Inhaltsverzeichnis

1 Installation

Für die Installation müssen folgende Schritte durchlaufen werden:

1.1 Installation der Dateien

Bitte kopieren Sie alle Dateien in das gewünschte Stammverzeichnis. Ihr Webserver muss PHP und MySQL unterstützen, um diese Webseite anbieten zu können. Unter Umständen müssen Sie die Pfade der Datei `./.htaccess` an Ihre Serverinstallation anpassen.

1.2 Setzen der Konstanten

Setzten Sie alle wichtigen Konstanten in `./config.php`. Die Verwendung und Bedeutung der Konstanten wird in der Datei erklärt. Achten Sie darauf, dass die Konstante `INSTALL` auf den Wert 1 gesetzt wird.

1.3 Installation der Datenbank

Stellen Sie sicher, dass die in `./config.php` angegebene Datenbank existiert. Führen Sie die Datei `./db_setup.php` mit PHP aus. Alternativ können Sie auch die url `http://www.IhreWebsite.ch/Pfad_zu_db_setup.php/db_setup.php` aufrufen.

1.4 Einrichten

Root-Benutzer Es wird ein Hauptbenutzer angelegt. Sie müssen für diesen Benutzer ein Passwort festlegen. Dies geschieht mit dem Aufruf von `http://www.IhreWebsite.ch/Pfad_zu_setRootPass.php/setRootPass.php?passwd=IhrPasswort`. Ersetzen Sie `IhrPasswort` mit einem sicheren Passwort.

Installation abschliessen Nun können Sie die Konstante `INSTALL` auf 0 setzen. Die Installation ist somit abgeschlossen.

Bemerkung Das `./uploads/` Verzeichnis muss für php beschreibbar sein. Setzen Sie wenn nötig die erforderlichen Rechte (zum Beispiel mit `chgrp http ./uploads`).

2 Umsetzung

2.1 Funktionsumfang

2.1.1 Geplant

1. Verschiedene Benutzer mit verschiedenen Aufgabenbereichen
2. Aufträge können von allen Benutzern bearbeitet werden

3. Jeder Auftrag soll einen Verlauf haben
4. Zuordnen von Aufträgen an Benutzern, weiterreichen von Aufträgen
5. Ein Auftrag durchläuft verschiedene Stufen

2.1.2 Umsetzung

Hier jeweils ein kurzer Kommentar zur Umsetzung der verschiedenen Zielsetzungen

1. Wurde umgesetzt mit 3 Benutzergruppen (Verwaltung, Lagerist, Arbeiter)
2. Alle Aufträge sind von zuständigen Benutzern veränderbar
3. Der Verlauf wurde als XML-Datei realisiert
4. Aufträge können zugeordnet und weitergereicht werden, jedoch fehlt eine Art Übersicht über die Zuordnungen
5. Der Auftrag durchläuft verschiedene Stufen (Erfassung, Abarbeiten, Abrechnen, Archiv)

2.2 Aufbau

2.2.1 Dateisystem

Unsere Ordnerstruktur ist so gestaltet, dass alle Dateien nach Funktion aufgeteilt werden. Das Zusammensetzen der dargestellten Seite wird dabei von der Datei `index.php` übernommen. Dass heisst, fast alle Funktionalitäten unserer Seite sind über `index.php` erreichbar. Javascript Dateien werden nur nach Bedarf geladen.

```
/
├── installfiles ..... Installationsdateien
├── templates ..... Vorlagen für die Seiten
├── css ..... CSS Dateien
├── js ..... Javascript Dateien
├── handlers ..... Ajax Handler / REST-Interface
├── img ..... Bilder
├── uploads ..... Hochgeladene Dateien
└── doc ..... Dokumentation
```

2.2.2 Seitenaufbau

Hier der exemplarische Aufbau unserer `index.php` Datei. Der jeweilige Seiteninhalt wird dynamisch als `<article></article>` eingebunden. Javascript Dateien werden spezifisch für jede Seite eingebunden.

