



Bilkent University

Department of Computer Engineering

Senior Design Project

Project Analysis Report

Project Name: PeerNews

Group Members: Şamil İraz
Enes Emre Erdem
Muhammed İkbâl Kazar

Supervisor: Prof. Özcan Öztürk

Jury Members: Prof. H. Altay Guvenir
Assoc. Prof. Selim Aksoy

Innovation Expert: Dr. Ali C. Begen

Analysis Report

Mar 25, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

- 1. Introduction**
- 2. Current System**
- 3. Proposed System**
 - 3.1. Overview**
 - 3.2. Functional Requirements**
 - 3.3. Nonfunctional Requirements**
 - 3.3.1. Usability**
 - 3.3.2. Scalability**
 - 3.3.3. Extensibility**
 - 3.3.4. Performance**
 - 3.3.5. Reliability**
 - 3.4. Pseudo Requirements**
 - 3.4.1. Economic Constraints**
 - 3.4.2. Implementation Constraints**
 - 3.4.3. Time Constraints**
 - 3.4.4. Ethical Constraints**
 - 3.4.5. Sustainability Constraints**
 - 3.5. System Models**
 - 3.5.1. Scenarios**
 - 3.5.2. Use Case Model**
 - 3.5.3. Object and Class Model**
 - 3.5.3.1. Protocol Object / Class Model**
 - 3.5.3.2. Connector Server Object / Class Model**
 - 3.5.3.3. User Interface Object / Class Model**
 - 3.5.4. Dynamic Object Models**
 - 3.5.4.1. Sequence Diagrams**
 - 3.5.4.1.1. SignIn Sequence Diagram**
 - 3.5.4.1.2. SignUp Sequence Diagram**
 - 3.5.4.1.3. Upvote Sequence Diagram**
 - 3.5.4.1.4. Post New Message Sequence Diagram**
 - 3.5.5. User Interface - Navigational Paths and Screen Mockups**
- 4. Glossary**
- 5. References**

1. Introduction

Media has been the main channel of communication between people since the invention of printing press. Lately, technological development migrated media from printing press to internet. These technological developments enabled media content creators to reach broader range of people because of the improvements in cost effectiveness and latency. Thanks to internet media, everyone is able to participate actively as content creator in social media. As a result, people's expectations from media have changed. People started getting more and more concerned about privacy, security and freedom of speech.

Media distribution has been centralized starting from the times of printing press where powerful authorities controlled the content. Even though technological developments enabled everyone to share their ideas rather than just consuming media, internet media such as Twitter, Facebook remained centralized. Centralized media distribution means trusted authority has the ability to abuse their power and restrict the freedom of speech by not allowing certain types of content to be distributed.

Decentralized architectures are becoming more popular due to recent improvements in peer to peer networking technologies. Crypto currencies such as Bitcoin, Ethereum and the file sharing systems such as Bittorrent are popular examples of renowned decentralized systems. PeerNews will provide a decentralized approach to media distribution by creating a peer to peer network among users. This decentralized architecture will not have a single point of failure or central authority. Thus everyone's freedom of speech rights will be protected by design.

2. Current System

There was a private messaging application from BitTorrent called Bleep that came out in 2015. It is no longer available, however, the nature of the application provided 1:1 messaging capabilities on multiple platforms with good privacy. However, the capabilities were only limited to personal messaging, not among the members of a group. A quick search reveals that this application is no longer available.

Among the centralized counterparts, there are several alternatives, but their rules regarding the freedom of speech and privacy are at least confusing, if not totally unacceptable to many. Reddit is one of them where the company has the sole power over the user accounts and the messages are distributed over their system.

3. Proposed System

3.1. Overview

PeerNews will be a desktop application targeting everyone who wants to participate in an absolutely independent media. Users are able to share and consume messages from various mediums such as image, video or plain text messages. When consuming content, users able to specify their preferences in order to filter and interact with the messages they are interested in.

PeerNews will consist of a peer to peer networking protocol and a client application. Messages will be used to create a torrent file. Hash of the torrent and several other attributes will be distributed through our own peer to peer network. Then we will make use of the existing torrent network to seed and download the actual content of the message which will be significantly larger in size. Client application will download the content of messages from the torrent network according to user preferences. Client application will receive message

from the peer to peer network and display them to the user appropriately. For instance, comments or upvotes will be an arbitrary message under the protocol layer but on the client application, they will be shown under the respective post.

Small messages containing metadata and torrent hash will be broadcasted to the whole network through peers. Each node will share the message with a certain number of its peers when they receive the message for the first time. This approach is latency and bandwidth efficient since each node needs to forward the message to a constant number of peers (at least 2), and the farthest node is guaranteed to receive the message in logarithmic time. Messages will be reserved in the client application for a set period of expiration time. Thus, users that were offline at the initial broadcast time of a message will still be able to request it from their peers. Once the expiration time is reached offline users will not be able to receive the message anymore. We believe that temporary nature of the messages will create an environment for users to share more freely.

3.2. Functional Requirements

- Users will have the option to download the source code and build the application themselves.
- Users will be authenticated using a P2P authentication protocol similar to Bitcoin's public-key cryptography based system. [1]
- Users will be able to publish contents without any controlling processes such as server.
- When a user joins the network, they will connect to connector node which will assist this new node on connecting to other nodes in the network.
- Users will be able to create content and broadcast the message to the network.
- Torrent hash of the message will be shared with the network, then underlying torrent network will be used to seed and download the actual content which will have a significantly larger size.

- Users will be able to subscribe topics that they are interested in.
- Users will be able to filter what messages to receive based on their preferences.
- Users will be able to fetch messages that were missed when they were offline for a certain period of expiration time which will be determined later on.
- Users will be able to upvote or downvote the news and messages they like or dislike. [2]
- Users will be able to sort their feed by number of upvotes. [2]
- Users will be able to give feedback about the network and application.
- Users will be able to write comments on messages from other users.
- Messages can be any kind of file. We will initially support plain text messages, images and videos.
- There will be two separate feeds. The first feed will be ranked according to personal preferences and the second feed will consist of popular messages.
- Users will be able to search on the metadata of the messages. Metadata will include tags, publisher and short description in addition to the torrent hash of the actual content.
- Users will be able to use same account in all platforms with the public-private keys.
- Users will be able to export their public and private keys to local storage.
- Users will be able to join network without providing any private information.
- Users will be able to copy other people public keys.
- Users will be able to search for a particular publisher by searching with publisher's public key.

3.3. Nonfunctional Requirements

3.3.1. Usability

- Interface will be user-friendly so that non-technical users are also encouraged to participate in the peer to peer network.

3.3.2. Scalability

- Since PeerNews does not have a centralized server, there cannot be an overload of users or messages. The performance issues are all related to the performance of WebTorrent.

3.3.3. Extensibility

- Users will be able to modify their client applications since it will be open source. This will allow for customizations on the client side.

3.3.4. Performance

- Latency of broadcasting a message should be less than 10 minutes in a crowded network of over 1 million nodes. Furthermore latency will be asymptotically bounded by $O(\log N)$ where N is the number of nodes in the network.

3.3.5. Reliability

- Application will be running all the time. Users will always have access to PeerNews.
- Messages will be accessible as long as there is at least one online node with that message since there is no single point of failure. Crash of one node does not harm any other node in the peer to peer network significantly.
- For the distribution of the actual content of the messages, existing torrent network will be used. Since torrent is an already established peer to peer network message-content delivery will be reliable.

- Authentication will be provided with the private-public key technology. As a result, there will not be any profile stealing.
- In the application, hashcash security system will be used so as to block any possible spam attacks to peers through network [3], [4].

3.4. Pseudo Requirements

3.4.1. Economic Constraints

- Application will be free of charge.
- Application will be open source with MIT Licence and community will do the maintenance and further improvements as open source contributors.
- Due to P2P nature of the architecture, client side of the application will consume more internet bandwidth than a usual social media product would.

3.4.2. Implementation Constraints

- Application will be a desktop application. A web version can be developed in the future.
- Github will be used for version control and development collaboration.
- Implementation will be done in Java mainly.
- All news will be transferred through the Torrent network. Users will upload the message to the Torrent network and receive its hash code. Then, they will share the hash code with their peers through the network.

3.4.3. Time Constraints

- The project will be finished in December, 2019.

3.4.4. Ethical Constraints

- Users will have absolute freedom of speech. Messages will not be sent through a server. Thus, there is no power above users which is able to ban or limit them.

- No information is hidden from anyone in the network. All data is public and will be transferred through the Torrent network.
- Since developers don't have any control over what is shared on the network, in no event shall the developers be liable for any claim, damages or other liability.

