

Team 10

Mippsy

Aaron Ho

Kaijun He

Mark Yang

Kean Jye Tan

Kaiyi Chen

Table of Contents

<u>Purpose</u>	3
<u>Design Outline</u>	5
<u>Design Issues</u>	6
<u>Design Details</u>	9

Purpose

People often rely on social media to spread awareness regarding cases of missing persons and pets. However, this form of communication is inefficient in spreading the news to the larger audience of the community. Our team has designed a centralized web app that will handle the reporting of both missing persons and pets cases, and allow the community to play an active role in the search.

This design document will showcase our functional requirements:

1. Users can create a post regarding their missing person or pet that will be visible to the public.
 - a. The post will be interpreted as a thread other users can look up and follow.
 - b. The post will display information crucial to helping the search.
2. Users will be able to look up posts by specific criterias.
 - a. Users will be able to search by name.
 - b. Users will be able to search by date.
 - c. Users will be able to search by geographic region.
 - d. Users will be able to search by species.
 - e. Users will be able to search by age.
 - f. Users will be able to search by gender.
3. Users will have the option to subscribe to a post.
 - a. Users will receive timely notifications and updates regarding posts they have followed.
4. Users will be able to interact with other users.
 - a. Users will be able to communicate with each other via private messaging.
 - b. Users will be able to comment on other users' posts.
 - c. Users will be able to upvote or downvote comments made by other users.
5. The user interface will be a smooth and simplistic web page
 - a. The web page will feature straightforward toggles and menus for users to use with ease.
 - b. The web page will display an attractive layout.
 - c. The web page will display sufficient information for users.
6. User will be able to view posts from their area.

- a. Users will receive feeds on posts based on their location their provide.
- b. Users can also subscribe to a certain location to receive notifications on that area.

7. Users will be able to easily share their reports to the community.

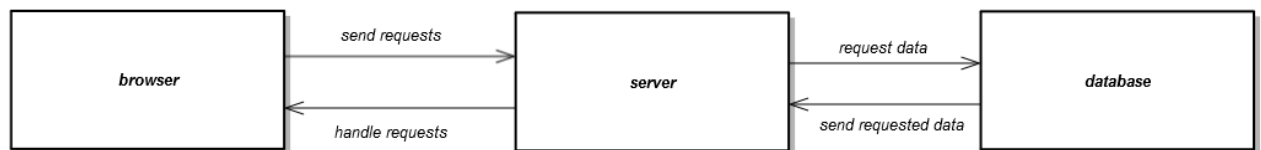
- a. Users will be able to share their posts to social medias such as Facebook.

Design Outline

Overview and Components:

This project will be mainly a web application that allows users to view and comment on posts created by other users or to create one themselves. Our system will use the Client-Server model where users will connect to our server to interact with our database to obtain and store information.

Figure 2.1: High Level Performance Of Our System Diagram



- **Browser:** The browser will allow the users to send requests and receive responses from the server in a user friendly webpage.
- **Server:** The server receives requests from users and depending on the user's requests, the server might be able to handle the requests directly in the server and send a response back to the user. However in some cases, the server might require to fetch data from the database and then send the data back to users.
- **Database:** The database stores all the data that both the server needs and the data the user sends the to database.

Design Issues

Issue 1: What language will the back end system be implemented in?

- Option 1: Php
- Option 2: python
- Option 3: ruby on rails

Decision: Option 1. This is because our the majority of our team members are more acquainted with the syntax of Php, being similar to the syntax of Java.

Issue 2: How will the server be set up?

- Option 1: Host a server on our machines.
- Option 2: Buy a domain name and a hosting service online.

Decision: Option 2. Due to the time constraint, we have decided to buy a hosting service online so that we can get started on the content of our application immediately.

Issue 3: How should the reports be displayed to the users

Option 1: Display in a vertical list view, updating and showing more information when user scroll till the end of page.

Option 2: Display in a vertical list view, providing users with numbered pages which will load and show users different information.

Option 3: Display in a horizontal view similar to a slideshow. The UI will have a horizontal bar showing a number of reports and user can navigate through arrows on the left and right side.

Decision: Both Option 2 and Option 3. Option 3 will be used in the main page to provide the user a clean view of most recent posts. Option 2 will be used when user wants to view more or search specifically for certain posts.

Issue 4: Who can modify a post?

Option 1: Only the poster is allowed to modify their own posts.

Option 2: Allow other users to add contents to a post.

