

Berufsakademie Sachsen
Staatliche Studienakademie Dresden
Studiengang Medieninformatik

Programmieraufgabe
Verteilte Zwischenablage - Clipboarder

Praxisbeleg für das Modul 3MI-UXUVS-50

eingereicht von:
Heik, David, 3002549, 3MI15-1
Weber, Philipp, 3002547, 3MI15-1

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	5
4.1 Server	5
4.2 Client	6
5 Funktionen	7
5.1 Registrierung	7
5.1.1 Registrierung aus Anwendersicht	7
5.1.2 Registrierung aus technischer Sicht	7
5.2 Aktivierung	8
5.2.1 Aktivierung aus Anwendersicht	8
5.2.2 Aktivierung aus technischer Sicht	9
5.3 Login	10
5.4 Eintrag hinzufügen	11
5.5 Eintrag benutzen	11
5.6 Eintrag entfernen	12
5.7 Einstellungen	12
5.7.1 Systemkonfiguration	12
5.7.2 Sitzung beenden (Ausloggen)	14
5.8 Passwort vergessen	14
6 Erweiterung und geplante Erweiterung	15
6.1 Einführung einer Blacklist	15
6.2 Änderung des Verschlüsselungs-Schlüssels	15
6.3 Löschen des Benutzerkontos	16
7 Anhang	A
7.1 Datenbankstruktur (SQL)	A
7.2 Screenshots	B

1 Motivation

Während eines zusätzlichen Excel-Kurses an der Berufsakademie Dresden, hat der Dozent eine Aufgabe gestellt, deren Lösung eine lange verschachtelte Formel war. Beim Vergleichen und anschließendem Abschreiben der Formel von der Projektionsleinwand, sind einige meiner Kommilitonen fast verzweifelt. Dies brachte uns zu der Idee, eine verteilte Zwischenablage zu benutzen. Diese könnte der Dozent nutzen, um die Formel hinein zu kopieren und sie im Anschluss mit uns teilen. Nach einer Recherche im Internet stellten wir fest, dass es etwas Derartiges als Programm noch nicht auf dem Markt gibt. Dies war der Grund, ein solches Tool für uns selbst zu entwickeln, das diese Aufgabe bewerkstelligen kann.

2 Zielstellung

Ziel der Software ist, die Zwischenablage in den Betriebssystemen *Windows 7* bzw. *Windows 10*, in einer privaten Umgebung (passwortgeschützt) zu archivieren. Diese Einträge sollen auch nach dem Ausschalten, 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, so dass mehrere Benutzer gleichzeitig, aber voneinander unabhängig eine eigene *Clipboarder*-Liste führen können. Ebenso soll es möglich sein, einen Account simultan von mehreren Computern aus zu nutzen (Beispiel: Excelkurs). Durch die Tastenkombination *STRG + C* werden unter *Windows* standardmäßig die aktuell markierten Textstellen in die lokale Zwischenablage kopiert. 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 Liste wächst bei jedem Kopiervorgang von Text. Mit einem Klick auf ein Element der *Clipboarder*-Liste soll der jeweilige Eintrag wieder in die lokale Zwischenablage geschrieben werden, so dass dieser mit *STRG + V* wieder eingefügt werden kann. Des Weiteren soll es möglich sein, nicht gewünschte Einträge dauerhaft aus der Liste zu entfernen.

3 Vorüberlegungen

3.1 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 Datenbanksystem aufzusetzen, wie beispielsweise *NoSQL* oder *MariaDB*. Da wir bereits während der Praxisphase Erfahrung mit *MySQL* sammeln konnten und unsere Kenntnisse zu dieser Technologie weiter vertiefen wollten, entschieden wir uns für dieses System und verwendeten dabei die standardmäßige Installation von Strato.

