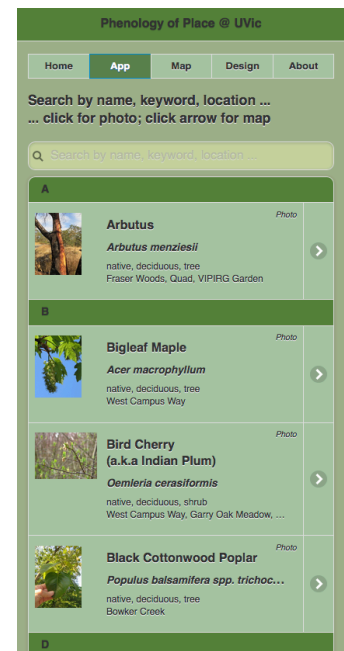
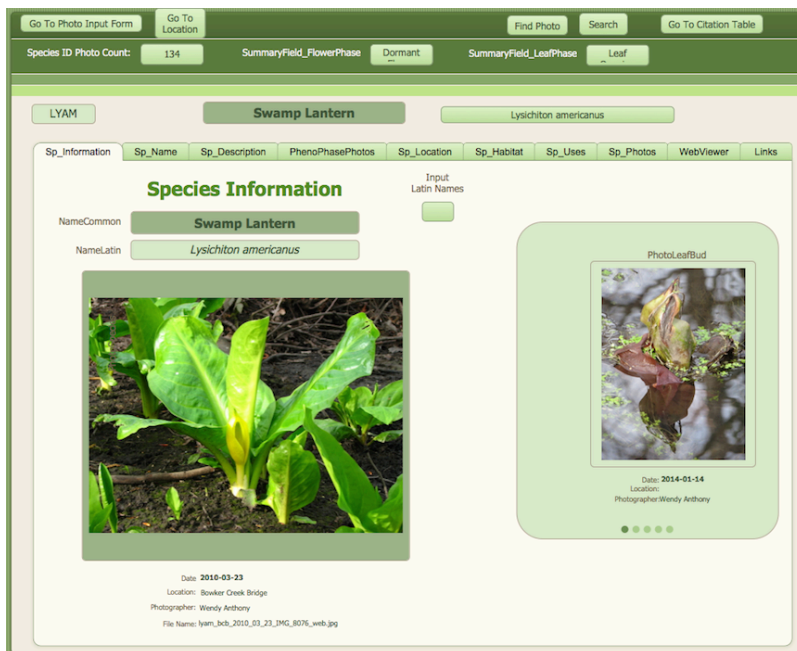


# A Project Manual: Using Native Plant Photographs and Digital Technology as Tools for Education and Research for a Phenology of Place at UVic



By Wendy Anthony  
for University of Victoria ES 490 Directed Studies pt. 2  
April 2015

Supervisors:  
Dr. Eric Higgs, Environmental Studies  
Dr. Yvonne Coady, Computer Science

## Introduction

***“Do not go where the path may lead,  
go instead where there’s no path and leave a trail.”  
- Ralph Waldo Emerson***

This Phenology of Place project was originally created as an Environmental Studies Directed Studies (see Appendixes I & II), to document growth stages of native plants growing at University of Victoria (UVic) campus, using digital photographs, and their seasonal data, to determine if any seasonal growth patterns emerge (see Appendix III – Questions that helped drive this research project).

A photo-based digital database was created using FileMaker Pro 13, and a 1000 sample photos from 5 years of native plant photos were imported, and organized by species, location and photo date in order to make these photos and data both useful and accessible, particularly in an on-line format. The data was exported in HTML and JSON formats into mobile-friendly web templates, to make the data available as a tool for online education and research. The companion document *User Testing: Native Plant Field Guide App for a Phenology of Place at UVic* provides illustrations and challenges for each of the Field Guide App versions developed during the design process, though this manual has not been user-tested.

As much of the necessary software procedures and coding information were new to me, and not always easy to find, this project manual will document the methods, procedures, code scripts and paths I’ve taken, leaving a breadcrumb trail to revisit some of the steps used to develop a database and a simple, mobile-friendly field guide to identify native plants at UVic.

## Table of Contents

|  |           |
|--|-----------|
| <b>Introduction</b> .....  | <b>ii</b> |
| <i>“Do not go where the path may lead,</i> .....                           | <b>ii</b> |
| <b>Table of Figures</b> .....  | <b>v</b>  |
| <b>I. Phenology</b> .....  | <b>1</b>  |
| <b>Phenology Definition</b> .....  | <b>1</b>  |
| <b>Potential Uses for Phenological Data</b> .....                          | <b>1</b>  |
| <b>Plant Phenological Phases</b> .....                                     | <b>1</b>  |
| <b>II. Digital Photographs as Citizen Science Data</b> .....               | <b>2</b>  |
| <b>Photography of Native Plants</b> .....                                  | <b>2</b>  |
| <b>Repeat Photography</b> .....  | <b>2</b>  |
| <b>Digital Technology</b> .....  | <b>2</b>  |
| <b>Citizen Science</b> .....   | <b>2</b>  |
| <b>III. FileMaker Pro Database Instruction Manual</b> .....                | <b>3</b>  |
| <b>Description of the FMP Database</b> .....                               | <b>3</b>  |
| <b>Set up FMP Database structure</b> .....                                 | <b>3</b>  |
| <b>How To Use the FMP Database</b> .....                                   | <b>3</b>  |
| <b>Managing Tables &amp; Layouts in FMP Database</b> .....                 | <b>4</b>  |
| Photo Input Table Layout in FMP Database.....                              | <b>4</b>  |
| Species Table in FMP Database.....   | <b>5</b>  |
| Species Photo Tab & Portal in FMP Database.....                            | <b>6</b>  |
| <b>Photo processing for FMP Database</b> .....                             | <b>7</b>  |
| File-naming Protocol.....  | <b>7</b>  |
| Photo Sizes.....   | <b>7</b>  |
| Creating .csv for importing EXIF data for FMP .....                        | <b>7</b>  |
| <b>Adding New Records to FMP Database</b> .....                            | <b>7</b>  |
| Creating A New Species Record in FMP .....                                 | <b>7</b>  |
| Creating A New Location in FMP.....  | <b>7</b>  |
| Importing Photos to FMP Database .....                                     | <b>8</b>  |
| Adding the Phenophase to FMP Database: .....                               | <b>8</b>  |
| <b>Code for Automatically Populating Data Fields in FMP Database</b> ..... | <b>9</b>  |
| <b>FileMaker Pro Scripts</b> .....   | <b>10</b> |
| <b>Code for Exporting FMP Data</b> .....                                   | <b>11</b> |
| Exporting GeoJSON data from FMP.....                                       | <b>13</b> |
| <b>Adding Weather &amp; Day-Length Data to FMP Database</b> .....          | <b>13</b> |
| Create a multi-year weather data .csv file.....                            | <b>13</b> |
| 1) Download Daily Weather Data Reports.....                                | <b>13</b> |
| 2a) Create a .csv file using Excel: .....                                  | <b>14</b> |
| 2b) Create a .csv file using Firefox SQLite Manager Add-On: .....          | <b>14</b> |
| 3) Create a Daily Sun Rise and Sun Set Table .csv file.....                | <b>14</b> |
| 4) Import Weather Data & Sun Rise, Sun Set Data into FMP database.....     | <b>14</b> |

|   |           |
|---|-----------|
| <b>IV. Native Plant Field Guide App Instruction Manual.....</b>               | <b>15</b> |
| <b>Description of the Field Guide App.....</b>                                | <b>15</b> |
| <b>How To Use The Field Guide App.....</b>                                    | <b>15</b> |
| <b>Responsive Web Design For the Field Guide App .....</b>                    | <b>16</b> |
| <b>User Interface &amp; Accessibility for Field Guide App.....</b>            | <b>16</b> |
| <b>The Code Used in the Field Guide App .....</b>                             | <b>17</b> |
| HTML5 .....   | 17        |
| CSS3 .....  | 17        |
| JavaScript.....   | 17        |
| jQuery.....   | 18        |
| jQuery Mobile .....   | 18        |
| <b>Mobile-friendly Web Page HTML Code For Embedded Google Maps .....</b>      | <b>18</b> |
| <b>Mobile-friendly Web Page HTML Code For Multiple Page .....</b>             | <b>19</b> |
| <b>Creating a Simple, Mobile-Friendly Image &amp; Data-based Web App.....</b> | <b>20</b> |
| <b>Species Information in Field Guide App.....</b>                            | <b>21</b> |
| <b>Using Photos in the Field Guide App .....</b>                              | <b>22</b> |
| Web Gallery: How to Photo Map the landscape with close-ups - using VUE .....  | 22        |
| Variations for Viewing Field Guide Photos.....                                | 23        |
| Image Slider / Slide Show .....   | 23        |
| <b>Using Google Maps in the Field Guide App .....</b>                         | <b>24</b> |
| Google Maps JavaScript API.....   | 24        |
| Create a Placeholder for Google Map in HTML Code.....                         | 24        |
| Info Window (one marker point) .....  | 25        |
| Info Window (populated by GeoJSON file) .....                                 | 26        |
| Centering the Google Maps for Responsiveness.....                             | 27        |
| Google Maps & GeoJSON Data.....   | 27        |
| Finding the User's Location on Google Maps .....                              | 28        |
| Geolocation Script (W3C 2014).....  | 28        |
| <b>V. Creating a Directed Studies Proposal.....</b>                           | <b>29</b> |
| <b>VI. Creating a Concept Map Using VUE .....</b>                             | <b>29</b> |
| <b>VII. Lessons &amp; Wish Lists .....</b>                                    | <b>31</b> |
| <b>Skills &amp; Knowledge Gained Through Phenology Project.....</b>           | <b>31</b> |
| <b>Suggestions For Wish List .....</b>  | <b>31</b> |
| <b>VIII. Challenges .....</b>   | <b>32</b> |
| <b>IX. Conclusions .....</b>  | <b>33</b> |
| <b>X. References .....</b>  | <b>34</b> |
| <b>Appendix I: Directed Studies Proposal pt. 1.....</b>                       | <b>36</b> |
| <b>Appendix II: Directed Studies Proposal pt. 2 .....</b>                     | <b>37</b> |
| <b>Appendix III: Questions That Helped Drive This Project.....</b>            | <b>38</b> |
| <b>Appendix IV: Concept Map for Phenology of Place Pt. 2.....</b>             | <b>39</b> |
| <b>Appendix V: Sequence Diagram for Field Guide App .....</b>                 | <b>40</b> |



## Table of Figures

|   |    |
|---|----|
| <i>Cover: FileMaker Pro Species Table; Native Plant Field Guide App</i> .....               | i  |
| <i>Figure 1: FileMaker Pro database Photo Input Layout Form.</i> .....                      | 4  |
| <i>Figure 2: FileMaker Pro database Species Table</i> .....                                 | 5  |
| <i>Figure 3: FileMaker Pro Database Species Photos Portal.</i> .....                        | 6  |
| <i>Figure 4 a,b,c: Native Plant Field Guide App Prototype.</i> .....                        | 16 |
| <i>Figure 5: Mobile-friendly web app prototype.</i> .....                                   | 20 |
| <i>Figure 6: Species info added to photo page.</i> .....                                    | 21 |
| <i>Figure 7: Species info populated by JSON.</i> .....                                      | 21 |
| <i>Figure 8: Web Gallery: Photomapping the Landscape with magnified close-ups.</i> .....    | 22 |
| <i>Figure 9: Phenophase Image slide-show navigated with arrows.</i> .....                   | 23 |
| <i>Figure 14: Directed Studies Proposal pt. 1 Pro Forma</i> .....                           | 36 |
| <i>Figure 15: Directed Studies Proposal pt. 2 Pro Forma</i> .....                           | 37 |
| <i>Figure 16: Concept Map for Phenology of Place Native Plant Field Guide App</i> .....     | 39 |
| <i>Figure 17: Sequence Diagram shows timing of interaction of objects and events.</i> ..... | 40 |

## I. Phenology

### Phenology Definition

- Phenology is the study of the seasonal events in the life cycles of plants and animals.
- A phenophase is an observable stage in the annual life cycle of a plant, consisting of a few days or weeks, with a start and an end point.
- When using phenophase categories to document seasonal changes in a species, common terms need to be used in order to make them comparable and exportable to other databases.
- Creating a glossary to explain the terms, and stages used for different species will be helpful for users new to phenological concepts.

### Potential Uses for Phenological Data

- Creating a seasonal calendar of the leafing and flowering stages of native plants.
- Monitoring and documenting seasonal changes in a systematic manner to contribute data for both education and research purposes.
- Analyzing phenological data to help determine if any seasonal patterns are apparent.
- Comparing phenological data to weather and day length data can help to determine if there is any correlation between patterns. This data analysis may be of interest to those studying the affect of climate change on local plant species.
- Documenting climate change, using by observing any biological responses to changing weather patterns, to determine if shifts of phenological cycles can be useful indicators.

### Plant Phenological Phases

- Document the date and growth phases of the leaf and flower stages using common terms to ensure that the data can be shared for research purposes.
- Provide definitions, and example photos to ensure that the correct phenophase is noted, in a consistent manner.
- Each species may have different growth phases due to the plant type.
  - e.g. evergreen needles vs deciduous leaves
- **Leaf Growth Phases:** dormant, bud swell, bud break, opening, growing, full leaf, bud set, leaf colour, leaf fall, n/a, unknown.
- **Flower Growth Phases:** dormant, bud form, bud splits, bud open, flower open, full flower, flower dying, dead flower, fruit forming, fruit ripe, fruit decaying, fruit fall, n/a, unknown.

