

HOCHSCHULE FÜR TECHNIK ZÜRICH

SEMESTERARBEIT VACATION PLANNER

## Handout: Präsentation

*Author:*  
Raffael SCHMID

*Dozent:*  
Beat SEELIGER

# 1 Einleitung

Auf der Basis von Lift und der Programmiersprache Scala wurde ein Prototyp zur Planung von Ferien innerhalb Gruppen oder Teams entwickelt. Ziel dieser Arbeit war nebst dem fertigen Prototypen auch die Einschätzung dieses relativ neuen Frameworks und der Sprache Scala. Das dafür notwendige Knowhow wurde teils zuvor, teils während der Semesterarbeit erarbeitet.

# 2 Analyse der Aufgabenstellung

Der Prototyp zur Ferienplanung soll im wesentlichen die folgenden Use Cases abdecken:

- Benutzeradministration von Team-Membren und -Ownern
- Erfassung von Teams und zuteilung von Mitarbeitern
- Eingabe und Bestätigung von Ferien für Mitarbeiter.

Während der Designphase sollen Rollenkonzept, Prozesse, Navigation und das Relationale Model definiert werden.

# 3 Grundlage

Scala entstand im Wesentlichen auch mit dem Ziel der Erneuerung der Sprache Java. Dies war aufgrund der Anforderungen an die Rückwärtskompatibilität nicht im erwünschten Masse möglich. Viele der Konzepte innerhalb Scala sind auch bereits aus anderen Sprachen bekannt, nebst dem grossen Einfluss aus der Java-Ecke gab es auch unterschiedliche andere Sprachen, die in Scala ihre Spuren hinterliessen: Erlang ist erkennbar durch die Actors, Standard ML durch das Typsystem, Haskell, Smalltalk, etc. Es handelt sich also nicht um einen Klon auf die JVM einer bereits existierenden Sprache wie bei Jython (Python), JRuby (Ruby), Clojure (Lisp).

## 3.1 Eigenschaften

### 3.1.1 Scala

Bei Scala handelt es sich nicht um eine rein Funktionale Sprache, allerdings erfüllt sie viele der dafür notwendigen Bedingungen:

- Anonyme Definition von Funktionen

- Funktionen werden wie Daten behandelt. D.h. es gibt in einer statisch typisierten Sprache einen passenden Datentyp zu jeder Funktion.
- Funktionen können als Parameter und/oder Rückgabewerte von anderen Funktionen sein.

Weitere Eigenschaften der Sprache Scala sind: objektorientiert, statisch Typisiert.

### 3.1.2 Lift

Lift ist ein Webframework basierend auf der Sprache Scala, viele der Funktionalitäten die Lift zur Verfügung stellt basieren deshalb auf dessen Sprachfeatures (Bsp: XML-Datentypen, etc.). Grundsätzlich sind viele Ansätze bei Lift relativ neu und innovativ und möglicherweise in gewissen Bereichen der Zeit etwas voraus. Für gewöhnliche Webseiten ist es m.E. nicht möglich, in Sachen Entwicklungsgeschwindigkeit mit bereits etablierten Frameworks wie Rails oder Grails mitzuhalten. Hinzu kommt, dass die Komplexität der darunterliegenden Sprache die Verbreitung des Frameworks nicht fördert.

## 3.2 Design und Konzeption

Bevor der Prototyp in Angriff genommen wurde, wurden basierend auf den Use Cases Rollenkonzept, Navigationskonzept und die Prozesse definiert. Das Relationale Modell der Datenbank wurde anhand dieser Konzepte definiert.

## 4 Implementation Prototyp

Als Basis für den Prototypen wurden folgende Frameworks verwendet:

- **Maven:** Als Basis für Lift-Projekte kann sowohl Maven als auch SBT<sup>1</sup> verwendet werden. Die grösste Verbreitung von Maven führt zu einem reichhaltigen Ökosystem an Plugins. So konnte beispielsweise das Projekt mittels Plugin auf die Stax-Plattform<sup>2</sup> deployt werden.
- **Flex:** Ein Teil der Funktionalität im Frontend wurden mittels Flex entwickelt. Der Zugriff auf das Backend findet via RESTful Webservices statt.

---

<sup>1</sup>Simple Build Tool

<sup>2</sup>Stax basiert auf der Amazon Elastic Cloud

- **JPA:** Als Basis für die Persistenz wurde aus verschiedenen Gründen (Reifegrad, Dokumentation) JPA 2.0 den Lift-internen Bibliotheken Mapper und Record vorgezogen.