

Twitter

Analysis and Design:

MySQL:

For the project, we are using MySQL as our relational database management system to store information about the users and the tweets made by these users.

MySQL is the world's most widely used open source [relational database management system](#) (RDBMS) that runs as a server providing multi-user access to a number of databases. It is also provided for all platforms like Windows and Linux.

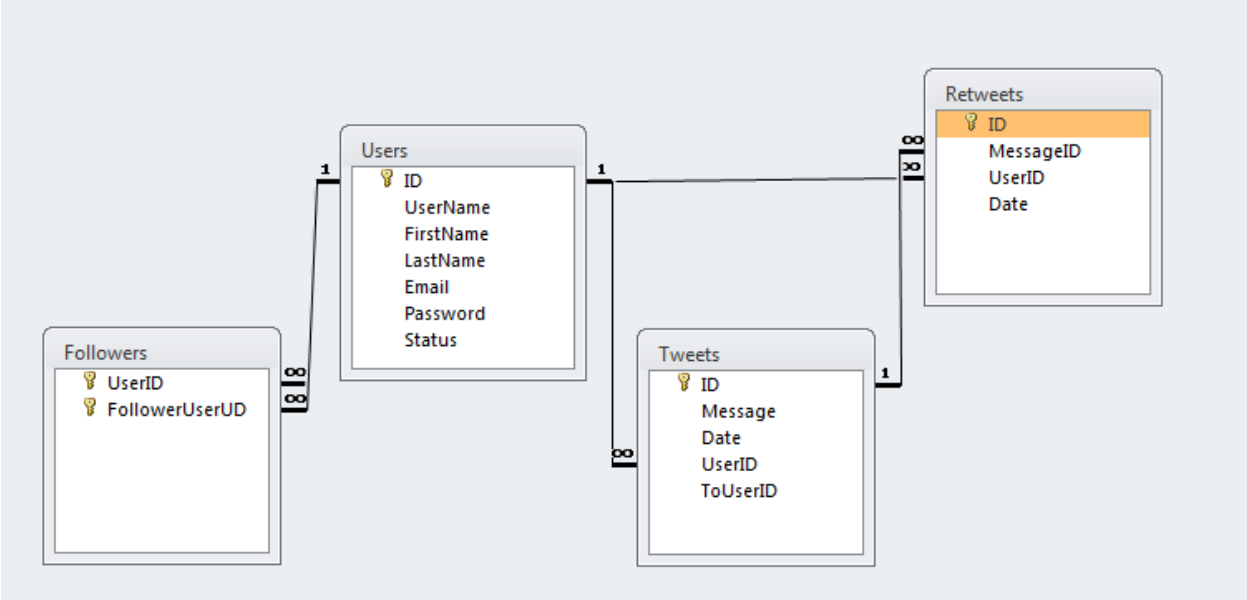
It can be easily installed from <http://dev.mysql.com/downloads/mysql/>.

The reason why we chose MySQL as our database, because it is supported by all platforms, easy to install and configure, and because the connection from java is very easy and straight forward.

A copy of the database is available at our bitbucket repository.

Database Design:

In this section, we will discuss about the database design of the twitter application which satisfies our needs to provide the main features.



In the above figure, our primary database design is shown where we have four tables:

- 1- The first table is the “Users” table, where all users along with their information are stored. Records are inserted in this table upon the user signing up.
- 2- The second table is the “Followers” table, where are stored all the people that the users follow.
- 3- The third table is the “Tweets” table, where all the users messages (tweet) are stored. This table is also used to get the private tweets of a user by the ToUserID field. If the value of the ToUserID is null, then it’s a general tweet otherwise it will be a private tweet to the specified user ID.
- 4- The fourth table is the “Retweets” table, where we have a message ID retweeted by a user.

Application Design:

In this session, we are going to describe the main design of our application by explaining the main classes that exist in the program according to their packages.

1- RMI Package:

The rmi package is where the client and the remote object which is the server exist. The following classes are the form the main parts of this package:

Server Interface

The server interface is the Remote class where it should be later implemented in one of its subclasses. This interface contains methods which should provide a functionality to connect to a database, login, sign in, tweet, re-tweet, send private message, and search for users and tags, and other functions that is made to satisfy the specifications of the application.

Server

The Server class extends UnicastRemoteObject and implements the ServerInterface. The Server class uses an rmi registry to register itself with “Server” name. The server implements all the functions of the ServerInterface and connects to a mysql database which is provided in the Downloads section in our repository. The details of each function with its description are provided in the class.

This class should be started first for the application’s Launcher to locate the server in the registry.

Client Interface

The client interface is where the client should connect to the server and display all tweets, private messages and more functions to the user. This interface is made Remote so that when two users are online and a followed user tweets or send a private message to the other user, the other user will be directly notified using remote methods which can be accessed externally.

Client

The client class extends UnicastRemoteObject and implements the ClientInterface.

Launcher

The Launcher class is the other class which will be executed with the server. This class is used to create a client and launch the GUI of the application.

2- Graphical Package:

The graphical package is responsible for the GUI of the application. The main GUI of the application is divided into four parts which corresponds to four panel existing in the MainFrame that are:

- 1- Main Panel: The main panel exists in the center of the GUI. It contains the a table of the users, tweets and retweets, date when a tweet was tweeted. The user can write click on a user or a tweet and send a private message to that user or retweet his tweet.
- 2- Search Panel: The search panel exists in the top of the GUI where the home, my tweets, private messages labels exist and the user can click on them to see the corresponding list. The search panel also contains text box that is used for searching for usernames and for tags.
- 3- Left Panel: The left panel is in the left of the GUI. It consists of Tweet, Following, Follower and Log Out. The user can click on them and see the corresponding lists.
- 4- Tweet Panel: The tweet panel

3- UnitTest Package:

The UnitTest Package is where we test some important methods of the server class. Four are our test:

- 1- Following: This class tests the method toggleFollowUser in the Server. We are testing when calling the method toggleFollowUser in the Server class the method should make the user follow or unfollow another user.

- 2- MyTweet: This class tests the method tweet and getMytweets in the Server Class. We test when calling the method tweet from the server, which means that the user entered a tweet, the size of getMyTweets should increase by one.
- 3- Tweet: This class tests the method tweet and getTweets in the Server Class. We test when calling the method tweet from the server (the user entered a tweet) the size of getTweets should increase by one for the user that is following the user that tweeted(the new tweet should appear in the lists of tweets of the user that is following the user that entered the tweer).
- 4- Private Message: This class tests the method sendPrivateMessage in Server. We check if the size of the getPrivateMessage has increased for the destUser after calling the method sendPrivateMessage from the srcUser (srcUser sent a private message to the destUser).