## II. Digital Photographs as Citizen Science Data

### Photography of Native Plants

- Getting good close-up shots may involve being willing to bend down, walk off the path, and keep coming back to get document the phenophase changes.
- The best time of day is either in the early morning, or late afternoon, when the lighting is best, and the wind is likely to be calmer.

### Repeat Photography

- Take repeat photos of the same plant, and if a tree or shrub, the same branch
- Return at 2-3 day intervals during the growing season to document changes in the growth stages of the leaves and flowers.
- Use the same camera location, the same camera settings (focal length and aperture, camera height), and preferably even the same camera model and photographer to ensure consistency; line up the edges of the photo with the same plants and landscape features as the preceding photos; take a photo of the photographer and camera setup, or at least the camera location.
- Shoot during the same time of day.

### Digital Technology

- Mobile smart phone and tablet devices, have on-board GPS geo-locators, and high-definition digital cameras.
- The digital landscape has changed to a higher proliferation of mobile devices that will require more thoughtful, responsive web design to accommodate a variety of users and tasks.

### Citizen Science

- When people engage with the natural environment around them, they become more familiar with seasonal changes, and may also have a tendency to care more about environments that have some regular contact with.
- As scientists need research data to analyze for potential patterns, they are increasingly turning to citizen scientist volunteers to participate in collecting digital data.
- Phenology photos need to be taken at regular, short intervals, requiring a team of knowledgeable volunteers about native plant phenophases, who are willing to choose a location and plant species to monitor, returning to take regular photographs over the seasons.
- The data would need to be submitted through some web app, with some method of validating the data.
- Providing a Field Identification Guide for the Native Plant species, along with photos of the phenophase stages to be monitored would help ensure the documented phenophase would use consistent standards of measurement, in order to make it useful for future research purposes.

## III. FileMaker Pro Database Instruction Manual

### Description of the FMP Database

- The database was created to organize a series of Native Plant photographs, with naming, date and location protocols, from which to research seasonal patterns in the growth stages of native plants at the UVic campus.
- FileMaker Pro 13 (FMP), is the relational database, which was used to create a database, consisting of tables of related data, which can be manually input, or imported as external .csv or image files.
- The Phenology of Place Photo database consists of 3 tables: Photo Data, including an embedded Google Maps, automatically marking the photo location for each record (see Figure 1), Species Data (see Figure 2), and Location Data.
- Individual records are created for each photo (corresponding to spreadsheet rows) by importing the photo and its EXIF data, and different input or output layouts can be created using fields (corresponding to spreadsheet columns) specific to the purpose of the layout, and in a manner best suited to the user.
  - Buttons can be used as a quick link between different layout
- Layouts can include a 'Portal', which shows records from related tables.
  - e.g. A species record can include a portal which can include data from each photo that is of this particular species (see Figure 3)
- Coded Scripts can be used for repeatedly used tasks (see page 9)

### Set up FMP Database structure

- Add Tables, and create relationships between tables
- Add Layouts specific to the tasks performed
- Add Tabs to reduce visual clutter, and organize fields
- Add Portals to view related records
  - e.g. Show which photos are all of the same species
- Add Image Containers
- Add button links to other Layouts

### How To Use the FMP Database

- All fields can be searched, and sorted by multiple values.
- All photos are geo-located, with the lat-long coordinates automatically populating an embedded Google Map, which is part of each record

## Managing Tables & Layouts in FMP Database

### Photo Input Table Layout in FMP Database

- This layout consists of three tabs (see Figure 1):
  - The default front tab is for Photo data input, where after the photo is imported, the filename and photo EXIF data automatically populate most of the fields; the user only needs to input data in the yellow boxes, for phenophase, and associated field data
  - Google Maps location
  - Large Photo Detail & Phenophases

The screenshot shows a FileMaker Pro database form for 'Red Flowering Currant' (Ribes sanguineum). The form is divided into several sections:

- Photo Data Input:** Fields for PhotoID (873), Year (2014), Month (07), Day (18), Date (2014-07-18), Time (2:38:41 PM), Season (Summer), and Year-Day (199).
- Map:** A photo of a flowering currant bush.
- Large Image:** Fields for File Name (risa\_img\_2014\_07\_18\_IMG\_7447.web.jpg), Original (risa\_img\_2014\_07\_18\_IMG\_7447.jpg), File Name (IMG\_7447.jpg), and File Path (file:///ADATA/HD710/Pictures/Canon Photos/!!UVic-Native-Plants-Phenology-Project/Photos\_In\_Database/WebPhotos-in-DB/risa\_img\_2014\_07\_18\_IMG\_7447.web.jpg). EXIF data includes EXIF AV: 4, EXIF FN: 4, EXIF Exp: 1/400, EXIF Focal Length: 4.5, EXIF ISO: 160, and EXIF SS: 8.643856.
- Location:** Fields for LocationID (irmg), SpeciesID (risa), Plant Nickname, Common Name (Red Flowering Currant), Latin Name (Ribes sanguineum), LocDesc, LocName (Ian Ross Memorial Garden DTB), and Eco-Type.
- Coordinates:** Lat calc (48.464888), Long calc (-123.314995), and Lat / Long (48.4648881).
- Phenology:** Fields for Flower Phase (Fruit Ripe), Flower #, Flower %, Leaf Phase (Leaf Colour), Leaf #, Canopy %, and Dead or Alive?
- Associated Data:** Fields for Associated PlantsID, InvasiveID, Animal AssocID, Bird AssocID, Insect AssocID, and Phenomena AssocID.
- Metadata:** CreatedBy (Wendy Anthony), On (2014-07-22), ModifiedBy (Wendy Anthony), On (2014-07-28).

Figure 1: FileMaker Pro database Photo Input Layout Form.

## Species Table in FMP Database

- Each species in the database has its own separate record in the Species table.
- The Species Layout consists of multiple tabs to organize the data and reduce visual clutter (see Figure 2).
  - Species information, Name, Description, Phenophase Photos in a Portal (see Figure 3), Location, Habitat, Uses, web links to species information, and a Web Viewer with embedded web pages linked to the particular species in two different online databases
- When in Photo Input Layout, clicking the button "Go To Species Table" will open the Species Table Layout to the same Species as in the Photo Input Layout.
- Species Layout Species Photos Tab will show all the photos in the database for that specific species.

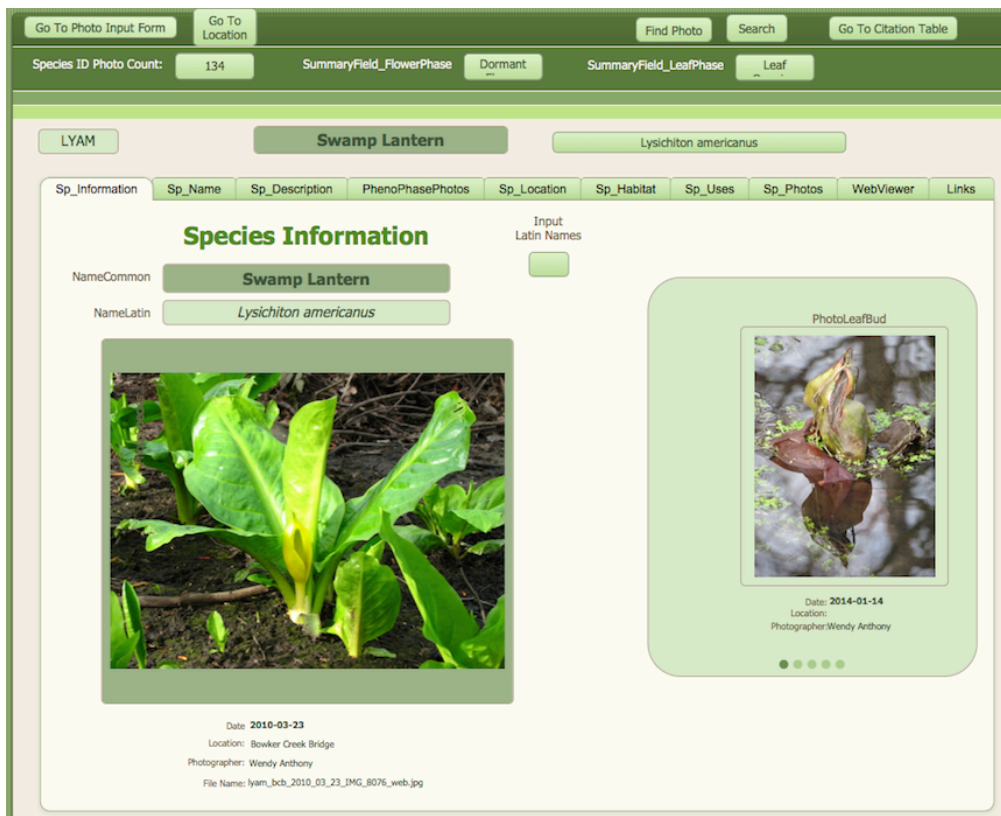


Figure 2: FileMaker Pro database Species Table.

## Species Photo Tab & Portal in FMP Database

- The species table phenophases tab contains a 'portal' view tool, which imports data fields from each of the photos that documents that particular species (see Figure 3)
- The List View of the portal is scrollable, totaling the total number of species listed, and showing a thumbnail of the image, as well as a button linking to a larger image

Phenology of Place at UVic with Native Plant Photography by Wendy Anthony

MAAQ Tall Oregon-grape Mahonia aquifolium






|   |  |   |
|---|--|---|
|    | Year-Day: 59<br>Year: 2011<br>Photo ID: 455<br>Date: 2011-02-28<br>Time: 1:15:09 PM<br>Season: Late Winter<br>Leaf Phase: Leaf Growing<br>Flower Phase: Flower Bud | File name: maaq_img_2011_02_28_IMG_3093_web.jpg<br>Location: Ian Ross Memorial Garden DTB<br>Location Description:<br>EXIF AV: 4 EXIF Focal Length: 37.8<br>EXIF Exp: 1/800 EXIF ISO:<br>EXIF FN: 4 EXIF SS: 9.643856           |
|   | Year-Day: 77<br>Year: 2014<br>Photo ID: 90<br>Date: 2014-03-18<br>Time: 6:44:11 PM<br>Season: Late Winter<br>Leaf Phase: Dormant Leaf<br>Flower Phase: Flower Bud  | File name: maaq_img_2014_03_18_IMG_4804_web.jpg<br>Location: Ian Ross Memorial Garden DTB<br>Location Description:<br>EXIF AV: 4.643856 EXIF Focal Length: 25.3<br>EXIF Exp: 1/25 EXIF ISO: 800<br>EXIF FN: 5 EXIF SS: 4.643856 |
|  | Year-Day: 93<br>Year: 2014<br>Photo ID: 532<br>Date: 2014-04-03<br>Time: 3:49:24 PM<br>Season: Spring<br>Leaf Phase:<br>Flower Phase:                              | File name: maaq_fw_2014_04_03_IMG_6015_web.jpg<br>Location: Fraser Woods<br>Location Description:<br>EXIF AV: 4 EXIF Focal Length: 4.5<br>EXIF Exp: 1/60 EXIF ISO: 80<br>EXIF FN: 4 EXIF SS: 5.906891                           |
|  | Year-Day: 93<br>Year: 2014<br>Photo ID: 533<br>Date: 2014-04-03<br>Time: 3:51:43 PM<br>Season: Spring<br>Leaf Phase:<br>Flower Phase:                              | File name: maaq_fw_2014_04_03_IMG_6027_web.jpg<br>Location: Fraser Woods<br>Location Description:<br>EXIF AV: 4 EXIF Focal Length: 4.5<br>EXIF Exp: 1/100 EXIF ISO: 160<br>EXIF FN: 4 EXIF SS: 6.643856                         |
|  | Year-Day: 174<br>Year: 2014<br>Photo ID: 225<br>Date: 2014-06-23<br>Time: 12:53:47<br>Season: Summer<br>Leaf Phase:<br>Flower Phase:                               | File name: maaq_dtbp_2014_06_23_IMG_2002_web.jpg<br>Location: David Turbin Building Parking Lot<br>Location Description:<br>EXIF AV: 4 EXIF Focal Length: 4.5<br>EXIF Exp: 1/80 EXIF ISO: 80<br>EXIF FN: 4 EXIF SS: 6.321028    |

Figure 3: FileMaker Pro Database Species Photos Portal.

## Photo processing for FMP Database

### File-naming Protocol

- The file name is used to automatically populate many data fields (see page 10), including: Common Name, Location, and Date. Correct naming is the first step in creating valid, unique data records.
  - SpeciesCode\_LocationCode\_Year\_Month\_Day\_IMG#.jpg
  - lyam\_bc9\_2014\_07\_17\_IMG1234.jpg

### Photo Sizes

- Resize each photo into multiple sizes, one as a thumbnail, one kept at high resolution for details, and a series of web sizes based on the device screen.

