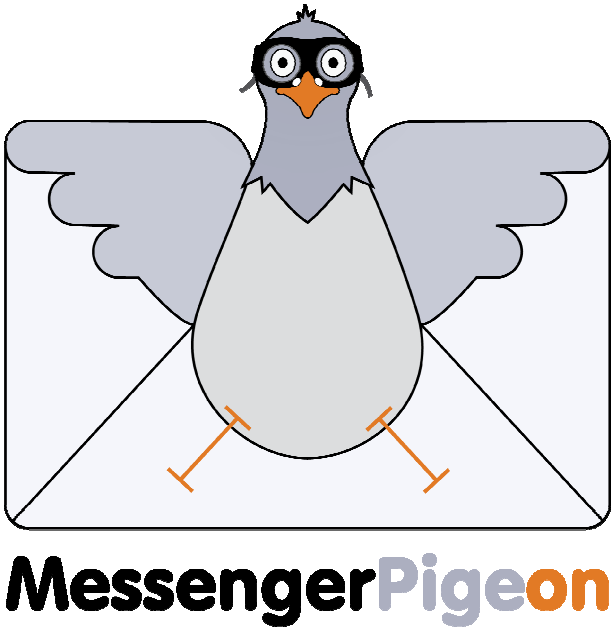
New Bulgarian University

Department: Informatics

BACHELOR THESIS:
Messenger-Pigeon



Submitted by:
Predrag Tasevski
Major: Informatics
Student No: F27024
E-mail: stepcellwolf@gmail.com

Advisor
Doc. Nikolai Gadzev

February 2009,
Sofia

ABSTRACT

The idea of developing Java LAN Massager came from the everyday using instant messaging tools, to create a small application with client-server architecture that can have a whole range of availability to the users and can run in no requirements of operating systems, and architecture of their systems (hardware and software).

To be used in any company that does not allow an instant messenger and prefers to use a internal company messenger to deliver the important message to the floor or the company, to be easy and simple to install, run, setup and configure, and to be a friendly design interface with simple and easy availabilities used by every age group.

ACKNOWLEDGEMENTS

This work would not have been possible without the support and encouragement of my colleague and friend, doc. Nikolai Gadzev, under whose supervision I chose this topic and began the thesis. Doc. Bozidar Sendov, from other university has also been of great help, and has assisted me in numerous ways, thought me and shown me how to develop in advance Java, long time ago and assisted me for the development of this application in his class, and the Informatics department at New Bulgarian University.

I would also like to thank my family and friends, on whose constant encouragement and love I have relied throughout my time at the University. I am grateful also to the example my older sister set for me, Juliana Tasevska and her friend for correcting my English errors. Their courage and conviction will always inspire me, and I hope to continue, in my own small way. It is to these people that I dedicate this work.

Table of Contents

LIST OF TABLES AND FIGURES

INTRODUCTION

The first computer messenger extrications was developed in the 1970s where the first available computer version was called DEC PDP-11. The application was based on a UNIX system with a command *talk* which enables user to directly talk with each others. The protocol that was used is *NTALK-protocol* has also been used by programs on the other operating systems, for example TalkR and WinTalk on Windows. (LAN messenger, 2009)

The first LAN (Local Area Network) messenger for Windows is WinPopup[1] and Abuse, a small utility included since Windows 3.11. WinPopup uses SMB (Server Message Block) / NetBIOS (Network Basic Input/ Output System) protocol and was intended to receive and send short text messages. Windows NT/200/XP improves upon this with A Messenger service, a Windows service compatible to WinPopup. On systems where this service is running, the received messages "pop up" as simple message boxes. Any software compatible with WinPopup, like a console utility NET SEND[2], can send such messages.

First more sophisticated version of LAN messenger is an Instant messaging (IM) from the middle of 1960s in operating systems like CTSS (Compatible Time-Sharing System) and Multics (Multiplexed Information and Computing Service), which predates the Internet. With the instant messaging one could share files, voice and video conversation, and many other futures.

Messenger-Pigeon is an application type of LAN messengers. A LAN messenger is an instant messaging program designed for a use within a single local area network (LAN), where instant messaging (IM) is a form of real-time communication between two or more people based on typed text. The text is conveyed via devices connected over a network such as Internet.

There are advantages using a LAN messenger over a normal instant messenger. The LAN messenger runs inside a company or private LAN, so only people who are inside the firewall will have access to the system. Communication data does not leave the LAN and the system can also not be spammed or get infected from the outside. The other advantages are

[1] Winpopup LAN Messenger
[2] Microsoft Net Send Documentation

that the LAN messengers can be categorized by needs of the individual company needs, as able to used it in different operation system and hardware components which requires minimum needs to be able to run or working with a multi operations' system (OS) as: Microsoft Windows 95, 98, NT, 2000, XP, 2003, Vista or newer, UNIX, Linux different distributions which are working with a different libraries and even with Mac OS X from the Apple.

The disadvantaged with LAN Messengers is that they are closed with the other futures, like: voice and video conversation, file sharing and many others, but still with a purpose of security reasons of the customers' demands and needs.

As today on market there are hundreds LAN messengers and Instant messengers with ability to work in more than one operating system or hardware components with a minimum system requirements. But still all of them are different in their own ways, like: interface, friendly usage, easy to setup, easy to install, customize by costumers/users needs, server-client, most important security and deployment.

Here are few examples of LAN Messengers:

- ePIGEON: http://www.epigeon-instant-messaging.com/index.html
- OfficePopup – WinPopup replacement: http://www.lanmessenger.com/
- Softros LAN messenger: http://messenger.softros.com/
- LanTalk NET messenger: http://www.lantalk.net/lantalk-net/ and many others.

Here are few examples of Instant Messengers:

- AIM: http://dashboard.aim.com/aim
- Jabber: http://www.jabber.org/web/Main_Page
- ICQ: http://www.icq.com/
- Yahoo! Messenger: http://messenger.yahoo.com/
- New Web Instant Messenger Meebo which supports multiple IM services : http://www.meebo.com/
- And Open-Source Code Pidgin, the universal chat client: http://www.pidgin.im/

OVERVIEW

Messenger-Pigeon is an easy-to-use LAN messaging application for effective intra-office communication. It is *client-server* software architecture which distinguishes client from server systems, which communicate over a computer network. A client-server application is a distributed system comprising both client and server software. A client software process may initiate a communication session, while the server waits for requests from any client. Client-server describes the relationship between two computers programs in which one program, the client program, makes a service request to another, the server program.

Messenger-Pigeon correctly identifies and works under Windows NT/2000/XP/Vista limited user accounts (without administrative privileges), UNIX, Linux with different distributions (eg. RedHat, Fedora, Ubuntu, SuSe and others) and MacOS X. **Messenger-Pigeon** comes with a variety of handy features such as: view the while content of user that are logged in the chat, and leave the chat, view the name of who is distributing a message, view of when particular user has joined the chat, view the table of contents when the user is connected with a server only as an administrator, and how many messages have been written to the chat, the administrator of the messenger can block the users of using un-propertied language or for other reasons and an intuitive interface. **Messenger-Pigeon** chatter utility offers strong ability of choosing Network Port[3] and IP[4] (Internet Protocol) Address to connect the clients, which gives a chance for the administrator to the network to setup the application on his own way.

The program is very stable when running under any operating system and on large or small TCP/IP networks, with different operating systems on same time. This application can be used for contacting individual users. All chat messages are count and saved in the database when the users are logged in to the chat. All correspondence can be looked up at any time. The program is very straightforward and requires no special training. It is ready to be used right after installation is completed. Configuring the program can be done by individual users without requesting any assistance from the IT staff or system administrators. The only thing that is needed to be connected to the server is IP Address and the elected Network Port, and

[3] TCP and UDP port
[4] IP address

no limitation of clients to establish the connection. It is optimized for minimum network traffic.

Messenger-Pigeon is an application for every company that started to experience communication problems?

- Problem with the corporate security and environment requirements by the company. Usage of the third party software for communicating outside of the office and breaking the job policy.

Have every employer in one chat, and on the hands to deliver them a quick message, inform your personnel about any important events or incidents in a matter of seconds. Make a control of when the employee has been logged in to the system and how many messages has sent to the other employees and many other reasons and futures that will be discuss in this document.

This Documentation is made for a purpose to deliver knowledge to an end users and administrators of how to setup and how to use the **Messenger-Pigeon**. What is a LAN messenger and why is it used for? How can anyone configure it in to the office, at home or as far as in the workshop? How can on get familiar with the interface of the two application server and client and how can it be configured and ran in different operating systems, software architecture.

