2011

Social Analytics Visualization

[Jung Layouts]

Project Supervisor

Dr. Kai Xu

Dept. of Computing & Multimedia Technology, Middlesex University, London

Nakul Sharma

Module Code: CMT4141

Student Number: M00333742

Acknowledgement

I would like to convey my heartiest Thanks to each and every one who was involved in this project and contributed either actively of passively.

First & Foremost, my deepest gratitude to my Project supervisor **Dr. Kai Xu** (Senior Lecturer, Dept. of Computing & Multimedia Technology, Middlesex University, London) who provided persistence guidance as and when I required. I learned a lot under his guidance and supervision and would like to seek future opportunities to work under him again.

Many thanks to **Dr. Carl Evans** (Director of PG Studies, Dept. of Computing and Multimedia Technology, Middlesex University, London) who supported me throughout the MS program and always encouraged me to perform well. Java (Object Oriented Programing Concepts) taught by him was used in this project extensively and was very useful.

Also, I would like to convey my thanks to **Dr. Ralph Moseley** (Senior Lecturer, Department of Computing and Multimedia Technology, Middlesex University, London) who taught me 'Internet Programming' and provided insight knowledge on the subject.

At the end, Last but not least, **Mr. Ed Currie, Dr. Franco Raimondi, Mrs. Bronwen Cassidy, Dr. Elke Dunker-Gassen and Ms. Nallini Selvaraj** for the affirm guidance they all provide to me throughout the MS Program.

Thanks & Regards

Nakul Sharma

Abstract

Social Networking has gained a tremendous popularity within past few years. With millions of users registered on such social networking sites there flows a huge amount of data every day. For some people this huge amount of information is of no use but for some it's worth to look into. As a small effort towards making the use of this huge information on board a new system is developed using which anyone can visualize the data of their interest in the desired layout and analyse it. This system operates on the database which is created and populated by Central Tweet Collector Application. Developed system can be quite useful for professionals and researchers in the field which involves the need of analysing social networking data.

1 Table of Contents

2 Introduction

Visualizations have played an important role in generating new insights in social network analysis and one such kind of social networking platform is 'Twitter' which is one of the most famous online platforms and provide services like social networking and micro-blogging, having more than 200 million users registered on it. Users share the data in terms of 'tweets' restricted to 140 characters each. With more than 200 million tweets being generated in a single day, there flows a huge amount of information. The idea of visualizing the huge data is not new but the way in which it is visualized is getting better day by day. Many professionals and researchers seek such kind of platform or tool using which a huge amount of data can be visualized and analyzed deeply. Out of many available graphs, layouts and patterns only few were selected which were found to be suitable for analyzing a huge network. One such kind of framework is Java Universal Network/Graph Framework which was used in this project to visualize the Twitter data filtered by the search keywords into different Layouts.

2.1 Project Scope

This project aims to develop a system which visualizes the Twitter data into a selected Network. Twitter data to be visualized will be filtered based on the search keyword provided by the users. Also user will have the option to choose among the different layouts. Visualization of data in this project will be implemented in ISOM Layout, KK Layout, FR Layout and Circle Layout.

Beside this, the system will provide an option to select the number of nodes/users to be represented in the visualized network. All the visualizations will be enabled with features like Rotation, Shearing, Zoom-In/Out and Pickable mode.

2.2 Objective

Some of the key objectives of this project are,

1. Identify and analyse the existing Visualization Methods for Social Network Data.
2. Design & Develop a new system which will represent the Social Network data collected from Central Database in different Layouts.
3. Test & Evaluate the new system by simulating the developed prototype.

The significance of this project is the provision of flexible and more understandable Layouts for novice users.

2.3 Methodologies

This project started with a thorough literature review and collecting a base knowledge on social networks, Visualization techniques and graph layouts. Then a research was conducted to select the visualization technique and graph layouts which need to be plotted using this system.

Then a thorough understanding of Java Universal Network/Graph (JUNG) Framework was achieved before getting on to the actual development of the system.

Upon completion of development of the system, an extensive testing and detailed evaluation was performed to ensure that we did something which we intended to do.

3 Literature review

3.1 Social Networks and Twitter

Social networking from past decade as emerged tremendously. Now it' not only used by individuals but also by corporate and institutions. Eventually, social networking has resulted in an easiest and convenient way to communicate and stay in touch with individuals, groups and communities. Number of users on such social networking platforms is growing exponentially day by day. Some user's use social networking for entertainment and to stay in touch with friends and family and some users use social networking professionally. A variety of users can be found on such social networking platforms. Because of large number of users on such social networks there flows a huge amount of information every day. For some this information is quite a worth and for some it's of no use. Most of the social networking platform now a days are web based where users can register and create an account and can start communicating with other users. Some are paid and some provide this service for free.

Twitter, Facebook, Google+ is few of its kind who are major players in this field. Most of the social networking platforms provide the feature for users to create the profile page where they can have all the related information about them. Other features like posting comments, sharing, status updates and messages are also common among these famous social networking platforms. These are actually the basic features a user looks for on any social networking platform. Most important feature which all social networking platforms provide is connection to other users. Some user makes connections to make friends and for entertainment and some make connections for business purpose. But some user's take it as a goal to have as many as links they can. [1]

Where there are many advantages like entertainment, communication, business growth, exposure etc. there are few disadvantages too. Privacy issue has been in limelight from past few years. Because of this issue many social networking platform providers were made to tighten the data privacy policies for their companies. Another few major concerns are Data Mining, Information leakage and Trolling.

One such famous and most emerging social networking platform is Twitter. Twitter was developed and launched by Jack Dorsey in Jul, 2006. As per Forbes [Web] Currently Twitter claims to have 200 Million registered users on board with around 110 Million Tweets being posted every day [2]. 110 Millions Tweets per day accounts for a huge amount of data which can be utilized by professionals and researchers. And to access this Twitter provides the API's for the developers. Developers can access the information using the API's only if the tweet is set to public by the user. Central Tweet Collector application utilized the same API's to fetch the data from the Twitter.

3.2 Visualization Techniques and JUNG

As it's correctly said that - 'A Picture conveys a lot more than thousand words'. Visualizing data is the best solution one can opt for when needs to analyze or monitor the huge amount of data. Visualizations can also be helpful when one need to study the huge and complex networks. Visualization has been studied and being used from past many decades and with time there has been a tremendous advancement in the techniques being followed to visualize the huge data in different layouts.

Network visualizations are extensively used for analyzing Social Networks. Every social Network has nodes (actors or vertex or Point or Agent) which are nothing but an entity which may represent any individual, organization, city or something else. Then there are Edges (Link or Tie or Line or Arc) which represents the connection or relationship between two Nodes. Furthermore, any network can be divided into groups and sub-groups which is subset of Nodes along with the edges in between them. Social Network Analysis is the measures the connection or relationship between the Actors which linked to each other via edges.[3]

Java Universal Network/Graph Framework (JUNG) provides the platform for the Java developers to visualize huge amount of data in different layouts. JUNG is an open source library and very famous because of following reasons:

- It provides a facility to represent directed and undirected graphs.
- Multi-Modal Graphs – Graphs can have different type of Nodes
- Annotations
- Graph theory algorithms
- Hypergraphs
- A separate framework for Visualization
- Mechanism for Filtering data to be plotted

Java Universal Network/Graph Framework (JUNG) not only provides an excellent platform to develop new networking visualization applications but also provides the capabilities to modify and enhance networking features on the existing system. *Netsight* is one of the most popular applications which is developed using the JUNG library. Beside this JUNG Library was used to enhance the famous tool – *Augur* which is one the famous code-analysis tool [4].

Network analysis is becoming very popular in various fields like - computer science, physics, statistics etc. Due to this emerging need in improvement of data analysis tools and technique many research panels are set and sponsored by big Institutes. Some of the famous data analysis techniques are graph visualization and drawing [5], physically-motivated models for characterizing large-scale properties of networks [6], Web page modeling Models [7], analyzing social networks using quantitative methodologies. [8]

3.3 Graph Layouts

A graph is a conceptual representation of a set of objects that may or may not be connected to each other by links. The interconnected objects are represented by mathematical concrete example of events called vertices and the links that connect pair of vertices are called edges. Generally a graph is represented in diagrammatic forms as a set of dots for the vertices and joined by lines or curves for the edges.

Graph or network-like structures are most commonly used for the visual representations of data. The use of Automatic layout of graphs in many applications makes the understanding of things better. Some of the applications where automatic layout of graphs are used are Visual Programming, Software

Engineering (ex: Repository Structures, Flow-Charts, Dependency visualization), Engineering (ex: Circuit Diagrams), and particularly in Web-Visualization almost always when relational data that has been obtained from automated operation such as a database or repository computation has to be visualized then we come across with automatic graph layout. Automatic layout of graphs is a very complex and mathematically challenging problem.

Different types of graphs are:

1. Trees (binary, ordered, rooted, free, etc)
2. Directed Acyclic Graphs
3. General Graphs (directed, undirected, etc)

4 Project Requirements

4.1 Requirements

4.1.1 Scope:

This project is to extract the Twitter data from the database created and populated by Centralized Tweet Collector (TweetCollecter) and plot the Graphs in following Layouts,

1. ISOM Layout
2. KK Layout
3. FR Layout
4. Circle Layout

All the above layouts were implemented using Java Universal Network-Graph (JUNG) Framework which is an open source library and provides interactive API's for Java applications. This project not only represents the plotted Graph's nodes as Twitter Users (with profile image and username) but also displays the Tweet Text on mouse-over. All the plotted Graphs (Layouts) were enabled with features like – Zoom –In/Out, Rotation, and Shearing. Also, all the Graphs (Layouts) are Pickable so that User can rearrange the nodes as per the requirement.

4.1.2 Project Software Requirements:

Following software were required to implement this project,

1. JDK 2.0
2. Eclipse IDE
3. MySQL Server 5.0
4. JUNG Library 2.2.0.1
5. VisualThoughts [http://code.google.com/p/social-analytics-visualisation]
6. TweetCollector [http://code.google.com/p/social-analytics-visualisation]
7. MySQL-Connector-Java-5.1.16-bin.jar
8. Commons-httpclient-3.1.jar
9. AppFrameWork.jar

4.1.2.1 Why Java?

First and foremost reason to choose Java as a development language for this project is that, Java is a platform independent language and can be extensively used over any platform. Its features like- flexibility, easy to use and GUI Development ease make this language the first choice for developing this system. Java has been in market from past 2 decades and has emerged as one of the most powerful languages in current market so selecting Java as the preferred language for this system was not a difficult decision to make.

4.1.3 Project Functional Requirements:

1. **SWRD_01:** Software shall use the Twitter Data extracted by TweetCollecter Application.

   Rationale: In order to get the latest Tweets and corresponding metadata, this system shopud extract the data from the database which is created and populated by TweetCollecter Application.

2. **SWRD_02:** Software shall generate the Twitter Network based on the extracted Data Collected by TweetCollector Application.

   Rationale: This system should extract the following fields from the database created by TweetCollecter Application and create the Jung Network,

   - Sender Username
   - Receiver Username
   - Tweet Text
   - Twitter User Profile Image Link

3. **SWRD_03:** Software shall plot the Data in CircleLayout using JUNG Library.

   Rationale: When user selects the 'Circle Layout' from the Dropdown Menu provided in the Integrated GUI, this system should plot the extracted data in Circular Jung Graph Layout.

4. **SWRD_04:** Software shall plot the Data in FRLayout using JUNG Library.

   Rationale: When user selects the 'FR Layout' from the Dropdown Menu provided in the Integrated GUI, this system should plot the extracted data in FR Jung Graph Layout.

5. **SWRD_05:** Software shall plot the Data in ISOMLayout using JUNG Library.

   Rationale: When user selects the 'ISOM Layout' from the Dropdown Menu provided in the Integrated GUI, this system should plot the extracted data in ISOM Jung Graph Layout.

6. **SWRD_06:** Software shall plot the Data in KKLayout using JUNG Library.

   Rationale: When user selects the 'KK Layout' from the Dropdown Menu provided in the Integrated GUI, this system should plot the extracted data in KK Jung Graph Layout.

7. **SWRD_07:** Software shall implement the Graph Editing/Interaction Functionality. User shall be able to view the Tweet Text when he clicks on the Node.

   Rationale: User should be able to interact with the plotted graph. That is to say, user should be able to access the following features,

- Zoom-In/Out
- Rotation of Graph
- Shearing of Graph
- Pickable Mode
- Use Twitter Profile Image for corresponding nodes
- Displays Twitter Text on Mouse-over on Nodes.

