

Zagazig University

Faculty OF Computers & Informatics

Information Technology Department

2013

Secure Video conference

Project Supervisor

DR| Waleed Kheder

Team Advisor

ENG| Mahmoud Mahdi

2012/2013

ZAGAZIG UNIVERSITY

FACULTY OF COMPUTERS & INFORMATICS

INFORMATION TECHNOLOGY DEPARTMENT

Dr. Waleed Khedr

Project Advisor

Eng. Mahmoud Wardi

SVC

SECURE VIDEO CONFERENCE

(VIDEO CHAT)

Report prepared by Mohamed

Department of Information Technology

PROJECT SUPERVISOR

Dr \ Waleed Khedr

TEAM ADVISOR

Eng\ Mahmoud Mahdi

PROJECT TEAM

Haitham Mohamed Kamel

Enas Mostafa Mohamed Bahgt

Asmaa El-Sayed Lotfi

Rwida Mohamed Mohamed

Department of information technology

2013

Perfice:

Secured conference pattern aimed to provide security during conference and video conference between two user

In very complex applications, there are some problems for developers, unless the application parses a few problems according to some criteria used in the applications become more complex, so it is better to divide the application into several parts.

تحت إشراف:

١/ وليد خضر

م/ محمود مهدي

فريق العمل

١. هيثم محمد كامل عبد الرحمن
٢. إيناس مصطفى محمد بهجت
٣. أسماء السيد المطفي رزق
٤. رويدا محمد محمد عوض

1.1 overview of secure video conference:

Secure video conference is a design pattern aimed to provide security during conference , and also separate video conference between more user.

In very complex all applications , the chaos of layers is really confusing for developers . unless the application phases are separating according to some criteria , updating the applications becomes a nightmare .these problems are noticeable in web and enterprise applications.

1.2 SVC architecture :

1)Secure model:

In this layer providing the security for the user of the program as user will be able to chat with other people without afraid of any stolen in call.

2)Video / audio model:

This model is represented by video & audio streaming and JMF " java media framework" provide this application to the program , as it is easy to chat with people

JMF package is used to develop softwares related to audio &video

3)Conference chat model:

This able to provide the user to chat with one or more other users.

1.3 overview of design patterns & the SVC design pattern:

Design patterns

A pattern is a proven solution to a problem in a context.

Design a pattern represent a solution to problems that arise when developing software within a particular context

i.e. patterns = problems , solution pairs in context

design a pattern use a consistent documentation approach

design a pattern

are often organized as creational , structural or behavioral

DETAILS OF SVC DESIGN PATTERN

- Name (essence of the pattern)
Secure video conference
- Context (where does this problem occur)

SVC Secure video conference is a design pattern aimed to provide security during conference , and also separate video conference between more user.

- Problem (definition of the reoccurring difficulty)

Define how to interface with the program and how to use it

- Solution (how do you solve the problem)

Use software engineering principle of "separation of concerns " to divide the application into 3 layers :

- Security
- Streaming
- Multiple video chat

3.1.3 RTP Applications

RTP applications are often divided into those that need to able to receive data from the network (RTP Clients) and those that need to be able to transmit data across the network (RTP Servers). Some applications do both.

