Unified Butterfly Recorder
Final Report

butterflies.ece.iastate.edu | butterflies@iastate.edu

Senior Design Team Dec13-08

Ryan Scheel

Curtis Ullerich

Julie Tillman

Cameron Whipple

Client: Reiman Gardens
Nathan Brockman
Anita Westphal

Advisor: Dr. Diane Rover

Table of Contents

Project Overview

Background and Motivation

Background

Butterflies serve as a great indicator species, meaning that their presence, absence, and abundance can be correlated to interesting environmental phenomena. Researchers and conservation workers use butterfly migration patterns and dispersal data to monitor components of the environment from local habitat restoration progress to large scale climate change.

Surveys

Myriad organizations around the world collect or organize the collection of butterfly data, most commonly through surveys. A survey is a collection of butterfly sightings gathered in one general location over a time span that is generally less than a few hours. As with most research, different data is collected with different purposes in mind, so the data required by surveyors for one organization can vary significantly from that of another. This creates problems when trying to look at national or global populations or other larger generalizations.

Protocols

A survey may collect as little information as the presence of a particular species within a county, or data as detailed as the condition of each specimen with light intensity readings at the exact sighting location. The type and method of data collection during a survey is dictated by protocols. Each protocol is designed to answer a specific type of research question. There are many protocols used today. Below are a few of the more common protocols.

Pollard Walk: A surveyor repeats an identical route several times over the course of years, recording sightings visible within a specific range of the path, aiming for consistency of recording. This can allow more rigorous statistical analysis.
Distance Sampling: Record the distance of sightings from a specific line or point in order to estimate local distribution and abundance.

Presence-Absence: Simply record whether a particular species has been sighted at all in an area.
Meandering: Similar to field trips, this protocol involves individuals or groups walking an indeterminate path looking for as many individuals as they can find.
Mark-Recapture: Surveyors capture butterflies with nets. They then place a mark on their wings and release them. Later they return to the same location and capture more butterflies, making notes when they find already marked butterflies.

Problems

There is no national or global standard for the type of data collected, how it is recorded, or the methods of collection. Data collected using one protocol can be similar to that of another protocol, but different enough to not be compatible. Current methods for collecting this data are often done on paper and not catalogued uniformly. There is no central warehouse for butterfly survey data, making large-scale research and conservation efforts challenging.

Objective

The fragmentation of the data collected, the collection method and how the data is stored is a very real problem in the worldwide community of butterfly researchers. By designing an Android mobile app to use during data collection, we aim to help reduce this problem by allowing easy collection of a superset of the data required for each of the many disparate survey protocols. We plan to create the app such that it can easily export data to centralized servers provided by other organizations. By communicating with professionals in the field during the design and development of this system, we hope to achieve wide adoption of our data collection app in the butterfly community and provide a product that matches or exceeds the quality and efficiency of existing survey methods for each protocol.

> "This app will help standardize the collection of data and has the potential of impacting conservation efforts both nationally and globally."
>
> *- Anita Westphal, Reiman Gardens*

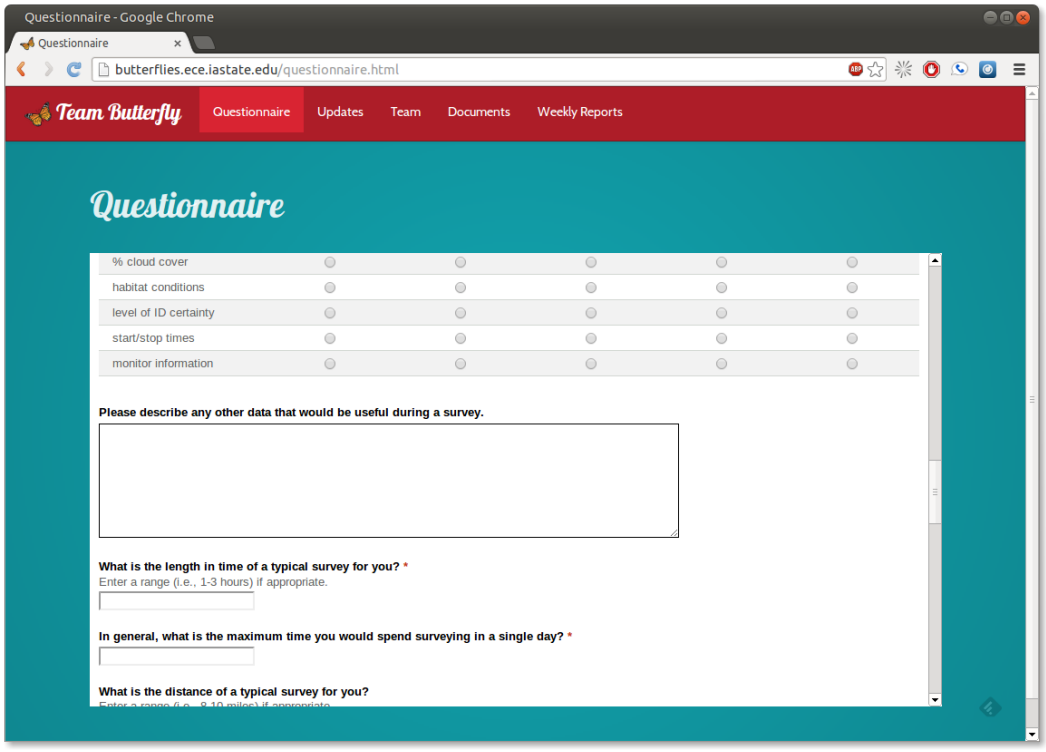
Target Devices

The deliverable for this project is an Android app for phones and tablets that run Android 4.0 or newer. Some such devices may not have all supported sensors (e.g. no barometer), in which case UBR uses as many sensors as possible, and simply does not collect data from missing sensors. An ideal device would have GPS, mobile data connection, light sensor, thermometer, barometer, humidity sensor, and a camera. For our rationale on which devices to support, see Design Tradeoffs.

Market and Literature Survey

Questionnaire

Nathan (our client) provided a list of thirty butterfly professionals and enthusiasts from around the world for us to contact for preliminary feedback. We designed a questionnaire with the objectives of determining which protocols are used the most, ascertaining which data points are perceived as most essential, gathering user interface preferences, and eliciting any helpful ideas. To view the response summary, see Appendix B.



We sent a questionnaire to surveyors to gather user requirements.

The results from this questionnaire validated most of our hypotheses. A few individuals were interested in further discussion. We conducted several Skype and conference calls with surveyors from as far as Germany. One of them was a leader of an organization from Chicago that had developed his own iOS app that assisted him in survey taking. While it wasn't near the extent that we are planning to develop, it was a good source of potential UI ideas.

One responder of particular importance was Thomas Naberhaus of Butterflies and Moths of North America (BAMONA). "The BAMONA project aims to serve as a one-stop database of butterfly and moth data that scientists can use to form or to address research questions. " BAMONA's information page provides more detailed information about the goals and background, which are very similar to the objectives of this project[1]. BAMONA currently has a web interface and database for hosting all records. Thomas proposed that we collaborate with him on the specification for the new system and develop the mobile apps to interface with BAMONA's database. We discussed this within our team and with our client to analyze the potential for success and vet potential risks of the collaboration. We determined that we would make the application with the capability to export to remote databases so that users will be able to export their data to BAMONA's database.

Protocols

Primarily through discussions with our clients, we reviewed the standards for existing survey protocols and determined the set of data fields we wanted to collect. We included this list of potential data points in the questionnaire in order to gauge market response and capture any fields we had overlooked. The list of fields we collect is available in the Detailed Design - Database section.

Existing Applications

By searching through the Google Play Store and talking with our client, we collected a list of mobile applications that perform related tasks. The full list is in Appendix C. Most of these suffered from the same problems as manual survey protocols: They collected a small set of information. This search verified that we would not be replicating existing technology with the development and release of our app.

[1] http://www.butterfliesandmoths.org/about

Functional Requirements

Android Application

Synopsis

Allows users to record all the relevant information about a butterfly sighting and its environment throughout the course of a survey. Once collected, the users will have the ability submit that information to a database or export it locally into CSV files.

List of Requirements

Must create a list of common butterfly species for numerous regions around the world. The application must support the presentation of butterflies that are indigenous to the particular region of the user. The user must also have the ability to submit anomalous sightings.

The following data points will be collected or calculated automatically by the app

- GPS
- waypoint of routes
- date and time
- light levels
- temperature
- windspeed and direction
- humidity
- cloud cover

The app will facilitate user input of the following data

- tally of sightings for each species
- habitat category
- habitat conditions
- data entry for mark recapture details
- categorized behavior notes: Mating, nectaring, basking, puddling, perching, patrolling
- site name
- surveyor name
- taxonomic tree-based classification
- miscellaneous comment section (animal life, plants present)
- differentiation of sections along route, distance/habitat category
- manual weather entry
- voucher photo capture

Web Export

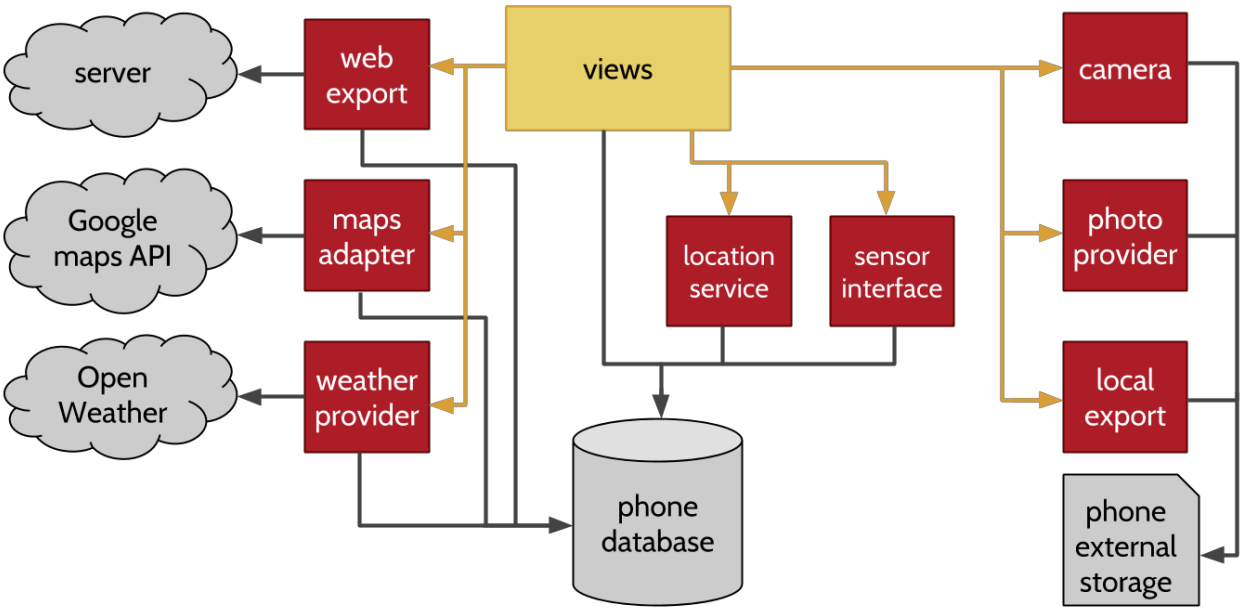
Allow submission of the survey and sightings data to a remote database.

Nonfunctional Requirements

- Performance: The battery must drain no more than 50% during a 4 hour survey on recommended settings with an average phone
- Ease of use: Users should require no training to record a sighting and perform a survey with the app
- Graceful failure: The app should exit cleanly in the face of exceptional conditions, saving user data prior to exit, and not causing the device to hang or restart
- Form factor adaptability: The app should function well on devices of all standard form factors
- Offline usability: The app should allow users to record data in the absence of a network connection
- Hardware adaptability: The app should still be usable on a device with a subset of the supported hardware features and sensors
- Minimal data transfer: The app should minimize the amount of network traffic necessary for submitting data to the server
- Network agnosticism: The app should submit data over wifi connections or cell networks
- Transactional data submissions: In the event of a lost network connection or otherwise failed data transmission, the app should retain all data locally and report a failed upload.
- Multi-task capability: The app should allow the user to move to another app in the middle of a survey or sighting and return without losing data
- Minimal resource usage: The app will minimize CPU and memory usage where possible

System Design

Architecture



System architecture. Our app architecture has three high-level categories of components: Views (UI, in yellow), data sources/storage (grey), and data interfaces (red).

