

Version: 0.1  
Autoren: Tobias Lanz, Mejdin Hatema   
Studiengang: BSC INF 2013.ZH1  
Ort: REGENSDORF

Pflichtenheft / Projektauftrag

Garden Designer

30. August 2015

# Versionen

|  |  |  |  |
| --- | --- | --- | --- |
| Autor | Datum | Version | Zusammenfassend |
| Tobias Lanz | 02.09.2015 | 0.1 | Initiale Struktur |
| Tobias Lanz | 02.09.2015 | 0.2 | Einleitung  Einbindung Mindmap  Grobplanung |
| Tobias Lanz | 22.10.2015 | 0.3 | Projektumfang |
| Tobias Lanz | 23.10.2015 | 0.4 | Abgrenzung |
| Tobias Lanz | 27.10.2015 | 0.5 | Information & Kommunikation |
| Tobias Lanz | 28.10.2015 | 0.6.1 | Lösungsvariante 2 - RIA Prototyp |
| Tobias Lanz | 31.10.2015 | 0.6.2 | Lösungsvariante 2 - RIA Prototyp |
| Tobias Lanz | 01.11.2015 | 0.7 | Realisierung Funktionaler Anforderungen |

Inhaltsverzeichnis

[1 Versionen 0](#_Toc434165890)

[2 Einleitung 0](#_Toc434165891)

[2.1 Leserkreis 0](#_Toc434165892)

[2.2 Umfang 0](#_Toc434165893)

[2.3 Abgrenzung 0](#_Toc434165894)

[3 Projekt Übersicht 0](#_Toc434165895)

[3.1 Projektorganisation 0](#_Toc434165896)

[3.2 Information & Kommunikation 0](#_Toc434165897)

[4 Anforderungsanalyse 0](#_Toc434165898)

[4.1 Ursprüngliche Ausgangslage 0](#_Toc434165899)

[5 Lösungsvariante 1 - Swing Prototyp 0](#_Toc434165900)

[6 Lösungsvariante 2 - RIA Prototyp 0](#_Toc434165901)

[6.1 Vorarbeiten 0](#_Toc434165902)

[6.2 Mittel & Methoden 0](#_Toc434165903)

[6.2.1 Bluemix 0](#_Toc434165904)

[6.2.2 DevOps (https://hub.jazz.net) 0](#_Toc434165905)

[6.2.3 Cloudfoundry 0](#_Toc434165906)

[6.2.4 Git 0](#_Toc434165907)

[6.2.5 Maven 0](#_Toc434165908)

[6.2.6 Selenium 0](#_Toc434165909)

[6.2.7 Vaadin 0](#_Toc434165910)

[6.3 Realisierung Funktionaler Anforderungen 0](#_Toc434165911)

[6.3.1 FR-1 - Login/Konto erstellen 0](#_Toc434165912)

[6.3.2 FR-2 - CRUD User Data 0](#_Toc434165913)

[6.3.3 FR-3 - CRUD Garden 0](#_Toc434165914)

[6.3.4 FR-4 - CRUD Garden Elements 0](#_Toc434165915)

[6.3.5 FR-5 - Garden Health 0](#_Toc434165916)

[6.3.6 FR-6 - Stammdaten Verwaltung durch Administration 0](#_Toc434165917)

[6.3.7 FR-7 - Alarmierung 0](#_Toc434165918)

[6.3.8 FR-8 - Forum CRUD 0](#_Toc434165919)

[6.3.9 FR-9 - Stammdaten Verwaltung durch Benutzer 0](#_Toc434165920)

[6.3.10 FR-10 - Garten exemplarisch gestalten 0](#_Toc434165921)

[6.4 Abhängigkeiten und Einflüsse 0](#_Toc434165922)

[6.5 Systemarchitektur 0](#_Toc434165923)

[6.6 Rahmenbedingungen 0](#_Toc434165924)

[6.7 Nutzen Revisionsplan 0](#_Toc434165925)

[6.8 Testkonzept 0](#_Toc434165926)

[6.9 UI 0](#_Toc434165927)

