MyMopsi: Managing location-based data collections

Juhani Järviö

UNIVERSITY OF
EASTERN FINLAND

Master's Thesis

University of Eastern Finland

School of Computing

December 2022

Abstract

Geo-tagged photo collections are increasing by the minute since mobile devices started to incorporate GPS sensors and cameras with decent photo quality. We begin the thesis with a review of existing tools for handling such collections and their features. Then we present MyMopsi, a tool developed by us to support the most essential features. MyMopsi applies state of the art methods for data security and privacy, server load balancing and enhanced interface that supports an interactive map and decluttering by using clustering.

Keywords: Geo-tagged, Photos, Collections, MyMopsi

University of Eastern Finland, Faculty of Science and Forestry, Joensuu

School of Computing

Computer Science

Juhani Järviö: MyMopsi: Managing location-based data collections

Master’s Thesis, 51 pages

Supervisor: Professor Pasi Fränti

December 2022

Foreword

Well, it only took 9 years, 4 months, and some number of days to get to this point but I finally managed to do it. Almost five years of that merely writing this document.

I would like to thank Professor Pasi Fränti for his guidance (and especially patience) during the writing of this thesis; I still do not quite understand how you have tolerated me for this long. I am also grateful for Doctor Radu Mariescu-Istodor for his help on the thesis and on the more practical side of the IT-project. I'd also like to thank Doctor Mikko Malinen for checking the final document.

In addition, I'd also like to extend my gratitude to the teachers at UEF. Without their teachings I wouldn't be here today. From a class of students trying to instruct a teacher how to open a bag of bread, to detecting glass bottles with machine learning and building robots, through three years of attempting to pass mandatory Swedish, and the one English and translation course where I was actually interested in academic writing; I doubt I'll forget my time studying computing, statistics, English, or economics anytime soon.

Finally, to my family and friends: you may now stop asking how my thesis (either of them) is going. It's finished. Concluded. Terminated. Over and done with. Valmis. Färdig. 完成した.

And hopefully I never have to write or see the word "thesis" ever again in my life. I'm a programmer, not a writer dammit!

2022-12-01 Juhani Järviö, FM

Abbreviations

I.E.	Latin for “Id Est”; it means “In Essence”
E.G.	Latin for “Exempli Gratia”; it means “for example”
GPS	Global Positioning System, a satellite-based radionavigation system owned by the US government
JPEG	A common method of lossy image compression for photos. Acronym for “Joint Photographic Experts Group”, the creators of the standard
iOS	Mobile operating system (OS) created by Apple Inc.
UEF	University of Eastern Finland
PC	Personal Computer
WYSIWYG	What You See Is What You Get, a form of editor. E.g., Microsoft Word
Exif	Exchangeable image file format. Technically JPEG is the compression, and Exif is the format
UI	User Interface; commonly paired with UX (User Experience)
URL	Uniform Resource Locator, colloquially known as a web address
EU	European Union
GDPR	EU’s General Data Protection Regulation; a regulation in EU law on data protection and privacy, and the reason why you see those cookie pop-ups on all websites.
PNG	Portable Network Graphics. Common image format, best for solid colour images, like logos, bad for photos
GIF	Graphics Interchange Format. Common (and very old) image format, used commonly for short and small videos. Rapidly being replaced by other better formats.
IPTC	International Press Telecommunications Council, a consortium of the world's major news agencies
HEIF	High Efficiency Image File Format, a proprietary format for photos
MB	Megabyte, a measure of file size
MIME	Multipurpose Internet Mail Extensions, a standard commonly used to indicate what type or format of media is being handled
SHA-1	Secure Hash Algorithm 1, a cryptographically broken but still widely used hash function
WebP	Image file format created by Google, intended to replace other common formats
PHP	A general-purpose scripting language geared toward web development, specifically on the server-side. PHP originally stood for “Personal Home

	Page”, but it now stands for the recursive initialism “PHP: Hypertext Pre-processor”
SVG	Scalable Vector Graphics, an XML-based vector graphics format, as opposed to a raster (or pixel) based format like JPEG
XML	Extensible Markup Language, a markup language and file format for storing and handling arbitrary data
XSS	Cross-site scripting, a type of security vulnerability that can be found in some web applications

Index

1 Introduction

With the rise of smart phones with cameras and GPS capability people can take geotagged photos easily. Currently one third of the population has smartphones and the number is expected only to climb. As smartphones have as standard features a camera and GPS capability, with ability to embed location data, this means that at least 3 billion people can take geo-tagged photos. In addition, in 2018 over 50 % of internet website traffic was through mobile devices. [1] [2] [3] [4]

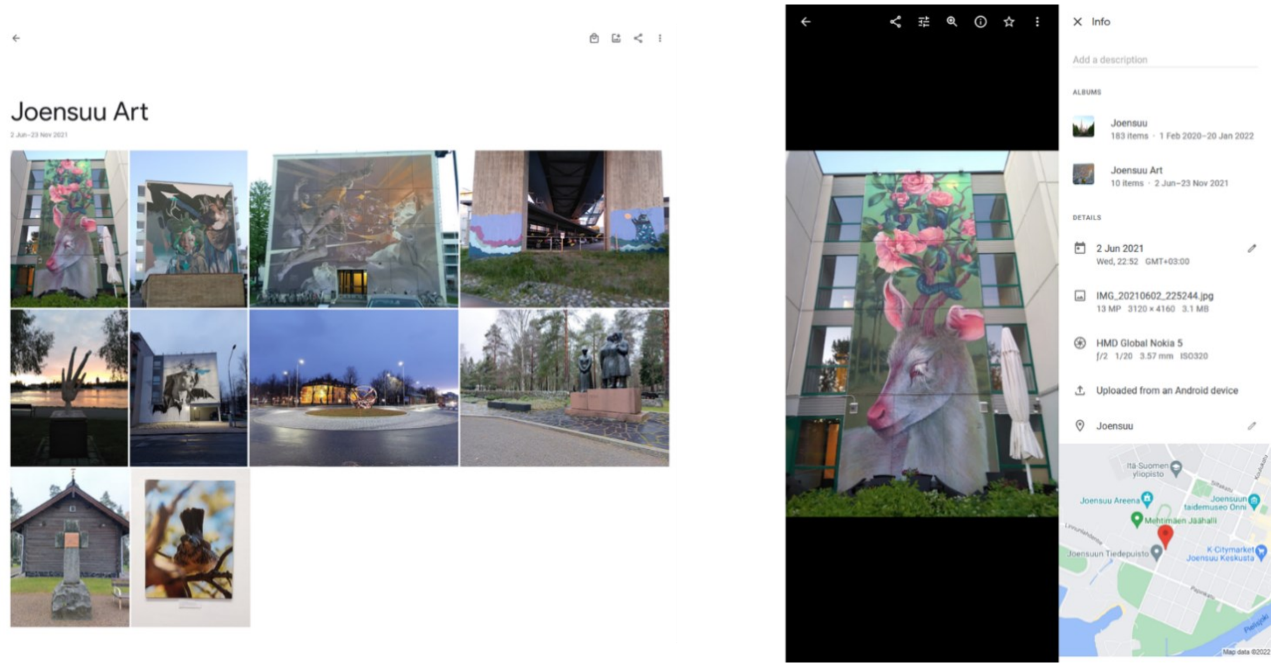
Even though geo-tagged photo collections are growing, few services allow users to view their collections on an interactive map. In fact, many media sharing services strip uploaded images of their metadata, most likely for privacy reasons, so most sites that allow content upload do not have this feature by design. For example, social media sites remove location metadata from uploaded images when uploading (in some cases, showing location is still possible, as the information is stored elsewhere). *Google Photos* is one of the services that allow to download the original image with metadata, but do not provide a map view for whole albums (see Figure 1).

[1] How Many People Have Smartphones Worldwide (Mar 2022) (bankmycell.com) | https://www.bankmycell.com/blog/how-many-phones-are-in-the-world

[2] Smartphones killing point-and-shoots, now take almost 1/3 of photos - Gigaom | https://gigaom.com/2011/12/22/smartphones-killing-point-and-shoots-now-take-almost-13-of-photos/

[3] Smartphone Sales Taking Toll on G.P.S. Devices - The New York Times (nytimes.com) | https://www.nytimes.com/2010/11/15/technology/15iht-navigate.html

[4] What Percentage of Internet Traffic Is Mobile in 2021? (hostingtribunal.com) | https://hostingtribunal.com/blog/mobile-percentage-of-traffic/



Joensuu Art

2 Jun–23 Nov 2021

Info

Add a description

ALBUMS

Joensuu
183 items · 1 Feb 2020–20 Jan 2022

Joensuu Art
10 items · 2 Jun–23 Nov 2021

DETAILS

2 Jun 2021
Wed, 22:52 GMT+03:00

IMG_20210602_225244.jpg
13 MP 3120 × 4160 3.1 MB

HMD Global Nokia 5
ƒ/2 1/20 3.57 mm ISO320

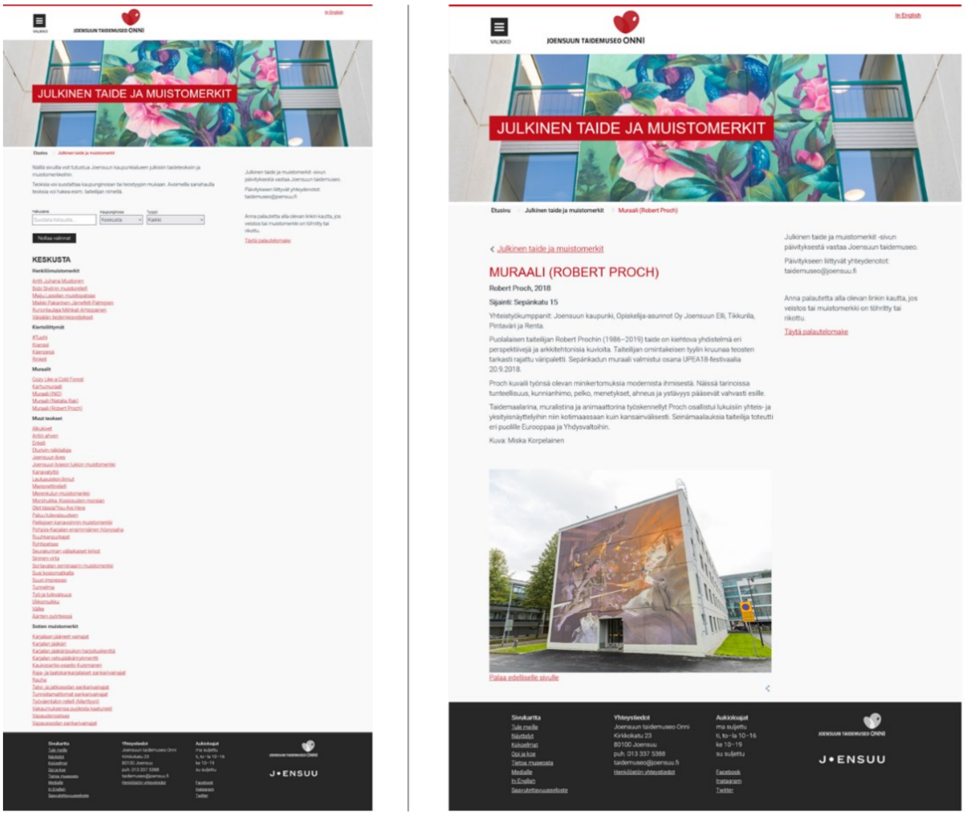
Uploaded from an Android device

Joensuu

Figure 1 – Collection view in Google Photos (left), with no map view in sight. Individual images can be viewed on a map, along with other metadata (right)

An ability to show photos on an interactive map would also be useful for commercial or public purposes. Take for example Joensuu's public art displays and memorials, managed by Joensuu Art Museum. [5] See Figure 2 for a picture of their website. As it is, you can only view an alphabetical list of the public works of art in Joensuu with no map view, even though this would be an ideal use case for an interactive map. It would be much more useful for a tourist interested in seeing the locations to view them on a map, but the individual listings only have the address. None of the other services explored in the following Section 2 however offer any idea way of managing this, and it would require building one from scratch by profession software developers. A public collection like this could even be used for recommendations for other users (Waga, Tabarcea, & Fränti, 2012). (Fränti, Chen, & Tabarcea, 2011) points to location being location being one of the four most significant identifiers of relevance when recommending and sharing media content online.

[5] Julkinen taide ja muistomerkit - joensuuntaidemuseo.fi | https://joensuuntaidemuseo.fi/julkinen-taide | Page only in Finnish



JOENSUUN TAIDEMUSEO ONNI

In English

JULKINEN TAIDE JA MUISTOMERKIT

Etusivu > Julkinen taide ja muistomerkit

Näillä sivuilla voit tutustua Joensuun kaupunkialueen julkisiin taideteoksiin ja muistomerkkeihin.

Teoksia voi suodattaa kaupunginosan tai teostyypin mukaan. Avoimella sanahaulla teoksia voi hakea esim. taiteilijan nimellä.

