

GeoMedia: A Platform to Share Comments and Multimedia Content Based on Location.

Alberto Morini

*Department of Mathematics
University of Padua
Padua, Italy*

alberto.morini@studenti.unipd.it

Prof. Claudio Enrico Palazzi

*Department of Mathematics
University of Padua
Padua, Italy*

cpalazzi@math.unipd.it

Abstract—Nowadays Online Social Networks (OSN) are the principal way to create connections among people and discover places. Platforms like Facebook, Twitter, and Instagram suggest content and profiles based not only on mutual connections or trending posts, but also on geographic location near to the end-user.

GeoMedia is a platform, developed as an Android mobile application, that provides a medium for users to share comments and multimedia files (such as photos, videos, and audio) based on their location. Then, users can easily see the content shared through GeoMedia moving across a geographical map.

Index Terms—Android, Social Networks, Online Social Networks, OSNS, Social Media, geolocation, microservices, client server, multimedia, media sharing, android location, http, three-tier, three-tier architecture, SQL, DBMS

I. INTRODUCTION

In the last decade, mobile applications have almost replaced traditional computer programs, especially when it comes to social networks. Not only because it is more convenient to use a mobile phone than a laptop, but also because mobile phones can give more functionality by using the physical sensors present in way to enrich the user experience.

The explosion of OSNS (Online Social Networks) led media companies to invest and explore almost all functionality of mobile devices, thus to make users feel more connected and satisfied by providing more relevant content. As a result, users spend more time on these platforms, generating profit for social media corporations. A perfect example is Tinder, a dating app where users can meet each other based on their geographical proximity. To reach this purpose, a variety of sensors can be used to profile the end user, often using the GPS to get a very precise location, otherwise opting to the proximal location provided by Access Point / Cell antenna.

However, utilizing these features can lead to faster battery drain, higher data usage, and potential privacy concerns. GeoMedia represents an open-source alternative, which does not share data to third systems and access to only necessary user information, thus to limit the energy consumption and protecting users' privacy.

II. RELATED WORKS

As slightly mentioned before, online social media are growing exponentially in terms of numbers and also of functionality

provided; in fact, there are several alternatives to GeoMedia but usually more complex and more invasive in terms of privacy.

In example

A. Google Maps

Google Maps [1] is a web mapping platform providing many functionality such as: route directions, traffic conditions and “Street View” [2] which function consent users to create an immersive 360° vision in a specific point on the map. The last function is the only way to enrich the geographical map whenever users want, but on the other hand is very complex to create a content because it's not a picture or a video but a 360° full panoramic; and it is not possible to insert a simple description text.

Another functionality which allows users to write and post media content (limited to photos or videos) is through a review or rating of a place such as a bar, museum or attraction point already created by the platform or the owner of the by the business.

B. Foursquare City Guide

Another Online Social Network that can be used to explore locations or share comments across a city was Foursquare City Guide [7] owned by Foursquare Labs Inc. [8] created in 2008 and later shut down in October 2024.

The app provides personalized recommendations of places near a user's current location based on users' previous browsing history and places visited before, also allowing users to upload content in the position they are.

After shutting down, the features of this social network will be earned by Swarm [9], another social media owned by the same company, which allows users to share their locations with friends and create a record of their experiences in a personal life-log.

C. Instagram

Instagram [3] is one of the biggest Online Social Networks of the last decade in terms of active users; the platform is profile-centric by allowing users to enrich their profile with photos or video, and potentially to manually add a location to the post by selecting a nearby place (i.e. a museum, a bar, a

city) or adding specific latitude and longitude coordinates. With this functionality, Instagram offers also the possibility to navigate through a map seeing various content published, but cannot limit posts to a defined geographical area. The only exclusivity that Instagram can give is with the “Instagram stories”, a feature which limits photos or videos from the moment of publication up to the next 24 hours.

Even Instagram can retrieve the location via GPS to place the position in the map near to the end users to shows them more appropriate content.

