# Overview

MyMosq membership project consists of three components:

* Membership management (what we currently have in MS Access)
  + add, remove and edit members
  + add, remove and edit payments
  + statistics, visualizations, queries etc
  + each user (editor or viewer) should be able to log-in, log-out, change the password and retrieve forgotton password
* Member app (what we currently have in db.moschee-wil.ch)
  + Each member should be able to log-in, log-out, change the password and retrieve forgotton password
  + Can view all details about his/her membership, including personal information, family members and payments
* Admin panel (similar to what we have in admin.mymosq.ch)
  + An admin shoud be able to log-in, log-out, change the password and retrieve forgotton password
  + An admin can view, create and remove users, and reset their passwords
  + Can assign roles to the users, currently viewer (only read) and editor (write) roles
  + Creates, views and edits the config file (not stored in the DB as table)

The superadmin manually creates the database server (if needed), the database (with all tables), and creates the credentials of at least one admin and one viewer to be used in the Member app.

# Tables

The following two tables will be used to store User and Member passwords:

* UserPassword
  + Email (as username)
  + Password (hashed)
* MemberPassword
  + Nr (membership ID)
  + Password (hashed)

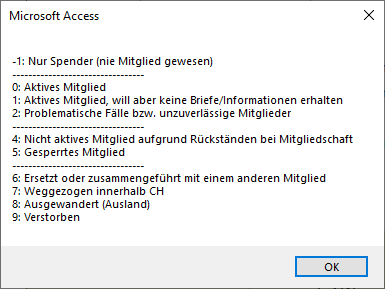
The following two tables will be used to store member’s and payments’ data

* “Person”
  + Nr (integer), unique, incremental, mandatory
  + Name, mandatory
  + Vorname, mandatory
  + Strasse
  + Plz (integer)
  + Ort
  + Land
  + Sprache
  + Beruf
  + Geburtsdatum (date)
  + Email
  + Telefon
  + Mobil
  + Beruf
  + H\_Herkunft\_Ort
  + H\_Herkunft\_Region
  + H\_Strasse
  + H\_Ort
  + H\_Region
  + H\_Land
  + Notiz (long text)
  + Passwort (hash)
  + Status (integer)
  + HeadMember (boolean, defines whether a normal member or a family member)
  + HeadMemberNr (integer, Nr of another Member)
  + HeadRelationship (string)
* “Zahlung”
  + ID (integer)
  + Person (linked to Nr of table Person)
  + Betrag (real/float)
  + Valuta (date)
  + Jahr (year)
  + Notiz (long text)
  + PaymentType (e.g. Mitgliedschaft, or ID of the payment in the config file)

# Config File

The config file is a json file which defines the structure of the data and the views of a particular mosque, e.g. Moschee-Wil config file:

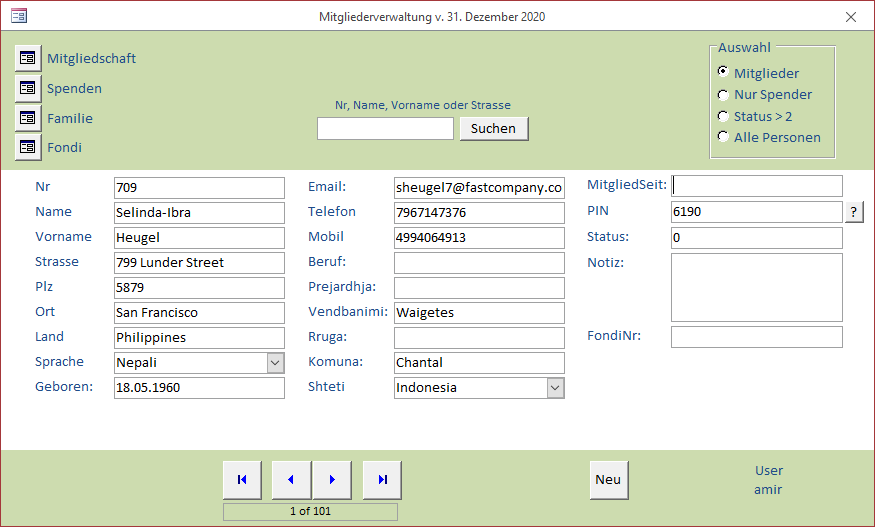
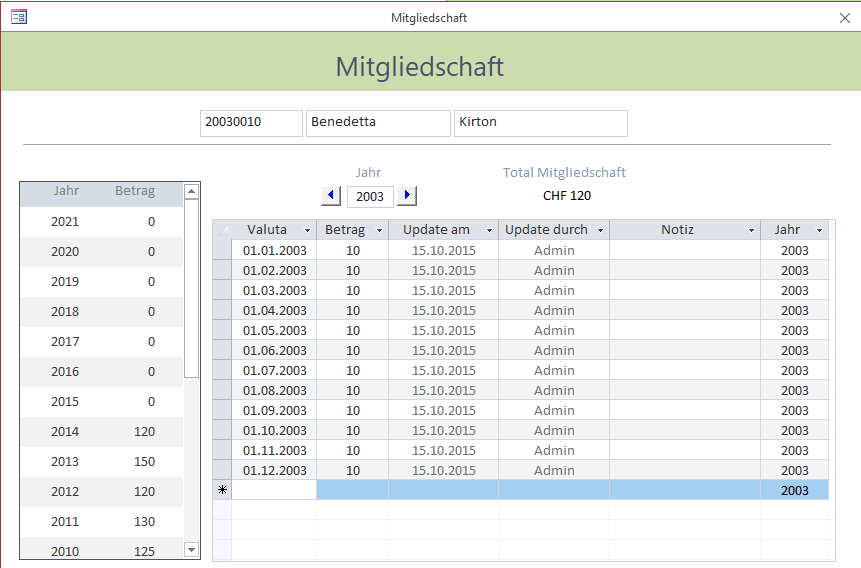
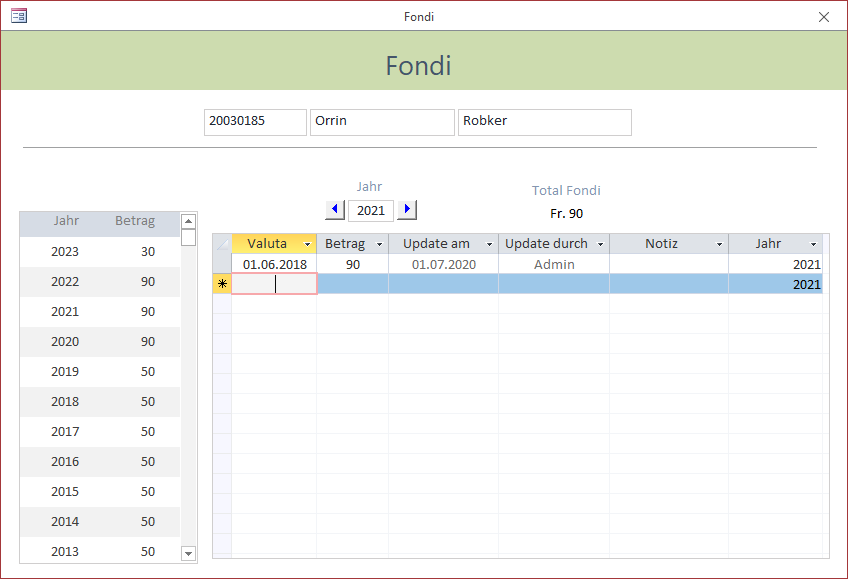
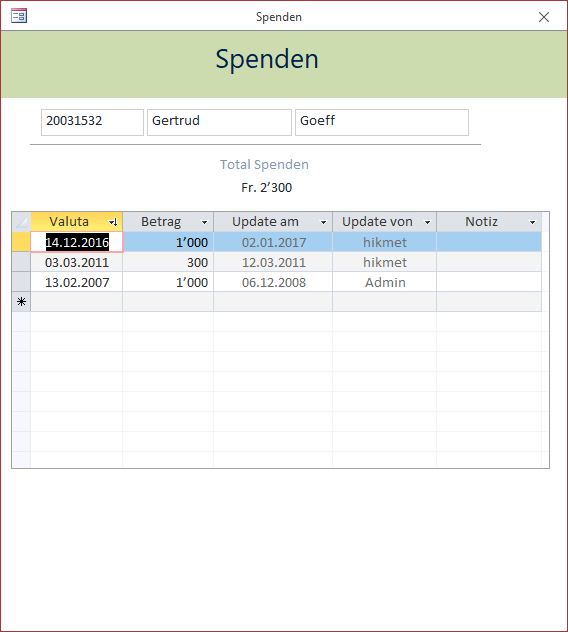
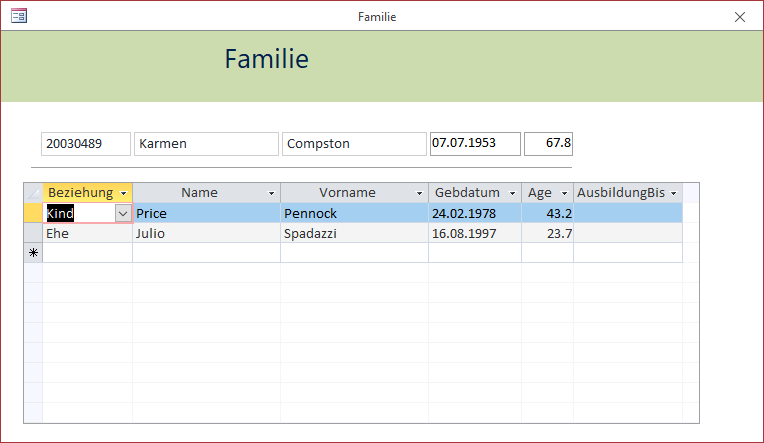
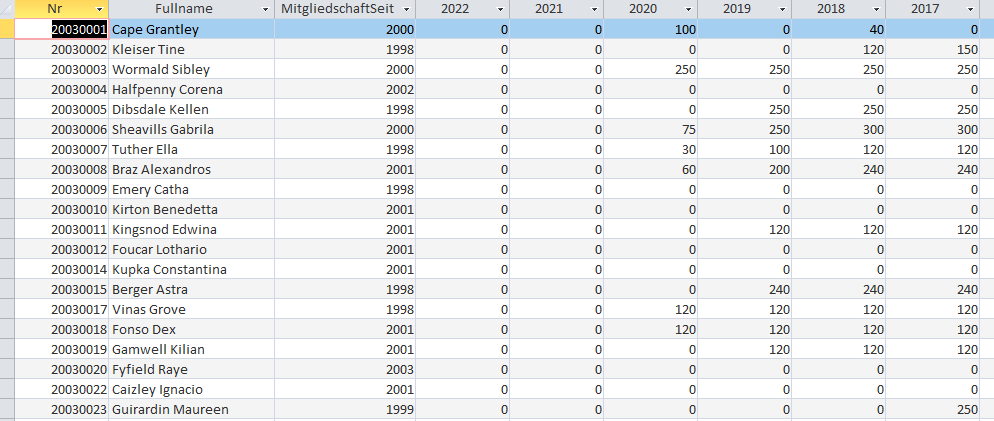
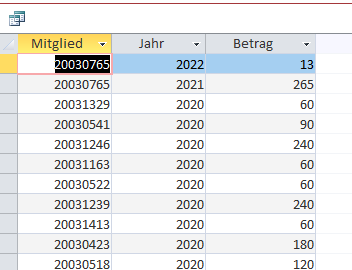
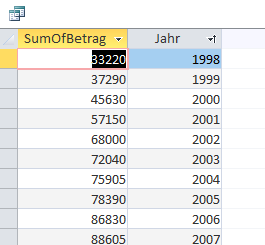
* Root
  + PaymentType
    - Payment
      * Name: “Mitgliedschaft”
      * Mode: “Yearly”
      * Minimum: 240
    - Payment
      * Name: “Spenden”
      * Mode: “Continuous”
    - Payment
      * Name: “Fondi”
      * Mode: “Yearly”
  + StatusMapping
    - Status
      * Code: -1
      * Description: “Nur Spender (nie Mitglied gewesen)“
    - Status
      * Code: 0
      * Description: “Aktives Mitglied“
    - ...
  + Email: “info@moschee-wil.ch”