There is no doubt about it; Instant Messaging is now an essential business tool. To enhance all the benefits of this tool, ***Messenger-Pigeon*** *Instant Message software ensures that your Instant Messaging is well managed, secure and integrated.*

Messenger-Pigeon *will enhance your business performance with instant internal communication solutions.*

SOFTWARE ARCHITECTURE

Software architecture is the set of decisions the software architect makes. It is commonly defined in terms of structural elements and relationships. Structural elements are identified and assigned responsibilities that client elements interact with through "contracted" interfaces. (Malan & Bredemeyer, 2005)

Messenger-Pigeon software architecture is shown in Figure 1 with the basic concept of structure, in acting as client-server software architecture, as an example.

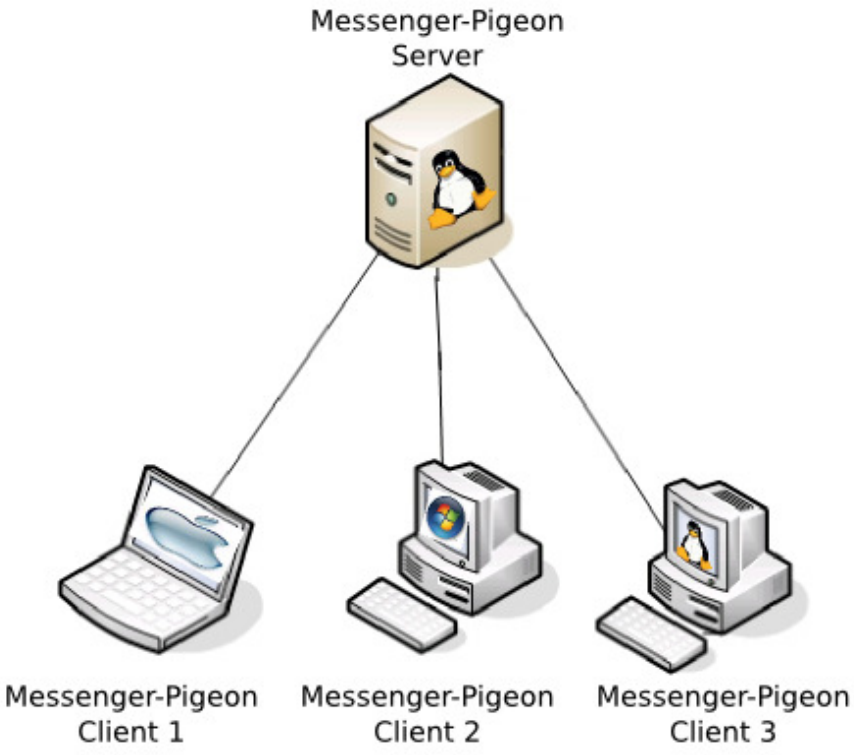


Figure 1

Structure of **Messenger-Pigeon** is based on:

- First we need to run the *server application* on one of the computers no matter in which operating system we started, as it is shown on Figure 1 that the server is run on Linux Operation System. Then we need to setup the database. After the

Messenger-Pigeon server application is started without any error messages or any pop-up windows that detect some error conflict of not being able to run the server application, it will successful establish peer / connection and create network port to the IP Address that the user has chosen, or configure it before starting the server.

- When first step is completed, then we can start running the *client application* of the **Messenger-Pigeon** as it is shown on the graphic 1 that we have connected 3 clients with a different operating systems:
    - Client 1: Running **Messenger-Pigeon** on laptop computer with Mac OS X / Apple operating system, and get connected with the server application through wireless connection.
    - Client 2: Is with operating system from Microsoft operating system, and it is connected to the server application through wire connections as it is the next client
    - Client 3: Has a same operating system as a server Linux or different distribution.

Three clients with different operating systems can connect in network through different sources (wire and wireless), to establish a successful connection through IP Address of the server and the Network Port that the administrator has assigned to the server application. Being aware of your firewall configuration and the picked port will be opened in the server system.

Important: the user who is installing this application or configuring in to the system to understand the steps, and the configuration settings that will be applied in this documentation. Should have:

- Networked basic picture map;
- Operating System;
- If firewall installed in to the system, what is there current status, is it running or it is stopped, what kind of network port or network connections and protocols are allowed to be connected to this computer and
- The IP Address statement, how the IP address are assigned to the other computers in the network.

INTERFACES ARCHITECTURE

Principles of user Interface Design are intended to improve the quality of user interface design. These principles are:

- *The structure principle*: Design should organize the user interface purposefully, in meaningful and useful ways based clear.
- *The simplicity principle*: The design should make simple, common tasks easy, communicating clearly and simply in the users own language, and providing good shortcuts.
- *The visibility principle*: The design should make all needed options and materials for a given task visible without distracting the user.
- *The feedback principle*: The design should keep users informed of actions or interpretations.
- *The tolerance principle*: The design should be flexible and tolerant, reducing the cost of mistakes.
- *The reuse principle*: The design should reuse internal and external components and behaviors. (Principles of User Interface Design, 2008)

All the above principles and quality it is guaranteed in **Messenger-Pigeon** in both applications: client and server. The interface design for **Messenger-Pigeon** it is simply and every one can understand and use it, with icons and menus.

Messenger-Pigeon interface is designed with the JFC (Java Foundation Classes) and Swing GUI (Graphical User Interfaces) Components (includes everything from buttons to split panes to tables), which encompass a group of features for building GUI's and adding rich graphics functionality and interactivity to Java applications that will allow the application to run in multi operating systems.

Because there is a two applications for **Messenger-Pigeon** we will discuss both, and the different concept and structure of applications.

Server Architecture Interface Design

On Figure 2, shows the simple interface design to server application of **Messenger-Pigeon**. Panels:

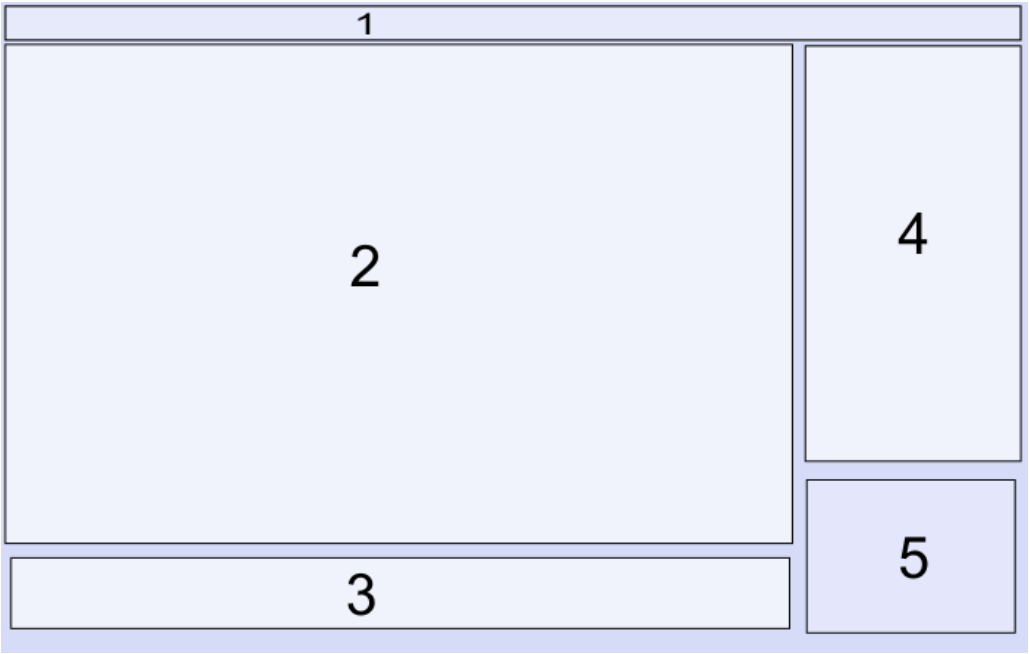


Figure 2

1. Menu, options: Options and Help.
2. Text Box: messages that are sent to be displayed to the chat.
3. Input text for chatting, to send a message to all users logged in the chat.
4. List of all users that are connected with the server.
5. Buttons to take an action of administrative tasks: send a message, change font and a block / terminate a client.

