**Project 01 Proposal**

**PhatChat**

**Josephine Lipkin, Justus Flerlage**

**April 19, 2015**

**Table of Contents**

1.0 Project Overview 2

1.1 Abstract 2

1.2 Target Customers 2

1.3 Keywords 3

1.4 Scope and Objectives 3

2.0 Team and Contraints 4

2.1 Team Profile 4

2.1 Challenges 5

2.2 Assumptions and Contraints 5

3.0 Deliverables and Milestones 5

3.1 Project Deliverables 5

3.2 Schedule and Budget Summary 6

**1.0 Project Overview**

**1.1 Abstract**

PhatChat 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 split 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**

* developed in C++11
* utilize “cmake” as cross platform build system
* chat is split between the client application and the server application
* each client(user) is connected to a server
* to connect to a server, the user types in server's ip address and port number into a provided window
* once a successful connection is established, the user can enter his messages into a small text box
* Pressing enter or clicking on send button on the window will send the client's/clients' messages to the server, which broadcasts the message to all connected clients
* Once the client receives a message, the message is displayed in a text box
* unique 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
* For the GUI, the 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
* the server application uses only SFML for threads and sockets

**2.0 Team and Contraints**

**2.1 Team Profile**

Josephine Lipkin

* moderate programming skills
* good design background

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
* avoid resource consuming applications
* designing a simple yet effective GUI

**2.2 Assumptions and Constraints**

* application will not be used on mobile devices
* if the server shutdowns, the data (chat history) will be lost

**3.0 Deliverables and Milestones**

**3.1 Project Deliverables**

* Project Proposal
* Manual
* Source Code
* “cmake” build script

**3.2 Schedule and Budget Summary**

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