Berufsakademie Sachsen   
Staatliche Studienakademie Dresden   
Studiengang Medieninformatik

**Programmieraufgabe  
Verteilte Zwischenablage - Clipboarder**

Praxisbeleg für das Moduls 3MI-UXUVS-50

eingereicht von:

Heik, David, 3002549, 3MI15

Weber, Philipp, 3002547, 3MI15

Tag der Einreichung: 23. Oktober 2017

**Inhaltsverzeichnis**

Inhaltsverzeichnis \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_II

1 Motivation 1

2 Zielstellung 1

3 Vorüberlegungen 2

3.1 Zusammenwirken von Server und Client 2

3.2 Datenbankstruktur 3

4 Einrichtung 4

4.1 Server 4

4.2 Client 5

5 Funktionen 5

5.1 Registrierung 5

5.1.1 Anwendersicht 5

5.1.2 Technisch 6

5.2 Aktivierung 6

5.2.1 Anwendersicht 6

5.2.2 Technisch 7

5.3 Login 7

5.4 Eintrag hinzufügen 8

5.5 Eintrag benutzen 8

5.6 Eintrag entfernen 8

5.7 Einstellungen 9

5.7.1 Systemkonfiguration 9

5.7.2 Sitzung beenden (Ausloggen) 10

5.8 Passwort vergessen 10

6 Erweiterung und geplante Erweiterung 10

6.1 Einführung einer Blacklist 10

6.2 Änderung des Verschlüsselungs-Schlüssels 11

6.3 Löschen des Benutzerkontos 11

7 Anhang A

7.1 Datenbankstruktur (SQL) A

7.2 Screenshots B

# Motivation

Während einen Zusatz Excel Kurses an der BA-Dresden hat der Dozent eine Aufgabe gestellt, dessen Lösung eine lange verschachtelte Formel war. Beim Vergleichen und anschließendem abschreiben vom Beamer sind einige meiner Kommilitonen fast verzweifelt. Dies brachte uns zu der Idee eine verteilte Zwischenablage zu benutzten. So dass der Dozent die Formel in seine Zwischenablage kopiert und diese im Anschluss mit uns teilt. Nach einer Recherche im Internet stellten wir fest, dass es etwas Derartiges als Programm noch nicht auf dem Markt gibt. Grund für uns selbst ein Tool zu entwickeln was diese Aufgabe bewerkstelligt.

# Zielstellung

Ziel der Software ist unter dem Betriebssystem *Windows7* und *Windows10*, in einer privaten Umgebung (Passwort geschützt) die Zwischenablage zu archivieren. Diese Einträge sollen auch nach dem Ausschalten des Computers bzw. beim Wechsel des Computers (z.B.: vom Büro zum Home-Office) verfügbar sein. Es soll die Möglichkeit bestehen sich über die Desktopanwendung zu registrieren, sodass mehrere Benutzer gleichzeitig, aber voneinander unabhängig eine eigene *Clipboarder*-Liste führen können. Es soll auch möglich sein, einen Account simultan von mehreren Computern aus zu nutzen (Beispiel: Excelkurs). Durch das drücken der Tastenkombination *STRG + C* werden unter Windows Standardmäßig die aktuell markierten Textstellen in die Lokale Zwischenablage gespeichert. Durch das Programm Clipboarder soll diese Tastenkombination ebenfalls erkannt werden. Dabei soll der aktuelle Wert aus der lokalen Zwischenablage als neuer Eintrag in der *Clipboarder*-Liste erscheinen. Diese *Clipboarder*-Liste wächst mit jedem Eintrag und soll mit einem Klick auf ein Element der *Clipboarder*-Liste wieder in die lokale Zwischenablage geschrieben werden, so dass sie mit *STRG + V* wieder eingefügt werden kann. Es soll auch möglich sein, nicht gewünschte Einträge dauerhaft aus der *Clipboarder*-Liste zu entfernen.

# Vorüberlegungen

## Zusammenwirken von Server und Client

Um der selbst gestellten Zielstellung gerecht zu werden sollte der Datenbestand zum einen zeitlich und zum anderen auch örtlich (potentiell auf jedem *Windows*-Computer mit Internetverbindung) stets verfügbar sein. Deswegen muss dieses Projekt als eine Client-/Serverlösung umgesetzt werden. So kann die Datenhaltung zentral gehalten werden. In unserem Fall haben wir uns für ein Hosting bei Strato entschieden, weil dieser Server bereits vor dem Projekt zur Verfügung stand, keine zusätzlichen Kosten verursacht und bereits eingerichtet war, so dass wir ohne großen Zeitaufwand mit dem Projekt beginnen konnten.

