# 國立交通大學

## 網路工程研究所

## 碩士論文

# Pebble2P：一個基於 Pebble 的跨平台點對點部落格系統

# Pebble2P: A Cross-platform P2P Blog System Based on Pebble

研究生：洪偉翔

指導教授：袁賢銘 教授

中華民國九十七年六月

# Pebble2P：一個基於 Pebble 的跨平台點對點部落格系統

學生：洪偉翔　　　　指導教授：袁賢銘

國立交通大學網路工程研究所

## 摘要

部落格系統在近年來非常流行，使用者可以在自己的部落格上發表文章。可是若透過部落格服務提供商（BSP）使用傳統 client/server 架設部落格，可能會因為 BSP 的政策因素導致使用者的部落格資料受到侵害。故有人提出以點對點架構作為架設部落格服務的方案。但與傳統部落格相比，他們的系統傳統部落格的使用者體驗相距甚遠，沒有好的分類文章的方式且沒有解決或只部份解決多媒體內容呈現的問題。本篇論文提出名為 Pebble2P 的解決方案，它結合了傳統部落格和 P2P 部落格平台，使得使用者可以得到與傳統部落格相同的使用經驗並同時保有 P2P 部落格系統的好處。

# Pebble2P: A Cross-platform P2P Blog System Based on Pebble

Student: Wei-Hsiang Hung Advisor: Shyan-Ming Yuan

Institute of Network Engineering

National Chiao Tung University

## Abstract

The blog services become popular in recently years. Web users can post their articles on their blog sites. On the other hand, the users do not have full control of the articles after posted while the blog services are provided by blog service providers (BSPs). So some people propose use peer-to-peer (P2P) technology to establish blog services. However, compare their systems with traditional blog system, the user experience of their systems are not alike traditional blog systems, they have no way to classify user's article and their system does not support or only partially support multimedia contents. This paper proposes a new platform witch try to combine traditional blog systems and P2P blog systems named Pebble2P. Thus, users can get the user experience the same with traditional blog and reserve the benefits of P2P blog systems.

# Acknowledgements

首先最要感謝的是袁賢銘老師兩年來的指導，自由開明的風格讓我受益良多，也讓我們有許多發揮的空間，接觸各方面不同的領域。另外，非常感謝擔任我的口試委員的梁德容老師、洪振偉老師、謝筱齡老師，他們在論文上所給予建議也有很大的幫助。同時，還要謝謝實驗室的秉哲學長、繼弘學長、永威學長、家鋒學長，及所有實驗室的碩班同學和學弟妹們在研究的過程中所給予諸多的指教及鼓勵，並一起為實驗室製造歡樂的氣氛。

最後，感謝要我父母親，有你們的支持與信任我才能專心致力於學習。謹以此文獻給我摯愛的家人。

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Preface

Nowadays, the Internet is becoming indispensable for human activities. The global network infrastructures have been established intensively and increasingly popular in our daily lives. More and more people, companies or other organizations maintain their own web sites. These web sites were built to provide information or services to the users. In early days, building a web site is too difficult to people without related background. With the progressing of the Internet technology, some web service provider provides users an easy way to build their own web sites. One of these services is blog, a contraction of the term "web log".

Although blog service providers (BSP) provide a convenient method to build a blog, there are drawbacks while blog services are built with traditional client/server architecture. So many people try to build their own blog by passing the BSPs. Unfortunately, most users do not have adequate computer knowledge; they are not capable to construct the blogs by themselves.

In recent years, computer power of PC and network bandwidth increased quickly. People are willing to dispense their computing power and share information with each other and peer-to-peer technology provides the possibility. So there were somebody try to use peer-to-peer technology to solve above problem. However, they did not provide some functionalities that normal blog have.

# 1.2 Problem Statement and Motivation

Blog system is one of hot services on the Internet nowadays. It is a kind of communication and becomes popular recent years. There are many blog service providers, for example, Blogger.com and WordPress.com. Use these service can build a blog easily, but there are also drawbacks. The blog contents are provided by authors, but authors cannot own their writing. Those articles are hold by the BSP. Whenever the BSP's policy changes, authors may bear losses of no reason. Therefore, some bloggers, the blog writers, set up blogs by themselves. Setting up a blog server is easy nowadays, but maintaining a blog server is hard to most of people. If the blog server data were broken and without any backup, blogger will lost all their articles.

P2P can solve above problem. Many P2P protocols provide auto replication, when a node on P2P network is failed. So P2P blog system may be a good solution to the maintenance problem. We found two implementations of P2P blog systems which will be introduced in section 2.5, and we found following problems.

1. They all have their own blog content browsers. But the blog content display of those browsers could not display multi-media contents between the text contents of a blog post.
2. They have no ways to classify blog posts of a blog, for example, tags or categories that traditional blog have.

To solve above problems, we decide to modify existed server-based blog software to P2P blog system. In this way, users can use common web browser to browse the blog articles.

Furthermore, Pebble[1] is lightweight, open source blog software, and it is implemented with Java EE. It's small, fast and feature-rich with unrivalled ease of installation and use. The more important feature is that all maintenance and administration can be performed through a web browser. That why we decide to modify Pebble from traditional client/server architecture to P2P architecture.

## 1.3 Outline of the Thesis

In Chapter 2, we discuss the background of peer-to-peer network, Chord protocol, blog system and related works. In Chapter 3, we show a diagram of system overview and describe implementation details of some important component in Pebble2P. Next, in Chapter 4 presents the results of Pebble2P and operations. We compare Pebble2P with another previous P2P blog system in Chapter 5. Finally, we give a conclusion and then talk about the future works in Chapter 6.

[1] http://pebble.sourceforge.net/

# 2 Background

## 2.1 Peer-to-Peer (P2P) Network

The data and computing were centralized in a small number of servers in traditional, and all the clients must be directly dependent on them for data or computing power. This approach is called server-based, or client/server. Peer to peer (P2P) systems use decentralized approaches to sharing distributed resources (e.g. computing, storage, or communication), avoiding storing or maintaining global state. Typically, P2P systems are designed to adapt to nodes entering and leaving the network and to balance resource use among member nodes.[1] There is an intuitive definition of Peer-to-peer is given by Clay Shirkey [30]: "Peer-to-peer is a class of applications that take advantage of resources storage, cycles, content, human presence available at the edges of the Internet.".

There are many successful P2P applications. For example, BitTorrent [5], eMule [4] are very famous P2P file sharing software, and Skype is well known P2P VoIP (Voice over Internet Protocol) software. Furthermore, P2P also have been used in network data storage [21], distributed indexing [20], and domain name service [25]. They show the advantages of P2P network and verify the practicability of P2P network architecture. Besides, Most of P2P network assume that all users on the P2P network are virtuous, but the fact is not. So there were many researches about P2P network security [26][27].

The main characteristics of the P2P systems are the ability to pool together and harness large amounts of resources, self-organization, load balancing, adaptation and fault-tolerance [3].

"There are two classes of P2P overlay networks: Structured and Unstructured" [2]. Structured P2P usually use a distributed hash table (DHT) as a substrate. Such a network use hash function to give resources a hash value, thus nodes can locate the resources such as peers, files, data objects, etc. The efficiency of structured P2P network depends on its design. There are many publications about DHT in the P2P network, i.e., Chord [6], Pastry [7], CAN [8].

Unstructured P2P network can be easily constructed. The network often uses flooding as the mechanism to send queries across the overlay with a limited scope, so the queries may not always be resolved if a peer is looking for rare data shared by only a few other peers.

A P2P network builds on the overlay network above the Internet. There are some definitions of overlay network: "An overlay network is a computer network which is built on top of another network." "Overlay networks can be constructed in order to permit routing messages to destinations not specified by an IP address."[23] "Overlay networks create a structured virtual topology above the basic transport protocol level that facilitates deterministic search and guarantees convergence." [9] So using virtual communication protocol, we can escape from original Internet communication protocol. For example, a structured P2P network maintains a DHT and routing messages to logical address without having IP address. Therefore, the P2P network can work smoothly by its specific protocol.

## 2.2 Chord

Chord [6] is one of structured peer-to-peer network protocol, and Figure 2-1 shows a Chord network example that consisted of 10 nodes storing 5 keys. Every Chord node has a “key” which is the hash function result of the node’s IP address. Such key is mapped to an identifier circle called “Chord ring” with N nodes, and to stand for address of the node. For example, N1, N8, ⋯, N56 in Figure 2-1 are all one of Chord node, and the number after N is the key. If a node want to insert a value, a value can be an address, document, or an arbitrary data item, into network, Chord also use hash function to generate a key and store the data item in the mapped node. We also use Figure 2-1 as example, if a data item’s hash value (key) is 1, it will be stored in N1, and data items with key 2~8 will be stored in N8.

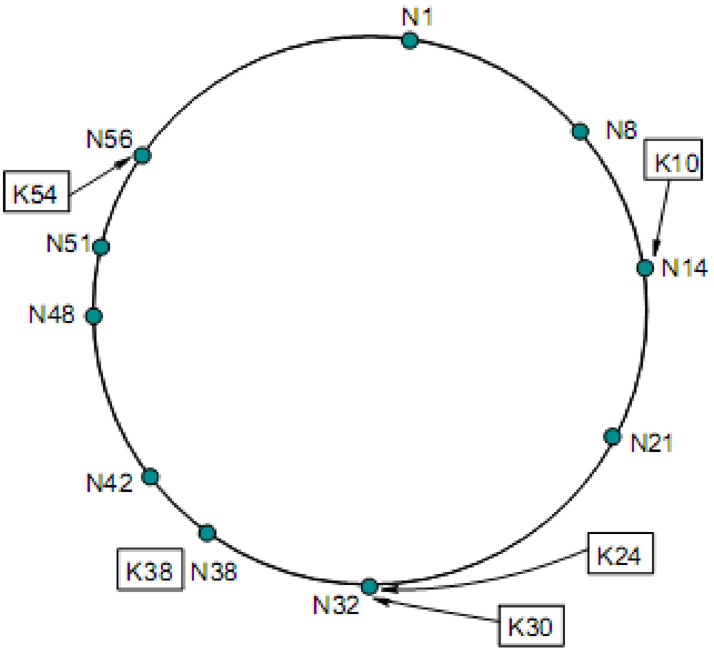
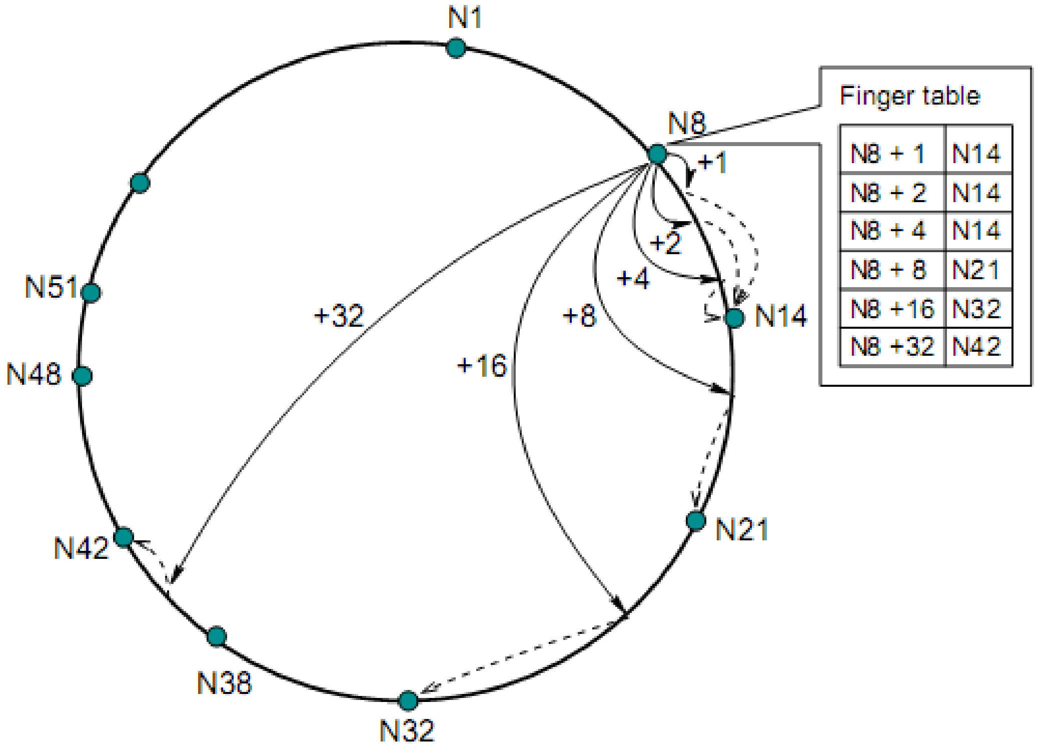


