



Citizen Science Bee Ecology

*A Major Qualifying Project submitted to the faculty at
Worcester Polytechnic Institute
in partial fulfilment of the requirements for the
Degree of Bachelor of Science*

By:

Jackson E. Oliva

April 27, 2017

Professor Carolina Ruiz, Project Advisor
Department of Computer Science, WPI

Professor Robert J. Gegear, Project Co-advisor
Department of Biology and Biotechnology, WPI

Professor Elizabeth F. Ryder, Project Co-advisor
Department of Biology and Biotechnology, WPI

Table of Contents

- Table of Figures 4
- Abstract 5
- Acknowledgements 6
- 1. Introduction 7
- 2. Background 8
 - 2.1 Bee Information and Bee Identification 8
 - 2.2 Citizen Science 10
 - 2.2.1 App Research 10
 - 2.2.2 User Incentives 11
 - 2.3 Mobile Application Development 12
 - 2.3.1 Building and Testing the Application 12
 - 2.3.2 The Database 13
 - 2.3.3 The User Interface 13
 - 2.4 Conclusion 14
- 3. Methodology 15
 - 3.1 Organizing the Application 15
 - 3.2 Setting up the Application User Interface 16
 - 3.3 Programming the Application Logic 17
 - 3.4 Storing the Collected Data 18
 - 3.5 Conclusion 18
- 4. Results 19
 - 4.1 Logging a Bee 19
 - 4.1.1 Choosing Photo/Video Capture 20

4.1.2 Choosing Guided or Manual Identification	20
4.1.3 The Guided Question Tree	21
4.1.4 The Coloration Slider	23
4.1.5 Selecting the Flower color and shape	24
4.1.6 Confirming and Submitting the Log.....	25
4.2 The ‘Beedex’	25
4.3 Tutorial and ‘My Logs’	27
4.3.1 The Tutorial	27
4.3.2 The ‘My Logs’ Feature.....	28
4.4 Conclusion.....	29
5. Conclusions and Future Work	30
References.....	32
Appendix A: Bumblebee Identification Tree.....	33

Table of Figures

Figure 1: Bumblebee sectional anatomy.....	8
Figure 2: Local New England bumblebee species.....	9
Figure 3: Identification tree snippet. Taken from (Heather, Murphy and Stevens, 2017).....	10
Figure 4: The home screen.....	19
Figure 5: Guidance selection screen.	21
Figure 6: Log a Bee guided identification	22
Figure 7: Log a Bee coloration slider.	23
Figure 8: Flower identification screen.	24
Figure 9: 'Beedex' menu screen.	26
Figure 10: 'Beedex' details screen.....	27
Figure 11: Tutorial screens.	28

Abstract

This MQP sought to allow the collection of bee population data while educating the public on the topic of bee diversity. The end goal of the project was to create a citizen science Android phone application that let users log sightings of bumblebees in the northeastern United States. The application was coded in Java using the Android Studio software and libraries. The final application came with three major features: the ability to log a bee, the ability to look up information about local bumblebee species, and the ability to view past submitted logs. The project was successful in creating a base application to accomplish its goals, and the application will be expanded with additional functionality by other project groups in the future.

Acknowledgements

I would like to thank the following individuals for the tremendous amount of effort and support each of them gave towards the project.

- **Professor Carolina Ruiz** of Worcester Polytechnic Institute for her feedback and encouragement throughout the MQP, which without the entire project would have been impossible.
- **Professor Robert Gegear** of Worcester Polytechnic Institute for his extensive knowledge of bumblebees which helped drive the project.
- **Professor Elizabeth Ryder** of Worcester Polytechnic Institute for helping to identify and correct important issues with project.
- **Kenedi Heather** from the Beecology IQP team for helping form the algorithm that lets users correctly identify a bee's species.
- **Rachel Murphy** from the Beecology IQP team for providing the data for the application's 'Beedex'.
- **Devin Stevens** from the Beecology IQP team for providing both the colored graphics and the real pictures of the bumblebees.

1. Introduction

At any point in time, there exists a multitude of important issues that pose risks to the ecosystems of our planet. One such issue that currently eludes the attention of society is the declining diversity in bee populations throughout the world, which has been caused by problems such as the use of pesticides and other human interaction with bee living spaces. Scientists could benefit from more recent bee population data to help better evaluate the problem. The general public could also use more education on the topics of bee diversity and their population declines to be more informed of this global-scale problem.

To address the issues at hand, this MQP partnered with an IQP (Heather, Murphy and Stevens, 2017) and sought to create a campaign to interact with the public and provide information about different species of bees in the Northeast United States. To narrow the scope of the project, the project decided to focus on the species of bumblebees in the New England region as opposed to the thousands of total bee species. This project developed a Citizen Science Android phone application designed to engage members of the public about the ten varieties of bumblebee that reside in the Northeast. A Citizen Science project, or a project that involves data input or analysis from the general public, was deemed to be an effective approach to simultaneously educate the public about bumblebees while providing more data about the diversity of bee populations in the region. This document focuses on the description of the mobile application design and implementation, and the functionality of the resulting app.

2. Background

The issue of declining bee diversity populations is a serious ecological problem. The project started by researching the issue to determine the most effective way of tackling the problem to educate the public about this issue and to engage the public in helping with field data collection.

2.1 Bee Information and Bee Identification

To start the MQP, there needed to be research done to better understand bumblebees and what species are local to the New England region. The bumblebee body is composed of three major sections, being the head, thorax, and abdomen. Particular colorations and features on these sections can help distinguish different species of bumblebees from each other. A diagram labeling the head, thorax, and abdomen of a bumblebee is shown in Figure 1 below.

