

1.INTRODUCTION

Social Networking - It's the way the 21st century communicates now. Social networking is the grouping of individuals into specific groups, like small rural communities or a neighborhood subdivision. Although social networking is possible in person, especially in the workplace, universities, and high schools, it is most popular online. This is because unlike most high schools, colleges, or workplaces, the internet is filled with millions of individuals who are looking to meet other people.

Social network is the mapping and measuring of relationships and flows between people, groups, organizations, computers, URLs, and other connected information/knowledge entities. The nodes in the network are the people and groups while the links show relationships or flows between the nodes. Social network provides both a visual and a mathematical analysis of human relationships.

Social Networking Website project itself is a huge project comprising various features like profile updating, friend's list organization and various other application to enhance the overall look and feel of the website. However, in this project I am basically working on two essential feature or module (PROFILE MANAGEMENT & FRIENDS ORGANIZATION).

PROFILE MANAGEMENT module maintain the profile of a user like name, like, dislikes, hobbies, status etc.

FRIENDS ORGANIZATION module maintains the friend list, handles request and sends request to the other user.

Profiles and Friends lists are two key features on social network sites. The third is a public commenting feature ('Testimonials', 'Comments', 'The Wall'). This feature allows individuals to comment on their Friends' profiles. These comments are displayed prominently and visible for anyone who has access to that profile.

1.1 Problem Statement

We define social network sites as web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site.

Since their introduction, social network sites (SNSs) such as MySpace, Facebook, Cyworld and Hi5 have attracted millions of users, many of whom have integrated these sites into their daily practices. As of this writing, there are hundreds of SNSs, with various technological affordances, supporting a wide range of interests and practices. While their key technological features are fairly consistent, the cultures that emerge around SNSs are varied. Most sites support the maintenance of pre-existing social networks, but others help strangers connect based on shared interests, political views, or activities. Some sites cater to diverse audiences, while others attract people based on common language or shared racial, sexual, religious, or nationality-based identities. Sites also vary in the extent to which they incorporate new information and communication tools, such as mobile connectivity, blogging, and photo/ video-sharing.

Social networking sites are not only for you to communicate or interact with other people globally but, this is also one effective way for business promotion. A lot of business minded people these days are now doing business online and use these social networking sites to respond to customer queries. It isn't just a social media site used to socialize with your friends but also, represents a huge pool of information from day to day living.

A social networking service is an online service, platform, or site that focuses on facilitating the building of social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Most social network services are web-based and provide means for users to interact over the Internet, such as e-mail and instant

messaging. Online community services are sometimes considered as a social network service, though in a broader sense, social network service usually means an individual-centered service whereas online community services are group-centered. Social networking sites allow users to share ideas, activities, events, and interests within their individual networks.

1.2 Domain Study

As of May 2013, almost three quarters (72%) of online U.S. adults use social networking sites, up from 67% in late 2012. When we first started asking about social networking sites in February 2005, just 8% of online adults said they used social networking sites.

Today, social networking site use is a major activity for internet users from a wide range of demographic groups. Younger adults are especially avid adopters, but social networking continues to grow in popularity for older adults as well. Six out of ten internet users ages 50-64 are social networking site users, as are 43% of those ages 65 and older. Although online seniors are less likely than other age groups to use social networking sites, adoption rates for those 65 and older have tripled in the last four years (from 13% in the spring of 2009 to 43% now).

The main types of social networking services are those that contain category places (such as former school year or classmates), means to connect with friends (usually with self-description pages), and a recommendation system linked to trust. Popular methods now combine many of Facebook, Google+, YouTube, LinkedIn, Instagram, Pinterest, Tumblr and Twitter widely used worldwide; Nexopia in Canada; Badoo, Bebo, VKontakte (Russia), Delphi (also called Delphi Forums), Draugiem.lv (mostly in Latvia), Hi5 (Europe), Hyves (mostly in The Netherlands), iWiW (mostly in Hungary), Nasza-Klasa, Soup (mostly in Poland), Glocals in Switzerland, Skyrock, The Sphere, StudiVZ (mostly in Germany), Tagged, Tuenti (mostly in Spain), and XING in parts of Europe; Hi5 and Orkut in South America and Central America; Mxit in Africa; and Cyworld, Mixi, Orkut, renren, weibo and Wretch in Asia and the Pacific Islands.

Many of these early communities focused on bringing people together to interact with each other through chat rooms, and encouraged users to share personal information and ideas via personal web pages by providing easy-to-use publishing tools and free or inexpensive web space. Some communities - such as Classmates.com - took a different approach by simply having people link to each other via email addresses. In the late 1990s, user profiles became a central feature of social networking sites, allowing users to compile lists of "friends" and search for other users with similar interests. New social networking methods were developed by the end of the 1990s, and many sites began to develop more advanced features for users to find and manage friends. This newer generation of social networking sites began to flourish with the emergence of SixDegrees.com in 1997 followed by Makeoutclub in 2000, HubCulture and Friendster in 2002 and soon became part of the Internet mainstream. Friendster was followed by MySpace and LinkedIn a year later, and eventually Bebo. Friendster became very popular in the Pacific Island. Orkut became the first social networking in Brazil and then also grew fast in India (Madhavan, 2007). Attesting to the rapid increase in social networking sites' popularity, by 2005, it was reported that MySpace was getting more page views than Google. Facebook, launched in 2004, became the largest social networking site in the world in early 2009. Facebook was first introduced (in 2004) as a Harvard social networking (Cassidy, 2006).

1.3 Existing Systems

According to the definition above, the first recognizable social network site launched in 1997. SixDegrees.com allowed users to create profiles, list their Friends and, beginning in 1998, surf the Friends lists. Each of these features existed in some form before SixDegrees, of course. Profiles existed on most major dating sites and many community sites. AIM and ICQ buddy lists supported lists of Friends, although those Friends were not visible to others.

SixDegrees was the first to combine these features. SixDegrees promoted itself as a tool to help people connect with and send messages to others. While SixDegrees attracted millions of users, it failed to become a sustainable business and, in 2000, the service closed. Looking back, its founder believes that SixDegrees was simply ahead of its time

(A.Weinreich, personal communication, July 11, 2007). While people were already flocking to the Internet, most did not have extended networks of friends who were online. Early adopters.com explained that there was little to do after accepting Friend requests, and most users were not interested in meeting strangers.

Classmates.com allowed people to affiliate with their high school or college and surf the network for others who were also affiliated, but users could not create profiles or list Friends until years later.

From 1997 to 2001, a number of community tools began supporting various combinations of profiles and publicly articulated Friends. AsianAvenue, BlackPlanet, and MiGente allowed users to create personal, professional, and dating profiles— users could identify Friends on their personal profiles without seeking approval for those connections (O. Wasow, personal communication, August 16, 2007). Likewise, shortly after its launch in 1999, LiveJournal listed one-directional connections on user pages. LiveJournal's creator suspects that he fashioned these Friends after instant messaging buddy lists (B. Fitzpatrick, personal communication, June 15, 2007)—on LiveJournal, people mark others as Friends to follow their journals and manage privacy settings. The Korean virtual worlds site Cyworld was started in 1999 and added SNS features in 2001, independent of these other sites (see Kim & Yun, this issue). Likewise, when the Swedish web community LunarStorm refashioned itself as an SNS in 2000, it contained Friends lists, guestbooks, and diary pages (D. Skog, personal communication, September 24, 2007).

1.4 Project Scope

- This system provides users to register their various types of profile like social, personal, general, professional.
- This system provides users to send a scrap message, images, and data files to their friends. User can maintain the scrap book whatever scraps he has send to users.
- The system provides user to upload the photos so that user can maintain own album.
- This system provides user to join the communities according to their scenario.

- This system provides the user to maintain their friend list and user can update their friend list.
- This system provides user to send invitation to another friend and can add to their friend list for future.

1.5 Organization Profile

Incorporated in 2004, as JAVASTREAM Technologies and later got registered as SYSINNOVA InfoTech Pvt. Ltd., it's an ISO 9001:2000 certified organization that operate through well-defined systems and procedures. They have been relentlessly endeavoring to provide end to end solutions to the Information Technology Industry. With our expertise developed through the profound experience we continue contributing in Technology Consulting, Software Development, Training & Talent Management Consulting and Software Development Services to the clients globally. With their headquarters in Bangalore, the IT silicon valley of India, their activities are spread over many locations in Karnataka and other states in India .A formidable manpower of over 58 qualified professionals and strong associations with Global Technology Leaders such as Oracle & SAP are undoubtedly their core strengths. SYSINNOVA InfoTech is an offshore software services and IT consulting company based in Bangalore, India. As a committed outsourcing partner and an IT vendor, our goal is to ensure cost effective, technical excellence and on-time deliveries. While they take care of their end-to-end programming and consulting needs, their clients focus on core business activities which correlate directly to their revenues and profitability. Strategic partnership with them gives their clients the access to latest technology, skilled manpower and scalable team which ultimately results in lower risk and higher ROI. Our core competency lies in web technologies, be it Java-J2EE, Spring, Hibernate, Oracle-XML Publisher, DBA tuning, Oracle Application implementation, PHP and the associated frameworks and CMSs like Joomla, Drupal, SharePoint (MOSS).

For over 7 years, they have been working with their clients to bring their creative ideas onto the web. Our typical engagement with our clients is that of an offshore IT vendor. Today, more than 50 mid sized enterprises and media agencies across the world rely on

them to save their invaluable time, resources and money. Their specialized technical and domain expertise can be hired for building modern web sites, web applications and complex enterprise applications.

2. SYSTEM REQUIREMENTS

2.1 Literature Survey

The Web-based social networking services make it possible to connect people who share interests and activities across political, economic, and geographic borders. Through e-mail and instant messaging, online communities are created where a gift economy and reciprocal altruism are encouraged through cooperation. Information is suited to a gift economy, as information is a non rival good and can be gifted at practically no cost.

Facebook and other social networking tools are increasingly the object of scholarly research. Scholars in many fields have begun to investigate the impact of social-networking sites, investigating how such sites may play into issues of identity, privacy, social capital, youth culture, and education.

Several websites are beginning to tap into the power of the social networking model for philanthropy. Such models provide a means for connecting otherwise fragmented industries and small organizations without the resources to reach a broader audience with interested users. Social networks are providing a different way for individuals to communicate digitally. These communities of hypertexts allow for the sharing of information and ideas, an old concept placed in a digital environment.

In 2011, HCL Technologies conducted research that showed that 50% of British employers had banned the use of social networking sites/services during office hours.

2.2 Functional Specifications

- **Server Object**

The Server class acts as a wrapper for all server functions for our social networking site. It essentially act as a link between all of the information such as accounts, account details, pages, notes, etc to our database. When any other model object such as a page is pulled from the server, a temporary copy is made. If that temporary copy is changed in any way. The new version must be sent to the server in order to update the permanent copy. The reason behind local copies is that all the necessary information for the object is sent over in one easy-to-use package. Then the update to the database can be done all at once by sending back that single object. There is no need for multiple functions or a function that takes a large number of parameters.

- **Account Object**

Each user who wants to use the site must create an account. This is the head class that all other objects use to determine what a user does and when the user did it. The account's information has four purposes: hold the login information, hold friend information, hold profile information, and hold privacy information with such a large amount of information to keep track of, the Account class would be very large and difficult to work with. Therefore to ease the load, the Account class was broken up into three different classes. There is the actual account class which keeps track of login information and friend information. It also holds the other two classes within it. Profile information was outsourced to the Account Details class, and privacy settings were outsourced to the Privacy Settings class. The only time the Account class needs to be updated is when the user changes his/her username and/or password. All other settings are handled by the Account Details and Privacy Settings classes.

- **Account Details Object**

An Account Details object is a helper class created whenever a new Account object is created. The object contains all the information that shows up in the user's profile. The

user can edit this by modifying his/her profile. Overall, this class has no other purpose but to be a helper class to its account object.

- **Privacy Settings Object**

A Privacy Settings object is the other helper class created whenever a new Account object is created. This object contains all the privacy settings that a user has, such as who can view his/her media or custom pages. This class is called any time a user visits a profile or content created by another user. However, it does not directly interact with the other model classes, only the view.

- **Chat Session Object, Event Object, Link Object, Note Object and Page Object**

These objects contain unique information for a particular type of action a user performs. All of these contain a reference to the account that owns them. Each object is a "working-copy" of an object in the Server. Anytime one of these objects is created on the Server, an entry of its creation is added to the news feed database.

- **Message Object**

A Message object is created when a user composes a new message to be sent to a friend. After it is confirmed that the friend is located in the database, the Message object adds its information to the database. When a user checks his or her inbox, a list of messages that were sent to the user will be shown in descending order of when they were received.

- **Wall Post Object**

A Wall Post object works very similar to a Message object. The only difference is that the Wall Post objects are viewable to all friends of the user. Another difference is that a Wall Post object can hold comments. If a user adds a wall post to his or her own account, their status will change to the new wall post.