3.4.5. Sustainability Constraints

- We will have our own nodes in the network for the purposes of monitoring. These monitoring nodes will alert developers when there is an unexpected error.

3.5. System Models

3.5.1. Scenarios

Scenario 1 Application Opening:

Actors: User

Entry Conditions: User opens the desktop application.

Exit Conditions: User closes the desktop application.

Main Flow of events:

1. Welcome page will be shown to users
2. Users will sign in or sign up

Scenario 2 Sign in:

Actors: Registered User

Pre Condition: User has a pair of keys.

Post Condition: Key pair is accepted by the application

Entry Condition: User clicks the sign in button in the main page.

Exit Condition: User is directed to the feed page.

Main Flow of events:

1. If the application already has a pair of keys stored in the file system it is used to sign in the user.

2. Else user is prompted to enter their key pair.
3. Application will verify that public private keys are compatible with each other.
4. User will be redirected to feed

Scenario 3 Sign Up:

Actors: New User

Post Condition: New key pair is generated as a valid pair and new user added to the network of application.

Entry Conditions: User clicks the sign up button in the main page.

Exit Conditions: User is directed to the feed page.

Main Flow of Events:

1. User will click the “Sign Up” button in the welcome page.
2. Application starts to generate new valid key pair for the new user.
3. Application will send the new user’s information(key pair) to other users in the network to notify them about new user.
4. User will be redirected to the feed page.

Scenario 4: Viewing Feed

Actors: Registered User

Pre Condition: User authenticates successfully.

Post Condition: Feeds are shown to the user if they are available.

Entry Conditions: User clicks the ‘Feed’ button.

Exit Conditions: User opens another page or exit application.

Main Flow of Events:

1. Application filters all messages and shows posts only.
2. Upvotes and downvotes related to each post are counted and showed next to them.
3. Post are listed one after another, with their titles.

Scenario 5: Viewing Popular Posts

Actors: Registered User

Pre Condition: User authenticates successfully.

Post Condition: Popular posts are shown to user.

Entry Condition: Registered user clicks to `Popular`.

Exit Condition: User opens another page or exit application.

Main Flow of Events:

1. Application filters all messages and collects posts.
2. The messages are sorted according to their number of upvotes and downvotes. Popular posts are listed on top.

Scenario 6: Subscribing Topics:

Actors: User

Pre Conditions: User authenticates successfully.

Post Condition: User subscribes to at least one topic.

Entry Condition: User opens the `Topic` page.

Exit Condition: User opens another page or closes application.

Main Flow of events:

1. User clicks the topics tab to view the list of topics.
2. User clicks the subscribe button on a selected set of topics.
3. Messages with recently selected topics will be added to the feed.

Scenario 7: Searching Posts:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: Searched posts are shown to user.

Entry Conditions: User opens the search page.

Exit Conditions: User opens another page or exits application.

Main Flow of events:

1. User enters keywords to the search bar.
2. Posts are filtered according to occurrences of the keywords in message titles.
3. Posts matching the given keywords are shown as a list of post titles and upvote/downvote numbers.

Scenario 8: Viewing a Post:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: Post is shown to user.

Entry Conditions: User clicks the post's title button.

Exit Conditions: User clicks back button to go back or closes the application.

Main Flow of events:

1. User clicks a post title to view its contents.
2. Application redirects to post page.
3. Post title, content, upvote/downvote numbers and comments are shown in the page.

Scenario 9: Voting a post:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: The message of vote has been created and published publicly.

Entry Conditions: User clicks to post to show it.

Exit Conditions: User clicks the upvote or downvote button.

Main Flow of events:

1. User clicks upvote or downvote buttons.
2. Voting message is broadcasted to the network.
3. Upvote/downvote numbers of the post is updated.

Scenario 10: Posting Plain-Text Message:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: The message of the new post has been created, the torrent file of this new post has been created and hashcode of torrent file is shared publicly.

Entry Conditions: User opens the new post page.

Exit Conditions: Users creates the new post by clicking 'Post' button.

Main Flow of Events:

1. User will click to "Post" button in the application.
2. User will select the "Plain Text" option to create text post.
3. User will write his message to the text box in the application page.
4. User will click to "send" button to post his text to other users.
5. Application will upload the post to the torrent and retrieve the hash code.
6. Application will share the hashcode with network through peers.

Scenario 11: Commenting to a Post:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: The message of the new post has been created, the torrent file of this new post has been created and hashcode of torrent file is shared publicly.

Entry Conditions: User opens the new post page.

Exit Conditions: Users creates the new post by clicking 'Post' button.

Main Flow of events:

1. User clicks comment button on a post.
2. Comment creation page is opened.

3. User writes a comment in a text box.
4. User clicks send comment button.
5. Comment message is created, signed and then broadcasted to the network.

Scenario 12: Posting Video:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: The message of the new post has been created, the torrent file of this new post has been created and hashcode of torrent file is shared publicly.

Entry Conditions: User opens the new post page.

Exit Conditions: Users creates the new post by clicking 'Post' button.

Main Flow of Events:

1. User will click to "Post" button in the application.
2. User will select the "Video" option to create text post.
3. User will choose the file in the video format that is wanted to be posted from the local storage.
4. User will click to "send" button to send the post.
5. Application will upload the post to the torrent and retrieve the hash code.
6. Application will share the hashcode with network through peers.

Scenario 13: Posting Image:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: The message of the new post has been created, the torrent file of this new post has been created and hashcode of torrent file is shared publicly.

Entry Conditions: User opens the new post page.

Exit Conditions: Users creates the new post by clicking 'Post' button.

Main Flow of Events:

1. User will click to "Post" button in the application.
2. User will select the "Image" option to create text post.
3. User will choose the file in the image format that is wanted to be posted from the local storage.
4. User will click to "send" button to post his text to other users.
5. Application will upload the post to the torrent and retrieve the hash code.
6. Application will share the hashcode with network through peers.

Scenario 14 Viewing User's Own Posts:

Actors: User

Pre Condition: User authenticates successfully.

Post Condition: Posts of user are shown to user if they are available.

Entry Conditions: User opens the profile page.

Exit Conditions: User goes back to feed by clicking the back button or exits the application.

Main Flow of events:

1. User clicks the profile button.
2. Profile page is shown the users which includes previously sent posts

Scenario 15 Exporting Keys:

Actors: User

Pre Condition: User authenticates successfully.

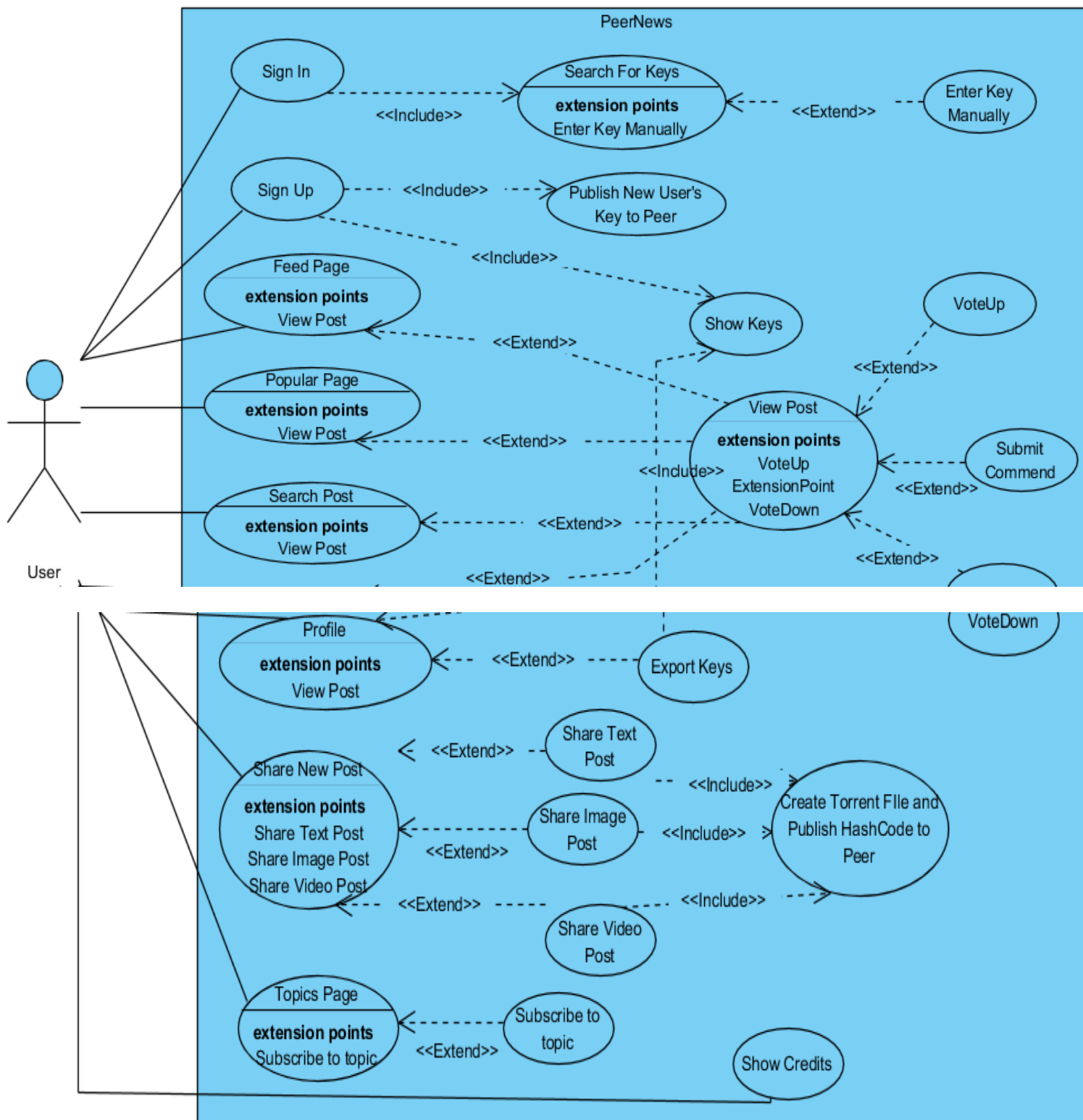
Post Condition: Keys of user are exported as a text file to local storage.

