

Online-Verwaltungstool für Mehrfamilienhaus

Aufgabe B



URL: <https://www.photoca.se>

Admin / PW: admin@admin.ch / admin

Projektarbeit Web-Engineering

Publikation: 2017

Autoren:

Kevin Trottmann, Matej Djuranec, Mikayil Bekooglu

Fachhochschule Nordwestschweiz

Inhaltsverzeichnis

1	Aufgabe	3
1.1	Ausgangslage.....	3
1.2	Ziele.....	3
2	Projektmanagement:	3
2.1	Planung.....	3
2.2	Team / Aufgabenzuteilung	4
2.3	Ablauf.....	4
2.4	Hilfsmittel.....	4
3	Anforderungsanalyse.....	5
3.1	Use Cases.....	5
3.2	Datenbankdiagramm.....	7
3.3	PHP Funktionen.....	8
4	Technische Umsetzung.....	12
4.1	Webserver (Metanet)	12
4.2	Zugangsdaten.....	12
4.3	Testing	12
4.4	Kurze Benutzeranleitung	13
5	Fazit.....	15

1 Aufgabe

1.1 Ausgangslage

Da ist ein Besitzer eines 12-Familienhauses, der die Verwaltung selbständig führen will. Wir wollen diesem Verwalter ein praktisches Onlinetool erstellen, welches er einfach über den Webbrowser bedienen kann.

1.2 Ziele

Folgendes wollen Sie jederzeit online abrufen können:

- Mieterspiegel (Adressen und Mietzinse)
- Erfasste Rechnungen (Reparatur-, Oel-, Wasser-, Strom-, Hauswartsrechnungen)
- Weitere Rechnungen sollen jederzeit erfasst werden können.
- Mietzinseingänge sollen einfach erfasst werden können.
- Ende Jahr soll man eine Abrechnung ausdrucken können.
- Die Heizkostenabrechnung und die Nebenkosten sollen ausgewiesen werden.
- Über Formulare können Daten eingegeben, mutiert oder gelöscht werden.

Erstellen Sie diese Online-Verwaltungshilfe für den Hausverwalter. Benutzen Sie Webformulare für die Dateneingabe und lösen Sie die Datenausgabe mit PHP oder Perl. Sie können dazu auch die Datenbank MySQL verwenden. Auch alle weiteren Sprachelemente, die Sie kennen, können eingesetzt werden (JavaScript, CSS, ASP usw.)

2 Projektmanagement:

2.1 Planung

Initialisierung:

- Themenwahl
- Vorbereitung Webserver & Datenbank & Domain
- Auswahl der Editoren
- Einrichten und Verbinden von Github

Planung

- Erstellung der Use Cases
- Zuteilung der Use Cases und Arbeiten

Realisierung

- Jeder programmiert einzelne Codesnippets und Funktionen
- Bei den Skypemeetings und Treffen werden diese dann zusammengeführt
- So haben wir Seite für Seite aufgebaut

Abschluss

- Vergleich Use Case mit Funktionalität
- Testing
- Code überarbeiten (formatieren & kommentieren)
- Fertigstellung der Dokumentation

2.2 Team / Aufgabenzuteilung

Kevin Trottmann:

- DB Connections
- Login / Registrierung
- Create, Insert, Change, Delete

Matej Djuranec:

- Gestaltung Bootstrap
- Welcome Page
- Formulare, Header, Footer
- Session Control

Mikayil Bekooglu:

- PDF Generator
- Overview Mieterspiegel
- Abrechnungen / Einnahmen

2.3 Ablauf

Wöchentliche Skype Termine, Dropbox Ordner für den Datenaustausch und eine To-do-Liste, in welcher laufend die anstehenden Pendenzen zugeteilt wurden, waren hauptsächlich unsere Kommunikationskanäle. Für dringendere Abklärungen nutzten wir den Gruppenchat in WhatsApp. Des Weiteren trafen wir uns regelmässig in der Schule, um unseren Wissensstand abzugleichen und am dem Projekt zu arbeiten.

2.4 Hilfsmittel

Dropbox: Dokumentation laufend geführt, To-Do-Liste

Webserver: Der Code wurde direkt auf dem Webserver programmiert und jeweils mit Github versioniert und gesichert. Jedes Projektmitglied hatte seine eigenen PHP-Dateien und konnte dort seine Funktionen programmieren und testen. Nach erfolgreichem Test wurde der Code in das Hauptverzeichnis kopiert und an die Webseite angehängt.