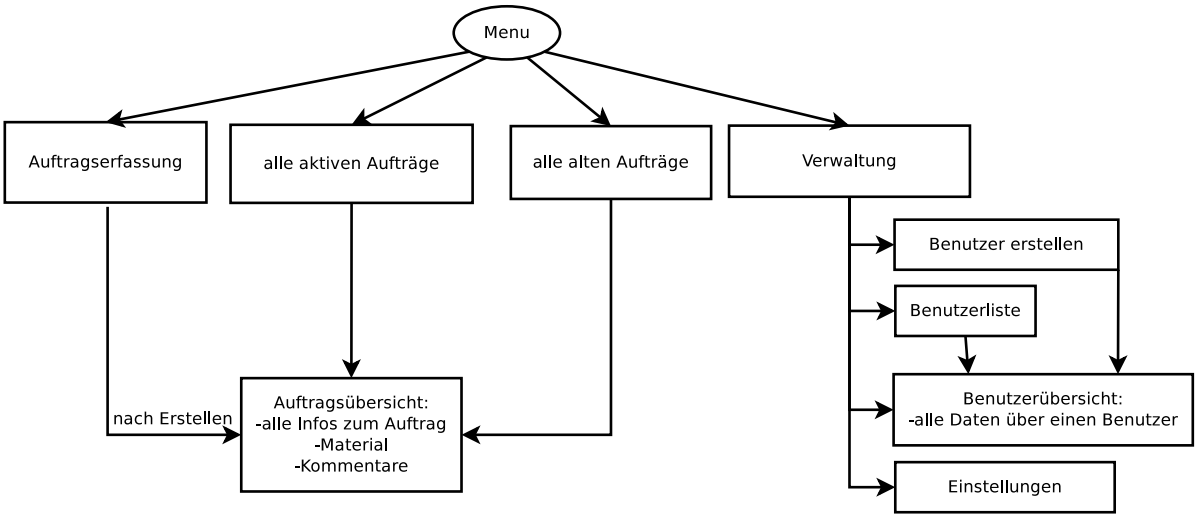
```
<html>
   <head>
      <!-- Hier sind alle CSS Dateien eingebunden -->
   </head>
   <body>
      <div id="content">
         <header>
            <!-- Hier ist das Menu -->
         </header>
         <!-- Benachrichtigungsfeld -->

         <article>
            <!-- hier ist der verlangte Seiteninhalt -->
         </article>


      </div>
      <!-- Hier sind alle Javascript Dateien eingebunden -->
   </body>
</html>
```

2.2.3 Benutzerinteraktionen

Unsere Seite setzt sehr viel Javascript für die Benutzerinteraktionen ein. Rund 7% unseres Codes ist Javascript. Wo möglich wurden viele Interaktionen mittels Ajax gelöst. Jedoch ist ein neu laden der Seite manchmal unumgänglich.

2.2.4 Benutzerführung

Folgende Graphik zeigt unsere Benutzerführung.



2.3 Datenbank

Auf folgenden Graphiken ist unser Datenbankschema dargestellt. Beim Verlauf wollten wir ursprünglich noch Benachrichtigungen für Benutzer zur Verfügung stellen. Aus Zeitgründen haben dies jedoch nicht mehr implementiert. Auch die Verknüpfung des Verlaufes mit den einzelnen Datenbankeinträgen ist zwar laut Schema vorgesehen, jedoch nicht implementiert. Wir haben absichtlich keine Trigger / Stored Procedures verwendet, da diese von den meisten Hostern nicht unterstützt werden.