Hakusana: Suodata listausta... Kaupunginosa: Keskusta Tyyppi: Kaikki

Nollaa valinnat

KESKUSTA

Henkilömuistomerkit

Antti Juhana Mustonen
Bobi Sivénin muistolaatta
Maiju Lassilan muistopatsas
Maikki Pakarinen Järnefelt-Palmgren
Runonlaulaja Mihkali Arhippainen
Väisälän tiedemiesveljekset

Kiertoliittymät

#Tuohi
Kransi
Käenpesä
Rinkeli

Muraalit

Cozy Like a Cold Forest
Karhumuraali
Muraali (INO)
Muraali (Natalia Rak)
Muraali (Robert Proch)

Muut teokset

Aikukset
Antin ahven
Enkeli
Eturivin näköaloja
Joensuun Aves
Joensuun lyseon lukion muistomerkki
Kanavatyttö
Laulupuiston lirinat
Marionettiteatteri
Merenkulun muistomerkki
Morsiuhukka, Kosisaudon morsian
Olet tässä/You Are Here

Julkinen taide ja muistomerkit -sivun päivityksestä vastaa Joensuun taidemuseo.

Päivitykseen liittyvät yhteydenotot: taidemuseo@joensuu.fi

Anna palautetta alla olevan linkin kautta, jos veistos tai muistomerkki on töhritty tai rikottu.

Täytä palautelomake

JOENSUUN TAIDEMUSEO ONNI

In English

JULKINEN TAIDE JA MUISTOMERKIT

Etusivu > Julkinen taide ja muistomerkit > Muraali (Robert Proch)

< Julkinen taide ja muistomerkit

MURAALI (ROBERT PROCH)

Robert Proch, 2018

Sijainti: Sepänkatu 15

Yhteistyökumppanit: Joensuun kaupunki, Opiskelija-asunnot Oy Joensuun Elli, Tikkurila, Pintaväri ja Renta.

Puolalaisen taiteilijan Robert Prochin (1986–2019) taide on kiehtova yhdistelmä eri perspektiivejä ja arkkitehtonisia kuvioita. Taiteilijan omintakeisen tyylin kruunaa teosten tarkasti rajattu väripaletti. Sepänkadun muraali valmistui osana UPEA18-festivaalia 20.9.2018.

Proch kuvaili työnsä olevan minikertomuksia modernista ihmisestä. Näissä tarinoissa tunteellisuus, kunnianhimo, pelko, menetykset, ahneus ja ystävyys pääsevät vahvasti esille.

Taidemaalarina, muralistina ja animaattorina työskennellyt Proch osallistui lukuisiin yhteis- ja yksityisnäyttelyihin niin kotimaassaan kuin kansainvälisesti. Seinämaalauksia taiteilija toteutti eri puolille Eurooppaa ja Yhdysvaltoihin.

Kuva: Miska Korpelainen

Palaa edelliselle sivulle

Julkinen taide ja muistomerkit -sivun päivityksestä vastaa Joensuun taidemuseo.

Päivitykseen liittyvät yhteydenotot: taidemuseo@joensuu.fi

Anna palautetta alla olevan linkin kautta, jos veistos tai muistomerkki on töhritty tai rikottu.

Täytä palautelomake

Sivukartta
Tule meille
Näyttelyt
Kokoelmat
Opi ja koe
Tietoa museosta
Medialle
In English
Saavutettavuusseloste

Yhteystiedot
Joensuun taidemuseo Onni
Kirkkokatu 23
80100 Joensuu
puh. 013 337 5388
taidemuseo@joensuu.fi
Henkilöstön yhteystiedot

Aukioloajat
ma suljettu
ti, to–la 10–16
ke 10–19
su suljettu

Facebook
Instagram
Twitter

JOENSUUN TAIDEMUSEO ONNI

J•ENSUU

Figure 2 – Joensuu Art Museum's website, where they list public works of art in Joensuu

(Nikulin & Buchko, 2020) describes several Information Retrieval Multimedia Systems, and compares them, including Google Photos and Apple Photos which are also looked in this thesis (though from different angle), along with pointing out some specific end-user needs, and calling for technical solution. This provides motivation for MyMopsi.

MyMopsi is a web-based service created to solve these problems. With it anyone can upload and manage image collections, which are viewable on a dynamic map, with clustering. Anyone can check individual image metadata (including location on a map) or create an account to manage larger collections. Public collections can be shared to other people, including anyone without an account. MyMopsi also enables adding or changing a location to an image. As privacy is important, any collection can be set to private, in which case only the owner can view it or any images it contains.

MyMopsi is an extension to Mopsi, a location-based social network developed by the Machine Learning Group, School of Computing at the University of Eastern where users can share their location, take geo-tagged photos, record routes, and get recommendations in the region. Today (November 2022), Mopsi contains over 65,000 geo-tagged photos total and MyMopsi helps to manage large collections such as the ones in Mopsi. It is also where the service gets its name. One of the key features of MyMopsi is the clustering system created for Mopsi, which allows dynamic, real-time clustering of large number of datapoints.

The thesis structure is as follows: Section 1 is a short introduction to the topic, which you are reading right now. Section 2 chronicles other related and/or competing services, with a focus on a select few key features that I think are important in MyMopsi. I go through the services listing which features they have. In section 3 I take these key features and go through them in MyMopsi, and how I have implemented them there, in surface detail, along with some other important aspects of the practical side of the project.

Section 4 contains possible future improvements to MyMopsi that could improve it but were out of the scope of this project. And finally, the last and fifth section contains final conclusions and opinion.

2 Feature comparison between competing and related services

In this section I will review some select competing or related services and look at some key features that I think are important and compare them between services. At the end of the section, I will summarize the results of the comparison. Features have been selected based on my subjective opinions and experience. List of features that I will focus on:

- **Account**: is one required or needed. An account can be used to save information or data between visits, enabling users to save collections for later use. Not requiring an account does enable easier and faster use for singular visits.
- **Desktop & mobile view**: can the services be used on multiple different devices. In today's smartphone centric world, it is important to have a usable mobile version (Djamasbi, et al., 2014). In most cases having a dedicated mobile app would be best for end-user but requires more work and ongoing maintenance. Because in all cases the services have a desktop view, this point is mostly about the existence of a mobile view.
- **Collections**: ability to group images into collections or albums. In this thesis I will mostly use the term "collection" even if the service uses "album", since in most cases they are interchangeable. (Torniai, Battle, & Cayzer, 2007)
- **Map view**: ability to view an image or images on a map. Key aspect for easy visualisation of large collections of photos. (Jaffe, Naaman, Tassa, & Davis, 2006)
- **Clustering**: if service has a map view for whole collections, does it have clustering?
- **Image formats supported**: the most common format is JPEG, and has been for a few decades now, but I do not wish to limit myself to such artificial boundaries.
- **Metadata handling**: does the service keep image metadata, and if so, allow to view it afterwards. All images – or computer files really – have metadata attached to them, embedded inside the file itself. In the case of a JPEG photo taken with a camera, this may include (but is not mandatory) information on camera (e.g., model name), information on author, when image was taken, and location (e.g., GPS coordinate, height, and/or specific address). This metadata can be also added later manually.
- **Monetization**: how is the service making money, and thus inconveniencing the user. This point is only for curiosity and completeness's sake, as *MyMopsi* will not have any monetisation involved, so

the comparison does not have any actual bearing on the technical side. Although it is good to keep in mind that nothing is free, and every extra feature has a cost.

Services that I will be looking at are:

- *Mopsi*
- *Google Photos* and *Google*'s *My Maps*
- *Apple Photos*
- *Pic2Map* (and other similar services, of which there are multiple)
- *Flickr*
- *Imgur*
- Social Media sites: *Twitter*, *Facebook*, and *Instagram*.

It is also possible to make your own solution, by combining server-side software like *GeoServer* and using *Leaflet* for managing client-side application or using *Google Maps* or *OpenStreetMap* (*OSM*). This enables you to create a complete custom solution for a user's specific needs, but it also requires skills and knowledge to create the app and is not comparable to what I am trying to do with *MyMopsi* (to allow easy access to anyone managing large photo collections for end-user). In fact, you would be creating *MyMopsi* yourself at that point.

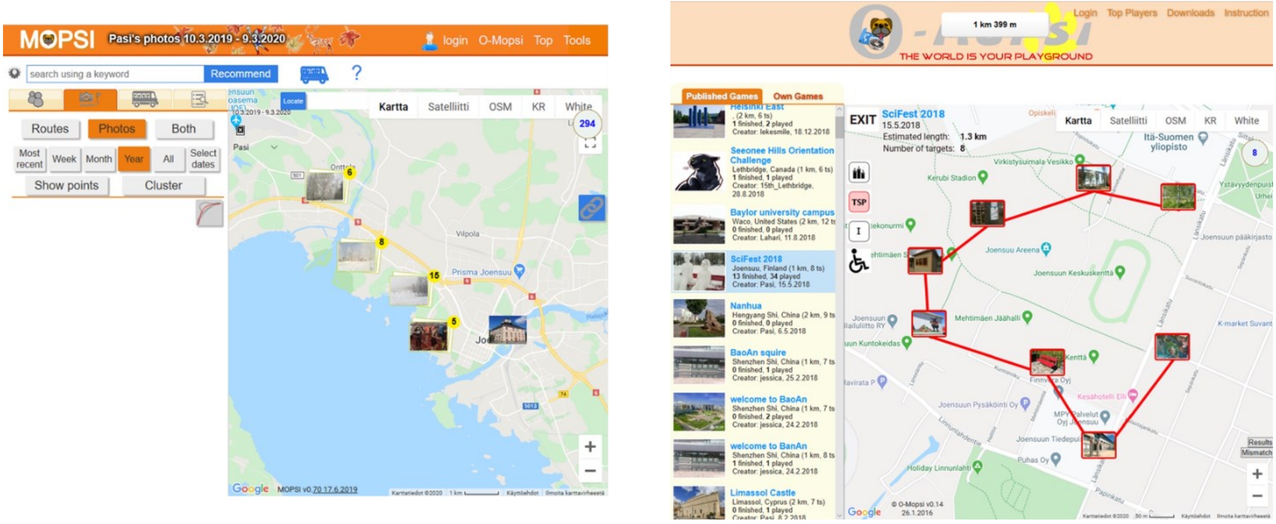
2.1 Mopsi

Mopsi [6] is a social network that helps people to discover who and what is around them. Its features include photo sharing, live tracking and chatting with friends. *Mopsi* was developed by the Machine Learning Group, School of Computing at the University of Eastern Finland. It provides location-based services, such as search, recommendation, route tracking, geo-tagged photo collection and bus schedules. *Mopsi* has been used for route management and visualization, route search, transport mode detection research, to name a few.

Related to *Mopsi* is *O-Mopsi*, a location-based orienteering game built for *Mopsi* users, using public images from Mopsi users, presented in more detail in (Fränti, Mariescu-Istodor, & Sengupta, O-Mopsi: Mobile Orienteering Game for Sightseeing, Exercising, and Education, 2017). Goal of the game is to visit several real-world locations, using a smartphone with GPS-capability, in fastest time

[6] Mopsi Project - University of Eastern Finland (uef.fi) | http://cs.uef.fi/mopsi/

and shortest distance. This is an example of traveling salesman problem, where the goal is to visit all cities in a given list, when distances are known and each city is visited only once. [7] In *O-Mopsi* games are made by users, and each point is marked by a photo with a GPS coordinate. The main challenge is finding the targets and deciding the order in in which to attach the targets to get least distance travelled in fastest time. The games present a mental challenge as well as out-doors exercise which bring positive health benefits. See Figure 3 for screenshot of both *Mopsi* (left) and *O-Mopsi* (right).



Helsinki East
, (2 km, 6 ts)
1 finished, 2 played
Creator: lekesmile, 18.12.2018

Seeonee Hills Orientation Challenge
Lethbridge, Canada (1 km, 6 ts)
1 finished, 1 played
Creator: 15th_Lethbridge, 28.8.2018

Baylor university campus
Waco, United States (2 km, 12 ts
0 finished, 0 played
Creator: Lahari, 11.8.2018

SciFest 2018
Joensuu, Finland (1 km, 8 ts)
13 finished, 34 played
Creator: Pasi, 15.5.2018

Nanhua
Hengyang Shi, China (2 km, 9 ts
0 finished, 0 played
Creator: Pasi, 6.5.2018

BaoAn squire
Shenzhen Shi, China (1 km, 7 ts
0 finished, 1 played
Creator: jessica, 25.2.2018

welcome to BaoAn
Shenzhen Shi, China (1 km, 7 ts
0 finished, 2 played
Creator: jessica, 24.2.2018

welcome to BanAn
Shenzhen Shi, China (1 km, 8 ts
1 finished, 1 played
Creator: jessica, 24.2.2018

Limassol Castle
Limassol, Cyprus (2 km, 7 ts)
0 finished, 1 played
Creator: Pasi, 8.2.2018

SciFest 2018
15.5.2018
Estimated length: 1.3 km
Number of targets: 8

Figure 3 – Mopsi UI (left), and O-Mopsi UI (right)

As for features, an account is required handling own photos, but anyone can view them. Site is mainly built for desktop, but there was a mobile app developed for Windows Phones, Android, and iOS. However, it is unsupported now and unavailable due to lack of maintenance. Difficult to say if having the mobile handling done through a browser would it have survived to this day, but mobile browsers have improved in the past decade, so separate mobile apps probably made more sense when they were made.

There is no ability to group photos into collections, but you can show all of them on a map or select a range of dates for photos to show on an interactive timeline. The map view also has a clustering, developed specifically for *Mopsi*, by the Machine Learning Group at UEF. This clustering system is

[7] Travelling salesman problem - Wikipedia | https://en.wikipedia.org/wiki/Travelling_salesman_problem

also used in *MyMopsi* and is presented in more detail in a further chapter. See Table 1 for summary of features.