**Figure 2-1 A Chord network topology example**

To accelerate the lookup up operation, every node maintains a finger table which records all successors. The successors of node *n* are nodes which are the first clockwise node of keys value $n + 2^i$-1 in Chord ring, which *n* is the node's key value and $1 \leq i \leq \log \mathrm{N}$. So when a node looks for a value, it can look up its finger table to route lookup message to which node. Figure 2-2 illustrate the finger table for N8, and successors of N8 are N14, N21, N32 and N42.



Finger table

|  |  |
|---|---|
| N8 + 1 | N14 |
| N8 + 2 | N14 |
| N8 + 4 | N14 |
| N8 + 8 | N21 |
| N8 +16 | N32 |
| N8 +32 | N42 |

**Figure 2-2 Chord finger table example**

To quote Frank, "Chord provides an efficient method of locating documents while placing few constraints on the applications that use it." [14] Chord has been implemented in various programming language and we list some them in Table 2-1.

**Table 2-1 Chord implementations**

| **Implementation** | **Language** |
|---|---|
| Chord[2] | C++ |
| Open Chord[3] | Java |
| NChord[4] | C# |
| chordjerl[5] | Erlang |

## 2.3 Blog System

“Blogs (also known as web logs, or weblogs) are web sites that contain frequently updated ‘posts’ with the most recent entry at the top of the page and the previous ones displayed reverse-chronologically.” [10].

Blogs are easy way to produce web page. A blog entry, which is a unit of blog article, is created just by filling in simple forms. A blog entry usually includes article contents (text, pictures, hyperlinks or videos), a permalink, post time, etc. The other users can use comments or trackbacks to interact with blog writer.

What contents are there in blogs? Bonnie [19] give us a good example, “Bloggers are driven to document their lives, provide commentary and opinions, express deeply felt emotions, articulate ideas through writing, and form and maintain community forums.”

People can build their own blog in two ways. The first is building the blog system by self. It includes writing blog programs or blog suit software such as WordPress or Pebble, and renting virtual space in the Internet or using personal

[2] http://pdos.csail.mit.edu/chord/
[3] http://sourceforge.net/projects/open-chord/
[4] http://sourceforge.net/projects/nchord
[5] http://github.com/jashmenn/chordjerl/tree/master

machine to store. The second way is to rely on blog service providers (BSPs) we have mentioned in Chapter 1.

The advantage of the first way is the user has the control of all data. But the users must maintain their own software, hardware and data. So they must have enough knowledge about computer technology. Therefore, it usually is adopted by the people having computer technology. By contrast, the second way makes people build blogs more convenient. Anyone can be a blog writer without handling the problems about system security, data backup and machine updates. So it is the main way to build a blog. Although building a blog through BSP is convenient, there are potential problems. Every blogger writes articles or data on the blog, but the controller is BSP not the author.

More and more bloggers do not want their efforts held in hands of BSP. Some bloggers begin to build blogs by themselves, but it is hard to maintain a server. To maintain a server, there are many things need to be done, for example, data backup is a big problem for many people.

Furthermore, the number of blog in the Internet is multiples of growth [13]. Each user wants to visit more and more blogs and is unable to stand the load. So the RSS [28]/Atom [29] technology is used to track user’s favorite blogs, it eliminates the matters that user has to open the web pages and find new information.

## 2.4 Pebble

Pebble is lightweight, open source blog software written by Simon Brown. Pebble is small, fast and feature-rich with unrivalled ease of installation and use. The more important feature is that all maintenance and administration can be performed through a web browser. Pebble is implemented with Java EE and JavaServer Pages(JSP). It use following technologies: Java Servlets, Spring Framework for whole application, and Acegi Security System for security.

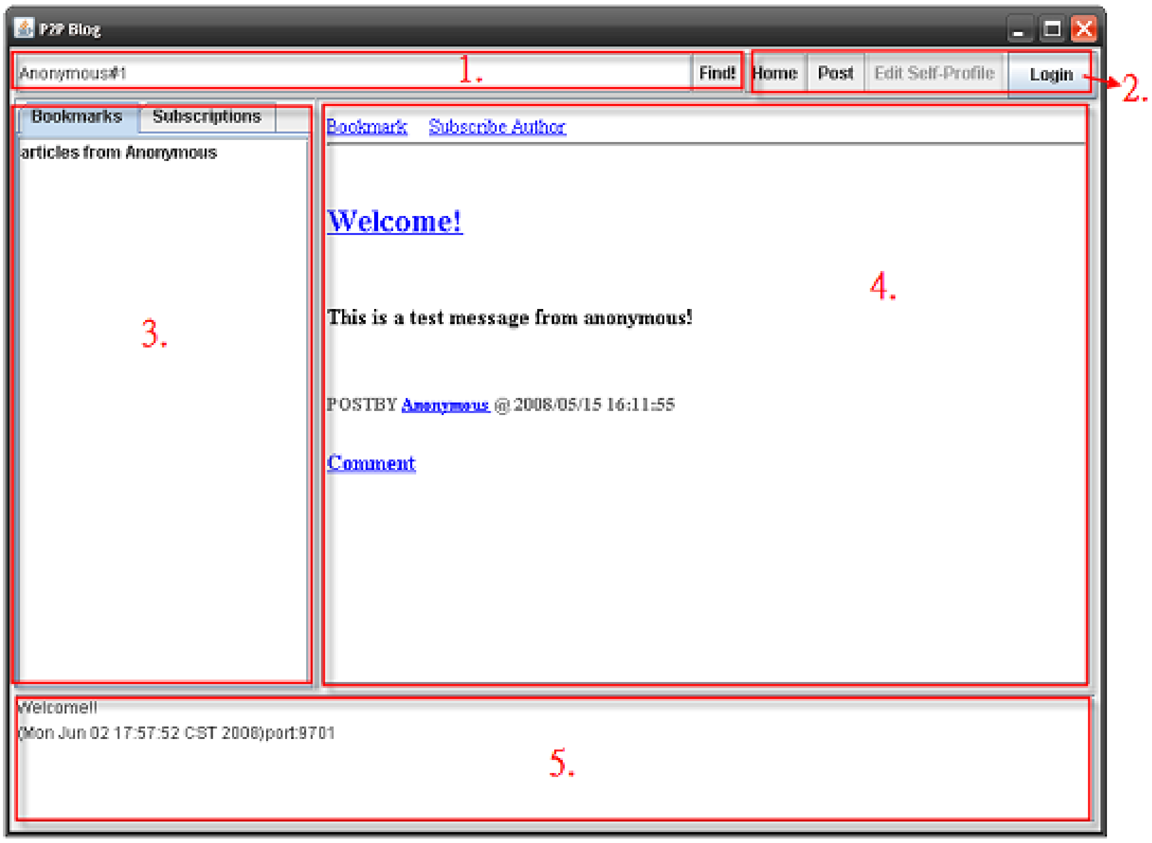
Pebble2P is implemented based on Pebble, so Pebble2P's user interface is the same with Pebble. We just re-write Pebble's data storage strategies from traditional client/server architecture to P2P architecture, and we try to reserve most Pebble's functionalities. So user can experience P2P blog systems just like to visit a common blog site.

## 2.5 Related Work

In this section, we introduce two applications similar with Pebble2P. The first one is P2PBlog which is implemented by our library member before. The other one is P2PmBlog. They both are p2p blog system solution.

### 2.5.1 P2PBlog

P2P Blog[12][24] is a P2P blog application. Figure 2-3 shows the main GUI of P2PBlog. The main characteristic of P2PBlog is that the application use OpenID [18] to resolve the anonymous feature of P2P network. Its GUI can be separated into five areas. Users can find an author or a blog entry in area 1, perform blog functions, such as login, edit user profile, post a blog entry or go to homepage, in area 2. Area 3 shows the user's bookmarks and subscriptions. Area 4 is the blog entry contents area. Finally, area 5 logs the application information.



articles from Anonymous

Welcome!

This is a test message from anonymous!

POSTBY Anonymous @ 2008/05/15 16:11:55

Welcome!!
(Mon Jun 02 17:57:52 CST 2008)port:9701

**Figure 2-3 P2PBlog screenshot**

P2PBlog has following components: Users use this system via the graphical user interface (GUI). GUI communicates with other components except P2P Pub/Sub Module (FreePastry). OpenID is responsible for OpenID integration, every node need to run a Web Server to handle the OpenID authentication. Static Data controls important static resources. Action Functions are the particular capabilities about the application. P2P Pub/Sub Module only manages P2P and Pub/Sub operations.

