Chat-Pack

System Specification

Stephan Do

Nico Vasquez

Thomas Antensteiner

HTBLA LEONDING

Content

[1. Initial Situation and Goals 3](#_Toc502760921)

[1.1. Initial Situation 3](#_Toc502760922)

[1.1.1. Application Domain 3](#_Toc502760923)

[1.1.2. Glossary 3](#_Toc502760924)

[1.1.3. Model of the Application Domain 3](#_Toc502760925)

[1.1.4. Overview of the Business Processes 3](#_Toc502760926)

[1.1.5. Description of the Business Processes 3](#_Toc502760927)

[1.2. Goal Definition 3](#_Toc502760928)

[2. Functional Requirements 4](#_Toc502760929)

[2.1. Use Case Diagrams 4](#_Toc502760930)

[2.2. Interact with friends 4](#_Toc502760931)

[3. Non-functional Requirements 5](#_Toc502760932)

[3.1. Usability 5](#_Toc502760933)

[3.2. Efficiency 5](#_Toc502760934)

[3.3. Maintainability 5](#_Toc502760935)

[3.4. Security 5](#_Toc502760936)

[3.5. Legal constraints 5](#_Toc502760937)

[4. Quantity Structure 6](#_Toc502760938)

[5. System Architecture and Interfaces 7](#_Toc502760939)

[6. Acceptance Criteria 7](#_Toc502760940)

[6.1. Manage friend-list 7](#_Toc502760941)

[6.2. Communicate with friends 7](#_Toc502760942)

[6.3. Get friends info 8](#_Toc502760943)

1. Initial Situation and Goals
   1. Initial Situation
      1. Application Domain
      2. Glossary

User: The User is the person who starts the program and logs in to use it

Friend list: The friend list is the list of profiles with are friends with your profile

Friend: A friend is another user who sent a friend-request to you and you accepted it or the other way.

* + 1. Model of the Application Domain
    2. Overview of the Business Processes
    3. Description of the Business Processes
  1. Goal Definition

1. Functional Requirements
   1. Use Case Diagrams
   2. Interact with friends
2. Non-functional Requirements
   1. Usability

As already mentioned we want to have a low memory use that the chat program can easily run in the background while working/playing videogames without delay/lags.

Also, the interface should be so clear that the user finds all he needs at his first look.

Another important point is that the user should be able individualize the overlay as he likes it.

The program should deliver messages instant (in under 5 seconds) *without any delay* so that, if the user is able to/wants to, he can respond immediately.

* 1. Efficiency

As already mentioned in the Usability paragraph, the program should deliver messages in under 5 seconds and also ,as mentioned in the previous paragraph, the program should use less memory than programs like skype or discord (We are trying to reach the goal of under 100 MB ). Finally it should open in under 10 seconds so that the user can start communicating.

* 1. Maintainability

If some bugs will appear we will fix them as soon as possible and we will add little features the users ask for.

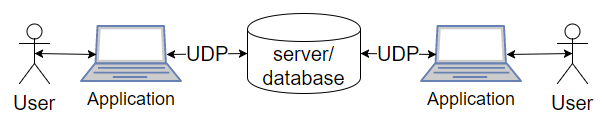
* 1. Security

The private data like the password won´t be passed on to anyone.

* 1. Legal constraints

As far as the project goes we didn´t find any standards we need to respect.

1. Quantity Structure
2. System Architecture and Interfaces



The Application is going to be the user-interface. And in the background the application will send/receive messages via UDP to the server where all users with password and ID are saved. The chat history is going to be saved with the application.

1. Acceptance Criteria
   1. Manage friend-list

|  |  |
| --- | --- |
| **functionality** | **Expected result** |
| Send a friend-request | The asked user gets a friend-request |
| Accept friend-request | A new profile is added to your and your new friends friend-list |
| Decline friend-request | The user who sent the friend-request gets a message |
| Remove friend | The profile is removed from your friend-list |

* 1. Communicate with friends

|  |  |
| --- | --- |
| functionality | Expected result |
| Send a message | The user gets the message on his screen |
| Start a call | The called user is getting a call-request |
| Accept call-request | The two users are in a conference where they can talk |
| Ignore call-request | The user who started the call gets a message |

* 1. Get friends info

|  |  |
| --- | --- |
| functionality | Expected result |
| Press the “stats” button | The user is able to see the stats between him/her and one of his/her friend |
| Look under the profile | See what your friend is currently doing |