

Module	CSA
Lecturer	AVET / GRTS / KPYS
Room	2.72 / 2.76 / 2.78 / 2.79 / 2.84 / 2.86 / 2.88

Distribution of points:

Exercise	1	2	3	4	5	Total
Points	20	20	20	20	20	100

Remarks:

For each exercise, the exam points will be awarded if and only if the required functionality described in the exercise works (runs with no exceptions)!

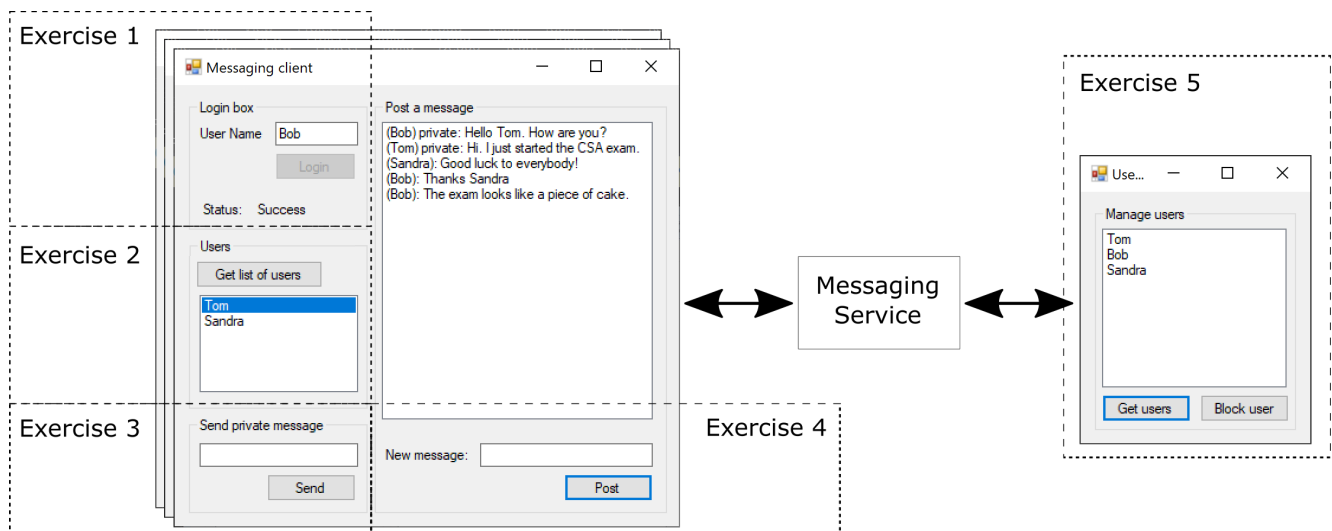


Figure 1. Example of the Client-Service application for messaging system.

Introduction

In this exam, you are going to implement a client-service system, for a student organization.

The messaging Service allows users to log into the system and send messages.

The Service stores a list of logged in users. Each user has:

- user name property (e.g. “Bob” or “Sandra”);
- a property which tells if user is blocked or not (true or false);
- a property to store user’s callback channel.

The given solution already contains two Client projects, each with one Form. The `MessagingClient` project is used by the users to send private (to a specific user) or public (to all users) messages. The `ManagerClient` is used by administrators of a student organization for blocking users. You should either add a new Service project to the existing solution, or you can make a new solution for the Service project. In this exam you are free to decide which type of endpoint configuration you will use (programmatic or administrative).

Exercise 1 [20 points]

Create functionality which after button “Login” is clicked, sends a user name to the Service and tries to log in a user. In the `labelLoginStatus` show if login was successful or not (see *Exercise 1* in Figure 1). A login is successful if there is no user in the Service with the given user name. The Service should store the list of all logged in users. Initially, the newly logged in user is not blocked. Additionally, if application is closed, user must be logged out from the Service.

Exercise 2 [20 points]

By clicking on the “Get list of users” button, the `MessagingClient` should retrieve the list of all ‘not blocked’ users from the Service and show their user names in the list box `lbUsers` (see *Exercise 2* in Figure 1). The name of the same (logged in) user should not be shown in his/her list of users.

Note: for each user, only the user name should be given to the Clients.

Exercise 3 [20 points]

A user (sender) should be able to send a private message to another user (receiver).

By clicking “Send” button (see *Exercise 3* in Figure 1), the sender sends a message entered in the text box `tbPrivateMessage` to a selected user from the list box `lbUsers`. The Service should forward the message to both, the sender and receiver, if:

- sender is not blocked;
- receiver is not blocked;
- receiver exists.

Therefore, if these three conditions are satisfied, a private message should appear in the list box `lbMessages` of both Clients, the sender and receiver.

A private message should appear in the following format: “(*{sender’s user name}*) private: (*{message text}*)”.

Exercise 4 [20 points]

A user (sender) should be able to send a public message to all 'not blocked' users.

By clicking "Post" button (see *Exercise 4* in Figure 1), the MessagingClient sends a message entered in the text box tbPublicMessage to the Service. The Service should forward a message to all users (receivers) if:

- sender is not blocked;
- receiver is not blocked.

Therefore, if these two conditions are satisfied, the public message should appear in the list box lbMessages of all 'not blocked' users.

A public message should appear in the following format: "{sender's user name}): {message text}".

Exercise 5 [20 points]

Add a new business endpoint to the Service: the new endpoint will be used by the system manager to retrieve all logged in users (both 'blocked' and 'not blocked') and block a user if necessary.

By clicking "Get users" button (see *Exercise 5* in Figure 1), the ManagerClient should retrieve the list of all logged in users and display them in the list box lbUsers.

By clicking "Block user" button, the ManagerClient should send the name of the selected user to the Service. The Service should block and inform the blocked user with the following message: "You have been blocked by admin".