Client Architecture Interface Design

The difference between server and client applications is that the client interface does not have administrator tasks. Figure 3

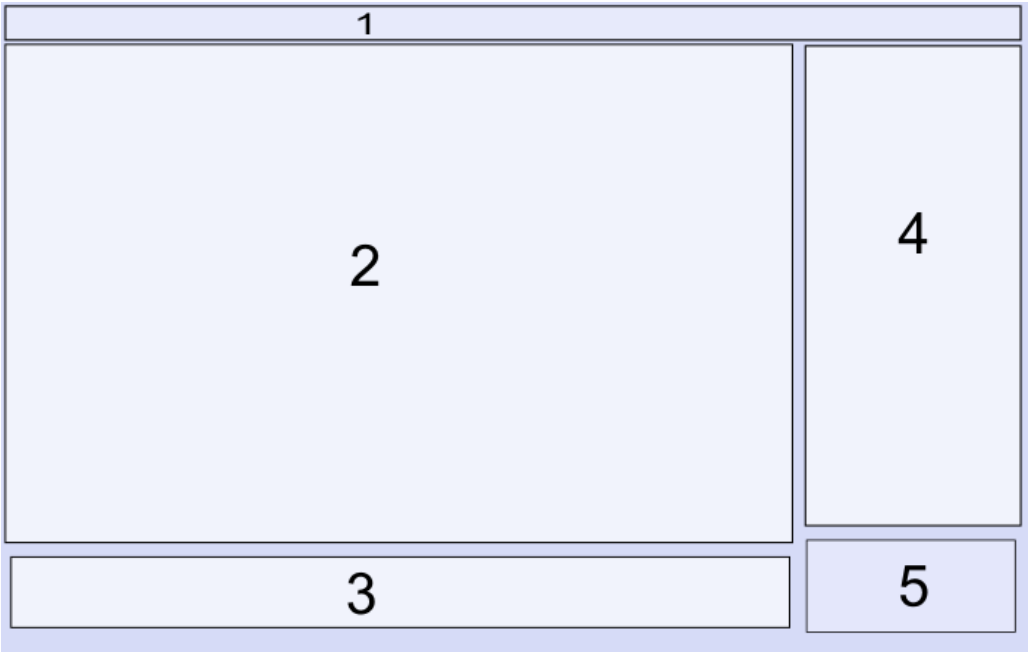


Figure 3

1. Menu with option of Option and Help.
2. Panel where the messages are displayed in chat with nicknames. (e.g. *Predrag: Hello Word!*)
3. Input text box to send the message to chat.
4. List of the users that are logged in the messenger.
5. Panel with buttons: Send and Font.

SOFTWARE PLATFORM

Platform describes hardware architecture or software framework, which allows software to run. Platforms include a computer's architecture, operating system, programming language and related runtime libraries or graphical user interface.

Messenger-Pigeon software platform it is developed with a Java[5] programming language. The following lines we will discuss about: history, basic introduction, software platform of Java.

[5] Java Home Page

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun Microsystems[6] Java platform. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. (Java Virtual Machine, 2009)

Java Virtual Machine (JVM) is set of computer software programs and data structures which use a virtual machine (software implementation of a machine (computer) that executes programs like a real machine) model for the execution of other computer programs and scripts. (Java Virtual Machine, 2009)

Java (software platform) refers to a number of computer software products and specifications from Sun Microsystems that together provide a system for developing application software and deploying it in a cross-platform environment.

In order for software to be considered cross-platform, it must be able to function on more than one computer architecture or operating system. This can be a time-consuming task given that different operating systems have different application programming interfaces or API's (Application Programming Interfaces). For example: Linux uses a different API from application software than Windows does.

From the above statement we can sum-up that **Messenger-Pigeon** can be run on both computer architectures: x86 and x86-64, with operating systems: Microsoft Windows, Linux, Mac OS X and Free BSD.

Client-server architecture on **Messenger-Pigeon** makes an easy way of illustrated in Java, as in connection-based communication such as TCP (Terminal Control Protocol) which **Messenger-Pigeon** is using; a server application binds a socket to a specific port number, which is **8000**. This has the effect of registering the server with the system to receive all data destined for that port. A client can then rendezvous with the server at the server's port, as illustrated here:

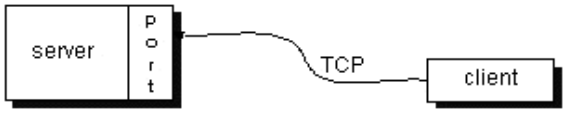


Figure 4

In Java, the networking classes in the JDK (Java Development Kit) are defined through the classes in *java.net*, where **Messenger-Pigeon** program can use TCP to communicate over the network.

[6] Sun Microsystems

To run the **Messenger-Pigeon** on your computer, operating system you will need to download the free latest recommended version Java from this link: http://www.java.com/en/.

PRODUCT DESCRIPTION

Messenger-Pigeon is an instant messaging software designed for the use within Small, Medium and Corporate Office's Local Area Network and home. This Intranet Messenger is an ideal replacement for INTERNET messengers in companies, which creates private and secure messaging, and improve internal communications, business productivity and relationships with colleagues inside the company. This software includes Instant Text Messaging.

Messenger-Pigeon software is compatible in LAN (Local Area Network), VPN (Virtual Private Network), WAN (Wide-Area Network), Terminal Service[7], Wi-Fi Networks. Software does not require any Internet connection, and it works across Ethernet port using TCP/IP protocol. Installation can be easily done without seeking system administrator help. Install the software in the computers, and the program is ready to use. Users do not need to add friends' list manually, it's automatically done. The employer has the opportunity of a full access to his employees when they have logged in to chat and how many massages have been sent from users to the chat. When the user has been logged in to the system it is automatically recorded with Date and Time in the database, which can be used for controlling your employee's time record entering the floor.

The Broadcast IP tool reduces network traffic in a sub network and it can run in Windows, Mac OS X & Linux.

It's very easy to use for beginners and offers a lot of functions even for professionals.

[7] Terminal Services is one the components of Microsoft Windows (both server and client versions) that allows a user to access applications.

SYSTEM REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer system. Those pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer version of software, system requirements tend to increase over time, industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. (System requirements, 2008)

The "Minimum System Requirements" must be satisfied for the software to be usable. Computers with lower specifications than the minimum requirements may sometimes also run the software. It is suggested, however, that the user will not have a representative experience of the software this way. Generally this set is regarded more of rule than a guideline. A system meeting this requirement will provide basic performance of a software application. (System requirements, 2008)

Recommended system requirements are often suggested by software vendors for optimal performance of the software. Although not a necessity, this set of requirements is often sought after by power users who expect to gain a better experience of software usability. Recommended System Requirements do not promise best possible performance of software and are treated as more of a guideline than a rule. Almost always a better system is available, or will be in future, to provide better performance. Also, exceeding by far these requirements does not guarantee to the user that everything will run with absolute smoothness and look its best.

Here is the specific minimum and recommended system requirements for a server and client applications.

Server

Table of System Requirements for **Messenger-Pigeon server.**

Table 1

Windows

	Minimum	Recommended
Processor	Pentium II / Celeron 568 MHz or equivalent	Pentium III or equivalent
RAM	256 MB	512 MB
Screen Resolution	800 x 600 pixels	1024 x 768 pixels
Operating System	Windows 98/2000/XP/Vista	Windows XP/Vista
Disk Space	50 MB	100 MB
Mac OS X		
	Minimum	Recommended
Processor	G3 or better	G4 or better
RAM	256 MB	512 MB
Screen Resolution	800 x 600 pixels	1024 x 768 pixels
Operating System	Mac OS X 10.3.9 or later	Mac OS X 10.3.9 or later
Disk Space	50 MB	100 MB
Linux		
	Minimum	Recommended
Processor	Pentium II / Celeron 568 MHz or equivalent	Pentium III or equivalent
RAM	256 MB	512 MB
Screen Resolution	800 x 600 pixels	1024 x 768 pixels
Operating System	Linux kernel 2.2.x,2.4.x, or 2.6.x, for the Intel x86 architecture	Red Hat, Debian, Mandrake, SuSe, Fedora
Disk Space	50 MB	100 MB

*Java is required

Client

Table of System Requirements for **Messenger-Pigeon client**:

Table 2

Windows		
	Minimum	Recommended
Processor	Pentium I or equivalent	Pentium I or equivalent
RAM	128 MB	256 MB
Screen Resolution	800 x 600 pixels	1024 x 768 pixels
Operating System	Windows 98/2000/XP/Vista	Windows XP/Vista
Disk Space	50 MB	50 MB
Mac OS X		
	Minimum	Recommended
Processor	G3 or better	G4 or better
RAM	128 MB	256 MB
Screen Resolution	800 x 600 pixels	1024 x 768 pixels
Operating System	Mac OS X 10.3.9 or later	Mac OS X 10.3.9 or later

Disk Space	50 MB	50 MB
Linux		
	Minimum	Recommended
Processor	Pentium I or equivalent	Pentium II or equivalent
RAM	128 MB	256 MB
Screen Resolution	800 x 600 pixels	1024 x 768 pixels
Operating System	Linux kernel 2.2.x,2.4.x, or 2.6.x, for the Intel x86 architecture	Red Hat, Debian, Mandrake, SuSe, Fedora
Disk Space	50 MB	50 MB

*Java is required

USING MESSENGER-PIGEON

In next few pages we will see how easy, simply it is to configure and use **Messenger-Pigeon** as server and client application.