Functional Decomposition

Views

These are represented by the yellow box in our diagram. These touch-centered interfaces are defined structurally in XML, with an associated Activity Java class. All functions of the UI are defined in callbacks that execute in response to system or user events (e.g. button press, swipe, system alarm). These display data using adapters that implement the Observer pattern to

asynchronously update the UI in response to underlying changes in the data.

Data Sources and Storage

Server

This is a write-only remote server to which UBR can export surveys over a network connection. In our current system, this is a Linux server hosted by ISU IT, running Apache and MySQL. We have designed UBR such that this can be replaced by or augmented with a third party remote database with a network interface.

Google Maps API

Google provides the Google Maps Android API v2 to allow android applications to include google maps. UBR uses the map to show the user's breadcrumb trail and sightings. Breadcrumbs are shown as a trail drawn with a polyline, and the sightings are shown with custom made markers. The color of these markers is determined from a hash of the butterflies species name and family.

OpenWeather

Open Weather provides web services through which applications can gain access to up to date weather information. Through their services, UBR is able to pull information such as the temperature, cloud cover, and wind speed. The weather services supports exporting the results of a query in a multitude of formats. UBR obtains its results in the form of a JSON message.

Phone Database

UBR uses two SQLite databases on the phone in its application-private storage. One contains solely user data with tables for surveys, sightings, and breadcrumbs. The other contains system information like the default list of butterflies. We introduce this separation as a way to hasten the creation of the butterfly tables within the database.

Phone External Storage

On most devices, this is an SD card, but implementation is device-specific. Any photos the user takes are stored here, as well as any data exports created by the user. Photos are on the SD card because of size limitations of internal storage, and exports are here to allow the user easy access to the files via any file manager application.

Interfaces to Data Sources and Storage

Web Export

Our implementation of this for our proof-of-concept server URL-encodes the survey and opens a connection with the server over HTTP to send the survey information.

Maps Adapter

Using the Google maps API to draw the map itself, this adapter reads the current survey's breadcrumbs to draw a path and sighting locations to show pinpoints on the map. This implements an Observer pattern for the database, so the map updates asynchronously.

Weather Provider

Upon beginning a survey, a task is launched to get the latest weather data from OpenWeather and write it directly to the survey table in the database. In case the device has no network connection, the provider simply does not collect data.

Location Service

Breadcrumbing in UBR is implemented as an Android Service, which is a long-running process not tied to a UI. This means that the user can exit UBR while doing a survey without cancelling location tracking. When beginning a survey, the view launches an Intent to connect to the location service, providing it a URI to the survey associated with the GPS points. If multiple surveys are running simultaneously, the location service receives the launching intent and builds a list of currently connect surveys in order to write the breadcrumbs to the table for each survey. When all surveys have disconnected from the service, again by Intent, the service exits.

Sensor Interface

UBR's environmental sensor interface requests instances of the device's default temperature, light, pressure, and humidity sensors. Of all those that are available, it registers listeners for device updates and write the first update of each to a given URI (a field of a sighting instance in the database), cancelling each listener after receiving the first value (as Android sensor interface does not support getting a single value). This request expires after a configurable number of seconds (set to 20 in UBR based on experimentation) to prevent battery draining the case of sensor failure. If after 20 seconds a sensor has not yet reported a value, the listener is cancelled.

Camera

UBR uses the device's default camera application via Intent to request that a photo be taken and stored to a given URI. In this way, the camera acts as an interface to external device storage.

Photo Provider

This component reads images into thumbnails from external storage for display in the application UI. Thumbnail size is configurable.

Local Export

Users may export any or all of the user-data tables to the device's external storage at the press of a button. This is implemented as background task on a different thread, and writes the tables as csv files to a timestamped folder in the apps external storage export directory..

Design Tradeoffs

Target Device and Operating System

Our client's original project proposal included apps for Android, iOS, and Windows Phone. As of November 2013, 43% of the world's smartphones run Android, compared to 21% of phones running iOS, and just 1.8% running Windows. While iOS has a better standing in the US, we are targeting a global market[2]. We decided to create our proof of concept in Android for this reason, bolstered by our team's greater experience with Android vs iOS. After analyzing the complexity of implementation on a single target device, we made the informed decision with our client to declare non-Android versions of the app out of scope for the project.

Within Android, we selected OS version 4.0+ (API 14+) as our target. Supporting Android versions older than 4.0 would require using special support libraries in order to achieve our current design. Estimates in May 2013 predicted that the proportion of devices running Android 4.0+ would rise by 15% to 70%. Based on this prediction and the added complexity of supporting older OSs, we officially supported Android 4.0 (API 14) and newer, mitigating the added complexity of supporting a rapidly shrinking portion of the market. At time of release 74.2% of phones are 4.0+[3].

Database Performance

One of UBR's objectives is to provide researchers with a standardized facility through which to collect and share information regarding butterflies. The application requires extensive internal data storage mechanisms to handle all the collected information. These mechanisms not only need to be able to reliably store the information but must also make it accessible to the application in a timely manner. This led to some interesting design decisions that needed to be made in regards to the performance of the application.

The application needed to store a list of all the butterfly species most common to many areas around the world. This list got to be quite large and was provided in a CSV format from our clients. We also needed to support the import of user defined CSV lists as well. Not only that, the butterflies added by the user needed to be segregated from those added by the developers. This segregation was desirable to allow developer controlled imports based on the particular region of the user. Thus, the users would be provided a pre-defined list of butterflies based on their location. As the provided CSV's contain hundreds of entries, the optimal method for quickly swapping regional butterfly lists would have been to drop a pre-built database onto the device.

[2] www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892
[3] developer.android.com/about/dashboards/index.html

The use of the pre-built database model would have resulted in performance problems for accessing the butterflies. In order to survive the upgrade, the table containing the user imported butterflies needed to be placed in a separate database. This separation of the tables into separate databases would have led to a significant impact on the applications performance. With the separation, generating the list of butterflies would require multiple SQL operations across the two databases. This kind of a design would not scale well as the introduction of larger lists was introduced.

In the end, we decided that we needed to find a design that would result in the optimal performance for the most common use case. For UBR, that case is reading the list of butterflies to be presented to the user. We decided to place all the butterflies into a single table within the database while also keeping track of their originating file. Having that single file optimizes the queries against the data while also providing a method by which to efficiently filter the butterflies shown to the user. This design does result in a the database creation and upgrade process taking a few extra seconds when a new version is installed. As this isn't common, we decided that the performance impact of the upgrades was acceptable.

Native vs Portable Implementation

In the spring we explored various cross-platform mobile development frameworks like PhoneGap, Appcelerator, and Titanium that allow creation of a single codebase for deploying to multiple platforms, as the original proposal discussed multiple platforms. Though our implementation is solely for Android, we include this discussion, as platform expansion is a near-term goal for our client. These have the disadvantage of having a common UI, which means that it can match the UI of at most one target platform, resulting in a mismatched experience for users on other platforms. These solutions also present difficulties when working with device sensors, which is a requirement for our implementation. These concerns disqualified such platforms from our design. We then moved on to exploring cross platform design patterns for the possibility of having a shared codebase written in C++ for the core application logic for multiple platforms. Because of the UI-heavy nature of our design, we decided that this would require *more* work in the end, regardless of the re-implementation time of this core logic, given that the ratio of this code base to the full application would be fairly small.

Detailed Design

Database

Schema

Breadcrumb Table

- key - ID
- Columns
  - Survey_ID → Links back to ID in Survey Table
  - Time - datetime
  - Latitude - GPS
  - Longitude - GPS
  - Accuracy - Accuracy of location reading
  - Speed - Speed device is moving at time of reading

Butterfly Table

- key - ID
- Columns
  - Generic_Name
  - Scientific_Name
  - Generic_Family
  - Scientific_Family
  - Generic_Sub_Family
  - Scientific_Sub_Family
  - region
  - Frequent
  - Custom_List
  - Source

Sighting Table

- key - ID
- Columns
  - Survey_ID → Links back to ID within Survey Table
  - Generic_Name
  - Scientific_Name

- Generic_Family
- Scientific_Family
- Generic_Sub_Family
- Scientific_Sub_Family
- Location
- Number
- Temperature
- Wind_Speed
- Wind_Direction
- Cloud_Cover
- Time - Time of the sighting
- Behavior - Activity that butterfly when seen
- Gender
- Condition
- Comment
- Transect
- Photo_Name → Links to the associated picture on device
- Pressure
- Ambient_Temperature
- Illuminance
- Relative_Humidity
- Wing_Length
- Mark_Found
- Mark_Added

Survey Table

- key - SurveyID
- Columns
  - ID → Links into Breadcrumbs and Sightings tables
  - Survey_Type
  - Name
  - Start_Time
  - End_Time
  - Surveyor_Count
  - Surveyor_Names
  - Location_Name
  - Comment
  - Habitat_Type
  - Habitat_Condition
  - Wind_Speed
  - Cloud_cover
  - Temperature
  - Engagement_Level
  - Radius
  - Uploaded
  - Service_Temperature
  - Service_Humidity

- Service_WindDirection
- Service_WindSpeed
- Service_Time
- Service_Pressure
- Service_Sunrise
- Service_Sunset
- Service_City - Location of weather station
- Service_Country - Location of weather station
- Service_MaxTemperature
- Service_MinTemperature
- Service_CloudCover
- Service_Rain
- Service_Snow

Transect Table

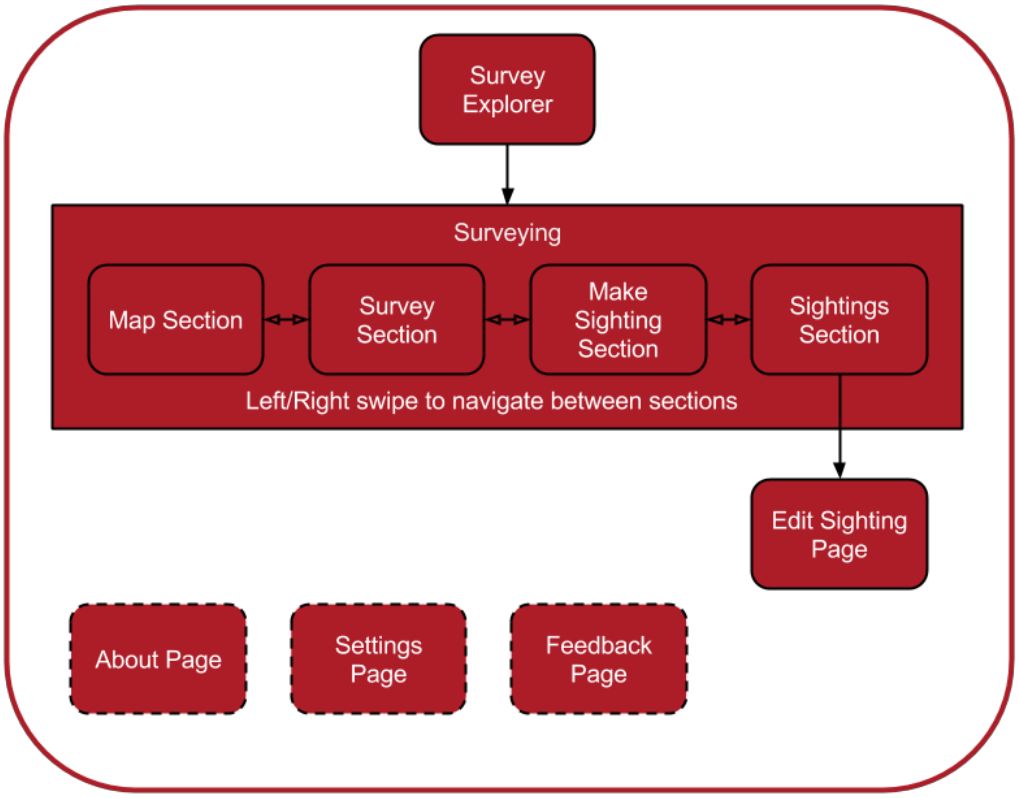
- key - ID
- Columns
  - Name
  - Description

Application User Interface

Target Audience

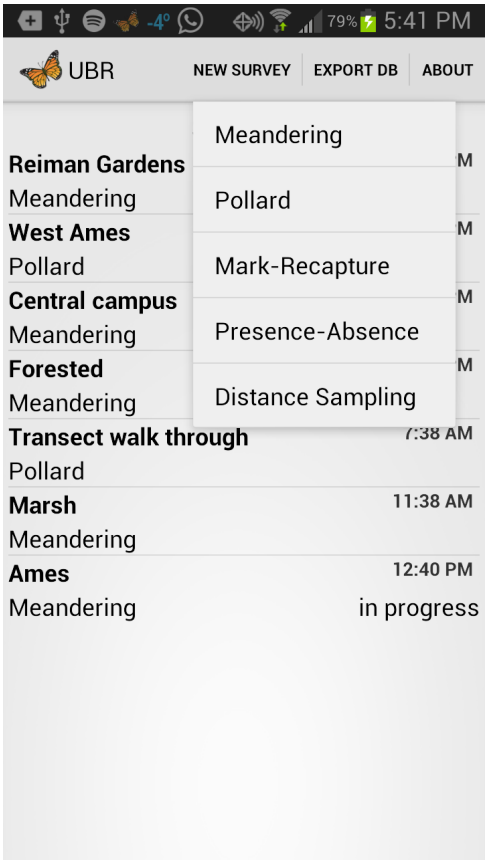
There are several kinds of people that currently perform butterfly surveys: Researchers, graduate students, conservation workers, citizen scientists, hobbyists, and volunteers with minimal experience. We will design this app to be easily understood upon first use, and fluidly functional for each of these users. The app has been released on the Google Play Store for anyone to use for free. The UI also has to be designed to adapt to varying screen sizes.

