

HOCHSCHULE FÜR TECHNIK ZÜRICH

SEMESTERARBEIT VACATION PLANNER

## Handout: Präsentation

*Author:*  
Raffael SCHMID

*Dozent:*  
Beat SEELIGER

# 1 Einleitung

Auf der Basis von Lift Webframework 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 Bewertung 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-Membnern und -Ownern
- Erfassung von Teams und zuteilung von Mitarbeitern
- Eingabe und Bestätigung von Ferien für Mitarbeiter.

# 3 Grundlage

Scala entstand im wesentlich auch dadurch, da die Weiterentwicklung von Java aus vielerleigründen (Rückwärtskompatibilität, etc.) nicht im erwünschten Masse möglich war. Nebst dem grossen Einfluss der Sprache Java gab es auch unterschiedliche andere Sprachen, die in Syntax und Semantik von Scala erkennbar sind: Haskell, Erlang, Standard ML, Smalltalk, etc. Es handelt also nicht um einen Klon einer bereits existierenden Sprache auf der JVM (Bsp: JRuby-Ruby, Clojure-Lisp, Jython-Python, etc.)

## 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

Bei Lift handelt es sich um ein Webframework auf der Basis von Scala. Viele der Funktionalitäten die Lift zur Verfügung stellt basieren deshalb relativ direkt auf den Sprachfeatures von Scala (Bsp: XML-Datentypen, etc.). Grundsätzlich sind viele Ansätze bei Lift relativ neu und innovativ und möglicherweise in gewissen Bereichen der Zeit etwas voraus. Für Anforderungen einer gewöhnlichen Webseite ist das Entwicklungstempo mit diesem Framework (zu) langsam. Meines Erachtens liegt das grosse Interesse daran insbesondere an der verwendeten Programmiersprache Scala.

## 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 grösse Verbreitung von Maven führt zu einem reichhaltigen Ökosystem an Plugins. So konnte beispielsweise das Projekt mittels dem Stax-Plugin auf diese 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.
- **JPA:** Als Basis für die Persistenz wurde aus verschiedenen Gründen (Reifegrad, Dokumentation) JPA 2.0 den Lift-internen Bibliotheken Mapper und Record vorgezogen.

---

<sup>1</sup>Simple Build Tool

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