3.1.4 RTP API Architecture

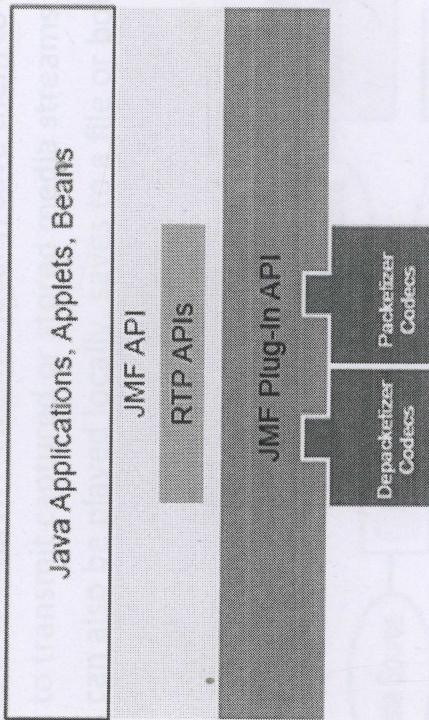


figure: High-level JMF RTP architecture

JMF enables the playback & transmission of RTP streams through the APIs defined in the `javax.media.rtp`, `javax.media.rtp.event` and `javax.media.rtp.rtcp` packages. JMF RTP APIs are designed to work with the capture, presentation & processing capabilities of JMF. Players & Processors are used to present and manipulate RTP media streams. JMF can be extended to support additional RTP-specific formats and dynamic payloads through the standard plug-in mechanism.

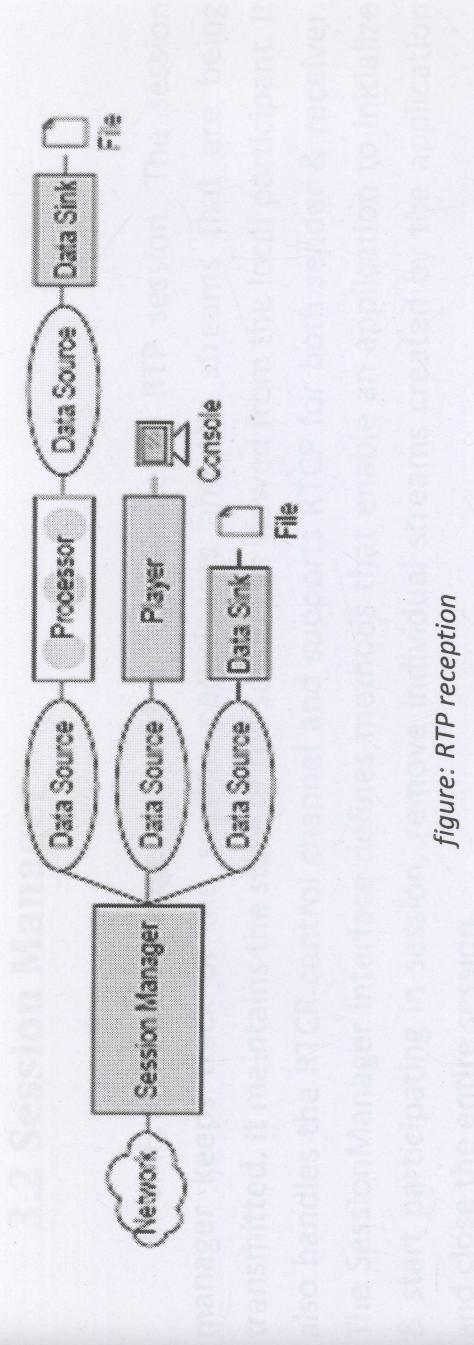


figure: RTP reception

User can play incoming RTP streams locally, save them to a file or both. Similarly they can use APIs to transmit captured or stored media streams across network. The outgoing streams can also be played locally, saves to a file or both.

