Dokumentation

# 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 Windows 7 und höher, 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 Zielstellung gerecht zu werden sollte der der Datenbestand zum einen zeitlich und zum anderen auch örtlich (potentiell auf jedem Windows-Rechner 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 Online-Quelle 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 Online-Quelle 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 PHP-Schnittstelle. 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 HTTP-Request, 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 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 Requests 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-Mail s persönlich anzusprechen. 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 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 (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 MySQL 5.5 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 „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 PHH-Skript in der Lage mit der Datenbank zu kommunizieren. Testen Sie die Verbindung zum Beispiel mit [http://localhost/ Clipboarder/activate.php](http://localhost/%20Clipboarder/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

Das PHP-Skript überprüft bei jeder Antragsstellung ob der Anfragende auch die entsprechende rechte hat.

Dieser empfängt alle HTTP-Request der Clients und ist an die MySQL Datenbank angebunden.

und Synchronisiert werden soll gespeichert werden soll und auch nach einem Wechsel soll an einer zentralen Stelle gespeichert werden. (Auf dem Server).

Die Clients liefern die

## 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 zusammen gebaut.  
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 das 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/](http://localhost/%20Clipboarder/). 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 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 Daten 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 zusammen gebaut.  
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 den „Uniform Resource Locator“ (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 überein werden Sie in die Anwendung eingeloggt. Sie werden 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 „Okay“ bestätigen können. Sie erden 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.

## Eintrag hinzufügen

Mit dem Shortcut „STRG + C“

## Eintrag Entfernen

## Sitzung beenden / Ausloggen

## Passwort Vergessen

## Setting

# Erweiterung und geplante Erweiterung

## Einführen einer Blacklist

Dafür ist eine Erweiterung der Datenbankstruktur notwendig. Eine neue Tabelle mit den Spalten „ID“ und „EMail“. Die ID ist dabei der

# Anhang

## 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;