How to setup the server and to run the client application in cross-platform environment in different operating systems? Get familiar with the interfaces, as we have already seen the interface architecture in both applications.

Before you are able to run the **Messenger-Pigeon** on your computer, operating system, you will need to download the free latest recommended version Java, as we described on Software Platform chapter, from this link: http://www.java.com/en/.

Click on the following icon on the page:

Figure 5

Please follow the instructions on the web page, for correct installation of Java.

If you do not have Java in your system you will not be able to run the **Messenger-Pigeon.**

Server

The Client-Server model of software architecture distinguishes the client system from server systems, which communicates over a computer network. A client-server application is a distributed system comprising both client and server software. Client software may initiate a communication session, while the server waits for requests from any server. (Client-server, 2009)

Client-Server describes the relationship between two computer programs in which one program, the client program, makes a service request to another, the server program. That is connected by direct port specified by the server. **Messenger-Pigeon** is using by default **8000** port on the network. Before you continue to configure the messenger you need to make sure that this port it is not used by any other application or service in your network. If the firewall in your system is not set to block 8000 port, that you add this port to allowed protocols or port in your current system firewall or internet security center.

In computer networking, a port is an application-specific or process-specific software construct serving as a communication's endpoint used by a Transport Layer[8] protocols of the Internet Protocol Suite[9] such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) A specific port is identified by its number, commonly known as the port number, the IP address it is associated with, and the protocol used for communication. Example ports:

Table 3

Port	Application
25	SMTP (Simple Mail Transfer Protocol)
110	POP3 (Post Office Protocol version 3)
80	HTTP (Hypertext Transfer Protocol)

Where each program binds a socket to it is end of the connection. To communicate, the client and the server each reads from and writes to the socket bound to the connection.

[8] Transport Layer / Transport protocols is a reliable and efficient end-to-end transport service between user process and server.

[9] Internet Protocol Suite is the set of communications protocols used for the Internet and other similar networks.

A socket is one end-point of a two-way communication link between two programs running on the network. Socket classes are used to represent the connection between a client program and a server program. The *java.net package provides two classes: Socket and ServerSocket*, that implement the client side of the connection and the server side of the connection, respectively. (Custom Networking, 2008)

On the client-side: The client knows the hostname / IP address of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendez-vous with the server on the server machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system. Figure 6

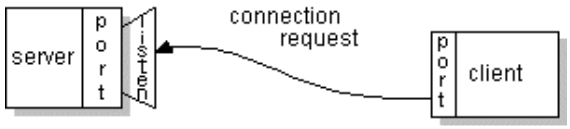


Figure 6

If everything goes well, server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket from connection requests while tending to the needs of the connected client. Figure 7

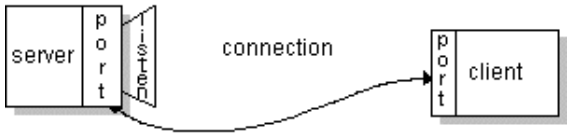


Figure 7

On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server.

The client and server can now communicate by writing to or reading from their sockets.

And end-point is a combination of an IP address and a port number. Every TCP connection can be uniquely identified by its two end-points. That way you can have a multiple connections between your host and the server.

Let as first see how to configure **Messenger-Pigeon** server application.

Configuration

Configuration of a server is easily made in different operating systems. First we need to setup a database called: ***pigeon***. File that is located in the directory of application is called *PigeonDateBase.mdb* for Windows and for MySQL for Linux, *PigeonDataBase.sql.*

Messenger-Pigeon it is set by the default port **8000**, if this port is already used or reserved by other application or a service it will display a message windows Figure 8 that will inform you that this port is already used. Where you will need to find which application or a service it is using the 8000 port in your network and terminate.

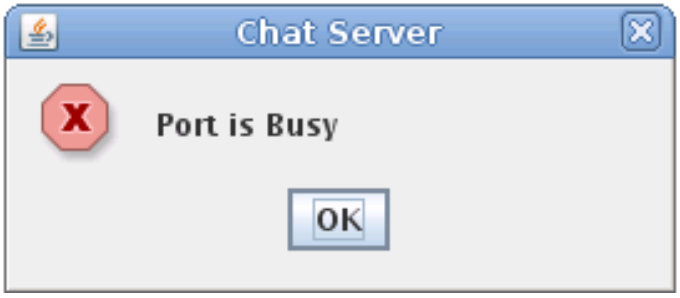


Figure 8

After network port is free and not set or blocked by any third party application you will need to restart the server application to take an effect, and if the result is without any error, that means the sever has established the socket local port and waiting for connections.

Configuration of a server is important for security, privileges and way of getting connection with the server for the clients

Here is the following different ways of configuration of database.

Windows – ODBC

Open Database Connectivity (ODBS) provides a standard software API (application programming interface) method for using Database Management System (DBMS).

ODBS is Microsoft's strategic interface for accessing data in a heterogeneous environment of relational and non- relational database management systems. Based on the Call Level Interface specification of the SQL Access Group, ODBC provides an open,

vendor- neutral way of accessing data stored in a variety of proprietary personal computer, minicomputer, and mainframe databases. (Article ID: 110093, 2007)

On run/search type: **ODBS** select **DataSources (ODBC)**,Figure 9. On next window click on **Add** at **User DNS** tab.

Figure 9

From the list select **Driver do Microsoft Access (*mdb)**, to next window on "**DataSourceName:"** input the text: **Pigeon** and afterwards on "**Database:"** click on **Select** where you need to specify the path of the file *PigeonDataBase.mdb*, then click **OK.** Now you should be able to see on the **User DNS** tab, on User Data Sources: a **Pigeon**, and then click **OK.** Figure 10

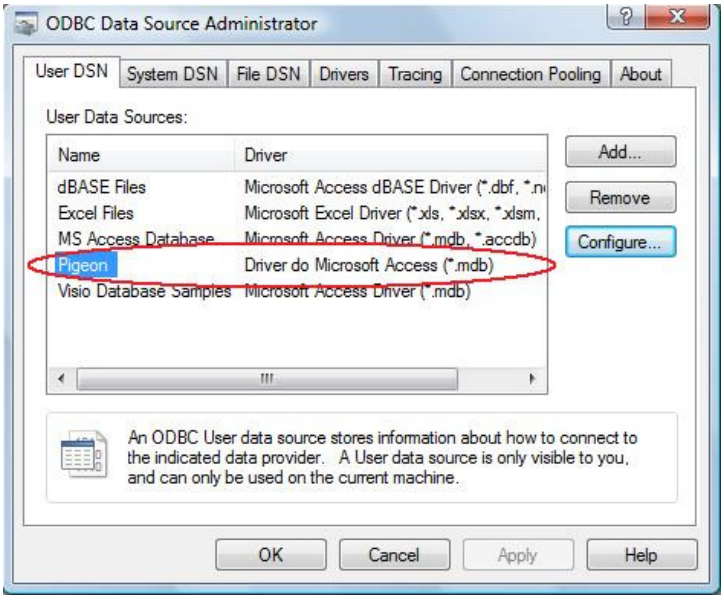


Figure 10

After the database setup, now we will need to setup a static IP address in the network that will allow a client from the network to connect to the chat. Network configuration depends on how it is setup in the building floor and how gets the IP addresses to the

computers. Way of configuring correctly and most security **Messenger-Pigeon** it is in hands of an administrator of the advance networking.

Messenger-Pigeon gets the local-host/loopback (127.0.0.1) as a default host name / IP address. Change settings and setup a different IP address and host name it is located in a system folder in: **C:\Windows\System32\drivers\etc** the file name is **hosts**. Or easy way on Run type:

notepad C:\Windows\System32\drivers\etc\hosts.

Here is the content of host file:

Hosts

```
# Copyright (c) 1993-2006 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

127.0.0.1 localhost
::1 localhost
```

To change the IP Address and the host name write as it is explained on example.

