

Social Networking: Project Report

Vlife(VirtualLife)

Naveen Kumar
140050013

Yathansh Kathuria
140050021

Rajat Chaturvedi
140050027

Suman Swaroop
140050032

November 3, 2016

INTRODUCTION

Social Networking is an on-line platform where people from all over the globe can communicate with each other and share their life moments easily. This Project encompasses all basic features of communication required with few extra functionality.

- ◇ Basic Functions: Account creation, Friends, Group Page/Group Communications, Posts, Events, Feed, Media Uploading, Tagging in posts.
- ◇ Extra Functionality: SlamBook, One to One Game Challenges, Group Messaging.

FUNCTIONAL SPECIFICATIONS:

This section explains each of the functionality in detail.

1. Account Creation

When a new user visits the site for the first time, he/she is asked to sign up on the website. The signing up process is divided into two steps. The first step involves filling up the login information which includes entering the mail address, phone number and the password. Each account should have a different email address and phone number. Step two deals with the personal information of the user. The user is asked to input his/her first, middle and last name, gender, date of birth, relationship status etc.. Most of these fields are optional except first name, gender and date of birth. Once the details are filled and verified, the user is redirected back to the login page.

2. Feed

Feed is the place where the account owner/user can see the various activities of the people he/she is friend with. These activities include any status update, media shared by friends and other feeds related to his/her interests. Posts to be shown on Feed will be generated algorithmically based on time, the user posted, number of likes/comments on posts and others such criteria.

3. Friends

Friends form the backbone of Social Networking. A user can add friend or remove anyone from their list of friends. A search bar to search for friends/groups based on their names/location. A User can go through a another user's accounts based on the privacy policy and send him a friend request. The other user receives a request which he/she can accept/ignore/decline. Following a user is also possible in which the user can have the posts of the followed user in his feed but not vice-versa.

4. Chatting/Private Messages

Along with posting public status and other medias, one can also send private messages to a specific person or a group of person using the chat feature. This is a point to point communication feature rather than a broadcast one and is only visible to the sender and the receiver.

Active/Inactive users list is categorized and listed on the chat bar. If both the user is online then real-time chatting is implemented just by querying the database again and again when messages

are exchanged.

5. Events

User can create Events specifying its small description along with the date and venue of the event and invite other users for the same. All invited users will get a notification regarding the event. Other users then can choose among the options going/interested/maybe regarding the event. All statistics related to the event will be displayed on the event page created.

6. Media-Photo/Audio/Video

Each user will have their own media section (this doesn't mean cloud storage for pictures). All media uploaded on the site by the user can be viewed on his/her Album section. For the Media Posted on other's Timeline/Wall, the other user just has a reference to the media. Both the user A and user B has the right to remove it.

7. Privacy Options

There are three levels of privacy options available for users to choose from: Open, Friends, Closed. Just after account creation the user's privacy will be set to open by default.

- ◇ Open: Profile information/Posts of the user is visible to everyone else.
- ◇ Friends: Profile Information/Posts (by the user) visibility is restricted only to friends.
- ◇ Closed: Profile Information not disclosed to anyone except the user himself. Post's visibility is restricted to friends. Current Profile Photo and Name of the user is publicly visible to everyone irrespective of the privacy option chosen.

8. Wall/Timeline

Each user has its own wall where the user can post or his/her friend can post text/media on his/her wall expressing views/thoughts. It is basically a way for many user to communicate publicly or simply leave their thoughts for each other.

9. Group Page

- ◇ Group Page has its own Wall. All Posts posted on this wall can be viewed only by the users in the group. Group Content is limited only to group members.
- ◇ The group page has an Admin. Admin has the right to remove or invite users to the group.
- ◇ Admin cannot remove himself from the group until and unless he/she either deletes the group or there are multiple admins.

10. SlamBook

SlamBook acts as a Personal virtual SlamBook for a particular user. Any Friend of a user can fill the virtual book. The book will have some non-trivial questions based on personal attachment.

- ◇ Each user can fill the book of a particular user only once. However he can edit it later on or remove his/her entry.
- ◇ None of the friends can view the SlamBook entries of another user except his own entry.
- ◇ A user has right over his own SlamBook. The User can delete a entry he/she doesn't like but can't edit it.

EXTRA SPECIFICATIONS(TENTATIVE)

1. One to One Game Challenges

We have thought of including a small gaming platform where a user can challenge other user for small multiplayer games. These games are made available by external gaming sites like miniclip.com.

2. Group Messaging

In this multiple users can chat in the same chat box. Each message send by a user is broadcasted/sent to all users in that group.