[6.10 GUI Mockups 0](#_Toc434165928)

[6.11 Technische Ressourcen 0](#_Toc434165929)

[6.12 Projektkosten & Nutzen 0](#_Toc434165930)

[6.13 Risiken 0](#_Toc434165931)

[6.14 Lieferobjekt Katalog 0](#_Toc434165932)

[7 Phasenplan 0](#_Toc434165933)

[8 Auftragsbestätigung - Variantenentscheid 0](#_Toc434165934)

[9 Anhang 0](#_Toc434165935)

# Einleitung

Die Urban Gardening Group, nachfolgend UGG genannt, ist durch den Zusammenschluss von drei Studierenden während des Fachhochschul Studiums entstanden. Während des 5ten Semesters wird von der UGG ein Garden Designer entwickelt. Das vorliegende Pflichtenheft liefert eine genauere Spezifikation der Arbeit, welche während dem Modul Projektmanagement entwickelt wird.

## Leserkreis

Das Pflichtenheft richtet sich an die Projektleitung, den Lenkungsausschuss des Projekts sowie an die betreuenden Dozenten des Moduls PA\_5, in welchem diese Projektarbeit durchgeführt wird.

## Umfang

Innerhalb des Pflichtenhefts werden die geforderten Leistungen aus dem Lastenheft übernommen und in zwei Varianten übersetzt. Die Varianten werden dem Auftraggeber mit einer Vollständigen Bewertung vorgelegt, die Implementation der Arbeit ist allerdings nicht Bestandteil dieses Dokuments.

Ausgehend von den erfassten Anforderungen wird eine lokale Client Lösung sowie eine Webbasierte Anwendung spezifiziert. Beide Varianten decken die im Lastenheft vorgegebenen Anforderungen vollständig und sind, sofern nicht ausdrücklich erwähnt, unabhängig voneinander zu betrachten. Es kann sein, dass bestimmte Lösungswege innerhalb beider Varianten vorkommen und sich diese Analog zueinander verhalten.

## Abgrenzung

Für beide Varianten werden keine Schulungskonzepte erstellt. Aus Zeitgründen wird zudem lediglich einer der Beiden Lösungsvarianten implementiert, der Variantenentscheid ist abhängig von der Bewertung und dem finalen Entscheid des Auftraggebers. Sollten weitere Änderungen an den Anforderungen vorgenommen werden, sind diese nicht zwingenden Bestandteile der Lösungsbeschreibung – es ist dem Auftraggeber überlassen, ob und zu welchem Zeitpunkt die Anpassungen gemäss formalem Änderungsantrag in die finale Implementation einfliessen.

# Projekt Übersicht

Im Rahmen der Variante 1 wird eine Lösung mit Swing erarbeitet, während in der Variante 2 eine Webbasierte „Rich Internet Application“, kurz RIA, erarbeitet wird. Die Spezifikation der Lösungsvariante 1 wird durch Mejdin Hatema vorgenommen und abgeliefert, während die webbasierte Lösung durch Tobias Lanz spezifiziert wird. Die Projektorganisation bleibt dabei unverändert und ist wie nachfolgend definiert.

