**Anforderungsheft**

Entwicklung einer portablen P2P Chat-Anwendung für den Einsatz in bestehenden Netzwerken

|  |  |
| --- | --- |
| Voraussetzung: | Lastenheft |
| Autoren: | Ludwig Cornely, Julian Hanfgarn |
| Version: | 1.0 |
| Zuletzt geändert: | 07.07.2016 |
| Hinweis: | Erklärungen zu technischen oder nicht hinreichend definierten Begriffen finden sich im Glossar am Ende dieses Dokuments. |

Inhaltsverzeichnis

[Abbildungsverzeichnis 3](#_Toc456087131)

[1. Zielbestimmung 4](#_Toc456087132)

[1.1 Musskriterien 4](#_Toc456087133)

[1.2 Wunschkriterien 4](#_Toc456087134)

[1.3 Abgrenzungskriterien 4](#_Toc456087135)

[2. Produkteinsatz 5](#_Toc456087136)

[2.1 Anwendungsbereiche 5](#_Toc456087137)

[2.2 Zielgruppen 5](#_Toc456087138)

[2.3 Betriebsbedingungen 5](#_Toc456087139)

[3 Produktumgebung 6](#_Toc456087140)

[3.1 Software 6](#_Toc456087141)

[3.2 Hardware 6](#_Toc456087142)

[3.3 Produkt-Schnittstellen 6](#_Toc456087143)

[4 Produktfunktionen 7](#_Toc456087144)

[4.1 Chat-Funktion 7](#_Toc456087145)

[4.2 Autorun-Funktion 7](#_Toc456087146)

[4.3 Dateitransfer-Funktion 7](#_Toc456087147)

[4.4 Verschlüsselungs-Funktion 8](#_Toc456087148)

[5 Produkt-Daten 9](#_Toc456087149)

[5.1 Produktdaten im Detail 9](#_Toc456087150)

[5.2 Übersicht der Daten und Beziehungen 9](#_Toc456087151)

[6 Benutzeroberfläche 10](#_Toc456087152)

[6.1 GUI 10](#_Toc456087153)

[6.2 NO-GUI 10](#_Toc456087154)

[7 Qualitäts-Zielbestimmung 11](#_Toc456087155)

[8 Globale Testszenarien / Testfälle 12](#_Toc456087156)

[9 Entwicklungsumgebung 14](#_Toc456087157)

[9.1 Software 14](#_Toc456087158)

[9.2 Hardware 14](#_Toc456087159)

[9.3 Orgware 14](#_Toc456087160)

[10 Glossar 15](#_Toc456087161)

# Abbildungsverzeichnis

[Abbildung 1: Use-Case-Diagramm Chatfunktion 7](#_Toc455414433)

[Abbildung 2: Use-Case-Diagramm Dateitransfer 8](#_Toc455414434)

# 1. Zielbestimmung

Ziel des Projektes ist es, eine portable Peer-to-Peer Chat-Anwendung für den Einsatz in bestehenden Netzwerken zu entwickeln.

## 1.1 Musskriterien

* Die Kommunikation soll nach dem Peer-to-Peer (P2P) Prinzip erfolgen.
* Die Anwendung soll ohne weitere manuelle Konfiguration oder Installation in einem bereits bestehenden Netzwerk funktionsfähig sein.
* Die Anwendung soll portabel, also über ein externes Medium ohne voriges Kopieren auf ein Hostsystem, ausführbar sein.

## 1.2 Wunschkriterien

* Verschlüsselung von Nachrichten
* Dateitransfer
* Betriebssystemunabhängige Autorun-Funktion
* Konsolen-Version

## 1.3 Abgrenzungskriterien

Die Anwendung dient primär dem Austausch von Kurznachrichten, um beispielsweise kleinere Support-Anfragen realisieren zu können. Weitere Spezifikationen der Anwendung sind optional.

# 2. Produkteinsatz

## 2.1 Anwendungsbereiche

Die Anwendung soll in einem beliebigen Unternehmen einsetzbar sein. Dabei beschränkt sich die Nutzung ausschließlich auf die Mitarbeiter.