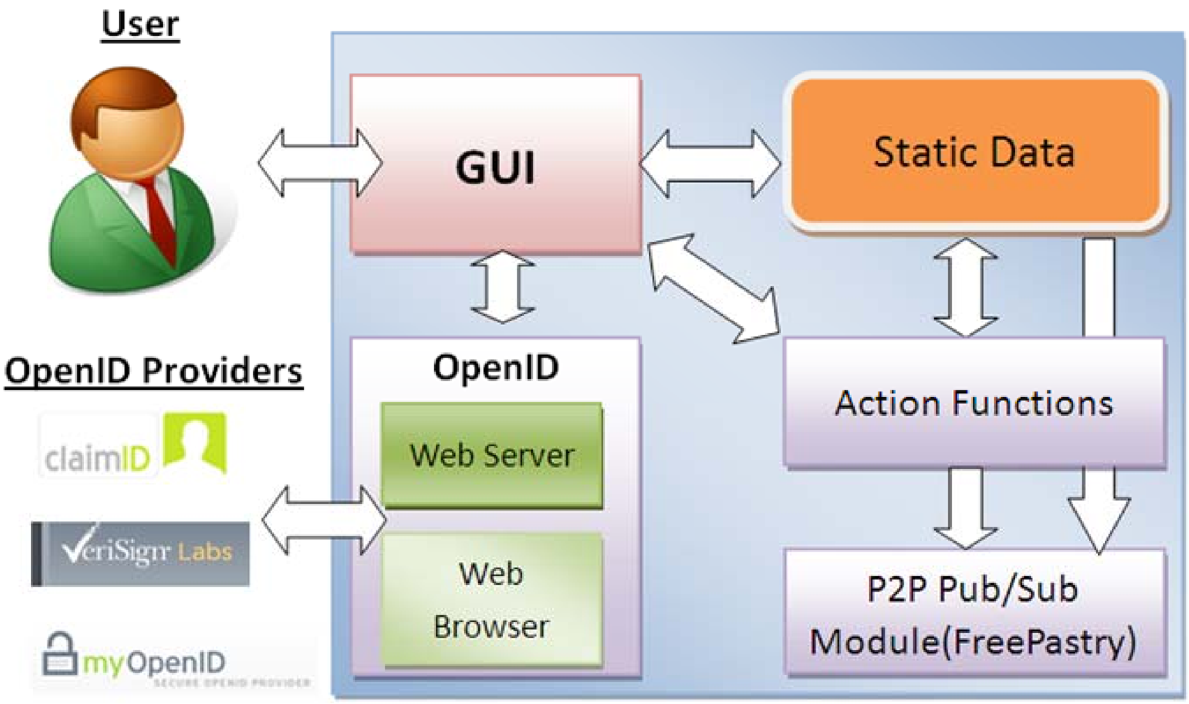


Figure 2-4 P2PBlog System Architecture

### 2.5.2 P2PmBlog

P2PmBlog[6] is one of the efforts in the Pepito project[7]. The system consists of "blog nodes" that form a peer-to-peer network and maintain the blog data, and a number of "blog clients" that can connect to such nodes to get the blog data via XMLRPC. The project provides three kinds of blog clients:

**Blog extractor:** A process that can push external blog data into the P2P network.

**Mobile blog reader:** Let users browse blog data in the P2P network using a mobile phone.

[6] http://www.sics.se/pepito/D5.2/www/p2pmblog.html

[7] http://www.sics.se/pepito/

**Desktop blog reader:** A java application which let user can use desktop computer to connect to a blog node and access to P2PmBlog system.

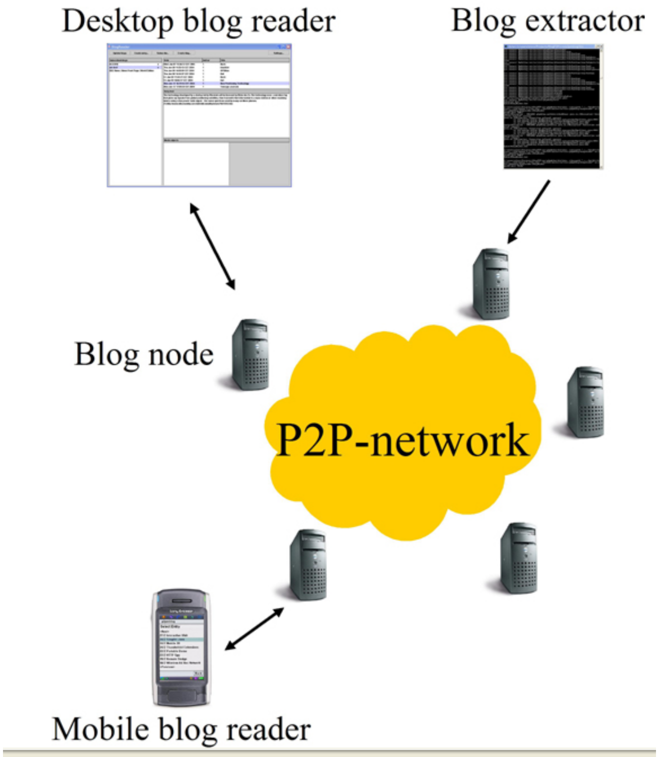


Figure 2-5 P2PmBlog overview

# 3 System Design and Implementation

In this chapter, we present the overview of Pebble2P blog system that includes Pebble2P web application and Pebble2P bootstrap server and both components in Pebble2P.

## 3.1 Overview

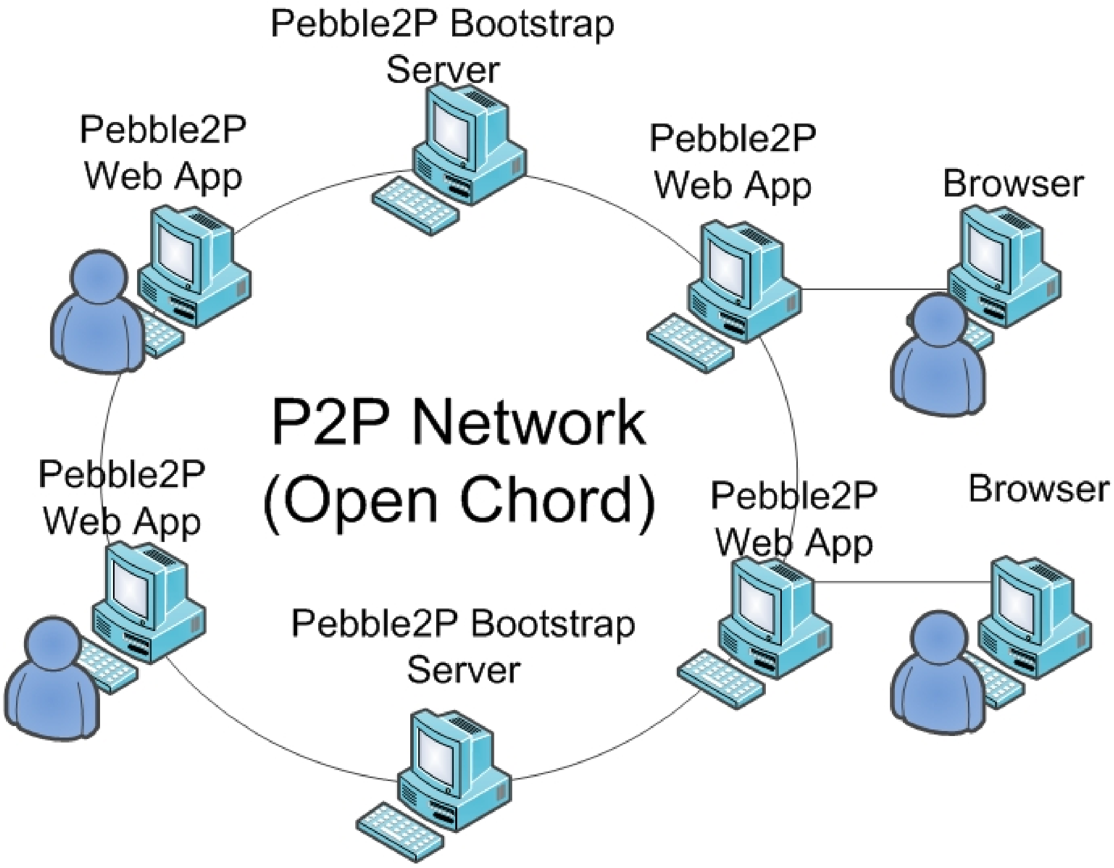


**Figure 3-1 Pebble2P overview**

Pebble2P contains two applications that are Pebble2P Bootstrap Server and Pebble2P Web Application. Pebble2P Bootstrap Server is a Java application and Pebble2P Web Application is a web application based on Pebble, a server based blog software implemented in Java, version 2.3.2. Figure 3-1 shows Pebble2P system will look like virtually, and indicate the relationships between these two components and users.

In the aspect of the users, Users (blog owners/viewers) can use a common web browser such as Internet Explorer (IE), Firefox, to connect to Pebble2P Web Application and do the operations like general blogs. For example, user can access a blog named blog1 through a URL http://[Pebble2P Web Application path]/blog1 and use another URL http://[Pebble2P Web Application path]/blog2 to visit another blog named blog2. If the user is not the owner of the blog, user can just view this blog, register a user account to create a new blog, etc; if user is the owner of the blog, the user can also post a blog entry or administrate the blog. The detail functionality demonstration will be showed in Chapter 4.

We would like to emphasize that both Pebble2P Bootstrap Server and Web Application all are peers in P2P network, and both of them could persistent blog domain data that are data objects of blog such as blogs, user details and blog entries. We will explain both applications in the following sections.

## 3.2 Pebble2P Bootstrap Server

Pebble2P Bootstrap Server is a Java graphical user interface (GUI) application and it is a peer in P2P network. Most of users do not need to run this application except that they want to build their own Pebble2P network. In other words, if they want to build another group of blogs, they can run a Pebble2P Bootstrap Server instance and let other Pebble2P Web Applications join into this network.

The design of Pebble2P Bootstrap Server can be separated into three layers, GUI, P2P abstract layer and Open Chord, showed as in Figure 3-2. Open Chord [15] layer is an implementation of Chord network. P2P Abstract layer provides APIs that let

GUI can control the network behaviors such as create a network, join into an existed network or leave from network. GUI layer let users can control the application, and users also can get the information of this server (peer) and view messages view P2P Abstract Layer. P2P Abstract Layer provides interfaces let GUI can control the server.

The benefit of this design is that if we want to change the network layer (Open Chord), we don't need to rewrite whole application. In other words, we can do that just rewrite the implementation of P2P Abstract Layer.

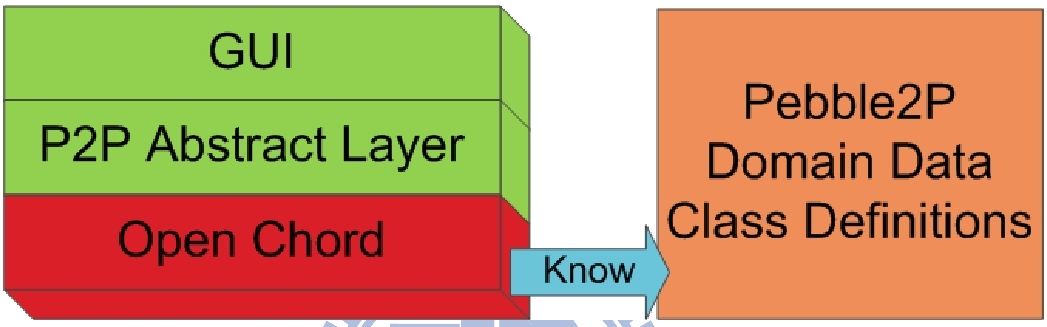


**Figure 3-2 Pebble2P Bootstrap Server 3-layer design**

Moreover, Pebble2P Bootstrap Server must know the implementation of Pebble2P domain classes, like Blog, BlogEntry, UserDetail and all related classes that we will show in next section, because of the limitation of Open Chord.

# 3.3 Pebble2P Web Application

Pebble2P Web Application is a blog web application based on Pebble and it is the main application of our system. Normal users can run this web application on a JSP 2.0/Servlet 2.4 compatible web servers such as Apache Tomcat or GlassFish, and then users can use any web browser like Internet Explorer or Firefox to connect to this application just like they connect to a website.