Bei Strato wird standardmäßig bei jedem Hosting Paket eine *MySQL* Datenbank bereitgestellt. Es ist aber auch möglich ein eigenes Datenbank System aufsetzten wie Beispielsweise *NoSQL* oder *MariaDB*. Da wir aber bereits während der Praxisphase bei unseren Praxispartnern gute Erfahrungen mit *MySQL* gemacht haben und unsere Kenntnisse zu dieser Technologie weiter vertiefen wollten, haben wir uns für dieses System entschieden und dabei die standardmäßige Installation von Strato verwendet.

Es wäre prinzipiell möglich die *MySQL* Verbindung aus der Client-Anwendung heraus aufzubauen. Aber bei dieser Implementierung müsste man die *MySQL* Verbindungsdaten (Server, User, Password, Datenbank) der Anwendung mitgeben. Man könnte sie entweder statisch programmieren, in einer Konfigurationsdatei speichern oder auch beim Programmstart aus einer Onlinequelle beziehen. Das Problem bei der ersten Variante ist, dass man nach Fertigstellung der Anwendung keine Änderungen mehr an den Zugangsdaten vorgenommen werden kann. Man müsste eine neue Version mit den aktualisierten Zugangsdaten erstellen. Bei Variante zwei und drei umgeht man dieses Problem, die Zugangsdaten wären hier immer aktuell. Aber hierbei müssten die Zugangsdaten im Klartext oder verschlüsselt Hinterlegt bzw. übertragen werden. Da jede *Java*-Anwendung auch wieder dekompiliert werden kann, könnte auch eine Verschlüsselte Konfigurationsdatei oder einer verschlüsselten Übermittlung aus einer Onlinequelle wieder von einem potentiellen Angreifer Entschlüsselt werden. Da der Schlüssel durch die dekompilierung wieder im Klartext auftaucht.

Um diese Probleme zu umgehen haben wir uns für eine andere Kommunikationsmethode mit der Datenbank entschieden und zwar über einen *Personal Home Page*-Schnittstelle (*php*). Der Verbindungsaufbau zur Datenbank wird über das *php*-Skript aufgebaut. Die Variablen und hinterlegten Passwörter stehen zwar im Klartext im Quellcode, aber sind von außen nicht einsehbar.

Der Client sendet aus der *Java*-Anwendung heraus einen *Hypertext Transfer Protoco-Request* (*http*) an den Host. Für die Aktivierungs-Funktion verwendet wir die *GET*-Methode. Diese ist zwar in der Größe für die Parameter begrenzt (maximal 8192 Bytes). Diese Größe ist für den Zweck der Aktivierung, wo nur der Token und die E-Mail benötigt wird vollkommend ausreichend. Des Weiteren können bei dieser Methode die Parameter in der *Uniform Resource Locator* (*URL*) mitgegeben werden, was es auch ermöglicht in der Aktivierungs-E-Mail einen klickbaren Link einzupflegen. So kann die Aktivierung mit einem Klick ausgeführt werden.

Für die restlichen *Request* nutzen wir die *POST*-Method. Da hier die Parameter im Body mitgeschickt werden. Über diese Methode können die mitgeschickten Parameter in abhängig von der Einstellungsgröße der *php.ini*-Datei nahezu beliebig groß gewählt werden. Somit stellen wir sicher, dass auch Texte die größer sind als 8192 Bytes gespeichert werden.

## Datenbankstruktur

Um Redundanzen zu entfernen haben wir die Datenbankstruktur auf drei Tabellen aufgeteilt. Die erste Tabelle *clipboarderuser* enthält alle 1 zu 1 Beziehungen. Jeder Benutzer hat eine eindeutige ID, diese ist eine einzigartig fortlaufende Ganzzahlige Zahl (*Integer*) mit einem Inkrement. Die ID bildet zugleich den Primärschlüssel dieser Tabelle. Die *EMail*-Spalte vom Typ *varchar* ist ebenfalls einzigartig, da jede Adresse nur einmal in unserem System verwendet werden kann. Den „Username“ vom Typ *text* verwenden wir um den Benutzer in den E-Mails persönlich ansprechen zu können. Zu Sicherung des Accounts verwendet der Benutzer ein Passwort, dieses wird gehast in der Spalte „Password“ hinterlegt und ist ebenfalls vom Typ *text*. Die Spalten *Registerdate* und *Activatedate* sind vom Typ *int*, wobei das Registrierungsdatum gesetzt sein muss (*NOT NULL*), da beim Erstellen des Datensatzes (während der Registrierung) das Datum vorhanden ist zwingend eingetragen werden muss. Das Aktivierungsdatum wird während der Aktivierung aktualisiert und nachgetragen. Ähnlich ist es auch mit den Spalten *Activatetoken* und *PasswordReset*, beide sind vom Typ *int* und speichern jeweils einen sechsstelligen Token, der vom *php*-Skript generiert wird. Der Aktivierungstoken wird ebenfalls währende der Registrierung erzeugt und muss ebenso zwingen gesetzt sein (*NOT NULL*). Der Token zum zurücksetzten des Passwortes wird in *PasswordReset* gespeichert. Die Funktion zum zurücksetzten des Passwortes, wird möglicherweise nie vom Benutzer verwendet, so wird auch nie ein Token generiert, deswegen benötigt diese Spalte nicht den Zusatz, dass sie niemals leer sein darf (*NOT NULL*).