- **Friends**

The most important feature is being able to add and remove friends. In our Social Networking site, making friends is a fairly straightforward process. Users can type in the name of a friend in the search bar at the top of their home page. The database is queried for an account that has the search term contained in the full name, any media files with the search terms in the description, any pages with the search terms in the titles, any links with the search terms

in the titles, and any notes with the search terms in the title. For example, User A could search for User B in the search bar. After clicking on User B's profile, User A will see a button that says Send Friend Request. Clicking on it will send an alert to User B that User A wants to be a friend. The friend request will now be in the friends list of User B, where he/she can either accept it or ignore it, letting it sit there indefinitely. If user B accepts the request, User A will be added to User B's friends list and vice versa. Being friends has its advantages. For example, only friends can chat to each other. Also, friends can view any part of a profile that is marked as friends only. Finally, for a user to view his/her friends and incoming friend requests, he/she just clicks on the friends tab which brings up a frame.

- **Account Creation**

When a user accesses the site for the first time, he/she must create an account before using any of the site features. The account creation process is broken into three sections. The first section deals with the login information and is required for the user to fill out. This includes the email, password, and password confirmation. The purpose behind the password confirmation is to ensure that the user didn't accidentally mistype when creating a password. The second section deals with information about who you are such as name, location, and gender. Most of these fields are optional except for your name and gender. It wouldn't be much of a social network if everyone was named anonymous. The final section deals with information about the users likes and dislikes, such as interests and activities. Unlike the other two sections, this section is completely optional. Once the

user clicks create account, a new account, account details, and privacy settings are added to the server, and the user is brought back to the login page

- **Privacy Options**

Privacy is very important feature for some people, and social networking is no exception to this. Our system provides three levels of privacy: open, friends only, and closed. A feature with an open privacy level is public and may be viewed by anyone. By contrast, a feature with a closed privacy level is completely private and can only be viewed by the account owner. A feature with a friends only privacy level is fairly self explanatory. By default, all privacy levels are set to open when a new account is created. To change privacy levels, the user can click on the options tab on the top menu bar. The current features with privacy levels are media share, account wall, user pages, notes, and the overall profile privacy.

The implementation of privacy is a fairly straightforward process. When the user makes changes to the privacy level, the Privacy Settings object in the account is updated with the new privacy settings. That working copy is then passed into the update method of the server, and that queries the database to set the record for that privacy settings object to the new values. When a user visits another profile, the profile owner is pulled from the database. A function called `can view page()` is and is passed the profile owner, the viewer profile and the page type. In the function the privacy settings object is pulled from the viewer profile and the page type determines which privacy setting is pulled from the object. If the setting is open, the function automatically returns true. If the setting is friends only, the function only returns true if the viewer profile is a friend of the page owner or if the viewer profile and the page owner are the same. Finally, if the setting is closed, the function only returns true if the viewer profile is the page owner. Once the function returns its answer the page will either do one of two things. If the function returns true, the viewer is allowed to view this page, and it will load normally. If the function returns false, the viewer is not supposed to view the page, and it will redirect to an error page.

- **Chatting**

One of the advantages of having friends is the ability to have live communications with them via chat. When a friend is online, he/she will appear in the chat tab as an available friend to chat with. To start a chat session, simply click on the name of the friend which will begin a new chat session. The little bar at the bottom will change from chat disabled to chatting with friend name as seen in the figure below. Also, the friend receiving the chat will get a notification that a new chat session has started. Then the two friends can chat with each other until one or both log out.

The chat feature is a little more complex than others as it requires a combination of JavaScript and JQuery to work. When a user clicks on a friend name to begin a chat, a chat request is sent to the database. On every page there is a JQuery function that queries the server for any new chat requests once a second. It needs to be JQuery, so the client can request for the server to run a check and return any relevant information. If it was pure PHP, the page would never load completely, because the page cannot finish loading until the server is done all preprocessing. The JavaScript portion of JQuery allows the client to ask the server to do more processing after the page has been loaded.[3] If there is a new chat request, a pop under is generated, notifying the receiver of the new chat that is starting or a new pending chat if the user is in another chat. When a user send a message, another JQuery function is called, to alert the server that it needs to update the chat session with a new message. At the same time, another JQuery function is running to ask the server once a second if the chat session has been updated. Overall, it is a lot of the client asking the server to send over any changes that exist.

- **Messaging**

Unlike with chatting, users can send a message to any other user. For user A to send a message to user B, he/she simply goes to the message center tab and clicks compose message. He then fills out the form as shown in the figure below with the email of the recipient, the message title, and the message content. The new message will then be in User B's inbox in the message center.

Messaging is implemented by storing a record of the message in the server. First a message object is constructed that takes in the to and from accounts, the message body, and the date it was sent. The message body is retrieved from a user submitted form, the from account is retrieved from the user session, and the receiver account is retrieved from the recipient email address. The receiver account is validated, to ensure the sender didn't try to send a message to a non-existent receiver. Finally, the new message is stored in the server. Retrieving messages sent to the user is a simple method of querying the server for any messages that have been sent to the user, and this is done by checking the receiver id of each message against the account id of the user. Any matches are returned to the inbox of the user as shown in the figure below.

- **Events**

In addition to messaging your friends, social networking sites are a great place to alert your friends of important upcoming events. For example, if a user is throwing a birthday party, he/she could create an event and invite friends he/she wants to attend or simply make it a public for anyone. The process of creating an event is as simple as filling out a form which looks like this.

After filling out the form, the user will be brought to the standard confirmation page. He/she can then view the new event by clicking on the Events tab and clicking on the new event. If the user wishes to edit some information about the event, he/she can click on the edit link next to the event where a similar form to creating an event will be displayed with all the fields populated with the current event information.

Storing the event is a fairly straightforward process. Once the form data is submitted, a new Event object is created storing the Account that created the event as well as all the information sent along with the form. That new event is then sent to the server, which creates a database query and stores the fields of the event object into equivalent fields of the event database table. Retrieving events is also a fairly straightforward process. A MySQL query is called to pull the contents of the event into an array, and that array is used to instantiate a new working copy of the event in an Event object.

- **Media Uploading**

As people use their social networking account, they will want to be able to upload funny or interesting images, video, music, etc., to share with their friends. The media upload section will be located at the media tab, where users are able to specify a file to upload as well as provide a short description of the file to be uploaded. In order to prevent users from uploading potentially malicious files such as executables, only certain file extensions are supported. These allowed extensions cover popular image extensions such as png, jpg, gif, and bitmap, video extensions wmv and avi, audio extensions mp3, wma, and wav, and some document files including txt, rtf, doc, and pdf.

The inner workings of this uploading process are surprisingly simple, since the HTTP server takes care of requesting the file from the client automatically. After the user clicks upload, the file is sent to the server and stored in a temporary location. Through PHP all information about these temporary files can be accessed through the \$ FILES variable. The temporary location is stored in that variable and can be used to pull the file name and file extension.[1] That extension is then checked against an array of allowed extensions. If the extension is in the list, the file is then moved to a permanent location in the media folder under a subfolder for the user's account. If the extension is not on the list, it is left in the temporary folder where it will be automatically deleted by the server once the php script completes.

- **Other features**

All the other features, such as the wall pages, and notes are fairly similar. Their main purpose is to add additional places to have places for additional content. All are created using similar methods, and are stored similarly but are used for different purposes.

1. Wall

The wall is a place where the account owner can express his current feeling or thoughts with a wall status. Other users can add their thoughts by creating wall posts that show up

below the status. It is basically a way for many users to communicate publicly or simply leave their thoughts about each other

2. Notes

Notes are very similar to wall posts with one major exception. Unlike wall posts which can be created by anyone, only the account owner can create notes. It's main purpose is for the account owner to create reminders for him/herself, or create announcements for any other users to see.

2.3 Non-Functional Specification

- Secure access of confidential data by user name and password. This application is secure for every kind of its users, because if any user logout from any session then nobody will be able to access his profile without knowing his confidential password.
- 24 X 7 availability
- Better component design to get better performance at peak time.
- The database used here is robust, reliable & fast. So users will have to wait for the output very short time.
- This application can be accessed from any type of platform.
- There is no case of redundancy in the database so it will not take extra memory space.
- Username & password are sent to the users via email after registration.
- Password recovery system is also provided in case of forgetting the password.

2.4 Software Tools Specification

- **MySQL**

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory

using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

- **Language: PHP**

PHP is a scripting language designed to fill the gap between SSI (Server Side Includes) and Perl, intended for the web environment. Its principal application is the implementation of web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. PHP is currently divided into two major versions: PHP 4 and PHP 5, although PHP 4 is deprecated and is no longer developed or supplied with critical bug fixes. PHP 6 is currently under development. PHP was designed by Rasmus Lerdorf to display his resume online and to collect data from his visitors.

PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications which can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.

PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a interface capability and can be used in standalone graphical applications.

The PHP language was originally implemented as an interpreter, and this is still the most popular implementation. Several compilers have been developed which decouple the PHP language from the interpreter. Advantages of compilation include better execution speed, static analysis, and improved interoperability with code written in other languages.

PHP includes free and open source libraries with the core build. PHP is a fundamentally Internet-aware system with modules built in for accessing File Transfer

Protocol (FTP) servers, many database servers, embedded SQL libraries such as embedded PostgreSQL, MySQL, Microsoft SQL Server and SQLite, LDAP servers, and others. Many functions familiar to C programmers such as those in the studio family are available in the standard PHP build.

- **Back End: PHP MyAdmin**

PHPMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. PhpMyAdmin supports a wide range of operations on MySQL, Maria DB and Drizzle. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

Features provided by the program include:

1. Web interface
2. MySQL database management
3. Import data from CSV and SQL
4. Export data to various formats: CSV, SQL, XML, PDF (via the TCPDF library), ISO/IEC 26300 - Open Document Text and Spreadsheet, Word, Excel, LaTeX and others
5. Administering multiple servers
6. Creating PDF graphics of the database layout
7. Creating complex queries using Query-by-Example (QBE)
8. Searching globally in a database or a subset of it
9. Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
10. Live charts to monitor MySQL server activity like connections, processes, CPU/Memory usage, etc.

2.5 Hardware Requirements

Processor	:	Intel Pentium IV 2.0 GHz and above
RAM	:	512 MB and above
Hard disk	:	80GB and above
Monitor	:	CRT or LCD monitor
Keyboard	:	Normal or Multimedia
Mouse	:	Compatible mouse

2.6 Software Requirements

Front End	:	LAMP
Language	:	PHP
Back End	:	PHP MyAdmin
Operation System	:	Windows XP or above
Browser	:	Any latest browser

3 Design Specifications

3.1 Modular Design

The application comprises the following major modules:

Register to be a member Module

This module provides functionalities for those people who wants to open an account. Applicants can post their views with personal and professional details. They can also update the profile as frequently as required. The member can also browse through the

friends profile available. Members can also get message alerts when their friends message them.

Profile Module

This module provides functionalities related to members profile. Logged users can see their details and if they wish to change any of their information they can edit it.

Admin Module

This module provides administrator related functionalities. Administrator manages entire application and maintains the profiles of all the registered users and their activities.

3.2 System Design

3.2.1 Data Flow Diagrams

Data flow diagrams model the flow of data into, through, and out of an information system:

- show the processes that change or transform data
- show the movement of data between processes
- represent a system as a network of processes which transform data flowing between them

The user screen flow shows what a user of the community will see. After successfully logging on, the user will be given various links (such as search users, search boards, view mail, etc.), and be able to select options from there, or go back to their home.