Entry Conditions: User opens the profile page.

Main Flow of events:

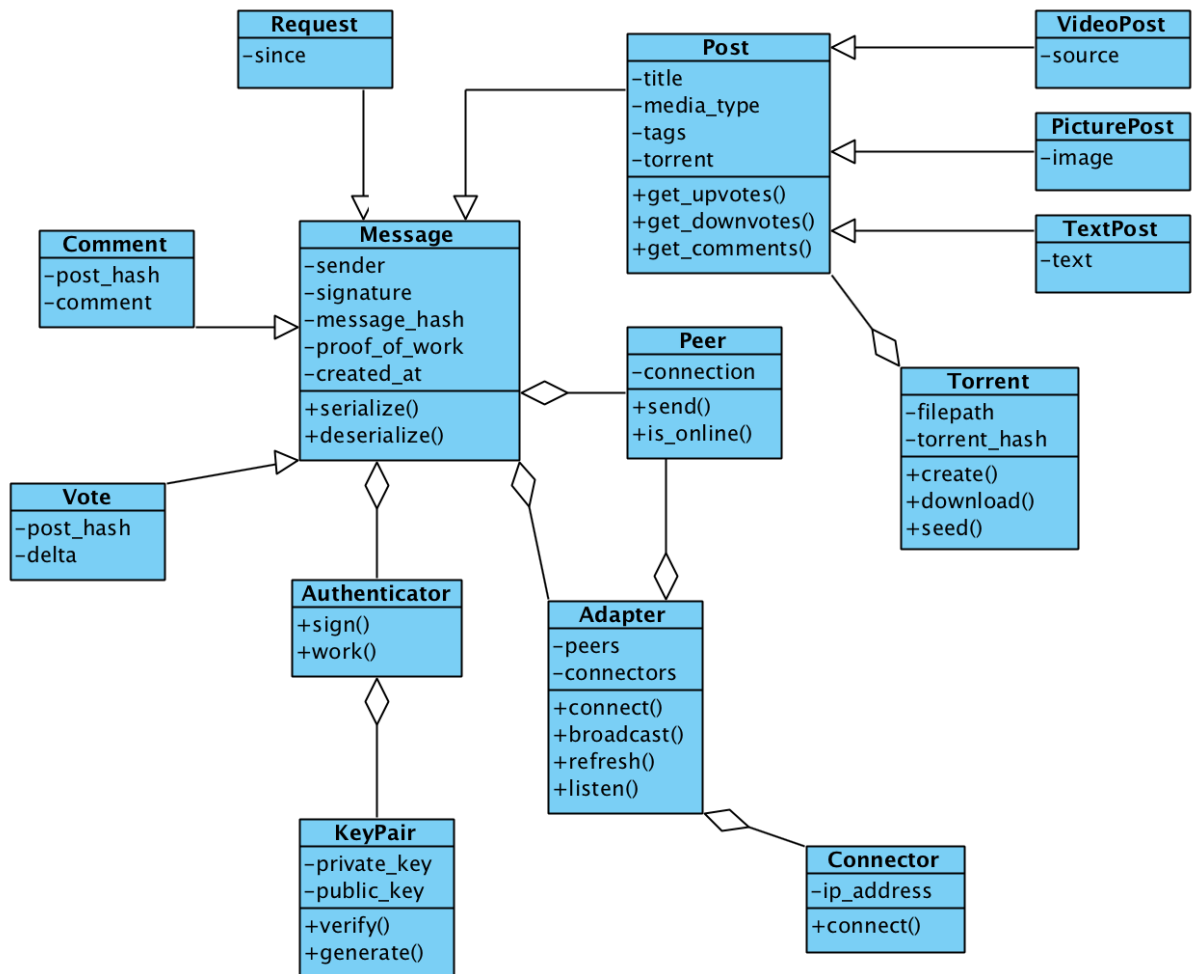
1. User navigates to the profile page.
2. User clicks export keys to view and download their public private key pair.
3. Application redirects user back to profile page.

3.5.2. Use Case Model



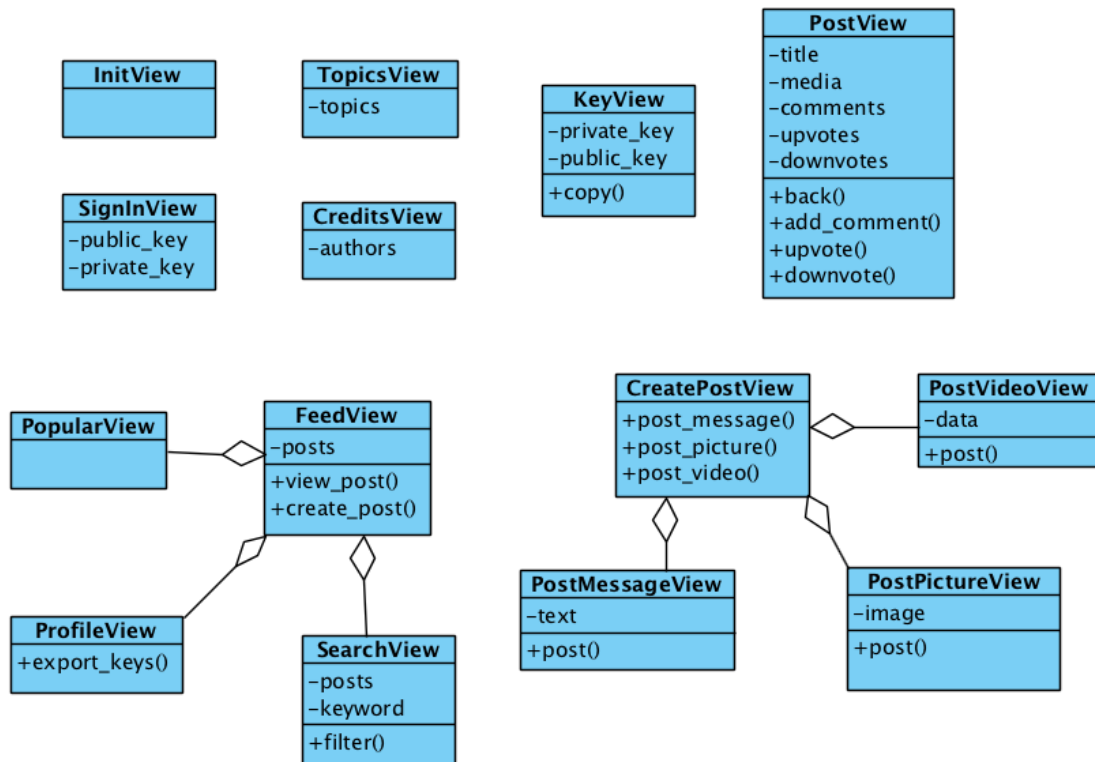
3.5.3. Object and Class Model

3.5.3.1. Protocol Class Diagram



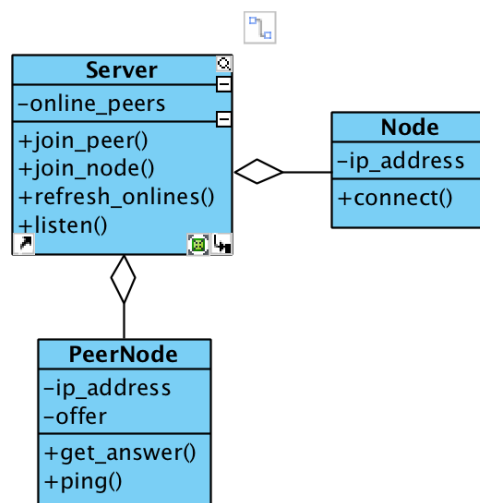
- Message class is the base a class that represents a basic message object. Each message will contain various meta-data such as signature, hash, proof of work.
- Request, Comment, Vote, Post are subclasses of the Message class. Each represents a particular type of message with additional fields.
- Authenticator class will be responsible for creating a digital signature and proof of work for a given message.
- KeyPair class contains private and public keys of the user. It will also be responsible for verification and initial generation of keys.
- Adapter class maintains a list of connectors and peers that are directly connected to the current user.
- Torrent class will be responsible for the connection with the torrent network.

