GENERATING BICYCLIST COUNTS USING VOLUNTEERED AND PROFESSIONAL GEOGRAPHIC INFORMATION THROUGH A MOBILE APPLICATION

by

Patricia Jula

A Thesis Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(GEOGRAPHIC INFORMATION SCIENCE AND TECHNOLOGY)

May 2015

DEDICATION

This thesis is dedicated to all the bicyclists harmed in the motorist-bicyclist collision data utilized in this project.

ACKNOWLEDGMENTS

I would like to thank my family including my mom, dad, and sister for their support throughout my graduate coursework and thesis writing. I really appreciate how encouraging you have been while I have been achieving this education.

Several people at the University of Southern California were essential for completing this application development project. I want to extend a big thanks to my thesis committee, composed of Dr. Yao-Yi Chiang, Dr. Robert Vos, and Dr. Darren Ruddell, who offered valued direction while working on this thesis. Dr. Chiang provided excellent and thorough review of my work and convinced me to complete more with this project than I would have thought I could accomplish. In the initial stages of this thesis project, Dr. Vos was very valuable in getting me to focus on creating this mashup development project. I'd also like to thank Dr. Vanessa Osborne for numerous readings of my thesis and for offering essential feedback.

I am very thankful to the co-workers and friends who made the time to test and provide valuable feedback on the application and website.

Lastly I would like to thank my husband, Dr. William Kronholm, who was very support of my educational goal, and provided highly appreciated debugging suggestions for my code while I worked on this thesis project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AADB	Average Annual Daily Bicyclists
AADT	Average Annual Daily Traffic
API	Application Programming Interface
BC	Bicyclists Count Application and Bicyclists Count Website
BCApp	Bicyclists Count Mobile Application
BCWeb	Bicyclists Count Website
GIS	Geographic Information Systems
IDE	Integrated Development Environment
LACBC	Los Angeles Count Bicyclist Coalition
LAPD	Los Angeles Police Department
LBS	Location Based Services
NHTSA	National Highway Traffic Safety Administration
OGC	Open Geospatial Consortium
OSM	Open Street Map
PGI	Professional Geographic Information
SCAG	Southern California Associate of Governments
SDK	Software Development Kit
SPFs	Safety Performance Functions
VGI	Volunteered Geographic Information

ABSTRACT

Counts of the number of bicyclists on roads give community-based organizations strength in appealing for improved bike infrastructure from city governments. Bicyclist count data can also be used in conjunction with vehicle counts and collision data to better understand factors that contribute to motorist-bicyclist collisions. Bicyclist count collection involves manual methods where volunteers fill out paper forms for bike coalitions, and automated methods such as video cameras set up on roads to capture bicyclist movement. This thesis presents a mobile application through which users generate bicyclist counts (Volunteered Geographic Information, VGI), and a website that provides a method for users to review these bicyclist counts. Both the application and website developed in this thesis contain motorist-bicyclist collision data derived from an authoritative source (Professional Geographic Information, PGI). Counting bicyclists in high collision areas can indicate of these areas see a high or low amount of bicyclists, making the motorist-bicyclist collision PGI germane. The bicyclist count collection method produced by this thesis serves as a model for community-based organizations that want to collect bicyclist counts by means of an inexpensive and automated method.

CHAPTER 1: INTRODUCTION

Generating bicyclist counts gives city governments an understanding of where to focus bike infrastructure planning that can improve bicyclist safety. Bicyclist counts refer to the process of tallying the number of bicyclists present on a road over a defined period of time in this thesis. Bike infrastructures that can be created based on these counts include a bike lane designated by a white stripe on the road, a bicycle symbol on the road, signage, or contrasting pavement (USDOT 2014). These lanes encourage predictable movement between bicyclists and motorists (USDOT 2014), a valuable component of city streets. Bicyclist count data can also be used in conjunction with vehicle counts and motorist-bicyclist collision data to better understand factors that contribute to these collisions, such as road design and vehicle speed (Dumbaugh and Li 2009).

Two overarching techniques of collecting bicyclist counts exist: manual and automated. These two techniques represent, respectively, a bicyclist count during which a person must be present to conduct the count and a bicyclist count in which a person does not need to be present to conduct the count. The National Bicyclist and Pedestrian Documentation (NBPD) Project (2014) define a manual bike count method called screenline counts, where a volunteer establishes an invisible line across a road and sidewalk, counting bicyclists that cross over this line during a set time period. A person conducts screenline counts from her/his position on the sidewalk. Alternatively, Automated count methods make use of video cameras or pneumatic tubes to count bicyclists (Zelt 2014; Malinovskiy, Zheng, and Wang 2009).

Bike coalitions comprise a group that organize bicyclist counts because these groups advocate for safer biking environments. These organizations can more effectively appeal for better bike infrastructure from city governments with bicyclist count data (LACBC 2011). The

Los Angeles County Bike Coalition (LACBC) collects bike counts, along with pedestrian counts, with the goal of advocating for bicyclists in transportation planning in Los Angeles County cities including Los Angeles, Glendale, and Culver City (LACBC 2011). In October 2014 the LACBC provided volunteers with standardized paper forms to collect bicyclist counts on LACBC selected streets (LACBC 2014c). The data captured during the 2014 LACBC count must be manually entered into the LACBC count database (LACBC 2014c). This data entry is potentially time-consuming and prone to errors. LACBC could save time by collecting counts through a mobile GIS application that automatically stores bike counts to a database.

This thesis presents a mobile GIS application, called Bicyclists Count, which provides an efficient method for collecting bicyclist counts through a mobile device and storing the results automatically on a cloud server. The collected bicyclist counts serve as the Volunteered Geographic Information (VGI) of this project. Motorist-bicyclist collision data for Los Angeles is also included in this project. The motorist-bicyclist collision data was professionally created, comprising the Professional Geographic Information (PGI) component of this thesis. In addition to creating BCApp, this thesis also involves creating the Bicyclists Count website (BCWeb, http://bicyclistscount.net) which gives its visitors a platform independent method of reviewing the BC data. Throughout this thesis references to both the BCApp and BCWeb will be referred to as BC. The data flow diagram in Figure 1 depicts the VGI, PGI, and BC components.

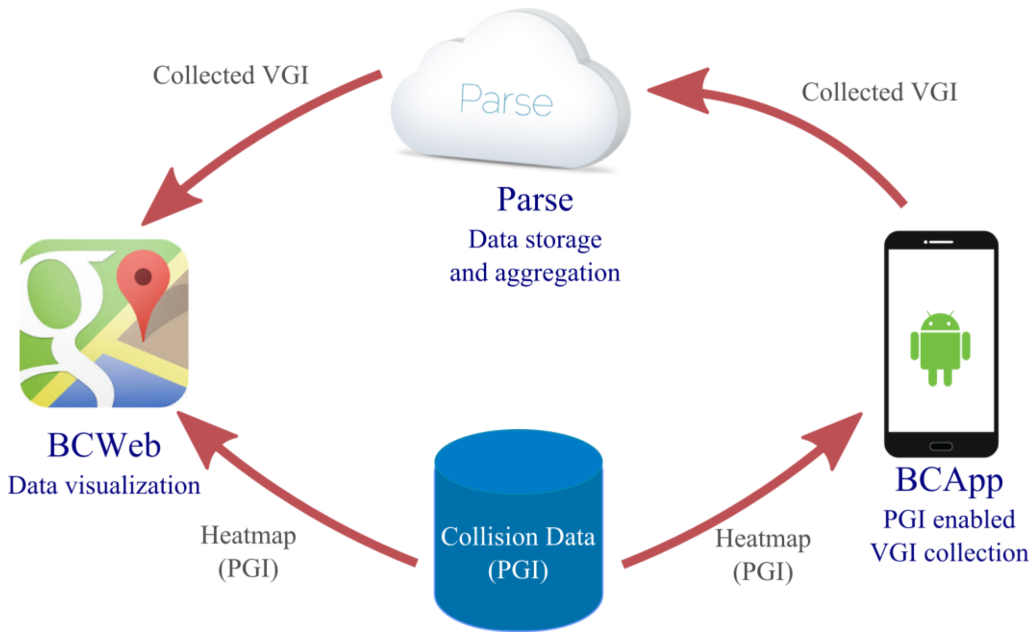


Figure 1 BC Data Flow

1.1 VGI and PGI Mashup

BCApp is a mashup application, meaning it contains a mixture of Volunteered Geographic Information (VGI) and Professional Geographic Information (PGI) sources. A motorist-bicyclist collision in the city of Los Angeles represents the Professional Geographic Information (PGI) portion of BCApp. This collision data is provided through a heatmap, which refers to a visualization that uses color to display the distribution and relative intensity of data, rather than displaying data points for each event (Google Developers 2014). The visual nature of the heatmap renders the collision data, which contains more than 3,000 records, easy to examine. The motorist-bicyclist collision heatmap (PGI) was created with motorist-bicyclist collision data provided by the Los Angeles Police Department (LAPD) for the years 2011-2012 (Carter 2013). More broadly, PGI can be defined as geographic information that is traditionally prepared and published by members of public-sector agencies using mapping software (Goodchild 2008).

The counts collected with BCApp represent the VGI portion of BC. VGI refers to volunteers collecting geographic data that can be used by others (Celino 2013). VGI has been on the rise with projects such as Wikimapia and OpenStreetMap (Celino 2013). BCApp contains a form of VGI where people generate data of a specific kind rather than describing place names or streets (Goodchild 2008). Although accuracy is a concern with VGI, a study has found that greater amounts of VGI collected at a location improve accuracy for this type of data (Zielstra and Zipf 2010). Zielstra and Zipf (2010) analyzed OpenStreetMap (OSM) and TeleAtlas, a proprietary data vendor, and found that OSM can replace proprietary data for larger cities in Germany such as Berlin and Munich. Since BCApp is available to download for free on the Google Play Store, this application could generate enough counts to make bike movement knowledge in some locations accurate.

1.2 Motivation

The motivation for creating BC is based on a desire to generate bicyclist counts, data that can direct city planners to roads that need better bike infrastructure. These improvements can make city streets in Los Angeles safer for bicyclists. Collecting bicyclist counts on streets with high motorist-bicyclist collisions, using the BCApp PGI as reference, can help to determine if these locations contain a high or low volume of bicyclists. Potentially a street with high collisions and high bicyclist counts will be prioritized by city planners for improved bike infrastructure. Research on how to effectively aggregate data that can improve bicyclist safety occurs in other cities in the United States. For example, in Boston, MA a team of students at Harvard proposed a method for streamlining Boston's motorist-bicyclist collision data, bicyclist counts, and related data from numerous city agencies into one database, giving policy makers a system to evaluate this data more effectively (Roeder 2013). If enough of the bicyclist count VGI is created with BCApp, it can potentially provide a better understanding of how city planners of Los Angeles can prioritize streets to, provide better bike infrastructure.

The benefits of bicycling and the prevalence of this form of transportation underscore the importance of BCApp. Bicycling is a useful form of transportation that offers many health benefits. In urban environments, bicycling is appealing since bicyclists can reach destinations in a practical amount of time (Cahill Delmelle, Thill, and Ha 2012). Unfortunately, motorist-cyclists accidents result in preventable deaths each year. According to the National Highway Traffic Safety Administration (NHTSA) in 2012 there were 726 bicyclists killed in motorist-cyclists accidents within the United States, compared with 682 in 2011 (NHTSA 2014). These deaths as well as injuries highlight the value of creating a tool to produce bicyclist counts.

1.3 Project Overview

BCApp provides a process to collect bicyclist counts through mobile devices that use the Android operating system. After downloading BCApp, the user reviews the PGI heatmap to find locations of high collisions. Counts near high collision locations generate data that can be valuable to city planners for bike infrastructure research. After the user arrives at the selected locations she/he draws an imaginary line across the road and proceeds to count bicyclists that cross that line on the road or sidewalk during a 15-minute interval. Using screenline counts to capture bicyclist activity are consistent with methods used to capture vehicle counts (NPBD Project 2014). Screenline counts are also used in pen and paper counts by LACBC. Bicyclists that cross the screenline multiple times during the 15-minute period are counted each time (NPBD Project 2014).

BCApp is designed to allow users to easily keep track of bicyclists and to require minimal user input. For example, BCApp automatically stores user location as latitude and longitude values. This location-awareness was built with Google Play Location Services, a tool that captures the geographic information of a mobile device (OGC 2014). This location awareness prevents BCApp users from needing to capturing latitude/longitude within their BCApp count form, as described in Chapter 3. Once users have counted bicyclists for 15 minutes, they click a button within BCApp that stores their count number. The user's post (i.e., the bicyclist count) automatically stores to the back-end database. BCApp bicyclist counts are added to BCWeb, as described in Chapter 3.

The city of Los Angeles serves as the study area for the BC application and website. Los Angeles represents a city seeking to encourage bicycling and reduce dependence on vehicles, particularly in the past several decades with city and countywide programs such as the Bicycle Plan in 2011, the Metro Bicycle Transportation Regional Plan in 2006, and the Los Angeles County Bicycle Plan in 1976 (LADCP 2011). These efforts to improve the bicycling environment make BC particularly relevant for the city of Los Angeles. In addition, the population size of Los Angeles encourages the generation of data to improve bicyclist safety. Los Angeles was the second largest city in the United States by population in 2010 (American FactFinder 2014). Another factor that makes BC appealing for Los Angeles is that this city contains a burgeoning bicyclist culture. The LACBC and Midnight Ridazz represent two groups promoting bicycling in Los Angeles (Midnight Ridazz 2004; LACBC 2014a). Figure 2 contains an outline of Los Angeles city boundaries.

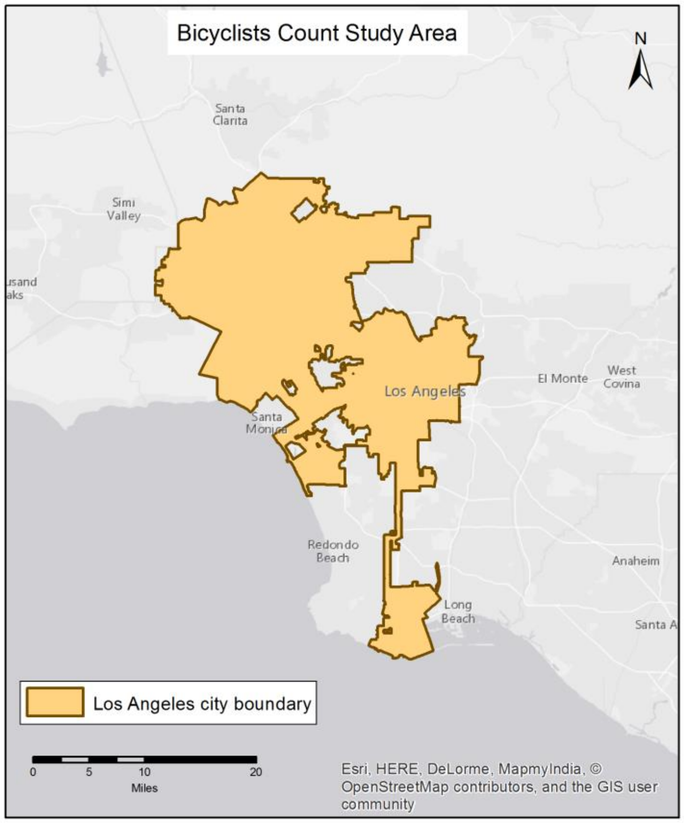


Figure 2 Study Area

1.4 Application Goals

Table 1 contains the BC project goals and the technical descriptions for achieving these goals.