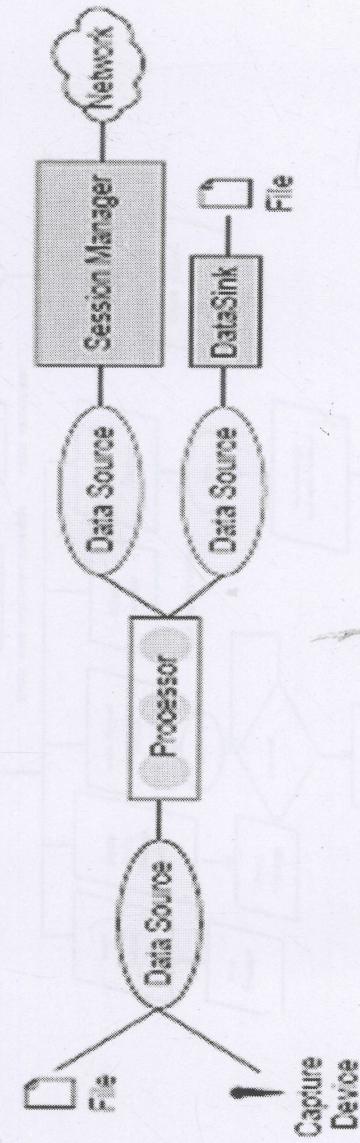
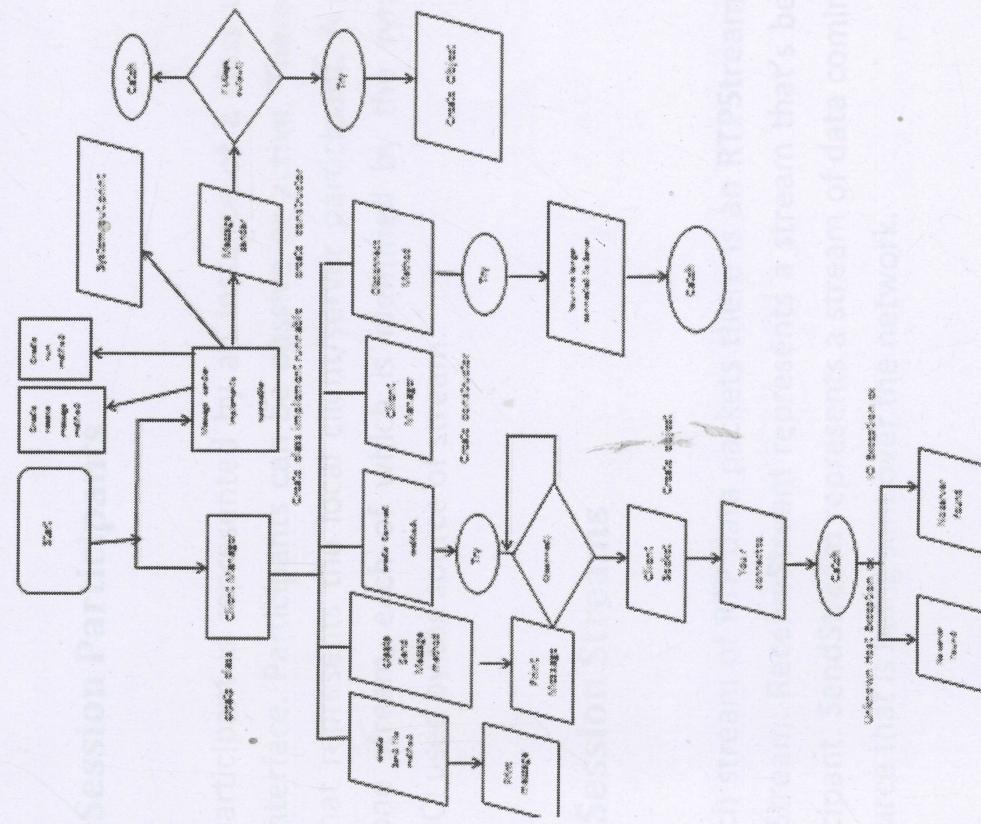


Figure: RTP transmission

3.2 Session Manager

In JMF, a **SessionManager** is used to coordinate an RTP session. The session manager keeps track of the session participants and the streams that are being transmitted. It maintains the state of the session as viewed from the local participant. It also handles the RTCP control channel and supports RTCP for both sender & receiver. The SessionManager interface defines methods that enable an application to initialize & start participating in session, remove individual streams created by the application and close the entire session.



卷之三

Print view							Propose table structure		Operations		Import		Export		Insert		Search		Structure		SQL		Browse		
Add 1		field(s)		@ At End of Table		@ At Beginning of Table		After id		>		Go													
Field	Type	Collation	Attributes	Null	Default	Extra	Action	Field	Type	Usage	Space usage	Statements	Value	Action	Field	Type	Usage	Space usage	Statements	Value	Action	Field	Type	Usage	Space usage
<input type="checkbox"/> id	int(14)			No		auto_increment	<input checked="" type="checkbox"/>																		
<input type="checkbox"/> user_name	varchar(255)	utf8_general_ci		No			<input checked="" type="checkbox"/>																		
<input type="checkbox"/> password	varchar(255)	utf8_general_ci		No			<input checked="" type="checkbox"/>																		
<input checked="" type="checkbox"/> Check All / Uncheck All With selected:																									

