

MỤC LỤC

DANH SÁCH TỪ VIẾT TẮT	3
LỜI CẢM ƠN	4
MỞ ĐẦU	5
Chương 1: GIỚI THIỆU TỔNG QUAN CHƯƠNG TRÌNH	6
1.1 Giới thiệu Bots	6
1.2 Mô tả bài toán.....	7
1.2.1. Mô hình chương trình:	7
1.2.1. Mô tả:	7
1.3 Giao thức Yahoo	7
1.3.1. Cấu trúc các gói tin:	9
1.3.2. Cơ chế đăng nhập:.....	10
1.4 Cơ sở dữ liệu JDBC - ODBC.....	13
1.5 Tìm hiểu ngôn ngữ XML	14
1.6 Môi trường làm việc.....	16
1.6.1. Môi trường lập trình Java.....	16
1.6.2. Tổng quan về Netbeans.....	21
Chương 2: NỘI DUNG CHƯƠNG TRÌNH.....	23
2.1 Tương tác Yahoo.....	23
2.1.1. Cơ chế đăng nhập.....	23
2.1.2. Cơ chế trả lời tự động	25
2.1.3. Cơ chế kiểm tra kết nối tự động và đăng nhập lại	27
2.1.4. Quản lý danh sách nick trên Yahoo	28
2.2 Phân tích và trả lời truy vấn	29
2.3 Tương tác cơ sở dữ liệu.....	34
2.3.1. Cơ chế hoạt động:	34
2.3.2. Kết nối cơ sở dữ liệu:	36
CHƯƠNG 3: CHƯƠNG TRÌNH ỨNG DỤNG.....	40
3.1 Yêu cầu phần cứng, phần mềm.....	40
3.2 Giao diện chương trình	40
KẾT LUẬN	49
TÀI LIỆU THAM KHẢO	50

DANH SÁCH HÌNH ẢNH

<i>Hình 1. 1: Mô hình tiến trình bài toán.</i>	7
<i>Hình 1. 2: Header chung của Yahoo! Messenger.</i>	9
<i>Hình 1. 3: Cấu trúc trường dữ liệu của Yahoo.</i>	9
<i>Hình 1. 4: Chuỗi các bước đăng nhập.</i>	10
<i>Hình 1. 5: Cấu trúc gói xác nhận được gửi tới Yahoo! Server.</i>	10
<i>Hình 1. 6: Cấu trúc gói xác nhận được gửi trả lại từ Yahoo! Server.</i>	11
<i>Hình 1. 7: Cấu trúc được Server gửi lại Client.</i>	11
<i>Hình 1. 8: Cấu trúc gói xác lập quyền được gửi tới server.</i>	11
<i>Hình 1. 9: Cấu trúc gói xác lập quyền nhận được từ server.</i>	12
<i>Hình 1. 10: Client gửi trả sự ủy quyền tới Server.</i>	12
<i>Hình 1. 11: Yahoo! Buddy List nhận từ server.</i>	13
<i>Hình 2. 1: Các chức năng hoạt động của chương trình.</i>	23
<i>Hình 2. 2: Sơ đồ cơ chế đăng nhập.</i>	24
<i>Hình 2. 3: Sơ đồ cơ chế nhận gửi tin nhắn.</i>	26
<i>Hình 2. 4: Sơ đồ mô tả quá trình tự động đăng nhập lại.</i>	27
<i>Hình 2. 5: Sơ đồ cơ chế tự động thêm nick.</i>	28
<i>Hình 2. 6: Sơ đồ quy trình truy vấn phân tích cú pháp.</i>	29
<i>Hình 2. 7: Kiến trúc của JDBC.</i>	35
<i>Hình 2. 8: Cơ chế hoạt động của JDBC – ODBC.</i>	36

DANH SÁCH TỪ VIẾT TẮT

<i>Từ viết tắt</i>	<i>Nghĩa tiếng anh</i>
XML	Extensible Markup Language
JVM	Java Virtual Machine
SQL	Structured Query Language
IP	Internet protocol
ICQ	Internet Chat Query
TCP	Transmission Protocol
API	Application Interface
JDBC	Java Database Connectivity
ODBC	Open Database Connectivity
CO	Connection Oriented
DBMS	Database Management System
CPU	Central Processing Unit
IDE	Integrated Development Environment
JDK	Java Development Kit
DNS	Domain Name System

LỜI CẢM ƠN

Trong lời đầu tiên của báo cáo đồ án tốt nghiệp “Xây dựng hệ thống trả lời tự động thông qua Yahoo Messenger” này, em muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện đồ án.

Trước hết, em xin chân thành cảm ơn Thầy Giáo - Ths. Lê Thụy, Giảng viên Khoa Công Nghệ Thông Tin, Trường ĐHDL Hải Phòng, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ em trong suốt quá trình thực hiện đồ án.

Xin chân thành cảm ơn GS.TS.NGƯT Trần Hữu Nghị Hiệu trưởng Trường Đại Học Dân Lập Hải Phòng, ban giám hiệu nhà trường, các thầy cô trong Khoa Công Nghệ Thông Tin và các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em cũng như các bạn khác trong suốt thời gian học tập và làm tốt nghiệp.

Cuối cùng em xin gửi lời cảm ơn đến gia đình, bạn bè, người thân đã giúp đỡ động viên em rất nhiều trong quá trình học tập và làm đồ án tốt nghiệp.

Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên đồ án thực hiện chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy cô giáo và các bạn để em có thêm kinh nghiệm và tiếp tục hoàn thiện đồ án của mình.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 24 tháng 11 năm 2012

Sinh viên

Nguyễn Thu Trang

MỞ ĐẦU

Trong xã hội phát triển thông tin đã thực sự trở thành nguồn tài nguyên quan trọng. Các mối quan hệ, tính trật tự và tổ chức là những thuộc tính phổ biến của mọi hệ thống kinh tế - xã hội. Hệ thống càng phát triển tức càng có nhiều yếu tố hợp thành và nhiều mối quan hệ giữa chúng thì trật tự càng phức tạp và do đó nội dung thông tin càng phong phú đến mức không thể xử lý nổi bằng những phương pháp và công cụ “truyền thống”. Để xử lý được lượng thông tin lớn như vậy đòi hỏi phải có những phương pháp và công cụ cần thiết.

Trong cuộc sống ngày nay công nghệ thông tin luôn được coi là ngành mũi nhọn được đặt nền hàng đầu. Các sản phẩm được tạo ra của ngành công nghệ thông tin đã và đang giúp con người xử lý các vấn đề đơn giản hơn, đáp ứng nhu cầu sống ngày càng hiện đại này.

Việc trao đổi dữ liệu, thông tin, hay trò chuyện trực tuyến, tâm sự, cảm xúc cũng được nhanh chóng bày tỏ, chia sẻ thông qua các phương tiện trên internet như: gmail, facebook, yahoo... nó giúp chúng ta rút ngắn khoảng cách, kết nối cộng đồng với nhau hơn.

Ngày nay, sự phát triển nhanh chóng của Internet đã cho phép người dùng trao đổi thông tin nhanh hơn thông qua các hệ thống tin nhắn nhanh và đàm thoại trực tuyến với chi phí ngày càng rẻ. Có rất nhiều nhà cung cấp dịch vụ nhắn tin nhanh trực tuyến, thông qua những chương trình như: Yahoo! Messenger, Windows Messenger, Google Talk... Ở Việt Nam, dịch vụ chat qua Yahoo! Messenger là phổ biến và được ưa chuộng nhất, do phù hợp với nhu cầu của đa số người dùng và hoàn toàn miễn phí.

Yahoo! Messenger là một chương trình chat, trao đổi thông tin, dữ liệu rất hữu dụng trong giới internet từ trước đến nay, nhưng không phải lúc nào ta cũng có thể ngồi trực nick yahoo để trả lời lại ngay những tin nhắn đến, vì vậy em đã chọn đề tài “Xây dựng hệ thống trả lời tự động thông qua Yahoo Messenger”. Đồ án gồm 3 chương:

Chương 1: Giới thiệu tổng quan chương trình.

Chương 2: Nội dung chương trình.

Chương 3: Chương trình ứng dụng.

Chương 1: GIỚI THIỆU TỔNG QUAN CHƯƠNG TRÌNH

1.1 Giới thiệu Bots

Bot hay robot mạng là các ứng dụng phần mềm chạy các tác vụ tự động hóa trên mạng. Thông thường thì bot thực hiện các tác vụ đơn giản và có cấu trúc lặp đi lặp lại với một tần suất cao hơn nhiều so với khả năng của một soạn thảo viên là con người. Ứng dụng rộng lớn của Bot là trong duyệt tự động Web theo kiểu bò loang (*Web spidering*), trong đó một chương trình tự động tìm kiếm, phân tích và sắp xếp thông tin từ các máy chủ web với tốc độ cao hơn nhiều lần tốc độ con người. Mỗi máy chủ có một file có tên robots.txt chứa các quy tắc cho việc bò loang tự động tại máy chủ đó, đây là các quy tắc mà con bot cần tuân theo.

Bot thường xuyên được sử dụng trong thế giới Internet. Một ví dụ về Bots đó là search engine. Người ta định nghĩa spider được dùng bởi các công cụ tìm kiếm trực tuyến, ánh xạ website và phần mềm đáp ứng theo yêu cầu trên IRC (như eggdrop) là robot.

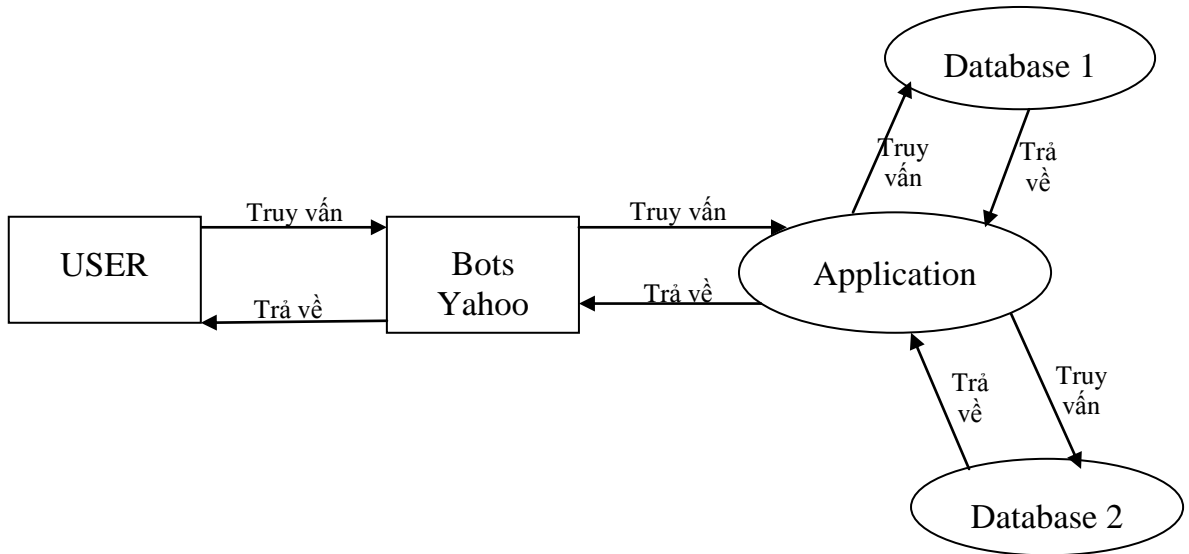
Bên cạnh các ứng dụng kể trên, bot còn có thể được cài đặt tại nơi đòi hỏi tốc độ phản ứng cao hơn tốc độ con người (chẳng hạn bot trò chơi điện tử và bot tại các trang web bán đấu giá) hoặc trong các tình huống cần đến sự bắt chước các hoạt động của con người như các chatbot – bot nói chuyện.

Nói một cách ngắn gọn khác, bot Yahoo là một dạng robot chúng được viết ra bởi các ngôn ngữ lập trình dùng để trả lời các request từ phía người dùng gửi tới. Ví dụ khi bạn add một bot nick vào, bạn gửi request có trong botfile, nó sẽ gửi về cho bạn hành động đã được thiết lập, có thể là một lời chào, có thể là một thông tin dự báo thời tiết, một status, điểm thi môn học,...

Hiện nay trên thực tế có rất nhiều những phần mềm chat trả lời tự động trên yahoo nhằm đáp ứng yêu cầu giải trí, cập nhật tin tức...trên yahoo chỉ cần một hành động add nickbot vào là người dùng có thể tìm kiếm các thông tin cần thiết trên bot. Ví dụ một số nick được dùng nhiều: wow.robot, yahoo friend9...

1.2 Mô tả bài toán

1.2.1. Mô hình chương trình:



Hình 1. 1: Mô hình tiến trình bài toán.

1.2.1. Mô tả:

Khi User gửi một yêu cầu truy vấn đến Bots Yahoo có cú pháp truy vấn cố định:

- Nếu User nhập sai cú pháp thì hệ thống sẽ trả về hướng dẫn cú pháp cho User.
- Nếu User nhập đúng thì dữ liệu nhập vào sẽ được chấp nhận thông qua Bots Yahoo gửi vào Application và đem so sánh với cơ sở dữ liệu trong database tương ứng, và sau khi so sánh có dữ liệu phù hợp thì kết quả sẽ được truy xuất ra Application và trả về cho Bots Yahoo, khi này thông tin được truy xuất ra sẽ tự động trả về cho User.

1.3 Giao thức Yahoo

Yahoo Messenger protocol là một protocol tầng Application chạy trên nền TCP, hoặc HTTP, SOCKS.

Do giao thức của Yahoo là giao thức đóng cho nên việc tìm hiểu cách thức hoạt động và trao đổi dữ liệu cần phải thực hiện một cách thủ công thông qua việc chặn bắt các gói tin gửi đi giữa các chương trình chat.

Việc tìm hiểu này được tiến hành thông qua chương trình chat rất phổ biến đó là chương trình Yahoo! Messenger. Qua việc theo dõi và ghi nhận thông tin, cấu trúc của các gói tin cũng như phương thức gửi dữ liệu của chương trình đã được ghi nhận. Từ đó để đưa ra cơ sở lý thuyết về giao thức dành riêng cho việc

truyền thông giữa chương trình và Yahoo sever được gọi là giao thức Yahoo (Yahoo! Protocol).

Yahoo! Messenger là một chương trình và giao thức nhắn tin có quảng cáo phổ biến cung cấp bởi Yahoo!. Đây là chương trình miễn phí có thể tải xuống để dùng với một tài khoản Yahoo!. Nó cho phép kết nối với các dịch vụ khác của Yahoo!, như là Yahoo! Mail, người sử dụng có thể tự động được thông báo khi có thư. Yahoo! Cung cấp các tính năng gọi điện từ máy tính tới máy tính là video call, truyền tập tin, dùng webcam, dịch vụ nhắn tin, và phòng chat với nhiều loại khác nhau.

Bên cạnh tính năng nhắn tin nhanh tương tự như của ICQ, Yahoo! Messenger cũng cung cấp (trên Microsoft Windows) nhiều tính năng độc đáo như: IMVironments (tùy biến giao diện cửa sổ Instant Message), tích hợp sổ địa chỉ và Custom Status Messages (tin nhắn trạng thái tùy chọn). Nó cũng là chương trình tin nhắn nhanh lớn đầu tiên có tính năng Buzzing và trạng thái âm nhạc. Và một tính năng mới được thêm vào gần đây là có thể chỉnh sửa hình đại diện (avatar).

Năm 2003 phiên bản Yahoo! Messenger 2.5.3 cho máy Mac được phát hành, là phiên bản ổn định hiện nay cho Mac OS X. Nó cung cấp ít chức năng hơn so với bản dành cho Windows và được báo cáo có khá nhiều lỗi. Vào tháng 6 năm 2006 Messenger cho Mac 3.0b1 đã phát hành bản beta, với một bản kế hoạch cho bản chính thức sau đó. Bản beta này có giao diện thân thiện hơn 2.5.3 và có hình đại diện, chức năng xem hình, BUZZ, và những tính năng khác cho giống như bản Windows. Tuy nhiên, nó vẫn thiếu các tính năng cao cấp hơn như gọi từ máy tính tới máy tính, sổ địa chỉ và vào phòng chat.