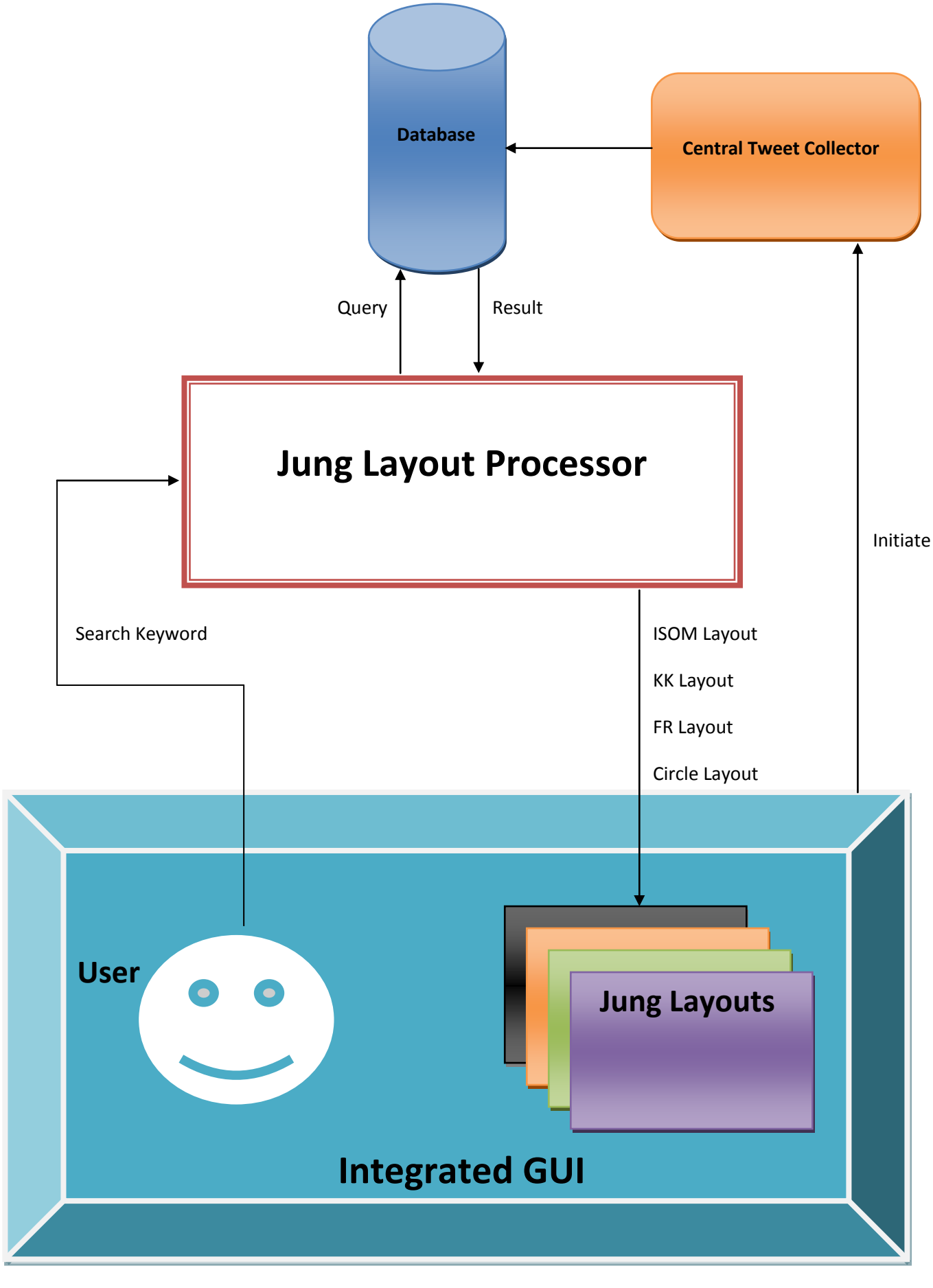
8. **SWRD_08:** Software shall integrate all the Layouts with VisualThoughts Application.

   Rationale: Jung Layouts should be integrated within the Integrated GUI along with other developed Layouts.

4.1.4 Project Non-Functional Requirements

1. CPU with minimum 2.4 GHz processing speed with a physical memory of 2 GB.
2. External library jars need to be set in Java class path.
3. Configure Global variables for MYSQL Database
4. For performance tuning ‘query_cache_size’ and ‘read_buffer_size’ need to be set to 512 MB and 32 MB correspondingly.
5. Configure path and settings for external libraries and API’s
6. To set the java heap, set VM argument (jvm) as "-Xms512M". This is the maximum size for physical memory.

4.2 Use Case


Database
Central Tweet Collector
Query
Result
Jung Layout Processor
Initiate
Search Keyword
ISOM Layout
KK Layout
FR Layout
Circle Layout
User
Jung Layouts
Integrated GUI

The figure above represents the use case diagram for the system. Users here have the access of application via Integrated GUI which consists of user options to provide the search keyword and the layout of the graph. When user provides the search keyword, selects the desired layout and selects the number of nodes to be presented on the graph, Jung layout processor queries the database (which in turn is created and populated by central tweet collector) and extracts the data related to the search keyword. Following are the sub tasks performed by the component – JUNG Layout Processor,

- Connects to the Database, execute query and store the result for processing.
- As a result from database it collects – Sender Usernames, Receiver Username, Tweet Text and User's Profile Picture.
- Based on the search keyword and collected users, it create the Nodes and Edges.
- Assign nodes as the user's profile picture.
- Assign Tweet Text on corresponding nodes.
- Enables plotted graph with features like – Rotation, Shearing, Zoom-In/Out and Pickable mode.

Output from JUNG Layout Processor is represented in the Integrated GUI. User can rearrange the Nodes and the edges as per the requirement as all the plotted graphs are enabled with pickable mode by default. This feature is enabled to make the session interactive. Also, to make the graph representation more appealing, user's profile pictures represents the nodes of the graph.

5 Project Analysis and Design

5.1 System Design

This System was implemented by considering very simple and sequential design flow. Simpler the design is, less complex the system will be. Figure below represents the sequential Diagram for the System. As Java programming language was opted to develop this system, the main and core aspects were utilized to the fullest while developing the system. Each functional module has its own class so that anyone can understand the internals of the system easily. Processing starts when User provides the inputs from Integrated GUI. System Connects to the Database and extracts the data based on the inputs passed by the user. Data extracted from the database is processed and used to create the graph. Data extracted from Database is kept as public so that any class can use it. Inbuilt functions from JUNG Library are utilised to draw the selected nodes and then corresponding edges. A separate class is created to visualize the created graph.

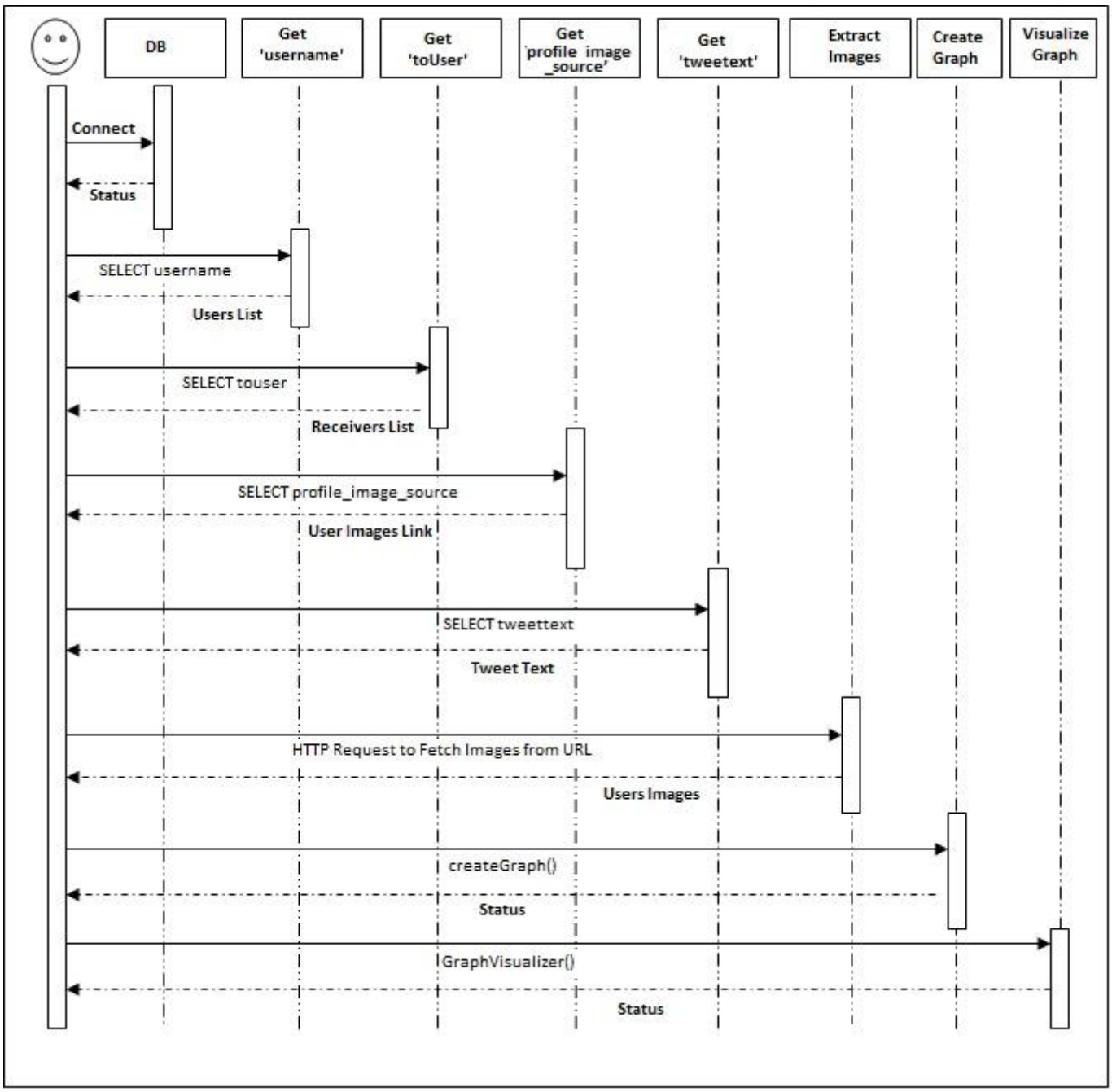


Figure 1: Sequence Diagram of the System

5.2 System Overview

Users have the access of the system via integrated GUI where he can provide the Search Keyword, Select the Number of Nodes in the Graph and the JUNG Layout for the Graph. Upon receiving all the inputs from the user, system connects to the database and queries for the data based on the Search Keyword provided by the user.

Data extraction from the database is limited based on the user selection on the Integrated GUI. Upon successful connection with the database, system extracts the usernames, receiver names, user's image links and tweet text. Extracted data is stored in Global lists which have the *'public'* scope. User's images are extracted by using simple http request on the extracted URL's from the database. Profile images are saved with the name same as the username. For users who don't have any profile image set in twitter,

three different kinds of profile images are used. Default images are also used in case if systems fail to extract the user image using http request.

Once the data is extracted from the database and stored in the local, system starts creating the graph. Graph is created by first creating the node for each username on the graph and then creating the edges within them. A separate Transformer class is created to represent the nodes as the corresponding user profile images. As the final Outcome the created graph is finally visualized on the Integrated GUI using the JUNG Library APIs.

5.2.1 Username

'username' field in database represents the Twitter username for the users. They are the tweet sender. Usernames from the database are extracted and stored in the global list. Upon collecting all the usernames from the database, the list is filtered to remove the duplicates to avoid the duplication of nodes in the graph.

5.2.2 To Users

'touser' filed in database also represents the Twitter username for the users but they are the receivers of tweets. This user list is also extracted from the database and stored in the global list. This list is also filtered to remove the duplication of users who in turn will be represented as nodes on the graph.

5.2.3 Profile Image Source

Users on Twitter can set the profile image of their choice and if not set by any user Twitter sets the default twitter profile image. Every user's profile image has a different link so these URL's are also extracted from the database and stored in the global list. Further image from each URL is downloaded using the http request and saved in the local with name as same as the username. Each user is represented by his profile picture as node on the created graph.

5.2.4 Tweet Text

Users on Twitter can send tweets and these public tweets are also collected in the database corresponding to the senders. 'tweettext' field in database represents the tweet text and extracted to the local. Further this tweet text is displayed when application user brings the mouse cursor on the node/user on the graph. Sender on graph displays the tweet text it sended and receivers on the graph displays the tweet text it received.

5.3 Security

As the database contains the data extracted from the Twitter which serves as the base input for this system and small modification can change the table structures in the database. This can be a potential failure reason for the system so the database is protected with the password so that application user should not alter the database.