In the end, this platform has a backbone ready to makes what GeoMedia do, but is not properly centered on that. Another aspect that users might consider is related to privacy concerns, since the social network is owned by Meta Platforms, Inc. a profit company known to share personal data with third parties [6]

III. GEOMEDIA

GeoMedia insert itself among the others Online Social Networks (OSNS), offering another way to share content in order to discover new places by adding a comment or a multimedia file in the current user location. In fact, the app provides just a geographical map, where users can fill it by posting not just photos or videos, but also recorded audio, music, files, or just textual comment. So, users are in charge of enriching the map of a city by creating a post, then for example, giving others the opportunity to discover secrets spots or facts about a place.

Unlike others applications seen before (i.e. Google Maps), the mobile app automatically get the location of the user posting by retrieving the information from the GPS placed inside the phones, bringing the online social network into the physical world. In fact, the app provides the feature to create some exclusive content by choosing to restrict the post published just in a determinate date and time or in a well defined geographical area.

An interesting scenario could be adopted by the city of Padua to explain one of its dictum: “città dei tre senza” (literally: city without the three). Which consists in three things that Padua has but not in a proper way:

- *The Lawn Without Grass*, which is “Prato della Valle” because the grass in it, is just ornamental is not really really a lawn like the word would intend.
- *Saint without a name*, which is “Saint Anthony of Padua” (Sant’Antonio da Padova) except that he wasn’t from Padua, but from Portugal; and the friar only lived the last of his years in Padua.
- *Coffee without doors*, and this is the “Pedrocchi Coffee”, because during war time was open all day (so the doors was removed) to provide shelter during the bombings.

So, in this example, the city of Padua could use GeoMedia to place some comments in these locations to tell the story of each peculiarity or even upload old photos to show as it was.

In the end, GeoMedia, offers a full solution to share contents in the geographical location of where the end users is; making

the Online Social Network (OSN) interactive with the real world and without the effort to pick the right position as usually happens with other social networks app.

The whole source code developed for this project is available on GitHub. [22]

IV. ARCHITECTURE

To ensure that the GeoMedia app functions as intended, it is necessary to design an opportune architecture, even if it is not strictly related to the mobile application itself. In fact, the adopted three-tier architecture shown in Figure 1, makes the app work as desired.

The designed solution is composed of three actors: the presentation layer, the application layer, and the data layer.

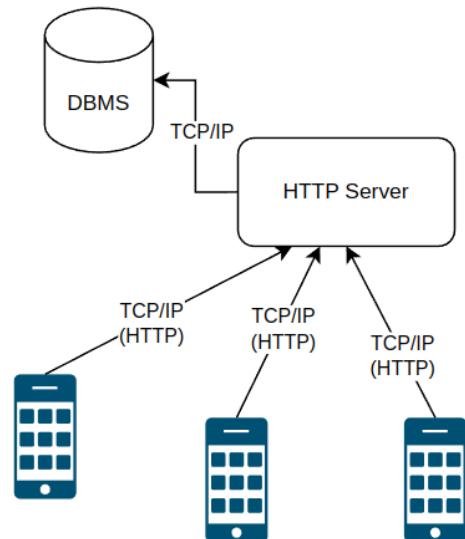


Fig. 1. Three-tier architecture for GeoMedia.

A. Presentation layer

Presentation layer also called client layer, is represented by mobile phones since GeoMedia is the Android app that serves as an interaction point with users and the designed back-end architecture.

By providing a user interface, users can view the published comments or share their own content; which in the last case, the mobile app, as presentation layer, has also the duty to retrieve data such as the current location, username and other metadata even if not directly inserted by the user.

B. Application layer

In this project, the application layer is represented by an HTTP server, which handle the requests received with the opportune functions and logic; for example: compute the distance from the given location to the exclusive posts.

The HTTP server exposes the endpoints that the front-end

(presentation layer) will fetch by making some HTTP request. Specifically, for each endpoint, the server will manipulate data with opportune SQL instruction through the Data Layer.

C. Data layer

The Data Layer consists into a Database Management System (DBMS), capable of performing CRUD operations (Create, Read, Update, Delete) on data stored in defined tables. For this project, the DBMS will also be used to create SQL stored procedures, offering a more flexible way to modify the select or insert of data queries in future extensions, without changing the code inside the server program.