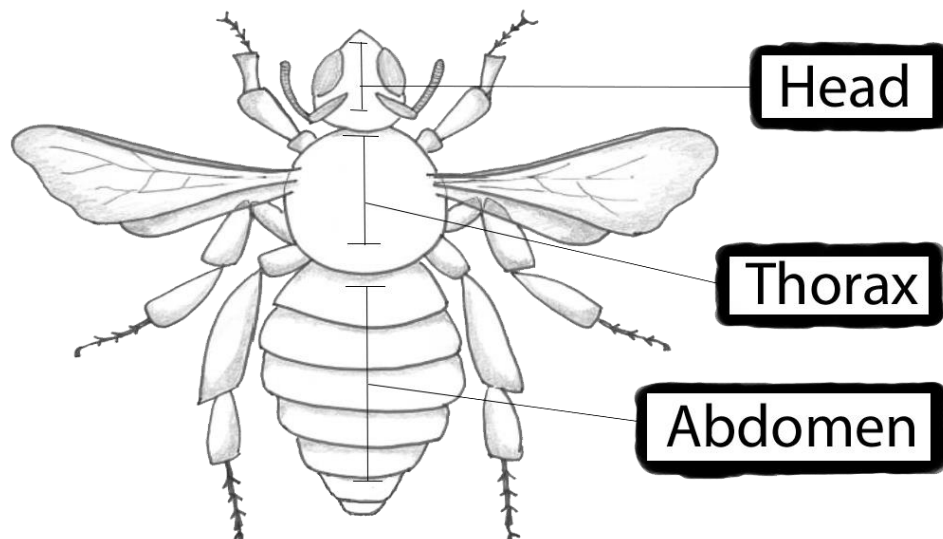


Figure 1: Bumblebee sectional anatomy.

Another important aspect of bumblebees is the different roles they serve for their hive. Worker bees are female and have the primary role of collecting pollen for the hive. Male bees have the purpose of mating with the queen to expand their colony. Queen bees are the sole leaders of their hive and are purposed with reproducing to keep their colony growing in size. Each of the three different role varieties has a different physical appearance, so bees from the same species can still look different from one another.

Knowing this information, the next step was to compile a list of all the local bumblebee species in the New England region. In the end, the list was only ten species long. The small size of the species list allowed the project to start on a narrow scope which could be expanded in the future. A list of the local bumblebee species with accompanying example pictures can be seen in Figure 2 below.

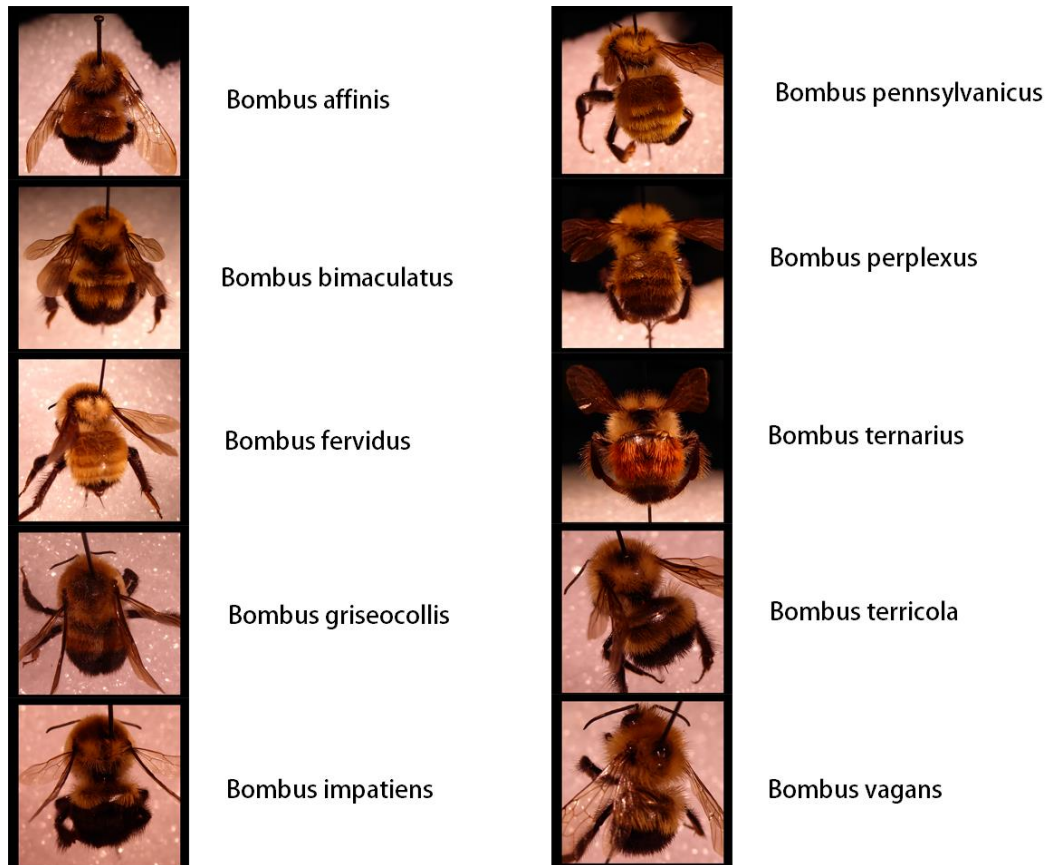


Figure 2: Local New England bumblebee species.

Bee species identification was an important part of the MQP, because the project needed easy ways for the average person to identify different varieties of bees apart from one another. It was deemed best to perform the identification of the different bee species by observing the colorations of the bee's head, thorax, and abdomen. This method allows users to spot differences in bee species by simply spotting unique coloration patterns on the bees. However, to allow a user to consistently correctly identify a bee's species, there needed to be an easy to follow procedure that directed a user towards the correct response. The solution was to create a tree of simple questions about various colorations on the bee. Each question would focus on a particular segment of the bee (head, thorax, and abdomen) and would include a limited number of

responses that narrowed down the list of species that the logged bee could potentially be. A snippet of the tree is shown in Figure 3 below. For the full tree diagram, see Appendix A.

