

# Unified Butterfly Recorder: iOS

DEC14-16: Eric Larssen - CJ Mankin - Sean Shickell

Client: Reiman Gardens, Nathan Brockman - Anita Westphal

Advisor: Dr. Diane Rover

## Introduction

Butterfly population levels are an effective way of indicating climate changes in an environment. The 2013 Unified Butterfly Recorder team created an Android application (app) that changed the way butterfly sightings are recorded around the world. Popularity of the app was exceptional, but Apple users, including several Apple-exclusive organizations, were unable to participate. Thus, the demand for a similar iOS app led to the creation of this team.



### Problem

Apple users are unable to utilize the Android app and are currently entering survey information manually using paper and pencil.

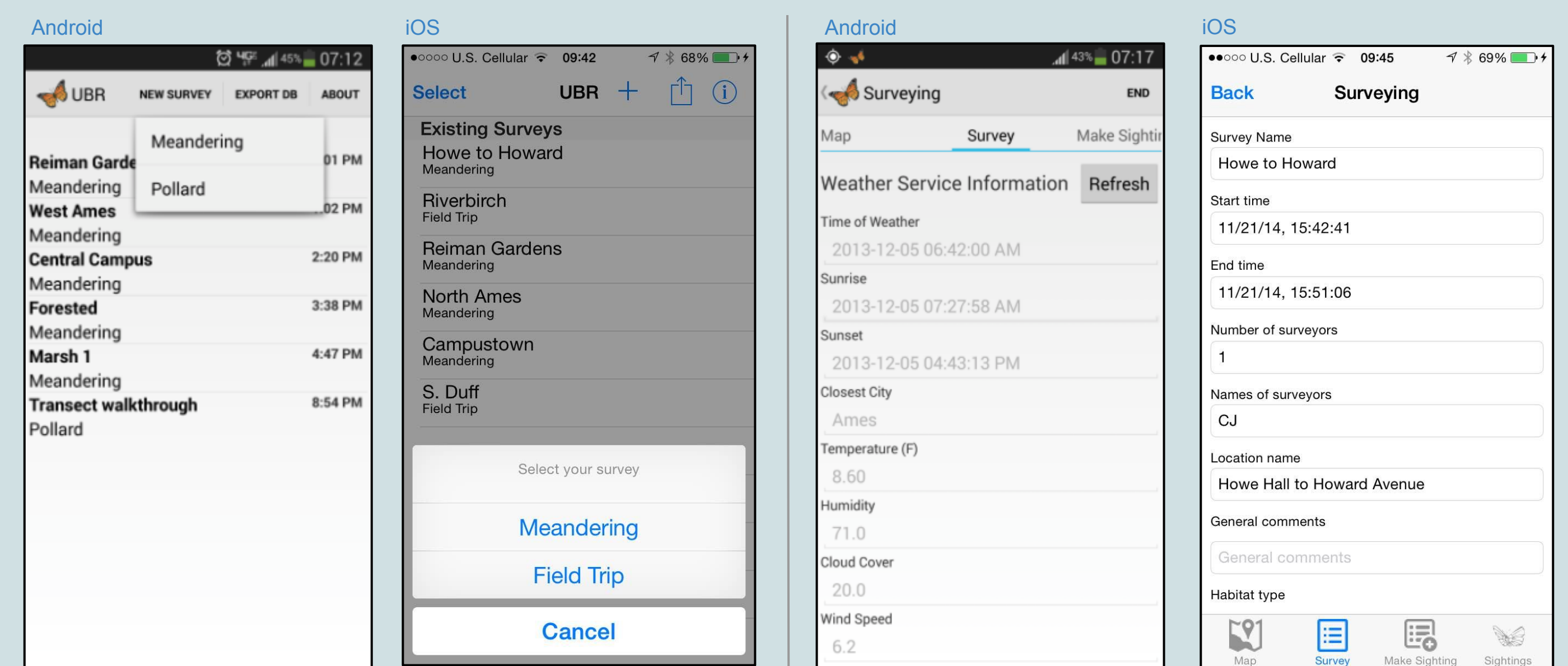
The aggregation of data requires manual effort and is therefore prone to human error.

