

# Matching Algorithmus

## Pseudo Code

### 1. Standort festlegen:

Standort des Nutzers feststellen oder eine Eingabe des Standorts zulassen.

```
Get currentLocation() or readline(Standort)
```

```
return val standort
```

### 2. Veranstaltungen in der Nähe finden:

Das System soll in einem Radius (veränderbarer Standardwert) um den Standort des Nutzers alle Veranstaltungen feststellen und eine Liste erstellen.

```
var radius = 30 km
```

```
or
```

```
readline (radius) = radius
```

```
Nearbyveranstaltungen(standort,radius)
```

```
//Mögliches Tool zur Radius Suche ist die Geometry Library der Google Maps API
```

```
return listof(veranstaltung_nähe)
```

### 3. Matching:

Mit allen Veranstaltungen, die im gewählten Radius des Standorts des Nutzers liegen, wird das Matching begonnen. Im Prototypen soll nur nach den gewählten Oberkategorien gematcht werden, wenn die Anzahl an Veranstaltungen zunimmt könnte man dies durch einen String-Algorithmus erweitern. Dieser soll die Veranstaltungsnamen vergleichen, um die Matching-Möglichkeiten zu verbessern.

Oberkategorie wird als Variable beim Veranstaltung Erstellen aus einer Reihe Vorgaben ausgewählt.

```
Matching (veranstaltung_nähe) {  
  Set Interessen(User.profil){  
    Val Oberkategorien = „Sport“  
  }  
  listsearch( listof(veranstaltung_nähe), var Oberkategorien)  
  return listof(sport_ veranstaltung_nähe)  
  // sport_ veranstaltung_nähe ist die Liste der Matches in der Oberkategorie Sport  
  // es könnten mehrere Oberkategorien durchgearbeitet werden  
  
  Matches = listof(veranstaltung_nähe) paired with all Oberkategorien  
  return Matches  
}
```

#### **4. Einladung versenden:**

Bei einem Match soll der Nutzer eine Einladung erhalten die er annehmen oder ablehnen kann.

```
Einladung_senden(Match) {  
  For each (Match) send invite(User)  
}
```

#### **5. Einladung annehmen/ablehnen:**

Wenn der Nutzer die Einladung annimmt wird er der Veranstaltung hinzugefügt und die Teilnehmeranzahl wird um eins inkrementiert. Sonst wird die Einladung verworfen.

```
Einladung_user () {  
  If user_acception (true) {
```

```
        add (User) to Veranstaltung
        and Veranstaltung.teilnehmerzahl ++ }
if user_acception (false) {
delete (Einladung ) }
}
```