## Projektorganisation

## Information & Kommunikation

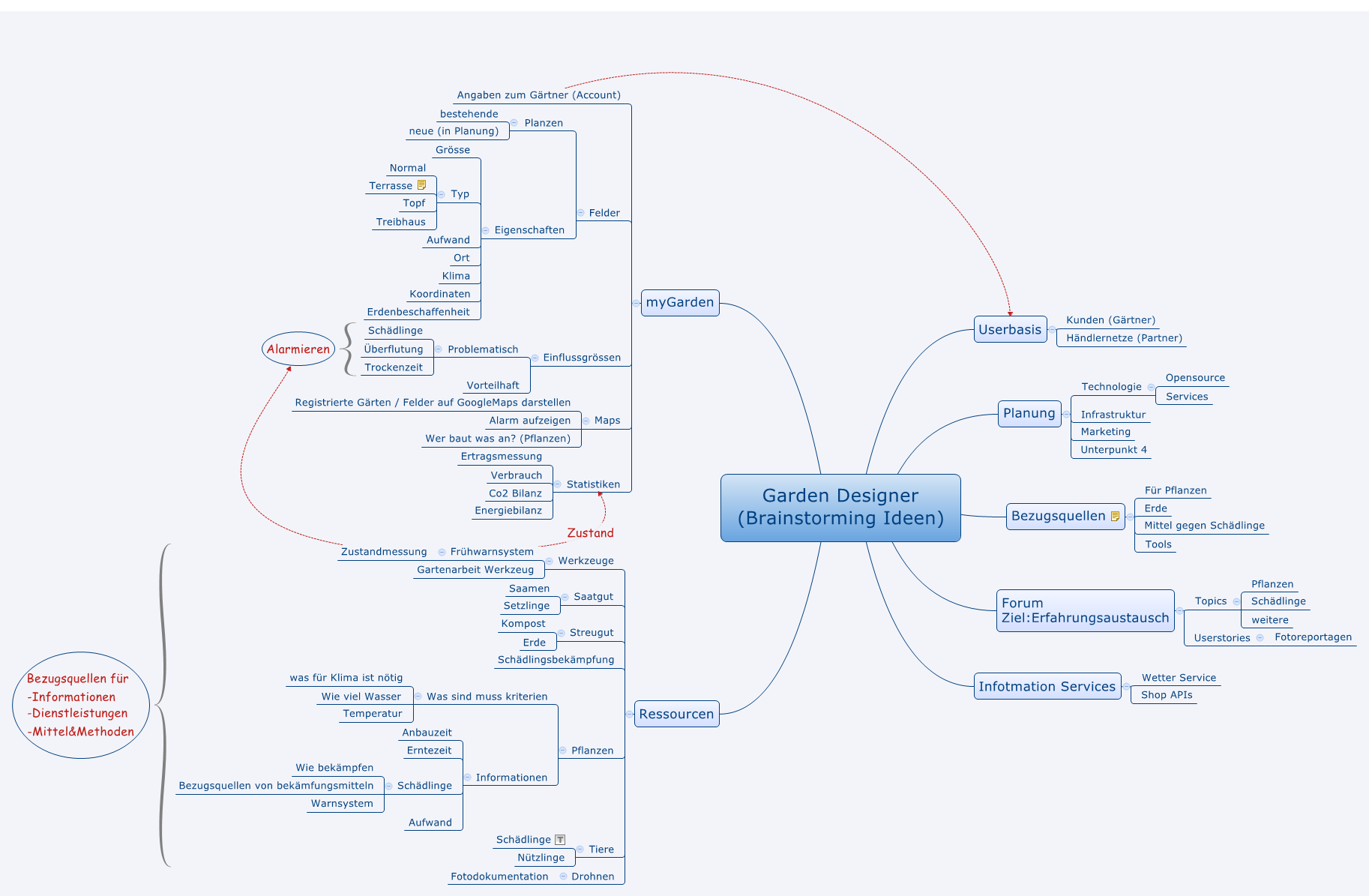
Sämtliche Kommunikation innerhalb des Projektes läuft über Hangout Chats. Für die Projektinvolvierten wurde eine Gruppe gebildet, über welche alle miteinander kommunizieren können. Für spezifischeren Informationsaustausch betreffend wichtiger Themen werden zudem spontan Telefonkonferenzen abgehalten, so dass komplexere Themen effizient und effektiv, also ohne Missverständnisse, diskutiert und abgehandelt werden können. In den regelmässigen Statusmeetings wird zudem fortlaufen der Projektfortschritt gemessen. Diese Meetings finden alle zwei Wochen statt, davon muss mindestens eines im Monat durch Präsenz aller beteiligten an der FFHS gehalten werden.