Die Tabelle *clipboarderclipboards* enthält die vom Benutzer hinzugefügten Zwischenablagen. Diese stehen in einer 1 zu n Beziehung zur Tabelle *clipboarderuser*. Jeder Benutzer kann keine, eine oder mehrere Zwischenablagen speichern. Dabei dient die Spalte *UserID* als Fremdschlüssel und referenziert auf die Tabelle *clipboarderuser*. Um einen Datensatz (*Tupel*) gezielt anzusprechen zu können, vergeben wir auch hier einen Primärschlüssel der einzigartig ist. Hierfür haben wir wieder eine Spalte mit der Bezeichnung *ID* und den Typ *int* implementiert, die auch über ein Inkrement verfügt und automatisch bei einem *Insert* hochzählt. In der Spalte *Content* speichern wir den eigentlichen Inhalt, da dieser aus verschieden Zeichen bestehen kann speichern wir ihn als Typ *text* ab. Die letzte Spalte dieser Tabelle *CreateDate* enthält den Zeitstempel der Erstellung und hat deswegen den Typ *int*. Alle Spalten müssen beim Erstellen eines Tupels gesetzt sein, deswegen hat jede Spalte die Bedingung *NOT NULL*.

In der letzten Tabelle *clipboarderlogin* werden alle Zugangstokens, die die Verifizierung via Passwort ersetzten hinterlegt. Es möglich, dass ein Benutzer mehrere Tokens besitzt, daraus ergibt sich eine 1 zu n Beziehung. Die Spalte *UserID* dient als Fremdschlüssel und referenziert auf die Tabelle *clipboarderuser*. Um einen Datensatz, ein Tupel gezielt anzusprechen zu können, vergeben wir auch hier einen Primärschlüssel der einzigartig ist. Hierfür haben wir ebenfalls eine Spalte mit der Bezeichnung *ID* und den Typ *int* implementiert, die auch über ein Inkrement verfügt und automatisch bei einem *Insert* hochzählt. In der Spalte *Token* speichern wir den vom *php*-Skript generierten Token ab, der Typ ist *int*. Die letzte Spalte dieser Tabelle *CreateDate* enthält den Zeitstempel der Erstellung und hat deswegen den Typ *int*. Alle Spalten müssen beim Erstellen eines Datensatzes gesetzt sein, deswegen hat jede Spalte die Bedingung *NOT NULL*.

# Einrichtung

## Server

Die Voraussetzungen sind relativ niedrig. Auf dem Server sollte *PHP5.6* oder höher, sowie *MySQL5.5* oder höher installiert sein. (Zum lokalen Testen empfehlen wir den *XAMPP* mit *PHP5.6* direkt vom Anbieter: *https://www.apachefriends.org/de/download.html* herunterzuladen, zu installieren und anschließend Apache und *MySQL* zu starten) Richten Sie zunächst die Tabellen ein. Gehen Sie dazu über *phpMyAdmin* in die Datenbankverwaltung von *MySQL*. (Lokal unter: *http://localhost/phpmyadmin/* zu finden) Wählen Sie eine existierende Datenbank aus oder erstellen Sie gegebenenfalls eine neue. Erstellen Sie als nächstes die Tabellen, in dem Sie auf den Reiter *SQL* klicken und den im Anhang befindlichen *SQL*-Code in das Textfeld kopieren und den Code ausführen. Danach ist die Datenbankstruktur eingerichtet. Importieren Sie nun alle *php*-Skripte, die sie auf der Abgegeben CD im Ordner *Server\_PHP* finden, in ein Verzeichnis oberhalb des *root*-Verzeichnisses. (Lokal zum Beispiel z.B.: *C:\xampp\htdocs\Clipboarder*) Im Anschluss daran müssen sie die Datei *getClipboarderConstant.inc.php* bearbeiten. Tragen Sie hier sowohl den Server (Lokal: *localhost*), den Benutzer und dessen Passwort ein. (Lokal hat *root*, ohne Passwort standardmäßig alle notwenigen Rechte) Bei *$sMySQLDBName* tragen Sie ihre ausgewählte Datenbank ein und speichern Sie die Datei ab. Nun ist das *php*-Skript in der Lage mit der Datenbank zu kommunizieren. Testen Sie die Verbindung zum Beispiel mit *http://localhost/ Clipboarder/activate.php*. Wenn kein *mysqli*-Warnung oder Fehlermeldung kommt war die Konfiguration erfolgreich. (Wenn bei Ihnen *Missing parameters* auftaucht ist der Test ebenfalls erfolgreich, da das Skript diese Meldung ausgibt, wenn der Parameter *email* oder *token* nicht gesetzt ist).