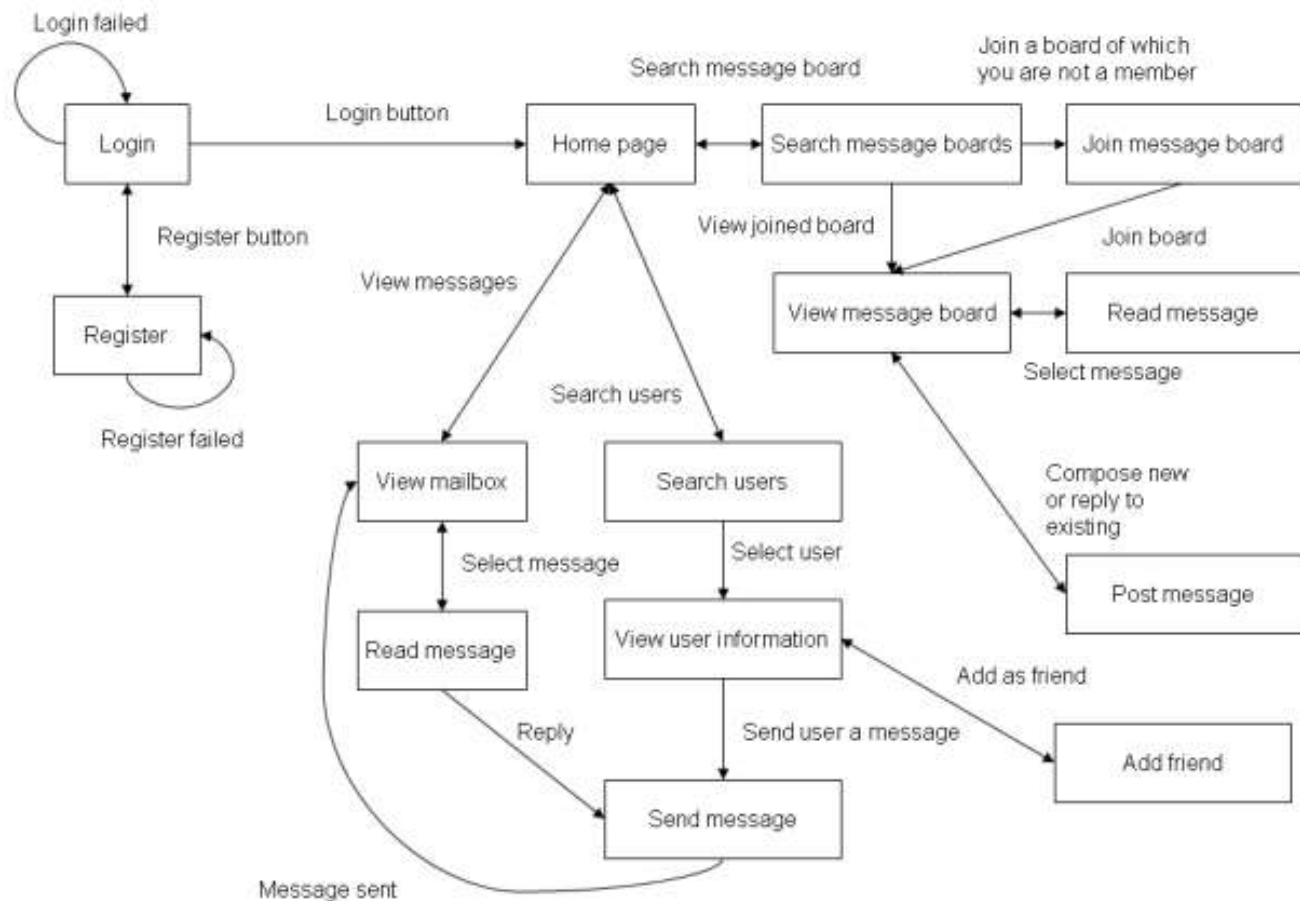


Figure 3.1 Client Screen Flow

Use case diagrams.

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating,

payment processing, and customer relations. A use case diagram contains four components. The boundary, which defines the system of interest in relation to the world around it.

- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

- **Login/Registration**

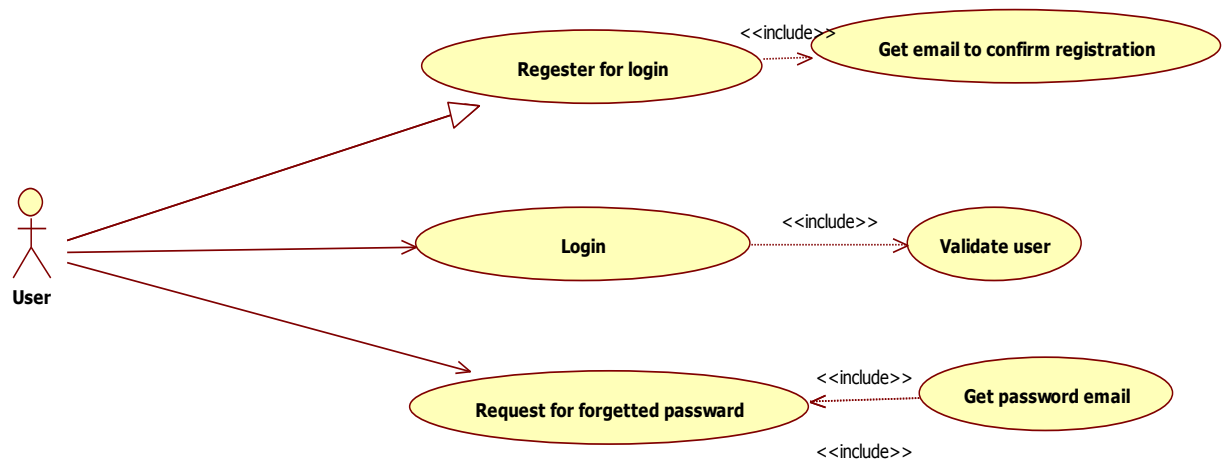


Fig 3.2 Login/Registration

- Home Page

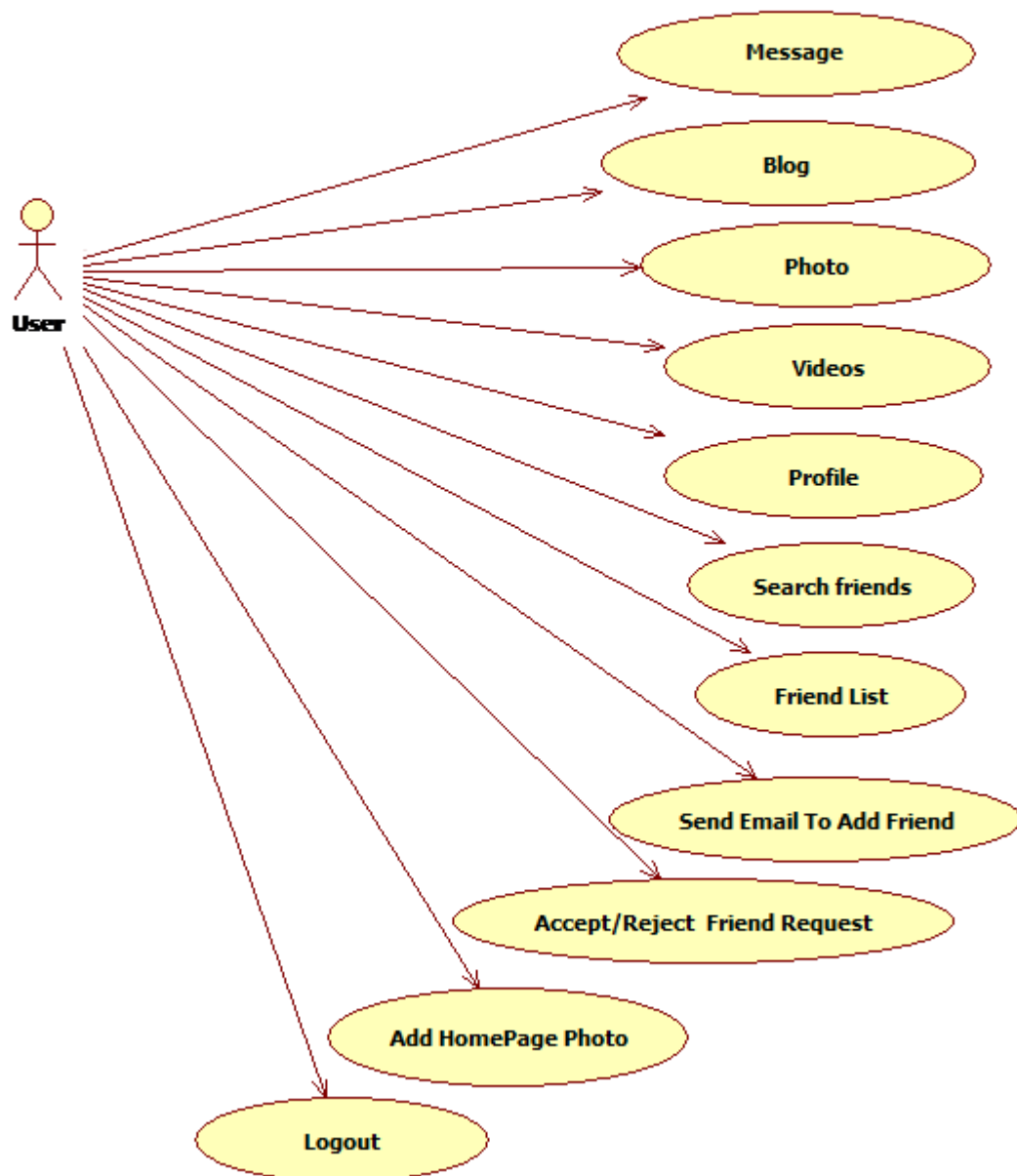


Fig 3.3 homepage

- **Wall Page**

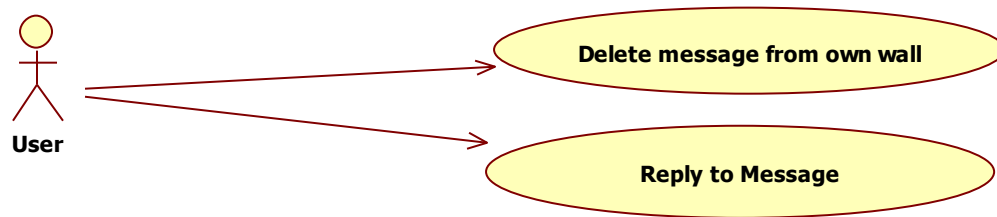


Fig 3.4 Wall page

- **Blog Page**

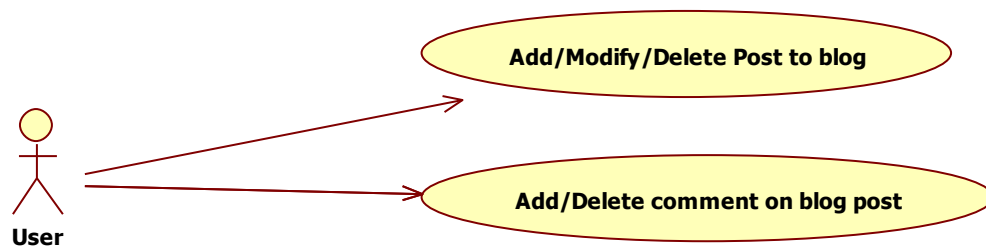


Fig 3.5 Blog page

- **Profile Page**

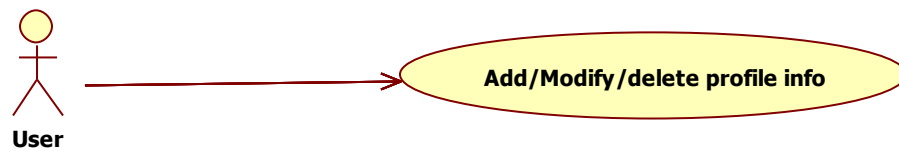


Fig 3.6 Profile page

- **Search Friend Page**

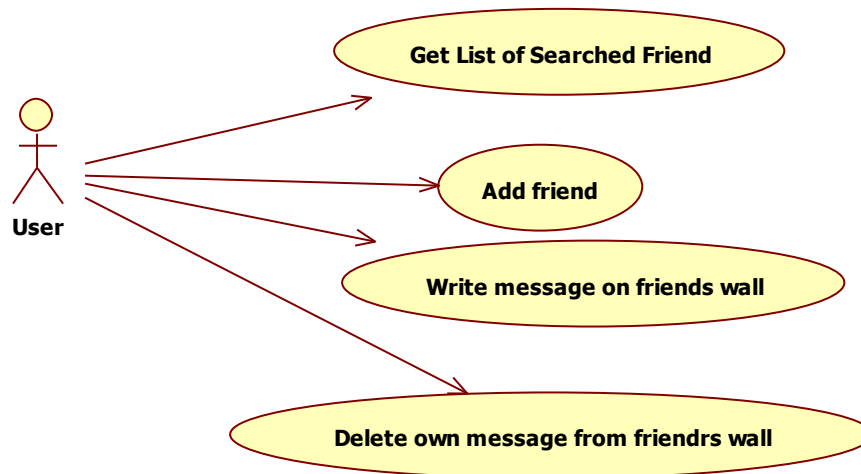


Fig 3.7 Search friend page

- **Friend List Page**

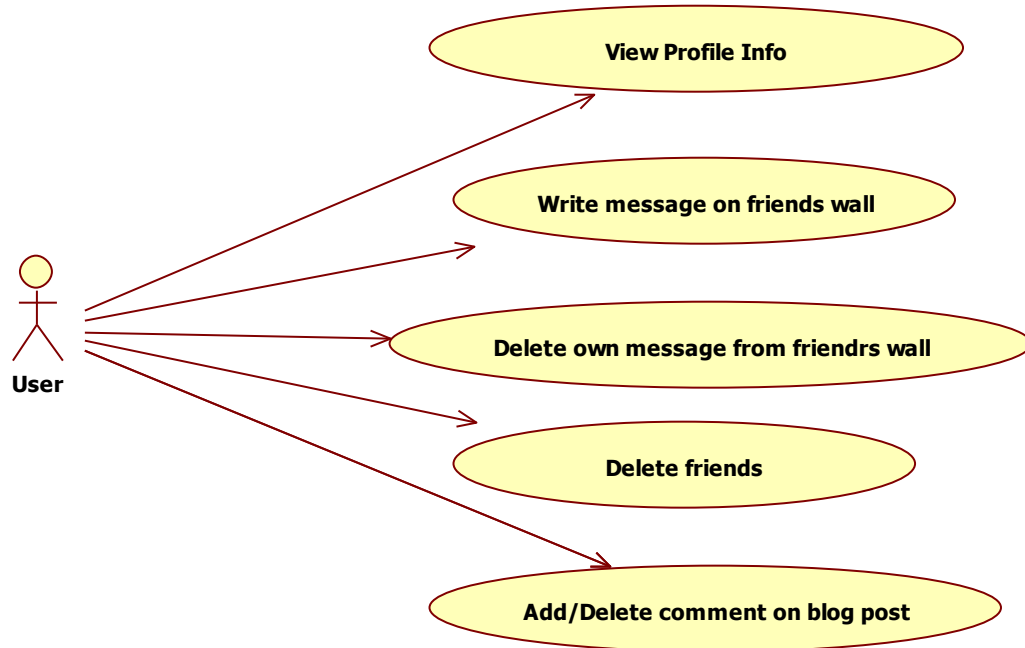


Fig 3.8 Friend List Page

- **Photo Page**

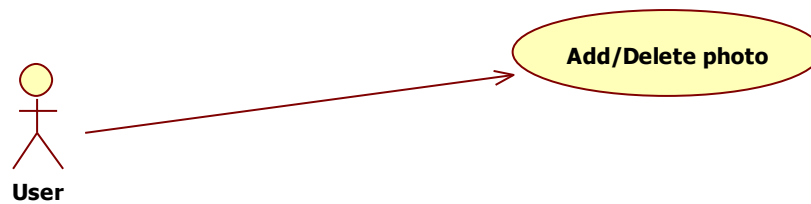


Fig 3.9 Photo page

3.2.2 Context Diagram

The highest level data flow diagram is the context diagram.