Tháng 7 năm 2006 Yahoo! Messenger đã tích hợp tin nhắn nhanh cho người dùng Windows Live Messenger và hoạt động hoàn hảo. Yahoo! Messenger đã không ngừng phát triển, nâng cấp cho ra các phiên bản mới đa chức năng, giao diện bắt mắt, thân thiện hơn. Các phiên bản được nâng cấp như: Yahoo! Messenger 8.1, Yahoo! Messenger 9.0 Beta, Yahoo! Messenger 10, Yahoo! Messenger 11, và phiên bản mới nhất hiện nay là Yahoo! Messenger 11.5.

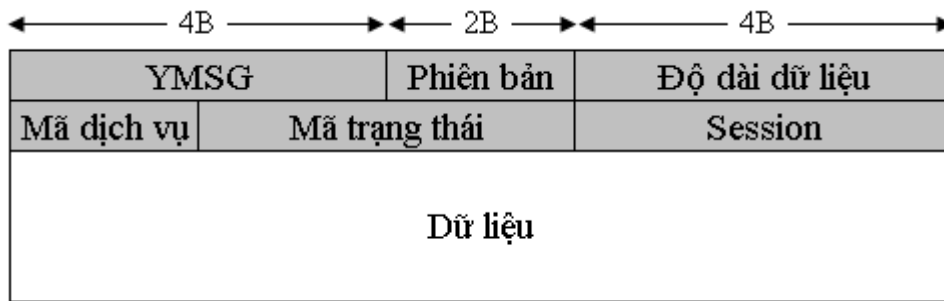
1.3.1. Cấu trúc các gói tin:

Qua quá trình capture Yahoo! Messenger packets từ phía client, ta thấy được YMSG luôn có cấu trúc gói tin (*Packet*) như sau:

[YMSG || Version || Packet length || Service || Status || Session ID || Content]

Tất cả truyền thông Yahoo! Đều sử dụng giao thức TCP thông qua truyền thông IP và dữ liệu Yahoo! Messenger nằm trong trường dữ liệu của gói TCP.

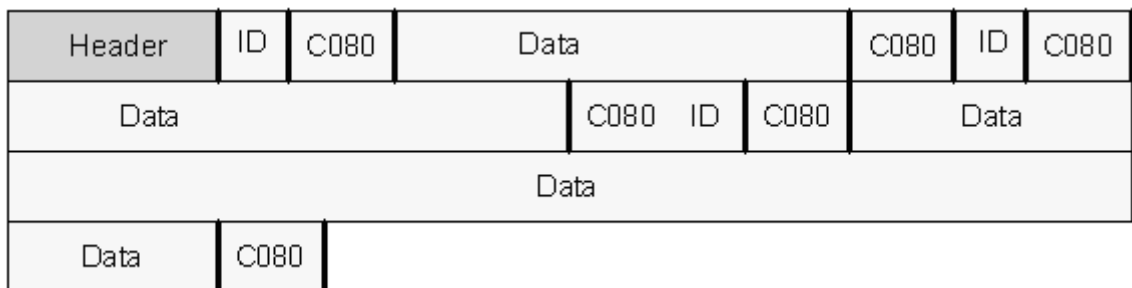
Yahoo! Mở rộng giao thức TCP/ IP bằng việc sử dụng header được tạo ra bởi định dạng header ở lớp ứng dụng của nó. Một header của Yahoo! Dài 20 bytes được nhận ra bởi 4 byte đầu tiên là “YMSG”. Header của Yahoo! Cũng bao gồm thông tin phiên bản của YMSG, độ dài thông báo, kiểu dịch vụ, trạng thái và ID của phiên làm việc (session ID).



Hình 1. 2: Header chung của Yahoo! Messenger.

Phần dữ liệu (Data) của gói Yahoo cũng tuân theo cấu trúc riêng. Ngay sau Session ID, trường dữ liệu có dạng FIELD ID, FIELD SEPARATOR, FIELD DATA, FIELD SEPARATOR,..., FIELD SEPARATOR. Trường ID được biểu diễn bằng một số nguyên ASCII có thể bao gồm một vài ký tự. Ngăn cách các trường là một dãy số hexa có giá trị là C0 80.

Hình 1.3 biểu diễn một cấu trúc trường dữ liệu của gói Yahoo:

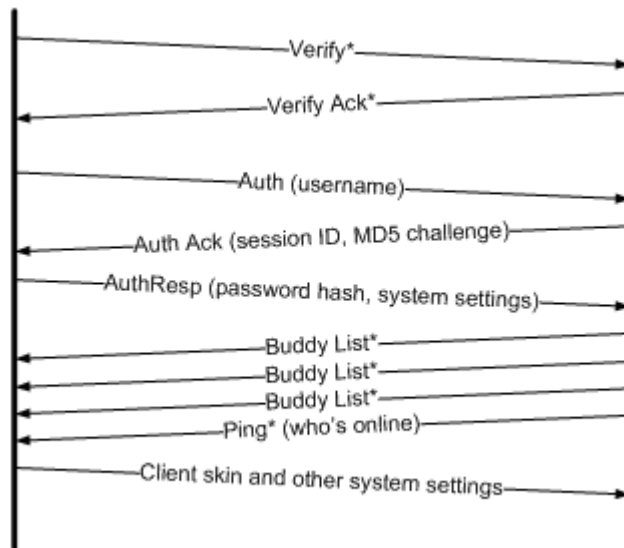


Hình 1. 3: Cấu trúc trường dữ liệu của Yahoo.

Như vậy, một gói Yahoo! Sẽ có dạng như trong hình bên trên. Ở dòng đầu hình vẽ bao gồm kiểu dịch vụ và phía trên bên phải cho thấy phương hướng truyền thông. Một danh mục các trường ID, kiểu dịch vụ, và trạng thái.

1.3.2. Cơ chế đăng nhập:

Để có thể sử dụng mạng Yahoo!, client phải đăng nhập với tên người dùng và mật khẩu. Thứ tự của các bước đăng nhập được trình bày trong hình dưới đây. Không phải tất cả các bước đều là cần thiết để đăng nhập vào mạng Yahoo! và các bước không bắt buộc được đánh dấu “*”.



Hình 1. 4: Chuỗi các bước đăng nhập.

Bước đầu tiên để đăng nhập là gửi một gói xác nhận, Yahoo_Verify, tới máy chủ Yahoo! Để kiểm tra nếu tồn tại một đường mạng dùng được. Cấu trúc gói này được trình bày trong các hình 1.5, hình 1.6 và hình 1.7:

Yahoo_Verify *Client → Server*

YMSG	Version 15	0
Verify (76)	Available	Null

Hình 1. 5: Cấu trúc gói xác nhận được gửi tới Yahoo! Server.

Chi tiết dữ liệu được gửi từ client tới server:

Yahoo_Verify *Server → Client*

	YMSG	Phiên bản	Độ dài d liệu	Dịch vụ	Trạng thái	Session
Giá trị	59 4d 53 47	00 0f	00 00 00 00	00 4c	00 00 00 00	00 00 00 00
Nội dung	YMSG	15	0	76	Available	Null

Yahoo_Verify *Server → Client*

YMSG		Version 15	0
Verify (76)	BRB		Null

Hình 1. 6: Cấu trúc gói xác nhận được gửi trả lại từ Yahoo! Server.

Chi tiết dữ liệu được Server gửi trả lại Client được mô tả như hình 1.7:

Yahoo_Verify *Client → Server*

	YMSG	Phiên bản	Độ dài d liệu	Dịch vụ	Trạng thái	Session
Giá trị	59 4d 53 47	00 00	00 00 00 00	00 4c	00 00 00 01	00 00 00 00
Nội dung	YMSG		0	76	BRB	Null

Hình 1. 7: Cấu trúc được Server gửi lại Client.

Khi truyền thông được xác nhận, sẽ tiến hành trao quyền bằng cách gửi đi gói Yahoo_Auth. Gói được gửi tới server bao gồm tên người dùng và yêu cầu dịch vụ. Server trả lời với xâu mời truyền thông (challenge) và session ID. Session ID được nhận từ server và sẽ được kèm theo các gói tin cho đến khi truyền thông kết thúc. Giai đoạn xác lập quyền được thể hiện trong hình 1.8 và hình 1.9:

Yahoo_Auth *Client → Server*

YMSG		Version 15	Length
Auth (13)	Available		Session ID
1: username			

Hình 1. 8: Cấu trúc gói xác lập quyền được gửi tới server.

Trong gói tin này, trường dữ liệu sẽ chứa xâu ký tự là tên người dùng, độ dài của trường dữ liệu cũng phụ thuộc vào đó.

YMSG		Version 15	Length
Auth (13)	BRB		Session ID
1: username 94: MD5 challenge string 13: "1"			

Hình 1. 9: Cấu trúc gói xác lập quyền nhận được từ server.

Khi client nhận được lời mời truyền thông, client sẽ thêm giá trị của xâu đó vào mật khẩu và tính toán mảng băm MD. Kết quả mảng băm sẽ được chia vào hai trường dữ liệu 6 và 96, và được gửi tới server. Cùng với mật khẩu hash, những dữ liệu về hệ thống khác như client version cũng được gửi qua gói AuthorizationResponse. Yahoo_AuthResp. Cấu trúc gói AuthResp được trình bày trong hình 1.10 và hình 1.11:

YMSG		Version 15	Length
AuthResp (6)	AuthResp (ZU.U)		Session ID
6: password hash #1 96: password hash #2 0: username 2: username 192: "-1" 2: "1" 1: username 135: client version 148: "360" 59: system settings			

Hình 1. 10: Client gửi trả sự ủy quyền tới Server.

Sau khi mật khẩu băm hợp lệ được thừa nhận từ server, người dùng sẽ được xem xét để đăng nhập và không cần thông tin khác từ client, server gửi buddy list (danh sách bạn chat) cho người dùng. Buddy list được chứa trong gói Yahoo_List.

<i>Yahoo_List</i>		<i>Server → Client</i>
YMSG	Version 15	Length
List (3)	BRB	Session ID
87: buddy list		

<i>Yahoo_List</i>		<i>Server → Client</i>
YMSG	Version 15	Length
List (3)	BRB	Session ID
87: buddy list continued		

<i>Yahoo_List</i>		<i>Server → Client</i>
YMSG	Version 15	Length
List (3)	BRB	Session ID
88: ignore list		
89: username		
59: system settings		
...		

Hình 1. 11: Yahoo! Buddy List nhận từ server.

Sau khi nhận được Buddy list từ server, các gói khác cũng được gửi tiếp theo. Những gói này là Yahoo_Ping chứa thông tin về những người đang online. Giống thông tin về Yahoo_List, Yahoo_Ping cũng qua nhiều gói, headers, hoặc cả hai. Thông tin về mỗi Buddy được chứa đựng trong cấu trúc gồm có username, tin nhắn gửi đi, và các thông tin về hệ thống khác. Một header thêm vào có thể cũng được gửi, cho biết những thông tin về các mail chưa được đọc thông qua Yahoo! Mail server.

Vào cuối quá trình đăng nhập, có thể server sẽ gửi tới client vài thông báo không cần thiết. Ví dụ: về kiểu skin của client, cập nhật launchCast Radio, đoạn trích, thời tiết, và Yahoo! Insider. Chương trình Yahoo! Update cũng có thể chạy song song với thao tác đăng nhập của chương trình.

1.4 Cơ sở dữ liệu JDBC - ODBC.

ODBC (*Open Database Connectivity* – kết nối cơ sở dữ liệu mở) cung cấp một phương pháp API phần mềm chuẩn cho việc sử dụng các hệ quản trị cơ sở

dữ liệu (DBMS). Mục đích của các nhà thiết kế ODBC là làm việc độc lập với các ngôn ngữ lập trình, các hệ cơ sở dữ liệu, các hệ điều hành.

ODBC là một phương pháp truy xuất cơ sở dữ liệu chuẩn được phát triển bởi SQL Access group vào năm 1992. Mục đích của ODBC là cung cấp cho các trình ứng dụng khả năng truy xuất dữ liệu bất kì mà không phải quan tâm hiện tại dữ liệu đang được quản lí bởi hệ quản trị cơ sở dữ liệu nào. ODBC làm được việc này bằng cách chèn một lớp trung gian vào giữa trình ứng dụng và hệ quản trị cơ sở dữ liệu. Lớp trung gian đó được gọi là một database driver(trình điều vận cơ sở dữ liệu). Mục đích của lớp trung gian này là chuyển đổi những câu truy vấn của trình ứng dụng thành những lệnh mà hệ quản trị cơ sở dữ liệu hiểu. Để làm được việc này thì cả hai trình ứng dụng và hệ quản trị cơ sở dữ liệu phải hiểu biết ODBC, tức là trình ứng dụng phải có khả năng tạo ra những lệnh ODBC và hệ quản trị cơ sở dữ liệu phải có khả năng đáp lại những lệnh đó.

Java Database Connectivity (JDBC) hỗ trợ ODBC dựa trên cơ sở dữ liệu và cung cấp một cơ sở dữ liệu độc lập.

JDBC có bốn phần chính, được sử dụng cho mỗi giai đoạn truy cập cơ sở dữ liệu:

- DriverManager: DriverManager tải lớp một trình điều khiển cơ sở dữ liệu trên cơ sở dữ liệu và cấu hình.
- Kết nối: lớp kết nối thực hiện confectioning và xác thực đến một cơ sở dữ liệu.
- Statement/PreparedStatement: Tuyên bố và các lớp học PreparedStatement gửi câu lệnh SQL để các công cụ cơ sở dữ liệu cho bộ tiền xử lý và thực hiện cuối cùng.
- ResultSet: lớp ResultSet cho phép kiểm tra các kết quả từ hành quyết.

1.5 Tìm hiểu ngôn ngữ XML

XML (Extensible Markup Language - Ngôn ngữ Đánh dấu Mở rộng) là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác. Đây là một tập con đơn giản của SGML, có khả năng mô tả nhiều loại dữ liệu khác nhau. Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet. Các ngôn ngữ dựa trên XML (thí dụ: RDF, RSS, MathML, XHTML, SVG, và cXML) được định nghĩa theo cách thông thường, cho phép các chương trình sửa đổi và kiểm tra hợp lệ bằng các ngôn ngữ này mà không cần có hiểu biết trước về hình thức của chúng.

Đặc điểm của XML:

XML cung cấp một phương tiện dùng văn bản (text) để mô tả thông tin và áp dụng một cấu trúc kiểu cây cho thông tin đó. Tại mức căn bản, mọi thông tin đều thể hiện dưới dạng text, chen giữa là các thẻ đánh dấu (markup) với nhiệm vụ ký hiệu sự phân chia thông tin thành một cấu trúc có thứ bậc của các dữ liệu ký tự, các phần tử dùng để chứa dữ liệu, và các thuộc tính của các phần tử đó. Về mặt đó, XML tương tự với các biểu thức S (*S-expression*) của ngôn ngữ lập trình LISP ở chỗ chúng đều mô tả các cấu trúc cây mà trong đó mỗi nút có thể có một danh sách tính chất của riêng mình.

Đơn vị cơ sở của XML là các ký tự theo định nghĩa của Universal Character Set (Bộ ký tự toàn cầu). Các ký tự được kết hợp theo các tổ hợp chuỗi hợp lệ để tạo thành một tài liệu XML. Tài liệu này gồm một hoặc nhiều thực thể, mỗi thực thể thường là một phần nào đó của các ký tự thuộc tài liệu, được mã hóa dưới dạng một chuỗi các bit và lưu trữ trong một tệp văn bản (text file).

Các tệp XML có thể dùng cho nhiều loại dữ liệu đa phương tiện. RFC3023 định nghĩa các loại “application/xml” và “text/xml”, với ý rằng dữ liệu được biểu diễn bằng XML mà không nói gì đến ngữ nghĩa của dữ liệu.

Bằng cách cho phép các tên dữ liệu, cấu trúc thứ bậc được phép, và ý nghĩa của các phần tử và thuộc tính có tính chất mở và có thể được định nghĩa bởi một giản đồ tùy biến được, XML cung cấp một cơ sở cú pháp cho việc tạo lập các ngôn ngữ đánh dấu dựa XML theo yêu cầu. Cú pháp chung của các ngôn ngữ đó là cố định — các tài liệu phải tuân theo các quy tắc chung của XML, bảo đảm rằng tất cả các phần mềm hiểu XML ít ra cũng phải có khả năng đọc (phân tích cú pháp – parse) và hiểu bố cục tương đối của thông tin trong các tài liệu đó. Giản đồ chỉ bổ sung một tập các ràng buộc cho các quy tắc cú pháp. Các giản đồ thường hạn chế tên của phần tử và thuộc tính và các cấu trúc thứ bậc được phép, ví dụ: chỉ cho phép một phần tử tên ‘ngày sinh’ chứa một phần tử tên ‘ngày’ và một phần tử có tên ‘tháng’, mỗi phần tử phải chứa đúng một ký tự. Đây là điểm khác biệt giữa XML và HTML. HTML có một bộ các phần tử và thuộc tính không mềm dẻo, chỉ có một tác dụng và không thể dùng cho mục đích khác.