Es wäre prinzipiell möglich, die *MySQL*-Verbindung aus der Client-Anwendung heraus aufzubauen. Bei dieser Implementierung müsste man jedoch die *MySQL*-Verbindungsdaten (Server, User, Password, Datenbank) als Parameter übergeben. 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 an den Zugangsdaten vornehmen kann. Man müsste eine neue Version mit den aktualisierten Zugangsdaten erstellen. Bei Variante zwei und drei umgeht man dieses Problem und die Zugangsdaten wären hier immer aktuell. Hierbei müssten die Zugangsdaten jedoch im Klartext oder verschlüsselt hinterlegt bzw. übertragen werden. Da jede *Java*-Anwendung auch wieder dekompiert werden kann, könnte auch eine verschlüsselte Konfigurationsdatei oder eine verschlüsselte Übermittlung aus einer Onlinequelle wieder von einem potentiellen Angreifer entschlüsselt werden. Der Schlüssel würde durch die Dekompilierung jedoch im Klartext sichtbar sein.

Um diese Probleme zu umgehen, haben wir uns für eine andere Kommunikationsmethode entschieden. Der Verbindungsaufbau zur Datenbank wird über das *php*-Skript (rekursives Akronym: *Hypertext Preprocessor*) aufgebaut. Die Variablen und hinterlegten Passwörter stehen zwar im Klartext im Quellcode, sind jedoch von außen nicht einsehbar.

Der Client sendet aus der *Java*-Anwendung heraus einen *HTTP-Request* (*Hypertext Transfer Protocol*) an den Host. Für die Aktivierungs-Funktion verwenden wir die *GET*-Methode. Diese ist zwar in der Größe für die Parameter begrenzt (maximal 8192 Bytes), jedoch für den Zweck der Aktivierung, bei der lediglich ein Token und die E-Mail-Adresse benötigt wird, vollkommend ausreichend. Des Weiteren können bei dieser Methode Parameter in der *URL* (*Uniform Resource Locator*) 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*-Methode. Da hier die Parameter im Body mitgeschickt werden. Über diese Methode können die mitgeschickten Parameter in Abhängigkeit von der Einstellungsgröße der *php.ini*-Datei nahezu beliebig groß gewählt werden. Somit stellen wir sicher, dass auch Texte die größer als 8192 Bytes sind, gespeichert werden.

3.2 Datenbankstruktur

Um Redundanzen zu vermeiden haben wir die Datenbankstruktur auf drei Tabellen aufgeteilt. Die erste Tabelle *clipboarduser* enthält alle 1-zu-1-Beziehungen. Jeder Benutzer hat eine ID, die eindeutig einem fortlaufenden ganzzahligen Wert (*Integer*) mit einem Inkrement zugeordnet ist. Diese bildet zugleich den Primärschlüssel dieser Tabelle. Die *E-Mail*-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. Zur Sicherung des Accounts verwendet der Benutzer ein Passwort, das gehasht in der Spalte „*Password*“ hinterlegt und ebenfalls vom Typ *text* ist. Die Spalten *Registerdate* und *Activatedate* sind vom Typ *int*. Hierbei muss das Registrierungsdatum gesetzt sein (*NOT NULL*), da beim Erstellen des Datensatzes (während der Registrierung) das Datum vorhanden ist. 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 während der Registrierung erzeugt und muss zwingend gesetzt sein (*NOT NULL*). Der Token zum Zurücksetzen des Passwortes wird in *PasswordReset* gespeichert.

Die Funktion zum Zurücksetzen 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.

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 gibt es eine Spalte mit der Bezeichnung *ID*, welche den Typ *int* besitzt, und über ein Inkrement verfügt, also automatisch bei einem *Insert* hochgezählt wird. In der Spalte *Content* speichern wir den eigentlichen Inhalt. Da dieser aus einer Kombination von Text, Ziffern und Sonderzeichen 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*. Da alle Spalten beim Erstellen eines Tupels gesetzt sein müssen, hat jede von ihnen die Bedingung *NOT NULL*.

