

CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Implementation of a Volunteered Geographic Information (VGI)

Mobile Application for Plant Inventory

A graduate project submitted in partial fulfillment of the requirements

For the degree of Master of Science in

Geographic Information Science

By

Amanda Nichole Lindgren

August 2017

The graduate project of Amanda Nichole Lindgren is approved:

Regan Maas, Ph.D.

Date

Edward Jackiewicz, Ph.D.

Date

Soheil Boroushaki, Ph.D., Chair

Date

ACKNOWLEDGEMENTS

As a graduate student coming into the Geography department from a different subject and with very limited background on Android application development, I extend my greatest appreciation and thanks to my advisor, and friend, Dr. Soheil Boroushaki. I never imagined that I would be capable of developing a mobile application if it wasn't for his encouragement, patience, and lending hand. I would also like to thank Dr. Regan Maas for patiently going through and critiquing every paper I wrote in her statistics class. Without her feedback, this paper would have been an even more painful read than it already is. This paper would also not be possible if it wasn't for Dr. Jackiewicz for coming to my aid when I needed it the most. I am also thankful for Tom Chen for the tremendous emotional support and late nights of proof reading. Lastly, I would like to thank the faculty and staff of the Geography and Environmental Studies department for providing such a positive environment for me by being some of the most kindest and supportive people at CSUN.

TABLE OF CONTENTS

Signature Page	ii
Acknowledgements.....	iii
List of Figures	v
Abstract	vi
Chapter 1: Introduction.....	1
1.1: Purpose.....	3
Chapter 2: Literature Review	4
2.1: Volunteered Geographic Information	4
2.1.1: Background.....	4
2.1.2: Volunteered Geographic Information in Today’s Settings	5
2.1.3: Benefits of Volunteered Geographic Information.....	7
2.1.4: Challenges of Volunteered Geographic Information	9
2.2: Mobile Volunteered Geographic Information.....	12
2.2.1: Background.....	12
2.2.2: Mobile Applications in Volunteered Geographic Information	13
2.3: Case Study - Importance of Oak Trees	16
Chapter 3: Methodology	18
3.1: Mobile Operating System	19
3.2: Application Development Tools	21
3.2.1: Extensible Markup Language (XML)	21
3.2.2: Java.....	23
3.2.3 CARTO.....	23
3.2.4 Google Drive	25
3.3: Preliminary Setup.....	26
3.4 Oak Tree Mapper Workflow	28
3.4.1 Map View Activity	31
3.4.2 Add Point Fragment.....	38
3.4.3 View Point Fragment.....	42
3.4.4 Oak Tree Query Fragment.....	44
Chapter 4: Conclusion.....	47
References.....	50

LIST OF FIGURES

Figure 1. General communication between the user interface and external services	19
Figure 2. Chart of global sales of mobile devices based on operating systems	20
Figure 3. List of Android platform versions	20
Figure 4. XML view hierarchy example	22
Figure 5. Device location request prompt	30
Figure 6. Map view activity diagram	32
Figure 7. Message displayed when internet is not connected	33
Figure 8. Flowchart of the major functionalities of the main user interface	37
Figure 9. Flowchart of the major functionalities of the main user interface's menu	38
Figure 10. Add point fragment diagram	40
Figure 11. Device storage request prompt	41
Figure 12. Postgresql sql function for uploading data to carto	43
Figure 13. User interface of the view point fragment	44
Figure 14. User interface of oak tree query fragment	46

ABSTRACT

Implementation of a Volunteered Geographic Information (VGI)

Mobile Application for Plant Inventory

By

Amanda N. Lindgren

Master of Science in Geographic Information Science

Volunteered Geographic Information (VGI), a phenomenon described by Goodchild (2007) as the practice of the general public collaborating in the collection of geographic data, has been taking place for centuries. Over the last two decades, the development of VGI web and mobile applications has increased greatly; however, the majority of VGI applications are web based with an emphasis on georeferencing and inputting GPS coordinates manually. The purpose of this project was to develop a VGI mobile application in order to create a more accessible spatial data collection tool that would provide users with the convenience of recording and uploading data in the field. The application developed for this project (Oak Tree Mapper) focused on generating an oak tree inventory within the state of California. Oak trees were chosen as the focus for this case study in response to a research which used remote sensing to identify and monitor areas of sudden oak death (Kelly 2002). The study was able to identify oak trees that were deceased, but did not confirm whether the cause of death was from sudden oak death. The study suggested that extensive field work would be required to improve the remote sensing methodologies.

The development of Oak Tree Mapper VGI mobile application focused on three major objectives: data collection, data exploration, and data querying. In order to implement these functionalities, the application was developed for Android devices using external services such as CARTO and Google Drive for spatial capabilities and data storage. Overall, the application provides the necessary means to record instances of sudden oak death and could provide the data needed to support remote sensing methodologies.

CHAPTER 1: INTRODUCTION

The practice of participation and collaboration in geographic data collection has been occurring for centuries; the volume of user-generated spatial data has increased drastically with the advances and accessibility of technology within the last two decades. Goodchild (2007) described this phenomenon as Volunteered Geographic Information (VGI), where citizens record, edit, and share geographic data on the web. One of the most recognizable VGI applications today is OpenStreetMap. This web-based VGI provides the general public with the essential tools to create, modify, delete and verify spatial features on an interactive global map. Volunteers, along with researchers, can then apply the spatial data to their own research or use the data for personal use (i.e. locating nearby stores, navigation, etc.) (Coast 2004). This format of data collection has allowed various research projects to benefit by reducing cost and time especially for underfunded organizations. In cities with limited budgets, turning to local community members to gather geographic data provides city officials with updated maps and the opportunity to prioritize the budget based on the data collected (Lathrop et al. 2005; Goodchild 2007; Johnson and Sieber 2013). There are many benefits to incorporating VGI into a research study, but there are also numerous challenges that researchers must consider.

Volunteers that participate in VGI studies normally do not have experience in spatial data collection and can increase the chances of data inaccuracy. Researchers have addressed these concerns and found that user credibility and lack of data management are the main factors affecting the quality of user-generated data (Goodchild 2007; Flanagan and Metzger 2000, 2007; Metzger 2007; Tulloch 2007). Credibility of the participant can be hard to determine especially if the source information about the collected data was not

generated or incomplete. The lack of background information on how the data was created would then deem the data content unviable (Cooper et al. 2011; Flanagan and Metzger 2008). The lack of data management also leads to problematic datasets. VGI is normally used to generate large quantities of data for organizations with limited to no funding and hiring an overseer to manage the incoming data may not be an option. This leads to redundant, outdated, and/or disorganized data, which can affect the overall quality of the data (Flanagan and Metzger 2008; Sester et al. 2014). There have been multiple studies that applied different methodologies to compensate for these challenges, but one of the major improvements to user-generated data was integrating mobile technology as a data collection tool (Song and Sun 2010).

In recent years, the usage of social media applications on smartphones takes up the majority of time a user spends on their phone (Böhmer et al. 2011). The recognition of this upsurge led researchers to developing mobile applications for VGI data collection in order to improve data quality and quantity. There are multiple types of tools built into smartphones that can be utilized to obtain and record data. These tools include: camera, global positioning system (GPS), microphone, compass, accelerometer, and gyroscope (Ferster and Coops 2013). The purpose of these tools differ between studies, but the GPS allows users to record accurate locational information compared to adding spatial features to a map manually. There are still multiple VGI studies that are web-based and create limitations on what participants can do. Expanding these studies to include mobile devices as a tool will then motivate volunteers to participate more in data collection within their community.

1.1: Purpose

The focus of this study will be to implement VGI into a mobile application where users can participate in updating flora inventory (oak trees) within their community. The case study will be centered on developing an oak tree inventory mobile application, called Oak Tree Mapper, for California, that will be able to record the coordinates of oak trees based on the mobile devices' GPS. Oak trees were chosen for this study due to the increasing risks that are threatening their population such as disease, overgrazing, and urbanization (Dahlgren et al. 2003; Gaman and Firman 2006). The participants will also be able to add information about the oak trees they geolocate such as species identification, presence of disease, and other features they may notice on or around the tree. Along with the field notes, the volunteers will be able to upload photographs of the physical characteristics of the trees to help with species identification and to document any additional growths on the tree (i.e. epiphyte and/or parasites). Once the volunteer has completed her/his observations, the location, field notes, and images will be stored and displayed on a map with other tree points collected from other participants. Lastly, the users will be able to: interact with the map and the oak tree points, view information and photos on individual oak trees, and conduct spatial or aspatial query selection based on attributes.

CHAPTER 2: LITERATURE REVIEW

2.1: Volunteered Geographic Information

2.1.1: Background

Goodchild (2007) described a web phenomenon where individuals in the general public would create and share data related to specific geographic locations and he called this phenomenon Volunteered Geographic Information (VGI). Unlike traditional geographic information, all the spatial data are acquired through citizens that volunteer their time and personal equipment for the purpose of making geographic knowledge available for users to apply towards future projects (Goodchild 2007; Elwood, Goodchild, and Sui 2012). Although Goodchild expresses VGI as a web phenomenon, geographic user-generated content has been around for centuries (Goodchild 2007; Muki Haklay 2013; Elwood, Goodchild, and Sui 2012). In 1507, a German cartographer, Martin Waldseemüller, sketched out a continent and labeled it after the explorer Amerigo Vespucci who had explored the continent. Although Waldseemüller later withdrew the label, the name had caught on and the continent was named America (Goodchild 2007). Instances like this one eventually lead to the Board on Geographic Names to be established in 1890 in order to maintain a constant naming system (U.S. Board on Geographic Names 2016). Although the board was created, there was an abundance of spatial features that only the locals knew and map makers would seek help from the public to determine the names attached to the spatial feature. Not all the names supplied by locals matched and a hierarchy of the geographic names would be used to resolve the conflict (Goodchild 2008). Providing place names voluntarily is just one example of previous user-generated content that officials have incorporated into official maps.

Another example that involves user-generated geographic content is the citizen science project, Christmas Bird Count (National Audubon Society 2016). Citizen science (Irwin 1995) is where citizens participate in collecting and analyzing data for scientific projects. Not all citizen science studies require spatial data, but studies that do are referred to as geographical citizen science and can be categorized within VGI (Muki Haklay 2013). Christmas Bird Count has been an ongoing project that started in 1900 and has been the longest running geographical citizen science project in the United States. The project was established by the National Audubon Society and provided volunteers the opportunity to collect bird population data for scientific research. Anyone can volunteer and collect data as long as the accumulated data are within a designated area stated in the guidelines developed by the National Audubon Society. The participants would originally submit their findings through the postal service, but with advances in technology, birdwatchers now have the option of submitting their observations online (National Audubon Society 2016; Elwood, Goodchild, and Sui 2012). The capability of web submission for the study drastically increased the amount of volunteers.

2.1.2: Volunteered Geographic Information in Today's Settings

With the popularity of social media, VGI has become a widespread mechanism for organizations to collect user-contributed spatial and aspatial data (Goodchild 2007). Currently, one of the most well-known VGI applications is OpenStreetMap, where volunteers are able to add and maintain data on roads, restaurants, parks, bike lanes, and much more. The website provides the necessary tools the volunteers need to edit data and to also add any comments on other contributors' work as a validation procedure. An

important feature of OpenStreetMap is open data, meaning that anyone can access and use the data for their own projects (Coast 2004). This is the standard concept for a VGI application, where users can create, update, delete, and download data; however, not all organizations give their users so much freedom. For example, the social media company, Facebook, allows users to assign geographic locations, or geotagging, to their photos, videos, and/or comments; but the company's terms of service gives the company the right to decide how users can access and remove their data (Facebook 2015).

VGI's contribution is not just limited to social media, but has had significant impact during times of crisis (Feick and Roche 2013; Goodchild 2008; Horita et al. 2013; Huang, Chan, and Hyder 2010; Roche, Propeck-Zimmermann, and Mericskay 2013; Zook et al. 2010). In 2005, Hurricane Katrina swept across the south-eastern United States, uprooting many communities along the way. The tropical storm left severe damage to the states along the Gulf Coast, leaving cities damaged beyond recognition with many people reported missing and all traditional communications were down except internet networks. Locals that still had access to web technology started posting information in forums about the aftermath in their community. This quickly led to Scipionus, an online, interactive map that allowed local and global users to publish their own geographic information in regards to the disaster. The people that were close to the disaster would upload data that included updates of damage, status of missing persons, locations of resources and more. For volunteers that were not local, they would provide data such as flood water levels, roadblocks, and other data that could be obtained through satellite imaging. Once Scipionus was launched, it quickly became the most used resource to access locational data in the hurricane disaster zones. The data were not just

used by the public but also by the news media outlets, telecommunication operators, and other website developers (Stoll 2005; Miller 2006; Roche, Propeck-Zimmermann, and Mericskay 2013). Providing interactive web mapping tools, like Scipionus and OpenStreetMap, to crowdsource locational data could prove to be beneficial for those in need.

2.1.3: Benefits of Volunteered Geographic Information

Keeping maps up to date can be costly and time consuming, leaving organizations with limited funding and outdated maps. In locations that were in these situations, the public took it upon themselves to contribute their time and resources towards updating geographic data in order to bring the maps up to date. In Dublin, Ireland, data regarding minor roads and topographies were more up to date on OpenStreetMap and Wikimapia when compared to Google Maps and Bing (Goodchild 2008). Due to limited costs and staff, government organizations were only able to maintain data on major locations within the city. This was problematic for locals and travelers when getting around or determining distance or travel time between places. With OpenStreetMap and Wikimapia, the locals were then able to volunteer geographic knowledge on minor roads, parks, and other points of interest that were not considered formal. Using residents to participate in this informal labeling of place names then becomes more reliable in comparison to official map makers (Goodchild 2008).

