

Zhuddle – A Social Networking Website

submitted by

**Mihir Bhavsar(101284)
Prannoy Roche(101285)**

**On
April 2013.**

Under the guidance of

Prof.Subhash Kumar

**Submitted in partial fulfilment of the
requirement's for qualifying for the**

BSc IT semester VI Examination

Submitted to



**ST. XAVIER'S COLLEGE, MUMBAI
MAHAPALIKA MARG
MUMBAI-400001**



ST. XAVIER'S COLLEGE, Mumbai

Project certificate for BSc.IT students

This is to certify that the project entitled

Zhuddle-A Social Networking Website

Undertaken at

St.Xavier's College

Mr.Prannoy Roche Seat No.101285

Mr.Mihir Bhavsar Seat No.101284

In partial fulfilment for the BSc.IT degree(Semester VI)
Examination had not been submitted for any other examination
and does not form part of any other course undergone by the
candidate.

It is further certified that he has completed all required phases
of the project.

Signature

Signature

Signature

(Internal Guide)

(Internal Examiner)

(External Examiner)

Signature

(HOD - IT)

(College Seal)

ACKNOWLEDGMENT

A project is always a coordinated, guided and scheduled team effort aimed at realizing common minimum goals. It is executed by team members, though it never reaches completion by team efforts alone, nor does it reach the far off shores, unless there are light houses to guide the ship sailing astray.

We express our deep gratitude and special thanks to our project guide **Prof.Subhash Kumar** for his rigorous effort and constant support to guide and provide encouragement for our project work. Without their vision and support this synopsis work would never have been possible.

We are also grateful to all the teaching and non-teaching staff, those who have directly or indirectly helped us to make this report a successful one. We are also thankful to our friends for helping us out in making this project report a success.

**PRANNOY ROCHE
MIHIR BHAVSAR**

ABSTRACT

What is a Social Networking Website?

A social networking website helps users to keep in touch with friends, loved ones and also enables you to meet new people and forge a strong relationship that lasts for life.

Users can share ideas, thoughts and even multimedia like images and videos. People can bond over similar interests and form groups to do the same.

Social networking has become a rage, especially among the youth which use it for giving voice to their ideas and even displaying their talents. This has led to social networks becoming more and more complex often using Data mining techniques to analyse user trends and suggest potential friends that the user could make in order to expand his friend circle.

Table of Contents

#	Title	Page No.
1	Synopsis	1
2	Objectives	6
3	System Design and Analysis	7
4	Proposed System	11
5	Diagrams ER Diagram Basic Data Flow diagram UML Diagram Data Flow Diagram	13
6	Gantt Chart	24
7	Software Engineering Processes	27
8	Software Engineering Process Models Applied	30
8	Software Engineering Project Management	38
9	Feasibility Study	42
10	Database Design	46
12	Screenshots	48
13	Code	67
14	Software Testing	106
15	Software Testing Process	107
16	Software Testing Strategies	108
17	Validation Checks	112
18	Future Enhancements	119
19	Bibliography	120

Synopsis

Title of the Project

Zhuddle - A Social Networking Website.

1. Introduction and Objective of the Project.

Social Networking is a nice form of entertainment, great for meeting people with similar interests, and can be a very effective business technique for entrepreneurs, writers, actors, musicians or artists.

However our social networking website comes with a few unique features. They are as follows:

- Users can host debates online and get people express their views on current affairs, politics, sports, culture and a variety of other hot topics.
- Users can interact via discussions and ask for opinions and suggestions from online friends or users.
- Users can customize the appearance of the user interface.

Scope of the Project

- People can interact on a more personal level giving them a broader outlook towards life.
- Helps maintaining relationships and establish new ones by reaching out to people you have never met before.
- Share problems and get people's opinion on how solve them.
- Users can upload pictures thoughts and views.
- Bring about social change.

Technology Used and Project Category:

- **Project Category:** RDBMS, Web Application.
- **Front end:** HTML, CSS, javascript.
- **Intermediate:** Php 5.
- **Back end:** MySql Server

Major Functions

- **Authorization:** There is a unique authority of the system which is the administration of the system.
- **Authentication:** Every user is authenticated against his email address and password.
- **User Form:** An interactive user friendly interface will be displayed for a new user during registration. All forms are under the administrator only. Some fields are mandatory to verify authenticity.
- **Administrator functions:**
 - Authentication of users after registration.
 - Updating the account details of the users.
 - Sending notifications to the users.
 - Deleting/Deactivating accounts based on the user's request.
 - Generating Reports for keeping a track of users and their activities .

- **User functions:**
 - Editing the main profile page.
 - Sending friend requests.
 - Stare views, send messages, share photos and other media.
 - Post comments and send personal messages

Operating Environment

Operating Systems Supported:

- Windows 98/ XP/VISTA/7

Internet Browsers Supported:

- Google Chrome.
- Mozilla Firefox.
- Internet Explorer 8.

Internet connection must be available.

Objectives

The main objectives of our project are as follows:

- ❖ To connect with friends , relatives and keep in touch.
- ❖ People can express views and share life's experiences.
- ❖ To provide a User-friendly interface so users can find it easy to navigate around the website.
- ❖ Users can also update themselves with the latest happenings around the world.
- ❖ Socialize with people and build a strong network.
- ❖ An easy way to share information and improve communications.
- ❖ User's can give the user-interface a personal touch by using various customization options.

System Analysis And Design

Working of the present system

Social Networking sites are becoming a major source of global news and delivering information.

People know everything about what is going on in another person's life. A user's profile page itself gives us a wealth of information about the user's attitude, psyche, way of life and other peculiarities.

Like-minded people can connect immediately and start socializing developing a strong bond in the process.

Users can also upload pictures - entire albums, videos and other multimedia.

Facebook, Orkut, and Myspace are some of the premier social networking websites which are used by billions of people around the world.

Some of the features of these websites are as follows:

1. Chat Feature:

Users can chat with one another using a compact **chatbox** that has been provided. They are often quick and reliable and have added-value features like **emoticons**.

2. Status Update:

Users can post anything from a joke to breaking some serious news. Feedback for the post is provided by a comment-box which holds the profile visitor's comments.

3. Like

Users can like pages about literally anything - from food to celebrities, movies to cars - the choice is yours. Even another's status update can be liked.

4. Language of the World:

Users can communicate using any language and can choose from a wide variety of options.

5. Making Friends:

Users can add new friends to their friend's list.

6. News Feed:

This is a feature which gives you real-time information on what your friends are upto. Any new status update, comment or new friend added is shown to you on the news-feed. This is also called '**'Stalking'**'.

7. Customizability:

User's can customize the appearance of their profile page.

8. Groups:

User's can form Groups and start discussing about various topics. Any news shared is shown only to group members.

9. Privacy options:

User's can choose from various privacy options and hide sensitive information and even block annoying users.

Drawbacks of the present system

1. Too many features:

These website's have too many features which often confuses a new user. Most of these features are generally not needed and they and they simply cram up the website.

2. Spam and scams! :

This problem wouldn't be on Facebook alone. Other social networking sites like Myspace and Bebo are full of spam and scams! Many users receive emails nearly every day from anonymous users asking to date and chat on other websites. It continuously becomes frustrating to receive all these useless spam emails and no solution is being revised.

3. Ruining your professional life!

You might be one of those millions of people who are and use Facebook to post offensive images of you and your friends. Though it is fun and provides humour for all your friends to see, it can be a major drawback when you apply for a job. Employers will want to know more about you and by searching on these social networking sites which contain your offensive content, employers may get repelled and could result in losing the job role that you applied for

Proposed System

Functional Requirements

'Zhuddle' is a social networking website which allows users to interact with one another and stay connected.

Users can share ideas and experiences and people can also get live news updates on their home page. Users can make friends and also send private messages.

Some of the key features are as follows:

Search Bar

Users will be able to search for other users using this search bar. Results would be dynamically displayed according to the keystrokes captured.

Messaging Feature

Users would be able to send private messages to one another. There is a complete inbox management system which lets you delete selected messages or clear your entire inbox. Also replies to a message would not be shown as a new message but as a part of the same thread.

Blab about something

People can share ideas, thoughts, jokes and whole lot of other stuff. Other users (friends of the person who posted the blab) can comment on it and give valuable feedback.

Profile Customization

Users can customize the colour scheme and background images of their profiles. It's great when you can personalize your own stuff.

Dynamic Scroll Bar

The home page initially displays only the latest posts and when the scroll bar reaches the end of the window, the next set of results are automatically loaded and displayed without page refresh. This makes for a very rich user experience.

Diagrams

Entity Relationship Diagram

An Entity-Relationship diagram (ERD) typically serves as the main deliverable of a conceptual data model. While newer approaches to E-R modelling have developed, the E-R approach is still cited by some professionals as “the premier model for conceptual database design”. An ERD is a logical representation of an organization’s data, and consists of three primary components:

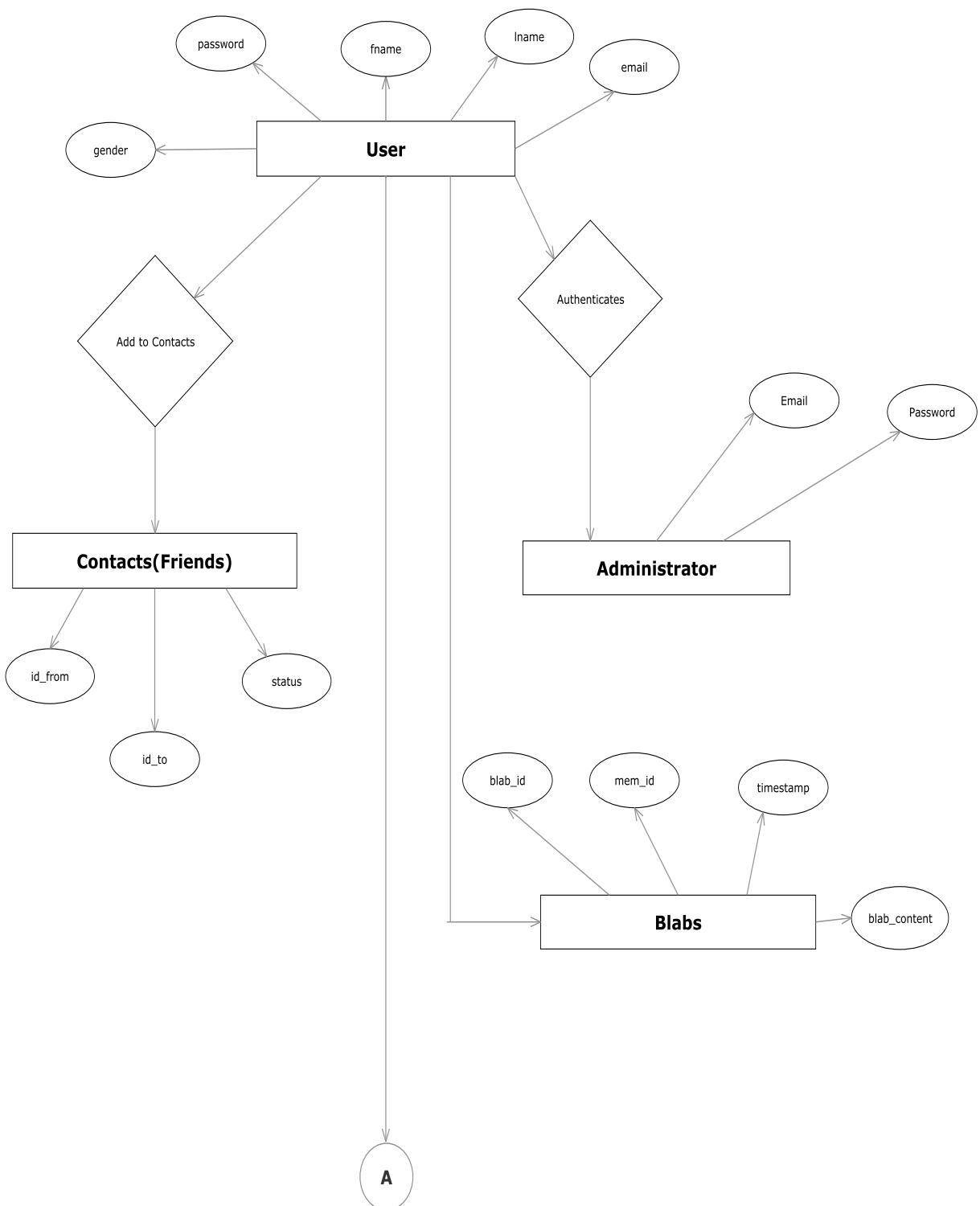
- **Entities** – Major categories of data and are represented by rectangles.
- **Attributes** – Characteristics of entities and are listed within entity rectangles
- **Relationships** – Business relationships between entities and are represented by lines

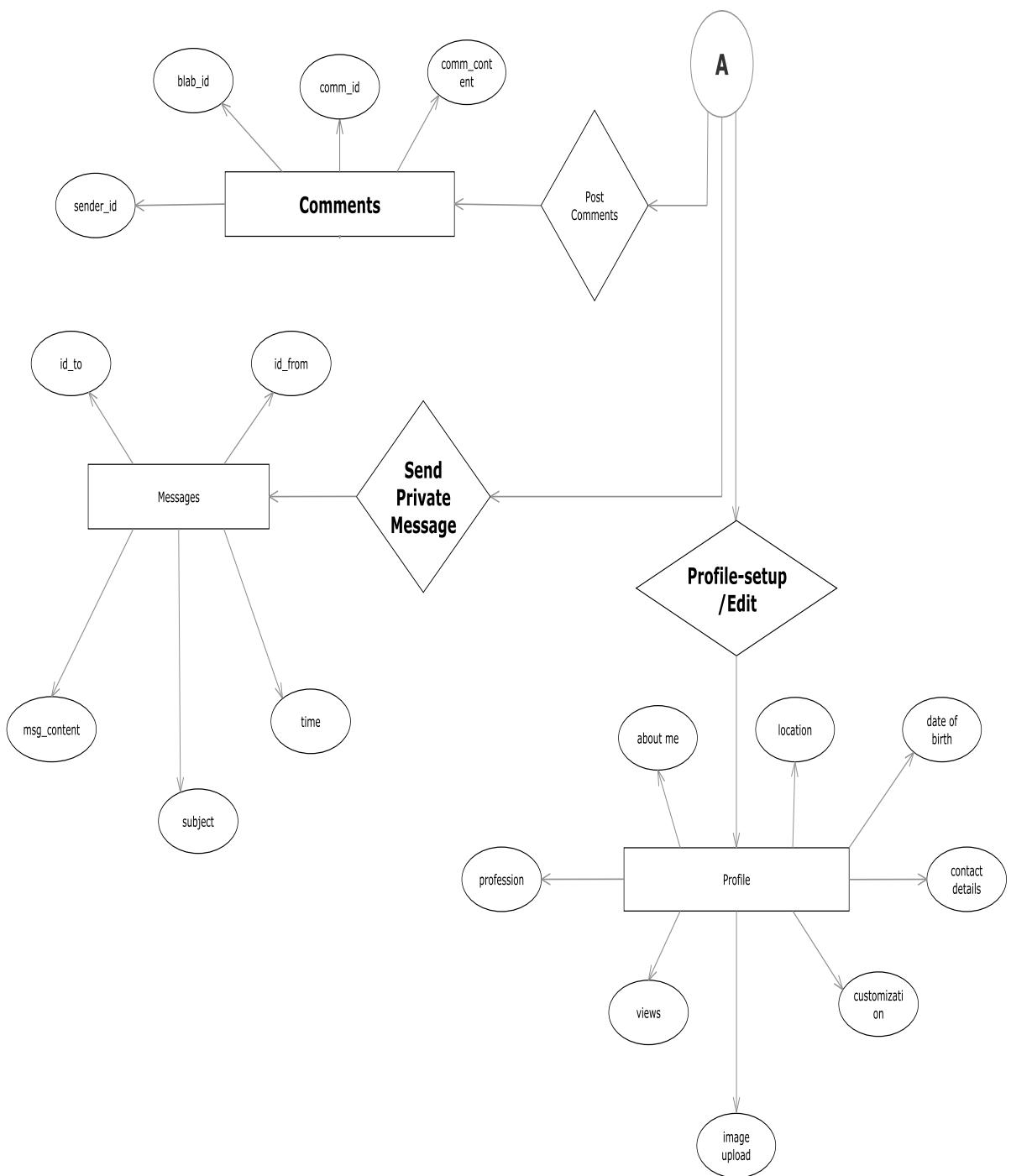
An **Entity** is a person, place, object, event, or concept that an organization wants to maintain data on. Each entity has a unique identity that differentiates it from other entities

An **Attribute** is a characteristic of an entity that is relevant to the organization. When defining an attribute, an analyst should state why the attribute is important, what is included in the attribute’s value, the source of the value, and whether or not that value can change. Again, a sound understanding of an organization’s business should assist the analyst in compiling relevant attributes.

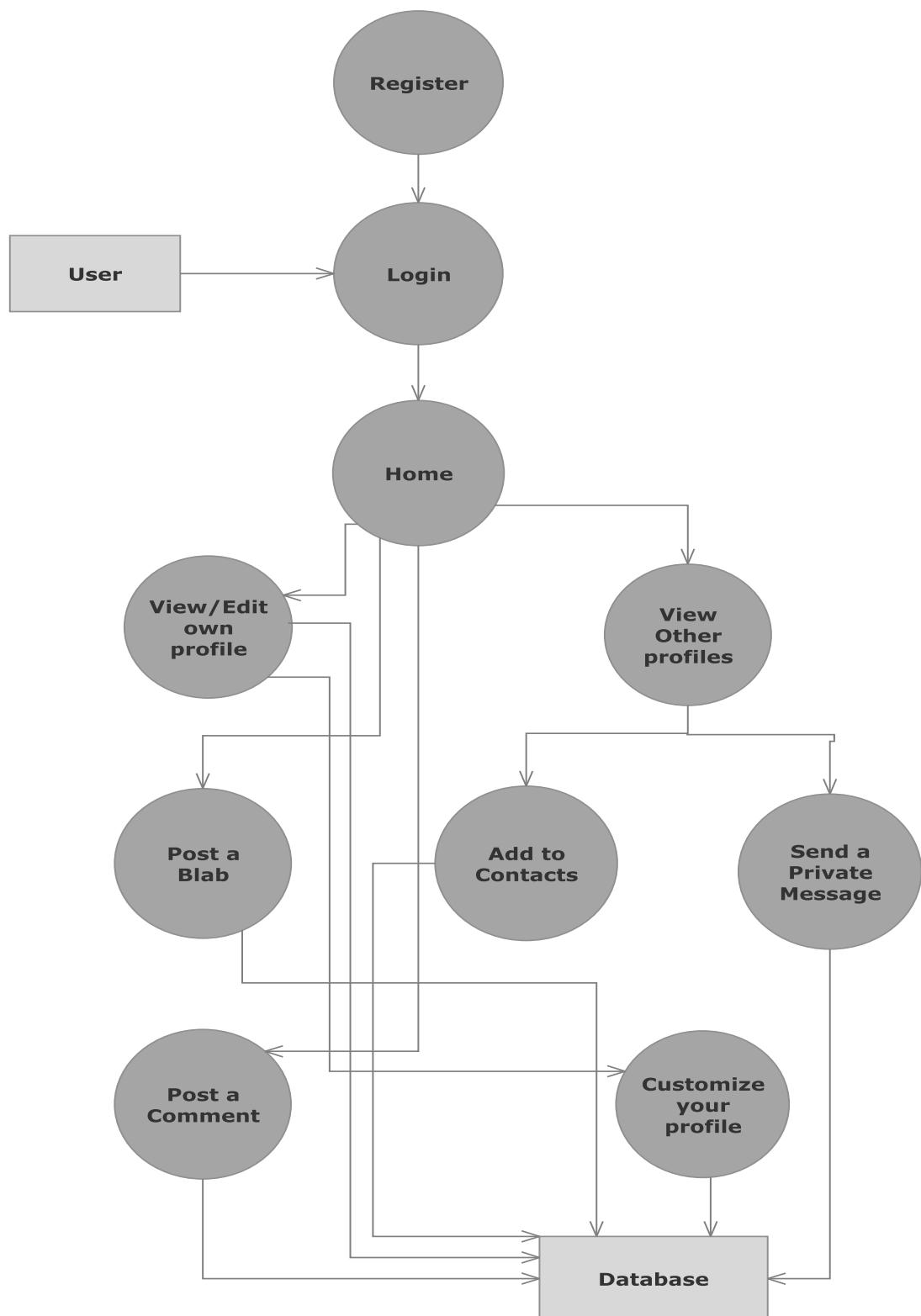
Relationships link the various components in an E-R diagram together. Relationships depict either some kind of event occurring or a natural link between entity instances.

