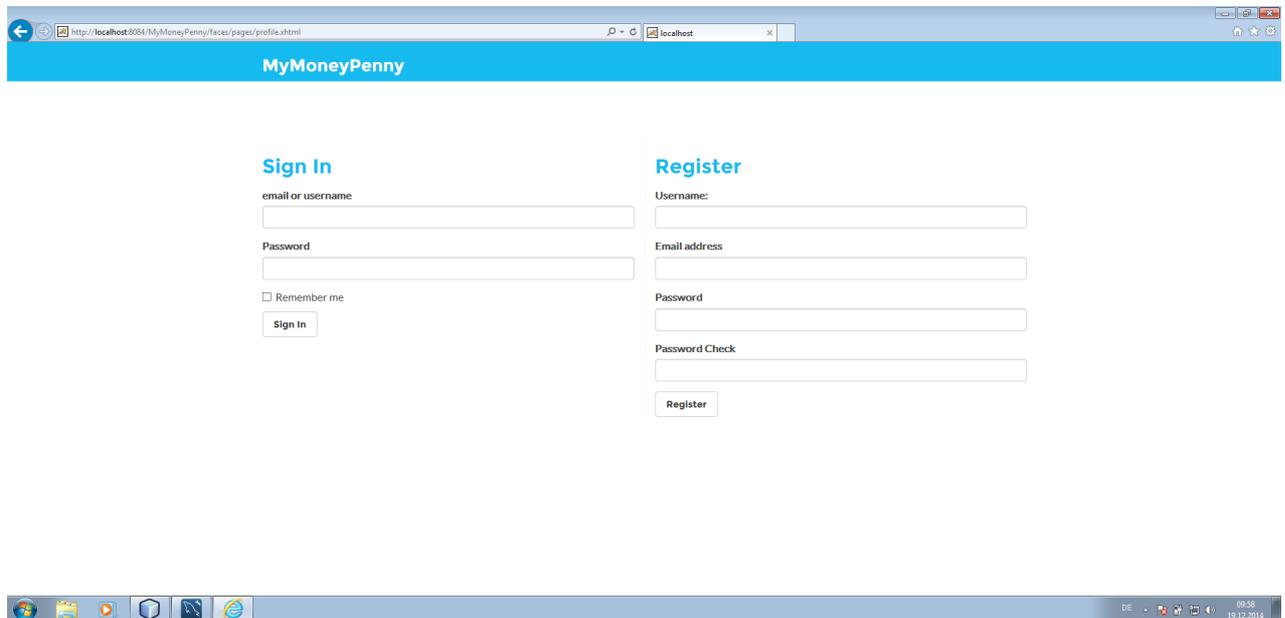


Abbildung 10: Login- sowie Registrationsseite

Security

Den Sicherheitsteil wurde mittels von JSF-Filtern realisiert. Hier bietet JSF die Möglichkeit, Filter in der Applikation für bestimmte Zugriffspfade zu registrieren, welche dann beim Aufruf vor dem senden der Antwort durchlaufen werden und die Antwort entsprechend manipulieren. Somit kann verhindert werden, dass unbefugte Zugriff auf heikle Funktionen der Webapplikation bekommen.

Hierbei wurde hier ein Zugrifffilter definiert, welcher beim Aufruf einer Seite den Benutzer auf den eingeloggten Zustand überprüft und anschliessend zu der gewünschten Seite weiterleitet, falls er die Anforderungen erfüllt. Andernfalls wird die Login-Seite angezeigt.

```
public class AccessFilter implements Filter {

    FilterConfig fc;

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        fc = filterConfig;
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
        throws IOException, ServletException {
        HttpServletRequest req = (HttpServletRequest) request;
        HttpServletResponse resp = (HttpServletResponse) response;
        HttpSession session = req.getSession(true);

        String pageRequested = req.getRequestURI().toString();

        SessionBean b = (SessionBean) session.getAttribute("sessionbean");

        if (b != null) {

            if (b.isLoggedin() && pageRequested.contains("signin.xhtml")) {
                resp.sendRedirect("pages/dashboard.xhtml");
            } else if (b.getUser().getRole() < 2 && pageRequested.contains("admin.xhtml")) {
                resp.sendRedirect("pages/profile.xhtml");
            } else {
                chain.doFilter(request, response);
            }

        } else {
            resp.sendRedirect("pages/signin.xhtml");
        }
    }
}
```

Abbildung 11: Source-Code für Accessfilter.java

Fragebogen

Der Fragebogen soll den Teil der Applikation symbolisieren, welcher die Business-Logik enthält. In unserem Beispiel werden Fragen aus der Datenbank gelesen, anschliessend kann der Benutzer einen Wert als Antwort angeben und diesen dann in der Datenbank speichern.