Table 1 BC Project Requirements

Application Requirement	**Technical Description**
Provide an alternative to paper form bicyclist count collection that automates data storage of bicyclist counts	Build a mobile GIS application that automatically stores the bicyclist count data
Develop an application using a platform that is affordable to build with and maintain	Build with the Android software stack
Automate storage of bicyclist counts within website to serve as reference data for city planners and dwellers	Develop an application that contains a back-end database that can be processed with a Python script
Implement features in BCApp that reduce data input by users	Capture user location automatically with the Android LocationListener (Android 2014b)
Provide motorist-bicyclist collision reference data so that users of the application know where bicyclist counts will be most useful	Geocode collision data provided by the LAPD to add to a heatmap of motorist-bicyclist collisions
Create a way to display the bicyclist counts and motorist-bicyclist collision data that is platform independent	Develop a web page of the collision and bicyclist count data
Provide users a way of storing bicyclist count value within BCApp	Reduce user input of bicyclist count values with Android (Android 2014b)
Ensure that Bicyclists Count works well and is intuitive for users	Test and debug application as based on user feedback
Ensure that Bicyclists Count works on a variety of devices	Create an application that works on Google Nexus 7 tablet, Samsung S5, and Samsung S3

1.5 Application Technology

BCApp was created with the Google Android software development kit (SDK). The Android software stack consists of an open source software system, middleware, and user applications (Gandhewar and Seikh 2010). Developing an application with open source components was part of the decision to create an Android application. This project developed BCApp through the Eclipse Integrated Development Environment (IDE, The Eclipse Foundation 2014), using the Android SDK and Java Development Kit (Gandhewar and Seikh 2010). Table 2 displays the components of BCApp.

Table 2 Software components of BCApp

Application Framework	Version	API Level	Java version	IDE
Android	Android 4.4 KitKat	20	1.7.0_51	Eclipse

Aside from a preference for building an open-source application there are other advantages to developing with Android. Compared with other operating systems such as Windows or Apple iOS, the popularity of devices sold with Android has been increasing. Android is estimated to capture approximately 45 percent of device shipments by operating systems in 2014 compared with 38 percent in 2013 (Gartner 2014). While the market for tablets covers a small amount of device shipment, the percentage is increasing. Tablets took approximately nine percent of all device shipments in 2013, compared with approximately 11 percent projected for 2014 (Gartner 2014). The percentage of tablet shipments is projected to continue growing in 2015 (Gartner 2014). The increasing popularity of Android provides motivation for developing BCApp for this operating system.

BCApp utilizes technology that is affordable, allowing this project to achieve the goal of being inexpensive to develop and maintain. Bicyclist counts generated with BCApp are automatically stored to a cloud-based online platform called Parse.com (Parse 2014). The platform Parse.com offers a freemium model of cloud storage (Parse 2014). Freemium refers to a business model that offers basic features for free and additional features for a cost (Kumar 2014). Developing BCApp with Parse.com, or Parse, is also appealing because this software platform allows developers to store their application data without configuring server side code (Taylor 2011).

The front-end, meaning the software through which users can interact with back-end data, is also reasonably priced. The Google Map Android and JavaScript APIs comprise the map features of BC. These APIs are free of charge unless a map receives more than 25,000 hits per day for more than 90 consecutive days (Google Developers 2014b). Using Parse and Google Maps creates an affordable development option compared to other development platforms. For example a server with ArcGIS Server installed is necessary to develop apps with the ArcGIS Runtime Software Development Kit (SDK) for Android (Esri 2013). The server configuration and cost are more appealing with Parse and Google Maps than other options such as ArcGIS.

1.6 Thesis Organization

This thesis is divided into five chapters. Chapter 2 summarizes research undertaken to understand the relationship between collisions and the number of bicyclists and vehicles in a U.S. city. Chapter 2 goes on to compare and contrast BCApp with other counting methods to highlight the strengths of this application. Chapter 3 presents the general approach taken in the application development. Chapter 4 discusses evaluations provided by testers of the BC application and website. Chapter 5 concludes with a discussion of the impact and future improvements to the BC application and website.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

Chapter 2 describes bicyclist safety research and different types of bicyclist count tools. Section 2.1 describes a research project on developing bicyclist safety performance functions (SPFs) for Boulder, Colorado (Nordback, Marshall, and Janson 2014). The Boulder study is relevant for this thesis project because it highlights the importance of bicyclist volume data in creating bicyclist safety research. Next, Section 2.2 reviews automated and manual bicyclist counts methods and compares them with BCApp. After that, Section 2.3 discusses the incentive for developing with the Android software stack and the Parse back-end platform to BCApp. Section 2.3 concludes Chapter 2 with providing an overview of other mobile GIS applications and compares and contrasts them with BCApp.

2.1 Motorist-Bicyclist Collision Research

Developing a tool that can efficiently produce large numbers of bicyclist counts for bicyclist accident prediction models is a major motivation for developing BCApp with this thesis project. Bicyclist counts are essential for creating bicyclist Safety Performance Functions (SPFs). SPFs are defined as accident prediction models that analyze traffic volume in conjunction with collision data to determine accident rates over a time period (Kononov and Allery 2012). Nordback, Marshall, and Janson (2014) estimate average annual bicyclist (AADB) and average annual traffic (AADT) from automated vehicle and bicyclist counters using estimation techniques similar to those employed by the Federal Highway Administration. This study does not define what constitutes sufficient data for estimating AADB and AADT.

The bicyclist SPFs produced by the Boulder study reveals several findings about vehicle collisions with bicyclists when considering the number of bicyclists on the road. The Boulder study found that to an extent there were fewer collisions per bicyclist with a greater volume of bicyclists, which can be considered a “safety in numbers” effect (Nordback, Marshall, and Janson 2014). The authors interpret this effect in several ways. First, Nordback, Marshall, and Janson (2014) state that more bicyclists present on a road might trigger safer driving by motorists and safer bicycling by bicyclists. Alternatively, the authors suggest bicyclists will frequent roads safer for biking. This study highlights how modeling the relationship between motorist-bicyclist collisions and bicyclist exposure to collisions gives cities data that can produce improved safety measures.

2.2 Bicyclist Counting Methods

BCApp allows users to generate bicyclist counts in a manner that automates as much of the collection as possible and is affordable to use. This section reviews different types of counting methods and compares them with features of the BCApp collection method. Bike counts occur in cities such as Los Angeles, CA, New York, NY, Portland, OR, and San Francisco, CA (LACBC 2014b; PBOT 2014; New York City DOT 2014; SFMTA 2013). The occurrence of bicyclist counts in cities across the United States underscores the value of developing BCApp. Although BCApp, is developed for Los Angeles, potentially this application can be utilized in other cities.

2.2.1 Automated Inductive Loop

The Boulder study described above employs an automated inductive loop counter to count bicyclists (Nordback, Marshall, and Janson 2014). Inductive loop counters are traffic sensors installed by cutting a shallow slot in pavement and placing an insulated loop wire within that slot (FHWA 2006). When a motorist or bicyclist passes over the wire, the electrical circuit flowing in the wire induces a voltage. This induction is registered by the controller cabinet of the wire as the passage of a bike or vehicle (FHWA 2006). From twelve automated inductive loop bicyclist counters set up throughout Boulder, the authors created estimates on annual bicyclist traffic.

While the automated inductive loop counter can count bicyclists for longer periods of time than users of BCApp, there are drawbacks to this method of counting bicyclists. SCAG and METRO (2013) state that automated inductive loop counters require a nearby source of electric power to function. BCApp does not require a power source.

The Boulder study does not state the expense of the Boulder inductive loop counters, although estimates can be captured for other locations. In Lincoln, Nebraska an estimate found six automated inductive loop counters would cost the city approximately $16,000 to $32,000, depending on the counter options and manufacturer (Pesnichak 2013). BCApp is free for users to download and utilize, provided they own an Android device.

Aside from being costly, an inductive loop counter offers a limited counting ability. An inductive loop counter installed in a bike lane does not capture bicyclists that travel outside of where the counter is installed. Users of BCApp count bicyclists along the entire span of a road and sidewalk through their screenline counts. Another difference between automated inductive loop counters and BCApp is how the methods are developed. One of the manufacturers of automated inductive loop counters has patented the technology used with their counter (Zelt 2014). In contrast, BCApp is not developed with patented technology, meaning it is easy for developers to download and modify the Android source code for BCApp.

2.2.2 Pneumatic Tubes

Another automated bicyclist count method involves laying a pneumatic tube across a road that contains mechanical counters at both ends; bicyclists are counted as they cross over the tube. Authors Hyde-Wright, Graham, and Nordback (2014) studied the use of pneumatic tubes to conduct short-term counts in Boulder, Colorado. This study found that pneumatic tubes failed to count bicyclists if they cycled over the tube too far from where the mechanical counters which are locate at the ends of a tube. Another drawback to pneumatic tube counters is that bicyclists riding side by side might be counted as one. BCApp users will probably not produce these counting errors.

In addition to the counting errors possible with pneumatic tubes, the cost of equipment for this counting method needs to be considered by cities government looking to implement bicyclist counts. These expenses include the initial purchase, plus labor costs and the occasional replacement of tubes (Proulx 2013). In December 2013, a report prepared for Davis, California found the initial purchase price of one pneumatic tube to be $2,500 (Proulx 2013). Assuming the cost of one pneumatic tube continues to cost $2,500, installing two of these tubes at ten intersections in Los Angeles would cost:

$50,000 = 10 X ($2,500 X 2)

Adding in the cost of labor for installation and replacement tubes and the expense grows. In comparison BCApp offers a count option that is free to users.

2.2.3 Video Camera

In addition to the automated collection methods that inductive loop and pneumatic tubes offer, video cameras can also be set up to automate capture of bicyclist presence on streets. Malinovskiy, Zheng, and Wang (2009) developed a method to count bicyclists and pedestrians with low-resolution video cameras. An algorithm developed with Visual C# (Microsoft 2014) extracted objects from the video. The algorithm developed for this study requires the pedestrian/bicyclist traffic is not too high ensuring that entities can be individually recorded for a few frames. One advantage of the video capture method is that it is easy to install using existing street infrastructure. For example, the video cameras can be installed on existing traffic poles. Similar to video capture, BCApp does not require any changes to street infrastructure.

Malinovskiy, Zheng, and Wang (2009) stated video camera based counts are inexpensive, although using this technology instead of an economical tool such as BCApp means the costs can accrue. For example, there can be are labor costs involved in re-watching video footage as well as video camera acquisition and maintenance costs (FHWA 2014). Bicyclist Count differs from video capture, pneumatic tube, and automated inductive loop counters in that it is available for free to users owning Android devices.

2.2.4 Paper and Pen

The BCApp collection method most resembles the paper and pen method of capturing bicyclist counts. The “paper and pen method” is also used by the Los Angeles County Bike Coalition (LACBC) and requires users to fill out a standardized form. Both the paper and pen and BCApp methods require a person to be physically present on a sidewalk to capture bicyclist counts. The paper and pen method involves users to count with paper, a writing surface such as a clipboard, and a pen or pencil in their possession. BCApp users only need a mobile device with BCApp installed to do a bicyclist count. Once paper and pen counts are completed the forms are sent to LACBC where members input the counts into a database. This input method requires manual data entry by LACBC that can be time consuming and prone to data entry error. Alternatively, after a BCApp user clicks the button to save a “post,” it sends the recorded bicyclist count to a back-end database. This automated storage of bicyclist counts results in BCApp producing a time savings over the data entry required with the paper and pen method. The BCApp count automation feature achieves the project goal of reducing time spent carrying out data entry of bicyclist count data.

2.3 Mobile GIS Application Development Research

Components of other mobile GIS applications influenced the development of Bicyclist Count. The Mobile Data Collection app, developed by GIS Cloud (2014), contains similar features found in BCApp. This application includes web forms that users can edit while out in the field, and web maps for users to explore. Mobile Data Collection requires users to have a GIS Cloud account. While users need to register their names and email addresses to use Bicyclist Count, they do not need to create an account with the Parse.com back-end to use BCApp.

I considered another popular mobile GIS platform for developing BCApp. Environmental Systems Research Institute (Esri) released the newest version of its Android mobile GIS application, called Collector for ArcGIS 10.2.5, in July 2014 (Esri 2014c). This application is not free to use. Users must have an ArcGIS organizational account to use this Collector app, and it requires an annual subscription plan (Esri 2014c). BCApp differs from Collector for ArcGIS because it is available for free. Moreover, BCApp is specifically geared toward generating bike count VGI, and contains a motorist-bicyclist collision heatmap that gives users the direction where counts are most beneficial. One interesting feature of ArcGIS Collector is that data can be collected when users are offline. This feature is currently not available with BCApp, although it can be considered for inclusion in a future release of the application.

2.3.1 Forest Resources Application Example

This project explored the value of mobile applications for field collection as compared to more traditional methods of collecting field data. A study conducted Kennedy et al. (2014) on the practicality of a smart-phone application to produce field collection of forest resources in

Ontario, Canada is pertinent for this research. This study developed an application with Microsoft Visual Studio 2005 and Esri ArcPad and designed this application to mimic a paper data collection form. The mobile application captures features including counts of tree species, condition, and characteristics of the site where data collection occurred.

In the Ontario forest resource study, there were benefits and disadvantages with using a mobile application to complete data collection when compared to a paper form. Data entry took longer with the smartphone than with the paper method, and there was a greater chance of entry error from users clicking the wrong buttons on the screen (Kennedy et al. 2014). In spite of these drawbacks, users collecting forestry data preferred carrying around a smartphone to using paper collection forms, a map, a camera, and a GPS unit.

The preference for data collection with a mobile device is also realized with BCApp. As described in Chapter 4, BCApp evaluators preferred collecting bicyclist counts through a mobile device over collecting with pen and paper. This preference for mobile device is probably because mobile devices improve the efficiency of gathering data and work well in many environments (Feixiang, Xiao, and Shaoliang 2013). In addition to improved efficiency, Kennedy et al. (2014) found the smartphone application studied in Ontario reduced post processing times because the data collected in forest survey did not need to be manually entered into the database. A time savings in data entry is also an advantage realized with BCApp since counts are automatically stored to the backend Parse.com database.

CHAPTER 3: DEVELOPMENT

Chapter 3 describes BC development, and ties in how the components accomplish project goals. First, section 3.1 explains the process of geocoding the motorist-bicyclist collision data. Second, section 3.2 reviews how this data is utilized in the motorist-bicyclist collisions heatmap. After that, section 3.3 describes how these features combine to produce an effective method for counting bicyclists through BCApp. Finally, section 3.4 summarizes the BCWeb creation and explains how this website augments the mobile application by providing a platform independent method for visitors to access BCApp content.

3.1 Collision Data Geocoding

The first step in creating BC is to geocode the collision data. Geocoding is the process of connecting a street address or intersection to latitude and longitude coordinates (Murray et al. 2011). The LAPD provided a table of collisions in Los Angeles (Carter 2013). This table contains attributes related to different types of collisions, including the collision data and time, collision location, and type of collision. The collision location is provided in one of two formats: either a street intersection or an address. The collision type is coded as one of the following: vehicle collision with a bicyclist, vehicle collision with a pedestrian, bicyclist collision with an open car door, or bicyclist collision with a vehicle. Figure 3 contains a capture of this table. Geoprocessing with several software tools is necessary before these data can be added to a heatmap that provides BCApp users with a visual reference for motorist-bicyclist collision density in Los Angeles.

