29.3.2015

Luca Tirabassi, Alex Theijs

Modulprüfung M120

Dokumentation und Eportfolio zum Chatprogramm

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc392531109)

[1.1 Aufgabenstellung Allgemein 3](#_Toc392531110)

[1.2 Spezialisierte Aufgaben für Unsere Gruppe 3](#_Toc392531111)

[1.2.1 Generieren eines \*.bfb – Files 3](#_Toc392531112)

[2 Analyse 4](#_Toc392531113)

[2.1 Problematik Codegrösse 4](#_Toc392531114)

[2.2 Problematik auslesen 4](#_Toc392531115)

[2.3 Problematik Koordinaten verarbeiten 4](#_Toc392531116)

[2.3.1 Alle Koordinatendifferenzen anpassen 4](#_Toc392531117)

[2.3.2 Es sollten Grundsätzlich immer 4 Linen pro milimeter enthalten sein. 4](#_Toc392531118)

[2.4 Problematik Grössen wählen 4](#_Toc392531119)

[2.5 Problematik Filestruktur und Befehle 4](#_Toc392531120)

[3 Konzept 6](#_Toc392531121)

[3.1 Programmablauf 6](#_Toc392531122)

[3.2 Programmstruktur 7](#_Toc392531123)

[3.3 Testplanung 13](#_Toc392531124)

[3.3.1 Test calculatefactors 13](#_Toc392531125)

[3.3.2 Test createGoToinstruction 13](#_Toc392531126)

[3.3.3 Test definefactors 13](#_Toc392531127)

[3.3.4 defineindexedlines 13](#_Toc392531128)

[3.3.5 findbiggestandsmallestcoordinates 13](#_Toc392531129)

[3.3.6 getusedlinesfornewobject 13](#_Toc392531130)

[3.3.7 isentryvalid 14](#_Toc392531131)

[3.3.8 setui 14](#_Toc392531132)

[3.3.9 writefile 14](#_Toc392531133)

[3.4 Userinterface 14](#_Toc392531134)

[3.4.1 Hauptinterface 14](#_Toc392531135)

[3.4.2 Falsche Eingabe 15](#_Toc392531136)

[3.4.3 Richtige Eingabe 15](#_Toc392531137)

[3.4.4 Error bei Erstellung 15](#_Toc392531138)

[4 Quellenverzeichnis 16](#_Toc392531139)

[5 Arbeitsplanung / Arbeitsjournal / Rechnung 17](#_Toc392531140)

[5.1 Arbeitsplanung 17](#_Toc392531141)

[Montag: 05.05.2014 17](#_Toc392531142)

[Montag: 12.05.2014 17](#_Toc392531143)

[Montag: 19.05.2014 17](#_Toc392531144)

[Montag: 26.05.2014 18](#_Toc392531145)

[Montag: 02.06.2014 18](#_Toc392531146)

[Montag: 16.06.2014 18](#_Toc392531147)

[Montag: 23.06.2014 18](#_Toc392531148)

[5.2 Arbeitsjournal 19](#_Toc392531149)

[Montag: 05.05.2014 19](#_Toc392531150)

[Montag: 12.05.2014 19](#_Toc392531151)

[Montag: 19.05.2014 20](#_Toc392531152)

[Montag: 26.05.2014 20](#_Toc392531153)

[Montag: 02.06.2014 20](#_Toc392531154)

[Montag: 09.06.2014 20](#_Toc392531155)

[Montag: 16.06.2014 21](#_Toc392531156)

[Sonntag: 22.06.2014 21](#_Toc392531157)

[Montag: 23.06.2014 21](#_Toc392531158)

[Samstag 28.06.2014 21](#_Toc392531159)

[Sonntag 29.06.2014 22](#_Toc392531160)

[6 Programmcode 23](#_Toc392531161)

[7 Projektablauf für jede Person 34](#_Toc392531162)

[7.1 Theijs Alex 34](#_Toc392531163)

[7.2 Kohler Carmen 34](#_Toc392531164)

[8 Rechnung für Konzept und Analyse 35](#_Toc392531165)

# 1 Aufgabenanalyse

## 1.1 Beschreibung des Problems / System

Unser gewähltes Betriebssystem ist Windows. Weil wir uns für ein Chatprogramm entschieden haben, soll es dem Client möglich sein mit anderen Clients über einen Server miteinander zu kommunizieren. Auch der Server sollte ein Microsoft Windows Betriebssystem haben, da wir mit .NET arbeiten. Des Weiteren sollte eine Internetverbindung vorhanden sein, oder zumindest ein lauffähiges Netzwerk, damit die verschiedenen Clients miteinander kommunizieren können.

## 1.2 Use-Cases für zentrale Aufgaben

Folgende Usecases haben wir ausgearbeitet:

1. User startet Programm
2. User gibt in Textfeld seinen Namen ein
3. User loggt sich ein
4. User wählt Chatpartner aus
5. User wählt Chattextbox aus
6. User schreibt in TextBox
7. User sendet geschriebene Nachricht
8. User erhält geschriebene Antwort
9. User schliesst Applikation

# 2 Benutzeranalyse

Unsere Zielgruppe sind unerfahrene Computerbenutzer, welche einfach und schnell mit einer kurzen Installation und wenigen Klicks miteinander einfach und schnell kommunizieren wollen.

# 3 Ergonomieanalyse

## 3.1 Grundsätze

* Das Chatprogramm soll den Benutzer nicht überfordern und nur das können was wirklich von Ihm erwartet wird.
* Das Programm sollte möglichst selbsterklärend sein 🡪 Das bedeutet, dass der Benutzer ohne jegliches lesen einer Dokumentation mit dem Programm umgehen kann.
* Unser Programm soll ähnlich funktional und Benutzeroberflächen technisch gestaltet sein wie bereits vorhandene Produkte, wie zb.: Skype, ICQ, MSN, Viber
* Dem Benutzer soll es nicht möglich sein falsche Eingaben zu tätigen. Falls dies doch passieren sollte, soll dem Benutzer angezeigt werden, was er falsch gemacht hat.
* Die Bedienbarkeit des Programms soll schnell und einfach lernbar und ausführbar sein