A relationship’s **degree** indicates the number of entity types that participate in a given relationship. The three most common relationship degrees are: *Unary* (between instances of one entity type), *Binary* (between instances of two entity types), and *Ternary* (between three entity types).





Basic Data Flow Diagram



UML DIAGRAM

UML stands for Unified Modeling Language. This object-oriented system of notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation. These renowned computer scientists fused their respective technologies into a single, standardized model. Today, UML is accepted by the Object Management Group (OMG) as the standard for modelling object oriented programs.

Types of UML Diagrams

UML defines nine types of diagrams: class (package), object, use case, sequence, collaboration, statechart, activity, component, and deployment.

Class Diagrams

Class diagrams are the backbone of almost every object oriented method, including UML. They describe the static structure of a system.

Package Diagrams

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

Object Diagrams

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

Use Case Diagrams

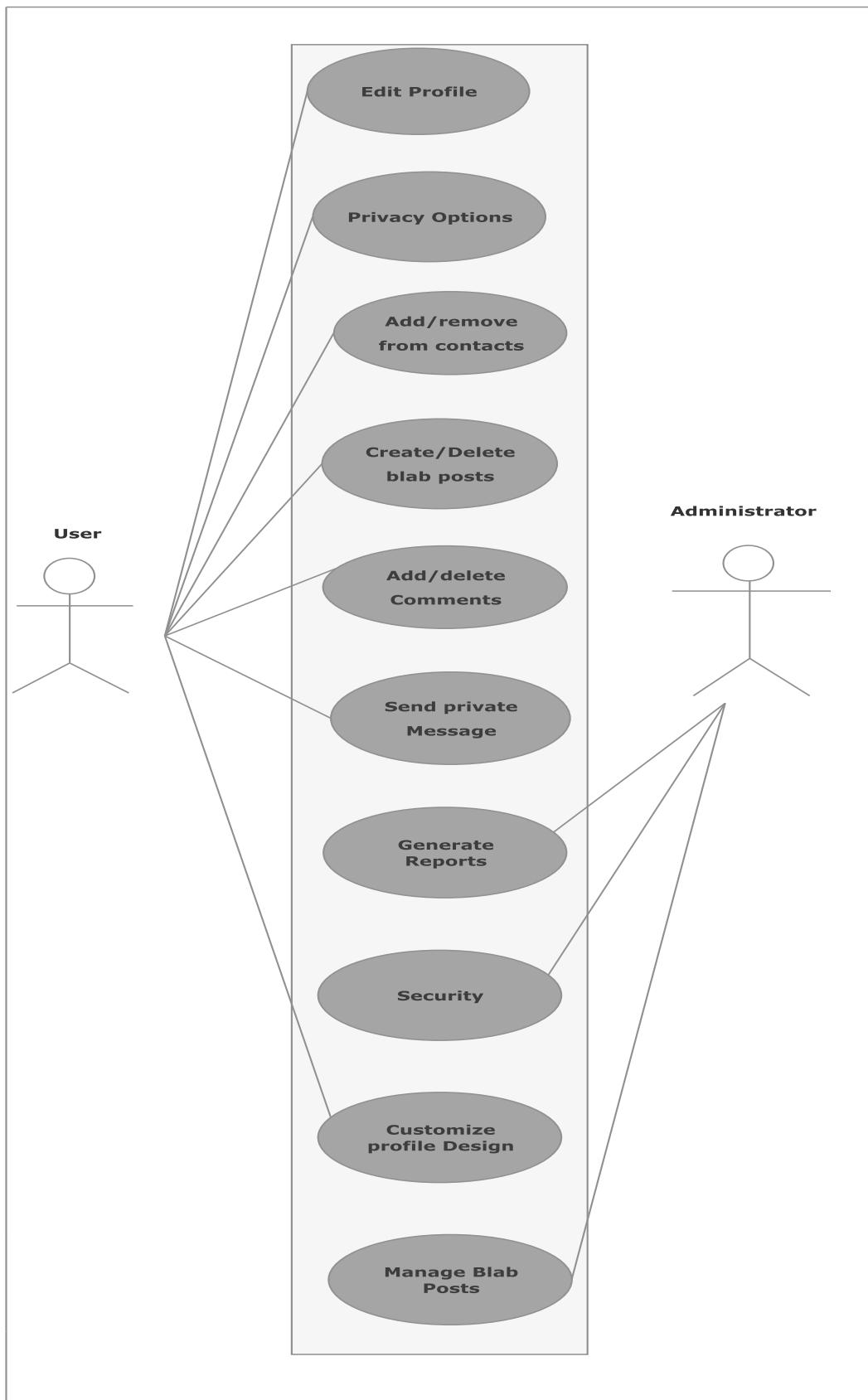
Use case diagrams model the functionality of system using actors and use cases.

Sequence Diagrams

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

Collaboration Diagrams

Collaboration diagrams represent interactions between objects as a series of sequenced messages. Collaboration diagrams describe both the static structure and the dynamic behaviour of a system.



Data Flow Diagram

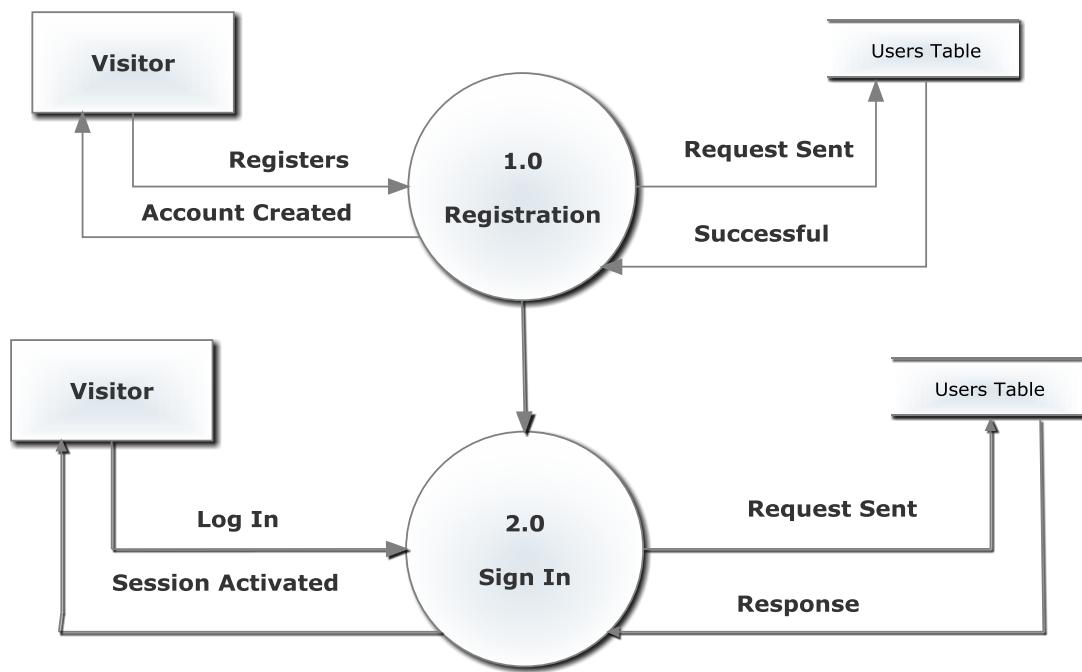
A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

It is common practice to draw the context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram the system's interactions with the outside world are modelled purely in terms of data flows across the *system boundary*. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

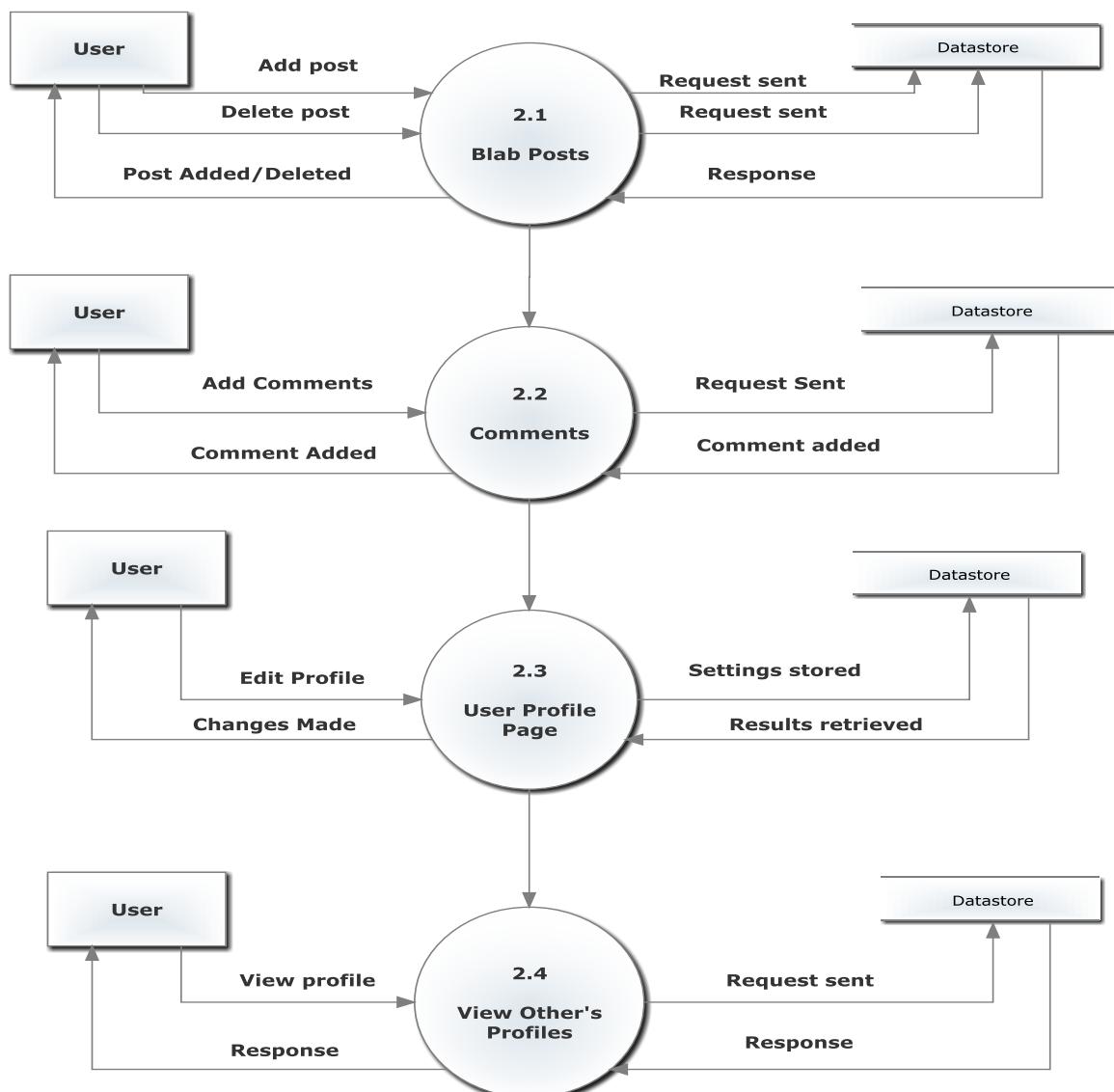
Level 0 DFD



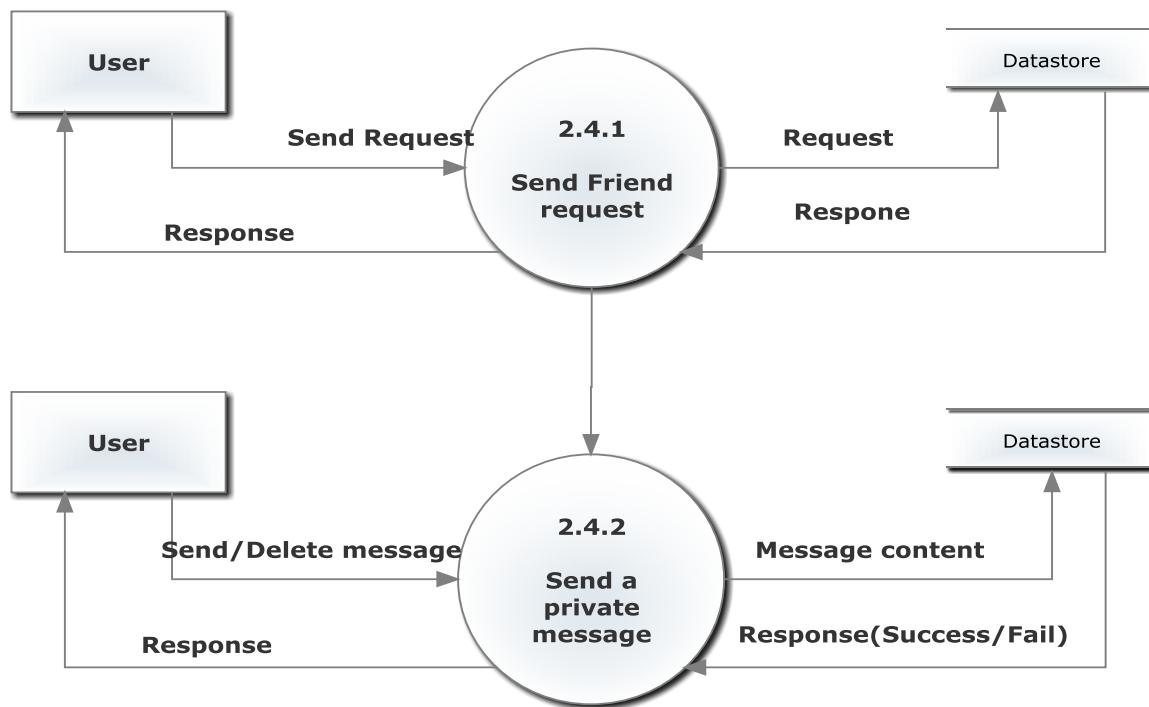
Response



Level 1 DFD



Level 2 DFD



Gantt Chart

A **Gantt chart** is a type of bar chart, developed by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.

Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced.^[1] In recognition of Henry Gantt's contributions, the Henry Laurence Gantt Medal is awarded for distinguished achievement in management and in community service. This chart is also used in Information Technology to represent data that has been collected.

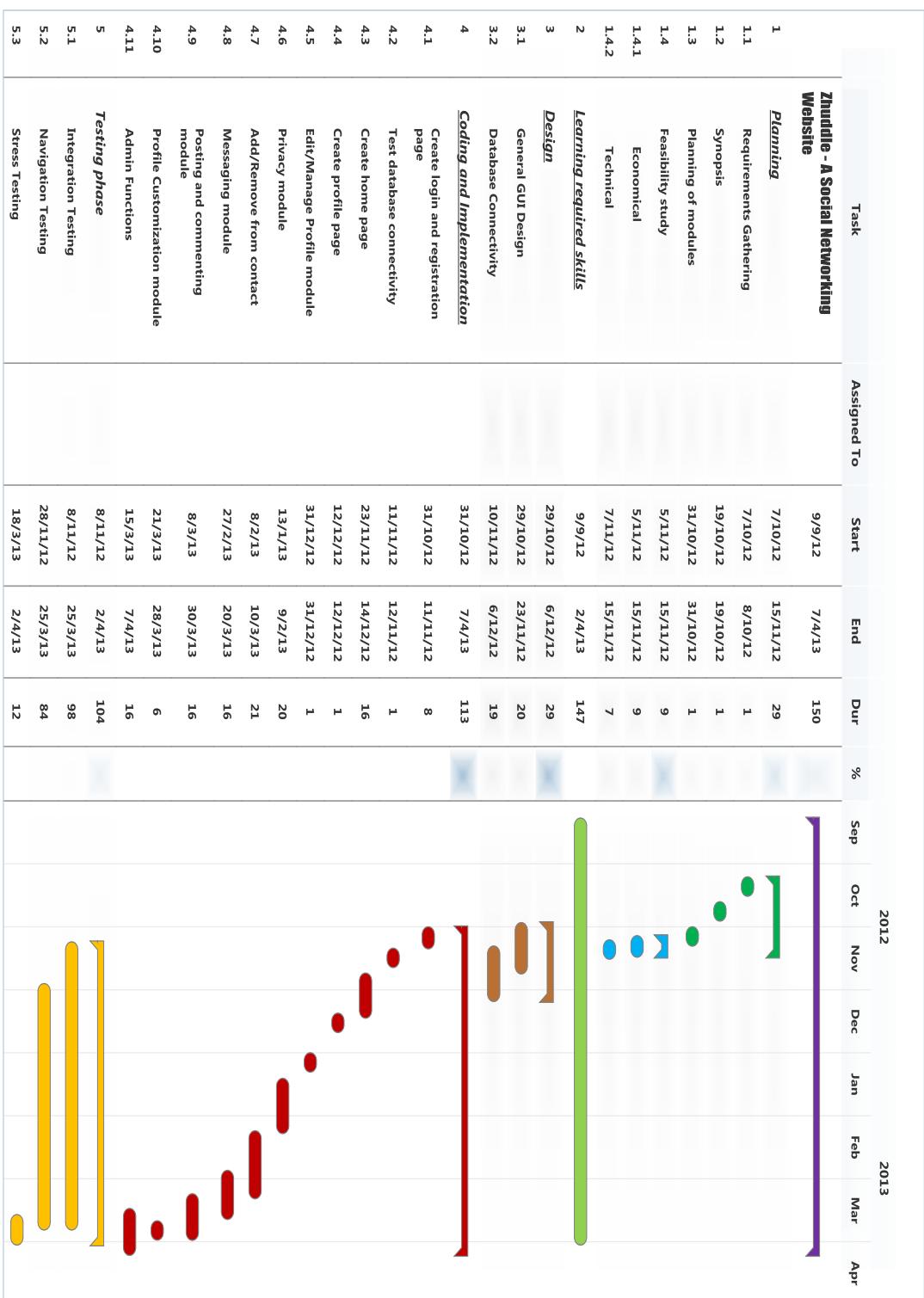
A common error made by those who equate Gantt chart design with project design is that they attempt to define the project work breakdown structure at the same time that they define schedule activities. This practice makes it very difficult to follow the 100% Rule. Instead the WBS should be fully defined to follow the 100% Rule, then the project schedule can be designed.