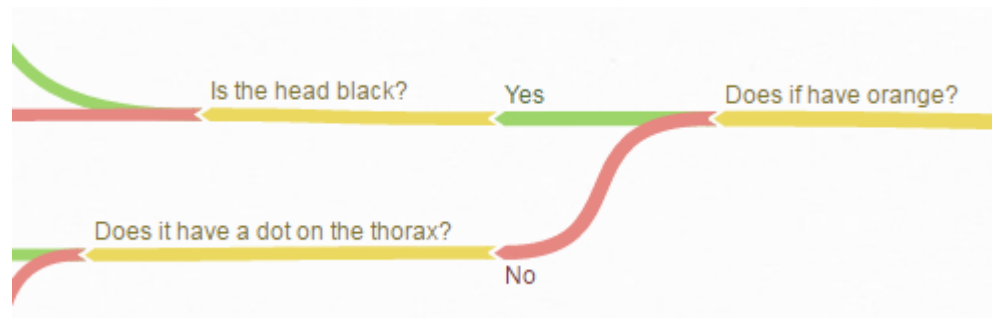


Figure 3: Identification tree snippet. Taken from (Heather, Murphy and Stevens, 2017)

2.2 Citizen Science

Because a Citizen Science application would be a major part of the MQP, the project wanted to analyze the benefits and drawbacks of using this format. For positives, Citizen Science projects are an effective way to get a large sample size and gather a broad range of data. Citizen Science is also useful as it is a relatively cheap and easy way to collect this information. However, the data gathered from these projects could be unreliable, as it is being submitted by random people. To counter this, the project needed a way to accurately provide the users with guidance when they submitted their data to make sure the data is valid. Another issue that can stem from Citizen Science is the potential lack of participation. This can be accounted for by providing some sort of incentive to contributing, such as entertainment or useful information.

2.2.1 App Research

For the construction of the application, the project first looked into similar applications to gather ideas for what makes them successful. The first application found was called Project Noah, which was published by National Geographic in 2010. Its purpose is to help people gather and document wildlife, amassing data on the species that they find in the process. The application incentivizes users to join in by assigning missions for particular species to find and awarding virtual badges to the users that contribute the most, providing a sense of satisfaction to users to encourage them to continue participating (Project Noah, 2013). Another Citizen Science application the project researched was Play to Cure: Genes in Space, which was launched in 2014 by Cancer Research UK. The app is seemingly just a fun game where you pilot a spaceship

through a maze of obstacles, but the user's progress in the game actually helps scientists with cancer research. The paths in the game coincide with paths through DNA microarray data, and users solving the mazes presented to them provides patterns in this data, giving scientists better insight towards figuring out how exactly cancer works (Play to Cure: Genes in Space, 2014). The third application that was looked into was PressureNet, which was published in 2011. The app gathers information from the atmospheric sensors on its users' smartphones, using the large pool of data to then predict accurate weather readings and relay them back to the users (PressureNet, 2011). Finally, the MQP observed an application called The Great British Bee Count, which had many similarities to the basis of the project. The app, created by Friends of the Earth, allowed citizens in the United Kingdom to contribute sightings of different bee species. Of the course of the app's runtime, there were a total of over 370,000 bees spotted, giving scientists a plethora of new data to analyze (Friends of the Earth, n.d.).

2.2.2 User Incentives

Each application that was reviewed had their own unique incentives to convince users to participate, further solidifying the idea that this was necessary for the project's Citizen Science application to be successful. There were a couple of potential options that could be used to encourage users to contribute to the project. One option was to include a game-like aspect to the application. This implementation could have a literal approach like Play to Cure: Genes in Space, in which users can play a game to participate. In the project's case, this could be a game paired alongside the main portion of the application that rewards users for submitting data. The game aspect of the project would certainly help draw in an audience, but it would be a significant time investment as well.

Another option for incentivizing participation in the project would be to follow Project Noah's example and create a rewards system of virtual badges and achievements through missions the app gives to users. This option, while potentially less enticing than a game, would be much simpler to implement and could have similar effects. Systems of achievements that users can complete have a fun lure to them, and as a result make doing the required work seem appealing. This has been shown through many games in the past that have had similar models, displaying that users will do various tasks to earn the virtual 'badge' of having completed an achievement (Madigan, 2016).

The project could also take on aspects of the PressureNet app, relaying accumulated information back to users. While this data would not be as useful as the weather predictions PressureNet provided, it would be interesting and fun for users to know what their participation has been contributing to. This could be accomplished with a list of bee statistics or possibly even a map showing the locations particular bee species have been cited. Depending on the implementation, this could potentially be a small or large time investment, so the inclusion or exclusion of this feature would have to be looked into further.

2.3 Mobile Application Development

Along with the design aspects, there would also be a various amount of technical aspects to the completion of the project. These were the parts of the project that needed to be investigated to make sure the project produced the intended deliverables by the project's conclusion.

2.3.1 Building and Testing the Application

The first major part is the development of the application itself. The project looked into the application being developed in Android Studio due to the high level of documentation and tutorials that exist for Android app programming. Learning the language would be less of an obstacle with more information and educational tools available, which would allow us to maximize time with the actual development of the application.

Testing the Android application when it was being built would become another technical aspect of the project. While having a physical Android phone is the most preferable situation, there is the option to use computer emulators to simulate the Android phone operating system. Phones are optimal because there is a quicker testing time for each build that is created, although there is the cost of obtaining the hardware required. On the other hand, using the emulators built into Android Studio is free, but the time testing between each build is longer, as it takes more time for the emulator to start up than an actual phone. However, there are workarounds to avoid most of the startup time by pre-loading operating system disk images.

2.3.2 The Database

The database that stores the information submitted from the project's application is another technical aspect of the project. This information would need to be stored locally on the smartphone as well as globally on a server. For the creation of the both of these databases, the project explored a variety of options. For data storage on the phone, the project looked into SQLite, which was another free, open-source program that improved upon the 'f_open' function in file systems. It allows the storage of data from the application to the user's phone, as SQLite works with the cross-formatting between different operating systems and file types (SQLite, 2016).