Table 1 – Mopsi features

Account	Mobile view	Collections	Clustering	Map	Formats	Keep metadata	View metadata	Monetisation
✓	✗	✓	✓	✓	JPEG	✓	✓	N/A

2.2 Google services

Google Photos [8] is a photo sharing and storage service developed by *Google*, mainly for Android phones for handling photos taken on device. They also allow users to store photos on the cloud and viewing them on a browser with limited free storage provided by *Google*. [9] As it is made by *Google*, it is very highly integrated with Android operating system, and such has a lot of features that work directly with the camera, but its main purpose is to work as a gallery for photos taken on a phone. Photos can be automatically uploaded to online cloud, either original size or compressed, with original metadata stored. The service also can analyse photos to detect places or people using machine intelligence. See following Table 2 for summary of features.

Table 2 – Google Photos features

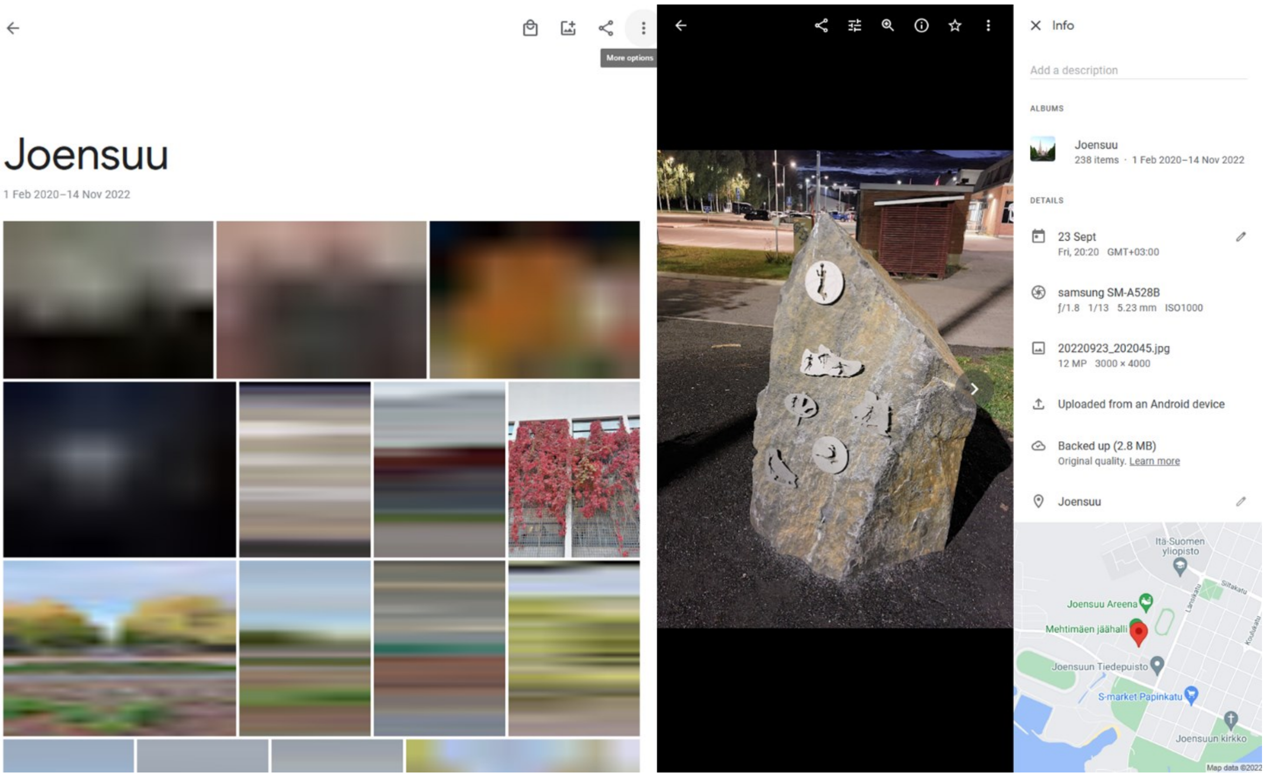
Account	Mobile view	Collections	Clustering	Map	Formats	Keep metadata	View metadata	Monetisation
✓	✓	✓	✗	✓	Most common	✓	✓	Ads / Subscription

Google Photos does require an account to use, but it is not locked to hardware; uploading via browser is possible. Photos can be arranged into collections, and it will even automatically suggest grouping photos together (based on for example date and/or location). There is a simple map view

[8] Google Photos | https://www.google.com/photos/about/

[9] Upload the Pictures, and Let Google Photos Do the Rest - The New York Times (nytimes.com) | https://www.nytimes.com/2015/06/04/technology/personaltech/upload-the-pictures-and-let-google-photos-do-the-rest.html

for individual images to see the location, but there is no way to see whole collections on a map and thus no clustering. As a sidenote, *Google Photos* mainly tries to use location data saved on image metadata, but if that is not available, it can estimate a location based on time taken and device location history. As for metadata, none of it is removed either on device or after upload to cloud (though it is possible to securely share photos without metadata), and user can view some basic metadata on *Google Photos* (mainly when and where photo was taken). See Figure 4 – Screenshot of Google Photos album view, and individual image metadata.Figure 4 below for a screenshot of the album view, and individual image with metadata.



Joensuu

1 Feb 2020–14 Nov 2022

Info

Add a description

ALBUMS

Joensuu
238 items · 1 Feb 2020–14 Nov 2022

DETAILS

23 Sept
Fri, 20:20 GMT+03:00

samsung SM-A528B
ƒ/1.8 1/13 5.23 mm ISO1000

20220923_202045.jpg
12 MP 3000 × 4000

Uploaded from an Android device

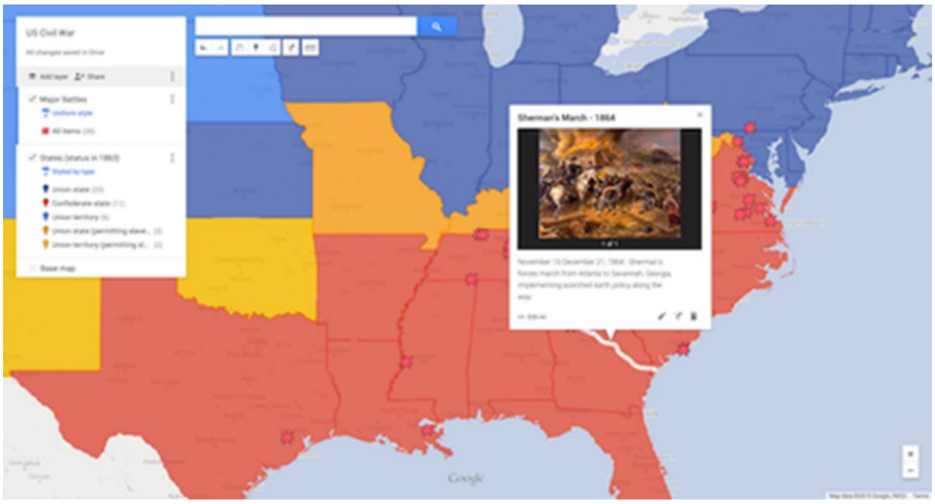
Backed up (2.8 MB)
Original quality. Learn more

Joensuu

Figure 4 – Screenshot of Google Photos album view, and individual image metadata.

When uploading images manually, most common formats are recognized and supported. Despite the name, Android will even suggest saving “memes” saved locally to the cloud, so really, it’s any images the user is handling on a phone. Monetisation is through ads (as is most of *Google* services, advertising and personal data being their main business), and an optional subscription for more storage space. *Google Photos* is not locked to Android, as it works on PCs via a browser (or through Google Drive) and can even be installed on Apple devices. Which makes sense, Google is not a hardware business, they deal in software and advertising. They want to get as many people using their services as possible, to get more data which is their main means of generating income. (Foerderer, Kude, Mithas, & Heinzl, 2018)

Google also has a *My Maps* [10] service, which allows you to make your own maps in a WYSIWYG [11] web-interface, viewable by anyone and which can be embedded on a webpage. A user can add not only images but also points, lines, and drawn shapes, as well as notes, which can then be shared to other people. The maps are mostly static, and for singular purposes, for example showing regions in coloured overlayed shapes and historic battles (see Figure 5). Images can be only imported from *Google Photos*, but *My Maps* does have support for all images that it supports.



November 15-December 21, 1864: Sherman's forces march from Atlanta to Savannah, Georgia, implementing scorched earth policy along the way.

Figure 5 – An example map created in Google's My Maps

Table 3 – Google's My Maps features

Account	Mobile view	Collections	Clustering	Map	Formats	Keep metadata	View Metadata	Monetisation
✓	✗	✓	✗	✓	Google Photos	✓	✗	Ads

[10] My Maps – About – Google Maps | https://www.google.com/maps/about/mymaps/

[11] WYSIWYG = What You See Is What You Get. Microsoft Office is an example of this; an opposite would be e.g., LaTeX.

2.3 Apple

Apple Photos [12] is *Apple*'s photo management and editing application for iOS operating system. It is bundled by default on all *Apple* devices and is integrated with *iCloud*, *Apple*'s cloud service. It can automatically add photos to collections, and has a map view, able to show all collections of photos at once, including clustering them as needed for easier viewing, see Figure 6 below. Unfortunately, access to *Apple Photos* requires not only an account, but also *Apple* hardware to access and use. Anyone can create an *Apple ID*, but it has very limited access to their services (e.g., *iCloud* is accessible but not *Photos*), as *Apple* is of course primarily a hardware manufacturer, and such the software is incentive to buy an iPhone instead of competing alternatives.

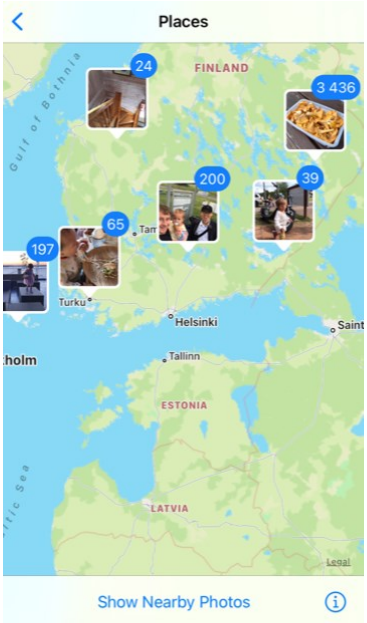


Figure 6 – Example screen capture taken from an iPhone, showing clustering used with photos.

Apple also does not keep any metadata on the image itself, removing it on upload to *Apple Photos*. This means that the metadata of images is kept separately from the image and locked to the Apple ecosystem. This approach is quite different from other companies and could be due to either security and privacy like social media sites do (e.g., *Facebook*), or it could be an effort to keep a user on their platform and highly control the experience. This would fit *Apple's* usual strategy since they like to keep their ecosystem very closed off. I doubt it is a storage space saving measure, since

[12] Photos for iOS and iPadOS - Apple | https://www.apple.com/ios/photos/ | Only viewable on Apple devices

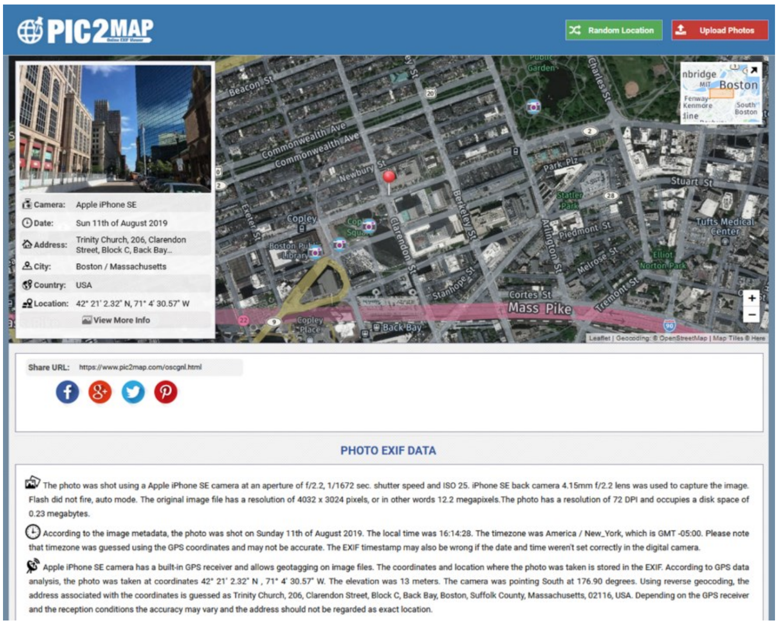
metadata takes very little space, especially since *Apple* controls what metadata goes into a photo to being with, as it is their camera software. As for security, it is possible to share images without metadata attached, so this should not be an issue for sharing purposes.

Table 4 – Apple Photos features summary

Account	Mobile view	Collections	Map	Clustering	Formats	Keep metadata	View Metadata	Monetisation
✓	✓	✓	✓	✓	Common	✗	✓	Hardware

2.4 Pic2Map

Pic2Map [13] is a free online EXIF data viewer with GPS support which allows you to locate and view your photos on a map. It allows you to upload one or more images, and it will show you the metadata of said photos – focus on location with a map, but also camera information and many others. See Figure 7 for screenshot of user interface (UI).