Another domain that benefits from VGI is land management and planning. Urban and environmental areas are continuously changing and require an endless upkeep within the municipalities around the world. There is little to no support to account for all the

updates within a city. Having government organizations reach out to the public to collect geographic data reduces costs and provides them the resources needed to develop and employ updates accordingly (Goodchild 2007; Johnson and Sieber 2013).

A paper by Lathrop et al. (2005) noted that VGI is not just limited to creating and updating geospatial data but also to include qualitative and quantitative attributes to spatial features. The project demonstrates that VGI expands into multiple fields outside of geography and can apply to any research that requires geospatial data. The study was able to bring citizens, government personnel (State Department of Environmental Protection), and academic researchers (Rutger University) together to collaborate on building a database in order to locate and determine areas of high density of vernal pools (Lathrop et al. 2005; NJ Division of Fish and Wildlife 2010; Tulloch 2008). Locating these small ephemeral bodies of water were essential since they provide habitats for distinct flora and fauna. The initial portion of the project was done using remote sensing to locate all vernal pools. However, in order to validate the results produced from the remote sensing and spatial analysis, the State Department of Environmental Protection and Rutger University developed an interactive web map that allowed volunteers to: validate existing vernal pools in the database; report new vernal pools not in the database; upload photographs of the pools and any nearby herpetological species; and include any other notes (water level, size, etc.) that may be relevant to the study (Lathrop et al. 2005; NJ Division of Fish and Wildlife 2010; Tulloch 2008).

Incorporating VGI into research can be cost and time efficient as seen in the crisis management of Hurricane Katrina and the vernal pool government collaboration. The flexibility of VGI also offers researchers in various fields a way of obtaining high

quantities of geographic information over large areas in a short periods of time. These are just some of the major benefits that VGI has to offer, however, researchers must also be cautious in their usage of VGI. Participants that volunteer in spatial data collection are not experts nor are they trained in proper field techniques. This leads to the questioning of the validity of public-generated geographic information, and a need for clear quality control measures.

2.1.4: Challenges of Volunteered Geographic Information

In many studies, citizens are the ideal candidates for data collection in local areas due to their endemic familiarity of the region. These projects normally end up with large quantities of data and can be easily accessed online (Rieh and Danielson 2007). However, researchers have voiced various concerns regarding the quality of the user-generated content since a majority of volunteers may have no prior background in the subject of interest. Of these concerns, user credibility and the lack of data management involved in VGI studies were noted the most (Goodchild 2007; Flanagan and Metzger 2000, 2007; Metzger 2007; Tulloch 2007).

The quality of VGI looks at the suitability of the user-generated content and determines whether the content satisfies the criteria required in the designated study (Esmaili, Naseri, and Esmaili 2013). The first major concern that impacts the quality of the data is the person(s) that collect the data. If the project does not include or emphasize on the reliability of the volunteers by including the metadata, the development behind the dataset becomes unknown to those entities that want to incorporate the geospatial content into their studies. There is no easy way to determine credibility of the information

without the source information, which renders the user-generated content unusable. This does not imply that all VGI projects do not provide source information. However, there have been instances where the metadata are available but the information is either in disorder and/or difficult to interpret. It is then up to those using the data to decide whether or not the data from the VGI are trustworthy. If researchers use data with unknown credibility and the assessment of the metadata are incorrect, the outcome could lead to false conclusions and severe consequences (Cooper et al. 2011; Flanagin and Metzger 2008).

The second major data quality concern is the lack of data management in VGI applications. One of the main reasons VGI has increased in popularity is due to the fact that it is cost-effective for organizations that are underfunded. In most occasions, hiring an individual to manage the database is outside the company's budget. The absence of an overseer may lead to a problematic dataset (i.e. disorganized, out of date, redundant, etc.) along with affecting the credibility of the metadata if not available or not maintained (Flanagin and Metzger 2008; Sester et al. 2014). The influence of the metadata and data management credibility on the quality of user-generated data can make VGI integrated research challenging; but previous studies have shown that there are methods to improve VGI quality.

In prior VGI research, the developers would hand select volunteers that were considered to be reliable, based on their education and/or experience, in order to enhance the quality of geographic data. This method only worked for projects that were confidential and with a relatively smaller datasets. For larger projects that required large volumes of data, researchers turned to volunteers for assistance; this led to "peer-to-

peer” evaluations, where volunteers would observe, update, and provide feedback on fellow participants’ inputs in order to reduce redundant, outdated data. The most popular and successful applications that use this method are Wikimapia and OpenStreetMap. Mordechai Haklay (2010) conducted an analysis comparing OpenStreetMap data to the Ordnance Survey dataset showing that approximately 80% of the OpenStreetMap data overlapped with the Ordnance dataset. The reason these two applications are so successful is due to the constant activity of the volunteers updating information. If the volunteers lose interest and participate less, then the peer review method would not work (Flanagin and Metzger 2008).

Hand selecting volunteers and peer-to-peer evaluations only work for specific studies. The development of frameworks to implement into future projects was explored as an alternative to validation of data collection for determining credibility (Rieh and Danielson 2007; Goodchild and Li 2012). Frameworks to assure data quality in VGI depend greatly on the purpose of the application when designing the project and can vary from study to study. There has yet to be a best practice when developing a VGI application, but a majority of them follow very similar approaches (Fonte et al. 2015).

There are multiple aspects that need to be considered and applied to the application. The researchers need to decide whether their applications will allow all citizens to participate or only those with specific credentials. This must be decided at the beginning of the design process since it impacts how much detail will be included in the data collection instructions. If all citizens can participate, too much detail may drive volunteers away and too little can lead to unreliable data. The next aspect that needs consideration is how detailed the metadata need to be. If the volunteer is collecting data

through a mobile device, information about the device (brand, model, camera, microphone, etc.) can be easily obtained through the application. By including the metadata collection into the framework of the application, the credibility of the data becomes more reliable.

The format of the data that will be entered must also be considered based on what the data will be used for and how it will be managed. Data that are recorded as free text can be harder to manage and interpreted compared to inputting quantitative data, selecting from a list of preset values such as categorical data of plant species, or using Boolean logic in the form of yes or no questions. Lastly, data management can either be monitored by assigned credible individuals, volunteers through discussion and updates, or by incorporating a mechanism or model to automatically sift through the data in real-time (Brando and Bucher 2010; Fonte et al. 2015; Foody et al. 2014, 2013; Goodchild and Li 2012; Mordechai Haklay 2010; Vandecasteele and Devillers 2013).

2.2: Mobile Volunteered Geographic Information

2.2.1: Background

The use of mobile applications has been increasing in recent years due to the upsurge in smartphone usage (Petsas et al. 2013). Among these applications, social media categories dominate the amount of time users spend on their phones (Böhmer et al. 2011). With the high usage of social media applications, mobile users become more willing to participate in social activities that they can access through their personal phones. Mobile developers have been taking advantage of this popularity by incorporating mobile technology as a tool to collect user-generated spatial data (Song and Sun 2010). This type

of data collection not only benefits under-funded research projects by saving time and money, but it also amplifies the number of volunteers participating in the project resulting in an upsurge in data quantity.

One of the main advantages of using smart phones as a data collection tool for VGI projects is that they are equipped with multiple features that can be utilized to obtain and document information. This includes accessing the camera, microphone, global positioning system (GPS), compass, accelerometer, and gyroscope (Ferster and Coops 2013). Researchers in various disciplines are presented with the choice of applying VGI methods into their research by volunteers having access to these tools.

2.2.2: Mobile Applications in Volunteered Geographic Information

There are multiple types of research that have implemented VGI mobile applications into the methodology. Some of the research disciplines that have used VGI mobile applications include: public health (Martí et al. 2012; Maisonneuve et al. 2009), ecology and conservation (Sullivan et al. 2009), education (Mayer 2010; Han et al. 2011; Ferster and Coops 2013), and urban environments (Alessandrini et al. 2014; Martí et al. 2012; Maisonneuve et al. 2009). In public health, noise pollution has been gradually increasing around the world due to the growth in urban development and has been shown to have a negative effect on human and animal behavior (Martí et al. 2012). This problem has led to government personnel approving a directive (European Commission 2002) that requires collection of noise pollution data within major cities in order to create action plans for each location. Hiring experts and paying for sound level meter equipment can be costly, especially for cities with budget constraints. In two separate studies

(Maisonneuve et al. 2009; Martí et al. 2012), VGI methods were applied to address the problem and produced mobile applications that could be used for monitoring noise pollution collected by the general public. Both applications, NoiseTube and Noise eQuest/Noise Battle, were successful, however, the design of the applications differed greatly. NoiseTube (Maisonneuve et al. 2009) measured the loudness level of the sound in real-time by applying algorithms to obtain more accurate data. Noise eQuest/Noise Battle (Martí et al. 2012) on the other hand focused more on long term collection and gamifying the application in order to keep volunteers motivated.

For ecology and conservation, a VGI application called eBird that was developed by the Cornell Lab of Ornithology and National Audubon Society gave bird watchers a chance to put their hobby towards science by recording their bird sightings using the camera on their mobile devices. Along with geolocating birds, the users were also able to upload photos, record bird identification, and keep track of all of the birds they spotted. The data would then be uploaded and automatically validated based on the type of bird and the location (based on previous observations and field guides). The amount of volunteers involved with the study was very high due to the birding community already tracking birds for own personal interest (Sullivan et al. 2009).

Not all VGI applications have such high success in promoting volunteers and must use other means to motivate them. An example of this is Project Budburst, created by the National Environmental Observatory Network and Chicago Botanical Garden for educational purposes, where students would take measurements of marked plants within the botanical garden over an extended period of time. In order to motivate long time participants, a game, Floracaching, was incorporated into the application. The game

would award points for revisiting and updating the data for marked plants and allow students to share their scores with others. The results showed that the students were highly motivated by the game and even more so by comparing scores with other participants (Mayer 2010; Han et al. 2011; Ferster and Coops 2013).

The applications that have been mentioned above have required active volunteers that are consistently submitting geographic data such as recordings, images, video, and surveys. Some individuals are not as eager to get out and collect data, but still want to be a part of a local study and develop an understanding of the projects at hand (Devictor, Whittaker, and Beltrame 2010). These volunteers are considered passive volunteers and collect data without engaging in the act of collecting data. They are able to participate by agreeing to let organizations track their daily activities and movements through their personal phone (Muki Haklay 2013). An example of this is an urban environment VGI application called Smart Road Sense which monitors the condition of road surfaces. Smart Road Sense records the movement of the car by accessing the accelerometer and gyroscope on the volunteer's smart phone along with GPS tracking location to determine areas in need of maintenance (Alessandroni et al. 2014; Muki Haklay 2013).

These are just a few examples of how VGI mobile applications can be applied to multiple types of studies. There are still many other studies that are only web-based and limit the participant's access to the spatial data. By expanding data collection tools to mobile devices, citizens will become more willing to participate in community data collection for VGI research.

2.3: Case Study - Importance of Oak Trees

In California, oak trees cover one-eighth of the state's extent (about 13 million acres) (Gaman and Firman 2006). However, an estimate of 82% of oak woodlands and forests are found on private property and cannot be monitored unless granted permission by the landowners (Sulak et al. 2008). What most proprietors may not know is how oak trees benefit the ecosystem and by not monitoring them properly can have a negative impact on surrounding areas (Standiford et al. 2002). A major benefit oak trees provide is improving the quality of California's water shed. About two-thirds of the drinking water obtained within California passes through the soil of oak woodlands where the soil beneath the canopy changes in properties due to the nutrient-cycling process from leaf litter. The decomposition from the fallen leaves resupply the soil with organic matter and nutrients, which enhances the drinking water quality (Dahlgren et al. 2003; O'Geen et al. 2010). Oak trees also contribute to the ecosystem by maintaining biodiversity (California Oak Woodland Community 2016). The canopy forms an understory habitat which provides sustainable environments for native flora and fauna (Fralish 2004; Knoot, Schulte, and Rickenbach 2010). If these trees are not managed properly, then the wildlife habitats and local biodiversity will decrease and soil erosion will occur (Bartolome, Allen-Diaz, and Tietje 1994).

Oaks can be incredibly durable when it comes to harsh winters and dry years; however, this does not make them completely resilient, especially to disease. In 1995, oak trees started perishing at a rapid rate due to a pathogen known as Sudden Oak Death (*Phytophthora ramorum*) (Garbelotto, Svihra, and Rizzo 2001). Within several years of discovering the disease, *Phytophthora ramorum* had already spread across 300km of

California's central coast and was impacting specific species of oaks (Rizzo and Garbelotto 2003). There were multiple methods used in local management to survey and contain the disease but this was difficult since the virus' reach was so extreme (Alexander and Lee 2010). Researchers then turned to remote sensing to identify highly concentrated areas of Sudden Oak Death, but the applied process was only able to identify oak trees that were either dead or alive (Kelly 2002). There have also been attempts to determine the species of oaks using multispectral data with no success (Weigel and Randolph 2013). In order to validate and improve these methods, the results must be compared to field collection of oak tree data. By developing an oak tree inventory mobile application, users will not only be able to record oak tree observation but also help validate the methods of previous studies while becoming educated on the effects oak trees have on the local environment along with becoming aware of the risks (i.e. disease, real estate, and overgrazing) that threaten oak trees today (Dahlgren et al. 2003; Gaman and Firman 2006).