### Creating .csv for importing EXIF data for FMP

- Use Adobe Bridge for extracting the EXIF data of each photo into a .csv file.
- Use the folder of new named web images before adding them to the folder of all photos in database.
  - Download script for creating csv file in Bridge (PS-Scripts, 2008)
    - To install unzip and place this script into Bridge's script folder:
    - Edit - Preferences -Startup Scripts
    - At the bottom click the "Reveal Button" this will open the folder where the script should be placed.
    - To use: Mouse Right click (menu) and select "CSV Data Collector"
    - Select a folder for the output file, enter filename (including extension)
    - Just tick the boxes of the EXIF fields to be extracted (e.g. Date Created, Exposure Time, FNumber, ISO Speed, Shutter Speed, Aperture Value, GPS Latitude, GPS Longitude)
- The csv file of EXIF data can be imported into the FMP Database Settings:
  - Update matching records (match filename); Don't import first record

## Adding New Records to FMP Database

### Creating A New Species Record in FMP

1. Create a New Record in the Species Layout
2. Update the Value List for Species names
3. In the Photo Input Layout > Manage Database > Common Name > add the new SpeciesCode and the Common Name
  - The database automatically calculates the Common Name from file name

### Creating A New Location in FMP

1. Create a New Record in the Location Layout
2. Update the Value List for Location
3. In the Photo Input Layout > Manage Database > LocName > add the new LocationCode and the Location Name
  - The database automatically calculates the Location Name from file name



### Importing Photos to FMP Database

1. Copy database to newly named file for a backup copy, before making changes
2. Use file-naming protocol to accurately name image files
3. Make sure all photos have lat/long info before adding them
  - a) GeoTag software: locate photo on map and assign coordinates
4. Create different sizes: original (good for focusing on details), web (1024 x 768 pixels), thumbnail (180 x 135 pixels)
5. Add new photos to appropriate folders
6. Note number of records in DB, and how many should be after finishing import
7. Use Bridge to write metadata script for importing EXIF data from new image files in folder, before adding photos to full database folder
8. Import .csv file of EXIF data, matching csv file headers to database fields
9. File > Import Records > Folder > Specify > Choose Folder > WebPhotos-in-DB > Continue > As Above
10. Move new files to usual web folder, and update path in Database
11. File > Import Records > File > Update matching records in found set; Don't import first record (match to FileName field)

### Adding the Phenophase to FMP Database:

- Each photo must have the leaf and flower phenophase growth stages noted in the database, by choosing from the drop-down lists.
- Viewing the large version of the photo shows detail to determine the phenophase growth stages; the drop-down list fields have been added to the large photo tab layout in order to more easily see and document growth stage details.
  1. Go to PhotoInput Layout > Choose Large Image Tab
  2. Choose Leaf and Flower phenophase from drop-down list
    - **Leaf Growth Phases:** dormant, bud swell, bud break, opening, growing, full leaf, bud set, leaf colour, leaf fall, n/a, unknown.
    - **Flower Growth Phases:** dormant, bud form, bud splits, bud open, flower open, full flower, flower dying, dead flower, fruit forming, fruit ripe, fruit decaying, fruit fall, n/a, unknown.
  3. Change to the Photo Data Input tab to add any field notes.
    - e.g. associated plants or animals, weather, or human activities.
- To enable sorting, the phenophases are also given a numeric value, which is automatically calculated based on the chosen phenophase.
  - **Leaf\_Phase#:** Case ( Leaf\_Phase = "Dormant\_Leaf"; "1"; Leaf\_Phase = "Leaf Bud Swell"; "2"; Leaf\_Phase = "Leaf Bud Break"; "3"; Leaf\_Phase = "Leaf Opening"; "4"; Leaf\_Phase = "Leaf Growing"; "5"; Leaf\_Phase = "Leaf Full"; "6"; Leaf\_Phase = "Leaf Bud Set"; "7"; Leaf\_Phase = "Leaf Colour"; "8"; Leaf\_Phase = "Leaf Fall"; "9"; Leaf\_Phase = "n/a"; "0" ; Leaf\_Phase = "??"; "0" )
  - **Flower\_Phase#:** Case ( Flower\_Phase = "Dormant Flower"; "1"; Flower\_Phase = "Flower Bud Form"; "2"; Flower\_Phase = "Flower Bud Splits"; "3"; Flower\_Phase = "Flower Bud Open"; "4"; Flower\_Phase = "Flower Open"; "5"; Flower\_Phase = "Full Flower"; "6"; Flower\_Phase = "Flower Dying"; "7"; Flower\_Phase = "Flower Dead"; "8"; Flower\_Phase = "Fruit Forming"; "9"; Flower\_Phase = "Fruit Ripe"; "10"; Flower\_Phase = "Fruit Decaying"; "11"; Flower\_Phase = "Cone"; "12"; Flower\_Phase = "??"; "0" )

## Code for Automatically Populating Data Fields in FMP Database

- The filename and photo EXIF data are used to automatically populate some fields.
- **Filename:** SpeciesCode\_LocationCode\_Year\_Month\_Day\_IMG#.jpg
  - lyam\_bc9\_2014\_07\_17\_IMG1234.jpg
- **Date**

The filename includes the date (year-mo-dy), which can be separated using FMP's MiddleWords code format. Numeric months can be sorted, though names are easier for users; Season were given numeric values to facilitate sorting, and the day of the year was assigned to each season to enable seasonal sorting.

  - **DateYear:** MiddleWords (FileName; 3; 1)
  - **DateMonth:** MiddleWords (FileName; 4; 1)
  - **DateDay:** MiddleWords (FileName; 5; 1)
  - **Date:** DateYear & "-" & DateMonth & "-" & DateDay
  - **DateMonth\_text:** Case(DateMonth="01"; "January"; DateMonth = "02"; "February"; DateMonth="03"; "March"; DateMonth = "04"; "April"; DateMonth = "05"; "May"; DateMonth = "06"; "June"; DateMonth = "07"; "July"; DateMonth = "08"; "August"; DateMonth = "09"; "September"; DateMonth = "10"; "October"; DateMonth = "11"; "November"; DateMonth = "12"; "December")
  - **DateSeason\_txt:** Case ( DateSeason# = "1" ; "Early Winter"; DateSeason# = "2"; "Late Winter"; DateSeason# = "3"; "Spring"; DateSeason# = "4"; "Summer"; DateSeason# = "5"; "Autumn"; DateSeason# = "6"; "Early Winter" )
  - **DateSeason#:** Case(DateYearDay="1...35"; "1"; DateYearDay="36...80"; "2"; DateYearDay="81...171"; "3"; DateYearDay="172...263"; "4"; DateYearDay="264...354"; "5"; DateYearDay="355...365"; "6";)
- **Location**

Each location is given a unique code, which is included in the file name.

  - **LocationID:** MiddleWords (FileName; 2; 1)
  - **LocName:** Case ( LocationID = "BC9"; "Bowker Creek Parking Lot 9"; LocationID = "ATGH"; "Alumni Trail / Gordon Head Rd"; LocationID = "BCB"; "Bowker Creek Bridge";)
- **Species**

Each species is given a unique four-digit character code based on the first two letters of the Genus name, and the first two letters of the species name. The code is included in the file name, and populates the Common Name field.

  - **CommonName:**Case (Species\_ID = "ARME"; "Arbutus"; Species\_ID = "RUDI"; "Himalayan Blackberry"; Species\_ID = "RUUR"; "Trailing Blackberry";)
- **Camera EXIF data**

Includes Date and Time, Lat/Long Coordinates, elevation data, Camera make, model, and photographer's name, which can be used to populate field values.
- **Value Lists**

Using drop-down lists populated with accepted values help reduce input errors: Common\_Names, Invasives, Leaf\_Phase, Location\_ID, Location\_Name, Month, Phenomena\_Assoc, Photo\_Quality, Photo\_View, Seasons, Slope, Species\_ID, Year

## FileMaker Pro Scripts

- For repeatedly used code, FMP Scripts can be invoked through the script menu, or by embedding the code in a clickable Button.
- When writing script code, add commands one line at a time, and test first to ensure results are as expected before adding more lines of code, Choose code commands from side bar list in Script window, and choose Layouts and fields from drop-down lists.
- Scripts can be created for: Printing to pdf, Searching by date, species, etc, Sorting by species, date, season, location, phenophase, and Exporting Records.
- **Automatically Input Season Field Name in FMP (based on YearDay)**
  1. Go to Layout [ "AutoPhoto\_Input\_Form\_Map\_AutoImport"  
(PhotoImport\_2014) ]
  2. Enter Find Mode [ ]
  3. Perform Find [ Specified Find Requests: Find Records; Criteria:  
PhotoImport\_2014::DateYearDay: "1...35" ]
  4. [ Restore ]
  5. Go to Field [ PhotoImport\_2014::DateSeason# ]
  6. Replace Field Contents [ Replace with calculation: "1" ]
  7. Enter Find Mode [ ]
  8. Modify Last Find
  9. Perform Find [ Specified Find Requests: Find Records; Criteria:  
PhotoImport\_2014::DateYearDay: "36...80" ] [ Restore ]
  10. Go to Field [ PhotoImport\_2014::DateSeason# ]
  11. Replace Field Contents [ Replace with calculation: "2" ]
  12. Enter Find Mode [ ]
  13. Modify Last Find
  14. Perform Find [ Specified Find Requests: Find Records; Criteria:  
PhotoImport\_2014::DateYearDay: "81...171" ] [ Restore ]
  15. Go to Field [ PhotoImport\_2014::DateSeason# ]
  16. Replace Field Contents [ Replace with calculation: "3" ]
  17. Enter Find Mode [ ]
  18. Modify Last Find
  19. Perform Find [ Specified Find Requests: Find Records; Criteria:  
PhotoImport\_2014::DateYearDay: "172...265" ] [ Restore ]
  20. Go to Field [ PhotoImport\_2014::DateSeason# ]
  21. Replace Field Contents [ Replace with calculation: "4" ]
  22. Enter Find Mode [ ]
  23. Modify Last Find
  24. Perform Find [ Specified Find Requests: Find Records; Criteria:  
PhotoImport\_2014::DateYearDay: "266...355" ] [ Restore ]
  25. Go to Field [ PhotoImport\_2014::DateSeason# ]
  26. Replace Field Contents [ Replace with calculation: "5" ]
  27. Enter Find Mode [ ]
  28. Modify Last Find
  29. Perform Find [ Specified Find Requests: Find Records; Criteria:  
PhotoImport\_2014::DateYearDay: "356...366" ] [ Restore ]
  30. Go to Field [ PhotoImport\_2014::DateSeason# ] Replace Field  
Contents [ Replace with calculation: "6" ]

## Code for Exporting FMP Data

- Data can be exported from FileMaker Pro Database in a variety of formats: exporting an Excel .xlsx file through the FMP menu File Export Records, or HTML, JSON or GeoJSON file exports through coded fields where the record data automatically populates code templates webpage elements.

### Export FMP Data to Excel

- CSV format is useful for importing data into spreadsheets (e.g. Excel) or other databases. Exported .csv files from FileMaker pro do not contain column header information, making the data of little use when exported.
  - Export data as an .xlsx file, open in Excel, and export as .csv file from Excel. This exported csv file will include the column headers.
- Export .csv file from Excel**
  - Open FMP layout with records to export
  - File > Export Records > Excel Worksheet (.xlsx) > Specify Field Order For
  - Export (File > Save Records As does NOT include header)
  - Open .xlsx file in Excel > File > Save As > .csv

### HTML (*Hyper-Text Markup Language*)

- HTML5 templates can be created for web pages (see also page 17).
- HTML code scripts can be created within FileMaker Pro using the relevant record fields as data to populate HTML template code.
- FMP HTML Code template used to populate a WebGallery:**

```
"<a name=\"\" & GetAsText ( DateSeason_txt_pi ) & GetAsText ( DateYear_pi ) & "\"></a>
<a href=\"http://web.uvic.ca/~wanthony/phenology/images/\" & GetAsText ( File Name ) & "\">
<img src=\"http://web.uvic.ca/~wanthony/phenology/images/\" & GetAsText ( File Name_tb ) & "\" width=\"180\" height=\"135\" title=\"\"
& GetAsText ( Photo_AltTitle_pi ) & "\" alt=\"\" & GetAsText ( Photo_AltTitle_pi ) & "\" class=\"imgtb\" border=\"1\"></a><br>
& GetAsText ( DateMonth_text_pi ) & " " & GetAsText ( DateDay_pi ) & ", "
& GetAsText ( DateYear_pi ) & "<br>(& GetAsText ( LocationID_pi ) & ")"
```

### JSON (*JavaScript Object Notation*)

- JSON is a data exchange format, readable by both humans and computers, is formatted in pairs of "Key" : "Value", which are separated by commas, and is enclosed within curly brackets. The data can also contain nested sets of key : values. No comments are allowed in JSON file format.
- Add JSON script to a database field so each record is automatically converted to JSON object format. Be sure to validate JSON before using on webpage.
- FMP JSON Data Template** formatted to export Species-related data. (Note that the quotation marks needed to be preceded by an escape character, as JavaScript uses quotes for coding purposes).

```
"{ \"name\" : \"\" & GetAsText ( NameCommon ) & "\",
  \"nameAlt\" : \"\" & GetAsText ( NameAlt1 ) & "\",
  \"latin\" : \"\" & GetAsText ( NameLatin ) & "\",
  \"division\" : \"\" & GetAsText ( NameDivision ) & "\",
  \"family\" : \"\" & GetAsText ( NameFamily_Common ) & "\",
```

```

\"familyLatin\" : \"\" & GetAsText ( NameFamily_Latin ) & "\",
\"speciesID\" : \"\" & GetAsText ( SpeciesID ) & "\",
\"speciesName\" : \"\" & GetAsText ( NameSpecies ) & "\",
\"speciesSub\" : \"\" & GetAsText ( NameSubSpecies ) & "\",
\"speciesFlower\" : \"\" & GetAsText ( Species_Flowering_Plants ) &
 "\",
\"speciesGroup\" : \"\" & GetAsText ( Species_Group ) & "\",
\"speciesNonVascular\" : \"\" & GetAsText (
Species_Non_Vascular_Plants ) & "\",
\"speciesVascular\" : \"\" & GetAsText ( Species_Vascular_Plants ) &
 "\",
\"location\" : \"\" & GetAsText ( LocUVic ) & "\",
\"habitat\" : \"\" & GetAsText ( Habitat ) & "\",
\"type\" : \"\" & GetAsText ( PlantType ) & "\",
\"evergreenDecid\" : \"\" & GetAsText ( EvergreenDeciduous ) & "\",
\"invasive\" : \"\" & GetAsText ( PlantInvasive ) & "\",
\"leafShape\" : \"\" & GetAsText ( LeafShape ) & "\",
\"leafStemNum\" : \"\" & GetAsText ( LeafPerStem# ) & "\",
\"flowerColour\" : \"\" & GetAsText ( FlowerColour ) & "\",
\"flowerPetal\" : \"\" & GetAsText ( FlowerPetal ) & "\",
\"flowerPetalNum\" : \"\" & GetAsText ( FlowerPetal# ) & "\",
\"flowerShape\" : \"\" & GetAsText ( FlowerShape ) & "\",
\"fruitShape\" : \"\" & GetAsText ( Fruit_Shape ) & "\",
\"fruitColour\" : \"\" & GetAsText ( Fruit_Colour ) & "\",
\"fruitType\" : \"\" & GetAsText ( Fruit_Type ) & "\",
\"usePart\" : \"\" & GetAsText ( UsePart ) & "\",
\"useTrad\" : \"\" & GetAsText ( UseTraditional ) & "\",
\"useWarn\" : \"\" & GetAsText ( UseWarning ) & "\", "