Camera: Apple iPhone SE
Date: Sun 11th of August 2019
Address: Trinity Church, 206, Clarendon Street, Block C, Back Bay...
City: Boston / Massachusetts
Country: USA
Location: 42° 21' 2.32" N, 71° 4' 30.57" W
View More Info

Share URL: https://www.pic2map.com/oscgnl.html

PHOTO EXIF DATA

The photo was shot using a Apple iPhone SE camera at an aperture of f/2.2, 1/1672 sec. shutter speed and ISO 25. iPhone SE back camera 4.15mm f/2.2 lens was used to capture the image. Flash did not fire, auto mode. The original image file has a resolution of 4032 x 3024 pixels, or in other words 12.2 megapixels.The photo has a resolution of 72 DPI and occupies a disk space of 0.23 megabytes.

According to the image metadata, the photo was shot on Sunday 11th of August 2019. The local time was 16:14:28. The timezone was America / New_York, which is GMT -05:00. Please note that timezone was guessed using the GPS coordinates and may not be accurate. The EXIF timestamp may also be wrong if the date and time weren't set correctly in the digital camera.

Apple iPhone SE camera has a built-in GPS receiver and allows geotagging on image files. The coordinates and location where the photo was taken is stored in the EXIF. According to GPS data analysis, the photo was taken at coordinates 42° 21' 2.32" N , 71° 4' 30.57" W. The elevation was 13 meters. The camera was pointing South at 176.90 degrees. Using reverse geocoding, the address associated with the coordinates is guessed as Trinity Church, 206, Clarendon Street, Block C, Back Bay, Boston, Suffolk County, Massachusetts, 02116, USA. Depending on the GPS receiver and the reception conditions the accuracy may vary and the address should not be regarded as exact location.

Figure 7 – Pic2Map user interface

[13] Photo Location & Online EXIF Data Viewer - Pic 2 Map | https://www.pic2map.com/

The service does not require any account, nor is there even a possibility of creating one. As there is no account to link to any uploaded images, there is also no way to save images or collections for later use. Sharing images is possible via a URL link, and uploads are kept for some undisclosed amount of time; in case of private images, they are deleted after a few days. You can upload multiple images at a time, so technically *Pic2Map* has collections, but it's not a fully-fledged out feature. Using the site on mobile is possible. The service shows ads for monetization.

As for metadata, the site shows most relevant information about a photo uploaded, including exact location, camera information (i.e., settings the image was taken with and specific model of camera). Location is shown on an embedded map, but there is no clustering if multiple photos are uploaded. You can upload multiple photos, which are collated into an album that, if made public, can be shared via a URL link (as mentioned before, no account system means this URL is the only way to access the photos later). However, only JPEG format photos are accepted, and the service is very particular that the image needs to be an actual photo (most likely based on available metadata). It seems that the service does not keep the original image or metadata, instead removing the data and compressing the image down to smaller size. See Table 5 for summary of features.

Table 5 – Pic2Map features

Account	Mobile view	Collections	Map	Clustering	Formats	Keep metadata	View metadata	Monetisation
✗	✓	✗	✓	✗	JPEG only	✗	✓	Ads

Images are marked public by default unless user clicks a checkbox, which is a dark pattern and illegal against EU's General Data Protection Regulation (GDPR) (data collection without explicit user consent). The "Keep photos private" checkbox also resets to off every time on page load, and the label is also not clickable, which while possibly not intentionally harmful is still annoying from a usability point of view. Basically, they have done everything possible so you would not click it. In *Pic2Map*'s Terms of Use they do say that "whether public or private, you can delete your uploaded photos or albums and all information will be removed permanently from our database." If you don't delete an image, they do have "a non-exclusive, royalty-free, perpetual, irrevocable worldwide license to use and to display online and in *Pic2Map* website." This is not to say other services do not have similar dark patterns, but this is just an easy example of one.

There are also multiple other services like what *Pic2Map* offers but with slightly varying feature set; for example: *Photo-location.net*, *Metapicz*, and *Whereisthepicture.com*. These services are highly like *Pic2Map*, as they allow the user to upload a photo and will then show where that image was taken and possibly other metadata. Some slight changes exist between services exist, for example some only handle one image at a time, and some allow multiple types of image formats.

2.5 Flickr

Flickr [14] is an image and video hosting service / online community, focused hosting and sharing high-resolution photography. It is popular among both professionals and amateurs. Unlike other services listed here, *Flickr* is heavily focused on users' ability to share and sell photos, giving prominent information on even the device images were taken on (though the photo itself is the main point). See Figure 8 for screenshot of UI.

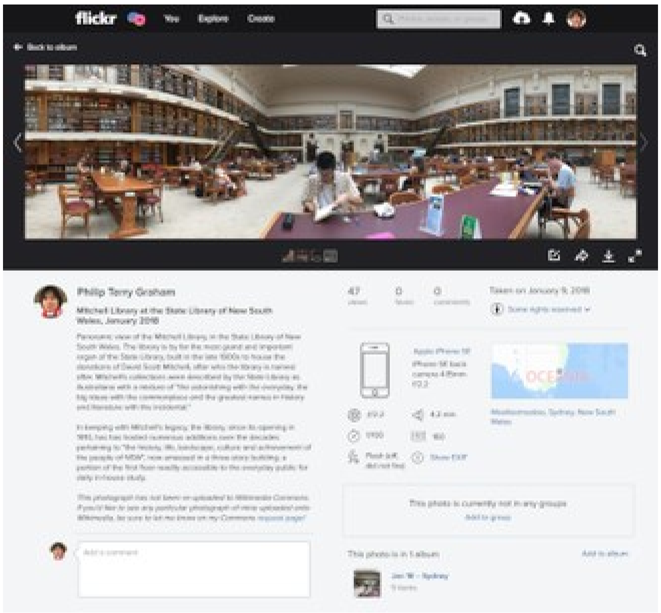


Figure 8 – Flickr UI, showing a single image (compressed for privacy and copyright reasons, taken from Wikipedia[15])

Flickr requires an account to upload and handle your own photos, but viewing public images is accessible to anyone. Most of the site is viewable from mobile devices through a browser, except

[14] Find your inspiration. | Flickr (https://www.flickr.com/)

[15] File:Flickr screenshot.png - Wikipedia | https://en.wikipedia.org/wiki/File:Flickr_screenshot.png

for uploading and handling collections only being designed for desktop access. You can upload photos (individually or multiple at once), which can be organised into collections. Metadata from them is automatically read and shown, including camera and location information, along with a map view. The public facing map view does not have any clustering, but when viewed from the editing view (*Organizr* as *Flickr* calls it) you can view all your photos with clustering (see Figure 9). This view does not let select a single collection though, only all photos at once. So technically they have clustering, it’s just hard to find and not visible to other users.

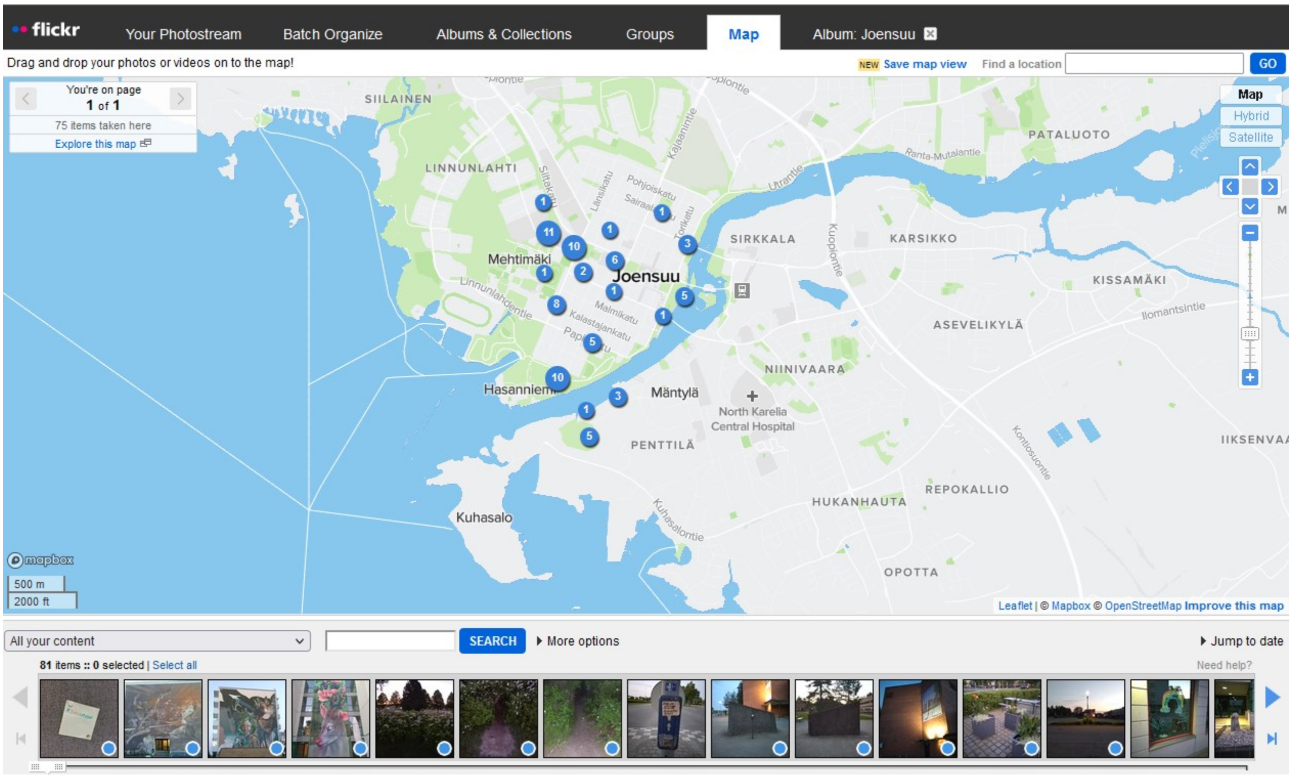


Figure 9 – Flickr organiser editor view, with clustering.

Flickr only accepts JPEG photos, and will keep the original metadata intact, along with ability to view it afterwards–prominently displayed in the public page for a photo. Their monetisation model is mainly advertisement, but they do offer subscription for more features, including the ability to upload more private photos. As a service, they do offer easy way to browse through other people’s photos but managing your own is a bit too much of a hassle, and the user interface doesn’t look like it has been updated in a while, especially in the collection editing side. See Table 6 for summary of features.

Table 6 – Flickr features table

Account	Mobile view	Collections	Map	Clustering	Formats	Keep metadata	View Metadata	Monetisation
✓	✓	✓	✓	✗	Photos, JPEG	✓	✓	Ads / subscription

2.6 Imgur

Imgur [16] is an online image sharing and image hosting website with a "focus on social gossip" [17]. It is not really related to what I'm doing with *MyMopsi*, as it is not focused on photos but more humorous images and videos. But still it is strongly associated with image sharing and focused on handling images online, so could be useful to look at it. See Figure 10 below for a screenshot.

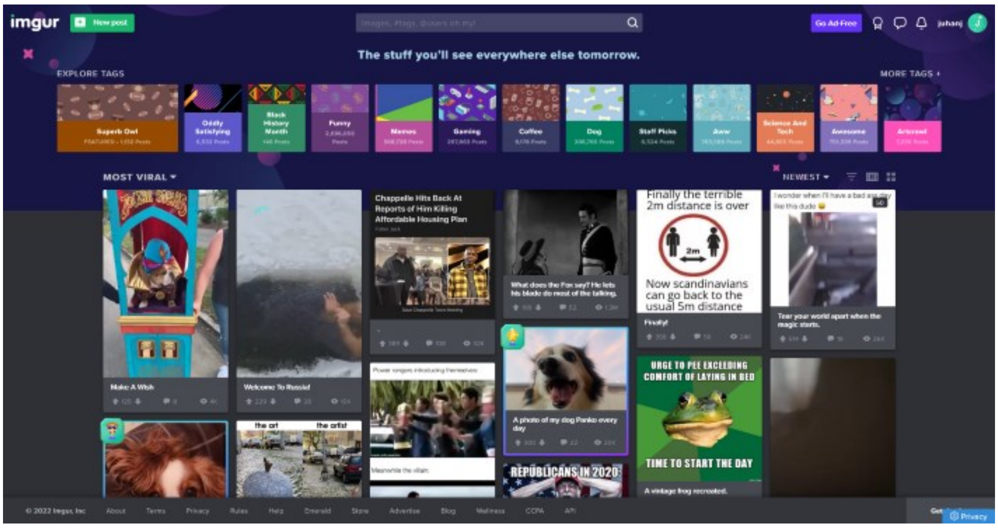


Figure 10 – Imgur front page UI

An account is required to upload and share images, but one is easy to create and requires no personal information. As a very sharing focused site, their website works on mobile devices as well

[16] Imgur: The magic of the Internet | https://imgur.com/

[17] https://en.wikipedia.org/wiki/Imgur | Quote from Wikipedia article, which does not state its source, so probably some random Wikipedia editor.

via browser and a separate optional mobile app, which they do push rather aggressively, a popular tactic with websites these days.

You can upload any images there–they accept most common image and video formats–which can be organised into collections. They strip all metadata from images when uploading, including location data. They do not have any map view, and such clustering either. Imgur does make it easy to quickly upload and share images of any kind, either publicly or privately, via a simple link which you can then embed to places. Convenient if you just want a place to quickly host an image to share (or even embed) somewhere, but for any location-based stuff, it is very useless. They do not advertise themselves with any focus on photos, so I consider this fine personally. See Table 7 directly below for summary table of features.

