



**King Saud University**  
**College of Computer and Information Sciences**  
**Department of Information Technology**

# **IT 328 Project**

**Socket Programming**

<b>Weighting of this coursework towards the Module mark (%):</b>	10%
--	-----

<b>Outline of Problem</b>	The common language of the Internet is IP. Programmers who create networked applications typically interface with IP via either the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP). Sockets offers an Application Programming Interface to both UDP and TCP. In this assignment you are to investigate the Sockets interface and produce a small networked application as outlined below.
<b>Problem</b>	<p>Instant messaging (IM) is a staple of the web and has been around almost since its inception, starting with simple text-based programs like talk and IRC and progressing to today's GUI-based IM clients from Yahoo, Microsoft, AOL, etc. In this project you will design and implement an IM system, including both the client and the server. The following characteristics constrain the design space of an IM system:</p> <ul style="list-style-type: none"> <li>• <b>Real-time communication.</b> An IM conversation happens in real time: one person types some text, presses "enter," and the other person (almost) immediately sees the text.</li> <li>• <b>Number of parties.</b> An IM conversation can happen between two or more people. Some systems only allow two people to communicate; others allow more than two people. <b>Most systems allow a person to be involved in multiple conversations at the same time.</b></li> <li>• <b>Based on typed text.</b> The main mode of communication is via text, as opposed to voice or video.</li> <li>• <b>Connected over a network.</b> The parties involved in the communication may be in physically remote locations, and are connected over the internet.</li> </ul> <p>Your task will be to design an instant messaging system <b><u>with the above properties</u></b>, as well as additional properties that you will incorporate into your design. This system will include a server component that handles the transfer of messages and other data, and a client component with a graphical user interface.</p>
<b>Purpose</b>	<p>The purpose of this project is twofold.</p> <ul style="list-style-type: none"> <li>• <b>First</b>, you will learn several Java technologies, including networking (to support connectivity over a network), sockets and I/O (to support real-time, text-based communication), and threads (to support two or more people communicating concurrently). State machines may be useful to specify certain aspects of the system's behavior.</li> <li>• <b>Second</b>, this project will introduce you to the state-of-the-art for enabling human-computer interaction: graphical user interfaces. You will: <ul style="list-style-type: none"> <li>◦ become familiar with Swing, a graphical user interface (GUI) toolkit for Java, that is similar to many other such toolkits;</li> <li>◦ learn important GUI programming concepts, including the notion of a <i>view hierarchy</i> and the <i>model-view-controller</i> design pattern;</li> <li>◦ use event-based programming and the publish-subscribe pattern;</li> <li>◦ use object modeling notation to explore and express these structures;</li> <li>◦ And confront user interface design challenges.</li> </ul> </li> </ul> <p>Throughout the project, you will need to design and implement mutable datatypes, paying particular attention to their specifications and how they interact with one another.</p>
<b>Specification</b>	<p>Implement an IM system in Java with the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Client.</b> The client is a program that opens a network connection with the IM server at a specified IP address and port number. Once the connection is open, the client program presents a graphical user interface for performing the <b><u>interactions listed below</u></b>.</li> <li>• <b>Server.</b> The server is a program that accepts connections from clients. A server should be able to maintain an unlimited number of open client connections, and clients should be able to connect and disconnect as they please.</li> </ul>

	<p>The server is responsible for managing the state of both clients and <i>conversations</i>.</p> <ul style="list-style-type: none"> <li>• <b>Conversations.</b> A conversation is an interactive text-exchange session between some number of clients and is the ultimate purpose of the IM system. The exact nature of a conversation is not specified, except to say that it allows clients to send text messages to each other. Messaging in a conversation should be instantaneous, in the sense that incoming messages should be displayed immediately, not held until the recipient requests them.</li> <li>• <b>Client/server interaction.</b> A client and server interact by exchanging messages in a protocol of your devising — the protocol is not specified. Using this protocol, <u>the user interface presented by the client should:</u> <ul style="list-style-type: none"> <li>○ Provide a facility for seeing which users are currently logged in;</li> <li>○ Provide a facility for creating conversations between all online users.</li> <li>○ Provide a facility for private conversation between 2 clients.</li> </ul> </li> <li>• <b>No authentication.</b> In a production system, logging in as a client would require some form of password authentication. For simplicity, this IM system will not use authentication, meaning that anyone can log in as a client and claim any username (an arbitrary string of maximum size of 10 characters) they choose.</li> </ul>
<b>Phases</b>	<p>There are two phases for this project:</p> <ul style="list-style-type: none"> <li>• For the <b>First phase (Week 9)</b>, you will be evaluated during your lab session on: <ul style="list-style-type: none"> <li>○ Interface code and run</li> <li>○ Connection code with clear comments and run</li> </ul> </li> <li>• For the <b>Second</b> and final Phase (<b>Week 11</b>): <ul style="list-style-type: none"> <li>○ your deliverable is a poster: <ul style="list-style-type: none"> <li>- to present your application features, its architecture and protocols (you might use the attached template).</li> <li>- will be evaluated on the following elements: design, creativity, content and communication skills</li> <li>- It should be of size A1</li> </ul> </li> <li>○ Also, you will meet with your lab instructor during the same week (time slots will be specified later) for the evaluation on the whole project.</li> </ul> </li> </ul>
<b>Resources Required</b>	JAVA compiler
<b>Hints</b>	<ul style="list-style-type: none"> <li>• This is a group coursework, to be carried out by groups of 3 students.</li> <li>• You must be able to run your IM by having one computer per each group member (for example: if you have five group members→ you should have at least five different PCs, a server and four or more clients).</li> <li>• Each group member must understand every detail in the project and able to answer any question with regard to the evaluation.</li> <li>• Cheating or copying others' solutions is not tolerated. If two solutions are similar, or any other form of cheating has been used, zero grades will be given to both teams.</li> </ul>
<b>Grading</b>	<p>The grading breakdown is as follows:</p> <ul style="list-style-type: none"> <li>○ <b>phase 1: (3 Marks)</b></li> <li>○ <b>phase 2 – Code Implementation and evaluation (7 marks)</b></li> </ul>