Unified Butterfly Recorder

iOS Development

Project Plan

ADVISOR: DR. DIANE ROVER CLIENT: NATHAN BROCKMAN

TEAM MEMBERS: ERIC LARSSEN (TEAM LEADER), SEAN SHICKELL (COMMUNICATOR), CHARLES MANKIN (WEBMASTER)

Project Charter

Project Name	Unified Butterfly Recorder iOS Development
Performing Division	iOS Development Division
Performing Group	Senior Design (EE/CprE/SE 491) Group Dec14-16
Product	UBR App for the Apple App Store

Prepared By

Document Owner(s)	Project / Organization Role
Eric Larssen	Team Leader, iOS Development, Scrum Master
Sean Shickell	Communications, iOS Development, Documents, Maps
Charles Mankin	Webmaster, iOS Development, Database

Project Charter Version Control

Version	Date	Author	Change Description
0.0	2/24/14	Sean Shickell	Document Created
0.1	2/26/14	Sean Shickell	Document formatted and properly filled with any pertaining information regarding this project and the Project Plan Rubric
0.2	4/09/14	Sean Shickell	Added Project Charter
0.3	4/14/14	CJ Mankin	Added information based on current project progress
1.0	4/16/14	Sean Shickell	Formatted to the new Project Plan Rubric, adding any required information as needed

Table of Contents

Project Overview

Project Statement

Background

Butterflies are great indicator species; from monitoring climate change to determining quality of habitat restoration, butterflies are used a lot to help answer scientific questions. Across the nation and around the world, however, there is a lack of information on native butterfly species, their annual dispersal and numbers. To help with this task the scientific community has begun relying on citizen scientists, which are volunteers of different backgrounds, to go out in the field and do the surveys.

In January 2013, Team Butterfly, Senior Design Team Dec13-08, took on the task of creating the Unified Butterfly Recorder (UBR) Android app. Throughout 2013, with the collaboration of the Reiman Gardens Entomology staff, UBR was designed, developed, and released on the Google Play store. This app has the potential to significantly and positively alter the course of global conservation research. It is being tested currently by researchers in the United States, Canada, Germany, the Netherlands and other countries around the world. In the spring of 2014, a comprehensive in-the-field study will be conducted, by RG Entomology staff, to test all aspects of the current application, as well as gathering information as to the usability of the app by researchers of different backgrounds, from volunteers to professionals. The app and results of this study will be presented at various conferences in the entomology and conservation fields.

The release and demonstration of UBR in the conservation community has produced excitement about the ease of data collection, ability to standardize survey results despite variable collection methods and the capability to manipulate analysis (mapping, graphing, other visuals for presentation) to benefit research and enhance conservation efforts.

Survey 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 specific 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.

Pollard Walk: A surveyor repeats an identical transect (route) different 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 transect 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 Survey: Similar to field trips, this protocol involves individuals or groups walking an indeterminate path looking for as many individuals as they can find.

(Check references) [1]

Our application will need to accommodate different input methods corresponding to the types of data collected by these surveys. The users must be able to choose which kind of survey they want to perform before a survey is created.

Problems

The problem statement from the Android UBR project is as follows:

> "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 useful. 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."

(Check references) [2]

The success of the Android UBR has taken the first step in tackling this challenge. The etymology communities has taken steps to adopt the UBR as an official surveying tool, with the help of Reiman Gardens researchers championing its use. There is still no central warehouse to dump surveying information from the myriad devices that complete the surveys, but our client continues to work with BAMONA to realize that goal.

For now, the most important next step is to get the application on as many devices as possible, so that we can get the surveys in the hands of as many surveyors as possible.

Objective

The success of the Unified Butterfly Recorder has made the fact that there is no iOS counterpart to the Android application unacceptable by current social standards. By not providing access to this application for those using Apple mobile devices, we are leaving out a substantial portion of surveyors, especially among the "citizen scientists" that many surveyors are relying on. It is time to bring the UBR to the App Store.

Our objective is to create an application that looks and feels like the Android UBR, while also looking and feeling like a traditional iOS Application. The subtle differences promise design challenges, but our goal is to create a seamless emulation of the original UBR. Surveyors who have already used the Android UBR should be able to migrate between the two effortlessly, and new UBR users should be able to start the application and jump right in to surveying, without any sustained training.