## Client

Zum Ausführen de auf *Java* basierenden Desktopanwendung ist eine Internetverbindung zwingend erforderlich. Sie benötigen zum Ausführen die aktuelle Version von der *Java Runtime Environment* (*JRE*) beziehbar von: *https://java.com/de/download/* oder das *Java Development Kit* (*JDK*) zum Entwickelten von Anwendungen beziehbar von: *http://www.oracle.com/technetwork/java/javase/downloads/index.html*.

Haben Sie entweder die *JRE* oder die *JDK* installiert, dann öffnen Sie anschließend den auf der CD befindlichen Ordner *Client\_Ausfuehrbar*. In ihm befindet sich eine ausführbare *start.bat* Datei welche Sie starten müssen. Eine Konsole öffnet sich, Sie startet die eigentliche *.jar* Datei und muss bis zur Beendigung des Programmes ausgeführt werden. Sie können die Konsole jedoch minimieren. Die Desktopanwendung ist nun einsatzbereit.

# Funktionen

## Registrierung

### Anwendersicht

Um den Service *Clipboarder* nutzen zu können, benötigt der Nutzer ein Benutzerkonto.

Sofern der Nutzer kein Zugriff auf ein bereits existierendes Benutzerkonto hat, muss er sich innerhalb der *Java*-Anwendung Registrieren.

Um zu dem Registrierungsformular zu gelangen müssen Sie im *Account*-Menü auf den Button *Register* klicken. Das Registrierungsformular öffnet sich. Tragen Sie hier Ihren Vornamen und Ihre E-Mail-Adresse ein. Vergeben Sie außerdem ein sicheres Passwort um Ihren Account zu schützen. Mit dem Klick auf *Register* schließen Sie den Vorgang ab.

Eine E-Mail mit dem Aktivierungs-Token wird an ihr hinterlegtes Postfach gesendet und eine Bestätigung mit der Meldung *Check your mails* taucht auf. Bestätigen Sie diese Meldung und fahren Sie mit dem Gliederungspunkt *Aktivierung* fort.

### Technisch

Durch den Klick auf *Register* wird ein *http-POST-Request* zusammengebaut.  
Dabei wird aus der Konfigurationsdatei (*config.properties*) der Hauptpfad (*globalurl*) geladen, in dem sich die *php*-Skripte befinden. Wir liefern es mit der der Voreinstellung aus, so dass der Pfad zu unserem Server von Strato zeigt. Wenn Sie einen lokalen Test ausführen wollen, ändern sie den Wert der *globalurl* zum Beispiel auf *http://localhost/ Clipboarder/*. Bei der Registrierungsfunktion wird dieser Pfad um die eigentliche Datei *register.inc.php* erweitert. Um die Informationen, die der Nutzer in das Formular der Desktopanwendung eingetragen an das *php*-Skript zu übermitteln zu können, müssen sie vor dem Absenden des Request eingebunden werden. Anschließend müssen die Informationen zu *UTF8* (*8-Bit UCS Transformation Format*) kodiert werden, da auch das Encoding auf dem Server sowie der Datenbank diesen Zeichensatz verwendet. Ist dies geschehen erfolgt der eigentliche verbindungsaufbau.

Der Server nimmt Ihre Anfrage entgegen und überprüft zunächst ob die Parameter *password*, *username* und *email* übermittelt wurden sind. Ist das der Fall, erfolgt im zweiten Schritt eine Überprüfung ob Ihre angegebene E-Mail-Adresse bereits in dem System existiert. Wenn dies nicht der Fall ist, wird ein Datensatz mit folgenden Inhalt erstellt.

ID Fortlaufende Nummer  
E-Mail Ihre eingetragene E-Mail -Adresse  
Username Ihr eingetragener Vorname  
Password Ihr eingetragenes Passwort wird mit dem  
 *CRYPT-BLOWFISH*-Algorithmus zu einem 60 Zeichen  
 langen String gehasht  
Registerdate Aktuelle Serverzeit in Sekunden seit dem 1.1.1970 um 01:00 Uhr Activatetoken Eine zufällige sechsstellige Zahl

Konnte der Datensatz erstellt werden wird an die angegebene E-Mail-Adresse eine E-Mail gesendet, die den Aktivierungstoken, den Sie benötigen um sich zu verifizieren enthält.

## Aktivierung

### Anwendersicht

Die zu aktivierende E-Mail-Adresse wird aus dem Registrierungsformular automatisch an das Aktivierungsformular innerhalb der Desktopanwendung übergeben, Sie müssen sie nicht erneut eintragen. Überprüfen Sie ihr E-Mail-Postfach. In der E-Mail mit dem Betreff *Deine Clipboarder-Aktivierung* finden Sie den Aktivierungstoken. Tragen Sie diesen sechsstelligen Schlüssel in die Desktopanwendung in das Feld *Token* ein. Um den Vorgang abzuschließen klicken sie Im Anschluss auf *Activate*. Der von Ihnen Token eingetragene Token wird nun überprüft. Stimmt er mit dem von uns erstellen Token überein, wird ihr Account aktiviert. Eine Bestätigung mit der Meldung *Succsessfully activated* taucht auf. Ihre E-Mail-Adresse ist nun bei uns verifiziert, Sie erhalten nun eine E-Mail mit der wir Sie herzlich bei unserem Service Begrüßen. Bestätigen Sie die Meldung in der Desktopanwendung und fahren Sie mit dem Gliederungspunkt *Login* fort.