Third party organizations need to convert the data from the Android app to their own systems, and currently have to do so manually.

### Solution

- Create both an iOS and server application
  - iOS
    - Mimic Android app look and functionality
    - Address possible loss of functionality between apps
    - Distribute through Apple's App Store
  - Server
    - Distribute data to third party organizations
    - Add layer of authentication
    - Avoid loss of data

## Usage Compatibility



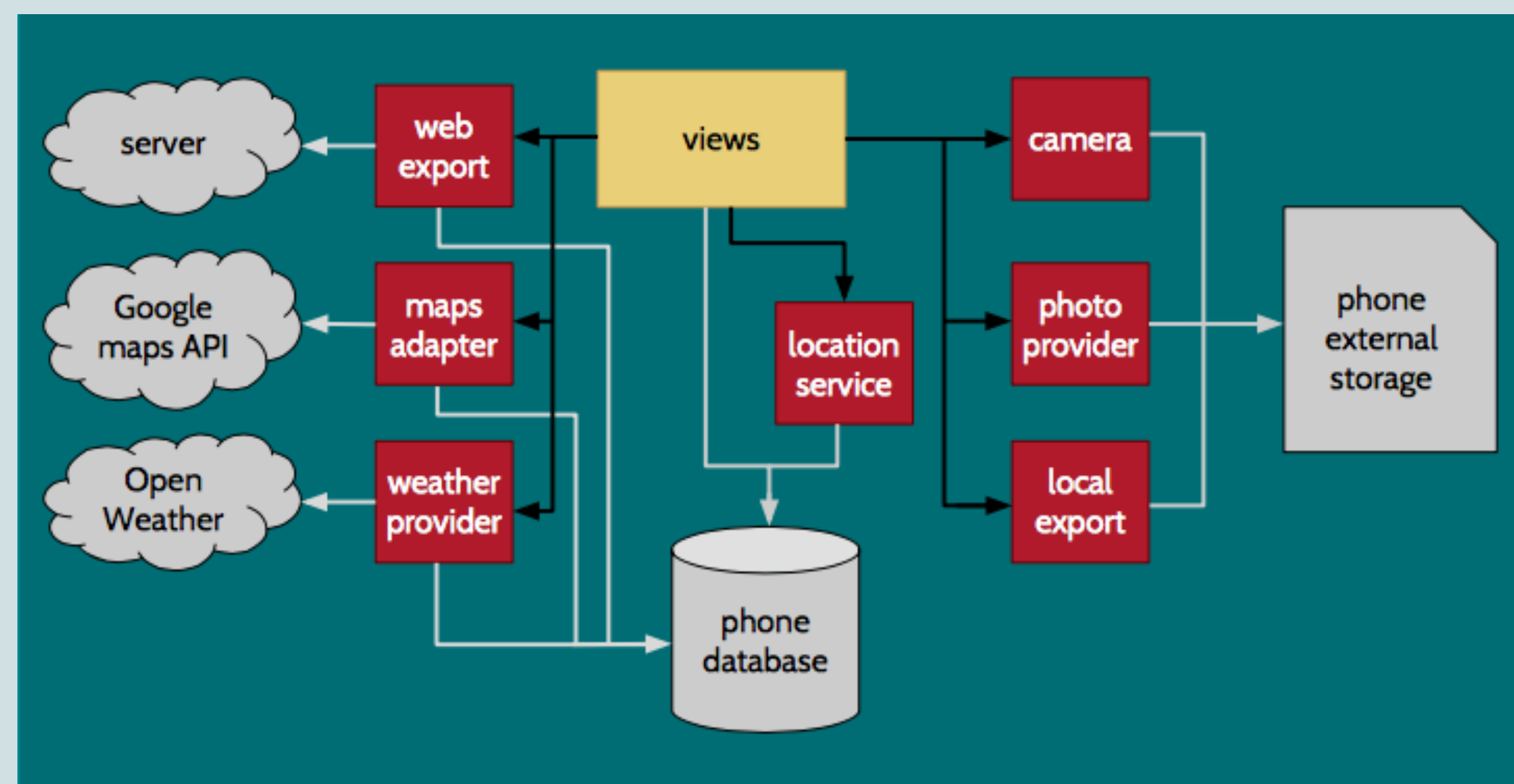
### Survey Selection

- ❖ Survey operations
- ❖ Easily recognizable action buttons

### Survey Information

- ❖ Tab bar navigation
- ❖ Breadcrumbs support

## Architecture



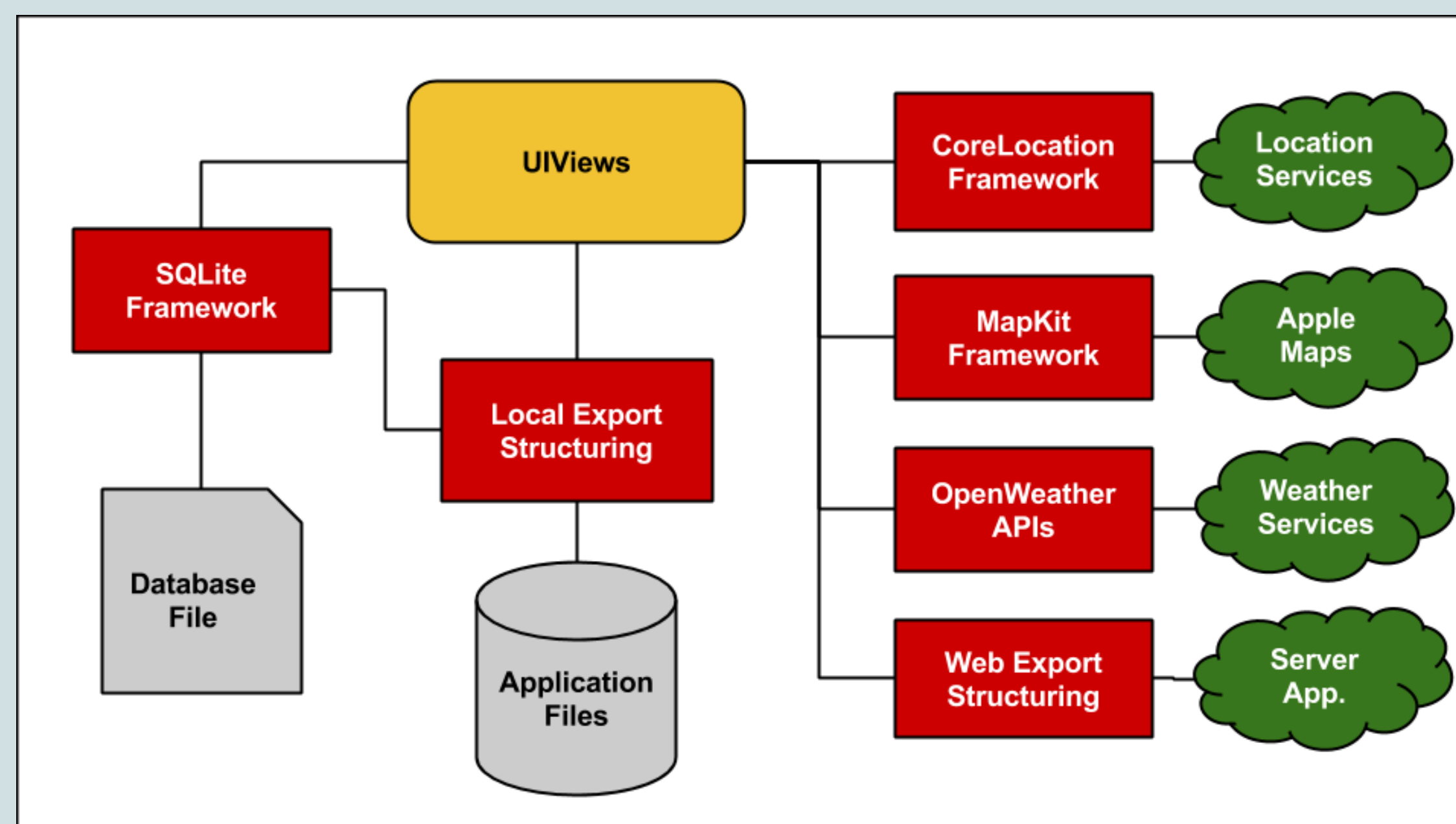
\*Photo taken and information summarized from poster of senior design team 06, December 2013: <http://butterflies.eco.iastate.edu/files/ubr-poster.pdf>