## 2.2 Zielgruppen

Zielgruppe sollen sämtliche Mitarbeiter sein, welche über einen Rechner am Arbeitsplatz verfügen. Es werden grundlegende Kenntnisse im computertechnischen Bereich vorausgesetzt. Jedoch soll die Applikation intuitiv bedienbar sein. Besondere Rechte für den Anwender werden nicht benötigt.

## 2.3 Betriebsbedingungen

Das Tool soll bei Bedarf und ohne Vorarbeiten ausgeführt werden können. Dabei befinden sich die Clients in einem normal temperierten Raum ohne Einwirkung jeglicher extremer physikalischen Randbedingungen.

# 3 Produktumgebung

## 3.1 Software

Die Software befindet sich auf einem Speichermedium, z.B. auf einem USB-Stick. Es wird Java auf dem Endsystem benötigt, um die Anwendung ausführen zu können.

## 3.2 Hardware

Die erforderliche Hardware beschränkt sich auf gängige Systeme. Es sind keine besonderen Geräte notwendig.

## 3.3 Produkt-Schnittstellen

Die Bedienung der Anwendung erfolgt mittels standardisierten Eingabegeräten (Maus/Tastatur).

# 4 Produktfunktionen

Es ergeben sich folgende Funktionalitäten für den Anwender.

## 4.1 Chat-Funktion

Dies ist die Kern-Funktion der Anwendung. Sie realisiert einen Kurznachrichten-Dienst, welcher die Kommunikation mit einem beliebigen Mitarbeiter im Netzwerk ermöglicht. Hierzu wird zunächst der Empfänger aus einer Liste ausgewählt, um ihn dann kontaktieren zu können. Zuvor wird automatisiert ein IP-Scan ausgeführt, um die in Frage kommenden Empfänger zu ermitteln.

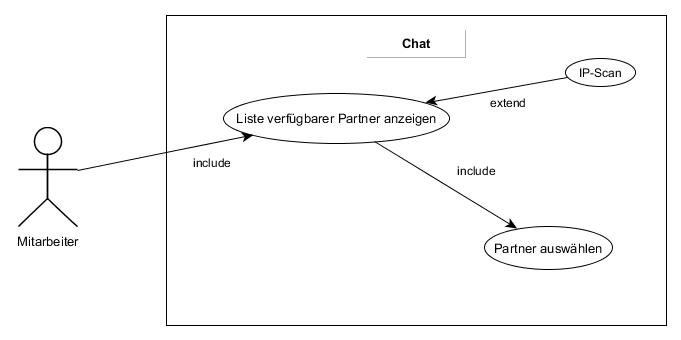


Abbildung 1: Use-Case-Diagramm Chatfunktion

## 4.2 Autorun-Funktion

Diese Funktion ist keine Funktion im generellen Sinne. Sie soll gewährleisten, dass das Programm startet, sobald das Speichermedium, auf dem es sich befindet, eingeführt und gelesen wurde. Eine einfache Bedienung, ohne vorher im Datei-Manager das entsprechende Medium ausgewählt zu haben, wird so ermöglicht. Auch soll die Funktion die Unabhängigkeit von Betriebssystemen beinhalten. Es ist also irrelevant, ob z.B. Windows oder Linux genutzt wird.

## 4.3 Dateitransfer-Funktion

Mit der Funktion soll dem Anwender die Möglichkeit gegeben werden, Dateien zu versenden. Es handelt sich bei der Funktion um eine Wunsch-Funktion. Sollte die Realisierung gelingen, wird in einem Dialogfenster die zu versendende Datei ausgewählt und anschließend übertragen. Dieser Schritt setzt natürlich voraus, dass der Speicherort der Datei bekannt ist. Auch eine Maximalgröße ist nicht auszuschließen, sodass keine sehr großen Dateien zum Transfer freigegeben sind.