Screen Flow Diagram



Page Descriptions

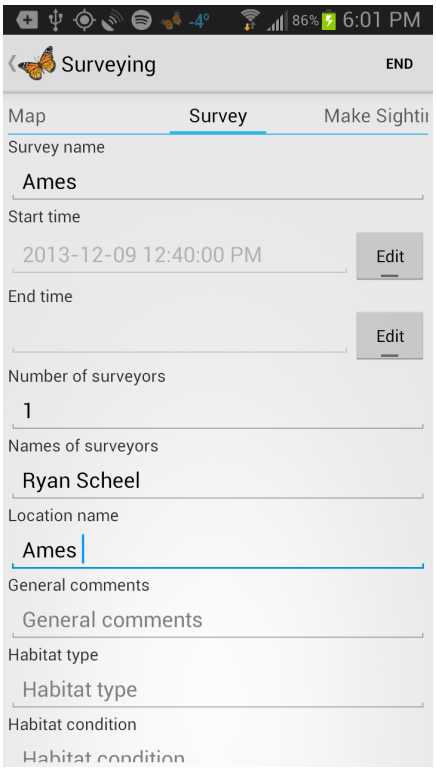
Survey Explorer



The start screen for UBR is a vertical list that contains a catalog of all surveys performed on the device. Users may view and update any record stored on the device by selecting the record to return to the surveying page. Users can also export all the data they've recorded using several different methods. They can export to local files, to our server, or email the data to themselves. Finally, the survey explorer's topmost option will be to create a new survey, This will take them to the surveying page.

Surveying

The surveying page will look a bit different for each protocol, to accommodate the different styles of collection. For each sighting and survey, no matter the protocol, the app will collect the same set of automatic data, as delineated in the requirements-to-deliverables mapping in the project plan. This feature of the system is to promote standardization of data collection without burdening users with the requirement of extra manual recording. In addition to the benefit of automatically collecting a larger amount of data than any one protocol, this automation will speed up the process of surveying for all users. Users will be able to include data outside the scope of the current protocol via an unobtrusive panel. The surveying page is organized into four sections, the user can swipe back and forth between these sections.



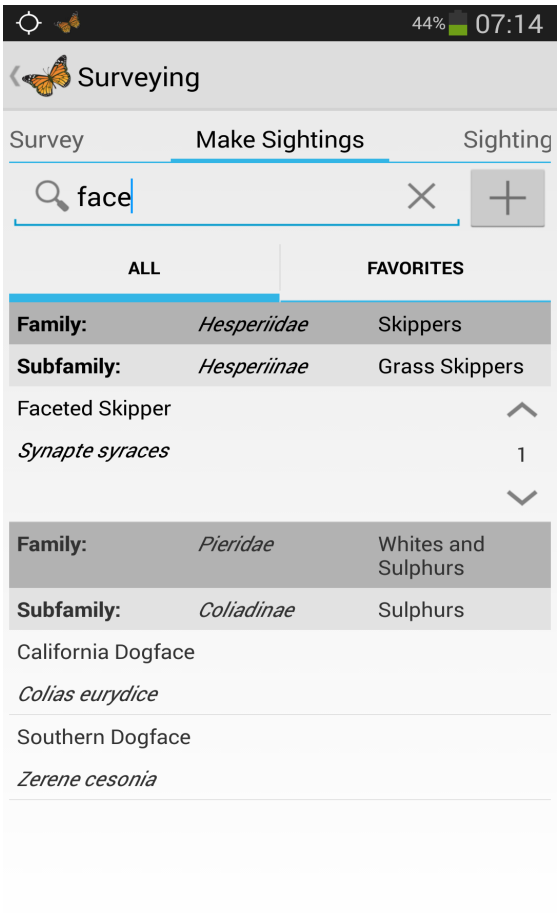
Section 1 - Survey

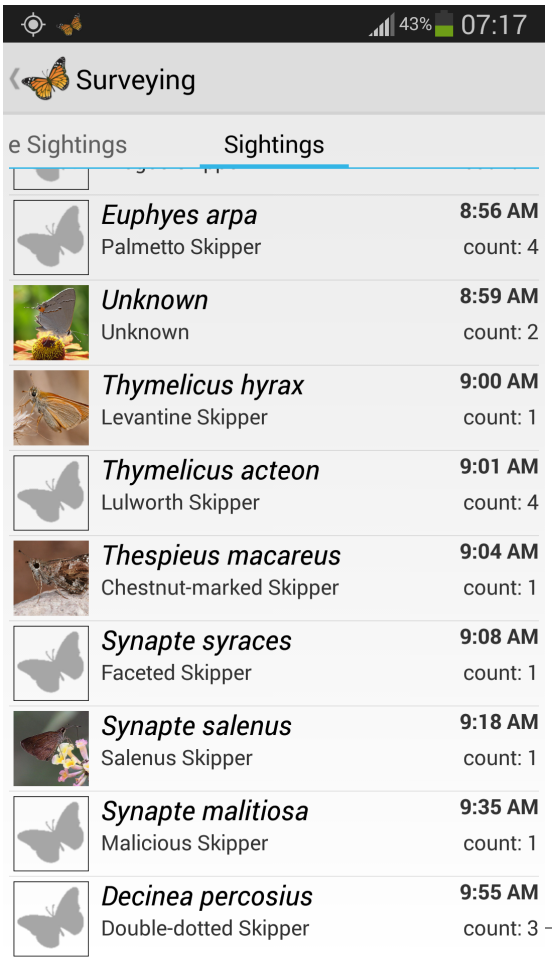
A survey consists of many sighting and some common data. The user can enter the common data at any point during a sighting. Pressing the Begin button in the Survey section automatically records a start time and pushing End will record the end time. These values may be manually edited. Common data also includes route waypoints, which are automatically plotted during the survey after the begin button is pushed, at an interval configurable in the app settings, defaulting to

every 30 seconds.

Section 2 - Make Sightings

Scrolling to the right from the Survey section will bring the user to the Make Sighting section. This page is a vertical list of butterflies, organized taxonomically and alphabetically. UBR comes with a comprehensive list of butterflies found in different areas in the world, but we also offer the option for a user to import their own custom lists. We provide a search mechanism and the ability to create a Favorites list to allow easier navigation of this lists. When a user sees a butterfly they can simply click it on the list and it will be recorded, along with gps, timestamp, and other auto filled data. Butterflies are frequently seen in groups so we also provide a counter when they select a butterfly.



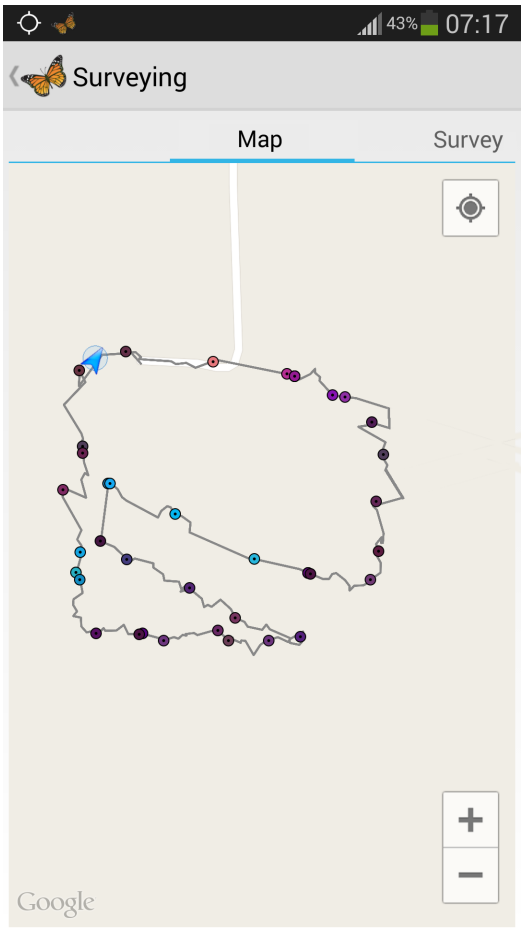


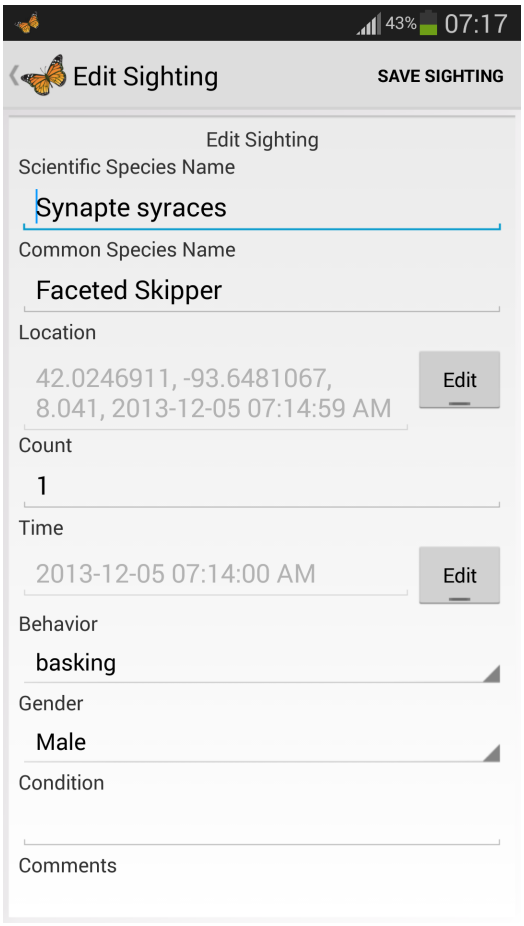
Section 3 - Sightings

Scrolling to the right again will bring the user to the Sightings section. Here the user is shown a list of all the sightings they have made while on this survey. They will also be able to see the thumbnails for any voucher photos the may have taken, and can select any sighting on the list to go into a more detailed Edit Sighting page.

Section 4 - Map

The leftmost section is the Map section. Here UBR makes use of Google Maps to show the user the route generated from the breadcrumbing, and all the sightings the user has taken along the way. The color of the sightings is actually generated using a hash of the butterfly name and family name. The family name determines the most significant bits of the color. This causes butterflies within the same family to appear similar in color.