# Member App

* Read the config file to understand the structure of the database
* Visualize components as outlined in the other document (convert app from .Net to JS)

# Membership Management App

* Read the config file to understand the structure of the database
* Members are all persons with HeadMember == true
* Entry page / Members Page:
  + List View: Show members as a long list (not all fields to show)
  + Member View: Show members individually including all fields (as in the picture below)
  + Search field (member’s fields to be searched dependent on the speed of search)
  + Filter by status
  + Navigation bars
  + Show list of payments tables (as defined in the config file), see left upper corner of the picture above
  + In both views: the abbility to add and remove members (dependent on the role)
* Payment page
  + Individual pages will be dynamically created based on the defined structure in the config file
  + Main information (Nr, Name, Surname) on top
  + Payments who have yearly mode
    - Summarized by year
    - Yearyl sums and navigation buttons
    - Clickable overview yearly table
    - List of payments for each year: editable, addable and removable single payments
    - Example 1: 
    - Example 2: 
  + Payments with continuous mode
    - Total sum
    - List of all payments: editable, addable and removable single payments
    - Example: 
* Family page
  + Family members are those users with HeadMember == false
  + HeadMember Nr, Name+Surname, Birthdate and age on top of page
  + List Family members with Name+Surname, Birthdate, Age and relationshsip to the head member
  + Ability to add, remove and edit family members
  + Example: 
* Exports (Excel) / Lists / Queries:
  + All queries to be stored as files in the cloud storage
  + Clear definitions of the input and output of the queries
  + Default queries
    - List of members including all fields based on the filters applied (status filter mainly)
    - Individual members including their family members
    - For each PaymentType
      * Yearly/total sums of individual members (as rows and columns) 
      * Yearly/total sums of all members 
  + Important queries from file and and show them in the same list as the above queries