5.4 Class Diagram

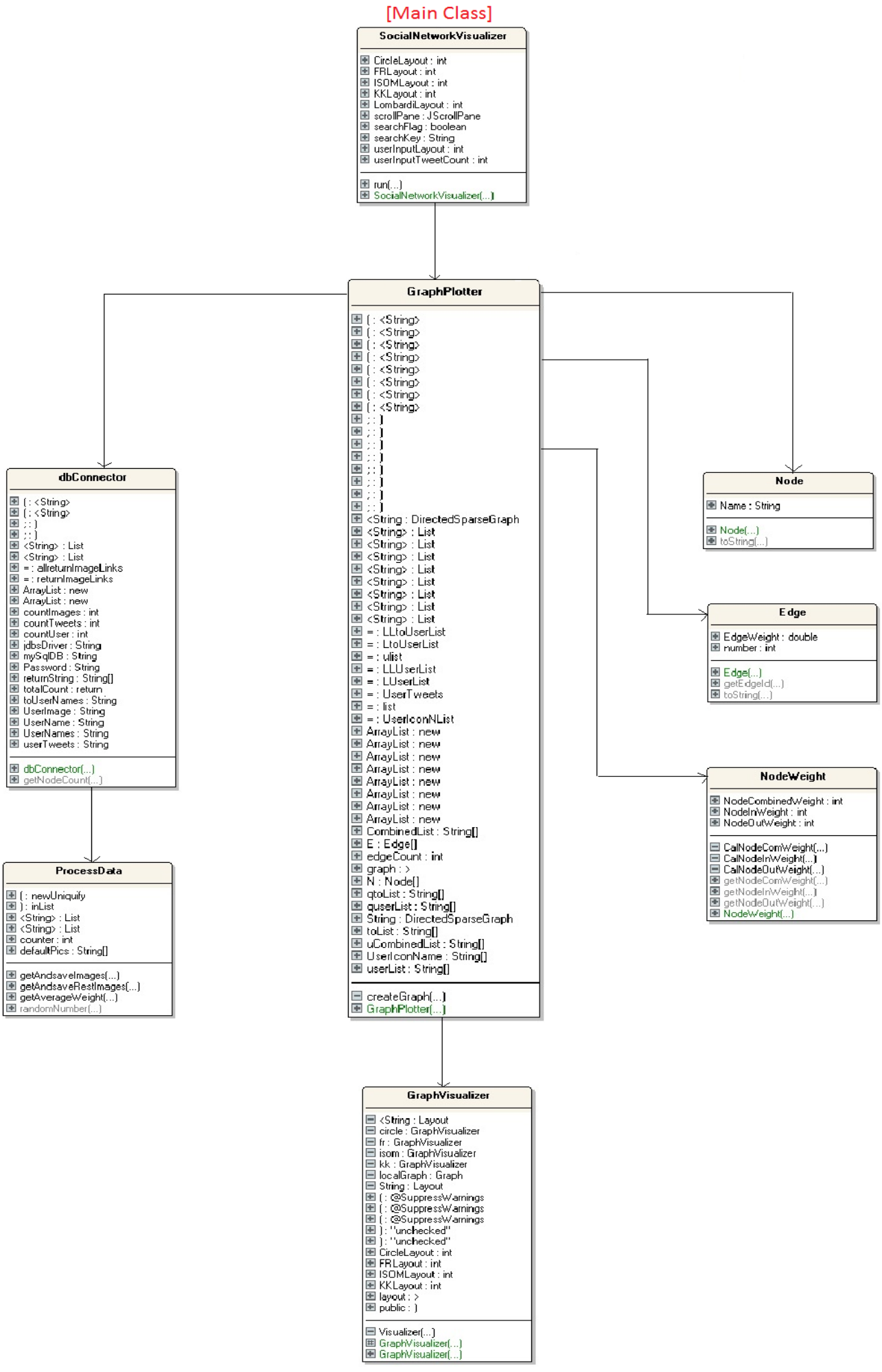


Figure 2: Class Diagram for the System

6 Project Implementation and Testing

6.1 Planning

Very first step on this project was planning. During this phase, after analysing the feasibility of the project a planned proposal was developed which was followed throughout the project. This project is divided into various phases like Requirement Analysis, Designing, Implementation, Testing and Evaluation. Each of these phases is further divided in to different activities. To maintain a smooth execution of the project a deadline for each activity was set in the plan. While setting the deadlines a serious consideration was taken to provide the buffer time to complete any activity. This means that any kind of external issues like illness, weekend leaves, and personal emergency etc. did not hinder the smooth execution of the project.

Activities and Deliverables for various phases were:

Phase	Activities	Deliverable
Requirement Analysis	Domain Understanding Requirements Collection Requirements Classification Conflict Resolution Requirement Validation	Formal System Requirements Document
Implementation & Integration	Coding Integration Self - Review	Formal Prototype of a System
Testing	Unit Testing Integration Testing System Testing Report Writing	Test Results & Reports
Evaluation	Questionnaire Report Writing	Evaluation Reports
Report Submission	Report Writing	Final Report

Table 1.0 – Project Plan

6.2 Implementation

Before implementation, a deep analysis of related systems and libraries was done to avoid any deadlocks during the implementation it may have occurred. Hardware and software requirements were analysed thoroughly but stepping to the implementation phase which helped in smooth execution of implementation phase. Implementation was further divided into subtasks to make sure the project completes on time. A great consideration towards time management and task management was followed during the implementation phase which resulted in the successful completion of this project on time.

6.2.1 Database Connectivity

MySQL 5.0 database was used in this project which was created and populated by Central Tweet Collector Application. Database being created and updated serves as the primary requirement for this project. Java provides an in-built class ‘java.sql’ which enable the users to perform sql operations. To make sure that program can connect to the database and can execute sql queries, external jar file mysql-connector was

included in the code project. This external library enables to perform operations like- Connection to the database, Driver management, execute sql queries and read the results returned from queries.

To connect to the database following parameters were set,

```
//Parameters used for database Connection
static String mySqlDB = "jdbc:mysql://localhost/twitter?user=root&password=password";
static String jdbsDriver = "com.mysql.jdbc.Driver";
static String UserName = "root";
static String Password = "password";
```

And once the connection to the database is estabishoshed, programmer can execute the sql queries and store the returned data in Result Sets.

```
/*----------------------------------------
  Creating a Statement to fetch User Names,
  Profile Images and Tweet Text
----------------------------------------*/
Statement User = myconnection.createStatement ();
User.executeQuery ("SELECT username,touser,profile_image_source,tweettext FROM tweetalldata");
ResultSet resultUser = User.getResultSet();
```

In this project only data from 'username', 'touser', 'profile_image_source' and 'tweettext' fields was extracted from the database.

6.2.2 Processing Data

Once the data from database is collected, it's required to process it and make it usable to plot graphs. Based on the inputs from the user from Integrated GUI data is filtered or screened. User list from database is extracted and duplicates are removed to make sure that senders representing the node do not repeat. Similarly, the global list for receivers is also filtered using the below method,

```
/*****************************************
This method is used remove duplicate items
from the list.
*****************************************/
public static  List<String> newUniquify( List<String> inList){

    HashSet<String> hash = new HashSet<String>();
    hash.addAll(inList);
    inList.clear();
    inList.addAll(hash);
    return inList;

}
```

To make graph look neat and tidy, weight on each node and edge is calculated so that the sub-networks having the higher weight can be prefered to be represented at the center of the graph and the sub-networks having lower weight can surround the former network. Below method is implemented to calculate the weight on each node/edge.

Field – 'profile_image_source' in database contains the url's for users profile image in Twitter and these images are downloaded to the local by using the method below. This method sends the http request to extract the images and saves it. For users not having the profile pic set in Twitter profile or if there is any problem in extracting the image from link because of broken links, 3 different king of twitter default profile pic is used. All the extracted images are saved as the name same as the corresponding username. Further these images represent the corresponding user on the graph as nodes.

```
/*****************************************
This method is used extract user profile
image from Url's and saves them in local.
*****************************************/
public static void getAndsaveImages(String[] imageUrl) throws IOException{

    byte[] b;
    int length;
    URL url;
    InputStream is;
    OutputStream os;
    String destinationFile, fileName;
    File file;

    for(int i =0 ; i< imageUrl.length; i++){

        fileName = imageUrl[i] + ".jpg";
        file = new File(fileName);
        if(file.exists() == false){

            destinationFile = "images//UsersProfilePics//" + GraphPlotter.quserList[i] + ".jpg"
            url = new URL(imageUrl[i]);
            try {
                is = url.openStream();
                os = new FileOutputStream(destinationFile);
                b = new byte[4096];

                while ((length = is.read(b)) != -1) {
                    os.write(b, 0, length);

                }

                is.close();
                os.close();
            }
            catch(IOException e){
                System.out.println("***Exception***");
                FileInputStream fn = new FileInputStream("images//UsersProfilePics//" +
                                                         defaultPics[randomNumber()]);
                os = new FileOutputStream(destinationFile);
                b = new byte[4096];

                while ((length = fn.read(b)) != -1) {
                    os.write(b, 0, length);
                }

                fn.close();
                os.close();
            }
            System.out.println(GraphPlotter.quserList[i] + "...." + Integer.toString(i));
        }
    }
}
```

6.2.3 Plotting Graph

After processing the data as per the requirement, system starts plotting the graph. To create nodes and edges separate dedicated java classes were developed. Each node on graph is created as an object of its class. Similarly, for edges. This design pattern is known as *Factory method* where each object of class is created with different data passed each time while creating an object (node or edge). In our case the data

passed while creating an object is username for nodes and edge Id number for edges which makes the graph easy to refer to. The method below is used to create the graph which is eventually nothing but creating the corresponding nodes and then linking the created nodes with edges (links). Traceability between the nodes is maintained by using global lists. In this project, JUNG Library's basic graph class – DirectedSparseGraph was used to create the Graph.

```
/****************************************
This method is used to create the graph
based on the Data collected from Database
****************************************/
private void createGraph(){

    //Creating a graph
    graph = new DirectedSparseGraph<String, String>();

    //Creating an array of Node objects
    N = new Node[uCombinedList.length];

    //Creating Node objects
    for (int i = 0; i < uCombinedList.length;i++){

            N[i] = new Node(uCombinedList[i]);
    }

    //Creating an array of Edge objects
    E = new Edge[userList.length];

    //Adding nodes & edges to the Graph
    for (int i = 0; i < userList.length;i++){

        graph.addEdge(Edge.getEdgeId(), userList[i], toList[i]);
    }
}
```

6.2.4 Visualizing Graph

6.2.4.1 Graph

In JUNG Library, Graph is responsible for representing the nodes and then linking each node with other relative nodes. These links are known as Edges. Edges can be directed or non-directed. In this project we implemented the directed edges. Each node on graph can be linked to multiple nodes.

6.2.4.2 Layout

Users have the option of choosing the layout from Integrated GUI. Available options provided for the users are – ISOM Layout, FR Layout, KK Layout and Circle Layout. Each layout has its own method in JUNG. So depending upon the user's choice, corresponding method is used to create the layout. Layout decides the positions for the nodes which will be represented in graph and depending upon the nodes position edges are drawn.

```
public GraphVisualizer(Graph graph, int Layout){

    //Getting a Main Graph in a local Graph
    this.localGraph = graph;

    //Selecting a layout based on the user's Choice
    switch (Layout) {

    case ISOMLayout:
        isom = new GraphVisualizer();
        layout = new ISOMLayout(isom.localGraph);
        Visualizer();
        break;

    case KKLayout:
        kk = new GraphVisualizer();
        layout = new KKLayout(kk.localGraph);
        Visualizer();
        break;

    case FRLayout:
        fr = new GraphVisualizer();
        layout = new FRLayout(fr.localGraph);
        Visualizer();
        break;

    case CircleLayout:
        circle = new GraphVisualizer();
        layout = new CircleLayout(circle.localGraph);
        Visualizer();
        break;
    }
}
```

6.2.4.3 Visualization Model

JUNG System is implemented in a Model-View-Controller (MVC) Model where Graph and Layout decides 'what we need to represent' and Renderer (explained later) decides 'how to represent'. VisualizationModel take control of Layouts. Whenever the state of VisualizationViewer changes, VisualizationModel interacts with it and animates the Layout.

```
/*****************************************
This method is used to create the visualizer
based on the layout selected by the User.
*****************************************/
@SuppressWarnings({ "unchecked", "rawtypes" })
private void Visualizer(){

//Setting the Layout size
layout.setSize(new Dimension(650,650));

//Creating the Visualizer
VisualizationViewer<String,String> visualizer = new VisualizationViewer<String,String>(layout);
visualizer.setPreferredSize(new Dimension(650,650));

//Creating a hash-map to store Icons
Map<String, Icon> iconsMap = new HashMap<String, Icon>();
for (int i = 0; i < GraphPlotter.uCombinedList.length; i++) {
    String tempString = "images//UsersProfilePics//" + GraphPlotter.uCombinedList[i] + ".jpg";
    try {
        Icon icon = new ImageIcon(tempString);
        iconsMap.put(GraphPlotter.uCombinedList[i], icon);

    } catch (Exception ex) {

        System.out.println("Error: Problem in creating a Hash-Map for Icons");
    }
}
```