For the server database, PostgreSQL was one option that is free to use and provides open source code, which were useful qualities for the project. The databases you could create with PostgreSQL were also only limited in size by hardware that is used, which would allow us flexibility in the later stages of the project if needed (PostgreSQL, 2016). Another option was MySQL, which had free as well as commercial software packages. The program is also supported by the WPI network, which allowed students to create databases for free (Oracle, 2016). However, these MySQL databases would be limited in size by the pre-defined hardware of the school machines, which could potentially cause problems.

The server database would also need to be able to communicate with the application that the project created. This means that users could submit data to the server database from their smartphones, but also that the database could provide feedback to users as it is updated. The server would need to handle a relatively heavy amount of traffic, as many different users could be submitting data from the application at any given time. The server database would also have to store a large amount of data, such as photos and location information, from the plethora of logs submitted using the application. Without a strong enough server or database storage, the application could cause users problems and delays and other conflicts when trying to contribute new information.

2.3.3 The User Interface

The graphical user interface design for the application would be another challenge for the MQP. The app's multiple features would have to be grouped accordingly and organized in a logical manner that makes them easy to locate and navigate. All of the relevant information for

each feature would need to fit on the smartphone's screen while remaining uncluttered and confusing. The interface would also have to be user-friendly and mobile-friendly. For example, buttons would have to be large enough for users to accurately hit on a touch screen. The interface would also have to be designed for the ease of use by all people, including those with colorblindness or other disabilities. For example, a good use of colorblind friendly design would be to not rely on colors for relaying important information, as it would isolate certain users who could not perceive this information.

2.4 Conclusion

The next chapter in the write-up is Methodology, which will detail the exact procedures that were used to complete the project. Many of these strategies that were chosen were derived from the research in the background.

3. Methodology

This chapter will detail the development decisions made and the obstacles encountered while building the Android application, as well as explain how each of these obstacles were handled.

3.1 Organizing the Application

The first task that needed to be accomplished was organizing the various ideas for the application into one coherent planned structure. Because time was limited, some potential features would also have to be cut out of the final design plans. The beginning step was to identify what would be the most important aspects of the application.

The first feature that would be necessary to appear in the application would be the ability to log a sighting of a bumblebee. A camera capture portion would be needed to allow users to document the bees through image or video. Because a vast majority of the users would not be qualified to perform an accurate species identification on a bumblebee, a section that guides the user to a correct identification would also be necessary. Dr. Robert Gegear, a WPI professor of Biology and one of the project's advisors, felt it would be important to gather data on the coloration of the bee and what flowers it might have been spotted near. These features would be grouped together into the "Log a Bee" portion of the application.

The application, planned to be an education tool as well as a data collecting tool, needed a feature to allow users to learn more about the bumblebees they were logging. The "Beedex" was coined as the solution to this issue, providing an index of the local bumblebee species with pieces of interesting information about each species. Deciding what needed to go into this index was a challenge, because the feature must strive to be informative without overwhelming the user with too much data. In the end, it was decided that the information should all fit on the screen with minimal scrolling and should prominently feature the species name and a sample picture of the bee species.

Many users would most likely want to be able to view their past logs to see their total contributions. Creating a feature that allowed users to look at their previous bee documentations would be one of the high priority tasks for the project. A database would be required to store the

logs over multiple sessions. This information was stored in a local database on the smartphone as well as a server-side database created by Xiaojun Wang, a graduate BCB student at WPI who has helped with the project.

The final feature that was put on high priority was a tutorial section. This provided basic instructions to users who are confused about how they should operate the application. It was decided that there should also be a graphic that labels the different segments of a bumblebee, allowing users to familiarize themselves with the naming conventions used within the application. The instructions and graphic were bundled together to form the complete tutorial feature.

3.2 Setting up the Application User Interface

The user interface (UI) for the application was designed using the tools provided in Android Studio. The software allows designers to use a series of layouts to organize widgets, like buttons and text fields, to form coherent screens that users can interact with in their applications. One way the designer can choose to create screens is by using a drag and drop system of assembling the widgets directly onto a preview of the application screen. The designer can also choose to edit based on a text view of the layout code, which is done in an XML format. To create the application for the project, a combination of both design methods was used. It was more convenient to add small details or to duplicate similar layouts using the text-based design, but it was more helpful to use the screen-preview design when creating a new screen from scratch.

Designing the user interface was tricky due to various aspects. Android Studio was an unfamiliar program to me, and the learning curve was steep at first without prior experience using widgets and layouts. There were plenty of resources online, fortunately, which helped give simple step-by-step introductions to each of the design features. Other screens in the application, like the log display screen, would require trickier elements such as dynamic layout generation. This required the integration of Java runtime code and could not be done with XML alone. Also an important challenge was the general aesthetic of the application.

I decided to go with a simplistic look design for the application. This would allow the primary focus to be towards the coding of new features, but would also make the application

look clean and avoid being too busy or confusing visually for users. To aid users visually, I made the applications color scheme as consistent as possible. Buttons that performed similar features between screens would have the same color. For example, all the 'Next' buttons were colored light-green to establish a color-functionality relationship. Consistency was also maintained as best as possible through the positioning of the various screen features. Titles would always be at the top, prominent features would always be at the center, and interactive widgets would either be at the center or the bottom portion of the screen. This consistency would allow users to more easily use the application, because they would quickly become familiar with the general layout of each screen.

3.3 Programming the Application Logic

Android Studio allows users to program their applications in Java or C/C++, although most of the software's features support the usage of Java. For this reason, I decided to go with the Java programming language for the project's application.

