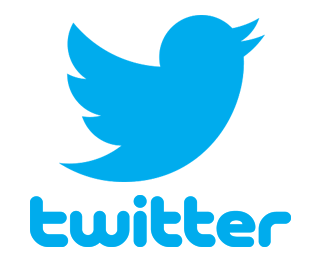
Database Systems Lab Project Proposal

[](https://www.google.com.pk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjz6qeBioXSAhVHwxQKHRvQA0kQjRwIBw&url=https://clipartfest.com/categories/view/fd0a1f393407033f6f2e9ff1755546ed02e2f1da/twitter-clipart-logo.html&psig=AFQjCNEUV_sKXkBYs3kbzstO69axuO2UJg&ust=1486800555676104)

Team Members

|  |  |
| --- | --- |
| Member Name | Roll number |
| Muhammad Asif | 13-4347 |
| Mudassar Sharif | 13-4155 |
| Shamama Safdar | 13-4019 |
| Bilal Anees | 13-4159 |

**Table of Contents**

[**1.** **Introduction** 1](#_Toc429138335)

[**1.1** **Project Overview** 1](#_Toc429138336)

[**1.2** **User Classes and Characteristics** 1](#_Toc429138337)

[**2.** **Functionality Points** 1](#_Toc429138338)

[**Appendix A: Glossary** 2](#_Toc429138339)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name(s)** | **Date** | **Reason(s) For Change(s)** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

Twitter is a kind of sister project implemented at undergraduate level that is almost similar to popular social networking website “Twitter” in functionalities and lay out. The purpose of making this project is to come up with the social website same as original Twitter and to know how these social websites are made. This document is intended for project advisors who are lab instructors and the structure and functionalities of the project Twitter is given below.

## **Project Overview**

Twitter is a look alike to “Twitter”, the popular social networking website. It has almost all the features of Twitter except, of course, its networking implementation. In Twitter, a person can have his personal account where he can post tweets, read tweets of people he is following, follow other people and send private messages to them. Twitter is implemented using Database Management Systems and web form and Master Page using ASP.NET.

## **User Classes and Characteristics**

1. Recreational/Casual Users

These will be the common users and they will be able to do almost everything that the user of original Twitter is able to do. They will be able to follow other users and can be followed, send private messages to people they are following, will have their own public page where they would be able to tweet and will be able to read tweets of other users.

1. Public Figure

Public Figure will be those users of Twitter who will have more than a million followers and they will be given a title of Public Figure, which will be written along their name on their public page.

1. Administrator

Administrators will be the special users who will be able to view everyone’s tweet and will be able to delete the tweet of any user or even accounts.

# **Functionality Points**

1. **Functionality 1 : Signup**

|  |  |
| --- | --- |
| **Signup** | |
| **User** | Non Account Holder User |
| **Steps** | |
| a) | A user can sign up for this website by providing username, password, Full name, Email and contact number on signup.aspx page. |
| b). | If this user name is already taken a message is shown "sorry, this user name already exists. Try another" |
| c). | In case the user name is available, a new account is created and the new user Is redirected to his/her Home.aspx page. |
| **Assumptions** | username, password, Full name will be not null, Contact number can be null |

1. **Functionality 2 : Delete Own Account**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Account Holder User |
| **Steps** | |
| a) | A user can delete his account. |
| b). | Ask for the email address for confirmation. |
| c). | When his account is deleted he/she will be directed signup.aspx |
| **Assumptions** | -Email address is taken to send mail on account where it will not be implemented. |

1. **Functionality 3 : Login/Logout**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Admin/User |
| **Steps** | |
| a) | User can login to the website by providing username and password on Adminlogin.aspx page. |
| b). | In case incorrect user name or password is provided following message is shown "Incorrect user name or password" |
| c). | In case of correct username and password user will be redirected to his/her AdminHome.aspx page. |
| **Assumptions** | - |

1. **Functionality 4 : Can have a Public Figure**

|  |  |
| --- | --- |
| **Login** | |
| **User** | User |
| **Steps** | |
| a) | User can make his own Public Figure. |
| b). | User can link that with his account. |
|  |  |
| **Assumptions** | Public Figure is the person who has 1 lac followers. |