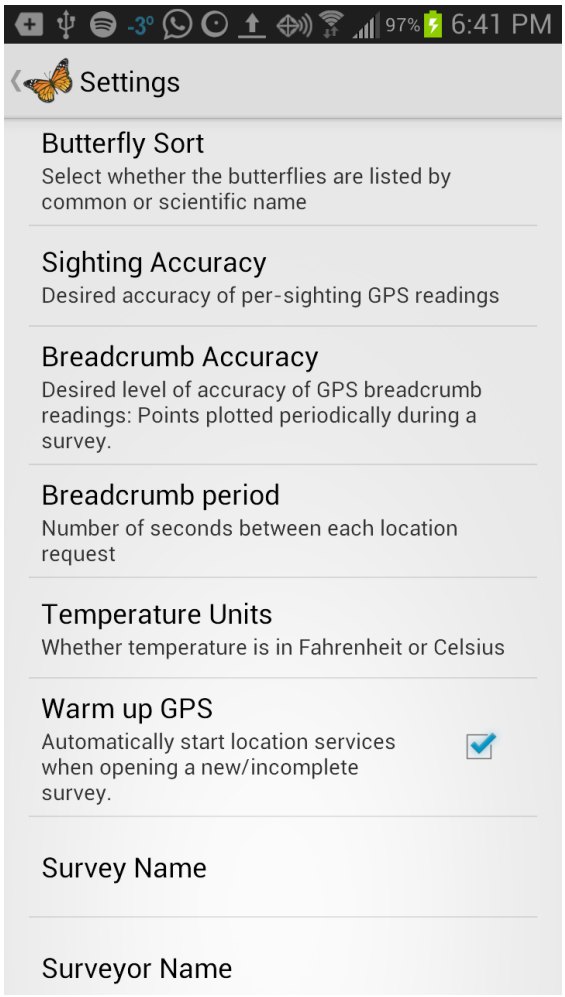
Edit Sighting Page

The user navigates to this page from the Sighting section above. Within this page the user can make more detailed observations about a specific sighting. They can also take a voucher photo, this is necessary in some instances to provide definitive proof of a sighting. Useful when a researcher sights a relatively rare butterfly in a new area.

Settings

The settings page is accessible from anywhere in the app via the android menu. UBR allows users to specify a number of settings:

- Default survey protocol
- Automatic upload preferences
- GPS accuracy
- Breadcrumb interval
- Preferred measurement units
- Local export directory
- Transect names
- Other miscellaneous settings



About Page

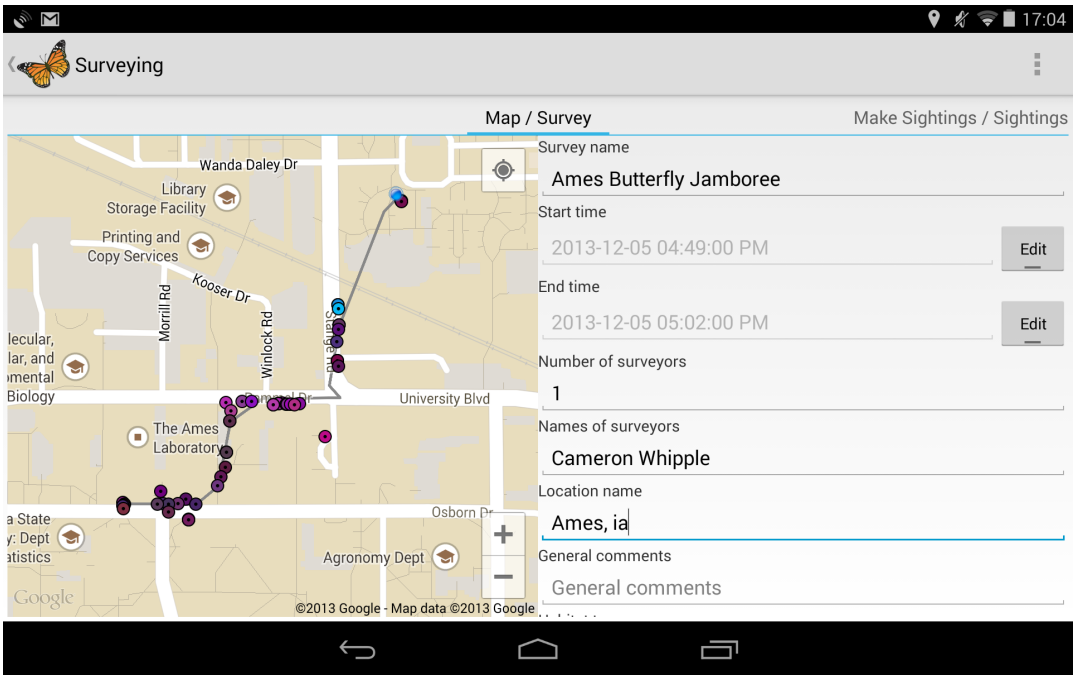
The about page is a simple page containing legal disclaimers and information about the project. It also contains information about the current version number.

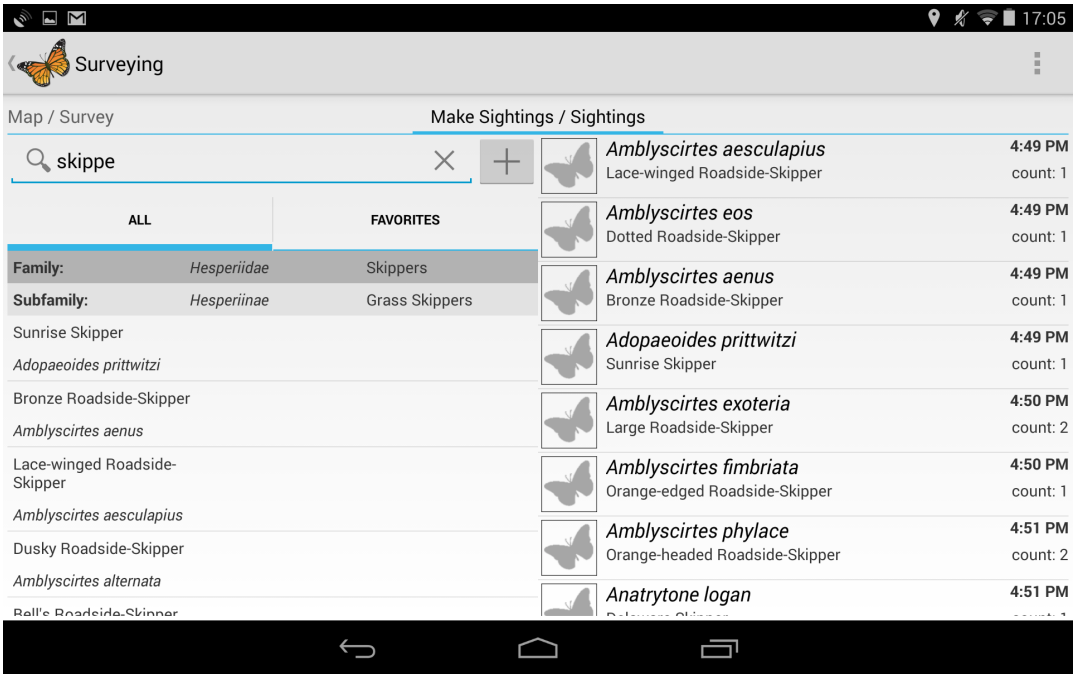
User feedback

For the prototype release, the app contains a page for submitting user feedback, accessible using the app's context menu. This page contains fields for providing bug reports and general comments. Each submission will have the option of including usage statistics.

Tablet Layouts

Tablets have much larger screens than phones, as a result simply stretching the phone UI on tablets would look unprofessional and wasteful. Our solution was to display two sections from the surveying page at a time. You can see this in the screenshots below.





Remote Export Support

Third Party Databases

To better enable remote sharing of data, our clients would like to ultimately create partnerships with existing butterfly data warehouses around the world to which UBR could export. BAMONA (Butterflies and Moths of North America) contacted us in the spring about supporting export from UBR. Shifting timelines for their development team have pushed this out of scope for our senior design team, but it remains as a potential future feature of UBR. To this end, we have designed UBR to be able to be able to export its collected data to a third-party database, the implementation of which just needs to follow our defined internal export interface. Another current butterfly record keeper, eButterfly, has expressed interest in supported export from UBR as well. This implementation would likely be done by a future senior design team on a related proposal by our clients from Reiman Gardens.

Proof-of-Concept Server

Through ISU IT, we set up a server running Redhat Linux. Beyond hosting our project website using Apache, we designed this server to accept survey uploads from UBR as a proof of concept for our remote export support. Any surveys sent to our PHP endpoint at butterflies.ece.iastate.edu/submit-survey.php will be inserted into a MySQL database. This is implemented client-side through a simple abstracted interface to demonstrate the ability to support export to third party databases.

Testing

Testing

Prototype

We deployed the minimum viable product of UBR in July, which allowed users to perform surveys using the Meandering protocol, with only time and location information collection for sightings. We released this using the alpha release feature of the Google Play store. This tool lets us restrict access to the Play store app using membership of a Google Group (ubralpha@googlegroups.com). Through Nathan, we invited all those who took our questionnaire and a few others to join the testing group and use UBR. Using this as our release system, our updates (and bugfixes) to the alpha release were automatically received by our users, making the release cycle simpler and more effective.

Scenario Validation

As we developed UBR using two-week sprints, we released new features and fixes every two weeks. Before each release, we performed scenario validation to detect regressions and test overall usability. Our scenario validation involved performing simulated surveys, using each feature of the app. By doing this, we were able to catch almost all regressions before deployment. Other issues (such as those specific to hardware our team doesn't have) were quickly reported by our client or user base.

We chose scenario validation as our primary means of testing due to the ever-changing nature of our project's requirements. We had a constant dialog open between our team, our client, and our other users about UBR's feature and usability. This meant that our requirements changed with each sprint, making automated testing infeasible.

Field testing

Reiman Gardens staff have used UBR extensively in the field during actual surveys. Our team has also accompanied conservation professionals on surveys in Ames, as well as Seattle, Washington. Using feedback from this, we have iteratively refined the process of performing a survey, minimizing the number of presses required for each sighting and improving the quality and

usefulness of GPS and other collected data.

Battery Usage

As battery is a major concern on mobile devices, we tested the drain of our most battery-intensive feature, breadcrumbing, using the highest accuracy and various collection intervals. This is compared to a baseline drain of 2% per hour without UBR running. Data was collected on a Samsung Galaxy S4 running Android 4.3.

Sampling Period (seconds)	battery drain per hour
5	7%
10	5%
15	4%
30	3%
60	2%

Usability Improvements Over Existing Methods

UBR speeds up the process of recorded basic preliminary survey information by three times. After the one-time process of creating a list of "favorite" butterflies in the app, recording of sightings during a survey shows more dramatic improvements. With a single touch of a butterfly in the list, UBR records common name, scientific species, ambient temperature, pressure, humidity, illuminance, GPS location, and a timestamp. If a user enters more manual information, behavior, sex, condition, and a photo can be recorded in just 30 more seconds.

Project Management

Standards

UBR aims to support all major survey protocols (for a listing, see the Background section). We have followed industry criteria for these, ensuring that we can collect a superset of the data required by each protocol.

Schedule and Timeline

Schedule

The first few months after receiving the project was spent discussing the project with our clients from Reiman Gardens and contacting butterfly professionals from around the world. Because our project came with an end goal but not specific requirements, a lot of time was spent solidifying the basic requirements. Towards the end of the first semester we started programming the prototype to be tested over the summer. The prototype with one survey protocol was completed by the end of July and real world users started the using the application and sent feedback with bug reports, missing functionality, and usability suggestions.

At the beginning of fall semester we met with our client to go over the feedback and see what they thought was most important to be added to or fixed in the application. The first few months of the semester we focused on adding major features which were completed by mid-September. By mid-October, the local database was determined. The next month was spent implementing the other survey protocols. From then to the end of the semester, most time was spent on finding and fixing bugs, adding minor functionality, and creating the poster, presentation, and final document.

Timeline of Major Milestones

Target Date	Milestone
Mar 1	Survey responses aggregated & requirements set
Apr 1	Divide tasks among team members; begin prototype
May 10	Finish Android application skeleton
July 31	Complete Android prototype
Aug 15	Review application testing and feedback
Sept 20	Major feature expansion complete
Oct 18	Butterfly database completed
Nov 15	All survey protocols implemented
Dec 9	Poster and presentation finalized

Work Breakdown

Team Contribution

We worked as a team to create our user requirements, working with our client and advisor. We all helping with the development of the application while focusing on our specific management positions. This allowed each member of the team to become an expert in different areas so we could polish the application as much as possible.

Component Managers

The component manager delegates tasks as necessary to meet our objectives by our target deadlines for their component. The component manager also became an expert in their area as to provide the best finished product.

Manager	Component
Ryan	User interface
Curtis	Documents, website, GPS
Cameron	Databases
Julie	Communication, presentation, and misc.

