# Photo Club Waalre – The App

# The Club

Map

Description automatically generated

*Waalre*

Photo Club Waalre is a photography club named after Waalre, a town in the south of The Netherlands. Since 1988 its club members meet to critique each other’s photos, organize excursions, and hold yearly photo expositions.

# Portfolios

A picture containing text, clock

Description automatically generated

App version 2.0.8 (build 4298)

The goal of this app is to showcase curated work of members of photo clubs.

The work is organized into ***portfolios***. Each portfolio covers that part of a photographer's work that was shared within a photo club. If a photographer is (or was) a member of more than one of the supported photo clubs, the app will show *multiple* portfolios for that photographer.

Each portfolio is shown in chronological order, with the latest work shown first, and typical spans multiple years.

We try to show the portfolios with **minimal distraction**. Web sites are often less optimal for this because they need to show browser controls (e.g., an address bar or tabs) and serve multiple purposes (e.g., show club news or contact information).

Originally the app only supported Photo Club Waalre (aka *Fotogroep Waalre*), but the current version supports **multiple** photo clubs. For a preview of what that looks like, drag down (“pull to refresh”) the screen of the Photo Club page. This loads a few random members from friendly neighboring clubs.

For Photo Club Waalre, we chose to also include the work of former club members. By default, former members are not visible, but you can configure the app to show them.

Contact us (app@FotogroepWaalre.nl) if you are interested in adding support for your photo club.

# Features and tips

Graphical user interface, text, application

Description automatically generated

*List of supported languages*

We regularly add features to the app. This requires some care, because we still want to maintain our focus on providing an optimal viewing experience. The app therefore explicitly does *not* aim to replace existing websites, but instead provides links to these websites.

The app supports the Dutch and English ***languages***. iOS automatically selects a language based on your device’s language settings (*Settings > General > Language & Region > Preferred Language Order*). iOS also allows you to overrule this per app (*Settings > Photo Club Waalre > Language*).

On the Portfolio page, you can swipe to ***Delete*** a member from the list. Don't worry - if the person is still a club member, that person will reappear the next time the app synchronizes its local data with online data. This swiping can be useful to remove members who are no longer on the online data members lists. In any case, swipe as much as you like (after all, it just might be therapeutical).

# The Animation

A picture containing text, sky, outdoor, sign

Description automatically generated

*© 2021 Greetje van Son*

When the app opens, it shows a large image corresponding to the app’s icon. If you tap somewhere ***inside*** the image, it zooms out to show the full image representing how digital cameras see color.

This involves a [Bayer color filter array](https://en.wikipedia.org/wiki/Bayer_filter) that filters the light per pixel. The filter array is shown here superimposed on a colorful photo.

Tapping ***inside*** the image allows you to zoom in or out to your heart's content. Tapping ***outside*** the image area ends the animation. You can trigger the animation again by restarting the app. A single tap ***outside*** the image allows you to skip the animation entirely.

# iOS, iPadOS, MacOS

A picture containing cup, indoor, glass, drink

Description automatically generated

*Apple*

The app runs on iPhone and iPad and requires iOS 15.0 or higher. iOS users tend to upgrade to the latest possible iOS version, which means that the app supports devices made in the last 5 or 6 years. The app also runs on the new generation of Macs with “Apple Silicon”.

The app uses various Apple software technologies including Swift (a programming language), SwiftUI (a user interface framework), Core Data (local data storage), Metal (graphics acceleration), concurrency (for fetching data), and animations.

The benefit of an iOS focus is that it allows the developers to focus on functionality rather than cross-platform and cross-version interoperability issues.

The drawback, however, is the lack of Android support, because the Android universe has its own set of software technologies. Apps that choose to support multiple platforms require more development effort and often contain more technical compromises.

There is another perspective on not targeting Android anytime soon: it implies the app cannot contain any essentially functionality for which there is no alternative. So, again, the app cannot replace a club's website.

To put the app's device support into perspective, consider the top-6 software environments. The top-6 platforms are, in [largest-first order](https://www.youtube.com/watch?v=7S9PBgS8HS8): web, Android, Windows, iOS, iPadOS, and MacOS. MacOS's market share is the smallest of the six but is still larger than all smaller platforms combined. Only the very largest apps (like MS Word, Twitter, Minecraft, Chrome) manage to cover the full range. But many well-known apps choose to focus their resources: Photoshop (Windows, MacOS, iPadOS), WhatsApp (iOS, Android, web - kind of), Pokémon Go (Android, iOS, iPadOS - kind of). So, covering three of the top-six is a decent enough score for now.

# Developers wanted

Text

Description automatically generated

*Fragment of the Swift source code*

We expect to put the app on GitHub to enable other software developers to contribute. Getting more developers speeds things up somewhat, but also gives fresh ideas and promotes continuity.

Welcome contributions include adding features, code improvements, ideas on architecture and interface definition, and possibly even a backend.

Contributions that don't require coding are also very welcome: beta testing via TestFlight, feature requests, translations, SVG icon design, and maybe UI/UX design.

A central design challenge for a next stage will be to provide a clean, standardized interface to retrieve data per photo club. The interface is needed to load the data, but also keeps the data within the app up to date. After all, membership data and portfolios change regularly. The current interface is essentially a plug-in design with an adaptor per photo club. This needs to be replaced by a standard data interface to avoid having to extend the source code whenever a new club comes aboard.

The app currently uses a software module per club. That module loads membership and portfolio data from the club’s server and merges it into the in-app database. For Photo Club Waalre, the membership data is read from a HTML table on a password protected part of the club’s [website](https://www.fotogroepwaalre.nl/). The portfolios use a somewhat more robust solution: they are read from XML files generated by a Lightroom Web plug-in called [JuiceBox-Pro](https://www.juicebox.net/tour/) Thus portfolios are created and managed as Lightroom *collections*. These collections are then uploaded to the webserver with a single *Upload* click (thus triggering JuiceBox-Pro) where they can be downloaded by the app.

Software developers or other volunteers who are willing to help can contact us via appdev@FotogroepWaalre.nl. Regarding the required level of development experience: you should be good to go if you have created a complete SwiftUI app before. Alternatively, an experienced developer who is learning SwiftUI would also fit the bill. Recommended courses for learning SwiftUI are Hacking with Swift's practical [100 days of SwiftUI](https://www.hackingwithswift.com/100/swiftui/)) or Stanford's fast paced and more academic online [CS193p course](https://cs193p.sites.stanford.edu/).

# The Model

Diagram

Description automatically generated

*Partial model*

As an intro for developers, here is an informal tour of the ***model***. The model is a [schema](https://en.wikipedia.org/wiki/Database_schema) or information model in the database sense, or “Model“ in the sense of the [Model-View-ViewModel](https://www.hackingwithswift.com/books/ios-swiftui/introducing-mvvm-into-your-swiftui-project) design pattern. The app uses a Swift-style MVVM design, meaning model data is stored in structs rather than in classes.

The model: every **PhotoClub** has **Members** of various types (current, former, etc.), while some Members have a formal role like chairman within the PhotoClub. Some of **Photographer**'s data (like name and birthday) are unrelated to being a Member of a PhotoClub. **Portfolios** represent the body of work of a Photographer in the context of a PhotoClub. A Portfolio contains **Images**, but an Image could be in multiple Portfolios depending on where the Image was shared. Member and Portfolio can technically be considered synonyms from a modeling perspective: we create exactly one Portfolio for each PhotoClub that a Photographer joined. They are thus a single table in the database.