

The implementation in the previous document has been improved. The new implementation works as follows:

- All functions now return a deque of messages that are in the form of

```
{  
    Connection : ID_OF_RECIPIENT  
    Text:      MESSAGE_TEXT  
}
```

- This is in contrast to the previous implementation, where functions only sent back a processed string that must be fed into postOffice at the end.
- Each command is now responsible for processing its own messages.
- Since each function is responsible for its own deque of processed messages, buildOutgoing and postOffice are no longer necessary.
- Each function uses either the helper function,
 - `std::deque<Message> processForRoom(Connection c, string text)`
 - Takes a string of text and creates processed outgoing messages for everyone in the same room as the user with the `userId c`
 - ex. `processForRoom(123, "Hello Room")` generates a deque with a "Hello Room" message to everyone in the same room as user 123

or

- `std::deque<Message> processForUser(Connection c, string text)`
 - Takes a string of text and creates a processed outgoing messages for only the user with the `userId c`
 - ex. `processForRoom(456, "Hi there")` generates a deque with one single message with text "Hi there", to user 456