Risks and Mitigation

Loss of a Team Member

There is always the possibility that a team member will be unable to continue his or her progress on the project. To mitigate this risk, we documented each component of the project and all interfaces so that they can be developed in effective isolation if necessary. Thankfully, we did not encounter this problem.

Possible Future Loss of Third Party Service

Our design of interfaces for external services (weather) will allow easy swapping of new services to conform to our existing usage of this data in the app.

Identification of Bottlenecks

We performed field testing, including user feedback built into the alpha/beta release of the application. We identified the most used and essential features and optimized those processes for maximum efficiency.

Communication and Outreach

Because our project has the goal of widespread adoption in the conservation community, we have made extra effort to effectively communicate with our global audience. We set up an Atom news feed on our website, through which we posted updates after each sprint. This enabled

those interested to receive push updates of UBR progress in their news reader. As discussed in section (...) we deployed our alpha release and bi-weekly updates through the Google Play store, to allow easy access for all Android users. Because UBR is installed through the app store, users are presented with the option to submit a report after a crash or "application not responding" error (ANR). Through our developer console, we can view the stack traces of the errors as well as device information. In the app itself we included a page through which users can submit feedback in the categories of bug, feature request, and comment. These are collected on our server. We had several users communicate with use via our listserv, butterflies@iastate.edu with comments, questions, feature requests, or bug reports. Several more communicated directly with our client, who informed us up feature requests at each post-sprint meeting.

Outreach

Our team presented to 300 high schoolers during the inaugural IT Adventures program at ISU. We discussed what it takes to develop an Android app and provided perspective on how engineering can apply to other fields like conservation.

We also presented at the ECpE department's Open Lab Night, in which approximately 100 students (primarily freshmen) learned about UBR and senior design in general.

Summary

Acknowledgments

Our team would like to thank our advisor, Dr. Diane Rover, for all of her input. She's had many senior design teams in the past and the quality of the advice she gave reflects that. We really appreciated that she was honest with us and was straightforward about ways that we could improve. She was flexible in meeting times, was always willing to look over our documents/presentations/poster outside of scheduled meeting times, came to our practice run throughs of presentations, and showed her support by attending the actual presentations.

We would also like to thank our clients, Nathan Brockman and Anita Westphal, from Reiman Gardens. Both Nathan and Anita were always very excited about the project. They gave us incredible help by getting us in touch with butterfly researchers from around the world and promoting the application at every conference they attended. They understood that we knew nothing about butterflies, which was appreciated, and also understood that they didn't know much about programming. The made our partnership flow smoothly as we were able to rely on each other's strengths and expertise.

Conclusion

The goal of the application was to streamline and standardize the data collected in butterfly surveys. We wanted a polished product that would be used by anyone doing a butterfly survey, from civilian scientists to professionals, no matter what survey protocol they were accustomed to using. We have gotten users from around the world to test the app and believe they plan to continue using it. There are also different people around the United States using the app that typically use different protocols.

Our clients and advisor are very pleased with UBR's feature set and usability. There will always be more features that can be added, but the app is feature-rich that we have released it publicly on the Google Play Store[4] and shared it for use by professionals for collection of data.

[4] https://play.google.com/store/apps/details?id=edu.iastate.ece.butterflies

Impact

Because our application standardizes what data is collected, data from all around the world can now be compared. Also, because it is able to export to external databases, users will be able to export their data to a common database for even easier collaboration. Conservation efforts will be helped immensely because of this. The application also makes going into the field and taking a survey easier so people may be willing to take more surveys. With more surveys being taken there will be more data for conservation groups to use.

Future Work

Resources

As this application is being released to the public and will be used in the future, maintainability has been a concern. Clear and precise documentation has been kept. All code has been pushed to a Github repository throughout development. The history in that repository could be a major help for any future developers as well. In the very near future, the team plans on being available to help with any bugs or minor tweaks that the application requires. Any major features to be added or system expansion will need to be done by other developers. Our clients have already submitted a proposal for another senior design team to take over our position and continue working towards the streamlining and standardizing of all butterfly data.

Improvements

Export to External Database

UBR has the capability to export to external databases. In order for the export to work, the app must be extended to interact with the services hosting that database. Extending the application can be performed quickly once an external database has been identified as UBR has been designed to support this functionality.

Additional Functionality

iOS Application

We have found some butterfly researchers who would like our application to run on iOS devices. Creating an iOS version of the application would increase the number of people who can use the application and contribute to the standardization and streamlining of data.

Appendix A: User Manual

Getting the App

Google Play Store

Unified Butterfly Recorder (UBR) can be downloaded from the Google Play Store at this link: https://play.google.com/store/apps/details?id=edu.iastate.ece.butterflies

Basic Surveying

Creating a New Survey

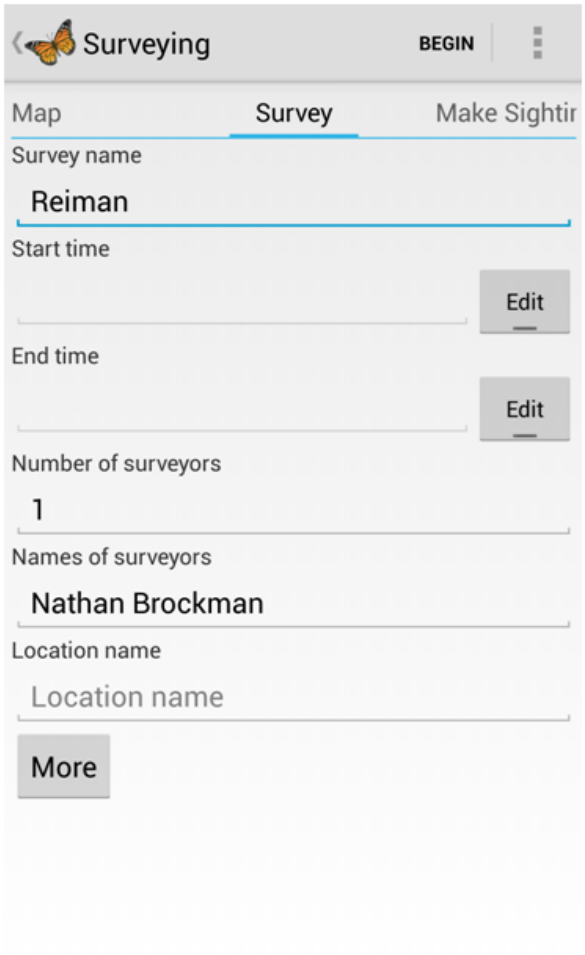
When UBR is first opened, it will begin on the Survey Explorer page. If you have not taken a survey yet, the list of surveys will be blank. Select "New Survey" at the top of the screen and choose your survey type from the drop down menu.

Presence Absence Protocol

If you selected Presence Absence, you will not have the following survey page. Skip to the "Taking a Survey" section.

Other Protocols

For all other protocols, you will be brought to the Survey page. Fill in as much of the information about the survey as you would like. Click the "More" button for more settings that can be set.



- Survey name: What the survey will be saved as (could be a name, route on a location, etc.).
- Start/end time: Recorded automatically when you start and stop the survey, based on your system time. You can edit these times later if need be.
- (For Pollard only) Transect division format: How transects are distinguished. Is each habitat a transect or is every x number of meters a transect.
- Viewing radius (meters): Maximum distance from a surveyor that an individual will be counted. This distance can be infinite if you want.
- Location name: Name of the location that you are surveying.

Select "More" for the following settings:

- Number of surveyors: How many people are participating in the survey.
- Names of surveyors: Name of each of the participants.
- General comments: Any other information you would like recorded.
- Habitat type: Description of the habitat being surveyed (i.e. forest, prairie, etc).
- Wind speed: Current wind speed.
  - calm: 0 mph (smoke rises vertically)
  - relatively still: 1-7 mph (wind felt on face)
  - moderately windy: 8-17 mph (leaves and small branches move)
  - windy: 18+ mph (small trees or large branches sway)
- Cloud cover: Current sky conditions.
  - clear: no clouds
  - mostly clear: less than half cloud cover
  - mostly cloudy: more than half cloud cover
  - cloudy: full cloud cover
- Temperature: Current temperature. The default units (Fahrenheit or Celsius) can be changed in Settings.

- Level of engagement: The amount of attention given to the survey.
  - 5: You are out just for the survey. No distractions.
  - 4: Your main reason for being out is to survey, but there are some distractions (i.e. you are talking to someone).
  - 3: You are doing a survey alongside another activity (i.e. walking a dog).
  - 2: You are mainly out for something other than the survey and are recording some butterflies along the way.
  - 1: You are out for reasons other than surveying and happened to see a few interesting butterflies you want to record.
- All settings below this point are weather information that can be recorded automatically from a station near you. Press "Refresh" to update this information.

Adding Sightings

Presence Absence Protocol

Once you have selected Presence Absence, you will be brought to the Make Sightings page. Making a sighting is the same as described below except once you have selected a butterfly species you will automatically be brought to the Edit Sighting page, which is described later on.

Other Protocols

Once you have entered all the information about the survey that you want to record, slide one screen to the right (swipe from the right side of the screen to the left) to get to the Make Sightings tab. You can always return to the Survey page by sliding back to the left to edit or add information.

On the Make Sightings tab you can use the search bar at the top to find a butterfly species, or look through the list of species. You can choose whether species are listed in order of common name or scientific name in Settings.

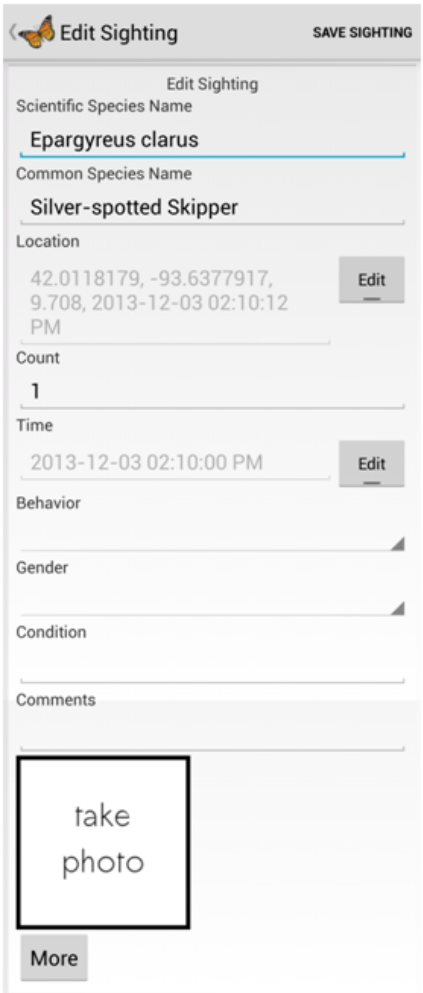
Use the tabs below the search bar to switch between the list of all butterflies and your Favorites list. You can add species to your Favorites list by tapping and holding on the

species name in the list and selecting "Add to Favorites" from the menu that appears. Favorites lists typically are used for butterflies that are seen frequently.

Tap a species in the list to record a sighting. Subsequent taps on the same species will record additional sightings. You can also use the arrows that appear to the right to add or subtract sightings. Multiple clicks (or incrementing the counter using the the arrows) will be saved as a single record of multiple sightings. You do not have to wait for the number to disappear before recording another species; it will disappear and be recorded as soon as you tap the next species or the timer runs out, which ever occurs first. When a sighting is added this way, time is automatically recorded based on your system's time. GPS coordinates of the sighting are also recorded automatically.

If you cannot identify a butterfly there are Unknown species you can select instead. There is a general Unknown, as well as an Unknown for each family and subfamily if you can narrow it down.

You can use the + button next to the search bar to add a new species to your list. You can also edit species information by tapping and holding on i t and selecting "Edit Species". You can import species lists in CSV format by tapping the button in the top right and selecting "Import List".



(For Pollard only) The transect button at the top lets you determine which section or habitat on your route the sightings were recorded in. When moving to a new transect, make sure to update what transect you are in.

Editing Sightings

Presence Absence Protocol

Because Presence Absence records a sighting of one species of butterfly, you will be brought directly to the Edit Sighting page and will not see the list of sightings. The Edit Sightings page is the same as the other protocols.

Other Protocols

Sliding one more screen to the right will bring you to the Sightings tab. This screen shows you all of the sightings you have recorded.

Tap a sighting to edit it. Tap and hold to either edit or delete a sighting.