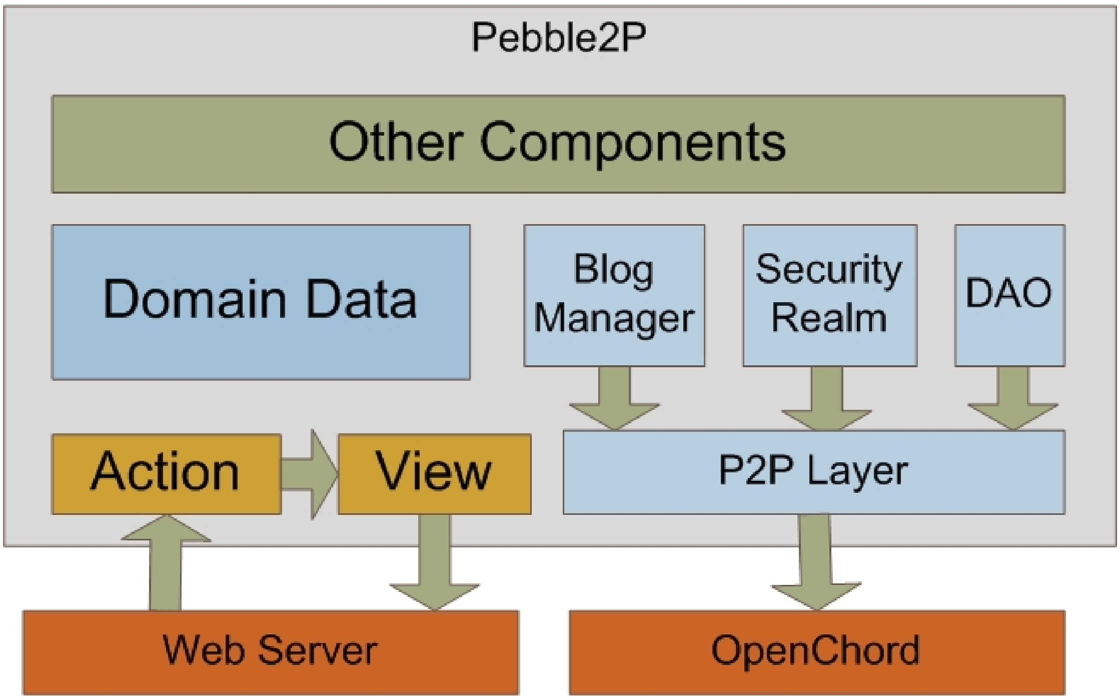


**Figure 3-3 Overview of Pebble2P Web Application**

Most of classes in this application are original implementation of Pebble. We alternate and add some classes to satisfy our requirements. Such classes we separate them into several components that are showed in Figure 3-3, and we will introduce them in following sections.

Besides, in original Pebble implementation, only the users that have administrator permission can add a user or create a new blog. In Pebble2P, we don’t have a role play of administrator. So we must add register functionality to Pebble2P. We will show the flow in Section 3.3.6.

## 3.3.1 Domain Data

The classes we need to put onto the P2P network we called domain data. The domain data classes are Blog, a class stands for a blog, BlogEntry, a class stands for a blog entry, and PebbleUserDetails, a class stands for a Pebble2P user profile. In order to put these objects onto network, we need to make these objects can be serialized. We will show some objects that we need to put onto the P2P network in these classes in following tables.

**Table 3-1 Organization of a Blog object**

| Name | Description |
|---|---|
| id | The blog owner's user name as blog id. |
| years, months, days | Record ids of all blog entries in specific date. |
| properties | Settings of this blog. |
| root category | The root of a category tree. |

**Table 3-2 Organization of a BlogEntry object**

| Name | Description |
|---|---|
| permalink | The permalink of this blog entry. |
| categories | The category that the blog entry falls into. |
| tags | The list of tags for this blog entry. |
| title, body, excerpt | Blog entry contents. |
| comments | the collection of comments for the blog entry |

**Table 3-3 Organization of a PeebleUserDetails object**

| Name | Description |
|---|---|
| username | The user name of this user |
| password | Encrypted password of this user. |
| name, email address, profile, etc. | User data. |

## 3.3.2 P2P Layer

```
public class P2P {
   public P2P();
   void updateUser(PebbleUserDetails pud);
   PebbleUserDetails retrieveUser(String username);
   void updateBlog(Blog blog);
   Blog retrieveBlog (String id);
   void updateBlogEntry(BlogEntry be);
   BlogEntry retrieveBlogEntry (String id);
   void removeBlogEntry(String id);
   void leave();
}
```

**Figure 3-4 Interfaces of P2P Layer**

P2P layer provide methods listed in Figure 3-4 that can manage blogs, blog entries and user details on the Chord network. If other classes want to manage objects on the network, they have no way except for invoking these methods.

If other developers want to modify the implementation of network layer, for example, change the Open Chord to another P2P protocol implementation, they can just rewrite this class without other changes. In general, if we want to implement a new P2P class, we can join into P2P network in constructor and then implement other interfaces. The method leave will be called before the P2P web application stops.

In our implementation, each domain data object has a string key that let us can use the key to insert the domain data object on the Chord network or retrieve the object from the network. The string key format of a blog is “Blog/[username]”, format of a blog entry is “BlogEntry/[blog entry id]”, format of a user is “User/[username]”.

### 3.3.3 Data Access Object (DAO)

Pebble implement an abstract class DAOFactory to set the strategy used to load and store blog entries. Pebble’s default DAO strategy is FileDAOFactroy which stores each blog entries as a XML file in file system.

In Pebble2P, we write a new DAOFactory called P2PDAOFactory. We use it to put blog entries onto and retrieve blog entries from the P2P network.

### 3.3.4 Security Realm

```
public interface SecurityRealm {
   public Collection<PebbleUserDetails> getUsers();

   public PebbleUserDetails getUser(String username);

   public void createUser(PebbleUserDetails pud);

   public void updateUser(PebbleUserDetails pud);

   public void changePassword(String username, String password);

   public void removeUser(String username);
}
```

**Figure 3-5 Interfaces of a security realm**

Security realm is defined by original Pebble. It is used to provide detailed user information and manage users. In other words, other components can manage user details via this class. Figure 3-5 shows the interfaces of a security realm. In order to put user profiles onto P2P network, we implemented the Pebble2PSecurityRealm.

### 3.3.5 Blog Manager

Blog manager is used to manage blog. Blog manager will be created upon the Pebble2P Web Application start, and every Pebble2P Web Application has only one instance. When a Blog manager is created, it will retrieve the Pebble2P default blog at first. Blog manager can start a blog, stop a blog, add a blog or get a blog instance known by this blog manager. For instance, when a user request a blog which is unknown by a blog manager, Pebble2P will retrieve the blog from P2P network, and add the blog into the blog manager. Then let blog manager start the blog. Besides, when Pebble2P Web Application is stopped, it will use blog manager to stop all blogs first.

### 3.3.6 User Register

When a user connects to a Pebble2P Web Application, user will see the homepage of default blog of Pebble2P. Then the user can press register button to register an account. Figure 3-6 shows the flow of user registration.

After user presses the “register” button, Pebble2P will send a register form to user’s browser. And then user fills the form and submits it to Pebble2P. Then Pebble2P checks the user is existed or not. If not, Pebble2P will create new user details and blogs for this user, and upload them to P2P network.

There is a fact that when a user completes registration, Pebble2P will create a blog for the user and use the username as the blog id. In other words, Pebble2P can only create a blog for a user.

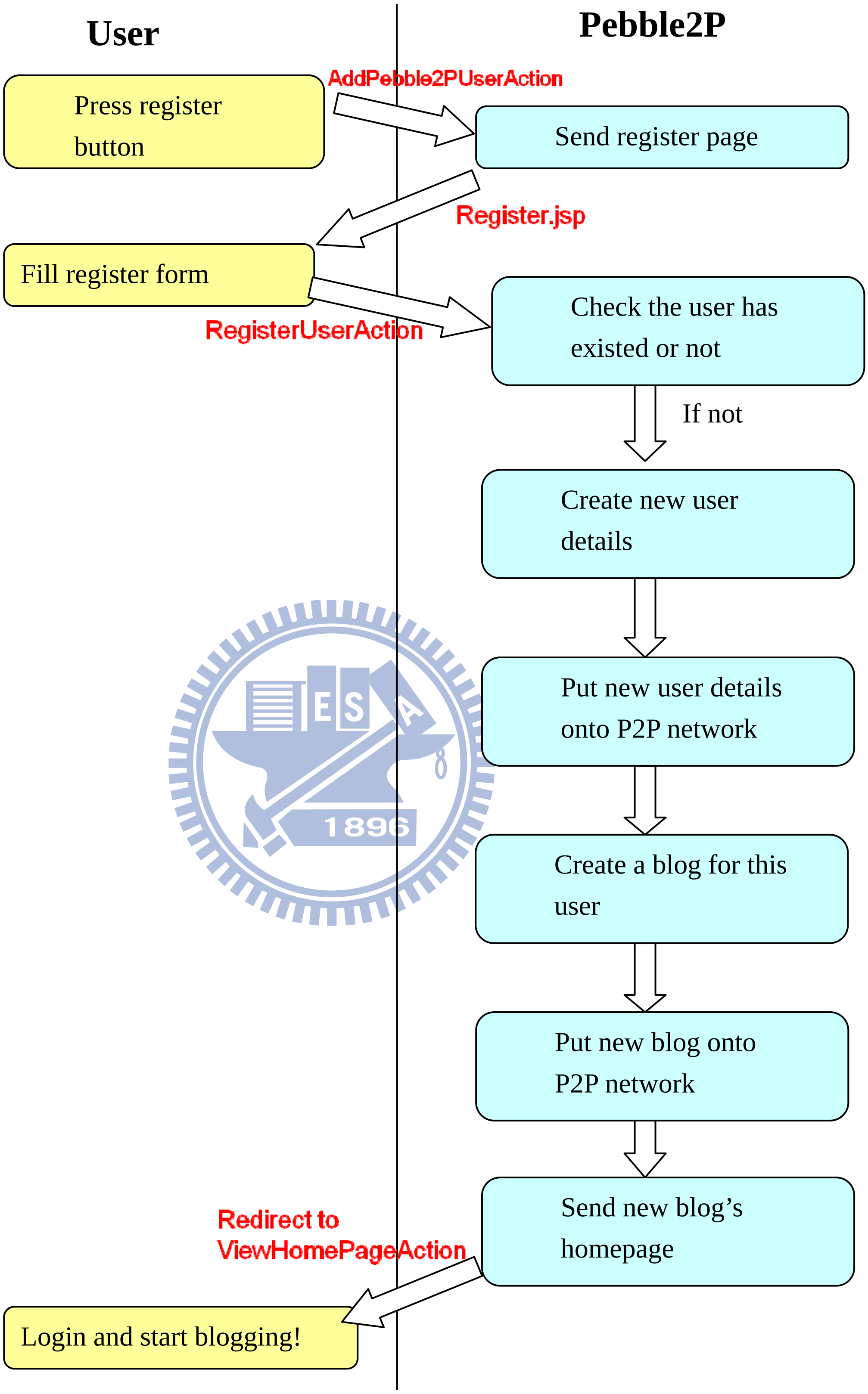

User
Pebble2P
Press register button
AddPebble2PUserAction
Send register page
Register.jsp
Fill register form
RegisterUserAction
Check the user has existed or not
If not
Create new user details
Put new user details onto P2P network
Create a blog for this user
Put new blog onto P2P network
Send new blog's homepage
Redirect to ViewHomePageAction
Login and start blogging!

**Figure 3-6 Flow chart of flow of user registration**

## 3.4 Tools and Libraries

In this program, we use numbers of open source tools and libraries for development. As shown below.

