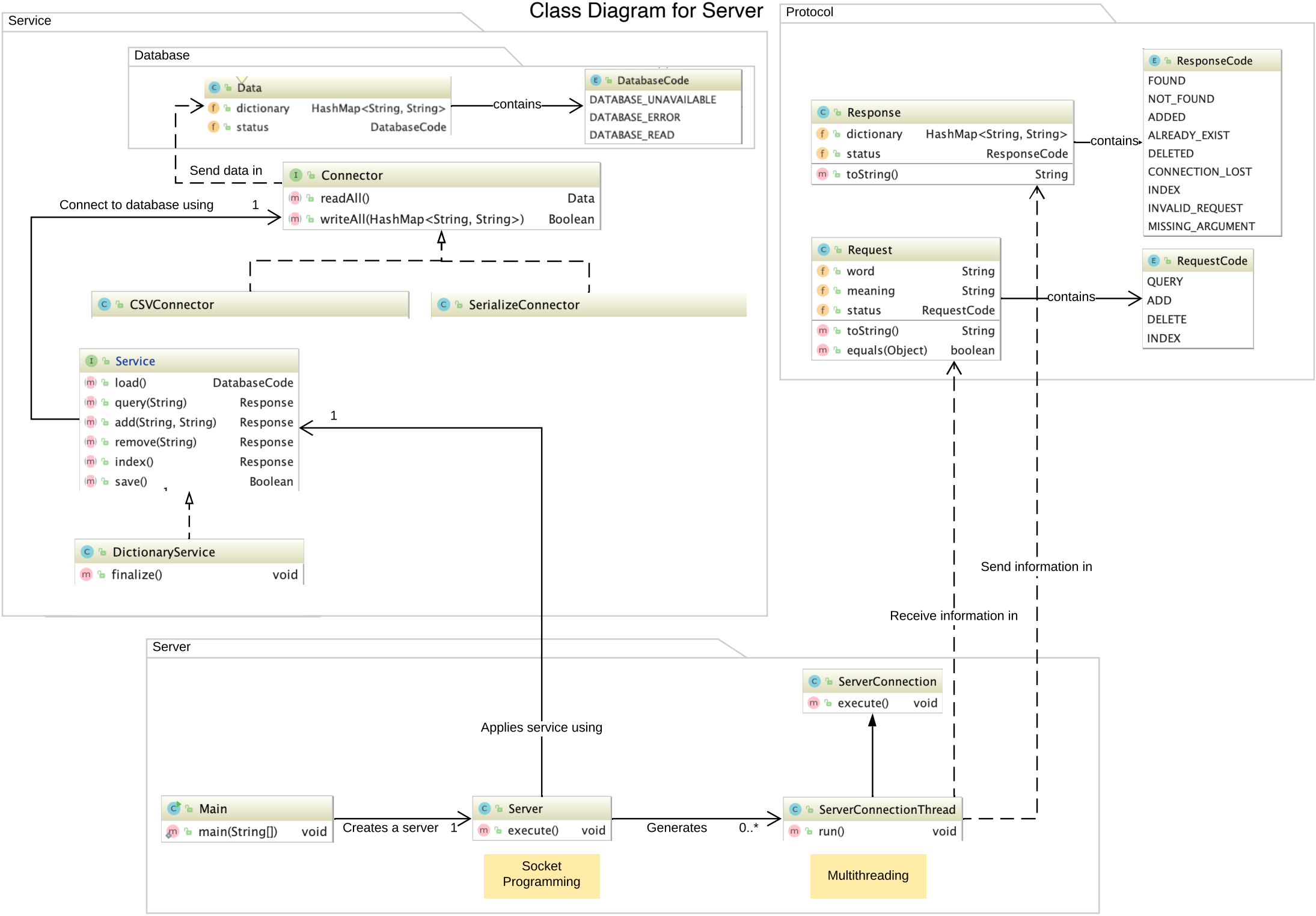
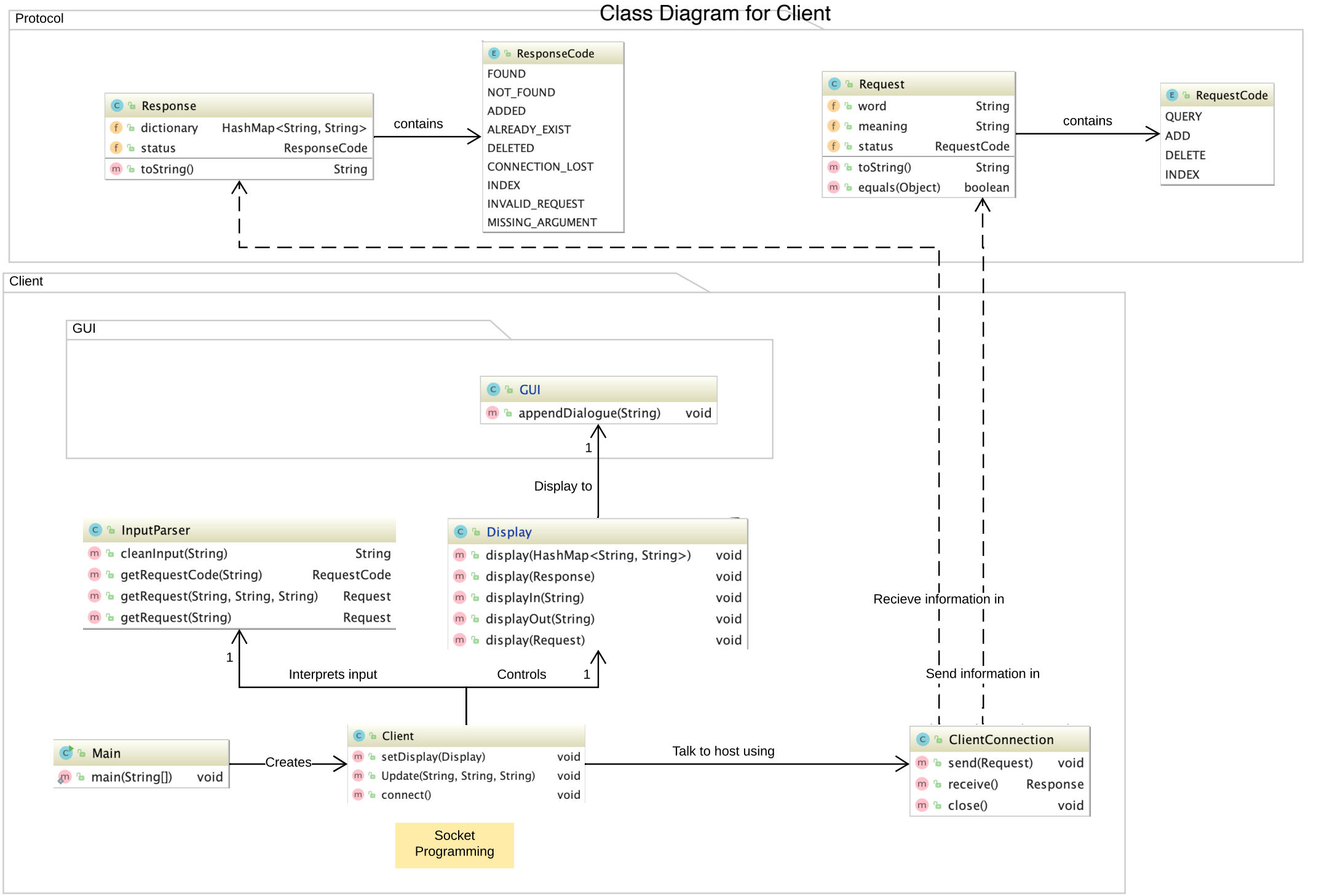
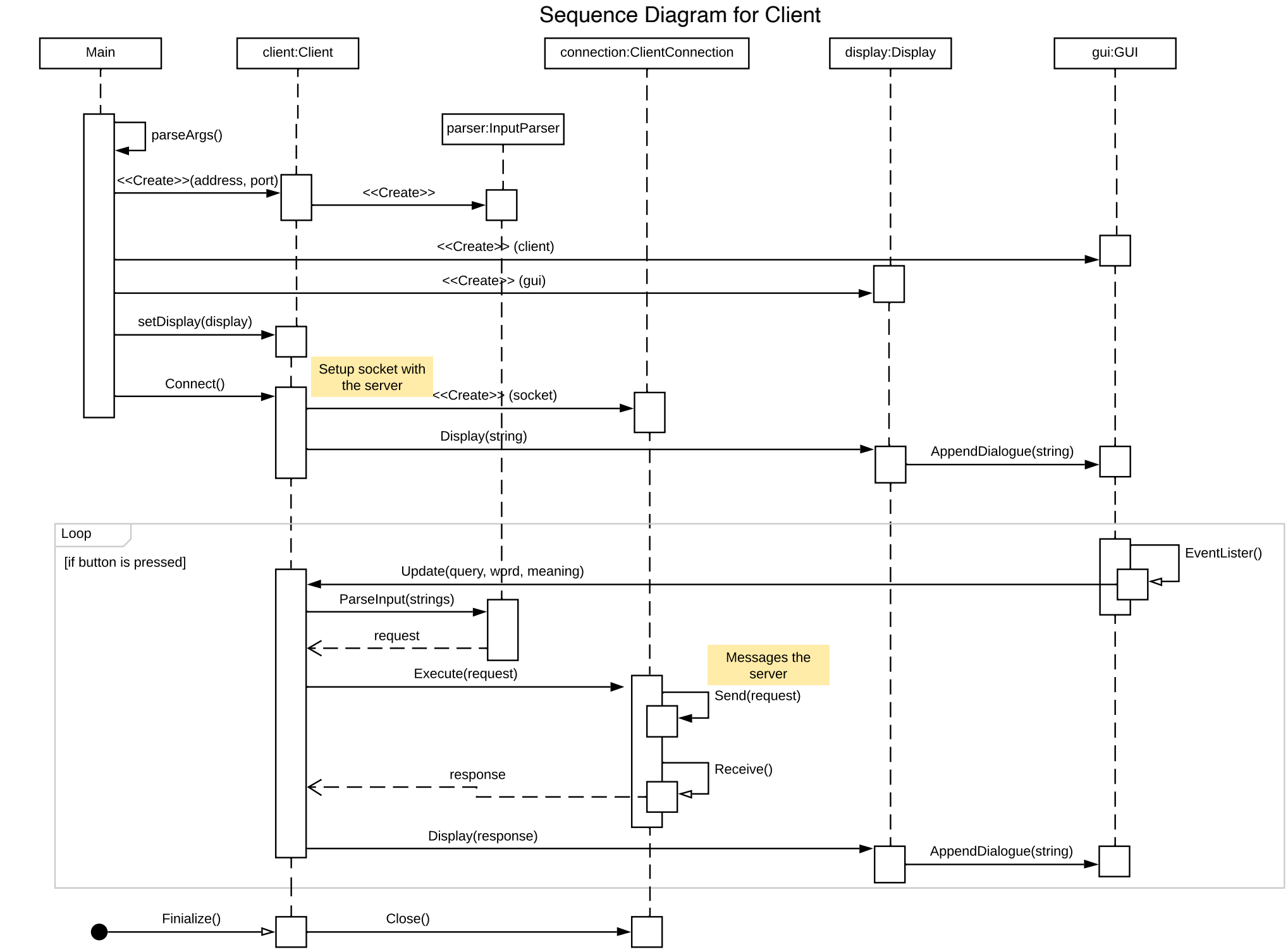
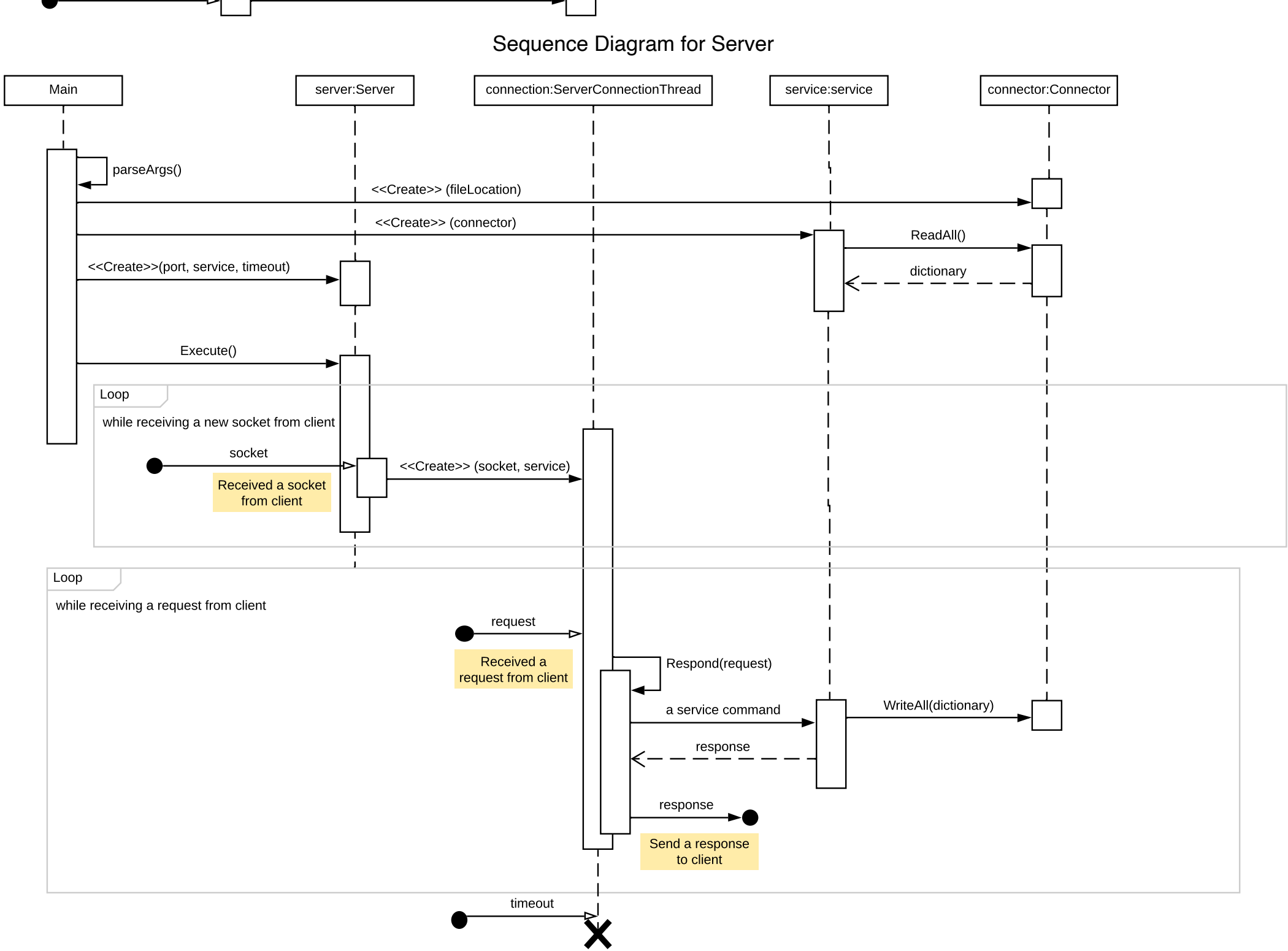
Class Diagram for Server




Class Diagram for Client
Protocol
Response
dictionary HashMap<String, String>
status ResponseCode
toString() String
contains
ResponseCode
FOUND
NOT_FOUND
ADDED
ALREADY_EXIST
DELETED
CONNECTION_LOST
INDEX
INVALID_REQUEST
MISSING_ARGUMENT
Request
word String
meaning String
status RequestCode
toString() String
equals(Object) boolean
contains
RequestCode
QUERY
ADD
DELETE
INDEX
Client