6.2.4.4 Renderer

Till now nodes/edges are created and the positions are decided but the appearance of these nodes and edges is still not decided. That's what a renderer in JUNG do. Renderer draws the nodes and edges as per the customization done. Renderers are responsible for decorating the graph. In this project as mentioned earlier, nodes will be represented by user's Twitter profile Pictures so to implement this Transformer Class - ***myVertexIconTransformer*** was created. This class helps in setting the image icon for each node. Further this transformer class is assigned to the Graph Renderer as shown below.

```
//Creating local Icon Transformer
final myVertexIconTransformer<String> vertexIconTransformer = new myVertexIconTransformer<String>()

//Setting Hash-Map for Icon on Transformer
vertexIconTransformer.setIconMap(iconsMap);

//A tool-tip Transformer, this will display tweets text on mouse-over.
visualizer.setVertexToolTipTransformer(new ToolTipView());

//Setting Visualizer's Renderer properties
visualizer.getRenderContext().setVertexIconTransformer(vertexIconTransformer);
//visualizer.getRenderContext().setVertexFillPaintTransformer(NodePaint);
visualizer.getRenderContext().setVertexLabelTransformer(new ToStringLabeller());
visualizer.getRenderContext().setEdgeLabelTransformer(new ToStringLabeller());
```

```
class myVertexIconTransformer<V> extends DefaultVertexIconTransformer<V> implements Transformer<V, Icon>{

    boolean Imagesfill = true;
    boolean Imagesoutline = false;

    public boolean isImagesfill() {
        return Imagesfill;
    }

    public void setImagesfill(boolean Imagesfill) {
        this .Imagesfill = Imagesfill;
    }

    public boolean isImagesoutline() {
        return Imagesoutline;
    }

    public void setImagesoutline(boolean Imagesoutline) {
        this .Imagesoutline = Imagesoutline;
    }

    public Icon transform(V ver) {
        if (Imagesfill) {
            return (Icon) iconMap.get(ver);}
        else {
            return null;
        }
    }
}
```

6.2.4.5 Pluggable Graph Mouse

To enable interaction with the created Graphs, JUNG provides a class known as – *PluggableGraphMouse*. This class can be extensively used to provide features like – Rotation, Shearing, Scaling (Zoom – In/Out) etc. In this system, all of these features are enabled in the created graph.

```
//PluggableGraphMouse properties
PluggableGraphMouse graphMouse = new PluggableGraphMouse();
graphMouse.add(new PickingGraphMousePlugin());  //Picking Mode is active
graphMouse.add(new RotatingGraphMousePlugin()); //Rotating Graph mode is active
graphMouse.add(new ShearingGraphMousePlugin()); //shearing Graph mode is active
graphMouse.add(new LabelEditingGraphMousePlugin ()); //Label Editing mode is active
graphMouse.add(new TranslatingGraphMousePlugin(MouseEvent.BUTTON1_MASK)); //Translating Graph mode
graphMouse.add(new ScalingGraphMousePlugin(new CrossoverScalingControl(), 0, 1.1f, 0.9f)); //Scalin
```

6.2.4.6 Visualization Viewer

Till now the user will not be able to view the created graph because it not yet plotted on the Viewer. VisualizationViewer Class in JUNG Library extends the Java's JPanel class and is used to visualize the graphs. VisualizationViewer follows the model and Renderer. This class also, handles the ***PluggableGraphMouse*** and applies ***myVertexIconTransformer*** in our system. Since VisualizationViewer extends the Java's JPanel class, we can utilize the functionality and features extensively which sought out the implementation up to some extent.

```
//Setting GraphMouse on Visualizer
visualizer.setGraphMouse(graphMouse);

JPanel JP = new JPanel();
JP.setSize(200, 200);
JP.add(visualizer);
SocialNetworkVisualizer.scrollPane.setViewportView(JP);
SocialNetworkVisualizer.scrollPane.setEnabled(true);
```

6.3 Testing

As an integral part of implementation, testing was done to ensure that the developed system works without and flaws to provide a healthy experience to the users. Testing phase was further divided into sub-phases for smoother execution of the project and also because of the reason that more kind of testing we perform much better the final products comes out to be.

6.3.1 Unit Testing

Unit testing has proved to be the most effective kind of testing till now. Maximum number of defects is found and thus reduces the fixture cost and reduces the overall release time for the software product. As the best practice unit testing is performed by the developer only. Each component of code was tested or analyzed accordingly not only to ensure the best quality of the developed software but also to make sure that code behaves in the same way as it was intended to. Unit testing was performed as and when the component was developed.

6.3.2 Integration Testing

After ensuring that each component in the code is working correctly, all the components were integrated as per the design and tested accordingly. Integration testing was very important and crucial activity involved in this project as it ensured that the developed components in code are interacting among themselves as per the design and as intended. A serious consideration was adopted to check that data transfers from one object to another properly and software-software Integration (SSI) is intact to enable the software execute smoothly without any flaws.

6.3.3 System Testing

As the final phase of testing, system tests were performed to ensure that application works as per the requirements defied in design phase. Functionality of complete system was tested for functionality from all aspects in this phase and to perform this following tests were conducted,

Test Case No.	Test Description	Initial Conditions	Inputs	Outputs	Result
1	To verify that Software extracts the tweets based on the 'Search Keyword' provided.	MySql database is created by importing the table 'dataextraction'. Application is running.	Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	Only Tweets having the keyword 'you' are extracted.	PASS, Same as Expected.
2	To verify that Software sets the field 'Select Nodes' to 20 by Default.	MySql database is created by importing the table 'dataextraction'.	Execute the Application. Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	'Select Nodes' field is set to 20 by Default. Selected Jung Layout is plotted with extracted 20 Tweets only.	PASS, Same as Expected.
3	To verify that Software plots the ISOM Layout	MySql database is created by importing the	Execute the Application.	ISOM Layout is Plotted properly.	PASS, Same as Expected.

	properly.	table 'dataextraction'. Application is running.	Search Keyword = 'you' Select ISOM Layout Click on the Button 'Visualise'		
4	To verify that Software plots the KK Layout properly.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select KK Layout Click on the Button 'Visualise'	KK Layout is Plotted properly.	PASS, Same as Expected.
5	To verify that Software plots the FR Layout properly.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select RF Layout Click on the Button 'Visualise'	FR Layout is Plotted properly.	PASS, Same as Expected.
6	To verify that Software plots the Circle Layout properly.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select Circle Layout Click on the Button 'Visualise'	Circle Layout is Plotted properly.	PASS, Same as Expected.
7	To verify that Software plots the Jung Layouts with User Profile Twitter image as node. [This Test is performed for all the Available Jung Layouts]	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select the Jung Layout Click on the Button 'Visualise'	Selected Jung Layout is Plotted properly with Nodes as the User's Profile Image.	PASS, Same as Expected.
8	To verify that Software plots the Jung Layouts with	MySql database is created by importing the	Execute the Application.	Selected Jung Layout is Plotted properly with	PASS, Same as Expected.

	Node name as corresponding User's Twitter username. [This Test is performed for all the Available Jung Layouts]	table 'dataextraction'. Application is running.	Search Keyword = 'you' Select the Jung Layout Click on the Button 'Visualise'	Node name as corresponding User's Twitter username.	
9	To verify that Software enables the Rotate feature for the plotted Jung Graph.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	Selected Layout is Plotted properly. Graph rotates when keyboard button – CTRL is used with the Mouse.	PASS, Same as Expected.
10	To verify that Software enables the Shearing feature for the plotted Jung Graph.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	Selected Layout is Plotted properly. Graph Shears when keyboard button – SHIFT is used with the Mouse.	PASS, Same as Expected.
11	To verify that Software enables the ZOOM-In & ZOOM-Out feature for the plotted Jung Graph.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	Selected Layout is Plotted properly. Graph is enabled with ZOOM-In & ZOOM-Out feature when Mouse Scroller is scrolled Down and Up respectively.	PASS, Same as Expected.
12	To verify that Software enables the Pickable Mode for the plotted Jung Graph.	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	Selected Layout is Plotted properly. Graph is enabled with Pickable Mode (User can re-arrange the nodes by dragging it anywhere on the plotted Graph.	PASS, Same as Expected.
13	To verify that Software displays the	MySql database is created by	Execute the Application.	Selected Layout is Plotted	PASS, Same as Expected.

	corresponding Tweet Text on Mouse Over for every nodes.	importing the table 'dataextraction'. Application is running.	Search Keyword = 'you' Select any Jung Layout Click on the Button 'Visualise'	properly. Software displays the corresponding Tweet Text on Mouse-over for every node.	
14	To verify that JUNG Layout Software Package integrates properly with IntegratedGUI. [This Test is performed for all the Available Jung Layouts]	MySql database is created by importing the table 'dataextraction'. Application is running.	Execute the Application. Search Keyword = 'you' Select the Jung Layout Click on the Button 'Visualise'	Selected Jung Layout is properly integrated with the IntegratedGUI.	PASS, Same as Expected.

Table 2.0 – Project Test Plan

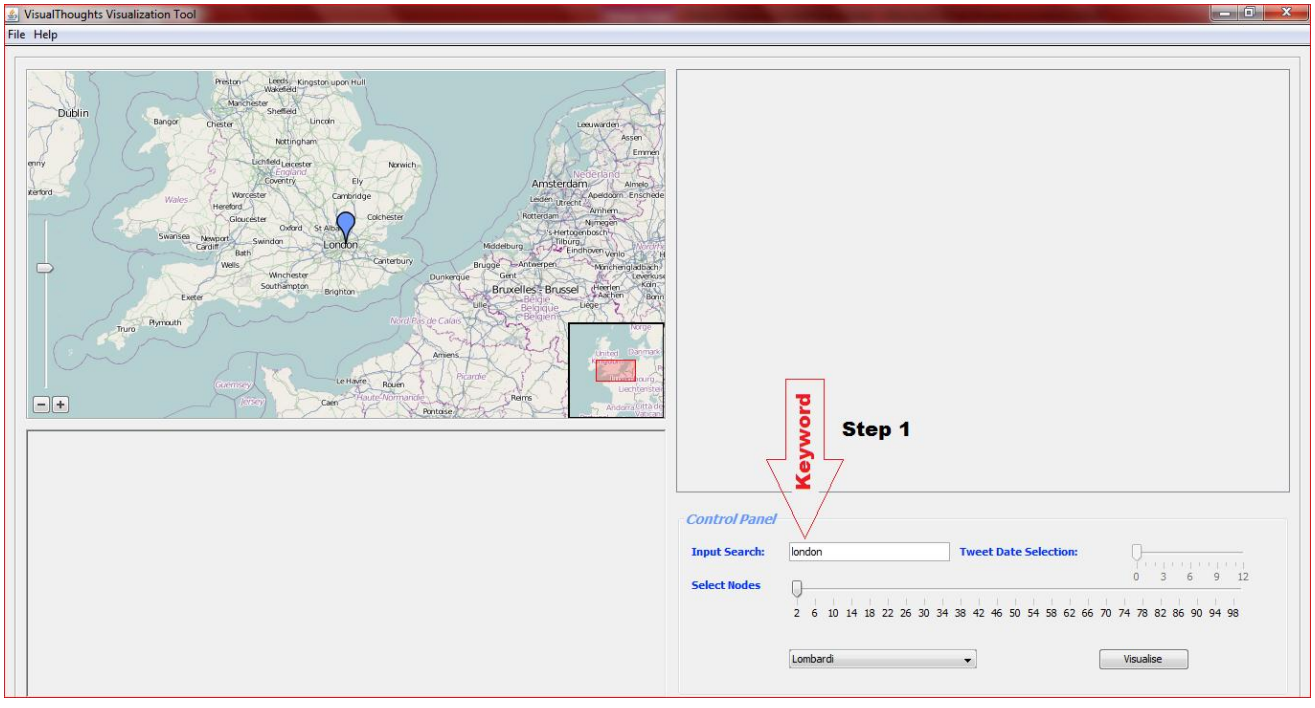
7 Project Demonstration & Evaluation

This section will demonstrate the functionality of this application. As mentioned earlier that JUNG Layouts are integrated with the Integrated GUI which was pre-developed. Integrated GUI is the main interface GUI for the users to provide the inputs/choices. Once the Integrated GUI is up and running, any Jung Layout can be generated using in 4 simple steps as described below:

7.1 Demonstration

7.1.1 Step 1

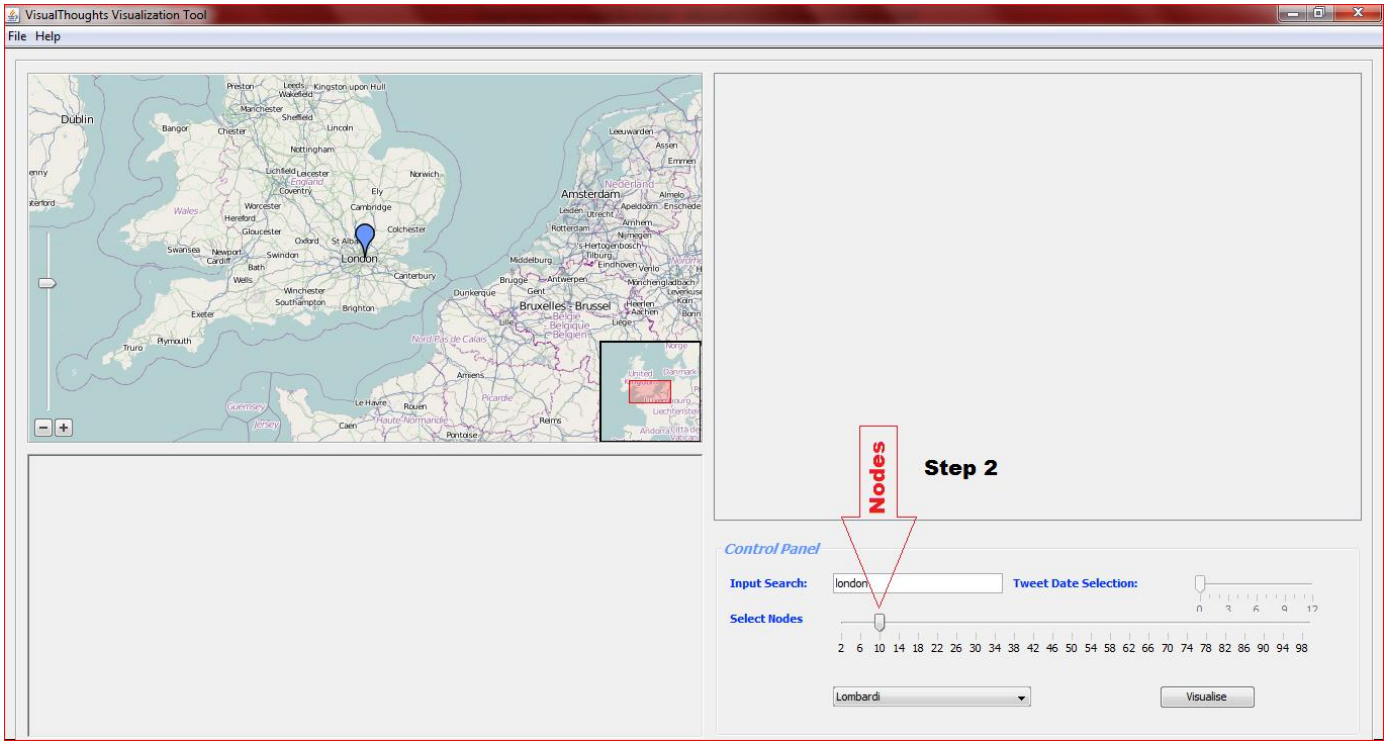
Once the Integrated GUI is up and running, provide the search keyword in the field – 'Input Search'. With this keyword application filters the data which it extracts from the database.



For this demonstration, we have used the search keyword – '**london**'.

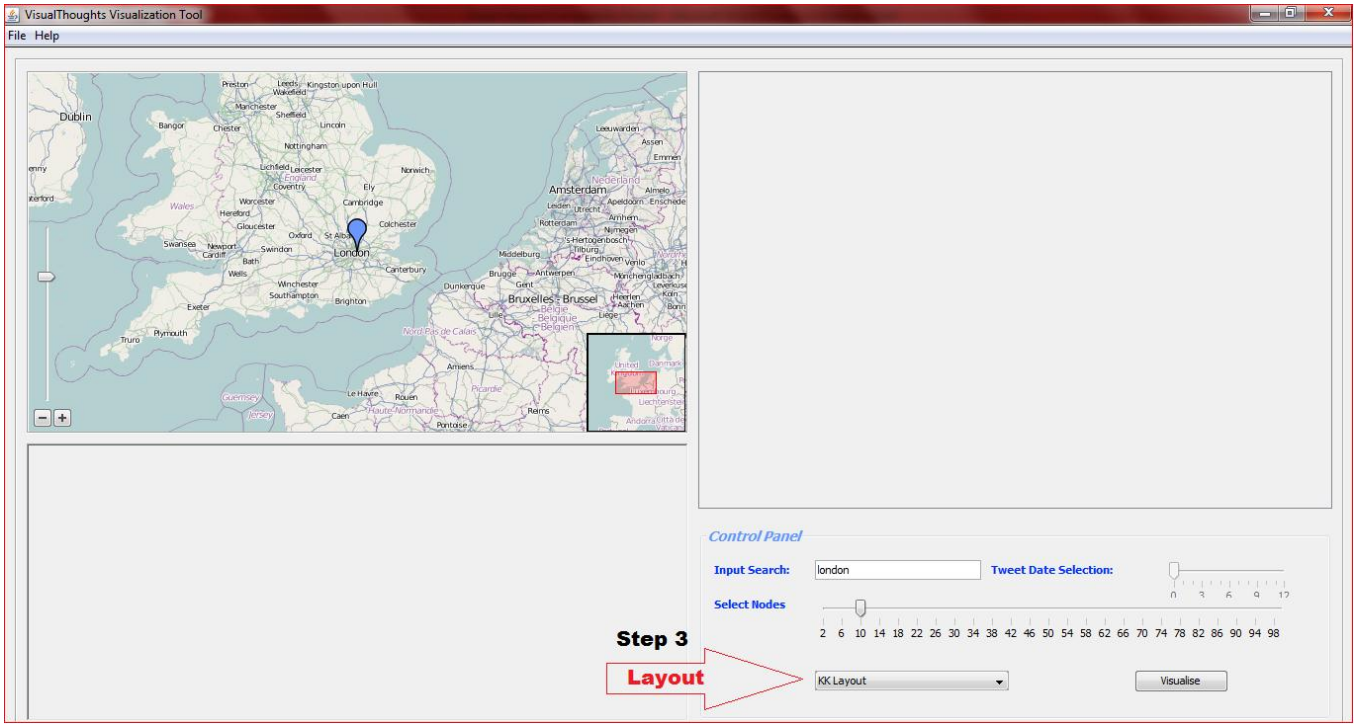
7.1.2 Step 2

After providing the search keyword –'london', user needs to select the number of nodes to be represented in the graph. This feature is enabled to filter the data from database so that graph should not have more nodes on the graph and should be simple and understandable. More number of nodes takes more time to be plotted. Also, the speed and application speed extensively depends upon the internet connection speed as the users profile pic will be downloaded using the urls over http connection. In this demo Number of nodes selected is 10.



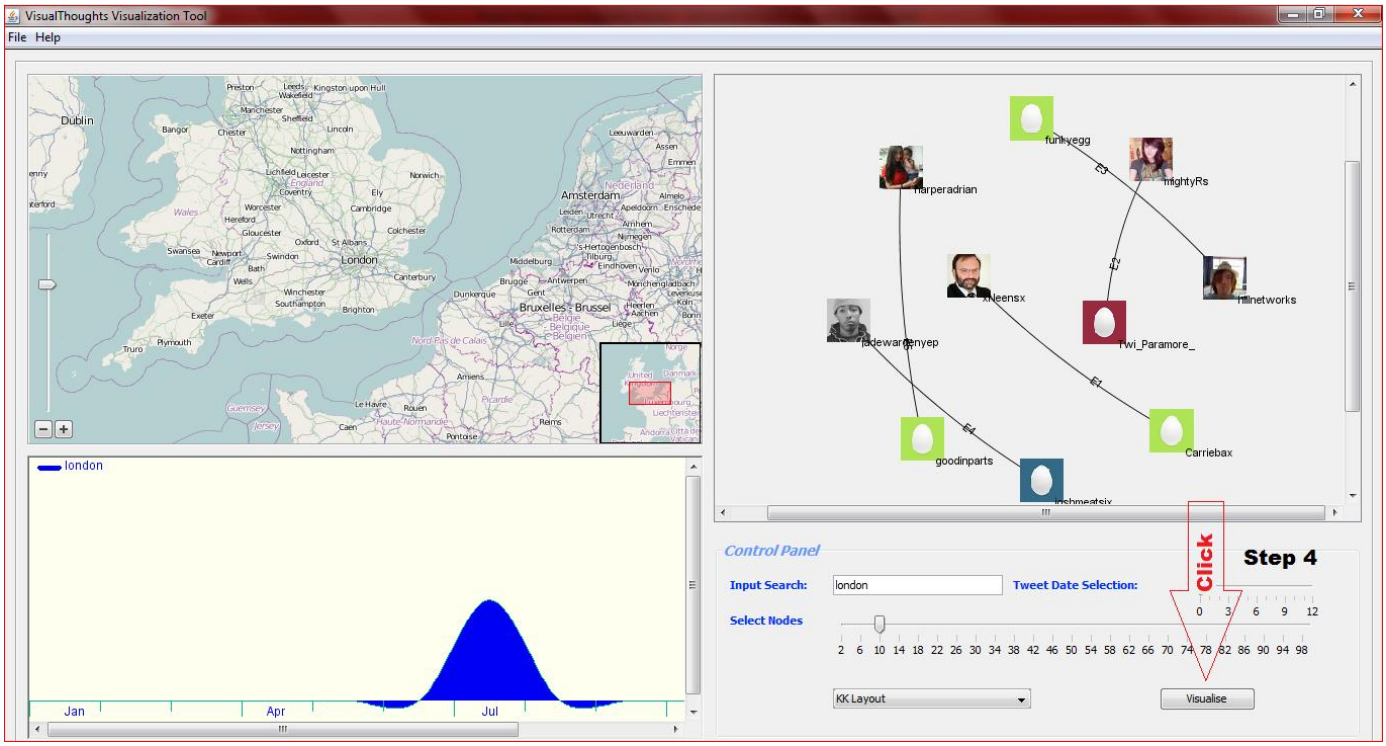
7.1.3 Step 3

Now user needs to select the Layout of his choice. Available choices in Integrated GUI are – ISOM Layout, KK Layout, FR Layout and Circle Layout. Layout decides the position of the nodes on the graph. In this demo, we have selected 'KK Layout'.



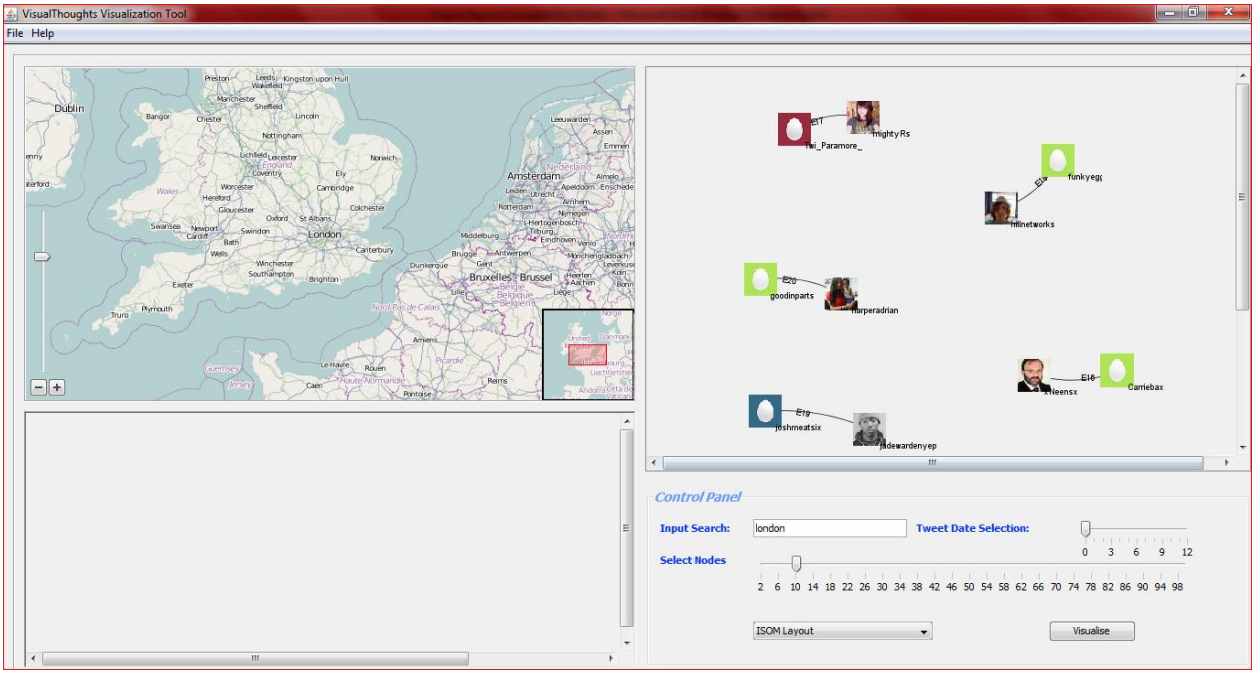
7.1.4 Step 4

Once the user has provided the inputs, it's time to visualize the graph. And to do this all user has to do is just click on the button – 'Visualise'. As soon as user clicks on this button, processing starts at the backend which is hidden from the user. Within few moments a graph appears on the GUI panel.