Although a Gantt chart is useful and valuable for small projects that fit on a single sheet or screen, they can become quite unwieldy for projects with more than about 30 activities. Larger Gantt charts may not be suitable for most computer displays. A related criticism is that Gantt charts communicate relatively little information per unit area of display. That is, projects are often considerably more complex than can be communicated effectively with a Gantt chart.

Gantt charts only represent part of the triple constraints (cost, time and scope) on projects, because they focus primarily on schedule management. Moreover, Gantt charts do not represent the size of a project or the relative size of work elements, therefore the magnitude of a behind-schedule condition is easily miscommunicated. If two projects are the same number of days behind schedule, the larger project has a larger effect on resource utilization, yet the Gantt does not represent this difference.

Although project management software can show schedule dependencies as lines between activities, displaying a large number of dependencies may result in a cluttered or unreadable chart.

Because the horizontal bars of a Gantt chart have a fixed height, they can misrepresent the time-phased workload (resource requirements) of a project, which may cause confusion especially in large projects. In the example shown in this article, Activities E and G appear to be the same size, but in reality they may be different orders of magnitude. A related criticism is that all activities of a Gantt chart show planned workload as constant. In practice, many activities (especially summary elements) have front-loaded or back-loaded work plans, so a Gantt chart with percent-complete shading may actually miscommunicate the true schedule performance status



Software Engineering Process

Process, Methods, and Tools

Software engineering is a layered technology. Any engineering approach (including software engineering) must rest on an organizational commitment to quality. Total quality management and similar philosophies foster a continuous process improvement culture, and this culture ultimately leads to the development of increasingly more mature approaches to software engineering. The bedrock that supports software engineering is a quality focus.

The foundation for software engineering is the *process* layer. Software engineering process is the glue that holds the technology layers together and enables rational and timely development of computer software. Process defines a framework for a set of *key process areas* that must be established for effective delivery of software engineering technology. The key process areas form the basis for management control of software projects and establish the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are established, quality is ensured, and change is properly managed. Software engineering *methods* provide the technical how-to's for building software. Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing, and support. Software engineering methods rely on a set of basic principles that govern each area of the technology and include modeling activities and other descriptive techniques. Software engineering *tools* provide automated or semi-automated support for the process and the methods.

A process framework establishes the foundation for a complete software engineering process by identifying a small number of *framework activities* that are applicable to all software projects, regardless of their size or complexity. In addition, the process framework encompasses a set of *umbrella activities* that are applicable across the entire software process. A generic process framework for software engineering encompasses five activities:

Communication. Before any technical work can commence, it is critically important to communicate and collaborate with the customer (and other Stakeholders) The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.

Planning. Any complicated journey can be simplified if a map exists. A software project is a complicated journey, and the planning activity creates a “map” that helps guide the team as it makes the journey. The map—called a software project plan—defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.

Modeling. Whether you’re a landscaper, a bridge builder, an aeronautical engineer, a carpenter, or an architect, you work with models every day. You create a “sketch” of the thing so that you’ll understand the big picture—what it will look like architecturally, how the constituent parts fit together, and many other characteristics. If required, you refine the sketch into greater and greater detail in an effort to better understand the problem and how you’re going to solve it. A software engineer does the same thing by creating models to better understand software requirements and the design that will achieve those requirements.

Construction. This activity combines code generation (either manual or automated) and the testing that is required to uncover errors in the code.

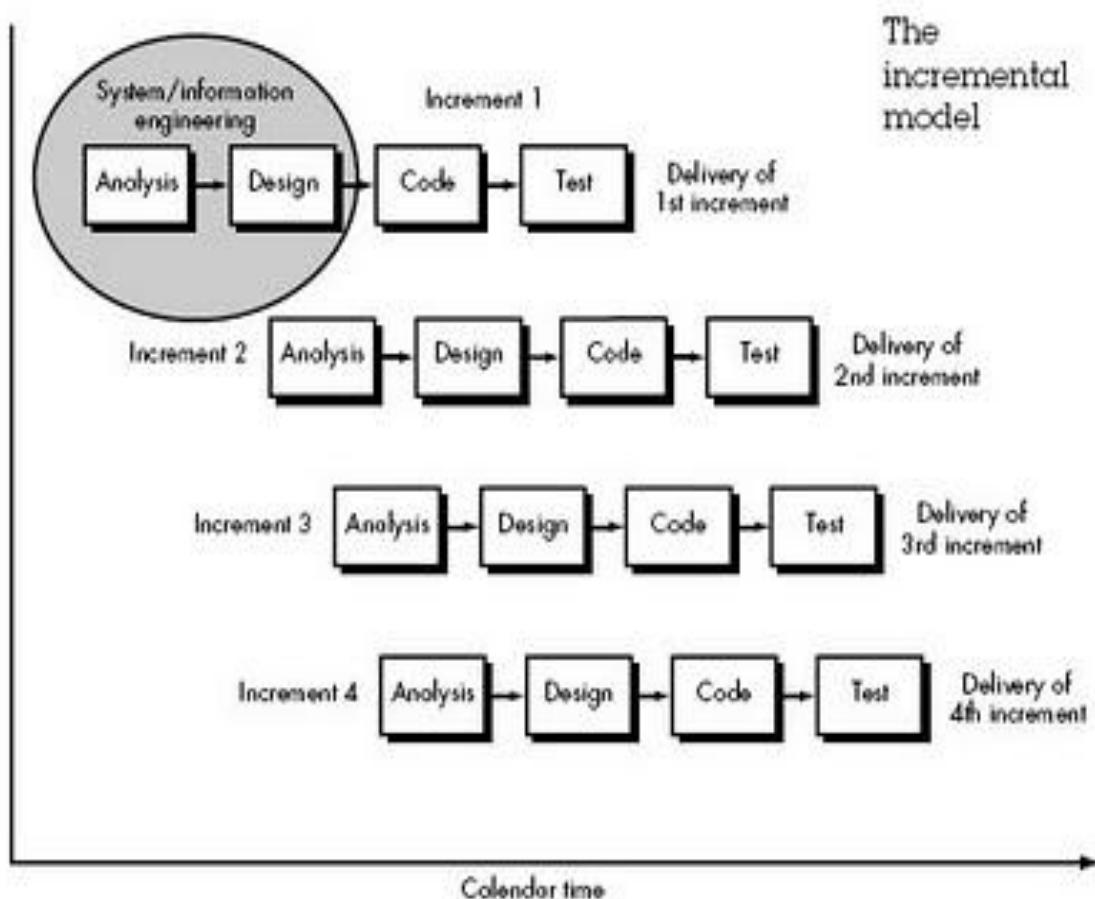
Deployment. The software (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

These five generic framework activities can be used during the development of small, simple programs, the creation of large Web applications, and for the engineering of large, complex computer-based systems. The details of the software process will be quite different in each case, but the framework activities remain the same.

Software Engineering Process Models

Applied

Incremental Model:



There are many situations in which initial software requirements are reasonably well defined, but the overall scope of the development effort precludes a purely linear process. In addition, there may be a compelling need to provide a limited set of software functionality to users quickly and then refine and expand on that functionality in later software releases. In such cases, you can choose a process model that is designed to produce the software in increments. The **incremental** model combines elements of linear and parallel process flows. The incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces deliverable “increments” of the software in a manner that is similar to the increments produced by an evolutionary process flow.

For example, word-processing software developed using the incremental paradigm might deliver basic file management, editing, and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in the third increment and advanced page layout capability in the fourth increment. It should be noted that the process flow for any increment can incorporate the prototyping paradigm.

When an incremental model is used, the first increment is often a **core product**. That is, basic requirements are addressed but many supplementary features (some known, others unknown) remain undelivered.

The incremental process model focuses on the delivery of an operational product with each increment. Early increments are stripped-down versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user. Incremental development is particularly useful when staffing is unavailable for a complete implementation by the business deadline that has been established for the project.

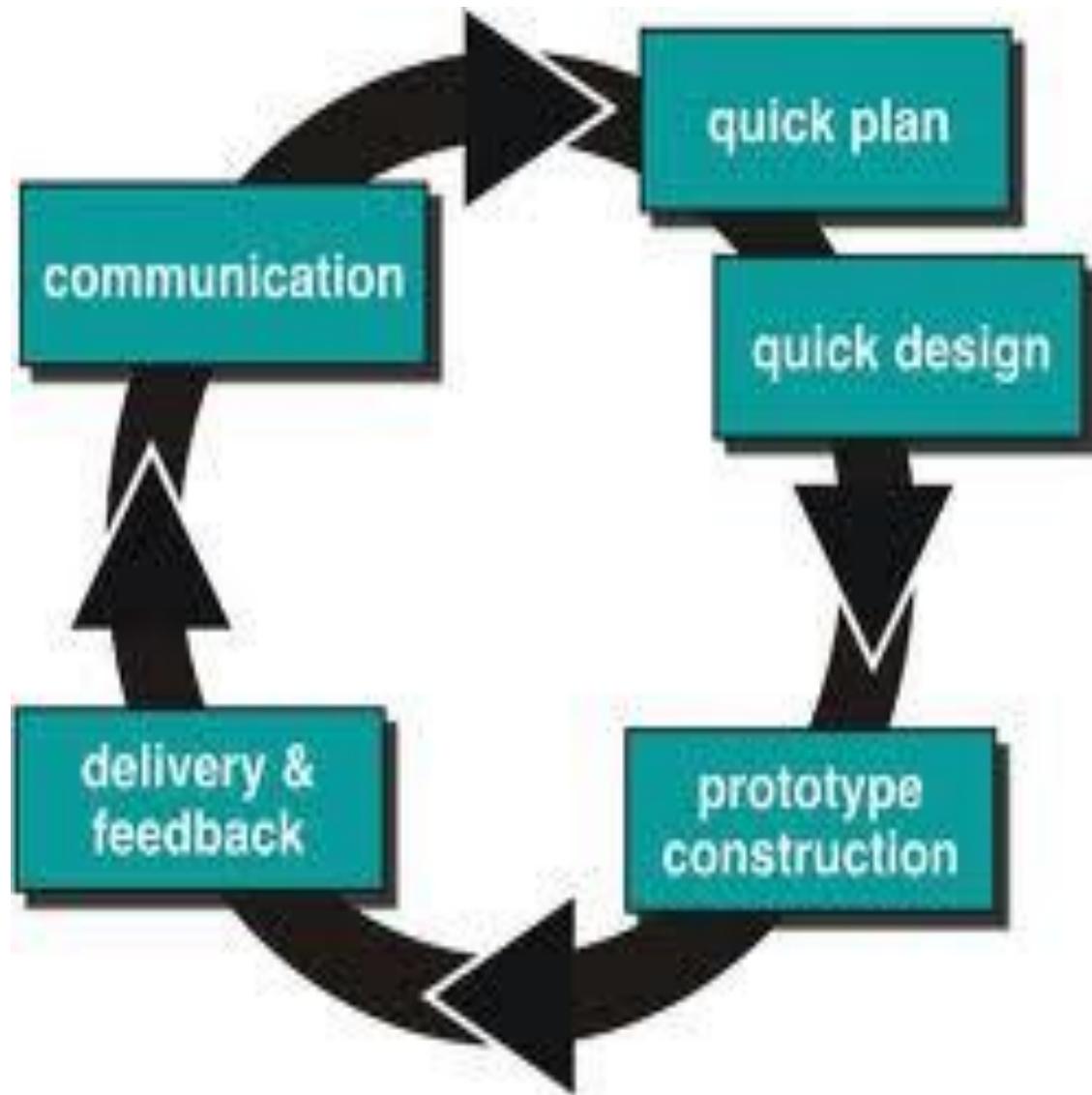
Early increments can be implemented with fewer people. If the core product is well received, then additional staff (if required) can be added to implement the next increment. In addition, increments can be planned to manage technical risks. For example, a major system might require the availability of new hardware that is under development and whose delivery date is uncertain. It might be possible to plan early increments in a way that avoids the use of this hardware, thereby enabling partial functionality to be delivered to end users without inordinate delay.

Evolutionary Process Models

Software, like all complex systems, evolves over a period of time. Business and product requirements often change as development proceeds, making a straight line path to an end product unrealistic; tight market deadlines make completion of a comprehensive software product impossible, but a limited version must be introduced to meet competitive or business pressure; a set of core product or system requirements is well understood, but the details of product or system extensions have yet to be defined. In these and similar situations, you need a process model that has been explicitly designed to accommodate a product that evolves over time.

Evolutionary models are iterative. They are characterized in a manner that enables you to develop increasingly more complete versions of the software.

The Prototyping Concept

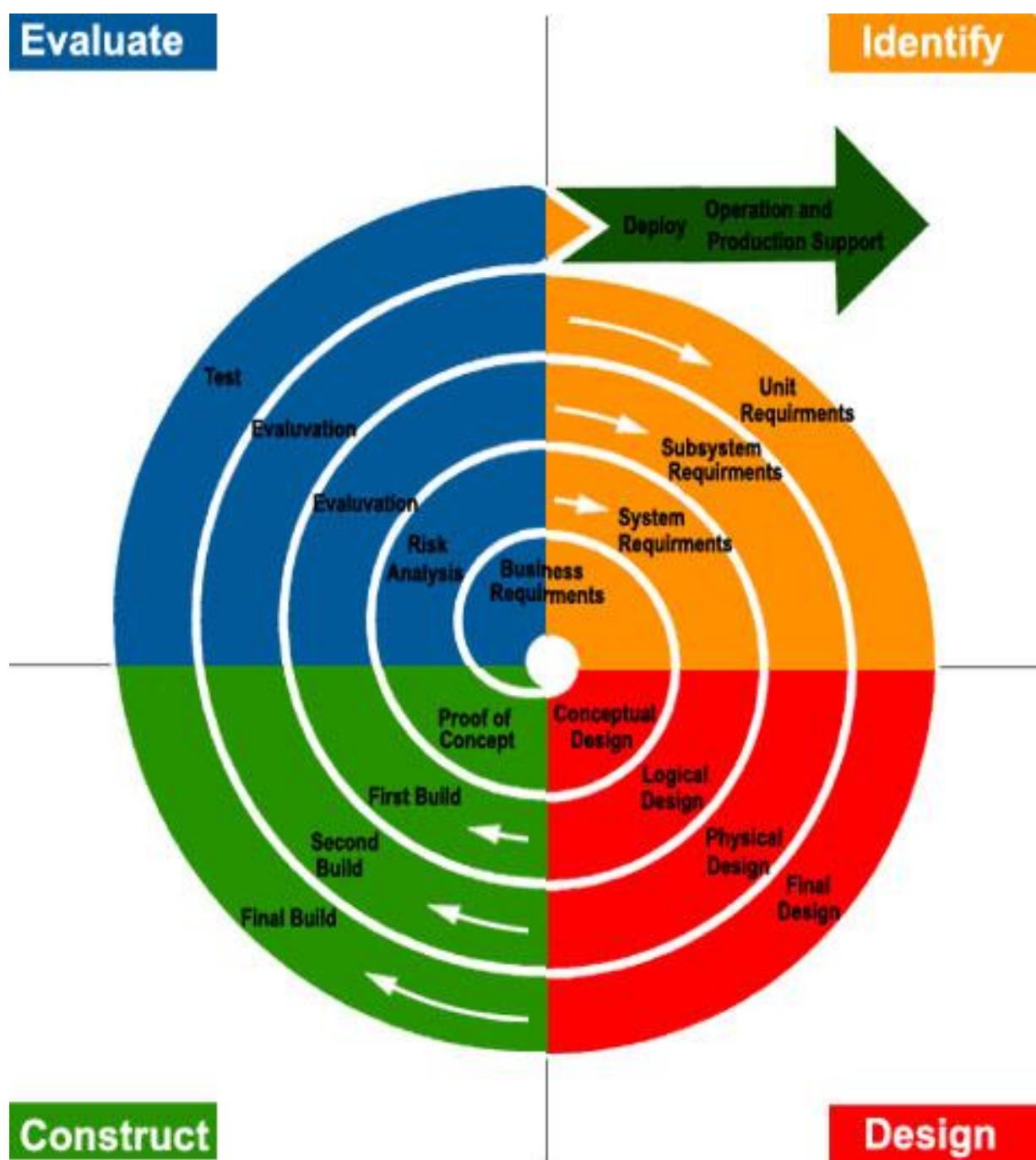


Often, a customer defines a set of general objectives for software, but does not identify detailed requirements for functions and features. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human-machine interaction should take. In these, and many other situations, a ***prototyping paradigm*** may offer the best approach. Although prototyping can be used as a stand-alone process model, it is more commonly used as a technique that can be implemented within the context of any one of the process models noted in this chapter. Regardless of the manner in which it is applied, the prototyping paradigm assists you and other stakeholders to better understand what is to be built when requirements are fuzzy.

The prototyping paradigm begins with communication. You meet with other stakeholders to define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A prototyping iteration is planned quickly, and modeling (in the form of a “quick design”) occurs. A quick design focuses on a representation of those aspects of the software that will be visible to end users (e.g., human interface layout or output display,format).

The quick design leads to the construction of a prototype. The prototype is deployed and evaluated by stakeholders, who provide feedback that is used to further refine requirements. Iteration occurs as the prototype is tuned to satisfy the needs of various stakeholders, while at the same time enabling you to better understand what needs to be done. Ideally, the prototype serves as a mechanism for identifying software requirements. If a working prototype is to be built, you can make use of existing program fragments or apply tools (e.g., report generators and window managers) that enable working programs to be generated quickly.

The Spiral Model



A spiral model is divided into a set of framework activities defined by the software engineering team. For illustrative purposes, I use the generic framework activities discussed earlier. Each of the framework activities represent one segment of the spiral path illustrated in Figure 2.7. As this evolutionary process begins, the software team performs activities that are implied by a circuit around the spiral in a clockwise direction, beginning at the center. Risk is considered as each revolution is made. ***Anchored point milestones***—a combination of work products and conditions that are attained along the path of the spiral—are noted for each evolutionary pass.

The first circuit around the spiral might result in the development of a product specification; subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software.

Each pass through the planning region results in adjustments to the project plan. Cost and schedule are adjusted based on feedback derived from the customer after delivery. In addition, the project manager adjusts the planned number of iterations required to complete the software.

Unlike other process models that end when software is delivered, the spiral model can be adapted to apply throughout the life of the computer software. Therefore, the first circuit around the spiral might represent a “concept development project” that starts at the core of the spiral and continues for multiple iterations until development is complete. If the concept is to be developed into an actual product, the process proceeds outward on the spiral and a “new product development project” commences. The new product will evolve through a number of iterations around the spiral. Later, a circuit around the spiral might be used to represent a “product enhancement project.”

In essence, the spiral, when characterized in this way, remains operative until the software is retired. There are times when the process is dormant, but whenever a change is initiated, the process starts at the appropriate entry point(e.g., product enhancement). The spiral model is a realistic approach to the development of large-scale systems and software. Because software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level.

The spiral model uses prototyping as a risk reduction mechanism but, more important, enables you to apply the prototyping approach at any stage in the evolution of the product. It maintains the systematic stepwise approach suggested by the classic life cycle but incorporates it into an iterative framework that more realistically reflects the real world. The spiral model demands a direct consideration of technical risks at all stages of the project and, if properly applied, should reduce risks before they become problematic.

But like other paradigms, the spiral model is not a panacea. It may be difficult to convince customers (particularly in contract situations) that the evolutionary approach is controllable. It demands considerable risk assessment expertise and relies on this expertise for success. If a major risk is not uncovered and managed, problems will undoubtedly occur.

Software Project Management

Effective software project management focuses on the four P's:

- People
- Product
- Process
- Project

The order is not arbitrary. The manager who forgets that software engineering work is an intensely human endeavor will never have success in project management. A manager who fails to encourage comprehensive stakeholder communication early in the evolution of a product risks building an elegant solution for the wrong problem.