Table 7 – Imgur features

Account	Mobile view	Collections	Map	Clustering	Formats	Keep metadata	View Metadata	Monetisation
✓	✓	✓	✗	✗	✓	✗	✗	Ads

2.7 Social media sites

Since core ideas of social networking are the same between the services looked at, I will be grouping them together because of that similarity. Services looked at are *Facebook* [18], *Instagram* [19], and *Twitter* [20]. They all have their nuances, but how they handle images is mostly same. *Twitter* is mostly focused on microblogging and sharing snippets of text (called "tweets"), with possibility to share images, which is however mostly incidental to their focus. *Facebook* users can share more general stuff, with more of a focus on specific circle of friends and/or family. *Instagram* is focused on sharing images and short videos.

[18] Facebook | https://www.facebook.com/

[19] Instagram | https://www.instagram.com/

[20] Twitter | https://twitter.com/

They all require accounts to use, going as far as being difficult or impossible to use without one. None of these offer any specific location-based services as far as photo collections go, nor do they have any map-based visual included with images; though Instagram at least will offer a suggestion on where an image was taken based on metadata, which will be included with the post. Only *Facebook* allows you to create collections of photos but not the other two services. All three services strip all metadata completely from uploaded images, and in the case of *Instagram*, even downloading any images (other's or own) seems to be impossible on their Android app at least. The services allow basic image formats, JPEG, PNG, and GIF.

Removing metadata is generally discouraged by International Press Telecommunication Council (IPTC) [21], but personally in this case I agree with the decision made by social media services and think that the main reason is privacy and security of users. These services are not intended as final storage solution for photos and as such I do not expect them to keep the photos unchanged. Nor do I think it is a space saving measure (though it's not a bad thing for them certainly to save some kilobytes at their scale of operations), as in my testing a typical photo only has approximately ~1 % metadata. [22] As for privacy, many users do not know just how much information contained is in a typical photo, which could be used against them, especially if the user has a lot of followers (more followers increase risk of stalkers and other dangers that come with fame), though even private individuals can face dangers from private information leaking. (Boyd, 2008)

See Table 8 for summary of features between the different services. Apart from collections they are all the same in the specific areas looked at.

Table 8 – Facebook, Instagram, and Twitter features summary

Account	Mobile view	Collections	Map	Clustering	Formats	Keep metadata	View Metadata	Monetisation
✓	✓	✗ FB: ✓	✗	✗	Basic	✗	✗	Ads / personal data

[21] Many Social Media Sites Still Remove Image Rights Information From Photos - IPTC | https://iptc.org/news/many-social-media-sites-still-remove-image-rights-information-from-photos/

[22] A typical photo is 5 megabytes. EXIF metadata is maximum 75 KB, and often less. Testing done with *Exiftool* (removal of metadata). Further information in IT-project technical documentation.

2.8 Summary & Conclusion

I have looked at several services: *Mopsi* (and *O-Mopsi*), *Google* services (*Photos* and *My Maps*), *Apple Photos*, *Pic2Map* (and the like), *Flickr*, *Imgur*, and lastly social media sites (*Facebook*, *Instagram*, and *Twitter*). I identify and compare several specific features that are important in *MyMopsi*. These features are **account functionality**; **mobile view**; **ability to group items into collections**; **map-view** with **clustering**; **formats supported**; **metadata handling**; and lastly **monetisation**. I will be going through these features here as a short summary.

Account functionality: In the different services looked at, this point ranged from *Pic2Map* not having any account, to *Facebook*/*Instagram* actively discouraging use of services without logging in and *Apple Photos* even being locked behind hardware. *MyMopsi* has a middle ground. Some pages are available without needing an account: viewing public collections and checking single image's metadata. However, an account is essential to saving, handling, and sharing collections. See Table 9 for quick summary.

Table 9 – Summary of services with account functionality

	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
Account	✓	✓	✓	✗	✓	✓	✓

Mobile view: supporting mobile devices is important, as majority of internet traffic from mobile devices. As well as main smart phones being main device for taking photos for most people. With services looked at most of them had some way to access with mobile devices with varying results of usefulness. *Mopsi* used to have support, and *Flickr* uploading process is not made for smaller screens in mind even though rest of the site is fine. *Google's My Maps* is not really meant for mobile devices, as the maps made there are meant to be embedded in other places.

Two different approaches: native applications, or adaptive website through a browser (in recent times has blurred the lines between these approaches with recent technology). Both have their advantages; native applications can be better with more focused approach, but for supporting more devices and operating systems it would require too much work for this project. An adaptive website means no worrying about operating system or device, although the result might not be as clean as a purpose-built native software. See Table 10 for quick summary.

Table 10 – summary of services with mobile view

	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
Mobile view	✗	✓ (My Maps: ✗)	✓	✓	½	✓	✓

Collections: grouping images into logical collections useful for end-users, makes handling of substantial number of images easier. Account-functionality important for this especially, as without it there is no way to access collections after leaving site. Most of the services looked at have collections in some form, only exceptions being *Pic2Map* (which does not have an account system to save collections for later use) and social media sites (which do not have any kind of collections, except for *Facebook*). No distinction was made in this section between private and public collections, but this separation exists in all services.

MyMopsi has both private and public collections; public collections and images could be used for other purposes as well; (Waga, Tabarcea, & Fränti, 2012) or (Setlur, Battestini, & Ding, 2009). Flickr in fact has recently changed their free model to only allow maximum of 50 private photos because of how their business model is built on sharing photos. See Table 11 for quick summary.

Table 11 – summary of services with ability to group items into collections

	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
Collections	✓	✓	✓	✗	✓	✓	✗ (Facebook: ✓)

Map: As a location-based application, having a functional, dynamic map is a core part of location-based service. In this point, there is a lot of variety between approaches. Social media sites and *Imgur* do not have a map at all (social media sites do use the location to show an address attached to a post if applicable). I do consider this fine, as they do not particularly advertise themselves as location-based applications, *Imgur* especially so. Arguably social media sites could benefit from a built-in map, *Facebook* for example could advertise a company's location without leaving the site. But when on mobile you could just click on the address and open an external application, the benefits seem small. Other ideas for handling photo collections have also been presented, for example (Viana, Bringel Filho, Gensel, Villanova-Oliver, & Martin, 2007).

Google Photos shows a map with location marked for a single image only; it does not have an option for whole collections on a map (ignoring *Google*'s *My Maps* as it has a slightly different purpose, and not useful to an average user).

The rest of the services have the same system. Full-sized map with capability to view whole collections in some capacity. Summary table of services with a map-view combined with clustering, see next table.

Clustering: the act of grouping logical group of items into, well, clusters. Strongly connected to the map-feature, as it is mandatory. As well as having a map (with collections) without clustering is also very limited, as clustering helps with clutter and readability. Of the services listed here, only *Apple* and *Mopsi* have clustering. Clustering used in *MyMopsi* (and *Mopsi*/*O-Mopsi* by extension) is explained in more detail in Section 3.3; there have been other attempts at visualization over the years, for example (Kisilevich, Mansmann, & Keim, 2010). *Flickr* has clustering, but only in one map only visible in an editing view for images, therefore only partially useful. See Table 12 for quick summary for both map-view, and clustering.

Table 12 – Summary of services with map-view, along with clustering as they are linked together in this context.

	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
Map-view	✓	✓	✓	✓	✓	✗	✗
Clustering	✓	✗	✓	✗	✗	✗	✗

Metadata handling: Apart from Imgur and social media sites, most services handling images do keep the metadata intact. Pic2Map being an outlier as it does not keep the original image at all. For social media (and Imgur) the metadata is not really that useful after uploading and is just more work and a security risk (due to personal/private information), whereas Google, Apple, and Flickr are a final storage solution for photos, with interest in keeping the original data in case a user wants to access that information later. (Ahern, et al., 2007)

As mentioned before, IPTC does say that metadata should not be removed from images, but there are valid reasons for doing so. [23] For example, personal photos being publicly accessible without the user knowledge of the information contained (where / how the photo was taken), which can be used for nefarious purposes. So, keeping the info means the service must ensure only authorised access. Google for example has a feature for sharing an image without metadata attached. [24] Instead of removing the metadata services should maybe try to educate users on what their images contain and provide better ways of sharing images without this data. Though trying to change human behaviour is probably a lost battle. (Smith, Szongott, Henne, & Voigt, 2012)

Viewing metadata mirrors how services keep it. What varies is what data they show to users. Pic2Map shows most (if not all) info on a photo, while Google for example only shows date image was taken, device name, and location. Flickr focuses more on device and camera attributes, which could be important for professional photographers who are interested in how the picture was taken. Imgur does not show any information on an image shared, nor do social media sites, as they completely remove any metadata from an image as discussed above. Social media sites do allow sharing this information attached to a post, but this is completely user editable and only what the user chooses. In some cases, Instagram for example, a location might be suggested based on where the image was taken.

See table 13 for summary table for both keeping and viewing metadata, as they are intrinsically linked.

[23] Social Media Sites Photo Metadata Test Results 2019 - IPTC | https://iptc.org/standards/photo-metadata/social-media-sites-photo-metadata-test-results-2019/

[24] What You Need to Know About Exif Data - Consumer Reports | https://www.consumerreports.org/privacy/what-can-you-tell-from-photo-exif-data-a2386546443/

Table 13 – summary of how services handle metadata

Metadata	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
… keep	✓	✓	✗	✗	✓	✗	✗
… view	✓	✓	✓	✓	✓	✗	✗

Image format support: most of the services mostly support only JPEG, in some cases some other common formats, like PNG (solid colour images and / or text), GIF (ancient format for short videos), or HEIF or raw (for saving photos). This is mostly fine, as supporting more formats just introduces complexity to software that is not necessary as the use-cases for rarer formats are... rare. However, for *MyMopsi*, I do not intend to intentionally limit support to any specific format.

Services which allow more formats are usually storage focused, like *Google* with *Google Drive*. *Pic2Map* on the other hand is very focused to just showing metadata in JPEG images and it would be rather pointless for it to support for example GIF to being with, as it wouldn't contain any of the extra information contained in a JPEG. Look at Table 14 for quick summary.

Table 14 – summary of different formats supported by looked-at services

	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
formats supported	JPEG	Most common	Most common	JPEG (explicitly photos)	photos, JPEG	✓ (all)	Basic

Monetization: as a research project / student graduation project *MyMopsi* will not and cannot have monetisation implemented (not that I would want to have any), so this point is mostly for curiosity. [25] Still, it is a good idea to acknowledge what costs are relevant and how other services approach

[25] Also, as an aside note, monetisation would involve a lot of long-term liabilities and responsibilities, which would be prohibitively difficult to navigate in a project of this scope.

this as it affects (sometimes greatly) usability of a service. In almost all cases advertising is the main approach (as it is on most websites; people are very cheap on the internet). Only *Mopsi* (for the same reason as *MyMopsi*) and *Apple* do not have advertising, though *Apple* is apparently moving in that direction as well. However, *Apple* is still mainly a hardware company.

There is also a difference between *Google* and social media sites compared to the other entries listed. They sell advertising to other companies, using personal data gathered from users, as opposed to for example *Pic2Map* which just shows advertising.

As the main cost of maintaining *MyMopsi*; the map and storage. The map used is *Google Maps*, which has a very limited free tier. Other solutions exist, for example hosting the map yourself as *Google Maps* does have running costs (*Pic2Map* has for example switched from *Google Maps* to *OpenStreetMap* probably for this very reason, to have more control and to lower costs). For simplicity in implementation this approach was not considered.

Storage is provided by the university, but even still is not unlimited, especially with large number of large photos that *MyMopsi* would be handling. It adds up fast, with large photos especially. In most services subscription is related to number of images allowed to upload. Flickr for example has started to enforce their limit of only maximum 50 private photos, due to how public photos work in their favour and draws users to their site.

See Table 15 for quick summary.

Table 15 – summary of monetization used by different services

	Mopsi	Google	Apple	P2M	Flickr	Imgur	Social media
Monetization	N/A	Subscription / ads	Hardware	Ads	Ads / subscription	Ads / subscription	Ads / personal data

As a conclusion for this feature comparison, all these services have some specific problem as a location-based application that *MyMopsi* aims to solve. Smaller services like *Pic2Map* aim to show the user the metadata / location-information of only one image at a time, and showing multiple images is not possible. Images cannot be saved individually, or into collections for easier handling. More powerful services like *Google's My Maps* provides better support for collection of images, but is much more difficult to use than *MyMopsi*, as it requires the images be loaded via *Google Photos*, where they must be uploaded first, not to mention a *Google* account. It is quite powerful as a tool for creating custom maps to embed somewhere, but for showing photo collections on a map it is quite clunky. Plus, none of the services have clustering, which is a problem, which is further elaborated on in Section 3.3.

3 MyMopsi Actual

In this section I will attempt to explain how I address the core features I pointed out in my previous chapter: account features, mobile friendly, ability to save collections, map-view with clustering, accept several formats, and lastly handling metadata. Not necessarily in the order given.

First, I will look at the user interface and experience shortly, followed by looking at the back-end side of thing with database design and storage. Afterwards I look at the clustering system that is one of the main points of *MyMopsi*. Lastly, I consider some security considerations that were thought of during development.

3.1 User interface and Experience

To log in and access core functionality, a user can either create a new *MyMopsi* account or use already existing *Mopsi*-account. A backend framework for other login services (e.g., *Google*-account) has been implemented, for further future additions; currently it is only usable for logging in with a *Mopsi* account. Using third-party social logins is popular, and such is good to outline implementation even if the feature is not fully implemented (Gafni & Nissim, 2014). Some parts of *MyMopsi* are accessible without logging in: viewing public collections, and viewing metadata from an individual image, à la *Pic2Map*, as these features do not need any user identifying data. *MyMopsi* Single Image could be useful for attracting users, so keeping it available for all is preferable.