CHAPTER 3: METHODOLOGY

The goal of this project was to develop a VGI mobile application called Oak Tree Mapper to provide volunteers with an accessible tool to record and store spatial data on oak trees. The application would allow the volunteers to view the data geographically and to be able to explore the data interactively on a mapping interface. Oak Tree Mapper was developed to be used on mobile devices with the Android operating system. The development of the application required the use of the Android API (Application Program Interface) libraries from the Android SDK (Software Development Kit) along with Java and XML (eXtensible Markup Language). However, in order to store and manage the data, and to implement spatial capabilities, external services were used. The main external service that was applied to Oak Tree Mapper was CARTO; a software as a service platform that provides spatial and aspatial data storage along with spatial functionalities. CARTO's services were accessed by the Oak Tree Mapper application through the use of the API for Android from the CARTO Mobile SDK. The last major external service applied to Oak Tree Mapper was Google Drive; where images captured by the volunteers would be stored within a public folder and accessed using the Google Drive API. Figure 1 shows the interactions between the components, which will be discussed in detail in this chapter.

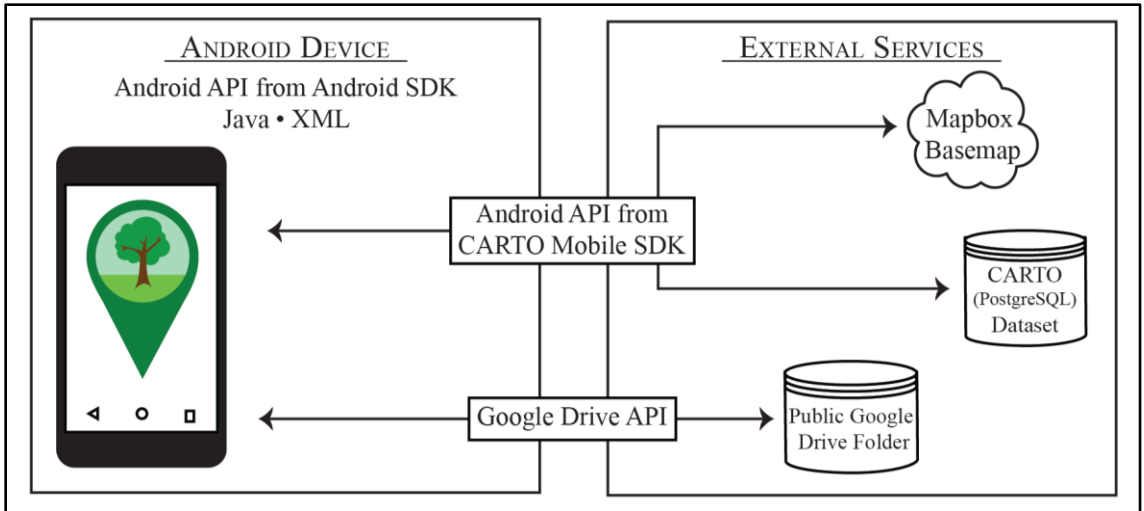


Figure 1: This diagram shows the programming components used to develop the user interface and the specific API libraries responsible for establishing the communication between the user interface and the external services.

3.1: Mobile Operating System

Before the design and development of the mobile application could begin, the operating system had to be determined. There were numerous mobile operating systems, but at the time of this project's production Android and iOS were the most widely used. As seen in Figure 2, between the two operating systems, Android was considered the most common operating system globally with 81.7% of mobile users being Android consumers (Gartner 2016). Within Android, there were multiple platform versions with the latest release of version 7.1 Nougat in December 2016. Although Nougat was the newest version of Android at the time, only 0.5% of Android consumers used this version. According to the recommendations from the Android developer site, applications should support about 90% of Androids active devices (Android Developers 2016). Figure 3 below shows the percentages of active devices based on individual platform versions, which was used to determine the ideal version for this application. In order to support the majority of these devices, 4.4 KitKat was set as the minimum required version, supporting up to 89.1% of active devices.

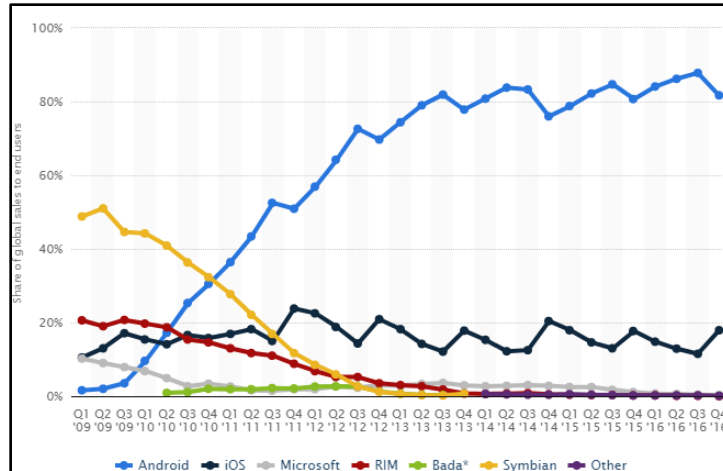


Figure 2: Graph showing the quarterly percentages of global sales of mobile devices based on the operating system from 2009 to 2016. (Gartner 2016)

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.8%
4.1.x	Jelly Bean	16	3.2%
4.2.x		17	4.6%
4.3		18	1.3%
4.4	KitKat	19	18.8%
5.0	Lollipop	21	8.7%
5.1		22	23.3%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24	6.6%
7.1		25	0.5%

Figure 3: Image showing the distribution of active devices for each Android platform version (Android Developers 2016). For more information on the individual versions, please refer to the following link:
<https://developer.android.com/about/dashboards/index.html>

3.2: Application Development Tools

There were multiple development tools and environments that could have been used for building an Android application. For this project, the official software development tool, Android Software Development Kit (SDK), was used for the construction of the Oak Tree Mapper application. Android SDK provides developers with the necessary tools and application program interface (API) libraries to create an Android application (Cornez and Cornez 2015). The API libraries consist of functions that are supplied by Android that allow developers to access their services and data. An abundance of documentation, tutorials, and sample code of Android's API was also available on the Android developer website along with public forums. Although Android's documentation was heavily accessible, experience in XML and Java was required in order to properly use the API libraries.

3.2.1: Extensible Markup Language (XML)

Android SDK generates the user interface of the application using eXtensible Markup Language (XML), which is a markup language that consists of information formatted into elements that are associated to one another as a hierarchy. The elements (i.e. layouts, text, images, user input, buttons, etc.) are displayed in rectangular "views" where parameters can be set to control the style, positioning, margins, sizing, and more. The overall layout and organization of the appearance is controlled by a "view hierarchy" that consists of child "views" nested within parent "views" (Android Developers 2017e). An example of this hierarchy in XML and how it is displayed can be seen below in Figure 4.

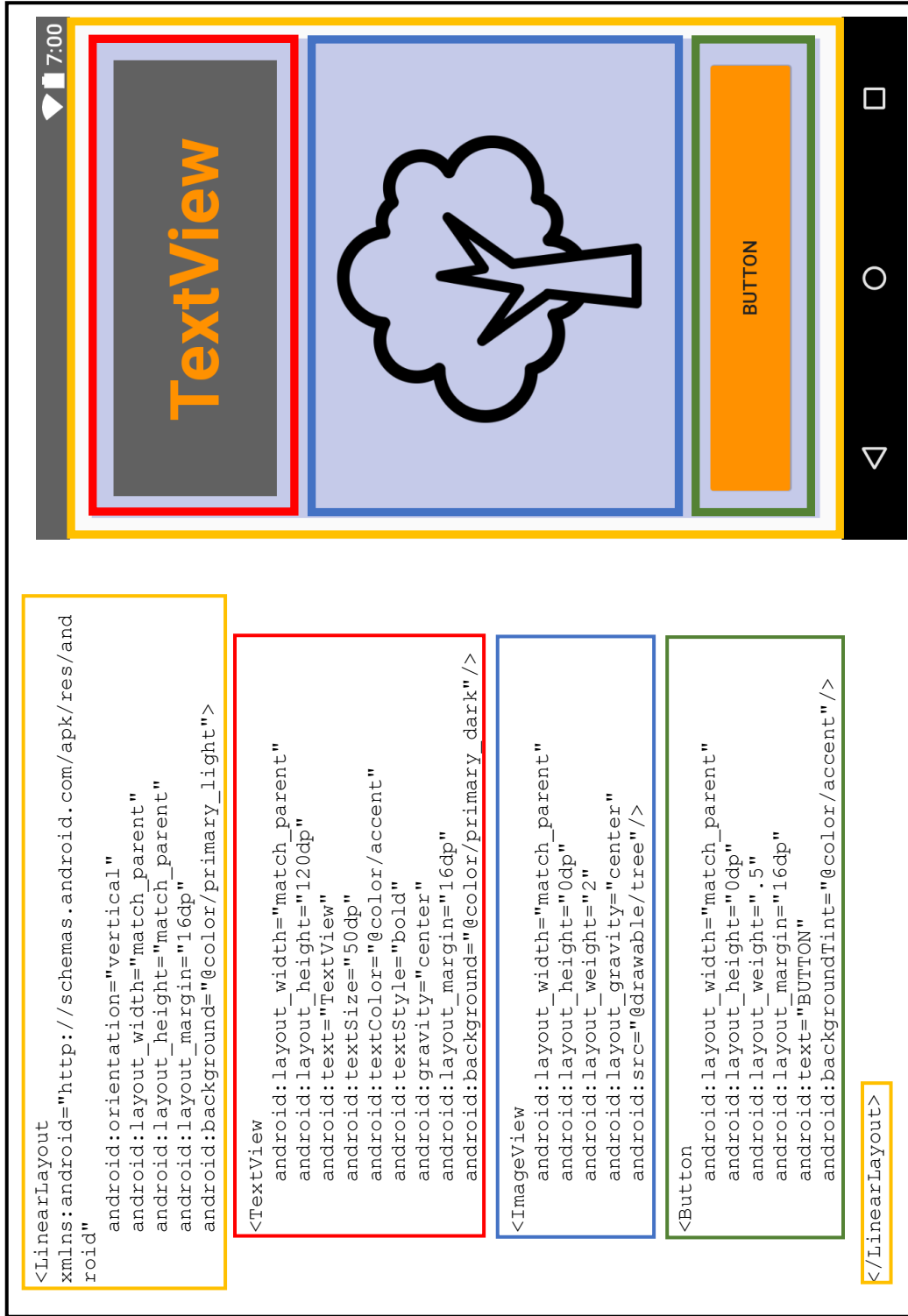


Figure 4: “View hierarchy” in XML (left) with Android user interface (right).

XML is also used for the application's manifest, which provides the Android operating system with the fundamental information about the application and is required in order for the system to run the application. According to the “App Manifest” (2017) on the Android developer website, there are many components that the application’s manifest file is responsible for. For example, when the manifest is created, it generates the application’s identification so that it can be distinguished from other applications on Android devices and the Google Play store. The manifest is also where permissions are stored, which allows the application to access contents and hardware on the user’s phone such as their location, camera, storage, and more (Android Developers 2017b). The application manifest and the design of the user interface are just two of the major uses of XML, but neither of these would function without the use of Java.

3.2.2: Java

Java is a “write once, run anywhere” programming language used for general application development. In Android, Java provides the functionality of the application when the user interacts with application’s user interface (i.e. button clicks, navigating between pages, request permissions, etc.). One of the advantages of Java is that it is platform independent and can run on any operating system with a Java Runtime Environment (JRE) without having to rewrite the code. The JRE uses a Java compiler to convert the code to Java bytecode and translates the bytecode using a Java virtual machine for that operating system (Kurniawan 2015). However, the default runtime environment on the Android operating system, Android Run Time (ART), does not support Java bytecode and uses a different compiler to convert the Java bytecode to

native machine code (Cornez and Cornez 2015). Instead of using multiple software applications to compile the source code, Android Studio (the official integrated development environment for Android) was used since it is considered to be an all-in-one package which included Android SDK and utilized the Java development kit to access the Java libraries and Java compiler (Android Developers 2017g).

The purpose of the Oak Tree Mapper application was to be able to record and store spatial data on oak trees while allowing users to view the data geographically, and to explore the data interactively. All of the development tools provided by the Android SDK that have been mentioned thus far all target the design and functionality of the user interface. However, in order to produce an interactive map and to store public data, implementing additional services into the application development was required.

3.2.3 CARTO

CARTO is a “freemium” (free of charge, but charged for more advanced features), Software as a Service platform built on PostgreSQL (database system) that provides services to develop a web geographic information system (GIS) and web mapping tools for commercial and noncommercial purposes through the use of the PostgreSQL spatial extension PostGIS (CARTO 2017f). The platform targets users and businesses that have little to no experience in GIS and allows them to manage and analyze their spatial data without relying on a GIS specialist. Users are able to access these tools using CARTO Builder, which is a user friendly web application where the user can visually manage their data (CARTO 2017a). The CARTO platform also supplies developers with array of API libraries that range from data management to geospatial

analysis and includes a Mobile SDK that can be downloaded and applied to mobile applications using the SDK's Android API libraries (CARTO 2017c, 2017b, 2017h, 2017g). CARTO also provides clients cloud storage where data can be requested, queried, and managed using CARTO SQL services included in both the Mobile SDK and CARTO Builder. The data are accessed from the services using PostgreSQL Structured Query Language (SQL) which is a language that focuses on database management. There are more products that CARTO has to offer, but for the purposes of this project only the CARTO Builder, Mobile SDK, and cloud storage were used.