Deliverables

Expected

1. A universal iOS application optimized for iOS 7 that can be used on any Apple mobile device
2. Updating the UBR Android app in response to field study results
3. Implement data visualization and manipulation feature for UBR

Non-Required

Windows mobile version of UBR

I/O Specifications

Arrows indicate the direction of information flow. (Check references) [3]

Apple device → Server

We are currently testing with a SQLite 3. The data will be exported as CSV files to the database.

Web server → Database

The web application will communicate with the database using Drupal's database abstraction layer in PHP. Both of these were set up by the previous group. (Check References)

Weather service → Apple device

We will interface with OpenWeatherMaps periodically to get weather information, we will have basic information (temp, wind speed & direction, …) displayed in a widget on our app.

Mapping service → Apple device

We will interface with Google Maps to get mapping information, we will have basic information (small map display) displayed in a widget on our app.

User Interface Specifications

The user will use the GUI to create, edit, and submit a survey to the database. We plan on many different types of people using this app, from Researchers and conservation workers to people who just started butterfly tracking as a hobby. Therefore, we need to make the app easy to use and understand (which is the "Apple" way), and yet with enough capability and functionality to support the advanced users.

As mentioned previously, one of our main goals is making our app appear as "Apple" as possible. Since the functionality might also be getting updated (as per user feedback of the Android version), we are planning to develop the iOS version with the user feedback in mind.

Moreover, simplicity and beauty are other fundamental goals were trying to accomplish. We plan to conform to Apple's UI Design Basics: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/, while developing our app to keep it simple, and use proper color, graphics, text, and object size to effectively show content without overcrowding the UI screen.

Hardware Specifications

Apple device

The user must have an Apple Device that can run iOS 7.0 or newer, preferably on a 64-bit device. The device must have network connectivity, either wifi or cellular connection capability. We will assume the smallest screen size to be the size of the iPhone 5 (or 5S, SC). Our layout will change appropriately to incorporate more and give more space between items as the screen size of the device being used is larger.

Server

The server will interface with the phone and the BOMONA database and will need to always be on, this was set up by the previous group and is something that we will leave in their control until other arrangements are made. Depending on user size, we will probably need to increase the amount of servers over time to meet load needs.

Third party hardware

Google Maps and Open Weather Maps provide servers, weather stations, network devices, and more that will all be used to get information for sightings. These are very large providers which have excellent track records for always being up and running. The same can be said of BOMONA (based on the previous group's choice to work with them).

Software Specifications

Operating system

This app will be designed to run on iOS 7.0 or newer.

Server

The server will interface with the phone and the BOMONA database and will need to always be on, this was set up by the previous group and is something that we will leave in their control until other arrangements are made. We will interface with the Server using the web interface.

(Check References) [4]

Database *(Schema)*

Main Table

- key - Record_ID
- Columns
  - Record_ID → Links into Environment and Pictures
  - Surveyor_ID → Links into surveyor Information
  - Survey_ID
  - Start_Time - datetime
  - End_Time - datetime
  - Site_Name / Route_Name
  - Latitude - GPS
  - Longitude - GPS
  - Amount_of_Butterflys_Seen

Weather Service Environment Table

- key - Record_ID
- Columns
  - Record_ID
  - Temperature
  - Wind_Speed
  - Wind_Direction
  - Cloud_Cover

User Environment Table

- key - Record_ID
- Columns
  - Record_ID
  - Habitat_Type/Category
  - Section_Number
  - Temperature
  - Wind_Speed

- Wind_Direction
- Light_Level
- Cloud_Cover
- Comments

Surveyor Table

- key - Surveyor_ID
- Columns
  - Surveyor_ID
  - Surveyor_Name
  - Surveyor_Email
  - Surveyor_Password
  - Surveyor_Phone
  - Surveyor_Organization
  - Surveyor_Comments

Pictures Storage

- key – Record_ID
- Columns
  - Record_ID
  - Path_to_Photo

Third party software

We will interface with Google Maps, Open Weather Map, and BOMONA using APIs and a similar method to the previous group (in regards to BOMONA).