In Android Studio, a screen of an application was created and organized as an object called an 'activity'. As a beginner to Android Studio, I decided to organize each individual feature and functionality of the application onto its own activity. I also decided to hardcode navigations between each separate activity, including 'next' and 'back' buttons. In retrospect, this was slightly inefficient, as this organization did not take advantage of the Android devices ability to remember past activity history, and also did not take advantage of the on-phone 'back' button. This error in organization was caused by my unfamiliarity with the Android phone features, but it did not impact the ending result of the application.

For the programming of the general features in the application, I was able to rely on the base Java libraries as well as the libraries provided by Android Studio. The only feature which required additional library support was the connection to the server-side SQL database. Android Studio comes equipped with many useful built-in objects, such as the `SQLiteDatabaseHelper`, that allowed relatively simple accessibility to otherwise tricky features to implement.

One of the trickier features to implement was the camera feature. Initial research online lead me to believe that I would have to implement camera logic from scratch. However, upon further research there was an ability to integrate the official Android camera feature into the

application. This allowed the implementation of the photo/video feature for logging a bee to be accomplishable in a reasonable amount of time.

3.4 Storing the Collected Data

To be able to study the data that was collected from the application, a database to store this information would be needed. In the early stages of the project, this storage was simply the on-phone local database that stored the past submitted logs. While this database did not help submit the bee logs for study, it helped concept the design for a server-side SQL database table in the future that would hold this information.

Later in the project, a graduate student in the BCB program started developing the central server database for the application. The program used for the database was PostgreSQL, and the table format was nearly identical to that of the on-phone application. Both databases contained similar data such as the logged bee's species, the location the bee was logged, and the time the log was submitted to the server.

As mentioned earlier, the custom PostgreSQL JDBC library was used to connect the application with the server. In the current state of the application, the application only interacts with the server-side database by inserting a new log into the table. In the future with more bees added to the app, the application could expand its functionality to also pull information dynamically from the server database into the 'Beedex'. This would allow users to log a greater variety of bee species and would make updating the scope of the application easier, as it would only require updating the database as opposed to updating the application's code.

3.5 Conclusion

This section covered a basic overview of the various aspects required to complete the production of the Android application. The next chapter will go over the final version of the application and each of its features in detail.

4. Results

This section goes into the details of how each major portion of the Android application was set up. There will be information on how the user can utilize each feature of the application as well as screenshots to display the final appearance of some of the screens. Each of these features is accessible through a button on the home screen of the application, which is shown below in Figure 4.



Figure 4: The home screen.

4.1 Logging a Bee

The feature to let a user log a sighting of a bumblebee is the main advertising point of the application, and it provides the bumblebee population data that will be used for further research.

4.1.1 Choosing Photo/Video Capture

When a user first selects the ‘Log a Bee’ feature, they will be taken to a choice of two buttons that allow the user to either take a picture or take a video of the bee they want to log. Hitting either of these buttons will bring up the integrated Android camera app with either the photo feature or the video feature ready. Once the user takes their photo/video and confirms it is ready for the next step, they are moved to the next screen which allows them to select which method of species identification they wish to use.

4.1.2 Choosing Guided or Manual Identification

The user can choose either to manually identify the species of bee they logged or to go through a guided identification process. The manual identification simply shows a list of the local bumblebee species and allows the user to select the correct species of the bee they captured in their photo/video. This feature is for advanced users who can identify different bee species on their own, and the application lets the users know that they should not select this feature unless they have extensive knowledge of the variety of local bumblebees. Most users will end up using the guided identification, as it is the recommended choice by the application. A screenshot of this screen is shown below in Figure 5.

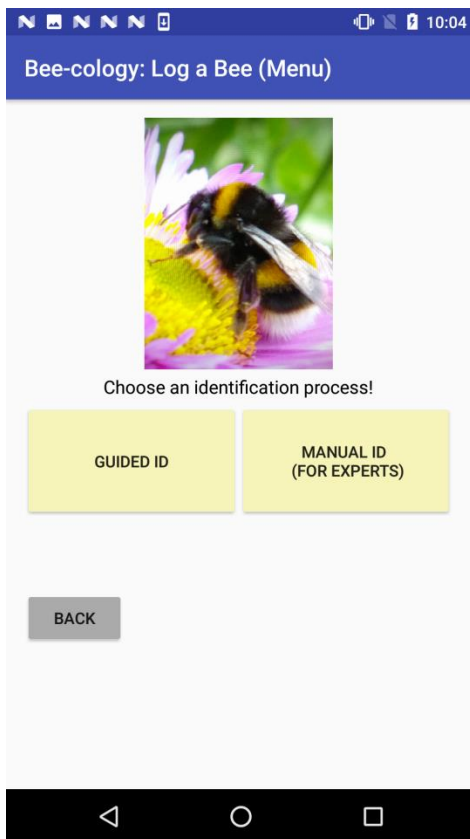


Figure 5: Guidance selection screen.

4.1.3 The Guided Question Tree

Users that choose the guided identification will be asked a series of questions to help narrow down the species of the logged bee. The branch of questions starts with asking the user whether their bee had an abdomen that was mostly black, mostly yellow, or roughly half black and half yellow. To help users, the question is accompanied with a graphic that displays what each of the three options would look like on their bee. Additionally, the picture of their bee is displayed in the upper portion of the screen, allowing users to compare their picture to the graphic to pick the most appropriate response to the question.

After the user answers the first question, they are asked a series of yes/no questions to help further narrow down the bee's species. Each question typically asks about the existence or absence of particular features on parts of the bee. The picture of the bee that the user took is constantly displayed through this process as a reference point. To help users decide correctly, each question comes with a unique graphic to highlight an example of what the question is

referencing. Users are also given the option to go back to a question if they feel that they may have given an incorrect answer previously. An example of a screen from this portion of the application is shown in Figure 6 below.