- The context diagram shows the interaction of the system with its environment in terms of data flows
- The context diagram defines the boundary of the system (the scope of the system)
- Only the data flows which leave the system and the data flows which come from outside the system are shown.

0 Level DFD: A level 0 DFD, also called a fundamental system model or context diagram represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively.

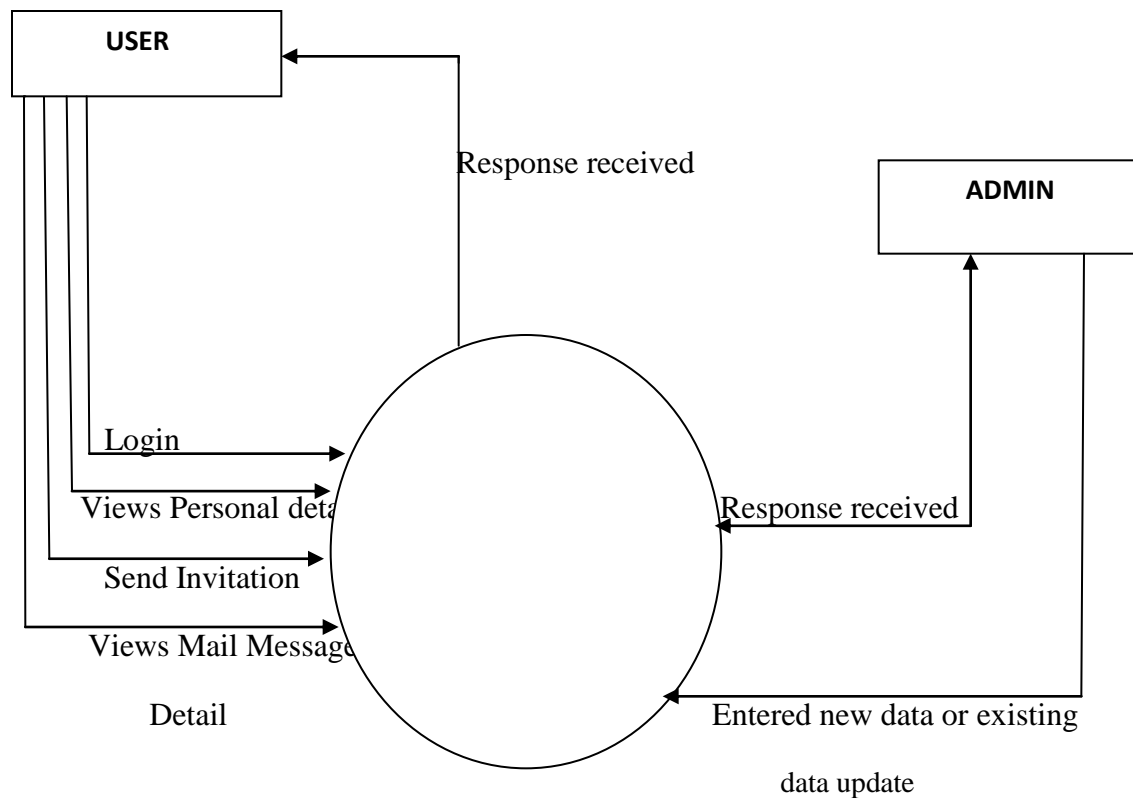


Fig 3.10 0 level DFD

1 Level DFD:

This level of DFD provide more detailed structure. It provides a detailed view of requirements and flow of data from 1 bubble to another.

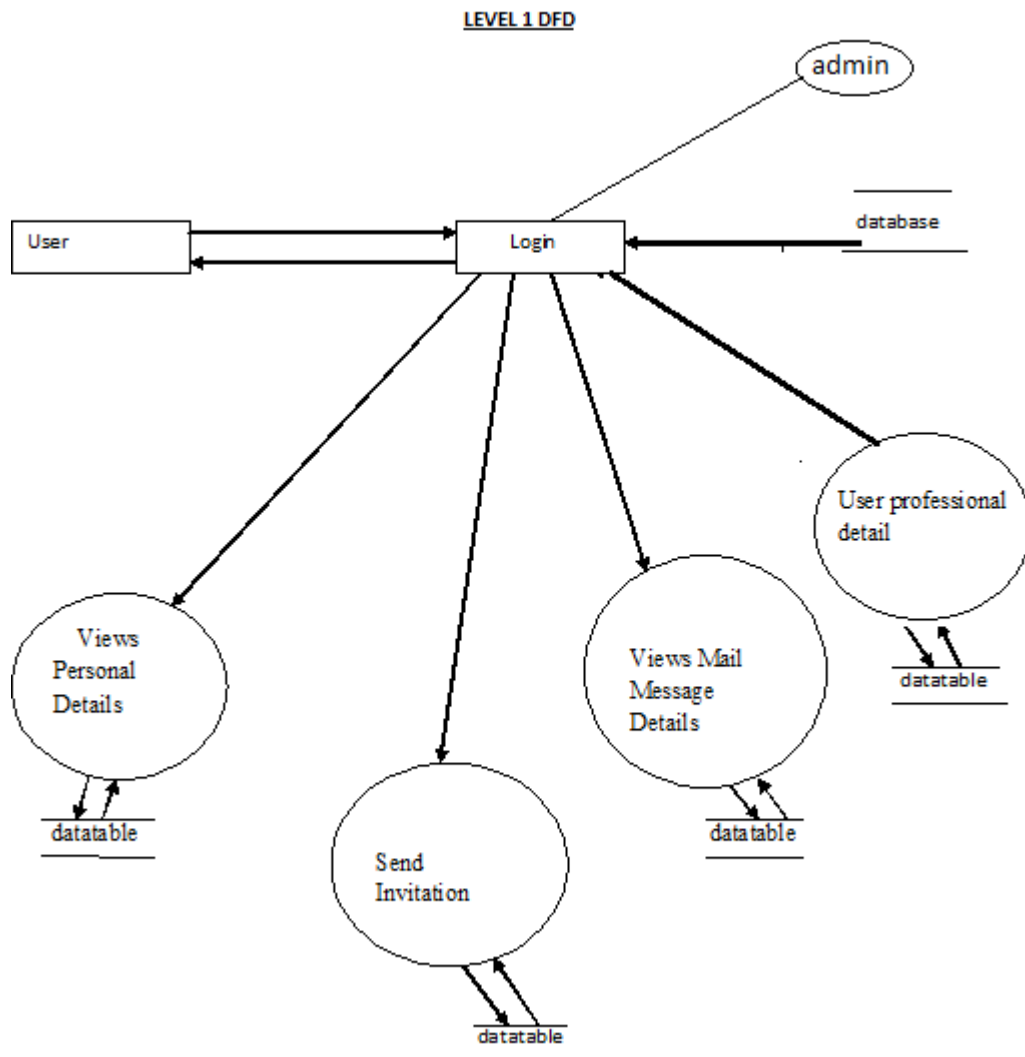


Fig 3.11 1 level DFD

3.2.3 ENTITY RELATIONSHIP DIAGRAM

The entity relationship model is a high level data model. It is based on a perception of a real world that consists of a collection of basic objects, called entities, and of relationship among these objects. It was developed to facilitate database design by allowing specification of an enterprise schema, which represent the overall logical structure of a database.

Entity: An entity is an object that has its existence in the real world. It includes all those “things” about which data is collected. An entity may be a tangible object such as a student, a place or a part. It may also be non-tangible such as an event, a job title or a customer account. For example, if we say that a customer buys goods, it means customer and goods are entities.

Diagrammatically, entities are represented in rectangles.

An Entity Set: It is a set of entities of the same type that share the same properties, or attributes. The set of all persons who are customers at a given bank, example, can be defined as the entity set customer.

Attributes: Attributes are units that describe the characteristics or properties of entities. In a database, entities are represented by tables and attributes by columns. For example, a customer entity might have numerous attributes such as code, name and addresses. Similarly, the goods entity may have attributes like code and price. They are drawn in elliptical shapes along with the entity rectangles.

The entity relationship diagram of mailing system is drawn on the next page:

ENTITY RELATIONSHIP DIAGRAM

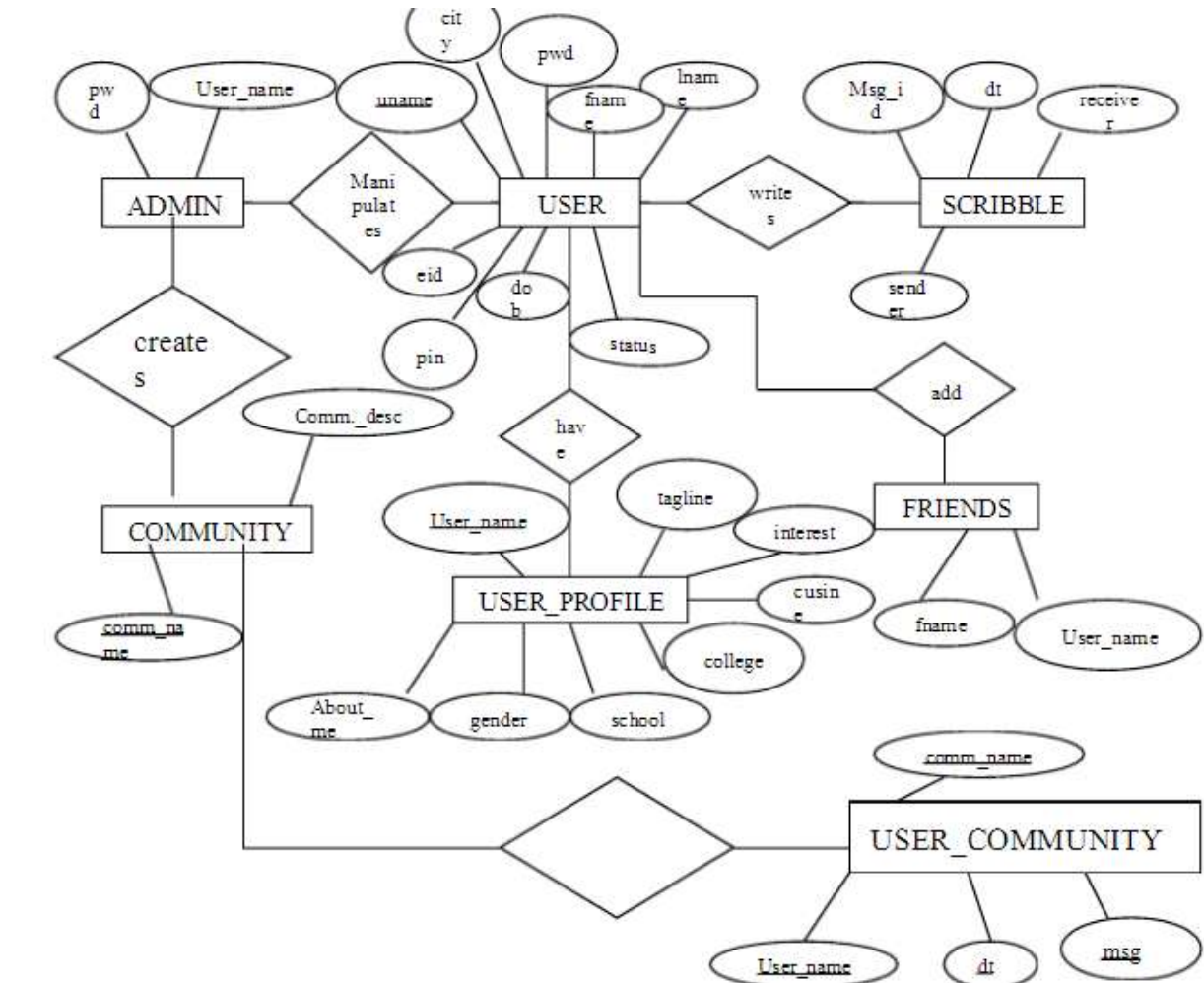


Fig 3.12 ERD

3.2.4 Database

Data base is used to store the relevant information of the individuals. A database is a collection of rows and columns in which rows indicates the tuple and column indicates the domain of table. Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters. Need to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of the logical design of the relation of the base data structures used to store the data. In the relational model these are the classes and named relationships. However, the term database design could also be used to apply to overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