Editor: Jedes Projektmitglied arbeitete mit unterschiedlichen Tools, da jedes Projektmitglied seine bevorzugte Entwicklungsumgebung hat. Folgende Tools wurden von uns verwendet: PHP-Storm und Notepad++

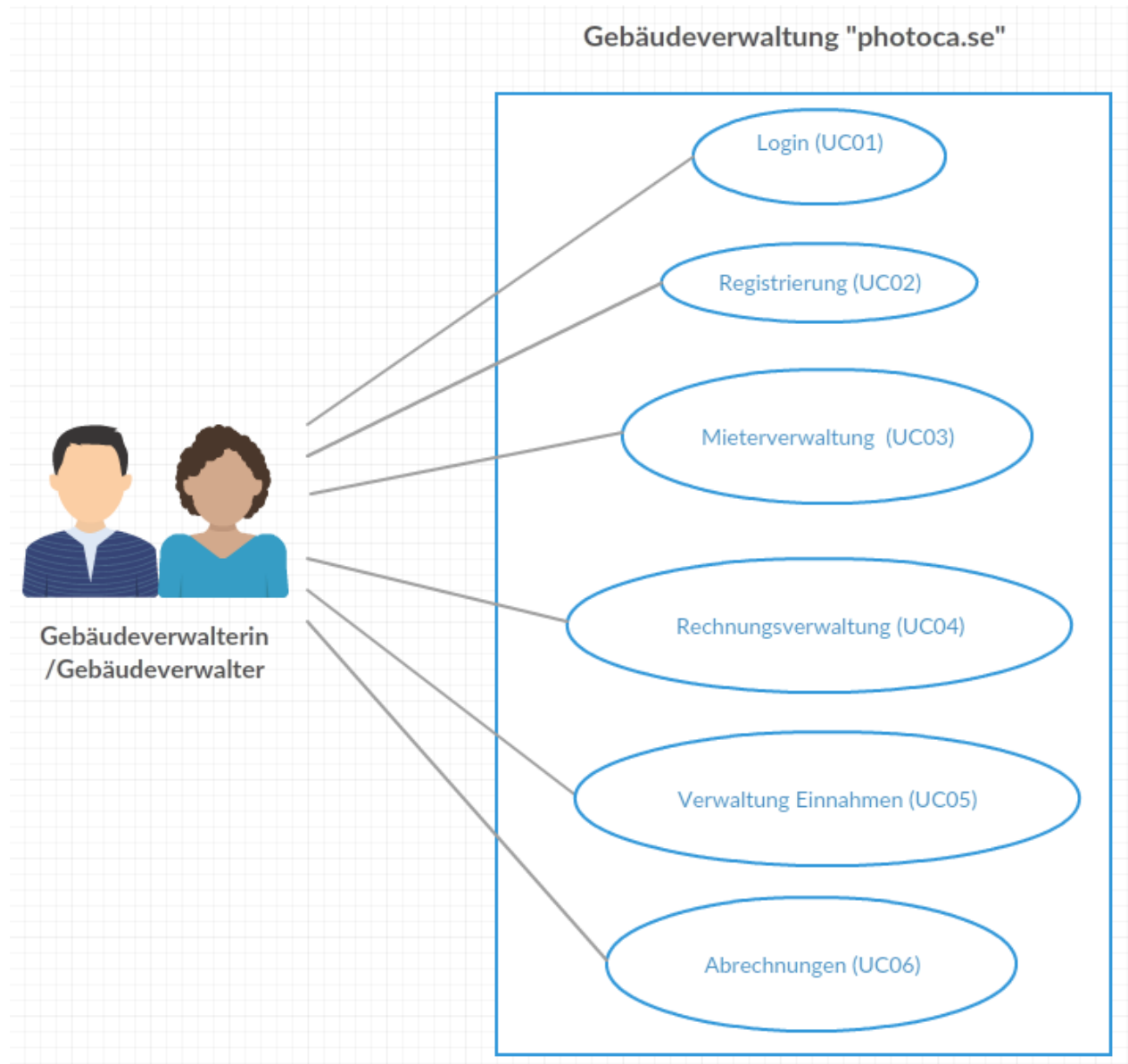
Github: Jeder hatte auf seinem Laptop die Desktopversion von Github, damit die Dateien lokal zusätzlich noch gesichert waren und so ein konfliktloser Austausch des Codes stattfinden konnte.

