**Table of Contents**

**1.0 Project Overview**

**1.1 Abstract**

The project is a simple group chat, a chat between mutliple users, using a chat client to send messages between them. Each message is transmitted to a server, which broadcasts the message to all connected clients. Therefore the chat is splitted between a client application and a server application.

**1.2 Target Customers**

Our chat targets anyone who wants to communicate with others in a group chat. This can be usefull everywhere where collaboration or socializing is needed.

* Team manager or the like (direct customer)
* Team members or the like (indirect customer)

**1.3 Keywords**

Group; Chat; Multi; User;

**1.4 Scope and Objectives**

* chat is splitted between the client application and the server application
* each client is connected to a server
* to connect to a server the user uses the servers ip address and port number, which are entered into a graphical formular
* on a successful connect the user can enter his messages into a small text box
* on enter or if the send button was pressed the client sends this message to the server, which broadcasts the message to all connected clients
* if a message is received by the client it gets drawed by a big text box
* usernames are used, so that each user is identifiable to other users
* the transmission control protocol (TCP) is used as transport protocol, since reliable connections and transfers between server and clients are needed
* the implementation of cross platform TCP sockets is done by the network module of the SFML library
* the implementation of cross platform thread handles is done by the system module of the SFML library
* as GUI library FLTK is used
* client and server use the same core code, which implements necessary functions and classes especially the chat protocol
* the client application uses SFML for threads and sockets and FLTK for the graphical user interface
* the server application uses only SFML for threads and sockets

**2.0 Team and Contraints**

**2.1 Team profile**

Josephine Lipkin

Justus Flerlage

* good programming skills, especially with C++
* already realized a few projects on his own

**2.1 Challenges**

* designing a simple chat protocol, which can be extended easily
* writing fast and reliable network code for both server and client

**2.2 Assumptions and Contraints**

**3.0 Deliverables and Milestones**

**3.1 Project Deliverables**

**3.2 Schedule and Budget Summary**

|  |  |
| --- | --- |
| Item | Date |
| Project Proposal | April 19, 2015 |
| Proposal Presentation | April 15, 2015 |
| Demonstration and Delivery | May 7, 2015 |