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

4 Einrichtung

4.1 Server

Die Voraussetzungen für die Einrichtung eines Servers, auf dem unsere Anwendung laufen kann, sind relativ niedrig. Auf dem Server sollte *PHP 5.6* oder höher, sowie *MySQL 5.5* oder höher installiert sein. Zum lokalen Testen empfehlen wir, *XAMPP* mit *PHP 5.6* direkt vom Anbieter: <https://www.apachefriends.org/de/download.html> herunterzuladen, zu installieren und anschließend Apache sowie *MySQL* zu starten. Richten Sie zunächst die Tabellen ein. Gehen Sie dazu über *phpMyAdmin* in die Datenbankverwaltung von *MySQL*. (Lokal unter: <http://localhost/phpmyadmin/>) Wählen Sie eine existierende Datenbank aus oder erstellen Sie gegebenenfalls eine neue. Erstellen Sie als nächstes die Tabellen, indem Sie auf den Reiter *SQL* klicken, anschließend 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 abgegebenen 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 notwendigen Rechte) Bei *\$sMySQLDBName* tragen Sie ihre ausgewählte Datenbank ein und speichern 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 Sie keine *mysql*-Warnung oder Fehlermeldung erhalten, war die Konfiguration erfolgreich. Falls bei Ihnen die Meldung „*Missing parameters*“ auftaucht, war der Test ebenfalls erfolgreich, da das Skript diese Meldung ausgibt, wenn einer der Parameter *email* oder *token* nicht gesetzt ist.

4.2 Client

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

Haben Sie entweder die *JRE* oder die *JDK* installiert, öffnen Sie anschließend den auf der CD befindlichen Ordner *Client_Ausfuehrbar*. In ihm befindet sich eine Datei namens *start.bat* die Sie ausführen müssen. Eine Konsole öffnet sich, welche die eigentliche *jar*-Datei startet und bis zur Beendigung des Programmes geöffnet bleiben muss. Sie können die Konsole jedoch minimieren. Die Desktopanwendung ist nun einsatzbereit.

5 Funktionen

5.1 Registrierung

5.1.1 Registrierung aus Anwendersicht

Damit man den Service *Clipboarder* nutzen kann, benötigt man ein Benutzerkonto. Sofern der Nutzer keinen 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.

5.1.2 Registrierung aus technischer Sicht

Durch den Klick auf *Register* wird ein *http-POST-Request* aufgebaut. Dabei wird aus der Konfigurationsdatei (*config.properties*) der Hauptpfad (*globalurl*) geladen, in dem sich die *php*-Skripte befinden. Wir liefern es mit 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 Requests eingebunden werden. Anschließend müssen die Informationen in *UTF8 (8-Bit UCS Transformation Format)* kodiert werden, da auch das Encoding auf dem Server, sowie die 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 worden sind. Ist dies der Fall, erfolgt im zweiten Schritt eine Überprüfung, ob Ihre angegebene E-Mail-Adresse bereits im System existiert. Ist dies nicht der Fall, wird ein Datensatz mit folgendem Inhalt erstellt.

ID	Fortlaufende Nummer
E-Mail	Ihre eingetragene E-Mail-Adresse
Username	Ihr eingetragener Vorname
Password	Ihr eingetragenes Passwort wird mit dem <i>CRYPT-BLOWFISH</i> -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 eine E-Mail an die angegebene E-Mail-Adresse gesendet, welche den Aktivierungstoken enthält, den Sie für die Verifizierung benötigen.

5.2 Aktivierung

5.2.1 Aktivierung aus Anwendersicht

Die zu aktivierende E-Mail-Adresse wird aus dem Registrierungsformular automatisch an das Aktivierungsformular innerhalb der Desktopanwendung übergeben. Sie müssen diese 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 *Successfully activated* taucht auf. Ihre E-Mail-Adresse ist nun bei uns verifiziert. Sie erhalten nun eine E-Mail mit der wir Sie recht herzlich bei unserem Service begrüßen. Bestätigen Sie die Meldung in der Desktopanwendung und fahren mit dem Gliederungspunkt *Login* fort.