**Table 3-4 Tools and Libraries**

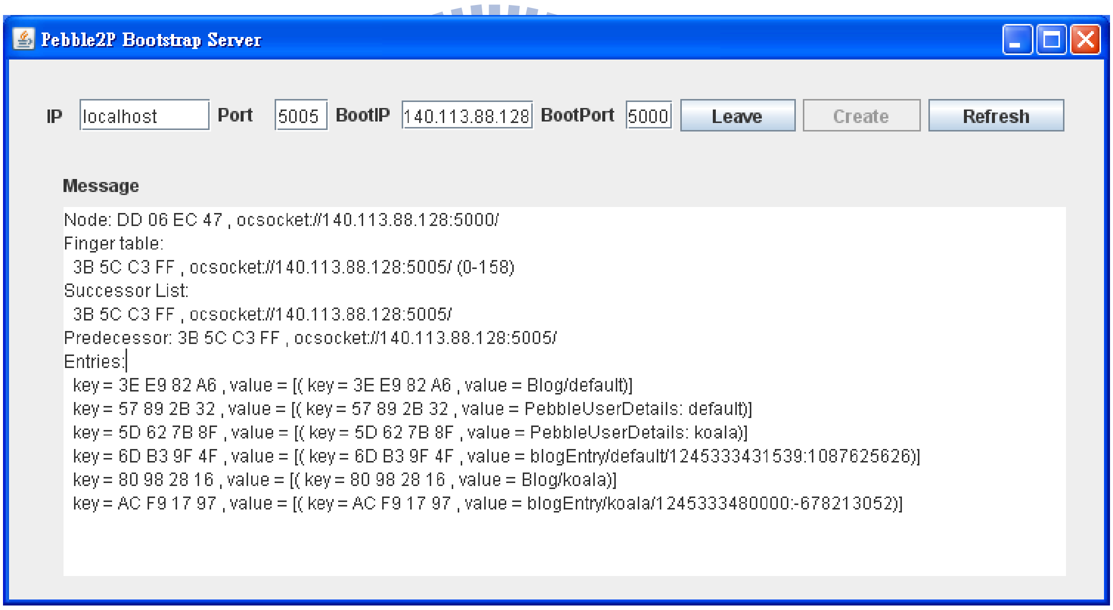
| Name | Usage | Version | License |
|---|---|---|---|
| Java EE [22] | Core | 1.6 | SUN |
| Pebble | Basis | 2.3.2 | BSD License |
| Open Chord [15] | P2P | 1.0.5 | GNU GPL |
| Java Servlet | JSP/Servlet | 2.4 | SUN |
| Spring Framework[8] | Web application | 1.2.8 | Apache Software License |
| Acegi-security for Spring | Security Realm | 1.0.1 | Apache License 2.0 |
| Tomcat | Web server | 6.0.18 | Apache Software License |

[8] http://www.springsource.org

# 4 System Demonstration

## 4.1 Bootstrap Server

The screenshot of Pebble2P Bootstrap Server is showed in Figure 4-1. Users can input those text fields and press a relative button to join into/leave from a network or create a new network. Notice that the "Create" button will use "BootIP" and "BootPort" to create a new P2P network. The "*Refresh*" button will refresh the peer information which is got from Open Chord API.



IP localhost Port 5005 BootIP 140.113.88.128 BootPort 5000

Message

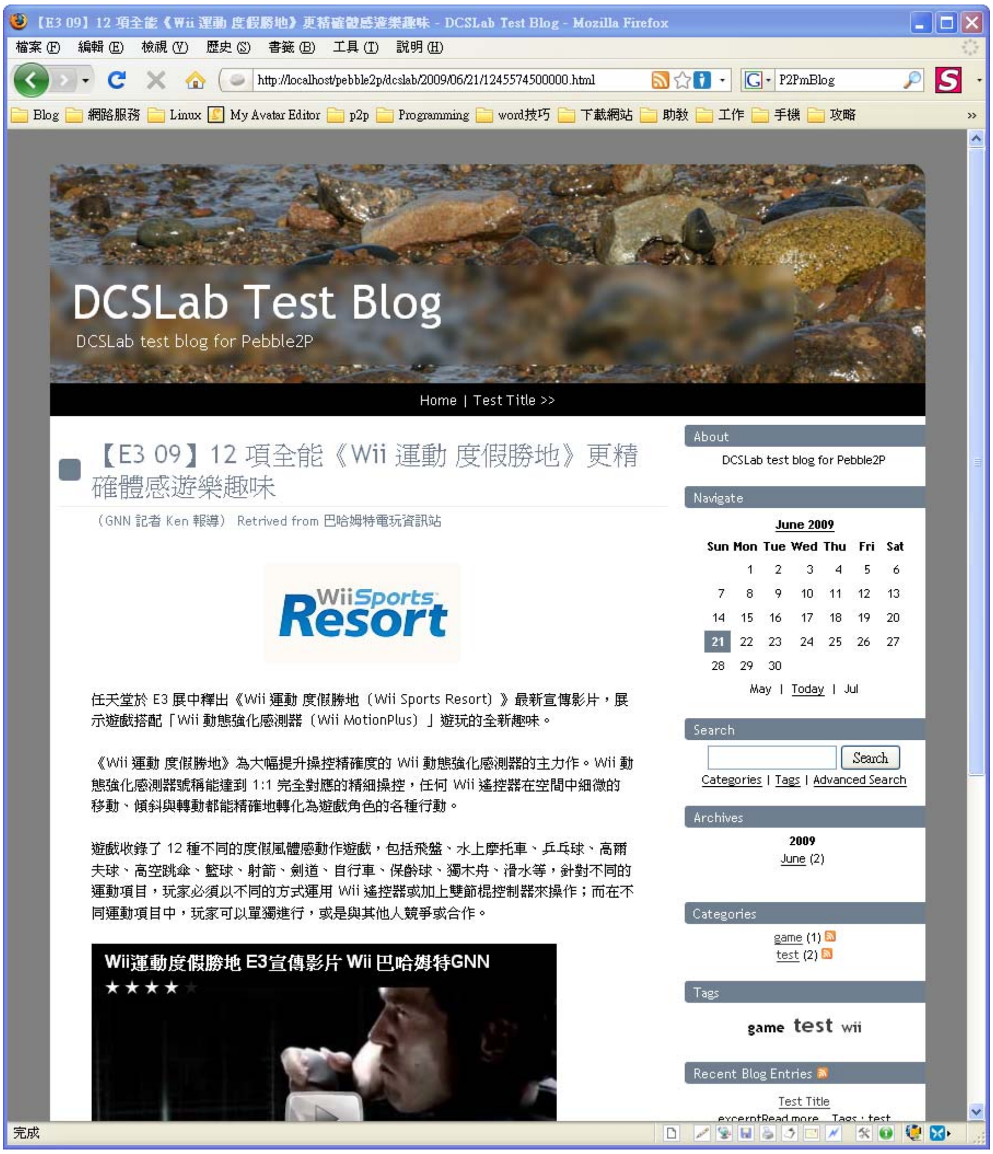
```
Node: DD 06 EC 47 , ocsocket://140.113.88.128:5000/
Finger table:
  3B 5C C3 FF , ocsocket://140.113.88.128:5005/ (0-158)
Successor List:
  3B 5C C3 FF , ocsocket://140.113.88.128:5005/
Predecessor: 3B 5C C3 FF , ocsocket://140.113.88.128:5005/
Entries:
  key = 3E E9 82 A6 , value = [( key = 3E E9 82 A6 , value = Blog/default)]
  key = 57 89 2B 32 , value = [( key = 57 89 2B 32 , value = PebbleUserDetails: default)]
  key = 5D 62 7B 8F , value = [( key = 5D 62 7B 8F , value = PebbleUserDetails: koala)]
  key = 6D B3 9F 4F , value = [( key = 6D B3 9F 4F , value = blogEntry/default/1245333431539:1087625626)]
  key = 80 98 28 16 , value = [( key = 80 98 28 16 , value = Blog/koala)]
  key = AC F9 17 97 , value = [( key = AC F9 17 97 , value = blogEntry/koala/1245333480000:-678213052)]
```

**Figure 4-1 Screenshot of Pebble2P Bootstrap Server**

# 4.2 Web Application

## 4.2.1 Overview

Figure 4-2 shows the screenshot when a user connect to Pebble2P Web Application via web browser. The contents are separated into two columns like traditional blog. The left column shows the blog contents and the right column shows other information, i.e., blog introduction, calendar, archives, tags, categories, login area, etc., of this blog.


【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味 - DCSLab Test Blog - Mozilla Firefox
檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)
Blog 網路服務 Linux My Avatar Editor p2p Programming word技巧 下載網站 助教 工作 手機 攻略
Home | Test Title >>
About
Navigate
Search
Archives
Categories
Tags
Recent Blog Entries
完成

http://localhost/pebble2p/dcslab/2009/06/21/1245574500000.html

P2PmBlog

DCSLab Test Blog

DCSLab test blog for Pebble2P

【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味

（GNN 記者 Ken 報導） Retrived from 巴哈姆特電玩資訊站

任天堂於 E3 展中釋出《Wii 運動 度假勝地（Wii Sports Resort）》最新宣傳影片，展示遊戲搭配「Wii 動態強化感測器（Wii MotionPlus）」遊玩的全新趣味。

《Wii 運動 度假勝地》為大幅提升操控精確度的 Wii 動態強化感測器的主力作。Wii 動態強化感測器號稱能達到 1:1 完全對應的精細操控，任何 Wii 遙控器在空間中細微的移動、傾斜與轉動都能精確地轉化為遊戲角色的各種行動。

遊戲收錄了 12 種不同的度假風體感動作遊戲，包括飛盤、水上摩托車、乒乓球、高爾夫球、高空跳傘、籃球、射箭、劍道、自行車、保齡球、獨木舟、滑水等，針對不同的運動項目，玩家必須以不同的方式運用 Wii 遙控器或加上雙節棍控制器來操作；而在不同運動項目中，玩家可以單獨進行，或是與其他人競爭或合作。