V. MOBILE APPLICATION

In way to create a mobile application, there are several frameworks and technologies options. For this project has been selected the “Ionic Framework” [10], an open source toolkit for building cross-platform mobile, web, and desktop applications using web technologies such as HTML, CSS, and JavaScript. One of the key advantages of the Ionic Framework is its ability to integrate web frameworks like ReactJS, that offers many functionality to provide a smooth and great user experience.

React also has some special functions called “hooks”, which are triggered on particular events making possible injection of personalized functions. In example the hook “useEffect()” allows to call functions at the component startup or when a property changes, giving to the developer the entry point to call some personalized functions; or instead the hook “useState()” allows an automatic refresh of components rendered in the user interface at the change of the data inside it. In fact, React, through Ionic Framework, has been integrated in the GeoMedia mobile application.

A. Key functionality

GeoMedia starts with a log-in component, where users can choose to create a new profile or to sign in whenever they already have an account.

Once logged into the app, GeoMedia offers a bottom menu with two sections:

- *Profile*, in this tab users can see their posts as list and open them; or change the server configuration parameters (i.e. IP Address or HTTP port)
- *Map*, this section is the heart of the app, by offering a navigation through a geographical map to look for comments and posts uploaded by other users. Furthermore this tab provide also the access to the publish function that allows users to create a new post based on their coordinates.

A single post is composed of some attributes:

- Title
- Comment
- Multimedia content (which is optional)
- And others metadata automatically retrieved (longitude, latitude, username, date, ...)

B. Map component

As cited before, the heart of GeoMedia app is the “Map” component, that utilizes Pigeon Map [17], which is a React component that encapsulates and provides a geographical interactive map; this library also allows to put some “markers” in specific location (with longitude and latitude coordinates as parameters), which in the case of GeoMedia those points represent a content posted.

There are different types of location indicator:

- Blue marker: the current user location (retrieved in real time by GPS)
- Red marker: a post with just a comment text
- Yellow marker: a post with a media attached
- Green marker: an exclusive post, so a content visible only at date and time or in a well defined geographical area range (i.e. within 5 km)

Respecting the three-tier model explained in Section IV, the client mobile application when fetching the server in order to retrieve data, provides the current position of the user. Then, the server as application layer, filters only the relative posts stored in the data layer; then returns an HTTP packet including the uploaded content with their kind and coordinates. The specific methods to filtering the exclusive content will explained in the dedicated subsection VI-A

As follows a snippet of Map Component created

```

1 //various imports ...
2 import { useContext, useEffect, useState } from "react";
3 const MyPMap = () => { //declare the Map component
4
5   const [PostList, setPostList] = useState([]) // to store the list of posts
6   const ctx = useContext(mycontext) //use the React context to retrieve data of other components
7
8   function getPosts() {
9     if (ctx?.UserPosition != null) {
10       doRequest("getPosts", { //do an HTTP
11         request to server
12         latitude: ctx?.UserPosition[0]
13         , longitude: ctx?.UserPosition[1]
14       }).then(res => {
15         setPostList(res)
16       })
17     }
18   ...
19
20   useEffect(() => {
21     getPosts(); //to call the post
22   }, []); //at the startup of component
23   return( //render the graphics
24     <>
25       <Map defaultCenter={ctx?.UserPosition}
26       defaultZoom={11}>
27         <ZoomControl />
28         <Marker width={50} anchor={ctx?.UserPosition} color={"#154c79"} />
29         {
30           PostList?.map(s => (
31             <Marker width={50} anchor={[s.LATITUDE, s.LONGITUDE]} color={(s?.MEDIATYPE?.length > 0) ? '#d6c531' : '#f23c3c'} />
32             onClick={() => {
33               ...
34             }
35           }
36         )
37       }
38     )
39   )
40 }
```

