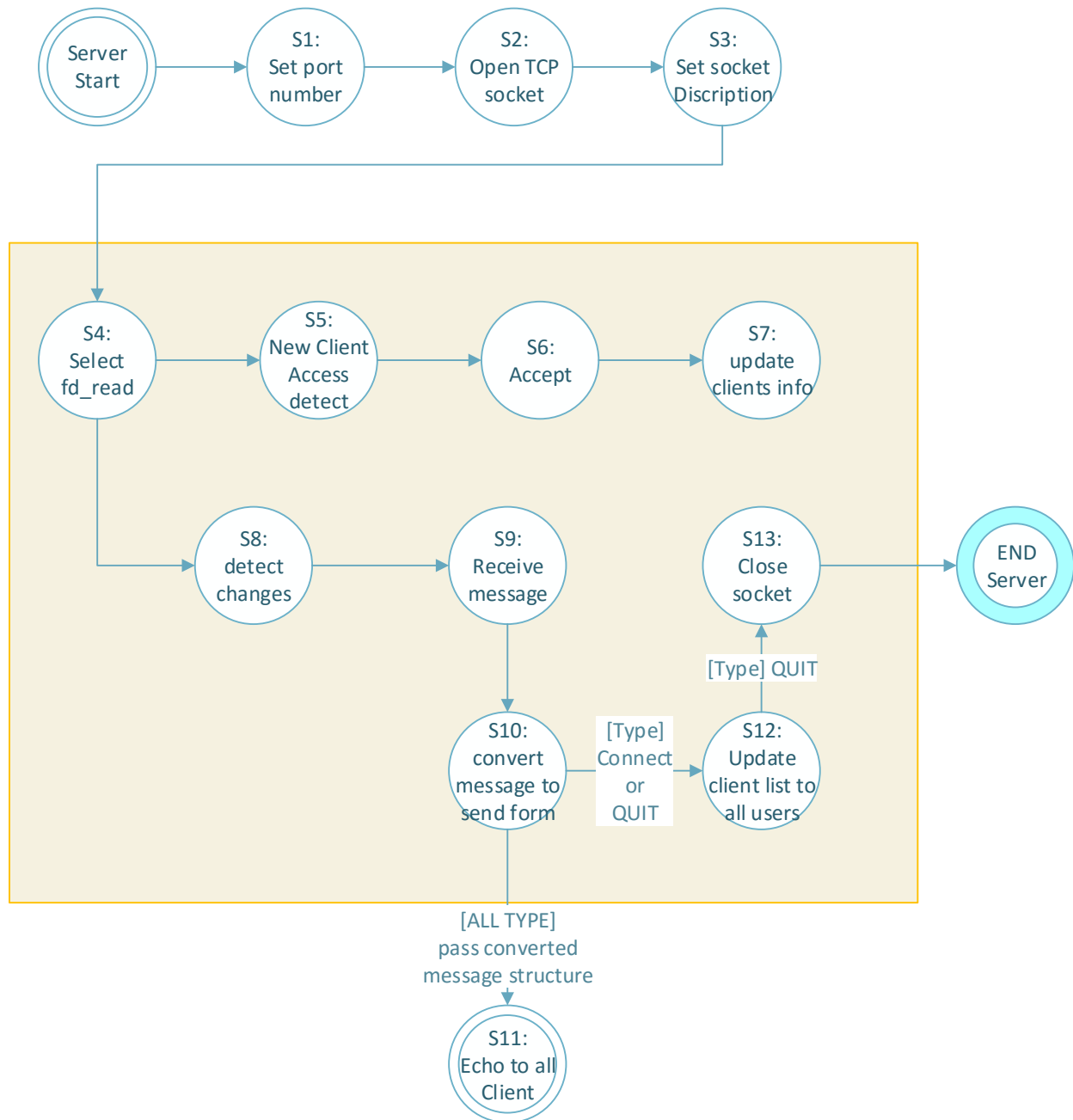


LINUX CHATTER BOX DESIGN

SERVER DIAGRAM



SERVER PSEUDO CODE

LINUX SERVER

FUNCTION MAIN

- Get port number
- Open file to write log
- Open socket
- Initizlie client list to save and FD_SET arrays for multiplexing
- Start multiplexing
 - Reset FD_SET
 - Accept connection and update client list and update FD_SET
 - If no more space for new user, send error message
 - Check user, and if multiplex signal is detected, receive message
 - Check message type and convert message and make sender thread
 - Save log using received message
 - If type is quit, close socket

FUNCTION OPEN SOCKET

- Create socket and initialize socket option to possible to be reused immediately
- Initialize address information
- Bind socket
- Listen socket

FUNCTION RECEIVE MESSAGE

- Receive message structure
- Check type of message structure
 - If connect type, reset user name and send user list
 - Generate send message depending on the type.