XML không hạn chế về việc nó được sử dụng như thế nào. Mặc dù XML về cơ bản là dạng text, các phần mềm với chức năng trừu tượng hóa nó thành các định dạng khác giàu thông tin hơn đã nhanh chóng xuất hiện, quá trình trừu tượng hóa này được thực hiện chủ yếu qua việc sử dụng các giản đồ định hướng

kiểu dữ liệu (*datatype-oriented schema*) và khuôn mẫu lập trình hướng đối tượng (mà trong đó, mỗi tài liệu XML được thao tác như là một đối tượng). Những phần mềm như vậy có thể coi XML như là dạng text đã được tuần tự hóa chỉ khi nó cần truyền dữ liệu qua mạng.

Với XML có thể đưa ý nghĩa vào các thẻ trong văn bản. Quan trọng hơn, máy tính sẽ dễ dàng hơn trong việc xử lý thông tin. Có ba phần tử chủ yếu để miêu tả các phần của một văn bản XML: thẻ, phần tử, và thuộc tính. Dưới đây là văn bản mẫu minh họa các thuật ngữ:

```
<address>
  <name>
    <first – name></first – name>
    <last – name></last – name>
  </name>
  <street></street>
  <city></city>
  <postal-code></postal-code>
</address>
```

- Thẻ là phần chữ giữa dấu ngoặc đơn bên trái (<) và ngoặc đơn bên phải (>). Có thẻ bắt đầu (như <name>) và thẻ kết thúc (như </name>).
- Phần tử là thẻ bắt đầu, thẻ kết thúc, và mọi thứ giữa chúng. Trong mẫu trên <name>, phần tử bao gồm ba phần tử con: <title>, <first-name>, <last-name>.
- Thuộc tính là một cặp giá trị tên trong thẻ bắt đầu của một phần tử.

1.6 Môi trường làm việc

1.6.1. Môi trường lập trình Java

a. Lịch sử hình thành và phát triển Java:

Java là một ngôn ngữ lập trình cấp cao theo hướng đối tượng do James Gosling và một số đồng nghiệp ở Sun Microsystems phát triển (với tên gọi ban đầu là Oak). Đây cũng là một phần trong dự án Green (các phần mềm điều khiển thiết bị điện tử dân dụng) của Sun.

Năm 1995 Oak trở thành Java với phiên bản 1.0. Sau đó, Java không ngừng được phát triển và lần lượt các phiên bản mới được Sun Microsystems

phát hành. Tới thời điểm này thì phiên bản Java SE (còn gọi là Dolphin) phát hành năm 2011 được coi là phiên bản mới nhất của java.

b. Một số đặc điểm của Java:

❖ Máy ảo Java (JVM – Java Virtual Machine):

Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (CPU Intel, CPU Solarix, CPU Macintosh...), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. Đối với CPU Intel chúng ta có thể chạy các hệ điều hành như Microsoft Windows, Unnix, Linux...

Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi .EXE còn trên Linux thì được biên dịch dưới dạng đuôi . ELE, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại.

Chìa khóa giúp Java giải quyết hai vấn đề bảo mật và tương thích có thể xem là mã dịch ra không phải là mã thực hiện, đó là mã Bytecode. Bytecode là tập chỉ thị của trình biên dịch Java và chỉ định sang mã máy lúc thi hành, hệ thống đó gọi là máy ảo Java – đây mà chương trình biên dịch mã lệnh sang mã Bytecode.

Việc dịch chương trình Java sang Bytecode có thể làm cho chương trình chạy trên nhiều môi trường hơn. Lý do là chỉ cần mỗi JVM cho mỗi môi trường, khi chương trình chạy chỉ cần cho phép JVM chạy chúng. Mặc dù có nhiều loại máy JVM trên nhiều môi trường nhưng mã dịch ra Bytecode là giống nhau trên tất cả các hệ máy.

Thông dịch: Java là ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi *.java đầu tiên được biên dịch thành tập tin có đuôi .class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.

Độc lập nền: Một chương trình viết bằng ngôn ngữ java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Linux, Unix,...) miễn sao ở đó có cài đặt máy ảo Java JVM. Viết một lần chạy mọi nơi (*Write once run anywhere*).

Hướng đối tượng: Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng được định nghĩa trước, thậm chí

hàm chính của một chương trình viết bằng Java (hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (*multi inheritance*) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.

Đa nhiệm – Đa luồng: Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiến trình có thể chạy song song cùng một thời điểm và tương tác với nhau.

Khả chuyển (portable): Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java JVM là có thể chạy trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java

Hỗ trợ mạnh cho việc phát triển ứng dụng: Công nghệ Java phát triển mạnh mẽ nhờ vào “đại gia Sun Microsystem” cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như: J2SE (*Java 2 Micro Edition*) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-sever; J2EE (*Java 2 Enterprise Edition*) hỗ trợ phát triển các ứng dụng thương mại, J2ME (*Java 2 Micro Edition*) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây,...

Các loại ứng dụng của Java:

- Ứng dụng console: không có giao diện GUI.
- Ứng dụng đồ họa: Có giao diện GUI.
- Applet: Nhúng trong các trang Web.
- Servlet: các class thực thi phía web server.
- JSP: các file nhúng mã Java và HTML.
- Ứng dụng EJB, RMI, JMS: xây dựng ứng dụng bởi nhiều thành phần ghép lại, giao tiếp từ xa.

Công cụ và môi trường lập trình Java: Hiện nay có rất nhiều môi trường phát triển Java (IDE – *Intergrated Development Environment*). Mỗi môi trường cung cấp cho lập trình viên những tiện ích lập trình ở mức độ khác nhau.

Một số IDE thông dụng là:

- Netbeans.
- Jcreator.
- Jbuilder.

- Eclipse.

c. *Các phân tử cơ sở của Java*

Từ khóa tổ chức lớp:

- Package: Tạo thư mục chứa các lớp trong cùng một chương trình.
- Import: nạp các lớp thư viện cần dùng.

Từ khóa định nghĩa lớp:

- Class: định nghĩa lớp (thuộc tính + phương thức).
- Extends: kế thừa lớp.
- Interface: định nghĩa giao diện.
- Implements: khai báo thực thi giao diện.

Từ khóa cho các biến và lớp:

- Public: khai báo lớp, thuộc tính, biến, phương thức có thể truy nhập ở mọi nơi trong chương trình.
- Private: khai báo thuộc tính, phương thức riêng cho từng lớp và chỉ cho phép truy nhập trong lớp đó.
- Protected: khai báo các thuộc tính, phương thức được bảo vệ, cho phép truy nhập ở lớp chứa chúng và lớp dẫn xuất của lớp đó.
- Static: định nghĩa các thuộc tính, phương thức tĩnh của lớp, dùng chung cho mọi đối tượng của lớp đó.
- Final: chỉ ra các thuộc tính, phương thức không được thay đổi sau khi đã định nghĩa.

Từ khóa khai báo kiểu dữ liệu cơ sở:

- Long: kiểu số nguyên dài có dấu, dài 64bits.
- Int: kiểu số nguyên chuẩn có dấu, 32bits.
- Short: kiểu số nguyên ngắn có dấu, 16bits.
- Byte: kiểu số nguyên một byte có dấu kích thước 8bits.
- Char: kiểu ký tự Unicode, mỗi ký tự chiếm 16bits.
- Float: kiểu số thực biểu diễn giá trị dạng dấu phẩy động, 32bits.
- Double: kiểu số thực biểu diễn giá trị dạng dấu phẩy động, 64bits.
- Boolean: kiểu dữ liệu logic với hai giá trị: true, false.
- Void: kiểu rỗng thường sử dụng cho những hàm không trả lại giá trị thông qua tên hàm.

Từ khóa cho các giá trị và biến:

- False: giá trị kiểu boolean (sai).
- True: giá trị kiểu boolean (đúng).
- Null: giá trị rỗng cho biết đối tượng không tồn tại.
- This: biến chỉ tới đối tượng hiện thời.
- Super: biến chỉ tới đối tượng của lớp cha (lớp cơ sở).

Từ khóa xử lý ngoại lệ:

- Throw, throws: bỏ qua một ngoại lệ.
- Try: thử thực hiện cho đến khi gặp một ngoại lệ.
- Catch: bắt một ngoại lệ.
- Finally: thực hiện đến cùng một khối lệnh cho dù ngoại lệ xảy ra.

Từ khóa lập và kiểm tra đối tượng:

- New: tạo lập một đối tượng mới.
- Instanceof: kiểm tra một đối tượng có thuộc lớp hay một giao diện hay không.

Từ khóa dòng điều khiển:

- Switch: chuyển điều khiển của chương trình theo các trường hợp của case.
- Case: trường hợp được tuyển chọn theo switch.
- Default: trường hợp mặc định.
- Break: thoát khỏi vòng lặp.
- If: lệnh điều kiện rẽ nhánh.
- Else: rẽ nhánh theo điều kiện ngược lại của if.

Từ khóa dòng điều khiển:

- Continue: quay trở lại đầu vòng lặp.
- Return: trả lại giá trị của phương thức thông qua phương thức hoặc thoát khỏi phương thức vô điều kiện.
- Do...while: vòng lặp không xác định thực hiện lệnh trước kiểm tra điều kiện sau.
- While: vòng lặp không xác định kiểm tra điều kiện trước, thực hiện lệnh sau.
- For: vòng lặp có số lần lặp xác định trước.

Chú thích trong Java:

- Chú thích trên một dòng: //lời chú thích.
- Chú thích trên nhiều dòng: /* lời chú thích */.

- Chú thích trong tài liệu java: `/** lời chú thích */`.

d. *Java SE Development Kit JDK – nền tảng của Sun Microsystems*

Java SE Development Kit là một bộ sưu tập phong phú của các công cụ được thiết kế để phát triển phần mềm Java và các ứng dụng trên JDK cung cấp hỗ trợ đầy đủ cho dòng lệnh và có một trình soạn thảo mã Java, cung cấp cho các nhà phát triển mạnh mẽ môi trường để tạo ra và thử nghiệm các ứng dụng của họ. Cộng đồng OpenJDK cung cấp truy cập công cộng với bộ sưu tập lớn các dự án với mục tiêu chung của thử nghiệm và phát triển nền tảng JDK. Tuy nhiên, một số bộ phận của cộng đồng là chỉ dành cho các thành viên của dự án này JDK có vai trò dự án nghiên cứu và đồng ý giấy phép Java nghiên cứu (JRL) về bức ảnh chụp nguồn JDK7 và JDK6 theo giấy phép JRL có sẵn thông qua kho lưu trữ subversion, chỉ có thể truy cập với JDK nhà nghiên cứu vai trò Contributor JDK dự án cho JDK Java SE Development Kit bao gồm:

- Công cụ phát triển nằm trong bin/ thư mục con để phát triển, thực hiện, chương trình gỡ lỗi và tài liệu.
- Java Runtime Environment nằm trong jre/ thư mục con, có chứa Java Virtual machine, các thư viện lớp và các tập tin khác cho phép thực hiện dựa trên các chương trình Java. Thư viện lớp bổ sung và hỗ trợ các tập tin nằm trong lib/ thư mục con. Applet Demo và ứng dụng nằm trong bản Demo/ thư mục con, có chứa các ví dụ lập trình nền tảng Java với mã nguồn.
- Mẫu mã (trong mẫu thư mục con) mẫu lập trình cho một số Java API.
- Mã nguồn (bên trong src.zip) có chứa các tập tin nguồn cho tất cả các lớp lõi Java API.
- Phân phối của JDK bao gồm IDE Netbeans, đó là một môi trường phát triển tích hợp mạnh mẽ để phát triển các ứng dụng trên nền tảng Java.

Địa chỉ để Download:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html>

1.6.2. Tổng quan về Netbeans

Netbeans IDE là một “môi trường phát triển tích hợp” (*Integrated Development Environment – IDE*) kiểu như Visual Studio của Microsoft và được xem là một bộ ứng dụng “*must – download*” dành cho các nhà phát triển phần mềm.

Netbeans IDE hỗ trợ nhiều hệ điều hành khác nhau như Windows, Mac, Linux, và Solaris. Netbean bao gồm một IDE mã nguồn mở và một nền tảng ứng dụng cho phép nhà phát triển nhanh chóng tạo nên các ứng dụng dành cho web, doanh nghiệp, desktop và thiết bị di động bằng các ngôn ngữ lập trình Java, C/C++, JavaScript, Ruby, Groovy, và PHP.

Netbeans IDE cung cấp nhiều tính năng như:

- QuickSearch.
- Giao diện thân thiện với người dùng.
- Tự động biên dịch khi lưu (Compile on Save).

Ngoài việc hỗ trợ Java mọi phiên bản (Java SE, Java EE, Java ME), Netbeans IDE còn là công cụ lý tưởng cho việc phát triển phần mềm bằng các ngôn ngữ PHP, C/C++, Groovy and Grails, Ruby and Rails, Ajax and JavaScript. Theo thời gian Netbeans cũng cho ra rất nhiều phiên bản và được nâng cấp và hoàn thiện nhiều, phiên bản Netbeans 7.3 là phiên bản mới nhất cho đến hiện nay.

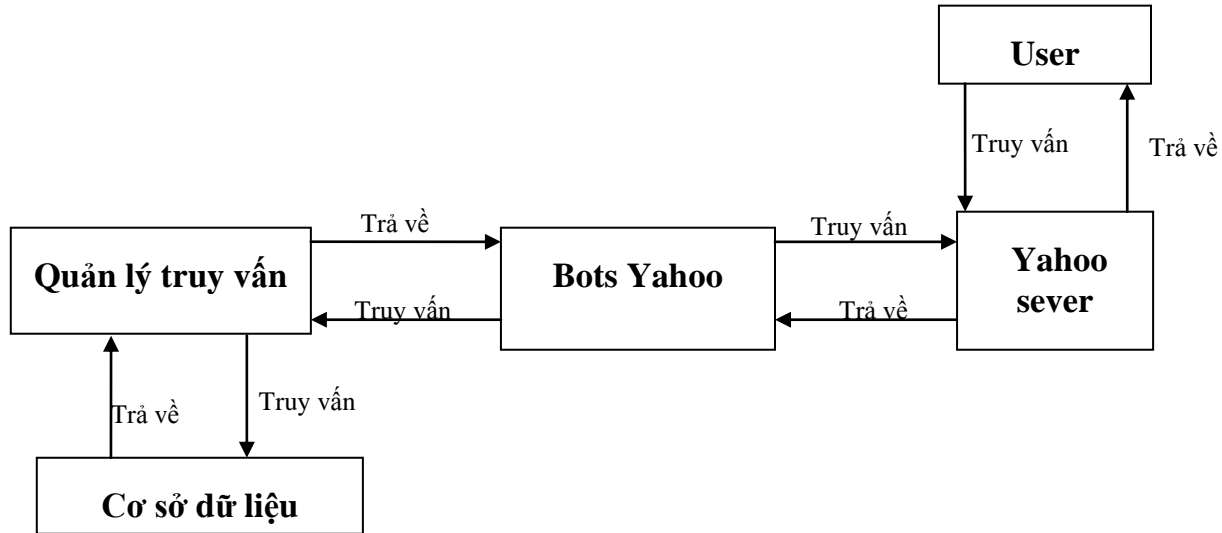
Trong đề tài nghiên cứu này sử dụng phiên bản Netbeans 7.2 được download tại trang: <http://netbeans.org/community/releases/72/index.html>

Netbeans IDE 7.2 cung cấp một hiệu suất cải thiện đáng kể và kinh nghiệm mã hóa, với khả năng đáng chú ý như tích hợp với Builder Scene cho trực quan tạo ra JavaFX các hình thức hỗ trợ cho các khuôn khổ PHP nhiều, hỗ trợ cập nhật Groovy, và nhiều cải tiến khác trong Java EE, Maven, C/C++ và NetBeans Platform, hỗ trợ cho JavaME 3.2 SDK.

