[Project Name: "Potatolk"]

Architecture/Design Document

Table of Contents

Version: (Last Created: <2.0>
Modifier: All team members (in particular the graphic designers: Gaole Elia & Grippi Giulio
Date: 12/08/2019
Description of Change: Added a particularly intuitive graphic appearance with a dark theme

---

1. Introduction

Architecture and Design:
The program code is structured in 8 distinct classes implemented in java.

1. Messages
2. Client Status
3. Control Messages
4. PacketReceiver
5. PacketInterpreter
6. Message
7. Connection
8. Group

Each of these classes performs the function of managing the title of the classes.
For example, of course the "messages" class checks and stores the messeges, and so on.

The objectives of this application are the creation of a chat through which the user can interact with other users, therefore used in particularly public contexts such as schools or companies. But if you want to implement it at home too.
The architecture created meets the requirements, apart from some bugs that will be fixed with

future versions.
The application is particularly usable because it has a simple and commonly used interface.

2. Logical View

The relationships between the classes described above are as follows:
Messages creates messages and communicates with Control Messages
Control Messages, manages messages with 2 basic methods
Client Status contains methods for doing user actions
Thread (which activates when you have successfully logged in)
PacketInterpreter: determines all the message opcodes
Connection: with it you connect and KEEP the connection
Group: maintains the group information of the chat

3. Process View

Only a single class named like "Thread" has been developed.
The "thread" class that activates when a successful login occurs.

4. Development View

The system developed theoretically adapts well enough to what has turned out on a practical level.
the classes interact well enough, which allows the whole application to function properly.

5. Use Case View

The main use cases,
as previously said, they could be those inserted in a school context, perhaps for the management of messages that must be sent between teams organized within the classroom of a school.
It could also be used as a chatting protocol between professors (always at school level).
It could be used in more business and professional contexts, but for this we prefer to wait for the development of better and more updated versions.
With the development of the most updated versions, this application could be put on the net, making it almost a social network, and therefore a public service.

6. Design Goals

Since it will not be possible to effectively express the quality of a design,
since the company is unable to judge its own design product,
it will be necessary to list the priorities that this application has set itself:

- the management of messages
- correct chat management

- sufficiently intuitive and pleasant graphics

7. High-Level Design (Architecture)

The structure of the classes and their functioning has been previously explained.
however, it is possible to view, through this link (https://docs.google.com/presentation/d/16NdDb0XYV2mN_gk75oJb8Mk-BwtCl631yxYhcMjBVlw/edit?usp=sharing) a presentation containing the description of each individual operation.
UML diagrams were not used for the creation of this project, as its realization was not difficult to manage, and therefore did not require the realization of the diagram in order to facilitate understanding.

8. Mid-Level Design

The application that makes use of this protocol (the potatolk) uses both dynamic and static elements. This is in the programming part and in the graphic part.
An example that clarifies this situation are the drop-down menus that concern a more dynamic part, on the contrary instead of all the static elements such as the buttons that the user can press, they all lead to mainly static pages.
In the functional programming part there are dynamic elements such as the functions performed by the classes, the continuous refresh of users in the chat group, and so on.

9. Physical View

The physical components of the entire structure with which the application was made are located in the school
ITIS Guglielmo Marconi (VR), the application in any case, when it should take development in future versions with improved functions, could be put on a server in order to make it public, but these are assumptions.

https://lh3.googleusercontent.com/proxy/DIJTmoc6k0hxPJ2DMuVdhiIUOWPo6kGv7gAr237P63niWwDYlylKoDUK8-DNV3tGg3Z4_CXNsqYd7JGS2Cls

10 . Use Case View

It is possible to view from the image content viewable through the following link

https://lh3.googleusercontent.com/proxy/DIJTmoc6k0hxPJ2DMuVdhiIUOWPo6kGv7gAr237P63niWwDYlylKoDUK8-DNV3tGg3Z4_CXNsqYd7JGS2Cls

We understand the primoridial context in which the application is used, which, as previously said, would seem the scholastic one as more questionable.
A proposed visualization is one in which the professor, who will be located in the "cattedra" section of the image, will naturally have greater privileges (still to be implemented in the application), compared to the students each of whom will be located in the "alunno x" section of the linked image.