```

## GeoJSON

- GeoJSON is a specialized version of JSON, containing coordinate information
- Longitude is listed before Latitude, which is opposite to the usual formatting that is recognized by Google Maps API (Butler et. al., 2008).
- GeoJSON cannot contain nested key : values.
- Example GeoJSON code format:

```

{ "type" : "FeatureCollection", "features" : [
  { "type" : "Feature",
    "geometry" : { "type" : "Point" ,
      "coordinates" : [ long, lat ],
      "name" : "pointName " }, },
  { "type" : "Feature",
    "geometry" : { "type" : "LineString" ,
      "coordinates" : [ [long, lat ], [long, lat ],
        [long, lat ] ]
      "name" : "LineStringName " },},
  { "type" : "Feature",
    "geometry" : { "type" : "Polygon" ,
      "coordinates" : [ [ [long, lat ], [long, lat ],
        [long, lat ] ]
      "name" : "poLygonName " } }]}

```

## Exporting GeoJSON data from FMP

- GeoJSON code can be exported from FileMaker Pro database to be used to populate the Google Maps Info Windows.
- Procedure for Exporting GeoJSON data from FMP
  1. Layout GeoJSON > Table View (sort by common name, year, yearday)
  2. File > Export Records > tab file (prevent added double quotes in .csv)
  3. Rename file with .json extension
  4. Add to beginning of file: {"type": "FeatureCollection", "features": [
  5. Add to end of file: ]}
  6. Validate the JSON file using jsonlint.com (JSONLint, 2015).
- Example of a GeoJSON template populated from data in each FMP record with the (field name) with brackets, using the FMP command GetAsText

```
{ "type" : "Feature", "geometry" : { "type" :
"Point", "coordinates" : [ " & GetAsText ( Long_c ) & ",
" & GetAsText ( Lat_c ) & " ] }, "properties" :
{ "name" : "\"" & GetAsText( CommonName ) & "\", \"latin\"
: "\"" & GetAsText ( LatinName ) & "\", \"location\" : "\"" &
GetAsText ( LocationID ) & "\", \"locName\" : "\"" &
GetAsText ( LocName ) & "\", \"date\" : "\"" & GetAsText (
Date ) & "\", \"season\" : "\"" & GetAsText ( DateSeason_txt
) & "\", \"imgID\" : "\"" & GetAsText ( File Name_origin )
& "\", \"filename\" : "\"" & GetAsText ( File Name ) & "\",
\"filenameetb\" : "\"" & GetAsText ( File Name_tb ) & "\",
\"leafphenophase\" : "\"" & GetAsText ( Leaf_Phase ) &
 "\", \"flowerphenophase\" : "\"" & GetAsText ( Flower_Phase )
& "\", \"flowercolour\" : "\"" & GetAsText (
Species_Table::FlowerColour ) & "\", \"minTempC\" : "\"" &
GetAsText ( UVic_weather_risaset_2009_2014::tempMinC ) &
 "\", \"maxTempC\" : "\"" & GetAsText (
UVic_weather_risaset_2009_2014::tempMaxC ) & "\",
\"sunRise\" : "\"" & GetAsText (
UVic_weather_risaset_2009_2014::SunRise ) & "\", \"sunSet\"
: "\"" & GetAsText ( UVic_weather_risaset_2009_2014::SunSet )
& "\", \"dayLightHours\" : "\"" & GetAsText (
UVic_weather_risaset_2009_2014::DayLightHours ) & "\", } }
```

## Adding Weather & Day-Length Data to FMP Database

### Create a multi-year weather data .csv file

Weather and daylength data can be imported as csv file into either Excel or Firefox SQLite

#### 1) Download Daily Weather Data Reports

Temperature and precipitation data is downloaded from Government of Canada Climate Data website, choosing the year, and csv data type (Government of Canada, 2015).

**2a) Create a .csv file using Excel:**

1. Import the downloaded .csv file into Excel
2. Remove the meta data from the top and save in a meta data file
3. Each yearly csv file can be appended to the end of the Excel file to create a multi-year weather data excel file.
4. Export the excel file as a .csv file.

**2b) Create a .csv file using Firefox SQLite Manager Add-On:**

1. In Firefox browser Tools Menu > Add-ons > Search add-ons for SQLite Manager > Click to install
2. Create and name a new database, and save file as UVicWeather.csv
3. Create a table with the following structure using the SQL statement:  

```
CREATE TABLE "UVic_DailyWeather_1993_2013" ("station" CHAR, "lat" FLOAT, "long" FLOAT, "elev" FLOAT, "date" DATETIME, "year" INTEGER, "monthID" INTEGER, "day" INTEGER, "tempMaxC" FLOAT, "tempMinC" FLOAT, "tempMeanC" FLOAT, "dd_0" FLOAT, "dd_U" FLOAT, "rain_mm" FLOAT, "snow_cm" FLOAT, "prec_mm" FLOAT, "snow_cm_g" FLOAT)
```
4. Import each year's .csv file to the table
5. Export as .csv file

**3) Create a Daily Sun Rise and Sun Set Table .csv file**

- **Access the Daily Sun Rise and Sun Set Tables** for 2014 the from National Research Council Canada Sunrise/sunset calculator website (National Research Council, 2015).
- **Import Daily Sun Rise and Sun Set Data to Excel**
  1. Copy year table to a text file, save as VicSunRiseSet-2014.txt
  2. In Excel > File > Import > Text File > Import > Text Import Wizard: Original DataType: Fixed Width > Import
  3. Remove extra rows; Rename Columns: Month, Day, SunRise, SunSet, DayLightHours
  4. Save As .csv file

**4) Import Weather Data & Sun Rise, Sun Set Data into FMP database**

1. To add YearDay.csv > Create .csv file in Excel > File > Import Records (Don't Import First Record; Match Records Based On: Year, Month, Day, Import YearDay>YearDay; Update Matching Records in Found Set) > File > Import Field Mapping > New Table > Import > File > Manage Relationships > Fields > Change Data Type > Add Year Day Field >
  2. Import 2014 UVic Weather data .csv into Excel > remove meta data from top of file > Change field names to be same as previous files: > Update Matching Records (Match Records Based On: Year, Month, Day,)
- Create a new Table for Weather and Day Length
    - Create Fields for: Year, YearDay, Day, Month (numeric for sorting), MonthID (name), Precipitation (mm), TemperatureMinimumC, TemperatureMaximumC, TemperatureMeanC, SunRise (Time), SunSet (Time), DayLength (hours)
    - Create a Relationship to PhotoImport table using the YearDay field

## IV. Native Plant Field Guide App Instruction Manual

### Description of the Field Guide App

- This mobile-friendly web app can help to identify native plants at the UVic campus, using digital photographs, and data-embedded code exported from a database of the photographs.
- This web-based application could be used for both educational and research purposes, with images used to identify the native plant species.
- Google Maps shows some species location based on photograph locations.
- The original target audience has been identified as being UVic students (Environmental Studies, Biology, Forestry, Environmental Restoration), restoration volunteers, outdoor enthusiasts interested in learning the identification of local native plants, and newcomers to the University or area. A sample of users have been involved in testing the app, during the design phase, in order to make sure the app is both useful and easy to use.
- A collection of digital photographs of native plants, taken between 2009 and 2015, were organized within a FileMaker Pro database to extract the data contained within the photos, and then exported in code scripts to be used in web page templates, along with JavaScript code, to create a functional web application design, responsive to a variety of computer devices sizes, including smart phones, tablets, laptops, and desktop computers.

### How To Use The Field Guide App

- The current, most stable working version (see Figure 4 a) is a Mobile-friendly design, and is responsive to different screen sizes. It includes a search bar, detailed and searchable species info, and links to an image, and a location marked on an embedded Google Map (see Figures 4 a,b,c).
- Using the JavaScript library JQuery Mobile to create a List View of different species includes a search function, and large button sizes for use with fingers and a mouse.
- The first species image (Arbutus) is actually loaded as part of the original page, which reduces page load time, though each of the other species are linked to images which are hosted on a separate webpage for each species (see Figure 4b).
- The map is linked to a separate file, which is populated by a GeoJSON file, exported from the database, for coordinates and species information (see Figure 4c).
- The sequence Diagram (see Appendix V) helps visualize the timing and interactions between various elements of the app.



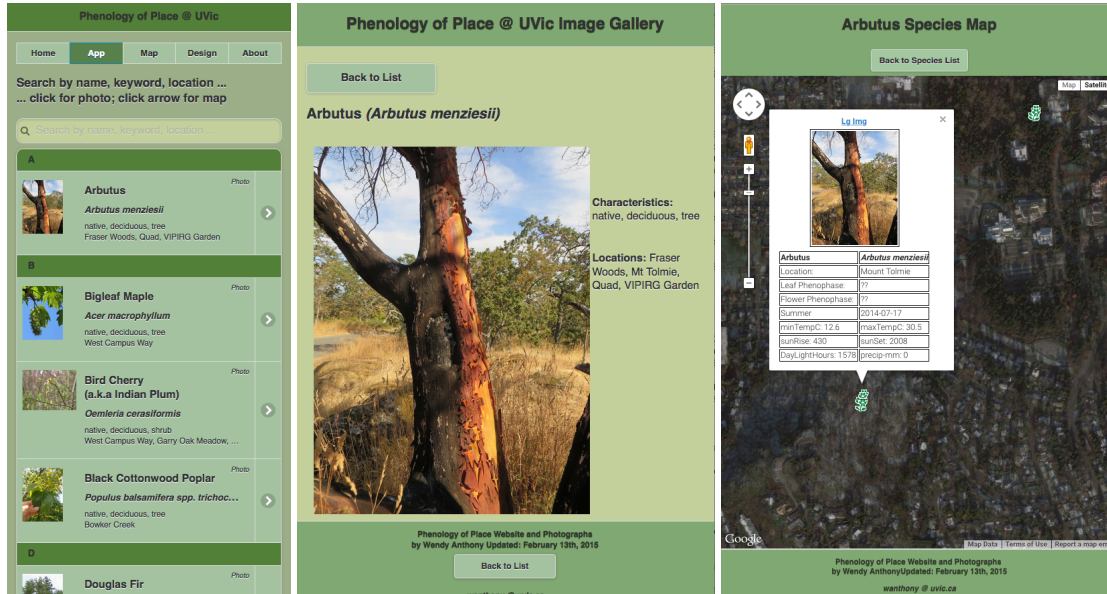


Figure 4 a,b,c: Native Plant Field Guide App Prototype.

## Responsive Web Design For the Field Guide App

- A responsive web page is accessible on a variety of device screen sizes. Building responsiveness into the design avoids the need to create multiple versions for each device. When creating code, the design needs to address both the layout and the user interaction (jQuery Mobile - rwd, n.d.).
  - Use units that are flexible, instead of absolute, to allow the HTML elements, images and media, to adapt to the size of the container.
  - Use CSS media queries, with multiple versions of `min-width` attribute, to set layout breakpoint rules for different screen widths and resolutions,
    - e.g. image stays a reasonable size, and the fonts remain legible.
  - Design for mobile versions first, before adding progressive enhancements and content for larger screen sizes, to ensure that a website will work effectively on the smaller screen sizes of mobile devices. Additional content, larger images and different navigation options can be progressively added to websites if intended to be viewed on larger devices.
  - Responsive web pages will respond to different screen size reducing the need for user to do a sideways scroll.

## User Interface & Accessibility for Field Guide App

- Backgrounds and font colour should consider user needs and abilities.
- Alt and Title (tool tips) image tags help screen readers for vision impairment.
- Navigation menus need to be usable in a variety of device screen sizes, and may need to be tailored to include different items on smaller devices.
- Buttons and menu items need to be of large enough size to be used with fingers (instead of a mouse) on mobile devices.
- Responsive design code will ensure the app is viewable in various screen sizes.
  - TV, desktop, laptop, tablet, mobile phone, smart watch, Future sizes
- Add Back Buttons for user to easily return to main page

## The Code Used in the Field Guide App

### HTML5

- Hyper Text Markup Language creates the structure of the webpage.
- Source code for a webpage can be viewed using the browser tools.
- Comments can be added to the code to help explain code use, without appearing on the web page `<!-- HTML comments are formatted like this -->`
- All HTML5 documents source code begins with `<!DOCTYPE html>` to alert the browser that the code follows the HTML5 coding protocols.
- Include within the `<head></head>` tags
  - **Meta tags:** (jQuery Mobile, Pages, no date).
    - Viewport ensures that regardless of the device width, the webpage will cover the full width of the device screen
    - The Charset specifies the character set in a meta tag for browsers to understand the code
  - **Link Tags:** CSS style sheets are linked to from within the link tag
  - **Script Tags:** Java Script source files can be linked to from within the script tag, or added to end of body, to load webpage before the script
- Structural elements are defined with `<div></div>`, and can be manipulated with JavaScript code.