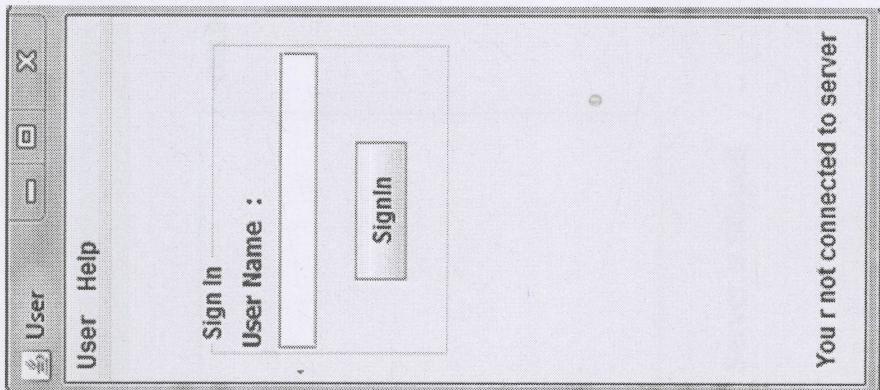
Figure: access data base using user name, and password

2) the user start to identify his data on database

Such as (user name, password , email)

Print view							Propose table structure		Operations		Import		Export		Insert		Search		Structure		SQL		Browse		
Add 1		field(s)		@ At End of Table		@ At Beginning of Table		After id		>		Go													
Field	Type	Collation	Attributes	Null	Default	Extra	Action	Field	Type	Usage	Space usage	Statements	Value	Action	Field	Type	Usage	Space usage	Statements	Value	Action	Field	Type	Usage	Space usage
<input type="checkbox"/> id	int(14)			No		auto_increment	<input checked="" type="checkbox"/>																		
<input type="checkbox"/> user	varchar(255)	utf8_general_ci		No			<input checked="" type="checkbox"/>																		
<input type="checkbox"/> pass	varchar(255)	utf8_general_ci		No			<input checked="" type="checkbox"/>																		
<input type="checkbox"/> mail	text	utf8_general_ci		No			<input checked="" type="checkbox"/>																		
<input checked="" type="checkbox"/> Check All / Uncheck All With selected:																									