OBJECTID	DATE	TIME	Method of collision	Direction	Street	Street suffix	Cross street	Cross street suffix	LOCATION
284	1/1/2012	15:45:00	3008		FICKETT		MALABAR		FICKETT & MALABAR
529	1/2/2012	19:00:00	3008		3RD	ST	VISTA	ST	3RD ST & VISTA ST
825	1/2/2012	17:15:00	3008		VANOWEN	ST	WILBUR	AV	VANOWEN ST & WILBUR AV
1432	1/2/2012	19:00:00	3008		COLDWATER	AV	ROSCOE	BL	COLDWATER CA AV & ROSCOE B
1433	1/2/2012	21:30:00	3008		LANKERSHIM		HUSTON		LANKERSHIM & HUSTON
2642	1/2/2012	15:00:00	3008		WILTON	PL	SUNSET	BL	WILTON PL & SUNSET BL
1566	1/3/2012	17:30:00	3008		FOOTHILL		ORO VISTA		FOOTHILL & ORO VISTA
1617	1/3/2012	16:20:00	3008		LASSEN	ST	INDEPENDENCE	AV	LASSEN ST & INDEPENDENCE AV
1618	1/3/2012	17:40:00	3008		TAMPA	AV	ROSCOE	BL	TAMPA AV & ROSCOE BL
1716	1/3/2012	17:55:00	3008		VAN NUYS	BL	ROSCOE		VAN NUYS BL & ROSCOE
128	1/4/2012	19:10:00	3008		SANTA MONI	BL	VIRGIL	AV	SANTA MONICA BL & VIRGIL AV
285	1/4/2012	11:00:00	3008		BOYLD	AV	8TH	ST	BOYLD AV & 8TH ST
286	1/4/2012	19:30:00	3008		INEZ	ST	EUCLID	AV	INEZ ST & EUCLID AV
1434	1/4/2012	20:30:00	3008		VINELAND		BURBANK		VINELAND & BURBANK
3006	1/4/2012	12:40:00	3008		SEPULVEDA	BL	VICTORY	BL	SEPULVEDA BL & VICTORY BL
826	1/5/2012	15:00:00	3008		RESEDA		WYANDOTTE		RESEDA & WYANDOTTE
1435	1/5/2012	11:30:00	3008		OXNARD		WHITSETT		OXNARD & WHITSETT
1436	1/5/2012	17:00:00	3008		LANKERSHIM		CHANDLER		LANKERSHIM & CHANDLER
1437	1/5/2012	17:30:00	3008		LANKERSHIM	BL	OXNARD	ST	LANKERSHIM BL & OXNARD ST
1964	1/5/2012	15:10:00	3008		COMMUNITY		FABLE		COMMUNITY & FABLE
1	1/6/2012	13:00:00	3008	S	GRAND	AV	HOPE	PL	S GRAND AV & HOPE PL
906	1/6/2012	12:15:00	3008		VERMONT	BL	FRANKLIN	AV	VERMONT BL & FRANKLIN AV

Figure 3 Collision Data

The Esri ArcGIS software (Esri 2014) provides suitable geoprocessing tools for preparing the collisions table for visualization through a heatmap. First, the collision table is imported into ArcMap software. Second, the motorist-bicyclist collisions are selected from the table and exported to another table. From here it is necessary to create a single field containing the address or intersection of the collision, because the table sent by the LAPD has the address or intersection of collision locations separated into different fields. Concatenating the location fields allows for the geocoding tool to be used. The ArcGIS Field Calculator handles the process of merging the location data into a single field. The location field in Figure 3 displays the merged location data. Once these steps are complete, the motorist-bicyclists collision data geocoding can begin.

An ArcGIS locator is employed to geocode the motorist-bicyclists collisions table. This address locator is created from a point feature class of Los Angeles addresses maintained by the city of Los Angeles Department of Public Works (DPW 2014). Feature class refers to homogenous geographic data that are stored as points, lines, or polygons (Esri 2014c). A map of the geocoded data is displayed in Figure 4.

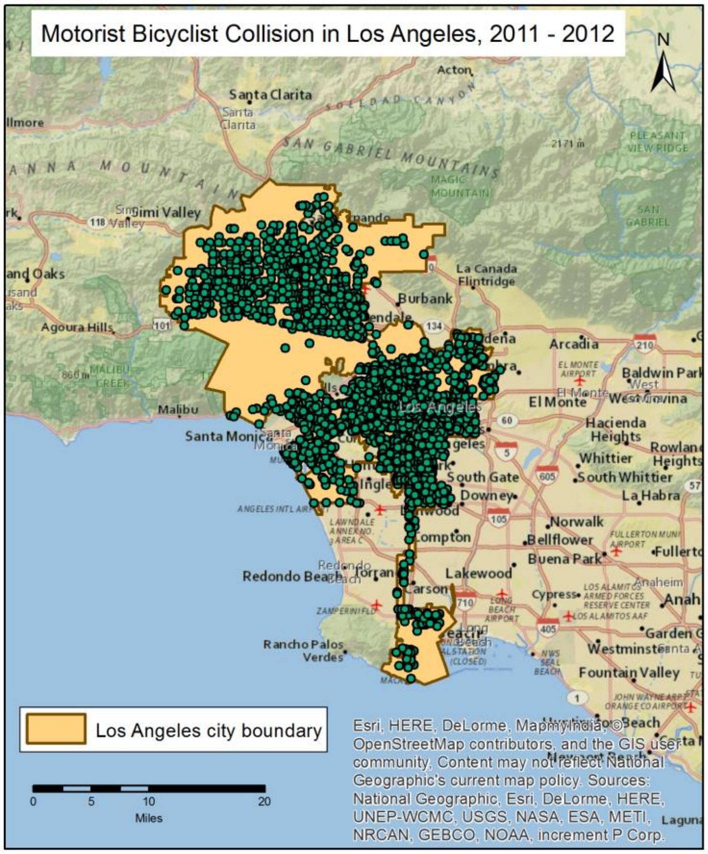


Figure 4 Geocoded motorist-bicyclist collisions

3.2 Collision Heatmap

The motorist-bicyclist collision heatmap in BCApp gives users a reference for where data collection is most essential. BCApp users can position themselves on the sidewalk near these high collision intersections to count bicyclists. Conducting counts in high collision locations produce data that can be used to examine the effectiveness of street design, such as the presence of bike lanes, on collision rates (SCAG and METRO 2013). While BCApp is geared toward bicycle riders, a group likely to be concerned with improving bike infrastructure, anyone can utilize this application provided they can situate themselves on a sidewalk in Los Angeles to count bicyclists. The heatmap included in the BC application and website displays collision history over the years 2011-2012 in Los Angeles.

Several steps are necessary to configure the motorist-bicyclist collision data for use in the heatmap visualization within BCApp. The Esri ArcGIS software provides tools to prepare the geocoded collision data for this visualization. First, using ArcGIS Calculate Geometry tool, fields containing the latitude and longitude are added to the motorist-bicyclist collision feature class. Next, this feature class is exported as a table. The heatmap utilizes the latitude/longitude values stored in this table.

The latitude and longitude fields of the collision table must be stored in a format that can be read by the Google heatmap. This heatmap consumes a JavaScript Object Notation (JSON) file. JSON is built from a collection of name and value pairs and a JSON contains an ordered list of values (Ecma 2013). The Microsoft Excel formula tool provides a quick option for formatting the latitude and longitude values for this JSON. Once the collision table is formatted using Microsoft Excel, data are transferred to a text file and saved as a JSON. Finally, the data are ready to be added to the Google Android heatmap via the Eclipse Integrated Development Environment (IDE). Figure 5 displays a capture of the properly formatted JSON collision data within Eclipse. Following the preparatory work with ArcGIS software and Microsoft Excel, the collision data can be loaded into a Google Android heatmap.


```
[
{ "lat" :  34.04073563, "lng" : -118.2540928 } ,
{ "lat" :  34.0657459, "lng" : -118.23655198 } ,
{ "lat" :  34.04159415, "lng" : -118.25846889 } ,
{ "lat" :  34.06675404, "lng" : -118.2358542 } ,
{ "lat" :  34.03437685, "lng" : -118.27203469 } ,
```


Figure 5 Collision JSON file

Users can alter the radius, gradient, and opacity of the motorist-bicyclist collision heatmap. Table 3 provides definitions for these options and the default and alternative values provided in BCApp heatmap. The gradient values given in Table 3 indicate the low intensity color of the heatmap.

Table 3 Heatmap options

Option	**Definition**	**Default Value**	**Alternative Value**
Radius	Size of Gaussian blur applied to heatmap, expressed in pixels	20	10
Gradient	Range of colors used in heatmap, from low to high intensity	Green	Blue
Opacity	Amount of transparency applied to heatmap, values range from 0 to 1 where 1 is completely opaque and 0 is completely transparent	0.7	0.4

Source: adapted from Google Developers (2014)

Figure 6 displays two visualizations of the BCApp heatmap, one visualization contains the default values, and the other displays the alternative values for the radius. Note that the heatmap in BCApp is for visual references and does not serve as statistically sound spatial analysis results since Google does not provide information on how the heatmap is generated from the input points.

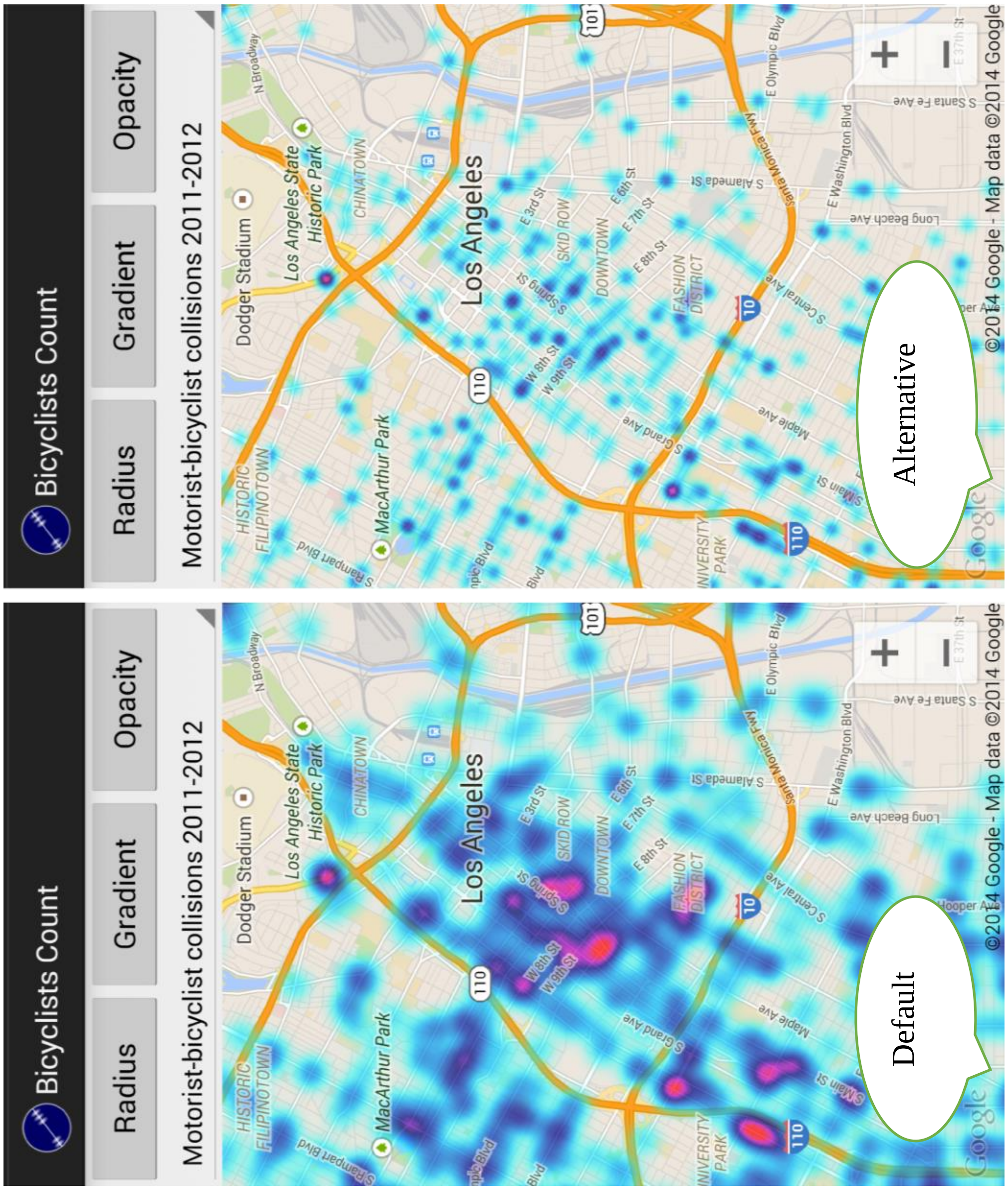


Figure 6 Default and Alternative Radii of Motorist-Bicyclist Collision Heatmap

3.3 Bicyclists Count Application

BCApp creation includes developing Java and XML code by means of the Eclipse IDE. This development process produces four activities available within BCApp. Activities provide a user interface for mobile application through which users can accomplish an event (Android

2014a). The first activity is a login screen. From this screen users log in or create an account to work with BCApp. Figure 7 displays the log in activity.

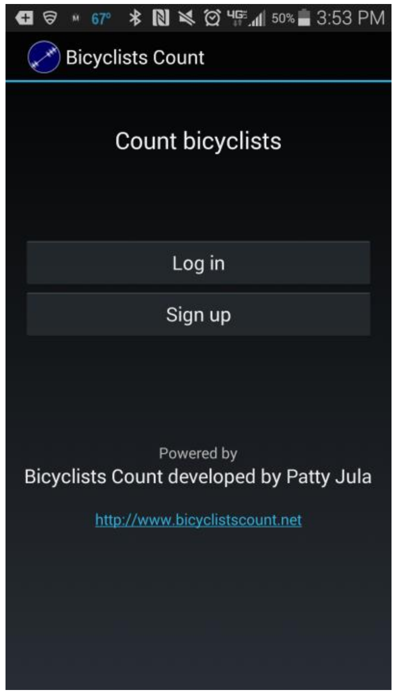


Figure 7 Log In Activity

After signing into BCApp, the main activity page opens to users. Several items are available from the main activity screen. At the top of the main activity are two buttons. The button on the left, labeled “Post count” allows users to open the form to begin counting bicyclists. The button on the right, labeled “Heatmap” gives users a view of the motorist-bicyclist collision heatmap. Figure 8 contains the main activity screen of BCApp.

The middle section of Figure 8 displays the location aware map. In this map, the circular point marker indicates the current user location. This location awareness achieves the project goal of reducing user input to BCApp. Automatically capturing user location using the

LocationListener tool provided by Google (Android 2014b) prevents users from needs to capture their exact location when working with BCApp.