Wii運動度假勝地 E3宣傳影片 Wii 巴哈姆特GNN

DCSLab test blog for Pebble2P

June 2009

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 |  |  |  |  |

May | Today | Jul

Search

Categories | Tags | Advanced Search

2009
June (2)

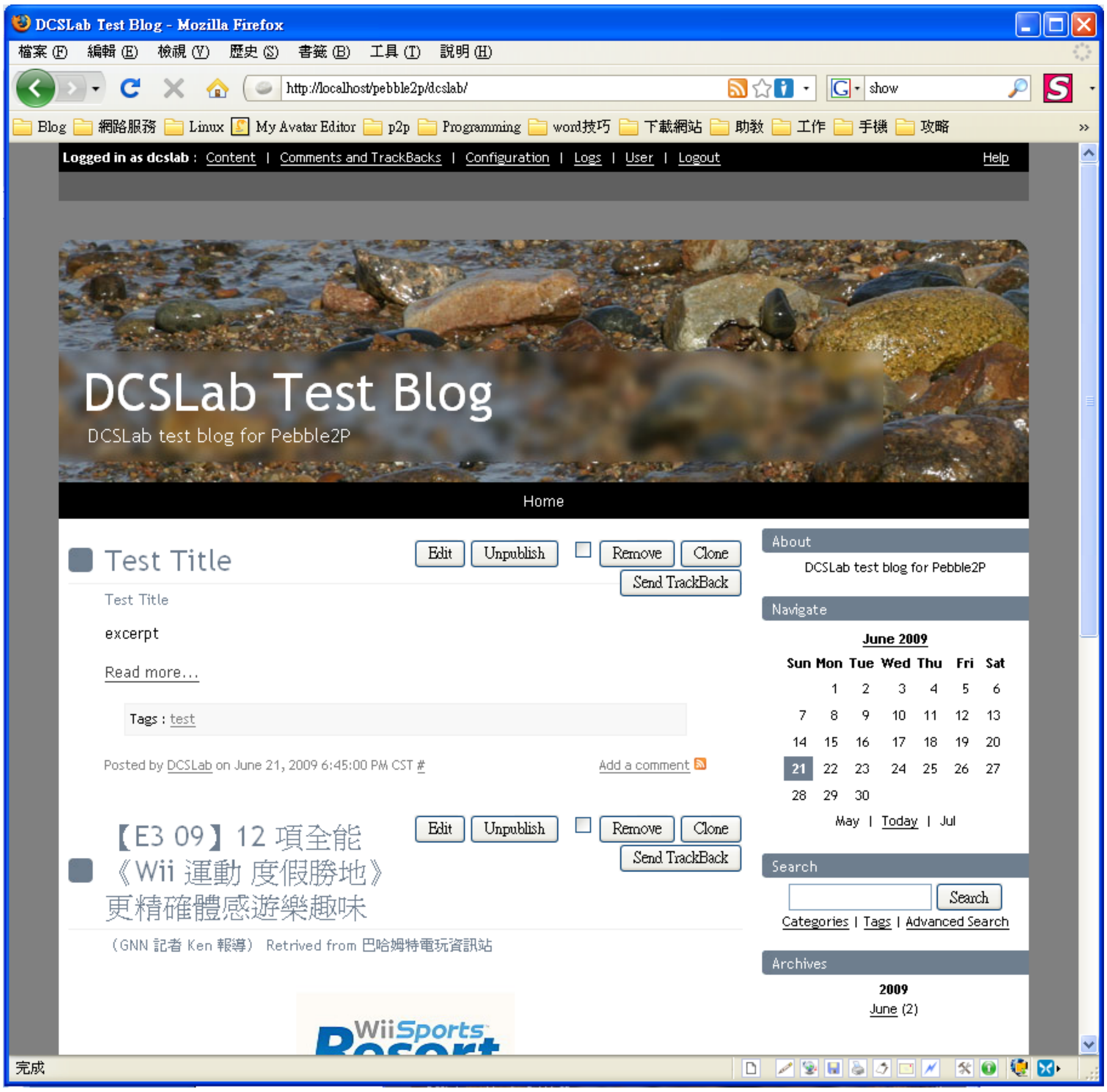
game (1)
test (2)

game test wii

Test Title
excerptRead more Tags : test

Figure 4-2 Pebble2P Web Application screenshot

If a user has logged in, Pebble2P will present an administrative bar. User can use this bar to manage this blog, create a new blog entry, edit user profile, etc. And there are buttons at right side of blog entries' titles. User can use these buttons to manage blog entries. The screen shot is showed in Figure 4-1.


DCSLab Test Blog - Mozilla Firefox
檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)
http://localhost/pebble2p/dcslab/
show
Blog 網路服務 Linux My Avatar Editor p2p Programming word技巧 下載網站 助教 工作 手機 攻略
Logged in as dcslab : Content | Comments and TrackBacks | Configuration | Logs | User | Logout
Help
Home
Edit Unpublish □ Remove Clone Send TrackBack
About
Navigate
Search
Categories | Tags | Advanced Search
Archives
完成

DCSLab Test Blog

DCSLab test blog for Pebble2P

Test Title

Test Title

excerpt

Read more…

Tags : test

Posted by DCSLab on June 21, 2009 6:45:00 PM CST #

Add a comment

【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味

（GNN 記者 Ken 報導） Retrived from 巴哈姆特電玩資訊站

DCSLab test blog for Pebble2P

| June 2009 | | | | | | |
|---|---|---|---|---|---|---|
| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | | | | |

May | Today | Jul

2009

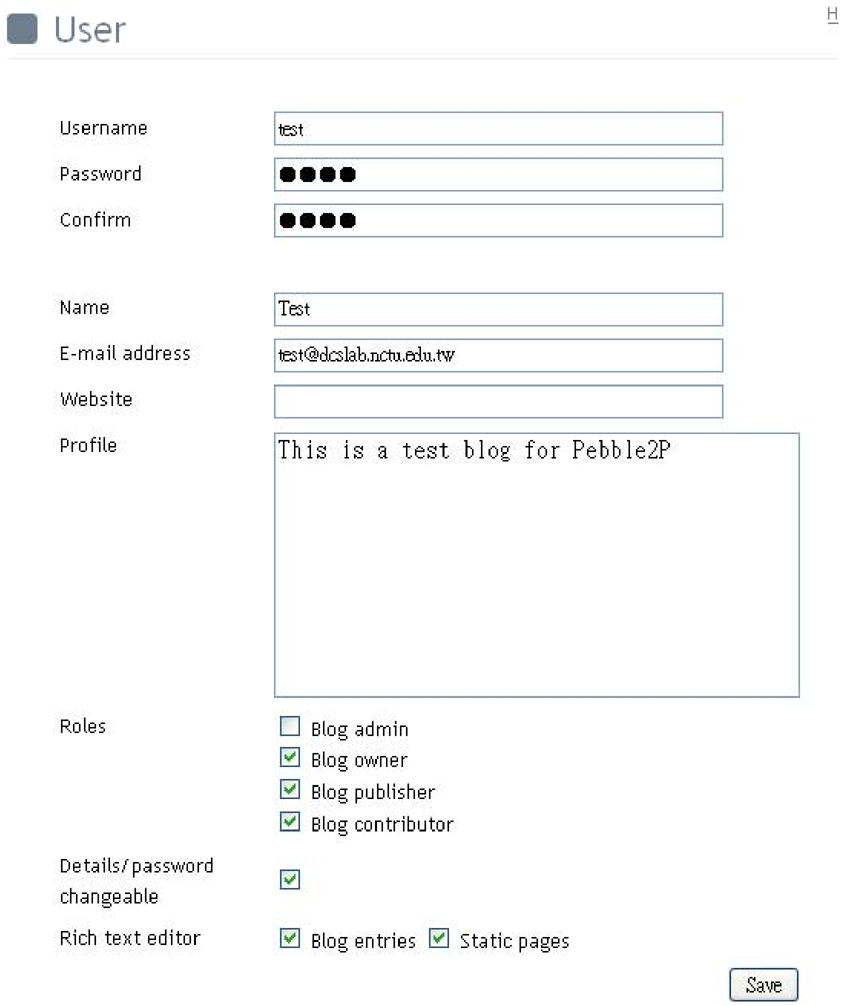
June (2)

**Figure 4-3 Screenshot of user logged in**

Because Pebble2P Web Application is based on Pebble, most of user interfaces in Pebble2P are the same with Pebble. Incidentally, for all partial screenshots in this chapter, they are presented in left column if not be mentioned.

### 4.2.2 Register

User can use the “*Register*” button in the login area of right column. When “Register” is pressed, Pebble2P will show a register form. User must fill this form to register a username and to create own blog.



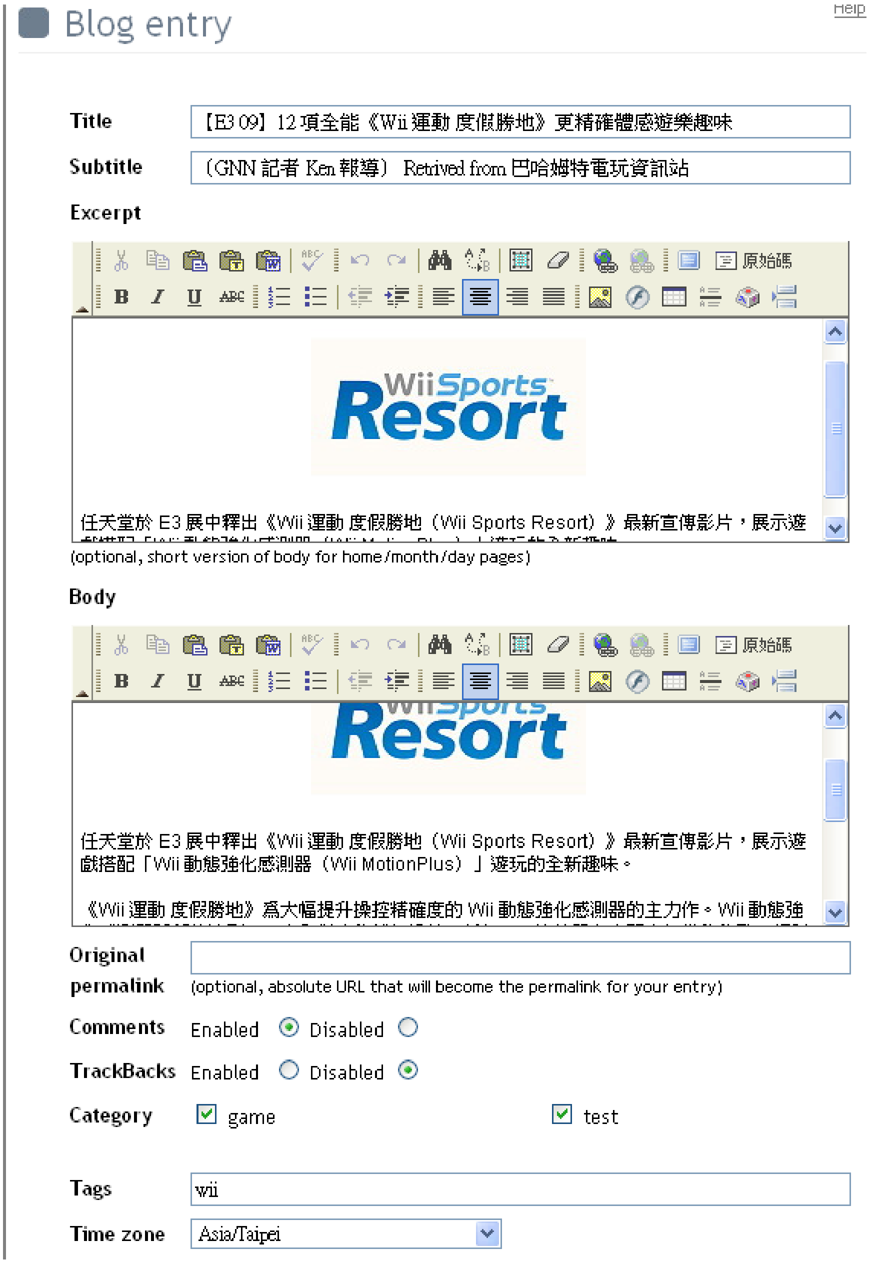
User

| | |
|---|---|
| Username | test |
| Password | ●●●● |
| Confirm | ●●●● |
| Name | Test |
| E-mail address | test@dcslab.nctu.edu.tw |
| Website | |
| Profile | This is a test blog for Pebble2P |
| Roles | □ Blog admin<br>☑ Blog owner<br>☑ Blog publisher<br>☑ Blog contributor |
| Details/password changeable | ☑ |
| Rich text editor | ☑ Blog entries ☑ Static pages |

**Figure 4-4 Partial screenshot of user registration**

### 4.2.3 Create a Blog Entry

User can move mouse on “*Content*” on the administrative bar, and click “*New blog entry*” to create a new blog entry. Use can create a new blog entry through fill the form showed in Figure 4-5. User can input blog entry contents, select categories, add tags to this entry, etc.