RELATIONAL SCHEMA

1. table USER:

attributes:

- a) u_id varchar(10) not null
- b) password char(128) not null
- c) email_id varchar(256)
- d) phone_no numeric(10,0)
- e) date_time timestamp

primary key u_id

check (email_id is not null or phone_no is not null)

2. table USER_PROFILE

attributes:

- a) u_id varchar(10) not null
- b) first_name varchar(50) not null
- c) middle_name varchar(50)
- d) last_name varchar(50)
- e) gender varchar(10) not null
- f) language varchar(100)
- g) bday date not null
- h) rel_status varchar(30)
- i) privacy varchar(10) not null
- j) country varchar(60)
- k) state varchar(60)
- l) city varchar(60)

primary key u_id

foreign key u_id referencing USER

foreign key country, city referencing LOCATION

check privacy in ('open', 'friends', 'closed')

3. table FRIEND

attributes:

- a) user varchar(10) not null
- b) friend varchar(10) not null
- c) status varchar(15) not null

primary key user, friend

foreign key user referencing USER

foreign key friend referencing USER

check status in ('waiting', 'following', 'accepted')

4. table LOCATION

attributes:

- a) country varchar(60) not null
- b) state varchar(60)

c) city varchar(60) not null
primary key city, country

4. table INSTITUTION

attributes:

- a) ins_id varchar(10) not null
- b) name varchar(150) not null
- c) country varchar(60)
- d) state varchar(60)
- e) city varchar(60)

primary key ins_id

foreign key country, city referencing LOCATION

5. table POST

attributes:

- a) p_id varchar(10) not null
- b) content text not null
- c) date_time timestamp
- d) country varchar(60)
- e) state varchar(60)
- f) city varchar(60)
- g) posted_by_id varchar(10) not null
- h) media_id bytea
- i) posted_to_id varchar(10)
- j) page_id varchar(10)

primary key p_id

foreign key country, city referencing LOCATION

foreign key posted_by_id referencing USER

foreign key media_id referencing BLOB

foreign key posted_to_id referencing USER

foreign key page_id referencing GROUP_PAGE

check (posted_to_id is not null xor page_id is not null)

6. table GROUP_PAGE

attributes:

- a) page_id varchar(10) not null
- b) description varchar(100)
- c) date_time timestamp
- d) page_pic bytea

primary key page_id

foreign key page_pic referencing BLOB

7. table QUESTION

attributes:

- a) q_id varchar(3) not null
 - b) question_description varchar(100) not null
- primary key q_id

8. table SLAM

attributes:

- a) slam_id varchar(10) not null
- b) date_time timestamp not null
- c) user_1 varchar(10) not null
- d) user_2 varchar(10) not null

primary key slam_id
foreign key user_1 referencing USER
foreign key user_2 referencing USER

9. table SLAM_QUEST

attributes:

- a) slam_id varchar(10) not null
- b) q_id varchar(10) not null
- c) answer text not null

primary key slam_id, q_id

foreign key slam_id referencing SLAM

foreign key q_id referencing QUESTION

10. table MESSAGE

attributes:

- a) msg_id varchar(15) not null
- b) content varchar(1024)
- c) user_1 varchar(10) not null
- d) user_2 varchar(10) not null
- e) media_id bytea

primary key msg_id

foreign key user_1 referencing USER

foreign key user_2 referencing USER

foreign key media_id referencing BLOB

11. table BLOB

attributes:

- a) med_id varchar(12) not null
- b) description varchar(100)
- c) content bytea not null
- d) country varchar(60)
- e) state varchar(60)
- f) city varchar(60)

primary key med_id

foreign key country, city referencing LOCATION

12. table EVENT

attributes:

- a) eve_id varchar(10) not null
- b) description text not null
- c) date_created timestamp
- d) date_event timestamp not null
- e) country varchar(60) not null
- f) state varchar(60)
- g) city varchar(60) not null
- h) created_by_id varchar(10) not null

primary key eve_id

foreign key country, city referencing LOCATION

foreign key created_by_id referencing USER

check date_event < current_timestamp

13. table EVENT_INVITES

attributes:

- a) eve_id varchar(10) not null

b) u_id varchar(10) not null
c) status varchar(15) not null
primary key eve_id, u_id
foreign key eve_id referencing EVENT
foreign key u_id referencing USER
check status in ('invited', 'notgoing', 'maybe', 'going')

14. table NESTED_POST

attributes:

a) p_id1 varchar(10) not null
b) p_id2 varchar(10) not null
primary key p_id1, p_id2
foreign key p_id1 referencing POST
foreign key p_id2 referencing POST

15. table GROUP_USER

attributes:

a) page_id varchar(10) not null
b) u_id varchar(10) not null
primary key page_id, u_id
foreign key page_id referencing GROUP_PAGE
foreign key u_id referencing USER

16. table NOTIFICATION

attributes:

a) not_id varchar(12) not null
b) content varchar(60) not null
c) eve_id varchar(10)
d) p_id varchar(10)
primary key not_id
foreign key eve_id referencing EVENT
foreign key p_id referencing POST
check (eve_id is not null xor p_id is not null)

17. table NOTIFY_TO

attributes:

a) not_id varchar(12) not null
b) u_id varchar(10) not null
primary key not_id, u_id
foreign key not_id referencing NOTIFICATION
foreign key u_id referencing USER

18. table POST_LIKE

attributes:

a) p_id varchar(10) not null
b) u_id varchar(10) not null
foreign key p_id referencing POST
foreign key u_id referencing USER_PROFILE

CURRENT PROGRESS:

Till date the following features have been completed:

1. Account Creation

Users can visit the site and can create their account on vLife

2. Feed

Every user has a news feed, which for now contains all the posts made by all other users (no filtering algorithm is being used)

3. Friends

For the time being you can either accept a friend request, or leave it as it is, but you cannot delete it.

