Q1) Design a network application protocol for this application and justify your design decision. In your protocol design, you should address the following. [35 points]

1. The communication patterns of the network application.

This application makes use of a TCP Socket to allow for bi-directional communication between the Server and the Clients. How the communication of this application works follows this pattern: Server is constantly running looking for a connection to be made from a Client, Client (Job Seeker or Job Creator) connects to Server depending on the Client type will cause certain functions to run. Connection with the Server will remain in tack until the Client has fulfilled all of it's duties whether it be posting a job then accepting Job Seekers or accepting a job and submitting the files the Job Creator was looking for. Ideally this communication pattern would run with no error and can be repeated an infinite amount of times.

2. Define the protocol design goals.

Due to timeboxing issues the protocol design goals need to be broken into different phases of development.

Phase 1 (Completed):

Let n denote set of positive real numbers

- Creating a Server which can handle n Clients at a time
- Server can send and receive messages from Clients
- Server Thread was created to cause a Blocking Socket (Reason Disclosed in further Phases)

Phase 2 (Complete):

- Two Clients created to distinguish their roles (EX: Job Seeker or Job Creator)
- Clients which are Job Creators can send a job to Server which will save this job to a list
- Clients which are Job Seekers can accept posted jobs (One job at a time)
- Clients which are Job Creators can start the job once all Job Seeker positions are filled
- Clients which are Job Seekers can submit their file to fulfill the job

Phase 3 (Incomplete):

- Account Creation and Login function will be created to help with security issues
  (Make use of Salted hashes)

Phase 4 (Incomplete):

- GUI development will start to allow for an easy to use interface
  - Login Screen
  - Job Posting Screen
  - File Submission Screen

3. Define the message format, structure, and semantics

*Client **(Job Creator)** Connects to Server*

Server - - - - - >Client

- Connection Message

Client - - - - - >Server

- Sends Job which they want listed (Job Name and # of Job Seekers needed)

Server - - - - - >Client

- Sends Job posting confirmation

Client - - - - - >Server

- Sends message telling Server it is waiting for Job Seeker

Server connects with Job Seeker

Server - - - - - > Client

- Sends confirmation message that job has been posted

Job Seeker/s Accept Job

- Sends message that Job Seeker spots are filled

Client - - - - - > Server

- Sends message to start job and is waiting for job completion

Job Seeker/s Finishes Job

Server - - - - - > Client

- Sends job completion message
- Sends needed file to Job Creator Client

Client - - - - - > Server

- Sends message accepting the completion message
- Receives and downloads Job file

Server Removes Job from List and Terminates Connection

*Client **(Job Seeker)** Connects to Server*

Server - - - - - >Client

- Sends connection message
- Sends Job List

Client - - - - - >Server

- Sends message pertaining to Job which they want to accept

Job Creator Accepts Job Seeker

Server - - - - - >Client

- Sends acceptance message

Client - - - - - >Server

- Sends necessary files to complete the job

Server saves the file/s

Server - - - - - > Client

- Sends confirmation of file being accepted

Server Terminates Connection

4. Design the communication rules.

-

-

-

Q2) Argue the need for a new application layer protocol for this network application instead of using existing standard protocols (e.g. HTTP, SMTP, WebSocket, etc.) [10 points]

When talking about an application layer protocol there are a couple base functions that need to be satisfied. If we were to argue about needing a new application layer protocol for this network application the following must be kept in mind when talking about application layers; facilitates the user to use the services of the network, it can be used to develop network-based applications, provides services like; user login, naming network devices, formatting messages and transfer of files, as well error handling and recovery of messages/files.

Breaking down what this network applications actions are, they can be categorized into two majors' components; File transferring/updating and communication between Server and multiple Clients. If there were a way to create a new application layer protocol the most useful function would be creating a global file store which is bound to the Server which can be accessed by every client, this protocol will be accompanied by Client priority numbers (Ex: 0 == Client can View Files, 1 == Client can Download Files, 2 == Clients can add Files, 3 == Client can Edit and Save Changes to Files, Each increase in priority number is given the functions of all previous priority numbers). Having this new application layer protocol would be perfect when considering what must be done for this network application, Having the Job Creators being of Priority 3 and the Job Seekers stating at Priority 0 once the Job Seeker accepts a job they will increase to Priority 1, after completion of this job they will move to Priority 2 then when the connection is terminated the Job Seeker will be set back to Priority 0 and the process can be repeated.

Q3) Provide the implication source code of your network application protocol with sufficient test cases based on the design goals, message philosophy (format, structure, semantics), and Communication rule.[25 points]