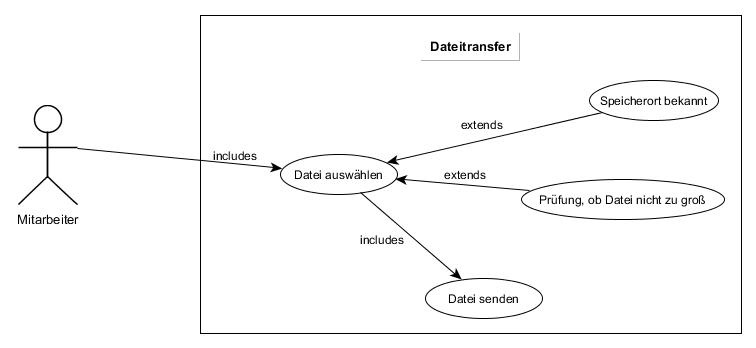


Abbildung 2: Use-Case-Diagramm Dateitransfer

## 4.4 Verschlüsselungs-Funktion

Es handelt sich hierbei ebenfalls um eine Wunsch-Funktion. Durch sie soll das Versenden und Empfangen von Nachrichten verschlüsselt geschehen, sodass Spionage durch einen Außenstehenden vermieden wird.

# 5 Produkt-Daten

## 5.1 Produktdaten im Detail

Die Anwendung erfordert keine direkte, konkrete Angabe von Daten. Der Datenfluss geschieht im Hintergrund. Jedes Gerät ist mit einer IP-Adresse versehen, welche nach weiteren im Netzwerk sucht. So wird dann der gewünschte Partner ausgewählt. Es wird von der Unicast-Technik Gebrauch gemacht.

## 5.2 Übersicht der Daten und Beziehungen

Es entsteht eine „Beziehung“ zwischen den beiden kommunizierenden Rechnern.

# 6 Benutzeroberfläche

Die Benutzeroberfläche der Anwendung wird mittels der Swing-Bibliotheken von Java realisiert und bietet Zugriff auf alle Programmfunktionen.

## 6.1 GUI

Verschiedene Teilbereiche der Applikation, wie z.B.: eine Liste verfügbarer Clients und ein Chatfenster, werden dabei getrennt von einander dargestellt. Die Oberfläche wird mit dem Ziel entworfen, intuitiv die Kernfunktion der Anwendung, den Chat, zur Verfügung zu stellen, und weitere Funktionen durch Auswahlmenüs verfügbar zu machen.

## 6.2 NO-GUI

Sollte das Budget des Projektes dies erlauben, wird eventuell zusätzlich eine No-GUI-Option integriert, welche die Anwendung in einem Konsolenfenster benutzbar macht. Dies würde der Nutzung an Systemen dienen, die über keine grafische Oberfläche verfügen. Die Steuerung fände vollständig mittels Tasteneingaben statt.

# 7 Qualitäts-Zielbestimmung

Die geplante Anwendung ist als Werkzeug mobiler Arbeitsgruppen konzipiert. Dadurch musste der Fokus auf Stabilität und flexible Einsetzbarkeit gelegt werden. Da die Applikation vor allem durch Fachkräfte eingesetzt werden soll, ist die Benutzerfreundlichkeit aller Funktionen außer dem grundlegenden Chat von geringerer Bedeutung. Um nicht mit den eigentlichen Aufgaben des Anwenders zu kollidieren muss die Anwendung speicherschonend sein, allerdings sind Anpassungen in diese Richtung aufgrund der Größe der Anwendung kaum zu erwarten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **sehr wichtig** | **wichtig** | **weniger wichtig** | **unwichtig** |
| **Benutzerfreundlichkeit** |  |  | x |  |
| **Stabilität** | x |  |  |  |
| **Performanz** |  | x |  |  |
| **Flexibilität** | x |  |  |  |

# 8 Globale Testszenarien / Testfälle

Diese Testfälle werden bei der Projektabnahme durchgeführt. Dabei wird (sofern nicht im Testfall anders beschrieben) vorausgesetzt, dass die eingesetzten Clients unkonfiguriert sind und noch keine Laufzeit hatten.

**/T010/ Netzwerkscan**

Zwei Clients werden zeitversetzt gestartet. Die automatische Konfiguration erzeugt einen Handshake im Netzwerk. Die verbundene IP und eine Client-ID werden angezeigt.