3.5.3.2. User Interface Class Diagram



- As it can be seen above, each page will be implemented as its separate class responsible for rendering the user interface and navigation.

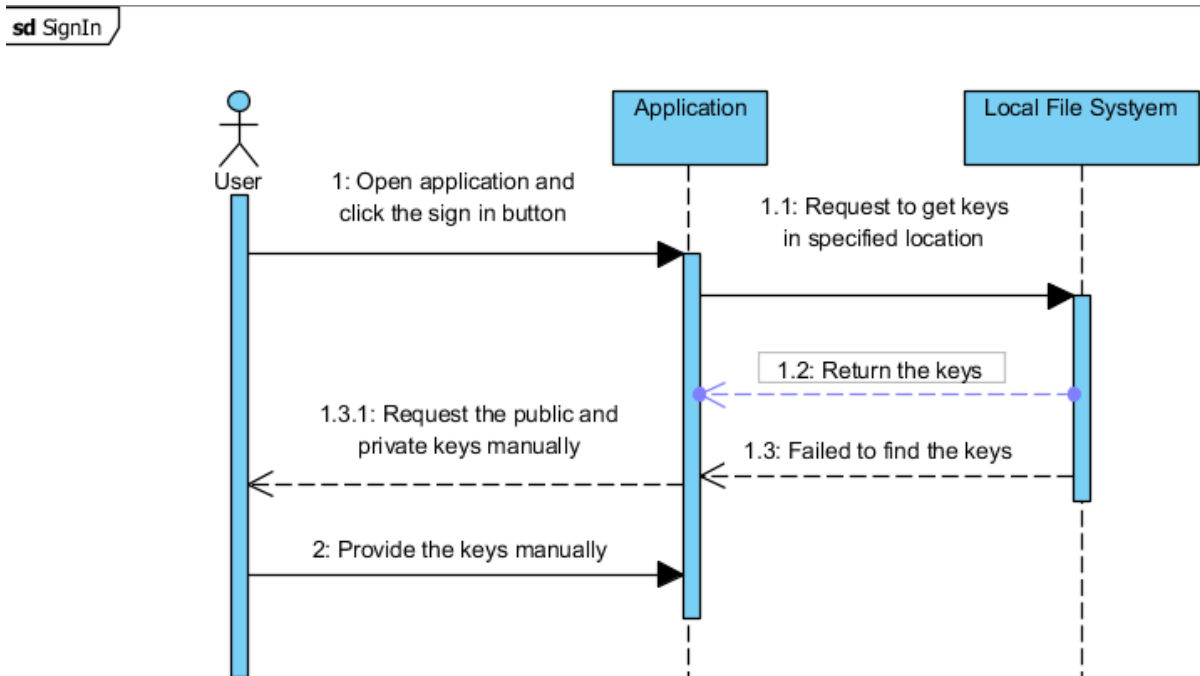
3.5.3.3. Connector Server Class Diagram



3.5.4. Dynamic Models

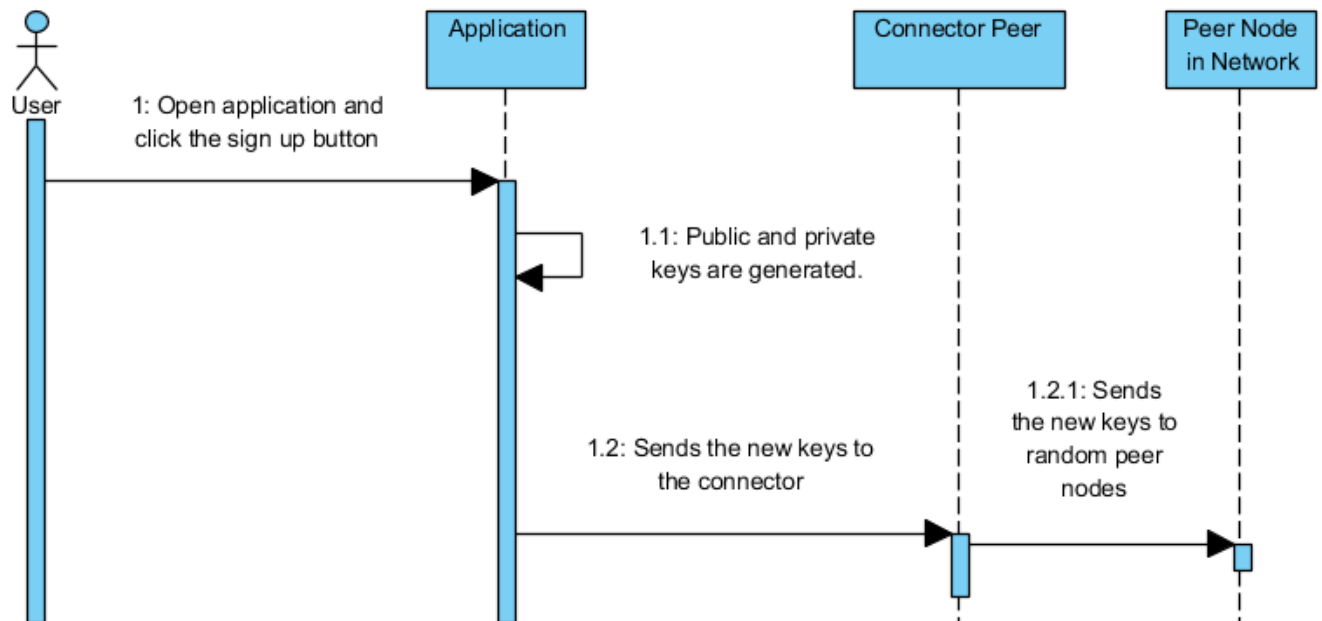
3.5.4.1. Sequence Diagrams

3.5.4.1.1. SignIn Sequence Diagram



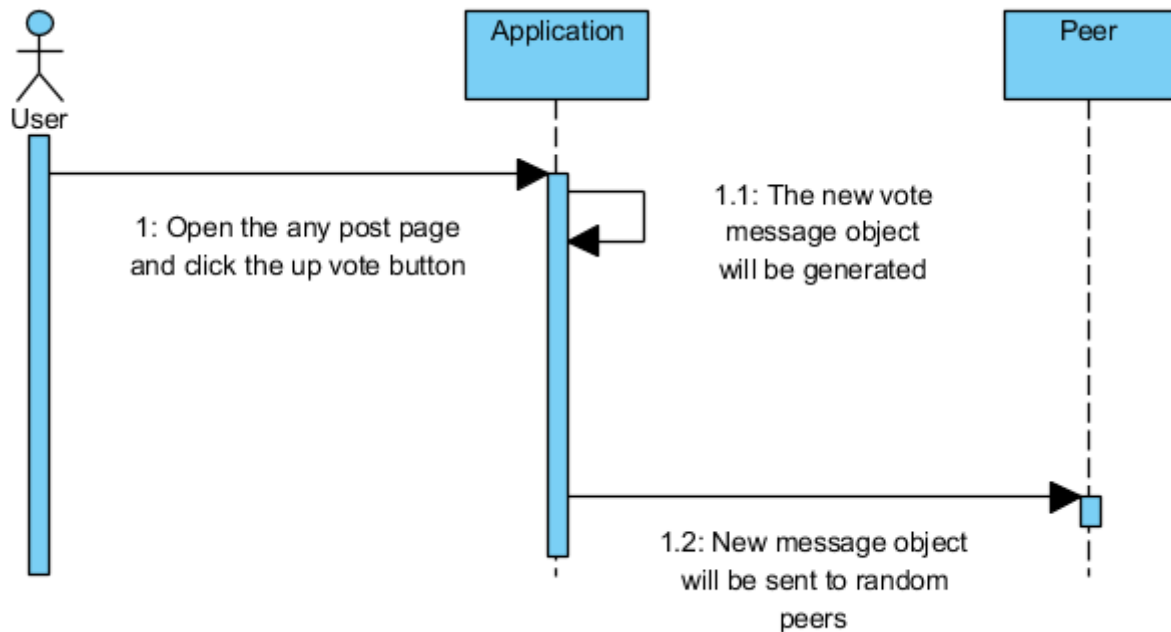
When the user opens the application, there will be two options for him/her: Sign In or Sign Up. After he clicks the sign in button, application will search the public and private key of user in the local file system. If application cannot find the keys, then it will request user to provide the public and private keys. If user provides, then application will be opened.

