

## **Begründung der gewählten Programmiersprachen:**

Grundsätzlich ist eine Vorgabe des Projektrahmens, dass das Projekt mit den Programmiersprachen Java und/oder JavaScript realisiert werden soll. An dieser Stelle wird in kurzer Form zwischen diesen Alternativen auf Client und Server Seite abgewogen.

### **Clientseite:**

Grundsätzlich besteht die Möglichkeit den Client in Java oder JavaScript zu realisieren. Da der Nutzer in unserem Projekt mobil ist, wie in der Contextual Task Analysis ergeben und somit das System auf einem mobilen Endgerät laufen soll, stehen hier die zwei Alternativen zur Verfügung. Zum einen eine reine Webapp, welche auf dem mobilen Endgerät aufgerufen wird und in JavaScript programmiert wird, oder eine native App in Java basierend auf dem Android Betriebssystem. Da wir jedoch Standortinformationen der Benutzer benötigen und ggf. mit Bildern arbeiten und diese lediglich durch Realisierung einer nativen App, durch den Zugriff auf Ressourcen des Betriebssystems möglich ist, ist die Wahl auf Java für den Client gefallen.

### **Serverseite:**

Hier besteht ebenfalls die Möglichkeit den Server in Java oder in Javascript zu implementieren. Die Vorteile eines Servers mit Java sind:

- Große Community
- Große Auswahl an Libraries
- Effizient im Hinblick auf CPU intensive Berechnungen
- Gute Auswahl an IDEs

Als Nachteil steht dem gegenüber, dass Java in der Implementierung sehr zeitaufwendig ist und die Effizienz auch Abhängig von der Ordentlichkeit der Implementierung ist.

Die Vorteile eines Servers mit JavaScript und Node.js sind:

- Schneller Entwicklungsprozess
- Einfachere Implementierung
- Große Auswahl an Modulen
- Bereits erworbene Erfahrung
- Gut auf eine hohe Benutzeranzahl skalierbar

Dem stehen die Nachteile gegenüber, dass es im Vergleich zu Java eine noch nicht so gereifte Technologie ist und, dass es nicht so viele IDEs für die Entwicklung gibt.

Da unser Vorgehen aus einer iterativen Vorgehensweise besteht und wir einen schnellen Entwicklungsprozess benötigen, welcher auf eine hohe Benutzerzahl skalierbar sein muss haben wir uns für die Entwicklung mit JavaScript und Node.js entschieden. Die überwiegenden Vorteile von Javascript und Node.js und die erworbene Erfahrung bestärken ebenfalls diese Vorgehensweise.