Chương 2: NỘI DUNG CHƯƠNG TRÌNH

2.1 Tương tác Yahoo

Chương trình trả lời tự động qua hệ thống tin nhắn nhanh Yahoo! Messenger bao gồm các chức năng:

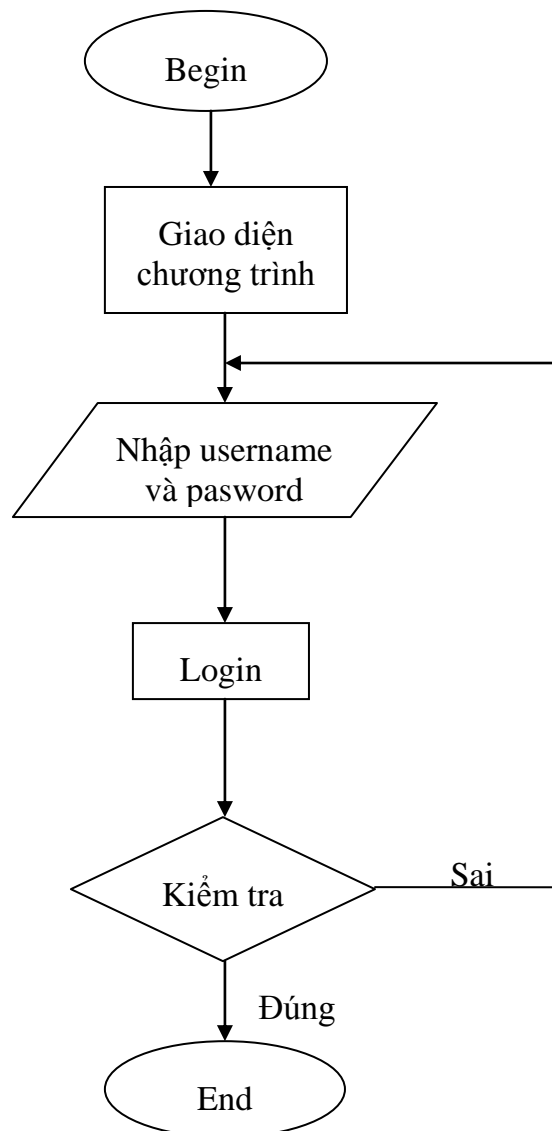


Hình 2. 1: Các chức năng hoạt động của chương trình.

2.1.1. Cơ chế đăng nhập

Để có thể sử dụng hệ thống nhắn tin qua Yahoo! Messenger, cần phải có một chức năng đăng nhập vào tài khoản Yahoo! Messenger đã được đăng ký trước bởi người quản trị. Cơ chế được bắt đầu khởi động chương trình người quản trị sẽ nhập tên *Username* và *password* của *Bot Yahoo*. Thông tin được thông qua máy chủ để kiểm tra nếu đúng thì kết thúc quá trình đăng nhập và chuyển qua bước tiếp theo, nếu kết quả là sai thì sẽ quay lại nhập *username* và *password*.





Hình 2. 2: Sơ đồ cơ chế đăng nhập.

Thao tác được thực hiện qua đoạn mã sau: (*openymsg library*)

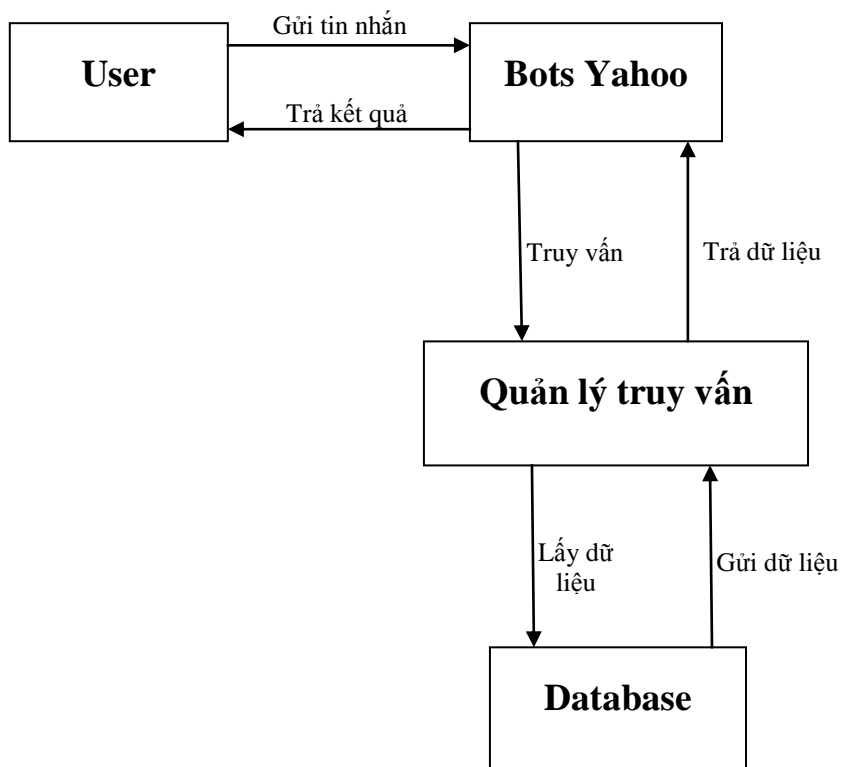
```

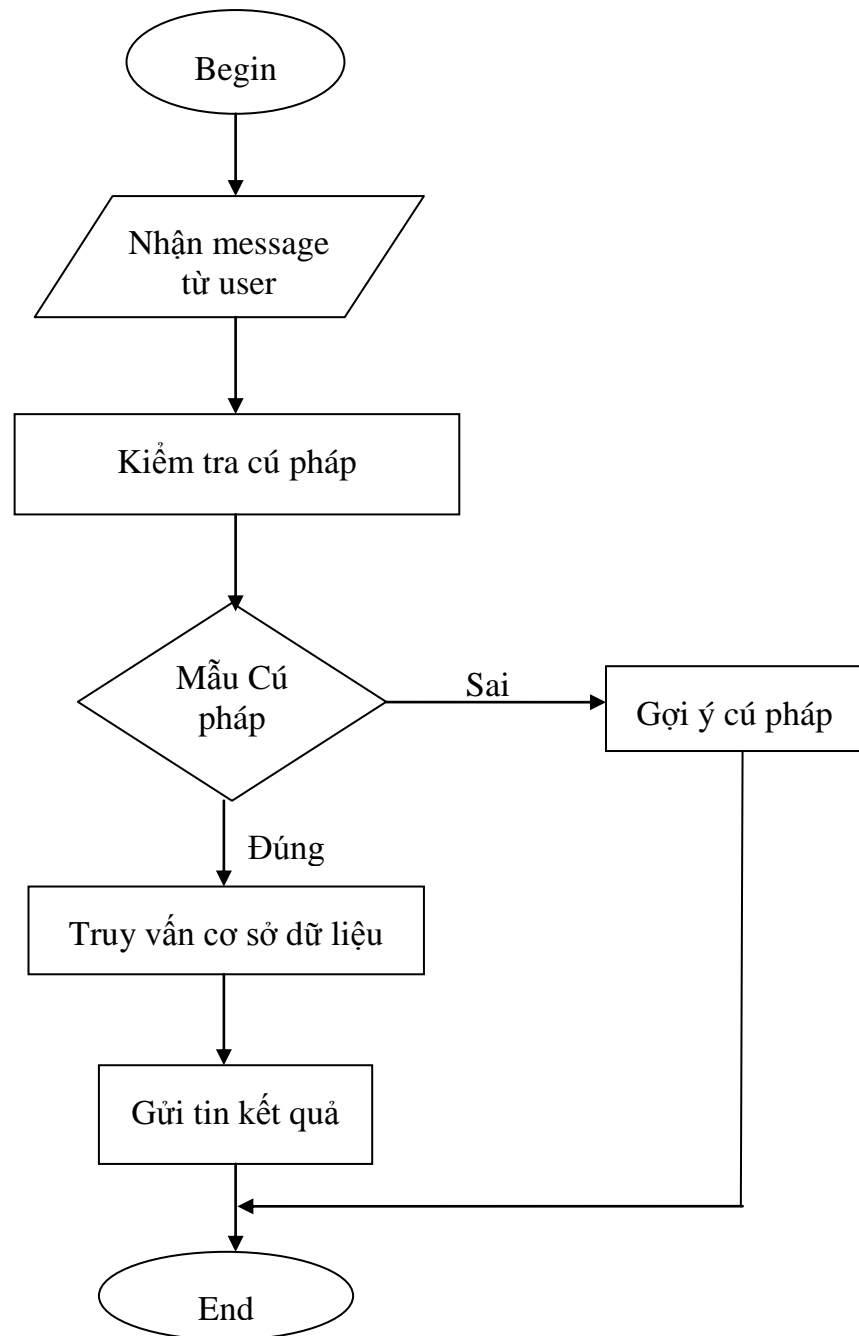
private Session S11; // khai báo biến Session với tên S11.
String username; // tên đăng nhập của người dùng.
String password; // mật khẩu của người dùng.
S11.login(String username, String password); //đăng nhập
S11.logout(); //đăng xuất
  
```

Lớp *Session* sử dụng phương thức *login* vào Yahoo với *username* và *password* là tên đăng nhập và mật khẩu đăng nhập của người dùng. Lớp *Session* sử dụng phương thức *logout* để thoát khỏi trạng thái đang đăng nhập của người dùng đó.

2.1.2. Cơ chế trả lời tự động

Khi nhận được tin nhắn từ người dùng gửi tới chương trình thông qua hệ thống tin nhắn nhanh Yahoo! Messenger, chương trình sẽ *kiểm tra mẫu tin nhắn* nhận được. Nếu mẫu tin nhắn *hợp lệ* thì chương trình sẽ bắt đầu *truy vấn cơ sở dữ liệu* và lấy ra thông tin phù hợp với yêu cầu và gửi trả lại người dùng dưới dạng tin nhắn. Trường hợp mẫu tin nhắn *không hợp lệ*, chương trình sẽ đưa ra *gợi ý về các mẫu tin nhắn* trong hệ thống và gửi tới người dùng.





Hình 2. 3: Sơ đồ cơ chế nhận gửi tin nhắn.

Thao tác thực hiện thể hiện qua đoạn mã:

`Private Session S11;`

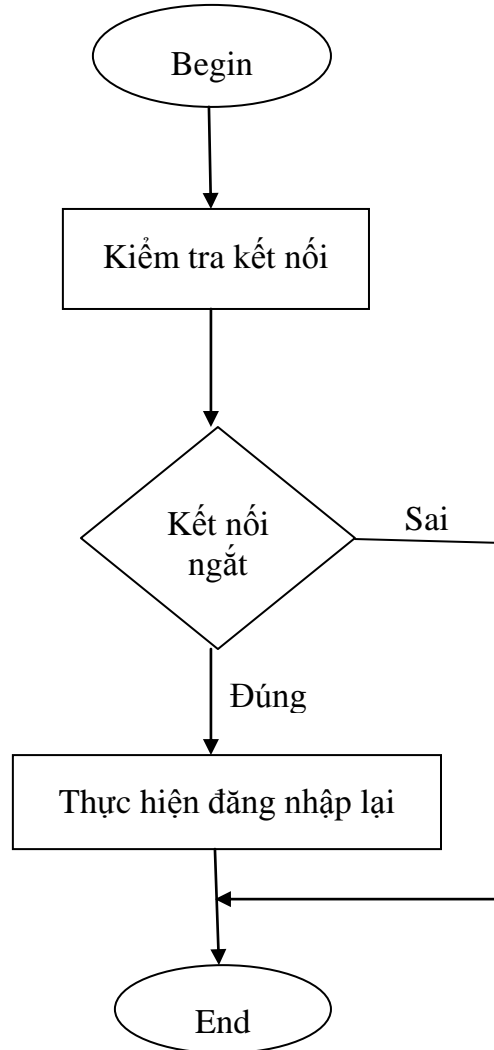
`jTextArea1.append(event.getFrom() + ": " + event.getMessage()+"\n");`

`//sự kiện bắt thông tin là tin nhắn của người dùng gửi tới Bot Yahoo.`

`S11.sendMessage(String username, String message); //gửi tin nhắn
message trả về cho người dùng username sau khi nhận được kết quả.`

2.1.3. Cơ chế kiểm tra kết nối tự động và đăng nhập lại

Trong quá trình hoạt động nếu chương trình và máy chủ không thực hiện được kết nối vì một lí do nào đó thì chương trình sẽ tự động kết nối lại cho tới khi có kết nối và thực hiện đăng nhập lại thành công.



Hình 2. 4: Sơ đồ mô tả quá trình tự động đăng nhập lại.

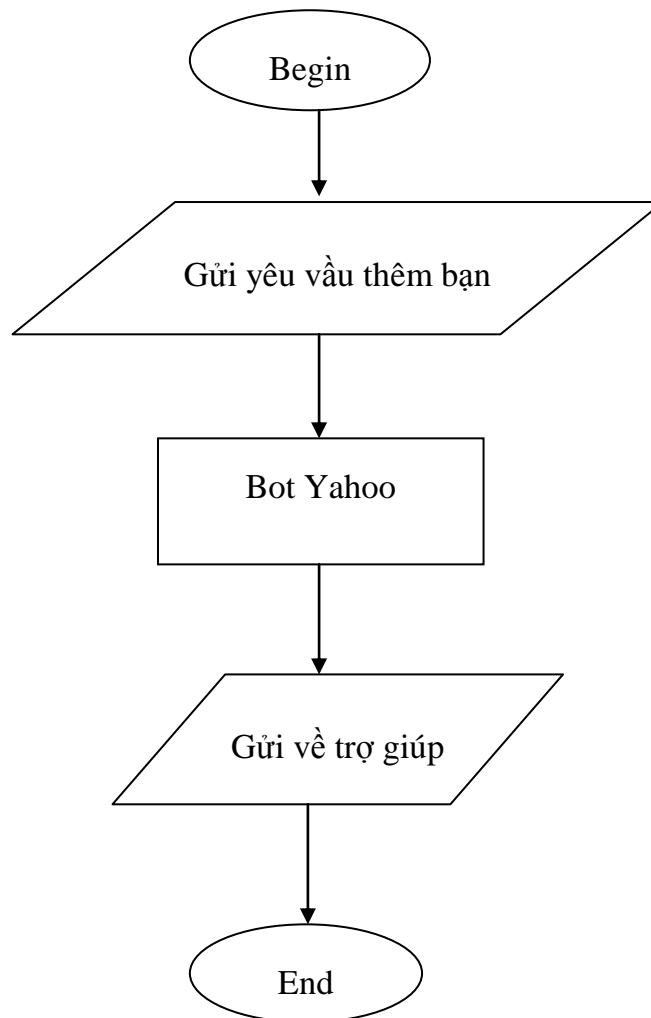
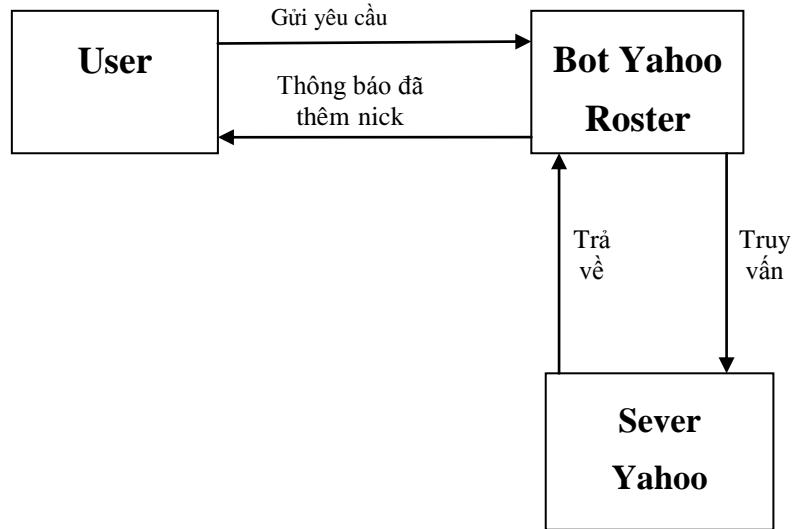
Đoạn mã thực hiện việc kiểm tra kết nối và tự động đăng nhập lại khi kết nối không còn:

```

if(S11.getSessionStatus()==SessionState.FAILED) {
    S11.login(username, password);
}
  
```

2.1.4. Quản lý danh sách nick trên Yahoo

Khi nhận được *yêu cầu thêm bạn bè* từ một người dùng mới, chương trình sẽ gửi ngay lại một trợ giúp, và *tự động thêm bạn* vào danh sách bạn bè có sẵn.



