Abstract—Geocrowdsourcing is a new way for institutions to obtain data from the crowd. It makes it possible to harness an effort from the crowd, in order to satisfy a need for information, in which geographic information (locations on the map) plays a key role. Surveys have been, and still are, important means for institutions to gather information from people. One survey might have a simple goal, be simple to perform, and be targeted at a limited population the surveyor has direct access to. But it is common for surveyors to not have direct personal access to their target population, and surveyors need innovative ways to reach their audiences. There might be a geographical component, and a good solution needs to make sure this isn’t a significant burden on participants. If the surveyor needs media like photographs and other digital files, this should also be attainable.

In the old days, surveying problems would have several inherent limitations. Nowadays, devices like Smartphones and Tablet computers have become widespread and popular, not only gradually replacing the old mobile phones, but also fulfilling people’s desire for information, entertainment, and new ways to make use of the Internet. Why not apply this to surveying? Can surveying also be made simpler, with just a couple of screen clicks or taps away from the user?

The implemented system demonstrates several functionalities that when combined provide a useful and cohesive tool for surveyors. It attempts to solve their whole surveying process, from Survey creation, configuration and deployment on a Web platform, with usage of geographical tools, to the management of participants and submission/exportation of Survey answers. This is shown to be possible, along with striving for friendly and easy to use user interfaces that make the participant’s task straightforward and simple.

Index Terms—Survey; crowdsourcing; mobility; geotagging; forms

I. INTRODUCTION

The Geo Crowd Surveys project is aimed at demonstrating the possibility of conducting Surveys that include a geographical component, and media files, using Smartphone technology. Specifically, developing a demonstrative system that provides users with a mobile App using their Smartphones’ location capabilities, cameras, and Internet access, to acquire, fill, and submit Survey answers. This is meant to show that it is possible to simplify the surveyor’s tasks, making them available on a Website, and also making it simple for users to answer and submit Surveys, wherever they may be, just by using an application on their smartphone or tablet devices.

This project is not meant to be restricted to scientific research purposes by closed groups. An analysis of crowdsourcing concepts, and the currently available systems is performed, on two sides. On one hand, the array of geographical/geological smartphone Apps packing a lot of functionality, that are not adequate for surveying a crowd. On the other hand, paid web platforms and commercial applications geared for purposes like market research, opinion polls, etc., but offering little or no geographic capabilities at all. This presents an interesting gap.

This paper presents a solution towards achieving the potential in the rift between the existing solutions. A Server-side project that can be run on a single machine (which doesn’t have the typical privacy concerns the cloud has) and that presents a Website as an interface, plus a smartphone application that can be distributed to participants. The Server project has geographic capabilities, and provides surveyors flexibility, namely through form building, user restrictions/authentication, and ease of exporting the answers. The App should allow users to answer Surveys on the terrain, including media files, and automatically geotag them.

Finally, the solution is evaluated and compared to others, and the obtained testing feedback is presented. Its extensibility and the future evolution of the project are discussed.

II. RELATED WORK

Surveying a crowd with integrated geographical components, as presented in this project, is a relatively unexplored area, even in the current smartphone era. In the old days, before personal computing, a Survey taking location into account would be expensive, typically requiring a dedicated and knowledgeable staff. With the digital age, things became easier, especially given the ability to get the current location using GPS receivers, and the possibility of using of text messages or email to submit answers. Nowadays, in the smartphone era, things are much simpler, but not yet easy enough.

In some cases, location capabilities make a big difference. This is true because some Surveyors have the need to know exactly where the answers they are getting come from. Surveys that can pinpoint occurrences, something happening in real world locations. Also, they may need to target specific surveys at specific areas in the real world.

With personal computing the rise of Smartphones, the general public has become more tech-savvy and able to participate in other people’s projects using the Internet. The recent mainstream introduction of globally-reaching Internet technologies such as crowdsourcing may be a solution to the limited participant pool with which researchers must sometimes work [1]. In developed
countries, smartphones are especially predominant among the younger generations.