Decision: We have chosen Option 2. Although it is more difficult to implement and it complicates the simple UI we are implementing, it may allow other users to provide more information to the search and may provide a huge benefit for the poster. Other users can only modify their own content while the poster have full administrative rights on their posts.

Issue 5: What will the database architecture be like?

Option 1: mysql

Option 2: mongoDB

Option 3: mariadb

Option 4: etc

Decision : Option 1. Since we are using PHP as our back end language, MySQL is the most popular database system used with PHP. Furthermore, MySQL is also the de-facto standard database system for websites. Thus, we will use MySQL as it have a more complete documentation and more examples compared to its competitors.

Issue 6: Should users be able to post as anonymous?

Option 1: Allow users to retain anonymity when posting or commenting

Option 2: Users are required to sign in to an account to post or comment

Decision: Option 1. We will allow users to view posts as anonymous but we will not allow users to post or comment while not signed into an account. This is because we want users to be able to maintain accountability when creating a post and commenting on one.

Issue 7: How do we determine user address when recommending posts to them?

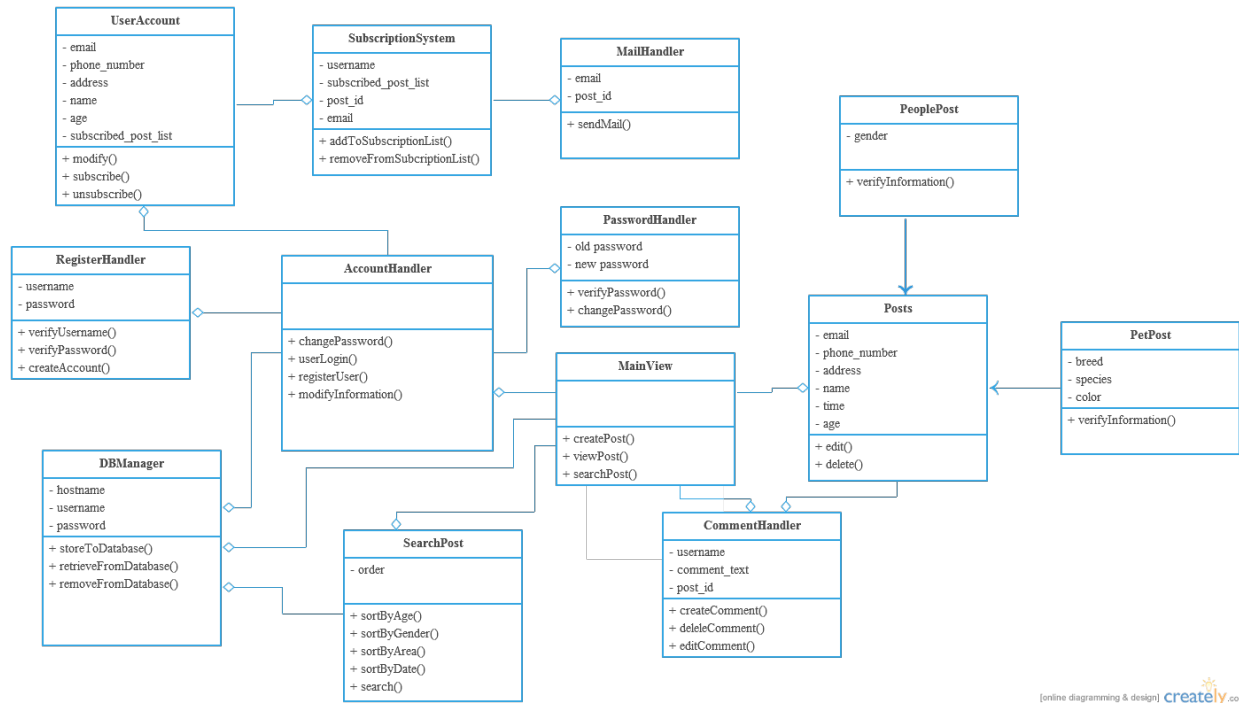
Option 1: use address from their personal profile

Option 2: extract client address based on their IP address

Decision: Option 1. We will recommend posts based on user input to exercise user freedom. It is also easy to implement given our current time constraint.

Design Details

Figure 4.1: Class Diagram



MainView:

The first thing users will be greeted with when accessing our web application is the homepage of the website on their web browsers. Guest users will be able to search and view posts. However in order for them be able to perform a wider variety of actions such as creating and commenting on posts, users are required to log into the system with an account. Users can register or log into the system with the help of the account handler class.