Settings page is for any possible options a user might need or want to adjust, that affect the whole site (for that instance/user). At the time of writing, it only contains a language switcher. A language selector is important for attracting a wider user base, and although the implementation in *MyMopsi* is a simple one, it is meant to be easy to expand with new languages (with some limitations regarding character sets and reading direction). As of time of writing, only English and Finnish are available. In addition to being able to select language, the site uses icons as much as possible to minimise needed translations.

MyMopsi also has the feature to upload a single image temporarily to view metadata and location, like *Pic2Map* looked at in section 2.4, as all the necessary parts for this were already implemented elsewhere on the site so it was an easy addition. Sometimes you just quickly want to see what information is in an image, or where it was taken assuming location is stored in the image. The image or metadata is only held temporarily during upload, and not stored anywhere after. This could also be a useful tool for attracting new users, as the threshold for testing it is low. As it is currently done, it is rather barebones as it was a late quick addition and never meant as a focus.

Collection handling is split into two pages: a list of collections, and a list of images in a single collection. Collections-page has a list with name and description as we as number of images listed with a thumbnail. A user can choose to either view own collections or public collections from a dropdown menu.

Clicking on a collection links to a page with all images with said collection, as well as links to edit said collection, a map-view, and the upload-page. On the page itself, you have all images listed, or if there are more than 100 images in the collection they are split into pages, with pagination controls shown for moving through the pictures. The image used in the list are only small thumbnails for faster loading; clicking on a thumbnail opens the full-sized image, along with name and description and some other information from the image.

On the upload page a user can upload image files to the server to be added to specified collection. There are very few limits to what a user can upload, which are either related to software compatibility and file types, or size. Size is more straightforward to explain, only images smaller than 10 megabytes (MB) are accepted. Most common JPEG images (i.e., normal photos in this case) are significantly smaller than this (around ~5 MB) when compressed in a lossy format, though most cameras these days also have an option for lossless formats which can be easily over 10 MB. This limit is purely to prevent the server overloading with huge images, which can even be an attack vector for a Denial-of-Service -attack. The server also checks for file type, specifically that the uploaded file is an image. This process checks the MIME-type [26] of an image written in the file metadata for this. There are no limits on specific image types beyond marked as "`image/*`", which can lead to problems on the client-side. The most obvious problem is an image not displaying, but I

[26] MIME: Multipurpose Internet Mail Extension | MIME - Wikipedia | https://en.wikipedia.org/wiki/MIME

do not consider this a problem at this state as this would be rare. Limiting to a specific list of allowed formats also makes any new formats in the future a problem. [27]

There is also duplicate checking on a per collections basis, so the same image cannot be uploaded to the same collection multiple times. However, there is no limit for same image being in multiple collections. This is based on calculating a quick and simple hash of the file, which is cryptographically insecure but good for comparing files. This hash is calculated in file contents so a different name would have no effect on the calculated hash.

The upload process on the client-side also presents a simple bucket sorting problem. The server has a limit of 20 MB for each request sent to it (for security reasons), and so the images must be sent in multiple batches. This is also done for user feedback; sending all the files at once means that user would have no way of knowing how the upload is progressing or how long is left. Sending in batches means that each batch must be filled with as many files as possible for maximum efficiency (we don’t want to send individual files, each request must be handled separately from start to finish by the server which costs time and processing power), which is the bucket sort problem. Added complexity is all the files having a variance of file size. The complexity of this is necessary as uploading 100 photos (~500 MB) can take more than 30 seconds, so user feedback during the process is needed.

Lastly, after a user is created and images uploaded into a collection, the map-view. Every collection (with images with location data) can be viewed on a map-page, based on *Google Maps* and a custom clustering solution created by the machine learning group at UEF. A more detailed explanation of the clustering system is presented in section 3.3 in more detail.

MyMopsi is also built to be usable on both desktop and mobile. While the desktop is the focus, the site has been tested on current up-to-date browsers on Android (Firefox and Chrome) to make sure all aspects work to some degree of acceptability. The mobile use is facilitated through a browser instead of a mobile app. Current tools and knowledge make it very easy to create websites that work on all common sizes of screen, devices, and operating systems. There would be little difference

[27] There have been several attempts to dethrone JPEG for decades now as the primary image format. Still, several are being developed, e.g., JPEG XL.

between a native application and a website, and it would force two different version in development.

3.2 Database design and storing files

The database is made of three main parts: User, Collection, and Image. The links between are thus: user ➞ collection ➞ image. The links are one-to-many, meaning, for example, one user can have multiple collections, but a collection can only have one owner. This could be changed to multiple owners to facilitate multiple people having control over a collection or an image, but this was not seen as a concern during development. In addition to these three main tables, there is also a table for third-party logins. The database should be easily extensible with new features, if needed for future.

Uploaded images are stored on the server unchanged, aside from name of the file due to convenience of software compatibility and security/privacy (though original name is also saved in database). Software compatibility refers to preventing character encoding issues from cross-platform behaviour (operating systems and/or languages). It also provides a modicum of anonymity, though this is debatable, because anyone who has access to the files already has access to the whole server. The images are also stored by collection, and there is simple duplicate checking in place on a per-collection basis, based on a simple SHA1-hash [28].

Important information to the functionality of the site is saved in the database on upload; the image-file after upload is only used for displaying the image when needed on client-side, so that only time metadata is read from the image is during upload. This is done to save time, as this can be rather time-consuming process; reading the metadata directly from file every time a collection is loaded can take a noticeable time. The rest of the metadata (if any exist) can be read from the image on demand and displayed to user. Downloading images or collections is not currently supported directly in UI, although I do not prevent users from downloading the image manually (via right-click context menu) [29].

[28] SHA-1 - Wikipedia | https://en.wikipedia.org/wiki/SHA-1 | Security Hash Algorithm 1, an older cryptographically broken but fast hashing algorithm

[29] Instagram (on the desktop website) for example intentionally prevents this (you cannot right click on an image).

As image files can get large, some thought was put into the possibility of saving space on the image files. Compressing uploaded image files into a lossy format like WebP-format is one way. This was discarded due to added complexity. With so many different image formats allowed, this would have added more requirements, and made the upload process even more complex. Another reason why only allowing some image file formats could be better, it allows for more control easier. Also, this would have irreversibly changed the users' images, which I would like to avoid as it would lose detail from the image, and possibly metadata, as well as introducing the problem of having to communicate to the user this change. Depending on the tech literacy level of a user this would at worst confuse and aggravate a potential user.

Another way of space saving considered was deleting metadata that was not needed, as according to some sources metadata can take significant space on an image. [30] (Fazal, Efficiency of Web Crawling for Geotagged Image Retrieval, 2019). However, in my testing this was not the case; metadata in a typical photo only takes about ~1 % of the file size (or approximately less than 100 KB (the EXIF standard in fact limits the metadata to a maximum of 75 kilobytes, not that there is anything stopping anyone from adding more)), which for me is not worth it in the added complexity. A typical image can even include a thumbnail of said image, and still be under 1% total metadata. In some rarer cases metadata can be significantly larger portion, if for example specific colour information is stored in the image file (as an example, *Adobe Photoshop* does this).

3.3 Dynamic clustering of images on the map

Clustering is the task of grouping a set of objects in groups (called clusters) based on similarity in one way or another. [31] There are several different ways of measuring that similarity based on what is being grouped and why; photos, for example, could be clustered with either time taken, or distance/location (for example Figure 11). In *MyMopsi* the goal is clutter removal (no overlapping points) with any number of photos as input and clustered in real-time, improving visual clarity. (Zhao, Shi, Liu, & Fränti, 2015) presents a new grid growing algorithm that suits well for geo-spatial

[30] How much smaller are images with EXIF data removed? - ShortPixel Blog | https://shortpixel.com/blog/how-much-smaller-can-be-images-without-exif-icc/

[31] Cluster analysis - Wikipedia | https://en.wikipedia.org/wiki/Cluster_analysis

data. It works fast (on demand, dynamic queries), and with no need to specify number of clusters or inputs, which is important when clustering collections of arbitrary number of photos in real-time.

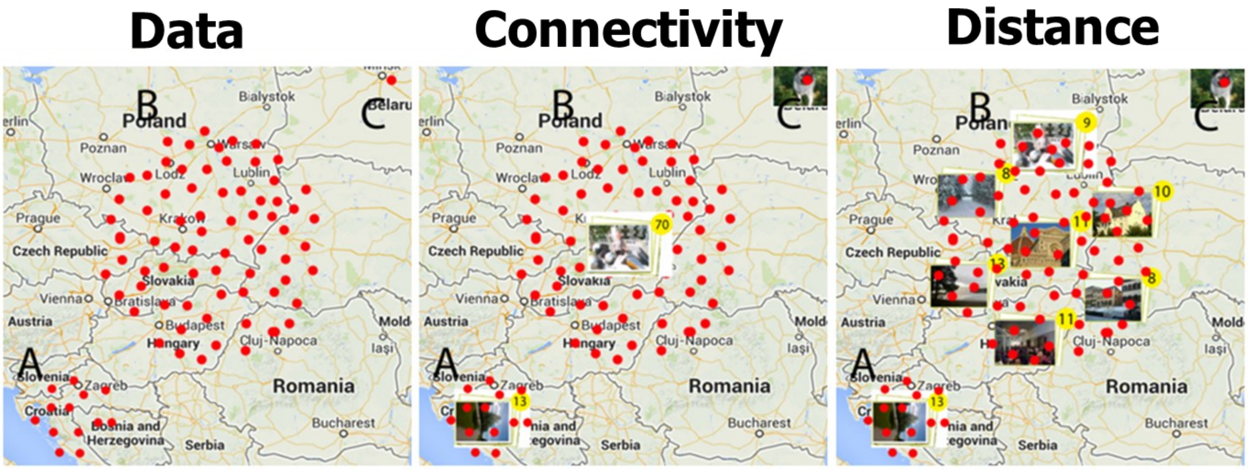


Figure 11 – Difference between clustering based on just connectivity (middle), versus based on distance and clutter removal (right). Taken from (Rezaei & Fränti, 2018)

(Rezaei & Fränti, 2018) adapts this algorithm to work specifically with photo collections for *Mopsi*. *MyMopsi* uses this same system for its clustering, with image thumbnails as the point and image representative which provides more information to the user as they navigate a map. See Figure 12 for the system in action.

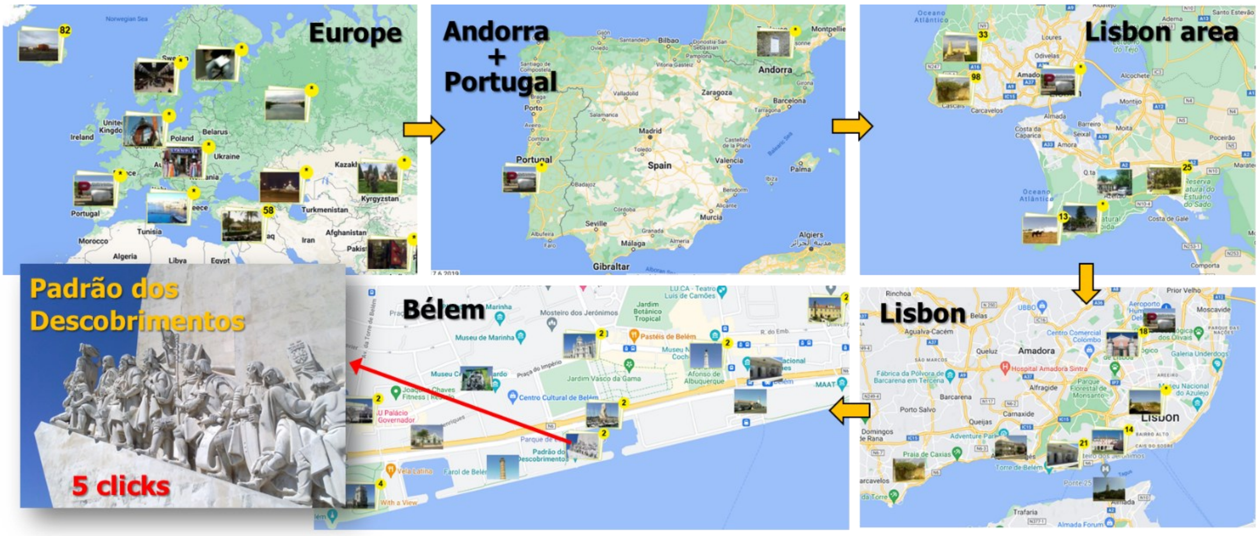


Figure 12 – An example of the clustering system in use, navigating from a zoomed-out view to a single image in five clicks.

When showing large collections of location data points three problems usually exist: clutter, slowness, and lack of dynamic queries. When trying to display many location points at once, you end up very quickly with clutter as points overlap and there is an information overload, and which leads to the user experience becoming unmanageable. Too many points also introduce slowness, as the system tries to handle all the points between the client and server-side, leading to worse user experience due to waiting around. Clustering means less for the client to process, and for the user to visually process while still having access to all items when necessary (see Figure 13). Some clustering solutions that solve one or both problems are then restricted to static queries only, with no user input. Hence need for dynamic queries.