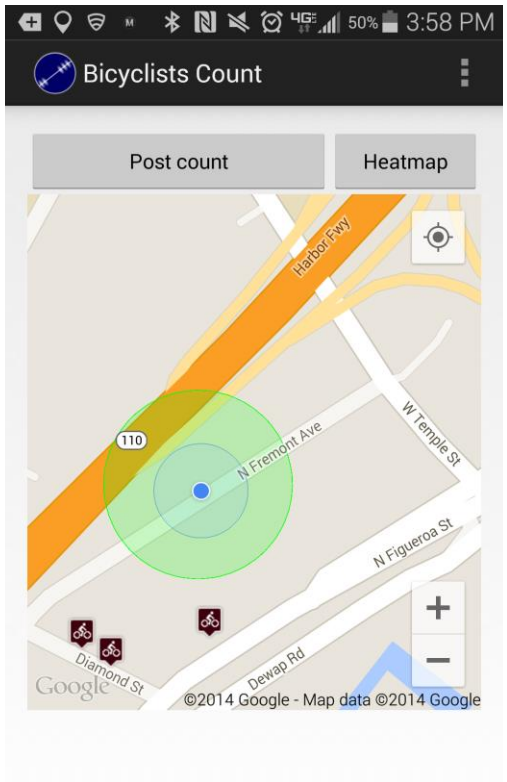


Figure 8 BCApp Main Activity – 250 feet Search Distance

Within Figure 8 a larger circle surrounds the user point marker. This circle is a specified distance around the user called “Search Distance.” The Search Distance tool allows users to review BCApp posts already generated near their location, provided posts exist in those Search Distances. The BCApp posts, depicted by bicyclist markers, represent the VGI component of this project. Users can review where VGI has been generated, similar to how OpenStreetMap contributors can view content generated by other users. In Figure 8 the user cannot review any of the previously generated BCApp posts since none of these posts, depicted by the bicyclist icons, are located in the Search Distance. In contrast, Figure 9 displays the 1,000 feet Search Distance, which encompasses the bicyclist markers.

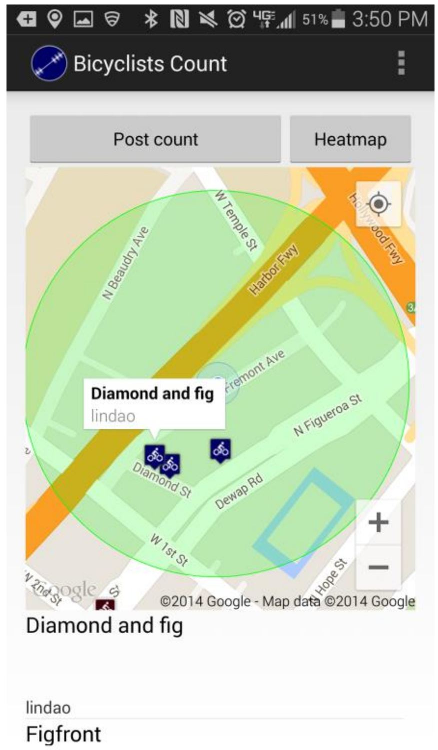


Figure 9 BCApp Main Activity - 1000 feet Search Distance

After the user changes the Search Distance from 250 feet to 1,000 feet, posts within that distance populate below the map and provide their post information via a popup window when the user clicks on the icon. At the bottom of the activity page is a scrollable list of the posts within that fall within the 1,000 feet Search Distance. The bicyclist marker color is dependent on Search Distance. Bicyclist markers in Figure 8 display with a dark maroon color, while the bicyclist markers in Figure 9 contain a dark blue color.

The Search Distance is set through the Settings activity, which is accessible by clicking the vertical dot icon in the top right corner of the main activity screen displayed in Figure 9. After clicking the icon, the setting activity displays. In this activity, users have three Search Distance options available: 250, 1,000, or 4,000 feet. Once users click a Search Distance from the Settings page and touch their Android device back button the map will update with their selected distance. The back button provides Android users with an option for navigating back through previous activity screens (Android 2014c). Figure 10 displays the Settings Activity.

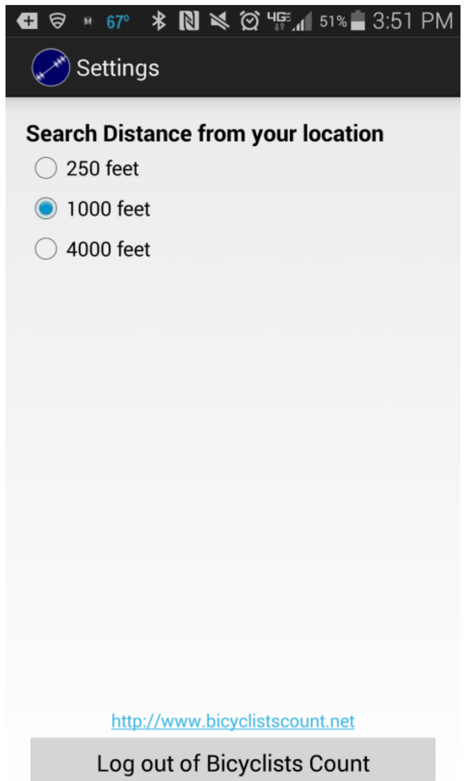


Figure 10 Settings Activity

3.3.1 BCApp Post Form

When a user clicks Post from within the main activity, a data collection, or Post form, is displayed on top of the main activity as shown in Figure 11.

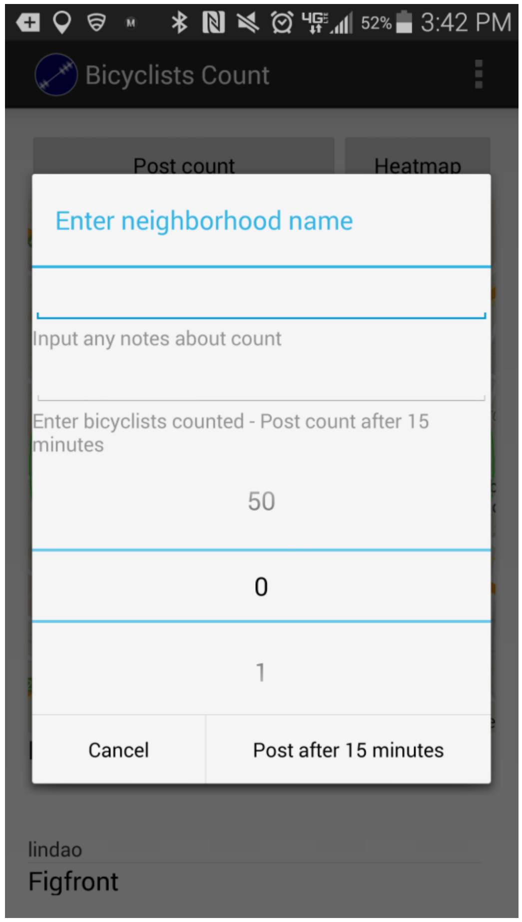


Figure 11 BCApp Post Form

To utilize the fields within the post form several development steps are necessary within the Eclipse Integrated Development Environment (IDE) and Parse.com. First the column name must be added to the BCApp back-end table in Parse.com. Next the form column names are added to the application through Eclipse software. After that the titles for these columns are specified. Table 4 depicts the rules and titles for these columns within the Post form.

Table 4 BCApp Post fields

Back-end field name	**Form title text**	**User input format**
Text	Enter neighborhood name	Text
Notes	Input notes about count	Text
bicyclistCount	Enter bicyclists counted	NumberPicker

There are three fields that users complete within the Post form during the process of collecting bicyclist counts. Within BCApp, users fill out the Enter neighborhood name text with a neighborhood description. This description can be an informal name for the area, or the nearest street intersection. BCApp automatically captures user location when they store the post, which prevents the need for a precise entry within this field. Next, users have the option to enter text under Input notes about count. This field contains user comments about the count they are collecting. For example, users can mention if it is raining during their count or if bike lanes are present on the street.

The final user input field in the Post form captures the number of bicyclists counted by users observing the street from the sidewalk. This field is titled Enter bicyclists counted and it features a NumberPicker (Figure 11). A NumberPicker enables a number from a predefined range to be selected by a user (Android 2014d). The NumberPicker provides users with a method of easily keeping track of bicyclists prior to posting, or saving, their count. These three fields serve as the user-populated values within BCApp.

3.3.2 Back-end Data Storage

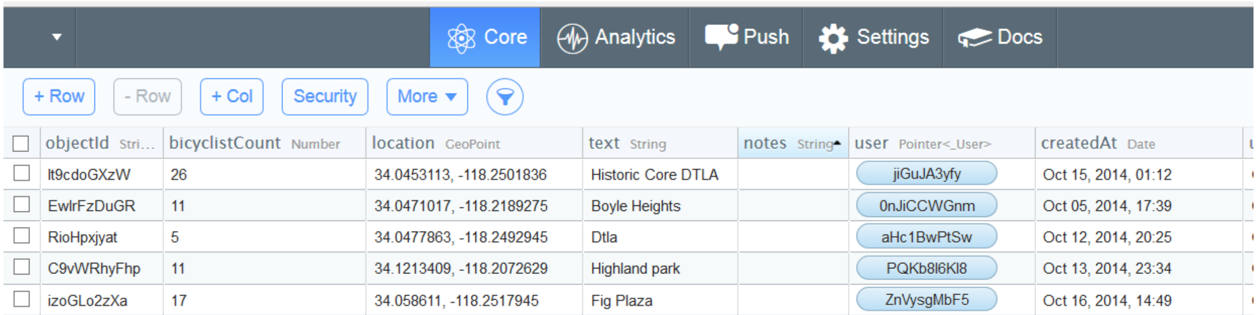
This section describes how data collected in post forms are stored in the BCApp back-end table. Once a user clicks the button “Post after 15 minutes” in Figure 11, the BCApp back-end table updates with that post as a new row in the table. These posts are stored within a table provided by the cloud-based platform called Parse.com. Parse.com offers a method for storing BCApp posts that precludes application developers from dealing with server set up and configuring a database to store user posts (Taylor 2011).

This table stores user-populated fields and automatically generated fields. The BCApp table stores the user-populated fields of text, notes, and bicyclistCount. BCApp also automatically stores information about users in the back-end table. These columns include location, as latitude and longitude in decimal degrees, createdAt, which captures the date and time of the post, and a username column. Figure 12 displays the BCApp table schema. This figure provides the Primary Key (PK) field. The required fields are shown in bold.

BicyclistsCount		
PK	**objectID**	**String**
	bicyclistCount	Number
	location	**GeoPoint**
	text	String
	notes	String
	user	**Pointer<user>**
	createdAt	**Date**

Figure 12 BCApp table schema

Figure 13 displays the BCApp table on Parse. The figure depicts the field names provided in Figure 12. The User field contains obfuscated values.



objectId String	bicyclistCount Number	location GeoPoint	text String	notes String	user Pointer<_User>	createdAt Date
lt9cdoGXzW	26	34.0453113, -118.2501836	Historic Core DTLA		jiGuJA3yfy	Oct 15, 2014, 01:12
EwlrFzDuGR	11	34.0471017, -118.2189275	Boyle Heights		0nJiCCWGnm	Oct 05, 2014, 17:39
RioHpxjyat	5	34.0477863, -118.2492945	Dtla		aHc1BwPtSw	Oct 12, 2014, 20:25
C9vWRhyFhp	11	34.1213409, -118.2072629	Highland park		PQKb8l6Kl8	Oct 13, 2014, 23:34
izoGLo2zXa	17	34.058611, -118.2517945	Fig Plaza		ZnVysgMbF5	Oct 16, 2014, 14:49

Figure 13 Bicyclists Count back-end table

3.4 Bicyclists Count Website

This section summarizes the features of the BCWeb and describes the complementary benefits of creating a website in addition to a mobile application. Next, this section provides a generalized description of the resources utilized to create the website and gives a synopsis of how the website gets updates from the BCApp, underscoring the value of automating updates.

The BCWeb is a valuable complement to the mobile application because, unlike BCApp, the website is not platform dependent. BCWeb is accessible to visitors running any operating system on their computer or mobile device. This differs from the BCApp that requires users have an Android device. Another feature of this website is that it provides documentation on the data sources of BCApp. Adding the data source documentation to the application would have created too much text for a mobile application. This documentation is better stored on the BCWeb.

The website contains the BCApp markers VGI overlaid on the motorist-bicyclist collisions PGI. Overlaying this data within the application would have produced a more crowded map. Unlike the application, visitors to the BC website can review all the BCApp posts. BCWeb features supplement the application, giving visitors additional information that is not found in the application.

The website development begins with creating Hyper Text Markup Language (HTML) pages called index.html, which holds the map, table.html, which contains the BCApp post table, and about.html, which serves as a brief author reference. HTML refers to the elements of a web page. The next step is to add references for the Cascading Style Sheets (CSS), which determines how a browser will display a webpage. The Bootstrap (Bootstrap 2014) framework of CSS and JavaScript, a programming language to create interactive effects in a website, provides a convenient option for website development and is utilized in this website. Minified Bootstrap files are downloaded and added to all three pages of the BCWeb. Finally script tags, meaning calls to external script links, must be added to the three pages. For example a script call to the Google JavaScript visualization library that is necessary for the heatmap is included in index.html. All three webpages contain minified scripts for Bootstrap and jQuery, a JavaScript library of widgets and effects.

The map in index.html contains the markers, generated by users of BCApp, overlaid on the heatmap. The motorist-bicyclist collision latitude and longitude values are provided to index.html as an array, meaning a list of values. Updates to this collision data occur infrequently, making a hardcoded array of this data a suitable option for the collision data. Figure 14 displays several rows of the collision data array.

```
var collisionData = [

new google.maps.LatLng(34.04073563, -118.2540928),
new google.maps.LatLng(34.0657459, -118.23655198),
new google.maps.LatLng(34.04159415, -118.25846889),
new google.maps.LatLng(34.06675404, -118.2358542),
new google.maps.LatLng(34.03437685, -118.27203469),
new google.maps.LatLng(34.04770033, -118.23929058),
new google.maps.LatLng(34.04795959, -118.24378936),
new google.maps.LatLng(34.04919319, -118.23906632),
new google.maps.LatLng(34.05156549, -118.25608105),
new google.maps.LatLng(34.0469046, -118.23808028),
new google.maps.LatLng(34.04090068, -118.25740179),
new google.maps.LatLng(34.04877588, -118.2517762),
```

Figure 14 BCWeb motorist-bicyclist collision data

Although collision data updates occur infrequently, more regular updates to BCApp necessitate the use of automation to reduce user entry to index.html. Automating the process of quickly adding these markers to the website reduces error and time necessary to update this dataset. A Python programming language script (see Appendix A) fulfills this automation need. First, the script connects to Parse.com to get the BCApp table, using the Parse REST Application Programming Interface (API, Parse 2014b). This REST API gives users the opportunity to download, upload, or query Parse tables. Second, the Python script pulls the necessary fields, including the latitude and longitude of the post, out of the table and places them into a JSON file. Finally, the Python script formats the JSON file so that its contents can be displayed on top of the Google heatmap. Figure 15 displays the markers overlaid on the Google heatmap.