1. **Functionality 5 : Can Tweet many times on his Public Page**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Public Figure |
| **Steps** | |
|  | |
| a) | Public Figure can add as many posts and delete them. |
| **Assumptions** |  |

1. **Functionality 6 : Can Follow Other Users**

|  |  |
| --- | --- |
| **Login** | |
| **User** | User |
| **Steps** | |
| a) | User can Follow other user. |
| b). | If user is not following the other person then follow button is displayed. |
|  |  |
| **Assumptions** | Follow button is available |

1. **Functionality 7 : Can UnFollow Other Users**

|  |  |
| --- | --- |
| **Login** | |
| **User** | User |
| **Steps** | |
| a) | User can UnFollow other user. |
| b). | If user is following the other person then unfollow button is displayed. |
|  |  |
| **Assumptions** | UnFollow button is available. |

1. **Functionality 8 : Can Send Private Messages**

|  |  |
| --- | --- |
| **Login** | |
| **User** | User |
| **Steps** | |
| a) | User can send messages. |
| b). | User can receive messages. |
|  |  |
| **Assumptions** | Any one of the user is following the other. |

**9. Functionality 9 Trends:**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Account Holder User |
| **Steps** | |
| a) | A user can create trends by using hash tags |
| b). | Make any number of tags |
| c). | Removed when account is deleted |
| **Assumptions** |  |

**10. Functionality 10 Add profile picture:**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Account Holder User |
| **Steps** | |
| a) | A user can upload a profile picture |
| b). | Any number of times |
| c). | - |
| **Assumptions** | - |

**11. Functionality 11 Retweet:**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Account Holder User |
| **Steps** | |
| a) | A user can re tweet a tweet of another user. |
| **Assumptions** | Somebody has tweeted before only then it can be retweeted. |

**12. Functionality 12 Admin Rights:**

|  |  |
| --- | --- |
| **Login** | |
| **User** | Admin |
| **Steps** | |
| a) | Admin can delete a user’s account. |
| b). | Block the account |
| c). | When his account is deleted he/she will be directed signup.aspx |
| **Assumptions** | -Email address is taken to send mail on account where it will not be implemented.  User can be reported. |

# **Database Diagram**

# **Appendix A: Glossary**

Account Holder User is also taken as normal user and as well as admin and public figure.

**Twitter Schema**

create table dbo.UoT

( User\_ID int primary key identity (1,1),

User\_Name varchar (20),

DP varchar (50),

DoB date,

Phone\_no varchar (50),

Email varchar (50),

Password varchar (50),

block int null

)

create table Admin(

User\_ID int primary key not null foreign key references UoT(User\_ID) on delete cascade on update cascade,

Admin\_Name varchar (20)

)

create table Followers(

User\_ID int foreign key references UoT(User\_ID) on delete no action on update no action,

f\_ids int foreign key references UoT(User\_ID) on delete no action on update no action,

primary key (User\_ID, f\_ids)

)

create table Trend(

Trend\_ID int primary key not null identity(1,1),

trend\_text varchar (21)

)

create table org\_tweet(

Tweet\_ID int primary key identity(1,1),

Tweet\_text varchar(270),

block bit

)

create table AllTweets(

Tweet\_ID int foreign key references org\_tweet(Tweet\_ID),

User\_ID int foreign key references UoT(User\_ID) on delete no action on update no action,

Date\_Time Datetime,

Retweet\_bit bit,

primary key (Tweet\_ID, User\_ID)

)

create table tweet\_trend

( Tweet\_ID int foreign key references org\_tweet(Tweet\_ID),

Trend\_ID int foreign key references Trend(Trend\_ID),

primary key (Tweet\_ID, Trend\_ID)

)

create table message(

msg\_Id int identity(1,1) primary key,

Sender\_Id int not null foreign key references UoT(User\_ID),

Reciever\_Id int not null foreign key references UoT(User\_ID),

Message varchar(500),

[Date and Time] datetimeoffset

)