Requirements and Analysis Document for . . .

Carl stling, Yazan Ghafir, Erik Gunnarsson, Mohamad Almasri

date
version

1 Introduction

This document will analyze and discuss the project "CEYMChat" created by group 21. The problem that the "CEYMChat" application tries to solve lies within human communication. By creating a simple chat application with an understandable Graphical User Interface (GUI), human interaction and communication will flourish. Anyone who wishes to communicate via online chat with friends or other users can easily do so using "CEYMChat". The application requires less personal data than other existing chat applications on the market. "CEYMChat" is a lightweight chat application developed using the java.net library. It requires no installation of software on the users device and requires minimal personal information during sign-up. No files need to be saved on the users device.

1.1 Definitions, acronyms and abbreviations

"Group 21" refers to Erik Gunnarsson, Carl stling, Yazan Ghafir and Mohamad Almasri.
"CEYMChat" refers to the application built by group 21 during this project.
"GUI" refers to a Graphical User Interface.
".net" refers to the java.net library used for network connections in the project.
"Multithreading" refers to the usage of multiple threads in a program in order to execute several tasks at the same time.

1.2 Definitions, acronyms, and abbreviations

2 Requirements

2.1 User Stories

Use the template from the course website and list all user stories here. It is fine to have them in an spreadsheet (or other application) at first, but they must end up here as well.

These user stories should describe what the user will be able to do. Write a the user stories in language of the customer, and give the a unique ID. List the user stories in priority order.

2.2 User interface

Sketches, drawings and explanations of the application user interface (possible navigation).
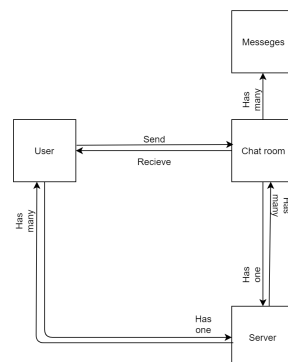
3 Domain model

Figure 1: High-level UML-diagram of the application

3.1 Class responsibilities

Name	Module	Summary
ClientMain	Client	This is the runnable class of the Client. It simply launches the program.
ClientModel	Client	This class contains the model of the Client. It is responsible for requests/replies to the server.
Message	model-lib	A generic class that encapsulates the data to be sent/received by the actors.
MessageFactory	model-lib	Factory for the Message-class. Narrows the generic element in Message to only be constructed with specified classes.

4 References