Scientists can view these devices as embedded sensors with the potential to take measurements of the Earth’s surface and processes [2]. S. Whitmeyer and D. Paor affirm that if the participants don’t need in depth knowledge of the topic, then Crowdsourcing is practical for rapid data collection [3]. The AMT (Amazon Mechanical Turk) platform [4–6] (created in 2005 and still running) is an example of Crowdsourcing where users get paid a specified rate for each contribution.

Crowdsourcing has evolved and taken new forms, especially in the past 5 years. From a tool where paying a crowd is cited as a necessity [4], into new possibilities where the crowd works for free, or even pays someone for them to create products (i.e. crowd funding).

Some geo crowdsourcing/surveying Apps are developed without having any sort of Server or other devices to establish direct contact with [3], [7]. In these cases, the results have to be exported by the user from their application, using some file format. This implies that a surveyor that is being supported by several users, using the App, has the job of collecting output files from each user.

Another possible alternative is to use pure online form building solutions. Arguably one of the easiest to use, most intuitive form building systems currently available (and free) is Google Forms [8]. It allows for quick and easy building of Surveys, easy sharing with participants, and also an easy way to either visualize results or export them. It does not, however, make use of mobile devices’ location capabilities, or have any type of location/coordinates related functionalities, for that matter. Users are not authenticated. Also, the answers are stored in the cloud, which might arise privacy concerns from users.

Some platforms gather and pay participants (paid crowdsourcing) to answer the available Surveys, like Google Opinion Rewards [9], Panel App [10], or Surveys on the Go [11]. Others have the possibility of “manual” distribution of the Surveys to select participants (via Social Networks, Email, or simply URLs) by surveyors who have a well-defined target audience, and know exactly who they want to answer their Surveys, like SurveyMonkey [12], QuestionPro [13], QuickTapSurvey [14], or LimeSurvey [15]. These Apps and platforms are generally directed at market research and opinion polling purposes, and have limited geographical capabilities or none at all.

III. System Goals, Functionalities and Roles

A. Essential System Goals

A Surveyor might only be interested in applying a Survey to some map areas. For example, a Survey might be meant for some beaches, but not for other areas around them. Surveys can be limited to one area, multiple areas, as in the Cascais Beach Photo Monitoring project [16] or none at all. Therefore, another goal is to enable Surveyors to place Surveys on specific areas of the map. Additionally, a Surveyor might want to limit some of their Surveys to a restricted group of participants, and have other Surveys be available to the entire participant population. Participants should be authenticated, and possibly distinguished from each other, in terms of permissions settings and group assignments.

In principle, this system can be used in a variety of situations. In the simplest situation, the system could be used by a single administrator and a close group of friends or colleagues. The actual envisioned (and possible) use is for a situation where an institution needs help from the general population, and there is an inherent motivation (e.g. altruistic motives) for people to help (like helping an environmental organization, or helping their local community, etc.) without the need for explicit financial rewards people to participate.

B. Basic Roles

Regular users (or “participants”) can answer Surveys, given that they fit the Survey’s user restrictions, and are inside the Survey’s geographical area. Also, a regular user has the option of downloading their own previous answers and files they have submitted. It is up to the Administrators to decide who these people will be.

Basic The people who need data collection are called surveyors. They can build and deploy their own custom surveys (name, description, and adding form fields that can be one of several types). They can also apply restrictions in terms of who is allowed to answer their Surveys, and define areas on the map for the Surveys to be deployed in. The Surveyors can export the data their Surveys have received. This is available as a Web Service. They also have the abilities regular users (participants) have.

Administrators have the power to create other user accounts, appoint surveyor status to other users, create user groups, assign other users to groups, and attribute specific individual permissions to each user or to a whole group. They can deactivate any Survey, even if not theirs. It is possible for them to verify users’ identities, by asking them to log in with Google or Facebook. They can also deactivate accounts, which can be useful to disable users with bad behavior. They also have the abilities Surveyors and regular users have.

The Webmaster is the person responsible for system maintenance. This person has the ultimate power to manage all users. The Webmaster is not necessarily involved in creating content/Surveying activity like the other users are.
C. Solution overview

The system makes use of web-based Internet technologies. The surveys are created and managed using a Website, and are answered by either Website users or smartphone application users. The survey presentation to the smartphone (Android) users is performed by the application. The App is also responsible for automatically using location information and geotagging the answers, and submitting them to a remote server. On the Website, that's done by clicking on a map to indicate positions.