3.5.4.1.2. SignUp Sequence Diagram



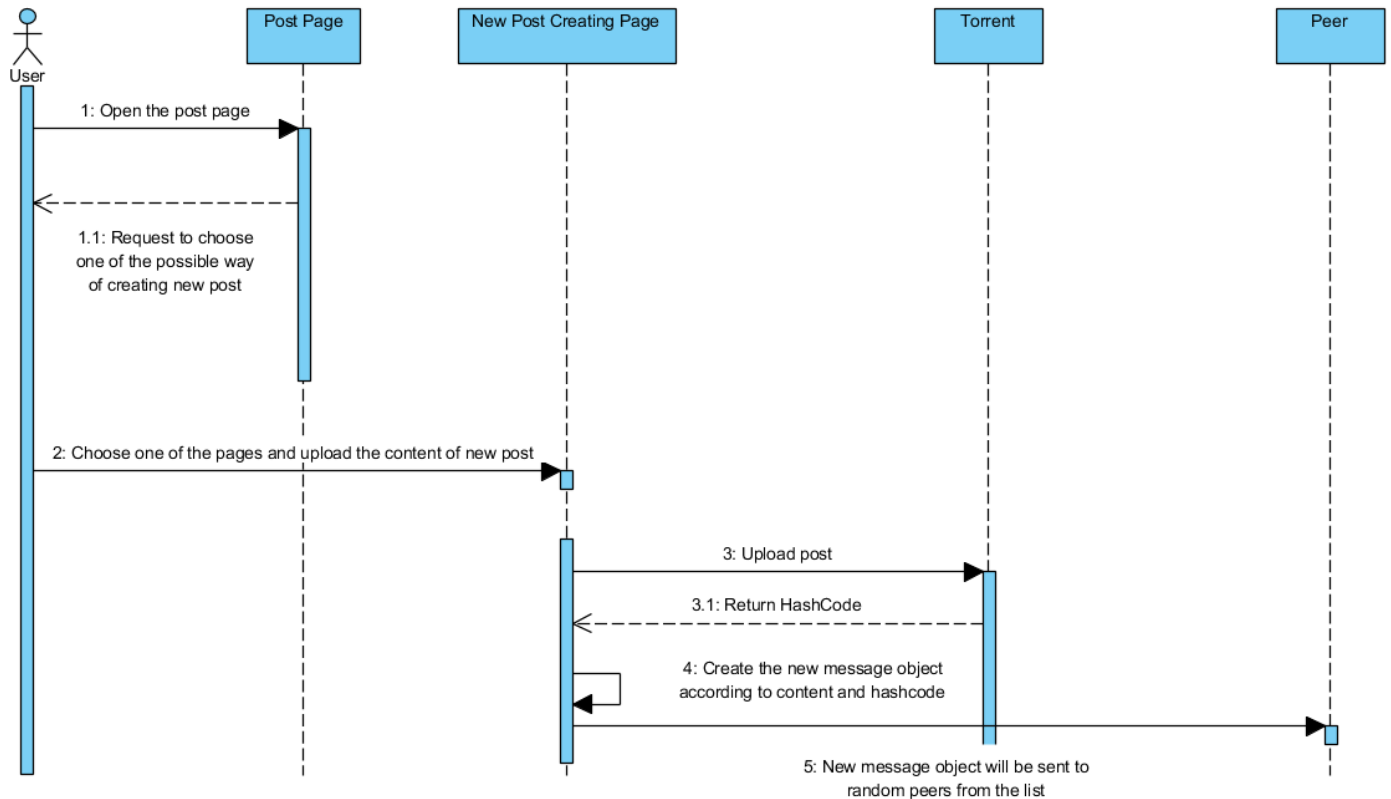
When the user opens the application, there will be two options for him/her: Sign In or Sign Up. After he clicks the sign up button, application will generate the new public and private keys for the new user. After creating the keys, application will send the new keys to the random connector computer in the network. This connector will share these new keys to the network through the random peer nodes.

3.5.4.1.3. UpVote Sequence Diagram



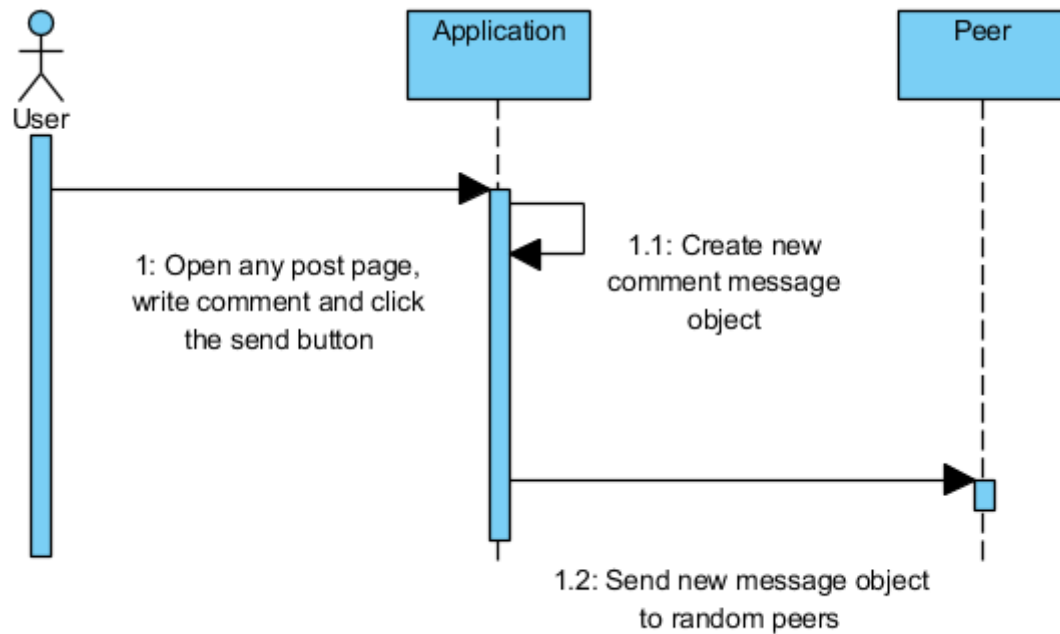
When user wants to upvote the post, s/he will click the upvote button. After clicking, application will create new vote message object and it will send the object to the random peers from the peer list. With this way, everyone in the network will be able to aware of this upvote transaction.