Account Handler:

The services that our application offers will require that our user has an account registered in our system. The account handler handles everything from account registration to account login. The account handler is also involved in other account-centered services such as changing the password of an account or simply editing details tied to an account.

Register Handler:

When new users to our system sign up for an account, the account handler class will call on the register handler class, which is purely in charge of handling account registration. Registering an account is simple and straightforward. The register handler prompts the user for an available username and password, verifies that it is in appropriate format, and proceeds to create the account tied to that login.

User Account:

When the user successfully logs into an account, the user account class handles filling and editing of the personal information of the accountholder. It also keeps track of the subscriptions made by the accountholder.

SubscriptionSystem:

Subscription services are handled by the Subscription System class. Subscribed posts of an account is interpreted as a list of post objects for simplicity and ease of implementing. When a user subscribes to a post, the subscription system class simply fetches the list of subscribed posts tied to an account and simply adds the posts to the list. In the case of unsubscribing to a post, the subscription system fetches the list and removes the specified post.

MailHandler:

When a user subscribes to a post, the user will receive updates and notifications revolving around that post. The updates and notifications come in the form of e-mails to the email address tied to the account. This service is handled by the mail handler class. The mail handler will fetch the user's email address from his account and send updates to the user who subscribed to the post with the given post id; unique to all posts.

PasswordHandler:

Another service offered to account holders is the option of changing their password. When a user attempts to change his password, the password handler prompts the user for the old account password to verify the authenticity before prompting the user for the new account password. The password handler confirms the user's input and then saves the user's input as the new password tied to that account.

SearchPost:

A main feature of our web application is its search system. Our app offers an extensive search system that displays results based on a variety of criteria such as age, gender, area and date. Users can also specify in what order they prefer the search results to be displayed in such as the most recent posts or the posts with the most views.

Posts:

The posts class acts a superclass to the two subclasses, the PeoplePost class and the PetPost class, that have a lot in common. The post class also handles the general actions involving posts such as the creation and modification of posts. When a user attempts to create, edit or delete a post of their own, the posts class will then coordinate the services accordingly.

PeoplePost:

There are two types of posts that our system handles, one of the two is a missing person post. A missing person post requires the user to fill in information regarding the missing person such as name, time went missing, age, gender, and description. The user is also required to fill in contact information that the user can be reached at such as their email, phone number and address.

PetPost:

Another type of post our system handles is a missing pet post. In this case, users are still required to fill in their contact information. However, a missing animal has different fields of information, compared to a person. A pet post will require the user to fill in information that applies to their pets such as it's name, age, species, breed and color for identification.

CommentHandler:

Every post contain a comment section for the public to interact and communicate with the poster. The comment section of every post will be handled by a comment handler class that is in charge of the display or removal of comments on a post. The comment handler will fetch a post id, the username of the account posting the comment, and the content of the comment. It will then record and display the comment on the specified post. When removing a comment, the comment handler simply erases the comment from the records of a post.

DBManager:

The DBManager class will handle all connections to the database. It will store or retrieve information from the database when requested.