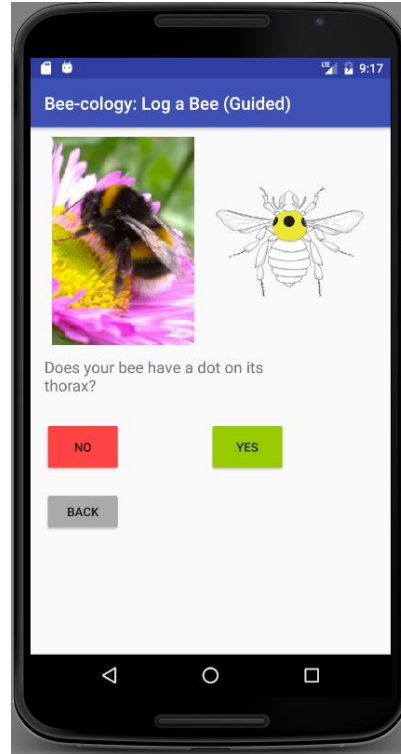


Figure 6: Log a Bee guided identification

If the application cannot narrow down the species of the bee through the chain of questions, the user will reach a stopping point in which the application tells the user it could not properly narrow down the bee's species. The user is given the option to either restart from the beginning of the questions or to go back a step, taking them to the last question they answered. Either choice will not invalidate the user's progress and will keep the bee picture they had taken.

The questions continue until the bee's identification has been narrowed down to one potential species, at which point the application will ask the user to confirm if the species correctly matches up with their bee. To help the user confirm, a bee picture of the species is shown next to the bee picture the user took to allow for side to side visual comparison. For additional help, a button that links to the specie's 'Beedex' page on the application will appear, allowing the user to view more information about the species before making a decision. More information about the 'Beedex' will be covered in section 4.2.

4.1.4 The Coloration Slider

Once the user has confirmed the species identification of their bee, they will be taken to a screen that allows them to best select the colorations of their bee's head, thorax, and abdomen. The user is given a series of buttons that iterate through the different colorations on a graphic of a bee. Based on the species of the user's bee, the user must best select the color patterns to allow the graphic to match the picture they took. An example of how this screen is set up and how the coloration selection graphic changes based on user input is shown below in Figure 7.

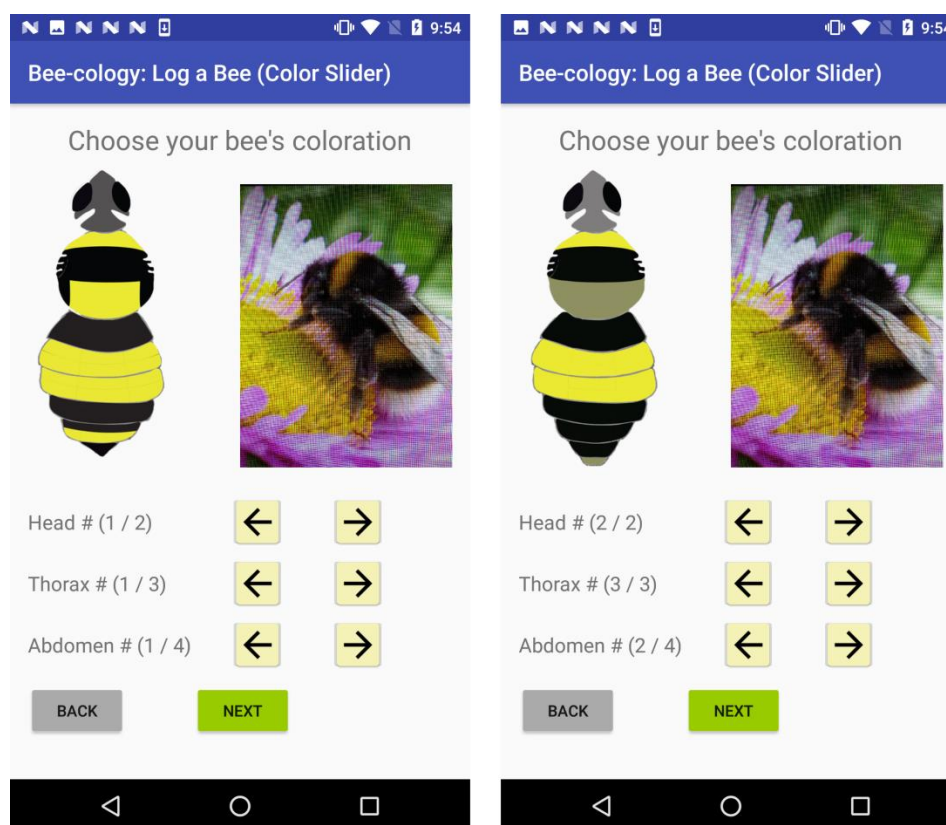


Figure 7: Log a Bee coloration slider.

Based on the species of the user's bee, the coloration options for the head, thorax, and abdomen will be different. This means that the user can only iterate through part colorations that are possible within that bee's species. Because the total number of colorations between all the bee species is greater than 100, allowing the user to be able to iterate through all of the different colorations would be confusing and tiresome to the user. Narrowing down the colorations allows users to more quickly and accurately pinpoint the correct color patterns on their bee without making the process frustrating.

4.1.5 Selecting the Flower color and shape

After they have confirmed their bee's colorations, the user will move onto a screen that prompts them to provide details of a flower the bee might have been spotted near. This feature was specifically suggested by Professor Gegear, who stated that knowing what flowers certain bee species are near would be a large help to scientists analyzing the data.

To identify the flower, users are asked to choose the shape of the flower and the flower's petal color. Both choices are picked through a drop-down menu with a limited number of selections. Because most users do not know the specific names of flower species, choosing these attributes is a simple way for a user of any experience to help correctly identify the flower to scientists who analyze the data. As not every bee will be spotted and logged near a flower, there is the option for the user to not fill in this information if it is not relevant for the log. An example of what the flower identification screen looks like is shown in Figure 8.