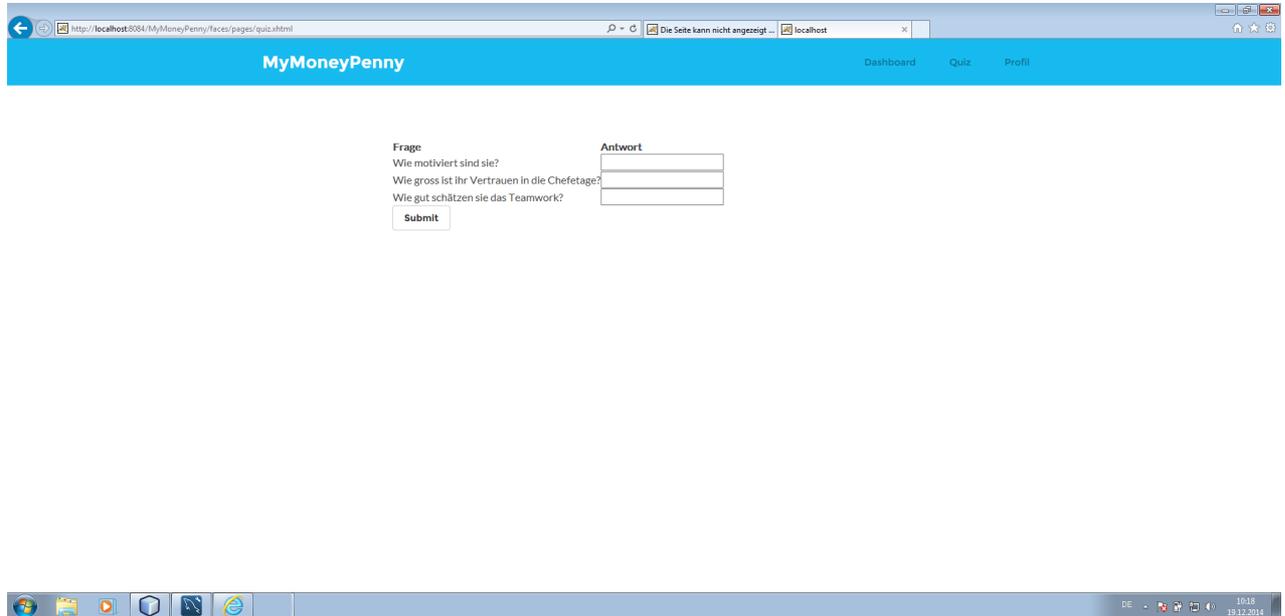


Abbildung 12: Auslesen einer Anzahl Fragen aus der Datenbank

Schlusswort & Ausblick

Die Technologien, die heutzutage in der Webentwicklung verwendet werden sind derart vielfältig, dass es schwer fällt einen guten Einstiegspunkt zu finden. Jedes Framework bietet andere Lösungsansätze an, sogar die reinen Konzepte unterscheiden sich stark. Dadurch gibt es viele im Kontext gute Lösungen aber wenig „best-practice“ Ansätze die übergreifend verwendet werden können.

Im Laufe dieser Arbeit stellte sich heraus, dass es bei der heute vorhandenen Vielfalt schwierig ist, einen Einstiegspunkt zu finden. Basierend auf unseren Java Kenntnissen JSF als Backendtechnologie einzusetzen, stellte sich nicht als so grosse Hilfe heraus, wie wir es erwartet hätten. Obwohl JSF auf Java basiert, wurde ersichtlich, dass sich das Verständnis des Frameworks als wichtiger herausstellte, als die Kenntnis der Programmiersprache. Da man als Entwickler nicht viel mehr aus der Grundtechnologie ziehen kann, wie wenn man mit einem Framework arbeitet, lohnt sich das Einlesen in ein mächtiges Framework eher als sich auf seine bereits vorhandenen Kenntnisse zu verlassen.

Der von uns definierte Lösungsansatz mit einer virtuellen Maschine überzeugt auch nicht mit Nachhaltigkeit, da sich speziell Webtechnologien heutzutage so schnell ändern und unsere VM laufend erneuert werden müsste.

Für zukünftige Webentwickler empfehlen wir möglichst modulare Frameworks mit einer grossen Community, welche das weiterführen und die Unterstützung bei der Benutzung dieser Technologie garantieren. Durch die grosse Verbreitung eines solchen Frameworks wird sichergestellt, dass eine grosse Anzahl von Online-Ressourcen vorhanden ist und es eine Menge von Lösungsansätzen gibt, an denen man sich orientieren kann.

Abbildungsverzeichnis

Abbildung 1: Elemente einer Webapplikation	11
Abbildung 2: Bewertungslegende	15
Leo Polovets, „CodingVC“ [Online] http://codingvc.com/which-technologies-do-startups-use-an-exploration-of-angellist-data	
Abbildung 3: Von Startups verwendete Frontendtechnologien	16
Leo Polovets, „CodingVC“ [Online] http://codingvc.com/which-technologies-do-startups-use-an-exploration-of-angellist-data	
Abbildung 5: Von Startups verwendete Datenbanktechnologien	20
Leo Polovets, „CodingVC“ [Online] http://codingvc.com/which-technologies-do-startups-use-an-exploration-of-angellist-data	
Abbildung 9: Willkommenseite basierend auf Bootstrap mit angepasstem Template.....	25
Abbildung 10: Login- sowie Registrationsseite	27
Abbildung 11: Source-Code für Accessfilter.java	28
Abbildung 12: Auslesen einer Anzahl Fragen aus der Datenbank	29

Quellenverzeichnis

[1] Peter Wayner, „15 hot programming trends -- and 15 going cold“ [Online]
<http://www.infoworld.com/about/about.html>

[2] Open Web Application Security Project, „Open Web Application Security Project“ [Online]
https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project#Core_Values

[3] Weitere Informationen zum Schweizer Datenschutzgesetz:
<http://www.edsb.ch/>
<http://www.edoeb.admin.ch/datenschutz/00683/index.html?lang=de>

[4] Cloud 9, „Cloud 9“ [Online] <http://www.c9.io>

[5] AngelList, „AngelList“ [Online] <https://angel.co>

[6] AngelList, „AngelList“ [Online] <https://angel.co/help/general#about>

[7] Wikipedia, „WikiPedia“ [Online] http://de.wikipedia.org/wiki/Ruby_on_Rails#Philosophie