❖ The program look like



❖ User can control the server to connect to it or not

We have 2 states of the server

1) Start connection

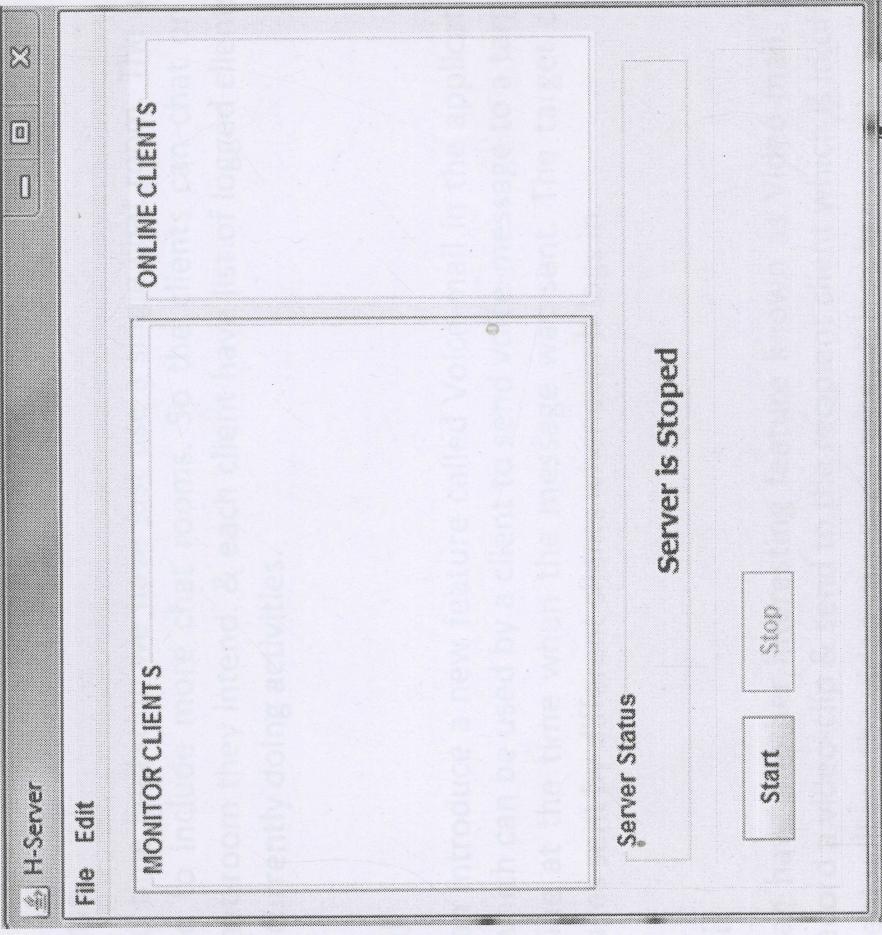
Server Status

Server is Listening on port : 12345
<input type="button" value="Start"/>
<input type="button" value="Stop"/>

2) Stop connection

Server Status
Server is Stopped
<input type="button" value="Start"/>
<input type="button" value="Stop"/>

- ❖ When the connection is completed the form of program will be :



The Transfer

We want to give facility of transferring files or connection between the clients. The files may be a text document, video files and other types of files. Various clients like audio, video conference or video conference can send & receive files to one another.

Conclusion

We have described our components as an application of Java programming language. Due to unavailability of some tools and lack in framework & Realizing limitations of Java, we have not implemented Chatting and Video Conference. However this kind of feature can be implemented in future more efficiently. Conference and video conference can be implemented in Java and its various platforms.