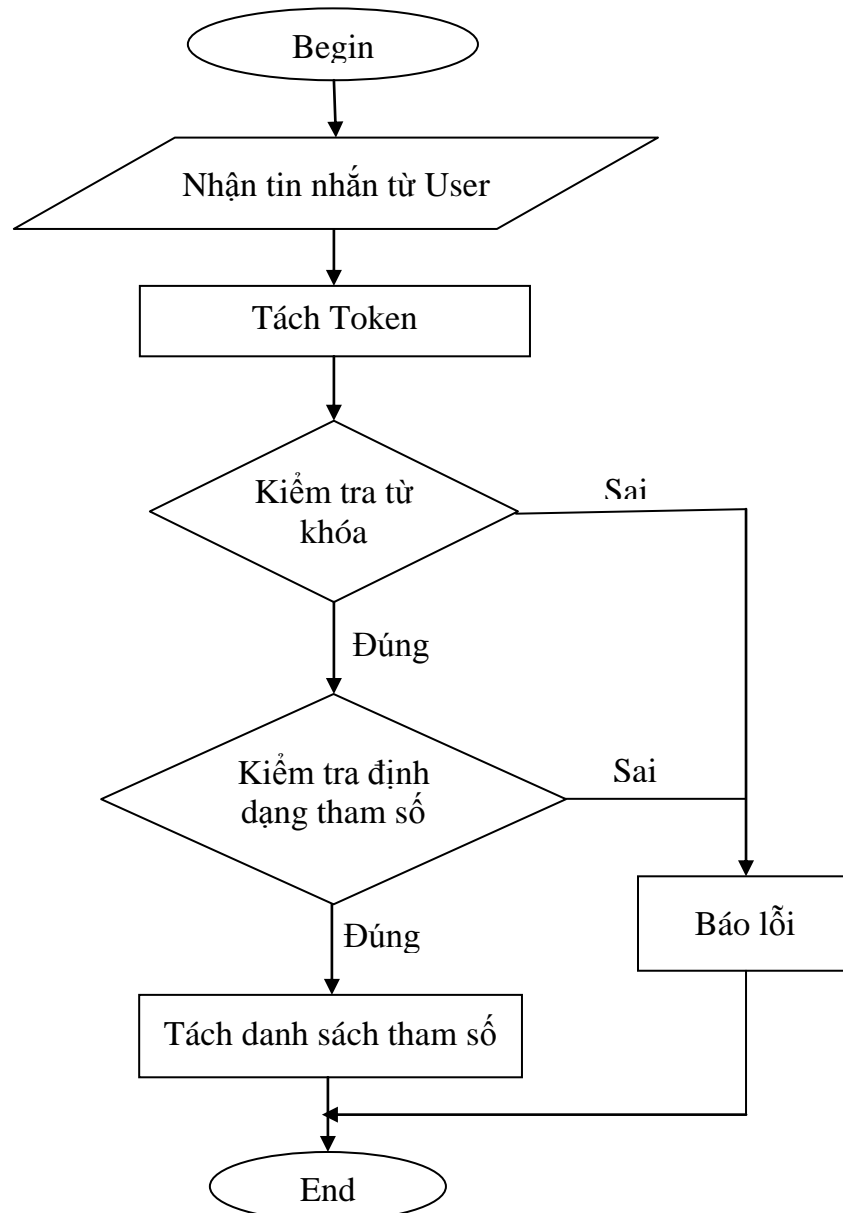
Hình 2. 5: Sơ đồ cơ chế tự động thêm nick.

2.2 Phân tích và trả lời truy vấn

Để có thể nhận và gửi trả dữ liệu cho người dùng một cách chính xác không thể bỏ qua công việc phân tích và trả lời truy vấn. Việc xử lý các chuỗi ký tự trong Java cần hỗ trợ bởi lớp String. Lớp String tập hợp các phương thức để xử lý chuỗi ký tự như tách, ghép, tìm kiếm, thay thế,...

Khi người dùng user gửi tin nhắn đến *Bot Yahoo* nhận được tin nhắn sẽ thực thi việc phân tích và kiểm tra cú pháp theo cú pháp được định sẵn theo trình tự các bước sau:

Quy trình kiểm tra truy vấn được xử lý theo quy trình như sau:



Hình 2. 6: Sơ đồ quy trình truy vấn phân tích cú pháp

Ví dụ:

- User gửi tin nhắn đến: input: TRADIEM 12345
- Cú pháp mẫu: TRADIEM [mã sinh viên]

Bước 1: Tách chuỗi ký tự :

Lớp *String* trong Java sẽ được sử dụng để tách từ trong cú pháp. Chuỗi nhận được gồm hai từ là TRADIEM và 12345. Cú pháp được thực hiện như sau

```
String s = "TRADIEM 12345";// chuỗi s là TRADIEM 12345.
```

```
String[] kq = s.split(" ");// tách chuỗi tại vị trí có dấu cách.
```

Khi nhận được xâu ký tự đầu vào thì con trỏ sẽ chạy từ đầu đến cuối xâu khi nó bắt được dấu cách thì nó sẽ tách chuỗi đó và cho vào mảng rồi lại tiếp tục cho đến khi thực hiện hết xâu đầu vào. Khi công việc tách xâu ký tự kết thúc sẽ được chuyển sang bước tiếp theo là kiểm tra từ khóa là xâu thứ nhất TRADIEM.

Bước 2: kiểm tra từ khóa vừa tách:

Lúc này từ khóa TRADIEM sẽ được đem so sánh với từ khóa cố định trong mẫu cú pháp. Và việc so sánh được thực hiện thông qua phương thức *equals()* của lớp *String* với kết quả trả về theo kiểu *Boolean*. Cú pháp được thực thi như sau:

```
String str1 = new String(TRADIEM);
```

```
String str2 = new String(TRADIEM);
```

```
Boolean k= str1.equals(str2);
```

Kết quả trả về là *k* bằng *true* vì *equals()* so sánh từng ký tự trong hai chuỗi và kết quả là hai chuỗi tương đương nhau. Ở đây phương thức *equals()* là phương thức so sánh chuỗi có phân biệt chữ hoa và chữ thường, còn phương thức *equalsIgnoreCase()* cũng là phương thức thực hiện so sánh nhưng không phân biệt chữ hoa, chữ thường.

Khi nhận được kết quả là đúng với bước kiểm tra từ khóa chuỗi thứ nhất TRADIEM sẽ chuyển sang bước tiếp theo là phân tích kiểm tra định dạng tham số biểu thức chính quy.

Bước 3: phân tích kiểm tra định dạng tham số biểu thức chính quy

Biểu thức chính quy (*Regular Expression*) mô tả một cách tường minh các xâu, được sử dụng như là ngôn ngữ vào của một số hệ thống như: *grep*, *sed*, *Lex*,

Flex,... các câu lệnh này được sử dụng tham số đầu vào có dạng biểu thức chính quy để xử lý sửa đổi, thay thế, tìm kiếm,... các chuỗi ký tự.

Một trong những ứng dụng điển hình của biểu thức chính quy là đặc tả bộ phân tích từ vựng của một chương trình dịch. Bộ phân tích từ vựng đọc chương trình nguồn và nhận biết các “token” là các chuỗi ký hiệu liên quan logic với nhau chẳng hạn như: từ khóa, danh biểu, số, ...

Một cú pháp mẫu của biểu thức chính quy được mô tả cấu trúc của chuỗi ký tự mà một biểu thức sẽ cố gắng tìm kiếm trong một chuỗi ký tự đầu vào. Một số cấu kiện mẫu phổ biến trong các chuỗi ký tự mẫu:

. bất kì ký tự nào.

X+ một hoặc nhiều X.

^ phép loại trừ các ký tự đứng sau. Ví dụ: [^0-9] (*non digit*).

[] một dải các ký tự hay chữ số.

\d bất kỳ số nào tùy chọn [0-9].

\D bất kỳ ký tự nào không phải số [^\d].

\w bất kỳ từ nào tùy chọn [a-zA-Z_0-9].

\W các ký tự không cấu thành từ. Ví dụ: [^\w] (*non word*).

\$ kết thúc chuỗi.

[abc] các ký tự a, b hoặc c.

[^abc] tất cả các ký tự trừ a, b hoặc c.

[a-zA-Z] các ký tự từ a đến z hoặc từ A đến Z.

[a-z&&[^m-p]] các ký tự từ a đến z trừ các ký tự từ m đến p.

X|Y tìm X hoặc Y.

XY tìm X trực tiếp theo sau Y.

Biểu thức chính quy có các tính chất đại số:

- Tính chất giao hoán: $a + b = b + a$.
- Tính chất kết hợp: $a + (b + c) = (a + b) + c$.
- Tính chất phân bố: $a(c + b) = ac + ab$.
- Một số tính chất khác:
 - $a + a = a$.
 - $(a^*)^* = a^*$.
 - $(a^*b^*)^* = (a + b)^*$.

- Ví dụ: các biểu thức chính quy:

$^[_A-Za-z0-9-]+\(\.[_A-Za-z0-9-]+\)^*@[A-Za-z0-9]+\(\.[A-Za-z0-9]+\)^*\(\.[A-Za-z]{2,}\)\$$ (*Email regular expression pattern*).

$^[_A-Za-z0-9-]+$: Bắt đầu một dòng với các ký tự A đến Z, a đến z, 0 đến 9 xuất hiện ít nhất một lần.

$(\.[_A-Za-z0-9-]+)^*$: Bắt đầu một nhóm thứ nhất theo bởi một dấu chấm “.” và chuỗi ký tự trong khung xuất hiện ít nhất một lần và kết thúc nhóm một với tùy chọn “*”.

$@[A-Za-z0-9-]+$: phải chứa biểu tượng “@” và chuỗi trong khung xuất hiện ít nhất một lần.

$(\.[A-Za-z0-9-]+)^*$: bắt đầu chuỗi thứ hai với dấu chấm và theo sau là ký tự trong khung [] xuất hiện ít nhất một lần với tùy chọn “*”.

$(\.[A-Za-z]{2,})&$: bắt đầu chuỗi thứ ba với dấu chấm “.” và ký tự trong khung xuất hiện ít nhất một lần với độ dài tối thiểu là 2. Kết thúc nhóm 3 bởi “&”.

$([^s]+\(\.(?i)(jpg|png|gif|bmp)\)\$)$ (*image file Extension regular expression pattern*)

$([^s]+)$: bắt đầu nhóm thứ nhất bởi dấu “(“, loại trừ khoảng trống.

$(\.(?i))$: bắt đầu nhóm thứ hai theo sau bởi dấu chấm “.”

$(jpg|png|gif|bmp)$: bắt đầu nhóm thứ ba với các chuỗi ký tự jpg hoặc png hoặc gif hoặc bmp và kết thúc chuỗi thứ ba.

$)\$$: kết thúc nhóm thứ 2, kết thúc chuỗi bởi “\$” và kết thúc nhóm 1.

- Biểu thức chính quy của ngôn ngữ Java gồm có ba lớp cốt lõi mà ta sử dụng thường xuyên hơn:

Pattern: mô tả mẫu chuỗi ký tự.

Matcher: kiểm tra một chuỗi ký tự xem có khớp với mẫu không.

PatternSyntaxException: báo một số vấn đề không thể chấp nhận được với mẫu mà ta đã thử.

Để thực hiện so sánh biểu thức chính quy trước tiên, chúng ta tạo một *Pattern* bằng cách gọi *compile()* một phương thức tĩnh trên *Pattern* với một chuỗi ký tự biểu diễn mẫu mà chúng ta muốn so khớp. Sử dụng cú pháp mẫu biểu thức chính quy. Ở đây chúng ta nên sử dụng một cấu kiện phổ biến là `\d` để kiểm tra chuỗi ký tự gồm số từ 0 đến 9.

Tiếp theo, chúng ta gọi *matcher()* trên *Pattern* đã tạo để tạo một cá thể *Matcher*. *Matcher* tìm kiếm chuỗi ký tự mà chúng ta đã chuyển cho nó để so khớp với chuỗi mẫu mà chúng ta đã dùng để tạo ra *Pattern*. Trên *Matcher* có các phương thức khác nhau để so sánh chuỗi:

- *Matches()* đơn giản cho chúng ta biết rằng toàn bộ chuỗi đầu vào có khớp với mẫu hay không.
- *Start()* cho ký tự bắt đầu trong chuỗi khớp đúng với mẫu.
- *End()* cho biết ký tự kết thúc trong chuỗi khớp với mẫu.

Quá trình được thực hiện qua đoạn cú pháp sau:

```
Pattern pattern = pattern.compile(\d);
Matcher matcher = pattern.matcher("12345");
Boolean didMatch = matcher.matches();
System.out.println(didMatch);
```

Lời gọi *matches()* trả về kết quả là *true* (đúng) vì ký tự đầu vào là một dãy số nằm trong *\d* là chuỗi ký tự mẫu gồm các chữ số. Khi nhận kết quả là đúng sẽ chuyển tiếp sang bước tiếp theo là tách danh sách tham số.

Kiểm tra định dạng tham đầu vào nếu kết quả là sai thì sẽ kết thúc bước phân tích và truy vấn và sẽ gửi thông báo lỗi về cho người dùng *User*.

Bước 4: tách danh sách tham số:

Sau khi kiểm tra định dạng tham số biểu thức chính quy chúng ta sẽ tách danh sách tham số đầu vào bằng cách sử dụng hàm *find()* của *matcher*.

Trước tiên, chúng ta tạo một chuỗi mẫu số từ 0 đến 9 là *"\d"* bởi phương thức *compile()* của lớp *Pattern*.

Sau đó gọi *matcher* trên *Pattern* bằng cách tạo một cá thể *Matcher* với chuỗi số cần tìm để tách. Phương thức *start()*, *end()* của lớp *matcher* sẽ in ra ký tự bắt đầu đến ký tự kết thúc của chuỗi đưa vào và nhóm ký tự thành từng mảng số bằng phương thức *group()*. vòng lặp *while()* được sử dụng để tìm cho đến khi kết thúc chuỗi đầu vào.

Quá trình được thực hiện qua đoạn mã sau:

```

Pattern pattern = Pattern.compile("\\d");
Matcher matcher = pattern.matcher("12345");
While (matcher.find())
{
    System.out.print("Start index: " + matcher.start());
    System.out.print("End index: " + matcher.end() + " ");
    System.out.println(matcher.group());
}

```

Quá trình tách danh sách tham số trên sẽ cho ta kết quả được các chuỗi có chứa số tương ứng với điều kiện mẫu "\\d". Kết thúc quá trình tách danh sách tham số chương trình sẽ được chuyển sang bước truy vấn cơ sở dữ liệu.