System Requirements

App Requirements

The system will include an iOS app that will be identical in functional requirements to that of its Android counterpart, so far as the iOS hardware allows. One of our main goals is to make it appear as "Apple" as possible. The Apple device will need to communicate with the weather service, OpenWeatherMap, along with being able to interface with Google Maps. The app will only allow users to submit data to the database.

The following was transferred from the previous group's website. (Check References) [5]

Functional Requirements

Android and iOS Applications

Synopsis

Allow users to record all relevant information of a butterfly sighting during a survey and submit that information to a database. The Android and iOS applications will conform to identical functional requirements, non-functional requirements, and output interfaces.

List of requirements

Must create a default list of common butterfly species based on the location of the survey. 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
- start/stop times
- light levels
- walking pace

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

- level of correct identification certainty
- surveyor information (level of expertise, contact info, # of survey takers - recorders, observers)
- taxonomic tree-based classification
- miscellaneous comment section (animal life, plants present)
- differentiation of sections along route, distance/habitat category

Web services

The following data will be imported by the server for each record

- temperature
- wind speed
- wind direction
- percent cloud cover

Database

Allow authenticated submissions through Drupal's XML-RPC interface

Web interface

Allow exporting of structured data (csv) based on queries

The following was transferred from the previous group's website. (Check References) [6]

Nonfunctional Requirements

Android and iOS applications

- Performance: The battery must last a minimum of 3 hours of surveying time on a recent mobile device with a quality battery.
- Ease of use: Users should require no training to record a sighting and perform a survey with the app
- Security: The authentication system must protect users' passwords and private information
- Graceful failure: The application 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 apps 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
- Security: The app should not leak users' BAMONA account information during authentication, or store account information in plain text

Web server

- Availability
  - The database should handle up to 10,000 requests per day
  - The database should store up to 30,000,000 records
- Security
  - The server should not allow submissions from a mobile app to override data in the database
  - The server should not allow mobile apps to query the database
- Maintainability
  - The database should allow inclusion of new data fields

Resource Requirements

Hardware

iOS Devices: We all have access to iOS operating devices to do testing and demoing.
Server: BAMONA will provide the physical server, and we have a backup running in a VM acquired through campus IT. Check Server section below for more information on BAMONA.

(Check references) [7]

System software

iPhone/iPad simulator for testing
SQLite3 framework for on-board database access
CoreLocation framework for GPS coordinates

Development software

Xcode 5 - We all have Mac Laptops in order to be able to develop the app using Xcode.
Eclipse/-ADT – For testing/viewing/editing UBR Android code
The database has already been set-up for us. This is a collaboration with BAMONA on the database, web server, and web application portions of the system.
Mac OS-X

Database

"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: http://www.butterfliesandmoths.org/about. BAMONA currently has a web interface and database for hosting all records.

(Check references) [8]

In addition, there will be a database included with the application that is used to hold all pre-loaded butterflies, as well as surveys and the sightings associated with each survey. The database will use the SQLite3 framework for iOS7

System Analysis

iOS app

We were assembled to build an iOS version of this app, and so we didn't have much choice there. However, since there was already a lot of research into mobile platforms done by the previous group, and they covered the Android portion of the mobile platform devices (and it having such a great response) it makes sense to cover Apple devices running iOS (since it's the next biggest, after devices running Android).

On the other hand, we chose iOS 7 to be our target operating system, because of many complex, yet very simple reasons. One being iBeacon, which is a Bluetooth Low Energy (BLE) location software/hardware that is just getting released with iOS 7 (along with a few others that have great potential with our app in more than one way). Another, is the fact that our app will be released in the Fall, when iOS 7 will be a familiar operating system in the mobile realm, and other mobile services (such as iBeacon) will have had time to mature and better integrate with 64-bit Apple Devices running iOS 7. Moreover, iOS 8 will be getting released at some point with its hardware compliment, the iPhone 6 (which could possibly get released in June or July). This would improve/add functionality to our app correlating with improvements in the hardware & software coming from Apple and third-party companies.

We have had talks about some of the hardware available and that a lot of Android devices have more sensors than Apple products. However, a lot of the main things are available on most smart phones. Those being, GPS, date & time, and camera. Using these in combination with the OpenWeatherMaps and Google Maps, we can store crucial information such as approximate locations, time-stamps, and weather conditions during sightings.