Table	Action	Rows	Type	Collation	Size
comment	Browse Structure Search Insert Empty Drop	12	MyISAM	latin1_swedish_ci	2.4 KiB
day	Browse Structure Search Insert Empty Drop	31	MyISAM	latin1_swedish_ci	2.3 KiB
friends	Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	1 KiB
likes	Browse Structure Search Insert Empty Drop	0	MyISAM	latin1_swedish_ci	1 KiB
members	Browse Structure Search Insert Empty Drop	13	MyISAM	latin1_swedish_ci	4.4 KiB
messages	Browse Structure Search Insert Empty Drop	0	MyISAM	latin1_swedish_ci	1 KiB
month	Browse Structure Search Insert Empty Drop	12	MyISAM	latin1_swedish_ci	2.2 KiB
photos	Browse Structure Search Insert Empty Drop	16	MyISAM	latin1_swedish_ci	2.8 KiB
postcomment	Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	2 KiB
year	Browse Structure Search Insert Empty Drop	43	MyISAM	latin1_swedish_ci	2.4 KiB
10 tables	Sum	129	InnoDB	latin1_swedish_ci	21.5 KiB

↑ ☐ Check All With selected: ▼

Print view Data Dictionary

Fig 3.13 List of tables

3.2.5 Data Dictionary

Table 3.1 Comment

Column	Type	Null	Default	Comments
comment_id	int(11)	No		
comment	text	No		
date_created	varchar(50)	No		
member_id	varchar(30)	No		

Table 3.2 Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	comment_id	12	A	No	

Table 3.3 Day

Column	Type	Null	Default	Comments
day_id	int(11)	No		
day	int(2)	No		

Table 3.4 Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	day_id	31	A	No	

Table 3.5 Friends

Column	Type	Null	Default	Comments
member_id	int(11)	No		
datetime	datetime	No		
status	varchar(11)	No		
friends_with	int(11)	No		

Table 3.6 Likes

Column	Type	Null	Default	Comments
like_id	int(11)	No		
remarks	text	No		
remarksby	varchar(30)	No		

Table 3.7 Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	like_id	0	A	No	

Table 3.8 Members

Column	Type	Null	Default	Comments
member_id	int(11)	No		
UserName	varchar(10)	No		
Password	varchar(80)	No		
FirstName	varchar(30)	No		
LastName	varchar(30)	No		

Address	varchar(200)	No		
ContactNo	varchar(14)	No		
Url	varchar(100)	No		
Birthdate	varchar(20)	No		
Gender	varchar(6)	No		
DateAdded	varchar(45)	No		
profImage	varchar(200)	No		
curcity	varchar(50)	No		
hometown	varchar(50)	Yes	<i>NULL</i>	
Interested	varchar(30)	Yes	<i>NULL</i>	
language	varchar(30)	Yes	<i>NULL</i>	
college	varchar(100)	Yes	<i>NULL</i>	
highschool	varchar(200)	Yes	<i>NULL</i>	
experiences	varchar(200)	Yes	<i>NULL</i>	
arts	text	Yes	<i>NULL</i>	
aboutme	text	Yes	<i>NULL</i>	
month	varchar(20)	Yes	<i>NULL</i>	
day	varchar(2)	Yes	<i>NULL</i>	
year	varchar(4)	Yes	<i>NULL</i>	
Stats	varchar(30)	Yes	<i>NULL</i>	

Table 3.9 Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	member_id	13	A	No	

Table 3.10 Messages

Column	Type	Null	Default	Comments
message_id	int(11)	No		
receiver	varchar(40)	No		
recipient	varchar(40)	No		
datetime	datetime	No		
content	varchar(100)	No		
status	varchar(6)	No		

Table 3.11 Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	message_id	0	A	No	

Table 3.12 Month

Column	Type	Null	Default	Comments
month_id	int(11)	No		

month	varchar(15)	No		
-------	-------------	----	--	--

Table 3.13 Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	month_id	12	A	No	

Table 3.14 Photos

Column	Type	Null	Default	Comments
photo_id	int(11)	No		
location	varchar(200)	No		
member_id	int(11)	No		

Table 3.15 Post comment

Column	Type	Null	Default	Comments
comment_id	int(11)	No		
content	text	No		
commentedby	varchar(30)	No		
pic	varchar(100)	No		
Id	int(40)	No		
date_created	varchar(50)	No		

4 IMPLEMENTATION DETAILS

4.1 SAMPLE SOURCE CODE

4.1.1 Homepage

```

if ( ! defined( 'ABSPATH' ) )
    exit;

/**
 * WP_Job_Manager class.
 */
class WP_Job_Manager {

    /**
     * Constructor - get the plugin hooked in and ready
     */
    public function __construct() {
        // Define constants
        define( 'JOB_MANAGER_VERSION', '1.7.3' );
        define( 'JOB_MANAGER_PLUGIN_DIR', untrailingslashit(
plugin_dir_path( __FILE__ ) ) );
        define( 'JOB_MANAGER_PLUGIN_URL', untrailingslashit( plugins_url(
basename( plugin_dir_path( __FILE__ ) ), basename( __FILE__ ) ) ) );

        // Includes
        include( 'wp-job-manager-functions.php' );
        include( 'wp-job-manager-template.php' );
        include( 'includes/class-wp-job-manager-post-types.php' );
        include( 'includes/class-wp-job-manager-ajax.php' );
        include( 'includes/class-wp-job-manager-shortcodes.php' );
        include( 'includes/class-wp-job-manager-api.php' );
        include( 'includes/class-wp-job-manager-forms.php' );
        include( 'includes/class-wp-job-manager-geocode.php' );

        if ( is_admin() )
            include( 'includes/admin/class-wp-job-manager-admin.php' );

        // Init classes
        $this->forms = new WP_Job_Manager_Forms();
        $this->post_types = new WP_Job_Manager_Post_Types();

        // Activation - works with symlinks
        register_activation_hook( basename( dirname( __FILE__ ) ) . '/' .
basename( __FILE__ ), array( $this->post_types, 'register_post_types' ), 10 );

```

```

        register_activation_hook( basename( dirname( __FILE__ ) ) . '/' .
basename( __FILE__ ), create_function( "", "include_once( 'includes/class-wp-job-
manager-install.php');" ), 10 );
        register_activation_hook( basename( dirname( __FILE__ ) ) . '/' .
basename( __FILE__ ), 'flush_rewrite_rules', 15 );

// Actions
add_action( 'plugins_loaded', array( $this, 'load_plugin_textdomain' ) );
add_action( 'switch_theme', array( $this->post_types, 'register_post_types'
), 10 );

add_action( 'switch_theme', 'flush_rewrite_rules', 15 );
add_action( 'widgets_init', create_function( "", "include_once(
'includes/class-wp-job-manager-widgets.php');" ) );
add_action( 'wp_enqueue_scripts', array( $this, 'frontend_scripts' ) );
add_action( 'admin_init', array( $this, 'updater' ) );
    }

/**
 * Handle Updates
 */
    public function updater() {
        if ( version_compare( JOB_MANAGER_VERSION, get_option(
'wp_job_manager_version' ), '>' ) )
            include_once( 'includes/class-wp-job-manager-install.php' );
    }

/**
 * Localisation
 */
    public function load_plugin_textdomain() {
        load_plugin_textdomain( 'wp-job-manager', false, dirname(
plugin_basename( __FILE__ ) ) . '/languages/' );
    }

/**
 * Register and enqueue scripts and css
 */
    public function frontend_scripts() {
        wp_register_script( 'wp-job-manager-ajax-filters',
JOB_MANAGER_PLUGIN_URL . '/assets/js/ajax-filters.min.js', array( 'jquery' ),
JOB_MANAGER_VERSION, true );
        wp_register_script( 'wp-job-manager-job-dashboard',
JOB_MANAGER_PLUGIN_URL . '/assets/js/job-dashboard.min.js', array( 'jquery' ),
JOB_MANAGER_VERSION, true );

```

```

        wp_register_script( 'wp-job-manager-job-application',
JOB_MANAGER_PLUGIN_URL . '/assets/js/job-application.min.js', array( 'jquery' ),
JOB_MANAGER_VERSION, true );
        wp_register_script( 'wp-job-manager-job-submission',
JOB_MANAGER_PLUGIN_URL . '/assets/js/job-submission.min.js', array( 'jquery' ),
JOB_MANAGER_VERSION, true );

        wp_localize_script( 'wp-job-manager-ajax-filters',
'job_manager_ajax_filters', array(
            'ajax_url' => admin_url('admin-ajax.php')
        ) );
        wp_localize_script( 'wp-job-manager-job-dashboard',
'job_manager_job_dashboard', array(
            'i18n_confirm_delete' => __( 'Are you sure you want to delete this
job?', 'wp-job-manager' ) ) );

```

4.1.2 Admin

```
<?php //encoding: utf-8
```

```

// Admin Settings
require_once( JOBMAN_DIR . '/admin-settings.php' );
// Frontend Display Settings
require_once( JOBMAN_DIR . '/admin-frontend-settings.php' );
// Job Form Setup
require_once( JOBMAN_DIR . '/admin-jobs-settings.php' );
// Job management
require_once( JOBMAN_DIR . '/admin-jobs.php' );
// Application form setup
require_once( JOBMAN_DIR . '/admin-application-form.php' );
// Applications
require_once( JOBMAN_DIR . '/admin-applications.php' );
// Emails
require_once( JOBMAN_DIR . '/admin-emails.php' );
// Interview Scheduling
require_once( JOBMAN_DIR . '/admin-interviews.php' );
// Comment handling functions
require_once( JOBMAN_DIR . '/admin-comments.php' );

function jobman_admin_setup() {
    $options = get_option( 'jobman_options' );

    // Setup the admin menu item
    $pages = array();

```

```

        add_menu_page( __( 'Job Manager', 'jobman' ), __( 'Job Manager', 'jobman' ),
        'publish_posts', 'jobman-conf', 'jobman_conf' );
        $pages[] = add_submenu_page( 'jobman-conf', __( 'Job Manager', 'jobman' ), __(
        'Settings', 'jobman' ), 'manage_options', 'jobman-conf', 'jobman_conf' );
        $pages[] = add_submenu_page( 'jobman-conf', __( 'Job Manager', 'jobman' ), __(
        'Add Job', 'jobman' ), 'publish_posts', 'jobman-add-job', 'jobman_add_job' );
        $pages[] = add_submenu_page( 'jobman-conf', __( 'Job Manager', 'jobman' ), __(
        'Jobs', 'jobman' ), 'publish_posts', 'jobman-list-jobs', 'jobman_list_jobs' );
        $pages[] = add_submenu_page( 'jobman-conf', __( 'Job Manager', 'jobman' ), __(
        'Applications', 'jobman' ), 'read_private_pages', 'jobman-list-applications',
        'jobman_list_applications' );
        $pages[] = add_submenu_page( 'jobman-conf', __( 'Job Manager', 'jobman' ), __(
        'Emails', 'jobman' ), 'read_private_pages', 'jobman-list-emails', 'jobman_list_emails' );

        if( $options['interviews'] )
            $pages[] = add_submenu_page( 'jobman-conf', __( 'Job Manager', 'jobman'
        ), __( 'Interviews', 'jobman' ), 'read_private_pages', 'jobman-interviews',
        'jobman_interviews' );

        // Load our header info
        foreach( $pages as $page ) {
            add_action( "admin_print_styles-$page", 'jobman_admin_print_styles' );
            add_action( "admin_print_scripts-$page", 'jobman_admin_print_scripts' );
            add_action( "admin_head-$page", 'jobman_admin_header' );
        }
    }

    function jobman_plugin_row_meta( $links, $file ) {
        if( JOBMAN_BASENAME == $file && ! get_option( 'pento_consulting' ) ) {
            $links[] = '<a
href="http://www.amazon.com/wishlist/1ORKI9ZG875BL">' . __( 'My Amazon Wish
List', 'jobman' ) . '</a>';
            $links[] = '<a href="https://www.paypal.com/cgi-
bin/webscr?cmd=_donations&business=gary%40pento%2enet&item_name=WordPress
%20Plugin%20(Job%20Manager)&item_number=Support%20Open%20Source&no_shi
pping=0&no_note=1&tax=0&currency_code=USD&lc=US&bn=PP%2dDonationsBF&c
harset=UTF%2d8">' . __( 'Donate with PayPal', 'jobman' ) . '</a>';
        }

        return $links;
    }

    function jobman_admin_print_styles() {
        global $wp_styles;

