

# Mobile Applications

## COMP 2601

### **Networking: Messaging**

## From last Class

- Learned to use sockets
  - Socket address needed to be 174.X.X.X
- Android clients need permission to use sockets
  - Had to modify AndroidManifest.xml
  - Cannot create socket on main activity thread
- Interacted using primitive data types
- No semantics associated with data
- Run to completion on server

## In Class

- Clone projects from last class
  - Well, you want to complete it to show TA 😊
- Create Message class:
  - Has a Header and a Body
  - Is serializable

## In Class

- Header has:
  - String id // Identity of sender; e.g., Bob
  - Long seqNo // Sequence number
  - String retId // Return id (same as id for now)
  - String type // Type of message (see later)
- Body has:
  - HashMap // Contains key/value pairs

## Messages

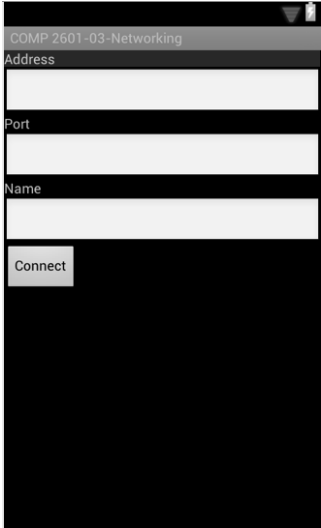
- Create messages for:
  - Connection
    - Effectively a login to the server
  - List Files
    - This will ask for a list of files from the server

## In Class

Activity has: Address field used for IP address for connection. Port field for connection, Name and Connect button.

Type valid IP address into Address Field. Use port between 1024 and 64K for Port. Name, say Bob

When **Connect** button pressed, connect to Server, generate a *new Activity* that contains a *ListView*

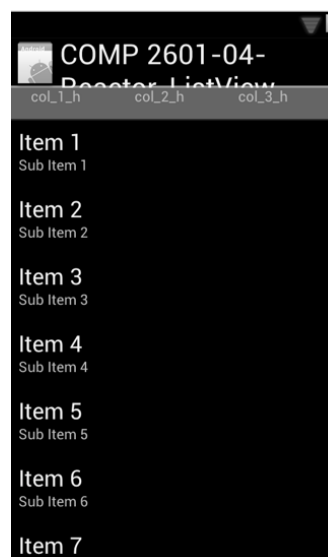
A screenshot of an Android application interface. At the top, there is a title bar that says "COMP 2601-03-Networking". Below the title bar, there are three text input fields. The first field is labeled "Address", the second is labeled "Port", and the third is labeled "Name". Below these three fields, there is a button labeled "Connect". The background of the application is dark.

## In Class

- When connected, button changes to “Connected”.
- The client then sends a message object to the server indicating that they are “logging in”.
- Server responds with “login response”
- Client then starts a new activity
- New activity sends “List files” message
- ...

## In Class

- List View →
- Should have 2 columns, one containing name the other can be empty (for now)



## In Class

- Server responds with a ListMessage that contains a list of fictitious files. Read them from a directory if you wish; hard coding them is okay.
- The ListMessage will have its HashMap with keys “file.x” where x = 1, 2, ... and the value will be the file name.
- Client updates ListView with files sent.

## In Class

- Server looks like slide 17 in *today's* notes.
- Server now processes *messages*, not primitive types.

## You must demonstrate:

- That the problems from the last class are completed.
- Messaging between client and server is implemented.
- A 2<sup>nd</sup> activity can be started on the client which contains the list of files generated by the server.

Will probably take more than this class

## Advanced: Not Required

- Modify your ListView to include a check box. The check box should be checked if the file type is .mp3 (or choose something to show that some check boxes will be checked and others not checked).