Mapping service

We chose to use Google Maps over Apple's Maps, because we feel that Google keeps better updated with road construction and other physical changes.

Weather service

OpenWeatherMap is a free service that provides weather data and has more than 40,000 weather stations worldwide (openweathermap.org), and is designed for web and smartphone applications. This felt like an obvious choice to get weather information from.

Web component, BAMONA, etc.

The web application and the arrangement with BAMONA were set up when the previous group was developing the Android portion of this project. We plan on keeping them the same and making the proper interface to properly communicate with them.

Simulations and Modeling

Unit tests

We will write unit test suites for the app using the Xcode Unit Test target. We will also use Xcode 5's coverage features to measure the code coverage of our unit test suites.

Database simulation

We currently have the SQLite3 database part (which includes being able to import data from a CSV file) ready and will soon be integrating it with a working iOS application that allows users to modify the data, and create tables (which is needed for the UBR database).

OpenWeatherMap simulation

We also have an online tutorial that walks through how to integrate OpenWeatherMap API into an iOS app. We plan on creating the example app provided in the tutorial to see all the coding needed to integrate Open Weather Map into our app.

Alpha Release (for demoing UI) OR Screen Flow Diagram with layout structure

We plan to have at the bare minimum an Alpha, or mockups of the UI, complete by the end of this semester in order to get much needed feedback on the look and feel. For more information, see Current Project Status below.

System Description

Block Diagram

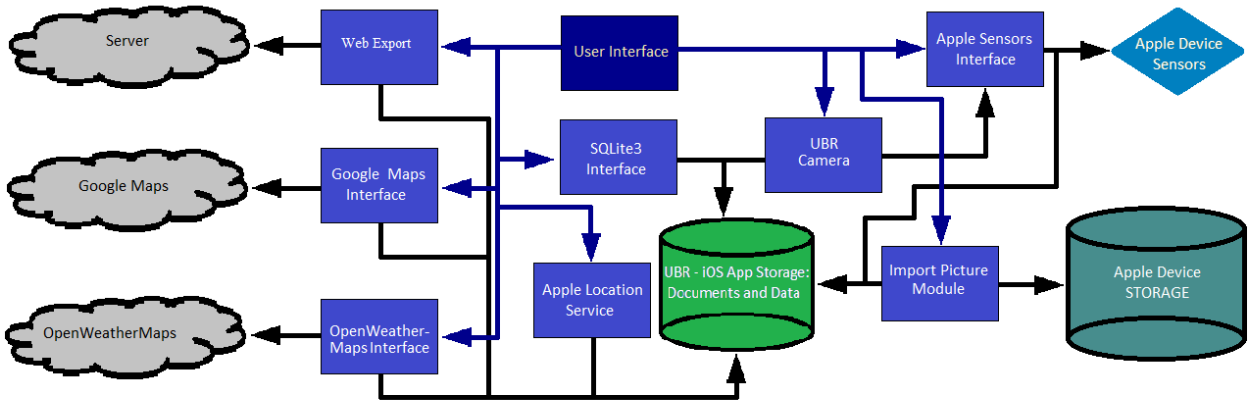

Fig. 1 UBR iOS App Sytem Architecture

UBR iOS Sytem Architecture: User Interface Controller (Dark Blue), data interfaces (Blue), App Storage (Green), Device Sensors (Sky Blue), Device Storage (Teal), Network-Connected Devices/Providers (Grey)

(Check References) [9-11]

Functional Decomposition

User interface

The mobile applications will provide a graphical interface to the user of the application. The interface allows the user to add butterfly sightings into the system. The application will collect numerous data points from the devices on board sensors, such as GPS location. The data will be stored locally until the user decides to upload it to the central database. The application will only allow for uploading new records to the database; as such, a user will not be able to pull data from database through the application.

Web server

The web server was set up by the previous group and will expose a CRUD interface for the mobile applications. This means that the web server will provide a means through which the applications will be able to authenticate users, upload survey records. The server will also provide a means through which the users would be able to access and export the survey records in the database. This functionality will not be provided to the mobile application. The users will interact with the web application on the server to submit queries and export the data that is returned from those queries.