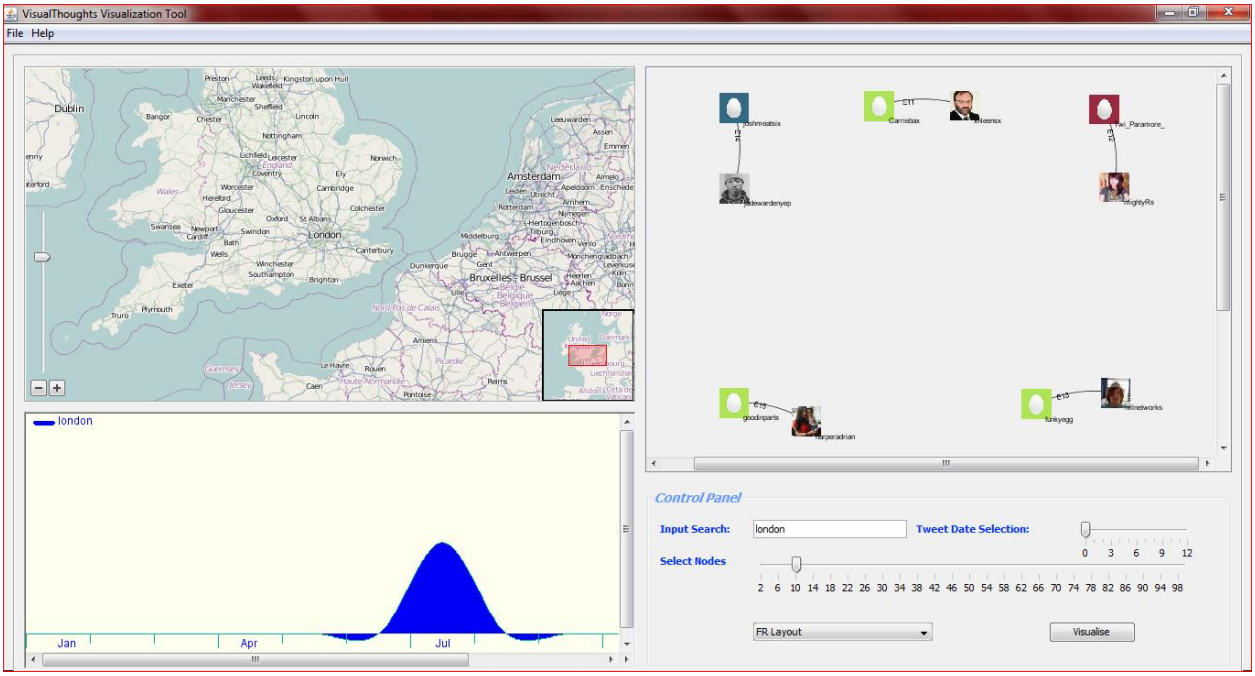


Other Available Layouts in the Integrated GUI are

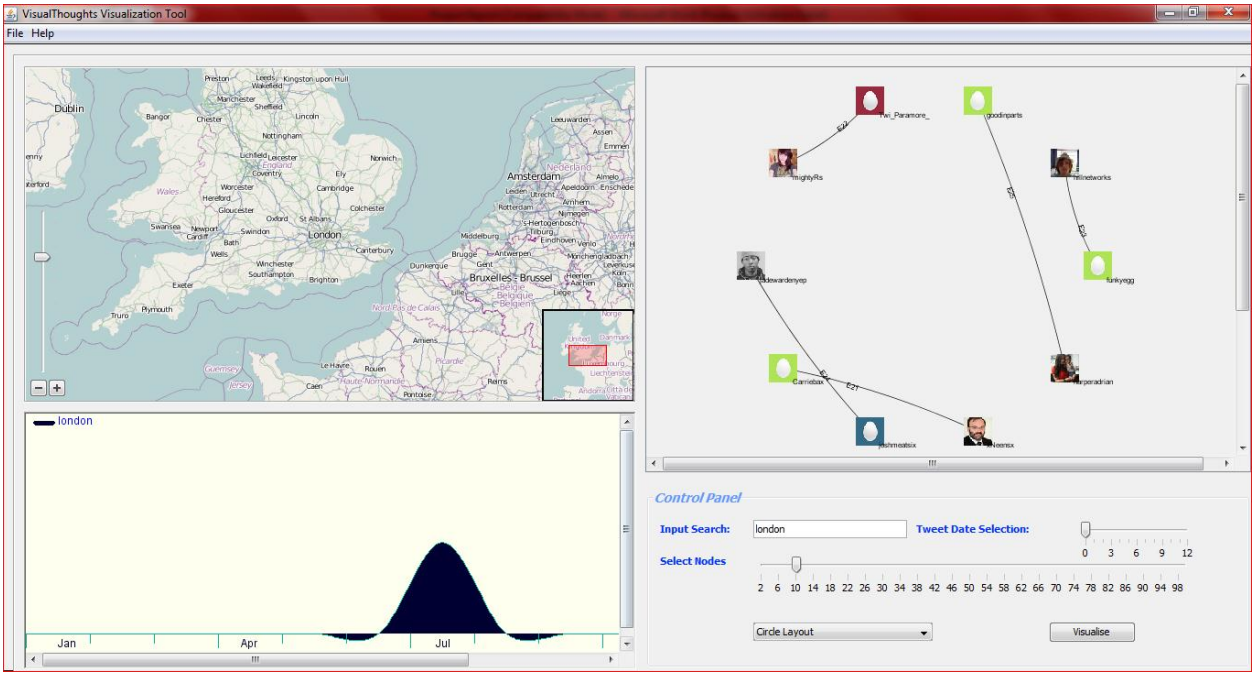
7.1.4.1 ISOM Layout



7.1.4.2 FR Layout



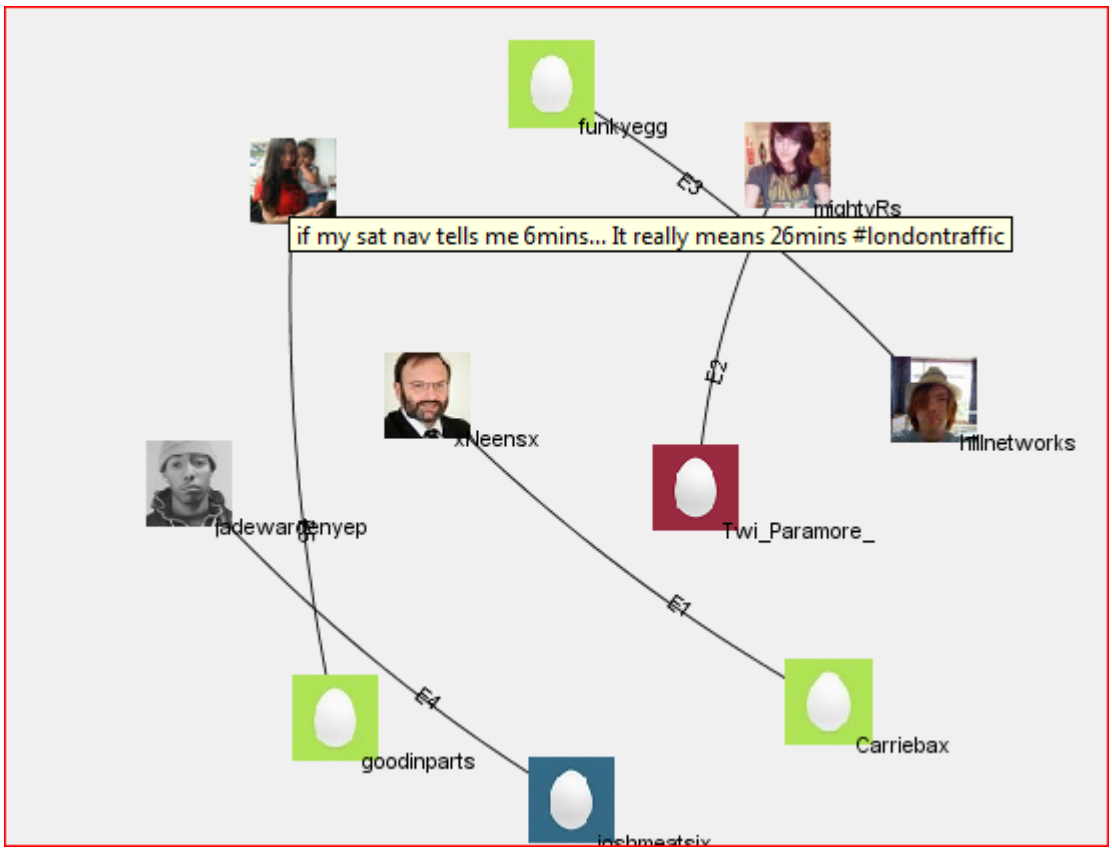
7.1.4.3 Circle Layout



7.2 Interaction with Graphs

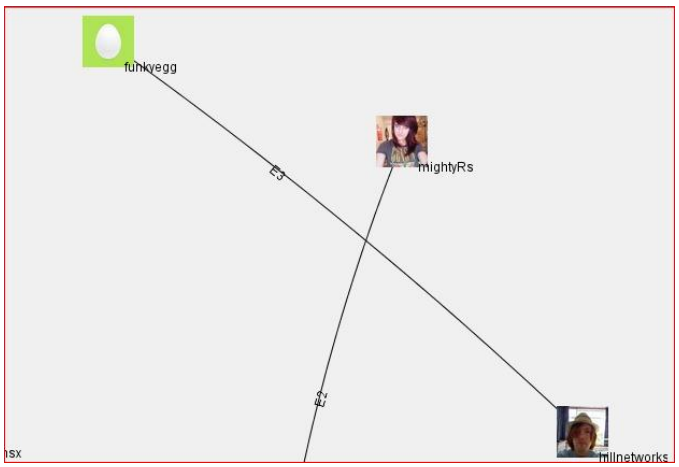
7.2.1 Tweet Text Display

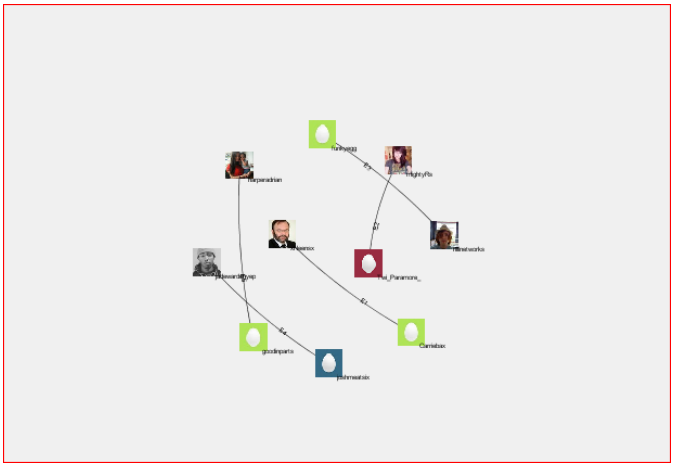
When user brings the mouse cursor on any node (User's picture on the graph) a Tweet Text is displayed. This feature enables the users to read the tweet which was sent by any Twiter User on the graph to another Twitter user.



