

1. Functional requirements

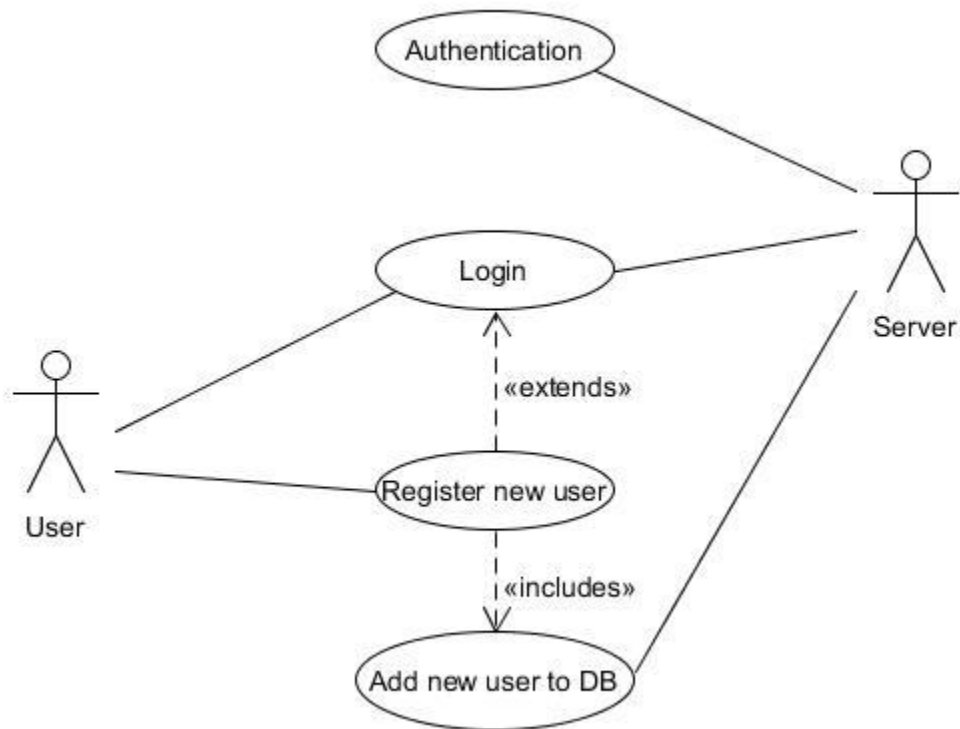
- 1.1 The user shall have the possibility to create a new account with unique Login and custom password. In that case the connection with the server shall be established first.
- 1.2 The user shall have the possibility to delete his own account. It is only possible if the user is connected to the server and confirms the deletion.
- 1.3 The user shall have the possibility to create his own chat room with chosen people. After selecting option to create his chat room the list of all currently online users shall be printed to the screen with an option to invite selected users, or to type their logins.
- 1.4 The user shall have the possibility to join public chat room.
- 1.5 The user shall have the possibility to accept an invite to a private chat room. An invite shall be contained in the invite list.
- 1.6 The user shall have the possibility to login to his account. Application shall check if the information typed in are correct.
- 1.7 The user shall have the possibility to exit program at any given time or to go back to main menu. In each screen there shall be an option listed.
- 1.8 The user shall have the possibility to view a list of all logged in users.
- 1.9 The program shall be command-line based.