The manager who pays little attention to the process runs the risk of inserting competent technical methods and tools into a vacuum. The manager who embarks without a solid project plan jeopardizes the success of the project.

The People

The cultivation of motivated, highly skilled software people has been discussed since the 1960s. In fact, the “people factor” is so important that the Software Engineering Institute has developed a *People Capability Maturity Model* (People-CMM), in recognition of the fact that “every organization needs to continually improve its ability to attract, develop, motivate, organize, and retain the workforce needed to accomplish its strategic business objectives”. The people capability maturity model defines the following key practice areas for software people:

- Staffing
- Communication and coordination,
- Work environment,
- Performance management
- Training compensation,
- Competency analysis and development,
- Career development,
- Workgroup development
- Team/culture development

The Product

Before a project can be planned, product objectives and scope should be established, alternative solutions should be considered, and technical and management constraints should be identified.

As a software developer, you and other stakeholders must meet to define product objectives and scope. In many cases, this activity begins as part of the system engineering or business process engineering and continues as the first step in software requirements engineering . Objectives identify the overall goals for the product (from the stakeholders' points of view) without considering how these goals will be achieved.

Scope identifies the primary data, functions, and behaviours that characterize the product, and more important, attempts to bound these characteristics in a quantitative manner.

The Process

A software process (Chapters 2 and 3) provides the framework from which a comprehensive plan for software development can be established. A small number of framework activities are applicable to all software projects, regardless of their size or complexity.

A number of different task sets—tasks, milestones, work products, and quality assurance points—enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team.

Finally, umbrella activities—such as software quality assurance, software configuration management, and measurement—overlay the process model. Umbrella activities are independent of any one framework activity and occur.

The Project

We conduct planned and controlled software projects for one primary reason—it is the only known way to manage complexity. And yet, software teams still struggle.

Although the success rate for present-day software projects may have improved somewhat, our project failure rate remains much higher than it should be.

To avoid project failure, a software project manager and the software engineers who build the product must avoid a set of common warning signs, understand the critical success factors that lead to good project management, and develop a common-sense approach for planning, monitoring, and controlling the project.

Feasibility Study

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behaviour of the system.

There are three parameters associated with feasibility study:

- Technical feasibility
- Economic feasibility
- Operational feasibility

Technical Feasibility

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include :

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.
- Facility to communicate data to distant locations.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirements.

For "Zhuddle" to run a server, database and a web browser are needed. It on Apache Server or MySQL via PHPMyAdmin.

Basic hardware requirement is a PC system, with colour monitor, internet connection and minimum RAM requirements and enough space on the hard disk.

Economic Feasibility

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost / Benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs.

If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an outgoing effort that improves in accuracy at each phase of the system life cycle.

The economic feasibility study evaluates the cost of the software development against the ultimate income of the benefits received from the developed system. There must be scope for profit after the successful completion of the project.

The software required to develop the software are readily available in the market at low prices and are thus economically feasible. To run "Zhuddle", all we require is an OS already installed in the system and a web browser which is also present. Latest web browsers are freely available online and can be downloaded.

Benefits far outweigh costs as the project can be easily expandable with a wide scope for new features and several users.

Operational Feasibility

This is mainly related to human organizational and political aspects.
The points to be considered are:

- What changes will be brought with the system?
- What organizational structure are disturbed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

This feasibility study is carried out by a small group of people who are familiar with information system technique and are skilled in system analysis and design process.

Proposed projects are beneficial only if they can be turned into information system that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed.

Database Design

Table Users

Column Name	Type	Attribute
id	int	Primary key
join_date	date	
fname	varchar	
lname	varchar	
email	varchar	
password	varchar	
gender	varchar	
state	int	
profile_pic	varchar	
temp_profile_pic	varchar	
contact_list	text	

Table User General Info

Column Name	Type	Attribute
id	int	
name	varchar	
gender	varchar	
dob	varchar	
Location_country	varchar	
Location_city	varchar	
profession	varchar	
relation	varchar	
About_self	varchar	

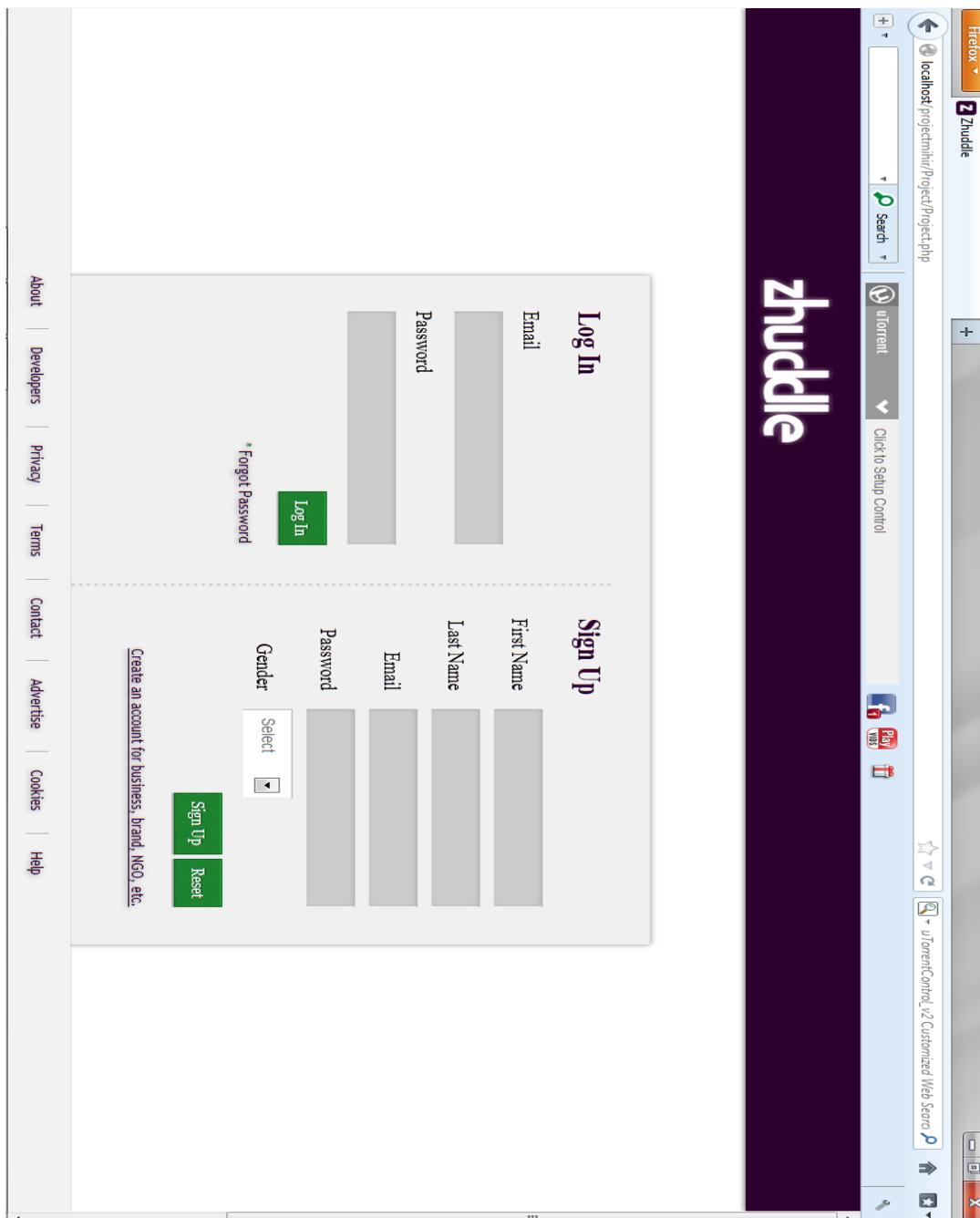
Table Blabs

Column Name	Type	Attribute
blab_id	int	Primary key
mem_id	int	Foreign Key
blab_content	varchar	
blab_date	datetime	
timestamp	int	

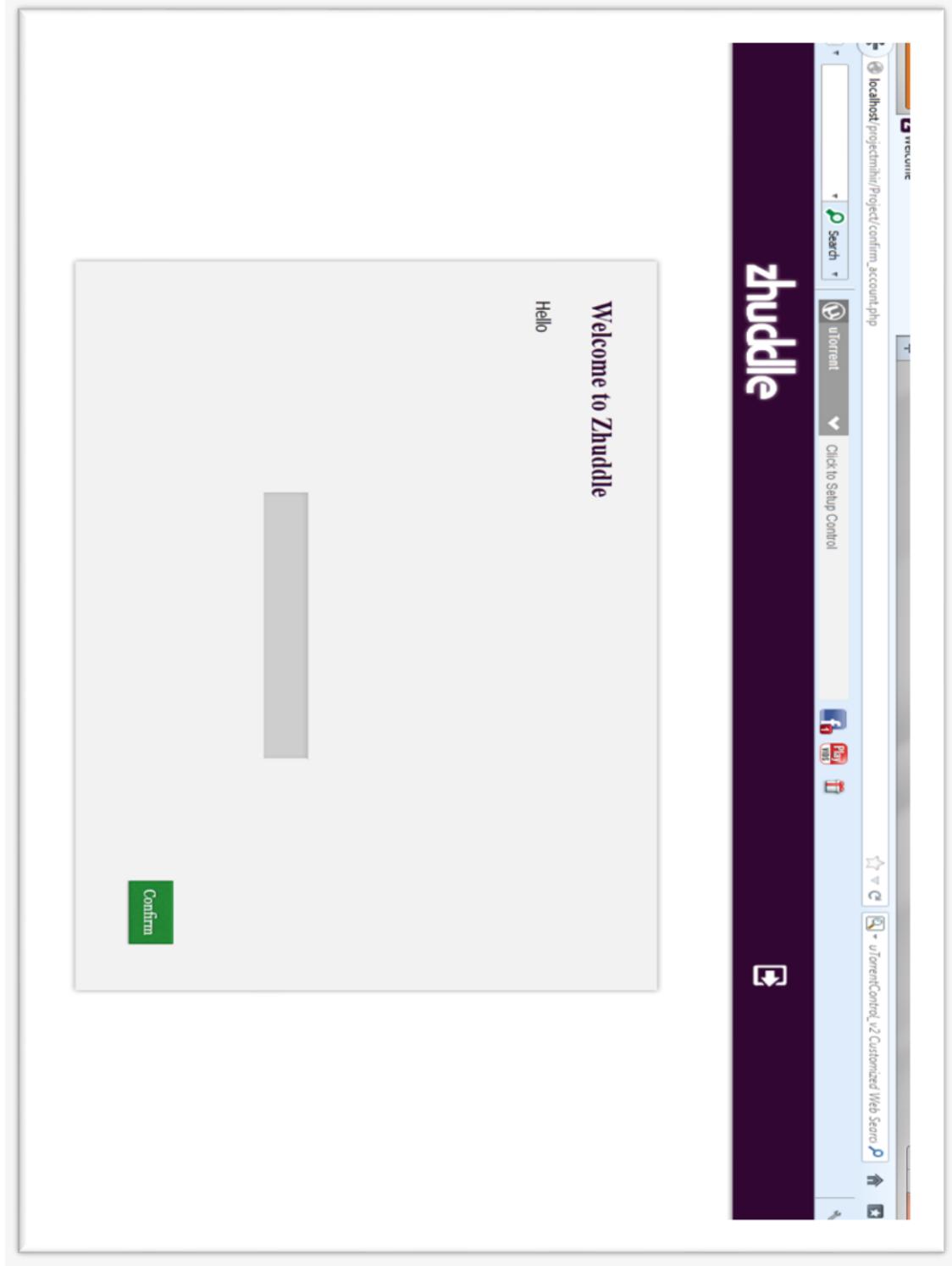
Table Responses

Column Name	Type	Attribute
type	varchar	
response	varchar	
id_from	int	
id_to	int	
time	datetime	
status	varchar	

Screenshots



Registration/Login page



Successful Registration/Authentication Page.

The screenshot shows a user profile page for 'Prannoy Roche' on a website called 'zhuddle'. The top navigation bar includes a search bar, a torrent link, and a 'Click to Setup Control' button. Below the header is a sidebar with links for 'Change Profile Photo', 'Responses [0]', 'Messages [0]', 'Requests [0]', and 'Contacts [4]'. The main content area displays the user's profile information in a grid format:

	General	Contact	Views	Privacy	Customize
Gender	Male	Select <input type="button" value="▼"/>	DD/MM/YYYY		
Date of Birth	03/07/1991				
Location	Chennai, India	Country <input type="button" value="▼"/>			
Profession	Student	Select <input type="button" value="▼"/>			
Relationship Status	Single	Select <input type="button" value="▼"/>			
About yourself	Hi				

At the bottom left are 'Save' and 'Cancel' buttons.

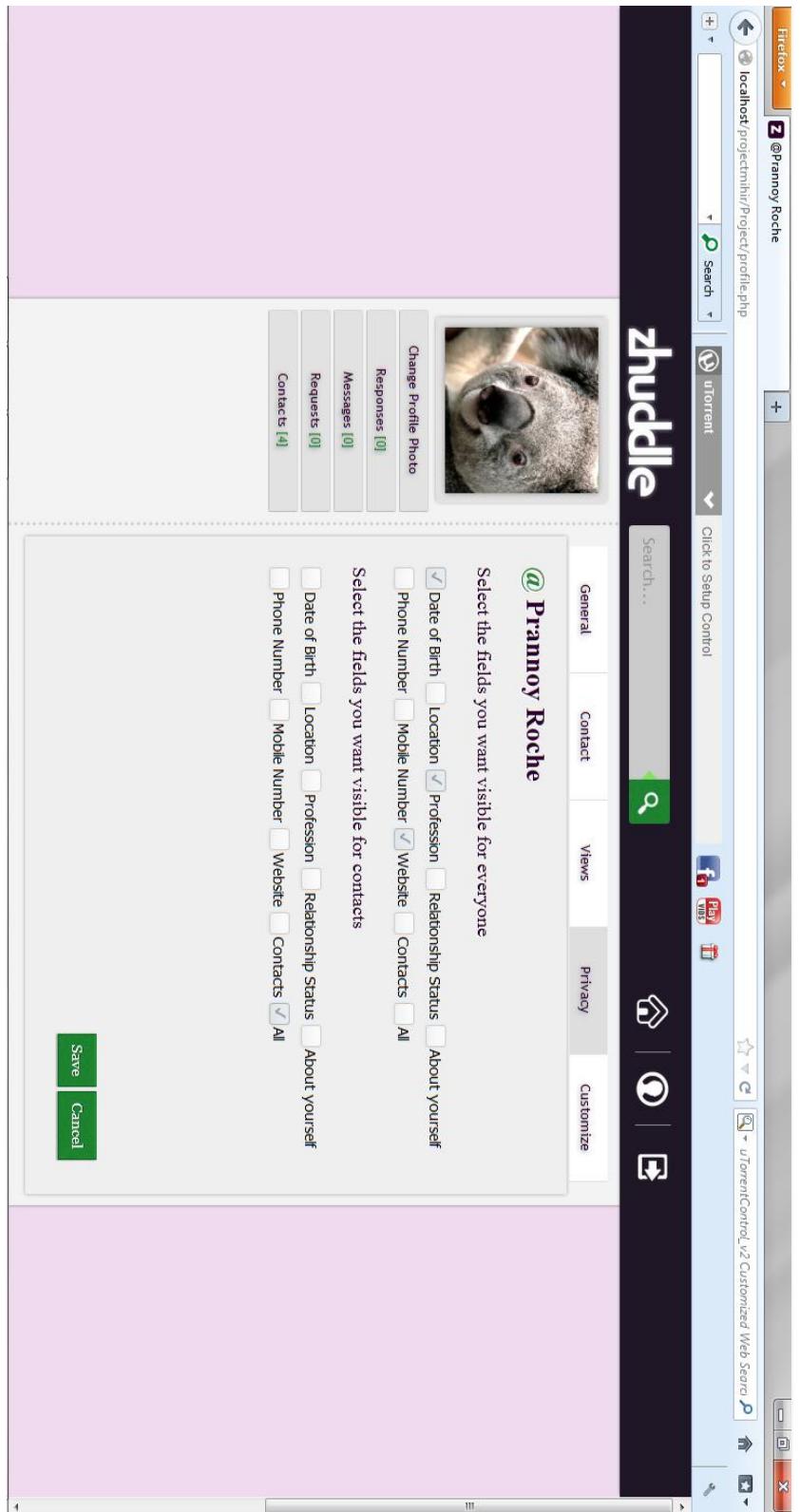
User Profile Page(Editing options)

The screenshot shows a profile editing interface for a user named 'Prannoy Roche'. The top navigation bar includes a search bar, a torrent download link, and a 'Click to Setup Control' button. Below the header is a sidebar with links for 'Change Profile Photo', 'Responses [0]', 'Messages [0]', 'Requests [0]', and 'Contact to [4]'. The main content area displays contact information in a table format:

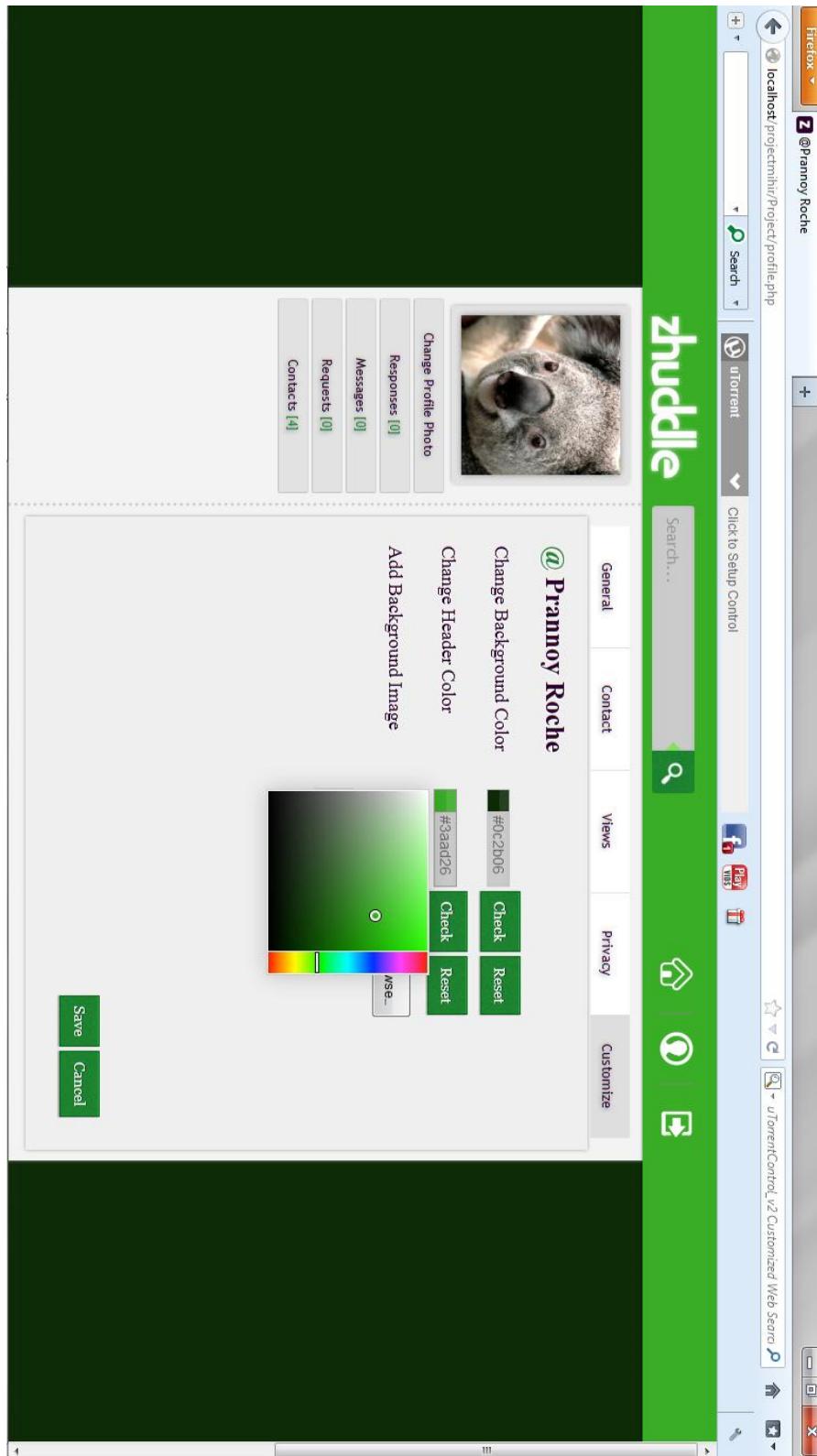
Email	prache@gmail.com
Phone Number	(022)28121876
Mobile Number	+919920429410
Website	www.facebook.com

At the bottom left of the content area are 'Save' and 'Cancel' buttons. The bottom right corner of the window has a close button.