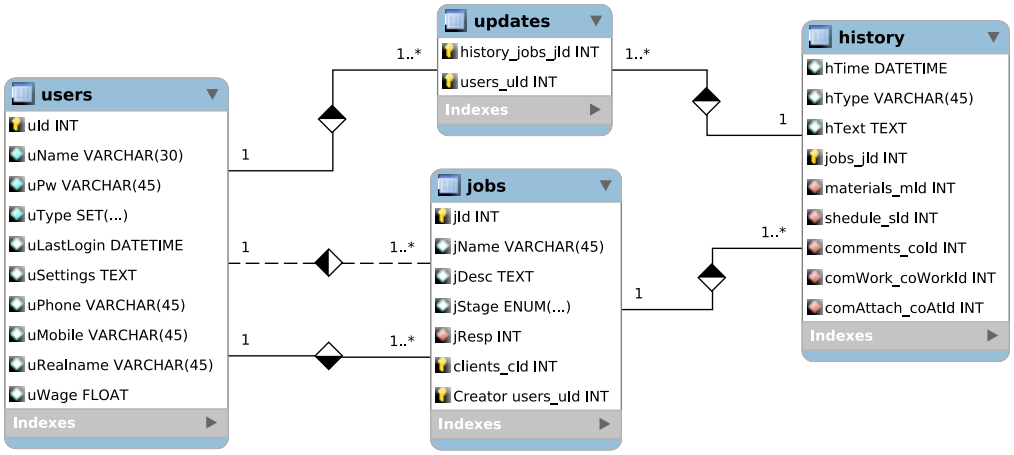


Abbildung 1: Tabellen welche für die Benachrichtigungen (Updates) und Rückverfolgbarkeit zuständig sind.