# Anforderungsanalyse

Während der Projektinitialisierungsphase wurden lediglich sehr grobe Vorstellungen über das Vorhaben besprochen. Im Lastenheft wurden diese Anforderungen konkretisiert und in einem Kriterienkatalog festgehalten. Es gilt nun die Kriterien des Garden Designers noch etwas genauer zu untersuchen, damit eine entsprechende Planung realistisch ist, damit der Aufwand des Vorhabens abgeschätzt werden kann. Im Lastenheft finden sich sämtliche Anforderungen, welche für beide Lösungsvarianten zutreffend sind.

## Ursprüngliche Ausgangslage

Ausgehend von der Ursprünglichen Mindmap der UGG wurden die Anforderungen entnommen. Diese Übersicht sollte dem Leser dabei hilfreich sein, wenn er sich schnell über die Zielformulierungen betreffend des Garden Designers informieren möchte. Ein detailliertes Bild des zu erwartenden Prototypen für den Garden Designer wird im jeweiligen Abschnitt „Lösungsvariante“ gegeben.



# Lösungsvariante 1 - Swing Prototyp

Für eine Desktop Lösung mit Swing haben alle Teammitglieder das nötige Knowhow. Diese Lösung ist mehr oder weniger ein lokales Konzept, welche auf dem Client installiert wird. Ein Fernzugriff ist nur über Fernzugriffssoftware möglich, wie „Remote Desktop“. Der Auftraggeber hat sich eine Weblösung gewünscht. Da diese Variante dem Wunsch des Auftraggebers nicht entspricht, ist diese als solche weiter nicht zu beachten.

# Lösungsvariante 2 - RIA Prototyp

Hinsichtlich der gestellten nicht funktionalen Anforderungen sind viele verschiedenen Technologien für den Prototypen Denkbar. Während in der Variante 1 ein sehr lokales Konzept vorgetragen wird, welches sich direkt auf dem jeweiligen Client installiert um die funktionalen Anforderungen zu decken, entspricht die Variante 2 vielmehr einem Verteilten Lösungsansatz. Entscheidend für einen erfolgreichen RIA Prototyp sind vor allem die Kenntnisse über die eingesetzte Technologie sowie die Einschätzung der damit verbundenen Risiken.

## Vorarbeiten

Aus bestehenden Projekten von Tobias Lanz kann für die Userverwaltung ein Konzept von Lotus Notes übernommen werden. Die Vorarbeiten beziehen sich dabei lediglich auf einen Service für User Login Daten. Die Vorarbeit eröffnet eine Basis zur Bereitstellung eines OAuth Provider Service, falls dies für die weiteren Vorhaben des Projektes als Sinnvoll erachtet wird. Innerhalb von Lotus Notes werden mittels eines Sekundären Directory Useridentifikationsdaten mit folgenden Daten bereitgestellt:

* Username
* Email
* Password
* Rolle

Der Service liefert lediglich eine Social ID, über welche ein User identifiziert und in einer externen Applikation über REST berechtigt werden kann.

## Mittel & Methoden

Innerhalb der Variante werden diverse Mittel beigezogen, damit die Arbeit am GardenDesigner effizient und effektiv vorangetrieben werden kann. Betreffend Effektivität sind vor allem bestehende Applikationen und Bausteine angesprochen, welche nicht durch die UGG entworfen werden müssen. Dabei werden sämtliche Lizenzrechtlichen innerhalb der Variante beschrieben und berücksichtigt. Der Auftraggeber hat keine weiteren Kosten durch die Verwendung, sofern durch den Auftraggeber nicht explizit darauf hingewiesen wird.

Mit Effizienz sind vor allem die von den Entwickler verwendeten Tools zur Erstellung der Applikation zu verstehen. Anlässlich einer allfälligen Übernahme des Betriebs der Applikation sind diese zur Vollständigkeit erwähnt.