There's an authentication system, with the option of using social authentication providers to identify the users. Participants don't necessarily have to create an account by filling a form or asking an administrator to do it. It is possible for a user to create an account simply by logging in with Facebook or Google. Participants' answers are automatically identified. Surveys have an owner (a surveyor) who can manage the survey and its restrictions, or retrieve the obtained data.

Surveyors can build custom surveys. They can restrict surveys to specific Areas on the map. It is also possible to restrict surveys to specific participants or groups of participants.

There is a Server entity, enabling a website and some web services, where Surveyors and Administrators can create and manage the Surveys and participants. The website also provides a main page where all the users can get an overview of their involvement and access most of the available functionalities. The website also provides an alternative (to the mobile application) way of answering the Surveys, using a web browser, although it requires pinpointing a location on the map, and manually uploading files from the user machine's file system. The Server also provides Web services, namely Survey answer exportation, downloading the submitted files, available Survey information exportation and area description exportation.

D. Main Implemented Features

Part of the Website is restricted, for Administrators and Surveyors (the regular users/participants don't have access to it). Group management is possible, for Administrators. They can manage user groups and their respective permission settings. This means that an Administrator can assign very specific permissions to each specific user group, which makes it easier when compared to just assigning roles individually, to each user. Administrators can also create new users and set their account information; it is possible to “suspend” a user, as there is an option for activating or deactivating an account. It is also possible to outright ban a user. The roles are not set in stone and it’s possible to create new roles and assign them specific permissions.

In terms of social authentication, it is possible for users to authenticate, on the website, using third party authentication providers. For the demonstration version of the system, Facebook or Google (OAuth2 protocol) can be used.

Surveyors can build custom forms. There aren't any pre-made forms, and it's up to Surveyor to decide what to ask from users. This sort of functionality empowers Surveyors and provides them a degree of independence from the system's developer/programmer, as new Surveys can be created at any time, without any developer/webmaster intervention.

There are several usable fields that surveyors can choose. The number of fields for each survey is not fixed, and is unlimited.

The Surveyor can demarcate the areas by clicking on the map and defining the area's polygon vertices. There is no limit on the number of vertices for the area polygons. It's all up to the Surveyor to decide where to place their Areas. These can be used to limit the surveys geographically.

Surveyors can also create and manage their own user restriction templates, to which they can add or remove users and/or user groups. The purpose for this is to deploy surveys that require some specific knowledge, or otherwise private Surveys, to a limited set of users chosen by the Surveyor.

An active survey is a combination of these features, as shown in Figure 1. These must have an associated form, but area and user restrictions are optional, and can change over time. The surveyor can export the answers, the participants can export their answers.

IV. System Architecture/Implementation

A client-server architecture is implemented, as shown in Figure 2 (client processes interact with individual servers, to access shared resources). The client processes are run on smartphones or personal computers. The Server has been tested on a machine running a Linux-based operating system and the Apache server software. Communication between the Server and the clients is performed over the Internet.
Clients access the server’s resources in a REST architectural style, using a set of stateless operations [17].

All the component interactions with the server occur over the same single interface, using HTTP for all services and resources. All resources that are made available to users have a unique resource identifier (URI). For example, there is a base URL for file exportation. The last element added to that URL will be a unique identifier for a file (as in base_URL/export/file identifier). This applies to everything in the server, whether requesting a Web page representing a Survey, making a submission, downloading a photo, exporting results, or exporting Area and Survey informations, and whether the resource is HTML, JavaScript, CSV, JSON, or a media file. Every Survey, file resource, and Area has its unique identifier. Answer submissions are performed to a URL using each Survey’s unique identifier. Some of the resource representations (the Web pages) contain links to other resources.

A. Django Server

The Server was developed using the Django framework, which is a free, open-source web framework that is written in Python. Python (the object oriented programming language) is used almost all throughout it (in this case, Python 2.7), including the settings. The only exceptions to this are found in the web templates that were created (HTML, JavaScript, JQuery, CSS, Django template language).