FileZilla: Wir haben nicht mit XAMPP gearbeitet, sondern mithilfe von FileZilla und einer online Datenbank direkt auf dem Webserver programmiert.

3 Anforderungsanalyse

3.1 Use Cases

Um die Anforderungen genauer aufzuzeigen, wurden Use Case Diagramme erstellt. Wir haben insgesamt sechs Use Cases erstellt:



3.1.1 UC01 – Login

UC01 - Login	
Beschreibung	Der User soll die Möglichkeit haben, sich mit seinen Logindaten auf «photoca.se» anzumelden. Externen Benutzern, die über kein Login verfügen, soll der Zugriff auf die internen Informationen wie Mieterspiegel, Abrechnung usw. verwehrt werden. Die einzelnen PHP-Seiten sollen ohne Login nicht direkt erreichbar sein.

3.1.2 UC02 - Registrierung

UC02 - Registrierung	
Beschreibung	Der Gebäudeverwalter soll die Möglichkeit haben, sich auf der Website zu registrieren und Login-Daten zu erhalten. Wenn die Login-Daten verloren gehen, muss sich der Gebäudeverwalter an den Administrator der Website wenden.

3.1.3 UC03 – Mieterverwaltung

UC03 - Mieterverwaltung	
Beschreibung	Der Gebäudeverwalter soll die Möglichkeit haben, neue Mieter zu erfassen. Dies soll über ein Formular «Neuer Mieter» geschehen. Zusätzlich muss die Möglichkeit bestehen, bestehende Daten zu bearbeiten oder zu löschen.

3.1.4 UC04 – Rechnungsverwaltung

UC04 - Rechnungsverwaltung	
Beschreibung	Der Gebäudeverwalter soll die Möglichkeit haben, Rechnungen auf der Website einzutragen. Dabei ist zwischen folgenden Typen zu unterscheiden: Heizkosten, Nebenkosten. Weiter sind Rechnungsarten zu definieren: Reparaturrechnung, Ölrechnung, Wasserrechnung, Stromrechnung, Hauswartsrechnung. Rechnungen sollen auch bearbeitet oder gelöscht werden können.

3.1.5 UC05 – Verwaltung Einnahmen/Ausgaben

UC05 – Verwaltung Einnahmen	
Beschreibung	Der Gebäudeverwalter soll die Möglichkeit haben, Einnahmen von Mietern auf der Website zu erfassen. Die Einnahmen werden manuell pro Mieter erfasst und können auch bearbeitet oder gelöscht werden.

3.1.6 UC06 – Abrechnungen

UC06 - Abrechnungen	
Beschreibung	Der Gebäudeverwalter soll die Möglichkeit haben, Jahresabrechnungen zu erstellen. Hierbei soll dargestellt werden, welche Mieteinnahmen erzielt wurden und welche Rechnungen bezahlt wurden. Dabei soll das Jahrestotal ausgewiesen werden.

3.2 Datenbankdiagramm

Host:localhost:3306 (MySQL)

Benutzer: db_root

Tabellen: 4

Größe: 80.0 KB

Tabellenstruktur:

db_photocase users
id : int(11) unsigned
email : varchar(64)
passwort : varchar(64)
vorname : varchar(64)
nachname : varchar(64)
created_at : timestamp

db_photocase mieter
ID : int(11)
Nachname : varchar(32)
Vorname : varchar(32)
Strasse : varchar(32)
PLZ : int(4)
Ort : varchar(32)
Liegenschaft : int(3)
Mietzins : double

db_photocase einnahmen
ID : int(11)
ID_Mieter : int(11)
Datum : date
Betrag : double

db_photocase rechnungen
ID : int(11)
Typ : varchar(32)
Art : varchar(32)
Rechnungstext : varchar(200)
Datum : date
Betrag : double

3.3 PHP Funktionen

Die wichtigsten Funktionen sind hier dargestellt – der restliche Code ist im Anhang «Code_photocase.zip» ersichtlich. Viele Funktionen bauen auf demselben Prinzip auf wie z.B. Add-Mieter oder Chg-Mieter.