### Bluemix

IBM® BluemixTM ist eine von IBM® im Zuge der SAAS, PAAS oder IAAS Bewegung gegründete Plattform. Sie wird innerhalb der Variante dazu verwendet einen Kostenlosen RIA Prototypen zu entwickeln, welcher speziell den Nicht Funktionalen Anforderungen gerecht werden soll. Die UGG verfügt während des Projekts über ein sehr geringes Budget. Die Verwendung der Plattform ist bis zu einem bestimmten Limit nicht mit Kosten verbunden, was für die Auftragnehmer daher durchaus von Vorteil ist. Selbstverständlich sind weitere IAAS Dienstleister innerhalb des World Wide Web möglich. Die Evaluation eines weiteren geeigneten IAAS Dienstleisters ist nicht Bestandteil dieser Variante, weil die Maturität für weitere IAAS Anbieter innerhalb der UGG nicht vorhanden ist. Die entwickelte Applikation kann durchaus auch auf andere Cloud Service Dienstleister übertragen werden, sofern das Wissen vorhanden ist. Ebenfalls ist eine ganzheitlich selbst gehostete Lösung dadurch nicht ausgeschlossen.

<https://console.ng.bluemix.net/>

### DevOps (<https://hub.jazz.net>)

Während der Entwicklung werden von der UGG wahlweise sowohl lokale wie auch Cloudbasierte Dienste in Anspruch genommen. DevOps ist eine von IBM® bereitgestellte Entwicklungsumgebung für die Cloud. Sämtliche Coding Aktivitäten können auf theoretischer Basis durch DevOps abgebildet werden, daraus resultiert allerdings nicht zwingend eine Abhängigkeit für den jeweiligen Entwickler, da die Entwicklungsumgebung durch die Verfügbarkeit von Git auch auf Lokale Unterstützungen übertragen werden kann.

<http://www.ibm.com/ibm/devops/us/en/>

### Cloudfoundry

Zur Bereitstellung des Prototypens innerhalb der IBM® BluemixTM Cloud wird Cloudfoundry verwendet. Cloudfoundry ist eine von mehreren Grossunternehmen vorangetrieben Technologie zur Propagierung von Software innerhalb einer Cloud.

Dabei wird die fertig gebaute Applikation, repräsentiert durch ein WAR File, auf die jeweilige Cloud Instanz des Entwickler publiziert. Die Bereitstellung des Prototyps ist dabei Bestandteil der Implementierung, ausgeschlossen ist dabei allerdings die Anleitung, weil sich innerhalb von BluemixTM oder anderen Quellen derartige Artikel bereits finden.

<https://www.cloudfoundry.org>

### Git

Damit die Entwickler gemeinsam an der Applikation arbeiten können und kein Fortschrittsverlust resultiert, wird GIT eingesetzt. Da GIT eine gängige Source Control Möglichkeit darstellt, wird hier auf eine ausführliche Erklärung verzichtet.

<https://git-scm.com/>

### Maven

Maven wird innerhalb der Variante 2 zur Handhabung von Abhängigkeiten verwendet. Dabei stellt es aber nicht ausschliesslich ein Dependency Management Tool dar, es wird auch dazu verwendet entsprechende Build Phasen während der Entwicklung zu unterstützen. Durch Maven werden die Unit und Komponententests abgebildet.

<https://maven.apache.org/>

### Selenium

Abschliessende Integrationstests werden mit Selenium durchgeführt. Voraussetzung dazu ist, dass mittels Maven bereits sämtliche dazu benötigten Unit und Komponenten Tests Fehlerfrei durchlaufen sind. Selenium ist bestens dazu geeignet, um RIA Abnahmetests zu realisieren.

<http://www.seleniumhq.org/>

### Vaadin

Bisher wurden lediglich Technologisch bedingte Mittel und Methoden eingeführt, welche zur Entwicklung benötigt werden. Das Userinterface für die Applikation wird - wie aus dem Prototypen Namen bereits abzuleiten ist - durch eine Rich Internet Application, kurz RIA, abgebildet. Die Evaluation weiterer RIA Frameworks wäre durchaus plausibel und Sinnvoll, fällt jedoch durch nachfolgende Punkte aus:

1. Eine weitere Variante wird durch Swing bereits als gegeben erachtet.
2. Innerhalb der UGG verfügt kein Mitglied vertieftes Wissen von einem anderen RIA Frameworks. Tobias Lanz befasst sich während seiner Arbeit ebenfalls mit Angular JS. Da dieses Framework jedoch auf JavaScript basiert, darf aufgrund der vorgegebenen Anforderung aus dem Lastenheft nicht damit gearbeitet werden.