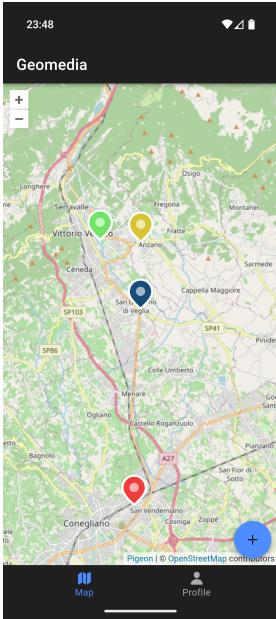


Fig. 2. Example of rendered content by the Map component

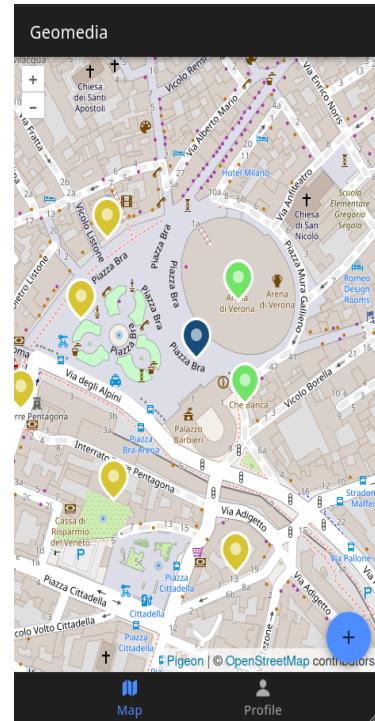


Fig. 3. Exclusive retrieving.

```

32     setPostSelected(s) // 
33     store the post selected to open it in the
34     opportuned component (viewPost)
35   }
36   ...
37 </Map>
38 ...
39 </>
40 );
41 }
42 export default MyPMap;

```

Listing 1. Map component snippet

The results rendered by the Map component can be seen in Figure 2, where in the left scenario two cities are included, useful case to cover wider area such rural landscapes; instead, in the right image, there's a more detailed scenario that enrich the center of a town by marking interesting attraction points with photos and comments.

C. Exclusivity

GeoMedia makes users interact with the real world by retrieving the real location and then fetching the server to gets the contents around them. In fact, a user, when creating a new post, can choose to make it exclusive by two factors: date or area.

An exclusive post by date will be visible from the moment it has been posted to the time indicated.

Moreover, users can choose a maximum distance within the marker will be visible by others, like creating a circular area around it, visible only if the user's current location is included.

As explained before, limited content are represented by green markers, in fact, Figure 3 compared to the right Figure 2, proves the correctly of GeoMedia by retrieving a post limited by area when the user (blue marker) is near.

This feature gives live to several use cases, for example GeoMedia could be also used to create interactive experiences and games such as: “scavenger hunt” or “hide and seek”.

D. View post

Another key component is the “ViewPost” as represented in Figure 4, which shows the content in the clicked marker by opening a new interface and fetching the server to retrieve any associated multimedia. Due to efficiency, media file attached in a post are retrieved from the server only when the posts is going to be opened. Instead, the list and relative coordinate of each visible post are requested at the opening of the application and at current location changing (thus to get exclusive posts). This component is not limited to just viewing the comment; it also provides the functionality to download the attached multimedia file and store it in the local storage of the mobile device. Media files are stored as “Base64” encoding into the data-layer, so the client is also able to encode and decode the attachments.

The owner of the post, of course, is able to delete it.

E. HTTP requests

The custom function doRequest() shown in the snippet of the Map component (Listing 1) is a wrapper to execute an HTTP request to the server. In fact, clients communicate with the server by adopting the HTTP protocol, then exchanging “XHR” (XMLHttpRequest) packets containing data in JSON (JavaScript Object Notation) format encapsulated into the body of the POST method of the protocol.

As follows, the code of doRequest function.

