GhostTalk.com

Software Requirements Specification

Version 1.0

January 21, 2012

Calvin Moore

Dominic Kotwica

Christian Argueta

1. **Introduction**

Over the past few years, we have seen a sharp increase in the number of users of social media websites. Recently a new type of social media has begun to appear and is catching on quite rapidly. This new trend is being called temporary social media (TSM for short). TSM has caught the eyes of many who wish to share their thoughts and actions, but without the permanent repercussions of other social media websites.

With these recent trends, it had been decided to design a TSM website called GhostTalk.com. GhostTalk.com will be a hybrid of both the older social media and the newer TSM to create a one-stop for our users’ social needs. What is meant by this previous statement is that while many of the posts will be temporary and deleted, some posts should be permanent when the user specifies. This feature will allow the user to keep messages and posts that are important to them, such as “I’m getting married today” and pictures taken at the wedding. Not only will this attract users of TSM, this will give our website popularity amongst social media users, too.

While deleting a post from the database after a certain amount of time may save space and increase security, it would limit the functionality of our website. To further explain, if a post is deleted from the database, it will no longer be viewable to anybody, including users that have not seen the post yet. This would greatly upset many of our prospective users and deter use of the site. While privacy is very important, most of our users will not need fear a cyber-attack from those they are hiding their posts. For this reason, the database will not permanently delete posts, but the user may choose to permanently delete a post from the database. This will allow for increased security that the user can control and allow the user to remove a post they regret before others possibly view the post.

The main users of GhostTalk.com will be teenagers, who want to connect to friends in private, and adults, who do not want companies finding information about them that they would rather keep between friends. As such, privacy is of the upmost concern while designing this website. Messages should not be able to be accessed by anyone after specified time. Furthermore, the upmost concern should be taken with copying the picture from the website. Creating a feature that will handle screenshot may deter users from copying a post without the original user’s permission.

In addition to the privacy, usability will also be a major concern of this project. GhostTalk.com will be a good outlet for many people, and, as such, should not add to the list of their frustrations. The website should have a clear and logical design that will allow users easily make posts and read their received posts. All features of GhostTalk.com should be clear and easy to find, as opposed to similar other websites.

1. **Definitions, Acronyms, and Abbreviations**

* **TSM –** Temporary Social Media, a medium in which users can interact with friends, families, and other acquaintances by sending posts through the medium to them that will last a short amount of time
* **ER Model** – Entity Relational Model
* **Posts** – messages and images sent to another user through a social media or TSM website
* **TSM Hybrid** – a hybrid between contemporary social media websites in which posts are permanent and the newest trend of TSM. This hybrid allows users to decide how long a post should remain on a website, including forever
* **Friends** – other users that a certain user knows and to which this user is connected through the social media site
* **Friendship** – signifies that a user has a connection with another user and therefore can receive posts from each other

**3. Scope and Constraints**

1. GhostTalk.com should provide temporary and permanent posts to certain users.
   1. The option will be given to the user to choose the amount of time the post should remain available for other users to view.
   2. The user should be able to determine which users receive their post, whether it be a certain user or a group that the user created.
   3. Posts should not be made public so that users outside of their friends can view any posts.
2. GhostTalk.com should allow users to requests friends and allow user to reject friendships.
   1. After a user accepts a friendship, the user should be able to remove the user as a friend.
   2. The user should be able to hide their profile from users they do not wish to know that they have an account.
      1. This feature may be useful for people who may feel like the other user would bully them or pressure them to do things unwise. Furthermore, it would make it easier for an employee, for example, to hide their profile from an employer.
3. GhostTalk.com will not identify possible friends for the user.
4. While users will be able to search for other users, they will not be able to search for hashtags of pictures sent to them or text within a message.
5. The database should not automatically delete posts from the database.
6. **Data Model**

This section will contain both the proposed schema and ER Model for GhostTalk.com. The schemas for the various entities are

User(

username: VARCHAR(15),

password: VARCHAR(20),

name: VARCHAR(255),

email: VARCHAR(255),

currentCity: VARCHAR(255),

birthdate: DATE,

age: INT

)

Requests(

senderName: VARCHAR(15),

receiverName: VARCHAR(15)

)

Groups(

name: VARCHAR(15)

)

Posts(

timestamp: TIME,

expirationTime: INT,

body: VARCHAR(255),

pic: BOOLEAN,

read: BOOLEAN

)

**User Entity**

The username should be no longer than 15 characters while the password can contain up to 20 characters. As names, emails, cities names can get quite long, the database will store up to 255 characters so as not to limit these attributes.

**Requests Entity**

Since requests are unique in that only one can be sent from a certain user to another certain user, no key or attributes are needed for a request.