3.5.4.1.4. Posting New Message Sequence Diagram



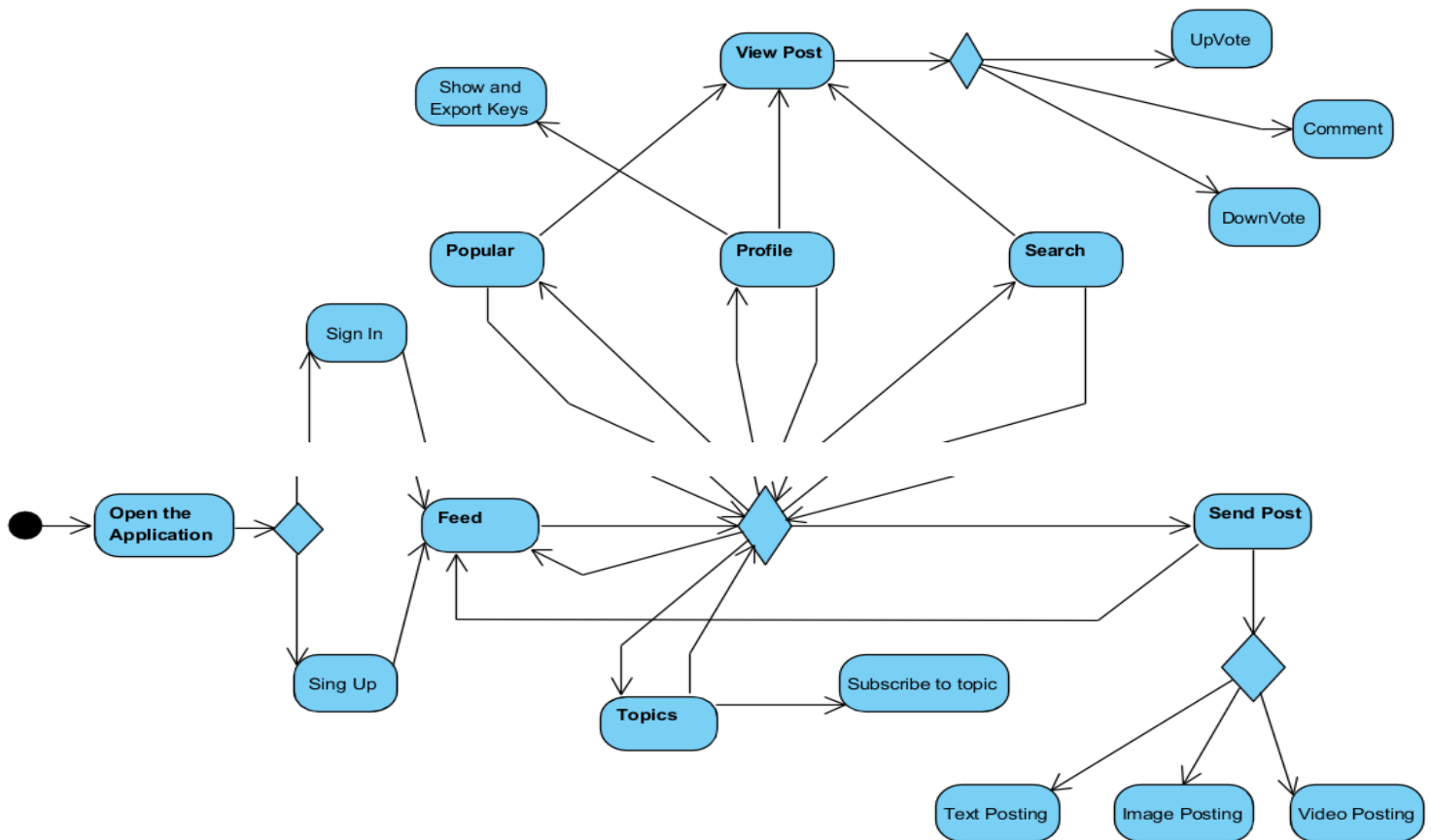
If user wants to create new post and share it, s/he will open the post page and choose one of the possible way of creating post: Text post, image post and video post. After choosing, user will add the contents of the post and click the send button. Application will upload the content to the WebTorrent and retrieve the hashcode from the torrent. It will create new message object according to hashcode and content from user. In addition, application will send this object to the peers.

3.5.4.1.5. Sending Comment Sequence Diagram



Users can post their comments under the original post. They will write their comment to the comment box and click the send button. Application will create new comment message object and publish it through peers.

3.5.4.2. Activity Diagram

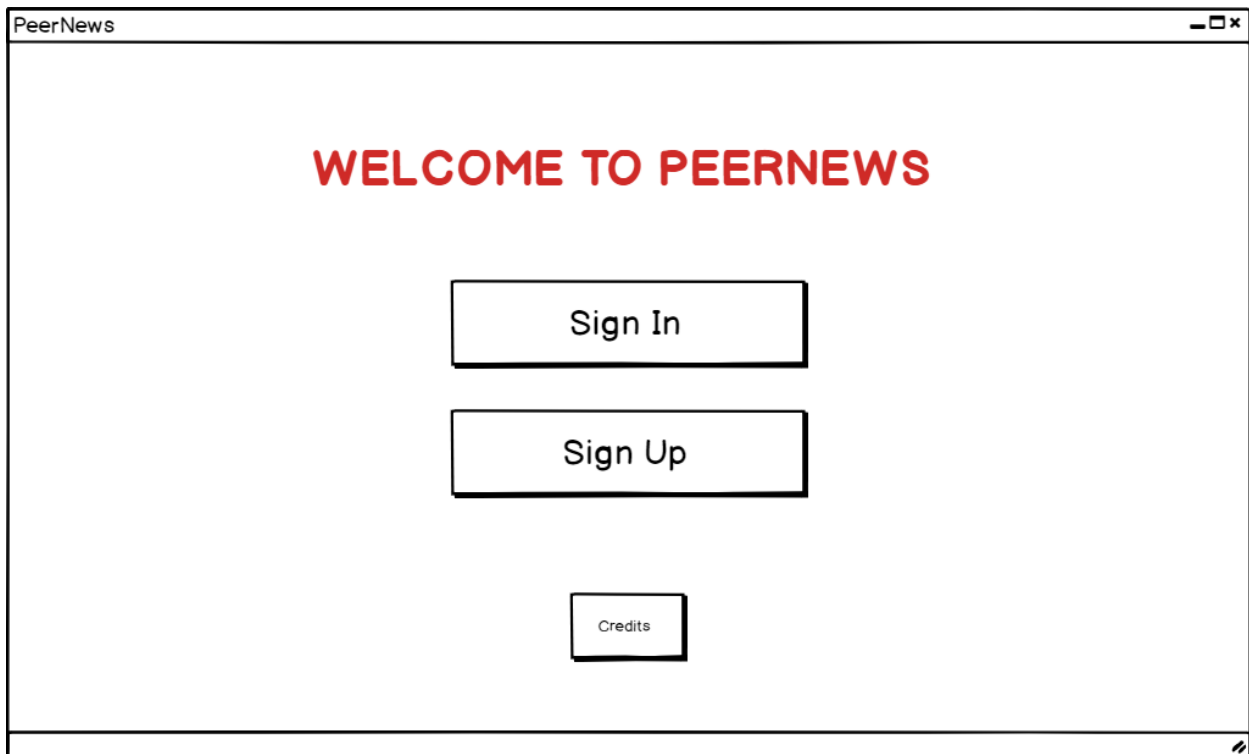


This activity diagram shows how our application can be used. Firstly, user will open the application. After opening the application, Main page will be shown to user and user will be able to either sign in or sign up. These options will authenticate the user to use application. After successfully authenticated, signing in and signing up will direct the user to the Feed page. In the Feed page, the new posts that are related with the subscribed topics of user will be listed. User will be able to open the many pages. User can open the Popular page. In this page, the popular posts among all of the posts will be shown to the user. User can open Topics page. In this page, the topics will be shown to the user and user will be able to subscribe or unsubscribe the topics according to his/her desires. User can open the Profile page. In the profile page, the posts that are sent by the user will be shown and also the public and private keys can

be exported from the Profile page. User will be able to search the posts from Search page according to topic, author or content. From all of these pages, user will be able to open the post by clicking the link of post. In the view page of post, user can upvote, downvote or comment to post. Also user can download the public key of the author from the post page and with this key, user will be able to search this author's posts from the Search page. In addition, user will be able to create new post by opening the SendPost page. In the SendPost page, there will be options to choose. These options are listed to specify the content of the new post. New post can be a text, image or video post. After choosing the option, user will add the content according to choice and send the post to share publicly. There will be option bar in all of the page except the viewing post page and send post page. With this bar, user will be able to navigate in the application to all pages. In the exception pages, there will be back button go Feed page. From all of the pages, user will be able to close the application just by clicking the exit button.

3.5.5. User Interface - Navigational Paths, Screen Mockups

Init Page



Users can click “Sign In” or “Sign up” to navigate to corresponding pages. They can also click credits to see developers and other credited entities.

Manual Sign In Page



The screenshot shows a web browser window titled "PeerNews". In the top-left corner of the page content, there is a black left-pointing arrow. The main text in the center reads: "We could not locate your private and public keys in your local file system. You need to provide your own private and public key to sign in." Below this, a green message states: "If you do not have an account yet just, go back to sign up for free !". There are two input fields: "Private Key :" followed by a text box, and "Public Key :" followed by a text box. At the bottom center is a "Sign In" button. The browser window has standard OS controls (minimize, maximize, close) in the top-right corner.

PeerNews

←

We could not locate your private and public keys in your local file system. You need to provide your own private and public key to sign in.

If you do not have an account yet just, go back to sign up for free !