### Technisch

Durch den Klick auf *Activate* wird ein *http-GET*-Request zusammengebaut.

Dabei wird wieder aus der Konfigurationsdatei (*config.properties*) der Hauptpfad (*globalurl*) geladen, in dem sich die *php*-Skripte befinden. Bei der Aktivierungsfunktion wird dieser Pfad um die eigentliche Datei *activate.php* erweitert. Um die Informationen, aus der Desktopanwendung an das *php*-Skript zu übertragen, müssen bei dieser Request-Methode die Parameter an die *URL* angehangen werden. Ist dies geschehen erfolgt der eigentliche verbindungsaufbau.

Der Server nimmt Ihre Anfrage entgegen und überprüft zunächst ob die *GET*-Parameter *email* und *token* übermittelt wurden sind. Ist das der Fall, erfolgt im zweiten Schritt eine Überprüfung ob Ihre angegebene E-Mail-Adresse bereits in dem System existiert und ob dabei das Aktivierungsdatum noch offen ist. Wenn dies der Fall ist, wird der Datensatz aktualisiert und der aktuelle Zeitstempel des Servers in die Spalte *Activatedate* eingetragen. Zeitgleich wird aus der Spalte *Activatetoken* der Token entfernt, da der Token nun verwendet wurde.

## Login

Direkt beim Starten der *Java*-Anwendung öffnet sich das *Account*-Menü, sofern Sie sich nicht bereits zu einem früheren Zeitpunkt in der Anwendung angemeldet und die Option *Remember Me* angehakt haben.

Tragen Sie ihre E-Mail-Adresse und ihr Passwort, was Sie bei der Registrierung vergeben haben ein. Wenn Sie sich nicht bei jeder Sitzung neu einloggen möchten, empfehlen wir Ihnen die Option *Remember Me* anzuhaken.

Mit dem Klick auf *Login* senden Sie eine Anfrage ihr Passwort Abzugleichen. Stimmt es mit dem von Ihnen hinterlegten Passwort überein werden Sie in die Anwendung eingeloggt. Sie werden im Anschluss in das *Clip*-Menü geleitet. Dort sehen Sie eine Liste mit letzten fünfzig Einträgen. Sollten Sie sich zum ersten Mal in Ihren Account einloggen und/oder noch keinen Eintrag hochgeladen haben sehen Sie zunächst einen Initial-Eintrag.

Sollte Ihr Passwort nicht übereinstimmen, erscheint die Information „Wrong password“ welche Sie mit „OK“ bestätigen können. Sie werden im Anschluss wieder zu dem Authentifizierungs-Bildschirm weitergeleitet.

Sollten Sie sich für die Option *Remember Me* entschieden haben wird währende des Login-Prozesses für Sie ein Token generiert. Dieser Token wird in der *config.properties* unter dem Parameter *token* gespeichert und sorgt dafür, dass sie beim erneuten Starten der Desktopanwendung direkt in Ihren Account eingeloggt werden. Der Token bleibt nach dem letzten Benutzungszeitpunkt der Desktopanwendung noch drei Tage gültig. Nach dem Ablauf der Frist wird der Token vom Server entfernt und Sie müssen sich über Ihre E-Mail und Passwort authentifizieren.

## Eintrag hinzufügen

Mit dem Shortcut *STRG + C* wird unter Windows standardmäßig maskierter Text in die lokale Zwischenablage gespeichert. Diese Funktion machen wir uns zu nutze. Wir triggern den selben Shortcut. Um einen Eintrag hinzufügen zu können, müssen die beiden Bedingungen erfüllt sein.

- erfolgreiche Authentisierung (entweder mit Passwort oder gültigen Token)

- im Einstellungsmenü muss die Option *Enable Recording* eingeschaltet sein

Ist dies der Fall so wird der *KeyListener* für diesen Shortcut aktiviert. Trifft das Ereignis ein, wird der Inhalt Ihrer Zwischenablage zunächst mit der Zeichenkette, die sich in der Konfigurationsdatei unter *cryptKey* befindet verschlüsselt. Im Anschluss wird eine Verbindung zum Server aufgebaut. Dort wird Anhand der übermittelten Informationen ein neuer Datensatz in die Datenbank geschrieben. Die Übermittelte Information Ihrer Zwischenablage ist aufgrund der verschlüsselten *Hypertext Transfer Protocol Secure*-Verbindung (*https* entspricht *http* jedoch mit der zusätzlichen Verwendung eines *Secure Sockets Layer*-Handshake-Protokolls), gegen das mögliche Mitschneiden von dritte gesichert. Durch die Verschlüsselung des Inhaltes der Zwischenablage innerhalb der Desktopanwendung stellen wir außerdem sicher, dass auch nach der Übertragung kein Unberechtigter an Ihre gespeicherten Informationen kommt. Dies könnte zum Beispiel durch den Administrator der Datenbank, bzw. ein Angreifer der einen Datenbankdump gewonnen hat der Fall sein. Die Desktopanwendung springt nach dem Hochladen zurück in das *Clips*-Menü und aktualisiert die Einträge.

