Story 1 – Send a message in a simplified email format
When I call the command line tool to send a message I'd like the "network" to be written to.

For example, if I call:
```
message joe@example.com "Hi there!"
```

network receives the message:
```
connect smtp
To: joe@example.com

Hi there!

disconnect
```

With newlines escaped this would look like:
```
"connect smtp\nTo: joe@example.com\n\nHi there!\n\ndisconnect\n"
```

Story 2 – Show an error if the address is invalid
If the email address doesn't contain an @, I'd like the "console" to receive an error message

For example, if I call:
```
message joewithnoatsign "hi there!"
```

Network should not receive any messages & console receives the message:
```
Invalid email address: jowithnoatsign
```

Story 3 – Show an error if the body is invalid
If the body of the message is omitted, or just left blank, that should be an a error too.

For example, if I call:
```
message joe@example.com
```

console receives the message:
```
Cannot send an email with no body.
```

Story 4 – Send to multiple email addresses
The application needs to be smart enough to detect that last argument is the body and the first n arguments are email address

For example, if I call:
```
message sally@example.com,joe@example.com "Hi everyone!"
```

network receives the message:
```
connect smtp
To: sally@example.com
To: joe@illustration.com

Hi everyone!

disconnect
```

Story 5 – Show a slightly different error if more than one email address is inval (We want our users to be happy don't we?)

For example, if I call
```
message noatsign,againnoatsign "Hi everyone!"
```

console receives the message:
```
Invalid email addresses: noatsign, againnoatsign
```

Story 6 – Send messages in another format
Imagine the tool supported a simplified chat protocol.

For example, if I call:
```
message –im alice@chat.example.com "hey there"
```

imnetwork receives the message:
```
connect chat
<alice@chat.example.com>(hey there)
disconnect
```

Story 7 – Send multiple chat messages
Chat messages should be sent separately to each recipient.

For example, if I call:
```
message –im alice@chat.com,bob@chat.com "Hello."
```

imnetwork receives the message:
```
connect chat
<alice@chat.example.com>(Hello.)
<bob@chat.example.com>(Hello.)
disconnect
```

Story 8 – Handles errors gracefully
If sending fails (for some reason the "network" isn't available), show a friendly error message.

For example, when the network is down and I call:

```
message lisa@somewhere.com "hi Lisa"
```

console receives the message:
```
Connection error. Please try again.
```