Blog entry

Help

Title 【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味

Subtitle （GNN 記者 Ken 報導） Retrived from 巴哈姆特電玩資訊站

Excerpt

任天堂於 E3 展中釋出《Wii 運動 度假勝地（Wii Sports Resort）》最新宣傳影片，展示遊

(optional, short version of body for home/month/day pages)

Body

任天堂於 E3 展中釋出《Wii 運動 度假勝地（Wii Sports Resort）》最新宣傳影片，展示遊戲搭配「Wii 動態強化感測器（Wii MotionPlus）」遊玩的全新趣味。

《Wii 運動 度假勝地》為大幅提升操控精確度的 Wii 動態強化感測器的主力作。Wii 動態強

Original permalink (optional, absolute URL that will become the permalink for your entry)

Comments Enabled ● Disabled ○

TrackBacks Enabled ○ Disabled ●

Category ☑ game ☑ test

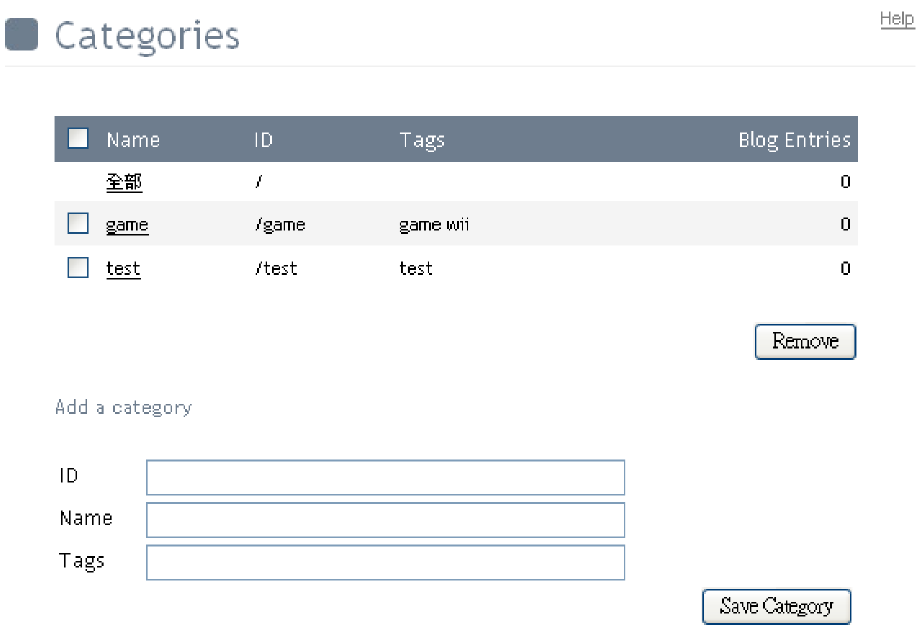
Tags wii

Time zone Asia/Taipei

**Figure 4-5 Create a blog entry**

### 4.2.4 Manage Categories

Categories are used to classify blog entries. User can manage categories of a blog through clicking “*Content*” → “Categories” on the administrative bar. Figure 4-6 shows the interface that user can manage categories of a blog. User can tag not only blog entries but also categories.



Categories

Help

| ☐ | Name | ID | Tags | Blog Entries |
|---|---|---|---|---|
| | 全部 | / | | 0 |
| ☐ | game | /game | game wii | 0 |
| ☐ | test | /test | test | 0 |

Add a category

ID

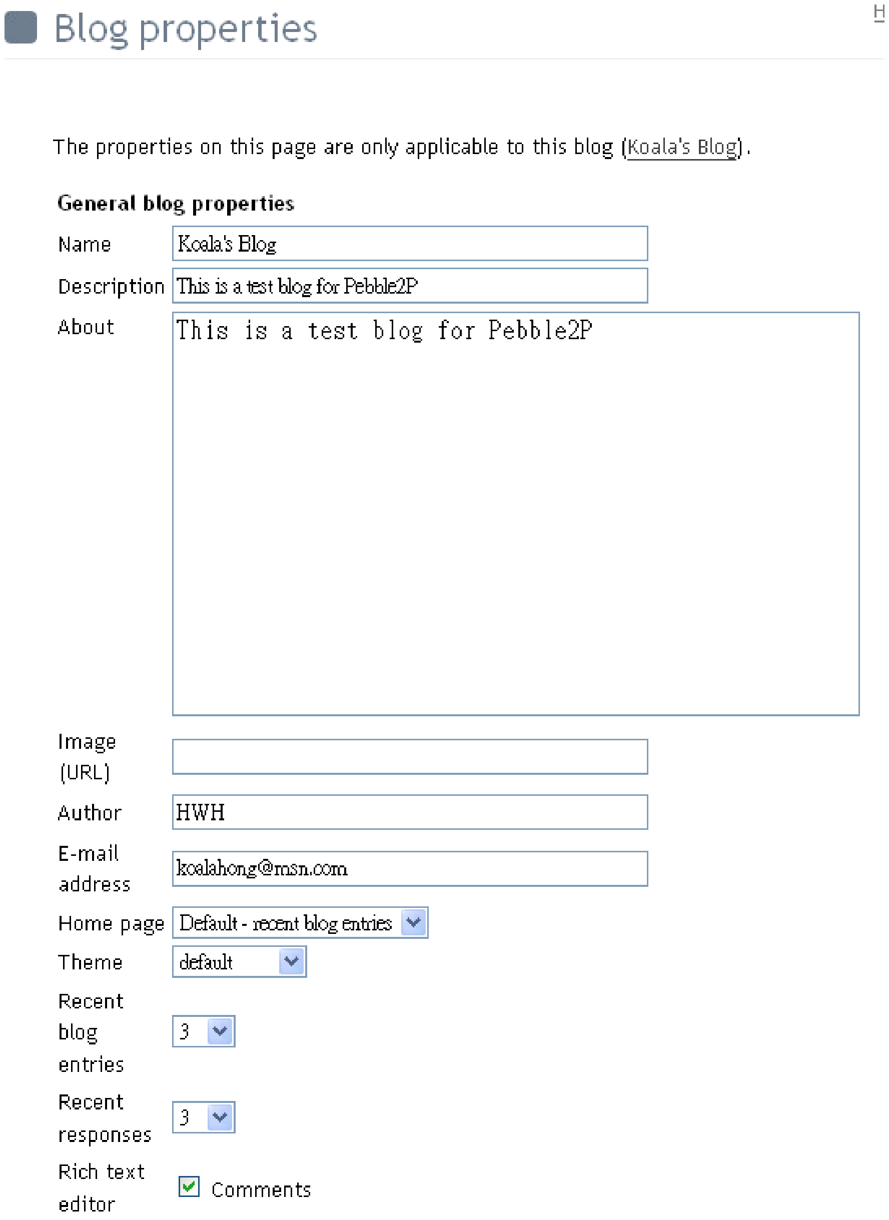
Name

Tags

**Figure 4-6 Partial screenshot of categories management**

### 4.2.5 Blog Setting

User can use the form showed in Figure 4-7 to configure blog proerties. This form is located at “*Configuration*” → “Properties” on administrative bar. Because of the limitation of Open Chord, user cannot change the blog themes.



Blog properties

The properties on this page are only applicable to this blog (Koala's Blog).

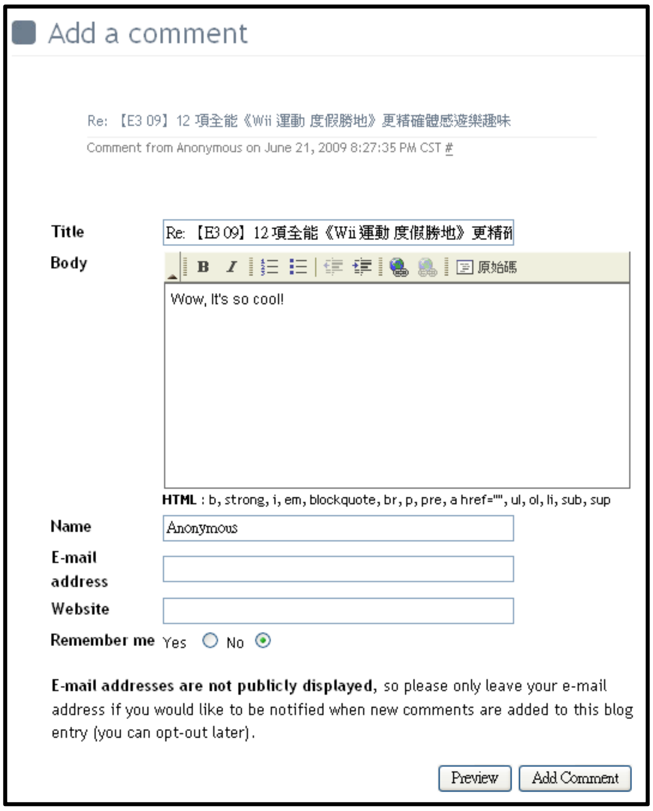
General blog properties

|   |   |
|---|---|
| Name | Koala's Blog |
| Description | This is a test blog for Pebble2P |
| About | This is a test blog for Pebble2P |
| Image (URL) |  |
| Author | HWH |
| E-mail address | koalahong@msn.com |
| Home page | Default - recent blog entries |
| Theme | default |
| Recent blog entries | 3 |
| Recent responses | 3 |
| Rich text editor | ☑ Comments |

Figure 4-7 Partial screenshot of blog configuration

## 4.2.6 Comment

A visitor can add comments to a blog entry. Figure 4-8 shows the screenshot of adding a comment to a blog entry. When a comment is added, it would not be showed at the bottom of a blog entry until the blog owner approves it (Figure 4-9).



Add a comment

Re: 【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味

Comment from Anonymous on June 21, 2009 8:27:35 PM CST #

**Title** Re: 【E3 09】12 項全能《Wii 運動 度假勝地》更精确

**Body** Wow, It's so cool!

**HTML** : b, strong, i, em, blockquote, br, p, pre, a href="", ul, ol, li, sub, sup

**Name** Anonymous

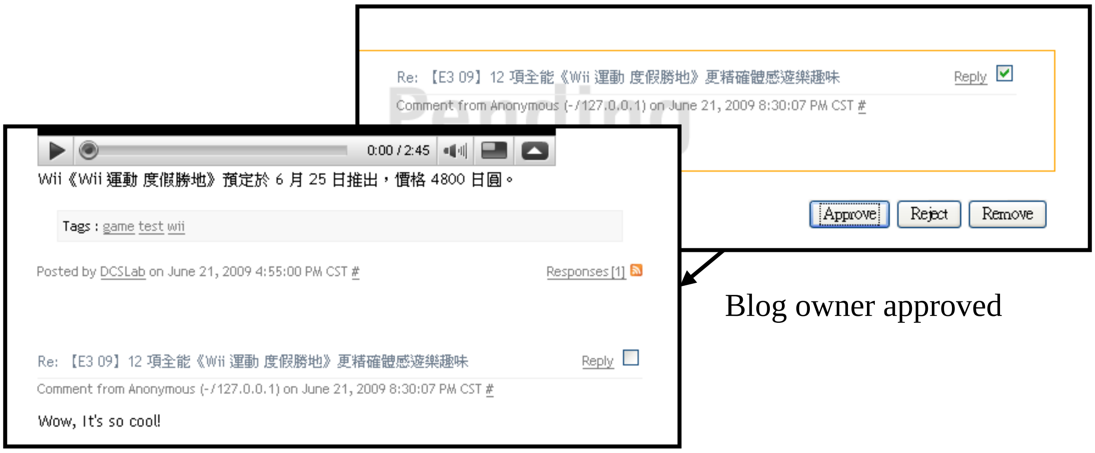
**E-mail address**

**Website**

**Remember me** Yes ○ No ●

**E-mail addresses are not publicly displayed,** so please only leave your e-mail address if you would like to be notified when new comments are added to this blog entry (you can opt-out later).