### Android

- ❖ User interface through Android Activity subclasses
- ❖ UI events trigger actions by intermediate services, adapters, or providers
- ❖ Intermediaries write data directly to application or device storage
- ❖ Views update information using Observer functions

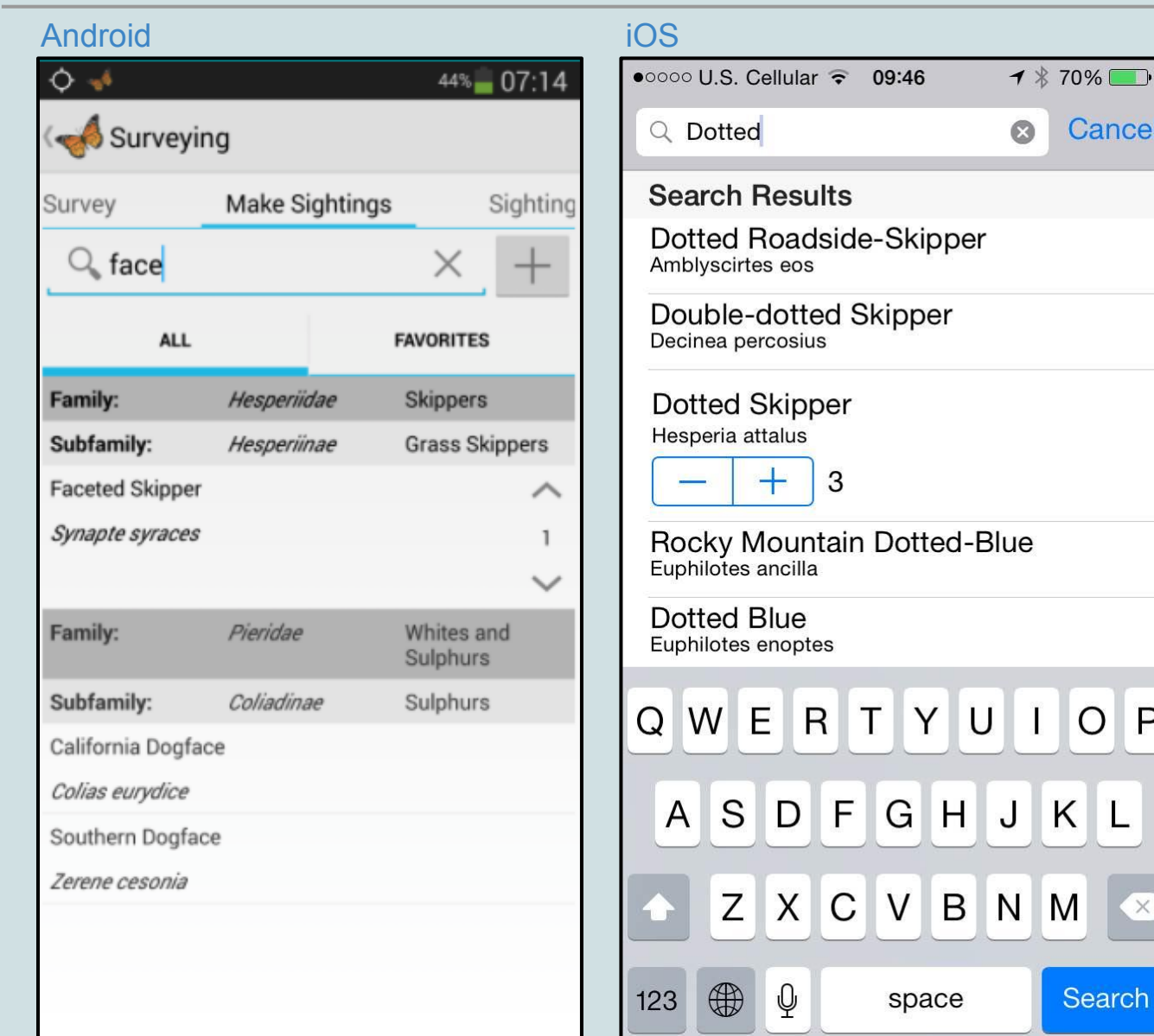
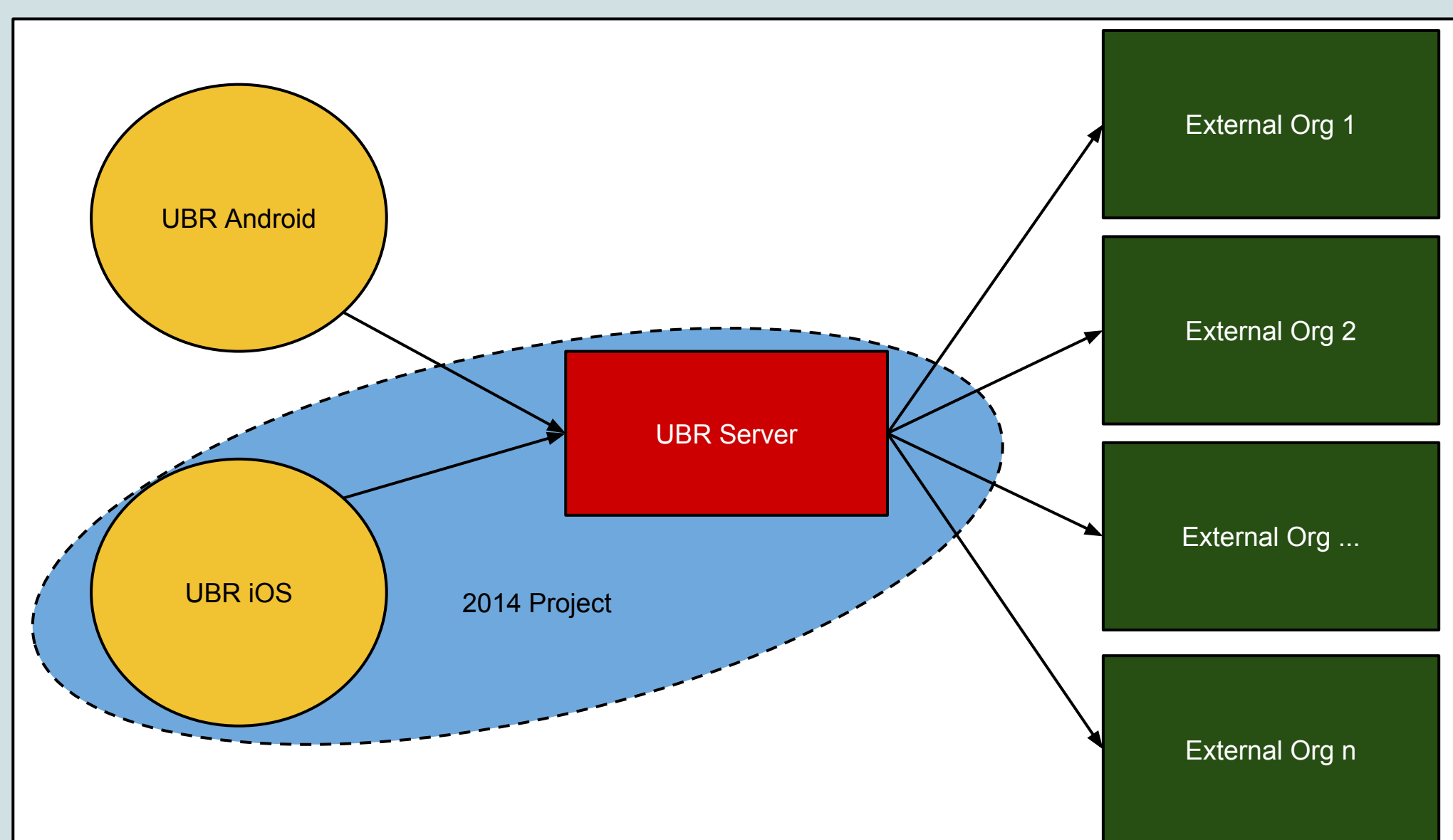
### iOS