Django’s extensibility was crucial, as it allowed for adding functionalities through usage of third party Apps and tools. The availability of the GeoDjango [18] module was also very important, because it allowed adding geographical functionality to the project. Also, the authentication system and administrative interface (which also are extensible) were an important asset.

One Django project was developed (corresponding to the Web Application run by the Server). At the moment, it makes use one App which was developed specifically for this project, and others which are third-party, open source apps available to the general public for usage.

The main purposes of the App that was developed from scratch are:

- Integrating 2 of the auxiliary Apps into the project (Forms Builder and Leaflet maps)
- Using GeoDjango functionalities;
- Establishing the main models like the Active Survey model, and the model for answer submissions;
- Extending Django’s User Model;
- Customizing the Django Admin. Namely, defining how the fields are to be presented to the Admin users and their help texts, and defining who can have access to/see what.
- Using Django Forms and Widgets;
- A URL configuration file implementing a RESTful interface. Defining the patterns for the URLs.
- View functions that correspond to each of the URLs, and perform the handling of those requests using either front end technologies or serialization capabilities;
- Front End Web development: creation of several Web templates and static files, used by the view functions to create the responses to requests.

The Django Web Application was developed locally and also tested on a Linux-based (Ubuntu) Server machine, running the Apache Server Software [19], at INESC-ID (Lisbon). Apache handles the requests from the outside, and communicates them to Django, using the Apache WSGI module (mod_wsgi).

1) Crucial Server Features

An important feature that was used is GeoDjango. Simple usage of formats that represent geometric data, like GeoJSON, would be enough, if not for the fact that the ability to perform geographical Querying to the database is necessary. GeoDjango covers that aspect. GeoDjango allows for various different SRIDs (Spatial Reference System Identity) for geometry fields. The default, current standard SRID (4326) was used, which corresponds to the World Geodetic System (WGS) number 84.

Leaflet [20] is an open-source JavaScript library for interactive maps. It was used for implementing geographical functionalities for the Django Website. Django-Leaflet [21] (a Django app) provides the embedding of Leaflet in Django projects. This is useful, as it allows the Geo Surveyors to easily delimit areas of the map by clicking on it, to be stored as Areas of Interest for surveying. Mapbox is a map layer provider that was used to provide the five map layers used by Leaflet, of which two use satellite imagery, and the rest are maps (different colors showing streets, land, and water).

A form building Django App (namedDjango-Forms-Builder) was used [22]. In order to gain some flexibility, and leaving the possibility for, in the future, substituting this form building App with alternative form building Apps/mechanisms, some decoupling was done. This
means that instead of having this App build the forms, serve them, handle their submissions, and handle result exportation, the only thing the App is used for is the actual form building. The other tasks are handled by the main Django App, developed specifically for this project.

Python Social Auth- Django [23] was used, and it provides the ability to configure new authentication backends for a Django Website. Two providers were configured (Google and Facebook (both using OAuth2)).

2) Execution Flow/Survey Presentation

When the Server gets a request for Surveys, evaluations are necessary to come up with the list of Surveys a specific user (the client making the request) can answer at a location. Restrictionless Surveys don’t impose a challenge, but on the other hand the Server needs to serve restricted Surveys only under correct circumstances.

![Active Surveys- Possible Restriction Situations](image)

In Figure 3, for case 1, the Survey has no restrictions and everyone can answer it. In case 2, user and group restrictions need to be checked. In case 3, geo-querying is necessary to determine survey availability for a location. In case 4, both case 2 and 3 checks need to be made.

A Web Page is generated by the Server and presented to the User, listing a button for each Survey that the user can answer given current conditions. By clicking on a button, the user makes a request for another Web page that presents the Survey itself.

Survey answers can be exported in CSV format text files, that store tabular data in plain text. As answers are structured in a tabular fashion in the file, each line represents a single answer.

B. Android Application

In order to demonstrate a working solution, a smartphone application was developed. A preference for developing using Java, using Android Studio, was the key factor in the decision between Android and iOS.