**Figure 4-8 Partial screenshot of adding a comment**



Re: 【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味 Reply ☑

Comment from Anonymous (-/127.0.0.1) on June 21, 2009 8:30:07 PM CST #

Wii《Wii 運動 度假勝地》預定於 6 月 25 日推出，價格 4800 日圓。

Tags : game test wii

Posted by DCSLab on June 21, 2009 4:55:00 PM CST # Responses [1]

Re: 【E3 09】12 項全能《Wii 運動 度假勝地》更精確體感遊樂趣味 Reply □

Comment from Anonymous (-/127.0.0.1) on June 21, 2009 8:30:07 PM CST #

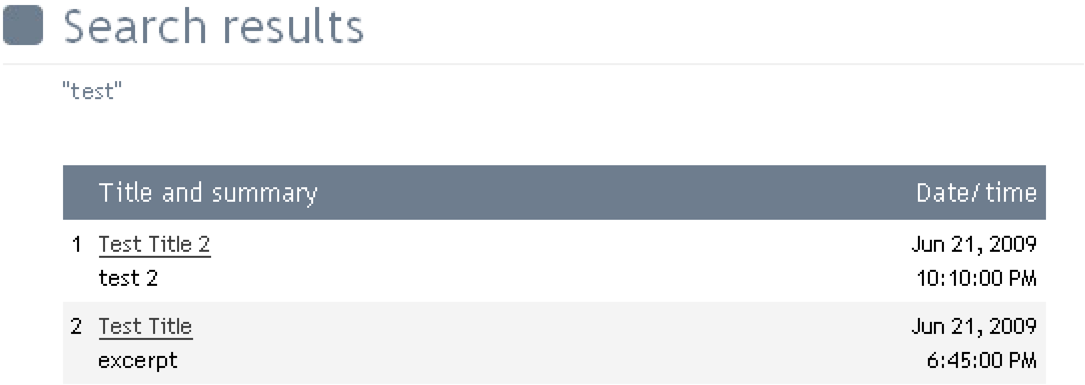
Wow, It's so cool!

Blog owner approved

**Figure 4-9 Approve a new comment**

### 4.2.7 Keyword Search

User can use search area in right column to search whole blog. Figure 4-10 shows an example result of keyword search.



Search results

"test"

| | Title and summary | Date/time |
|---|---|---|
| 1 | Test Title 2<br>test 2 | Jun 21, 2009<br>10:10:00 PM |
| 2 | Test Title<br>excerpt | Jun 21, 2009<br>6:45:00 PM |

**Figure 4-10 Result of keyword search**

# 5 Comparison

In this chapter, we compare Pebble2P with P2P Blog and P2PmBlog in following table:

**Table 5-1 Comparison with P2P Blog and P2PmBlog**

| | **Pebble2P** | **P2P Blog** | **P2PmBlog** | **Traditional blog (C/S Blog)** |
|---|---|---|---|---|
| **P2P Library** | Open Chord | FreePastry | P2PS [16] | **N/A** |
| **P2P protocol** | Chord | Pastry | Tango [17] | **N/A** |
| **Registration** | To register on system | OpenID | To register on system | To register on system |
| **Browser** | Web browser | Custom windows application | PC or Mobile client | Web browser |
| **Multimedia content** | Yes | No | independent column | Yes |
| **Keyword search** | Yes | No | No | Yes |
| **Article classify** | archives, tags, categories | No | No | archives, tags, categories |
| **Subscription** | RSS | Scribe | RSS, Atom | RSS, Atom |
| **Platform** | Cross platform | Windows only | Cross platform | N/A |

Although P2P Blog uses OpenID to perform registration, OpenID accounts are usually too long to remember. Different with other two systems must use their own browsers; Pebble2P can use a common web browser to view a blog. So users can post a blog entry with multimedia contents like pictures or videos. Pebble2P can archive blog entries by date automatically and users can use tags and categories to classify article like traditional client/server blog systems in Pebble2P. All three implementations have subscription mechanism. Although all three P2P blog systems are implemented in Java, P2P Blog use an embedded browser based on Internet Explorer, so P2P Blog can run on Windows only. On the other hand, Pebble2P and P2PmBlog are cross-platform.

In Table 5-1, we can find that Pebble2P's functionalities and user experience is more like with traditional blog. The main difference between Pebble2P and traditional blog is Pebble2P use P2P technology to store the blog data.

# 6 Conclusion and Future Works

## 6.1 Conclusion

This paper proposes another P2P blog system named Pebble2P to improve the user experience in P2P blog system. Users of Pebble2P can simply use a web browser, and then connect to Pebble2P Web Application to start blogging and users can present the contents of articles in not only text but multimedia contents, i.e., pictures, videos. The user experience of Pebble2P is Pebble2P also reserved the advantages that other P2P blog systems have, for example, distributed architecture, auto replication, protecting human right against BSPs. Besides, Pebble2P is a platform independent P2P blog system.

However, Pebble2P still has drawbacks. Although Pebble2P based on Pebble, some of Pebble's functionalities, i.e., user custom plug-in, file uploading, cannot be integrated in Pebble2P. P2P network needs enough users to keep the network operation steady. Moreover, the system security is also a problem on each host in Pebble2P.

## 6.2 Future Works

We shall settle following problems in near future:

1. Users of Pebble can write their own plug-in to add additional functionalities. But Open Chord does not support remote class loading so far. So users can only use Pebble official plug-ins build-in Pebble. We will wait for new release of Open Chord or implement our own remote class loader by ourselves.
2. Spring Framework 2.0 or newer versions provide some API to support OpenID authentication. We can upgrade Spring Framework in Pebble2P to support OpenID authentication.
3. Pebble2P do not support file uploading, so the multimedia content should depend on other service providers such as Youtube[9], Flickr[10]. We will implement this part if we find a good mechanism to divide large files to several fragments.
4. The keyword search functionality of Pebble2P is inherited from Pebble. It needs an indexer to index whole blog. When the number of blog entries grows up, it will be inefficient. We will try to improve keyword with P2P mechanism.

[9] http://www.youtube.com/

[10] http://www.flickr.com/

# References

[1] Russ Rew, " FISHING FOR DATA, PIER TO PIER", 20th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, Seattle, CA, USA, January 2004

[2] K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim. "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes", Communications Surveys & Tutorials, IEEE, 2005.

[3] Datta, A.K., Gradinariu, M., Raynal, M., Simon, G. "Anonymous publish/subscribe in p2p networks", Proceedings of the 17th International Symposium on Parallel and Distributed Processing, pp.74.1, Washington, DC, USA: IEEE Computer Society, 2003.

[4] Y. Kulbak and D. Bickson, "The emule protocol specification," 2005.

[5] J. Pouwelse, P. Garbacki, D. Epema and H. Sips, "The Bittorrent P2P File-Sharing System: Measurements and Analysis", Peer-to-Peer Systems IV, pp. 205-216, 2005.

[6] Ion Stoica , Robert Morris , David Karger , M. Frans Kaashoek , Hari Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications", Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, p.149-160, San Diego, California, United States, August 2001.

[7] Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg,

Germany, pp. 329-350, November, 2001.

[8] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. “A Scalable Content-Addressable Network”, In Proceedings of ACM SIGCOMM 2001.

[9] Diego Doval and Donal O’Mahony. “Overlay networks: A scalable alternative for P2P.” IEEE Internet computing 7:44, pp. 79-82, IEEE Computer Society, 2003.

[10] Mark Brady. “Blogging: personal participation in public knowledge building on the web”, Chimera Working Paper 2005-02>. Colchester, England, February 2005.

[11] Paul-Alexandru Chirita, Stratos Idreos, Manolis Koubarakis, Wolfgang Nejdl. “Designing Semantic Publish/Subscribe Networks Using Super-Peers.”, Semantic Web and Peer-to-Peer. Springer Verlag, 2005.

[12] Chen-Pu Lin, Yung-Wei Kao, Shyan-Ming Yuan. “A P2P Blog System with OpenID Integration”, Third 2008 International Conference on Convergence and Hybrid Information Technology, 2008.

[13] © Technorati, Inc. “State of the Blogosphere / 2008”, Retrieved June 9, 2009, from http://technorati.com/blogging/state-of-the-blogosphere/

[14] Frank Dabek, Emma Brunskill, M. Frans Kaashoek, David Karger, Robert Morris, Ion Stoica, Hari Balakrishnan, "Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service.", Eighth Workshop on Hot Topics in Operating Systems, pp.0081, 2001.

[15] Sven Kalle and Karsten Loesing.“Open Chord version 1.0.4 User’s Manual”, 2007

[16] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, John Cowan, ed. XML 1.1 (Second Edition), W3C Recommendation, 16 August 2006.

[17] Carton, B., Mesaros, V. “Improving the scalability of logarithmic-degree DHT-based peer-to-peer networks”, Euro-Par-International Conference on Parallel Processing 2004.

[18] David Recordon, Drummond Reed. “OpenID 2.0: a platform for user-centric identity management”, Proceedings of the second ACM workshop on Digital identity management, pp. 11-16, Alexandria, Virginia, USA, 2006.

[19] Bonnie A. Nardi, Diane J. Schiano, Michelle Gumbrecht, and Luke Swartz. “Why We Blog”, COMMUNICATIONS OF THE ACM December 2004/Vol. 47, No. 12, pp. 41-46, December 2004.

[20] I. Stoica, D. Adkins, S. Zhuang, S. Shenker and S. Surana, "Internet indirection infrastructure," in SIGCOMM '02: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 73-86 , 2002.

[21] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris and I. Stoica, "Wide-area cooperative storage with CFS," SIGOPS Oper. Syst. Rev., vol. 35, pp. 202-215, 2001.

[22] Eric Armstrong, Jennifer Ball, Stephanie Bodoff, Debbie Bode Carson, Ian Evans, Dale Green, Kim Haase, Eric Jendrock. “The J2EETM 1.4 Tutorial. For Sun Java System Application Server Platform Edition 8.2”, December 2005.

[23] Wikipedia (n. d.). Overlay network. Retrieved June 6, 2009, from http://en.wikipedia.org/wiki/Overlay_network

[24] Chen-Pu Lin. “A P2P Blog System with OpenID Integration”, 交通大學，碩士論文，民國 97 年。

[25] Russ Cox, Athicha Muthitacharoen, Robert T. Morris. “Serving DNS using a Peer-to-Peer Lookup Service”, the proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, MA, March, 2002.

[26] Emil Sit and Robert T. Morris. “Security Considerations for Peer-to-Peer Distributed Hash Tables”, the proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, MA, March, 2002.

[27] Dan S. Wallach. “A Survey of Peer-to-Peer Security Issues.”, Lecture Notes in Computer Science, volume 2609, Springer Berlin / Heidelberg, January 2003.

[28] Harvard, “RSS 2.0 Specification.”, January 2003, retrieved from http://cyber.law.harvard.edu/rss/rss.html

[29] Mark Nottingham and Robert Sayre. “RFC 4287: The Atom Syndication Format”, December 2005.

[30] A. Oram, editor. Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O’Reilly & Associates, March 2001.