### CSS3

- Cascading Style Sheets creates the style format for the webpage.
- External Style Sheets are referenced to within the `<head><link></link></head>` section.
  - Page-specific style code is added within a `<style></style>` element
  - Use id (unique identifier can only be used for one element) or class (identifier can be used for more than one element on the webpage).
- Comments can be added to the css code to help explain the usage.
  - `/* multiple CSS`  
`line comments */`      `// single line CSS comments`
- Changing an element in the style sheet will change all webpage elements that this style pertains to.
- Media queries can be used to change style depending on maximum width needed. e.g. Set image maximum width to 100% to fill the image container

### JavaScript

- Put the reference to .js files in the `<head>` section of the html page.
- When browser finds the `<script></script>` element, a request is sent to the server to access the url for the hosted .js file.
- JavaScript and JSON files must be located on the same server as the web page to prevent any Cross Browser security issue.
- Place the local JavaScript code just before the closing body tag.
- `/* multiple JavaScript`  
`line comments */`      `// single line JavaScript comments`

## jQuery

- The jQuery Library is needed to use the jQuery Mobile, and is listed first within the <head></head> section of the webpage.
  - <script src="http://jquery.js "></script>

## jQuery Mobile

- The jQuery Mobile library file is used to design mobile-friendly webpages, and is linked to from the <head></head> section, after the jQuery library reference.
  - jQuery Mobile has it's own .css file to link to
    - <link rel="stylesheet" href="http://jquery.mobile.css" />
  - Theme Roller can be used to create customized themes, for creating colour combinations, font family, corner radii of boxes, header, footer, body, links, buttons, which are downloaded as a .css file
    - <link rel="stylesheet" href="http://ThemeRollerUrl.css" />
  - Library file is linked to in the <head> section of the webpage
    - <script src="http://jquery.mobile.js "></script>
  - Multiple Pages can link from one page (jQuery Mobile – pages, n.d.; see page 19); Although technically only one web page, each page section has its own header and footer, with a back button on page two to easy return to page one.

## Mobile-friendly Web Page HTML Code For Embedded Google Maps

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="utf-8">
<!--Add CSS files before Scripts, Add jQuery script before jQuery Mobile -->
<link rel="stylesheet" href="jquery.mobile-1.4.5.css" />
<link rel="stylesheet" href="http://jquery.mobile.css" />
<link rel="stylesheet" href="http://ThemeRollerUrl.css" />
<script src="jquery-1.11.1.min.js"></script>
<script src="http://jquery.mobile.js "></script>
<style><!-- additional CSS style code can be added here for page-specific code -->
</style>
</head>
<body>
<div data-url="map-page" data-role="page" id="map-page">
<!-- Navigation bars are located in the header (w3schools TryIt, 2015) -->
<div data-role="header">
  <h1>Welcome To My Homepage</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Page Two</a></li>
      <li><a href="#">Search</a></li>
    </ul>
  </div> <!--End of navbar -->
</div> <!--End of header -->
<!-- placeholder for Google Maps Canvas -->
<div role="main" class="ui-content" id="map-canvas">Map Loads Here...
</div> <!--End of embedded Google Maps Canvas -->
</div> <!--End of map-page -->
<script>// JavaScript code specific to the webpage is included near end of body
</script>
</body></html>
```

## Mobile-friendly Web Page HTML Code For Multiple Page

```

<body>
<!-- Start of first page: #pageone -->
<div data-role="page" id="pageone">
  <div data-role="header">
    <h1>Welcome To Page One</h1>
  </div> <!--End of header -->
<div role="main" class="ui-content">
<!-- Search -->
<form class="ui-filterable">
<input id="myFilter" data-type="search" placeholder="Search by name, keyword, location ..."
></form>
<!-- List View -->
<ul data-role="listview" data-filter="true" data-input="#myFilter" data-autodividers="true"
data-inset="true">
<li><a href="#pagetwo" data-ajax="true"><h3>Arbutus</h3> <span class="latin">Arbutus menziesii</span><p
class="spinfo">native, deciduous, tree<br />Fraser Woods, Mt Tolmie, Quad, VIPIRG
Garden</p><p class="ui-li-aside">Photo</p></a><a
href="http://web.uvic.ca/~wanthony/phenology/PoP-map-2b-arme.htm" data-ajax="false">Species
Map</a></li></ul>
<!-- end of List View -->
</div> <!--End of content -->
<!-- footer pageone -->
<div data-role="footer" id="footercolour">
<p>Phenology of Place Website and Photographs by Wendy Anthony
<a href="#pageone" data-direction="reverse" data-role="button" data-theme="a"
class="backb">Back to List</a></p>
</div><!-- end of footer pagetwo -->
</div> <!--End of pageone -->
<!-- Start of second page: #pagetwo -->
<div data-role="page" id="pagetwo" data-theme="a">
<div data-role="header" id="headercolour">
  <h1>Phenology of Place Image Gallery</h1>
</div><!-- end of header pagetwo -->
<div data-role="content" data-theme="a">
<!-- Back Button to pageone -->
<div><a href="#pageone" data-direction="reverse" data-role="button" data-theme="a"
class="backb">Back to List</a></div>
<!-- Species List View -->
  <h2>Arbutus <i>(Arbutus menziesii)</i></h2>
  <div id="pimg"><table><tr>
    <td></td>
    <td><b>Characteristics:</b> native, deciduous, tree<br />
    <b>Locations:</b> Fraser Woods, Quad, Finnerty Parking Lot, VIPIRG Garden, Mt.
Tolmie</td>
  </tr></table> </div>
</div><!-- end of content pagetwo -->
<!-- footer pagetwo -->
<div data-role="footer" id="footercolour">
<p>Phenology of Place Website and Photographs by Wendy Anthony
<a href="#pageone" data-direction="reverse" data-role="button" data-theme="a"
class="backb">Back to List</a></p>
</div><!-- end of footer pagetwo -->
</div><!-- end of pagetwo -->
</body>

```

## Creating a Simple, Mobile-Friendly Image & Data-based Web App

- A simple, mobile-friendly web app can be developed using jQuery Mobile code snippets for creating a List View of items, i.e. Native or Invasive plant species, which can be linked to other elements, like images and/or maps (see Figure 5).

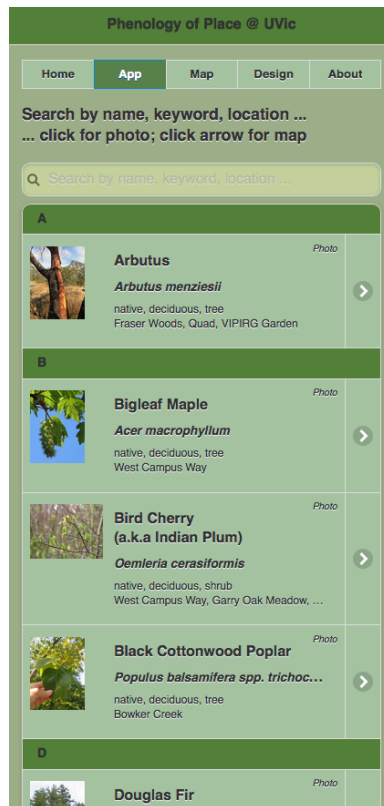


Figure 5: Mobile-friendly web app prototype.

### Create a jQuery Mobile List View

1. Add image tb with link to larger image
2. Add second link to list, at far right side (text in 2<sup>nd</sup> link displays when mouse hovers over link)
 