5.2.2 Aktivierung aus technischer Sicht

Durch den Klick auf *Activate* wird ein *http-GET*-Request aufgebaut. 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* angehängen 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 worden sind. Ist dies der Fall, erfolgt im zweiten Schritt eine Überprüfung, ob Ihre angegebene E-Mail-Adresse bereits im 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 dieser nun verwendet wurde.

5.3 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 ein, das Sie bei der Registrierung vergeben haben. 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 den 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 mit dem in der Datenbank übereinstimmen, erscheint die Meldung „*Wrong password*“, welche Sie mit „*OK*“ bestätigen können. Sie werden im Anschluss wieder zum Authentifizierungs-Bildschirm weitergeleitet.

Sollten Sie sich für die Option *Remember Me* entschieden haben, wird während des Login-Prozesses ein Token für Sie 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 Ihr Passwort erneut authentifizieren.

5.4 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, indem wir den selben Shortcut triggern. Um einen Eintrag hinzufügen zu können, müssen die folgenden Bedingungen erfüllt sein.

- erfolgreiche Authentifizierung (entweder mit Passwort oder gültigem Token)
- im Einstellungsmenü muss die Option *Enable Recording* eingeschaltet sein

Ist dies der Fall, 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 Informationen Ihrer Zwischenablage sind aufgrund der verschlüsselten *https*-Verbindung (*Hypertext Transfer Protocol Secure: http* jedoch mit der zusätzlichen Verwendung eines *Secure Sockets Layer*-Handshake-Protokolls), gegen das mögliche Mitschneiden von Dritten 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. einen Angreifer, der einen Datenbankdump hat, der Fall sein. Die Desktopanwendung springt nach dem Hochladen zurück in das *Clip*-Menü und aktualisiert die Einträge.

5.5 Eintrag benutzen

Durch das Klicken auf einen Eintrag der Liste im *Clip*-Menu, wird Ihre lokale Zwischenablage von *Windows* mit dem Wert, der sich hinter dem Listeneintrag verbirgt, überschrieben. Es werden immer nur die ersten zwölf Zeichen eines jeden Eintrages angezeigt. Zeilenumbrüche werden aus den Listeneinträgen 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 nutzen.

5.6 Eintrag entfernen

Im *Clip*-Menü werden standardmäßig die letzten 50 Einträge dargestellt. Die Anzahl der darzustellenden Elemente können Sie in der *config.properties* unter dem Parameter *number* einstellen. Passen nicht alle Elemente auf eine Seite, werden entsprechend zusätzliche Seiten angelegt. Möchten Sie einen Eintrag dauerhaft entfernen, klicken Sie auf das neben dem Eintrag stehende Kästchen. Daraufhin startet 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 *JSON*-Format (*JavaScript Object Notation*) 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 nächstjüngere Eintrag hinterher und es werden wieder 50 Einträge dargestellt.

5.7 Einstellungen

5.7.1 Systemkonfiguration

Unter 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* setzen und somit kontrollieren, ob das Programm beim Drücken des Shortcuts *STRG + C* den Inhalt der Zwischenablage in die Datenbank schreiben soll. Somit geben wir dem Benutzer die Möglichkeit, einen *nur-Lese*-Modus zu verwenden. Im nächsten 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. Nachdem Sie eine Änderung innerhalb der Auswahlbox getroffen haben, wird der Inhalt des *Cascading Style Sheets* (CSS) geladen und die Desktopanwendung neu gezeichnet, so dass das Theme direkt ohne Neustart angewendet wird. Zeitgleich wird in der Konfigurationsdatei automatisch der 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 verwenden. Es ist dabei egal wo die CSS-Datei auf dem Dateisystem hinterlegt ist, denn der Pfad kann über den *Choose*-Button ausgewählt werden.

Nach Klicken des *Choose*-Buttons öffnet sich ein Dialog-Fenster in dem der Ordner, der die gewünschte *CSS*-Datei(en) beinhaltet, ausgewählt werden kann. Nachdem Sie den ausgewählten Ordner im 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. Hierbei steht der erste Wert für die Breite und 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 einen ganzzahligen Wert ein. Zum Übernehmen, drücken Sie die Taste *ENTER*, klicken 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. Zeitgleich wird der neue Wert in der Konfigurationsdatei gespeichert.