4. Timeline

Each user has his/her own timeline but notifications and the option to edit the profile are still pending

5. Media

As of now users can upload only images (and not audio and video). They can also set their profile pictures

6. Chatting

The chat option is up and running but still lacks a presentable interface

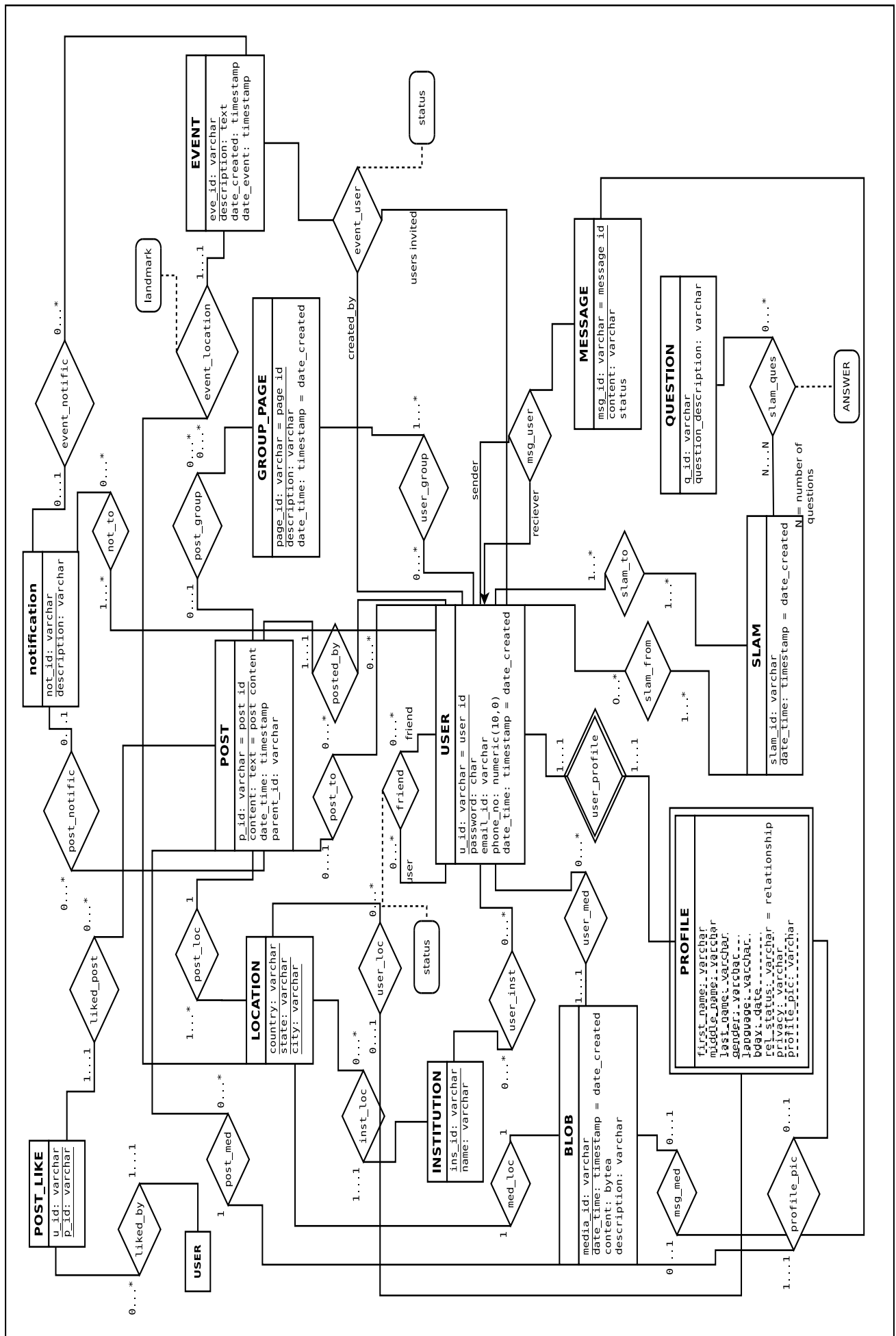


Figure 1: ER- Diagram