2. Non-functional requirements

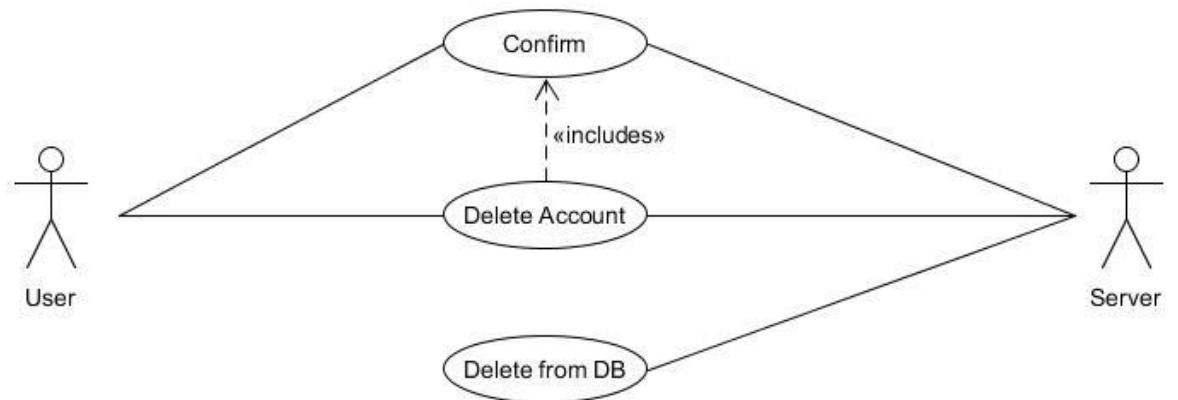
- 2.1 The application code and documentation shall be written in English language .
- 2.2 The application code shall be stored in a version control system.
- 2.3 The server application design shall incorporate the Thread Pool design pattern.
- 2.4 Upon exit the application shall perform a resource clean-up – free all resources and perform proper threads exit.
- 2.5 The application design shall be done in UMLet and contain: Use case diagrams, Class diagram, Sequence diagram.
- 2.6 The application cod shall be written in C++.
- 2.7 The application shall be compiled with g++.
- 2.8 The application shall use sockets to communicate client program with server program.
- 2.9 The application code shall be tested with unit tests.