Database

We are currently testing with a SQLite 3 database, and investigating Core Data, the main Apple database framework, as a possible replacement. The data will imported and exported as CSV files.

Web application

The web application was developed in PHP by the last group using Drupal as a backend and database abstraction layer. The application provides users with the ability to perform queries on the survey data and export those results in a common format, such as CSV.

Weather service

We will be using OpenWeatherMap, a free weather data and forecast API.

Mapping service

Google Maps API will be used as our mapping system.

Storage devices

We will store data in the App's "Documents and Data" storage. Items such as pictures, thumbnails, and CSV files. Pictures can be imported from the Apple Device storage into our App's storage.

Testing

Unit tests

We plan to write unit test suites using the Xcode Unit Test target (and possibly some third party programs), and use tools provided by Xcode's to measure the code coverage of our unit test suites. Xcode provides extensive coverage features, trying to provide all a developer needs to create amazing iOS apps.

Beta testing

If we don't have the app ready at the end of this semester, then we will plan to have it out by no later than the beginning of July, allowing Reiman Gardens to perform surveys in the field using their Apple devices and giving us our first user feedback.

Work Breakdown Structure

Timeline

Date	Milestone
February 7	Project Team assembly, Client/Advisor/Team introductions
February 28	Project Website in use
March 14	Pre-Alpha version release and/or mock-up of basic functionality and UI
April 1	Divide tasks among team members
May 26	Alpha version release for surveyor testing with 1 major protocol - Meandering
August 18	Feedback collected, begin improvements; work on implementing protocols similar to Meandering
October 1	Beta version release. Initial feedback improvements made. Begin work on implementing all survey protocols
November 1	All known bugs fixed; all survey protocols implemented; version 1.0 release
December 16	Poster and presentation finalized

Risks and Mitigation Strategies

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 plan on having plenty of documentation and communication to ensure picking up the slack in such case. With the use of GitHub and other software development tools, all code will be available to all members of the group.

Data license

We need to explore the legal implications of ownership of the data submitted through our application into the database. We need to determine what the rights of users will be with respect to their own data and data submitted by others. BAMONA currently has defined standards for this, which are viewable here: http://www.butterfliesandmoths.org/faq/what-are-terms-use. We will further explore this with our client in the interest of providing an open community.

(Check references) [12]

Lack of Experience

This group does not have experience developing applications for iOS, and only limited experience developing mobile applications. As such, we have designated the first three months of the semester for learning how to use Objective-C and Xcode 5 to build iOS applications.

We will start by using dedicated tutorials that can be found on the web, like www.codeschool.com, and then search for code scenarios or walkthroughs that pertain to a specific facet of our project. Our client and advisor are aware of the challenges this will present, and have agreed to this learning program, provided that progress is continually evident.

Cost Consideration

Software Development

All software development is done through collaborations with the previous group, the staff from Reiman Gardens, our Advisor Dr. Rover, and the hard work of this senior design team at no cost. The cost for this team is instead paid by a class grade at the end of this semester in Senior Design 1 and next semester in Senior Design 2, along with each member of this team gaining valuable experience working on an actual mobile application that will be released to the world.

Production Costs

ISU has a contract set up through Apple for any costs associated with making an iOS app available on the App Store. There is an App Committee review process that each app needs to go through to qualify for ISU funding.

Market Research

Survey Database

The previous group was able to make contact with the Butterflies and Moths of North America (BAMONA) organization. BAMONA attempts to serve as a one-stop database for all butterfly and moth data in the area. The previous group agreed to tailor their design specifications for data exporting to match BAMONA's database. This process is still in progress, and we will aim to adhere to the same export specifications that the Android group is implementing. More about the BAMONA project can be found at: http://www.butterfliesandmoths.org/about

Protocols

In attempting to create a seamless UBR experience between the two applications, we will strive to collect the same information corresponding to specific survey protocols and process them in consistent ways. The Android group has done the work in determining the data fields that most surveyors wish to collect, and we will follow their model.

Need

It is a well-known fact that Android devices are the dominant mobile platforms worldwide, but iOS devices have much higher numbers relative to global averages in the US, Canada, and Australia. (cite - https://econsultancy.com/blog/64376-65-of-global-smartphone-owners-use-android-os-stats#i.1vp69g36dlcrfu) At Reiman Gardens alone, several surveyors, including one of our clients Anita Westphal, are restricted to using the Android UBR on borrowed devices rather than using UBR on their personal iOS devices. It is clear that there is a high demand among the surveying population for an iOS equivalent to the Android UBR.