**/T011/ Chatnachricht**

Zwei Clients werden aktiviert, ein Handshake ausgeführt. An einem Endgerät wird der Kommunikationspartner ausgewählt und ein Chatfenster geöffnet. Der erste Nutzer versendet eine Textnachricht. Der zweite Nutzer beantwortet diese.

**/T012/ Verbindungsverlust**

Ein Client mit offener Verbindung wird vom Netz getrennt. Eine Fehlermeldung wird angezeit, der Verlauf archiviert und ein Netzwerkscan gestartet.

**/T020/ Code Injection**

Ein verbundener Client sendet ein ausführbares Codestück in Java und/oder einer Konsolen-Scriptsprache.

**/T021/ Speichergröße**

Eine sehr große Menge Text (ca. 50MB) wird von einem Client verschickt.

**/T031/ Speichermedium wird entfernt**

Der USB, auf dem die Daten eines Laufenden Clients gespeichert sind, wird entfernt. Die Anwendung speichert den Verlauf auf einer integrierten Festplatte (sofern verfügbar) und beendet sich selber mit einer Fehlermeldung.

**/T032/ Unerwarteter Shutdown**

Einer der Host-Computer wird heruntergefahren und/oder vom Strom getrennt. Der verbleibende Client registriert einen Verbindungsverlust.

**/T040/ Systemübergreifende Funktionalität**

Die Clients werden unter verschiedenen Betriebssystemen ausgeführt.

**/T041/ Mehrere Verbindungen**

Eine Reihe von Clients baut zufällige Verbindungen untereinander auf, sodass mehrere Clients mehr als eine Verbindung halten.

**/T042/ Verbindungsgrenzen**

Es wird versuch, eine bereits bestehende Verbindung erneut aufzubauen. Das bestehende Fenster rückt in den Vordergrund.

# 9 Entwicklungsumgebung

## 9.1 Software

Es werden die Eclipse IDE für Java, sowie verschiedene Texteditoren verwendet. Weiterhin kommen Git und ein Repositorium auf GitHub.com zum Einsatz. Die Entwicklung findet unter GNU/Linux und Windows statt.

## 9.2 Hardware

Es werden PCs in den Computer-Pools des b.i.b international College und Privatrechner verwendet.

## 9.3 Orgware

Keine.

# 10 Glossar

|  |  |
| --- | --- |
| Applikation | kurz: App; Synonym für Anwendung, Programm auf dem Computer oder aber auf einem Mobilfunkgerät |
| Autorun-Funktion | Programmstart direkt bei Einführen des Speichermediums |
| Chat | Dienst, zum Austausch von Kurznachrichten |
| Client | (dt. „Kunde“) hier: der Endnutzer mit seinem Rechner |
| Datei-Manager | Bsp. „Windows-Explorer“, stellt Verzeichnisse und Dateien auf verschiedenen Speichermedien dar |
| GUI | Graphical User Interface – Die graphische Benutzeroberfläche zur Steuerung des Programms. |
| Hardware | physikalische, greifbare Komponenten eines Rechners |
| Host | (dt. „Wirt“, „Gastgeber“, „Veranstalter“) hier ist das System des Anwenders gemeint |
| IP-Adresse | Adresse, die Geräten, die sich in einem Netzwerk befinden, zugewiesen wird |
| Netzwerk | Verbindung von mehreren Computern |
| NO-GUI | siehe GUI; Eine Programmvariante oder -option ohne GUI. Die Darstellung erfolgt innerhalb einer Konsole und ausschließlich mit Unicode-Zeichen. |
| Peer-to-Peer | (dt. „Rechner-Rechner“) bezeichnet die Verbindung in einem Rechnernetz zwecks Kommunikation |
| Software | Anwendung, die auf dem Rechner ausgeführt werden kann; nicht greifbar |
| System | aus mehreren Komponenten bestehend, hier: Computer-System |
| Tool | kleines, schmales Werkzeug zur Lösung einfacher Probleme; nur notwendige Funktionen enthalten |
| Unicast | Nachrichtenübertragung mit einem Sender und Empfänger |
| Use-Case-Diagramm | stellt Strukturen, Verhalten, Beziehungen und Abhängigkeiten von Anwendungen dar |