```
<li><a href= "ImageLg.jpg">ImageSm.jpg</a>
      <a href= " #">Species Info</a>
</li>
```
3. Add location, on separate line so it will show on mobile
4. **Search Filter** (w3schools.com, 2015)
  1. Elements need to declare the attribute `data-filter="true"`
  2. Add placeholder text "Search by name, keyword, location"
5. Add extra info to right side, using class `ul-li-aside` (tends to write over the left side when screen size is at smallest)

## Species Information in Field Guide App

- Species information added to the front page of the app can be searched using the List View search element (see Figure 5).
- Species information added to the photo page cannot be searched from the List View page, though more information can be added to the photo page than can be included in the List View (see Figure 6).
- Species information is added by using a JSON file to populate the webpage, though if the webpage does not use the jQuery Mobile List View structure, the page is very long, and can only be searched using the browser (see Figure 7).

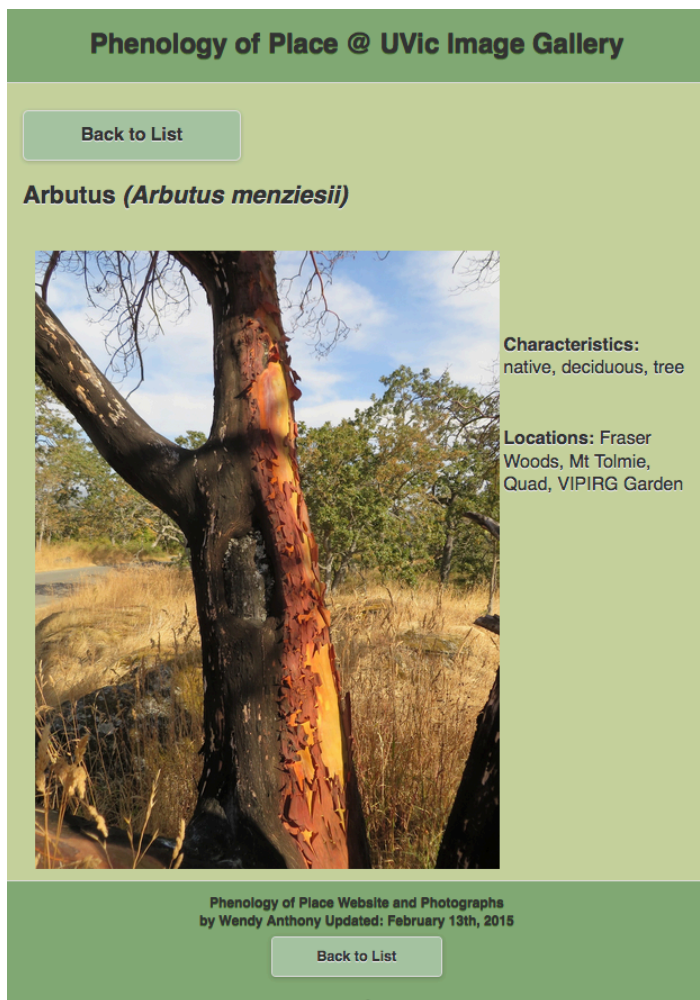


Figure 6: Species info added to photo page.

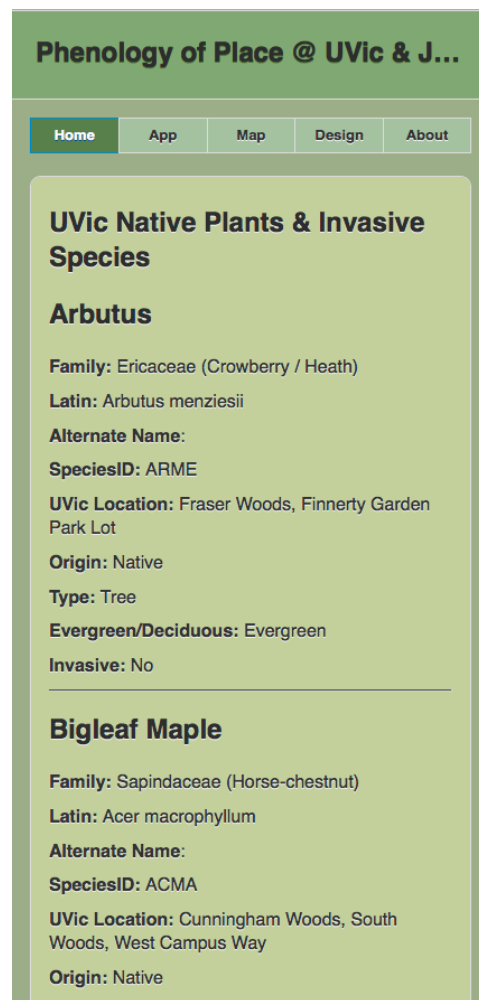


Figure 7: Species info populated by JSON.



## Using Photos in the Field Guide App

- Early attempts to create image galleries resulted in seeds for the idea to create a database, to easily sort plant images by species, date, season, location, and a web-base application, to help identify species and phenophase growth stages.

### Web Gallery: How to Photo Map the landscape with close-ups - using VUE

- One landscape photo can be populated with links to four close-up images, using the VUE concept mapping software (VUE, 2015) (see Figure 8).
- Using the same camera location, without moving, zoom in for close-ups of several of the plants. Use VUE to overlay these zoomed photos as clickable links.



Figure 8: Web Gallery: Photomapping the Landscape with magnified close-ups.

1. Make a VUE map with a static background, linking to photos (see pages 30-31)
2. Make node > insert photo > press "v" > move photo out of node > delete node >
3. Layers > lock key makes background layer (rename) > click eye makes invisible
4. New layer for photos > make node > make fill transparent >
5. Make outline thick bright/contrast to see against background photo
6. Shape square to match photo edges > position node over area of photo to link to larger, close-up image >
7. R-click on node > Add URL to node > reposition node so photo sits nicely over background photo
8. Save as an HTML web page

## Variations for Viewing Field Guide Photos

1. A List View species links to the image file when clicked, with no control over the display, as it is not included on a html page, nor does it have a back button.
2. Link to an image webpage, which includes only one page per image, as well as a back button to navigate back to the List View. If the design or code changes, each separate image page will need to be maintained or changed.
3. To reduce page loading time, jQuery Mobile allows a single HTML webpage to include a multi-page structure, as is done with the Arbutus List View image (jQuery Mobile, Pages, no date).
4. In order to include multiple images of a single species, in a variety of phenophase growth stages, a slide show can be created on the image page.

## Image Slider / Slide Show

- A slide show can show images of different phenophase stages in the growth cycle of one species, to assist with identification during different seasons.
- Only one species has a slideshow, in place of a single image, which is accessed by clicking the Swamp Lantern, a.k.a. Skunk Cabbage List View Item (see Figure 9).
- All photos must be the same dimensions and orientation (portrait or landscape).
- Right and Left arrows are used to navigate between photos, and the back button returns user to the List View.

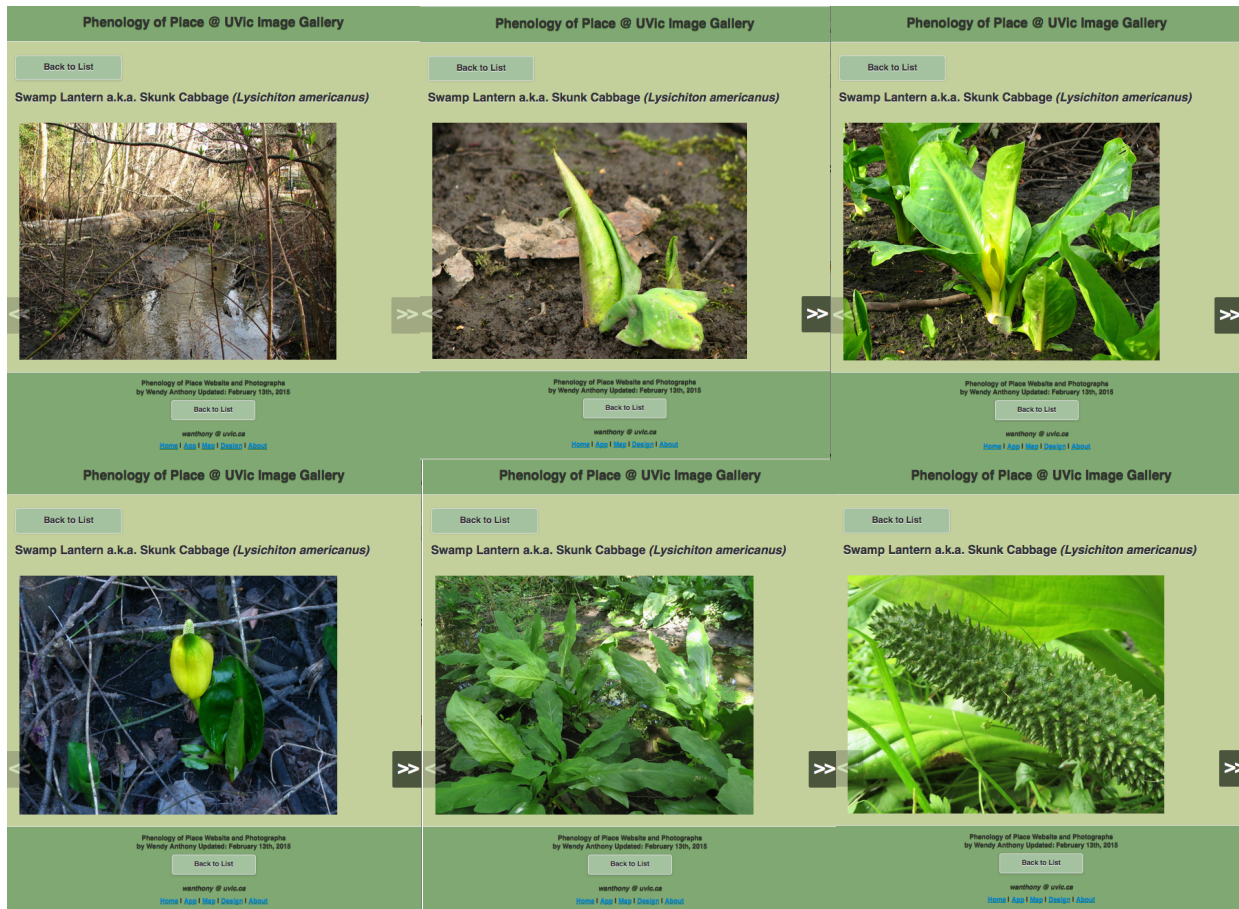


Figure 9: Phenophase Image slide-show navigated with arrows.



### Image Slide Show Source Code for Field Guide App

- In keeping with the Phenology theme, users requested to see images of the species in different leaf and flower phenophase growth stages, in order for the plant to be recognized in different seasons. Swamp Lantern is the only example where the Image webpage contains a slideshow of different phenophases, (see Figure 9).
- A responsive slide show can be created using HTML5, CSS3, and JavaScript (Lee, 2013).
- The slide show is enclosed in a div element, the new HTML5 figure element encloses the figcaption and the previous and next navigation arrow elements.
- The image that will appear when the slideshow is first loaded uses the class “show” in the figure element.
- The images use a percentage width to ensure that the slideshow will be responsive to different device screen widths.
- Left & Right Arrows are used to navigate between the phenophase photos; the Back Button returns to the List View page.
- ```
<div class="diy-slideshow">
<figure class="show">
<figcaption>Swamp Lantern a.k.a. Skunk Cabbage Bowker Creek Parklot 9
2011-03-02</figcaption>
<span class="prev"><<</span><span class="next">>></span></figure>
<figure>
<figcaption>Swamp Lantern a.k.a. Skunk Cabbage Bowker Creek Parklot 9
2011-03-02</figcaption>
<span class="prev"><<</span><span class="next">>></span></figure>
</div>
```

### Using Google Maps in the Field Guide App

#### Google Maps JavaScript API

- Link to the Google Maps JavaScript API in the head section of the HTML document, to help Google Maps load quickly on mobile devices (Google Developers - G, 2015).
- ```
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>
```

#### Create a Placeholder for Google Map in HTML Code

- A `<div></div>` element is created in the html document as a placeholder for the JavaScript to place the Google Maps in the webpage when the code runs.
  - ```
<div data-url="map-page" data-role="page" id="map-page">
<div role="main" class="ui-content" id="map-canvas">Map Loads
Here... </div>
</div>
```

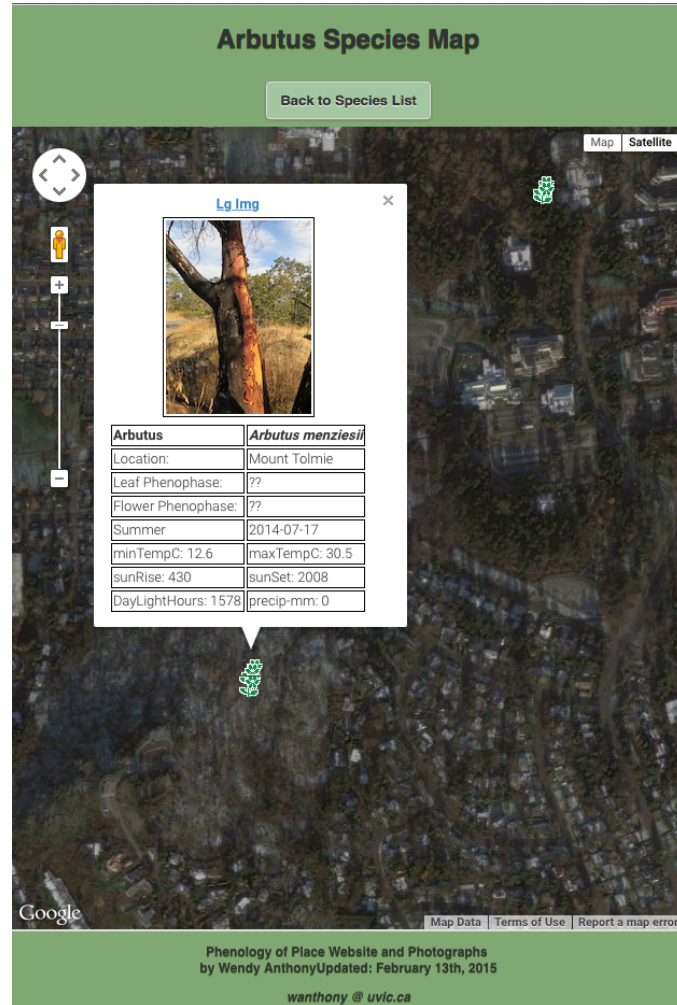
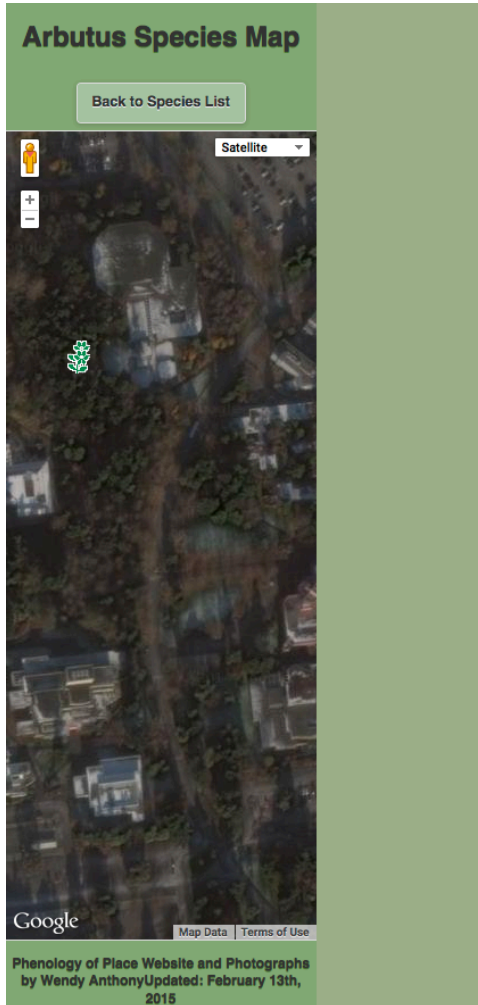


Figure 10: One Species Location Map. Figure 11: Species location map with open Info Window.

### Info Window (one marker point)

- Content String is placed within the initialize function of the JavaScript code (JSFiddle, n.d.) (see Figure 10).

```

<script>
function initialize() {
  var UVicLatLng = new google.maps.LatLng(48.463269, -123.311581);
  var mapOptions = {
    zoom: 15,
    center: UVicLatLng
  };
  var map = new google.maps.Map(document.getElementById('map-
    canvas'), mapOptions);
  var contentString = '<div id="contentStr">' +
    '<h3>UVic</h3>' +
    '<p>content here ...</p>'
    '</div>';
  var infowindow = new google.maps.InfoWindow({
    content: contentString,
    maxWidth: 200
  });
}

```

```

var marker = new google.maps.Marker({
  position: UVicLatLng,
  map: map,
  title: 'UVic'
});
google.maps.event.addListener(marker, 'click', function() {
  infowindow.open(map,marker);
});
}
google.maps.event.addDomListener(window, 'load', initialize);
</script>

```

### Info Window (populated by GeoJSON file)

- A variable var is created to hold the data for each property, then the property is added to the setContent, using "quotes" to enclose any text or html tag (Google Developers - G, 2015; JSFiddle, n.d.) (see Figure 11).
- When adding an img, use 'single quotes' for html code, with the "double quotes" used for the string values.

```

<script>
function initialize() {features = [];map = new
google.maps.Map(document.getElementById('map-canvas'), {zoom: 17,center: {lat:
48.463591, lng: -123.316296},mapTypeId: google.maps.MapTypeId.SATELLITE});
var infowindow = new google.maps.InfoWindow();
map.data.setStyle({icon:'http://web.uvic.ca/~wanthony/phenology/images/garden.png'});
google.maps.event.addListener(map, 'click', function(){infowindow.close();});
map.data.addListener('click', function(event){
var imgtb = event.feature.getProperty("filenametb");
var img = event.feature.getProperty("filename");
var name = event.feature.getProperty("name");
var latin = event.feature.getProperty("latin");
var locName = event.feature.getProperty("locName");
var date = event.feature.getProperty("date");
var season = event.feature.getProperty("season");
var leafph = event.feature.getProperty("leafphenophase");
var flowerph = event.feature.getProperty("flowerphenophase");
var minTempC = event.feature.getProperty("minTempC");
var maxTempC = event.feature.getProperty("maxTempC");var sunRise =
event.feature.getProperty("sunRise");
var sunSet = event.feature.getProperty("sunSet");
var dayLight = event.feature.getProperty("dayLightHrs");
var precip = event.feature.getProperty("precip");

infowindow.setContent("
<div style='width:100%; max-width:350px; text-align:center;'> +
"<table style='border:0px solid black; text-align:center;'> +
"<tr style='text-align: center;'> +
"<td style='border:0px solid black;' colspan='2'> +
"<a href='http://web.uvic.ca/~wanthony/phenology/images/' + img + "' data-
ajax='false'>Lg Img</a> +
"</td> + "</tr> +
"<tr style='text-align: center;'> +
"<td style='border:0px solid black;' colspan='2'> +
"<img style='border:1px solid black; padding:2px;'
src='http://web.uvic.ca/~wanthony/phenology/images/' + imgtb + "' /> + "</td> +
"</tr> +
"<tr style='text-align: left;'> +

```

