Problem 1

Overview

Networking in Java

As a Library, Network Programming was introduced in Java to enable communication between any two components connected on the same network. The *java.net* package interfaces low level communication details thus empowering a developed to focus on solving bigger problems at hand.

The *java.net* package supports 2 protocols

1. **TCP** (Transmission Control Protocol) - A connection-oriented protocol that is used to transmit packets of data across the network reliably.
2. **UDP** (User Datagram Protocol) - A connectionless protocol used to transmit data across the network in an unreliable fashion.

Socket Programming

Sockets are used to communicate over the network and interface over the TCP Layer. Communication is established after the ***accept()*** method is referenced. After the connection is established, I/O streams are used to send and receive messages.

TCP is a two way communication protocol that can be used to stream data from both sides at the same time.

Data Structures Used

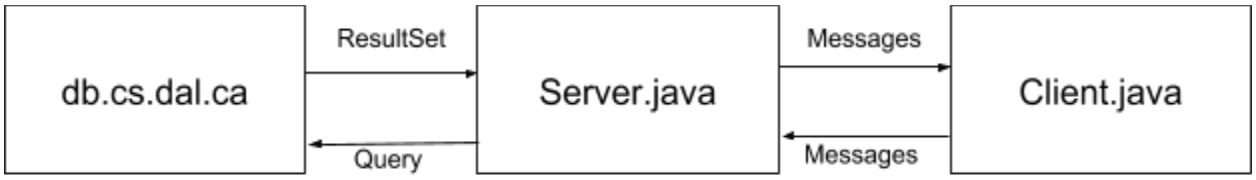
1. InputFrame

Datatype	Field Name
String	operation
String	target
String	protocol
Map<String, String>	headers

2. OutputFrame

Datatype	Field Name
String	operation
String	statusCode
String	status
Map<String, String>	headers

3-tier Architecture for Network Socket Connection



- The 3 tier Architecture includes a Server Layer that communicates with the cs.dal database when the Client requests for data or access.
- Business Logic for converting user/employee queries into database commands exists in the server.
- On response from the database, messages are relayed back to the Client.

Request Format

```
operation<sp>target<sp>version CRLF
headerKey1:headerValue1
...
...
<body>
```

Response Format

```
version<sp>responseCode<sp>response CRLF
headerKey1:headerValue1
...
...
<body>
```

Functions or Commands

1. Auth

Used for authenticating an employee to gain access to database. A correct authentication returns a 200 whereas an incorrect authentication returns a 401.

On correct authentication, a cookie value is returned which can be retained by the client and sent for any successive calls

2. Logout

Used for logging out a user from current session.

a. Cookie Auth Value checked to confirm if it's an authorized logout operation
b. Stored cookie value is deleted
c. Response returned

3. List

Product, Customers and Order Items can be listed out using this command.

1. List of products present in the database
2. List of customers present in the database
3. List of orders placed by current customer

4. New

This command text is primarily used for creating a new Order for a specific customer by adding products to the order

5. Add

This command text is used for adding items to the current order. The quantity of these items can also be included as part of the request frame

6. Order

This command text is used for placing the order and stopping any further items from being added

7. Drop

This command is used for dropping the current order and reverting back all the necessary flags to a default state.

Assumptions

1. One strong assumption towards this implementation is that Commands can be sent atomically, i.e, for one execution of the Server program, only one command (AUTH, LOGOUT, etc) can be executed.

   The reason for this is the lack of an end-frame signal. **End_frame** needs to be appended after every request command to denote the end of a frame and expect the start of a new frame.

Bibliography and References

1. https://dzone.com/articles/generate-random-alpha-numeric - To generate a random cookie
2. https://stackoverflow.com/questions/4234985/how-to-for-each-the-hashmap