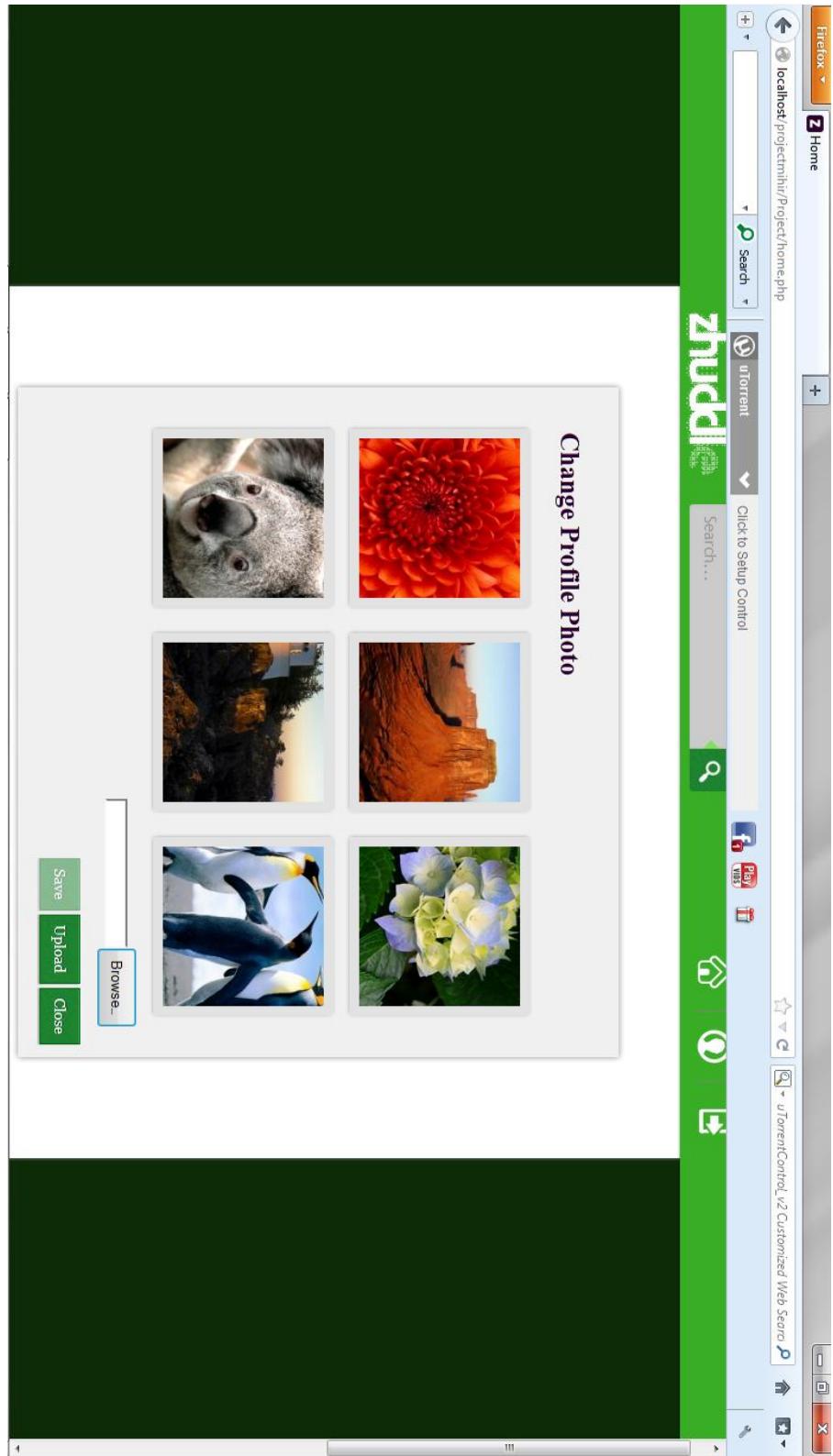
Profile Page Editing (Contact Information)



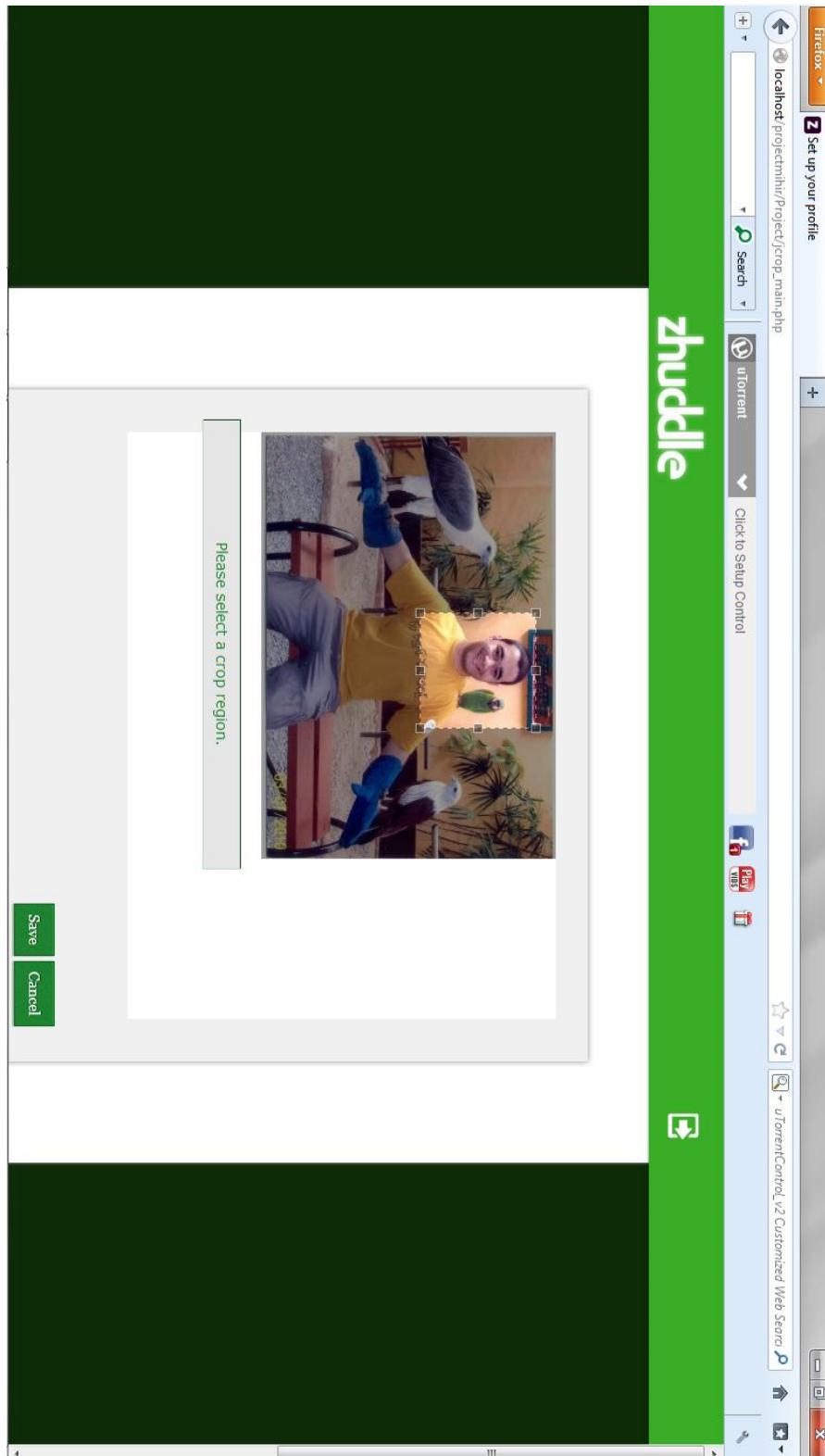
User Privacy Option Settings.



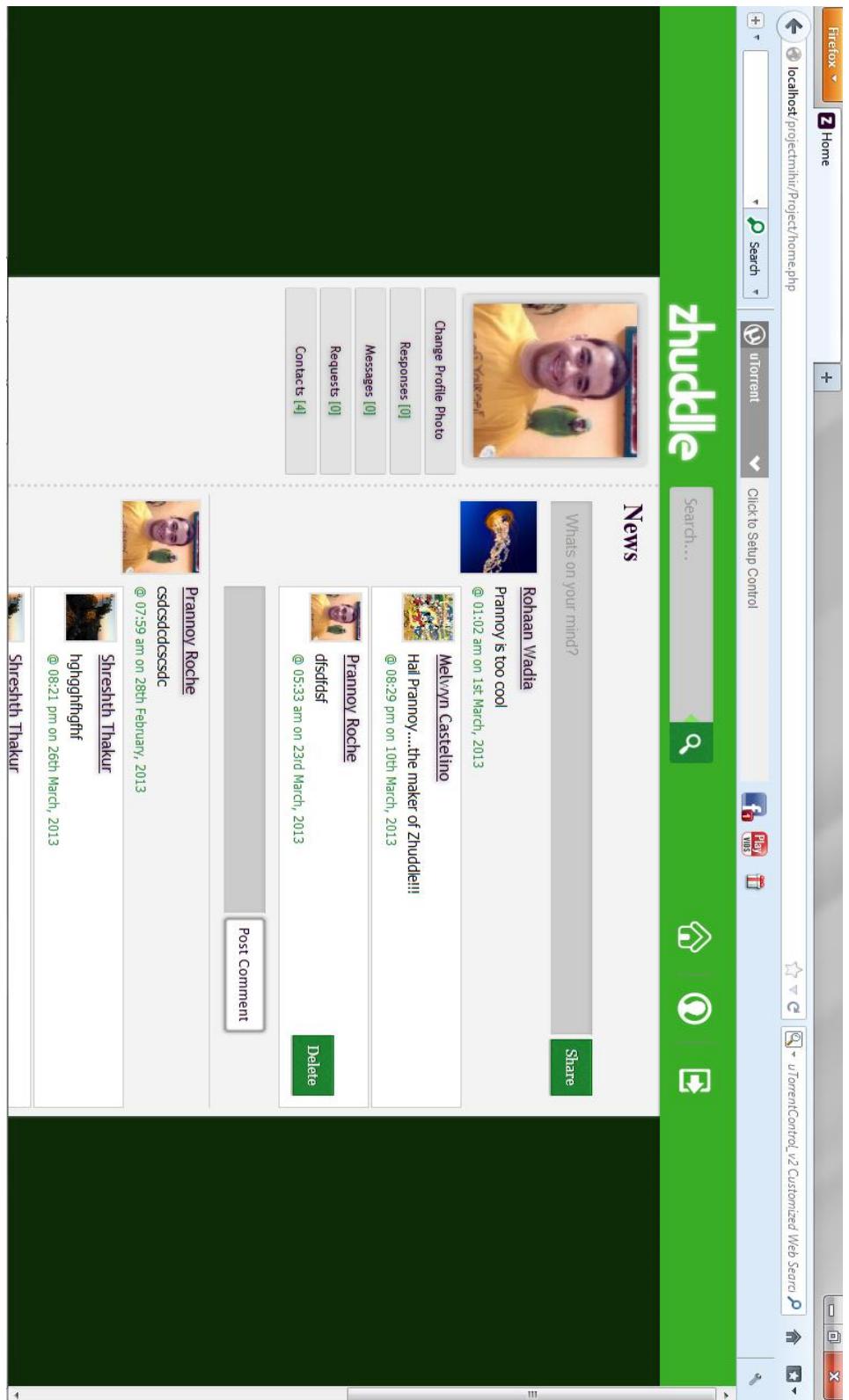
User Profile Customization Options.



Upload/Change Profile Picture.



Profile Picture Cropping.



The result after cropping the image.

Firefox ▾

localhost/projectmihir/Project/home.php

Search uTorrent

Click to Setup Control

zhuddle

Search...

News

What's on your mind?

Share

Rehaan Wadia
Prannoy is too cool
④ 01:02 am on 1st March, 2013

Melvyn Casteling
Hai Prannoy...the maker of Zhuddle!!!
④ 08:29 pm on 10th March, 2013

Prannoy Roche
Yes I am..thanks!!!
④ 10:04 am on 30th March, 2013

Mihir Bhavsar
Hey..I did more work than Prannoy
④ 10:14 am on 30th March, 2013

Prannoy Roche
ok dude..chill..you're cool too!
④ 10:16 am on 30th March, 2013

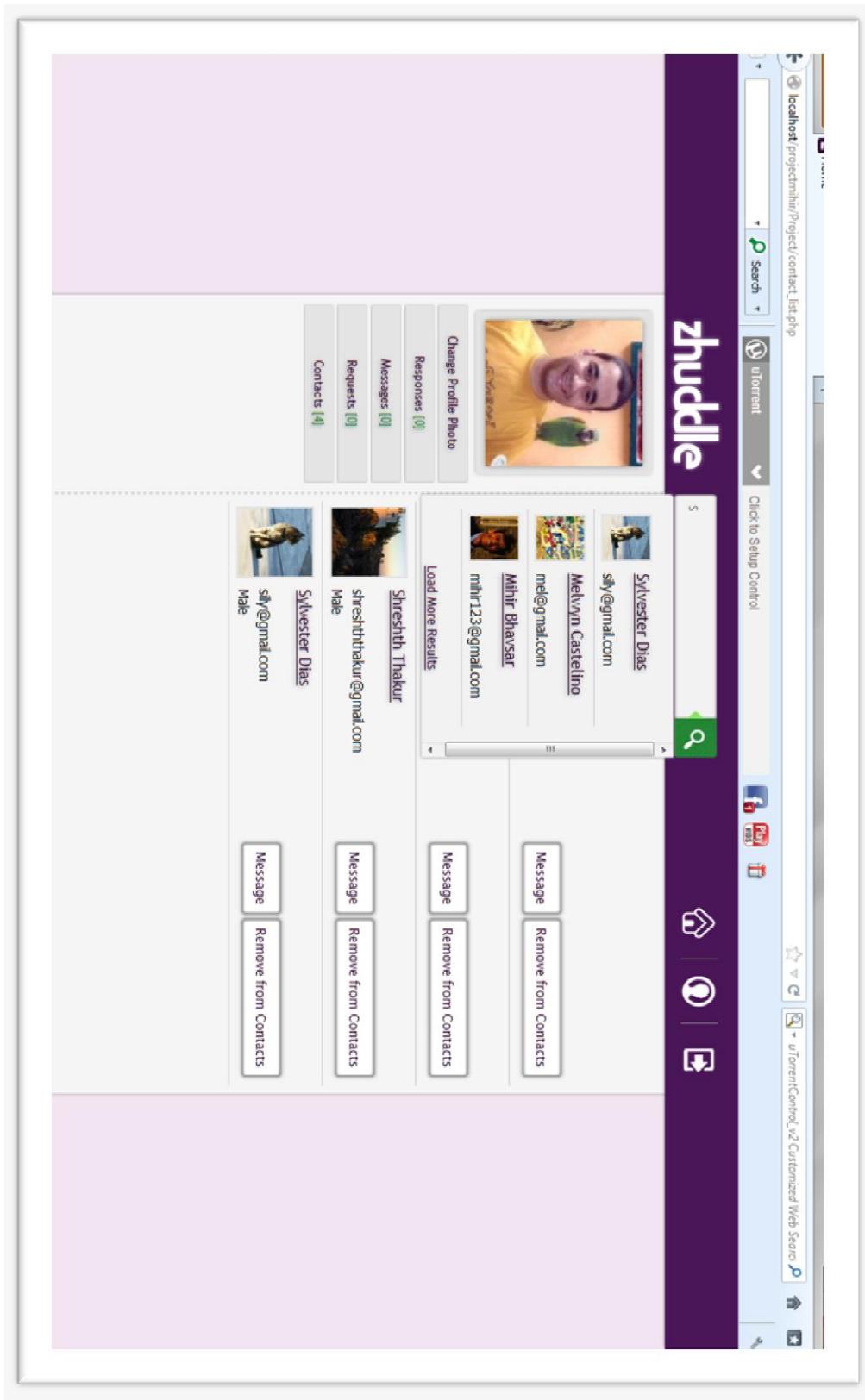
Delete

Post Comment

EN 10:18 30-03-2013

The screenshot shows the Zhuddle user home page. At the top, there's a navigation bar with links for Home, Search, and uTorrent, along with a 'Click to Setup Control' button. Below the navigation is the Zhuddle logo and a search bar. The main area is titled 'News' and displays a timeline of posts from other users. Each post includes a profile picture, the user's name, their message, the timestamp, and a 'Delete' button. The interface is clean with a purple header and a light gray background for the news feed.

User Home Page.



The Zhuddle Search Box.

Zhuddle

Search...

Found 8 Results.