Session Control:

```
session_start();

//Überprüft ob User eingeloggt
if (!isset($_SESSION['userid'])) {
    header("location:http://photoca.se/index.php");
}
```

Startet den Session controller und überprüft, ob der aktuelle User sich eingeloggt hat. Falls nicht, wird der User an die Login-Seite weitergeleitet.

```
session_start();
session_destroy();

if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

header("Location: http://photoca.se/index.php");
die();
```

Beim Logout wird die Session mit dem Destroyer aufgelöst – Cookies werden gelöscht, sodass nach dem Logout die einzelnen Seiten manuell auch nicht mehr erreichbar sind.

DB Connection:

```
$host="localhost";
$benutzer="db_root";
$password="aVu8k&13";
$dbname="db_photocase";

$link=mysqli_connect($host,$benutzer,$password,$dbname) or
die
("Keine Verbindung zur DB moeglich");
mysqli_set_charset($link, 'utf8');
```

Die Verbindung zur Datenbank findet über das File db.connection.php statt. Dieses ist in jedem PHP-File inkludiert. Darin hinterlegt sind Datenbankname, Benutzer und Passwort.

Darstellung Mieter (HTML):

```
<body>
<div class="container">
  <div class="page-header">
    <h3>Mieter</h3></div>
  <div class="table-responsive">
    <table class="table table-striped">
      <thead>
        <tr>
          <th>Mieter ID</th>
          <th>Nachname</th>
          <th>Vorname</th>
          <th>Strasse</th>
          <th>PLZ</th>
          <th>Ort</th>
          <th>Liegenschafts-Nr.</th>
          <th>Mietzins</th>
          <th>Löschen</th>
          <th>Bearbeiten</th>
        </tr>
      </thead>
      <tbody>
        <?php
$res_mieter = mysqli_query($link, "SELECT * FROM mieter");
while ($datensatz = mysqli_fetch_assoc($res_mieter)) {

    echo "<tr><td>" . $datensatz["ID"] . "</td>";
    echo "<td>" . $datensatz["Nachname"] . "</td>";
    echo "<td>" . $datensatz["Vorname"] . "</td>";
    echo "<td>" . $datensatz["Strasse"] . "</td>";
    echo "<td>" . $datensatz["PLZ"] . "</td>";
    echo "<td>" . $datensatz["Ort"] . "</td>";
    echo "<td>" . $datensatz["Liegenschaft"] . "</td>";
    echo "<td>" . $datensatz["Mietzins"] . "</td>";
    echo "<td><form action='db_del_mieter.php' method='POST'><input type='submit' class='btn btn-primary'
name='löschen' value='DEL'> <input type='hidden' name='ID' value='" . $datensatz["ID"] . "'></form></td>";
    echo "<td><form action='db_chg_mieter.php' method='POST'><input type='submit' class='btn btn-primary'
name='bearbeiten' value='CHG'> <input type='hidden' name='ID' value='" . $datensatz["ID"] . "'></form></td></tr>";

  }
  <?>
    </tbody>
  </table>
  <a href="form_neuermieter.php" class="btn btn-primary" type="button"> Mieter erfassen </a>
  <a href="pdfgeneratormieter.php" target="_blank" class="btn btn-primary">Drucken</a>

  <br>
  <br>
  <br>
</div>
</div>
</body>
```

Der Mieterspiegel ist als Tabelle in HTML aufgebaut und fügt mit einer FOR-Schleife die Daten aus der Datenbank ein. Für jede Zeile wird ein Input-Formular erstellt, damit die Buttons DEL & CHG abgefragt werden können.

Die Tabelle dient als Vorlage für die Darstellung der Rechnungen.