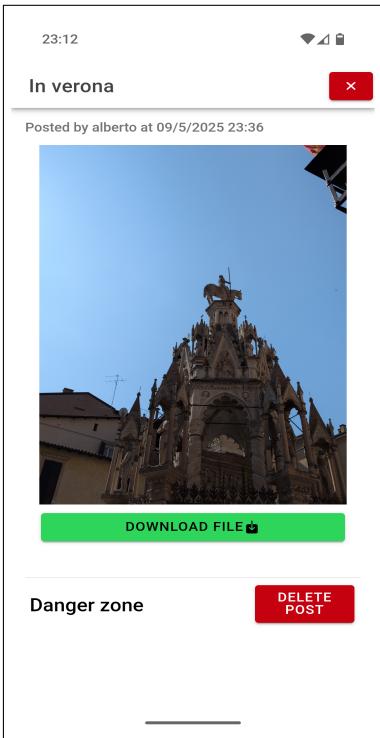


Fig. 4. Example of Post View.

```

1 //various import
2 import { Storage } from '@ionic/storage';
3 const store = new Storage();
4 store.create(); //start the cache storage
5 export const doRequest = (api, body = {}) => {
6   return store.get("ServerConfig").then(
7     ServerConfig => { //get config from cache
8       return fetch("http://" + ServerConfig.
9         ipserver + ":" + ServerConfig.port + "/" + api,
10        {
11          method: "POST",
12          mode: "cors",
13          body: JSON.stringify(body)
14        }).then(res => res.json()).then(res => {
15          return res;
16        }).catch(err => {
17          return err;
18        });
19    }
20  )
21 }

```

Listing 2. Method doRequest to fetch server

To simplify the server configuration in the app, has been installed the “Ionic Storage” [18] packet, which allows to store data into the application’s cache installed in the mobile phones. In order to use this feature an installation is required via terminal command:

```
$ npm install @ionic/storage
```

F. Ionic Capacitor

To access physical sensors on Android, such as geolocation and file storage, Ionic Capacitor [19] can be integrated into the Ionic Framework project. Capacitor is a collection of libraries

that make possible the use of native components inside web applications, then providing the API to adopt at JavaScript level.

Geodata require two specific modules of Capacitor, in specific:

- GeoLocation [11]: In way to get latitude and longitude coordinate of user logged into the app
- FileSystem [12]: To store file into mobile file-system whenever users want to download the media uploaded

These plugins can be installed via a shell command, respectively:

```

1 $ npm install @capacitor/geolocation
2 $ npm install @capacitor/filesystem

```

The geolocation package provides the method getCurrentPosition() which returns an Object with properties: latitude, longitude, altitude; and for each attribute the accuracy.

The altitude, even if not relevant for this project, could be integrated in future development thus to create more peculiar scenarios and use cases.

G. Building for Android

A key strength of the Ionic Framework is its ability to build web applications into native mobile apps; in fact, the framework acts as a wrapper/interpreter for the webpage, essentially embedding a custom browser inside the app.

In order to create an Android Package (APK) for installing the app, some shell commands are required as reported in the official documentation, such as:

```

1 $ npm install @capacitor/android //to add the
2                                         capacitor
3 $ npm install @capacitor/filesystem
4 $ npx cap add android //to denote the android
5                                         platform, since IonicFramework can also build
6                                         for "iOS" operating system
7 $ npx cap build android

```

With the last command, the application will be deployed in production mode and then encapsulated into a bundle that later will be compiled by Android Studio [20] tool and the relative SDK.

At this point, the Android Manifest needs to be modified, by adding the opportune permissions which the users will grant when requested on startup/installation of the app.

The only permission requested is the access to location, if denied the application will not work; a potential solution to avoid this scenario is to use the proximal location provided by antennas of the IPS, but could result in a very different position than the real one.

Another customization is to allow clear traffic (HTTP), because since from Android 8 for security reasons, only HTTPS packets are allowed by default. In fact, HTTP protocol is an insecure format where data is not encrypted and it's intended to be used just in the development phase, so the explicit declaration in the manifest is required in way to make the Android OS create an exception.

```

1 <application
2   ...

```

```

3 android:requestLegacyExternalStorage="true"
4 android:largeHeap="true"
5 android:usesCleartextTraffic="true"
6>
7...
8</application>
9...
10<uses-permission android:name="android.permission.
    ACCESS_COARSE_LOCATION" />
11<uses-permission android:name="android.permission.
    ACCESS_FINE_LOCATION" />
12<uses-feature android:name="android.hardware.
    location.gps" />

```

Listing 3. Snippet of Android Manifest

VI. HTTP SERVER

The HTTP server takes place as application layer, which represent the bridge between data stored and the client application explained before. The GeoMedia HTTP server uses NodeJS [13] as engine to start a listening process on a designated network port; even more, the server exposes several paths called endpoints, each of which corresponds to a specific function within the application. In fact, to every endpoint exposed, there's an execution command of an SQL stored procedure, passing the input data as parameters and reading the result set as outcome, then returning data in JSON notation to the client requesting.