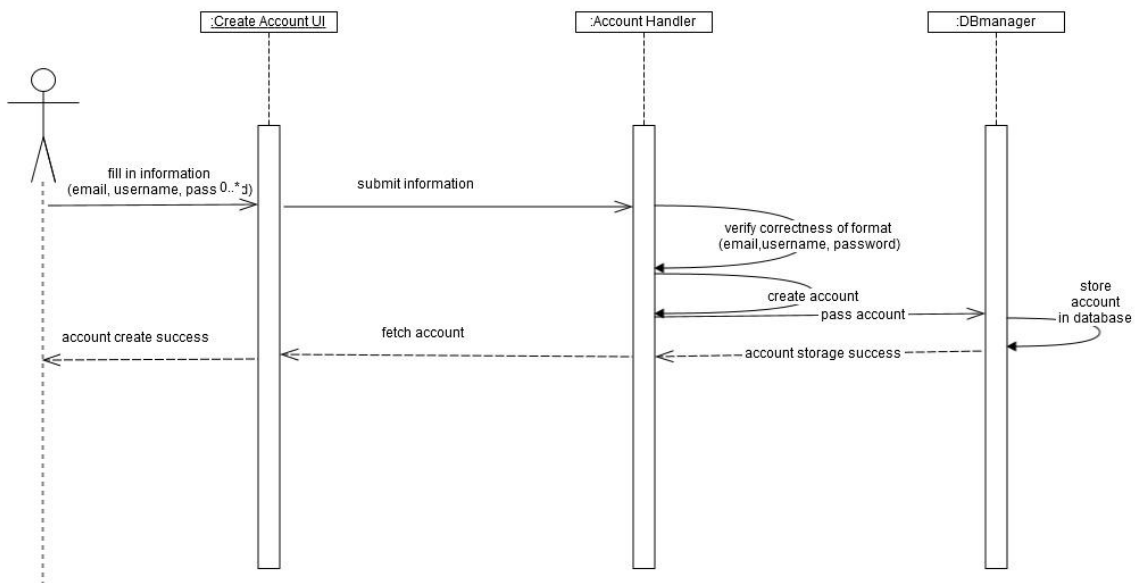


Figure 4.2: The sequence of events when a user creates an account

Create a new account: this allows users sign up an account and fill in personal information.

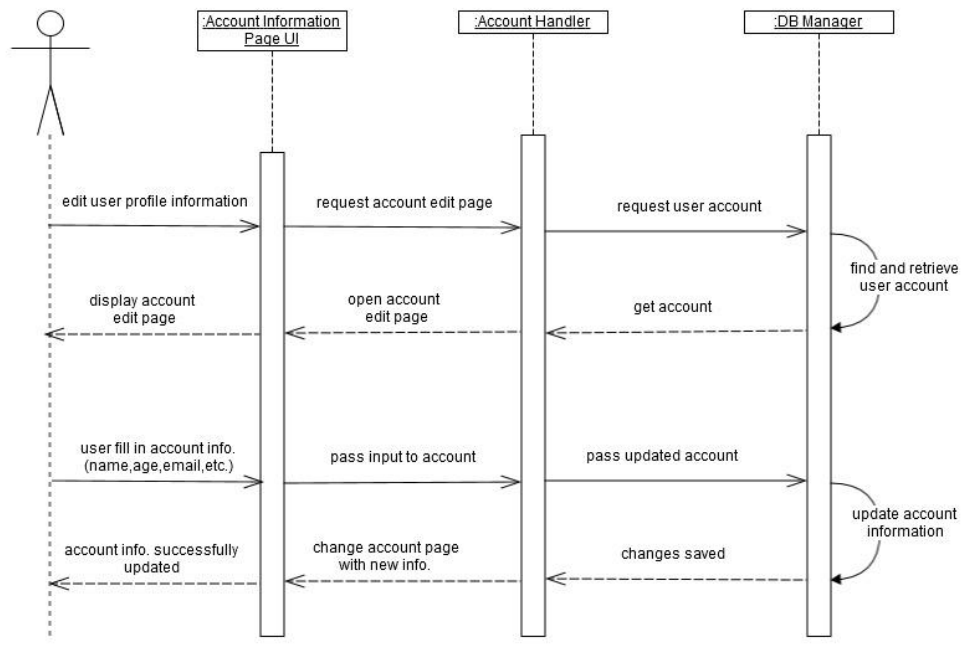


Figure 4.3: The sequence of events when a user fills in or edit his account information.

Edit personal account information: this allows a user change his/her personal profile information.