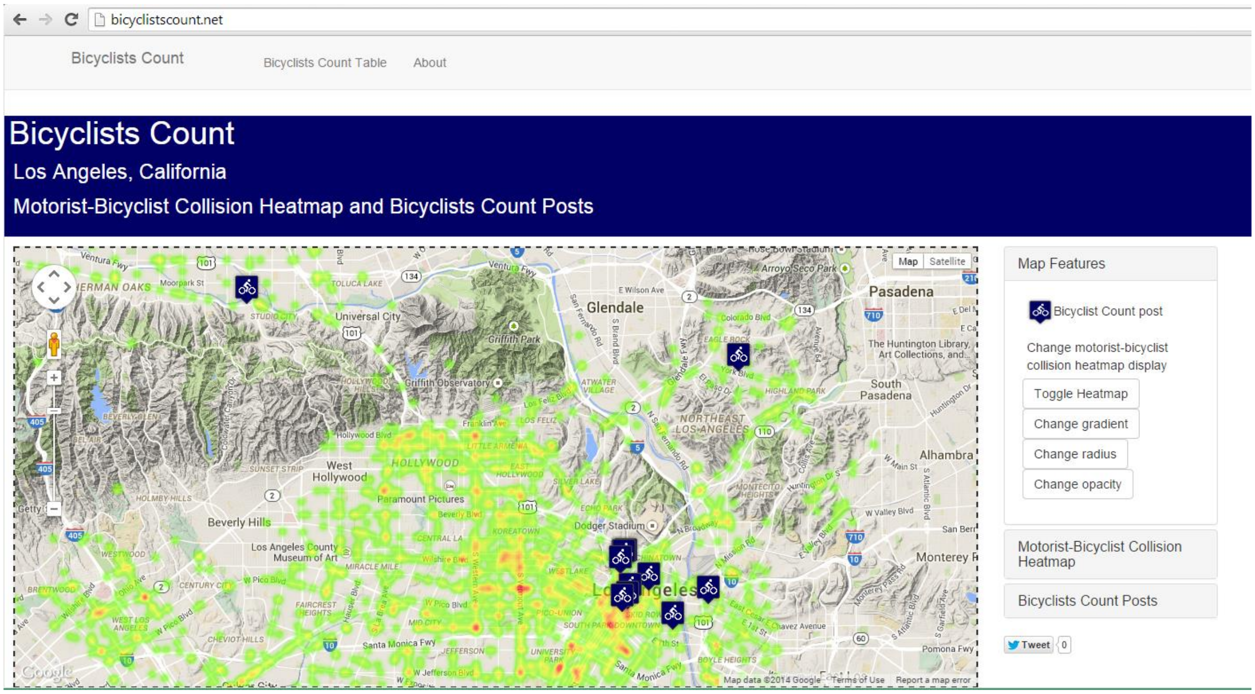


Figure 15 Bicyclists Count Website

Users can interact with the heatmap and markers within the web map in Figure 15. For example, visitors can adjust the gradient, radius, and opacity of the map through the buttons on the right side of the page within the jQuery accordion. This jQuery accordion provides collapsible panels within a limited extent of space within a webpage. In addition to altering the heatmap display and reviewing the markers, users can also review the data sources within the accordion. Figure 16 displays the motorist-bicyclist collision heatmap and BCApp posts metadata. This documentation is more extensive than what would easily fit within BCApp, giving the website additional practicality of being able to provide this information.

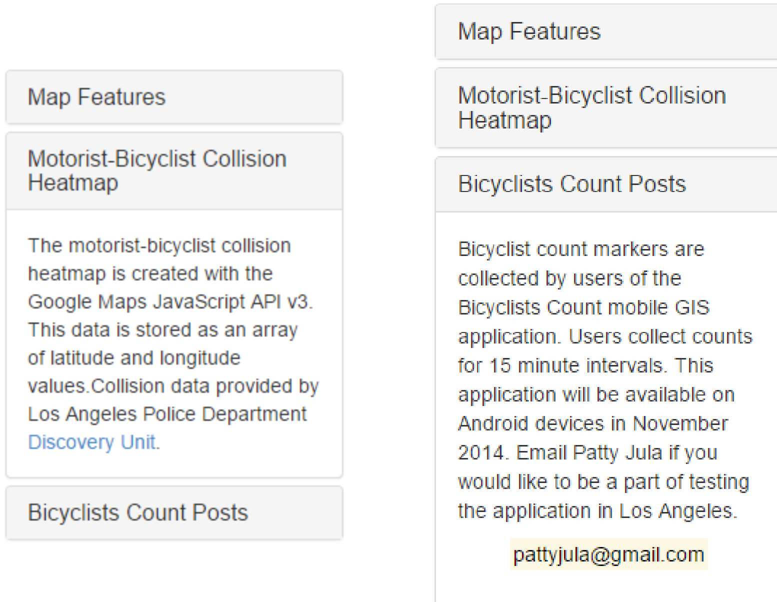


Figure 16 BCWeb metadata

For consistency the bicyclist icon in this map is the same design as the marker in the application. These markers are provided by a map icons website maintained by Nicolas Mollet (Mollet 2014). Users can click on the bicyclist markers to see the neighborhood name and bicyclist count of that marker in a popup window. Figure 18 displays the contents of one of these popup windows.

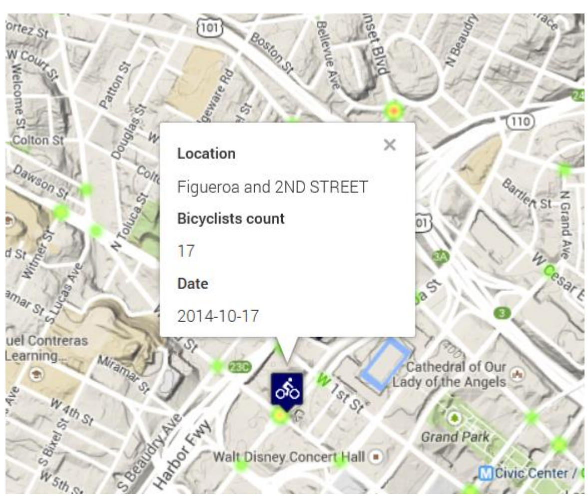
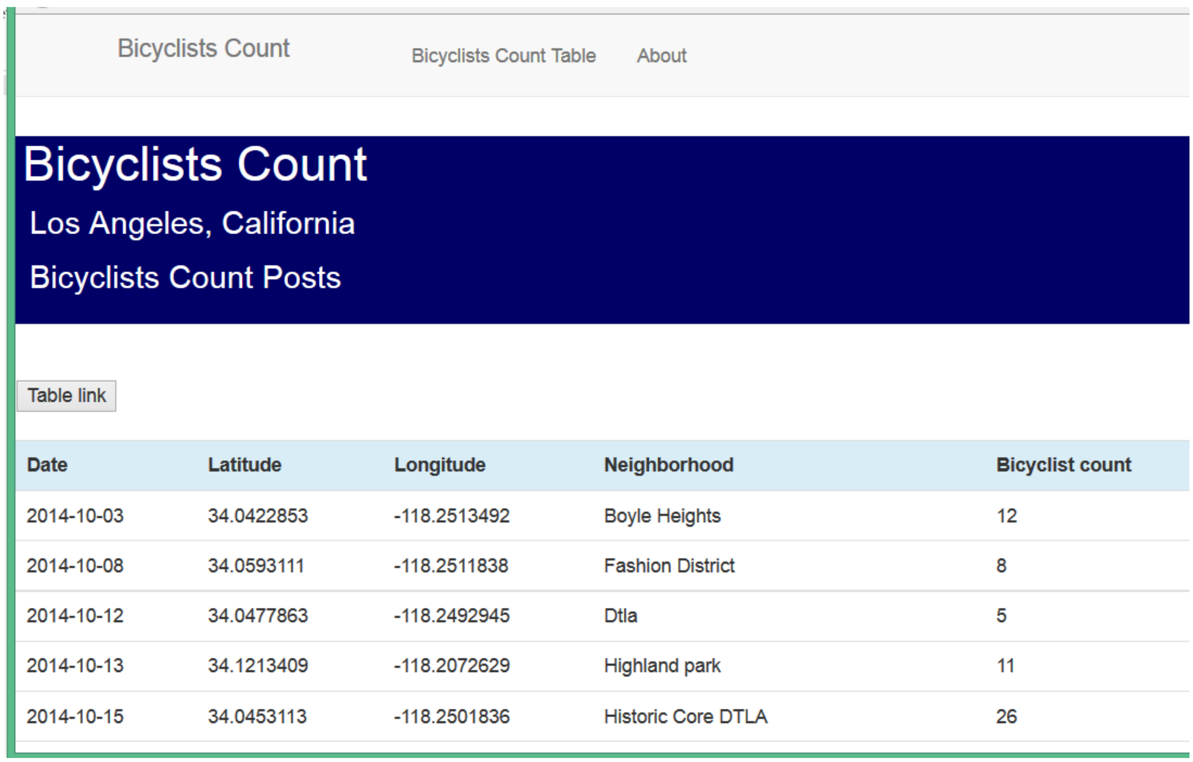


Figure 17 BCWeb popup window

The next step of the web site development is to create the reference table of the Bicyclists Count posts. This table is accessed from the "Bicyclists Count Table" link in the navigation bar. A JSON file of BCApp posts are added to table.html. When a user moves their cursor over a row in the table, the background color of the row darkens slightly. The table is coded to display the date, latitude, longitude, neighborhood, and bicyclist count values of each post. Figure 18 displays the Bicyclists Count Table page.


Bicyclists Count   Bicyclists Count Table   About

Bicyclists Count

Los Angeles, California

Bicyclists Count Posts

Table link

Date	Latitude	Longitude	Neighborhood	Bicyclist count
2014-10-03	34.0422853	-118.2513492	Boyle Heights	12
2014-10-08	34.0593111	-118.2511838	Fashion District	8
2014-10-12	34.0477863	-118.2492945	Dtla	5
2014-10-13	34.1213409	-118.2072629	Highland park	11
2014-10-15	34.0453113	-118.2501836	Historic Core DTLA	26

Figure 18 Bicyclists Count Table

Visitors can also access the JSON file by clicking the “Table link” button to open the JSON file as displayed in Figure 19.

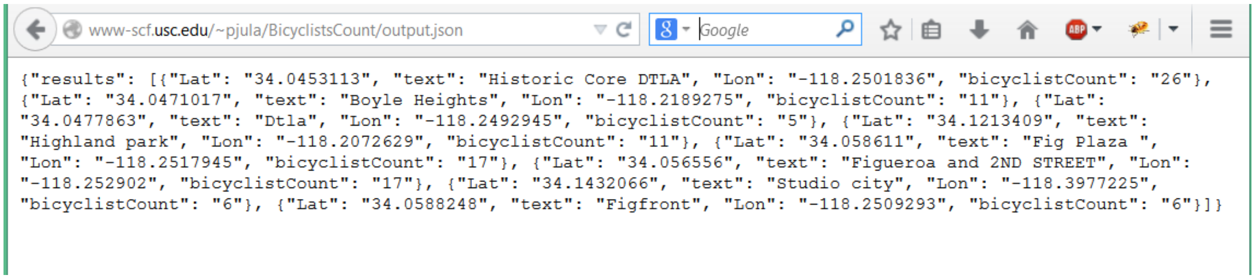


Figure 19 Bicyclists Count JSON data

In addition to giving visitors with a variety of operating systems the chance to review and access BCApp data, this website serves another purpose; providing a link to the BCWeb generated interest by users in evaluating the application. Chapter 4 details feedback from BC evaluators and provides feedback from an LACBC member on the practicality of using a mobile application instead of a pen and paper collection method.

CHAPTER 4: APPLICATION EVALUATION

Chapter 4 describes the BC evaluation methodology. First, Section 4.1 reviews the evaluator selection process. Second, Section 4.2 discusses how evaluators work with BCApp, including how the bicyclist counts are captured. Third, Section 4.3 reviews the BC Survey (Bicyclists Count Survey 2014). In this survey, evaluators gave feedback that determines how well the BC application and website meet the project goals (Table 1). Finally, Section 4.4 provides feedback from a phone interview with an LACBC member about BC.

4.1 Evaluator Selection

In October 2014, I made a request for evaluators through a post on the social networking website, Facebook, and sent an email to a group of co-workers requesting their help in testing BCApp and filling out the survey which contains a link to BCWeb. Through these requests, ten evaluators completed testing and evaluation of BC. The BCApp testing was completed in neighborhoods such as downtown Los Angeles, Little Tokyo, the Arts District, Highland Park, and Studio City.

All the evaluators completed the BCApp testing and submitted their survey responses during October 2014. Of the BC evaluators, eight live or work in Los Angeles, another is based in Irvine, CA and one evaluator lives in Philadelphia, PA. Two of these evaluators possess approximately five years of GIS experience each, and the rest of the evaluators hold a nominal knowledge of GIS.

Half the BCApp evaluations were conducted on weekdays, one evaluation occurred on a city holiday (October 13, Columbus Day), and four evaluations were conducted on weekends. The weekday evaluations were completed in the morning near 8:00 a.m., around noon, and during evening rush hour at 6:00 p.m. The weekend testing occurred in the afternoon, between

1:00 p.m. and 4:00 p.m. BCApp was not available at the time of evaluation in the Google Play Store, and I needed to be present with the evaluators so they could use BCApp on my mobile device, a Samsung S5. The evaluations were completed on days that were agreeable to both the evaluator and me. Evaluations by my co-workers could conveniently be completed on weekdays. Evaluators who lived further away tested BCApp on weekends. For example, the evaluation completed by a friend in Irvine was done on a Sunday afternoon, since that was a convenient day for both of us to meet in Little Tokyo for BCApp testing.

4.2 BCApp Evaluation Process

Once evaluators opened BCApp, I instructed them to look at the motorist-bicyclist collision heatmap prior to conducting the bicyclist count. Some of the evaluators wanted to re-position themselves to conduct the count closer to high collision locations. This repositioning would mean walking a few city blocks at most. Some of the evaluators seemed to already have a count location in mind and the motorist-bicyclist collision heatmap did not have much influence on them. For example, the count locations in Studio City and the Arts District already seemed to be pre-selected locations by the evaluators. Both of these evaluators are bicyclist commuters in Los Angeles, and perhaps they had had experiences or made observations about bicyclist movement in these locations that compelled them to want to capture counts in these areas.

Once situated on the selected sidewalk, evaluators began working with BCApp by creating a username and password. After that, BCApp would display the VGI of bicyclist counts map displayed in Figure 20.

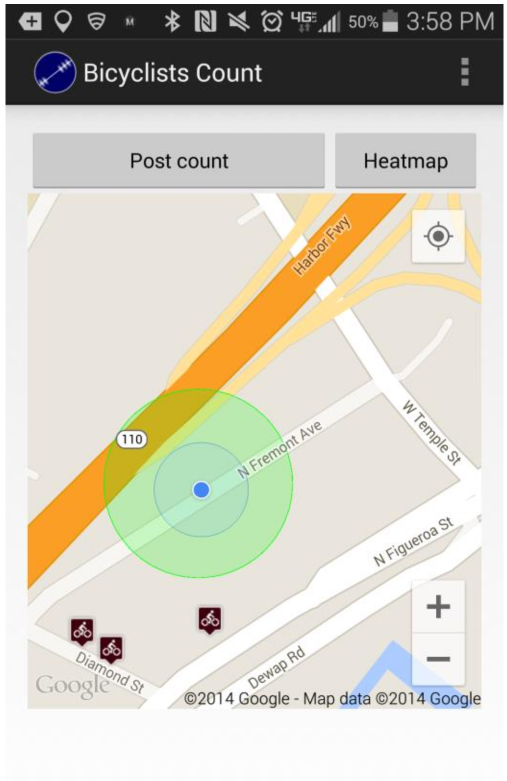


Figure 20 VGI map visible to evaluators on opening BCApp

After the map in Figure 20 was visible and zoomed into user location, I requested users click the Heatmap button to view the motorist-bicyclist collision heatmap found in Figure 21.