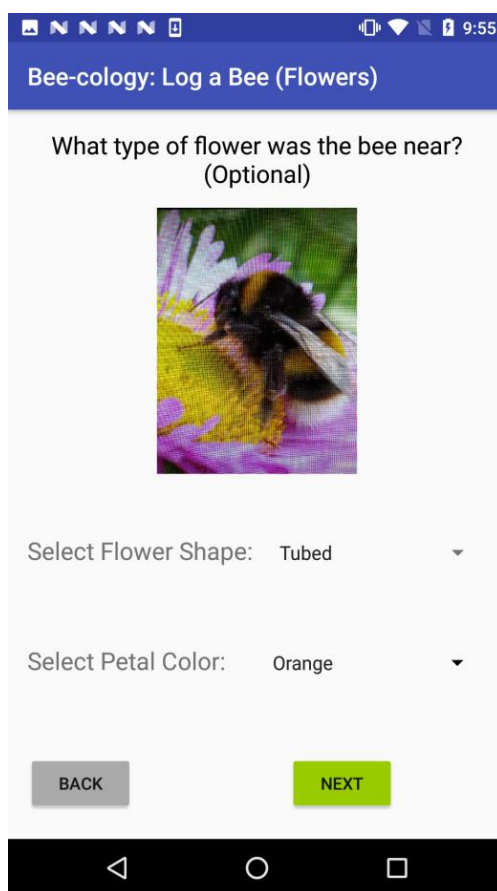


Figure 8: Flower identification screen.

4.1.6 Confirming and Submitting the Log

The final step that the user will take in the process of logging a bee is the confirmation of all their submission details. This screen shows all the information that the user has previously completed in their log, including the picture of their bee, the colorations they selected, and the bee's species. Once the user has looked at the details and has decided whether they are correct or not, they may choose to confirm the submission of the log or go back to correct the details.

Confirming the submission will relay information to the user that the log has been successfully submitted and will take the user back to the home screen of the application. This confirmation puts the information from the log onto a local database on the smartphone as well as submitting the information to the server-side database.

4.2 The 'Beedex'

The 'Beedex', as coined by the project group, is essentially a library of information about each of the local bumblebee species that the user can access from the application. When the 'Beedex' is selected from the home screen, the user is taken to a scrollable list of all the different bee species. Each species is labeled with a picture and its Latin name. Users can find any particular species they want to learn more about and select it via a button labeled 'Learn More'. An example of what the 'Beedex' menu looks like is shown below in Figure 9.



Figure 9: 'Beedex' menu screen.

When the user selects a bee species, they are taken to a screen containing details about the species. This information includes a picture, the specie's Latin and English names, the specie's active times, and a small physical description of species. An example of such a screen is shown below in Figure 10.

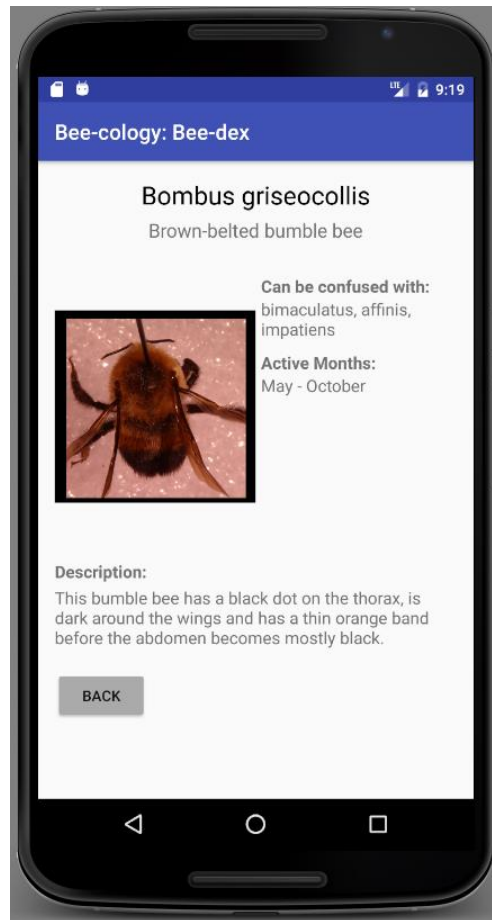


Figure 10: 'Beedex' details screen.

This feature is important to the application because it allows curious users to learn more about the local bumblebees. Because one of the major issues tackled with the project was lack of public awareness on the decline of bee population and diversity, creating a resource in the application for people to learn more about bees was an important goal.

4.3 Tutorial and 'My Logs'

The final features of the application are the tutorial and the 'My Logs' screen. These features serve to both help the user become acclimated to the flow of the application and to help the user view their past accomplishments within the application.

4.3.1 The Tutorial

This feature includes a basic walkthrough of steps for how you would go about logging a bee using the application. This allows the user to understand the basic idea of what purpose for

which they should use the application. The tutorial also includes a graphic that details the different sections of a bumblebee and by what names they are referenced. While the guided identification has detailed helper graphics to show users what each question is asking, it still helps the user to have an understanding of these terms before attempting to log a bee. Screenshots of the tutorial feature are shown in Figure 11 below.