4.3 Diagrams

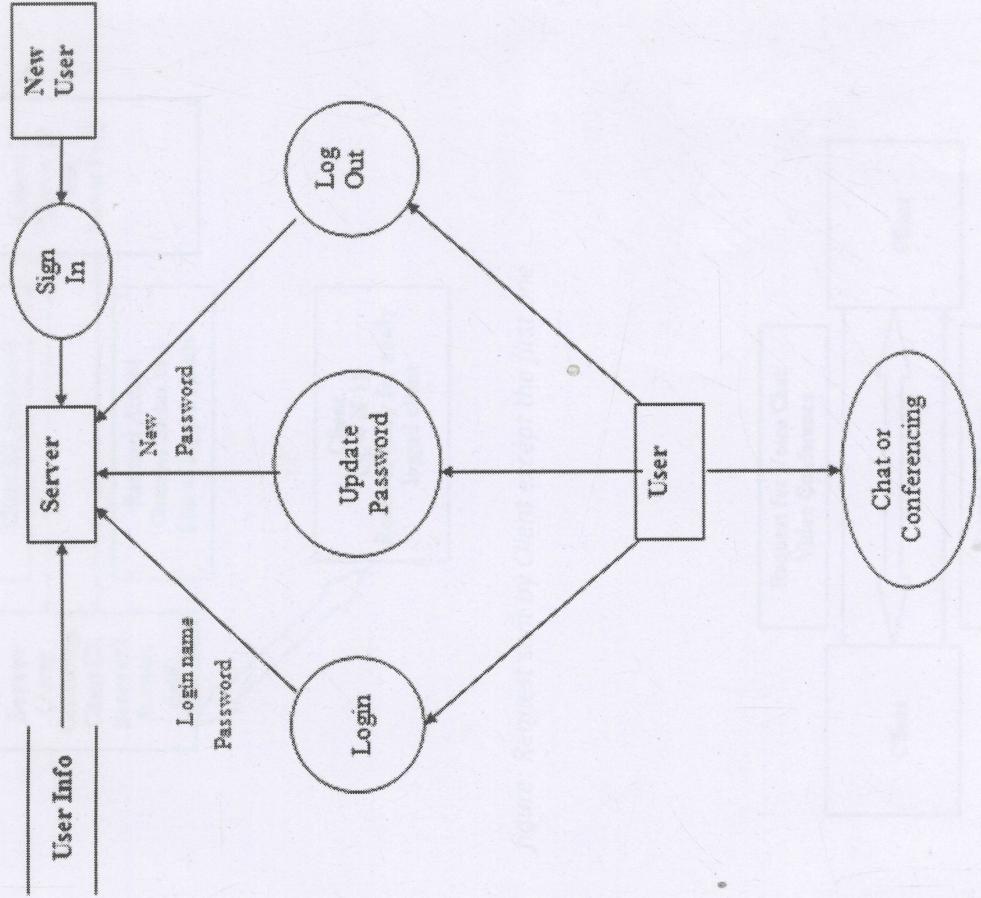


Figure: dataflow diagram

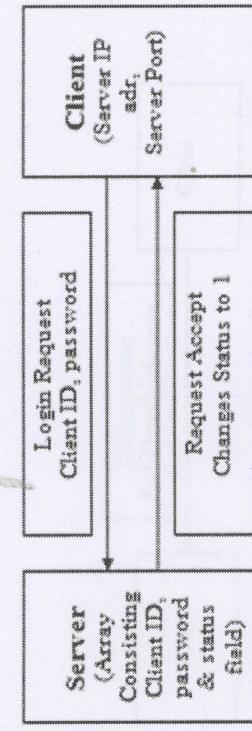


figure: Login Request sent by first Client

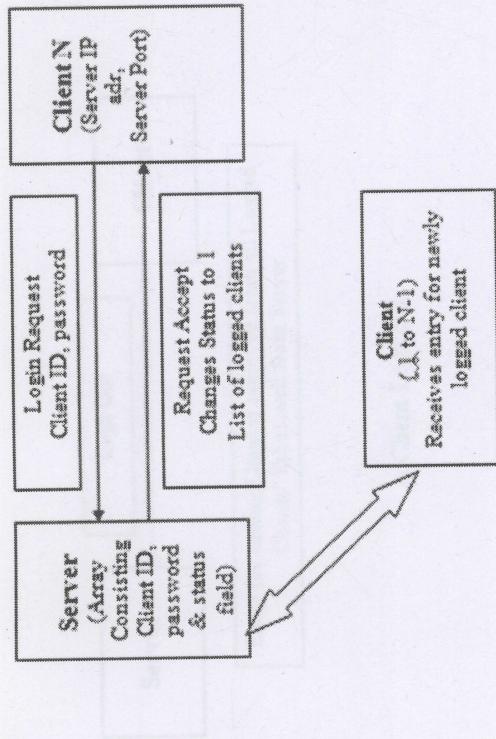
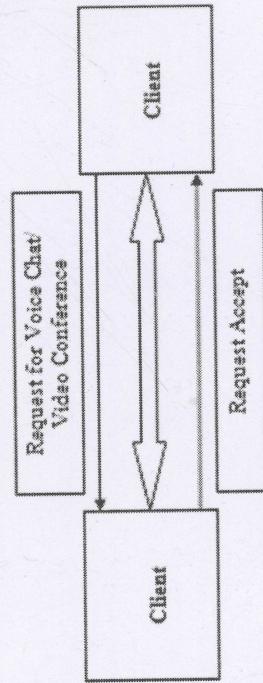
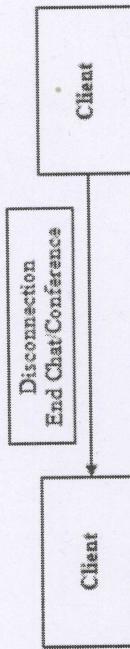


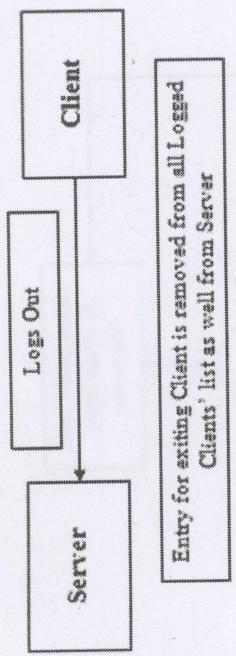
figure: Request sent by Client except the first one