The packages handled in this platform are in JSON (JavaScript Object Notation) content-type thus to allow a rapid extendability of data included and providing an easily human-readable format. Only the POST method is implemented, since every request/response include data and parameters encapsulated inside the body of the HTTP packet.

The server requires just one external dependency not included natively, which is “tedious” [14] (installable via ‘\$ npm install tedious’) a module to establish a connection to Microsoft SQL Database (DBMS) then to execute query commands and transactions.

As follows a snippet of the HTTP server created in order to handle the GeoMedia HTTP requests.

```

1const http = require("http");
2const port = 9911
3http.createServer((req, res) => {
4    let body = "";
5    req.on("data", (chunk) => {
6        body += chunk
7    });
8    req.on("end", () => {
9        ...
10       body = JSON.parse(body)
11       ...
12       switch (req.url) {
13           case "/doLogin": //LOGIN OR REGISTER
14               dispatcher.doLogin(body.username,
15                   body.password, body.newuser).then(resQuery => {
16                       sendResponse(res, 200, resQuery)
17                   //this method sends back the data
18                   }).catch(err => {
19                       sendResponse(res, 500, err)
20                   })
21                   break;
22       ....
23       case "/getPosts":

```

```

22         dispatcher.getPosts(body?.LATITUDE,
23             body?.LONGITUDE, body?.USERNAME).then(resQuery=>{
24                 sendResponse(res, 200, resQuery)
25             }).catch(err=>{
26                 sendResponse(res, 500, err)
27             })
28         ...
29     default:
30         sendResponse(res, 404, { 'msg': "Unknown path:" + req.url })
31         break;
32     }
33   })
34 }).listen(port);

```

Listing 4. Snippet of GeoMedia HTTP server

The dispatcher class is just a helper, used to wrap the execution of SQL stored procedure and manipulate data whenever necessary.

For example:

```

1const SQL_MANAGER = require("./SQL_MANAGER"); // 
2                                wrapper for tedious package
3
4function checkAREA(areaKM, post_latitude,
5    post_longitude, curr_latitude, curr_longitude) {
6
7    var dLat = (post_latitude - curr_latitude) *
8        Math.PI / 180;
9    var dLon = (post_longitude - curr_longitude) *
10       Math.PI / 180;
11   var a = 0.5 - Math.cos(dLat) / 2 + Math.cos(
12       curr_latitude * Math.PI / 180) * Math.cos(
13       post_latitude * Math.PI / 180) * (1 - Math.cos(
14       dLon)) / 2;
15
16   d = Math.round(6371000 * 2 * Math.asin(Math.sqrt(
17       a))); // in meters
18   if (d <= areaKM * 1000) { //transform areaKM to
19       meters
20           return true
21   } else {
22       return false
23   }
24
25
26async function getPosts(curr_latitude = null,
27    curr_longitude = null, username = null) {
28
29    let posts = await SQL_MANAGER.selectQuery(
30        SQL_MANAGER.loadConfig(), "EXEC GETPOSTS
31        @USERNAME=' " + username + "'")
32
33    let results = []
34    posts.forEach(pp => {
35        if (pp.AREA_KM != null) { // check if users'
36            position is within the post availability
37            if (checkAREA(pp.AREA_KM, pp.LATITUDE,
38                pp.LONGITUDE, curr_latitude, curr_longitude)) {
39
40                results.push(pp)
41            }
42        } else {
43            results.push(pp)
44        }
45    })
46    return results
47}
48
49function getMediaPost(postid){
50    return SQL_MANAGER.selectQuery(SQL_MANAGER.
51        loadConfig(), "EXEC GETMEDIAPOST @POSTID=" + postid
52    )
53}

```

Listing 5. snippet example of dispatcher.js class

A. Vincenty's formulae for distance computation

As shown in Listing 5, the filtering for exclusive content is done partially in the application layer. In fact, server executes the respective SQL stored procedure into the DBMS which already filters the expired posts limited by date, then server computes the visible radius of each exclusive post, then returns just the ones within the users' current location.

The method to compute the distance among the current user location and the position of the limited content within the kilometers chosen, is an implementation of Vincenty's formulae [23].