```

"<td style='border:1px solid black; '>" + "<b>" + name + "</b>" +
"</td>" +
"<td style='border:1px solid black; '>" + "<b><i>" + latin + "</b></i>" + "</td>" +
"</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + "Location: " +
"</td>" +
"<td style='border:1px solid black; '>" + locName +
"</td>" + "</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + "Leaf Phenophase: " + "</td>" + "<td
style='border:1px solid black; '>" + leafph +
"</td>" + "</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + "Flower Phenophase: " + "</td>" + "<td
style='border:1px solid black; '>" + flowerph +
"</td>" + "</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + season + "</td>" +
"<td style='border:1px solid black; '>" + date +
"</td>" + "</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + "minTempC: " + minTempC + "</td>" + "<td
style='border:1px solid black; '>" + "maxTempC: " + maxTempC +
"</td>" + "</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + "sunRise: " + sunRise + "</td>" + "<td
style='border:1px solid black; '>" + "sunSet: " + sunSet +
"</td>" + "</tr>" +
"<tr style='text-align: left; '>" +
"<td style='border:1px solid black; '>" + "DayLightHours: " + dayLight + "</td>" +
"<td style='border:1px solid black; '>" + "precip-mm: " + precip +
"</td>" + "</tr>" + "</table>" + "</div>" );

infowindow.setPosition(event.feature.getGeometry().get());
infowindow.setOptions({pixelOffset: new google.maps.Size(0,-30)});
infowindow.open(map); });

// load the GeoJSON file
map.data.loadGeoJson('http://web.uvic.ca/~wanthony/phenology/googlemap/acma-
geojson.json');

// centre the map for responsiveness (Stack Overflow, 2012)
var center; function calculateCenter() {center = map.getCenter();}
google.maps.event.addDomListener(map, 'idle', function() {calculateCenter();});
google.maps.event.addDomListener(window, 'resize', function(){map.setCenter(center);})
google.maps.event.addDomListener(window, 'load', initialize);
</script>

```

## Centering the Google Maps for Responsiveness

- Centre responsive Google Maps when browser is resized (Stack Overflow, 2012).

## Google Maps & GeoJSON Data

- Add to function initialize(), after all the Google Maps variables and Info Windows HTML code has been added :
- To use the GeoJSON file, make a new one & remove the beginning {curly brackets}, the JSON file must start & end with [square brackets].
  - The JSON file needs to be in the same folder as the html file that calls it

## Finding the User's Location on Google Maps

- HTML5 coding, through the browser, allows the user's location to be shared after asking user permission. Once user consent is given, and the location is found, the latitude and longitude are placed with a marker on a Google Map embedded in the webpage (jQuery Mobile, 2013; McGrath, 2013; Mozilla Developer Network, 2015), (see Figures 12 and 13).
- Users will need to trust the app before allowing their location to be shared, linking to a privacy policy can detail what the location data will be used for.

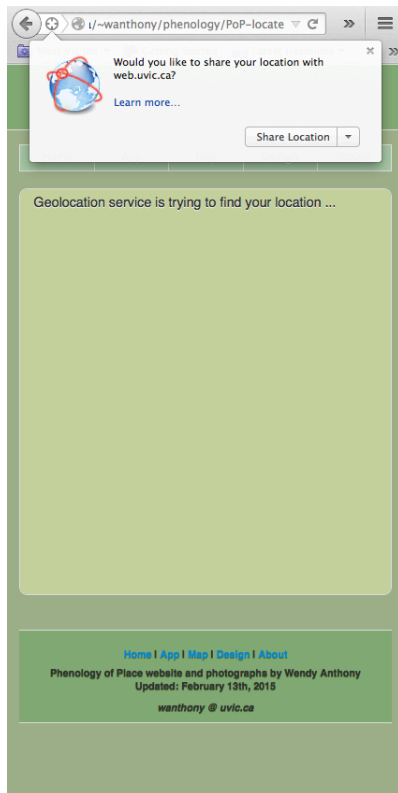


Figure 12: User permission required to share location.

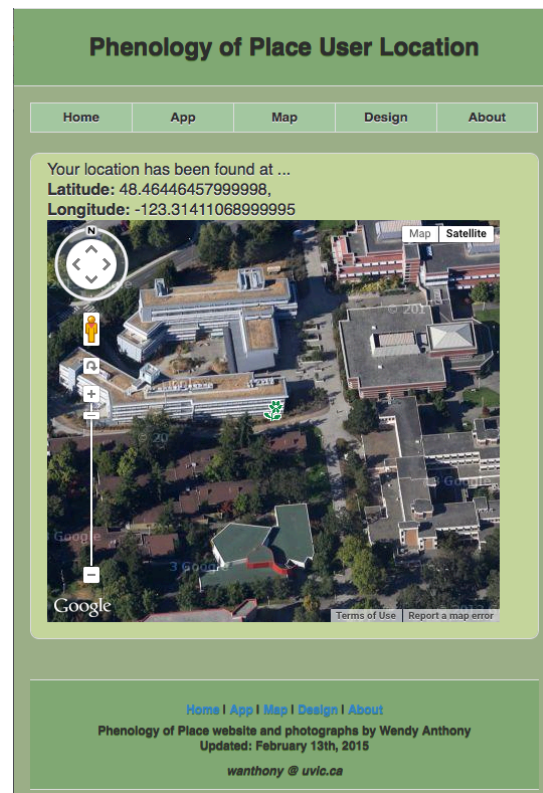


Figure 13: User's location found.

## Geolocation Script (W3C 2014).

```
<script type="text/javascript">
if (navigator.geolocation) {
document.getElementById("msg").innerHTML = "Geolocation service is trying to find
your location ...";
navigator.geolocation.getCurrentPosition(success, fail);}
else {document.getElementById("msg").innerHTML = "Your browser does not support
Geolocation service";}}
document.addEventListener("DOMContentLoaded", init, false);
function fail(position) {
document.getElementById("msg").innerHTML = "Geolocation service cannot find you at
this time ...";}
function success(position) {
var lat = position.coords.latitude;
```