Current Project Status

Team Website

The website has been created to hold all documentation from this semester and the next, as well as links to the Android UBR and Reiman Gardens UBR websites for any users looking for up-to-date information on the status of the app.

Starting in April, the website will also contain updates from development sprints that the group will conduct.

Prototype

Currently learning the iOS development language and environment. We have gained access to the previous group's GitHub repository and are studying their code and system architecture to better understand all the components and behavior of the UBR app.

We have created some UI designs that resemble the Android UBR system, and that we believe hold true to the typical iOS user interface models. These designs have been cleared by our client, and development of the Alpha prototype has begun. Our goal is to have this released through the Xcode 5 beta test model by the end of May.

Server

As of now, the BAMONA server is not accepting data from the Android UBR. Any data exporting is done to the local storage disk on the device or to the Android UBR website

server. We will therefore design a system by which we can submit survey data from the application to the web server given to us for our website, as a proof-of-concept.

If in the process of development the BAMONA server begins accepting surveys, we can modify our design to send it to BAMONA. We will also provide a local file exporting mechanism, to be included in the Alpha prototype. This file can then be extracted from the device at a later time.

Conclusion

The production of this app is demand driven. There was (is) so much excitement about the production of the Android version of UBR, it was only a matter of time before this app was expanded to include iOS devices.

This app will definitely deliver more than expected. Our team will develop this app with all iOS has to offer and integrate Google Maps and Open Weather Map for up to date mapping and weather information.

We will also be in contact with the previous group, so that even before we're finished, we can make adjustments to menu layouts, settings, tools, and more based on current user feedback of the Android version and what the previous team sees fit.

This, combined with implementing Apple's UI Design Basics, will help us keep the design simple, intuitive, and elegant, while still giving all the functionality available to Android users.

References

Continuation

This project is a continuation from the Butterfly Population Survey app (butterflies.ece.iastate.edu), and therefore we are working with the previous team on several things. We found a few things overlapping between our projects and thus a few things and been moved over from their Documents to ours. Below is what was transferred and the rationale behind why these items have been incorporated into our Project Plan.

1. Survey Protocols

These were transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-project-plan.pdf → Problem Statement → Protocols and not modified as these are standards to the butterfly sighting community

2. Problems

These were transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-project-plan.pdf → Problem Statement → Problems and not modified because it was meant to be quoted

3. I/O Specifications

These were transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-design-doc.pdf → Detailed Design → Component Interface. For this portion I mimicked the layout and transferred the "Web server→Database" as is since both of those were set up by the previous group.

4. Database (*Schema)*

This part was transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-design-doc.pdf → Detailed Design → Database. Since they set up the database and we plan on interfacing with the one they set up, we have to abide by the schema they set up.

5, 6. Functional & Nonfunctional Requirements

These were transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-design-doc.pdf → Background → Functional Requirements & … → Nonfunctional Requirements and not modified as we are trying to have the same functionality regardless of platform.

7. Hardware

Transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-project-plan.pdf → Resources → Hardware → Server and not modified as we will be using the same hardware.

8. Database

This information was transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-project-plan.pdf → Market and Literature Survey → Questionnaire & … → Hardware & … → Risks and Mitigation Strategies → BAMONA and modified to give a brief summary

9-11. Mobile applications, Web server, and Web application

These were transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-design-doc.pdf → System Design → Functional Decomposition → Mobile applications & … → Web server & … → Web application. For Mobile applications we changed the title to "User interface" to better match our block diagram. For the Web server and Web application, we simply inserted the text "was set up by the previous group and" and "by the last group" and alter some text to insure the inserted text matched properly with the syntax. The reason for this being that it gives credit to the previous group and lets the reader know that those items have been set up, and that our job then becomes interfacing with these items properly.

12. Data license

This was transferred directly from http://butterflies.ece.iastate.edu/files/butterfly-project-plan.pdf → Risks and Mitigation Strategies → Data license and was not modified as it deals with the relationship between our app and the BAMONA database.