Name	Email	Gender	Action
Sylvester Dias	sil@gmail.com	Male	In Contacts
Melvyn Castelino	mel@gmail.com	Male	Add to Contacts
Mihir Bhavsar	mihir123@gmail.com	Male	In Contacts
Shreshth Thakur	shreshththakur@gmail.com	Male	In Contacts
Santy Dsouza	santy@gmail.com	Male	Add to Contacts
Tom Sawyer	tom@gmail.com	Male	Add to Contacts

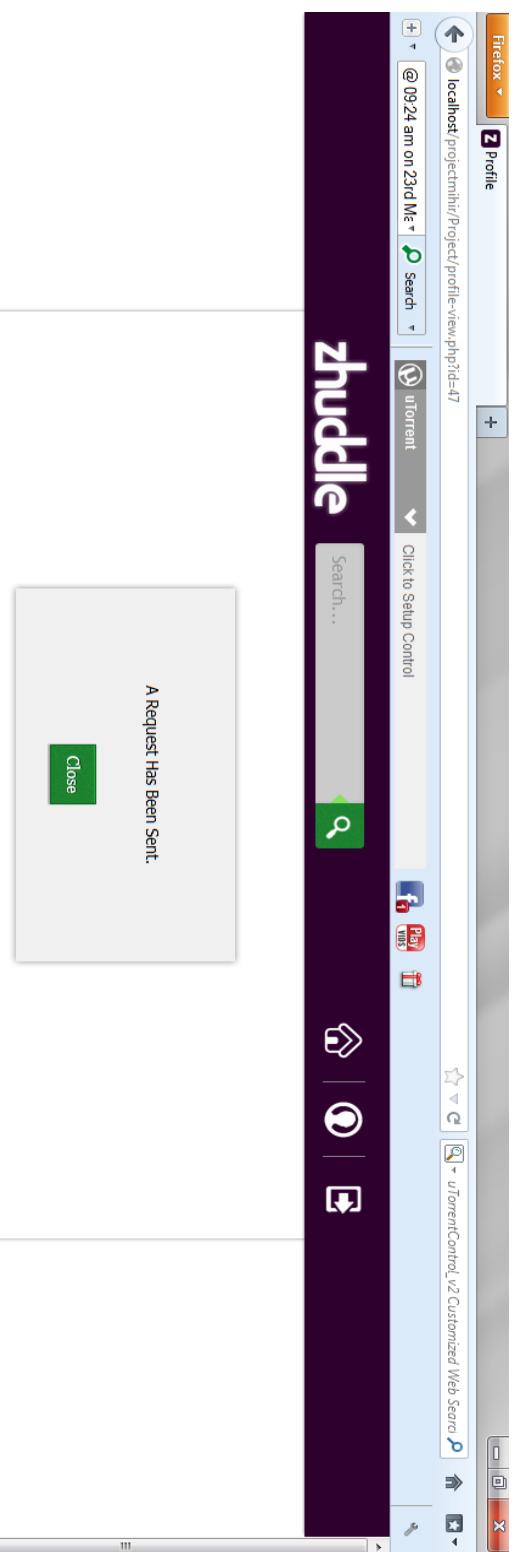
Search results.

The screenshot shows a Firefox browser window with the URL `localhost/projectmini/Project/contactList.php`. The page has a purple header with the 'zhuddle' logo and a search bar. Below the header is a section titled 'Contacts' containing four entries:

Contact	Email	Gender	Action Buttons
Gary Kirsten	gary@gmail.com	Male	Message Remove from Contacts
Mihir Bhavsar	mifir123@gmail.com	Male	Message Remove from Contacts
Shreshth Thakur	shreshththakur@gmail.com	Male	Message Remove from Contacts
Sylvester Dias	silly@gmail.com	Male	Message Remove from Contacts

At the bottom of the page, there is a note: "Click to Setup Control". The browser interface includes standard navigation buttons (Back, Forward, Stop, Refresh) and a toolbar with icons for Home, Search, and Torrent.

Contact List(Friends)



Sending a Friend Request.

The screenshot shows a Firefox browser window with the title bar "Zhuddle". The address bar displays "localhost/projectminir/Project/contact_requests.php" and the time "@ 09:24 am on 23rd Mar". The page content is a friend request interface for the user "Shreshth Thakur".

Requests

Shreshth Thakur
The above person wants to add you to his contacts.
Accept this request only if you know the person.

Sylvester Dias
The above person wants to add you to his contacts.
Accept this request only if you know the person.

Action Buttons:
Accept | Reject

Below the requests, there are navigation links: "Change Profile Photo", "Responses [0]", "Messages [0]", "Requests [2]", and "Contacts [2]".

Friend Requests.

Zhuddle

localhost:projectminiproject/inbox.php

Search

uTorrent

Click to Setup Control

Send a Message

Messages

<input type="checkbox"/>	Sent By	Subject	Time
<input type="checkbox"/>	Rohaan Wadia	fsdfsfds	@ 10:30 am on 30th March, 2013
<input type="checkbox"/>	Pramoy Roche	Hey	@ 10:26 am on 30th March, 2013
<input type="checkbox"/>	Pramoy Roche	fdffs	@ 09:24 am on 23rd March, 2013
<input type="checkbox"/>	Pramoy Roche	cdscdfscf	@ 05:56 am on 23rd March, 2013
<input type="checkbox"/>	Sylvester Dias	sdgsgsdsdsgvs	@ 10:03 pm on 28th February, 2013
<input type="checkbox"/>	Pramoy Roche	asadasda	@ 07:21 pm on 28th February, 2013
<input type="checkbox"/>	Pramoy Roche	fwtfewfewef	@ 07:19 pm on 28th February, 2013

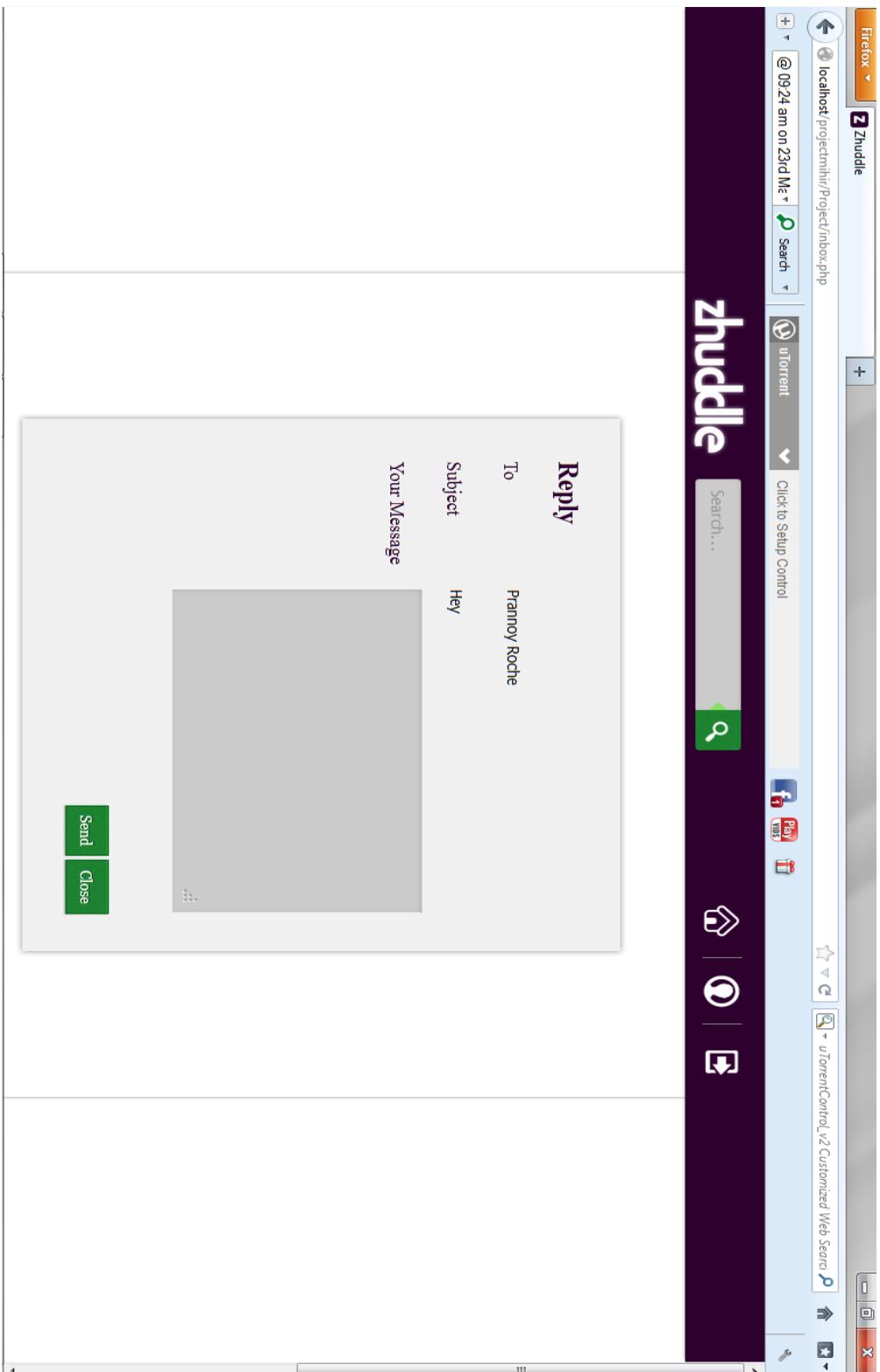
Messages(User's inbox).

The screenshot shows a Firefox browser window with the title bar "Firefox" and the tab "Zhuddle". The address bar displays the URL "localhost/projectminir/Project/inbox.php". The main content area is titled "zhuddle" and features a dark purple header with social sharing icons for Facebook, Twitter, and LinkedIn. Below the header is a search bar with placeholder text "Search...". The main content area is titled "Messages" and contains a list of messages. Each message item includes a "Delete" button, the "Sent By" field (e.g., Rohan Wadia), the "Subject" field (e.g., fsdfsd), and the "Time" field (e.g., @ 10:30 am on 30th March, 2013). The messages listed are:

- Rohan Wadia (fsdfsd) sent a message at 10:30 am on 30th March, 2013. The message content is "Hey".
- Pranay Roche (Hey) replied at 10:26 am on 30th March, 2013. The message content is "Hey wassup".
- Pranay Roche (tffff) sent a message at 09:24 am on 23rd March, 2013.
- Sylvester Dias (cdscdcsc) sent a message at 05:56 pm on 23rd March, 2013.
- Pranay Roche (asadasda) sent a message at 07:21 pm on 28th February, 2013.
- Pranay Roche (wffefwfewf) sent a message at 07:19 pm on 28th February, 2013.

At the bottom of the message list, there is a "Send a Message" button.

Opening a message.



Replies to a message.

The screenshot shows a Firefox browser window with the Zhuddle application open. The URL in the address bar is `localhost/projectminiir/project/responses.php`. The page displays a notification for a friend request from Pranoy Roche, dated February 15, 2013. The notification includes a small profile picture of Pranoy Roche and the text: "The above person has added you to your contacts." Below the notification, there is a search bar with placeholder text "Search..." and a green magnifying glass icon. To the right of the search bar are several social media sharing icons: Facebook, YouTube, Twitter, and LinkedIn. On the far right of the interface, there is a dark sidebar with the Zhuddle logo and navigation links for "Responses", "Messages", "Requests", and "Contacts". The top of the browser window shows standard Firefox controls like back, forward, and search.

Responses(Notification of accepted friend requests etc).

Codes

Mysql_connect.php

```
<?php  
  
$db_server = mysql_connect("localhost","root","");
if(!$db_server)
{
    die("Unable to connect to mysql:".mysql_error());
}
mysql_select_db("zhuddle") or die("unable to find the
database".mysql_error());  
  
?>
```

Login.php

```
<?php  
include("mysql_connect.php");  
?>  
<?php  
sleep(1);  
$db="";  
$dbpassword="";  
$dbname="";  
$id="";  
$error_message = "";  
  
$email=trim($_POST['logemail']);  
$password=trim($_POST['logpassword']);  
  
if($email && $password)  
{  
    $query='SELECT * FROM users WHERE  
        EMAIL="'.$email.'";  
    $result=mysql_query($query);  
    $numrows=mysql_num_rows($result);  
}
```

```
if($numrows!=0)
{
    while($row=mysql_fetch_assoc($result))
    {
        $db=$row['email'];
        $dbpassword=$row['password'];
        $dbname=$row['f_name'];
        $id=$row['id'];
    }

    if($email==$db && $password==$dbpassword)
    {
        $_SESSION['id']=$id;
        $_SESSION['email']=$db;
        $_SESSION['password']=$dbpassword;
        $return['error'] = false;
        $return['msg'] = "";
        echo json_encode($return);
    }
}

else
```

```
{  
    $error_message="Invalid email or password."  
}  
}  
  
else  
{  
    $error_message="Invalid email or    password."  
}  
}  
  
if (!empty($error_message))  
{  
    $return['error'] = true;  
    $return['msg'] = "".$error_message;  
  
    echo json_encode($return);  
    exit();  
}  
  
if(!$result) die("Database access failed".mysql_error());  
?>
```

Home.php

```
<?php
sleep(1);
include("dbheader.php");
if(isset($_SESSION['email']))
{
    // echo"<meta http-
equiv=\"refresh\"content=\"0;url=http://localhost/Project/home.ph
p\"";
$query='SELECT * FROM users WHERE
EMAIL="'. $_SESSION['email'].'";
$result=mysql_query($query);
while($row=mysql_fetch_assoc($result))
{
    $id=$row['id'];
}
$logquery='SELECT id,login FROM log WHERE id="'.$id.'";
$logresult=mysql_query($logquery);
$ini_no=1;
if (mysql_num_rows($logresult) == 0 )
{

```

```

$logquery='INSERT into log VALUES("'.$id.'","'.$ini_no.'","")';

$logresult=mysql_query($logquery);

}

else

{

while($row=mysql_fetch_assoc($logresult))

{

$logid=$row['id'];

$login_nos=$row['login'];

}

$login_nos=$login_nos+1;

$logupdatequery='UPDATE log set login="'.$login_nos.'" where
id="'.$logid.'';

$logupdateresult=mysql_query($logupdatequery);

}

$custquery='SELECT * from customize where id="'.$id.'";

$custresult=mysql_query($custquery);

$custcount=mysql_num_rows($custresult);

while($row=mysql_fetch_assoc($custresult))

{

```

```
$bgcolor=$row['bg_color'];

$headercolor=$row['header_color'];

}

}

else

{

    header("location:Project.php");

}

$check_pic=mysql_query("SELECT * FROM users WHERE
email='".$email"'");

$get_pic_row=mysql_fetch_assoc($check_pic);

$profile_pic_db=$get_pic_row['profile_pic'];

if($profile_pic_db=="")
{
    $profile_pic="";
}

else
{
    $profile_pic=$profile_pic_db;
```

```
}

    $num_req=mysql_query("SELECT * FROM requests WHERE
id_to=$id AND status='Request-Sent'");

    $num_rows=mysql_num_rows($num_req);

    $num_res=mysql_query("SELECT * FROM responses WHERE
id_from=$id and status='Unseen'");



$num_rows_res=mysql_num_rows($num_res);

$num_messages=mysql_query("SELECT * FROM private_messages
WHERE id_to=$id and Rstatus='Unseen'");

    $num_rows_messages=mysql_num_rows($num_messages);

?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>

<title>Home</title>
```

```
<link rel="stylesheet" type="text/css"
      href="stylesheet/stylesheet1.css" />

<link rel="Shorcut Icon" href="zhuddle_icon_small.ico">

<script type="text/javascript" language="javascript"
      src="js/jquery.js"></script>

<script type="text/javascript" language="javascript"
      src="js/search_func.js"></script>

<script type="text/javascript" language="javascript" src="js/search-
empty.js"></script>

<script type="text/javascript" src="js/photoZoom.min.js"></script>

<script type="text/javascript" src="js/profile-update.js"></script>

<script type="text/javascript">

    $(document).ready(function(){

        $("#profile-pic").photoZoom();

        $(".img-div").photoZoom();

    });

</script>

<script type="text/javascript">
```

```
$(document).ready(function() {
    $("#options a,#icons a").click(function()
    {
        $('#logo-div').hide();
        $("#anime-logo-div").css("display","inline");
    });
});

$(document).ready(function(e) {
    $("#error-box1").hide();
    $("#blab-container").show();
    $("#overlay-inside").hide();
    $("#overlay-inside2").hide();
    $("#overlay").hide();
    $("#overlay2").hide();
    $(".getblabs").hide();
    var url="blab_msg_parse.php";
    $.post(url,{},function(data)
    {
        $("#blab-container").html(data);
    });
});
```

```
$(' #blabform').submit(function(){$('input[type=submit]',  
this).attr('disabled', 'disabled');});  
  
$(function() {  
  
    //  
  
    $(".blabpostbtn").click(function(e){  
        e.preventDefault();  
  
        var blabposterid = $("#blabposterid");  
  
        var blabcontent = $("#blabcontent");  
  
        var url="blab_msg_parse.php";  
  
        if (blabcontent.val()=='') {  
            e.preventDefault();  
  
            //$("#error-box1").html('Please type a Blab.').fadeIn('slow');  
  
            //  
  
        } else {  
            $.post(url,{ blabid: blabposterid.val(), blab: blabcontent.val(), },  
                  function(data) {  
  
                    document.blabform.blabcontent.value='';  
  
                    $("#blab-container").html(data);  
  
                });  
  
        } } );  
  
//
```

```
});

$(function() {
    $(".newbutton").live("click",function()
    {
        var url="comment_parse.php";
        var url1="blab_msg_parse.php";
        var blabid=$(this).attr("id");
        var comment=$("#commentcontent"+ blabid);
        if (comment.val() == "") {
        }
    else {
```

```
$.post(url,{ comment:comment.val(), blab_id:blabid },  
        function(data) {  
  
    document.getElementById("commentcontent"+ blabid).value="";  
  
      
  
    $("#comments-box"+blabid).append(data);  
    $("#comment-box"+blabid).focus();  
  
});  
}  
});  
  
$(".button_delete").live("click",function()  
{  
    var url="delete_blabs.php";  
    var url1="blab_msg_parse.php";  
    var blabid=$(this).attr("id");  
    $.post(url,{blab_to_delete:blabid},function(data){  
  
})
```

```
$.post(url1,{ },function(data){  
    $("#blab-container").html(data);  
});  
});  
  
$(".delete_comments").live("click",function()  
{  
    var url="delete_comments.php";  
    var url1="blab_msg_parse.php";  
    var commid=$(this).attr("id");  
    $.post(url,{comm_to_delete:commid},function(data){  
  
        $("#comm"+commid).slideUp();  
        $("#comment-box"+blabid).focus();  
    });});});  
  
$(window).scroll(function(){  
if ($(window).scrollTop() == $(document).height() -  
$(window).height()){  
    var time($(".results:last").attr("id"));  
    var url1="getmore_blabs.php";  
    $.post(url1,{time:time},function(data){  
        $(".results").append(data);  
    });  
}  
});
```

```
$.post(url1,{timeval:time},function(data){  
    if(data!="")  
    {  
        $(".getblabs").empty();  
        var height=$("#right-box").css("height");  
        $("#main-box").css("height","+=660");  
    }  
    else  
    {  
        $(".getblabs").hide();  
    }  
    $("#blab-container").append(data);  
});  
}  
});
```

```
</script>

</head>

<?php
if($custcount==0)
echo '<body>';
else
{
    echo '<body style="background-color:'.$bgcolor.'">';
}
?>
<!----HEADER AREA---->

<?php
if($custcount==0)
echo '<div id="header">';
else
echo '<div id="header" style="background-color:'.$headercolor.'">';
?>
<div id="header-container">
    <div id="logo-div">
        <a href="home.php"></a>
    </div>
```

```
<div id="anime-logo-div">
    <a href=""></a>
</div>

<div id="search-box">
    <form class="form-wrapper cf"
action="view_contact.php" method="get">
        <input type="text" id="search-textbox"
class="text-box" placeholder="Search..." name="x"
onkeyup="autoSuggest()" autocomplete="off" />
        <button type="submit" id="search-
button"></button>
        <input type="hidden" id="id-textbox"
value="<?php echo $id?>"/>
    </form>
    <div id="autosuggest-container" >
        </div>
    </div>
<div id="icons">
    <a href="home.php" class="icon">
        
    </a>
    <a href="profile.php" class="icon">
```

```



        </a>

        <a href="logout.php" class="icon">

        </a>

    </div>

</div>

<?php

    $contact_list=$get_pic_row['contact_list'];

    $contacts=explode(",",$contact_list);

    $contacts_num=count($contacts);

    for($i=0;$i<$contacts_num-1;$i++)

    {

        $new[$i]=$contacts[$i];

    }

    $new_rows=count($new);

?>

<!----MAIN AREA---->

<div id="main">