<https://vaadin.com/home>

#### Add-ons

Innerhalb der Applikation wird für bestimmte Anwendungsfälle bestehender Code durch Code Injection herbeigezogen. Dabei wird, wie bereits erwähnt, darauf geachtet das keine weiteren Kosten auf Basis von Lizenzen entstehen.

<https://vaadin.com/directory#!browse>

Nachfolgend findet sich eine Liste der verwendeten Add-ons innerhalb des Garden Designer Prototyps:

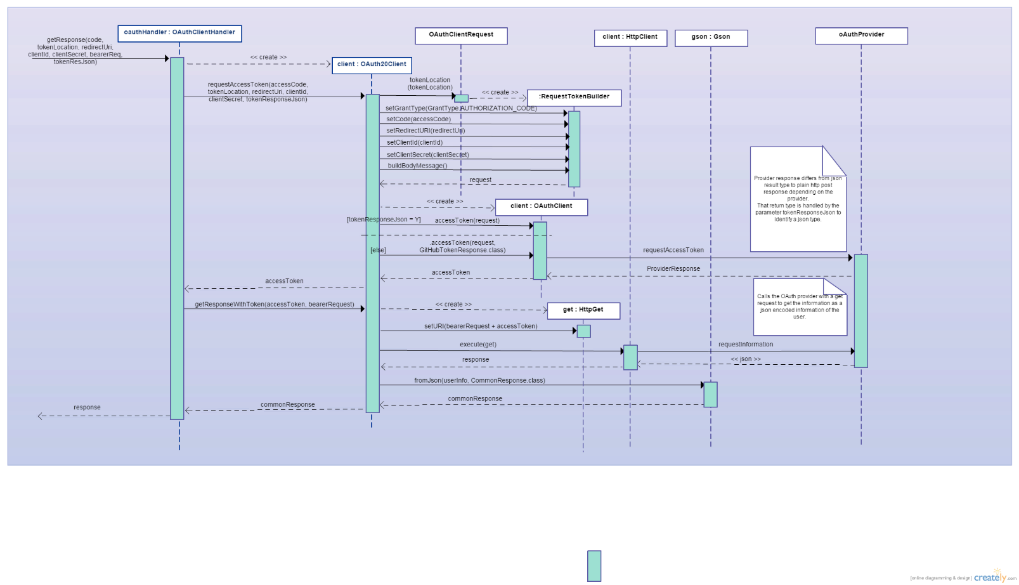
## Realisierung Funktionaler Anforderungen

Nachfolgen werden die Lösungsvorschläge für die definierten Funktionalen Anforderungen wiedergegeben.

### Business Logic Komponenten

### FR-1 - Login/Konto erstellen

In der Variante wird mittels Oauth 2.0 Client gearbeitet. Die Lösung befindet sich in der Cloud und soll keine kritischen User Daten verwalten. Garteninformationen werden nicht als kritisch betrachtet und stellen gemäss Auftragnehmer keine erweitert zu schützende Information dar, Login Daten jedoch schon. Daher wird innerhalb der Variante 2 darauf verzichtet, Informationen zu Personen zu speichern. Die Identifikation zu einem Garten wird lediglich durch eine Social ID dargestellt. Die Social ID ist die einzige kritische Information, welche innerhalb der Cloud persistiert wird und durch erweiterte Sicherungsmassnahmen zu schützen ist. Nachfolgend findet sich ein offizielles Schema des OAuth 2.0 Client Standard, welcher in der Applikation realisiert wird.

 Mittels bestehender OAuth Provider wird lediglich die Useridentifikation gelöst. Weitere Social Media Inhalte via OAuth einzubinden ist theoretisch möglich, dieser werden jedoch für den Prototypen nicht realisiert.