GUI
GUI
appendDialogue(String) void
1
Display to
InputParser
cleanInput(String) String
getRequestCode(String) RequestCode
getRequest(String, String, String) Request
getRequest(String) Request
Display
display(HashMap<String, String>) void
display(Response) void
displayIn(String) void
displayOut(String) void
display(Request) void
Recieve information in
1
Interprets input
Controls
1
Send information in
Main
main(String[]) void
Creates
Client
setDisplay(Display) void
Update(String, String, String) void
connect() void
Talk to host using
ClientConnection
send(Request) void
receive() Response
close() void
Socket Programming


Sequence Diagram for Client
Main
client:Client
connection:ClientConnection
display:Display
gui:GUI
parser:InputParser
parseArgs()
<<Create>>(address, port)
<<Create>>
<<Create>> (client)
<<Create>> (gui)
setDisplay(display)
Setup socket with the server
Connect()
<<Create>> (socket)
Display(string)
AppendDialogue(string)
Loop
[if button is pressed]
EventLister()
Update(query, wprd, meaning)
ParseInput(strings)
request
Messages the server
Execute(request)
Send(request)
Receive()
response
Display(response)
AppendDialogue(string)
Finialize()
Close()


Sequence Diagram for Server
Main
server:Server
connection:ServerConnectionThread
service:service
connector:Connector
parseArgs()
<<Create>> (fileLocation)
<<Create>> (connector)
ReadAll()
dictionary
<<Create>>(port, service, timeout)
Execute()
Loop
while receiving a new socket from client
socket
Received a socket from client
<<Create>> (socket, service)
Loop
while receiving a request from client
request
Received a request from client
Respond(request)
a service command
WriteAll(dictionary)
response
response
Send a response to client
timeout