## Eintrag benutzen

Durch das klicken auf einen Eintrag der Liste im *Clips*-Menu wird ihre Lokale Zwischenablage von Windows mit dem Wert, der sich in hinter dem Listeintrag verbirgt überschrieben. Es werden immer nur die ersten zwölf Zeichen eines jeden Eintrag angezeigt. Zeilenumbrüche werden auf den Listeinträgen ebenfalls entfernt. In der Zwischenablage befindet sich nun der *Plain*-Text, so wie sie ihn kopiert haben. In voller Länge und mit Zeilenumbrüchen. Mit dem Shortcut *STRG + V*  können Sie nun diesen in der Zwischenablage befindlichen Text wieder einfügen und damit nutzen.

## Eintrag entfernen

Im *Clip*-Menü werden standardmäßig die letzten 50 Einträge dargestellt. Die Anzahl der dazustellenden Elemente können Sie in der *config.properties* unter dem Parameter *number* einstellen. Passen nicht alle Elemente auf einer Seite, werden entsprechend zusätzliche Seiten angelegt. Möchten Sie einen Eintrag dauerhaft entfernen, so klicken Sie auf das neben dem Eintrag stehende Kästchen. Es starten ein Verbindungsaufbau zum Server und der zum Eintrag zugehörige Datensatz wird aus der Datenbank gelöscht. Im Anschluss daran startet ein erneuter Verbindungsaufbau, der dafür sorgt, dass die Liste mit den letzten Einträgen aktualisiert wird. Die vom Server gesendeten Daten liegen in diesem Fall im *JavaScript Object Notation*-Format (*JSON*) vor. Der Inhalt Ihrer Zwischenablagen wird dabei verschlüsselt übertragen. Mit dem in der Konfigurationsdatei hinterlegten Parameter cryptKey werden diese Inhalte entschlüsselt und die Darstellung in der Desktopanwendung aktualisiert. Sollten vor dem Löschvorgang mehr als die beispielsweise 50 Einträge in Ihrem Benutzerkonto existieren, so rutscht der nächst jüngste Eintrag hinterher und es werden wieder 50 Einträge dargestellt.

## Einstellungen

### Systemkonfiguration

In dem Menüpunkt *Settings* finden sie die Einstellungsmöglichkeiten die der Anwender problemlos vornehmen kann, ohne dass es negative Auswirkungen auf die Funktionalität der Desktopanwendung gibt. Der wichtigste Einstellungspunkt kommt direkt als erstes. Hier kann der Benutzer die Option *Enable Recording* setzten und somit kontrollieren ob das Programm beim drücken des *STRG +C* Shortcuts den Inhalt der Zwischenablage in die Datenbank schreiben soll. Somit geben wir den Benutzer die Möglichkeit einen *nur-Lese*-Modus zu verwenden. Im nächstens Abschnitt kann der Benutzer die Desktopanwendung individuell gestalten. Es werden fünf Designs mitgeliefert. Darunter ein eher helleres *Office* Theme und vier bunte, eher dunkel und modern gestaltete Themes. Dieses kann über das Dropdown-Menü ausgewählt werden. Nach dem Sie eine Änderung innerhalb der Auswahlbox getroffen haben, wird der Inhalt des *Cascading Style Sheets* (*CSS*) geladen und die Desktopanwendung neu gezeichnet. Sodass das Theme direkt ohne Neustart angewendet wird. Zeitgleich wird in der Konfigurationsdatei automatisch Parameter „style“ mit dem Dateinamen der ausgewählten *CSS*-Datei überschrieben. Der erfahrene Benutzer kann sich anhand der mitgelieferten *CSS*-Dateien auch ein eigenes Theme erstellen und dieses natürlich auch verwenden. Es ist dabei egal wo auf dem Dateisystem die *CSS*-Datei hinterlegt ist, denn der Pfad kann über den *Choose*-Button ausgewählt werden. Nach dem klicken des *Choose*-Buttons öffnet sich ein Dialog-Fenster in dem der Ordner, der die gewünschte *CSS*-Datei beinhaltet ausgewählt werden kann. Nach dem Sie den ausgewählten Ordner in dem Dialog-Fenster bestätigt haben wird automatisch der ausgewählte Pfad als neuer Wert für den Parameter *stylePath* in die Konfigurationsdatei geschrieben. Zeitgleich werden alle in dem Ordner enthaltenen *CSS*-Dateien in dem Dropdown-Menü aufgelistet. Hier können Sie wieder eine Datei auswählen. Das Theme wird wieder geändert wie oben bereits beschrieben.