Usecase Erweiterung (out of Scope):

1. Social Data Import
   1. Facebook
   2. LinkedIn
   3. Twitter
   4. Xing

### FR-5 - Garden Health

Für die Statistiken des Garten, welche zur Betrachtung der Gartengesundheit nötig sind, kommen mehrere Varianten in Frage. Aufgrund der Kosten für die Charting Software innerhalb von Vaadin werden mehrere Lösungspermutationen erwähnt.

1. Remote JS Statistics Engine
2. Remote Java Statistics Engine,
3. Vaadin Charting Software

***Implementiert werden aus Zeitgründen keine externen Clients. Daher fallen Permutation 1 und 2 aus.***

Die Architektur für den Health Bereich wird aber darauf ausgelegt Remote Clients zu unterstützen. Das Rest Layer wird innerhalb dieser Variant implementiert, jedoch kein Client.

Da es sich um einen Prototypen handelt, geht Tobias Lanz nicht von Lizenzrechtlich problematischem Einsatz aus. Allfällige Kosten betreffend dieser Angelegenheit übernimmt Tobias Lanz.

### FR-6 - Stammdaten Verwaltung durch Administration

Wenn die Stammdaten Verwaltung innerhalb der Applikation vorgenommen werden soll, bedeutet dies Automatisch Rollenbasierten Zugriffsrechte zu gewähren – diese können unteranderem mit Shiro[[1]](#footnote-1) oder eigen Implementationen gemacht werden –aktuell wird dieser Ansatz als sehr unrealistisch eingeschätzt, wenn der Release auf den 18.12.2015 fertiggestellt werden soll.

Die Stammdaten werden zu Beginn statisch erfasst.

### FR-7 – Alarmierung

Die Alarmierung kann durch IBM Bluemix Push Support schnell auf viele Smartphones propagiert werden. Dadurch würden allerdings zusätzlich kosten anfallen, welche erwähnt sein müssen.

Wenn an andere Übermittlungssysteme wie Email oder SMS gedacht wird, muss berücksichtigt werden das auch dafür zusätzlich kosten anfallen. Beide Dienste können ähnlich wie der Push Support als Dienstleistung eingekauft werden.

Erneut stellt ein Usecase aus Zeitkritischen Gründen

### Entity Persistenz

Alle Elemente des Garden Designers werden durch JPA Annotierte Beans von Services abgegriffen. Dazu müssen die Entitätsklassen entsprechend mit folgenden Annotationen geschrieben sein:

Class  
@Entity

@NamedQueries

ID Field

@Id  
@GeneratedValue(strategy = GenerationType.AUTO)  
  
weitere @Column, @Enumerated

Nachfolgend werden die registrierten CRUD Usecases für den Graden Designer beschrieben. Für alle diese Entitäten gilt obiges. Zusätzlich sollen alle entitäten mit einem Zeitstämpel versehen werden, damit der Erstell Zeitpunkt festgelegt ist und die letzte Änderungszeit dokumentiert ist.

### FR-2 - CRUD User Data

Da innerhalb der Applikation keine kritische Userdaten verwaltet werden sollen, sofern der User dies nicht ausdrücklich wünscht, werden keine persönlichen Daten innerhalb der Applikation in Bezug auf den eigenen Garten publiziert.

Usecase Erweiterung (out of Scope):

Der User kann über einen Import Button und die Ausdrückliche Zustimmung seine Personen Daten aus dem benutzen OAuth Profile herbeiziehen. Dabei bestätigt der User explizit, dass seine Daten in der Cloud gespeichert und gegebenenfalls publiziert werden.

### FR-3 - CRUD Garden

Der Garten ist eines der Komplexeren Objekte innerhalb des Garden Designers. Für ihn wird zusätzlich zur spezifischen Service Persistence Unit eine erweiterte, spezialisierte Persistence Unit names bereitgestellt. Diese wiederum behandelt

### FR-4 - CRUD Garden Elements

Innerhalb des Gartens befinden sich Felder. Innerhalb dieser Felder können Pflanzen angepflanzt werden. Der Garden Designer unterstützt den Gärtner bei der Organisation seiner Pflanzen.

