**CHAPTER 4**

**PROJECT REQUIREMENTS DEFINITION**

**4.1 Project Perspective**

Users get themselves registered to our system. During this process of registration, users create profiles of their own which also includes their id from the social networks. With the help of this id that is provided by the users, their data from social networks is fetched, analyzed and stored in a format called as User Description Format (UDF).

Once the user inputs the context, our system identifies the stored UDFs which satisfy the constraints and conditions of the given context. The given context has to be relevant to the user and also has to provide some kind of benefit to the user.

Once the users are registered with our system and the UDFs are generated, and the user enters a brief description about the context, the system identifies the contexts which are congruent with the UDFs and recommends the same to users. The user can either accept it or reject it. These contexts which are accepted or rejected are stored along with UDF and they can be reused.

When the users input contexts to our system, the UDF may not have the necessary attributes. In such situations, the user is also responsible to enter the attributes that are required for their contexts. We recommend these contexts to those users who are congruent with some of the contexts defined by this user. If the users have recently registered with our system, then we recommend these contexts to the people who are congruent with the contexts which have some of the attributes of newly defining context.

Once the UDF is defined and updated with the contexts, generating a congruent network for some given context can be simple as selecting those UDFs that have the given context with them. This is only possible for the predefined contexts. For the newly defined contexts, all UDFs will have to be analyzed and UDFs having the required attributes are selected.

Once the UDFs are refined, the congruence weight is calculated for each UDF. The UDFs which have a congruence weight greater than a threshold (defined by user) are identified to be as congruent.

**4.2 Project Functions**

The major facilities that are provided by our system are the following:

* Creation of Egocentric Networks
* Handling the perception of ego by different people
* Creation of context
* Capturing the ego of a person
* Matching the ego between people
* Matching the ego of a person with the context in which he is in
* Creation of Egocentric Congruent Networks for the given context
* Measuring the degree and stability of the Egocentric Congruent Network

**4.3 User Classes and Characteristics**

The users of this system will be able to register themselves, and thereby create a profile of their own, according to the UDF. The user can also create a context, by entering a brief description about the context, and then can choose the contexts from those that are recommended to the user, or the user can give a term for the context. Similarly, the user can also add properties to the contexts.

* 1. **Operating Environment**

4.4.1 Hardware requirements

* Processor : Intel Core i3 and above
* RAM : 2GB and above
* Hard disk : 20GB and above

4.4.2 Software requirements

* Operating System : Windows or linux
* Database : MongoDB
* Language : Python 2.7
* Access to different social networking sites
* Python packages : Pymongo, NLTK and BeautifulSoup
* Browser : Firefox 20.0.1 and above

**4.5 Design and Implementation Constraints**

All users must be connected with the system in all the appropriate social networks. In this project, Facebook, Twitter and LinkedIn are used as the primary social networks. Social connections are treated differently by different social networks. Facebook interprets people as friends whereas Twitter interprets as followers and LinkedIn as connections. Hence, the user must be a friend of our system in Facebook, a follower on Twitter and a connection on LinkedIn.

**4.6 User Documentation**

For this project to be effective, users should be able to express their requirements as contexts. The system will provide users with guidelines in order to convert the requirements into corresponding contexts. As contexts are expressed in terms of attributes, sample contexts defined with attributes will be provided.

**4.7 Assumptions and Dependencies**

Users will portray themselves differently in different social networks. This portrayal of the user will depend on the purpose that is intended and also on the benefits that can be reaped from the various social networks. Here representation is the quantity of information which the user is willing to show to others who are in the same social network.

Since the users can have different levels of benefits from this system, the strength of their connection with the system will also vary. Based on the strength of the connection the users will have, recommendations can be made. Recommendations, once done, will have to be beneficial to the users. As recommendations can be beneficial, we assume that the users will accept the recommendations made. The decision to accept or reject the recommendations, solely depends on the choice of the users.

**CHAPTER 5**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**5.1 External Interface Requirements**

5.1.1 User Interfaces

* Interface for creating new contexts or using existing contexts
* Interface for registering new users into the system
* Interface for changing the profile of a user
* Interface for viewing generated statistics

5.1.2 Software Interfaces

* Operating system : 64–bit linux 2.4 or above
* Database : MongoDB
* Language : JyThon (java Python)
* PyMongo package for python – this also requires nose, setup tools and sphinx packages for python
* NLTK package for python
* BeautifulSoup package for python
* Web 2.0 standards

**5.2 Functional Requirements**

5.2.1 Generation of egocentric networks, for each node

* For individual nodes, egocentric networks are generated, taking into consideration only the attributes of that particular node.
* Each ego network can be visualized as a star network.
* The attributes of the central node is called as distinguishing attributes of the egocentric network in that social network.