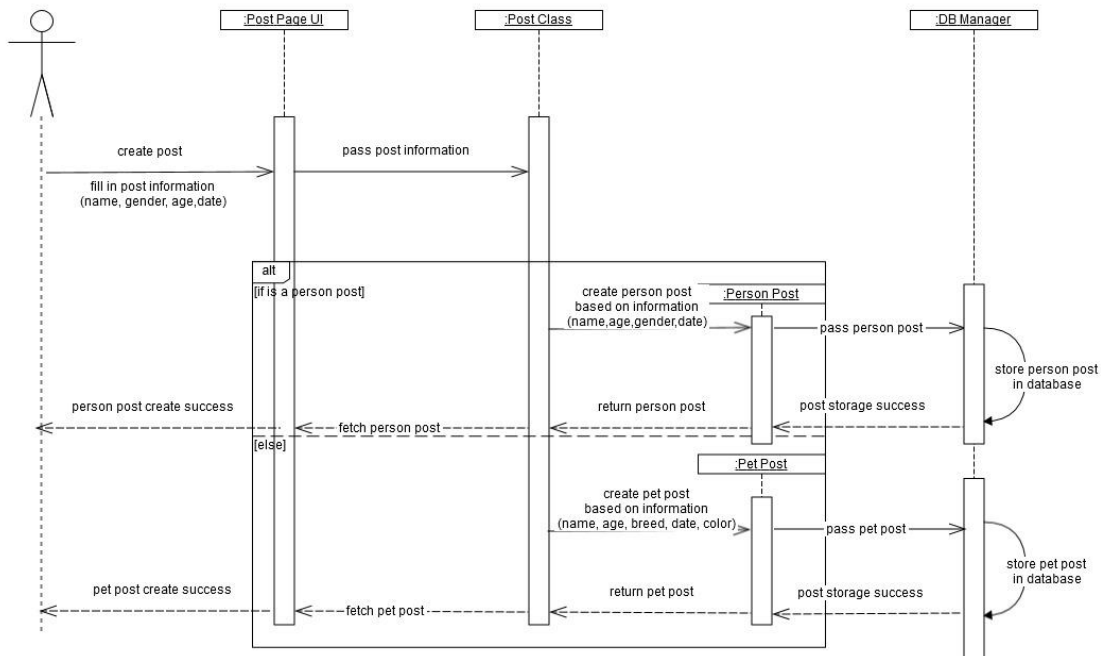


Figure 4.4: The sequence of events when a user creates a post.

Create a post: this allows users make a new post.

