# COMP3331 - Networks Assignment: Networking eDocuments (Report)

**Program Design**

**How System Works**

* Both reader and server have different directories, to support the ability for independent databases (which may have conflicting names) to be used
* Reader
  + Has a directory for each book, each containing the book's pages and contents
  + The first message (containing data) sent to the server will be the reader's username, mode (push/pull), . This message has a particular format, described in 'Message Design'.
* Server
  + *Server Database*  
    There are 3 files to contain the forum posts from the server side: Post info, Post Content, and a list of usernames of those who read the posts. Each of these files contains lines which each represent a portion of info stored about that particular forum post. The PostID (serial number) is used to relate these files together to form details for a particular forum post. These line formats are summarised below:

|  |  |
| --- | --- |
| File | Line format |
| Post Info | [PostID]=[SenderName],[BookName], [PageNum],[LineNum] |
| Post Content | [PostID]=[Content string] |
| Read Identities | [PostID]=[Name1],[Name2],[Name3]... |

By keeping these files, it allows the data to be preserved when the server is turned off (compared to when the database is stored in program memory). It also allows administrative tasks to be conducted (eg examining content).  
  
Thus the database initialisation is simply compiling a list of all serial numbers for forum posts that have been used, so the server knows what numbers can be generated for new forum posts.

**Message Design**

* Reader
  + Introduction message: used to provide the username and mode to the reader  
    '#Intro#Username=[UserName]#Opmode=[Opmode]'

Design Considerations and Tradeoffs

**Possible Improvements**

* Server Database
  + By convention of databases, 'BookName' can be replaced by an integer ID

**Issues**

* When using Ctrl+C to close the server, it needed to wait about 60 seconds before . To fix this, I set the options for the socket as 're-usable'. Not sure if there will be any future issues when a TCP tuple for a previous packet is recreated.

**Code References**

* Code used as a framework for creating sockets, and making connections between a server and client taken from "Socket Programming HOWTO" (https://docs.python.org/2/howto/sockets.html)
* Code used for handling multiple clients using 'Select' for networks was adapted from "Python Network Programming: Echo Server with Select"  
  (http://ilab.cs.byu.edu/python/select/echoserver.html)
* The template code for constructing and running threads was adapted from "Python Multithreaded Programming"  
  (http://www.tutorialspoint.com/python/python\_multithreading.htm)