Registrierung:

```
if (isset($_GET['register'])) {
    $error = false;
    $email = $_POST['email'];
    $password = $_POST['password'];
    $password2 = $_POST['password2'];
    $vorname = $_POST['vorname'];
    $nachname = $_POST['nachname'];

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo 'Bitte eine gültige E-Mail-Adresse eingeben<br>';
        $error = true;
    }

    if (strlen($password) == 0) {
        echo 'Bitte ein Passwort angeben<br>';
        $error = true;
    }

    if ($password != $password2) {
        echo 'Die Passwörter müssen übereinstimmen<br>';
        $error = true;
    }

    //Überprüfe, dass die E-Mail-Adresse noch nicht registriert wurde
    if (!$error) {
        $statement = $pdo->prepare("SELECT * FROM users WHERE email = :email");
        $result = $statement->execute(array('email' => $email));
        $user = $statement->fetch();

        if ($user != false) {
            echo "alert('Diese E-Mail-Adresse ist bereits vergeben')";
            $error = true;
        }
    }

    //Keine Fehler, wir können den Nutzer registrieren
    if (!$error) {
        $password = md5($password);

        $statement = $pdo->prepare("INSERT INTO users (email, password,vorname,nachname) VALUES (:email, :password, :vorname, :nachname)");
        $result = $statement->execute(array('email' => $email, 'password' => $password, 'vorname' => $vorname, 'nachname' => $nachname));

        if ($result) {
            echo "<script>window.location.href = \"https://www.photoca.se\";</script>";
        } else {
            echo "alert('Beim Abspeichern ist leider ein Fehler aufgetreten')";
        }
    }
}
```

Das Registrierungsformular überprüft die Gültigkeit der Email-Adresse. Sollte diese nicht oder fehlerhaft angegeben worden sein, wird der Nutzer gewarnt und aufgefordert, eine gültige Adresse anzugeben.

Wenn keine Fehler vorhanden sind, werden die Daten in die Datenbank geschrieben. Der Nutzer wird wieder auf die Login-Page geleitet.

Vorgang Add-Mieter:

Über den Button «Mieter erfassen» auf der Seite «mieter.php» kommt man zum Formular (form_neuermieter.php), in welchem man die Angaben zum Mieter abfüllen muss. Mit der Bestätigung durch den Button «Erfassen» wird man zur Seite «db_add_mieter.php» weitergeleitet. Diese lädt die Daten in die Datenbank und leitet den User wieder auf die Mieterübersicht weiter. Die Struktur beim Bearbeiten eines Mieters ist analog, nur wird dort ein vorausgefülltes Formular dargestellt, wo die Daten dann verändert werden können.

```
include "db.connection.php";

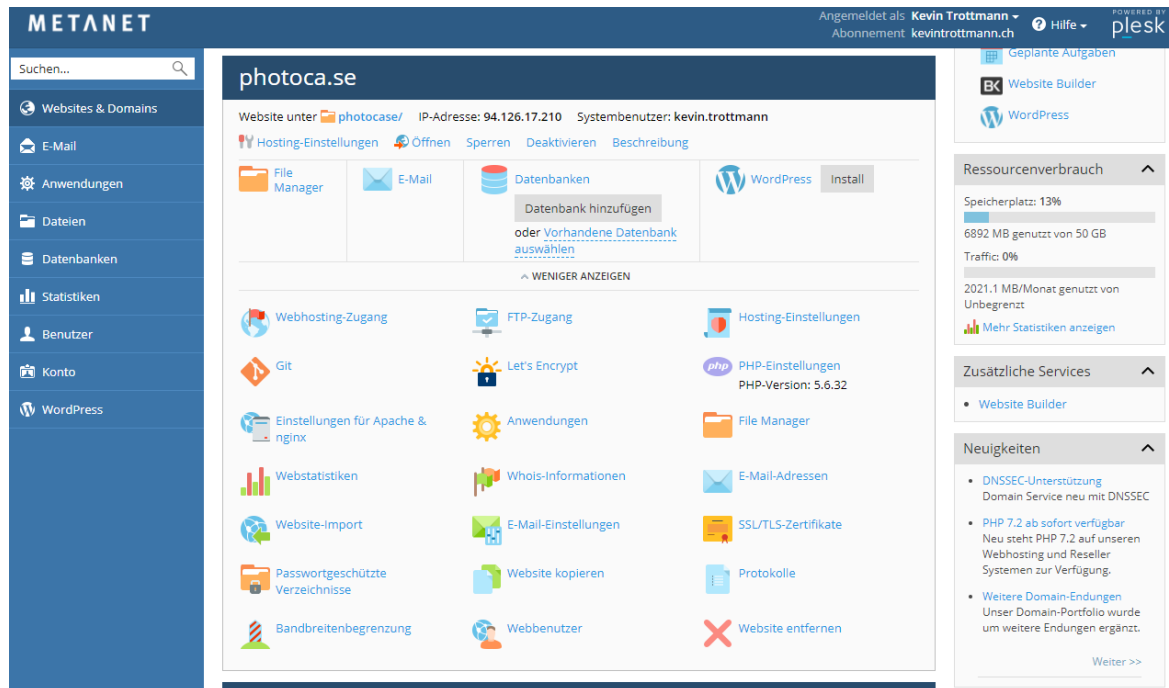
$error=false;
$vorname = $_POST['vorname'];
$nachname = $_POST['nachname'];
$strasse = $_POST['strasse'];
$plz = $_POST['plz'];
$ort = $_POST['ort'];
$liegenschaft = $_POST['liegenschaft'];
$mietzins = $_POST['mietzins'];

//Keine Fehler, wir können den Nutzer registrieren
$addsql = "INSERT INTO mieter (Vorname,Nachname,Strasse,PLZ,Ort,Liegenschaft,Mietzins) VALUES ('".$vorname."','".$nachname."','".$strasse."','".$plz."','".$ort."','".$liegenschaft."','".$mietzins."')";
$eintragen = mysqli_query($link,$addsql);

echo '<script>window.location.href = "mieter.php";</script>';
```