Editing allows you to change the species name, the number of individuals seen, as well as adding extra information. You can note interesting or unusual behavior, gender of a butterfly, or the condition it is in, such as what wing damage it has.

You can tap the "Take Photo" button to take a picture of the butterfly with your device's camera. This photo will be displayed next to this sighting on the Sightings tab.

Tapping the "More" button allows you to edit the family and subfamily information of the butterfly. Tap "Save Sighting" when you have finished editing to save your changes.

Map View

The tab farthest to the left (slide to the left three times if on the Sightings tab) is the Map tab (available for all protocols except Presence Absence). This displays a map of the world with the GPS data of your survey laid out on it.

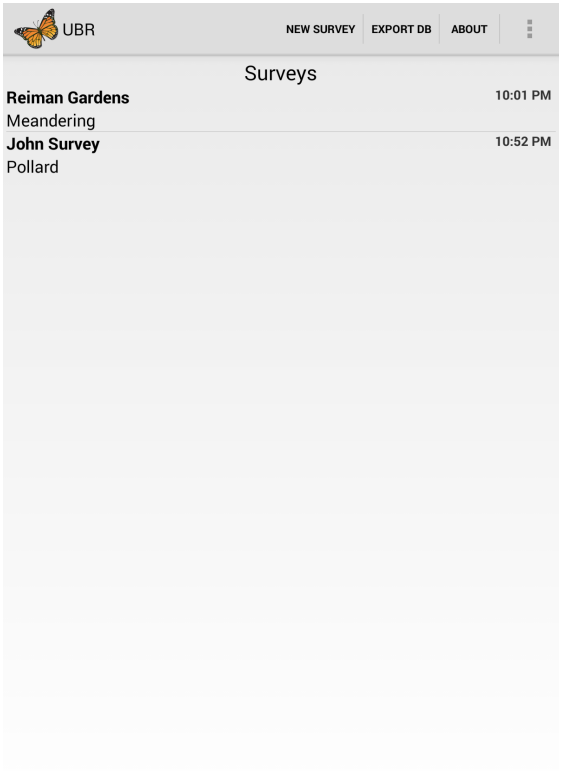
Press the crosshair button in the top left to zoom in on your survey location. The grey line is your breadcrumb data, i.e. the general route you took during the survey. The dots are the locations of each sighting you have recorded. A blue dot with a circle around it marks your current location.

To return to the survey tab, tap its name at the top.

Ending a Survey

For all protocols except Presence Absence, surveys must be ended. When you have finished your survey, navigate back to the Survey tab and tap "End" at the top. This will stop the recording of GPS breadcrumbs and automatically record your end time. You can go back and change information any time if you need to.

Options for Completed Surveys



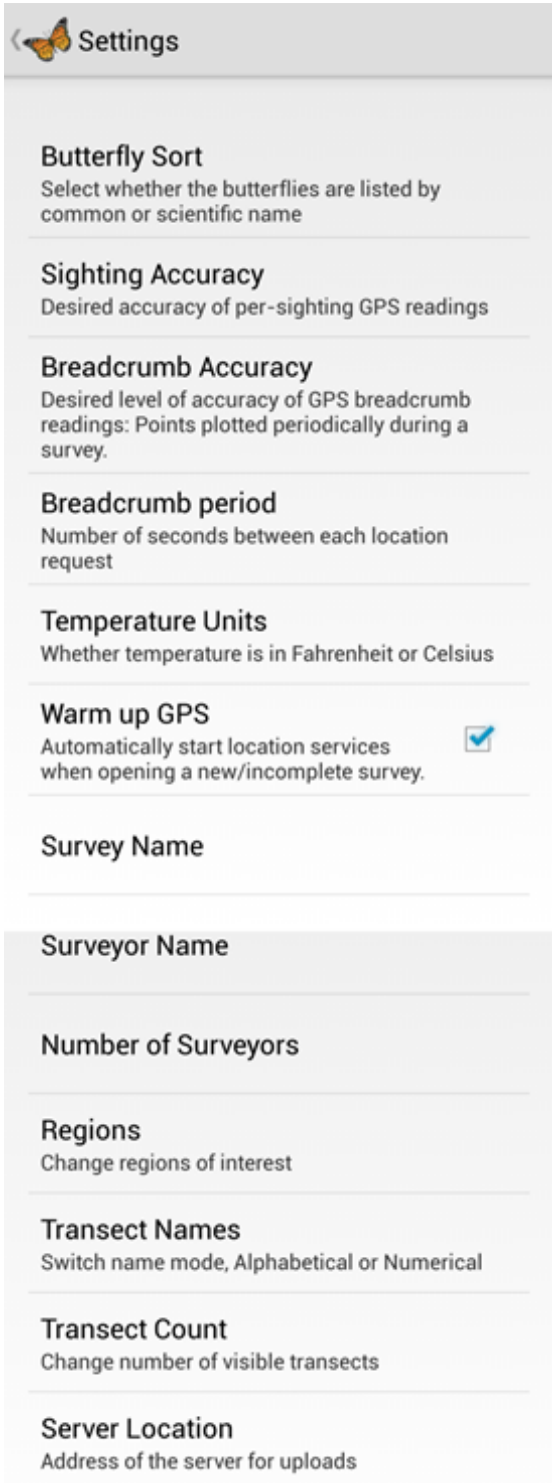
When UBR is opened, the Survey Explorer is opened. If in a survey or after finishing a survey you can get back to the Survey Explorer by clicking Surveying in the top left corner. If you return to the Survey Explorer while a survey is still running, it will not end the survey and you can click on that survey to go back to it and continue taking sightings or end the survey.

Tapping and holding on a survey will bring up a menu with three options:

- Summary: Gives a list of each species recorded and how many individuals of each were seen.
- Upload: Uploads the survey data to a selected server (see Settings on next page).
- Delete: Delete the selected survey.

Tap "Export DB" at the top to save all survey data to your device's storage. That data is saved as six files:

- breadcrumbs.csv: Breadcrumb GPS data
- imported)butterflies.csv: Butterfly species data from imported lists or that were added in the application
- regional_butterflies.csv: Butterfly species data from default regional lists
- sightings.csv: Data from recorded sightings
- surveys.csv: Information recorded about the surveys (date, time, type of survey, etc.)
- transects.csv: Information about the transects used



Settings

To change program settings, tap the menu button on your device and select Settings.

- Butterfly Sort: Determines whether the species list is sorted by common name or scientific name. Set to scientific name by default.
- Sighting Accuracy: Determines the accuracy of GPS coordinates recorded per sighting. Default is High.
  - Low: Coordinates accurate to within 500m; will wait 20 seconds for an accurate reading.
  - Medium: Coordinates accurate to within 100m; will wait 30 seconds for an accurate reading.
  - High: Coordinates accurate to within 10m; will wait 40 seconds for an accurate reading.
- Breadcrumb Period: Determines the interval (in seconds) between recording each breadcrumb coordinate. Default is 60.
- Temperature Units: Determines whether temperature is recorded in Fahrenheit or Celsius. Default is Fahrenheit.
- Warmup GPS: If turned on, will begin tracking your coordinates before the survey has started to imporve initial accuracy. Default is On.
- Survey Name, Surveyor Name, and Number of Surveyors: Determine the default information for each survey.
- Regions: Selects which species list(s) you would like to use in your survey.
- Transect Names: Determines whether transects are names as letters or numbers. Default is Alphabetical.
- Transect Count: Number of transects you can select. Default is 5.
- Server Location: URL entered here determines the server that survey data is uploaded to when you choose to upload a survey from the survey list (must be a server that UBR can

communincate with).

Appendix B: Questionnaire

Curtis Ullerich

Edit this form

17 responses

Summary

See complete responses Publish analytics

Name

[redacted]

Email

[redacted]

Organization

[redacted]

How often do you perform a Pollard survey?

6-8 times per year couple times a year 8 times per year about 6 times/year 2 12 plus times per year never Never never Coordinating one Daily to several times per week 1/week never every week At least 6 times per year 6-8 times per year 8 to 14 per year

How often do you perform a Mark Recapture survey?

Never never Never occasionally 0 Have used mark-recapture once never Never never Performed 2 Daily to several times per week in particular years never never never 1 time per year (unrelated to the Butterfly Monitoring Networks) never rarley

How often do you perform a Distance Sampling survey?

Not sure what Distance Sampling is never Never rarely 2 never never Never never never Frequently during April/May and Sept. (observing seasonal migrants) never never never never never never

How often do you perform a Presence Absence survey?

Not sure what Presence Absence survey is couple times a year 4 times per year 10+ times per year 3 1-2 times per year 3-4 days a week Never never 1-2 times a year Continuously (online citizen science reports) 2/year never every year Rarely never rarley

How often do you perform a Meandering survey?

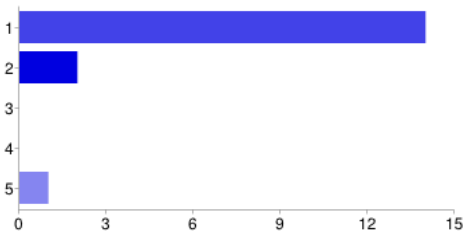
Not sure what Meandering survey is couple times a year Never about 6 times per year 3 6 plus times per year never 40-50 times per year multiple times a week in season never never weekly never never Rarely never ~3 times a year

If you conduct any other types of surveys, please briefly describe the method and list how often you perform them.

I survey miscellaneous sites around Iowa for butterflies using a random walking survey designed to record species presence and abundance at that single point in time. I record species as I encounter them during the walk using a pen in my field notebook, then estimate the quantity of each species seen during the walk when the walk is complete. This may be the same as a Meandering survey, I had just never heard that term before. I visit most sites once a season, and some two or three times a season. Visiting multiple times during a season and in multiple seasons gives a sense of butterfly population trends over time, but that is not the main objective my surveys. I collect moths about twice a week at a blacklight. Will probably do more butterflies with the right apps. An app would make it easier to track pictures and specimens. Rarely conduct a transect survey, but do not believe they are efficient.

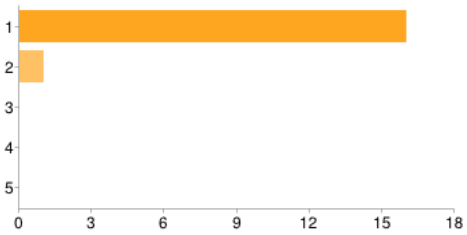
Which types of data would be useful to collect? [GPS coordinates of sighting points]

1	**14**	82%
2	**2**	12%



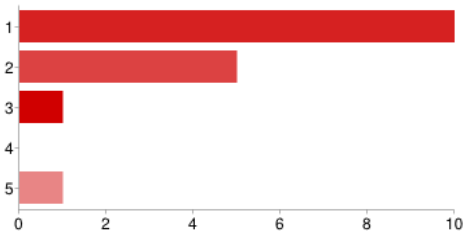
3	**0**	0%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [date of sightings]



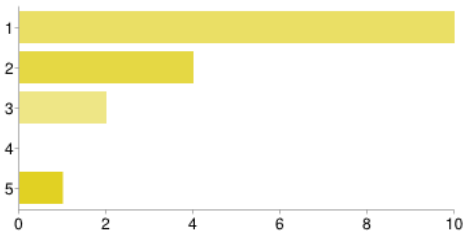
1	**16**	94%
2	**1**	6%
3	**0**	0%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [time of sightings]



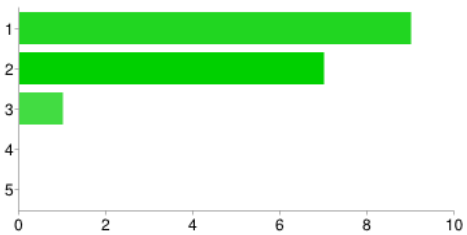
1	**10**	59%
2	**5**	29%
3	**1**	6%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [GPS waypoints of routes]



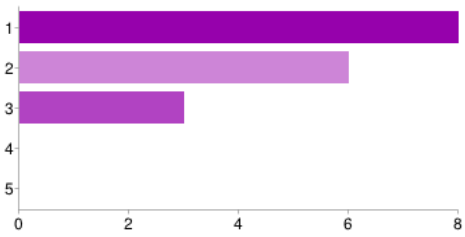
1	**10**	59%
2	**4**	24%
3	**2**	12%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [temperature]