- ❖ User Interface through iOS UIView subclasses
- ❖ UI events trigger methods provided by Apple Frameworks (e.g. CoreLocation)
- ❖ Writing data is restricted to application filesystem
- ❖ Separate frameworks for location and mapping
- ❖ Views update automatically as they appear



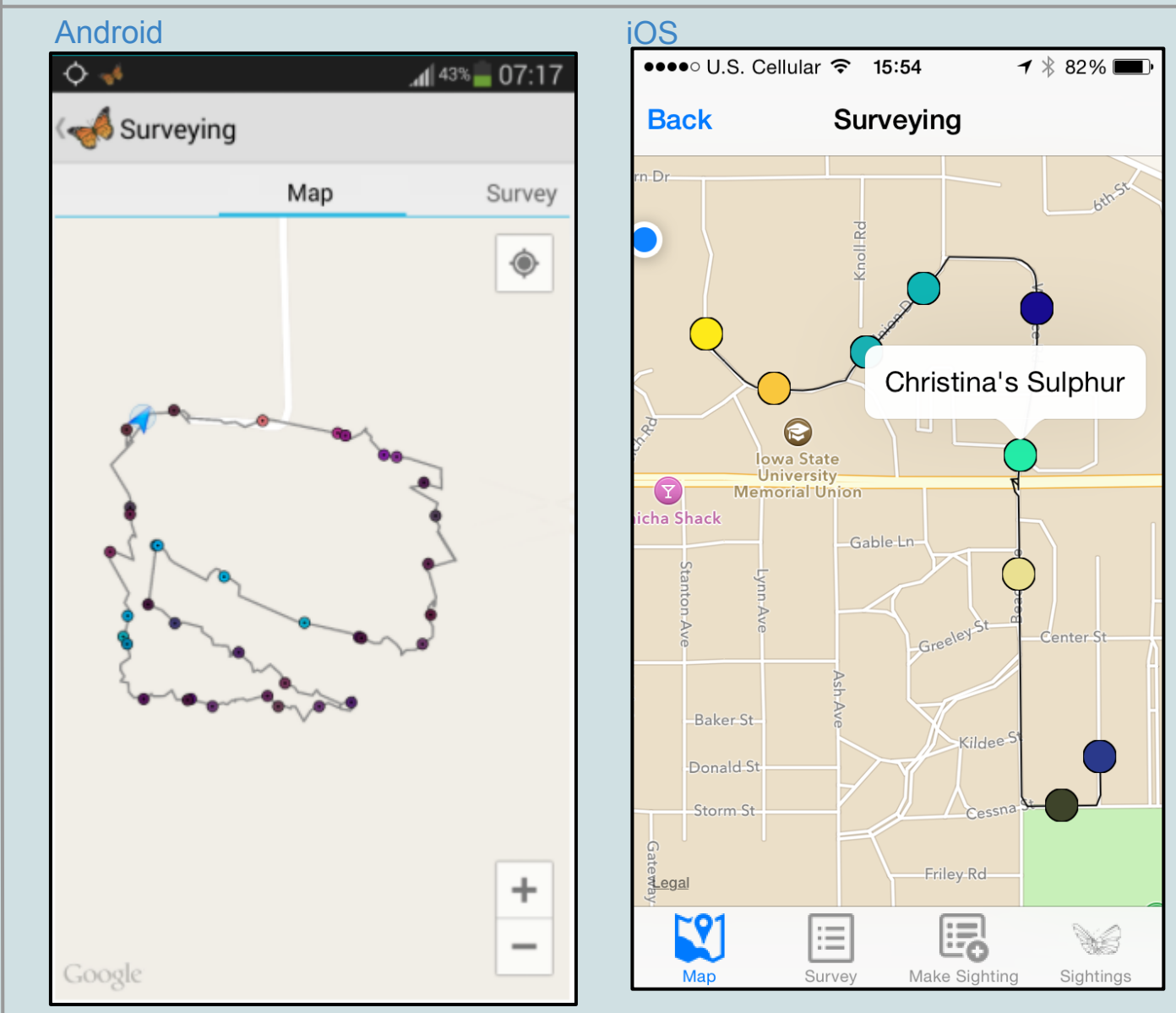
### External Interface

- ❖ An app sends data to the UBR Server
- ❖ The UBR Server delegates information to one or more user-selected butterfly databases
- ❖ A database will specify how it will receive the data