Example:

192.168.0.1 hostname

Now as we configure the database and the host name, IP address for the server you can run the **Messenger-Pigeon** on Microsoft Windows operating system and go to the next section of how to use it.

Linux – MySQL

MySQL is a relational database management system (RDBMS). The program runs as a server providing multi-user access to a number of databases. For more information please visit the following link: http://www.mysql.com/products/enterprise/server.html.
In different distributions there is a different way of installing the MySQL server. The best way is manuals how to install on the product website.

After setting up the MySQL server you will need to add the database table in to the server, which **Messenger-Pigeon** can get connected with. The best way to do this task is to install MySQL GUI client or PhpMyAdmin where you can get the instructions on the following link: http://www.phpmyadmin.net/home_page/index.php.

For easier setup is to use a MySQL Administrator GUI tool Figure 11, you can download from following link: http://dev.mysql.com/downloads/gui-tools/5.0.html

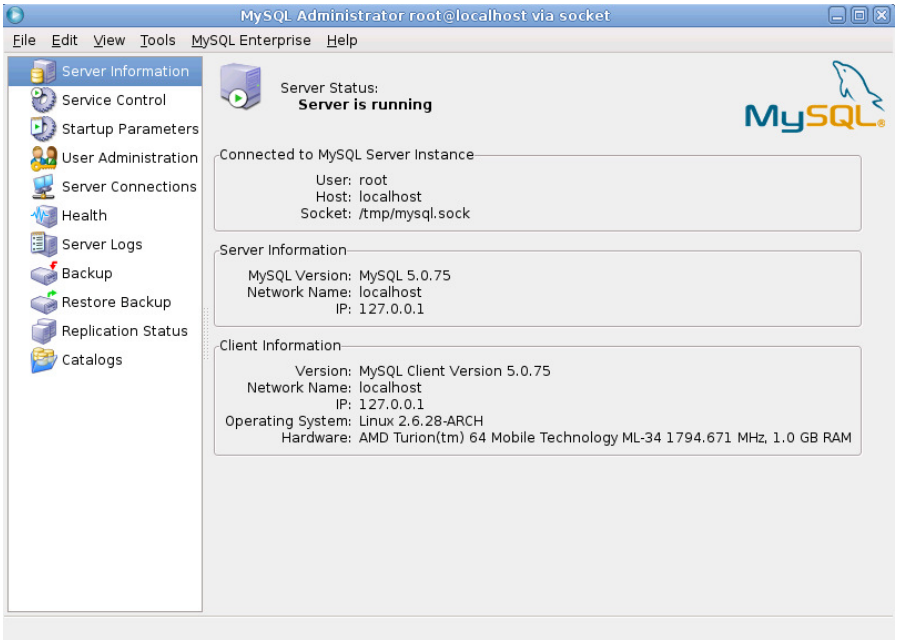


Figure 11

How to setup the database for **Messenger-Pigeon** in phpMyAdmin:

1. Open you browser and on the address bar type: http://localhost/phpMyAdmin/, screen Figure 12 where to provide a Username and Password for your MySQL server that has been already specified.

Figure 12

2. After successfully logging in to the phpMyAdmin, you need to create a database name for the server to access the table. To create a database you need to input the text in to the field for creating a new database Figure 13. Name the database: **pigeon**, and click **Create**.

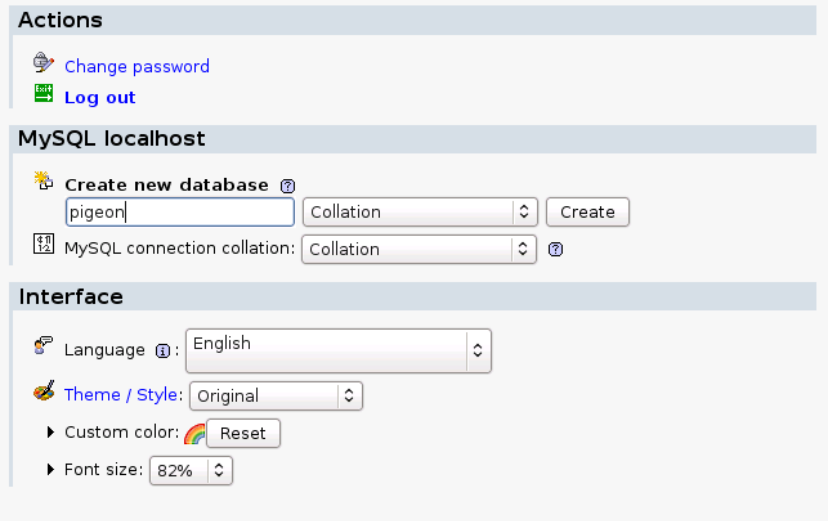


Figure 13

3. After creating the database, you need to import the database table. Click on the **Import** tab. (Figure 14) Click on **Browse…** and select the path where is *PigeonDatabase.sql*, and click on **GO**.

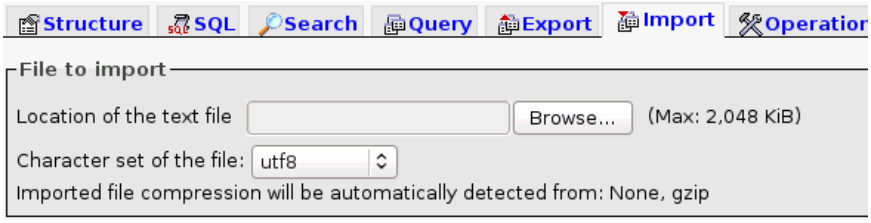


Figure 14

4. Three steps competed, on the left frame of phpMyAdmin you should be able to see: **pigeon (1)**. Means that the table was correctly imported to the pigeon database.

We are done trough half of the procedure. As we did in Windows, for more specific IP address and host name, we will do in Linux too. In Linux there are two ways of changing your host name:

1. Type in a terminal / console mode the following:

```
hostname new_name_of_host
```

2. Or make change it forever, when you restart or shutdown the computer. Open a terminal mode with administrative rights or root and type the following in terminal in your favorite text editor as on example shown nano text editor:

```
nano /etc/hosts
```

Here is the content of the hosts file in Linux:

/etc/hosts

```
#
# /etc/hosts: static lookup table for host names
#
#<ip-address> <hostname.domain.org> <hostname>

127.0.0.1 localhost.localdomain localhost

# End of file
```

You can specify a certain IP address and a host name for your server.

Example:

192.168.1.1 *localhost_domain localhost*

Now you are ready to go to the next step, we specified and IP Address and host name for the server in Linux, and setup the database for the server application.

Using a server

In both operating system Messenger-Pigeon can be run in same way. The difference is of what kind of distribution you are using for Linux. The bellow picture shows how to run it in Microsoft Windows and Linux operating systems, run it with the following script in the terminal / console ad a regular user mode:

java -jar "Destination_here_is_located_file: messenger-pigeon.jar"

Below picture shows how to run the **Messenger-Pigeon**. Figure 15

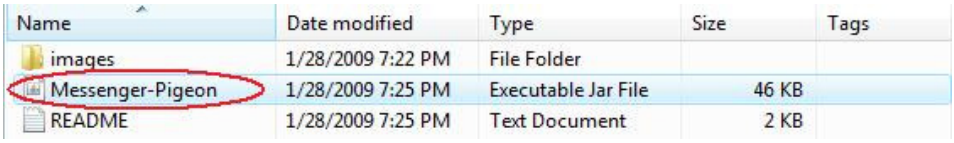


Figure 15

After you run **Messenger-Pigeon** server application, you will see the following window. (Figure 16)The basic interface elements are described below.