3. Use Case Diagrams

3.1 Logging



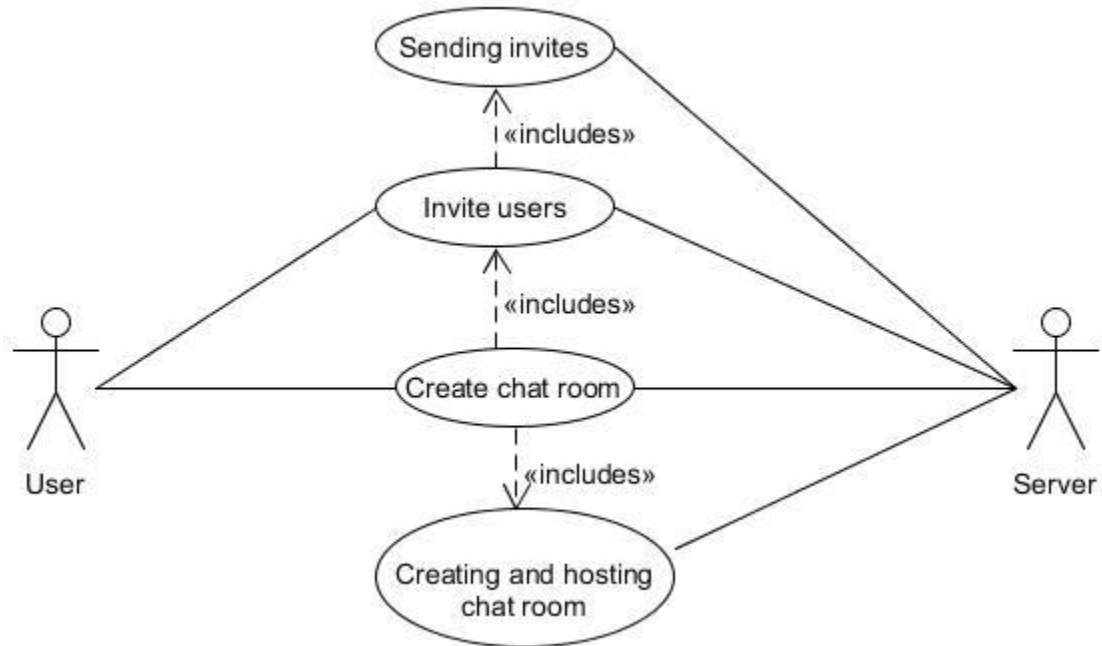
3.2 Deleting



3.3 Joining Public Chat room

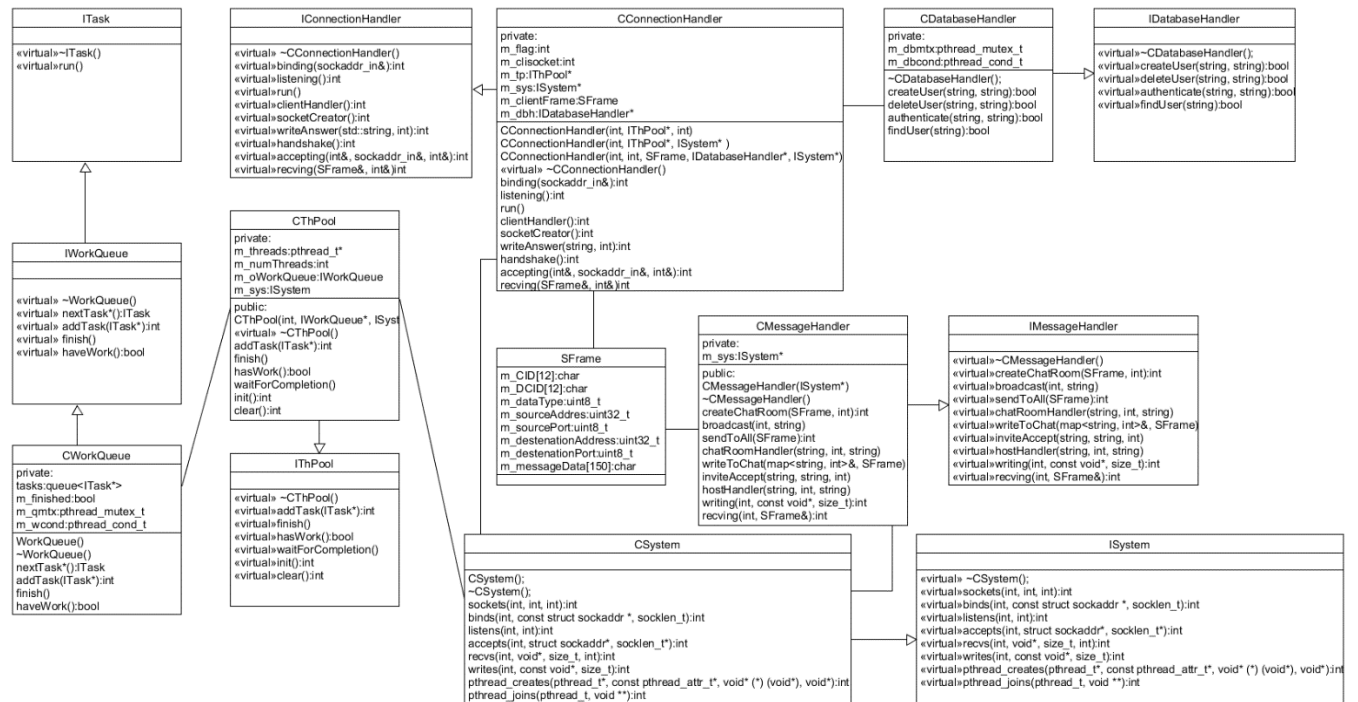


3.4 Creating Private Chat room

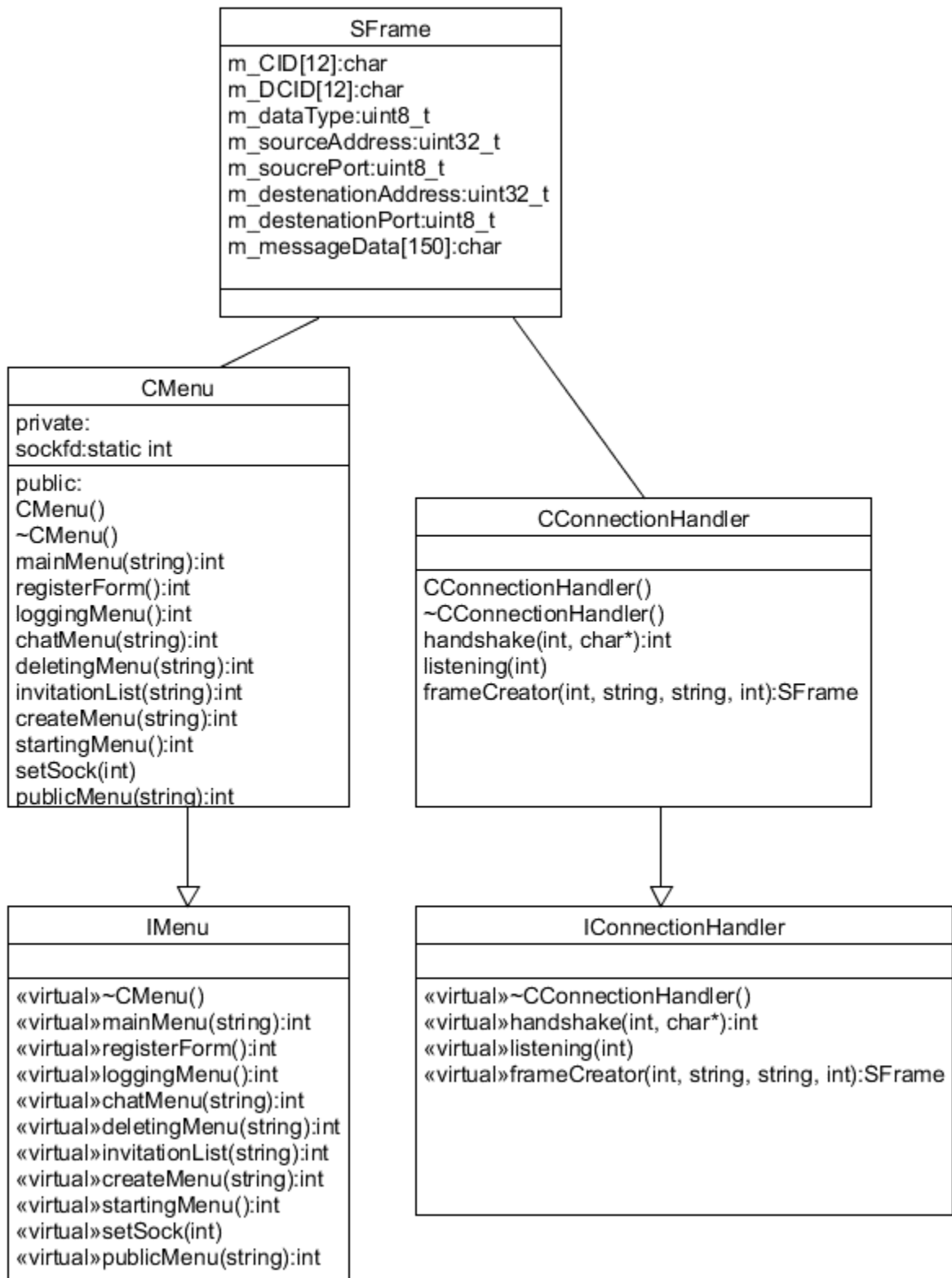


4. Class diagrams

4.1 Server class diagram

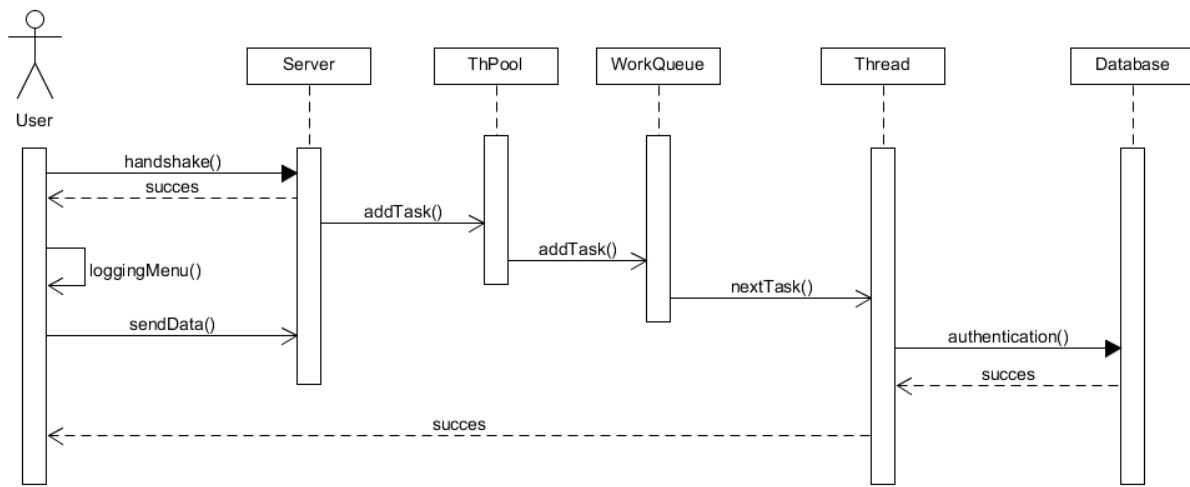


4.2 Client class diagram

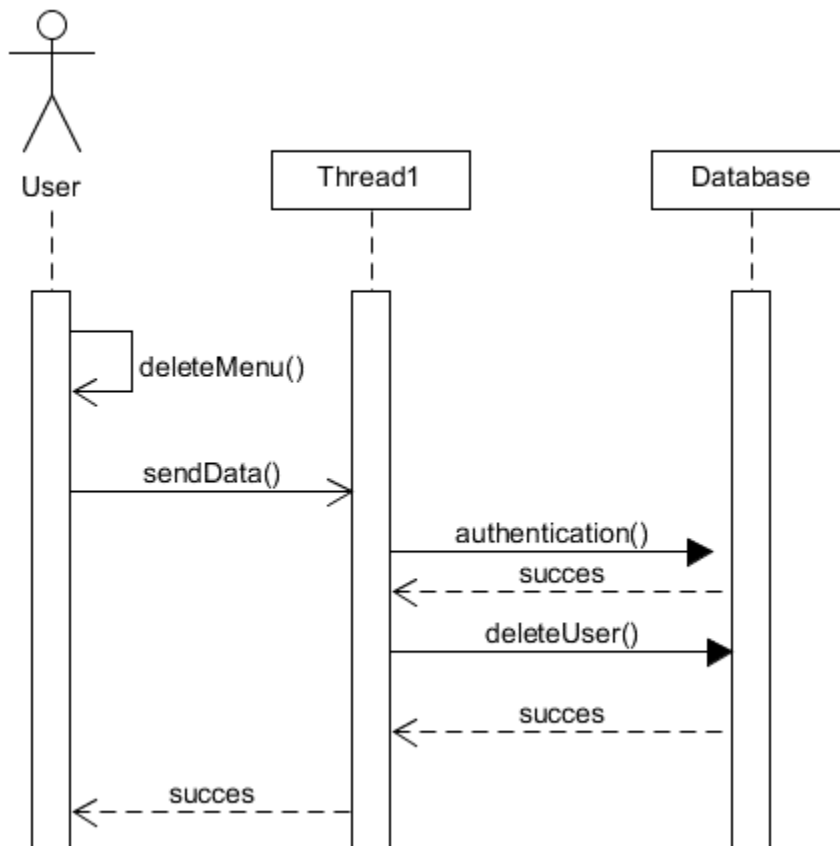


5. Sequence diagram

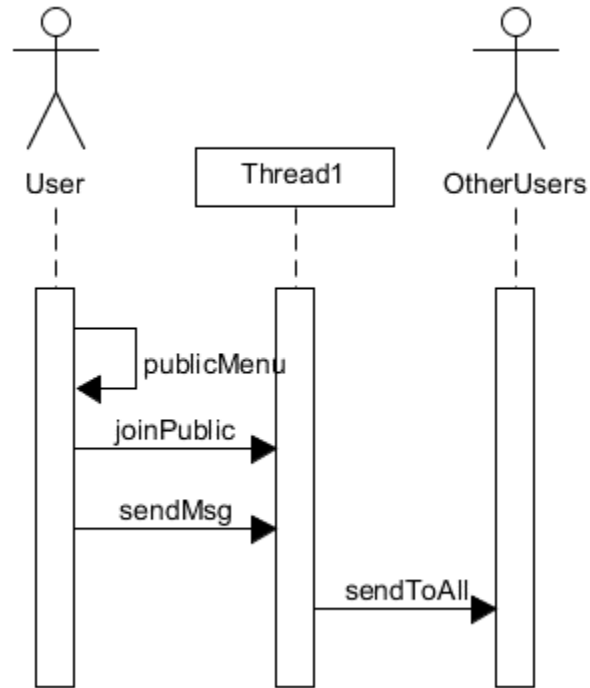
5.1 Logging



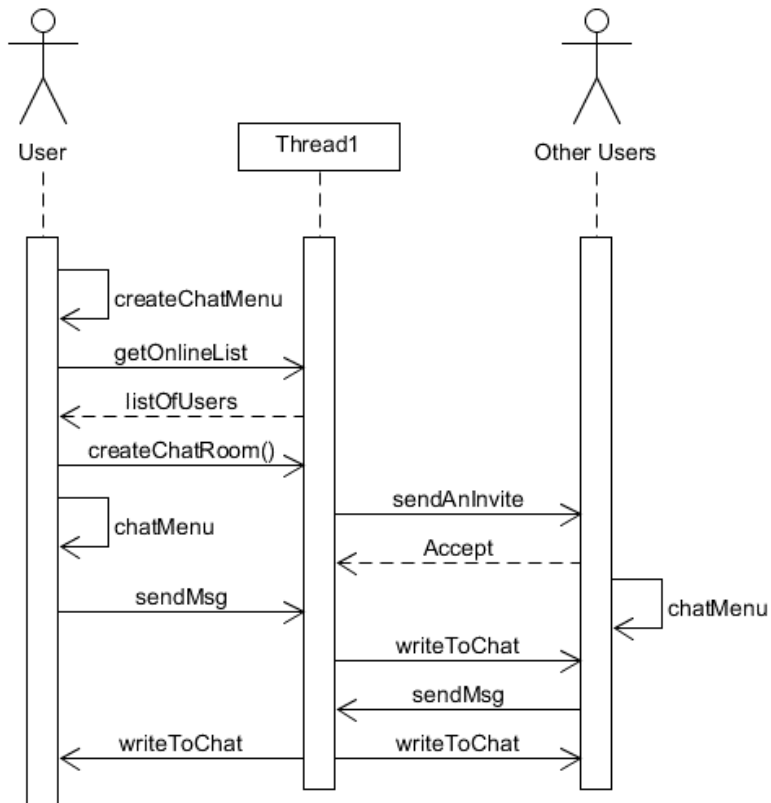
5.2 Deleting



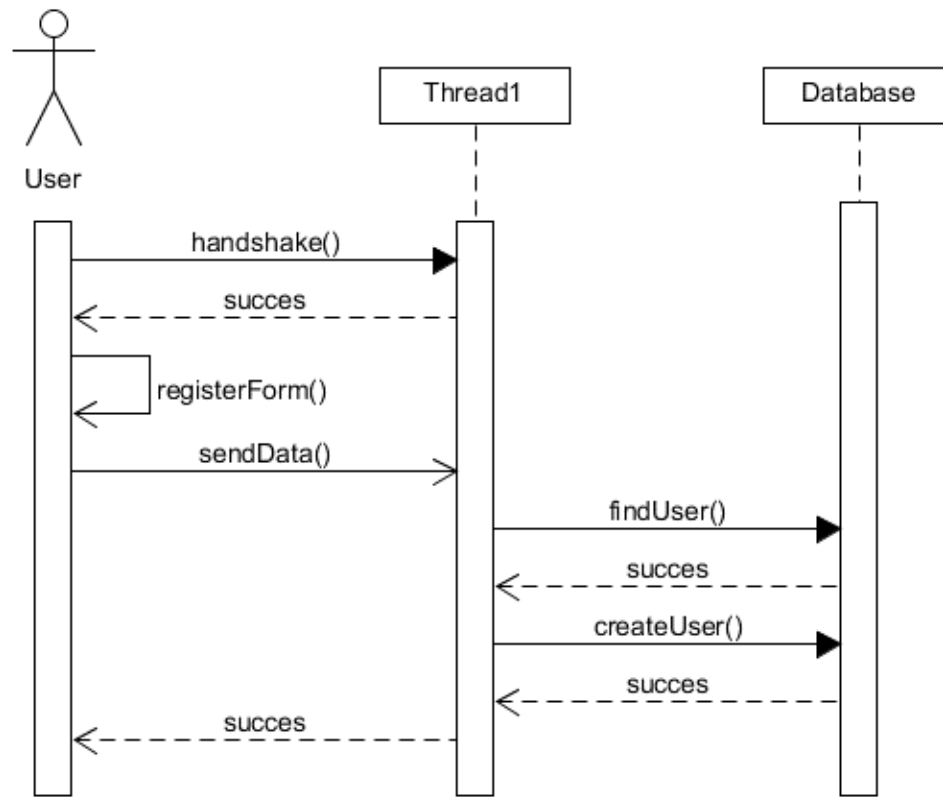
5.3 Join Public



5.4 Creating private chat room



5.5 Registering new user



6. Flow

6.1 Logging

Name:	Logging
Preconditions:	<ul style="list-style-type: none">- user have to have internet connection- user have to type in right port number for the right server