The application depends on CARTO for spatial functionalities such as implementation of the interactive basemap and the interactive point (oak trees) layer. Users can zoom and pan the map to explore and view the data, or click on the individual points to bring up a display to see the particular oak tree information. CARTO also provides database functionality through CARTO SQL, which allows users to create, request, and query oak tree data points using the application interface.

3.2.4 Google Drive

One of the goals for the Oak Tree Mapper application was to allow users to capture images of oak trees and to upload the photos to a cloud service that could be accessed later by other users. CARTO, however, only supports spatial and aspatial data in a text format and a separate public data storage was needed to house the images (CARTO 2017e). The benefit of incorporating Google Drive into the application is that the settings of a Google Drive folder can be set to public where multiple people can view and/or edit files in the folder. Another advantage of using Google Drive is that users should already

have a Google account at the time of downloading this application. This allows the users to have access to the Google Drive without having to create a new account for other image hosting services such as Flickr or Instagram (Google 2017a; Flickr 2017; Instagram 2017). Lastly, the Google Developers website provides a Drive API for Android along with an extensive amount documentation and examples for easy implementation (Google Developers 2017a).

3.3: Preliminary Setup

In Android Studio, the default structure of the application is automatically generated when starting a new project (Android Developers 2017c). One of the main modules within the structure included the `AndroidManifest.xml` file which consisted of the application identification along with other information about the application. The application identification is an important component since it is used to register the application to the CARTO and Google Developer services (CARTO 2017i; Google Developers 2017a). Once the services were activated, the library modules (API libraries and other code required to use the service) provided by the services were added to the dependencies section of the `build.gradle` script. The `build.gradle` script is a build system that compiles the source code and other resources the application requires. The Google Play Service Location and Android Support libraries needed for this application were also included within the `build.gradle` script along with the minimum and target versions of the Android SDK which were set at 19 (4.4 KitKat) and 25 (7.1 Nougat).

In addition to storing information about the application, the `AndroidManifest.xml` file is where specific requests are made to the Android operating system. When developing

an Android application, parts of the operating system are not available until permission has been granted to the application. A permission is a security measure that prevents the application from accessing sensitive information on the device unless the user gives consent. There are two types of permissions: normal and dangerous. Normal permissions include parts of the system that do not intrude on the user's privacy and automatically grants the application permission to connect to the requested services. The normal permissions that were applied to the Oak Tree Mapper application were `INTERNET` (connects to the internet through cellular network or wireless fidelity) and `ACCESS_NETWORK_STATE` (checks if the device is connected to the internet). Unlike normal permissions, the dangerous permissions encroach on the user's privacy and the user must be prompted to either accept or dismiss the application's access. The dangerous permissions that were implemented into the Oak Tree Mapper application included: `ACCESS_FINE_LOCATION` (precise location of the mobile device), `GET_ACCOUNTS` (used to access Google accounts on the Android device), `READ_EXTERNAL_STORAGE`, and `WRITE_EXTERNAL_STORAGE` (allows the application to read and write to the devices external storage) (Android Developers 2017i, 2017h).

The last portion of the preliminary setup was setting up the Oak Tree Mapper's data table using CARTO Builder. The data table in CARTO is based on the PostGIS extension of PostgreSQL and contains locational data using the geometry data type (point feature type and the geographic coordinate system [EPSG:4326]) (CARTO 2017d). Along with spatial data, the data table also contains aspatial attributes for each oak tree point. There are eight fields that were included in the data table (Table 1) for this application. The first two fields, "cartodb_id" and "the_geom", are managed by CARTO;

“cartodb_id” is a numeric field that is automatically populated by CARTO Builder sequentially for identification purposes; “the_geom” comprises the point feature of the oak tree in the form of a geometry. The remaining fields that had been added to the data table focused on storing details about individual oak trees. Three of these fields consisted of the oak tree species, presences of sudden oak death, and any further comments about the tree the user wanted to include; the final three fields referenced to unique identifiers of oak tree images stored within the public Google Drive folder designated to the Oak Tree Mapper application.

Table 1: Data dictionary for CARTO dataset attributes. For this project, string fields were not required to have content.

Field Name	Data Type	Definition
cartodb_id*	Number	Feature identification that CARTO Builder sequentially populates for every new oak tree that is added
the_geom*	GeoJSON	Consists of the oak trees geographic location and feature type
image_acorn	String	Unique identification of the oak trees acorn image in the Google Drive folder
image_leaf	String	Unique identification of the oak trees leaf image in the Google Drive folder
image_tree	String	Unique identification of the oak tree image in the Google Drive folder
notes	String	Extra comments about the oak tree
oakspecies	String	Oak tree species
sod	Boolean	Whether the oak tree has sudden oak death or not

* Standard fields created by CARTO.

3.4 Oak Tree Mapper Workflow

The main objective when developing Oak Tree Mapper was to focus on providing volunteers with a tool to collect spatial and aspatial oak tree data with an emphasis on sudden oak death presence. The gathered data would ultimately be applied to validation methods for previous studies focusing on sudden oak death. In order to encompass the fullest capabilities in Oak Tree Mapper, the application was broken into three major categories: data collection, data exploration, and data querying.

Before the objective functionalities of the Oak Tree Mapper could be developed, the essential aspects of the application had to be established. The first major feature necessary for the application to operate was obtaining internet access in order to connect to the CARTO services. Without internet connection, then the application would cease to function until the device was in range of a network source. The second major feature the application required was accessing the mobile device's GPS to obtain the user's geographic coordinates. Based on the best practices from the Android Developers website, Google Play services location API was employed to make the application locationally aware (Android Developers 2017f; Google Developers 2017b). Along with connecting to the Google Play services location API, the application is required to prompt the user to allow access to the device's GPS (Figure 5). After these key components were implemented, the main functionalities and layout of the application could be applied.

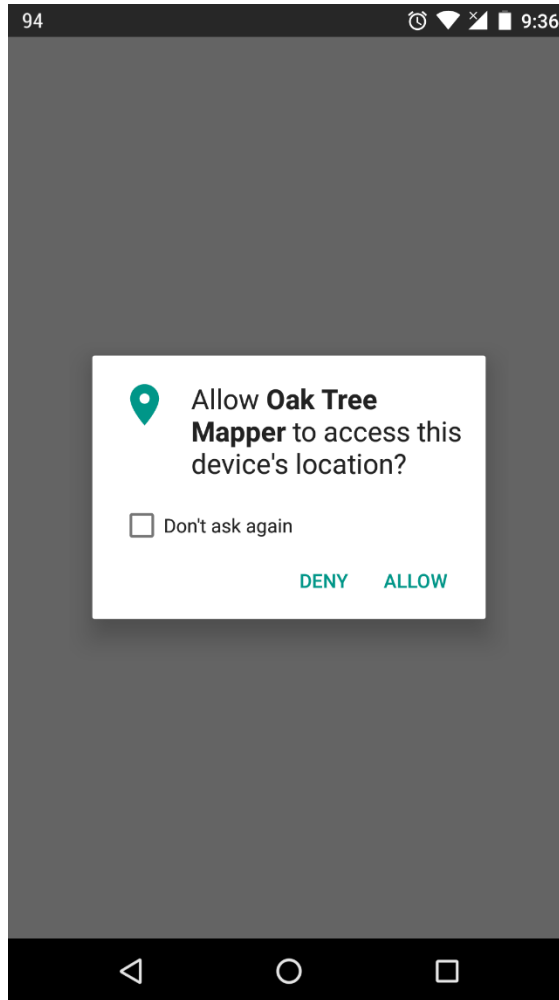


Figure 5: The prompt that is displayed when Oak Tree Mapper requests access to the device's GPS.

When developing the user interface of the application, each page in the application the viewer interacts with is called an Activity. An Activity (part of the Android API) consists of methods supported by Java code that is responsible for the interactivity of the user interface along with assigning the XML layout of the page (Android Developers 2017a). For example, when Oak Tree Mapper is opened, the first page that is called is the `MapViewActivity`, where the method `mapView.getLayers.add(layer)` sets and displays an interactive basemap on the user interface and `setContentView(R.layout.`

`map_layout` displays the XML layout file. One thing to note is that Activities cannot run simultaneously and Fragments (part of the user interface that can be displayed in an Activity) must be applied in order to view multiple windows on one page. Similar to Activities, Fragments contain their own methods and can display XML layouts, but depend on Activities in order to run (Android Developers 2017d). For example, when the user adds a new oak tree point in Oak Tree Mapper, a pop up window (Fragment) opens over the Activity where the user can input information about the oak tree. In Oak Tree Mapper, the main Activity, `MapViewActivity`, provides the core visual components, digital interactive map, and interactivity that allows users to open other Activities and Fragments.

3.4.1 Map View Activity

The first page to display when Oak Tree Mapper is launched is the `MapViewActivity` (Figure 6). Within this Activity, the first operation performed is setting the content of the user interface provided in the `map_layout.xml` file. Once set, the Activity checks whether the device is connected to the internet; if connection is not available, a message will display stating that the device does not have internet access and requires internet connection to use the application (Figure 7). Upon establishing a connection to the internet, the `MapViewActivity` attempts to access CARTO services using two pieces of information: the API key (obtained when the application was registered to CARTO) to initialize CARTO services and the name of the account that owns the data table that Oak Tree Mapper is trying to access (public data tables do not

require a password to access). After MapViewActivity establishes the proper connections, the remainder of the page can be loaded.

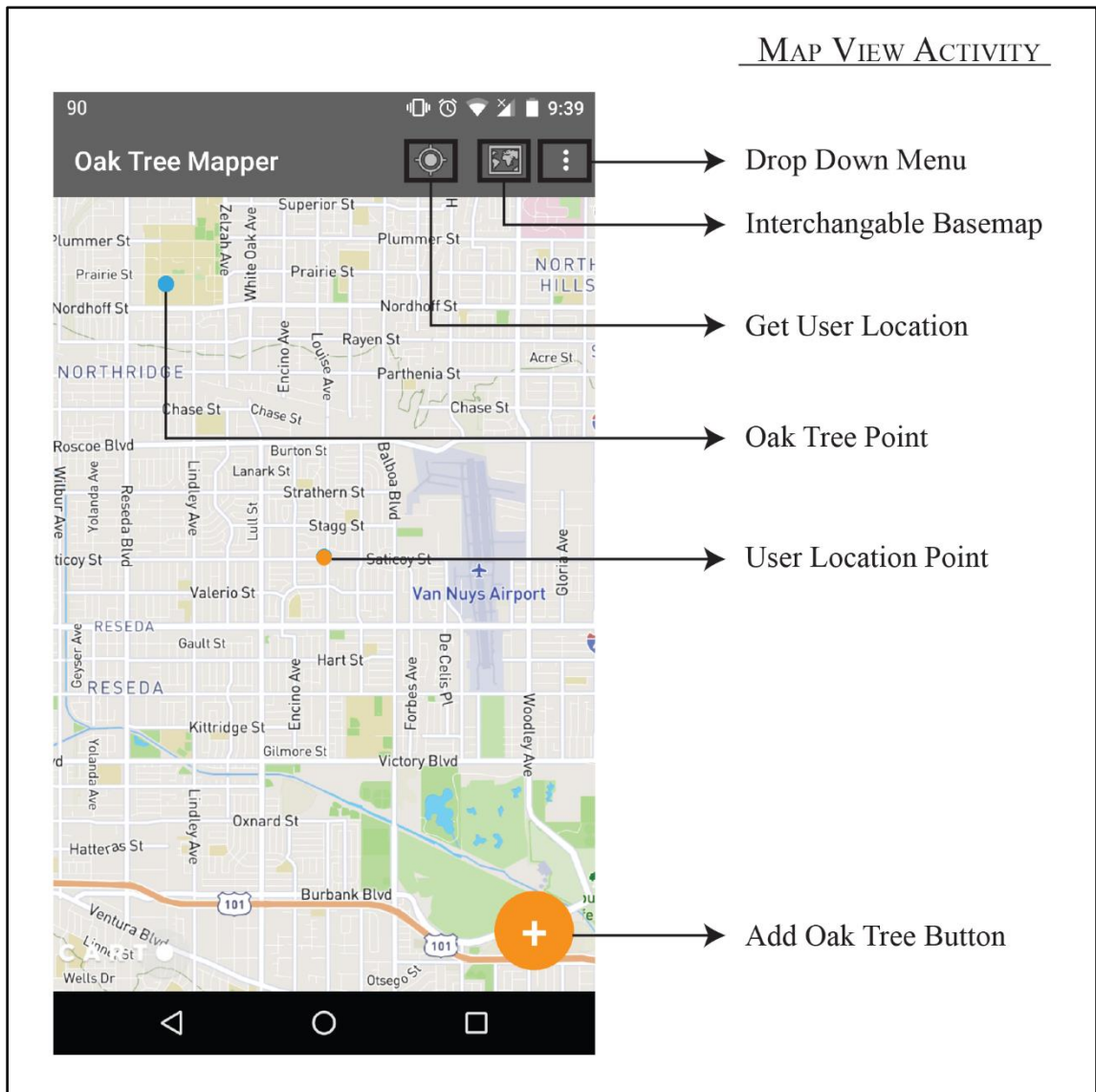


Figure 6: The above diagram is the first page to display when Oak Tree Mapper is launched. The user interface layout is provided from the main_layout.xml file, while the functionalities are part of the MapViewActivity.