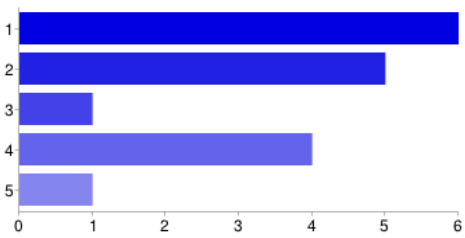
1	**9**	53%
2	**7**	41%
3	**1**	6%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [wind speed]



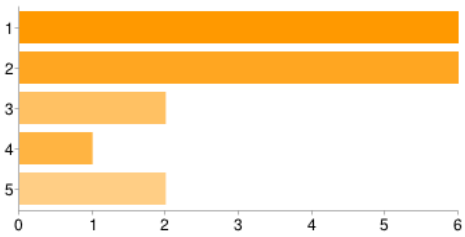
1	**8**	47%
2	**6**	35%
3	**3**	18%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [wind direction]



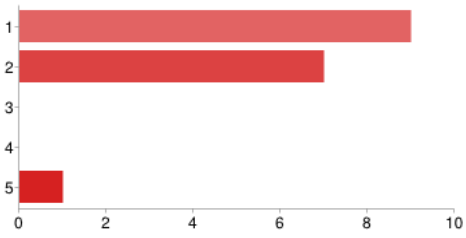
1	**6**	35%
2	**5**	29%
3	**1**	6%
4	**4**	24%
5	**1**	6%

Which types of data would be useful to collect? [light levels]



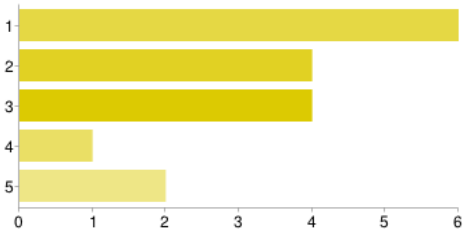
1	**6**	35%
2	**6**	35%
3	**2**	12%
4	**1**	6%
5	**2**	12%

Which types of data would be useful to collect? [photos]



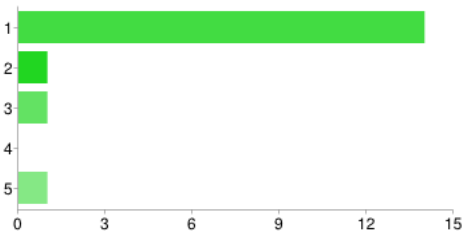
1	**9**	53%
2	**7**	41%
3	**0**	0%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [walking pace]



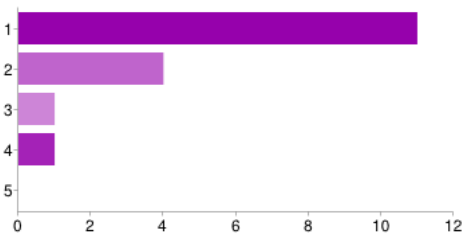
1	**6**	35%
2	**4**	24%
3	**4**	24%
4	**1**	6%
5	**2**	12%

Which types of data would be useful to collect? [tally of sightings for each species]



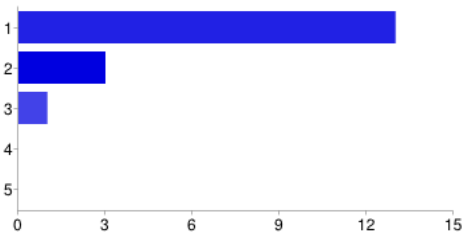
1	**14**	82%
2	**1**	6%
3	**1**	6%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [habitat category]



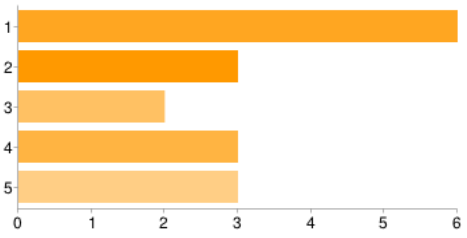
1	**11**	65%
2	**4**	24%
3	**1**	6%
4	**1**	6%
5	**0**	0%

Which types of data would be useful to collect? [differentiation of sections along route]



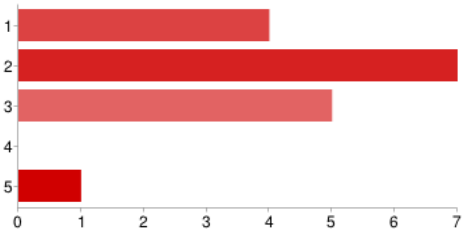
1	**13**	76%
2	**3**	18%
3	**1**	6%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [need for marked specimen data]



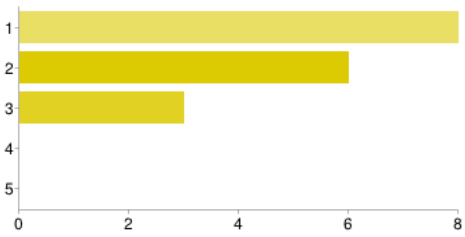
1	**6**	35%
2	**3**	18%
3	**2**	12%
4	**3**	18%
5	**3**	18%

Which types of data would be useful to collect? [behavior notes]



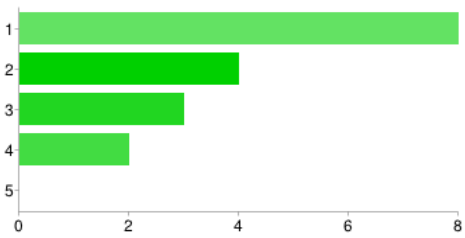
1	**4**	24%
2	**7**	41%
3	**5**	29%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [self generated species list]



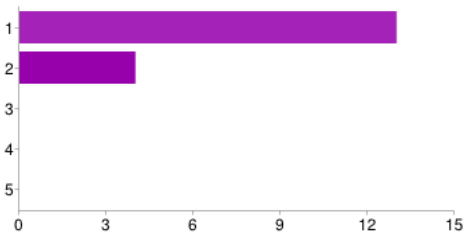
1	**8**	47%
2	**6**	35%
3	**3**	18%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [app generated species list]



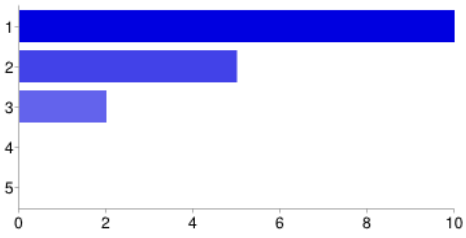
1	**8**	47%
2	**4**	24%
3	**3**	18%
4	**2**	12%
5	**0**	0%

Which types of data would be useful to collect? [survey site name]



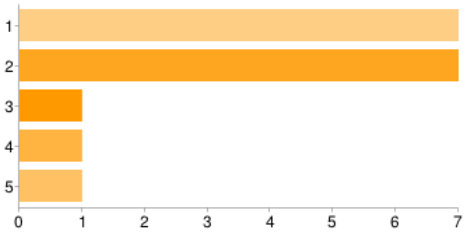
1	**13**	76%
2	**4**	24%
3	**0**	0%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [% cloud cover]



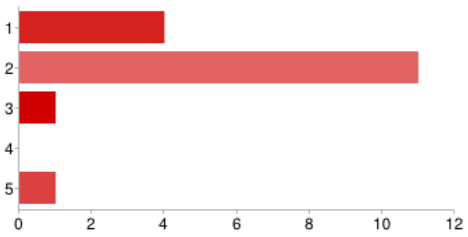
1	**10**	59%
2	**5**	29%
3	**2**	12%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [habitat conditions]



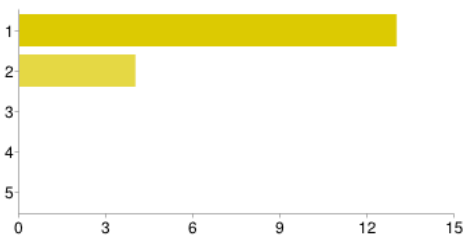
1	**7**	41%
2	**7**	41%
3	**1**	6%
4	**1**	6%
5	**1**	6%

Which types of data would be useful to collect? [level of ID certainty]



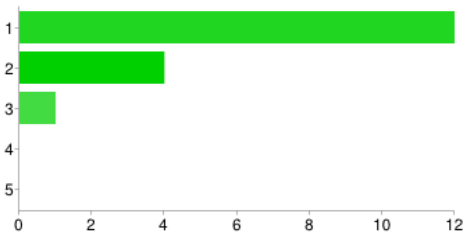
1	**4**	24%
2	**11**	65%
3	**1**	6%
4	**0**	0%
5	**1**	6%

Which types of data would be useful to collect? [start/stop times]



1	**13**	76%
2	**4**	24%
3	**0**	0%
4	**0**	0%
5	**0**	0%

Which types of data would be useful to collect? [monitor information]



1	**12**	71%
2	**4**	24%
3	**1**	6%
4	**0**	0%
5	**0**	0%

Please describe any other data that would be useful during a survey.

Description of survey site county, state, and ownership, i.e. "Folsom Point Preserve (TNC-owned), Mills County, Iowa. Also a place to record miscellaneous remarks such as "one of the Tiger Swallowtails was a black-form female" or "Butterfly netted and released for identification purposes." Assistant name Length of pauses Not necessarily needed for TN (but still might use it), but based on protocol comparisons, might be useful for OH since they do record: Larvae sighted Plants in bloom Energy sources (including nectar sources, sap, feces, etc.) possibly plant phenology - bloom nectar plants plants used as oviposition substrate degree days? min-max on temperature wing wear? Condition of the specimen -- whether just hatched or was it a worn individual -- might be helpful, but not critical. The ability to collect specific latitude and longitude of individuals of rare species is important but in general only the ability to collect a single location latitude and longitude of a particular site is needed. Also important is the ability to collect notes of other life forms such as plants, birds, mammals, other insects, etc. Free format mode for this latter information would be fine. unrecognized specimen "complex" of two or more species, or family-level recognition Names of discrete points within a survey route (e.g., Territory 1) Number of individuals observed at a particular point at a survey Status of each individual observed at each point (for territories: occupant, intruder, vagrant), and Occurrence and number of territorial interactions at a particular point. (e.g., Territory 1: 1 occupant, 1 intruder, 3 interactions; Territory 2: 2 occupants, 4 interactions) For migration observations: compass direction, estimated height above ground, estimated linear distance of butterfly from observer (for a distance sampling survey). We also use a field called note taker. It's left blank for people doing their routes solo. A comments field would be essential. We also should figure out how partial identifications will be dealt with.

What is the length in time of a typical survey for you?

0.5-3 hours 2hrs 1-2 hours 1-1.5 hrs at a site 1-2 hrs 2-3 hours 2-3 hours 30 minutes - 4 hours 1-3 hours 30-40 min 1-3 hours 1-4 hours 0 45 min 1 -2 hours 0.75-2 hours 2 hours

In general, what is the maximum time you would spend surveying in a single day?

All day 4hrs 3 hours 5 hours 6 6 hours 8-10 hours 8 hours 8 hours over multiple sites volunteers in Israel do it once in 2 weeks 3 hours for Pollard; 5 hr. for distance sampling of migrants 32 years 0 2h 3 hours 1 survey per day, no more than 2.5 hours 2 - 4 hours

What is the distance of a typical survey for you?

A few hundred feet up to 1.5 miles 2miles 1-2 miles 2-5 miles 1-3 miles 1-2 miles 2-3 miles 1 mile 1 mile for each site 300-600 m 4 miles 0 1 km 1 - 3 miles about a mile 2 miles

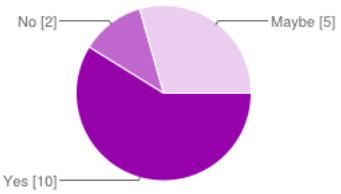
In what format(s) do you currently store your data?

Data from field notebook is supplemented with a written narrative when the survey is complete, then both are entered into an Excel spreadsheet for a complete record. excel spreadsheet and AVISYS Excel spreadsheet Word, Excel, GPS in garmin .gdb, Google Earth maps, jpg for images database database - online SQL Server, and local Access DB online database AviSys database spreadsheet and database online database Notebook; Excel spreadsheet xml mysql database SQL database database Filemaker Pro Database. We also have online entry available. Data from this system are stored in an HTML log and uploaded into the Filemaker Database at the end of each season. excel

How would you like to be able to review and use your data?