4 Technische Umsetzung

4.1 Webserver (Metanet)



4.2 Zugangsdaten

URL: <https://www.photoca.se>

Admin / PW: admin@admin.ch / admin

Normal: Man muss sich registrieren

4.3 Testing

Nach der Programmierung jeder PHP Funktion wurden verschiedene Tests durchgeführt. Anschliessend an diese Tests wurde das System auf verschiedenen Browsern, auf Mobile- und Desktopgeräten aus Benutzersicht getestet.

Am Ende der Arbeit haben wir dann in der Gruppe alle Funktionalitäten mit den Use Cases verglichen und so die ganze Homepage nochmals überprüft.

4.4 Kurze Benutzeranleitung

Login

Email

Password

Log In

Registrieren?

PHOTOCASE

Mieter

Rechnungen

Einnahmen

Abrechnungen

Logout

Willkommen auf Photoca.se

Was ist Photocase?

Sie sind Besitzer eines 12-Familienhauses, der die Verwaltung selbständig führen will. Mithilfe von Photoca.se erhalten Sie ein praktisches Onlinetool, welches man einfach über den Webbrowser bedienen kann.

Folgendes können Sie jederzeit online abrufen:

-Mieterspiegel (Adressen und Mietzins)

-Erfasste Rechnungen (Reparatur-, Öl-, Wasser-, Strom-, Hauswartsrechnungen)

-Weitere Rechnungen sollen jederzeit erfasst werden können.

-Mietzinseingänge sollen einfach erfasst werden können.

-Ende Jahr soll man eine Abrechnung ausdrucken können.

-Die Heizkostenabrechnung und die Nebenkosten sollen ausgewiesen werden.

-Über Formulare können Daten eingegeben, mutiert oder gelöscht werden.

Photocase 2017 - Matej Djuranec, Mikayil Bekooglu, Kevin Trottmann

PHOTOCASE

Mieter

Rechnungen

Einnahmen

Abrechnungen

Logout

Mieter

Mieter ID	Nachname	Vorname	Strasse	PLZ	Ort	Liegenschafts-Nr.	Mietzins	Löschen	Bearbeiten
4	Tüscher	Patrick	Reinweg 13	5000	Aarau	1	6000	DEL	CHG
16	Stierli	Michael	Strasse 20	5555	Stadt	2	1000	DEL	CHG
21	Burri	Nicola	Einestrasse 20	4444	Wohnfelden	3	15000	DEL	CHG
22	Trottmann	Kevin	Eine Strasse 20	5525	Fischbach-Gölikon	4	1500	DEL	CHG
23	Djuranec	Matej	Im Gugel 33	5522	Tägerig	5	1100	DEL	CHG
24	Roth	Marco	Im Gugel 33	5522	Tägerig	6	1100	DEL	CHG
27	Hitz	Dominic	Bremgarterstrasse 7	5525	Fischbach-Gölikon	7	1700	DEL	CHG
28	Stierli	Ramon	Bodenackerstrasse 42	5525	Fischbach-Gölikon	8	1600	DEL	CHG
29	Seiler	Erika	Uetlibergstrasse 231	8000	Zürich	9	2300	DEL	CHG
30	Power	Max	Ziegelackfeld 3	8919	Rottenschwil	10	1700	DEL	CHG
31	Meier	Guido	Winterstrasse 6	8919	Rottenschwil	11	1400	DEL	CHG
32	Stolz	Karsten	Buchrain 76	5017	Erlinsbach	12	1250	DEL	CHG

Mieter erfassen

Drucken

Login-Page:

Hier könne Sie sich mit Email & Passwort einloggen. Eine Registrierung ist hierfür notwendig – dazu müssen Sie den Link Registrierung anklicken und das Formular ausfüllen.