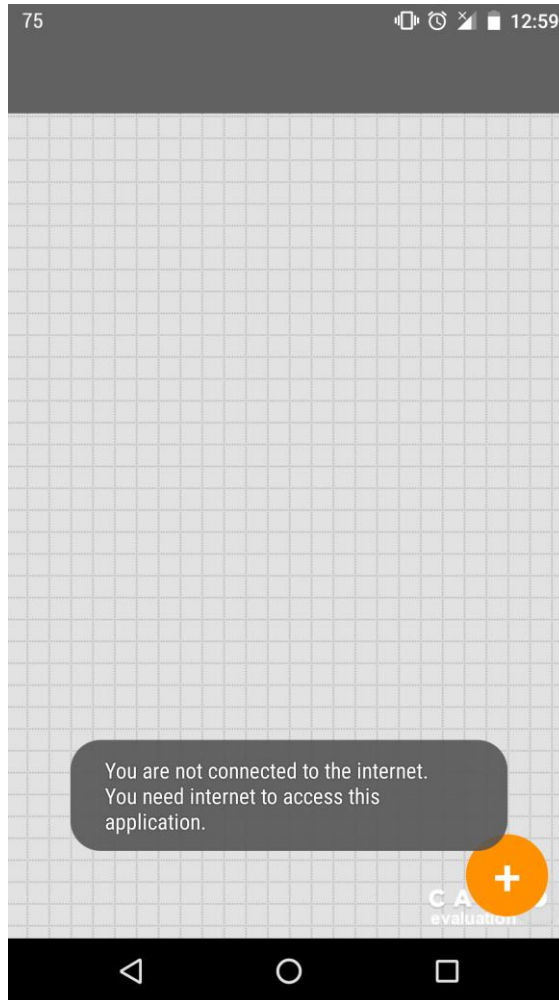


Figure 7: The display of Oak Tree Mapper when internet connection is not available.

When designing the Oak Tree Mapper application, the goal was to make the user interface simple and easy to use with the main layout focusing around an interactive map. In order to display an interactive map, CARTO contains a “MapView” that can be implemented into the `map_layout.xml` file of the Oak Tree Mapper. The “MapView” is a “view” that supports the processing and interaction of the map; however, the map layers must be created and initialized in the `MapViewActivity`. There are two types of layers used for this application, a vector tile layer for the basemap and vector layers for point features. When creating the basemap layer, the first requirement was the CARTO service

which loads the tile data source progressively using a HTTP connection. The tile data source for the Oak Tree Mapper was acquired from Mapbox, a third party mapping platform, using a designated web address for the Mapbox Outdoors style. The tile data source is then used to define the raster tile layer that is added to the map “view”. The process of creating the point features is very similar to the basemap process where data sources are created specifically for vector layers. There are two vector layers in the Oak Tree Mapper application: the oak tree data points and the user’s location. Unlike the basemaps vector tile layer, the vector layer data sources do not require any parameters and are used for storing the point geometry and attributes. Since the oak tree vector layer requires accessing information from the CARTO data set online, a request using the CARTO SQL services is sent to retrieve all the oak tree data from the data set. The data returned are then stored within the vector layer data source and used to create point geometry. The attributes of the oak trees are then stored to the corresponding point geometry within the vector layer data source.

In contrast, the user’s location vector layer requires the mobile device’s location that is obtained through a separate Activity (UserLocationActivity). When activated, the UserLocationActivity checks whether the user had previously granted the application permission to access the device’s location. If the permission was not already approved, then the application prompts the user for access. Once authorization has been given, a request for the device’s location is sent using the Google Play service location API. The location, returned in decimal degrees, is then sent back to the MapViewActivity and the UserLocationActivity closes. Since the coordinates return as decimal degrees, the location is converted into CARTO’s default projection, EPSG: 3857 (Web Auxiliary

Mercator), before storing the location in the vector layer data source. After the vector layer data sources have been defined, the vector layers can then be created and displayed over the basemap.

The default interactivity of any layer created is not defined, with the exception of panning and zooming of the basemap layer. In order to implement the responses to the user's actions, event listeners for the vector layers were established. An event listener is where a function is set to "listen" and activate methods for specific click events (i.e. single, double, long, or dual) on the user interface. In the `MapViewActivity`, a vector element event listener was defined and applied only to the oak tree point features emphasizing on single click events. Any single click event on a point would then display a `ViewPointFragment` on top of the basemap with information and images of the individual oak tree.

The final components of the `MapViewActivity` are the action bar and a button for adding data points. The action bar consists of a menu and the functionalities of each item of the menu. The format of the menu items are comprised in a drop-down menu along with icons displayed on the action bar for items expected to be used frequently (Figure 6). The icons provide the users an easy access for repositioning the map to their current location using the `UserLocationActivity` and interchanging the basemap between the Mapbox Outdoor theme and satellite imagery layout by updating the tile data source. The drop-down menu contains several options that lead to other pages or open Fragments. Two of these options are informative pages about the application and oak trees while the last item opens a tool (`OakTreeQueryFragment`) for querying oak tree data (more information on querying will be discussed later). The last piece of the `MapViewActivity`

is a button that opens a new Fragment (AddPointFragment) that allows users to add oak tree feature data points.

The major responsibility of the MapViewActivity is to provide pathways to the different tools of the Oak Tree Mapper application. As mentioned earlier, the three major functionalities of the application are data collection, data exploration, and data querying. Each of these components are represented within individual Fragments. Two of the Fragments, AddPointFragment and ViewPointFragment, are directly associated with the MapViewActivity while the third component, OakTreeQueryFragment, is connected through the drop-down menu of the MapViewActivity. Figures 8 and 9 provide an illustration of how the pathways connect the different Fragments to the MapViewActivity and other resources.

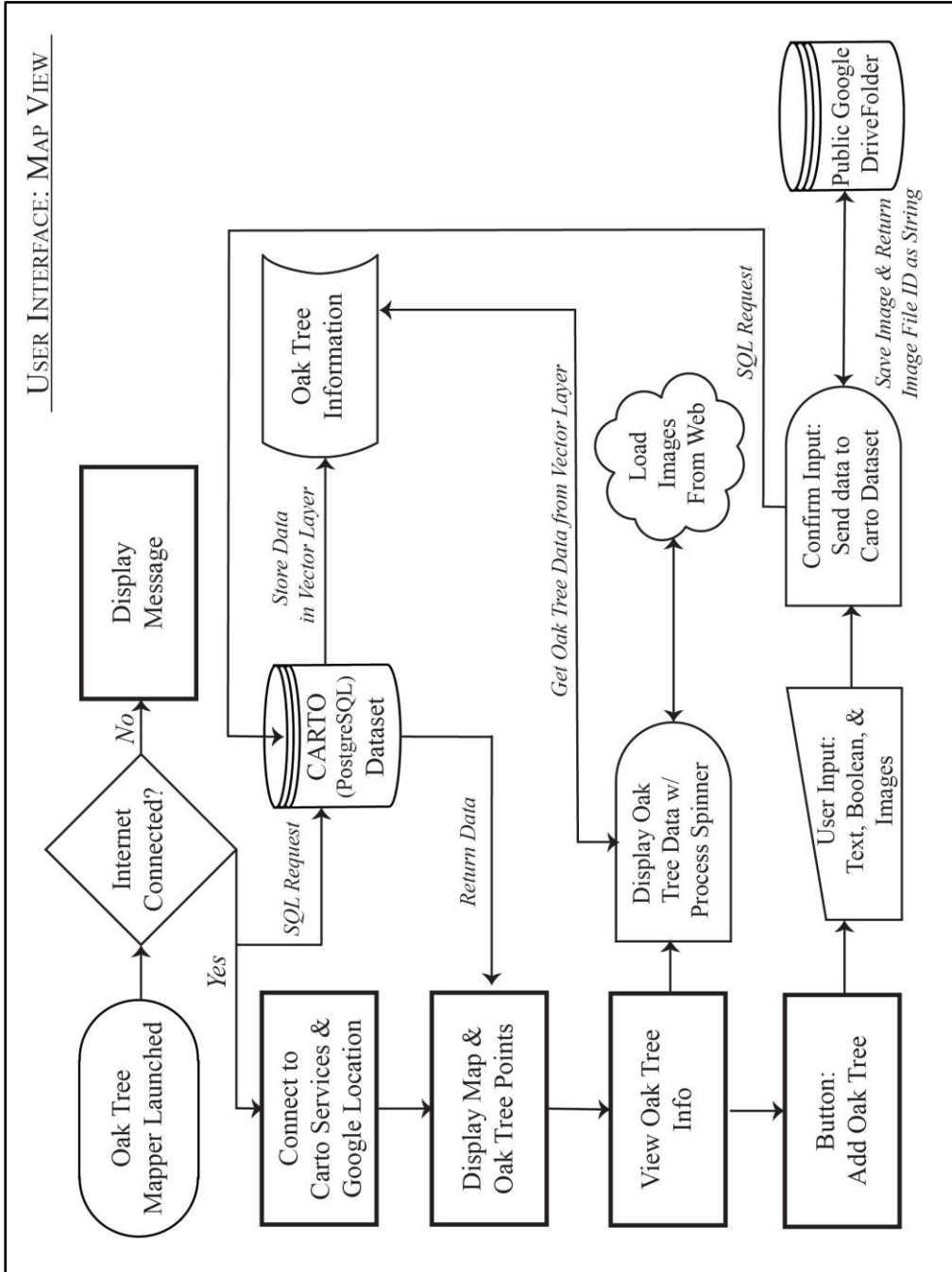


Figure 8: The above flowchart shows the major pathways and the functionalities of the main user interface of the Oak Tree Mappers Map ViewActivity.

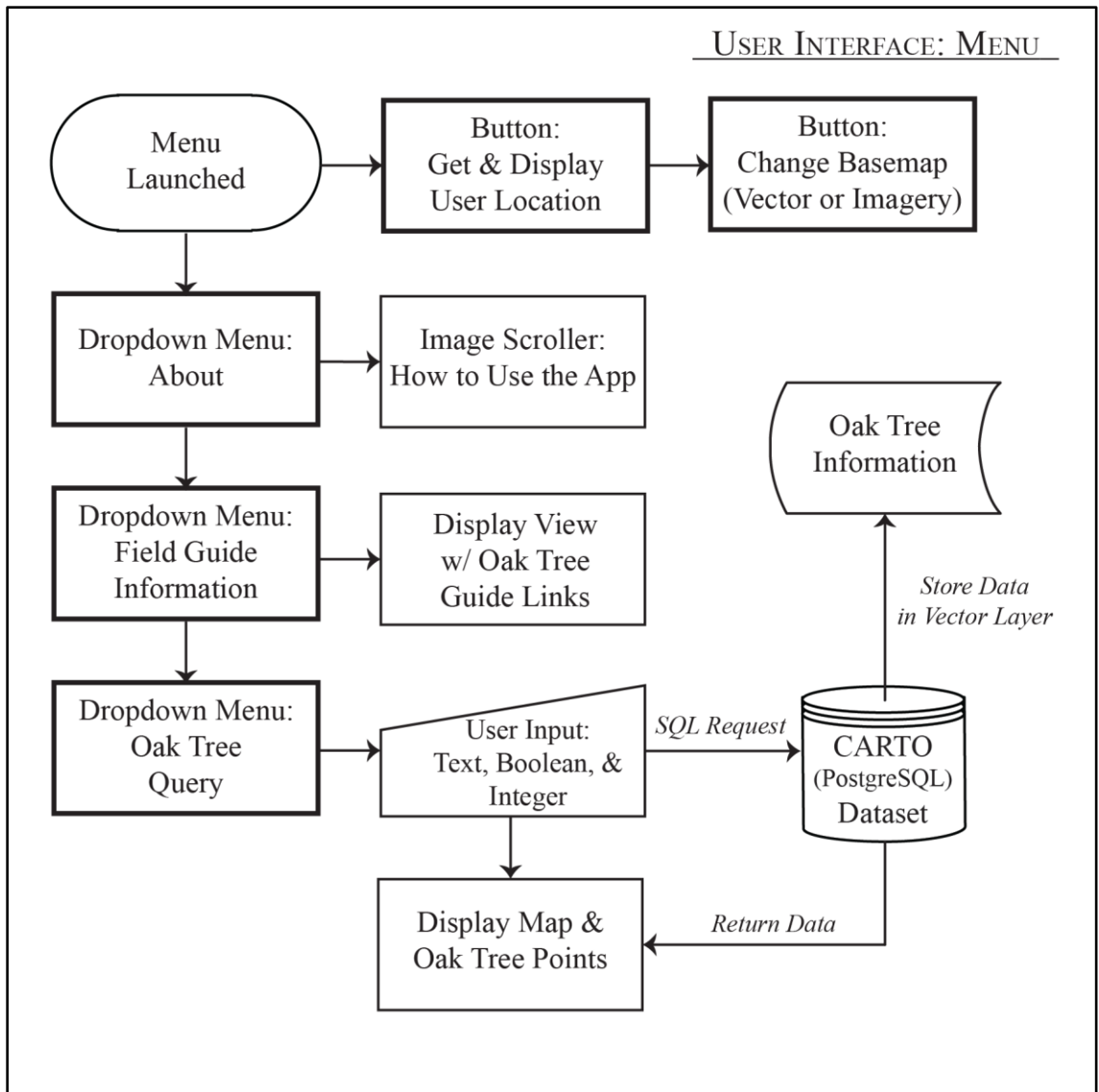


Figure 9: The above flowchart shows the major pathways and functionalities of the menu displayed on the main user interface of the Oak Tree Mapper application. The menu is a part of the MapViewActivity.