Private Key :

Public Key :

Sign In

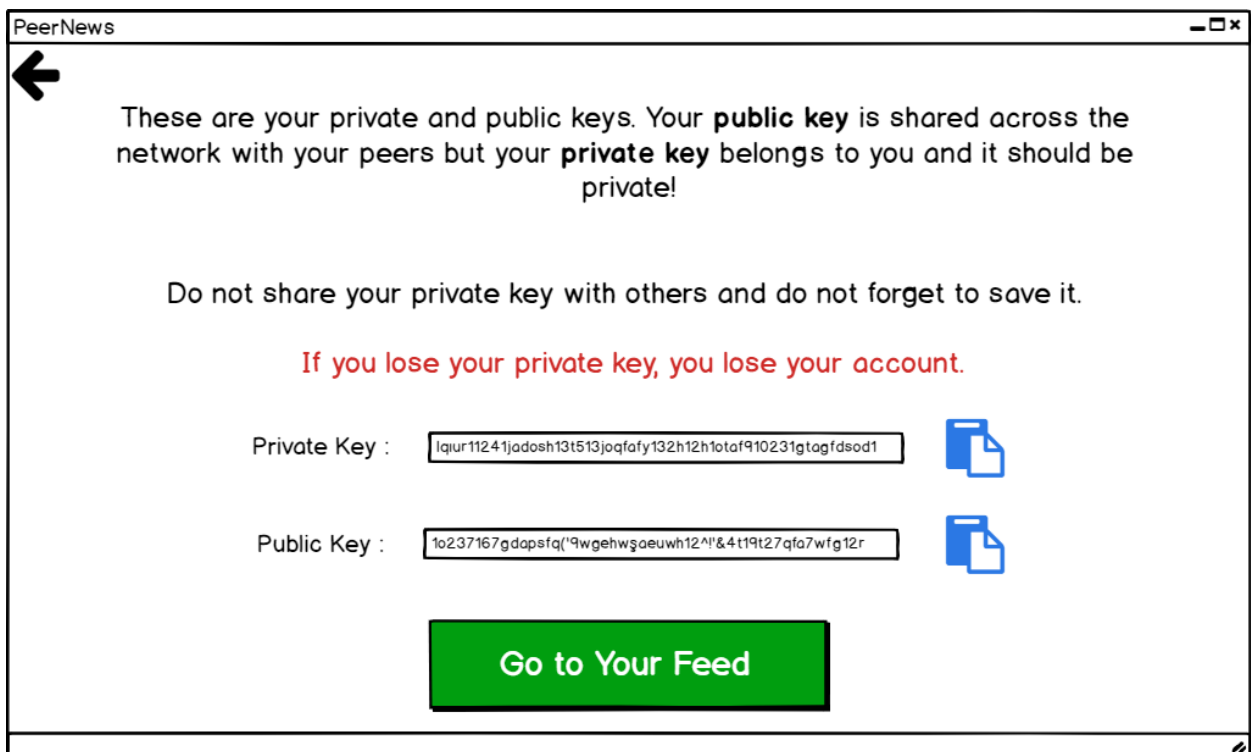
Users can enter their private and public keys to verify and login with their existing key pair. If an existing key pair is found, user is automatically signed in and redirected to feed page. Back button at the top right corner of the page can be clicked to navigate to the previous page. This is consistent across all pages.

Credits Page



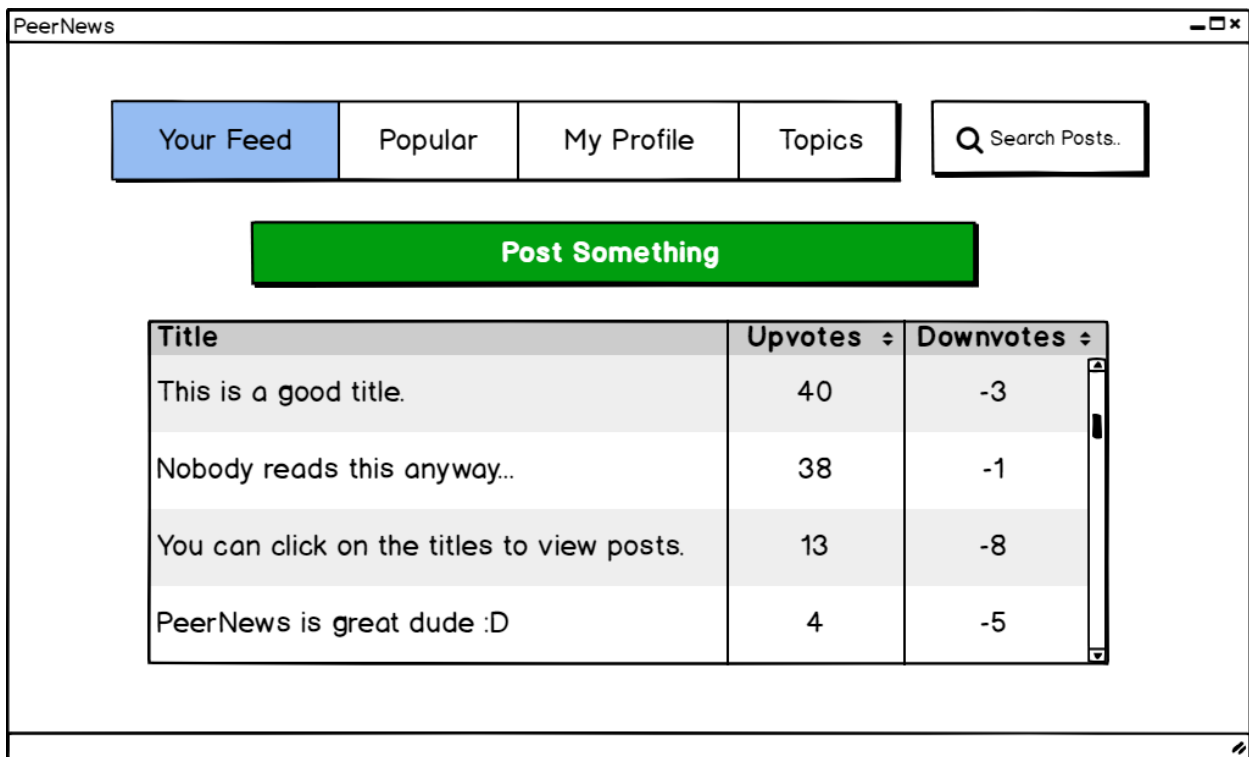
Developers of the project and open source libraries used by the application in the future will be displayed on credits page.

Public and Private Key Display Page



For a signed in user, their public and private keys will be shown on this page. This page will be shown right after sign up as well to let the new user take a note of the recently generated key pair. Warnings are also shown to let users know that they should take a backup of their private keys.

News Feed Page



News Feed Page allows users to view a list of posts shown as a row of title, upvotes and downvotes. Users can select one of the tabs above to navigate to different pages. Search bar can be used to search a keyword among all post titles. Entering some text into the search bar takes the user to search results page. Users can also click “Post Something” to go to the page where they can create a new post.

Popular Page

PeerNews

Your Feed

Popular

My Profile

Topics

Q Search Posts..

Post Something

Title	Upvotes	Downvotes
My post reached popular.	40	-3
Popular posts among all are listed here	38	-1
You can click on the titles to view posts.	13	-8
PeerNews is great dude :D	4	-5

Popular page is where users can see a list of posts that are popular in the network.

Popularity is determined by votes. Most popular post is shown first and so on.