Normal flow:	<ol style="list-style-type: none">1. User starts the application2. User choses logging option3. User types in his password and login4. User types in port number for the server he wants to connect to5. User confirms what he types in6. Application try's to connect to the server7. Server accepts the connection and returns information of success8. Server authenticates the data given by user and returns the information of success
Exceptional flow:	<ul style="list-style-type: none">- User's application crashes- User choses wrong option- User types in wrong password or login- User does not have an account- User types in wrong port- User does not confirms his input- Application returns timeout- Server discards the connection

6.2 Deleting an account

Name:	Deleting an account
Preconditions:	<ul style="list-style-type: none">- User is logged in to his account- User is in the main menu
Normal flow:	<ol style="list-style-type: none">1. User selects option to delete his account from the main menu2. User types in his password and confirms his choice3. Server authenticates the data given by user4. Server deletes an account from database5. User is given an information about success of the deletion6. User is forced to logout
Exceptional flow:	<ul style="list-style-type: none">- User choses wrong option- User types in wrong password- User does not confirms his input- Application returns timeout- Server cannot deletes and account

6.3 Join Public chat

Name:	Joining public chat room
Preconditions:	<ul style="list-style-type: none">- User is logged in to his account- User is in the main menu
Normal flow:	<ol style="list-style-type: none">1. User choses the option to join public chat from the main menu2. Server moves user to public chat