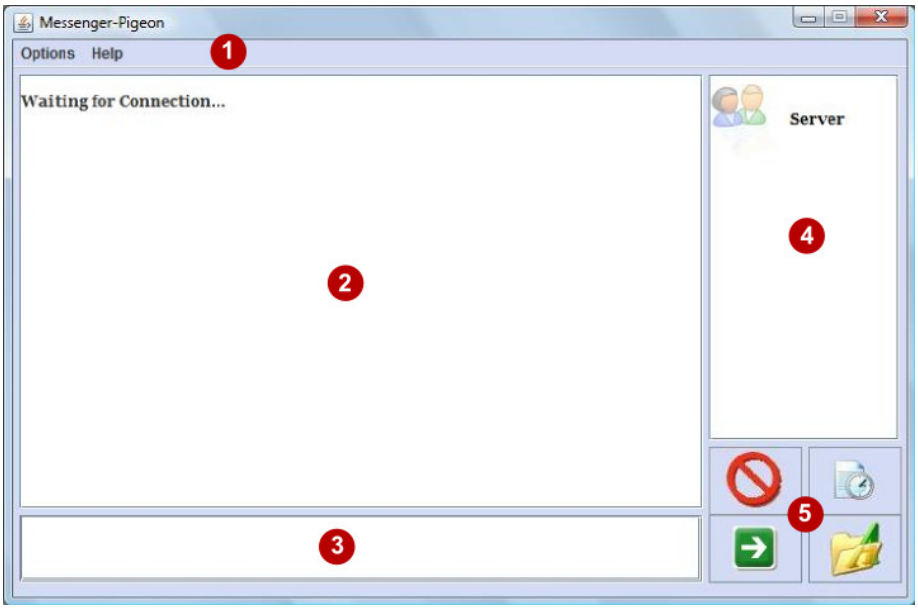


Figure 16

1. Main Menu, following options: Options (Figure 17) and Help (Figure 18). Options are:

Figure 17

Figure 18

- Nick Name: were you can change the nick name for the administrator or with CTRL+N from the keyboard.
- View Table: where you can see a table when and which client has joined the chat, subtotal messages that has been written in the chat (Figure 19) available with CTRL+E. This is connected through the dependency of operating system and database server.

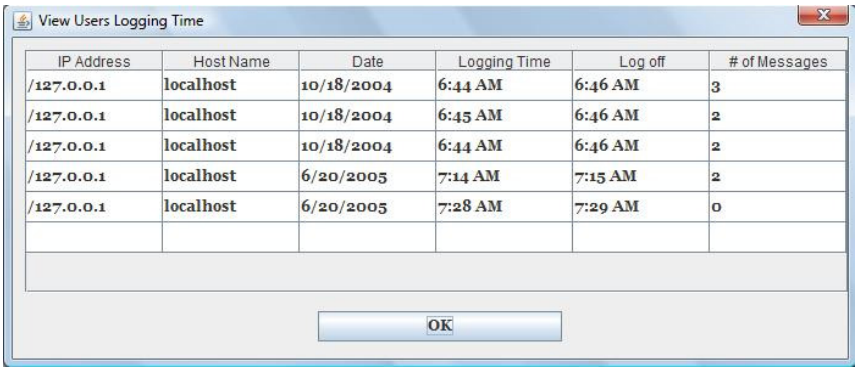


Figure 19

- Exit, ALT+4, exit the server and close the connection.

On Help menu we have only an About (Figure 20) available with ALT+1.

Figure 20

2. It is for all messages to be displayed and all the information when some of the clients join the chat, leave the chat or when the administrator takes and action for some client in the chat. Example:
    - When user is logged in to the chat the message is displayed: *Predrag is Connected.*
    - When we terminated user: *Predrag Forcely Terminated.*
    - When user leave the chat: *Predrag Left Conversation*.
3. It is used for inputting the massage to the chat to users. It can be sent with one click of **Enter** or by the Sent Button.
4. List of users that are logged in the chat, and can be selected and deselected. (Figure 21) *Predrag* is selected and the others are not selected.

Figure 21

5. Four buttons with different actions:

 Is used to send the message to the chat or just press ENTER on the keyboard.

 Font button, where you can select different style of font, size, color (Figure 22) and different Fonts, clicking this button it will pup-up another window called Font Selector shown on the picture bellow (Figure 23).

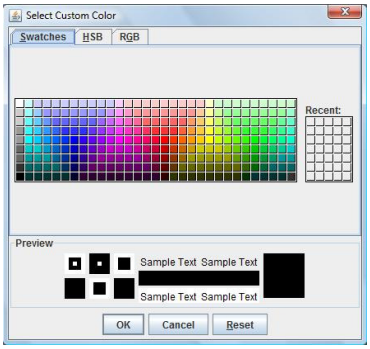


Figure 22

Color selection can be done with three different ways: Sqatches, HSB or RGB.

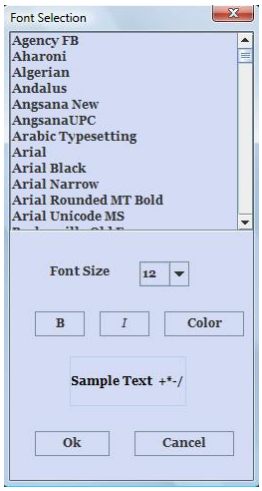


Figure 23

 Called Kill or terminate the selected user. To be able to terminate a user from the chat first need to select the user and then press the button. After the action is taken in panel number 2 will be displayed a massage: ***Predrag Forcely Terminated***, where *Predrag* is the user that we terminated and does not have more rights to write a message to the chat.

 The View button is used to display information when the user logged in to the chat. The user has to be selected before you click. (Figure 24)

Figure 24

After the introduction of the basic task, buttons and familiarity with the interface of **Messenger-Pigeon** you are ready to use it by your own.

Messenger-Pigeon server is easy to setup and use for any kind of computer level persons. There is no need from you to know an advance networking, be an administrator or be a computer guru to be able to setup in correct way.

Ones as it is a setup, every time when you start the **Messenger-Pigeon** you will use the same settings, and you can shut down your server or restart after system updates. There is no need to change the settings or any other task.

The next chapter is how to use a **Messenger-Pigeon** as a client application, which does not request any settings or setups; you just need to start the application.

Client

After a setup and configuration of a Server application of **Messenger-Pigeon**, you are ready to start the client application and get correctly connected with the server and start chatting. Before, you will need to know the interface of client application, task to be field for establishing the connection and abilities.

Therefore, before you start the client application, make sure that the server is running correctly, without any errors and it is connected through the network. It is waiting for clients to be connected and on second penal it is a written message: **Waiting for Connections …**, shown on Figure 16.

If the server it is not running or the port has been changed or the IP address it is not correctly specified in the client application you will get a pop-up window: **Connection Does not Established!** (Figure 25)

Figure 25

To correct this error message, the server it is running and that the port is not changed or blocked by your firewall or application in your system.

Using a client

When you first run the client as it is shown on the Figure 15 with double click. The first Input window (Figure 26) for **Messenger-Pigeon** is:

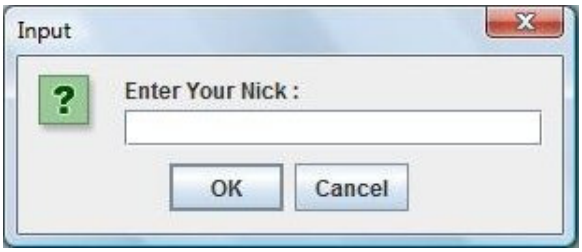


Figure 26

Type the nick name that you would like to use it for a chat. Example: *Predrag*. After click on **OK**, will pop-up new input window (Figure 27) for IP address of the server.

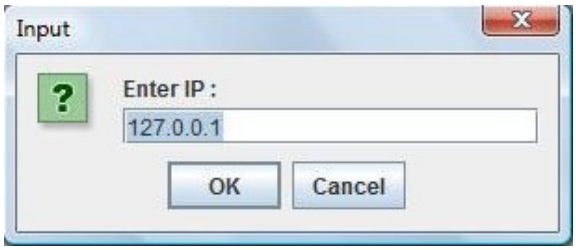


Figure 27

On the IP Address input field, you will need to input the server IP address, as you configured it on a hosts file. If you have not made any changes than it is best way to use the IP address of the server. Example shown that we are using the local-host IP address or 127.0.0.1. After clicking **OK** button pop-up a new window (Figure 28) with a message that the connection has been established.

Figure 28

If the connection has been failed to connect with the server, will show the Figure 25, and if you type a same nick name as other user in a chat, on panel 2 will show you and message: ***Please Change Your Nick Because Your nick is already in use!*** Way of fixing this error is by closing the application and run it again.

After the correct establishment of the connection, we will enter to the **Messenger-Pigeon** client application (Figure 29).

