

PROJEKT RAMÓN

1. Kurzbeschreibung

Es soll ein Formular erstellt werden mit dem Kunden sich per Eingabe von E-Mail-Adresse und Wunschpasswort mit Mindestanforderungen auf der Website registrieren bzw. einloggen können (gleiches Formular für Login und Register). Die Eingabe der übrigen Kundendaten soll im nachhinein in der Sektion „Account“ erfolgen. Die Kunden sollen in der Lage sein Ihre Daten nach dem Login jederzeit einzusehen und zu ändern. Fehlermeldungen bei der Formulareingabe sollen für jedes Feld unabhängig voneinander erfolgen. Erst nach Eingabe der erforderlichen Daten, soll der Zugriff auf den (fiktiven) Service möglich sein (=Pflichtfelder). Valide Daten sollen gespeichert werden auch wenn das Formular nicht vollständig ausgefüllt wurde.

2. Sitemap

- > Login / Register
- > Account
- > Service

3. Skizze der Dynamischen Seiten

Service Account **Login**

Login / Sign Up

Email *

Password *

SUBMIT

Service **Account** Login

Account

Email

Phone *

Company Name *

Address Line 1 *

Address Line 2

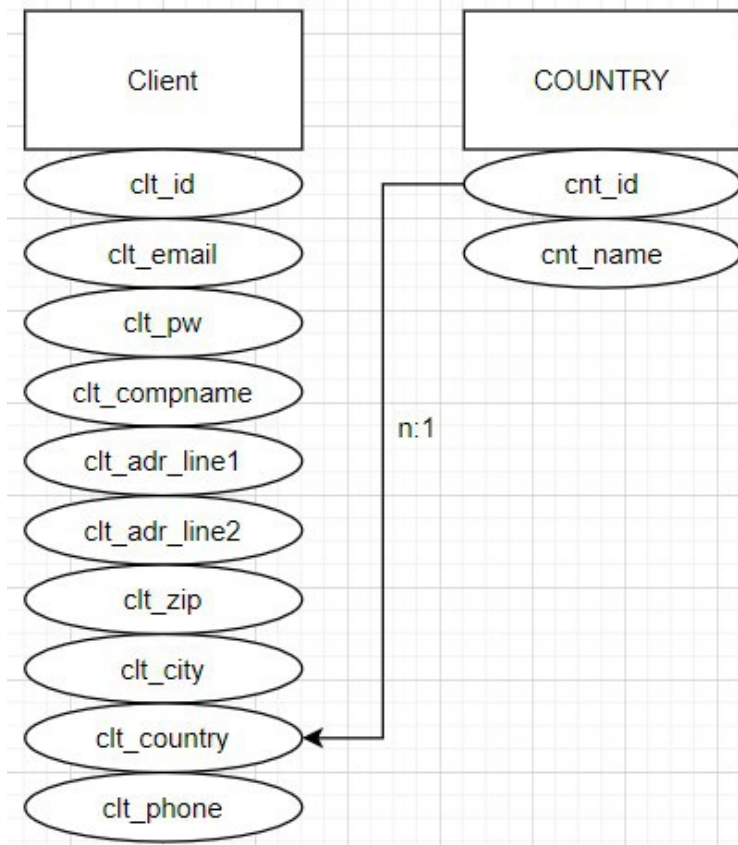
Zip Code *

City *

Country *

SAVE

4. Skizze des Datenbankentwurfs



5. Herausforderungen

- > Register und Login, wenn E-Mail-Adresse noch nicht in der Datenbank bzw. nur Login wenn E-Mail-Adresse schon in der Datenbank
- > Weiterleitung zur Seite „Account“ mit Textinfo „You have to add your company data before you can start using our service.“ bei Klick auf „Service“, wenn noch keine Unternehmensdaten eingetragen wurden.
- > Länderauswahl als Drop Down Menu
- > SQL Injections vermeiden, Sicherheit muss Gewährleistet sein
- > Password Encryption
- > Vollständige nicht manipulierbare Plausibilitätskontrolle und für Email, Country, Phone, Company Name
- > Doppeltprüfung für Email, Phone, Companyname, VAT,
- > Passwort (Sonderzeichen-, Capital-, Number- und Minimum Length-Anforderung),
- > Teilplausibilitätskontrolle für sonstige Daten
- > Session starten bei Login / Session beenden bei Logout
- > Session beenden wenn Browser geschlossen wird

- > Nichts was verarbeitet wird darf manipulierbar sein
- > Datenbank muss ordentlich sein und darf keine Redundanzen und Inkonsistenzen zulassen (Doppeltprüfung, ID-Inkrement, Pflichtfelder, Plausibilität, Beziehungen)

Optional:

- > Session beenden wenn User länger als 30 Minuten nicht aktiv (optional)
- > Ländervorwahl separat
- > Email-Domain-Blacklist in Datenbank und zur Plausibilitätskontrolle hinzufügen
- > Änderungen durch Eingabe des Passwortes bestätigen
- > Änderung des Passwortes

6. Zeitplan

Sonntag

- 6.0 Konzept Schreiben
- 6.1 Datenbank
- 6.2 HTML + CSS für Website und Unterseiten
- 6.3 Seitennavigation

Montag

- 6.4 Sub-Skriptseiten für Auslagerung
- 6.5 Globale Variablen Deklarieren (Eingabe, Verarbeitung, Speicherung, Abruf)

Dienstag

- 6.6 Masterseite Globale Logikstruktur konstruieren (Bedingte Blöcke)
- 6.7 Eingabe und Speicherung von Daten

Mittwoch

- 6.8 Verarbeitung, Plausibilitätskontrolle und Ausgabe von Daten

Donnerstag

- 6.9 Backtesting, Formelle Kontrolle, Codeästhetik
- 6.10 Optionale / Zusätzliche Funktionalitäten

Freitag

- 6.11 Präsentation Vorbereiten

7. Prinzipien

- > Jeden noch so kleinen Codeblock mit Kommentar bezeichnen
- > Sauber und Strukturiert Coden und Einrücken
- > Möglichst wenig hardcoden
- > Klar an Zeitablauf halten (Prioritäten setzen)
- > Am besten keine strukturellen Experimente
- > Klar an gelerntem orientieren und nochmal verinnerlichen

8. Hinweise

- > Personen des Kunden für dieses Projekt zu umfangreich, da es mehrere Ansprechpartner geben kann die verschiedene Funktionen, Kontaktdaten und Berechtigungen haben können (CEO, Rechnungsabteilung, Projektmanager, Bauleiter)
- > Vollständige Plausibilitätskontrolle für manche Daten zu Umfangreich für dieses Projekt, zB Adressdaten mit Google Maps API