```

```
<section>
    <div id="main-box">
        <div id="left-box">
            <div id="profile-pic">
                <div>
                    <?php
                        if($profile_pic=="")
                            echo "<img src=images/blank.jpg
width=150 height=150/>";
                        else
                            echo "<img src=$profile_pic
width=150 height=150/>";
                    ?>
                </div>
                </div>
            <div id="options">
                <?php
                    if($profile_pic=="")
                    {
                        echo "<a href="" class=tabbox id=add-profile-
pic>Add a Profile Photo</a>";
                    }
                
```

```
else
    echo "<a href=" class=tabbox id=change-
profile-pic>Change Profile Photo</a>";
    ?>
    <a href="responses.php"
class="tabbox">Responses <span class="nos">[<?php echo
$num_rows_res?>]</span></a>
    <a href="inbox.php" class="tabbox">Messages <span
class="nos">[<?php echo $num_rows_messages?>]</span></a>
        <a href="contact_requests.php"
class="tabbox">Requests <span class="nos">[<?php echo
$num_rows?>]</span></a>
        <a href="contact_list.php"
class="tabbox">Contacts <span class="nos">[<?php echo
$new_rows?>]</span></a>
    </div>
</div>
<div id="overlay">
    <div id="overlay-inside">
        <p id="reply-label" class="labels">Add a
Profile Photo</p>
```

```
<form enctype="multipart/form-data" method="post"
action="image_upload_1.php">

    <div id="orig-img">

        </div>

        <input name="uploaded_file" type="file"
id="image" class="filebutton" />

    </div>

    <input type="submit" class="button" id="upload" value="Upload"/>

    <input type="button" class="button"
id="close" value="Close"/>

    <input type="hidden" name="question"
value="3">

</div>

<div id="cropped-img">

    </div>

</form>

</div>

</div>

<div id="overlay2">

    <div id="overlay-inside2">

        <p id="reply-label" class="labels">Change Profile
Photo</p>

    </div>

</div>
```

```
<?php  
$files = glob("uploads/$id/*.*");  
for ($i=0; $i<count($files); $i++)  
{  
    $num = $files[$i];  
    echo '<div class="img-div"><a href=""  
id="'.$i.'" class="img-link"></a></div>';  
}  
?>  
  
<form enctype="multipart/form-data" method="post"  
action="image_upload_1.php">  
    <input name="uploaded_file"  
type="file" id="image" class="filebutton" />  
    <input type="submit" class="button"  
id="save-change-pic" value="Save" disabled="disabled"/>  
    <input type="submit" class="button"  
id="upload-button" value="Upload" />  
    <input type="button" class="button"  
id="close2" value="Close"/>  
    <input type="hidden" name="question"  
value="3" />
```

```
<div id="cropped-img">  
    </div>  
    </form>  
    </div>  
    </div>  
  
<div id="right-box">  
    <p class="labels" id="tags">News</p>  
    <div id="blabs">  
  
<form action="" enctype="multipart/form-data" method="POST"  
id="blabform" name="blabform">  
  
    <input type="text" name="blab-content" class="text-box"  
id="blabcontent" placeholder="Whats on your mind?"  
autocomplete="off"/>  
  
    <button type="submit" value="Share" class="blabpostbtn"  
name="blabpost" id="blabsubmitbtn">Share</button>  
  
    <input type="hidden" name="blabposterid" id="blabposterid"  
value=<?php echo $id;?>/>  
    </form>  
    </div>  
</section>  
</div>  
</body>  
</html>
```

Blab msg parse.php

```
<?php include("dbheader.php");

$blabberid="";
$blabberDisplayList="";
$display_comments="";
$id="";
$blabid="";
$comm_date="";
$timestamp="";
if($_POST)

{
$blabberid=$_POST['blabid'];

$prevent_dp = mysql_query("SELECT mem_id FROM blabs WHERE
mem_id=".mysql_real_escape_string($blabberid)." AND blab_date
between subtime(now(),'00000-00-00 00:00:20') and now()");

$nr = mysql_num_rows($prevent_dp);

if ($nr > 0){

    echo 'Please wait 20 seconds between your blab.';

    exit();
}
```

```
// Process the blab once it has been sent

if (isset($_POST['blab'])) {

    // Escape and prepare our variables for insertion into the database

    $time=time();

    $blab = htmlspecialchars($_POST['blab']); // Convert html tags and
    such to html entities which are safer to store and display

    $blab = mysql_real_escape_string($blab); // Just in case anything
    malicious is not converted, we escape those characters here

    $blab = mysql_real_escape_string($blab); // Just in case anything
    malicious is not converted, we escape those characters here

    ...

    $sql = "INSERT INTO blabs
(mem_id,blab_content,blab_date,timestamp) VALUES
('$blabberid','$blab',now(),$time)";

    if (!mysql_query($sql)) {

        echo 'Could not post blab! An insertion query error has
occured.';

    }

    else

    {

    }

}

}
```

```
$sql_blabs = mysql_query("SELECT blab_id, mem_id, blab_content,
blab_date,timestamp FROM blabs ORDER BY timestamp DESC LIMIT
4");

$blabberDisplayList = "";
$blabber_pic="";
$comments_list="";
$comm_id="";
$older="";

while($row = mysql_fetch_array($sql_blabs)){
$blabid = $row["blab_id"];
$older=$row["timestamp"];
$uid = $row["mem_id"];
$the_blab = $row["blab_content"];
$blab_date = $row["blab_date"];
$blab_d=date_create($blab_date);
$blab_date = date_format($blab_d, 'h:i a \o\n jS F, Y');
$timestamp=$row['timestamp'];
// Inner sql query
```

```

$sql_mem_data = mysql_query("SELECT * FROM users WHERE
id='$uid' LIMIT 1");

while($row = mysql_fetch_array($sql_mem_data)){
    $uid = $row["id"];
    $ufirstname = $row["f_name"];
    $ulastname = $row["l_name"];
    $ufirstname = substr($ufirstname, 0, 10);
    ////////////// Mechanism to Display Pic. See if they
have uploaded a pic or not ///////////////////////////////
    $blabber_pic = $row['profile_pic'];

    $sql_comments=mysql_query("SELECT * FROM
comments WHERE id_blab=$blabid ORDER BY comm_date ASC ");

    $count=mysql_query("SELECT COUNT(*) FROM
comments WHERE id_blab=$blabid ORDER BY comm_date");

    while($row1=mysql_fetch_array($sql_comments))

    {
        $comments_list=$row1["comm_content"];
        $commentator=$row1["sender_id"];
        $comm_id=$row1["comm_id"];
        $comm_date=$row1["comm_date"];
        $comm_date_d=date_create($comm_date);
        $comm_date=date_format($comm_date_d, 'h:i
a \o\n jS F, Y');
    }
}

```

```

$retrieve_commentator=mysql_query("SELECT * FROM users WHERE
id=$commentator");

while($results=mysql_fetch_assoc($retrieve_commentator))

{

$commentator_pic=$results["profile_pic"];

$f_name=$results["f_name"];

$l_name=$results["l_name"];



if($commentator==$_SESSION['id'])

{



$comments_list_formatted=stripslashes(wordwrap(nl2br($com
ments_list), 65, "<br>", true));

$display_comments.='

95


```

```
else
{
    $comments_list_formatted=stripslashes(wordwrap(nl2br($comments
_list), 65, "<br>", true));
    $display_comments.='<div class="comments-
class"><ul class="link-list search-list"><li><a href="profile-
view.php?id='.$commentator.'"><img class="search-pic"
src='.$commentator_pic.' />'."\$f_name.' '$l_name.'</a><p
class="blabs-p">'."\$comments_list_formatted.'<br /><span
class="time-span-class">@"
'."\$comm_date.'</span></p></li></ul></div>';
}

}
if($_SESSION['id']==$uid)
{
    $the_blab_formatted=stripslashes(wordwrap(nl2br($the_blab),
56, "<br>", true));
    $blabberDisplayList .='<div id="'.$timestamp.'"
class="results"><form id="blab_post_'"."\$uid.'" method="POST"><ul
```

```

class="link-list search-list">><li><a href="profile.php"><img
class="search-pic1" src='$.blabber_pic.' height="35"
width="35"/>'$.firstname.'  '$lastname.'</a><p class="blabs-
p">'$.the_blab_formatted.<br /><span class="time-span-class">@
'$.blab_date.'</span></p><br />

<div id="comments-box'$.blabid.'">
'$.display_comments.'

</div><form method="POST" action=""><input type="text"
id="commentcontent'$.blabid.'" class="text-box"/> <input
type="button" id="'.$.blabid.'" class="newbutton" value="Post
Comment" /> <button type="button" class="button_delete"
id="'.$.blabid.'" value="Delete">Delete</button><input type="hidden"
name="blabid" id="blabid" value="'.$.blabid.'" /></form>

</form></div></li></ul>';

$display_comments="";
}

else
{
    $the_blab_formatted=stripslashes(wordwrap(nl2br($the_blab),
56, "<br>", true));

        $blabberDisplayList .= '<div id="'.$.timestamp.'"'
class="results"><form id="blab_post_"'$.uid.'" method="POST"><ul

```

```
class="link-list search-list"><li><a href="profile-
view.php?id='.$uid.'"><img class="search-pic1" src='.$blabber_pic.'
height="35" width="35"/>'.'$firstname.' '.'$lastname.'</a><p
class="blabs-p">'.'$the_blab_formatted.'<br /><span class="time-
span-class">@ '.$blab_date.'</span></p><br />

<div id="comments-box'.'.$blabid.'">
'$display_comments.'

</div><form method="POST" action=""><input type="text"
id="commentcontent'.'.$blabid.'" class="text-box"/> <input
type="button" class="newbutton" id="'.$blabid.'" value="Post
Comment"/><input type="hidden" name="blabid" id="'.$blabid.'"
value="'.$blabid.'"></form>
```

```
</form></div></li></ul>';

}

}

}
```

```
$blabberDisplayList.='
<br/><br/><div class="getblabs"
id="'.$blabid.'"><br/><br/></div>';

echo $blabberDisplayList;
//to display the blabs

?>
```

Private_msg_parse.php

```
<?php

include("loggedincheck.php");

// PREVENT DOUBLE POSTS
///////////
///////////

if($_POST)

{
$checkuserid="";

$checkuserid=$_POST['senderid'];

$prevent_dp = mysql_query("SELECT id FROM private_messages
WHERE id_from=".mysql_real_escape_string($checkuserid)." AND
time_sent between subtime(now(),'0:0:20') and now()");

$nr = mysql_num_rows($prevent_dp);

if ($nr > 0){

    echo 'Please wait for 20 seconds.';

    exit();

}

///////////
///////////

// PREVENT MORE THAN 30 IN ONE DAY FROM THIS MEMBER
///////////
///////////
```

```
$sql = mysql_query("SELECT id FROM private_messages WHERE  
id_from='$checkuserid' AND DATE(time_sent) = DATE(NOW()) LIMIT  
40");  
  
$numRows = mysql_num_rows($sql);  
  
if ($numRows > 30) {  
  
    echo 'You can only send 30 Private Messages per day.';  
  
    exit();  
  
}  
  
// Process the message once it has been sent  
  
if (isset($_POST['message'])) {  
  
    // Escape and prepare our variables for insertion into the database  
  
    $to      = $_POST['recID'];  
  
    $from   = $_POST['senderid'];  
  
    $pageid = $_POST['pageID'];  
  
    if($pageid==1)  
  
    {  
  
        $messageid = $_POST['messID'];  
  
    }  
  
    else if($pageid==2)  
  
    {  
  
        $messageid=0;  
  
    }  
  
}
```

```

// $toName      = ($_POST['rcpntName']);
// $fromName = ($_POST['senderName']);

$sub = htmlspecialchars($_POST['subject']); // Convert html tags
and such to html entities which are safer to store and display

$msg = htmlspecialchars($_POST['message']); // Convert html tags
and such to html entities which are safer to store and display

$sub = mysql_real_escape_string($sub); // Just in case anything
malicious is not converted, we escape those characters here

$msg = mysql_real_escape_string($msg); // Just in case anything
malicious is not converted, we escape those characters here

// Handle all pm form specific error checking here

if (empty($to) || empty($from) || empty($sub) || empty($msg)) {

    echo 'Missing Data to continue';

    exit();

} else {

    // Delete the message residing at the tail end of their list
    so they cannot archive more than 100 PMs ----

    $sqldeleteTail = mysql_query("SELECT * FROM
private_messages WHERE id_to='$to' ORDER BY time_sent DESC
LIMIT 0,100");

    $dci = 1;
}

```

```

while($row = mysql_fetch_array($sqldeleteTail)){
    $pm_id = $row["id"];
    if ($dci > 99) {
        $deleteTail = mysql_query("DELETE
FROM private_msg WHERE id='$pm_id'");
    }
    $dci++;
}

// End delete any comments past 100 off of the tail end ---
-----
// INSERT the data into your table now
if($pageid==1)
{
    $sql = "INSERT INTO private_messages (id,id_to,id_from,
time_sent,subject,message,Sstatus,Rstatus) VALUES
('$messageid','$to', '$from', now(), '$sub', '$msg', 'Seen', 'Unseen')";

    if (!mysql_query($sql)) {
        echo 'Could not send message! An insertion query
error has occurred.';

        exit();
    }
}
else {

```

```
echo 'Message sent successfully';

        exit();

    } // close else after the sql DB INSERT check

}

else

{

    $selmaxid=mysql_query("SELECT id from private_messages
ORDER BY id DESC LIMIT 1");

    $numros=mysql_num_rows($selmaxid);

    if($numros=0)

    {

        $maxidadd=1;

    }

    else

    {

        while($insertmess=mysql_fetch_array($selmaxid))

        {

            $maxidadd=$insertmess['id'];

            $maxidadd=$maxidadd+1;

        }

    }

}
```

```

$sql1 = "INSERT INTO private_messages (id,id_to,id_from, time_sent,
subject,message,Sstatus,Rstatus) VALUES ('$maxidadd','$to', '$from',
now(), '$sub', '$msg', 'Seen', 'Unseen')";

    if (!mysql_query($sql1)) {

        echo 'Could not send message! An insertion query
error has occured.';

        exit();

    }

}

echo 'Message sent successfully';

exit();

} // close else after the sql DB INSERT check

}

}// Close if (empty($sub) || empty($msg)) {

}

}

else

{

    echo'error occurred';

}

// Close if (isset($_POST['message'])) {

?>

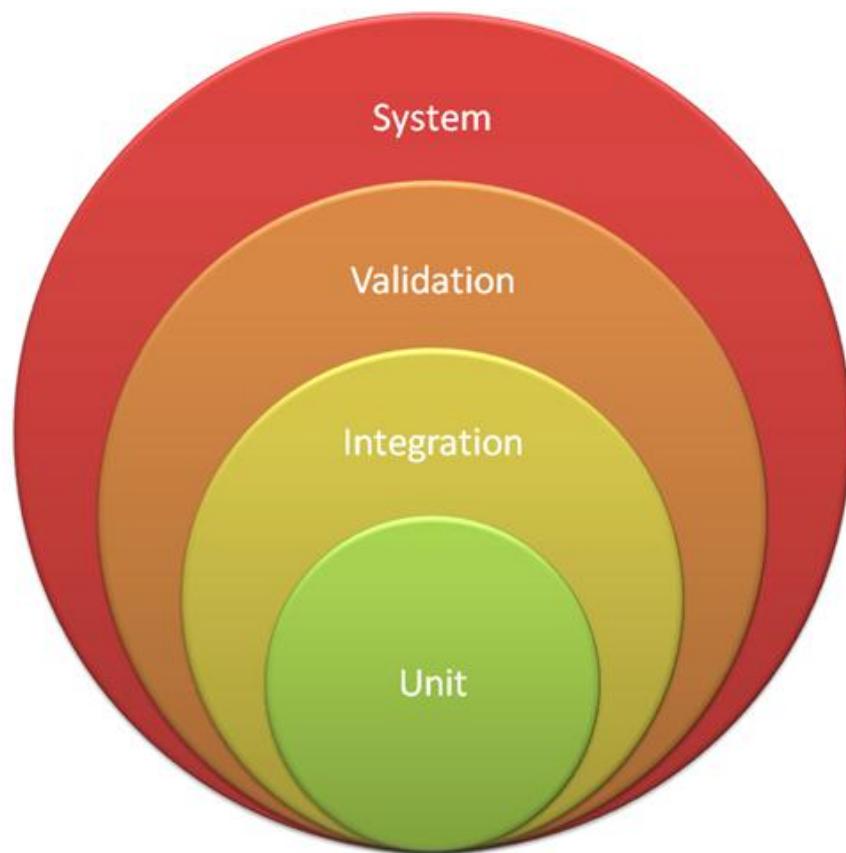
```

Software Testing

Testing is a set of activities that can be planned in advance and conducted systematically. For this reason a template for software testing—a set of steps into which you can place specific test case design techniques and testing methods—should be defined for the software process.

Verification and Validation

Software testing is one element of a broader topic that is often referred to as verification and validation (V&V). Verification refers to the set of tasks that ensure that software correctly implements a specific function. Validation refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.



Software Testing Process

Unit testing

It begins at the vortex of the spiral and concentrates on each unit (e.g., component, class, or WebApp content object) of the software as implemented in source code.

Integration testing

The focus is on design and the construction of the software architecture. It addresses the issues associated with the dual problems of verification and program construction. Test case design techniques that focus on inputs and outputs are more prevalent during integration, although techniques that exercise specific program paths may be used to ensure coverage of major control paths. After the software has been integrated (constructed), a set of *high-order tests* is conducted. Validation criteria (established during requirements analysis) must be evaluated.

Validation testing

Provides final assurance that software meets all informational, functional, behavioural, and performance requirements. The last high-order testing step falls outside the boundary of software engineering and into the broader context of computer system engineering. Software, once validated, must be combined with other system elements (e.g., hardware, people, databases).

System testing verifies that all elements mesh properly and that overall system function/performance is achieved

Software Testing Strategies

White-box testing, sometimes called *glass-box testing*, is a test-case design philosophy that uses the control structure described as part of component-level design to derive test cases. Using white-box testing methods, you can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bound.
- exercise internal data structures to ensure their validity.

Basis Path Testing(White-box Testing)

The basis path method enables the test-case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set of execution paths.

• Flow Graph Notation

Before we consider the basis path method, a simple notation for the representation of control flow, called a *flow graph* (or *program graph*) must be introduced. The flow graph depicts logical control flow using the notation.

- **Independent path**

It is any path through the program that introduces at least one new set of processing statements or a new condition. When stated in terms of a flow graph, an independent path must move along at least one edge that has not been traversed before the path is defined.

- **Cyclomatic complexity**

It is a software metric that provides a quantitative measure of the logical complexity of a program. When used in the context of the basis path testing method, the value computed for cyclomatic complexity defines the number of independent paths in the basis set of a program and provides you with an upper bound for the number of tests that must be conducted to ensure that all statements have been executed at least once.

❖ The number of regions of the flow graph corresponds to the **cyclomatic complexity**.

❖ Cyclomatic complexity $V(G)$ for a flow graph G is defined as
 $V(G) = E - N + 2$

where E is the number of flow graph edges and N is the number of flow graph nodes.

❖ Cyclomatic complexity $V(G)$ for a flow graph G is also defined as
 $V(G) = P + 1$ where P is the number of predicate nodes contained in the flow graph G .

Black Box Testing

Black-box testing, also called behavioral testing, focuses on the functional requirements of the software. That is, black-box testing techniques enable you to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black-box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than whitebox methods.