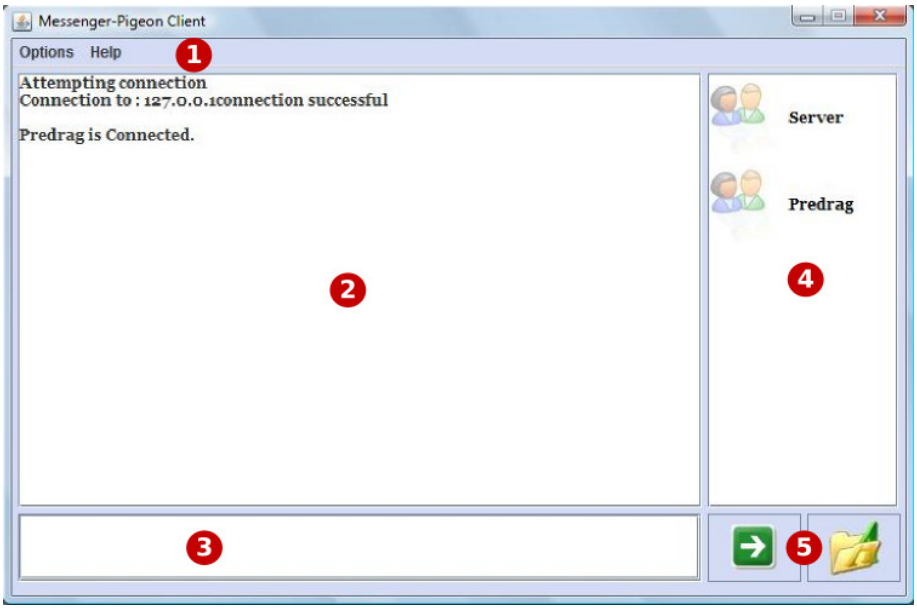


Figure 29

As you can see that on the second panel we received the messages:

Attempting conncection

Connection to: 127.0.0.1Connection successful

Predrag is Connected.

Where the client application has established the connection with server and now the new user with a nick name Predrag has connected to the chat.

When you run **Messenger-Pigeon**, you see the following window (Figure 29). The basic interface elements are described below.

1. Main menu bar, where there is a two options:
   - Options: where there is only an **Exit**, that can be run by clicking ALT+4;
   - Help: where there is only an **About** choice, it will open a window (Figure 30) or with ALT+1 from the keyboard.

Figure 30

2. Chat panel where all the messages that are sent to the users it will be displayed here and appropriate nick name, name of the user who is sending the message. In this panel the other content of the messages that will be displayed are:
   - When someone has connected to the chat with message: *Predrag Is Connected.*
3. This panel it is input field for sending a message to chat. In this field you can type a message without any set of maximum characters. All the messages will be displayed on the panel 2 (Figure 31).

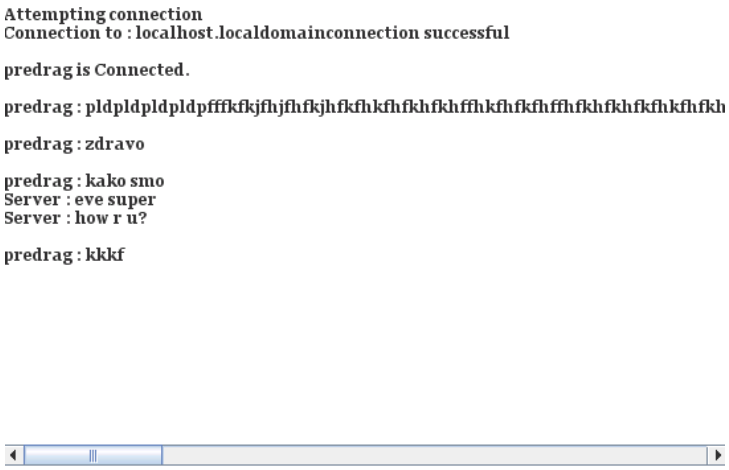


Figure 31

4. On this panel, you will see all the users that are connected to the chat (Figure 32).

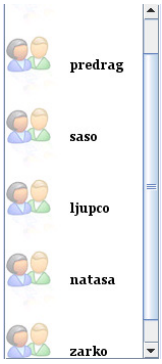


Figure 32

5. This is a main tool-bar, where you can use two options:

 Send. Where you can send the message with one click of this icon or by pressing an ENTER.


Font, were you can change the fonts for your panel 2 as it is shown on the Figure 29. When you click on this icon / button it will bring the font dialog window box (Figure 33) of ability to select which font, size, color (Figure 34) or style you won't.

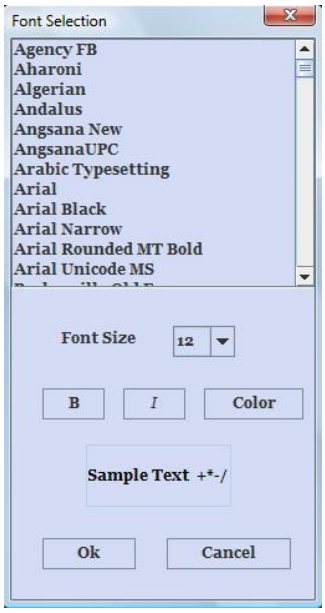


Figure 33

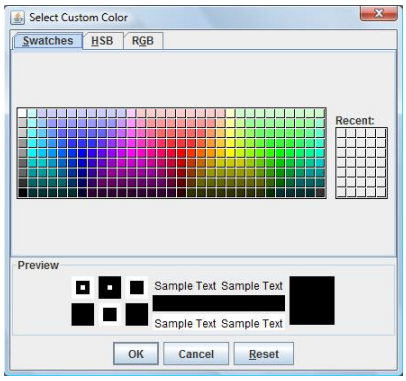


Figure 34

As you can see on Figure 33 there is a Simple Text box, where it is showing which style, font and size will take an affect after pressing OK button. Here is an example of how the **Messenger-Pigeon** will look like after changing the fonts:

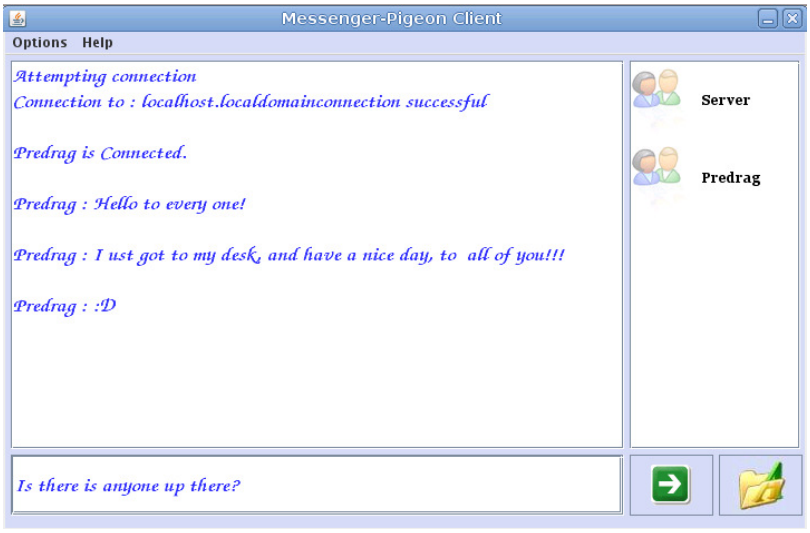


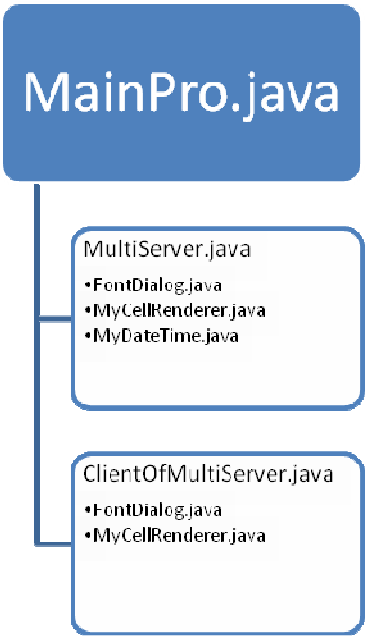
Figure 35

The **Messenger-Pigeon** it is very simple and very easy to run it, in every operating system. The interface it is user friendly in simple that everyone can use it in every day work.

DEVELOPING TASK

Developing of Messenger-Pigeon were made with a big aim form few books that I have found in the book store in Bulgaria and Macedonia for advance networking programming and few tutorials of how to program a messenger in Java. You can find them in the list of Bibliography used. Here is the reference: (Courtois, 1997), (Harold, Java Network Programming, Edition: 3, illustrated, revised, 2004), (Pitt, 2006), (Richard Monson-Haefel, 2000), (Courtois, 1997) and others.

Here is the developing structure of **Messenger-Pigeon**:



Java Class Library is a set of dynamically loadable libraries that Java applications can call at runtime. Because the Java Platform is not dependent on any specific operating system, applications cannot rely on any of the existing libraries. Instead, the Java Platform provides a comprehensive set of standard class libraries, containing much of the same reusable functions commonly found in modern operating systems.