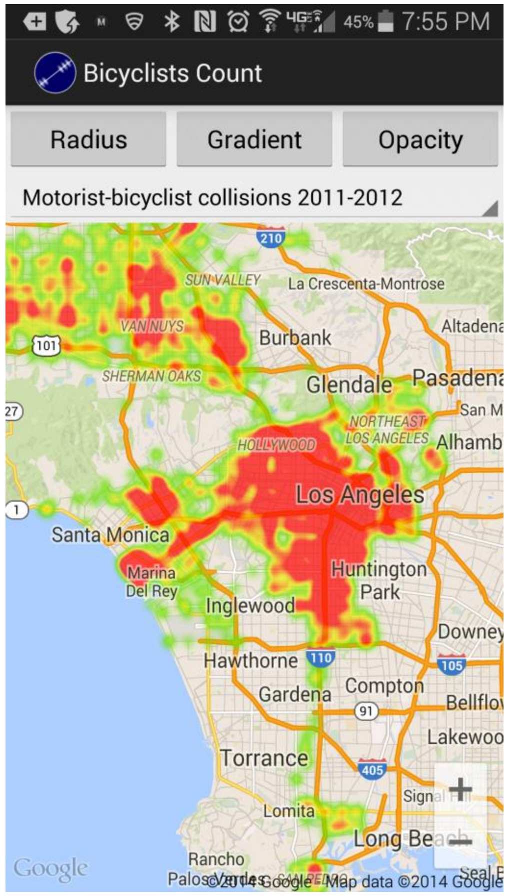


Figure 21 Motorist-bicyclist collision heatmap viewed by evaluators

Evaluators would spend variable amounts of time viewing the heatmap. Many of the evaluators had already viewed the heatmap which I provided in the Facebook and email requests for evaluators. Once evaluators seemed complete with viewing the heatmap, I would instruct them to touch the Android back button to return to the map in Figure 21. Finally, I would tell users to click the Post count button in Figure 21. On clicking this button, BCApp displays the post form found in Figure 22.

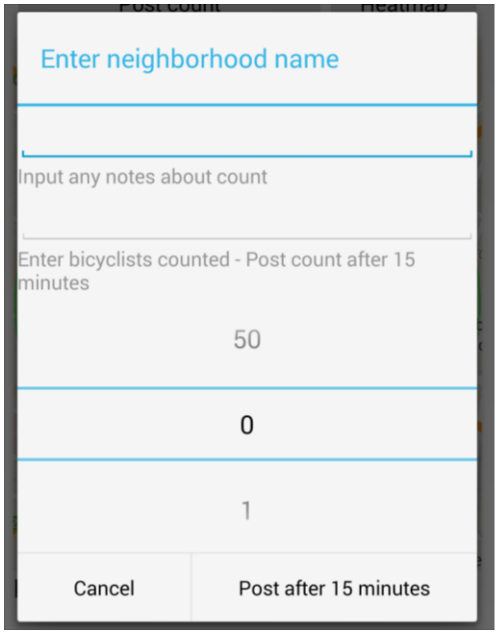


Figure 22 Post form

Evaluators would fill in a neighborhood name and conduct a count, keeping track of the current bicyclist count number with the NumberPicker displayed in Figure 22.

Within the “Input any notes about the count,” one evaluator provided text in the field. Possibly if a more specific statement had been provided such as “State whether a bike lane is present,” this field would have been filled out by more evaluators. After a 15 minute interval, the users would click the button in Figure 22 that states “Post after 15 minutes” effectively saving their post to the Parse.com database for BCApp. This would complete the evaluator use of BCApp for this evaluation process.

On the same day that users completed their evaluation of BCApp, I would email evaluators a hyperlink to the BC survey, which I created through Google Forms (Google 2014). This survey is found in Appendix B, and the results are discussed in section 4.3.

4.2.1 BCApp Technique

Each evaluator tested BCApp for a 15-minute interval from a position on a sidewalk where she or he could observe the road. I instructed evaluators to draw an imaginary line across the street and count bicyclists observed crossing over the line. This imaginary line method is derived from LACBC (LACBC 2014c). This method gives users a focused location to count, as opposed to attempting to count bicyclists within visibility. Users also counted bicyclists observed crossing over the line along the sidewalk on either side of the road.

4.3. BC Survey

BC survey contains eleven questions and is divided into three sections to gain an understanding of evaluator background and whether BC achieved project goals. The first section asks questions to gain an understanding of evaluator demographics, along with their bike riding frequency and mobile device preferences. In the second part of the survey, evaluators respond to questions that determine how well the BCApp meets project goals. In review, goals of this mobile application include: allowing users to quickly capture bicyclist counts, providing a motorist-bicyclist collision heatmap to serve as reference for users, and offering tools within the application that reduce data entry by users. In the third section evaluators provide feedback that indicates whether the BCWeb achieves its project goals. This website was developed to meet project goals of providing an overlay of the Bicyclists Count posts on the motorist-bicyclist collision heatmap and offering a platform independent method for visitors to review BCApp data.

Users respond to questions in two types of formats, assessing how well a project goal is met and allowing users to provide more open-ended responses. With the first type of question, users click the button that corresponds with their answer. These buttons contain responses such as “Yes,” “Slightly,” or “No” and are meant to determine whether the project goal associated with the question was achieved. In other questions, users can enter text, allowing for suggestions on how the application or website can be improved. This text entry feedback serves to generate additional ideas on how the application or website can be improved.

4.3.1 Survey Questions on Evaluator Background

BC Survey collects data on bicycle riding frequency, mobile device usage, and evaluator demographics. BCApp is geared toward bicyclists who are more likely to have an interest in bicycle counting and safety. The first question of this section of the survey asks:

1. **“Have you ridden a bike in Los Angeles within the last month?”**

Six evaluators responded “yes” to this question and four responded “no.” Although only slightly more than half of all survey respondents have biked recently in Los Angeles, a few of the other evaluators also ride bikes. Both the evaluator who lives in Philadelphia and another who lives in Torrance, CA, own bicycles although they responded “no” to this question. The responses and additional knowledge indicate the evaluators mostly fall within the target user group of being concerned about motorist-bicyclist collisions. Figure 23 displays the survey responses for Question One.

Figure 23 Survey Question #1 Responses

Because BCApp is an Android application, the next question seeks to determine whether or not the evaluators would be able to download BCApp onto their device. Question two asks testers:

2. **"What type of mobile device do you use primarily?"**

Eight of the testers responded "iOS" and two responded "Android." Although the majority of evaluators would be unable to download the application onto their device, the survey sample size is small. The opportunity to create BCApp inexpensively makes developing for Android more appealing than developing for Apple iOS. Figure 24 provides survey responses for question two.

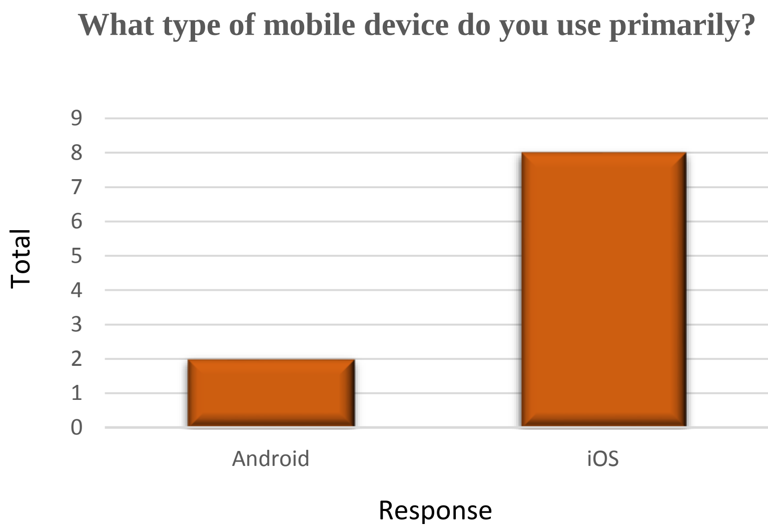


Figure 24 Survey Question #2 Responses

The final two questions of the evaluator background section of the survey seek to capture the demographics of survey respondents. This information attempts to determine if the survey evaluator age and gender are well distributed.

Question Three seeks to capture the age range of survey respondents. This question requests users to:

3. **"Please provide your age range"**

Age range responses are provided as 18-30 years old, 31-40 years old, 41-50 years old, and so on. Seventy percent of evaluators selected the range 31-40 years old, two responded 18-30 years old, and one responded 61-70 years old. Although the majority of evaluators fall in the 31-40 years old age range, the spread of evaluator age is wide, with at least 31 years between the youngest and oldest evaluators. Figure 25 provides responses to this question.

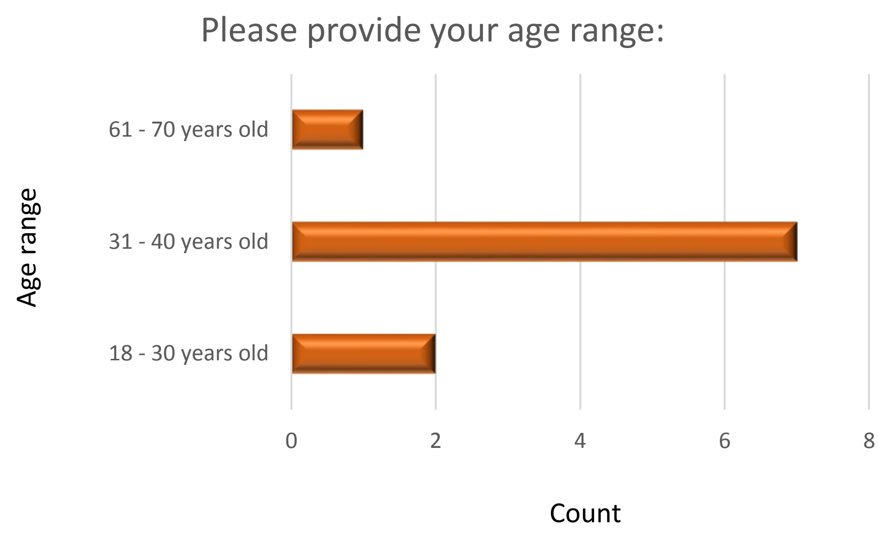


Figure 25 Survey Question #3 Responses

The final question of the evaluator background part of the survey requests evaluators to:

4. "Please provide your gender."

Possible responses to this optional question are "female," "male," and "other." Sixty percent of evaluators responded female, and forty percent responded male. This indicates the evaluator gender for this survey is well distributed. Figure 26 contains the responses this question.

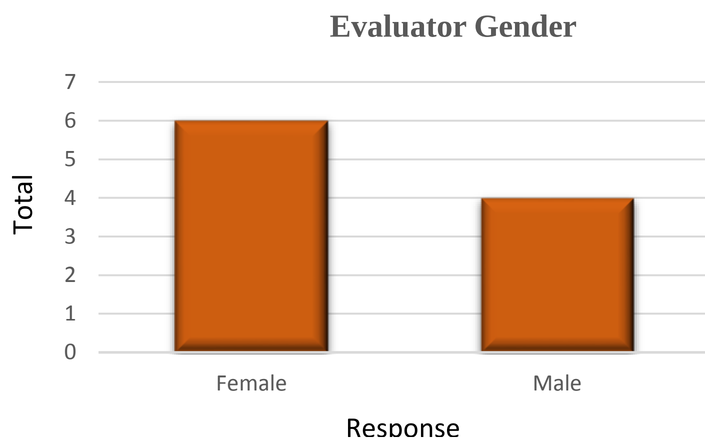


Figure 26 Survey Question #4 Responses

4.3.2 Survey on BCApp

One of the goals of BCApp is to provide a method for storing bicyclist counts that is preferable to a pen and paper collection method for data collectors. The next question of the survey gathers information on whether this goal is achieved with a question in which the user provides an answer by selecting a button with labels of “yes,” “slightly,” or “no” to answer. Question Five asks:

5. **“Would you prefer using Bicyclists Count to a pen and paper method of collecting bicyclist counts?”**

Of the ten respondents, eight answered “yes” they would prefer using BCApp to pen and paper. Two of the respondents answered “no” they would not prefer BCApp to a pen and paper method. Because 80 percent of respondents stated they would prefer BCApp to a pen and paper collection method, the application meets its intended purpose of providing a collection method preferred over paper and pen. The novelty of data collection through a mobile application might have produced this user preference. Figure 27 displays responses to this question.

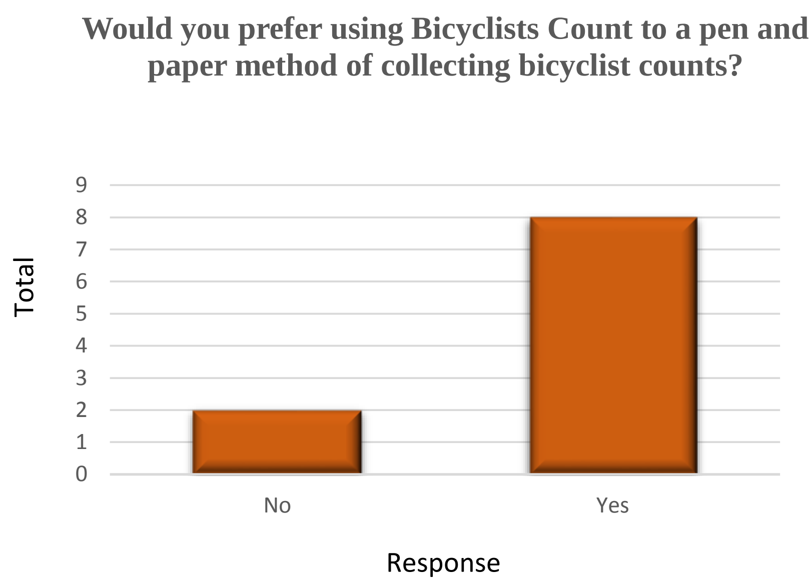


Figure 27 Graph of Survey Question #5 Answers

The second question seeks to determine whether the functionalities in the application are worthwhile to users for the purpose of collecting a bicyclist count. This question asks

6. **"Are the functionalities provided in Bicyclists Count useful?"**

Nine evaluators responded yes, and one responded slightly. These responses indicate the application meets its goal of creating a mobile application method for users to collect bicyclist counts. Figure 28 provides a graph of Question 6 responses.