### FR-8 - Forum CRUD

### FR-9 - Stammdaten Verwaltung durch Benutzer

### FR-10 - Garten exemplarisch gestalten

## Abhängigkeiten und Einflüsse

* Cloud Deployment Technologie
* Plugin Availability
* Dependency Injection

## Systemarchitektur

## Rahmenbedingungen

## Nutzen Revisionsplan

## Testkonzept

## UI

## GUI Mockups

## Technische Ressourcen

## Projektkosten & Nutzen

## Risiken

## Lieferobjekt Katalog

# Lösungsvariante 3 – PHP & MySQL Prototyp

Der Auftraggeber erwartet eine Weblösung, aufgrund dessen, zeigt sich die PHP Variante auch als eine Möglichkeit. PHP wird fast immer oder sehr oft in Kombination mit MySQL verwendet, deswegen wird diese Variante als Paket von PHP & MySQL analysiert.

## Vorarbeiten

Um die Variante zu realisieren, brauchen wir einen PHP Provider, wo wir die Applikation hochladen und testen können. Eine Domäne und Platz auf einem Webserver steht zu Verfügung. Der Provider ist Kreativ-Media.

## Git

Wie in der 2. Variante, für die Versionskontrolle wird GIT verwendet.

## Eclipse

Eclipse bietet eine komfortable Entwicklungsumgebung und deckt alle notwendigen Funktionen, welche wir mit benötigen, um die UGG erfolgreich abzuschliessen.

## PHP Unit

PHP Unit ist ein Framework zum Testen von PHP-Skripten. Es ist besonders für automatisierte Tests einzelner Einheiten (Klassen, Methoden) geeignet. Es basiert auf dem xUnit-Konzept, welches auch für andere Sprachen genutzt wird, wie zum Beispiel jUnit für JAVA.

## Vorteile

1. Keine Installation nötig. Weder bei Client noch auf dem Server. Der Provider stellt die installierte Version von PHP und MySQL zu Verfügung.
2. Die Belastung des Servers, auf dem PHP läuft, ist gering. PHP ist als Modul auf dem Server eingebunden, das bedeutet, dass PHP ein Teil des Servers ist und immer geladen ist. Somit entfällt der Overhead der Prozesserzeugung.
3. Der Inhalt (Content) einer Webseite und die Art der Präsentation sind getrennt. Die Daten, des Inhaltes werden in MySQL gespeichert, während das Layout in Dateien als „Snippets„ deponiert sind.
4. PHP beinhaltet schon in seiner grundlegenden Form eine sehr große Anzahl verschiedener Bibliotheken und Funktionen. Besonders hervorzuheben sind hier Protokolle und Datenbankanbindungen. Durch diese Vielzahl an Schnittstellen bringt PHP bereits alles mit, was zur Implementierung einer Web-Applikation notwendig ist.
5. PHP ist Browser unabhängig.
6. PHP läuft auf Linux/Unixt und auf Windows. Das bietet einen grossen Vorteil, da es sehr leicht ist, bei geänderten Anforderungen auf andere, leistungsfähigere oder sichere Systeme umzusteigen.
7. PHP ist kostenlos. Es fallen bei der Benutzung von PHP keinerlei Lizenzgebühren an.
8. PHP ist Open Source Software und kann damit die gesamten Vorteile, die dieses System mit sich bringt, für sich nutzen.

## Nachteile

1. Da PHP eine interpretierte Skriptsprache ist, ergibt sich ein Geschwindigkeitsnachteil bei der Ausführung von PHP gestützten Seiten. Vor jedem Durchlauf des Programms muss das Skript neu übersetzt werden, was sich speziell bei stark unter Last betriebenen Servern bemerkbar machen dürfte. Für unsere Applikation stellt das kein grosses Problem dar, da sie nur ein Last-Test von 5-10 User bestehen muss.
2. Quellcode ist offen. Bei der Weitergabe der Applikation ist der Quellcode nicht geschützt, da die Skripte aus reinen Textdateien bestehen.

# Phasenplan

# Auftragsbestätigung - Variantenentscheid

# Anhang

1. <https://github.com/vaadin-kim/shiro-example> [↑](#footnote-ref-1)