This method was developed in 1975 by Thaddeus Vincenty [24] a Polish American geodesist who worked with the U.S. Air Force and later the National Geodetic Survey, and is based on the assumption that the Earth is an oblate spheroid, and hence are more accurate than methods that assume a spherical Earth such as great-circle distance [25].

For GeoMedia application, has been implemented just the first method of Vincenty's solution, which computes the location of a point that is a given distance and azimuth (direction) from another point with an accuracy within 0.5 mm on the Earth ellipsoid.

B. Microservices

In the monolithic architecture designed, the server represents a single access node, which is a fragile solution in a real-world scenario, as it represents a "single point of failure" where the failure of this node makes the whole system unavailable.

There are many alternatives to mitigate this event. A potential approach is to replicate the HTTP server across different machines, introducing the challenge of synchronizing multiple main servers.

A more sophisticated solution is to change the whole design, creating a distributed architecture by adopting microservices [15]. With this architecture, server is fragmented into multiple smaller processes that act one for each specific function, as theoretically shown in Figure 5. For example, one microservice could handle the post retrieval process, another could manage the accounts, another the creation of the posts, and so on. This approach not only improves reliability, but also helps to distribute traffic loads thus to preventing the system from an overwhelmed; a common problem with social media platforms when a mass request of a viral content happens, causing the "break the Internet" [26] effect.

VII. DBMS

Following the three-tier design, DBMS (DataBase Management System) represents the Data-layer, providing the function to maintain and to query data in it. Microsoft SQL Server (MSSQL) [16] has been selected to implement this role.

All the posts and user profiles present in GeoMedia are stored in dedicated structure, specifically in database table:

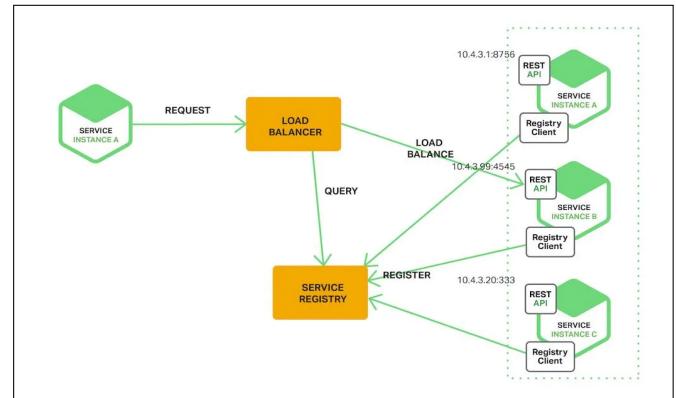


Fig. 5. Example of microservices architecture. [15]

- 'USERS' for users' profiles
- 'POSTS' for posts with all relative attributes and metadata

```

1 CREATE TABLE USERS (
2   ID INT IDENTITY,
3   USERNAME VARCHAR(50) PRIMARY KEY,
4   PASSWORD NVARCHAR(256) NOT NULL, --stored the
5   hash of MD5 function
6 );
7 CREATE TABLE POSTS(
8   ID INT IDENTITY PRIMARY KEY,
9   AUTHOR VARCHAR(50) FOREIGN KEY REFERENCES USERS (
10   USERNAME),
11   TITLE VARCHAR(50),
12   COMMENT VARCHAR(MAX),
13   MEDIATYPE VARCHAR(100), -- IMAGE/JPG, AUDIO/M4A
14   MEDIADATA NVARCHAR(MAX), --BASE64
15   POSTDATETIME DATETIME,
16   COORD_X FLOAT,
17   COORD_Y FLOAT,
18 );
19 
```

Listing 6. Tables creation

Data is handled through stored procedures, which give to the data layer isolation and independency from the rest of architecture, so further modification (both to application or data layer) would not impact each other.