Nach dem Login werden Sie zur Welcomepage geführt. Darauf befinden sich momentan die Anforderungen an die Website, wie sie auch in der Aufgabenstellung definiert sind. Diese Seite ist ein reiner Platzhalter. Die oben ersichtliche Navbar ist auf jeder Seite vorhanden.

Die Seiten Mieter, Einnahmen und Abrechnungen sind grundsätzlich ähnlich aufgebaut. Ersichtlich ist jeweils die Ausgabe der Daten aus der Datenbank. Rechts in jeder Zeile finden Sie die Buttons DEL (löschen) & CHG (verändern). Über den Button Mieter können neue Mieter erfasst werden. Der Button Drucken generiert ein PDF mit der aktuellen Tabelle.

PHOTOCASEMieterRechnungenEinnahmenAbrechnungenLogout

Neuen Mieter erfassen

Vorname:

Nachname:

Strasse:

PLZ (4 Zahlen):

Ort:

Liegenschaft:

Mietzins:

Erfassen

Photocase 2017 - Matej Djuranec, Mikayil Bekooglu, Kevin Trottmann

Das ist ein Eingabeformular für die Erfassung eines neuen Mieters. Die Formulare sehen alle ähnlich aus. Es ist darauf zu achten, dass alle Felder ausgefüllt werden. Mit dem Button «Erfassen» bestätigen Sie die Eingabe und die Daten werden in die Datenbank übernommen.

PHOTOCASEMieterRechnungenEinnahmenAbrechnungenLogout

Abrechnungen

Jahr der Abrechnung:

Abrechnung Drucken

Jahr der Abrechnung:

Heizkosten Drucken

Jahr der Abrechnung:

Nebenkosten Drucken

Photocase 2017 - Matej Djuranec, Mikayil Bekooglu, Kevin Trottmann

Die Abrechnungsseite beinhaltet drei Buttons – mit jedem kann die spezifische Abrechnung generiert werden. Das Dropdown definiert jeweils das Jahr der Abrechnung. Sobald die Buttons betätigt werden öffnet ein neuer Tab mit den entsprechend gewählten Daten.

pdfgeneratoreinnahmen.php1 / 1

Einnahmen

ID	Nachname	Vorname	Rechnungstext	Datum	Betrag in CHF
1	Tüscher	Patrick		2017-12-01	3,000.00
14	Trottmann	Kevin		2017-12-16	23,000.00
16	Djuranec	Matej		2017-11-20	15,550.25
17	Trottmann	Kevin		2017-08-01	2,300.00
18	Tüscher	Patrick		2016-09-19	2,300.00
19	Roth	Marco		2016-11-04	1,700.00

Nachdem die Abrechnung geöffnet wurde, wird das entsprechende PDF im Browser angezeigt. Über den Druckbutton kann das PDF in Papierformat ausgegeben werden.

5 Fazit

Wir haben zuerst mit einer eigenen Aufgabenstellung begonnen. Am Anfang haben wir versucht die UseCases umzusetzen, mussten aber nach ein paar Wochen das Thema wechseln weil dieses zu komplex und zeitaufwendig war.

Mit dem neuen Thema konnten wir in kürzester Zeit Fortschritte verbuchen. Vieles war bereits umgesetzt durch die Vorarbeit, die wir für unser eigenes Thema aufgewendet haben. So konnten wir schnell ein Grundkonstrukt erstellen und mit den Verbesserungen beginnen.

Die Arbeit hat uns viel Spass bereitet, auch wenn wir immer wieder mal an gewissen Enden angestanden sind und nicht mehr weiter wussten. Unsere Herangehensweise wäre für ein nächstes Projekt sicher verbesserungswürdig, auch wenn wir zum Schluss erfolgreich waren.