Exceptional flow:	<ul style="list-style-type: none">- User choses wrong option- Application returns timeout

6.4 Creating private chat room

Name:	Creating private chat room
Preconditions:	<ul style="list-style-type: none">- User is logged in to his account- User is in the main menu
Normal flow:	<ol style="list-style-type: none">1. User choses the option to create his own chat room2. Server creates chat room3. Server gives user list of online users to invite4. User type in the name of chosen user or users5. Server sends invites to the invited users6. Invited users accept the invitation7. Server puts all users in the chat room
Exceptional flow:	<ul style="list-style-type: none">- User choses wrong option- Application returns timeout- User types in wrong user name to invite- Invited user does not accepts the invite

6.5 Registering of new user

Name:	Registering new user
Preconditions:	<ul style="list-style-type: none">- user have to have internet connection- user have to type in right port number for the right server
Normal flow:	<ol style="list-style-type: none">1. User starts the application2. User choses option to register his new account3. User types in his login, two times his password and port number of the server4. User confirms what he types in5. Application try's to connect to the server6. Server accepts the connection and returns information of success7. Server checks if there is an account with the same login8. Server adds new user to the database9. Server returns user an information of successful creation of the new account
Exceptional flow:	<ul style="list-style-type: none">- User choses wrong option- Application returns timeout- User types in wrong server port- User's login is already taken- User does not confirms his input