The program flow is:

|  |  |
| --- | --- |
| **Program flow** | |
| **Main command flow:** | **Notes:** |
| NGTN |  |
| NGTN <METHOD> | You can call this at any time outside of data to renegotiate the method and keys via Diffie-Helman. |
| LOGN USERNAME PASSWORD DOB | Your credentials are shown below this table.  Passwords are hashed + salted. |
| HELO <DOMAIN> |  |
| MAIL FROM: <DOMAIN> |  |
| RCPT TO: <DOMAIN> | You can enter multiple one after another to keep adding more recipients. |
| DATA | End with <CRLF>.<CRLF>, you can enter more RCPT TO’s here if you wish as well. |
| **Other commands:** | |
| RSET | RSETs the user to the base step as if just connected, requiring renegotiation and logging in again. |
| NOOP | Resets timeout. |
| VRFY <name> |  |
| EXPN <group name> |  |
| TURN | As it’s optional this command returns a not available message as per the RFC821 spec. |
| HASH <line number in username and password file> | This is partially functional. It reads a salt and password from a given line in the ‘UandP.txt’ (username and passwords) file.  It then produces an MD5 hash of the password and salt then replaces the password + salt with the hash.  The issue the method has is that it reads the \n at the end of the file and includes that within the hash causing the hash to differ from that of the user login leading to user being unable to login. If this issue was resolved it would lead to being able to easily create users.  I did consider adding admin functions such as create user but decided it was a bad idea to leave an internet facing SMTP server with the ability to login and create accounts freely. I felt that the ability to hash details would be a good middle ground. |
| EHLO | Not supported, will return not supported message in response. |
| DLTM | Deletes currently logged in user’s entire mailbox. |
| DLET | Intended to delete currently logged in users mail filtered by subject. Currently non-functional, was mid implementation. You can find it at the bottom of the server lib with appropriate comments.  I have disabled it by setting the state requirement to something that cannot occur as noted in the comments |

Currently only a Caesar cipher is implemented as an encryption method as the spec rewards no bonuses for other methods. However, during development I have designed it with adding future methods in mind. It should take no longer than 30 minutes + time to find or create the code for the encryption method in question to add further methods.

(USERNAME| PASSWORD | DOB) – Enter values as shown below for LOGN:

10 1 1/1/1990

20 2 1/1/1990

30 3 1/1/1990

40 4 1/1/1990

50 5 1/1/1990

admin password 99/99/9999

My users do not have a username outside of their address currently. To reach a user please enter <username>@<domain>.

Almost all my values are defined in the initialiser or via file and loaded at runtime. Due to this you are able to redefine my domain by changing self.\_my\_domain in the \_my\_initialise method. By default, my domain is **@mymaildomain.com** (case sensitive per RFC821).

You can **enable** or **disable** **encryption** of **files** and **logs** via a **self.\_base\_key** you can find in the \_my\_initialise method. I have left a comment above it. By **default**, it is at zero meaning my logs and emails will **not** be encrypted. This is for your ease of marking (can’t mark logs if you can’t see them etc).

**Timeout** has been set to **4 minutes** to offer ample time to test and document without worry of timing out. You **can change this** in the init.

Shown below is two groups that are present along with their members.

|  |  |
| --- | --- |
| **Group: (**groups are case-insensitive per RFC spec) | **Members:** |
| staff | 10, 20, admin |
| students | 20, 30, 40, admin |

Logs are formatted as follows:

DATETIME | USER\_LOGGED\_IN | MESSAGE | HASH OF PREVIOUS MESSAGE

Note: As previously mentioned, by default log and email encryption will be set to 0 (disabled). You can quickly change this in \_my\_initialiser in the server lib.

Further note: To fulfil the requirement of implementing all error codes even if they cannot be reached, I decided it would be best to implement a code dictionary. I hope that is satisfactory.

Thanks!