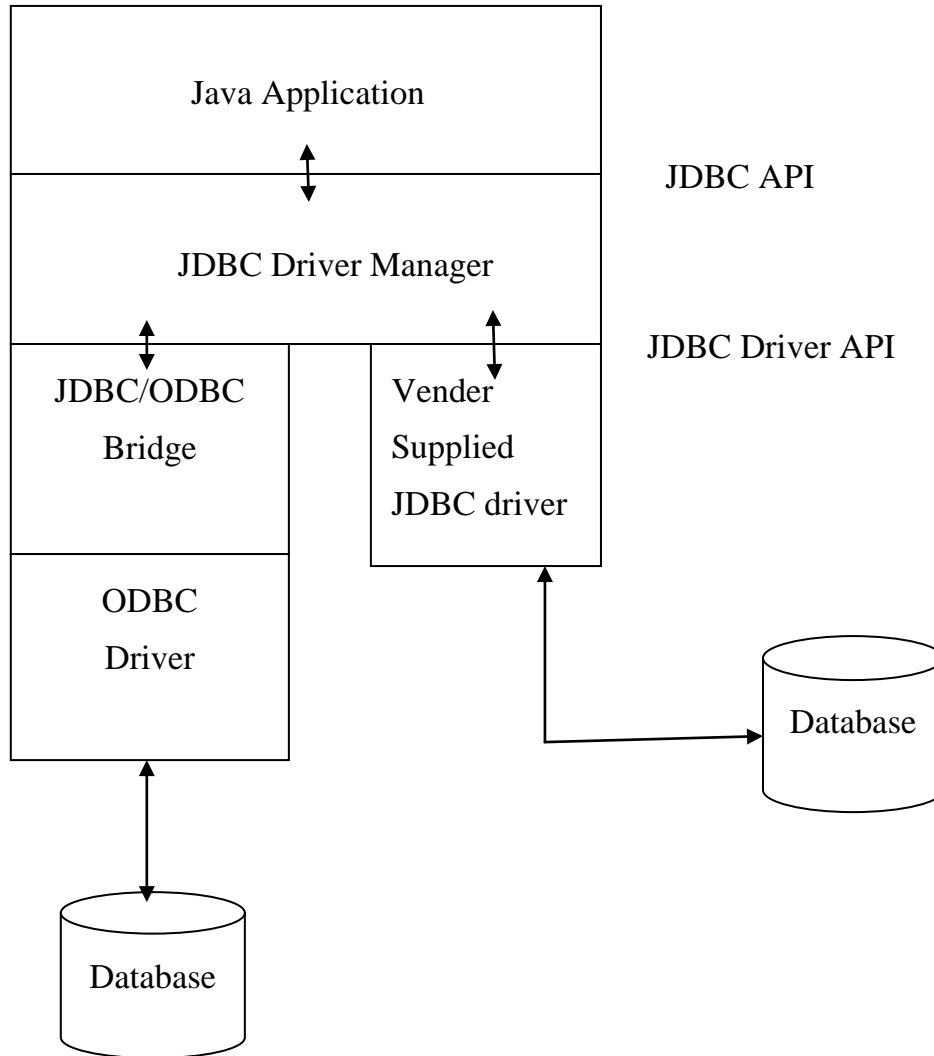
2.3 Tương tác cơ sở dữ liệu

2.3.1. Cơ chế hoạt động:

Kiến trúc của JDBC tương tự như kiến trúc ODBC do Microsoft xây dựng. Theo kiến trúc này các thao tác liên quan đến cơ sở dữ liệu trong chương trình được thực hiện thông qua các JDBC API. Sau đó các JDBC API sẽ truyền các yêu cầu của chương trình đến bộ quản lý trình điều khiển JDBC, là bộ phận có nhiệm vụ lựa chọn trình điều khiển thích hợp để có thể làm việc với cơ sở dữ liệu mà chương trình muốn kết nối.

Kiến trúc của JDBC gồm hai tầng:

- Tầng 1: là các JDBC API, có nhiệm vụ chuyển các câu lệnh SQL cho bộ quản lý trình điều khiển JDBC.
- Tầng 2: các JDBC Driver API, thực hiện nhiệm vụ liên hệ với trình điều khiển của hệ quản trị cơ sở dữ liệu cụ thể.



Hình 2. 7: Kiến trúc của JDBC.

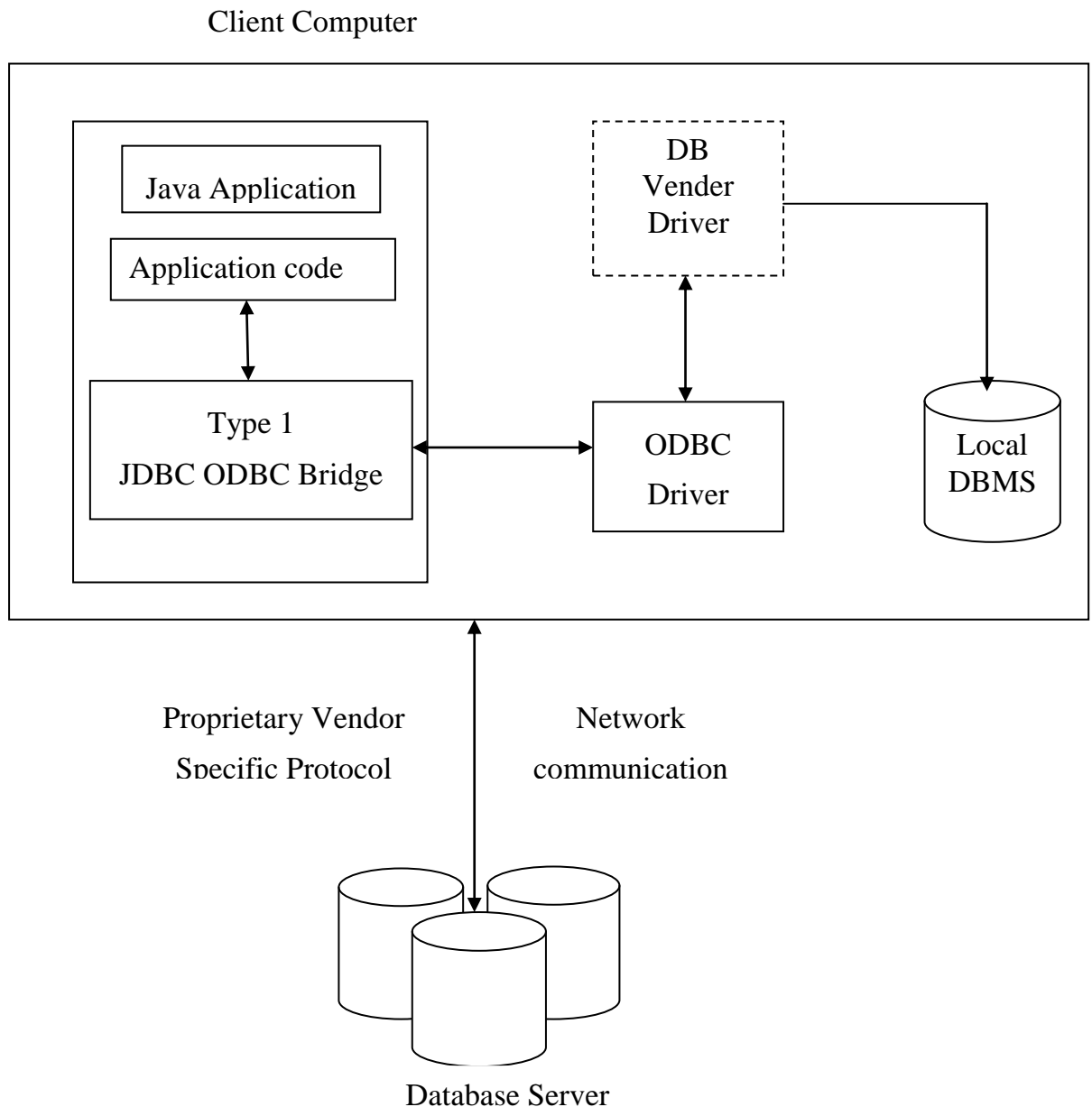
JDBC có cơ chế hoạt động tương tự như ODBC nhưng thông qua các đối tượng:

- Chương trình Java tạo một đối tượng kết nối (*connection object*) để thực hiện kết nối với cơ sở dữ liệu.
- Tạo một đối tượng câu lệnh (*Statement*) và chuyển các câu lệnh SQL đến cơ sở dữ liệu thông qua đối tượng này và nhận về kết quả. Để giảm thời gian trễ, các tập tin JDBC class thường được nạp sẵn trên máy khách.
- Các lớp JDBC nằm trong khối `java.sql` và mọi chương trình java đều dùng các đối tượng và phương thức trong khối `java.sql` để đọc và viết lên nguồn dữ liệu.

Trong chương trình này Java dùng JDBC sử dụng trình điều khiển thư viện cầu nối JDBC – ODBC cho nguồn dữ liệu cần giao tiếp.

Cầu nối JDBC – ODBC: cung cấp các truy cập JDBC thông qua các truy cập ODBC. Vì vậy trong cơ sở dữ liệu máy khách, các lệnh ODBC và lệnh cơ sở dữ liệu phải được nạp trước trong máy khách đó và luôn được cung cấp kèm trong bộ J2SE với tên: *sun.jdbc.odbc.JdbcOdbcDriver*.

Cơ chế hoạt động của JDBC – ODBC được thể hiện qua hình sau:



Hình 2. 8: Cơ chế hoạt động của JDBC – ODBC

2.3.2. Kết nối cơ sở dữ liệu:

Để kết nối với cơ sở dữ liệu, JDBC đòi hỏi hai yếu tố là trình điều khiển driver (tương ứng với các kiểu kết nối) và thông tin để kết nối (như địa chỉ máy chủ, tên tài khoản đăng nhập và mật khẩu...).

Có hai cách để tạo kết nối SQL Sever trong Java sử dụng JDBC – ODBC bridge

Cách 1: tạo một “**System data source**” thông qua công cụ “**ODBC Data Source Administrator**” của hệ điều hành Windows, sau đó viết một chương trình bằng Java thực hiện kết nối đến DNS (*Data Source Name*) đã tạo và đọc dữ liệu từ Database.

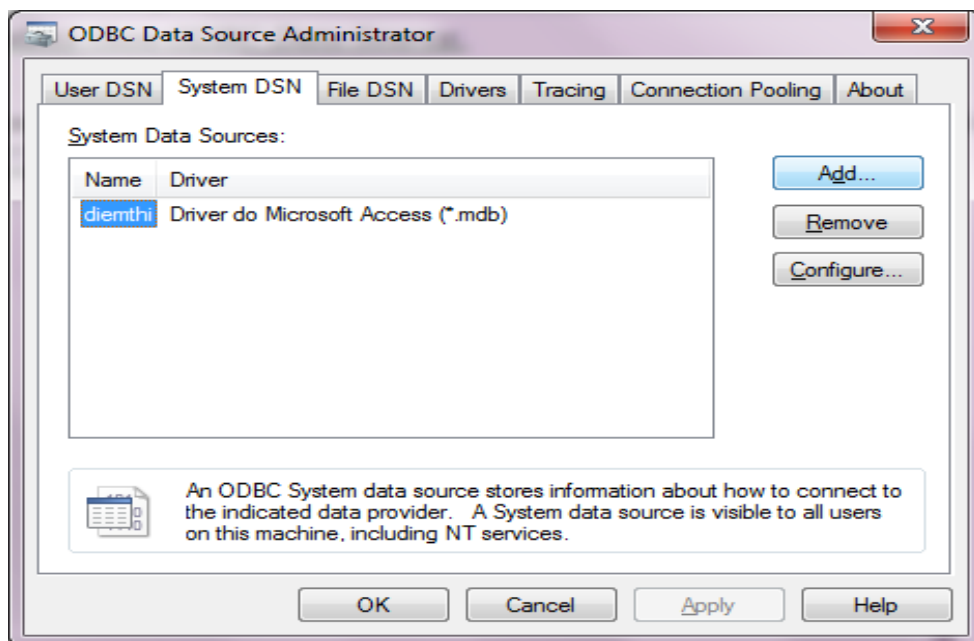
Cách 2: sử dụng chuỗi kết nối trực tiếp đến Database theo cú pháp có dạng

“**jdbc:odbc:driver={SQLSever};Sever=<SeverName>;Database=<YourDB>;UserName=<dbUser>;Password=<*>**”

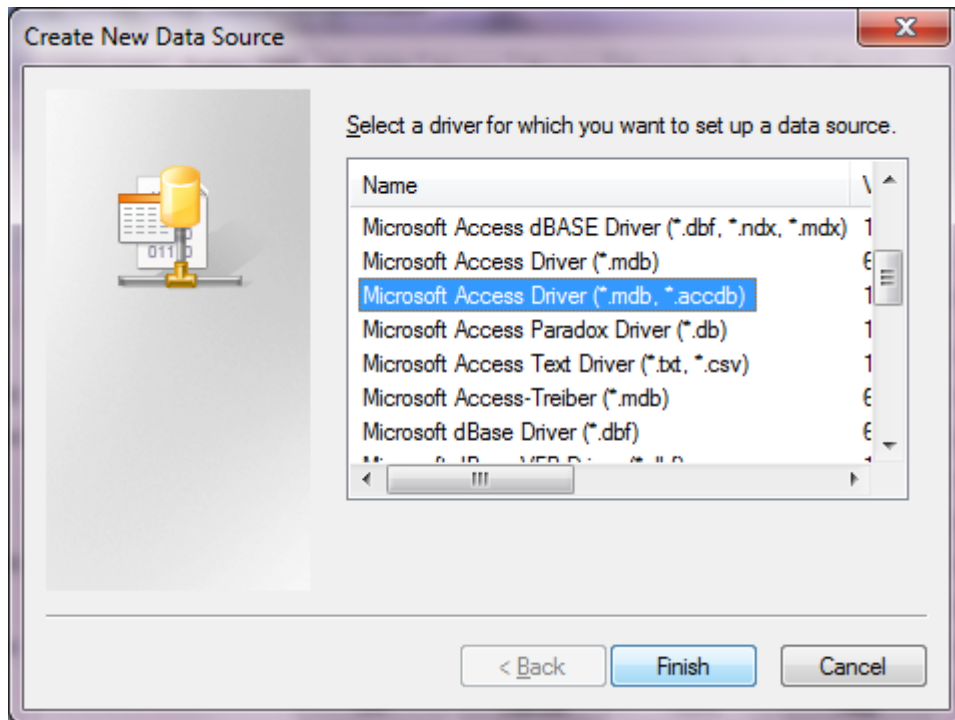
Trong chương trình này ta sử dụng cách thứ nhất kết nối database sử dụng JDBC -ODBC dựa trên DNS. Được thực thi qua hai bước chính là: đăng ký trình điều khiển JDBC; và thực thi phương thức *getConnection()* của lớp *DriverManager*.

Bước 1: thực hiện kết nối :

Đầu tiên cần phải tạo DataSource Name cho cơ sở dữ liệu bằng cách vào *Control Panel* và chọn *ODBC Data Source*.

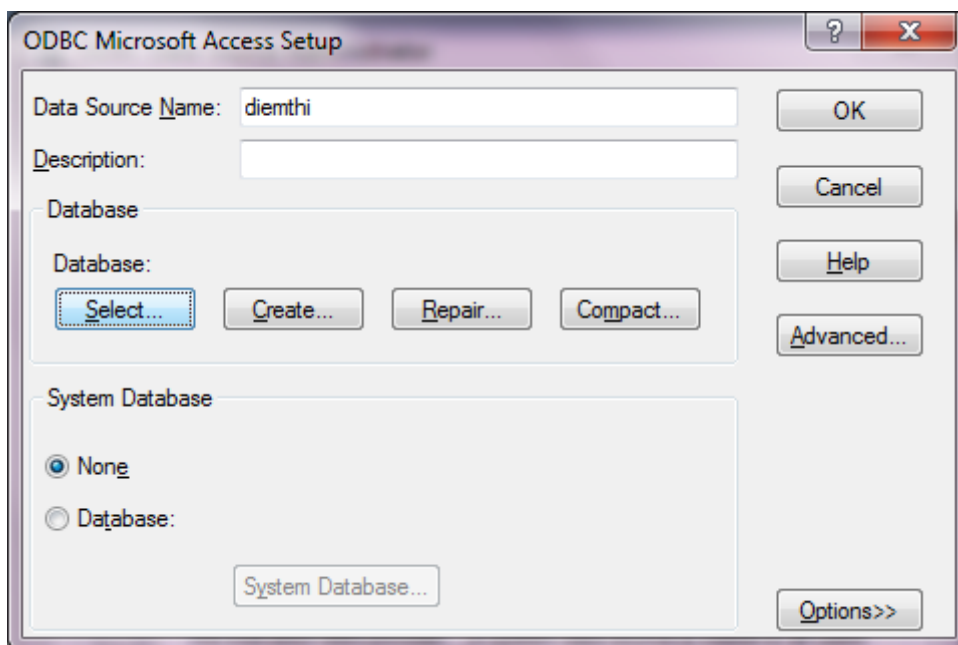


Chọn **Add** khi này danh sách các trình điều khiển cơ sở dữ liệu hiện có sẽ hiển thị:

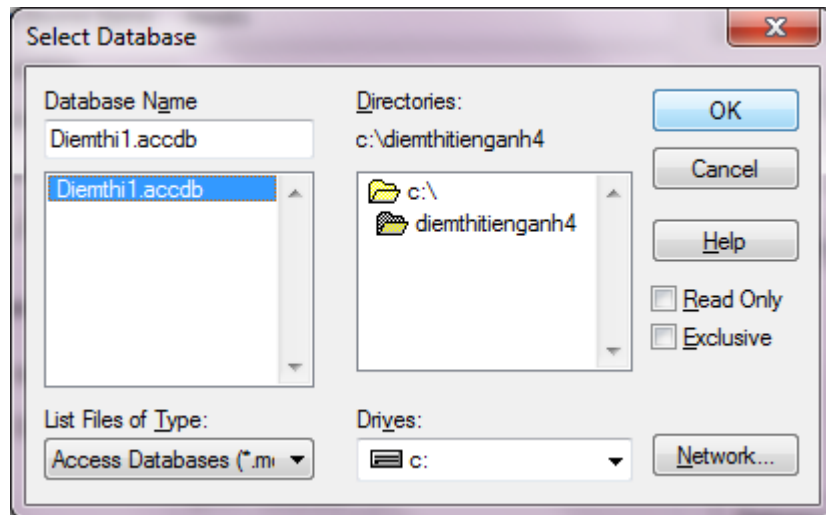


Ta chọn *Microsoft Access Driver (*.mdb, *.accdb)* hoặc trình điều khiển nào ta sử dụng và nhấn *Finish*.