```

```

        wp_enqueue_style( 'jobman-admin', JOBMAN_URL . '/css/admin.css', false,
JOBMAN_VERSION, 'all' );
        wp_enqueue_style( 'jobman-admin-ie7', JOBMAN_URL . '/css/admin-ie7.css',
false, JOBMAN_VERSION, 'all' );
        wp_enqueue_style( 'jobman-admin-print', JOBMAN_URL . '/css/admin-print.css',
false, JOBMAN_VERSION, 'print' );
        wp_enqueue_style( 'jobman-admin-print-ie7', JOBMAN_URL . '/css/admin-print-
ie7.css', false, JOBMAN_VERSION, 'print' );
        wp_enqueue_style( 'dashboard' );

        $wp_styles->add_data( 'jobman-admin-print-ie7', 'conditional', 'lte IE 7' );
        $wp_styles->add_data( 'jobman-admin-ie7', 'conditional', 'lte IE 7' );
    }

function jobman_admin_print_scripts() {
    wp_enqueue_script( 'jobman-admin', JOBMAN_URL . '/js/admin.js', false,
JOBMAN_VERSION );
    wp_deregister_script( 'jquery-ui-datepicker' );
    wp_enqueue_script( 'jquery-ui' );
    wp_enqueue_script( 'jquery-ui-datepicker', JOBMAN_URL . '/js/jquery-ui-
datepicker.js', false, JOBMAN_VERSION );
    wp_enqueue_script( 'dashboard' );
}

function jobman_admin_header() {
    global $wp_version;
    $options = get_option( 'jobman_options' );

    $textareas = array();
    foreach( $options['job_fields'] as $id => $field ) {
        if( 'textarea' == $field['type'] ) {
            $textareas[] = $id;
        }
    }
}

?>
<script type="text/javascript">
//
addLoadEvent(function() {
    jQuery(".datepicker").datepicker({

dateFormat: 'yy-mm-dd',
changeMonth: true,
changeYear: true,
gotoCurrent: true,
showOn: 'button',
buttonImage: '&lt;?php echo

JOBMAN_URL ?&gt;/images/calendar.gif',
</pre>
<hr/>
</div>
<div data-bbox="171 919 517 936" data-label="Page-Footer">Department of Computer Science, Christ University</div>
```

```

buttonImageOnly: true
    });
    jQuery(".column-cb > *").click(function() { jQuery(".check-column >
*").attr('checked', jQuery(this).is(':checked')) } );
    jQuery("div.star-holder img").click(function() {
        var cssclass = jQuery(this).parent().attr("class");
        var count = cssclass.replace("star star", "");
        jQuery(this).parent().parent().find('input[name="jobman-
rating"]').attr("value", count);
        jQuery(this).parent().parent().find("div.star-rating").css("width", (count *
19) + "px");

        var data = jQuery(this).parent().parent().find('input[name="callbackid"]');
        var func = jQuery(this).parent().parent().find('input[name="callbackfunction"]');
        var callback;
        if( data.length > 0 ) {
            callback = {
                action: func[0].value,
                appid: data[0].value,
                rating: count
            };

            jQuery.post( ajaxurl, callback );
        }
    });

    jQuery("div.star-holder img").mouseenter(function() {
        var cssclass = jQuery(this).parent().attr("class");
        var count = cssclass.replace("star star", "");
        jQuery(this).parent().parent().find("div.star-rating").css("width", (count *
19) + "px");
    });

    jQuery("div.star-holder img").mouseleave(function() {
        var count = jQuery(this).parent().parent().find('input[name="jobman-
rating"]').attr("value");
        jQuery(this).parent().parent().find("div.star-rating").css("width", (count *
19) + "px");
    });
}
<?php
}
?>
});

function jobman_reset_rating( id, func ) {

```

```

jQuery( "#jobman-rating-" + id ).attr("value", 0);
jQuery( "#jobman-star-rating-" + id ).css("width", "0px");

if( "filter" != id ) {
    callback = {
        action: func,
        appid: id,
        rating: 0
    };

    jQuery.post( ajaxurl, callback );
}
}
//]]>
</script>
<?php
}

function jobman_print_settings_tabs() {
    $tabs = array(
        'admin' => __( 'Admin Settings', 'jobman' ),
        'display' => __( 'Display Settings', 'jobman' ),
        'appform' => __( 'App. Form Settings', 'jobman' ),
        'jobform' => __( 'Job Form Settings', 'jobman' ),
    );

    if( ! array_key_exists( 'tab', $_REQUEST ) )
        $_REQUEST['tab'] = 'admin';
?>

    <h2 class="jobman-tabs">

<?php
    foreach( $tabs as $key => $title ) {
        $class = "";
        if( $key != $_REQUEST['tab'] )
            $class = ' menu-tab-inactive';
        echo "<a href='" . admin_url( "admin.php?page=jobman-
conf&tab=$key" ) . "' class='menu-tabs$class'>$title</a>";
    }
?>

    </h2>

<?php
}

function jobman_print_other_projects_box() {
?>

```

```

        <a
href="http://ponyedit.com/?utm_source=jobman&utm_medium=banner&utm_campaign
=Job%2BManager"><strong>PonyEdit</strong></a>
        <p><?php _e( 'If you like Job Manager and WordPress, but hate working
on remote servers, you\'ll want to check out PonyEdit - a fast new text editor that plays
nicely with the cloud!', 'jobman' ) ?></p>
<?php
}

function jobman_print_donate_box() {
?>
        <p><?php _e( "If this plugin helps you find that perfect new employee, I'd
appreciate it if you shared the love, by way of my Donate or Amazon Wish List links
below.", 'jobman' ) ?></p>
        <ul>
                <li><a
href="http://www.amazon.com/wishlist/1ORKI9ZG875BL"><?php _e( 'My Amazon
Wish List', 'jobman' ) ?></a></li>
                <li><a href="https://www.paypal.com/cgi-
bin/webscr?cmd=_donations&business=gary%40pento%2enet&item_name=WordPress
%20Plugin%20(Job%20Manager)&item_number=Support%20Open%20Source&no_shi
pping=0&no_note=1&tax=0&currency_code=USD&lc=US&bn=PP%2dDonationsBF&c
harset=UTF%2d8"><?php _e( 'Donate with PayPal', 'jobman' ) ?></a></li>
        </ul>
<?php
}

function jobman_print_about_box() {
?>
        <ul>
                <li><a href="http://pento.net/"><?php _e( "Gary Pendergast's
Blog", 'jobman' ) ?></a></li>
                <li><a href="http://twitter.com/garypendergast"><?php _e(
'Follow me on Twitter!', 'jobman' ) ?></a></li>
                <li><a href="http://pento.net/projects/wordpress-job-manager-
plugin/"><?php _e( 'Plugin Homepage', 'jobman' ) ?></a></li>
                <li><a href="http://code.google.com/p/wordpress-job-
manager/issues/list"><?php _e( 'Submit a Bug/Feature Request', 'jobman' ) ?></a></li>
        </ul>
<?php
}

function jobman_print_translators_box() {
?>
        <p><?php _e( "If you're using Job Manager in a language other than
English, you have some of my wonderful translators to thank for it!", 'jobman' ) ?></p>

```

```

        <p><?php printf( __( "If you're fluent in a language not listed here, and
would like to appear on this list, please <a href='%1s'>contact me</a>!", 'jobman' ),
'http://pento.net/contact/' ) ?>
        <ul>
            <li><strong><?php _e( 'Arabic', 'jobman' ) ?></strong> - <a
href="http://www.ghalebi.info/">Ali Al-Ghalebi</a></li>
            <li><strong><?php _e( 'Czech', 'jobman' ) ?></strong> - <a
href="mailto:prdlik@centrum.cz">Lukas</a></li>
            <li><strong><?php _e( 'Danish', 'jobman' ) ?></strong> - <a
href="http://www.lithin.com/">Christian Olesen</a>, <a
href="http://www.lithin.com/">Caspar Lange</a></li>
            <li><strong><?php _e( 'Dutch', 'jobman' ) ?></strong> - <a
href="http://www.centrologic.nl/">Patrick Tessels</a>, <a
href="http://webtaurus.nl/">Henk van den Bor</a></li>
            <li><strong><?php _e( 'Estonian', 'jobman' ) ?></strong> - Robert
Jakobson</li>
            <li><strong><?php _e( 'French', 'jobman' ) ?></strong> - <a
href="http://www.procure-smart.com/">Fabrice Fotso</a>, Vincent Clady</li>
            <li><strong><?php _e( 'German', 'jobman' ) ?></strong> - <a
href="http://www.tolingo.com/">tolingo translations</a>, <a href="http://www.la-palma-
diving.com/">Joachim Richter</a></li>
            <li><strong><?php _e( 'Portuguese (Brazil)', 'jobman' )
?></strong> - <a href="http://www.alex fj.com.br/">Alex Ferreira</a></li>
            <li><strong><?php _e( 'Spanish', 'jobman' ) ?></strong> - <a
href="http://www.tradiart.com">TradiArt</a></li>
            <li><strong><?php _e( 'Swedish', 'jobman' ) ?></strong> - <a
href="http://www.saxekon.se/">Berndt Axelsson</a></li>
        </ul>
    <?php
    }
    ?>

```

4.1.3 Registration

```

define( 'RPR_VERSION', '3.9.9' );
define( 'RPR_ACTIVATION_REQUIRED', '3.9.6' );

if ( !class_exists( 'Register_Plus_Redux' ) ) {
    class Register_Plus_Redux {
        private /*.array[string]mixed.*/ $options;

        public /*.void.*/ function __construct() {
            register_activation_hook( __FILE__, array( $this, 'rpr_activation' )
        );
    }
}

```

```

        register_deactivation_hook( __FILE__, array(
'Register_Plus_Redux', 'rpr_uninstall' ) );
        register_uninstall_hook( __FILE__, array( 'Register_Plus_Redux',
'rpr_uninstall' ) );

        add_action( 'init', array( $this, 'rpr_i18n_init' ), 10, 1 );

        if ( !is_multisite() ) {
            add_filter( 'pre_user_login', array( $this,
'rpr_filter_pre_user_login_swp' ), 10, 1 ); // Changes user_login to user_email
        }

        add_action( 'admin_enqueue_scripts', array( $this,
'rpr_admin_enqueue_scripts' ), 10, 1 );

        add_action( 'show_user_profile', array( $this,
'rpr_show_custom_fields' ), 10, 1 ); // Runs near the end of the user profile editing screen.
        add_action( 'edit_user_profile', array( $this,
'rpr_show_custom_fields' ), 10, 1 ); // Runs near the end of the user profile editing screen
in the admin menus.
        add_action( 'profile_update', array( $this, 'rpr_save_custom_fields'
), 10, 1 ); // Runs when a user's profile is updated. Action function argument: user ID.

        add_action( 'admin_footer-profile.php', array( $this,
'rpr_admin_footer' ), 10, 0 ); // Runs in the HTML <head> section of the admin panel of a
page or a plugin-generated page.
        add_action( 'admin_footer-user-edit.php', array( $this,
'rpr_admin_footer' ), 10, 0 ); // Runs in the HTML <head> section of the admin panel of a
page or a plugin-generated page.
    }

    public /*.void.*/ function rpr_activation() {
        global $wp_roles;
        add_role( 'rpr_unverified', 'Unverified' );
        update_option( 'register_plus_redux_last_activated',
RPR_ACTIVATION_REQUIRED );
    }

    public static /*.void.*/ function rpr_uninstall() {
        global $wp_roles;
        remove_role( 'rpr_unverified' );
        delete_option( 'register_plus_redux_last_activated' );
    }

    public static /*.mixed.*/ function default_options( $option = '' )
    {

```

```

$blogname = stripslashes( wp_specialchars_decode( get_option(
'blogname' ), ENT_QUOTES ) );
$options = array(
    'verify_user_email' => is_multisite() ? '1' : '0',
    'message_verify_user_email' => is_multisite() ?
        __( "<h2>%user_login% is your new
username</h2>\n<p>But, before you can start using your new username, <strong>you
must activate it</strong></p>\n<p>Check your inbox at
<strong>%user_email%</strong> and click the link given.</p>\n<p>If you do not
activate your username within two days, you will have to sign up again.</p>", 'register-
plus-redux' ) :
        __( 'Please verify your account using the
verification link sent to your email address.', 'register-plus-redux' ),
    'verify_user_admin' => '0',
    'message_verify_user_admin' => __( 'Your account will be
reviewed by an administrator and you will be notified when it is activated.', 'register-plus-
redux' ),
    'delete_unverified_users_after' => is_multisite() ? 0 : 7,
    'autologin_user' => '0',

    'username_is_email' => '0',
    'double_check_email' => '0',
    'user_set_password' => '0',
    'min_password_length' => 6,
    'disable_password_confirmation' => '0',
    'show_password_meter' => '0',
    'message_empty_password' => 'Strength Indicator',
    'message_short_password' => 'Too Short',
    'message_bad_password' => 'Bad Password',
    'message_good_password' => 'Good Password',
    'message_strong_password' => 'Strong Password',
    'message_mismatch_password' => 'Password Mismatch',
    'enable_invitation_code' => '0',
    'require_invitation_code' => '0',
    'invitation_code_case_sensitive' => '0',
    'invitation_code_unique' => '0',
    'enable_invitation_tracking_widget' => '0',
    'show_disclaimer' => '0',
    'message_disclaimer_title' => 'Disclaimer',
    'require_disclaimer_agree' => '1',
    'message_disclaimer_agree' => 'Accept the Disclaimer',
    'show_license' => '0',
    'message_license_title' => 'License Agreement',
    'require_license_agree' => '1',
    'message_license_agree' => 'Accept the License
Agreement',