3.4.2 Add Point Fragment

On the main user interface created by the MapViewActivity, a button is generated to direct users to the AddPointFragment where oak tree data points can be added (Figure 6). When the Fragment is initiated, a new XML layout (add_point_layout.xml) is overlaid on top of the MapViewActivity in the form of a small window (Figure 10). The design and the functionality of the AddPointFragment allow the user to document information

about oak trees by providing the user with several recording options. The first of these options are the three separate camera buttons to photograph the tree, leaf, and acorn for identification purposes. When any of the camera buttons are clicked, the application first checks whether the device has a camera. If a camera is inaccessible, the application will resume back to the AddPointFragment. Once the application confirms a camera is present, the application proceeds by prompting the user if Oak Tree Mapper can read and write to the device's storage (Figure 11). The storage is necessary to temporarily store the oak tree photos until the user finishes recording oak tree information. Without access to the device's storage, the user cannot capture oak tree photos for the application. If at any point the user decides not to record a data point but already captured photos of the tree, the cancel button (or clicking outside the Fragment) will delete the saved images and close the AddPointFragment.

The remainder of the options cover the attributes of the oak tree such as identification, verification of sudden oak death infection, and any other observations the user notices. The identification of the oak tree is determined by the user with the help of the field guide links in the MapViewActivity menu. Along with the field guide links, there is background information on sudden oak death which include images of how the disease appears on oak trees. Since the data will be available to query, the species common names were presented as a drop-down list and any sudden oak death observations were recorded as a Boolean (true/false) data type in the form of a checkbox to prevent imprecise queries due to spelling errors. The last option the user can use to document attributes is an editable text "view" where users can input any notes they have

on the oak tree. Unlike the oak tree identification and the verification of sudden oak death, the comments will not be used as a querying parameter and are recorded as is.

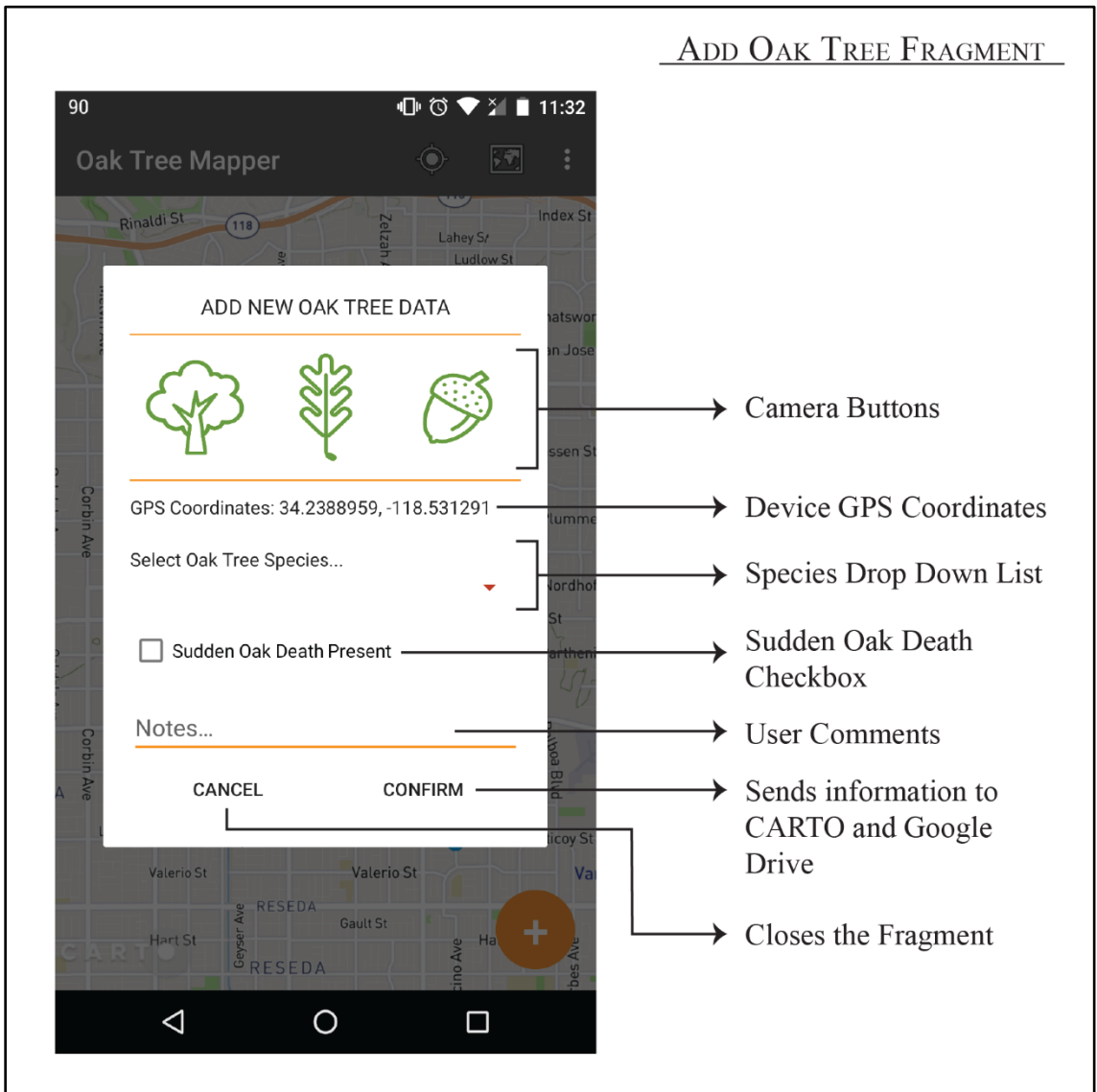


Figure 10: This diagram is the AddPointFragment displayed over the MapViewActivity. This window provides users with the tools to add oak tree data points and attributes. The content of the user interface is set using the add_point_layout.xml file.

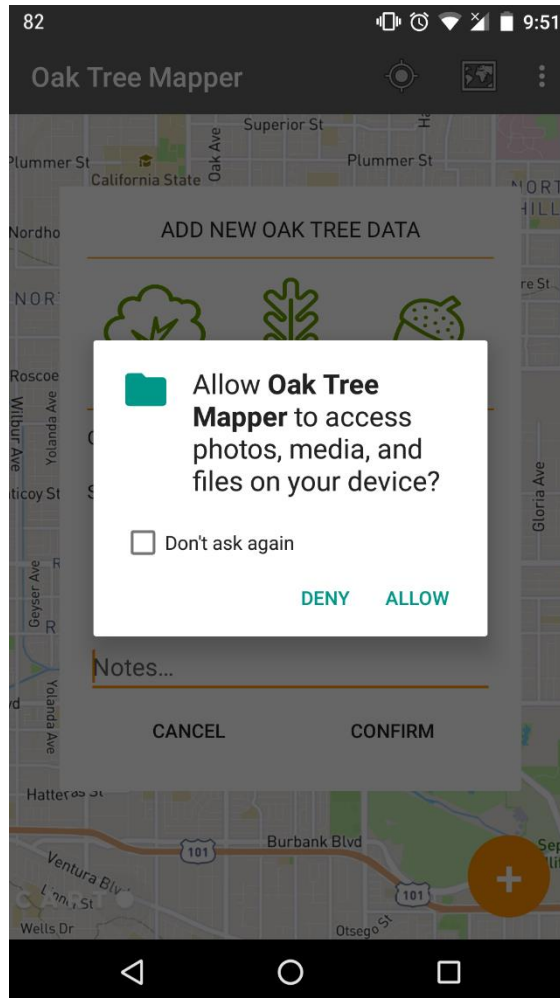


Figure 11: The prompt that is displayed when Oak Tree Mapper requests the user in order to read or write to the device's storage.

Once the user has completed documenting the oak trees information, the data are uploaded using the confirm button. As mentioned above, the images are stored in a public Google Drive folder which requires the application to prompt the user to login into their Google account. If the user rejects the request, the application will stay on the AddPointFragment display; otherwise the application will connect to the public folder. After the connection has been confirmed, the oak tree images will upload to the folder where each image is assigned a unique file ID; which are then returned to the application as a string. Once the file identifiers have been returned, all the information recorded in

the AddPointFragment along with the image identifications are uploaded to the CARTO data table using CARTO SQL services.

In order to upload the data to the data table, a function had to be created in CARTO Builder using the Dataset Options editor window. The parameters of the function (Figure 12) were set to accept inputs for each field within the data table (Table 1) including the geometry field (the_geom) which consisted of the latitude and longitude of the oak tree. The function would then insert the data into the corresponding fields within the table; any missing text values (images, identification, and notes) would be assigned an empty string before uploading to the table. After the CARTO SQL services uploads the data, the temporary images stored on the phone would then be removed and the AddPointFragment will close. Lastly, the MapViewActivity's interactive map would update with the new point added to the map.

3.4.3 View Point Fragment

In the development of the oak tree point features, a vector element event listener is setup to open the ViewPointFragment when a user clicks on a point providing a convenient method for data exploration. Similar to the AddPointFragment, the display of the ViewPointFragment overlays on top of the MapViewActivity in the form of a small window (Figure 13). Within the layout, the ViewPointFragment displays the attributes for the specific oak tree data point that was clicked on. However, before the information can be displayed, the images must be retrieved from the public Google Drive folder using the assigned file identification appended to the end of the following web address: [https://drive.google.com/uc?id=\[Google Drive Image ID\]](https://drive.google.com/uc?id=[Google Drive Image ID]). After the images are obtained, they are displayed as thumbnails which can be enlarged when clicked.

```

DROP FUNCTION IF EXISTS oakApp_addData(numeric, numeric, text, text,
boolean, text, text, text);

CREATE OR REPLACE FUNCTION oakApp_addData(
lat numeric,
lon numeric,
species text,
notes text,
sod boolean,
acorn text,
leaf text,
tree text)

RETURNS TABLE(cartodb_id INT)

LANGUAGE 'plpgsql' SECURITY DEFINER
  RETURNS NULL ON NULL INPUT
AS $$
DECLARE
  sql text;
BEGIN
sql:= 'WITH do_insert AS (
      INSERT INTO acorn_table(the_geom, oakspecies, notes, sod,
      image_acorn, image_leaf, image_tree) VALUES
      (ST_SetSRID(ST_MakePoint('||lon||','||lat||'), 4326),'
      ||quote_literal(species)||','
      ||quote_literal(notes)||','
      ||sod||','
      ||quote_literal(acorn)||','
      ||quote_literal(leaf)||','
      ||quote_literal(tree)||')'
      ||'RETURNING cartodb_id)'
      ||'SELECT cartodb_id FROM do_insert';

RETURN QUERY EXECUTE sql;

END;
$$;

--Grant access to the public user
GRANT EXECUTE ON
FUNCTION oakApp_addData(numeric, numeric, text, text, boolean, text,
text, text)
TO publicuser;

```

Figure 12: Above is the function used to insert data into the CARTO data table. The function was created and stored in the database using the CARTO SQL services provided in the CARTO Builder's Dataset Options. When needed, the application would connect to the services and call the function with the required parameters.