Cửa sổ cấu hình cho tập tin Access sẽ xuất hiện và nhập tên file dữ liệu vào ô *Data Source Name*.



Chọn *Select* cửa sổ Select Database xuất hiện, bạn chọn tập tin cơ sở dữ liệu cần tạo Data Source Name cuối cùng nhấn OK để kết thúc.



Sau khi đã tạo DNS cho tập tin diemthi1.accdb, chúng ta chuyển sang bước tiếp theo là xây dựng đoạn mã để tiến hành kết nối với tập tin diemthi1.accdb.

Bước 2: Để có thể thực thi các hàm ta phải import các thư viện:

```
java.sql.Connection; java.sql.DriverManager; java.sql.SQLException;
```

```
private Connection con=null;// khai báo Connection bằng rỗng.
```

```
private String connectionURL = "jdbc:odbc:diemthi1";
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); // đăng ký trình điều khiển để sử dụng câu lệnh.
```

```
Connection = DriverManager.getConnection(dbURL,"","");
```

```
sqlQuery = "SELECT * FROM diemthi1" + " WHERE '" + decodedMessage +  
"=MASV";// truy vấn vào cơ sở dữ liệu để lấy trường từ bảng diemthi1 với điều kiện mã sinh viên khớp với mã sinh viên trong bảng.
```

CHƯƠNG 3: CHƯƠNG TRÌNH ỨNG DỤNG

3.1 Yêu cầu phần cứng, phần mềm

a. Yêu cầu phần cứng:

Chip P₄ 1,6GHz

RAM 512 MB

Card đồ họa 64 MB

Directx 9.0

b. Yêu cầu phần mềm:

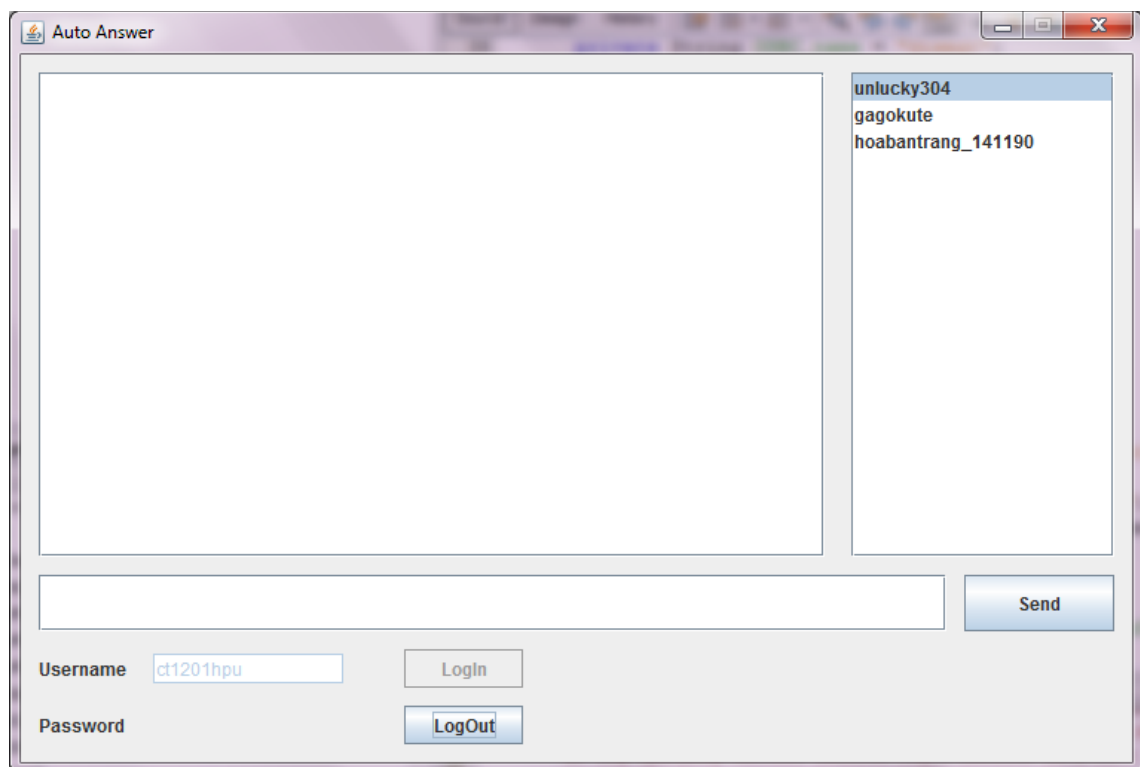
Hệ điều hành tối thiểu: Windows 2000.

Môi trường lập trình Java SE Development Kit JDK.

Môi trường phát triển tích hợp Netbeans 7.2.

3.2 Giao diện chương trình

a. Giao diện chính:



Chúng ta đăng nhập Bot Yahoo với tên đăng nhập là “CT1201HPU” và password là “trang121181”.

Mã nguồn đăng nhập:

```

S1 = new Session();
System.out.println(S1.getSessionStatus().toString());

if(S1==null)
{
    JOptionPane.showMessageDialog(JF, "Khong the tao duoc ket noi!");
}

try {
    S11.login(username, password);
    S11.setStatus("I'm DHP Bots", false);
} catch (IllegalStateException | IOException | FailedLoginException ex) {
    Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
} catch (AccountLockedException ex) {
    Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
} catch (LoginRefusedException ex) {
    Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
}

S11.addSessionListener(new Nhan_Event());

```

Mã nguồn nạp danh sách các Nick:

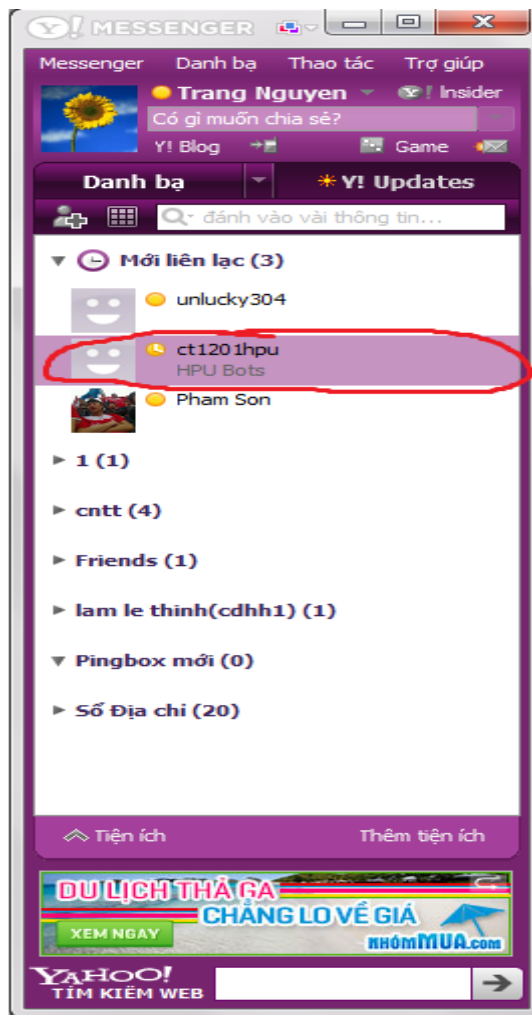
```

RT = S11.getRoster();

for(YahooUser yu: RT.toArray(new YahooUser[RT.size()]))
{
    JL1.addElement(yu.getId());
}

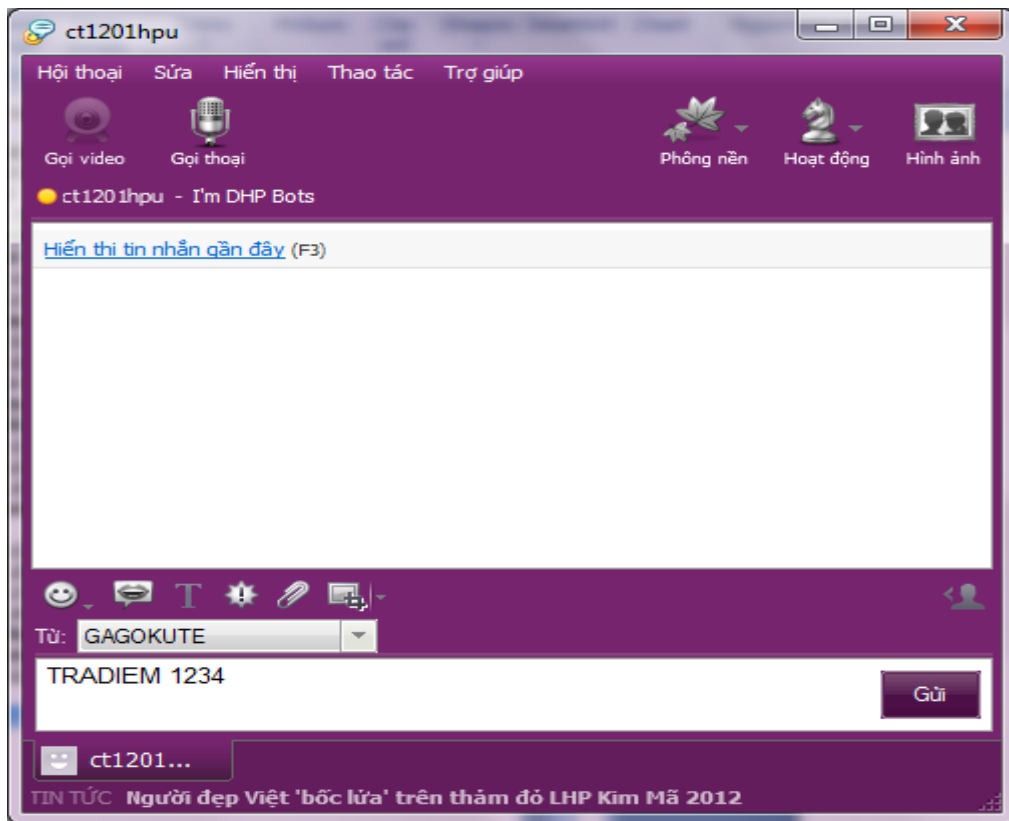
```

Giao diện của người dùng add nick Bot Yahoo “CT1201HPU”:

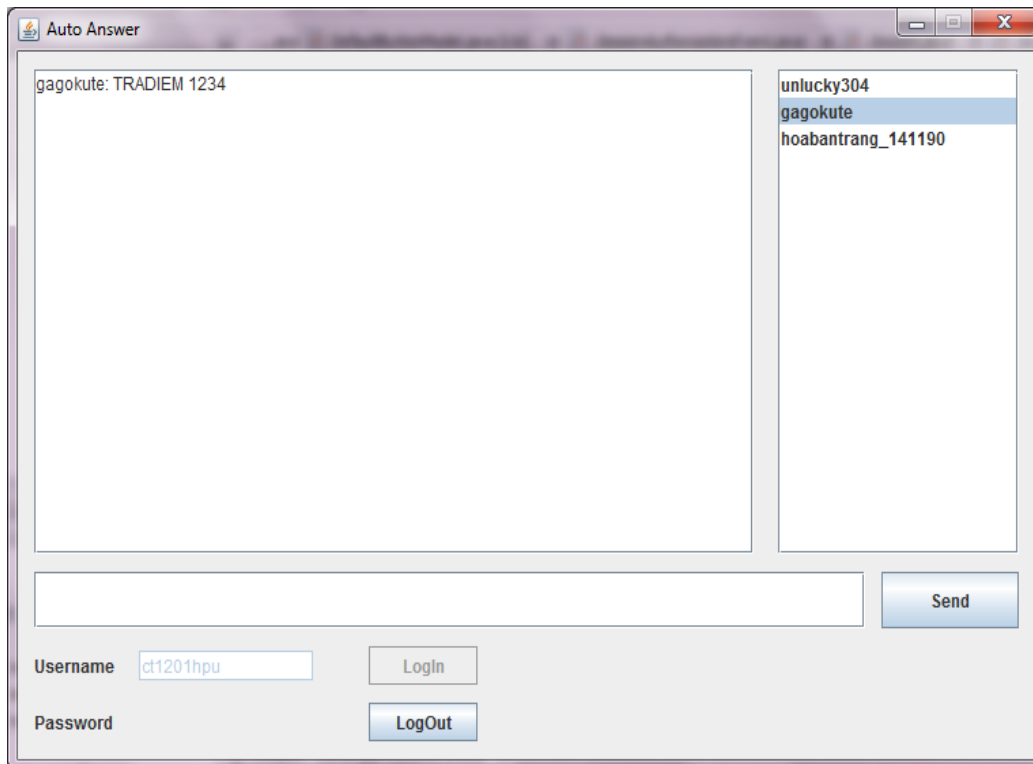


Người dùng dịch vụ của hệ thống, sử dụng chương trình Yahoo Messenger được tải và cài đặt miễn phí tại trang chủ của Yahoo tại địa chỉ www.yahoo.com. Chương trình này là chương trình chat miễn phí được sử dụng rất phổ biến tại Việt Nam. Người dùng đầu cuối sử dụng chương trình này để nhắn tin theo cú pháp đến hệ thống trả lời tự động để lấy các thông tin cần thiết.

b. Giao diện gửi tin nhắn đến Bot Yahoo:



c. Giao diện nhận tin nhắn của Bot Yahoo:



Nạp chồng phương thức nhận thông điệp từ người dùng đầu cuối:

```

@Override public void messageReceived(SessionEvent event) {

    Pattern p1;

    Matcher m1;

    Statement st1;

    ResultSet rec;

    String inputmsg = event.getMessage();

    String s = "(\\b[0-9C]*\\b)";

    if( inputmsg.split(" ")[0].compareToIgnoreCase("TRADIEM") == 0)

    {

        p1 = Pattern.compile(s);

        m1 = p1.matcher(inputmsg);

        while(m1.find()) {

            if(!m1.group().toString().isEmpty())

            {

                System.out.println(m1.group());

                try {

                    st1 = con1.createStatement();

                    rec = st1.executeQuery("SELECT Hoten, Diem FROM diemqt WHERE

                        Masv='"+m1.group()+"'");

                    while (rec.next()) {

                        try {

                            S11.sendMessage(currentNick, rec.getString(1) + ":\t" + rec.getString(2));

                        } catch (IllegalStateException | IOException ex) {

                            Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);

                        }

                    }

                }

                System.out.println(i);

                st1.close();

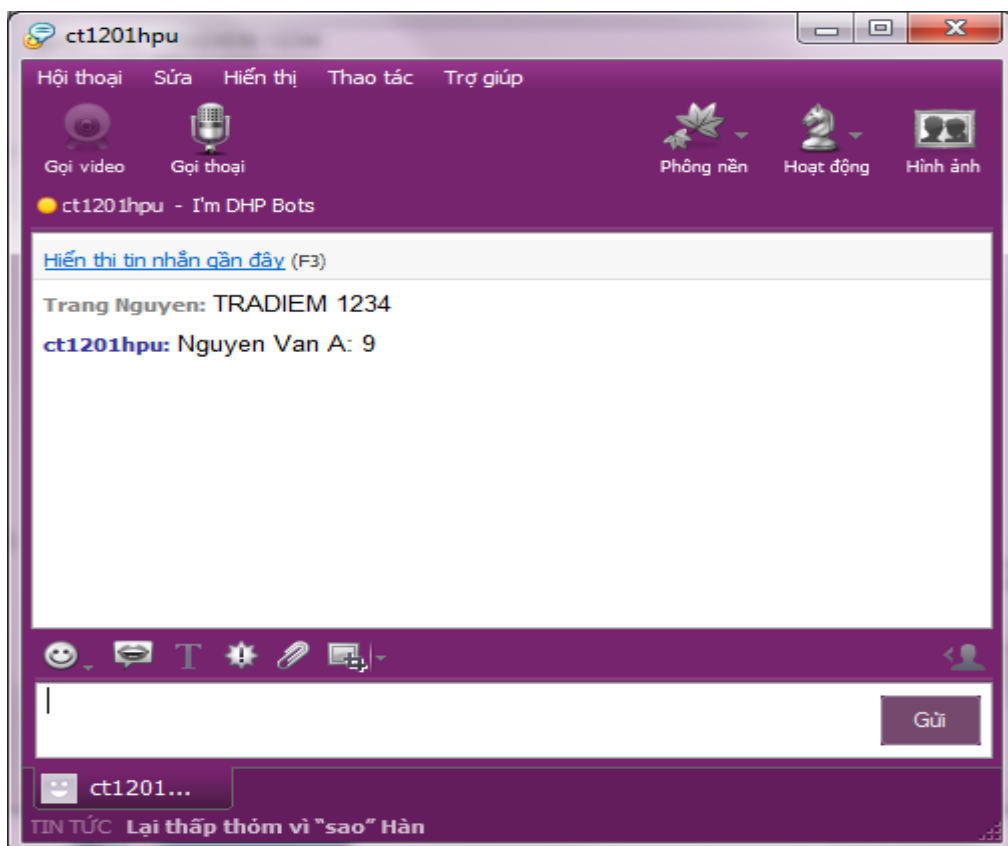
```

```

    } catch (SQLException ex) {
        System.out.println("LOI SQL"+ex.getMessage());
        Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
}
} //comprare IF
else
{
    S11.sendMessage(currentNick, "Sai cú pháp! TRADIEM [masinhvien]");
}
}
}

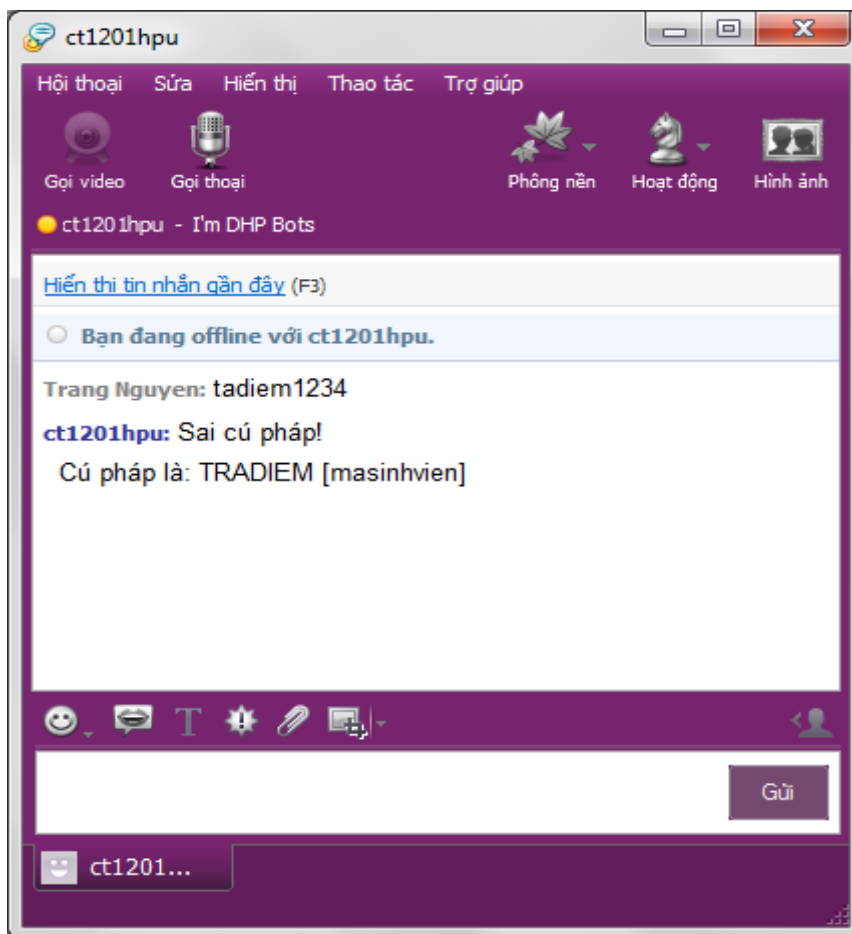
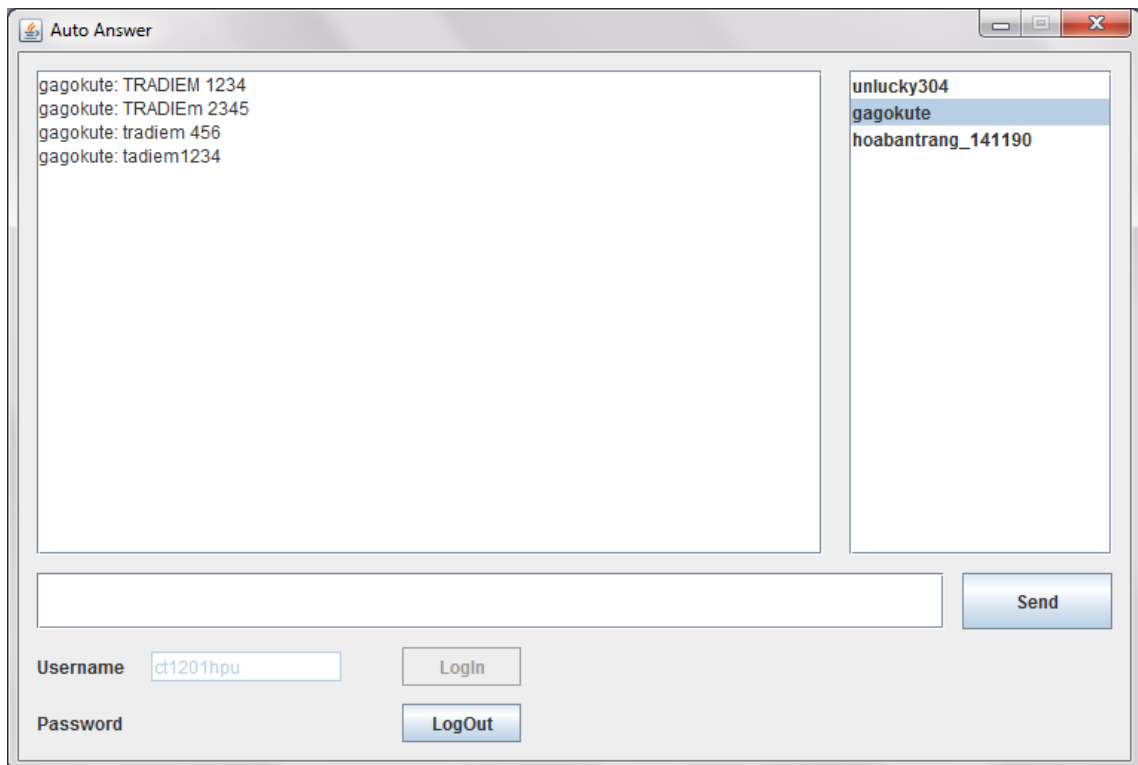
```

d. Giao diện nhận tin nhắn trả lời tự động từ Bot Yahoo:



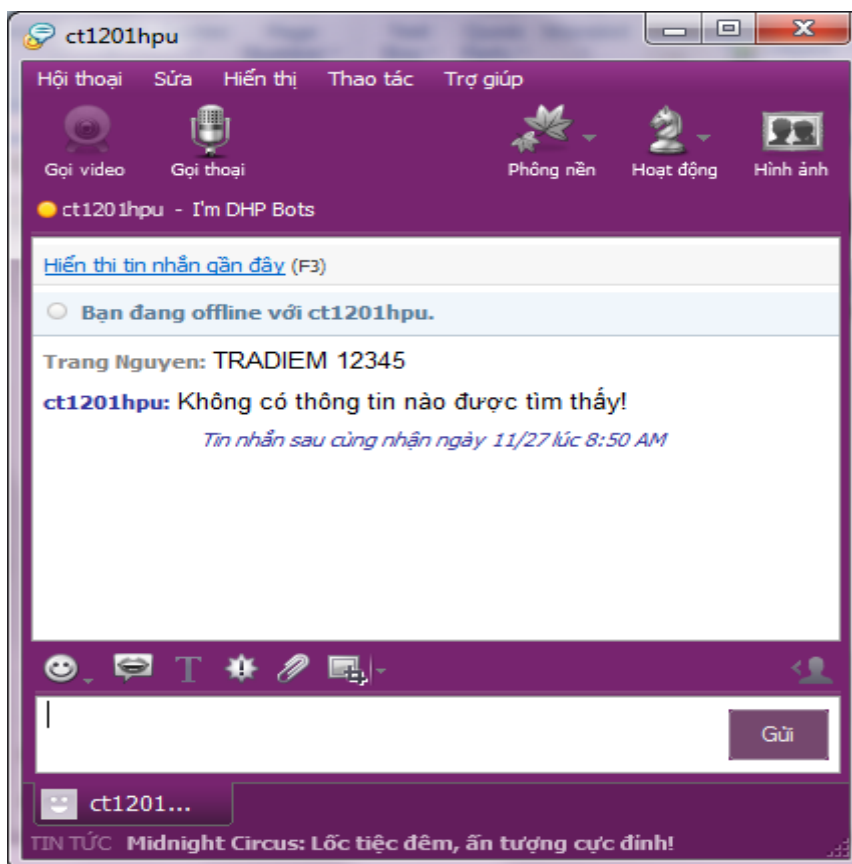
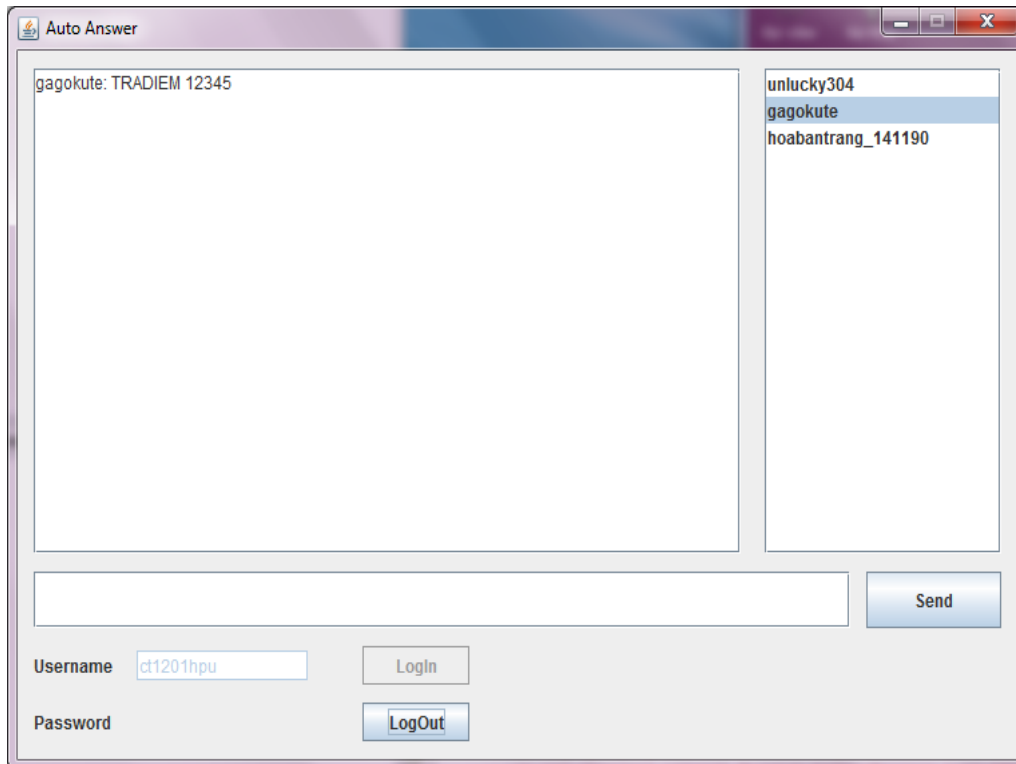
Khi chúng ta gửi tin nhắn yêu cầu đúng cú pháp “TRADIEM [mã sinh viên]” thì Bot Yahoo sẽ trả về họ tên và điểm quá trình của sinh viên đó.

e. Giao diện gửi và nhận tin nhắn sai cú pháp:



Khi chúng ta gửi yêu cầu sai cú pháp thì Bot Yahoo sẽ gửi thông báo là sai cú pháp và hướng dẫn cú pháp đúng.

f. Giao diện nhận và trả lời khi không có cơ sở dữ liệu cần tìm:



Một số phương thức được nạp chồng trong lớp SessionListener để xử lý các sự kiện của phiên làm việc Session:

```
class Nhan_Event extends SessionAdapter {
    @Override public void messageReceived(SessionEvent event) {
        //Phương thức xử lý sự kiện nhận thông điệp.
    }
    @Override public void offlineMessageReceived(SessionEvent se) {
        //Phương thức xử lý sự kiện nhận thông điệp offline.
    }
    @Override public void connectionClosed(SessionEvent se) {
        //Phương thức xử lý sự kiện ngắt kết nối.
    }
    @Override public void friendAddedReceived(SessionFriendEvent sfe) {
        //Phương thức xử lý sự kiện add friend
    }
    @Override public void contactRequestReceived(SessionAuthorizationEvent se) {
        //Phương thức xử lý sự kiện yêu cầu kết bạn
    }
}
```


KẾT LUẬN

Với đề tài “Xây dựng hệ thống trả lời tự động thông qua Yahoo Messenger”. Em đã mang những kiến thức được học ở nhà trường đem vận dụng vào thực tế để xây dựng chương trình này. Qua đó em có điều kiện trau dồi, nâng cao kiến thức đã học.

Với yêu cầu của đề tài thực tế để xây dựng thêm ứng dụng Yahoo Messenger em đã đạt được một số kết quả sau:

- Tìm hiểu và làm quen về môi trường phát triển tích hợp Netbeans IDE 7.2.
- Tìm hiểu các cơ chế làm việc, lập trình của Yahoo.
- Tìm hiểu cách thức tạo chương trình cho phép tương tác với Yahoo, nhằm triển khai cơ chế trả lời tự động.
- Xây dựng được chương trình tra cứu điềm cho sinh viên thông qua Yahoo Messenger để ứng dụng thực tiễn.

Sau khi xây dựng, chương trình đã có thể thực hiện được các chức năng giao tiếp với người dùng thông qua hệ thống máy chủ Yahoo, gửi và nhận tin nhắn, trả lời tự động với nội dung phù hợp nhằm đáp ứng nhu cầu tra cứu thông tin của người dùng.

Tuy nhiên, để có thể ứng dụng rộng rãi hơn trong các trường hợp khác nhau đòi hỏi chương trình phải có tính linh hoạt hơn nữa trong thao tác trả lời tự động và làm việc với các cơ sở dữ liệu khác nhau.

Sau đây là một số hướng phát triển của đề tài:

- Tiếp tục hoàn thiện các chức năng của chương trình.
- Xây dựng chức năng cho phép chương trình làm việc với các hệ cơ sở dữ liệu khác nhau.
- Có cơ chế phòng vệ tốt trong việc phòng ngừa những tấn công từ người dùng với mục đích phá hoại hoạt động của chương trình.

Trong thời gian tới, em sẽ tiếp tục phát triển chương trình nhằm hoàn thiện các chức năng của chương trình để chương trình hoạt động tốt hơn.

Do kiến thức và thời gian còn hạn chế nên đồ án tốt nghiệp của em không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đánh giá, đóng góp của các thầy cô giáo và các bạn để đồ án của em được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

1. <http://sourceforge.net/apps/trac/openymsg> Java Yahoo Messenger Library (OpenYMSG).
2. <http://netbeans.org/community/releases/72/> NetBeans IDE 7.2.1 Release Information.
3. Java.sun.com.
4. Lập trình Java nâng cao – Đoàn Văn Ban – nhà xuất bản Khoa Học và Kỹ Thuật.
5. Java tập 1, 2 (Ấn bản dành cho sinh viên) – nhà xuất bản Thống kê – Phương Lan (chủ biên).