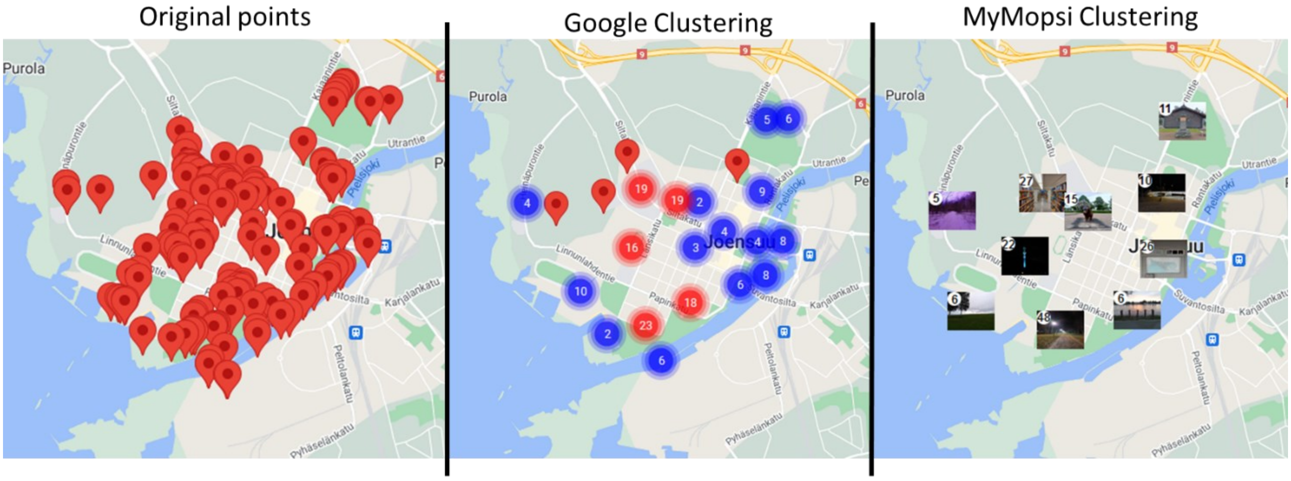


Figure 13 – Difference between no clustering, and two different clustering systems. Centre is Google's Marker Clustering API, right is Mopsi/MyMopsi thumbnail clustering

Clustering can solve the clutter problem, though it might not be the main goal of clustering. Other solutions exist, e.g., random sampling, where you take a random sample of points and only show those (see Figure 14). This is not ideal, as it can give a different result on every load, and because of randomness doesn't show true distribution. The user also needs to be able to interact with every item on the map, which is not possible with random sampling.

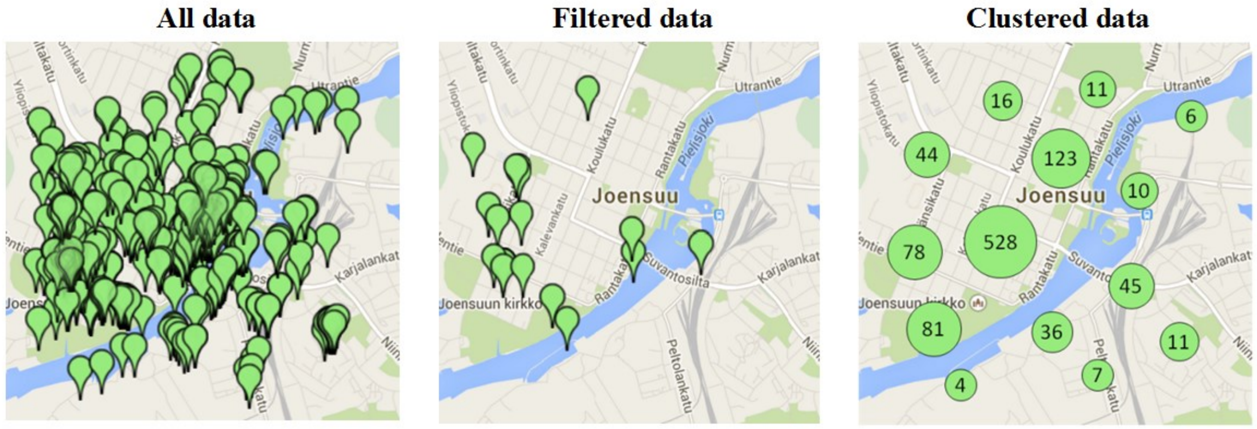


Figure 14 – Example of random sample (filtered data) vs clustering. From (Rezaei & Fränti, 2018)

On the speed side very large data sets, especially when dealing with large files like images, have the problem of slowness, which clustering can help. Not all clustering algorithms are fast time complexity (see Table 16). The grid-growing algorithm is built to be fast and scalable, and tested for up to a million points. In *MyMopsi* this many points in a single collection (or the whole service really) would never realistically be reached, as a million points would be several terabytes of storage of just photos, which now that I say it out loud is not actually that much for a server. The system allows for both client-side and server-side clustering. With server-side clustering considerable amount of connection bandwidth between the client and the server is saved because only the points that are visible need to be transferred to the client, including thumbnails. Even when doing the clustering client-side most of the rendering and processing is spent on fetching the thumbnails of the images, which saves a lot of time if only a fraction needs to be displayed. See Table 16 below for more information. Only overlap-based and grid-based support large data, and only the latter supports parallel processing which can speed up the process a lot.

Table 16 – Comparison of different clustering algorithms, from (Rezaei & Fränti, 2018)

Clustering algorithm	**Time complexity**	**Memory complexity**	**Large data?**	**Parallel processing?**
K-means	O(IKN)	O(N)	No	Yes
Overlap-based	O(KN)	O(N)	Yes	No
DBSCAN	O(N log N)	O(N)	No	No
Centroid-link	O(N^2)	O(N^2)	No	No
Grid-based	O(N)	O(N)	Yes	Yes

3.4 Security considerations

In this chapter I look at some security considerations, that are important in a project like this. This list is however in no way exhaustive, merely main ones I have come up with (order not indicative of importance):

1. authenticating users
2. image metadata and personal information within
3. validating any user input: file uploads, names, and descriptions
4. Password system, since to some services this is still difficult

1st issue is how to authenticate users. The password system is done with current best practices, and thus should not be a weak point. A PHP session cookie is used to link current active session to logged in user, and thus spoofing a user this way is very difficult. This is normal behaviour and browsers do this automatically when communicating with a server. Unfortunately, there is no “Remember me” -functionality currently, so a user must log in again every they wish to use the service (this would happen every time close and open a browser, as the session is invalidated when this happens). Technically this does increase security as there are less points of failure, but for user convenience this is usually a feature expected by default. When a user is logged in, the system has checks in place in several places for checking that the current user has authorisation to view the content being accessed, which is whenever a user tries to view a collection or an image or upload new images. Which leads neatly to…

2nd issue, which is handling image metadata, and the personal information within. When you take a photo with any camera, it may store in the image file several (possibly very personally identifying) pieces of information. Not only detailed camera settings (focus, aperture), but location data (GPS coordinate, including address and possibly even altitude), and as such making sure that only owner can access the image (if set to private). Removing the metadata altogether would be simplest solution but I don't want to do this for two reasons: 1) modifying the users' photos without their permission or knowledge, and 2) some standards organisations (e.g., IPTC) recommend against removing metadata.

So, we must control access to the image file. First step is storing the image outside browser access entirely. A server has a specific directory that contains the files (pages) accessible by browser, and anything outside is not accessible via an URL, but can still be accessed via other means on the server. This way I can control the fetching of the image and check for user privileges and if the collection is private. In this case, I merely let users choose to mark the collection as public if they want. This access check happens whenever a request is sent to the server for the image. However, there is research on how even without explicit image metadata information (e.g., location) can be referred based on visual content alone, thus making them vulnerable, from harmless targeted advertising to more serious criminal attacks. (Choi, et al., 2017) Defending against this would require modifying the image, which is probably not be what the user wants, but good to keep in mind for reasons why preventing unauthorized access is good, even if you remove the metadata. There are also other, more complicated ways of controlling access like for example a complex trust-based system presented in (Xu, Bao, Zhu, & Zhang, 2018).

3rd issue is validating any user input. There is a common saying "never trust user input", as that is the most accessible and easiest avenue for attack (if left completely unchecked). In any system with user input there is untrusted environment (which is anything client has direct access to; the client browser in the case of *MyMopsi*) and trusted environment (in web-development this would be the server side, and database). In *MyMopsi* there are two main places where user input is used: file upload, and any text form fields (name or description edit).

File uploads may pose risks to the server, and possibly even other users. Some image formats can have scripts inside them, for example SVG (a vector-based image format that uses XML like syntax inside). Another way is to disguise an executable as an image file, by merely changing the file name. In *MyMopsi* neither of these cases pose a credible threat; in the second case, the file would be caught on the server side when checking the file contents. In the first case of hidden code inside a valid image file, that would require running the code inside at some point, which does not happen by accident, it would have to be intentionally run by the end-user; merely displaying the image will do nothing. Browsers do not like running random code randomly. One certain way of eliminating this risk would be to recreate the image from scratch on upload, but this presents its own problems, mainly with handling different filetypes and preserving all metadata. Another attack would be sending a very large file to try and crash the server (a form of Denial-of-Service attack), but this is prevented by rejecting any upload request bigger than 20 MB (approximately 4–5 normal high-quality photos).

Another place of user input is any form inputs, especially text; checking for malicious content in any character strings the user can give, for example names, descriptions, and file names. Anything the user can change and is then saved in the database, or possibly shown to other users (untrusted data moved to trusted environment). Unsecured, a user could change content in the database or affect other users' systems through a cross-site scripting (XSS) attack. This problem is solved by securely encoding any user input, so that it is not possible to run either as a database command, or as a client-side script. As such, not an issue.

Attacks with malicious content can be dangerous to both server- and client-side. On the server-side, attacks on infrastructure can at best disable the website or at worst target the server itself, either disabling it or taking control. On the client-side malicious exploits could be used to target specific users or any user, to steal data or take control of client device. All this would require running the code inside the file, which is very highly unlikely to happen (basically impossible unless someone does something dumb), in *MyMopsi* images are never shown in a way that would entertain this possibility.

4th issue is password security. *MyMopsi* uses current good practices for its password system. A minimum length of 8, no maximum length (for user), with no limits on what the password should contain, and no asking to change it regularly. The password is created using standard built-in PHP functions for encrypting and decrypting password using modern algorithms. The password is never saved or logged in clear text anywhere and is only handled in one file to help maintenance. (Acker, Hausknecht, & Sabelfeld, 2017)

The minimum recommended length for passwords is 8 characters, though these days with technology improving brute forcing this is doable so even longer minimum is justifiable. I have stuck with 8 for user convenience (this is not a high security application after all). The minimum length is for higher entropy for encryption. There is no maximum length visible to user, although there is a limit of 265 for the form input, and the server-side system only actually uses the first ~80 characters due to the algorithm having a built-in limit. This limit is for performance, as too many characters will take too long. Trying to upload a several hundred- or thousand-character string as a password can in fact be an attack vector for a Denial-of-Service attack.

Odd password requirements and requiring changing password at regular intervals leads to bad password, as users will inevitably create short, memorable passwords to humans, which are easy to guess for computers. As a rule of thumb, length is the most important thing in a password (assuming you're not using one from a "Most Common Passwords"-list, obviously. It still needs randomness in some form.)

Even with all these, I only used “good practices” three paragraphs before instead of “best”, as more could be done to make the system even better for users. For example, providing users hints on stronger passwords on creation, or checking a new password for previous known breaches online and most common used passwords would also help.

In conclusion for this section, security is always an endless race against new threats, but in *MyMopsi* I do try to take it as seriously as possible.

4 Possible improvements

Software development is often a question of compromises and when to stop. Like any other software, *MyMopsi* could always be better, have more features. Scope creep is real however, and as such these features will be only ideas for future improvements. This section will document some of these ideas. This is not an exhaustive list, nor is it in any order.