5.7.2 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-Adresse aus der Konfigurationsdatei zu entfernen, klicken Sie auf den Button *Forget Me*. 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-Adresse und dem dazugehörigen Passwort authentifizieren.

5.8 Passwort vergessen

Sollten Sie einmal Ihr Passwort vergessen haben, können Sie es in wenigen Schritten zurücksetzen. Gehen Sie zunächst innerhalb der Desktopanwendung auf das *Account*-Menü und klicken Sie auf den 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 einem Klick auf *Confirm*. Wurde die E-Mail in unserem System gefunden und bereits aktiviert, so erhalten Sie die Mitteilung *Successfully request, check your mails to change*. Bestätigen Sie diese Meldung anschließend 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 das Formular aus. Tragen Sie hier 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 vergebenen Passwort im *Account*-Menü einloggen.

6 Erweiterung und geplante Erweiterung

6.1 Einführung einer Blacklist

Für die Einführung einer Blacklist, ist zunächst eine Erweiterung der Datenbankstruktur notwendig. Eine neue Tabelle mit dem Namen *clipboarderBlackList* und den Spalten *ID* und *Email* würde hier sinnvoll sein. Für jede Aktion, bei der eine E-Mail gesendet wird, 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-Mails, die unser System bisher versendet, überarbeitet werden. In jede E-Mail-Vorlage muss ein Link eingebaut werden, der beim Aufruf dafür sorgt, dass die E-Mail-Adresse des Empfängers in die Liste aufgenommen wird. Zeitaufwandsschätzung: 2,5 Stunden.

6.2 Änderung des Verschlüsselungs-Schlüssels

Bisher ist es nicht möglich, den *cryptKey* in der Konfigurationsdatei zu ändern, falls bereits Zwischenablagen mit dem Konto verknüpft sind. Es müssen erst alle Zwischenablagen (zurzeit noch jede einzeln) gelöscht werden, so dass 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 welches 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. Danach müssten die 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 in der Datenbank zurückgesetzt werden müssen, sondern auch der alte *cryptKey* wiederhergestellt werden muss, so dass die bisherigen Zwischenablagen weiter entschlüsselt werden können.

Zeitaufwandsschätzung: 8 Stunden.

6.3 Löschen des Benutzerkontos

Wenn sich ein Benutzer dazu entschließt, sein Benutzerkonto bei uns zu entfernen, 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 dies für die Benutzer in Zukunft angenehmer zu gestalten, haben wir einen zusätzlichen Button *delete account* im *Settings*-Menü geplant. Geplant ist, dafür eine Meldung mit der Frage: „*Are you sure you want to delete your account irrevocably?*“ anzuzeigen, falls dieser gedrückt wird. 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 und all seine Daten werden aus der Datenbank entfernt. Abschließend erhält er eine E-Mail mit der Bestätigung der Kontolöschung.