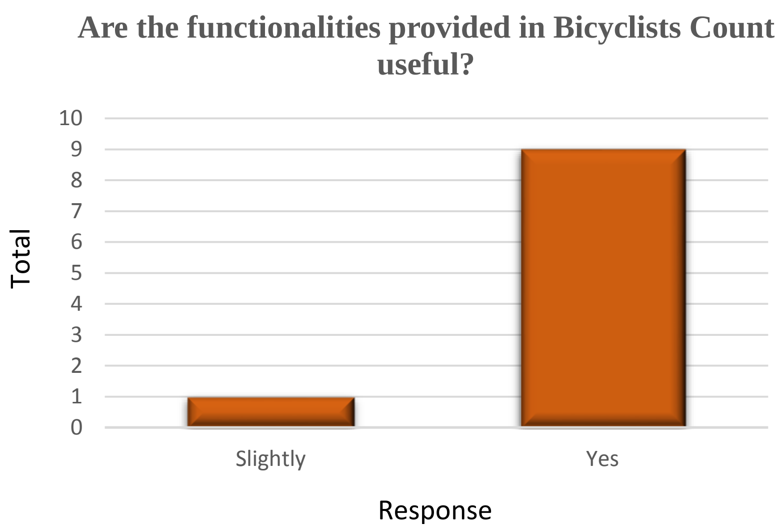


Figure 28 Survey Question #6 Answers

One of the goals of BCApp is providing a heatmap so that users can reference the map to determine where bicyclist counts are most necessary. Toward this goal the survey asks:

7. **"Did the collision heatmap help determine where you want to collect bicyclist counts?"**

Fifty percent of the respondents stated yes, four responded no, and one responded slightly. Figure 29 displays the graph of answers to this question.

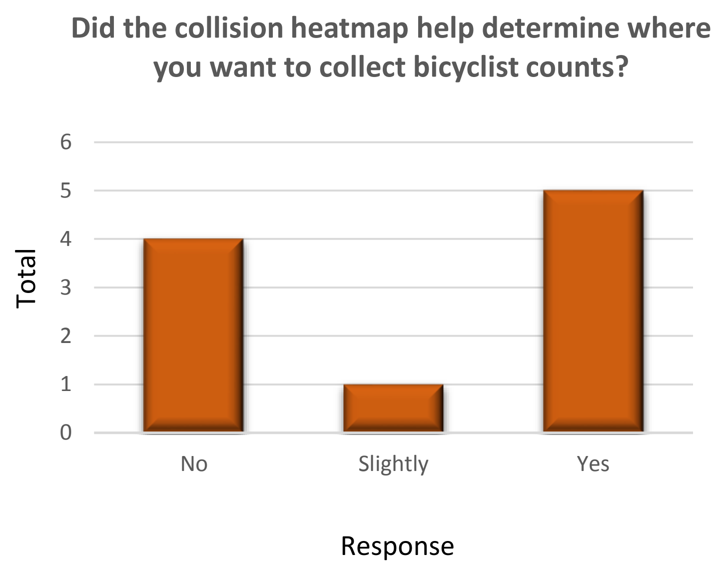


Figure 29 Survey Question #7 Answers

There are several explanations for why more evaluators did not find the collision heatmap affected their count location. BCApp was not yet available in the Google Play Store at the time of testing. This meant that the tests were conducted by meeting the author at a pre-determined location within the city of Los Angeles. Often times the BCApp test occurred near the meeting location. Potentially users of BCApp who download this application from the Google Play Store will look at the collision heatmap prior to going outside to conduct a count. A better question might have been “Would you feel more inclined to conduct counts in high collision areas?” to determine if the heatmap might influence users’ decisions in the future.

The next survey question serves to get an idea of how easy it is to work with BCApp. This question asks:

8. "Is Bicyclists Count easy to use?"

Eight respondents answered "Very easy" and two respondents answered "somewhat easy." These responses indicate that BCApp is intuitive for users. Figure 30 displays survey responses to this question.

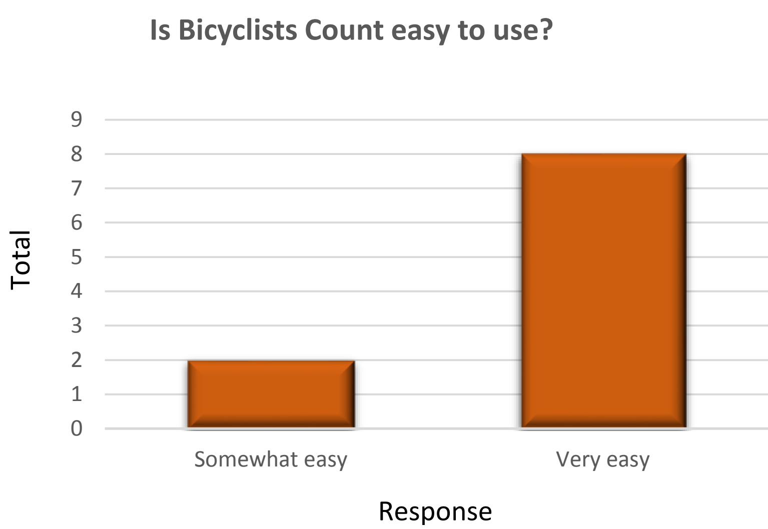


Figure 30 Survey Question #8 Answers

Another goal of BCApp is to minimize data entry by users. To accomplish this goal the application employs NumberPicker allowing users to select a number instead of entering the value. To get a sense of whether testers felt the NumberPicker results in a reduced amount of user input, the next question asks

9. **"Did the number picker reduce the data entry work within Bicyclists Count?"**

Six respondents answered yes, three answered slightly, and one provided no as their response. Ninety percent of evaluators indicate the NumberPicker reduced data entry to some extent, indicating this feature is successful in reducing user entry to the application. Figure 31 displays responses to this question.

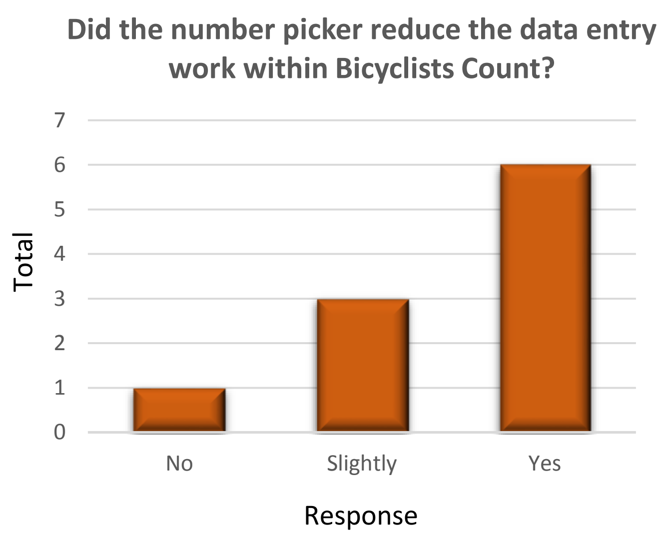


Figure 31 Survey Question #9 Responses

The final part of the BC Survey application questions is open-ended, giving users an opportunity to provide suggestions for BCApp. This tenth question of the survey asks:

10. “Can you suggest improvements for the Bicyclists Count application?”

Six of the ten evaluators provided suggestions. Three of these evaluators suggested including a timer component to the application to make it obvious when the 15-minute count is complete. Rather than a timer, BCApp requires users keep track of the time by way of the clock on their mobile devices. Timer functionality would be beneficial for users of the application and is described in Chapter 5 as a future enhancement. Several other features were suggested for the application. One user suggested creating additional “embedded help text” within the application which could give users the opportunity to learn the application faster. Another tester suggested providing an option in the Post form through which users can share whether it is raining during the count. Although users might be less inclined to capture a count in poor weather, it would be important to indicate if there is rain because this might result in users observing fewer bicyclists than would be present on a rain-free day. While users could share this information within the “Enter notes about count” field, providing a weather related question in BCApp such as “Is it raining?” with “yes” or “no” checkboxes would probably capture this information more consistently.

4.3.3 Survey on BCWeb

The final set of questions within the survey seeks to gain information on whether the BCWeb meets its intended purpose as provided in Chapter 1. In review, BCWeb intends to provide an overlay of the BCApp posts with the collision heatmap. The first BCWeb question states:

11. "Visit the Bicyclists Count website - http://bicyclistscount.net. Does having the heatmap and bicyclist counts overlaid in the same map provide additional insights?"

This question received nine "yes" responses and one "no" response. The person who responded "no" to this question also answered "no" to the previous question, which asks if BCApp seems like a better option than a paper and pen collection method, indicating this user might not be the most technologically-inclined. Figure 32 shows the responses to this question.

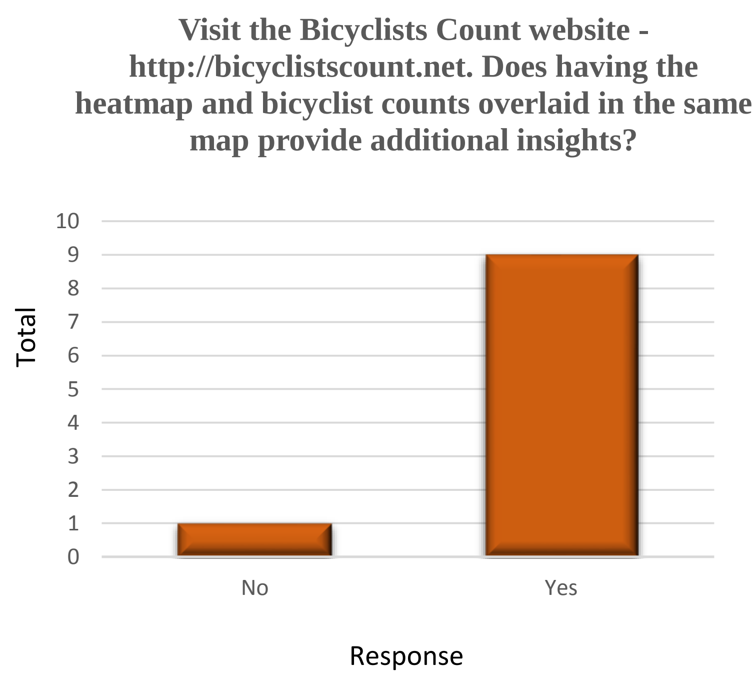


Figure 32 Survey Question #11 Responses

The final survey question provides users with a free form text field for responses. This question asks:

12. "Can you suggest improvements for the Bicyclists Count website?"

Six testers provided suggestions for the website, including adding additional BCWeb data, changing the legend, and expanding the motorist-bicyclist collision heatmap. One evaluator stated that total values for BCApp usage such as "Total Bikes Counted" or "Total Time Spent Counting Bikes" should be added to BCWeb. These values would underscore the effort that goes into creating BCApp posts. Another user provided a suggestion for the legend, stating that a heatmap legend would help to identify high motorist-bicyclist collision locations.

One evaluator suggested expanding the scope of the motorist-bicyclist collision heatmap by creating an "expanded heatmap with stats for other SoCal locations." Creating motorist-bicyclist collision heatmap for other locations would increase the relevance of this application for other locations. Overall the BCWeb feedback demonstrates that it serves as a strong complementary addition to the mobile application.

4.4 Feedback from LACBC Member

While the BC Survey provides insight on whether the application and website meet their intended purposes, conducting a phone interview with an LACBC member about this software provides a broader perspective of its utility. In addition to counting bicyclists, this group also counts pedestrians. Alek Bartrosouf, the Policy and Campaigns Manager with LACBC, said a mobile application would be much easier to work with than a paper form (Bartrosouf 2014). Bartrosouf organized the 2014 bicycle and pedestrian count for LACBC. This response indicates that BCApp is applicable for bicyclist count data collection on a widespread level.

Data storage differs between BCApp and a pen and paper collection method. BCApp stores counts automatically to the Parse database as described in Chapter 3. With the LACBC paper form collection method, count results must be input to a database following the count. Bartrosouf stated: “Skipping the step entirely of entering data into database would save a lot of time” (Bartrosouf 2014). This response indicates the time savings produced in capturing BCApp posts would make it appealing for an organization such as LACBC. Although BCApp provides useful features, further work on this application would be necessary to make it viable for an LACBC count. In this regard, Bartrosouf said: “If the application could be streamlined with the form data that would make it really helpful.” The LACBC count forms collect data not included in BCApp including the direction of bicyclist movement as well as pedestrian counts. Additional functionality for future development of BCApp is outlined in Chapter 5.

CHAPTER 5 CONCLUSION AND FUTURE WORK

This chapter describes the contribution of the application and website toward bicyclist safety and reviews how BC meets the goals described in Chapter 1. First, Section 5.1 discusses the value of creating BC. Second, Section 5.2 reviews goals achieved with this project. Third, Section 5.3 discusses how BC can be improved. Finally, Section 5.4 concludes Chapter 5 with discussing the work necessary to implement BC in other cities, highlighting how the automation found within BC makes this implementation feasible.

5.1 BC Impact

BC is a mashup project composed of VGI and PGI that generates bike count data which can lead to improved bicyclist safety. As Nordback, Marshall, and Jansen (2014) state the improvement of collision data and bike count collection can lead to a better understanding of unsafe roadways. These authors were able to model a bicyclist safety performance function by modeling bicyclist count, vehicle counts, and motorist-bicyclist collisions as described in Chapter 2. While Nordback, Marshall, and Jansen (2014) had sufficient bike count data to complete their bicyclist safety modeling study in Boulder, they write that bicyclist count data is often scarce, making this type of safety modeling challenging. The data generated with BCApp improves the bike count data available in Los Angeles, allowing this type of modeling to be more viable. The utility of BCApp will continue to increase as more users work with this application, which was deployed to the Google Play Store in November 2014. As bicycling is increasingly viewed as a viable form of transportation, a bike counting method such as BCApp becomes more germane.

5.2 Goals Achieved

BCApp achieves the goal provided in Chapter 1 of creating a cost-effective mobile application through which users can generate bicyclist counts in the city of Los Angeles. Developing with Parse, Android, and Google Map helped to achieve this goal of creating an affordable application. Other goals of the application are met as confirmed with feedback from evaluators and through the LACBC phone interview. The survey results for the application and website developed with this thesis are very positive with regards to BCApp providing a functional alternative to paper and pen data collection. The phone interview responses from Alek Bartrosouf with the LACBC were also positive. This preference for a mobile application is probably based in part on how the application collects the data in an electronic format; there is no need to transfer the data from a paper form to a database with the application. Another goal of the application is to provide tools to reduce user entry. Most BCApp evaluators thought the NumberPicker served to reduce data entry, indicating that this goal is met. Another goal is to create an intuitive application for users with a range of GIS experience. Although only two of the evaluators have GIS experience, most evaluators also found BCApp easy to use, demonstrating the application is intuitive to users.

I tested BCApp on several Android devices to determine if the software installs and runs correctly. A goal of this project is that BCApp run successfully on Android devices. BCApp runs correctly on Android operating system devices including Samsung S5 and S3 phones and a Google Nexus 7 tablet, indicating this project goal is achieved.

The BCWeb serves complementary goals to the application. One of these goals is to overlay the BCApp posts on the motorist-bicyclist collision heatmap. This overlay quickly gives viewers an idea of where additional BCApp posts are needed in addition to giving bicyclists an understanding of high collision locations in Los Angeles. The majority of evaluators found the BCWeb map visualization beneficial. Another goal of this application development process is to automate the update process of the BCWeb. The Python script described in Chapter 3 meets this goal. The creation of this automation process is favorable since it reduces maintenance time necessary for BCWeb. The count method available with BCApp

5.3 Future Improvements

Additional features within the BCApp form would make it more appropriate for an organization such as LACBC to use. It is likely this form would need to be extended to capture pedestrian counts to become more useful for LACBC. The LACBC 2014 count requests users to collect bike as well as pedestrian counts (LACBC 2014c). In addition, adding checkboxes by which users could capture whether a bike lane is present on the road where they are collecting counts would also make BCApp more similar to the LACBC form. Fortunately these additions to BCApp would be trivial because the framework for editing the form and adding new fields to the Parse database have already been accomplished, as outlined in Chapter 3.