UI Mockup

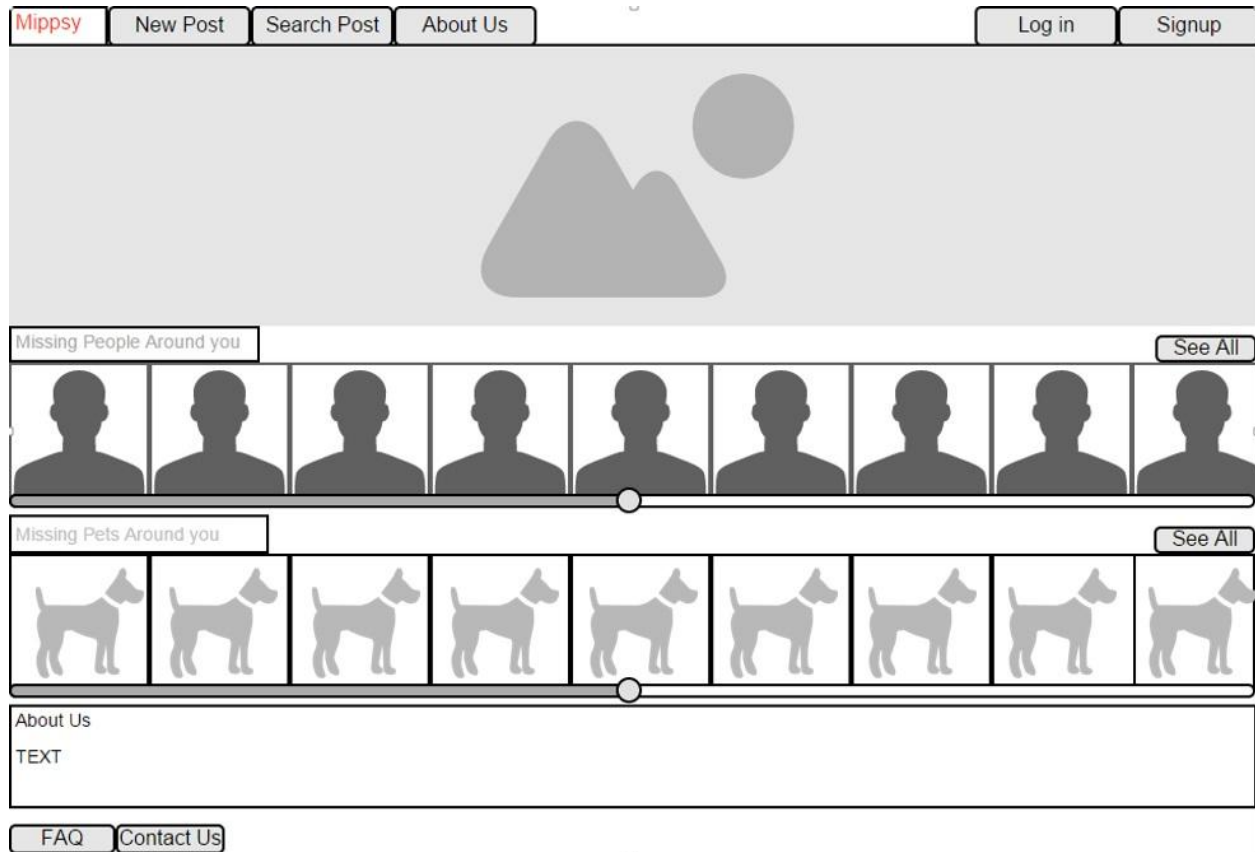


Figure 4.5: Main Page

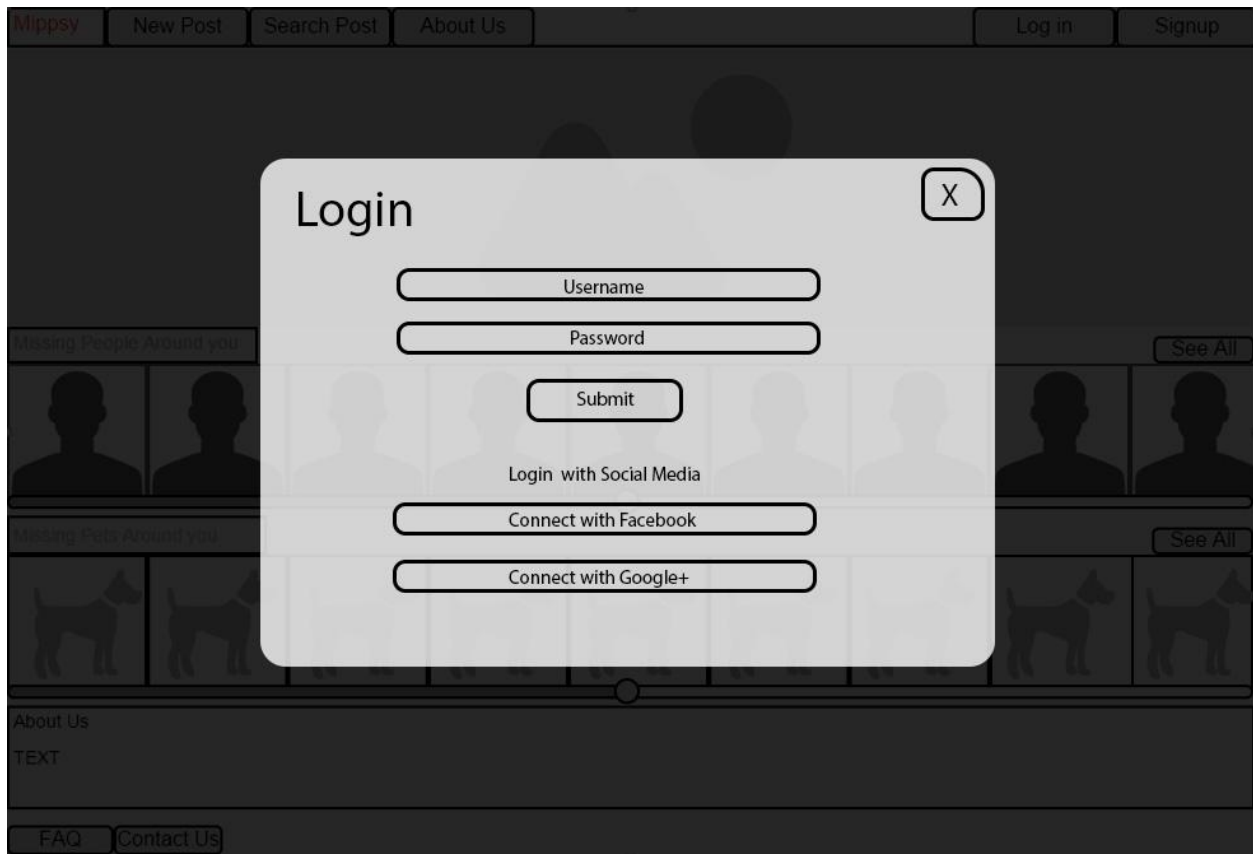



Figure 4.6 : Login overlay window

Mippsy

New Post


Search Post

About Us

 ↓

Log Off

Create New Report



Upload Image

Information

Name:

Age:

Missing Area:

Gender:

Extra Information:

Contact Information

Name:

Address:

Phone Number:

Email Address:

Submit

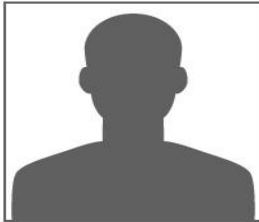
Cancel

FAQ

Contact Us

Figure 4.7: Post creation

Report Name



Information

Name: Age:

Missing Area: Gender:

Extra Information:

Contact Information

Name:

Address:

Phone Number:

Email Address:

Comments

Figure 4.8:Report View