7 Anhang

7.1 Datenbankstruktur (SQL)

```
--
-- Tabellenstruktur für Tabelle `clipboarduser`
--
CREATE TABLE IF NOT EXISTS `clipboarduser` (
  `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 `clipboarduser`
--
ALTER TABLE `clipboarduser`
ADD PRIMARY KEY (`ID`), ADD UNIQUE KEY `EMail` (`EMail`);
--
-- AUTO_INCREMENT für Tabelle `clipboarduser`
--
ALTER TABLE `clipboarduser`
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 `clipboarduser` (`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 `clipboarduser` (`ID`);
--
-- AUTO_INCREMENT für Tabelle `clipboarderlogin`
--

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

7.2 Screenshots

AccountSettings–

Authentification

E-Mail

Password

☐ Remember me

Login

Register

Forgot Password

AccountSettings–

Registration

Firstname

E-Mail

Password

Register

AccountSettings–

Settings

☒ Enable Recording

CSS Directory

src\com\heikweber\clipboard

Choose

Theme

Office

200px

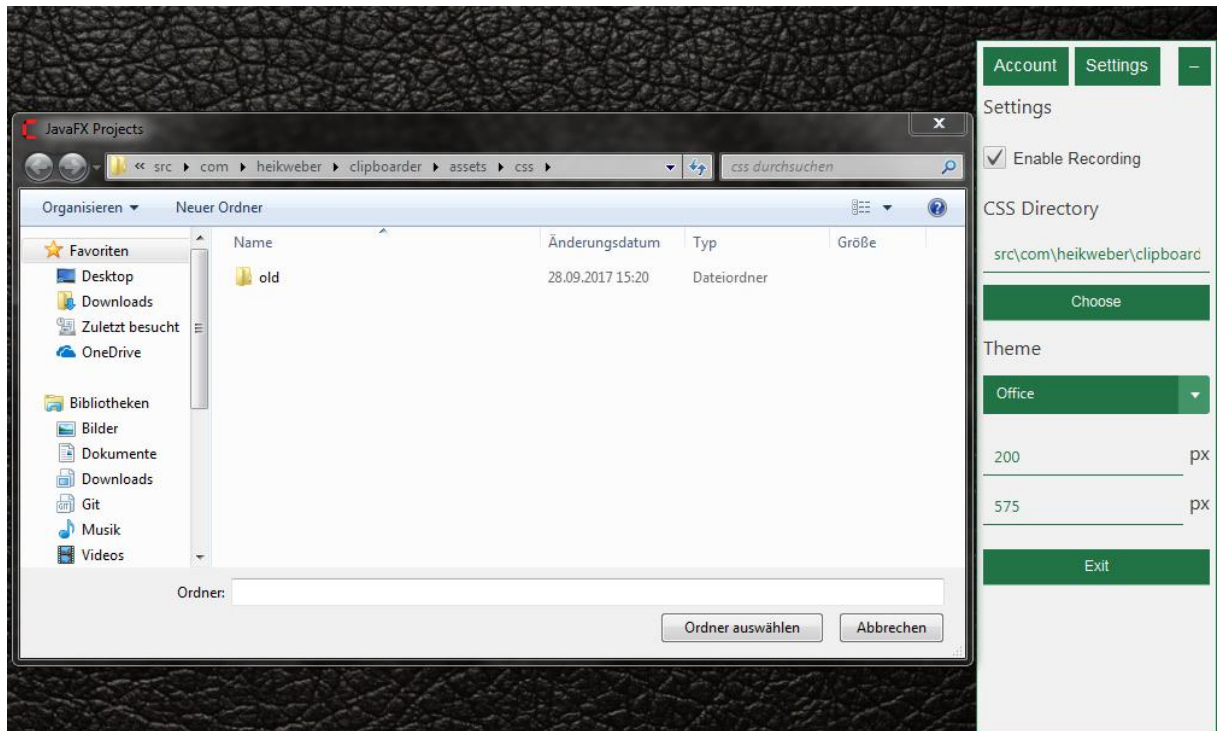
575px

Exit

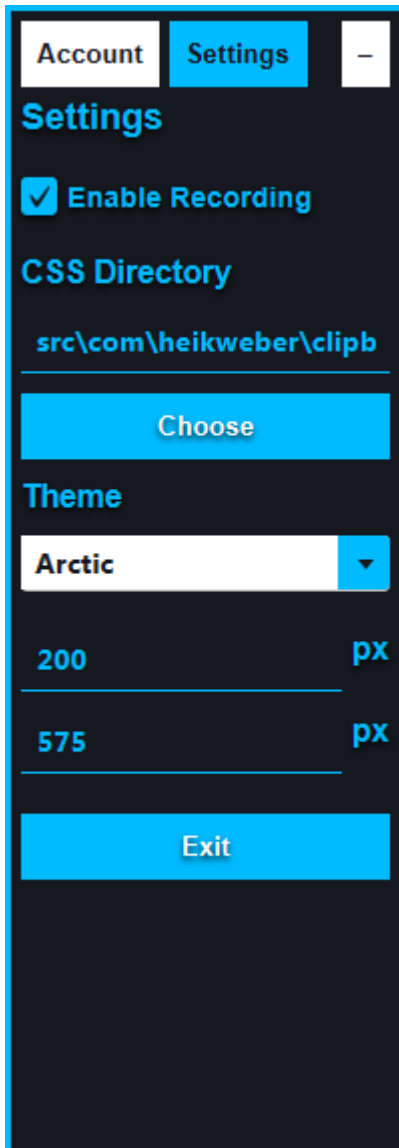
Account-Menü

Registrierungs-Menü

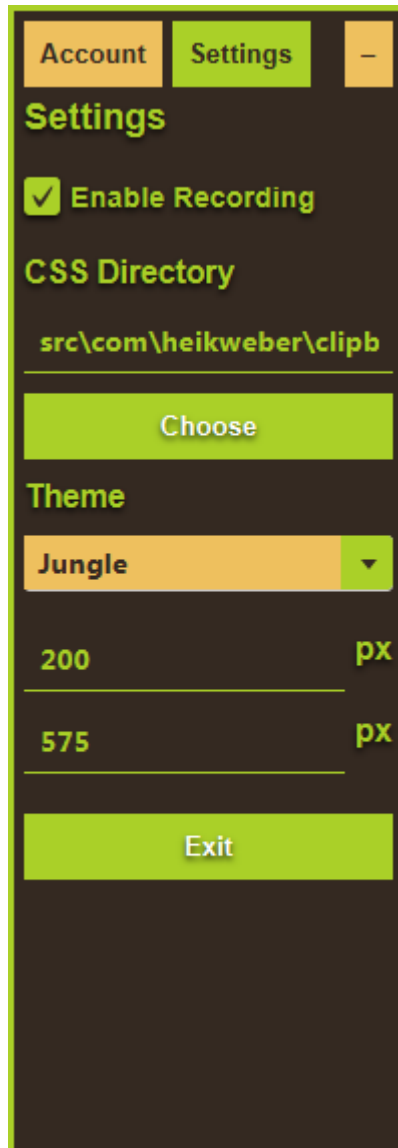
Settings-Menü



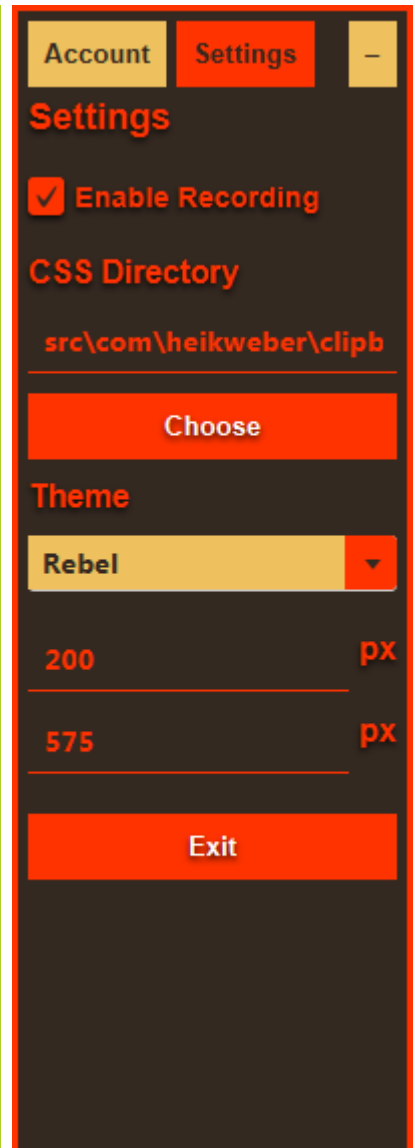
Auswahl des CSS-Ordnerns



Theme: Arctic



Theme: Jungle



Theme: Rebel