7.2.2 Zoom-In/Zoom-Out

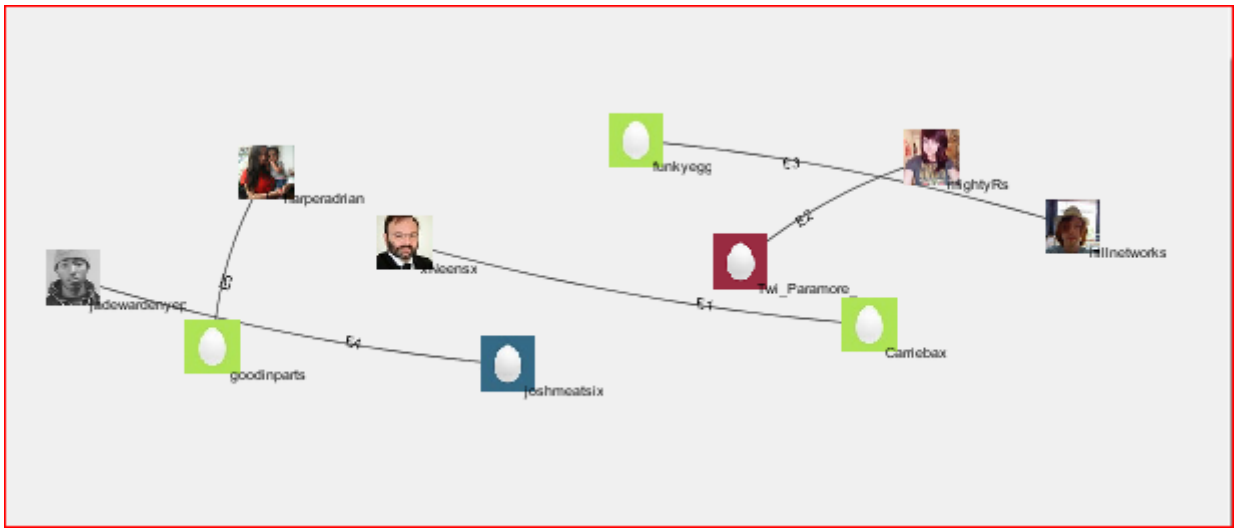
When user scroll-down the mouse scroller, plotted graph is zoomed-in which helps the users to analyse the complex network easily. On scrolling-up the mouse scroller, user can zoom-out the plotted graph which can help them in viewing the entire network in one view shot.





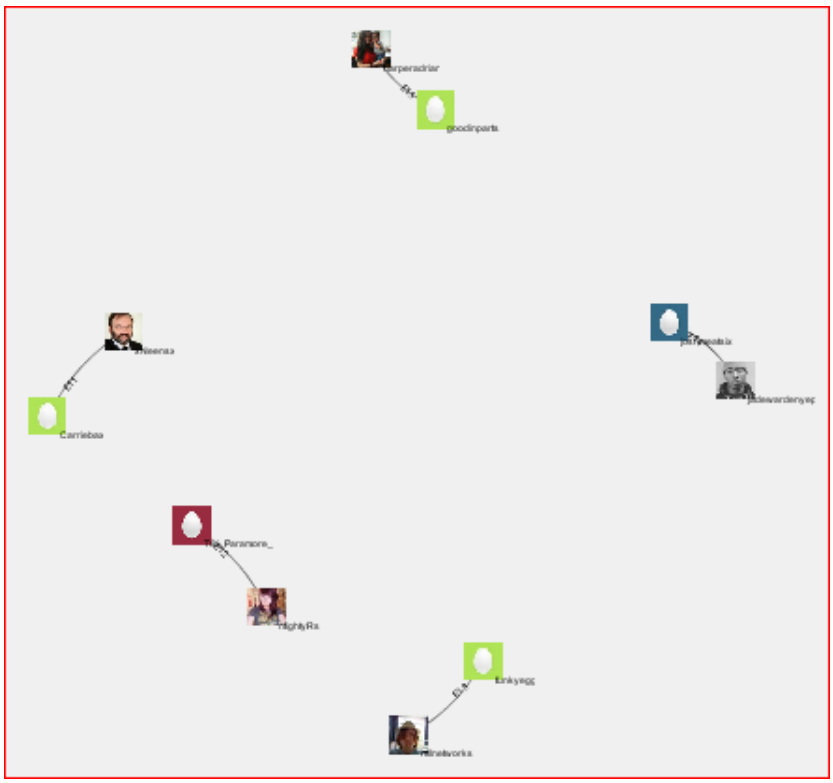
7.2.3 Shearing

When user holds the *CTRL* key on the keyboard and moves the mouse right and left, plotted graph shears right and left respectively.



7.2.4 Rotation

When user holds the *SHIFT* key on the keyboard and moves the mouse up and down, plotted graph rotates clock-wise and anti-clockwise respectively.



7.3 Evaluation

Testing alone is not sufficed to ensure that the project is complete and the developed product is ready to use extensively so this project was evaluated too at the end. A questionnaire was prepared taking into the consideration the main aspects of the product and the features it provides. A group of 20 volunteers were identified and selected to participate in the Evaluation questionnaire. As an ethical aspect, feedback received from each participant and their observations was not shared with any other participant and kept confidential.

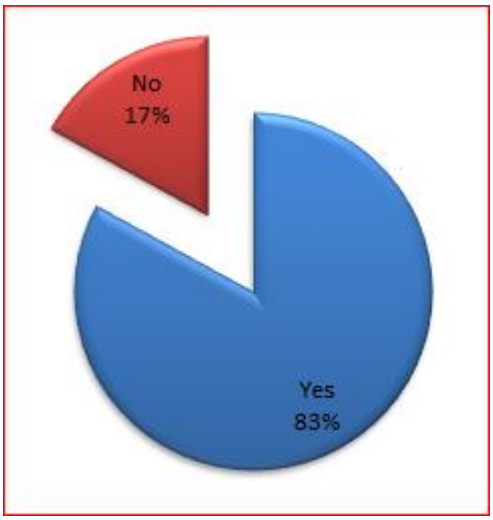
Upon completion of questionnaire, the result was thoroughly and deeply analyzed to know the loop holes in the developed product. Following questions were asked in the questionnaire.

7.3.1 Questionnaire

We started with the basic questions to know about the awareness of participants on social networking site like twitter.

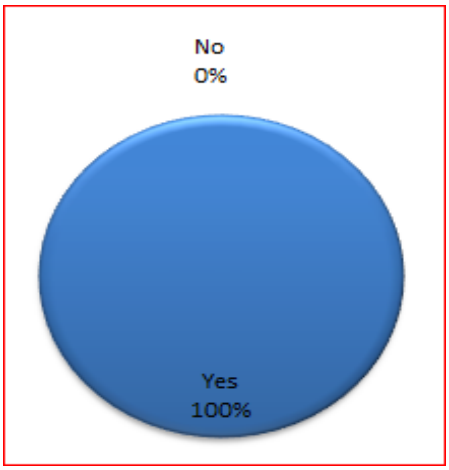
1. Do you know what Twitter is?

2. Are you on Twitter?



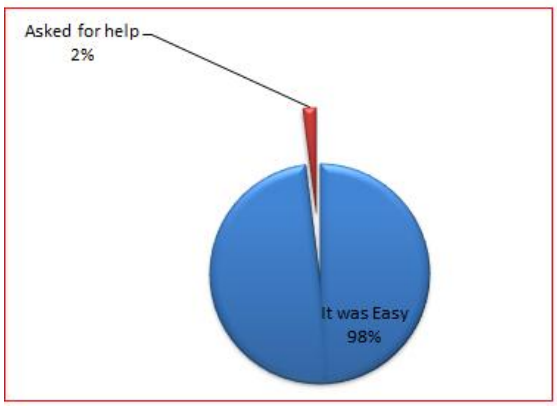
Then brief information about the developed application was provided to the participants along with the demonstration. After demonstrating the functionality and features of the application participants were asked to use the application on their own. Following questions were asked after the practical session.

3. Were you able to plot a Jung Layout Graph using Integrated GUI?

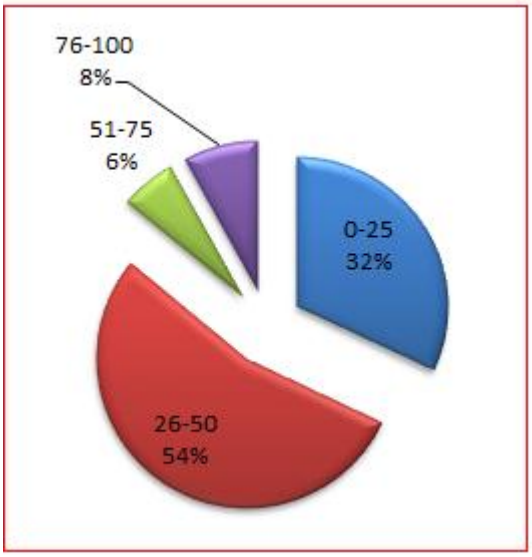


From the feedback received from the above question it was ensured that the developed application is very easy to understand.

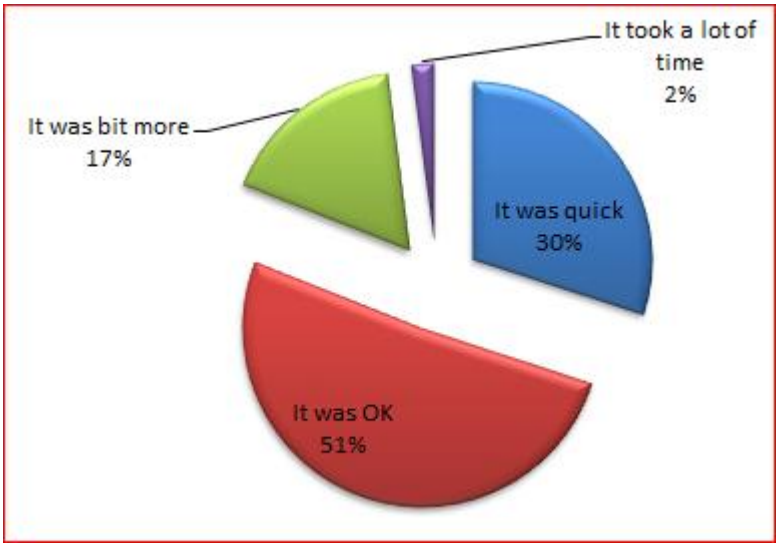
4. Did you find it easy to use or consulted anyone to know how to use it?



5. How many nodes did you try plotting on the graph?

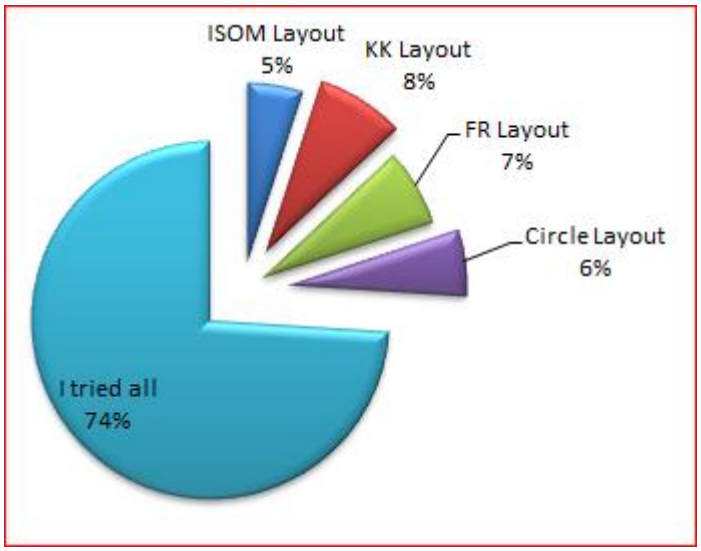


6. How much time did the application took to display the graph?

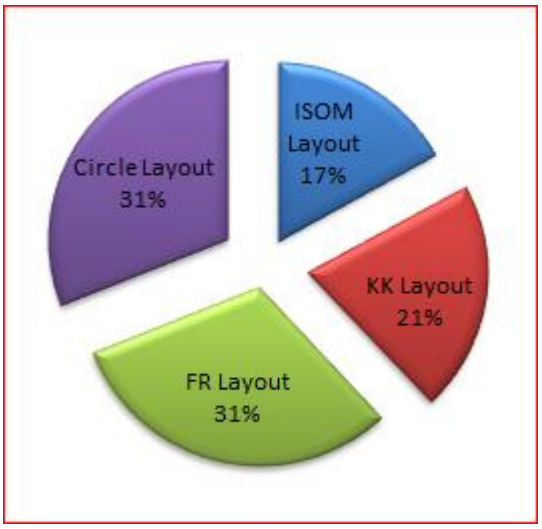


Above question provided a very healthy feedback which let us know that the there is still a possibility of imporving the application processing speed at backend. As a result of this feedback the application source code was optimized further more to increase the processing speed of application at backend.

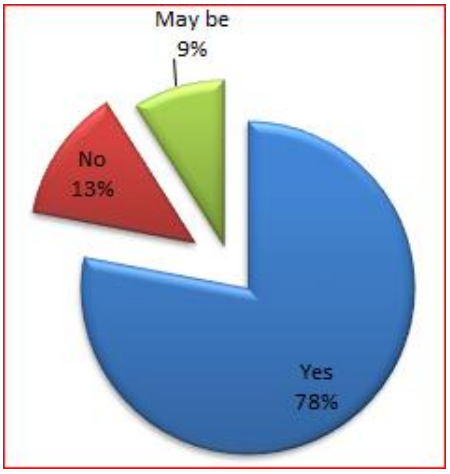
7. Which Layout you Selected?