Beyond developing BC for use by a bike coalition or city government, this project is freely available to citizens interested in capturing data that can provide a better understanding of where they live, work, and commute. As mentioned in Chapter 4, several evaluators of BCApp had predefined locations in mind where they wanted to conduct their bicyclist count. This intentional location selection indicates there are people interested in learning and collecting data that can improve their bicycling routes. BCApp represents a data collection tool empowers users to capture data about their environment.

As some of the survey respondents suggested, including a timer in the application would be beneficial for the application. A timer would increase the accuracy of the counts collected, since users would have a clear cut off time for a count. Potentially a timer would inspire additional counts. If a user notices several bicyclists as about to cross over the invisible line the user drew across the road just as a 15 minute count interval is ending, the user might be inspired to conduct another 15 minute count to capture those bicyclists. Possibly BCApp will develop an enthusiastic user group committed to counting bicyclists outside of an organization such as LACBC.

5.4 Applying BC in other Cities

The motorist-bicyclist collision data that serves as the PGI of this mashup application prompted interest in seeing this type of data in other locations. As mentioned in Chapter 4, one evaluator wanted the motorist-bicyclist collision heatmap created for other part of southern California. When BCWeb was shared on Facebook, one person stated he “would love to see Portland hooked up on this [motorist-bicyclist collision heat] map.” This heatmap creation is possible when there is motorist-bicyclist collision data with locational information, as is the case with the data the LAPD provided. BC can be advanced to include other cities where there is motorist-bicyclist collision data and people motivated to capture bicyclist counts. The automated process of formatting the BCApp posts generated by users of the application for the BCWeb ensures that updating the website with new posts is efficient. This automation lends BCApp to be applied in other cities and towns with an interest in collecting bicyclist counts.

REFERENCES

Android. 2014a. Activity. Android Developers. http://developer.android.com/reference/android/app/Activity.html (accessed 23 October 2014).

___. 2014b. LocationListener. Android Developers. http://developer.android.com/reference/android/location/LocationListener.html (accessed 6 December 2014).

___. 2014c. Navigation with Back and Up. Android Developers. http://developer.android.com/design/patterns/navigation.html (accessed 23 October 2014).

___. 2014d. NumberPicker. Android Developers. http://developer.android.com/reference/android/widget/NumberPicker.html (accessed 10 October 2014).

American FactFinder. 2014. Annual Estimates of the Resident Population for Incorporated Places of 50,000 or More, Ranked by July 1, 2013 Population: April 1, 2010 to July 1, 2013 - United States -- Places of 50,000+ Population. United States Census Bureau. http://factfinder2.census.gov/faces/tableservices/jsf/pages/productview.xhtml?src=bkmk (accessed 5 September 2014).

Bartrosouf, A. Phone interview with author. November 3, 2014.

Bicyclists Count Survey. 2014. Google Forms. http://goo.gl/forms/n1unPO1inj (accessed 7 December 2014).

Bootstrap. 2014. http://getbootstrap.com/ (accessed 10 October 2014).

Carter, A. 2013. Email with author, October 9, 2013.

Celino, I. 2013. Human Computation VGI Provenance: Semantic Web-Based Representation and Publishing. *IEEE Transactions on Geoscience and Remote Sensing* 51(11): 5137-44.

Chaney, R. A., and C. Kim. 2013. Characterizing Bicycle Collisions by Neighborhood in a Large Midwestern City. *Health Promotion Practice* 15:232-242.

DPW (Department of Public Works). 2014. Los Angeles Address Feature Class. Feature class available through author job.

Dumbaugh, E, and W. Li. 2010. Designing for the Safety of Pedestrians, Cyclists, and Motorists in Urban Environments. Journal of the American Planning Association 77(1):69-88.

Ecma International. 2013. The JSON Data Interchange Format. http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf (accessed 8 October 2014).

Esri. 2013. ArcGIS Runtime SDK for Android. http://resources.arcgis.com/en/help/android-sdk/concepts/index.html#/system_requirements/011900000004000000/ (accessed 7 September 2014).

___. 2014. An introduction to commonly used geoprocessing tools. http://resources.arcgis.com/en/help/main/10.1/index.html#//018p00000002000000 (accessed 6 December 2014).

___. 2014b. Collector for ArcGIS. http://doc.arcgis.com/en/collector/#systemRequirements (accessed 14 September 2014).

___. 2014c. Feature class basics. http://webhelp.esri.com/arcgisserver/9.3/java/index.htm#geodatabases/feature_class_basics.htm (accessed 2 November 2014).

Feixiang, C., M. Xiao, and N. Shaoliang. 2013. Organization and Correction of Spatial Data in Mobile GIS. *Journal of Networks* 8(7):1514-20.

FHWA (Federal Highway Administration). 2014. Pedestrian and Bicycle Sensor Technology. http://www.fhwa.dot.gov/policyinformation/travel_monitoring/pubs/pedbikedata.cfm#Question_15 (accessed 28 November 2014).

FHWA (Federal Highway Administration). 2006. Sensor Technology. United States Department of Transportation. http://www.fhwa.dot.gov/publications/research/operations/its/06108/02.cfm (accessed 28 November 2014).

Gandhewar, N. and R. Sheikh. 2010. Google Android: An Emerging Software Platform For Mobile Devices. Special Issue, *International Journal on Computer Science and Engineering* 12-17.

Gartner. 2013. Gartner Says Worldwide PC, Tablet and Mobile Phone Shipments to Grow 4.5 Percent in 2013 as Lower-Priced Devices Drive Growth. http://www.gartner.com/newsroom/id/2791017 (accessed 1 August 2014).

GIS Cloud. 2014. Mobile Data Collection. Google. https://play.google.com/store/apps/details?id=com.giscloud.mdc&hl=en (accessed 30 July 2014).

Google. 2014. Google Forms. http://www.google.com/forms/about/ (accessed 18 December 2014).

Goodchild, M.F. 2008. Commentary: whither VGI? *GeoJournal* 72(3/4):239-44.

Google Developers. 2014. Google Maps Android Heatmap Utility. https://developers.google.com/maps/documentation/android/utility/heatmap (accessed 14 August 2014).

___. 2014b. Usage Limits and Billing. https://developers.google.com/maps/documentation/javascript/usage (accessed 4 November 2014).

Hyde-Wright, A., B. Graham, and K. Nordback. 2014. Counting Bicyclists with Pneumatic Tube Counters on Shared Roadways. *ITE (Institute of Transportation Engineers) Journal* 84(2):32-7.

Jianu, R. and D.H. Laidlaw. 2013. What google maps can do for biomedical data dissemination: examples and a design study. *BMC Research Notes* 6:179.

Kennedy, R., R. McLeman, M. Sawada, J. Smigielski. 2014. Use of Smartphone Technology for Small-Scale Silviculture: A Test of Low-Cost Technology in Eastern Ontario. *Small-scale Forestry* 13: 101-115.

Kononov, J. and B. Allery. 2003. Level of Service of Safety: Conceptual Blueprint and Analytical Framework. *Transportation Research Record* 1840: 57-66.

Kumar, V. 2014. Making "Freemium" Work. *Harvard Business Review.* http://hbr.org/2014/05/making-freemium-work/ar/1 (accessed 14 September 2014).

LACBC (Los Angeles County Bicycle Coalition). 2014. About the Los Angeles County Bicycle Coalition. Available from http://la-bike.org/about (accessed 30 July 2014).

___. 2014b. Bicycle and Pedestrian Counts. http://la-bike.org/projects/bicycle-pedestrian-counts (accessed 7 November 2014).

___. 2014c. L.A. Bike and Ped Count. https://drive.google.com/file/d/0B62WzEc7Bn7zOFdZcFhmQ3lYa2s/view LACBC (accessed 16 November 2014).

LADCP (Los Angeles Department of City Planning). 2011. 2010 Bicycle Plan. http://planning.lacity.org/cwd/gnlpln/transelt/NewBikePlan/Txt/LA%20CITY%20BICYCLE%20PLAN.pdf (accessed 28 November 2014).

Microsoft. 2014. Visual C#. http://msdn.microsoft.com/en-us/library/kx37x362.aspx (accessed 7 December 2014).

Midnight Ridazz. 2004. About the Ridazz. Available from http://www.midnightridazz.com/about.php (accessed 14 August 2014).

Mollet, N. 2014. Map Icons Collection. http://mapicons.nicolasmollet.com/ (accessed 7 December 2014).

Murray, A.T., T.H. Grubesic, R. Wei, and E. Mack. 2011. A Hybrid Geocoding Methodology for Spatio-Temporal Data. *Transactions in GIS* 15(6):795-809.

NBPD Project (National Bike and Pedestrian Documentation Project). 2014. National Bicycle and Pedestrian Documentation Project. http://bikepeddocumentation.org/ (accessed 20 December 2014).

New York City DOT (Department of Transportation). 2014. Bicyclists. The City of New York. http://www.nyc.gov/html/dot/html/bicyclists/bike-counts.shtml (accessed 28 July 2014).

NHTSA (National Highway Traffic Safety Administration). 2014. FARS Data Tables. http://www-fars.nhtsa.dot.gov/People/PeoplePedalcyclists.aspx (accessed 31 August 2014).

Nordback, K.L., W. E. Marshall, and B. N. Janson. 2014. Bicyclist safety performance functions for a U.S. city. *Accident Analysis and Prevention* 65:114-122.

OGC (Open Geospatial Consortium). 2014. http://www.opengeospatial.org/ogc/glossary/l (accessed 8 September 2014).

Parse. 2014. Parse Core. https://www.parse.com/plans (accessed 7 September 2014).

Parse. 2014b. Rest API. https://parse.com/docs/rest (accessed 7 December 2014).

PBOT (Portland Bureau of Transportation). 2014. Summer Bike Count. City of Portland, Oregon. https://www.portlandoregon.gov/transportation/article/490280 (accessed 28 November 2014).

Pesnichak, D. 2013. Pedestrian and Bicycle Counters. https://www.lincoln.ne.gov/city/plan/mpo/tech/reports/2013/130815/BikePed.pdf (accessed 13 December 2014).

Proulx, F. 2013. City of Davis Bicycle and Pedestrian Counting Plan. http://bicycles.cityofdavis.org/Media/Default/Documents/PDF/Bicycles/Beyond%20Platinum%20Bicycle%20Action%20Plan/Appendix%20D%20-%20Bicycle%20and%20Pedestrian%20Counting%20Plan.pdf (accessed 15 December 2014.)

Roeder, A. 2013. HSPH students propose data-sharing system to improve bicycle safety in Boston. *Harvard School of Public Health.* http://www.hsph.harvard.edu/news/features/hsph-students-spring-challenge-bike-safety-boston/ (accessed 29 November 2014).

SCAG and METRO (Southern California Association of Government and Los Angeles Metropolitan Transportation Authority) 2013. Conducting Bicycle and Pedestrian Counts. http://www.bikecounts.luskin.ucla.edu/ (accessed 4 November 2014).

The Eclipse Foundation. 2014. Eclipse. https://www.eclipse.org/ (accessed 7 December 2014).

Taylor, C. 2011. How Parse wants to make mobile backends easy. *Gigaom.* https://gigaom.com/2011/08/08/parse/ (accessed 4 November 2014).

USDOT (U.S. Department of Transportation). 2014. Bicycle Lanes. Federal Highway Administration. http://www.pedbikeinfo.org/planning/facilities_bike_bikelanes.cfm (accessed 28 November 2014).

Zelt. 2014. Zelt Inductive Loops. http://www.eco-compteur.com/ZELT-Inductive-Loops-.html?wpid=39418 (accessed 13 December 2014).

Zielstra D. and A. Zipf. "A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany." Paper presented at the 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal, 2010.

APPENDIX A: PYTHON SCRIPT TO CREATE FORMATTED JSON FILE

```python
#NOTES: Pull table from Parse.com and format for use in Google Maps web map
#LIMITATIONS: Must use Parse.com ID and REST API keys
#
#AUTHOR: Patty Jula
#
#DATE: October 24, 2014
#
# Import modules
import json,httplib,urllib
#
connection = httplib.HTTPSConnection('api.parse.com', 443) # Must use this address and port
connection.connect()
jsonOut = "C:/GIS/Patty/bicyclistscountnet/json/output.json" # Storage for final json

# Connect to Parse table with ID and REST keys
connection.request('GET', '/1/classes/Posts', '', {
       "X-Parse-Application-Id": "yvnrgLjmaAxg0vvF9gYj3gpFHMYWdZ0BvJ9r768k",
       "X-Parse-REST-API-Key": "CviAhgEEILdi94w8Pon43dNhZpaQHxosaq6JtllA"

     })

response = json.loads(connection.getresponse().read())
d = {}
recordDict = {"results":[]}
#Format each result in response for use with Google webmap
for result in response["results"]:
        # Pulling only the necessary fields from result
        bicyclistCount = str(result["bicyclistCount"])
        text = str(result["text"])
        lat = str(result["location"]["latitude"])
        lon = str(result["location"]["longitude"])
        date = str(result["createdAt"])[:10]
        record = "bicyclistCount :"  + '"%s"' % bicyclistCount
        #print record
#       recordDict = dict([("bicyclistsCount", bicyclistCount), ("text", text)])
        record = {"bicyclistCount":bicyclistCount, "text": text, "Lat": lat, "Lon": lon, "date":date}
        recordDict["results"].append(record)
        print record

with open(jsonOut, 'w') as outfile:

        # Output formatted json to outfile
        json.dump(recordDict, outfile)

connection.close() # Close connections
```

APPENDIX B: GOOGLE EVALUATION SURVEY

Bicyclists Count Survey

* Required

Have you ridden a bike in Los Angeles within the past month? *

- ◯ Yes
- ◯ No

What type of mobile device do you use primarily? *

- ◯ Android
- ◯ iOS
- ◯ Windows
- ◯ Other

Please provide your age range: *

- ◯ 18 - 30 years old
- ◯ 31 - 40 years old
- ◯ 41 - 50 years old
- ◯ 51 - 60 years old
- ◯ 61 - 70 years old
- ◯ No response

Gender:

- ◯ Female
- ◯ Male
- ◯ Other

Would you prefer using Bicyclists Count to a pen and paper method of collecting bicyclist counts? *

- ◯ Yes
- ◯ Slightly
- ◯ No

Are the functionalities provided in Bicyclists Count useful? *

- ◯ Yes
- ◯ Slightly
- ◯ No

Did the collision heatmap help determine where you want to collect bicyclist counts? *

- ◯ Yes
- ◯ Slightly
- ◯ No

Is Bicyclists Count easy to use? *

- ◯ Not easy
- ◯ Somewhat easy
- ◯ Very easy

Did the number picker reduce the data entry work within Bicyclists Count? *

◯ Yes

◯ Slightly

◯ No

Can you suggest improvements for the Bicyclists Count application?

[ ]

Visit the Bicyclists Count website - http://bicyclistscount.net. Does having the heatmap and bicyclist counts overlaid in the same map provide additional insights? *

◯ Yes

◯ No

Can you suggest improvements for the Bicyclists Count website?

[ ]

Submit