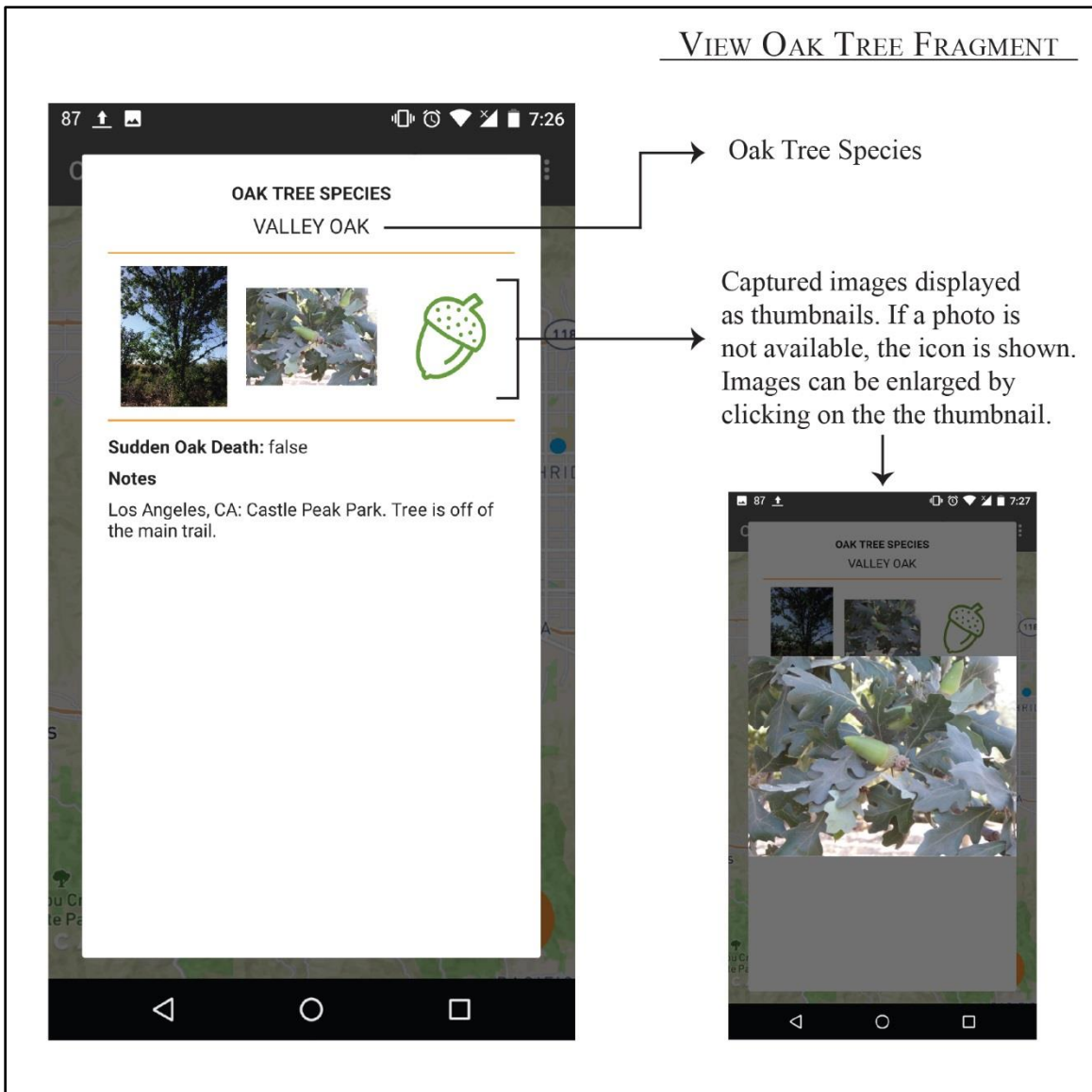


Figure 13: This diagram is the ViewPointFragment displayed over the MapViewActivity. This window provides users with information on oak tree data point that was clicked on. The content of the user interface is set using the view_point_layout.xml file.

3.4.4 Oak Tree Query Fragment

The last major function that the Oak Tree Mapper offers is querying oak trees based on spatial and aspatial parameters. The tool is accessible through the drop-down menu of the MapViewActivity with the XML layout (query_layout.xml) in a similar layout as the add and view point Fragments (Figure 14). The spatial parameter that the oak trees can be queried by is the distance based on the current coordinate of the device

using either kilometers or miles, while the the aspatial parameters include oak tree species and the presence or absence of sudden oak death. The method of input is similar to that of the AddPointFragment using drop-down lists and an editable text “view” with the inclusion of radio buttons instead of a check box for sudden oak death. Once the user confirms the constraints of their query, a SQL statement is concatenated together, sent back to the MapViewActivity, and the OakTreeQueryFragment closes. In the MapViewActivity, the CARTO SQL services are then used to request data from the CARTO data set based on the parameters provided in the SQL statement. Before the services are called, the vector data source and vector layer are cleared of all data in order to house the returned oak tree point features from CARTO SQL services. The updated vector layer will then be displayed on the basemap with the extent of the map showing all oak trees that meet the criteria of the query. However, if there are no data that meet the queries constraints, the vector layer will not exhibit any points. When the user is done viewing the queried data, the user location menu icon can be used to reset the map to display all oak tree data or a new query can be initialized.

These three primary functions of this application provide a simple geospatial data collection tool that can be applied to studies that require gathering information in the field. The Oak Tree Mapper was developed specifically to be applied as a validation method for geolocating oak trees with sudden oak death along with oak tree identification. However, depending on the purpose of the project, the application can be developed to suit the needs of any VGI project.

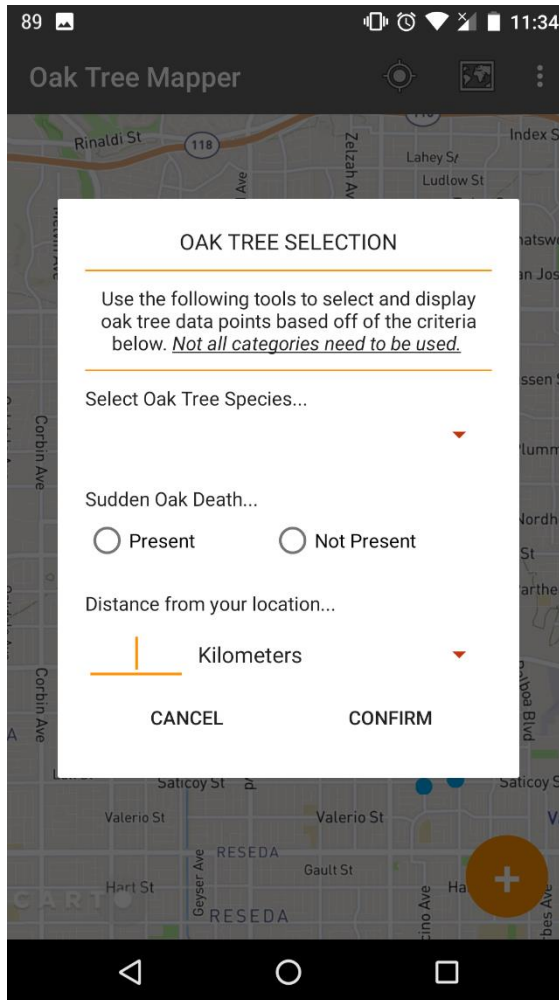


Figure 14: The above is the user interface of the OakTreeQueryFragment used for querying oak tree data points. One or more of the options can be used for querying.

CHAPTER 4: CONCLUSION

The development of the Oak Tree Mapper focused on implementing VGI as an approach to validate previous remote sensing methodology that aimed to identify oak tree species and check for presence of sudden oak death. The overall functionalities of Oak Tree Mapper were developed successfully with several limitations that need to be taken into consideration if similar applications were to be developed. The first limitation of Oak Tree Mapper is that it requires internet in order to function, which can be problematic in remote areas that do not have wireless fidelity and/or cellular network access. In order to address this problem, the application would need to have offline capabilities where a user can temporarily store recorded data on their device until the internet is available to upload the data to the designated cloud storages. The disadvantage of taking this course is that it would require the application to continue running in the background after the application has been dismissed. This method would be costly on the battery unless the user manually stopped the application. The usage of the device's battery reserve could be avoided by storing data on the device until the application is used again in an area with internet access.

Another approach developers could consider is to not change the internet requirements and acquire the data through other methods. One example of this is iNaturalist (a citizen science web/mobile application used for recording biodiversity observations) which cannot operate without internet access, but allows users to upload photos that were taken from devices with a GPS. iNaturalist will obtain the coordinates directly from the uploaded photo or request the user to manually input the location if the location is not available ("iNaturalist.org" 2017). These are just two examples of how the

internet limitation can be addressed, but depending on the specifications, developers can apply multiple solutions to fit their needs.

The second limitation of Oak Tree Mapper that needs to be taken into consideration are the services of the CARTO platform and Google Drive. Both CARTO and Google Drive are freemium platforms that require payments for more advanced features. Since Oak Tree Mapper was only used for testing purposes, the advanced features of CARTO and extra storage capacity of Google Drive were not necessary. If the project were to expand, the free services CARTO and Google Drive provide may not be enough and paying for the advanced features may be necessary (CARTO 2017j; Google 2017b). An alternative to using external services is to house a private server that supports PostgreSQL with the PostGIS extension. Both spatial/aspatial data and images can be stored in the database (PostgreSQL) and all spatial functions can be applied using the PostGIS extension (“PostGIS — Spatial and Geographic Objects for PostgreSQL” 2017, “Chapter 7. Storing Binary Data” 2017).

In addition to the Oak Tree Mapper’s limitations, the application still lacked in validation techniques to increase the reliability of the data collected by the volunteers. Since the Oak Tree Mapper was just a prototype, including validation methods did not seem applicable without volunteers to test the application. However, if the application were to be built upon and tested, developers would need to consider implementing a validation process into the application.

Incorporating any validation methods into an application can be challenging since there are no best practices to follow. For the Oak Tree Mapper, only two attributes would require validation: the oak tree species and sudden oak death presence. Since the

volunteers would be capturing images of the oak trees, either volunteers (peer-to-peer) or administrators could update inaccurate data without going into the field. If a peer-to-peer approach were to be taken, more concerns arise on whether volunteers should be given editing capabilities and if all edits should be documented. This would require another data table to store the edit history and more storage would be required. If a project is able to obtain the additional storage, this route would be ideal; however, for studies that are limited on resources, an administrator to conduct the updates would be the most affordable option.

The overall takeaway from developing the Oak Tree Mapper is that there is no one specific approach to building a mobile VGI application. This application specifically targeted mobile users with Android devices; however, depending on the project and the area of the study, the most widely used operating system can change from place to place. In situations where statistics on mobile device usage are not available, developers also have several options to target all mobile devices. These include, but are not limited to, choosing a programming language that can code across multiple platforms or creating a web application with an implemented mobile view mode to be accessed on any mobile devices web browser. There are also code free application builders available for users that have little to no coding experience. However, these can only build basic applications that do not require advanced functions. These are some examples of different approaches that can be applied when developing a mobile VGI application with the Oak Tree Mapper as a one approach. With the various software packages and programming languages available, developers have numerous methods they can apply to future mobile VGI applications.

REFERENCES

- Alessandrini, G., L. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. Paolini, A. Seraghi, et al. 2014. "Smartroadsense: Collaborative Road Surface Condition Monitoring." *Proc. of UBIKOMM-2014. IARIA*.
- Alexander, Janice, and Christopher A. Lee. 2010. "Lessons Learned from a Decade of Sudden Oak Death in California: Evaluating Local Management." *Environmental Management* 46 (3): 315–328.
- Android Developers. 2016. "Dashboards | Platform Versions." *Dashboards | Android Developers*. November 7. <https://developer.android.com/about/dashboards/index.html>.
- . 2017a. "Activity | Android Developers." Accessed June 22. <https://developer.android.com/reference/android/app/Activity.html>.
- . 2017b. "App Manifest | Android Developers." Accessed June 2. <https://developer.android.com/guide/topics/manifest/manifest-intro.html>.
- . 2017c. "Create a Project | Android Studio." Accessed June 16. <https://developer.android.com/studio/projects/create-project.html>.
- . 2017d. "Fragment | Android Developers." Accessed June 22. <https://developer.android.com/reference/android/app/Fragment.html>.
- . 2017e. "Glossary | Android Developers." Accessed June 22. <https://developer.android.com/guide/appendix/glossary.html>.
- . 2017f. "Making Your App Location-Aware | Android Developers." Accessed June 21. <https://developer.android.com/training/location/index.html>.
- . 2017g. "Meet Android Studio | Android Studio." Accessed June 7. <https://developer.android.com/studio/intro/index.html>.
- . 2017h. "Normal Permissions | Android Developers." Accessed June 17. <https://developer.android.com/guide/topics/permissions/normal-permissions.html>.
- . 2017i. "Requesting Permissions | Android Developers." Accessed June 17. <https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous>.
- Bartolome, J. W., B. H. Allen-Diaz, and W. D. Tietje. 1994. "The Effect of *Quercus Douglasii* Removal on Understory Yield and Composition." *Journal of Range Management*, 151–154.
- Böhmer, Matthias, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. "Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage." In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, 47–56. ACM.
- Brando, Carmen, and Bénédicte Bucher. 2010. "Quality in User Generated Spatial Content: A Matter of Specifications." In *Proceedings of the 13th AGILE International Conference on Geographic Information Science*, 11–14. Springer Verlag: Guimar aes, Portugal.

- California Oak Woodland Community. 2016. "California Oaks." Non Profit Organization. *California Oak Foundation*. Accessed October 31. <http://new.californiaoaks.org/>.
- CARTO. 2017a. "CARTO Builder — CARTO." Accessed June 12. <https://carto.com/builder/>.
- . 2017b. "CARTO Documentation — CARTO Documentation." Accessed June 12. <https://carto.com/docs>.
- . 2017c. "CARTO Engine — CARTO." Accessed June 12. <https://carto.com/engine/>.
- . 2017d. "FAQs — Datasets and Data — CARTO Documentation." Accessed July 14. <https://carto.com/docs/faqs/datasets-and-data>.
- . 2017e. "Import API — Importing Geospatial Data — CARTO Documentation." Accessed June 14. <https://carto.com/docs/carto-engine/import-api/importing-geospatial-data#supported-geospatial-data-formats>.
- . 2017f. "Location Intelligence Software — CARTO." Accessed June 12. <https://carto.com/>.
- . 2017g. "Mobile SDK — API Reference — CARTO." Accessed June 12. <https://carto.com/docs/carto-engine/mobile-sdk/api-reference>.
- . 2017h. "Mobile SDK — CARTO." Accessed June 12. <https://carto.com/docs/carto-engine/mobile-sdk>.
- . 2017i. "Mobile SDK — Getting Started." Accessed June 17. <https://carto.com/docs/carto-engine/mobile-sdk/getting-started#prerequisites>.
- . 2017j. "Pricing — CARTO." Accessed June 12. <https://carto.com/>.
- "Chapter 7. Storing Binary Data." 2017. Accessed July 14. <https://jdbc.postgresql.org/documentation/83/binary-data.html>.
- Coast, Steve. 2004. "OpenStreetMap." Collaborative Mapping. *OpenStreetMap*. <https://www.openstreetmap.org/>.
- Cooper, Antony K., Serena Coetzee, Iwona Kaczmarek, Derrick G. Kourie, Adam Iwaniak, and Tomasz Kubik. 2011. "Challenges for Quality in Volunteered Geographical Information." In *Proceedings of the AfricaGEO 2011 Conference, Cape Town, South Africa*. Vol. 31.
- Cornez, Richard, and Trish Cornez. 2015. *Android Programming Concepts*. Jones & Bartlett Learning.
- Dahlgren, Randy, William Horwath, K. W. Tate, Trina Camping, and others. 2003. "Blue Oak Enhance Soil Quality in California Oak Woodlands." *California Agriculture* 57 (2): 42–47.
- Devictor, Vincent, Robert J. Whittaker, and Coralie Beltrame. 2010. "Beyond Scarcity: Citizen Science Programmes as Useful Tools for Conservation Biogeography." *Diversity and Distributions* 16 (3): 354–362.
- Elwood, Sarah, Michael F. Goodchild, and Daniel Z. Sui. 2012. "Researching Volunteered Geographic Information: Spatial Data, Geographic Research, and New Social Practice." *Annals of the Association of American Geographers* 102 (3): 571–590.