Im dritten Bereich kann die Auflösung des Fensters eingestellt werden die erste Box ist für die Breite, der zweite für die Höhe der Desktopanwendung. Dabei kann der Wert nicht beliebig klein gewählt werden, da sichergestellt werden muss, dass alle Elemente dargestellt werden können. Die Maximale Größe ist auf die Auflösung des Bildschirmes minus den in der Konfigurationsdatei vergebenen Offset beschränkt. Zum Ändern der Parameter klicken Sie auf eines der Textfelder und tragen sie einen ganzzahligen wert ein. Zum Übernehmen klicken sie auf ein beliebiges anderes Element (auch außerhalb der Desktopanwendung) oder bestätigen Sie die Eingabe mit den Pfeiltasten Ihrer Tastatur. Der neue Wert wird sofort übernommen und das Fenster entsprechend angepasst, zweitgleich wird der neue Wert in der Konfigurationsdatei gespeichert.

### Sitzung beenden (Ausloggen)

Zum Beenden der Sitzung klicken Sie auf den Button *Exit* im *Settings*-Menü. Die Konsole und die Desktopanwendung werden geschlossen. Sie sind nun ausgeloggt, sofern Sie nicht die Option *Remember Me* im *Account*-Menü gesetzt haben. Haben sie diese Option gesetzt finden Sie ebenfalls im *Settings*-Menü direkt unter dem Button *Exit* den Button *Forget Me*. Um auch den generierten Token und die E-Mail aus der Konfigurationsdatei zu entfernen, klicken sie auf den *Forget Me*-Button. Die Parameter *mail* und *token* werden nun innerhalb der Konfigurationsdatei zurückgesetzt. Der Token wird nach drei Tagen vom Server entfernt. Nach einem Neustart der Anwendung können Sie sich wieder mit Ihrer E-Mail und dem dazugehörigen Passwort authentisieren.

## Passwort vergessen

Sollten Sie einmal Ihr Passwort vergessen haben, können Sie es in wenigen Schritten zurücksetzten. Gehen Sie zunächst innerhalb der Desktopanwendung auf das *Account*-Menü und klicken Sie auf den ganz unten befindlichen Button *Forget Password*. Das Formular *Reset your password* erscheint. Tragen Sie in das Feld *E-Mail* Ihre E-Mail-Adresse ein und bestätigen Sie ihre Eingabe mit einen klicke auf *Confirm*. Wurde die E-Mail in unserem System gefunden und bereits aktiviert, so erhalten die die Mitteilung *Successfully request, check your mails to change* Bestätigen Sie diese Meldung anschließen mit *OK*. Zeitgleich erhalten Sie eine E-Mail mit dem Betreff *Clipboarder Passwort vergessen*. In der Desktopanwendung öffnet sich das Formular *Setup new password*. Öffnen Sie die E-Mail und füllen Sie das Formular aus. Tragen Sie wieder Ihre E-Mail-Adresse, den Token aus der E-Mail und Ihr gewünschtes neues Passwort ein. War der Token gültig, so erscheint nun die Meldung *Password changed*. Ihr Passwort wurde erfolgreich geändert. Sie erhalten eine E-Mail mit einer Bestätigung über den Passwortwechsel. Sie können sich nun mit Ihrem neu vergebene Passwort im *Account*-Menü einloggen.

# Erweiterung und geplante Erweiterung

## Einführung einer Blacklist

Dafür ist zunächst eine Erweiterung der Datenbankstruktur notwendig. Eine neue Tabelle mit dem Namen *clipboarderBlackList* und den Spalten *ID* und *Email* notwendig. Die vor jeder Aktion die eine E-Mail sendet muss nun in den *php*-Skripten überprüft werden ob der Empfänger mit dieser E-Mail in der *BlackList* steht. Ist dies der Fall, so muss die Aktion Übersprungen werden. Des Weiteren müssen alle E-Mail die unser System bisher versendet überarbeitet werden. In jede E-Mail-Vorlage muss einen Link eingebaut werden, der beim Aufruf dafür sorgt, dass die E-Mail-Adresse des Empfängers in die Liste aufgenommen wird.   
Zeitaufwandsschätzung: 2,5 Stunden