Topics Page

PeerNews

Your Feed

Popular

My Profile

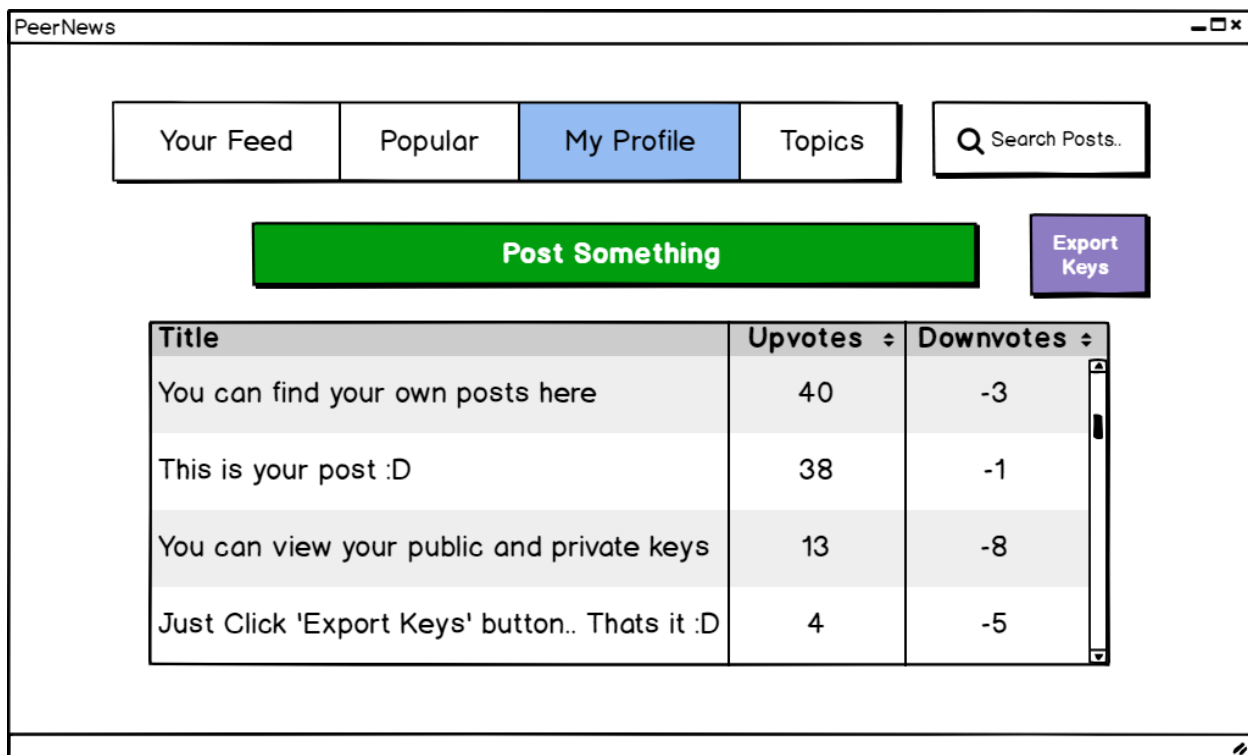
Topics

Q Search Topics

Topic	
Football	Subscribe
CS491	Subscribed
Education	Subscribe
Bilkent	Subscribed

Users can subscribe to different topics and see their existing subscriptions. Users can enter keywords to search in the list of topics.

Profile Page

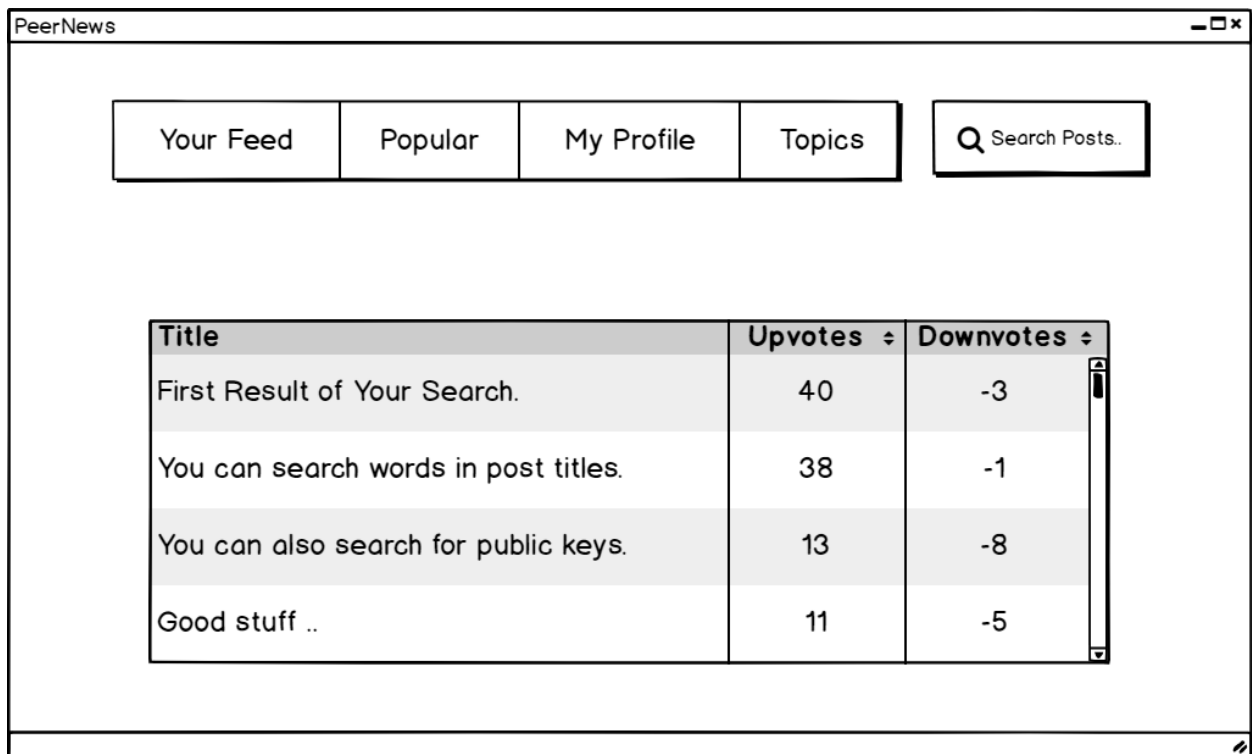


The screenshot shows the 'My Profile' page in the PeerNews application. At the top, there is a navigation bar with four tabs: 'Your Feed', 'Popular', 'My Profile' (which is highlighted in blue), and 'Topics'. To the right of these tabs is a search bar labeled 'Search Posts..'. Below the navigation bar, there is a green button labeled 'Post Something' and a purple button labeled 'Export Keys'. The main content area displays a table of the user's posts. The table has three columns: 'Title', 'Upvotes', and 'Downvotes'. There are four rows of posts listed. A scrollbar is visible on the right side of the table, indicating that there are more posts than can be displayed at once.

Title	Upvotes	Downvotes
You can find your own posts here	40	-3
This is your post :D	38	-1
You can view your public and private keys	13	-8
Just Click 'Export Keys' button.. Thats it :D	4	-5

Profile page lets the users see their previous posts in the network. In addition, users can also click “Export Keys” to view and download their public-private key pair.

Search Results Page



Search Results page shows the results of the search made by user. A search can be done by simply searching keywords in post titles or searching public keys of users. When a keyword search is done, all posts which their titles include the keyword will be listed. However, when a public key search is done, the posts of the user whose public key is the exact match of the searched public key will be listed. Keyword search is flexible but public key search requires exact match.

PeerNews

←

Share your thoughts with us through peers.

You can share text messages, images and even videos.

💬 Post Text Message

🖼️ Post Image

📹 Post Video

Post page allows users to pick one of the three options that are plain text, picture and video.

Post Message Page

PeerNews

←

Enter Your Title Here..

Enter your text message here..

selected topic1, topic2, topic3

Browse..

Post it !

Topic 4 ▾

+

Post Message Page allows users to write a plain text message. Users can also attach one or more topics related their post. Clicking “Post” broadcasts the message to whole network starting from directly connected peers.

Post Image Page


PeerNews

⏏

⬅

Enter Your Title Here..

Your Image Preview :



selected topic1, topic2, topic3

Browse..

Post it !

Topic 4 ▾

+

Post Image Page allows users to browse their filesystem to select an image file. Image file is used to create a picture post and broadcasted to the network.

Post Video Page

PeerNews

←

Enter Your Title Here..

Your Video Preview :

▶

🔊 🔍

[selected topic1](#), [topic2](#), [topic3](#)

Browse..

Post it !

Topic 4 ▾

+

Post Video Page similarly allows users to create a video post.

View Post Page

PeerNews

←

This is a post title. Hello World !

Hello, this is my first post in PeerNews.

This application is great. I love it.

[topic1](#), [topic2](#), [topic3](#), [First Post Hype](#)

Welcome! We are glad you like it.

Heyyy

Good to see you here !

These are all comments of this post.

Enter your comment here...

Comment

Copy Public Key of Post Owner

↑
18

↓
-5

Users can click the title of a post to view its content. Title, content, topics, comments, upvotes, downvotes are shown on the page. Users can write a comment to this post at the bottom of the page. It is also possible to copy the public key of the post owner.

4. Glossary

Message: Serialized message object that could be a post, comment, vote etc.

Peer: Client applications running on desktop.

Connector: Servers that help establish connection between peers through hole punching.

Feed: The list of all posts filtered according to the user-topic subscriptions.

Public Key: Public id that is shared with other users. Used to verify users.

Private Key: Private id that is not shared with other users. Used to sign posts.

5. References

- [1] "Managing Trust in Peer-to-Peer Systems." *10 Best Practices for Secure Software Development | Security, Data and Privacy | Subject Areas | Publishing and Editorial | BCS - The Chartered Institute for IT*, ITNOWextra, Jan. 2006, www.bcs.org/content/conWebDoc/3059.
- [2] Kamvar, Sepandar D., et al. "The Eigentrust Algorithm for Reputation Management in P2P Networks." *Contents: Using the Digital Library*, ACM, 20 May 2003, dl.acm.org/citation.cfm?id=775242.
- [3] Adam Back, "Hashcash - A Denial of Service Counter-Measure", technical report, August 2002
- [4] Dwork, Cynthia; Naor, Moni (1993). "Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology". CRYPTO'92: Lecture Notes in Computer Science No. 740. Springer: 139–147.