5.2.2 Elimination of egocentric networks based on the required context

* Some of the egocentric networks that do not relate with the required context are eliminated.
* This is done by matching the attributes of the context, that are derived from the context tree, and the distinguishing attributes of the individual egocentric networks.

5.2.3 Reduction in the number of nodes of each egocentric network, based on the context

* The number of nodes in the remaining egocentric networks is reduced.
* This is done by matching the attributes of the individual nodes with that of the context, and this gives us a weight.
* Those nodes whose weights are below a threshold are eliminated.
* This results in context specific egocentric networks.

5.2.4 Generation of egocentric congruent network

* From the remaining ego networks that belong to that context, all the nodes in all the networks are merged to generate the egocentric congruent network.
* Else, the most suitable context specific egocentric network can be suggested as egocentric congruent network.

5.2.5 Measurement of the degree and stability of the egocentric congruent network

* Once the egocentric congruent networks are generated, the degree and stability of the network is measured in order to understand the characteristics of the network.
* The degree can be measured by determining the level of congruence between the nodes in the network.
* The stability of the network is measured by determining the changes that happen in the network in the lifetime of the context.
* Higher the variations that happen in the network, lower is the stability of the network.

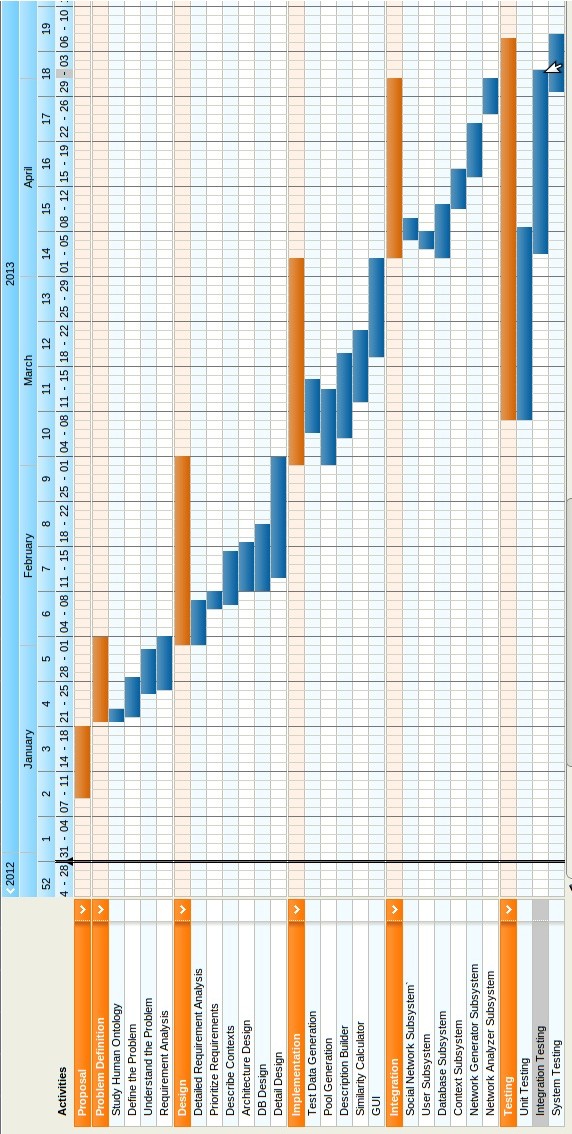
**5.3 Non functional requirements**

Since this project is web based and it involves various social networking sites, the system has to meet many non functional requirements. But, due to the limitation of time and infrastructure, our implementation has been restricted to meet the following non functional requirements:

* Implementation is capable of handling up to 10,000 user profiles.
* Should support at least 100 simultaneous users.
* Should be capable of handling at least 100 to 1000 contexts.
* The response time for generation of the egocentric congruent network should be less than 3 minutes.
* Profiles that have been changed by the users, should be updated within a time of 24 hours.

**CHAPTER 6**

**GANTT CHART**

****

***Fig. 6.1*** *Gantt chart*