FUNCTION INITCLIENTINFO

Initialize a client socket number and name on the clientList array

FUNCTION CREATEMESSAGE

Add user name from client list and fill message structure using information parameters

FUNCTION SENDLIST

Send a list of client users

FUNCTION WRITE MESSAGE

Check whole client list from first to last index

 If client name is same with message client name, not send

 If it is not empty array of clientList, send message to the client

 If it is connect or quit message, send client list to update

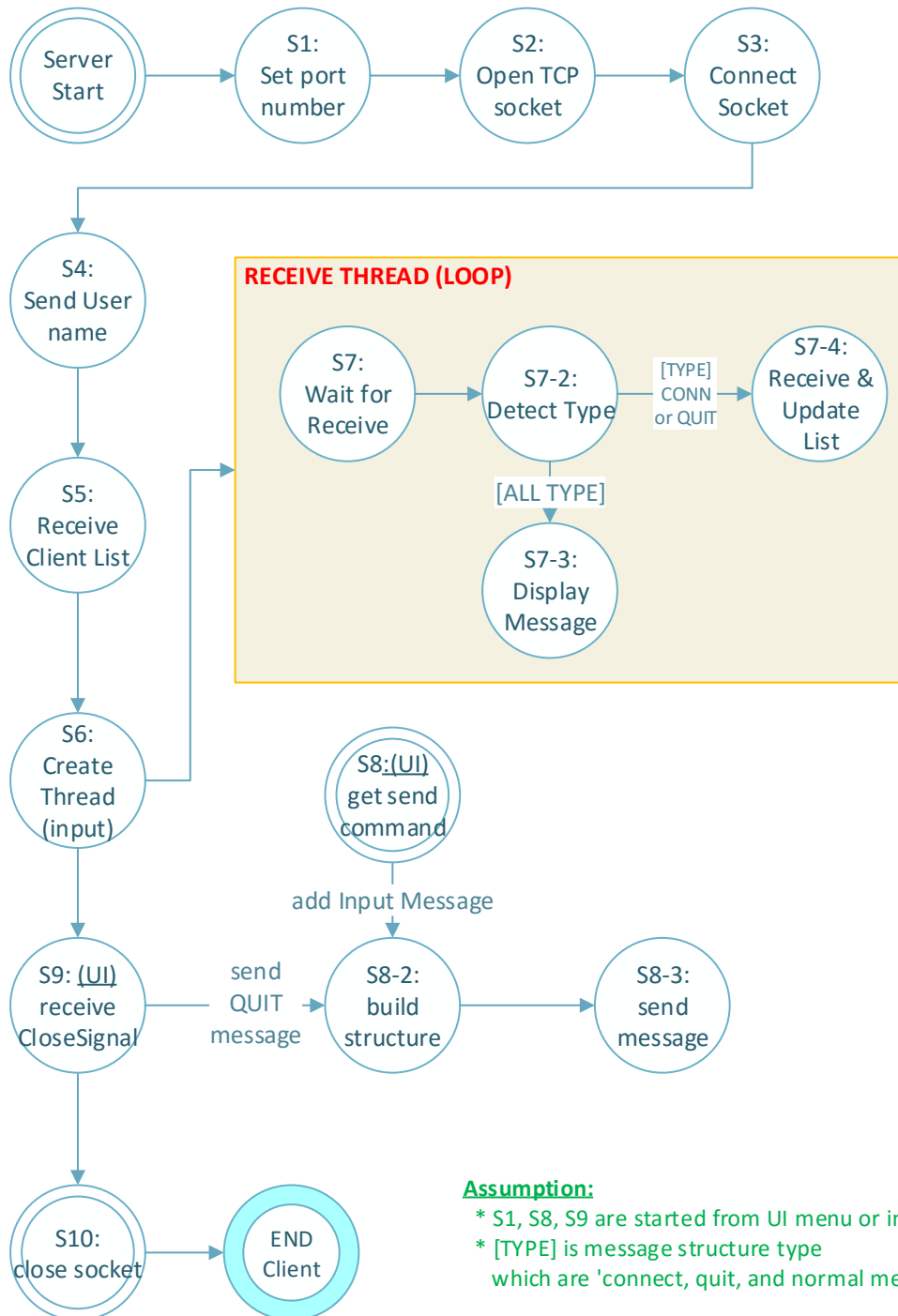
FUNCTION WRITEONE

 send message to only 1 client

FUNCTION DISPLAY LIST

 if the client list is not empty, display name and socket number

CLIENT DIAGRAM



CLIENT PSEUDO CODE

CLIENTSRC CLASS FUNCTION

CONSTRUCTOR

Initialize parent object for UI

FUNCTION CLIENTSTART

Create socket

Initialize socket address

Connect to the server

Send personal information

Receive user list

Create thread to receive message asynchronously

FUNCTION SENDPERSONALINFO

Read nickname of client user and make structure which is CONN type

FUNCTION CREATEMESSAGE

Fill message structure using input information

Write message

 write message structure with type info

 if type is quite, close socket

NON CLASS FUNCTIONS

FUNCTION RECVLIST

Receive message as a client structure type

Display the list of client users.

FUNCTION READMSG

Keep read message

- Check message type

- If message type is 'connect' or 'quit' receive client list

- Else display message with user name