```

```

'show_privacy_policy' => '0',
'message_privacy_policy_title' => 'Privacy Policy',
'require_privacy_policy_agree' => '1',
'message_privacy_policy_agree' => 'Accept the Privacy
Policy',

'default_css' => '1',
'required_fields_style' => 'border:solid 1px #E6DB55;
background-color:#FFFFE0;',
'required_fields_asterisk' => '0',
'starting_tabindex' => 0,

/*
'datepicker_firstdayofweek' => 6,
'datepicker_dateformat' => 'mm/dd/yyyy',
'datepicker_startdate' => '',
'datepicker_calyear' => '',
'datepicker_calmonth' => 'cur',
*/

'disable_user_message_registered' => '0',
'disable_user_message_created' => '0',
'custom_user_message' => '0',
'user_message_from_email' => get_option( 'admin_email' ),
'user_message_from_name' => $blogname,
'user_message_subject' => '[' . $blogname . ']' . __( 'Your
Login Information', 'register-plus-redux' ),
'user_message_body' => "Username:
%user_login%\nPassword: %user_password%\n\n%site_url%\n",
'send_user_message_in_html' => '0',
'user_message_newline_as_br' => '0',
'custom_verification_message' => '0',
'verification_message_from_email' => get_option(
'admin_email' ),
'verification_message_from_name' => $blogname,
'verification_message_subject' => '[' . $blogname . ']' . __(
'Verify Your Account', 'register-plus-redux' ),
'verification_message_body' => "Verification URL:
%verification_url%\nPlease use the above link to verify your email address and activate
your account\n",
'send_verification_message_in_html' => '0',
'verification_message_newline_as_br' => '0',

'disable_admin_message_registered' => '0',
'disable_admin_message_created' => '0',
'admin_message_when_verified' => '0',
'custom_admin_message' => '0',

```



```

        'admin_message_from_email' => get_option( 'admin_email'
    ),
        'admin_message_from_name' => $blogname,
        'admin_message_subject' => '[' . $blogname . ']' . __( 'New
User Registered', 'register-plus-redux' ),
        'admin_message_body' => "New user registered on your
site %blogname%\n\nUsername: %user_login%\nE-mail: %user_email%\n",
        'send_admin_message_in_html' => '0',
        'admin_message_newline_as_br' => '0'
    );
    if ( !empty( $option ) ) {
        if ( array_key_exists( $option, $options ) ) {
            return $options[$option];
        }
        else {
            //TODO: Trigger event this would be odd
            return FALSE;
        }
    }
    return $options;
}

public /*.bool.*/ function rpr_update_options( /*.array[string]mixed.*/
$options ) {
    if ( empty( $options ) && empty( $this->options ) ) return FALSE;
    if ( !empty( $options ) ) {
        update_option( 'register_plus_redux_options', $options );
        $this->options = $options;
    }
    else {
        update_option( 'register_plus_redux_options', $this-
>options );
    }
    return TRUE;
}

private /*.void.*/ function rpr_load_options( $force_refresh = FALSE ) {
    if ( empty( $this->options ) || $force_refresh === TRUE ) {
        $this->options = get_option( 'register_plus_redux_options'
);
    }
    if ( empty( $this->options ) ) {
        $this->rpr_update_options(
Register_Plus_Redux::default_options() );
    }
}

```

```

        public /*.mixed.*/ function rpr_get_option( /*.string.*/ $option ) {
            if ( empty( $option ) ) return NULL;
            $this->rpr_load_options( FALSE );
            if ( array_key_exists( $option, $this->options ) ) {
                return $this->options[$option];
            }
            return NULL;
        }
    }
}

```

4.1.4 Friends request

```

<?php
global $table_prefix, $wpdb;
$table_prefix = $wpdb->prefix;

/**
 * Constants
 */
define('VERSION', '1.1.2-c');
define('JP_PLUGIN_DIR', ABSPATH . 'wp-content/plugins/wpcareers');
define('JP_PLUGIN_URL', plugins_url('wpcareers'));

/**
 * @brief Autoload files.
 */
require_once(dirname(__FILE__) . '/include/jp_functions.php');
require_once(dirname(__FILE__) . '/include/jp_securimage.php');
require_once(dirname(__FILE__) . '/include/jp_GADlink.php');
require_once(dirname(__FILE__) . '/admin/jp_admin_posts.php');
require_once(dirname(__FILE__) . '/admin/jp_admin.php');
require_once(dirname(__FILE__) . '/admin/jp_admin_setup.php');
require_once(dirname(__FILE__) . '/admin/jp_admin_structure.php');
require_once(dirname(__FILE__) . '/admin/jp_admin_utilities.php');
require_once(dirname(__FILE__) . '/jp_post_job.php');
require_once(dirname(__FILE__) . '/jp_post_resume.php');
require_once(dirname(__FILE__) . '/jp_list_job.php');
require_once(dirname(__FILE__) . '/jp_list_resume.php');
require_once(dirname(__FILE__) . '/jp_list_category.php');
require_once(dirname(__FILE__) . '/jp_list_resume.php');
require_once(dirname(__FILE__) . '/jp_search.php');
require_once(dirname(__FILE__) . '/jp_main.php');

```

```

require_once(dirname(__FILE__) . '/jp_view_job.php');
require_once(dirname(__FILE__) . '/jp_view_resume.php');

/**
 * get_language() - Get HTTP header accept languages
 */
$locale = get_locale();
if(!empty($locale)) {
    $lng = preg_split ('/_/', $locale );
    $languageFile = JP_PLUGIN_DIR . '/language/lang_'. $lng[0] . '.php';
}
if (!empty($languageFile) && file_exists($languageFile)) {
    require_once($languageFile);
} else {
    require_once(JP_PLUGIN_DIR . '/language/lang_en.php');
}

function jp_rm_generator_filter() { return ""; }

/**
 * Initialize the plugin
 */
add_action('plugins_loaded', create_function('$a', 'global $wpcareers; $wpcareers = new WP_careers();'));
add_filter('the_content', 'wpcareers_page_handle_content');
add_filter('the_title', 'wpcareers_page_handle_title');
add_filter('wp_list_pages', 'wpcareers_page_handle_titlechange');
add_filter('single_post_title', 'wpcareers_page_handle_pagetitle');
add_filter('query_vars', 'wpcareers_query_vars');
add_filter('the_generator', 'jp_rm_generator_filter');
/**
 * Assigns each respective variable.
 */
date_default_timezone_set('UTC'); // php5.1
if (!isset($_GET)) $_GET = $_HTTP_GET_VARS;
if (!isset($_POST)) $_POST = $_HTTP_POST_VARS;
if (!isset($_SERVER)) $_SERVER = $_HTTP_SERVER_VARS;
if (!isset($_COOKIE)) $_COOKIE = $_HTTP_COOKIE_VARS;

$wpca_settings = get_option('wpcareers');
if (isset($_REQUEST["wpcareers_action"])){
    $_SERVER["REQUEST_URI"]
=dirname(dirname($_SERVER["PHP_SELF"]))."/". $wpca_settings['slug']."/";
    $_SERVER["REQUEST_URI"] = stripslashes($_SERVER["REQUEST_URI"]);