In example, when storing a post, all the content and the author are passed as parameters, then manipulated and formatted into the opportune stored procedure inside the DBMS.

```

1 CREATE PROCEDURE NEWPOST @AUTHOR VARCHAR(50),
2 @POSTCONTENT NVARCHAR(MAX)
3 AS BEGIN
4   INSERT INTO POSTS (AUTHOR, TITLE, COMMENT, MEDIATYPE,
5   MEDIADATA, POSTDATETIME, LATITUDE, LONGITUDE)
6   SELECT @AUTHOR, JJ.title, JJ.comment, JJ.mediatype,
7   JJ.media_b64, GETDATE(), JJ.latitude ,JJ.
8   longitude
9   FROM OPENJSON (@POSTCONTENT, '$') WITH(
10   title VARCHAR(50),
11   comment VARCHAR(MAX),
12   media_b64 NVARCHAR(MAX),
13   mediatype VARCHAR(100),
14   latitude FLOAT, longitude FLOAT
15 ) JJ
16 END;
17 
```

Listing 7. Stored Procedure of post creation

VIII. FUTURE EXTENSIONS

In conclusion, GeoMedia offers an open-source and easily extensible solution for sharing not just visual content but also audio, text, files across a geographical interactive map.

A further idea of development is a *Time capsule* function: storing data inside small sensors distributed through a city, forests, or wide landscapes. These sensors would be reachable only with a short-range medium like Bluetooth or Wi-Fi, or even NFC; so creating a Wireless ad hoc network (WANET) [27] to access the data.

This idea gives life to several use cases, such as way-points to mark hiking routes in the mountains, or acting time capsules in some panoramic points.

REFERENCES

- [1] Google Maps as social media, provided by Google. https://en.wikipedia.org/wiki/Google_Maps
- [2] Street View, functionality of Google Maps. https://en.wikipedia.org/wiki/Google_Street_View
- [3] Instagram, Online Social Media owned by Meta Platforms, Inc.
- [4] Meta Platforms, Inc. https://en.wikipedia.org/wiki/Meta_Platforms
- [5] Instagram stories functionality, to publish time-limited content <https://about.instagram.com/features/stories>
- [6] Meta Accused Of Still Flouting Privacy Rules With AI Training Data <https://www.forbes.com/sites/emmawoollacott/2025/05/15/meta-accused-of-still-flouting-privacy-rules-with-ai-training-data/>
- [7] Foursquare City Guide as City explorer social media. https://en.wikipedia.org/wiki/Foursquare_City_Guide
- [8] Foursquare's official home page. <https://foursquare.com/>
- [9] Foursquare's SwarmApp official home page. <https://www.swarmapp.com/>
- [10] Ionic Framework is cross-platform framework to create web-application and export them to mobile platform. <http://www.overleaf.com>
- [11] Ionic Capacitor - Geolocation package to access to GPS location on mobile devices. <https://capacitorjs.com/docs/apis/geolocation>
- [12] Ionic Capacitor - FileSystem package to store data on mobile devices <https://capacitorjs.com/docs/apis/filesystem>
- [13] NodeJS as server engine <https://nodejs.org/en>
- [14] Tedious - Node Package able to handle SQL Server communication. <https://tediousjs.github.io/tedious>
- [15] Microservices panoramic and image in figure <https://medium.com/the-modern-scientist/introduction-to-microservices-architecture-f0c7efe79f1>
- [16] Microsoft SQL Server 2022 as DBMSQ for data layer <https://www.microsoft.com/en-us/sql-server/sql-server-2022>
- [17] Pigeon Map as React Component to render a geographical map. <https://pigeon-maps.js.org>
- [18] Ionic Storage to store the configuration of server. <https://ionicframework.com/docs/react/storage>
- [19] Ionic Capacitor official website as documentation for the modules installed. <https://capacitorjs.com>
- [20] Android Studio SDK official website, as a tool for building the Android APK. <https://developer.android.com/studio>
- [21] React JS as chosen front-end framework. <https://react.dev/>
- [22] GitHub public repository for GeoMedia project. <https://github.com/albertomorini/GeoMedia>
- [23] Vincenty solutions of geodesics on the ellipsoid. <https://www.movable-type.co.uk/scripts/latlong-vincenty.html>
- [24] Bibliography of the Polish American geodesist Thaddeus Vincenty. https://en.wikipedia.org/wiki/Thaddeus_Vincenty
- [25] Great-circle distance or spherical distance method. https://en.wikipedia.org/wiki/Great-circle_distance
- [26] The Break the Internet effect. <https://dictionary.cambridge.org/dictionary/english/break-the-internet>
- [27] WANET (Wireless ad hoc Network) explained. <https://www.extnoc.com/learn/networking/wireless-ad-hoc-network>