Here is a class Hierarchy of a **Messenger-Pigeon** Source Code:

- java.lang.Object
  - java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
    - java.awt.Container

- java.awt.Window(implements javax.accessibility.Accessible)
  - java.awt.Dialog
    - javax.swing.JDialog (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
      - **FontDialog** (implements java.awt.event.ActionListener, javax.swing.event.ListSelectionListener)
  - java.awt.Frame (implements java.awt.MenuContainer)
    - javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
      - **ClientOfMultiServer** (implements java.awt.event.ActionListener)
      - **MultiServer** (implements java.awt.event.ActionListener)

- **MyDateTime**

MainPro.java is a main class to start the server and it is called the MultiServer.java class.

MultiServer.java class is used for running the server, designing the interface, menu, frames / panels, and connecting with MySQL or ODBC database, opening a socked / port so that client application can get connected.

How to create a buttons and menus:

Example 1:

```
public void actionPerformed(ActionEvent ae)
        {
                Object obj = ae.getSource(); //createing a object for access to menus and
buttons
                String strLabel = ae.getActionCommand();
                . . . . . .. ..
        }
```

Example of source code, how to connect with SQL database server to get and record the information for: View Table.

SQL

```
try
                {   //connecting with database ODBC or SQL: LANMessenger
                        Class.forName("com.imaginary.sql.msql.MsqlDriver");
                        String dsnSource = "jdbc:msql://localhost:3306/pigeon";
                        Connection con1 =
DriverManager.getConnection(dsnSource,"root","predrag");
                        st1 =
con1.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPD
ATABLE);
                        rs1 = st1.executeQuery(query1);
                }
                catch(ClassNotFoundException cnfe){}    //it cannot found the database / table
on the SQL server
                catch(SQLException sqle){}//when the SQL is not connected
```

With ODBC it is a different way, here is the example:

ODBC

```
try
                {
                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                        String dsnSource = "jdbc:odbc:Pigeon";
                        con1 = DriverManager.getConnection(dsnSource);
                        st1 =
con1.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPD
ATABLE);
                        rs1 = st1.executeQuery(query1);

                }
                catch(ClassNotFoundException cnfe){}
                catch(SQLException sqle){}
```

Source code how the server is run from the class:

Starting server:

```
public void runServer()     //start the server
{
        try
                {
                        serversocket = new ServerSocket(port);
                        ..... . ..  .. . .. . .. . . . . .. . . . .. . . .
                        //checking when user is logged in to the system and it is adding to the
list of users and it is adding it to class MyCellRenderer. Java and it is giving a ability to
enable the buttons Kill and others…
```

```
                }catch(IOException ioe){
                        JOptionPane.showMessageDialog(this,"Port is Busy","Chat
Server",JOptionPane.ERROR_MESSAGE);
                }
}
```

Updates of new user to the list and to the database is done with the following code:

Updates:

```
public NewThread(Socket csocket,JTextField messArea,JTextArea displayArea,JButton
sendBtn,JButton killBtn,JButton statusBtn,int count)
                { . . . . . .  .  .  .  .  .  .  .  .  .}
```

Attributes *public void run(){...}* and *private void getStreams(){...}* is checking if connection is established on the default port(8000), if it is will call the *private void processConnection(){...}* and it is sending a message that the connection is established, after checking if the nick name that you have choosen is already been used by other user or not, if it is not then it continues and gives a message that the user is connected to the chat and added to the list, sending to class MyCellRenderer to be added to the list on panel 4 on Figure 2 and Figure 3. Continuing with *private void closeConnection(){....}* whose user has logged out from chat. *public void actionPerformed(ActionEvent ae){...}* when we terminate / kill user and for the status of the buttons when to be enabled or disabled.

Structure of the database of Messenger-Pigeon with table pigeon:

Table 4

No(Key)	Text
ip	Text
hostName	Text
date	Text
logonTime	Text
logoffTime	Text
noMessage	Text

Table of the Messenger-Pigeon classes with overview description:

Table 5

Class	Description
MainPro.java	This is a main class, called a MultiServer.java class to start the server.

MultiServer.java	This class is for running the server, designing the interface, menu, frames / panels, and connecting with MySQL or ODBC database, opening a socked / port so that client application can get connected.
ClientOfMultiServer.java	This class is for designing the client interface application and creating sockets that can connect with the server.
MyCellRenderer.java	It used for creating a right panel for the server and client application for the list of users that are available to the chat. Where you have options to select user and update the connection of a new user to the chat. And changing the icon style with select and deselected.
MyDateTime.java	It is setting the Gregorian calendar and time to be recorded as AM and PM format.
FontDialog.java	It is a class for changing the font at the server and client application.

The other information's and the details of developing task are in the source code as a comment.

CONCLUSION

Today the usage of Internet it is more different than it was before. Users are using it for chatting and communicating between the users and contacts that are provided with applications like specified in Introduction. The instant messengers change the whole concept of Internet and voice over internet applications (e.g. Skype). The usage of these applications in companies and in floor irritates the employers and there is a need to close the employees of outside world, so that the employees will take all interests in work. But still, out there in the floor there is still need of applications or tools that can be deliver fast, easy and specific message to the floor, in a second to the employees send a schedule or change of specification on business. That's why I developed an application that will be used for this purpose as multi platform for all operating systems.

Messenger-Pigeon is an application developed in Java platform programming language as on purpose of no additional requirements of library or platform independence. Different LAN Chatting applications are developed in different programming languages, with different ways, interfaces and usage availabilities.

Messenger-Pigeon is an application with client-server software architecture. The interface it is very simple and easy to use for all different age users. It is simple and easy of configuration and setup the server with the two different database sources (ODBC and MySQL).

As it is presented in the above the availabilities of **Messenger-Pigeon** are:

- Configure and run in multi operating systems;
- Simple usage and friendly interface in both applications (server and client);
- Send a message to the whole list of users in chat without any limitations of maximum characters;
- Change the Font style and color of messages that are delivered through the chat;
- Users with same nick name will be not allowed and will inform that the nick name has been already used.
- Administrator of the server can change the name of his own, by default is Server.

- Administrator of the server, network can take few administrative tasks:
  - Terminate the user from the list, and blocked the user to be able to distribute any message to the others.
  - View a table of database when and time that user has been logged in to the chat and how many messages has been delivered to the chat for a day.
  - Configure the database source in ODBC or MySQL.

The summery of this thesis is that **Messenger-Pigeon** as an LAN Chatting Messenger can be used in business and home environment. It can save the money and time for distributing an important message to all users that are in the chat and can be used in Internet and intranet environments. It is secured way for chatting and communication.

We believe that **Messenger-Pigeon** as an application will find a wide area of applications, first of all because it is simplicity, in other words the fact that it is easily used, its effectiveness, in the sense that it provides easy and complete results. It is hoped that the final product will reach the market and its end users.

BIBLIOGRAPHY

Article ID: 110093. (2007, March 29). Retrieved from Microsoft Knowledge Base: http://support.microsoft.com/kb/110093

Bruno, E. (2005). *Java Messaging.* Charles River Media.

Client-server. (2009, February 19). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Client-server

Courtois, T. (1997). *Java Networking and Communications.* Prentice Hall PTR.

Custom Networking. (2008, February 14). Retrieved from The Java Tutorials: http://java.sun.com/docs/books/tutorial/

Dick Steflik, P. S. (2000). *Advanced Java Networking, Edition: 2, illustrated, revised.* Michigen, USA: Prentice Hall PTR.

Harold, E. R. (2006). *Java I/O, Edition: 2, illustrated.* O'Reilly.

Harold, E. R. (2004). *Java Network Programming, Edition: 3, illustrated, revised.* O'Reilly.

Instant messaging. (2009, February 17). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Instant_messaging

Java Virtual Machine. (2009, February 11). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Java_virtual_machine

Kenneth L. Calvert, M. J. (2008). *TCP/IP Sockets in Java: Practical Guide for Programmers.* Morgan Kaufmann.

LAN messenger. (2009, February 14). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/LAN_messenger

Malan, R., & Bredemeyer, D. (2005, April 28). ARCHITECTURE RESOURCES For Enterprise Advantage. Bloomington, IN, USA. Retrieved from Bredemeyer Consulting.

Nataraj Nagaratnam, B. M. (1996). *Java Networking and AWT API Superbible: The Comprehensive Reference for the Java Programming Language.* Waite Group Press.

Pitt, E. (2006). *Fundamental Networking in Java.* Springer.

Principles of User Interface Design. (2008, December 17). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Principles_of_User_Interface_Design

Richard Monson-Haefel, D. A. (2000). *Java Message Service.* O'Reilly.

System requirements. (2008, December 21). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/System_requirements