```

4.2 SCREENSHOTS



fig 4.1 Login page

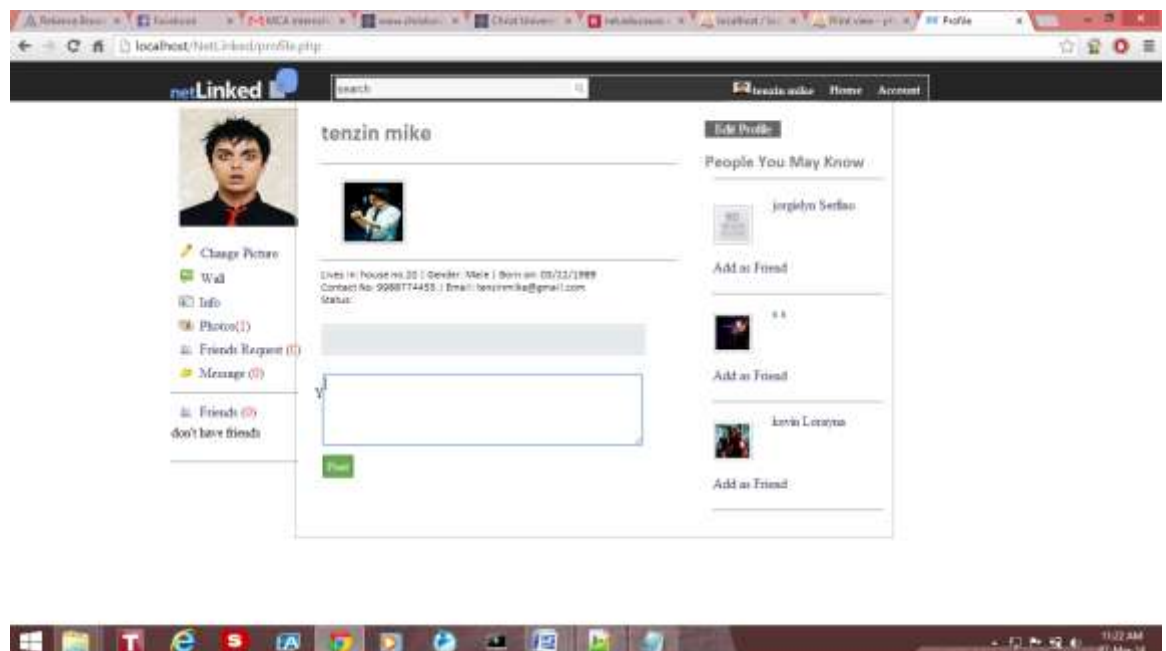


fig4.2 Homepage



fig4.3 Friends page

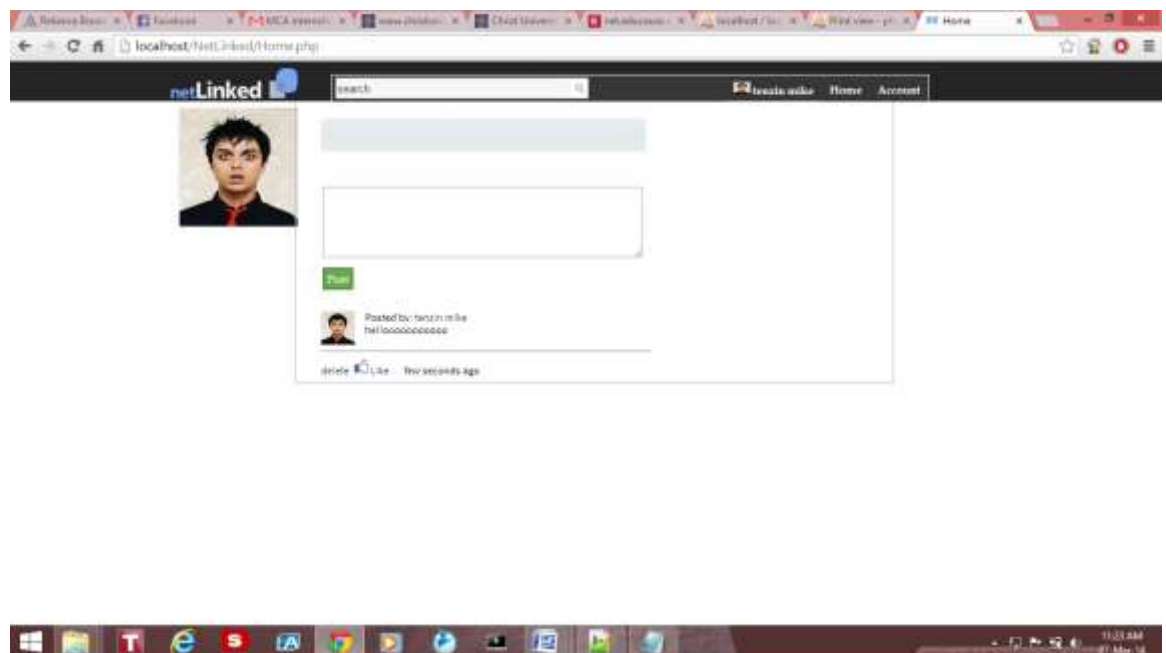


fig4.4 Post page

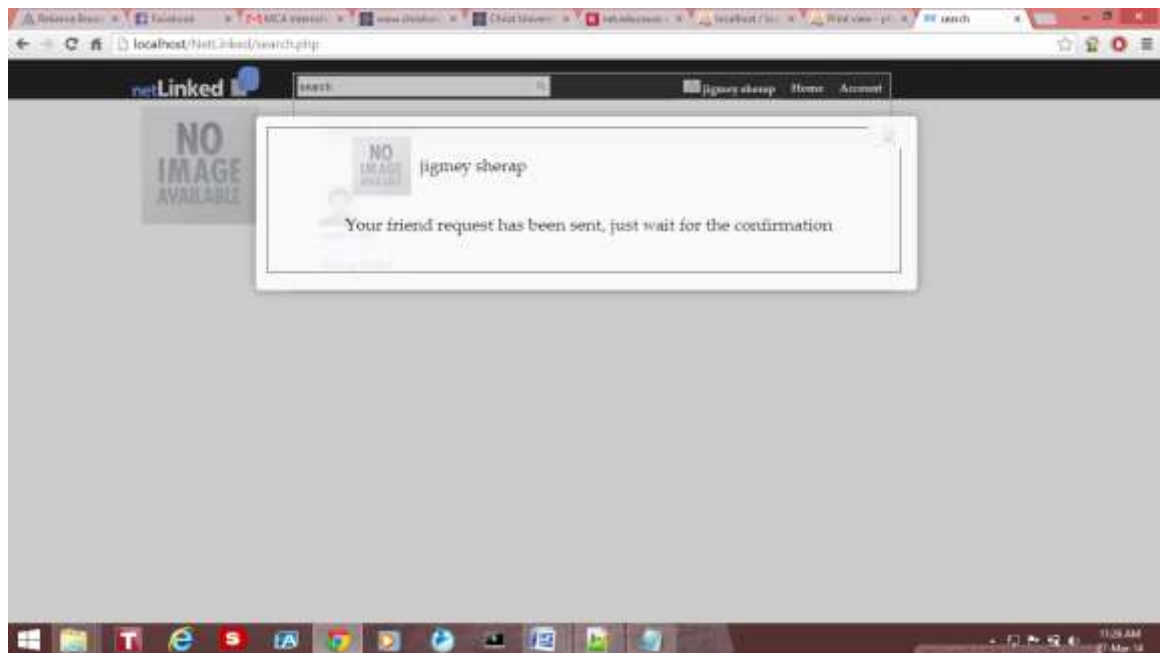


fig4.5 friend's request page

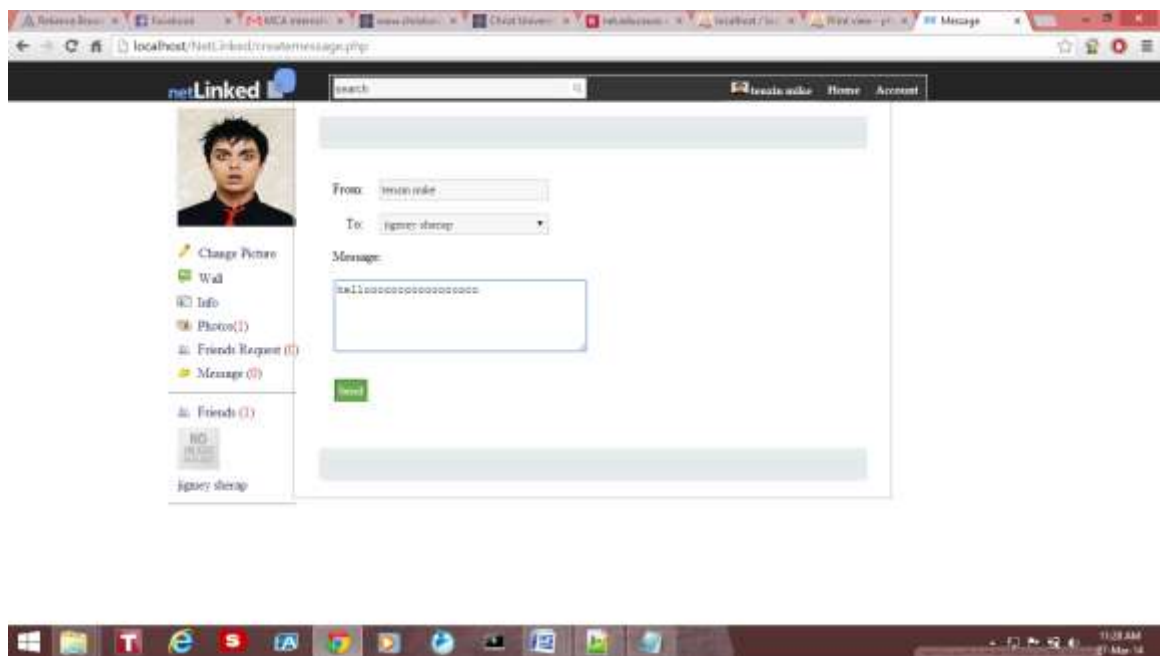


fig4.6 Message page

5. TESTING

- **System Testing**

System testing is a critical element of quality assurance and represents the ultimate review of analysis, design and coding. Test case design focuses on a set of techniques for the creation of test because that meet overall testing objective. When a system is developed it is hoped that it performs properly. The main purpose of testing an information system is to find the errors and correct them. The scope of system testing should include both manual and computerized operations. System testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls.

System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted in accordance to the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner.

The process of analyzing the software item to detect the differences between existing or required condition and evaluate the features of the software items. The thorough testing of the system before release of the software needs to be done vide the various test cases and modes so that the software becomes devoid of bugs and uses minimum space requirements as well as minimum time to perform. The test cases were selected beforehand with expected results defined and actual results recorded for comparison. The selection of test cases is done vide “White Box Testing” technique to check the internal programming logic and efficiency and vide ”Black Box Testing” technique to check software requirement fulfillment with intension of finding maximum number of errors with minimum effort and time. Although test cases are a design by considering the cyclomatic complexity, conditional test, still the software code is not in its optional form, as all other possible alternative parts in the software are not considered. At the integration level, the software will be passing to the third party tests which would further enhance the software optimality and efficiency.

- **TEST DATA IMPLEMENTATION AND THEIR RESULT ON:**

The quality and standardization of the software / application package depends truly on the various predefined testing norms and on the performances of the software over those norms. There are various standards existing in the software industry the engineered end product strives to achieve viz. ISO 9002 SEI CMM Level5 etc. These standards are achieved only when the concerned software fulfils the tests as per the respective testing norms predefined in them vide the various test cases and parameters using the CASE topologies. Generally, software is tested both on a stand-alone mode as well after integrating all the modules in the system vide deferent available testing methods/norms.

The following Flow Graph methodology was used while testing the software:

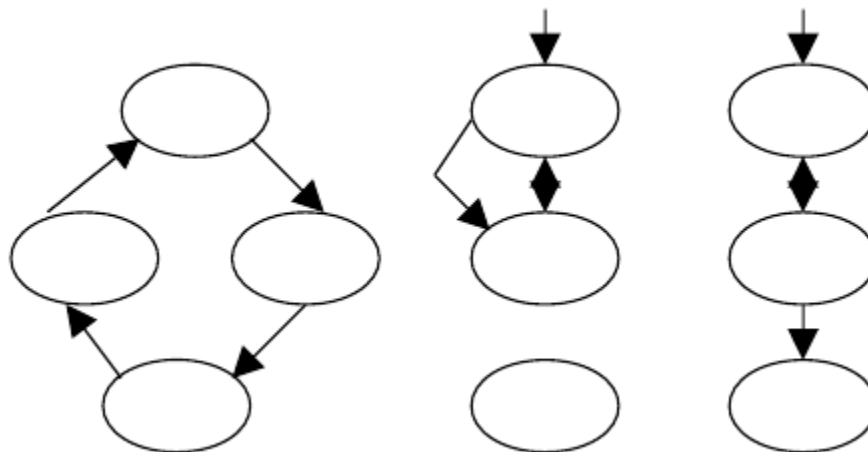


fig 5.1 Flow graph

Here each circle represents one or more non branching procedural language or source code statements in Flow Graph. While performing Condition Testing Domain Testing methodology was selected. While performing Loop Testing simple loops, concatenated loops, nested and unstructured loops were tested thoroughly.

- **TEST CHARACTERS :**

- 1.A good test has a high probability of finding an error.
- 2.A good test is not redundant.
- 3.A good test should be “best of breed”.
- 4.A good test should be neither too simple nor too complex.

- **BLACK BOX TESTING:**

The method of Black Box Testing is used by the software engineer to derive the required results of the test cases:

- 1.Black Box Testing alludes to test that are conducted at the software interface.
- 2.A Black Box Test examines some fundamental aspect of a system with little regard for the internal logic structure of the software.
- 3.A limited number of important logical paths can be selected and exercised.
- 4.Important data structure can be probed for validity.

Black box testing was performed to find errors in the following categories:-

- Incorrect or missing functions
- Graphics error.
- Errors in data in binary format.
- Error in data in integer format.
- File error.
- Pointer error.
- Memory access error.
- Variable error .
- Performance error

- **WHITE BOX TESTING:**

White Box Testing is sometimes called Glass Box Testing. Using White Box Testing methods the software engineer can derive the following test cases:

1. Guarantee that all independent paths within a module have been exercised at least once.
2. Exercise all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds.
4. Exercise internal data structures to ensure the validity.

In White Box Testing efforts were made to handle the following:-

- Number of input parameters equal to number of arguments.
- Parameters and arguments attributes match.
- Number of arguments transmitted is called modules equal to attributes of parameters..
- Unit system of argument transmitted is called modules equal unit system of parameter.
- Number of attributes and order of arguments to build in functions correct.
- Any references to parameters not associated to build in functions correct.
- Input only arguments altered.
- Global variable definition consistent across module.
- Files attributes correct.
- Format specifications matches I/O specification.
- Files opened before use.

- File closed while working is going on.
- I/O errors handled.
- Any textual errors in output information.

- **UNIT TESTING:**

The unit testing is performed to test the validity of the individual units. This is done in the coding phase with the interactive testing. Thus it itself constitutes a majority of functionality test for each logical unit.

- **INTEGRITY TESTING:**

When all the development of all the units or modules is completed and integrated the integrity test phase is started. In this phase the interface between the modules are tested. This phase basically verifies whether inter module exchange of information and events are as per required system behavior.

- **VALIDATION TESTING:**

Tests were performed to find conformity with the requirements. Plans and procedures were designed to ensure that all functional requirements are satisfied. The software was alpha-tested. There are two goals in preparing test plans. Firstly, a properly detailed test plan demonstrates that the program specifications are understood completely. Secondly, the test plan is used during program testing to prove the correctness of the program.

5.1 TEST CASES

Login:

Sl No	Input Values	Test case	Conditional being checked	Result
1	Email	Empty	Please Enter valid Username	Successful
3	Email	Already Exists or not	Login ID should be unique	Successful
4	Password	Empty	Please Enter valid Password	Successful
5	Password	If wrong Password	Enter Password	Successful
6	Password	Length	Length should be less than or equal to 10 character	Successful

Table 5.1 login table

Registration:

Sl No	Input Values	Test case	Conditional being checked	Result
1.	First Name	Empty	It must not be empty	Successful
2	Last Name	Empty	Last Name must not be empty	Successful
3	Email	Empty	Enter valid Email ID.	Successful
4	Password	Empty	Enter valid Password.	Successful

5	Password	Length	Minimum 8 characters required	Successful
6	Confirm Password	Empty	Password and confirmation password must be same	Successful
7	Date Of Birth	Select	Enter valid Username and Password.	Successful

Table 5.2 Registration table

Edit Profile:

Sl No	Input Values	Test case	Conditional being checked	Result
1.	First Name	Null	First Name must not be empty	Successful
2	Last Name	Empty	Last Name must not be empty	Successful
3	City	Empty	City must not be empty	Successful
4	State	Empty	state must not be empty	Successful
5	Pin code	Empty	PIN code must not be empty	Successful
6	Country	SELECT	Please select country	Successful
7	High School	Empty	High School must not be empty	Successful
8	College	SELECT	Please select college	Successful
9	Course	Empty	Course must not be empty	Successful

Table 5.3 Profile table

Photos:

Sl No	Input Values	Test case	Conditional being checked	Result
1.	Image Title	Null	Image title must not be empty	Successful
2	Add Image	BROWSE	Please browse image	Successful
3	Delete Image	Select	Please select image to delete	Successful

Table 5.4 Photos table

6. CONCLUSION

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources –that would generate a proper system for ONLINE SOCIAL NETWORKING.

While making the system, an eye has been kept on making it as user-friendly. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs. As in case of any system development process where there are a number of shortcomings, there have been some shortcomings in the development of this system also.

There are some of the areas of improvement which couldn't be implemented due to time constraints. One such feature was online chat where members can chat with his friends through this website. I also couldn't implement the scrap book and the selling item page for now but we still have two months of internship left in the company so defiantly we will do it in coming months.

7 .REFERENCES

- [1] Prof Mishra. “*Social networking sites effect*” 14 Jan.2014
<http://articles.timesofindia.indiatimes.com/2013-06-23/india/40146190_1.html>
- [2] Upender Singh ,” *Popularity of SNSs among Indian Non-IT Students*”.12Jan.2014. 29 Feb. 2014
< http://ijeit.com/vol%202/Issue%207/IJEIT1412201301_32.pdf>
- [3] Indira Jain *Social networking trends* New Delhi: The Times Of India
- [4] Matas, Alina, “*SNSs becomes an on-line opportunity for employers and students*”, The Washington Post, Nov. 7, 1993, pg. H2
- [5] ”*Building a Career Path*”, The Washington Post, Jan. 19, 1998, pg. F05
- [6] 16 Jan 2014.28 Jan.2014.
<http://en.wikipedia.org/wiki/SNSs_website>.
- [7] 02 Feb.2014.21 Feb.2014.
<<http://www.medianama.com/2010/07/223-SNSs-in-india-the-make-of-finding-a-dream/>>.