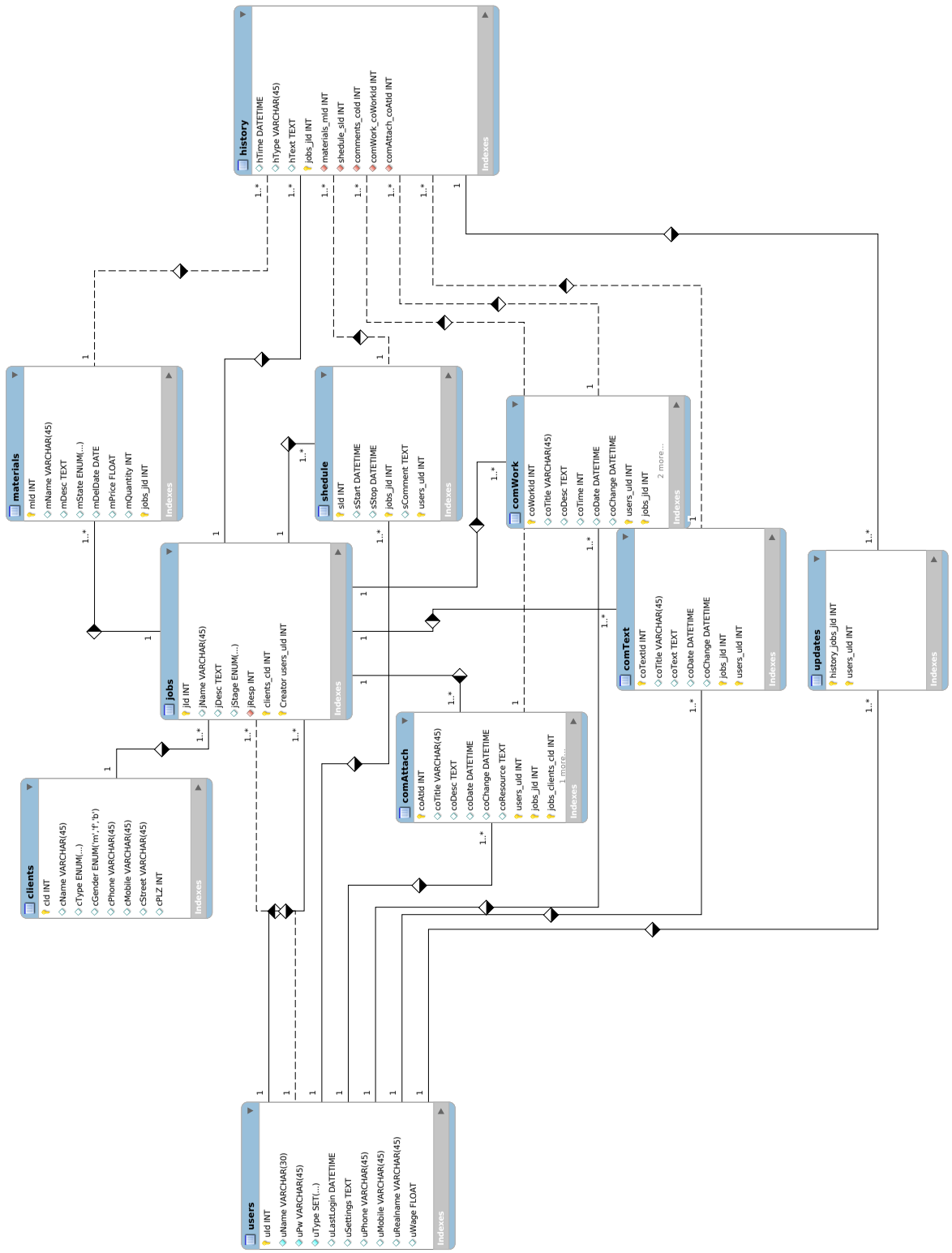


Abbildung 2: Das gesamte Datebankschema

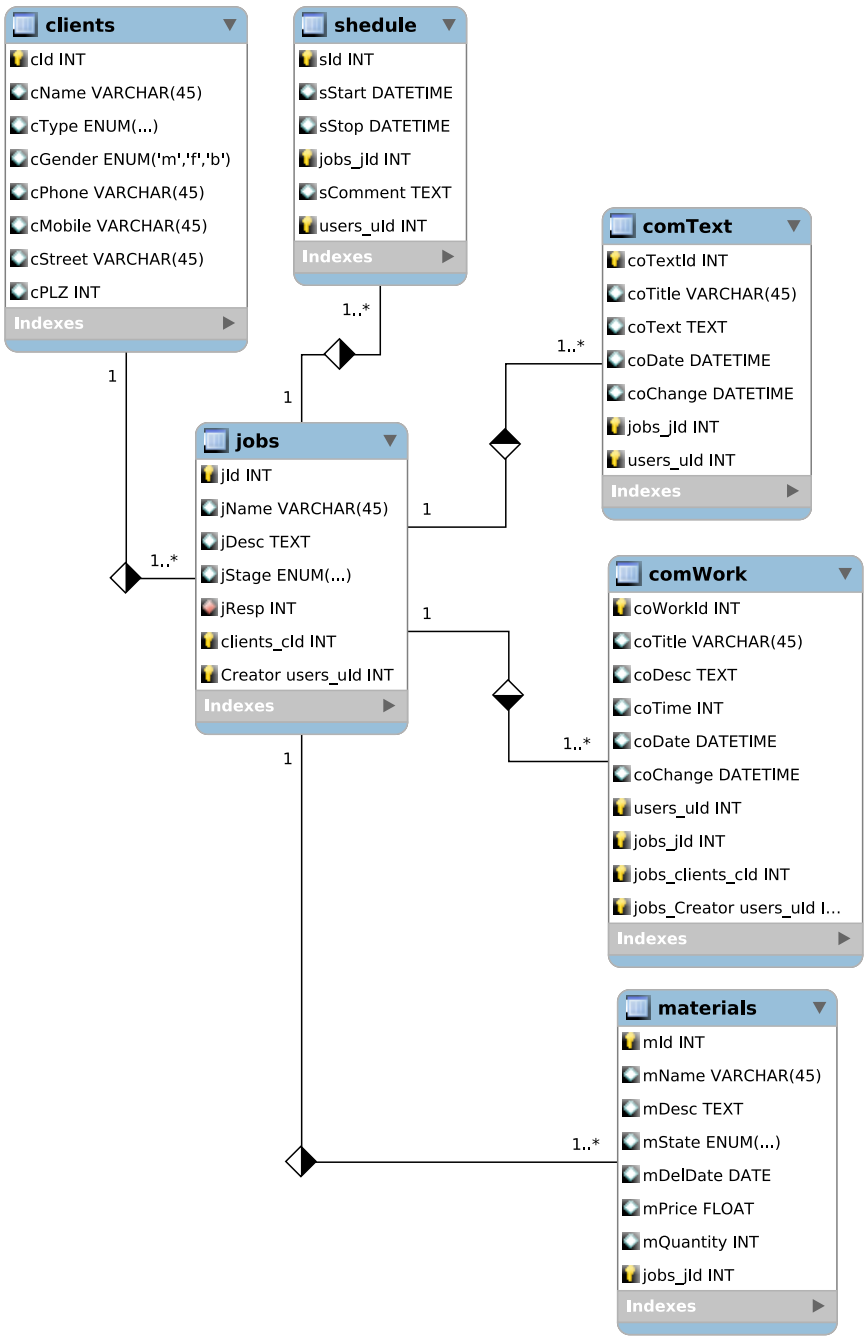


Abbildung 3: Alle Tabellen in denen Informationen betreffend eines Auftrages sind.

2.4 Verwendete Komponenten und Bibliotheken

- jQuery http://jquery.com/ (Javascript-Bibliothek)
- Iconic https://github.com/somerandomdude/Iconic(Icons)
- SuperBox http://pierrebertet.net/projects/jquery_superbox/(Javascript Overlay)
- tinyMCE www.tinymce.com/(Texteditor)
- AjaxFileUploader https://github.com/jfeldstein/jQuery.AjaxFileUpload.js(Asynchron Dateien hochladen)

2.5 REST-Interface

Um auf das REST-Interface zugreifen zu können, ist ein Login notwendig. Ein solches Login funktioniert wie folgt:
http://mysite.com/handlers/startSessionREST.php?name=root&passwd=1234
(Cookies müssen aktiviert sein)
Nun kann das gesammte REST-Interface genutzt werden. Die php-Skripte im /handlers/ Ordner sind jeweils kommentiert, welche Parameter sie akzeptieren und wofür diese stehen. Zusätzlich ist es möglich mit Tools wie 'Live HTTP headers' jeweils Beispiele zu erhalten, da die Website selbst auch das REST-Interface nutzt.

3 Benutzeroberfläche

3.1 Verwaltung

In der Verwaltung eingeteilte User haben das Privileg neue Benutzer hinzuzufügen und Passwörter anderer User zurückzusetzen.

3.2 Lagerist

