Simple Group Chat (Server)

## Problem Description

In this problem, you are tasked to write a basic server code for a simple group chat application using sockets in Python. The server is capable of accepting multiple client connections, receiving messages from any connected client, and then broadcasting these messages to all other connected clients.

## Server Code Overview

The provided incomplete server code sets up a TCP socket that listens for incoming connections on a specified host and port. When a client connects, it expects to receive a message that identifies the user (a nickname, for instance). Once a user is connected, they can send messages to the server, which will then broadcast these messages to all other connected users.

Key functionalities include:

- Accepting new client connections
- Receiving a message from a client
- Broadcasting a message to all other clients (except the sender)
- Handling client disconnections and errors

The code also includes a basic unit test setup using unittest and unittest.mock to test functionalities like message receiving, message broadcasting, and accepting new connections.

Your submission should include the complete server code with all the requested features implemented.

Expected Server Output:

```
Testing accept new connection ...
Listening for connections on 127.0.0.1:65432...
Accepted new connection from 127.0.0.1:12345, nickname: TestUser
bind called with: call(('127.0.0.1', 65432))
listen called with: call()
accept called with: call()
recv called with: call(1024)
Testing broadcast ...
send receiver 1 called with: call(b'Hello, Group!')
send receiver 2 called with: call(b'Hello, Group!')
Testing receive message empty ...
False is False
Testing receive message exception ...
False is False
Testing receive message successful ...
recv return value: b'Hello, World!'
test attribute passed: b'Hello, World!' is equal to b'Hello, World!'
```