1) App Architecture

The Android application’s architecture is based around the components Activity (typically each of the App's different screens), Service (running in the background), and Intents (that perform the interactions between components).

There is one Activity (that is started as the application is started) that serves as the welcome screen for the application, and that allows the user to turn a background service on or off. The second Activity is used to display online content (webpages) from the Server, within the App. A third activity is used by the user to force the App to update the current location with the highest precision possible, and to manually search for Surveys for the user's current location. There’s a fourth Activity with the sole purpose of defining a custom base class for the other Activities (they extend it). The idea behind this fourth activity is for the other activities to have identical look, namely having an “action bar” at the top of the screen, and presenting an icon on it, that pops up a menu.

![App Architecture](image)

A Web Page is generated by the Server and presented to the User, listing a button for each Survey that the user can answer given current conditions. By clicking on a button, the user makes a request for another Web page that presents the Survey itself.

Survey answers can be exported in CSV format text files, that store tabular data in plain text. As answers are structured in a tabular fashion in the file, each line represents a single answer.

B. Android Application

In order to demonstrate a working solution, a smartphone application was developed. A preference for developing using Java, using Android Studio, was the key factor in the decision between Android and iOS.

1) App Architecture

The Android application's architecture is based around the components Activity (typically each of the App's different screens), Service (running in the background), and Intents (that perform the interactions between components).

There is one Activity (that is started as the application is started) that serves as the welcome screen for the application, and that allows the user to turn a background service on or off. The second Activity is used to display online content (webpages) from the Server, within the App. A third activity is used by the user to force the App to update the current location with the highest precision possible, and to manually search for Surveys for the user's current location. There’s a fourth Activity with the sole purpose of defining a custom base class for the other Activities (they extend it). The idea behind this fourth activity is for the other activities to have identical look, namely having an “action bar” at the top of the screen, and presenting an icon on it, that pops up a menu.

2) Background Service

The idea for the background Service is to have it automatically check for restricted Surveys, and notify the user when these become available. Just checking for Surveys based on some time period having passed might not be the best solution for the app. If the user has not been moving around and just stays in the same place, or moves little, that would mean an unnecessary use of the user’s Internet connection, to make a request to the server.

The practical solution to this implied the app taking movement into account to decide when to make a new check for Surveys. The Service should only makes requests to the (remote) Server if the user has moved over a minimum distance, away from the last position where a check to the Server for Surveys happened. It is possible for displacements to happen, without provoking another request to the Server. A new request may
happen every few seconds, and it will only occur if the user’s distance to the spot of the last request becomes greater than a minimum distance. For demonstration purposes, it was set at 50 meters. The App uses an HTTP client to contact the Server. If area restricted surveys are available, an Android system notification is launched to warn the user.

Google provides the basis for a solution, by way of their APIs, namely using Google Play Services and the Fused Location Provider API [24].

It is also possible for the user to manually look for Surveys, without using the background service. This activity shows the user their current coordinates, also obtained using the Google Play Services. These are updated every second, with the maximum precision that’s possible, and the user gets to know if location is working or not (small movements will make it change, and typically the rightmost decimal digits tend to fluctuate). The user can tap on a button to see the Survey list.

V. System Evaluation

Table 1:
System comparison with similar alternatives

<table>
<thead>
<tr>
<th>Data Storage</th>
<th>Costs</th>
<th>Participant Auth</th>
<th>Social Auth</th>
<th>Offline Func.</th>
<th>Mobile Apps</th>
<th>Browser Answering (via Web Interface)</th>
<th>Form Building</th>
<th>Conditional Forms</th>
<th>Survey Location</th>
<th>Map Areas</th>
<th>Users/Groups Restrictions</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geo Crowd Surveys (GCS)</td>
<td>Own Server</td>
<td>No</td>
<td>OAuth</td>
<td>Yes</td>
<td>Possible</td>
<td>Evolution</td>
<td>Android (Web)</td>
<td>Yes</td>
<td>Yes/Plugin</td>
<td>Possible</td>
<td>Evolution</td>
<td>Auto/Map</td>
</tr>
<tr>
<td>Google Forms</td>
<td>Cloud</td>
<td>No</td>
<td>Anonymous</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ESRI Survey123</td>
<td>Cloud</td>
<td>Yes</td>
<td>Local Account</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Auto/Map</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PDF Forms</td>
<td>-</td>
<td>No</td>
<td>Anonymous</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SurveyMonkey</td>
<td>Cloud</td>
<td>Yes</td>
<td>OAuth</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LimeSurvey</td>
<td>Own Server</td>
<td>Yes</td>
<td>Local Account</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Map</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Open Data Kit collect</td>
<td>Own Server</td>
<td>No</td>
<td>Local Account</td>
<td>No</td>
<td>Yes</td>
<td>Android</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Auto/Map</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Mappt</td>
<td>App</td>
<td>Yes</td>
<td>No Surveying</td>
<td>-</td>
<td>Yes</td>
<td>Android</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>Auto/Map</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>