Der Lagerist kann ihm zugewiesene Aufträge editieren, Materialien editieren, hinzufügen und den Auftrag einer andern Person zuweisen.

3.3 Arbeiter

Der Arbeiter kann ihm zugewiesene Aufträge editieren und an andere Personen zuweisen.

4 Fazit

4.1 Einschränkungen

- Ein Nutzer kann nur Aufträge sehen/bearbeiten, die ihm zugewiesen sind(Es handelt sich hier um eine gewollte Einschränkung).
- Gewisse Interaktionen sind nicht Asynchron ausgeführt
- Benutzerverwaltung ist rudimentär

4.2 Problemstellen

- Sind die Zugriffsrechte im Dateisystem so gesetzt , dass `./uploads/` nicht durch php mit Dateien gefüllt werden kann, so kann dies zu unerwünschtem Verhalten führen.
- Da php die hochgeladenen Dateien im `/tmp` Verzeichnis zwischenspeichert, kann dies zu Problemen führen, wenn `/tmp` als tmpfs gemounted ist und die Summe über die Grössen aller gleichzeitig hochgeladenen Dateien zu gross ist.
- Die Website funktioniert nicht korrekt mit alten Internet Explorer Versionen. Wir bitten IE Nutzer deshalb einen Browser mit HTML5-Unterstützung zu benutzen.

4.3 Lessons learned

- Asynchronder Dateiupload geht nicht nativ mit jQuery.
- Auswerten von grösseren Formularen ist sehr repetitive Arbeit. Diesbezüglich war das von uns gewählte Projekt suboptimal.
- Ajax benötigt klare Abmachungen der Schnittstellen und ist sehr zeitaufwendig
- Browser $\neq$ Browser
- Keine langen Unterbrüche machen (Einarbeitungszeit)