**Groups Entity**

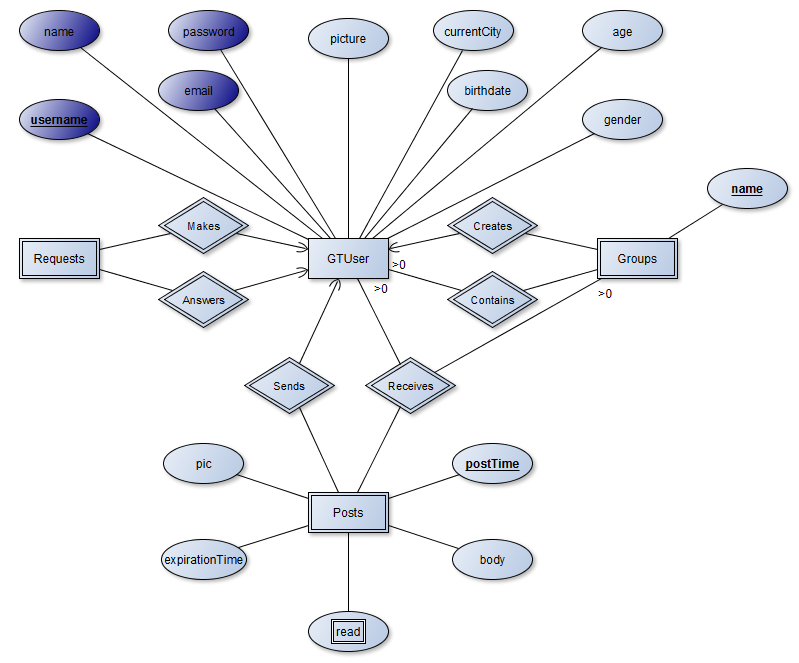
The Groups entity contains two partial keys. One is called name, which will be the name assigned to the group by the user entity, and the other will be the username of the user that created the group.

**Posts Entity**

The timestamp attribute will contain the exact time the post was made by the user entity and will include milliseconds. This attribute will be a partial key along with the username of the user that created the post. Since it is very unlikely that a single user will create two posts at the same exact millisecond, these two keys should make a unique key. Furthermore, each post will contain an expirationTime. This attribute will store the time allotted for each user to view the post. The Message and Picture entities have an “isa” relationship with posts.

While these last three entities may be thought of as relationships themselves, I personally find this way easier to comprehend and a better notation.

Below is the ER Model of the above schema. An alternative form is available upon request in which the Requests, Posts, and Groups entities are instead relationships.



1. **Functional Requirements**
2. Creating an account
   1. When a new user first uses GhostTalk.com, they will have to create a new account. This account will have to have a unique username (between 5 and 15 characters) and a password larger than 7 characters.
   2. Other information that will be required when the user logs in will include:
      1. Email Address
      2. Name
      3. Birthdate
      4. Age
3. Finding and Adding Friends
   1. The user should be able to seach for another user using one or more of the following attributes
      1. Username
      2. Email Address
      3. Name
      4. Users can narrow seaches further using the person’s birthdate or age, but cannot search based on these attributes
      5. When a search is conducted, the results should appear in alphabetical order by name, then username, then email address.
   2. Users can request other users to be their friends
      1. Requests can be accepted or denied by the reciever
      2. Requests can be cancelled by the sender
      3. If a user does not want another user to see them, they can hide from this user
4. Creating/Viewing Posts
   1. When creating a post, users will be able to set the amount of time they wish the post to be viewed by the users receiving the post
   2. Posts times should include:
      1. 1-10 seconds
         1. When any post is only has this long left before it is removed from the viewer’s sight, a countdown should begin in the top corner of the post.
      2. 1 minutes
      3. 5 minutes
      4. 1 hour
      5. Permanently
   3. After a post has been viewed by the other user for the time alloted, the post should no longer be visible to the user.
   4. Once a post has been viewed, the countdown should not stop until it has run out, even if the viewer is not on that particular post.
   5. If a user does not want to see a post anymore, they should be able to delete it from their view permanently.
5. Searching Through Posts
   1. While posts cannot be searched for hashtags or text within a message, posts should be able to be sorted and searched for by date sent or for the user that sent the post.
   2. When the search returns, the dresults should be sorted according to username.
6. Taking a Screenshot of the Webpage
   1. GhostTalk.com should offer a way to take screenshots of a page
      1. When a user wants to take a screenshot of a picture, GhostTalk.com should ask the other author if this is ok before the user has access to the screenshot.
         1. This function will prevent users from taking screenshots that the author would rather keep between theirselve and their friends.
      2. As a precaution, when the GhostTalk.com window does not have focus, a popup or picture should appear to block any attempts to get a screenshot outside of the program.
         1. While this method will not be perfect it should make it inconvient enough to discourage screenshots without the author’s permission.
         2. This popup would be the website’s logo and should explain why the screen is no longer visible.
7. Creating Groups
   1. Every user has one group, this is the group of all of their friends.
   2. After a user has at least one friend, the user can begin to group their friends. These groups will make sending posts to a group of users easier.
   3. Groups will have a name of the user’s choosing.
8. User Interface
   1. The user interface should have easy access to all features and have a aesthetically pleasing GUI.
   2. The interface should not show all posts at once as many social media sites have as a layout.
      1. This will allow users to view posts without accidentally setting off a timer for another post.
      2. Posts should rather be viewable through an interface similar to an email interface.
      3. Before a user opens a post, they should see the following features of the user that sent the post:
         1. Username
         2. Name
         3. Expiration Time
         4. Time sent
9. Deleting a Post from GhostTalk.com Permanently
   1. If a user does not wish a post to be available to him or any friends who have not seen the post yet, they should be able to delete the post from the database.
   2. When deleting a post from the database, a warning message should appear telling the user the possible risks of other users not seeing the post yet.
      1. If the user accepts, the post will be permanently deleted from the database.
      2. If the user declines, the post will remain in the database.