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

- Incorrect or missing functions.
- Interface errors.
- errors in data structures or external
- Database access.
- Behaviour or performance errors
- Initialization and termination errors.

Types of Black-Box Testing Includes:

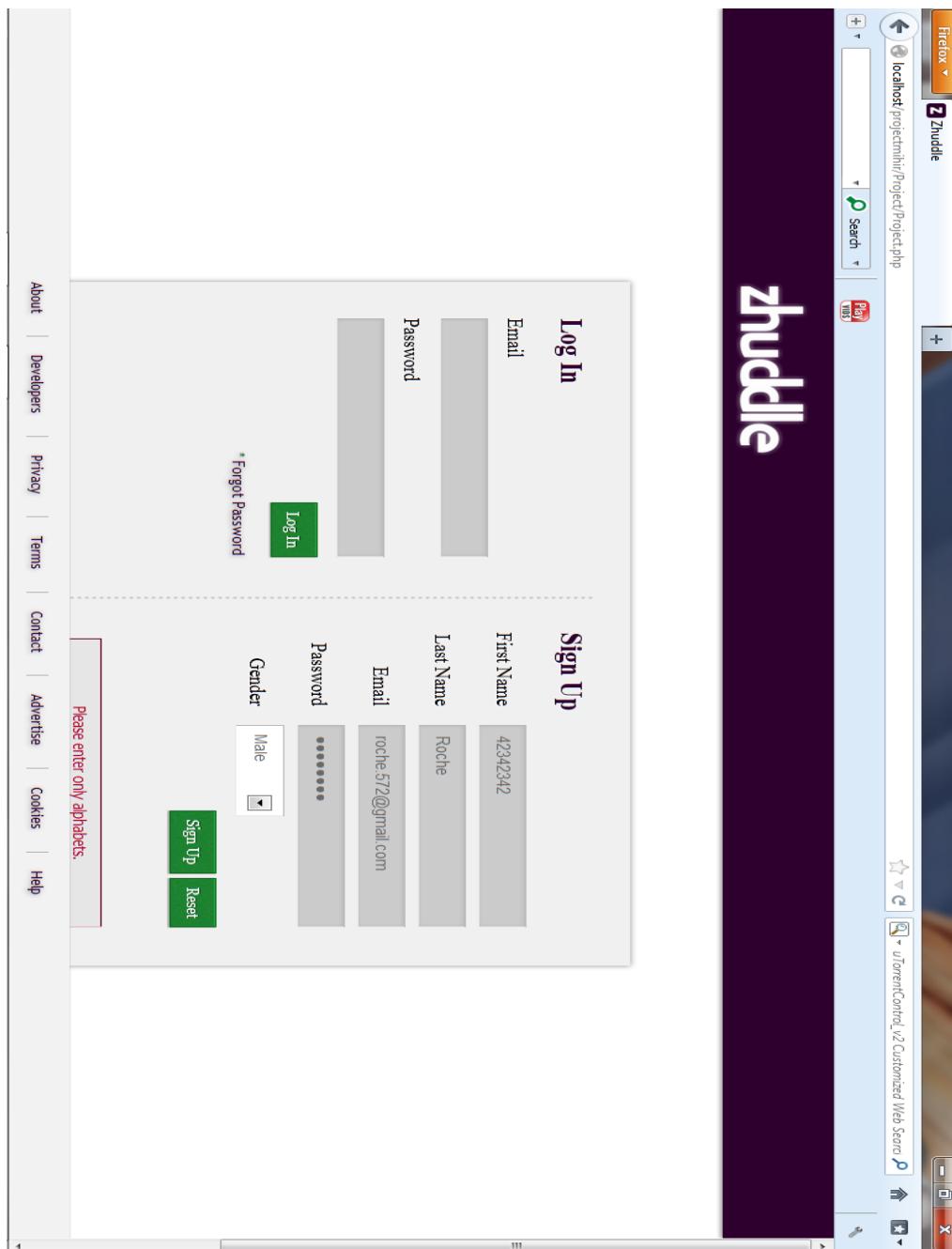
Equivalence partitioning

It is a black-box testing method that divides the input domain of a program into classes of data from which test cases can be derived. An ideal test case single-handedly uncovers a class of errors (e.g., incorrect processing of all character data) that might otherwise require many test cases to be executed before the general error is observed.

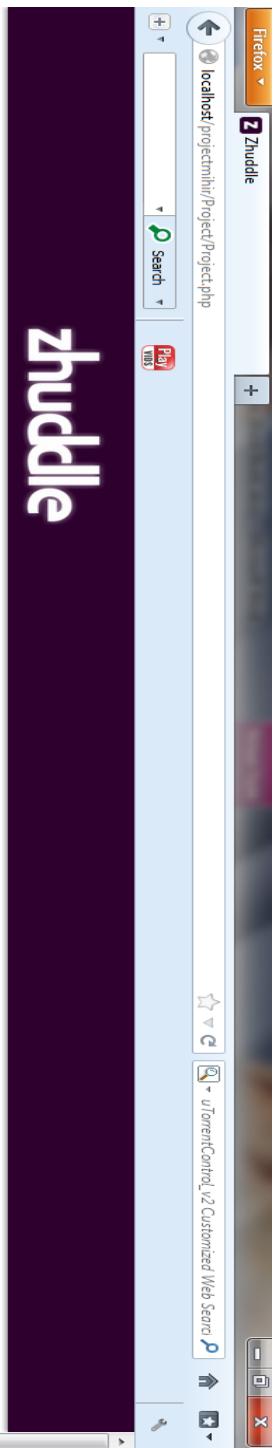
Boundary Value Analysis

A greater number of errors occurs at the boundaries of the input domain rather than in the “center.” It is for this reason that boundary value analysis (BVA) has been developed as a testing technique. Boundary value analysis leads to a selection of test cases that exercise bounding values. Boundary value analysis is a test-case design technique that complements equivalence partitioning. Rather than selecting any element of an equivalence class, BVA leads to the selection of test cases at the “edges” of the class.

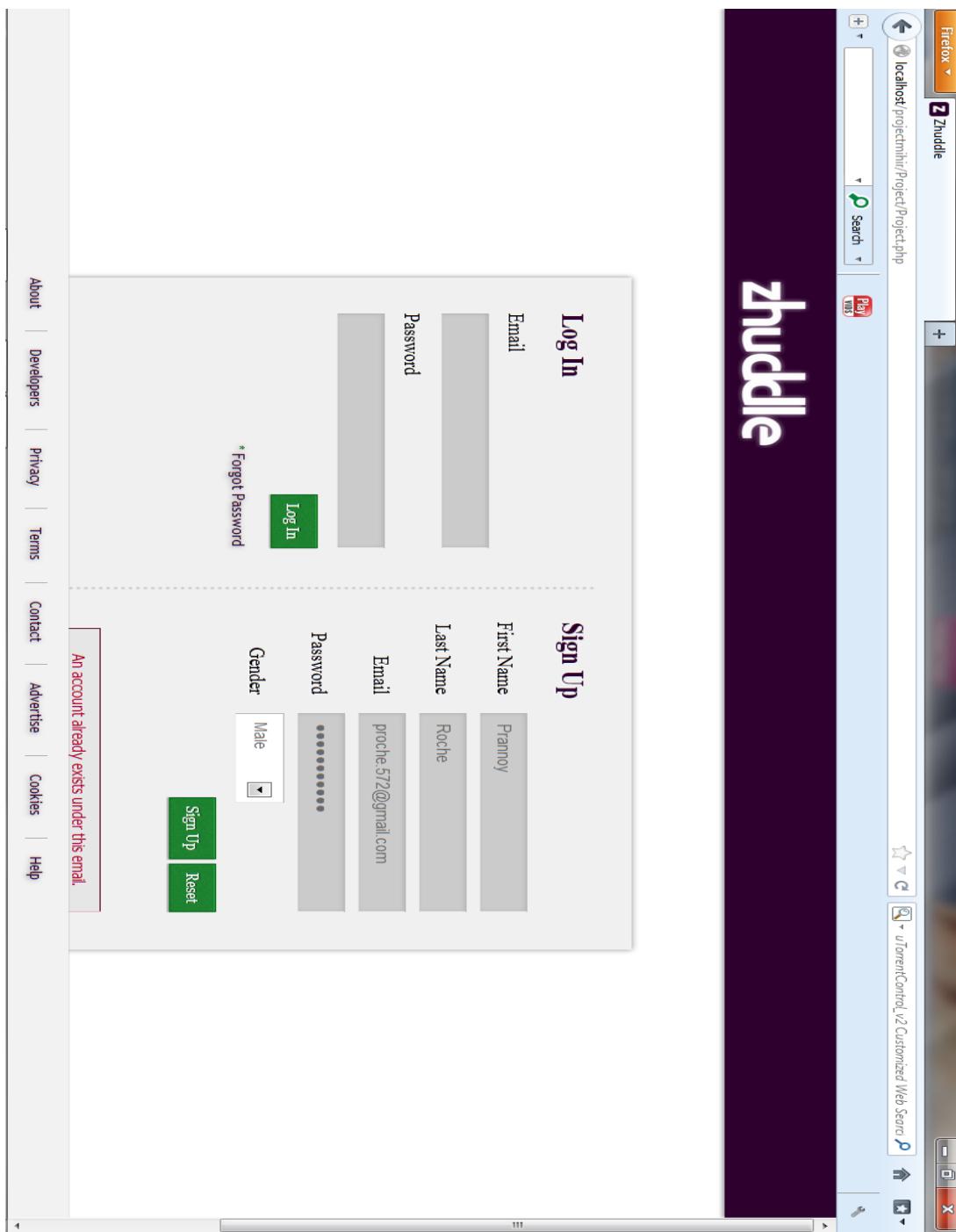
Validation Checks



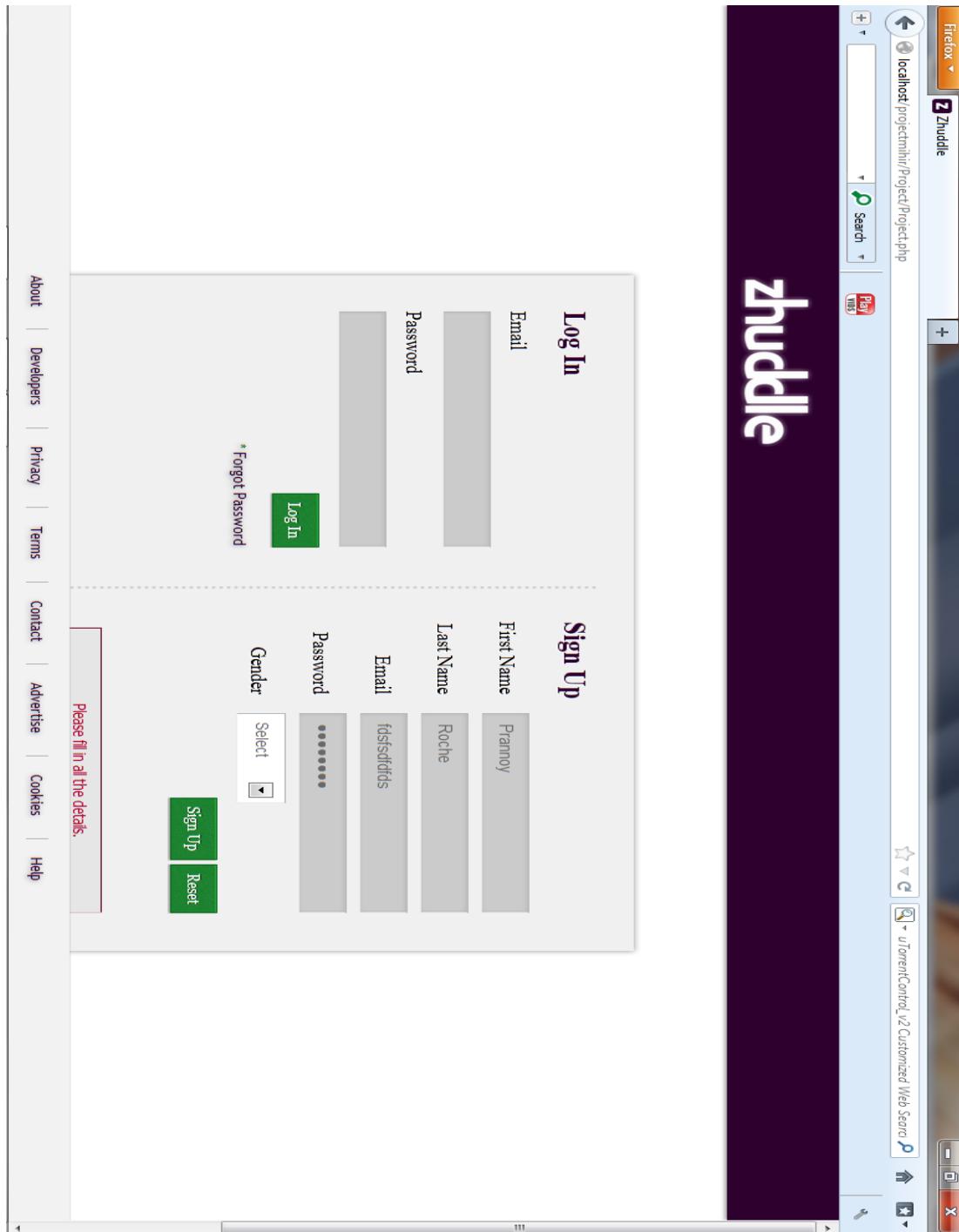
First Name and Last Name Validation



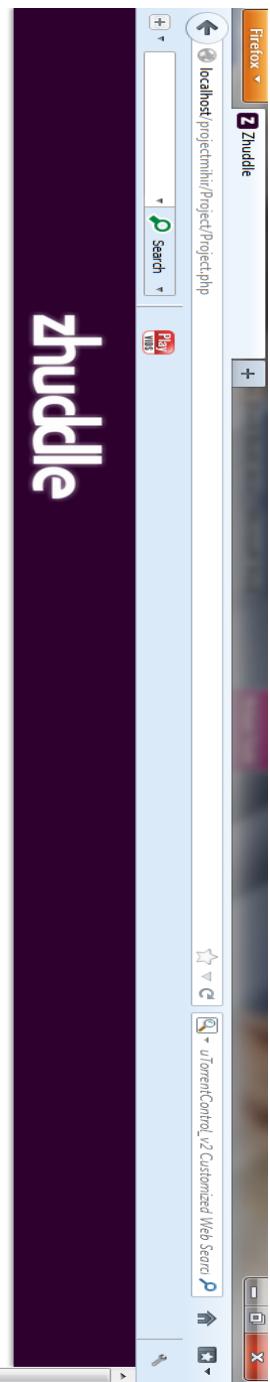
Email Validation



Check Availability of Email



Registration form validation to check if all fields are filled

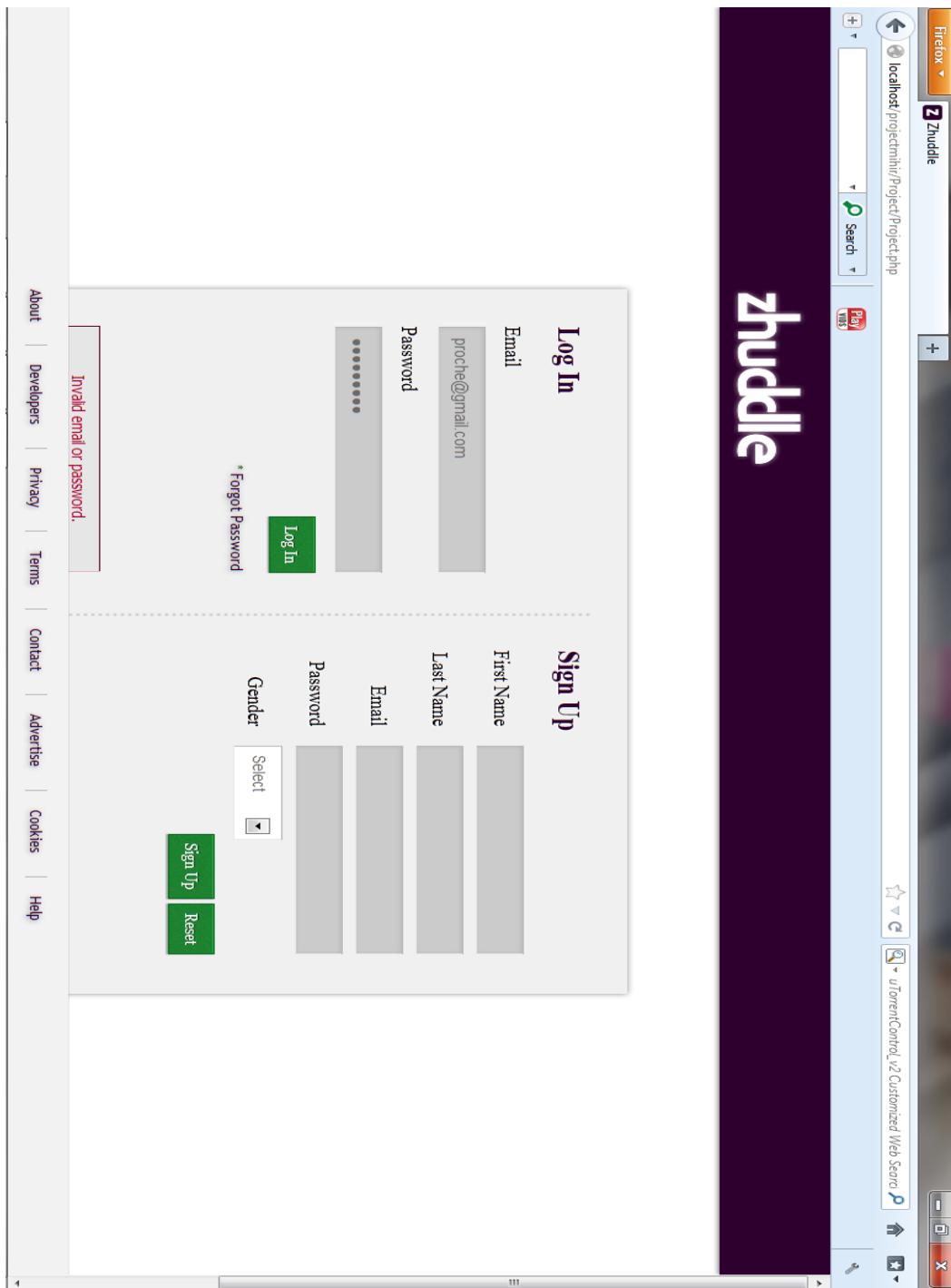


Password Validation

116



Login validation to check if all fields have been entered



Authentication during login attempt

Future Enhancements

‘Zhuddle’ was developed within a span of 4 months. It was a project of high complexity and due to the time constraint involved we couldn’t incorporate all the features of a social networking website thus leaving a huge scope for future enrichment. Some of the enhancements that can be easily implemented are

Chat Feature

Users can have a private chat box and can chat with anyone they want within their friends list.

Friend Suggestion

An algorithm can be implemented which suggests potential friends to a user based on the number of mutual friends, similarity of interests and institutions attended (schools and universities), location and other such parameters.

Better Image Management

Users can organize pictures and create albums for better image management.

Uploading and sharing files

Users can upload and share videos, presentations and other kinds of media.

Bibliography

Websites visited

<http://www.w3schools.com/>

<http://www.9lessons.info/>

<http://www.wikipedia.com/>

<http://www.webintersect.com/>

<http://www.easyphp.com/>

Books referred

Software engineering -A practitioner's approach by Roger S Pressman