8. Which Layout you think is the easiest and convinient to understand?



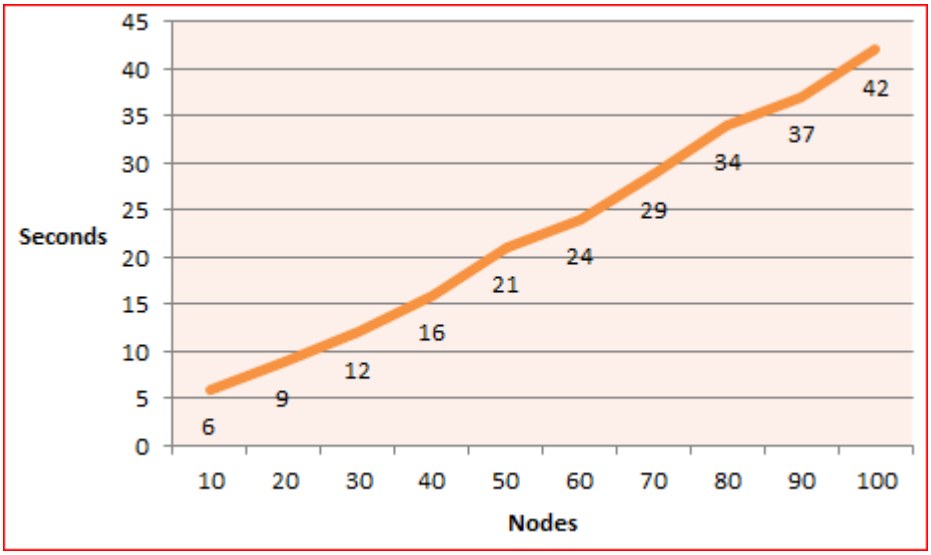
9. Do you think this application serves the purpose it's intended for?



Feedback received from the above question was also very interesting. This was understood from the feedback that still more features need to be added to satisfy the users experience on this application. As an action on this feedback, future work on this application was decided which is described later in this report.

7.3.2 Performance Evaluation

After conducting the questionnaire and analysing the feedback received, we performed the performance evaluation was performed for each layout. As per the performance evaluation, we tried to plot graph for the entire range of nodes. Starting from 10, node count was incremented by 10 for every cycle. For each cycle the time taken by the application to plot display the graph was noted. Taken time included the time taken by the application at backend to fetch data from the database, process the fetched data, extract user images from the urls fetched from the database, creating the graph and finally visualizing the network graph on Integrated GUI. Results received from the performance evaluation were quite satisfactory. A graph for this performance evaluation is as below,



8 Critical Evaluation

8.1 Revisiting the Project Aim

Project aim was to develop a system which can represent different JUNG Layouts based on the data extracted by Central Tweet Collector Application. Layouts included in this project were – ISOM Layout, KK Layout, FR Layout and Circle Layout. Not only displaying the graphs in selected layout suffice the project aim but also it was required to search the tweets with specific input search keyword provided by the user. User was also provided with an option to select the number of nodes (users) he wants to plot in the network (graph). Furthermore, all the plotted graphs were enabled with the features like – Shearing, Rotation, Zoom-In/Out and Pickable Mode. User will also be able to view the Tweet Text when mouse cursor is brought over any node/User. Also, nodes in the graph were represented by the user's twitter profile image. All the Layouts were required to be integrated within the Integrated GUI.

With an excellent guidance from the Project Supervisor and continuous dedication towards developing this system, all the system requirements were fulfilled and aim of the project was achieved successfully.

8.2 Revisiting the Design

A lot of analysis was done and considerable effort was spent while creating the design requirements for this system. Form the top view system seems to have achieved all of the design requirements but when we look at the internals of the system there still exist the possibility of improvement, especially on the robustness part of the system. If more time would have been provided, we definitely would have improved the system performance by optimizing the design and the coding.

8.3 What did I learn?

At the beginning of the this project it looked quite difficult to meet all the requirements within the small duration on time but later when we moved on and dig more into the project, bits started to fall into pieces with time. Besides gaining hands on experience and knowledge on various technologies like – Java, JUNG, MySQL, Java Swing, Appframework I also learned how to manage number of tasks at same time and schedule it priority wise. Additionally, I get a chance to work under the expert supervision and also with talented and hard working co-workers. Entire journey throughout this project was very fruitful and just because of this experience I gained within such a short time, I decided to continue working furthermore on the improvement of this project.

9 Window of Improvement (Future Work)

Based on the Testing and evaluation performed at the end of the project, it was noticed that there still exists the possibility of improvement in this application. Firstly, '*editing feature'* on the graph can be provided using which users can edit or remove the unwanted node (user) from the graph. Secondly, an option can be provided on the Integrated GUI using which user can view the '*next selected number of nodes*' on the graph. This feature can be very useful as user may need not to provide all the inputs again with new number of selected nodes every time.

10 Conclusion

Twitter is an excellent source of information which can be used by many professionals and individuals if arranged or formatted in a proper manner. By using this newly developed system, users actually can analyse and use the large amount of twitter data. Most important aspect of this new system is its user friendly and easy to understand visualization Layouts. This system also has an option to provide the keyword to broaden their search. Besides this application is also provides the interactive features like – Rotation, Shearing, Zoom-In/Out and Pickable Mode. Also, Application displays the Tweet Text when Mouse cursor is brought over any node/User. Overall, the main intent of the application was met and the project completed successfully within the time frame which was estimated before starting the project in Planning Phase.

11 References

[1] d. boyd. Friends, Friendsters, and Top 8: Writing community into being on social network sites. First Monday, 11(12), 2006.

[2]Forbes (2011), http://www.forbes.com/sites/oliverchiang/2011/01/19/twitter-hits-nearly-200m-users-110m-tweets-per-day-focuses-on-global-expansion/

[3] Scott, J., 1992, *Social Network Analysis.* Newbury Park CA: Sage.

[4] Froehlich J, Dourish P (2004). "Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams." In "Proceedings of the International Conference on Software Engineering," pp. 387–396.

[5] Battista GD, Eades P, Tamassia R, Tollis IG (1999). Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall.

[6] Watts DJ, Strogatz SH (1998). "Collective dynamics of 'small-world' networks." Nature, 393, 440–442.

[7] Brin S, Page L (1998). "The anatomy of a large-scale hypertextual Web search engine." Computer Networks and ISDN Systems, 30(1–7), 107–117. URL citeseer.ist.psu.edu/ brin98anatomy.html.

[8] Wasserman S, Faust K (1994). Social Network Analysis. Cambridge University Press, Cambridge.

[9]ANTONIOU, Grigoris and Frank Van HARMELEN. 2004. *A Semantic Web Primer*. The MIT Press.

[10] CHO, Junghoo, Hector GARCIA-MOLINA, and Lawrence PAGE. 1998. Efficient crawling through URL ordering. *In*: *Computer Networks and ISDN Systems*., pp.161-172.

[11] BONACICH, Phillip, Amalya OLIVER, and Tom A.B. SNIJDERS. 1998. Controlling for size in centrality scores. *Social Networks*. **20**, p.135–141.

[12] CHEN, Chaomei. *Information visualization: beyond the horizon*.

[13] BONNIE A. NARDI, Steve Whittaker, Ellen Isaacs, Mike Creech, Jeff Johnson, John Hainsworth. 2001. ContactMap: Integrating Communication and Information Through Visualizing Personal Social Networks. *Forthcoming in Communications of the ACM*.

[14] DIESTEL, Reinhard. 2006. *Graph theory*. Springer.

[15] BRIN, Sergey and Lawrence PAGE. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *In*: *In Computer Networks and ISDN Systems*., pp.107-117.

[16] Java Comparison (2010) in wiki.processing.org, Retrieved October 2011, http://wiki.processing.org/w/Java_Comparison

[17] Contact us (2011) In Twitter.com, Retrieved October 2011 from http://twitter.com/about/contact

[18] Jason Pontin, "From Many Tweets, One Loud Voice on the Internet," The New York Times, April 22, 2007

[19] Elke Dunker(2008), MSc Project Proposal Template, Middlesex University 2008

[20] Twitter (2011) In Wikipedia, Retrieved October 10th 2011 from http://en.wikipedia.org/wiki/Twitter

Appendices

Questionnaire:

Question 1: Do you know what Twitter is?

Options: Yes | No

Question 2: Are you on Twitter?

Options: Yes | No

Question 3: Were you able to plot a Jung Layout Graph using Integrated GUI?

Options: Yes | No

Question 4: Did you find it easy to use or consulted anyone to know how to use it?

Options: Asked for help | It was Easy

Question 5: How many nodes did you try plotting on the graph?

Options: 0-25 |26-50 | 51-75 | 76-100

Question 6: How much time did the application took to display the graph?

Options: It was Quick | It was Ok | It was bit more | It took a lot of time

Question 7: Which Layout you Selected?

Options: ISOM Layout | KK Layout | FR Layout | Circle Layout | I tried all

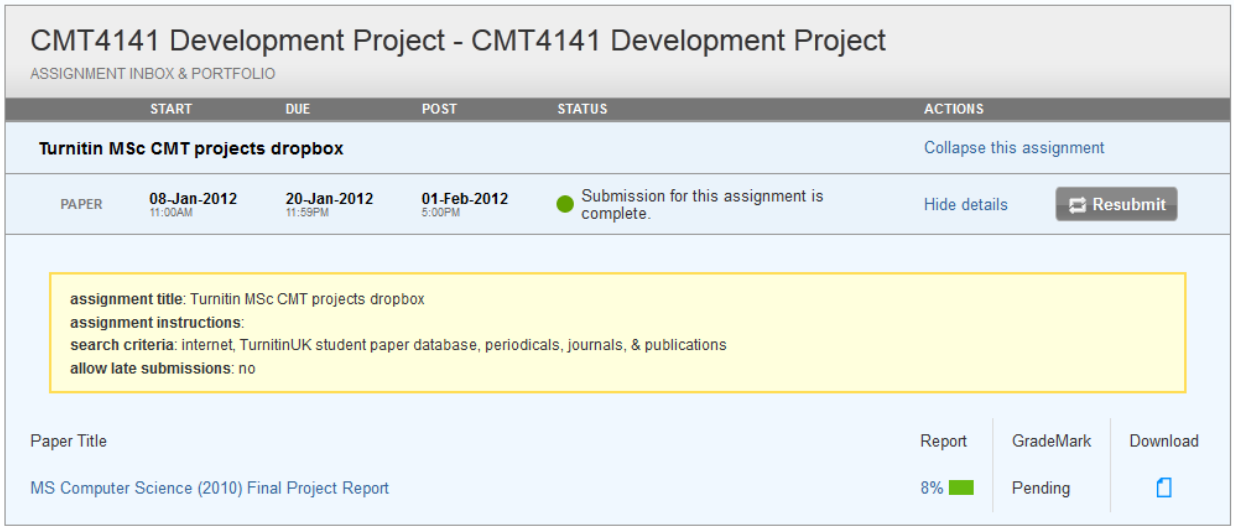
Question 8: Which Layout you think is the easiest and convenient to understand?

Options: ISOM Layout | KK Layout | FR Layout | Circle Layout

Question 9: Do you think this application serves the purpose it's intended for?

Options: Yes | no | May be

Turnit-In Report:



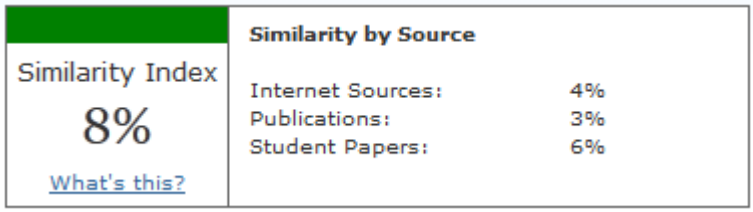
CMT4141 Development Project - CMT4141 Development Project

ASSIGNMENT INBOX & PORTFOLIO

	START	DUE	POST	STATUS	ACTIONS
Turnitin MSc CMT projects dropbox					Collapse this assignment
PAPER	08-Jan-2012 11:00AM	20-Jan-2012 11:59PM	01-Feb-2012 5:00PM	Submission for this assignment is complete.	Hide details Resubmit

assignment title: Turnitin MSc CMT projects dropbox
assignment instructions:
search criteria: internet, TurnitinUK student paper database, periodicals, journals, & publications
allow late submissions: no

Paper Title	Report	GradeMark	Download
MS Computer Science (2010) Final Project Report	8%	Pending	



Similarity Index

8%

What's this?

Similarity by Source

Internet Sources:	4%
Publications:	3%
Student Papers:	6%