- Esmaili, Roya, Farzin Naseri, and Ali Esmaili. 2013. "Quality Assessment of Volunteered Geographic Information." *American Journal of Geographic Information System* 2 (2): 19–26.
- European Commission. 2002. "Environmental Noise Directive." Government. *Directive 2002/49/EC Assessment and Management of Environmental Noise*.
http://ec.europa.eu/environment/noise/directive_en.htm.
- Facebook. 2015. "Facebook: Terms of Service." Terms of Service. January 30.
<https://www.facebook.com/terms>.
- Feick, Rob, and Stéphane Roche. 2013. "Understanding the Value of VGI." In *Crowdsourcing Geographic Knowledge*, 15–29. Springer.
- Ferster, Colin J., and Nicholas C. Coops. 2013. "A Review of Earth Observation Using Mobile Personal Communication Devices." *Computers & Geosciences* 51: 339–349.
- Flanagin, Andrew J., and Miriam J. Metzger. 2000. "Perceptions of Internet Information Credibility." *Journalism & Mass Communication Quarterly* 77 (3): 515–540.
- . 2007. "The Role of Site Features, User Attributes, and Information Verification Behaviors on the Perceived Credibility of Web-Based Information." *New Media & Society* 9 (2): 319–342.
- . 2008. "The Credibility of Volunteered Geographic Information." *GeoJournal* 72 (3–4): 137–148.
- Flickr. 2017. "Flickr Services." Accessed June 15. <https://www.flickr.com/services/api/auth.oauth.html>.
- Fonte, C. C., Lucy Bastin, G. Foody, T. Kellenberger, N. Kerle, P. Mooney, A.-M. Olteanu-Raimond, and L. See. 2015. "VGI Quality Control." *ISPRS Geospatial Week 2015*, 317–324.
- Foody, Giles M., L. See, S. Fritz, M. Van der Velde, C. Perger, C. Schill, D. S. Boyd, and A. Comber. 2014. "Accurate Attribute Mapping from Volunteered Geographic Information: Issues of Volunteer Quantity and Quality." *The Cartographic Journal*, 1743277413Y–0000000070.
- Foody, Giles M., L. See, Steffen Fritz, Marijn Van der Velde, Christoph Perger, Christian Schill, and Doreen S. Boyd. 2013. "Assessing the Accuracy of Volunteered Geographic Information Arising from Multiple Contributors to an Internet Based Collaborative Project." *Transactions in GIS* 17 (6): 847–860.
- Fralish, James S. 2004. "The Keystone Role of Oak and Hickory in the Central Hardwood Forest."
- Gaman, Tom, and Jeffrey Firman. 2006. *Oaks 2040: The Status and Future of Oaks in California*. California Oak Foundation Oakland, CA, USA:
- Garbelotto, Matteo M., Pavel Svihra, and David M. Rizzo. 2001. "New Pests and Diseases: Sudden Oak Death Syndrome Fells 3 Oak Species." *California Agriculture* 55 (1).
<http://escholarship.org/uc/item/7pg5c8vc>.
- Gartner. 2016. "Global Mobile OS Market Share in Sales to End Users from 1st Quarter 2009 to 1st Quarter 2016." *Statista*. Accessed November 14.
<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.

- Goodchild, Michael F. 2007. "Citizens as Sensors: The World of Volunteered Geography." *GeoJournal* 69 (4): 211–221.
- . 2008. "Commentary: Whither VGI?" *GeoJournal* 72 (3): 239–244.
- Goodchild, Michael F., and Linna Li. 2012. "Assuring the Quality of Volunteered Geographic Information." *Spatial Statistics* 1: 110–120.
- Google. 2017a. "Google Drive - Cloud Storage & File Backup." Accessed June 15. www.google.com/drive/.
- . 2017b. "Pricing Guide." Accessed July 14. www.google.com/drive/pricing/.
- Google Developers. 2017a. "Android Quickstart | Drive REST API." *Google Developers*. Accessed June 15. <https://developers.google.com/drive/v3/web/quickstart/android>.
- . 2017b. "Com.google.android.gms.location | Google APIs for Android." *Google Developers*. Accessed June 21. <https://developers.google.com/android/reference/com/google/android/gms/location/package-summary>.
- Haklay, Mordechai. 2010. "How Good Is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets." *Environment and Planning B: Planning and Design* 37 (4): 682–703.
- Haklay, Muki. 2013. "Citizen Science and Volunteered Geographic Information: Overview and Typology of Participation." In *Crowdsourcing Geographic Knowledge*, 105–122. Springer.
- Han, Kyungsik, Eric A. Graham, Dylan Vassallo, and Deborah Estrin. 2011. "Enhancing Motivation in a Mobile Participatory Sensing Project through Gaming." In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, 1443–1448. IEEE.
- Horita, Flávio Eduardo Aoki, Livia Castro Degrossi, Luiz Fernando Gomes de Assis, Alexander Zipf, and João Porto de Albuquerque. 2013. "The Use of Volunteered Geographic Information (VGI) and Crowdsourcing in Disaster Management: A Systematic Literature Review."
- Huang, Cheng-Min, Edward Chan, and Adnan A. Hyder. 2010. "Web 2.0 and Internet Social Networking: A New Tool for Disaster Management?-Lessons from Taiwan." *BMC Medical Informatics and Decision Making* 10 (1): 57.
- "iNaturalist.org." 2017. *iNaturalist.org*. Accessed July 2. <http://www.inaturalist.org/>.
- Instagram. 2017. "Instagram Developer Documentation." Accessed June 15. <https://www.instagram.com/developer/authentication/>.
- Irwin, Alan. 1995. *Citizen Science: A Study of People, Expertise and Sustainable Development*. Psychology Press.
- Johnson, Peter A., and Renee E. Sieber. 2013. "Situating the Adoption of VGI by Government." In *Crowdsourcing Geographic Knowledge*, 65–81. Springer.

- Kelly, Nina Maggi. 2002. "Monitoring Sudden Oak Death in California Using High-Resolution Imagery." *USDA-Forest Service General Technical Report PSW-GTR-184*, 799–810.
- Knoot, Tricia G., Lisa A. Schulte, and Mark Rickenbach. 2010. "Oak Conservation and Restoration on Private Forestlands: Negotiating a Social-Ecological Landscape." *Environmental Management* 45 (1): 155–164.
- Kurniawan, Budi. 2015. *Java for Android, Second Edition*. 2nd ed. Brainy Software.
- Lathrop, Richard G., Paul Montesano, Jason Tesauro, and Brian Zarate. 2005. "Statewide Mapping and Assessment of Vernal Pools: A New Jersey Case Study." *Journal of Environmental Management* 76 (3): 230–238.
- Maisonneuve, Nicolas, Matthias Stevens, Maria E. Niessen, and Luc Steels. 2009. "NoiseTube: Measuring and Mapping Noise Pollution with Mobile Phones." In *Information Technologies in Environmental Engineering*, 215–228. Springer.
- Martí, Irene Garcia, Luis E. Rodríguez, Mauricia Benedito, Sergi Trilles, Arturo Beltrán, Laura Díaz, and Joaquín Huerta. 2012. "Mobile Application for Noise Pollution Monitoring through Gamification Techniques." In *International Conference on Entertainment Computing*, 562–571. Springer.
- Mayer, Amy. 2010. "Phenology and Citizen Science Volunteers Have Documented Seasonal Events for More than a Century, and Scientific Studies Are Benefiting from the Data." *BioScience* 60 (3): 172–175.
- Metzger, Miriam J. 2007. "Making Sense of Credibility on the Web: Models for Evaluating Online Information and Recommendations for Future Research." *Journal of the American Society for Information Science and Technology* 58 (13): 2078–2091.
- Miller, Christopher C. 2006. "A Beast in the Field: The Google Maps Mashup as GIS/2." *Cartographica: The International Journal for Geographic Information and Geovisualization* 41 (3): 187–199.
- National Audubon Society. 2016. "Christmas Bird Count." *National Audubon Society*. Accessed October 24. <http://www.audubon.org/conservation/science/christmas-bird-count>.
- NJ Division of Fish and Wildlife. 2010. "New Jersey Division of Fish & Wildlife: Vernal Pool Project Update." Government. *NJ Division of Fish & Wildlife*. September 3. <http://www.state.nj.us/dep/fgw/ensp/vrnpoolupdate.htm>.
- O'Geen, A., R. Dahlgren, Alexandre Swarowsky, D. Lewis, M. Singer, and others. 2010. "Research Connects Soil Hydrology and Stream Water Chemistry in California Oak Woodlands." *California Agriculture* 64 (2): 78–84.
- Petsas, Thanasis, Antonis Papadogiannakis, Michalis Polychronakis, Evangelos P. Markatos, and Thomas Karagiannis. 2013. "Rise of the Planet of the Apps: A Systematic Study of the Mobile App Ecosystem." In *Proceedings of the 2013 Conference on Internet Measurement Conference*, 277–290. ACM.
- "PostGIS — Spatial and Geographic Objects for PostgreSQL." 2017. Accessed July 14. <http://postgis.net/>.
- Rieh, Soo Young, and David R. Danielson. 2007. "Credibility: A Multidisciplinary Framework." *Annual Review of Information Science and Technology* 41 (1): 307–364.

- Rizzo, David M., and Matteo Garbelotto. 2003. "Sudden Oak Death: Endangering California and Oregon Forest Ecosystems." *Frontiers in Ecology and the Environment* 1 (4): 197–204.
- Roche, Stephane, Eliane Propeck-Zimmermann, and Boris Mericskay. 2013. "GeoWeb and Crisis Management: Issues and Perspectives of Volunteered Geographic Information." *GeoJournal* 78 (1): 21–40.
- Sester, Monika, Jamal Jokar Arsanjani, Ralf Klammer, Dirk Burghardt, and Jan-Henrik Haurert. 2014. "Integrating and Generalising Volunteered Geographic Information." In *Abstracting Geographic Information in a Data Rich World*, 119–155. Springer.
- Song, Weidong, and Guibo Sun. 2010. "The Role of Mobile Volunteered Geographic Information in Urban Management." In *2010 18th International Conference on Geoinformatics*, 1–5. IEEE.
- Standiford, Richard B., W. J. McShea, W. M. Healy, and others. 2002. "California's Oak Woodlands." *Oak Forest Ecosystems: Ecology and Management for Wildlife*, 280–303.
- Stoll, Greg. 2005. "Hurricane Information Maps." Hurricane Crisis Management. *Hurricane Information Maps*. December 16. <https://gregstoll.dyndns.org/scipionus/>.
- Sulak, Adriana, Lynn Huntsinger, Sheila Barry, and Larry Forero. 2008. "Public Land Grazing for Private Land Conservation?" In *2008. Proceedings of the Sixth California Oak Symposium: Today's Challenges, Tomorrow's Opportunities. Gen. Tech. Rep. PSW-GTR-217. Albany, CA: US Department of Agriculture, Forest Service, Pacific Southwest Research Station*, 7–18. http://www.carangeland.org/images/Public_Land_Grazing_for_Private_Land_Conservation.pdf.
- Sullivan, Brian L., Christopher L. Wood, Marshall J. Iliff, Rick E. Bonney, Daniel Fink, and Steve Kelling. 2009. "eBird: A Citizen-Based Bird Observation Network in the Biological Sciences." *Biological Conservation* 142 (10): 2282–2292.
- Tulloch, David L. 2007. "Many, Many Maps: Empowerment and Online Participatory Mapping." *First Monday* 12 (2).
- . 2008. "Is VGI Participation? From Vernal Pools to Video Games." *GeoJournal* 72 (3–4): 161–171.
- U.S. Board on Geographic Names. 2016. "U.S. Board on Geographic Names." Government. August 2. <http://geonames.usgs.gov/>.
- Vandecasteele, A., and R. Devillers. 2013. "Improving Volunteered Geographic Data Quality Using Semantic Similarity Measurements." *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (1): 143–148.
- Weigel, Dale R., and J. C. Randolph. 2013. "Spectral Reflectance of Five Hardwood Tree Species in Southern Indiana."
- Zook, Matthew, Mark Graham, Taylor Shelton, and Sean Gorman. 2010. "Volunteered Geographic Information and Crowdsourcing Disaster Relief: A Case Study of the Haitian Earthquake." *World Medical & Health Policy* 2 (2): 7–33.