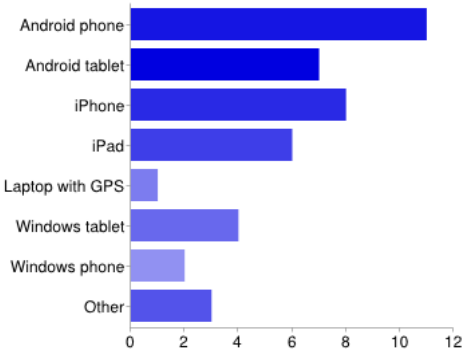
Spreadsheet and some sort of visual mapping excel spreadsheet and data on maps Custom web interface, mapping spreadsheet, web, mobile spreadsheet Spreadsheet, text files, custom web interface on the mobile device, mapping, custom web interface Spreadsheet spreadsheet, SQL database SQL database Spreadsheet, mapping, graphic and statistical analysis, custom web interface, SQL database, API access spreadsheet imported to bamona and reviewed on butterfliesandmoths.org website SQL database I'd be OK with spreadsheet or a delimited text file I would like to be able to generate reports, work with maps and do spatial analysis

Would you or your organization be interested in utilizing a central database solution if available?



Yes	**10**	59%
No	**2**	12%
Maybe	**5**	29%

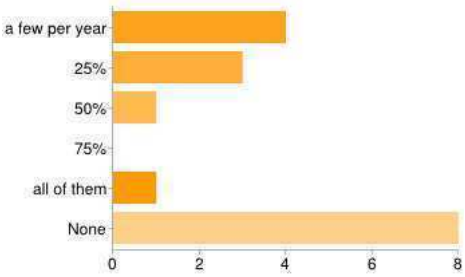
Which devices would you prefer to use for surveys?



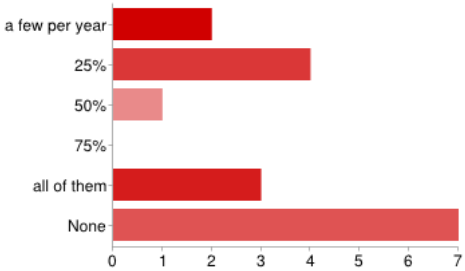
Android phone	**11**	26%
Android tablet	**7**	17%
iPhone	**8**	19%
iPad	**6**	14%
Laptop with GPS	**1**	2%
Windows tablet	**4**	10%
Windows phone	**2**	5%
Other	**3**	7%

How many of your surveyors fit into the following categories? [One-time users]

a few per year	**4**	24%
25%	**3**	18%
50%	**1**	6%
75%	**0**	0%
all of them	**1**	6%
None	**8**	47%

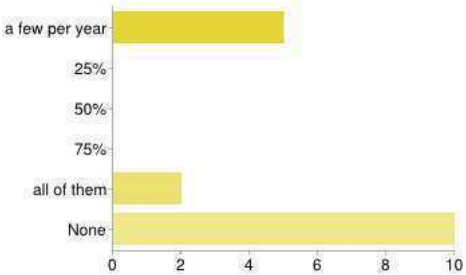


How many of your surveyors fit into the following categories? [First-year users]



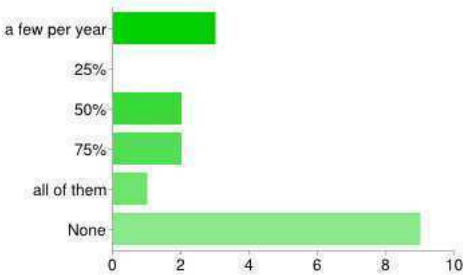
a few per year	**2**	12%
25%	**4**	24%
50%	**1**	6%
75%	**0**	0%
all of them	**3**	18%
None	**7**	41%

How many of your surveyors fit into the following categories? [Paid staff]



a few per year	**5**	29%
25%	**0**	0%
50%	**0**	0%
75%	**0**	0%
all of them	**2**	12%
None	**10**	59%

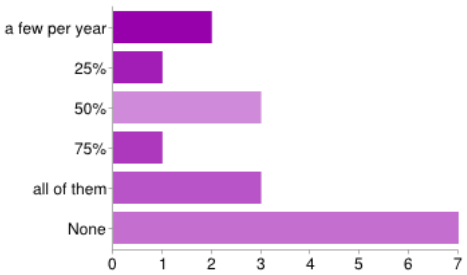
How many of your surveyors fit into the following categories? [Trained volunteers]



a few per year	**3**	18%
25%	**0**	0%
50%	**2**	12%
75%	**2**	12%
all of them	**1**	6%
None	**9**	53%

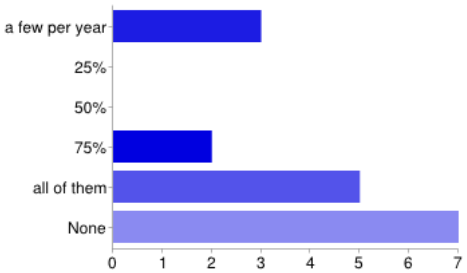
How many of your surveyors fit into the following categories? [Self-taught volunteers]

a few per year	**2**	12%
25%	**1**	6%
50%	**3**	18%
75%	**1**	6%
all of them	**3**	18%



None	**7**	41%

How many of your surveyors fit into the following categories? [Just me]



a few per year	**3**	18%
25%	**0**	0%
50%	**0**	0%
75%	**2**	12%
all of them	**5**	29%
None	**7**	41%

Please give any other comments or ideas you may have.

I would give priority to being able to record the exact location of sightings during a survey and of the survey route itself because most devices now have GPS tracking capability, and to me it would be the most important item needing to be known if the survey is to be replicated in the future. Within a site there may be multiple habitat types and if habitat-specific butterflies are present, knowing which habitats they are found in is a big help in trying to relocate them in the future. If the app had photos to help with ID that would be great. An app that shows tracks and wpts would be good. I think the app could be adapted to do bird surveys. I do some of those and the same data is used but would just need to be birds instead of butterflies. Most apps have you mark a point and note whether you are moving or not but don't track route. Should be able to enter unknowns at various taxonomic levels. Would be best if it can be used across a wide variety of platforms since volunteers are not likely to all have the same hardware/software and may change frequently. Of top priority for me is to have a central database that we can access on the Internet to enter all of our Iowa butterfly/skipper observations. Then have the ability to search for ALL records (from other individuals as well as mine) at a given site, or by a given species, or by a given observer. You might consider adding a data field that indicates under what conditions the observer documented the species -- e.g., collected, caught and released, photograph, good sighting, fair sighting, etc. The main challenge to me is the ability to use a fragile electronic device in the field while also carrying camera and binoculars. Important that the app will enable adding individuals of a given species to an existing list, and minimum re-uploading events. It is critical to enable people to a-priori decide, or know, where the data are to be sent to and how will sensitive data (e.g. rare species) be handled. The Pollard-like surveys I do are somewhat different than the usual design. My daily surveys are bicycle routes through a series of butterfly territories on campus. Territories are fixed locations at which there is a relatively high probability of finding a butterfly; butterflies are much less likely to be found away from territories. Thus there is much more focus on examining particular points along the route than there is on monitoring a continuous strip as is typically done in a Pollard survey. Hi, I am the developer of the butterfliesandmoths.org web application and will be working with Leslie Ries and Doug Taron to create a repository for all of the data on a new version of the bamona applicaiton. I would be happy to meet with you and discuss how to move forward with a greater degree of cooperation. From my point of view it is important not to mix pollard walks with detection/non detection surveys (a better description than presence absence). An app specially for one purpose is probably the best.

Describe an ideal user interface for performing a butterfly survey.

Here's what jumps to my mind. The first page is a series of text boxes where users enter the name of the site they're surveying and its location, the weather, their start time (this could probably be obtained automatically from the device), description of habitats at site... possibly some other info. The user hits a "Start" button which activates the GPS tracking and starts a time recorder and the device displays a series of collapsed drop-down boxes with the name of the butterfly families (Swallowtails, Whites/Sulphurs, Blues/Hairstreaks/Coppers, Brush-footed Butterflies, and Skippers). Upon seeing a butterfly, the user taps the dropdown and an expanded list of species appears, and the user taps the entry of the butterfly species that was seen, at which point the device makes a GPS and time recording. Upon completion of the survey the user hits a "Stop" button to stop the GPS tracking and record the survey ending time. Start the survey (app would know location and time), keep track of route, allow marking species seen and location, and would allow the download of data and also make a map showing the route and wpts. If the data were in a master location (check out birdlog app and see some of the features. It allows you to email the list to yourself but you have to go to ebird to see other data.), then the data is available to others to use. I like the idea of data being open for others to see and use. Easy to read in bright sunlight (high contrast and large type) One screen for general data (weather, site, etc.). Click submit and it goes to Start screen Click Start and it records start time and takes you to sightings screen. Pause/Play button on sightings screen to

automatically record long pauses for ID Once finished click "Finish" and it automatically records stop time and takes you to a review screen so you can proofread data. Click "Save" and it saves and takes you to tally screen. Thanks for working on this! Data categories used need to be quickly accessible; user should be able to modify and save a form with just categories needed for survey; then other info can be added at beginning and/or end - temp, rH, wind, plants blooming, etc. Other than what I described in the last section, no other comments. I sometimes use a Garmin GPS device with mapping capabilities, but nothing more. Ideally it would be great to have an auditory interface or some other hands-free way to enter and confirm data in the field as often as needed. Next best way would be to enter data through touch screen back at the car. Prefer to do it by phone or skype when you have the time and interest! Kind regards, Guy Something that would allow me to quickly collect data along with environmental variables and geographic coordinates, and then to readily download data to various analytical, graphing, mapping, and database programs. Let's discuss! This description is for a Pollard transect. The user interface should not require any changes to the survey protocol beyond simply recording data electronically rather than with pen and paper. The screen on the interface should resemble as much as possible the paper form that a program already uses. There might be a setup step that the user performs once before taking the interface out into the field. This setup might involve setting default values like the monitor and site names. All species name entry should be menu driven. During the setup, the user might pre-load the interface with those species that they most typically observe. For each field visit, the user would open the ap and indicate that they wish to begin a new route. Data recording should not involve typing anything on a keypad once one is out in the field. Buttons are preferable to menus for most tasks. The device that I have been collecting data with allows the user to toggle up and down in selected cells in the data table in units of 1. I like that system a lot. Being able to decrease as well as increase value in the cells is essential, because there is always a risk of mis-entry that you would want to be able to correct on the spot. It should be easy, while in the field, to enter species not pre-loaded onto the data sheet interface. This should be menu-driven. When the route is complete, the user should be able to submit the data directly from the field. The developers should bear in mind that this device will typically be used in bright sunshine. The interface color scheme and font choices should allow the text on the screen to be easily read under these conditions.

Number of daily responses

Appendix C: Existing Apps

Wildlife or Animal Tracking Apps

eReca - UK Butterflies
https://play.google.com/store/apps/details?id=com.ereca.specrec&hl=en

Bugs count
https://play.google.com/store/apps/details?id=uk.ac.nhm.bugscount

Record Wildlife
https://play.google.com/store/apps/details?id=com.record.wild.life

Wildlife Log
https://play.google.com/store/apps/details?id=appinventor.ai_Ockmeyer.WildlifeLog

Africa: Live
https://play.google.com/store/apps/details?id=com.kruger.live.working

Sightings Tracker
https://play.google.com/store/apps/details?id=za.co.hrdit.stracker

Bird Atlas Recording Software
https://play.google.com/store/apps/details?id=com.smartphone2b.birdatlas

Kruger Park Sightings
https://play.google.com/store/apps/details?id=com.wKrugerSightings

General Mapping and Geo-location Apps

GPS Grid Reference
https://play.google.com/store/apps/details?id=com.luck.GgridReference

GPS Waypoints Navigator
https://play.google.com/store/apps/details?id=com.luck.GgridReference

Gmemo for Field Survey
https://play.google.com/store/apps/details?id=jp.android.Gmemo

Survey
https://play.google.com/store/apps/details?id=com.survey

GIS4Mobile
https://play.google.com/store/apps/details?id=com.gis.gis4mobile

2GIS
https://play.google.com/store/apps/details?id=ru.dublgis.dgismobile

Maverick
https://play.google.com/store/apps/details?id=com.codesector.maverick.lite

Path Tracking
https://play.google.com/store/apps/details?id=com.pathtracker