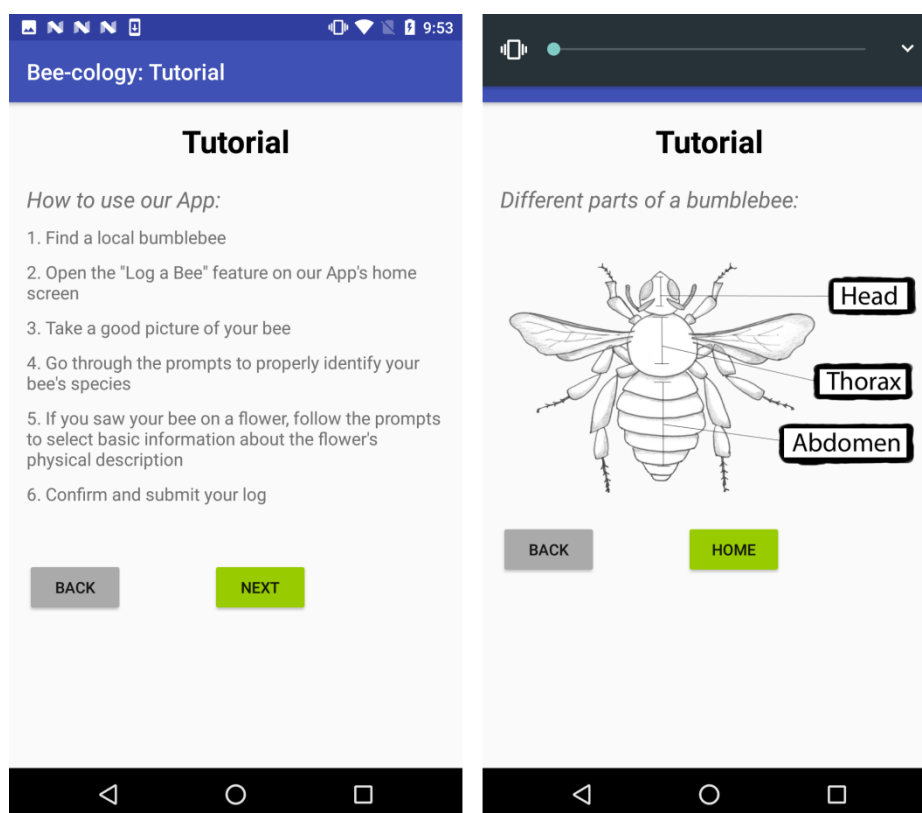


Figure 11: Tutorial screens.

4.3.2 The 'My Logs' Feature

The 'My Logs' feature allows users to look at the details of their past submitted logs. The application accomplishes this by storing the results of each completed submitted log in a database on the smartphone. This means that the data is kept between sessions and is not lost until the application is uninstalled from the phone. While the 'My Logs' feature does not affect the functionality of the application, it allows users to review their progress through their past contributions, which can motivate users to want to contribute more data.

4.4 Conclusion

This section covered the overview of every feature developed for the Android application. The next chapter will go over some concluding remarks, including future work that is planned and possible improvements to the application.

5. Conclusions and Future Work

The MQP was successful in creating a starting point for future work by developing a base application with most of the major functions and features available. With no prior experience coding applications, the framework for the project's application was constructed from scratch and included nearly all of the major features. With the application in its current state, a user can take a picture of a bumblebee, go through a process to identify its species, and submit the resulting log to their phone. Because this was the primary goal of the MQP, the project concluded successfully.

However, there were ways the project could have been improved. Because of inexperience in coding for Android smartphones, there were some features of the application that ended up being redundant or improperly organized for the Android architecture, such as the addition of back buttons onto each screen or the separation of each feature into its own activity. Inexperience with the Android Studio coding environment also caused delays, because some techniques and coding strategies that were unique to the software and its libraries were not initially known. Some features of the application did not get developed as much as they should have been, such as the video capture feature or the 'My Logs' feature. Early development of these features should have been a higher priority and would have made the application more user-ready.

This project has the potential for expansion in the future, including the immediately possibilities, such as increasing the number of bees that users are able to log and extending the region where the application is accessible beyond New England. At the time of the project, Dongsheng Wang, a graduate CS student at WPI, is working on a web-based version of the Android application to help attain a larger user base. Xiaojun Wang, a graduate BCB student at WPI, is also working on the server-side database for which the log data from the Android application and the web application will be stored.

The server-side database has a lot of potential to change the structure of the application. At the start of the project all of the information, such as that in the 'Beedex', was hard-coded directly into the application. Future iterations of the application can change this to pull the

information directly from the server. This also allows the application to be updated more easily in the future, as you will only have to edit the database and not the code directly.

For future work on the Android application, new MQP and IQP teams are already formed to further develop and improve various aspects of the application. These teams plan to add improved bee pictures into the application as well as adding a map feature that shows where bees have been logged. When the application is finally polished and finished with all its features complete, it will be available to the general public through the Android App store. With its official release, the application will be able to gather lots of useful data for scientists while increasing the public's attention towards local bee species, essentially addressing the two problems that this MQP focused on.

References

- About SQLite. (2016). Retrieved October 13, 2016, from <https://sqlite.org/about.html>
- Friends of the Earth. (n.d.). Join the generation that saves our bees. Retrieved October 13, 2016, from <https://www.foe.co.uk/page/bee-cause>
- Kenedi E Heather, Rachel Lee Murphy, and Devin T Stevens. “Bee-cology: Educating the New England Area on the Importance of Pollinator Diversity”. Interactive Qualifying Project. Worcester Polytechnic Institute. March 2017.
- Madigan, J. (2016, July 04). “Why Do Achievements, Trophies, and Badges Work?”. Retrieved October 10, 2016, from <http://www.psychologyofgames.com/2016/07/why-do-achievements-trophies-and-badges-work/>
- Moisset, B. (2010, November 26). Native Bees of North America. Retrieved October 12, 2016, from <http://bugguide.net/node/view/475348>
- Oracle. (2016). MySQL. Retrieved October 13, 2016, from <https://www.mysql.com/>
- Play to Cure: Genes in Space. (2014, February 4). Retrieved October 13, 2016, from <http://scienceblog.cancerresearchuk.org/2014/02/04/download-our-revolutionary-mobile-game-to-help-speed-up-cancer-research/>
- PostgreSQL. (2016). Retrieved October 13, 2016, from <https://www.postgresql.org/>
- PressureNet is joining Sunshine. (n.d.). Retrieved October 13, 2016, from <http://www.pressurenet.io/>
- Project Noah | Networked Organisms And Habitats. (2013). Retrieved October 13, 2016, from <http://www.projectnoah.org/>

Appendix A: Bumblebee Identification Tree

Taken from (Heather, Murphy, and Stevens, 2017)