## Änderung des Verschlüsselungs-Schlüssels

Bisher ist es nicht möglich den *cryptKey* in der Konfigurationsdatei zu ändern, wenn bereits Zwischenablagen mit dem Konto verknüpft sind. Es müssen erst alle Zwischenablagen (zurzeit noch jede einzeln) gelöscht werden. So das mit dem Benutzerkonto keine Zwischenablage mehr Verbunden ist (entspricht dem Zustand direkt nach der Aktivierung). Besser wäre es, wenn wir die Änderung des *cryptKey* über die Desktopanwendung anbieten würden. Dafür müsste ein neues Formular erstellt werden, in das der neue *cryptKey* eingetragen werden kann. Im Anschluss müssten zunächst alle bisherigen Zwischenablagen heruntergeladen werden. Diese müssten mit dem alten *cryptKey* entschlüsselt, in einer Liste zwischengespeichert, und im Anschluss mit dem neuen *cryptKey* verschlüsselt werden. Im daran Anschluss müssen Zwischenablagen-Datensätze des Benutzers an den Server übermittelt werden und in der Datenbank schließlich aktualisiert werden. Im Fall eines Verbindungsabbruches oder eines Fehlers in der Datenbank müssten alle Informationen zurückgerollt werden. Das bedeutet nicht nur, dass die Informationen auf der Datenbank rückgängig gemacht werden müssen, sondern auch der alte *cryptKey* wieder hergestellt werden muss, so dass die bisherigen Zwischenablagen weiter entschlüsselt werden können.

Zeitaufwandsschätzung: 8 Stunden

## Löschen des Benutzerkontos

Wenn ein Benutzer sich entschließt sein Benutzerkonto bei uns zu schließen, müssen zeitgleich auch alle mit dem Konto Verbundenen Datensätze wie Token und Zwischenablage entfernt werden. Dies kann momentan nur per Hand vom Datenbankadministrator David Heik erfolgen. Um das für die Benutzer in Zukunft angenehmer zu gestalten, haben wir einen zusätzlichen Button *delete account* im *Setting*-Menü geplant. Geplant ist, falls dieser gedrückt wird, soll sich zunächst eine Meldung öffnen mit der Frage: „*Are you sure you want to delete your account irrevocably?“.* Wird diese mit *yes* Bestätigt so wird der Benutzer in einem neuen Formular nach seinem Kennwort gefragt. Trägt er dieses korrekt ein, wird er ausgeloggt, all Seine Daten werden von der Datenbank entfernt und erhält eine E-Mail mit dem wir die Löschung des Kontos bestätigen.

# Anhang

## Datenbankstruktur (SQL)

--  
-- Tabellenstruktur für Tabelle `clipboarderuser`  
--

CREATE TABLE IF NOT EXISTS `clipboarderuser` (  
`ID` int(11) NOT NULL,  
 `EMail` varchar(255) NOT NULL,  
 `Username` text NOT NULL,  
 `Password` text NOT NULL,  
 `Registerdate` int(11) NOT NULL,  
 `Activatedate` int(11),  
 `Activatetoken` int(6) NOT NULL,  
 `PasswordReset` int(6)  
) ENGINE=InnoDB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8;

--  
-- Indizes für die Tabelle `clipboarderuser`  
--

ALTER TABLE `clipboarderuser`  
ADD PRIMARY KEY (`ID`), ADD UNIQUE KEY `EMail` (`EMail`);

--  
-- AUTO\_INCREMENT für Tabelle `clipboarderuser`  
--

ALTER TABLE `clipboarderuser`  
MODIFY `ID` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=0;  
  
--  
-- Tabellenstruktur für Tabelle `clipboarderclipboards`  
--

CREATE TABLE IF NOT EXISTS `clipboarderclipboards` (  
`ID` int(11) NOT NULL,  
 `UserID` int(11) NOT NULL,  
 `Content` text NOT NULL,  
 `CreateDate` int(11) NOT NULL  
) ENGINE=InnoDB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8;

--  
-- Indizes für die Tabelle `clipboarderclipboards`  
--

ALTER TABLE `clipboarderclipboards`  
 ADD PRIMARY KEY (`ID`);

ALTER TABLE `clipboarderclipboards`  
ADD FOREIGN KEY (`UserID`) REFERENCES `clipboarderuser`(`ID`);

--  
-- AUTO\_INCREMENT für Tabelle `clipboarderclipboards`  
--

ALTER TABLE `clipboarderclipboards`  
MODIFY `ID` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=0;

--  
-- Tabellenstruktur für Tabelle `clipboarderlogin`  
--

CREATE TABLE IF NOT EXISTS `clipboarderlogin` (  
`ID` int(11) NOT NULL,  
 `UserID` int(11) NOT NULL,  
 `Token` text NOT NULL,  
 `CreateDate` int(11) NOT NULL  
) ENGINE=InnoDB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8;

--  
-- Indizes für die Tabelle `clipboarderlogin`  
--

ALTER TABLE `clipboarderlogin`  
ADD PRIMARY KEY (`ID`);

ALTER TABLE `clipboarderlogin`  
ADD FOREIGN KEY (`UserID`) REFERENCES `clipboarderuser`(`ID`);

--  
-- AUTO\_INCREMENT für Tabelle `clipboarderlogin`  
--

ALTER TABLE `clipboarderlogin`  
MODIFY `ID` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=0;

## Screenshots