Voice Chat / Video Conference Request & Acceptance



A Client ends Chat/Conference



A Client Logs out

Figure: Creating a Client

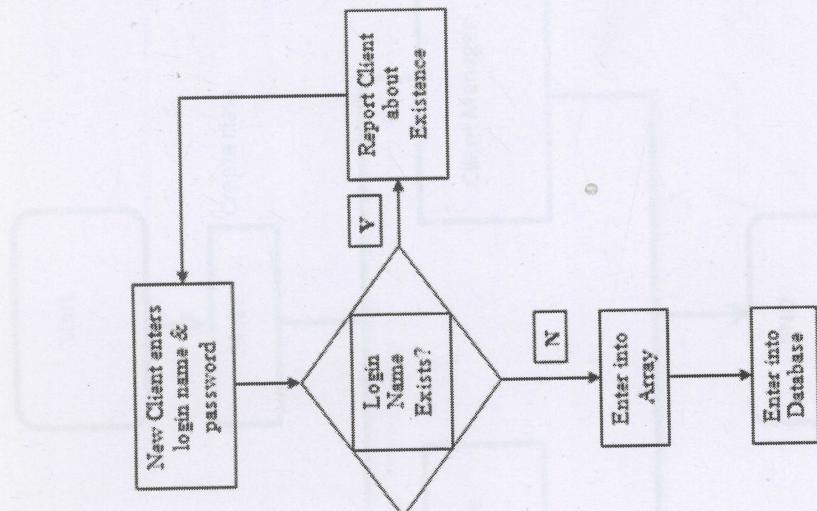