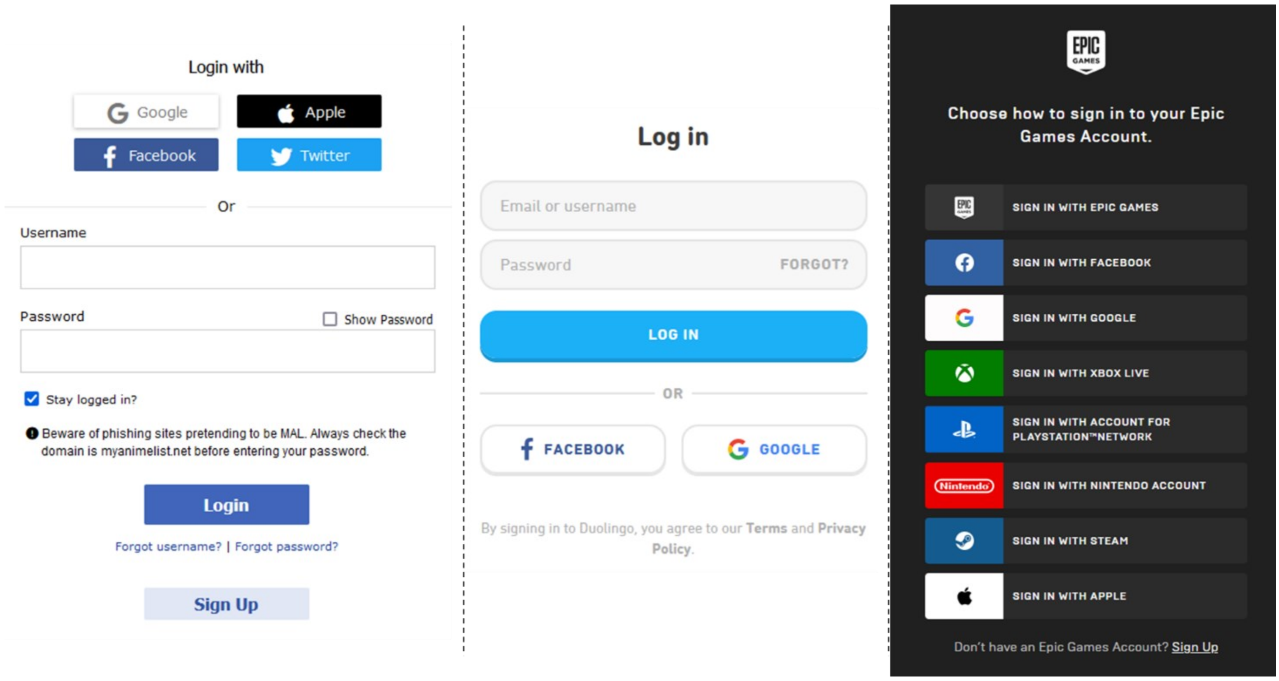
MyMopsi I could have had less steps between a user and their images. User interface design might seem subjective in many situations, but human behaviour is surprisingly similar in this regard. The collections-page, showing all collections, could have been a map, with each collection as a node (location of which would have been an average of GPS-coordinates in the collection). This would have strengthened the idea of location-based service that *MyMopsi* is supposed to be, and allowed the user to interactively see all collections, along with clustering. With the clustering I would predict it would also scale better (user experience wise) as number of collections increases. Especially listing all public collections in a list is a terribly boring experience currently. As a fun side, this could also be combined with showing all images from all collections, perhaps coloured based on collection. A little fun something for user to dive in.

As for uploading images, one addition could be the ability to import and upload images from other services, either whole (with image stored locally in *MyMopsi*) or just data used in *MyMopsi* (with image linked as an URL and not stored in *MyMopsi* (unknown if any service makes this possible)). This might be another way to attract users, as users are likely to already have images upload to another site and possibly not have them locally on their own computer stored at all anymore. However, usefulness is debatable, because as discussed in Section 2 many services remove the metadata from images and so the inclusion of location data would be questionable. So only specific sources would be useful (Fazal, 2019). Even then, as an inclusion it would be self-contained to add regarding practical implementation.

A quality-of-life feature that would be good for users would be the ability to recover an account from a forgotten password. Implementing this is not particularly difficult but was never a priority. It would increase usability and decrease the load on any possible support staff, as helping users would be automatised in those situations. It would also slightly increase user security as not having this feature means a user is more likely to choose an easier to remember, but more insecure password.

This feature would of course not actually recover a password, but merely let them choose a new one after authenticating themselves through email.

Another quality-of-life addition would be to be able to login with third-party services, e.g., *Google* or *Facebook*, in addition to a *MyMopsi* or *Mopsi* account. See Figure 15 below for examples of social login. Upside of this would be increase of security for user as they would only have to ensure that the only one account is secure, and *Google* for example cares more about user security and privacy than a random small website. Only having one login also decreases the risk of reusing passwords for user. It would also decrease the threshold for a user to try the service, as they wouldn't have to create an account (Karegar, Gerber, Volkamer, & Fischer-Hübner, 2018). Implementing this does have the habit of increasing maintenance work as the API used could change without notice, breaking stuff, leading to bad experience for users that used that.



Beware of phishing sites pretending to be MAL. Always check the domain is myanimelist.net before entering your password.

By signing in to Duolingo, you agree to our Terms and Privacy Policy.

Choose how to sign in to your Epic Games Account.

Don't have an Epic Games Account? Sign Up

Figure 15 – Examples of social login being used in MyAnimeList, Duolingo, and Epic Games Store. Note the difference in ordering compared to traditional login. Does that imply preference from a service one way or another? I don't know and this thesis does not answer this question.

There are also several articles discussing downsides of third-party social logins (Kontaxis, Polychronakis, & Markatos, 2012), (Zbořil, 2021). Choosing to log in with third-party account is a choice between several variables, e.g., security, convenience, and privacy. You exchange increased security and convenience for privacy. Both parties involved in the login process now know of the other. For example, if a fictional website *WebX* used *Facebook* login two things would happen: 1) *WebX* now has any information that the user agrees to give over from *Facebook* which may be more than desired initially; and 2) *Facebook* knows about user using *WebX* and could use this information for their own purposes, e.g., marketing. Using only one login also introduces a single point of failure, which means if you lose access the third-party social login account, you also lose access to any service that used that login information.

Ideas above are mostly generic user interface improvements. (Kandiga, et al., 2020) presented a more ambitious idea using visual image content for search features and analysis, possibly offering users more options based on it. (Waga, Tabarcea, & Fränti, 2012) shows that user generated collections could also be used for generating points of interest and used for recommendations for other users. For example, if there are a lot of pictures from a specific restaurant, suggest that location to more users.

Another addition would be location-based games. Both O-Mopsi (Fränti, Mariescu-Istodor, & Sengupta, 2017) and Puzzle-Mopsi (Fränti & Kong, 2022) suffer from not having enough location-tagged images as targets. MyMopsi could be used for repository of images for such games (Fazal, Mariescu-Istodor, & Fränti, 2021). This feature would require changes on both sides, but an API for accessing public collections (or, with some additional checks, private) could be useful. If a user wishes to export their images somewhere else this would also be helpful. As it stands, there is no easy and convenient way to get images off the platform. This should be remediated.

On the backend I will keep it short as improvements there are not as evident or as interesting. As much as I like my language switching system, it is rather simplistic and has limitations (as mentioned in previous chapters). If those limitations are exceeded, or if the development team grows, it might be necessary to switch to a more robust and standard system (but also significantly more complex). For another point, (Zhang, Yang, Yang, Lin, & Zhang, 2020) presents another, more efficient way of storing and accessing location-based objects. This is another thing that might be important if the scope of *MyMopsi* even increases.

5 Conclusion

In this thesis, a new application called *MyMopsi* is presented for handling large collections of location-based images.

MyMopsi manages to implement the key features presented in Section 2 in a way that offers best features of other services while avoiding negative features such as clustered map-view in *Google*, account-functionality in *Pic2Map*, and the closed eco-system of *Apple*.

Of the existing services studied, only one did not have account support, which makes the said application more of a curiosity. Having collections and managing them demands some way to save them. Other services lacked in a proper map support, with only one having clustering, which was locked behind a closed eco-system. Social media sites are removing all metadata from images, so they are not much better for handling geospatial data collections. This presents a niche for *MyMopsi*, as there are not many services that let you see large collections of photos on a map. On the other hand, some of the features presented in MyMopsi were too much. For example, image file format support seems a rather useless dead-end, and there are good reasons why other services only support select formats.

Competing with platform providers' own solutions for photo collections would be an exercise in futility, as discussed in (Foerderer, Kude, Mithas, & Heinzl, 2018), so *MyMopsi*'s focus for future should be the map system and clustering, with improvements in usability related to that. Could be used for multiple purposes, personal photos or displaying a public collection to multiple people. For future direction, the platform should be polished for end-user. As for more features, *MyMopsi* is a clean platform for new additions, as it is not burdened with old technology. The map should be the focus, as a mere photo collection handling tool, it would be competing with much bigger fish. Further extensions could be made with an API, which other sites could use to access public data of MyMopsi.

References

Acker, S. V., Hausknecht, D., & Sabelfeld, A. (2017). Measuring login webpage security. *Proceedings of the Symposium on Applied Computing*.

Ahern, S., Eckles, D., Good, N., King, S., Naaman, M., & Nair, R. (2007). Over-exposed? Privacy patterns and considerations in online and mobile photo sharing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 357-366).

Boyd, D. (2008). Facebook's Privacy Trainwreck. *Convergence*, 13-20.

Choi, J., Larson, M., Li, X., Li, K., Friedland, G., & Hanjalic, A. (2017). The Geo-Privacy Bonus of Popular Photo Enhancements. *ICMR '17: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval* (pp. 84–92). New York, NY, USA: Association for Computing Machinery.

Djamasbi, S., McAuliffe, D., Gomez, W., Kardzhaliyski, G., Liu, W., & Oglesby, F. (2014). Designing for Success: Creating Business Value with Mobile User Experience (UX). *International Conference on HCI in Business*, (pp. 299-306).

Fazal, N. (2019). Efficiency of Web Crawling for Geotagged Image Retrieval. *Webology*, 16-39.

Fazal, N., Mariescu-Istodor, R., & Fränti, P. (2021). Using Open Street Map for Content Creation in Location-Based Games. *2021 29th Conference of Open Innovations Association (FRUCT)*, (pp. 109-117).

Foerderer, J., Kude, T., Mithas, S., & Heinzl, A. (2018). Does Platform Owner's Entry Crowd Out Innovation? Evidence from Google Photos. *Information Systems Research, volume 29, issue 2*, 444-460.

Fränti, P., & Kong, L. (2022). Puzzle-Mopsi: a location puzzle game. *Applied Computing and Intelligence, 2*(3).

Fränti, P., Chen, J., & Tabarcea, A. (2011). Four Aspects of Relevance in Sharing Location-based Media: Content, Time, Location and Network. *WEBIST 2011 - Proceedings of the 7th International Conference on Web Information Systems and Technologies*, (pp. 413-417).

Fränti, P., Mariescu-Istodor, R., & Sengupta, L. (2017). O-Mopsi: Mobile Orienteering Game for Sightseeing, Exercising, and Education. *ACM Transactions on Multimedia Computing, Communications and Applications*, 1-25.

Gafni, R., & Nissim, D. (2014). To Social Login or not Login? Exploring Factors Affecting the Decision. *Issues in Informing Science and Information Technology*, 57-72.

Jaffe, A., Naaman, M., Tassa, T., & Davis, M. (2006). Generating Summaries and Visualization for Large Collections of Geo-Referenced Photographs. *Proceedings of the 8th ACM international workshop on Multimedia information retrieval.* Association for Computing Machinery.

Kandiga, S., Kini, U. N., Kini, U. K., Yashwin, Mamatha, G., & Harisha. (2020). Image Processing and Location based Image Querier(LBIQ). *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, (pp. 856-861).

Karegar, F., Gerber, N., Volkamer, M., & Fischer-Hübner, S. (2018). Helping John to Make Informed Decisions on Using Social Login. *Proceedings of the 33rd Annual ACM Symposium on Applied Computing* (pp. 1165–1174). Pau, France: Association for Computing Machinery.

Kisilevich, S., Mansmann, F., & Keim, D. (2010). P-DBSCAN: A Density Based Clustering Algorithm for Exploration and Analysis of Attractive Areas Using Collections of Geo-Tagged Photos. *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application.* Association for Computing Machinery.

Kontaxis, G., Polychronakis, M., & Markatos, E. (2012). Minimizing information disclosure to third parties in social login platforms. *International Journal of Information Security*.

Liu, C., Zhu, T., Zhang, J., & Zhou, W. (2022). Privacy Intelligence: A Survey on Image Privacy in Online Social Networks. *ACM Computing Surveys*.

Nikulin, D., & Buchko, O. (2020). Automated Approach for the Importing the New Photo Set to Private Photo Album to Make it More Searchable. *NaUKMA Research Papers. Computer Science, vol. 3, no. 0*, 141-148.

Rezaei, M., & Fränti, P. (2018). Real-Time Clustering of Large Geo-Referenced Data for Visualizing on Map. *Advances in Electrical and Computer Engineering*, 63-74.

Setlur, V., Battestini, A., & Ding, X. (2009). Travel scrapbooks: Creating rich visual travel narratives. *2009 IEEE International Conference on Multimedia and Expo*, 1314-1317.

Smith, M., Szongott, C., Henne, B., & Voigt, G. v. (2012). Big data privacy issues in public social media. *2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 1-6.

Torniai, C., Battle, S., & Cayzer, S. (2007). Sharing, Discovering and Browsing Geotagged Pictures on the World Wide Web. *The Geospatial Web, Advanced Information and Knowledge Processing*, 159.

Viana, W., Bringel Filho, J., Gensel, J., Villanova-Oliver, M., & Martin, H. (2007). PhotoMap – Automatic Spatiotemporal Annotation for Mobile Photos. *Web and Wireless Geographical Information Systems, 7th International Symposium*, (pp. 187-201).

Waga, K., Tabarcea, A., & Fränti, P. (2012). Recommendation of Points of Interest from User Generated Data Collection. *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (pp. 550-555). IEEE.

Xu, L., Bao, T., Zhu, L., & Zhang, Y. (2018). Trust-Based Privacy-Preserving Photo Sharing in Online Social Networks. *IEEE Transactions on Multimedia*, 1-1.

Zbořil, M. (2021). Security Concerns of Social Login Usage at the 3rd Party Cloud Services. *Proceedings of the international scientific conference Hradec Economic Days 2021*, (pp. 939-947).

Zhang, J., Yang, C., Yang, Q., Lin, Y., & Zhang, Y. (2020). HGeoHashBase: An Optimized Storage Model of Spatial Objects for Location-Based Services. *Frontiers of Computer Science*.

Zhao, Q., Shi, Y., Liu, Q., & Fränti, P. (2015). A grid-growing clustering algorithm for geo-spatial data. *Pattern Recognition Letters*, 77-84.