A. Comparison to other similar systems

To evaluate the System, it is imperative to compare it with other relevant systems. A free form builder/surveyor system (Google Forms [8]), a simple system of distributing PDF forms and submitting answers over the Internet, three commercial systems (the geo-enabled Survey123 [25] and LimeSurvey [15], and the consumer/data analysis focused SurveyMonkey [12]), an open source field data collection tool set (Open Data Kit [26]), and a GIS App (Mappt [27]) were analysed for this purpose.

Mappt is an example of a modern GIS-enabled App that packs a lot of geographic functionalities. In a situation with a group of a few participants it may be viable, but as the number of participants increases the surveyor’s tasks become increasingly tedious and exhausting, which makes this App unusable for crowd surveying.

All Systems except GCS have to rely on the users to know where, on the map, surveys are pertinent and need answers. Survey123, LimeSurvey, and Open Data Kit rely on users reading survey descriptions for that. Having Area restrictions gives GCS a significant advantage if the system is to be used in a crowdsourcing (crowd surveying) situation. Users can be linked to a server that has several Surveys spread around in the real world, without being overwhelmed by the Survey list, and not needing to worry about picking the ones that are relevant to their specific location beforehand.

The same applies to surveys being relevant for certain users and not affecting others. For example, a geologic survey may require specific geology skills to answer. It doesn’t make sense to make it available to every user. Some systems have no restrictions. For example, Open Data Kit will make every survey available to everyone that participates in a given server. For others, the way to impose user restrictions is by having the surveyor manually distribute survey URLs to each participant. GCS is the only that allows surveyors to simply publish their surveys, maybe for every participant, but, when

Fig. 5. User movement, and background Survey checks
appropriate, limit them to specific users, or perhaps more importantly, groups of users. Several different user groups can coexist in the same server without being a nuisance to each other.

Most systems are closed and cannot be changed by outside parties. Besides having flexibility/customizability as advantages, having a system be open to changes can get users closer to it, as they can try to improve it and implement the changes they want. Open Data Kit and LimeSurvey consist of open source code and that has been a strength as it garnered user support to evolve over the years, and GCS can also stand to benefit from being released as open source.

B. Feedback on the System

Feedback on the functioning GCS (demonstration version) system was provided by the Portuguese Environment Agency (APA) [28], that needs surveying systems to get geo-referenced information about coastal areas. The CSV answer exportation feature was deemed positive, as it is a standard format that’s compact, and the possibility of selecting exportation dates is also desirable. The fact that there’s currently only an Android App and no iOS App is not ideal, but the existence of a browser interface that can be used on iOS with a browser somewhat mitigates that. The possibility for social login was also appreciated as an alternative way to identify their participants. The menus (for the Web platform) where users can manage their surveys, survey answers, and look for surveys were appreciated for their user friendliness.

The Coast and Coastal Protection Department at APA are interested in systems to solve their surveying problems, for coastal geological occurrences [29] and were positively surprised by the ability to restrict surveys to areas (and the Leaflet/Mapbox interface for area selection) and showing different surveys according to the user location, and with the ability for user and group restrictions.

With regard to survey presentation and form answering, the button that adds extra file upload fields was welcomed, as their researchers want to get as many terrain photographs as they can, because participants are not photography professionals, and as such the more pictures the researchers have to choose from, the better the quality of results that they can hope to achieve.