Figure: create new user

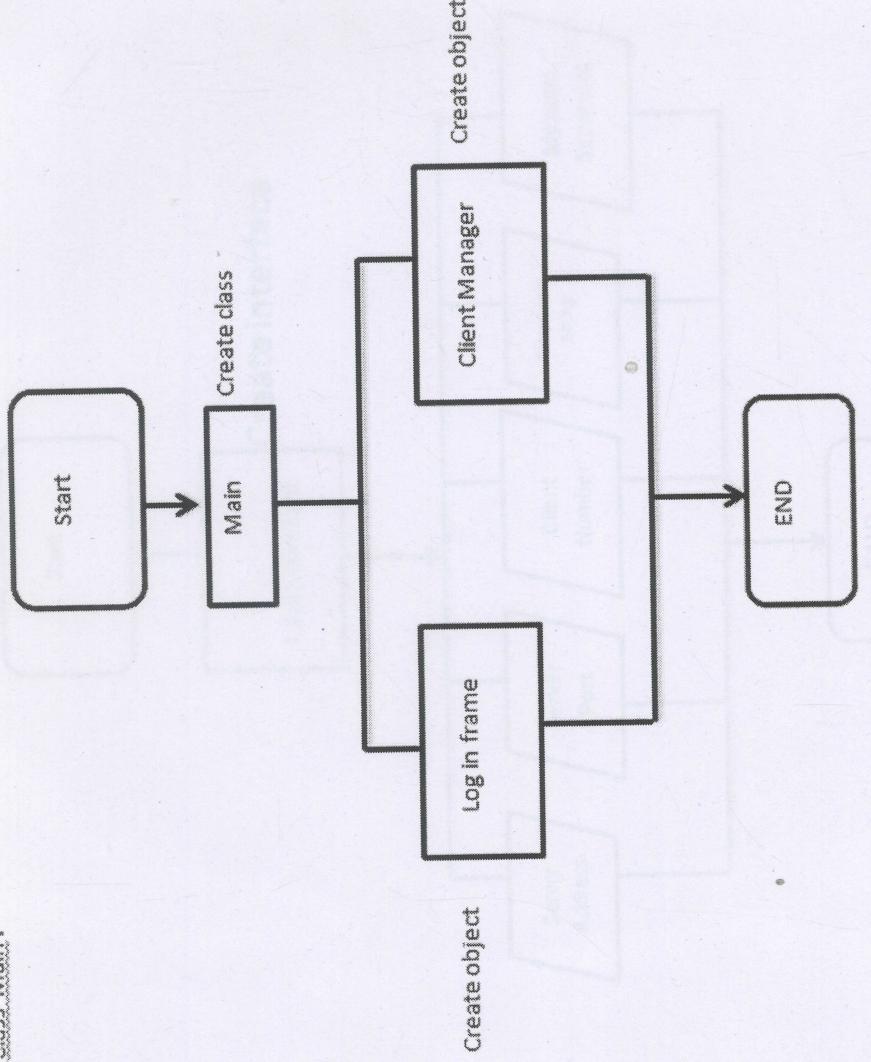


Figure: client connection

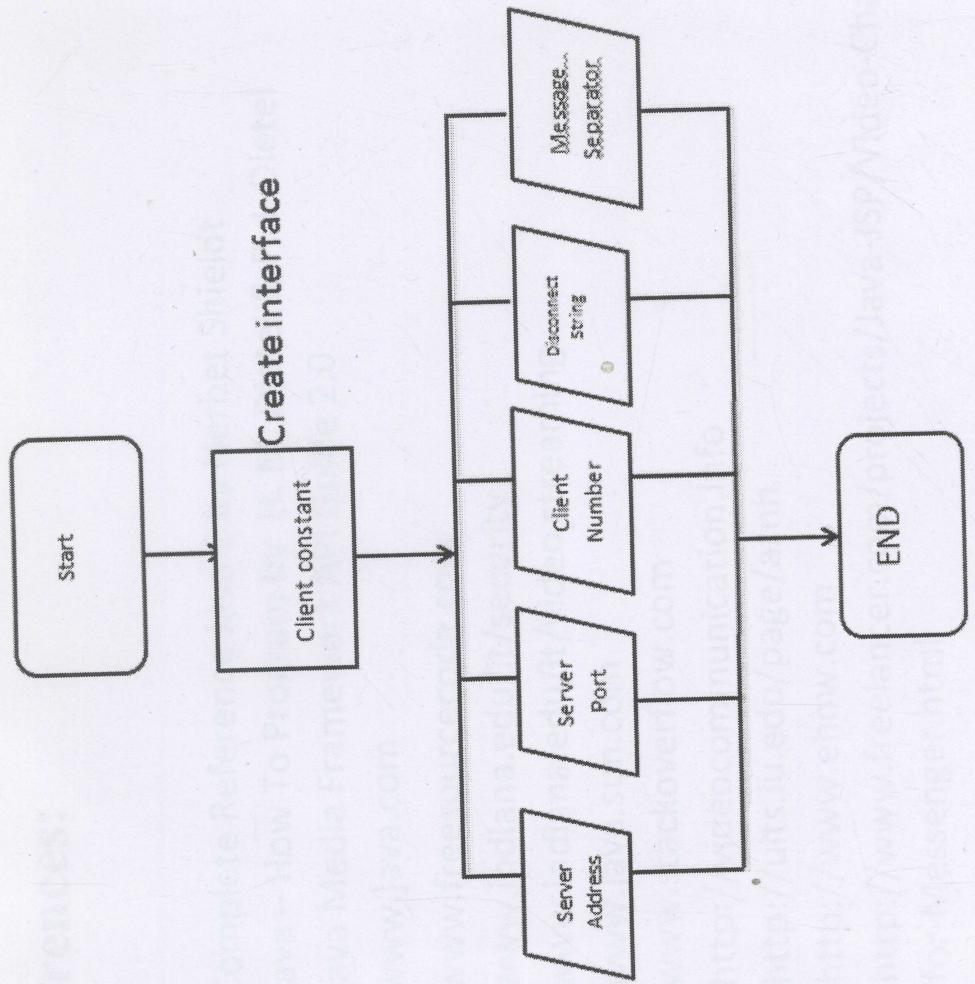


Figure: server connection