**Search Result Ranking Criteria:**

Gurus are ranked from 1-100 based on the following weights. Based on this the

gurus would be shown.

1. Featured Guru (or Paid) Guru (Weight - 50 points)

2. Social Guru (Connected in Facebook) Dynamic (can be difficult) (1st level - 10,

2nd level 7, 3rd level 5 points) OR static

4. Active Guru (Number of invitations sent in last month 10, last login, number of

enquiry received)

5. Recommended Guru (20) (10 for testimony, 10 for endorsement)

6. Complete Guru (10) ((a) Wordings less than 100) (b) Photos

7. Verified Guru (10) (Email Verified, Phone Verification)

Can be difficult (we can do sorting later on based on the zipcode distance)

8. Local Guru (within City - 10 points, outside city 0 points)

(note: seq. no. 3 is skipped in original SRS)

**Mathematical Calculation for the above mentioned rank criteria while searching service:**

Step 1: Search in sphinx to find IDs of services first. Max IDs picked up for any search criteria and location criteria will be 10K.

i.e. if someone looking for doctor in minto park he will get to see only 10K result which is sufficiently large.

Step 2: Compose comma separated IDs in PHP for use in MySQL Query as “IN” clause

Step 3: Compose Order by SQL like this -

Select id from service\_table s

where s.id IN ($id\_csv)

Order By

IF(s.featured,5\*108,0) +

CHECK\_FB\_LEVEL($visitor\_fb\_id, s.fb\_L1\_id\_csv,9x107,s.fb\_L2\_id\_csv,7x107,s.fb\_L3\_id\_csv,5x107)+

(

s.no\_of\_company\_service\_provider\_added\_this\_month+

(30-Max(30,s.day\_since\_last\_login))+

s.no\_of\_person\_contacted

) \* 105+

(s.no\_of\_testimony+s.total\_no\_of\_service\_providers\_endrosement)\*104+

(s.percentage\_of\_service\_profile\_completed+s.percentage\_of\_all\_service\_providers\_profile\_completed)/2\*104+

(

Max(s.verified\_guru\_count,7)+

IF(s.owner\_email\_verifed,1,0)+

IF(s.owner\_phone\_verifed,1,0)

)\*103

;

Note: Store user day\_since\_last\_login with service\_table as well. That field will update if any user attached to that service login.

Note: total\_no\_of\_service\_providers\_endrosement column is required in the service table

Note: Two columns are required in the service table - percentage\_of\_service\_profile\_completed, percentage\_of\_all\_service\_providers\_profile\_completed

Note: verified\_guru\_count is required in service table that depicts total of attached service providers account is verified.

**Rank Table** (This will be the implementation)

**service\_id**

**user\_id**

**featured** = 0,1

**fb\_level** = 0,1,2,3 --- 0 when userid is 0

**active\_level** = combination of above 3 = s.no\_of\_company\_service\_provider\_added\_this\_month+

(30-Max(30,s.day\_since\_last\_login))+

s.no\_of\_person\_contacted

**end\_recommended** =

s.no\_of\_testimony+s.total\_no\_of\_service\_providers\_endrosement

**completion**=

s.percentage\_of\_service\_profile\_completed+s.percentage\_of\_all\_service\_providers\_profile\_completed)/2

**verified**= Min(s.verified\_guru\_count,7)+

IF(s.owner\_email\_verifed,1,0)+

IF(s.owner\_phone\_verifed,1,0)

index (user\_id, featured,fb\_level,active\_level,end\_recomended,completion,verified,service\_id)

select service\_id from rank table

where

userid=$visitor\_user\_id or userid=0

order by featured,fb\_level,active\_level,end\_recomended,completion,verified