C. Mobile Data/Energy Consumption

The application is not particularly demanding, either power-wise or data-wise. It is also relatively light as installation should require under 3 Megabytes of the device’s storage space, for the demonstration version.

If the user turns the background service on, then the location services (likely using GPS, when away from known Wi-Fi networks) will be used. The service is not very demanding on the location services. It was set up to ask for updates only every 10 seconds, and these happen if the location services have determined that there’s been a displacement of over 50 meters from the last location. If there are other applications using location services, these updates might be faster. Furthermore, it stands to reason that using the service only makes sense when the user is moving around and interested in participating.

The background service energy demands were tested on a Samsung Galaxy tablet device (model SM-T113). Wi-Fi was on, to connect to a smartphone that was providing an Internet connection (but no location services at all). The tablet device’s location services were turned on, the application’s background service (automatic survey searching) was on, and testing was conducted on the streets. The device’s screen was turned off for most of the time. In three separate tests, with the duration of exactly 1 hour, the device’s battery level suffered a 3% decrease (after each test). In the exact same circumstances, four tests were conducted using Google’s Google Maps [30] Android application, using its navigation mode (which gets location updates even when the screen is turned off). Two of the tests were conducted with the walking mode, and two using the driving mode. All showed exactly the same energy consumption, as after each 1 hour test the battery level had decreased by 3%. As such, GCS’s Android application background service (which makes use Google’s Play Services location functionalities, configured for relatively slow updates) shows the same energy consumption as the Google Maps Android application in navigation mode.

Under similar conditions (Wi-Fi on, location services on but not being used), but with the GCS application’s automatic search off (background service turned off), after an hour the device either showed no difference in battery level or just a 1% decrease.

The application doesn’t perform any (even moderately) demanding computational tasks, nor does it require any special graphical capabilities from the device.

The application needs an active Internet connection at all times to function properly, being able to contact the remote Server. It will use the device’s current connection, whether it is a Wi-Fi network or a cellular network.

VI. Conclusions and Future Work

A. Conclusions

This thesis presents an innovative solution for surveying with a geographic component. It introduces a new way to survey a crowd, where location plays a key part, and where surveys effectively become features on the map.

The crowd auditing/inquiry systems that are currently available, that are geared to reach big audiences are
well adjusted to consumer/crowd research but don’t present significant geographic components. On the other hand, GIS enabled platforms present better suited solutions geographically-wise, but are tailored towards closed surveying with researcher groups as participants. Scientific studies and research aimed at these problems tend to have shortcomings, generally being very incomplete as they either leave the surveyor or the participants with burdensome, tedious tasks. The resulting system (GCS) results from a study of these solutions, their strengths, and perhaps more importantly, their flaws and what they leave unexplored.

GCS relies on open source technology to achieve its ends. This sort of solution is possible without the need for a big development team or company, because frameworks like Django allow the developer to use several pieces, several tools with pre-packed functionality, and plug them together with their own code to achieve a relatively powerful end product.

A system evaluation was performed that showed that there is currently no dominant, “killer” solution that overshadows all others, and that this area is still open for new developments. It is shown that despite GCS not being that dominant solution, it achieves an interesting alternative, because of a unique functionality set.

In summary, GCS provides a complete solution, from survey creation to answer collection, that is showing some valuable advantages over already existing alternative systems, right from the outset. Open source server-side Apps and tools, a modular server architecture, and the current abundance of open source smartphone libraries present the opportunity to keep evolving the system.

B. Future Work

Overall Extensibility is one of the key strengths of this project. It is not inherently limited to current capabilities, and its architecture allows for vast extension options. The system was planned to be extendable and upgradeable in the future, without requiring major overhauls/significant architecture changes.

In terms of these future possibilities, one of the main sources of interest was the possibility of developing offline smartphone App functionalities.

A particular concern was enabling future use of different form building systems. It’s viable to go a different route and use other (different, or better altogether, according to the practical scenario) form building systems, whether Django Apps or outside tools.

It’s also worthy to note that developing an iOS App, similar in functionality to the Android App that was created, is also possible.

REFERENCES