### Make Sighting

- ❖ Subtle interface changes
- ❖ Modified cell layout



### Map

- ❖ Less precise GPS
- ❖ Information callouts

\*Android screenshots taken from poster of senior design team 06, December 2013: <http://butterflies.eco.iastate.edu/files/ubr-poster.pdf>

## Design and Development Requirements

### Functional

- Automatic data collection using device sensors for both surveys and sightings
- Online data collection using location services
- Support for multiple survey protocols
- Data export to device storage, iCloud, server

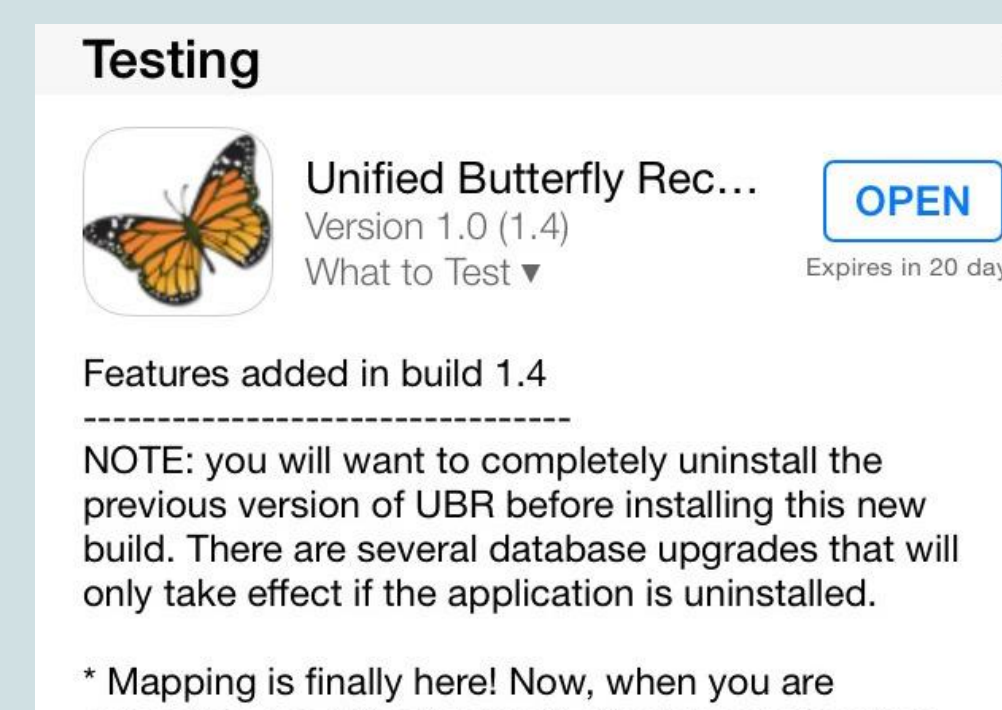
### Non-Functional

- Stay consistent to the Android workflow design
- Maintain Apple Human Design standards
- Preserve consistent user experience across platforms
- Ensure similar database schemas and data collection between apps

## Development and Testing

- ❖ Agile Kanban board using Trello
- ❖ Weekly stand-up meeting with Advisor
- ❖ Comprehensive code reviews

- ❖ Weekly releases post-MVP
  - Descriptive release notes
  - Releases field tested
  - 15 users granted beta testing access
  - 4 organizations interested in consumption of data
- ❖ Client meetings weekly to receive feedback from community



## Tools and Frameworks

- iOS development tool - Xcode 6
- SQLite framework for database emulation
- OpenWeather APIs
- MapKit services through Apple Maps
- Integrated beta testing with TestFlight