```

var lng = position.coords.longitude;
document.getElementById("msg").innerHTML = "Your location has been found at ...
<br><b>Latitude:</b> " + lat + ",
<br><b>Longitude:</b> " + lng;
var latlng = new google.maps.LatLng(lat, lng);
var options = {
    zoom: 18,
    center: latlng,
    mapTypeId: google.maps.MapTypeId.SATELLITE};
var map = new google.maps.Map(document.getElementById("map"), options);
var marker = new google.maps.Marker({
    position: latlng,
    map: map,
    icon: "http://web.uvic.ca/~wanthony/phenology/images/garden.png",
    title: "You are here"});}
</script>

```

## V. Creating a Directed Studies Proposal

1. Develop an idea into a concept, and map the core elements (see Part VI).
2. Create a proposal for the concept to be explored in a directed study.
3. Contact supervisors with request for supervised directed studies.
4. Develop Evaluation techniques and marking strategies, with supervisor.
5. Fill in Pro Forma form, supervisor's signature, submit for approval (department head, faculty advisors, course registration) (see Appendixes I, II).
6. Once course is registered, pay tuition fees can be paid.
7. Discuss what kind of supervision and support will be offered, how to arrange.
8. Arrange schedule for meetings with supervisor, for regular updates, questions
9. Set and confirm Due Dates for Assignments.
10. Confirm all dates by return e-mails, and re-confirm closer to each date.
11. Keep in regular contact with supervisor, ask questions, and confirm timelines.
12. Complete assignments.
13. Make a Presentation of the Direct Studies project.

## VI. Creating a Concept Map Using VUE

VUE (Visual Understanding Environment), is a free Concept Mapping software download, used by many academic institutions to facilitate learning (see Appendix IV and V, Figures 12 and 13).

### Create & Save A New Map

- 1 Open VUE > New Map
- 2 File > Save As [MapName.vue]
- 3 Make regular saves to ensure your work is preserved

### Create A New Node/Concept - with label

- 1 [Press Key] N (Node tool) > Click-&-Drag to form a new Node
- 2 Name node by typing to replace blue box
- 3 To choose Shape of Node: > Windows > Formatting Palette > Shape > choose shape
- 4 To choose Colour of Node: > Windows > Formatting Palette > Color > choose from palette
- 5 Click outside node

**Create A New Node - without label**

- 1 [Press Key] N (Node tool) > Click-&-Drag to form a new Node
- 2 [hit] Delete key to avoid naming node
- 3 To choose Shape of Node: > Windows > Formatting Palette > Shape > choose shape
- 4 To choose Colour of Node: > Windows > Formatting Palette > Color > choose from palette
- 5 Click outside node

**Link Two Concepts to Show Relationships Between Nodes/Concepts**

- 1 [Press Key] L (link mode tool) > drag from one node to other
- 2 Click inside one node, drag to other node, release mouse click when 2nd node turns green
- 3 Type relationship phrase in blue label
- 4 Choose which end of link has arrow > Windows > Formatting Palette > e.g. Start: none, End: >;  
for a 2-way arrow, choose Start <, End >
- 5 Choose Shape of Link: > Windows > Formatting Palette > Link > e.g. straight, one curve, multi
- 6 Choose Line Weight of Link: > Windows > Formatting Palette > Weight > e.g. 0, 1-6 pixels
- 7 Choose Type of Line of Link: > Windows > Formatting Palette > Stroke > solid, dotted, dashed
- 8 Click outside node

**Add an Image** (Note: by default, each Node allows only either an image or an URL link, not both)

- 1 Right-Click Node > Add Image/File [find image to link to]

**Add an URL (to create an Interactive Map)**

- 1 Right-Click in the top section of original node > Add URL

**Node Inside A Node**

- 1 [Press Key] N (Node tool) > Click-&-Drag to form a Node
- 2 Name node by typing to replace blue box
- 3 Click outside node
- 4 [Press Key] N (Node tool) > Click-&-Drag to form a new Node
- 5 Name node by typing to replace blue box
- 6 Click outside node
- 7 Click-Drag-&-Drop new Node inside of original Node (original Node turns green when ready)

**Rapid Prototyping Tool** (To draw a link from an existing node and creates a new node.)

- 1 [Press Key] R **OR** Click the Rapid Prototyping tool (the fourth tool from the left on the toolbar)
- 2 Click an existing node
- 3 Drag out into the window. The link is created and extended
- 4 Release the mouse. A linked node is created
- 5 Type a name for the node
- 6 Click the link > Type a label for the link > Save

**To use the keyboard shortcut for Prototyping Tool**

- 1 Click the Selection tool. The cursor changes to an arrow
- 2 Click on the node and hold down Alt (Windows) or Option (Mac)
- 3 Drag out to empty canvas and release. A new link and node is created

**Keyboard Shortcuts**

- [Press Key] N > Node tool (to create node/concept)
- [Press Key] L > Link mode tool (to create link between nodes)
- [Press Key] R > Rapid prototyping tool (to create link with node at end)
- [Press Key] S > Select tool (to move nodes or links)

## VII. Lessons & Wish Lists

### Skills & Knowledge Gained Through Phenology Project

- Digital Photography
  - Botanical Photography Best Practices
  - Repeat photography
- Phenology
  - Phenophase ID
  - Climate change
  - Citizen Science
- Code Development & Design
  - FileMaker Pro 13 relational database design
  - HTML5 & CSS3
  - Java Script
    - jQuery Mobile
    - JSON (scripted export from FMP) & GeoJSON & Google Maps
  - Web Development Best Practices
    - Responsive, interactive web programming
- Marketing:
  - ES490 idea > course credit
  - User Testing

### Suggestions For Wish List

- Maps
  - Find User's moving location on Google Maps - relate to markers and trails
  - Open Layers vs Google Maps
  - Leaflet.js to use GeoJSON with Open Layer Maps
  - Aggregate markers by species in Google Maps
  - Use proximity of user's changing location on Google Maps in relation to species location marker when user is moving along a trail
- HTML forms for user searches, and data input, including the ability to upload citizen science, crowd-sourced photos, of various phenophases, to the server
  - Learn Php & MySQL to create HTML forms
- Add section "What to expect to see When/Where", using a seasonal chart and map
- Add Glossary – don't use technical terms (e.g. phenophase)
- Add other info e.g. Traditional uses, edible vs poisonous
- Compare to similar species, especially Native vs Invasive
- Make downloadable app to use in field without Wi-Fi
- Include other locations: with similar species, with native species not found at UVic
- Learn how to use versioning software repositories, like Git Hub



## VIII. Challenges

- The image and map pages are all individual files, which would need to be maintained separately. Ideally code would populate the image and map pages automatically based on the choice of species from the List View.
- **Image Challenges**
  - Adding more details to the List View, expands the section height to be larger than the thumbnail image.
  - Determine image sizes needed for CSS media device-size code breakpoints
  - Image was no longer responsive after adding back button
  - Separate web page needed for each species image - any change requires changing each page, individually for each species
- **List View Challenges**
  - The information on the species detail page cannot be searched from the Search tool on the main List View page.
  - In order for the species information to be searchable, the details need to be included in the List View text on the first page.
  - Including enough information to be searched in limited space.
- **Google Maps Challenges**
  - Multiple location markers proved too difficult to determine which marker to click; use cluster markers for species and location to limit visual clutter.
  - Still looking for code for getting location to respond to user movement, to show user's proximity to species location
  - Google Maps API may change in the future, resulting in broken web pages, or markers that no longer show on maps, requiring ongoing maintenance
- **User Testing Challenges**
  - Need more photos in database for different phenophase photos for each of the species in the List
  - Create methods and protocols for adapting field guide app code for other users e.g. UVic Herbarium
- **Challenges Resulting From Design, Code & Devices**
  - Different devices with different coding needs will come on the market, and will need to be accounted for by using responsive design, and CSS Media elements to create screen size breakpoints for style changes
  - Images, maps and webpage may need different software and API code
  - Mobile Device handle drop-down lists differently
  - Add new features from user test feedback, before asking for user testing
  - JavaScript/JSON use to populated one page versus one-page/species
  - JSON file needs to be on the same server for JQuery Mobile & AJAX to work
  - JSON file includes all the data in one file, and downloaded to user's device
  - Mobile Web Pages won't work without JavaScript enabled
  - Find a meaningful way to visually portray phenological data, along with weather and day length data

## **IX. Conclusions**

Designing and developing this mobile-friendly web application may end up being a long, continuing process as new functions are developed and added, new user tests are conducted, and the app is redesigned, retested, repeatedly, until all currently desired functions, and maybe even some from the wish list are included. This project will continue to be a work-in-progress, with more native plant phenophase photographs added each year, as the seasons continue to turn.

Someday I hope this Phenology of Place project will develop into more than simply a labour of love, but into a useful data model for documenting phenological data through digital photography, and finding ways to incorporate weather and day length data, in order to analyze any effects from the changing weather patterns on the phenological growth phases of the local native plants at UVic.

## X. References

Butler, Howard, Martin Daly, Allan Doyle, Sean Gillies, Tim Schaub, Christopher Schmidt. (2008). *The GeoJSON Format Specification*. Retrieved March 25, 2015 from <http://geojson.org/geojson-spec.html>

Google Developers -G. (2015). *Google Maps JavaScript API v3*. Retrieved on February 13, 2015 from <http://code.google.com/apis/maps/documentation/javascript/basics.html>

Government of Canada. (2015). *Daily Data Report: Victoria University CS, British Columbia*. Retrieved January 25, 2015 from [http://climate.weather.gc.ca/climateData/dailydata\\_e.html?StationID=6812&timeframe=2&cmdB1=Go&Year=2009&Month=5&Day=9](http://climate.weather.gc.ca/climateData/dailydata_e.html?StationID=6812&timeframe=2&cmdB1=Go&Year=2009&Month=5&Day=9)

jQuery Mobile. (2013). *Google Maps*. Retrieved February 13, 2015 from <http://demos.jquerymobile.com/1.4.0/map-geolocation/>

jQuery Mobile. (no date). *Pages*. Retrieved February 13, 2015 from <http://demos.jquerymobile.com/1.4.5/pages/>

jQuery Mobile. (no date). *Responsive Web Design*. Retrieved April 6, 2015 from <http://demos.jquerymobile.com/1.4.5/rwd/>

JSFiddle. (n.d.). *Using GeoJSON to Create Google Maps InfoWindows*. Retrieved February 13, 2015 from <http://jsfiddle.net/9aZcv/>

JSONLint. (2015). *The JSON Validator*. Retrieved February 13, 2015 from <http://jsonlint.com>

Lee, Mark. (2013). *A simple DIY responsive image slideshow made with HTML5, CSS3, and JavaScript*. Retrieved February 13, 2015 from <http://themarklee.com/2013/12/26/simple-diy-responsive-slideshow-made-html5-css3-javascript/>

McGrath, Mike. (2013). *Locating Users*. In *JavaScript in easy steps* (Pages 202-203). Warwickshire, UK: Easy Steps Limited.

Mozilla Developer Network. (2015). *Using geolocation*. Retrieved February 13, 2015 from [https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/Using\\_geolocation](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/Using_geolocation)

National Research Council Canada. (2015). *Sunrise/sunset calculator*. Retrieved January 25, 2015 from <http://www.nrc-cnrc.gc.ca/eng/services/sunrise/advanced.html>

Stack Overflow. (2012). *Center Google Maps (V3) on browser resize (responsive)*. Retrieved February 13, 2015 from <http://stackoverflow.com/questions/8792676/center-google-maps-v3-on-browser-resize-responsive>

PS-Scripts. (2008). *CSV Data Collector*. Retrieved August 2015 from <http://www.ps-scripts.com/bb/viewtopic.php?f=19&t=2365&sid=6fe234bdba938828e87aecebce0f1b71>

VUE. (2015). *About Vue*. Retrieved January 1, 2015 from <http://vue.tufts.edu/about/index.cfm>

W3C. (2014). *Geolocation API Specification*. Retrieved February 13, 2015 from <http://dev.w3.org/geo/api/spec-source.html>

w3schools

. (2015). *jQuery Mobile Filters*. Retrieved April 7, 2015 from [http://www.w3schools.com/jquerymobile/jquerymobile\\_filters.asp](http://www.w3schools.com/jquerymobile/jquerymobile_filters.asp)

w3schools. (2015). *Try It: Nav bar*. Retrieved February 13, 2015 from [http://www.w3schools.com/jquerymobile/tryit.asp?filename=tryjqmob\\_navbars](http://www.w3schools.com/jquerymobile/tryit.asp?filename=tryjqmob_navbars)

## Appendix I: Directed Studies Proposal pt. 1

Three final research papers were actually produced:

1. Developing Best Practices for Native Plant Flower Photography
2. Documenting Landscape Change: Photo Point Monitoring and Repeat Photography
3. Online Databases Track Plant Phenology With Digital Images From Citizen Scientists

ES490 Directed Studies – Proposal – Wendy Anthony – May 10, 2014

---

**Title:**

Developing a Phenology of Place at UVic Through Citizen Science Native Plant Photography: Designing a Database To Determine Seasonal Patterns

**Question:**

Can a collection of photographs, taken over a 5-year period by amateur citizen scientist, provide enough useful data to develop the basis of a seasonal calendar of native plant growth phases over their photosynthetic period?

**Course Description:**

Read about phenology and repeat plant photography. Choose representative samples from unnamed photos. Create a database and populate with data mined from photos. Organize representative photo samples in on-line gallery. Write paper: review and analyze literature, document database design and data collection process, look for patterns, discuss, future suggestions, conclusions.

**Content:**

1. Create on-line Photo Gallery of chosen indicator species
  - a. Sort through photographs I've taken at UVic campus 2009-2014
  - b. Find representative samples in various flower & leaf growth stages
  - c. Choose a few indicator species to focus on (spring & autumn)
  - d. Identify sites for repeat photos, species verification, GPS location
  - e. Continue to take photos
2. Design a Database of Native Plant Phenological Info at UVic
  - a. Liaise with other plant database administrators
  - b. Logically name photos to self-sort, and related to database
  - c. Populate with data gleaned from photos (e.g. 100)
  - d. Organize, analyze, summarize data, note patterns
3. Write a paper
  - a. Review literature about phenology using photo analysis to establish patterns and citizen scientists to collect data
  - b. Summarize photo database design method
  - c. Discuss: results, repeat photos, research merit, recommendations
  - d. Appendix: Database Summary Tables, Photo Gallery example, Guide to UVic Native Plant Seasons, Map Plant Locations

**Skills to Learn:**

Design research project; take photographs attending to Data collection; mine photos for ecological data; design native plant database; create web gallery.

**Deliverables & Proposed Evaluation Technique:**

40% - Paper: Develop Phenology of Place at UVic by Designing Photo Database  
 40% - Database: Design, populate with chosen indicator species  
 20% - On-line Photo Gallery: Selected Photos from Database

*Figure 14: Directed Studies Proposal pt. 1 Pro Forma*

## Appendix II: Directed Studies Proposal pt. 2

### ES490 Directed Studies Phenology of Place at UVic - pt2 Wendy Anthony

#### CONTENT:

A continuation of ES490 Phenology of Place @ UVic pt1 to explore :

1. Study local phenological patterns using local native plant phenological data, weather data, and day-length data
2. Research methods of creating interactive web apps as phenological teaching tools, using mobile-friendly responsive-design, with phenological and location data from digital images

#### TEXT:

No text required. Mobile devices for design and testing would be useful.

#### EVALUATION TECHNIQUE:

**50% - Manual:** step-by-step story of the project: **(due Apr 10, 2015)**

**Pt1:** building a photograph archive, creating a database with relevant field data, assessing phenology stage, research citizen science, best practices of botanical photography, photo monitoring

**Pt2a:** document the process of incorporating local native plant phenological data with weather and day-length data into a database, creating visualizations to look for any patterns

**Pt2b:** document the design, implementation and testing of mobile, responsive, interactive web apps, with the incorporation of phenological and location data

**25% - End Product - Initial Testing - Pt 1 (due Mar 2015)**

- Testing the mobile-friendly field guide to local native plants

**15% - End Product - Beta App Pt 2 (due Apr 10, 2015)**

- example of a mobile-friendly field guide to local native plants

**10% - Presentation (due Apr 10, 2015)**

- show end product: phenological visualization, Field Guide app
- describe process of phenology project

*Figure 15: Directed Studies Proposal pt. 2 Pro Forma*

## Appendix III: Questions That Helped Drive This Project

- Do native plant species have regular seasonal phenological phase patterns?
  - Do these patterns correspond to some weather or day length signals that might offer an opportunity to study the impact of changing climate patterns on seasonal plant growth patterns?
  - What kind of statistical analysis needs to be performed to determine if patterns are significant?
  - e.g. is Spring 2015 early?
  - Identify some “Indicator Plants” to monitor
    - e.g. Indian Plum (first shrubs to bud green leaves), Flowering Red Current (one of first shrubs to flower). Salmonberry (first nations hunting and fishing cycle indicator)
- What kind of evidence needs to be collected to document change?
  - How to use Photographs in an efficient manner?
  - What indicators need to be monitored (e.g. phenological phases)?
  - Weather data to determine any changes in patterns
- How to design a database to collect phenological information?
  - How to export data in a useable format?
- How to create a base of citizen science volunteers to gather geo-located photo data to contribute to a long-term data set?
  - Create a field guide for identifying species, and phenophase stages
  - Create seasonal field hikes to collect data on specific plants
  - Create an app/website for uploading photos
- How might users use this database and/or field guide app?
  - What kind of testing can users help with?
  - How to use testing feedback?
  - What to do about problems/challenges revealed?
  - How to identify whom the audience is?
  - How to get users to test app?
  - How will users interact with the app?
- What kind of technology can be used?
  - Non-proprietary, Open-Source
  - Useful across multiple Devices & Platforms?
    - e.g. mobile-friendly
    - Responsive design
  - How to use web tools to facilitate data sharing?
- How to make data available?
  - Browser-based – JSON file
  - Server-based – php & MySQL, HTML forms

## Appendix IV: Concept Map for Phenology of Place Pt. 2

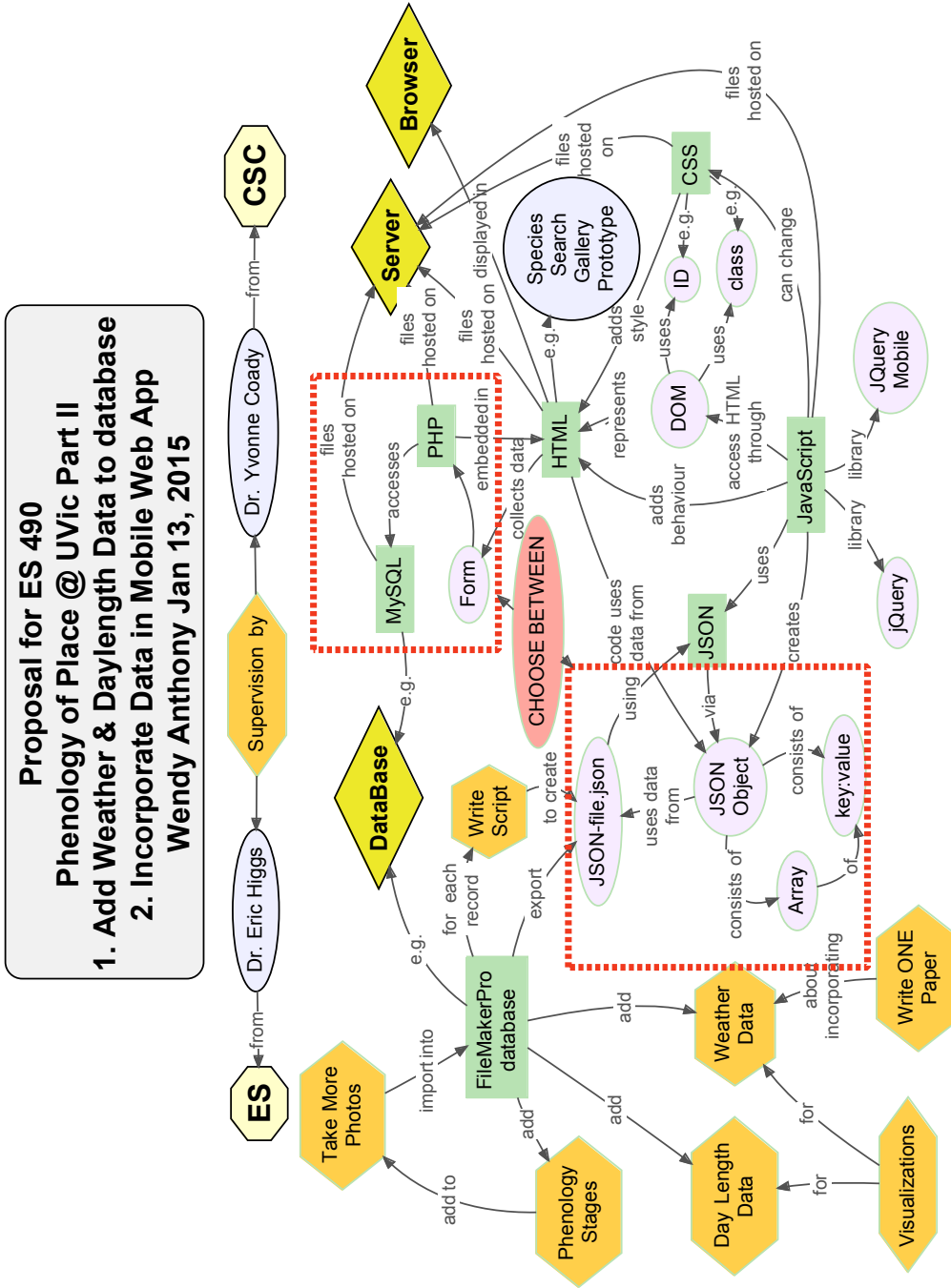


Figure 16: Concept Map for Phenology of Place Native Plant Field Guide App



## Appendix V: Sequence Diagram for Field Guide App

A Sequence Diagram can be used to explain the interaction and sequence of events that occur during the various user interactions with the field guide app (see Figure 13)

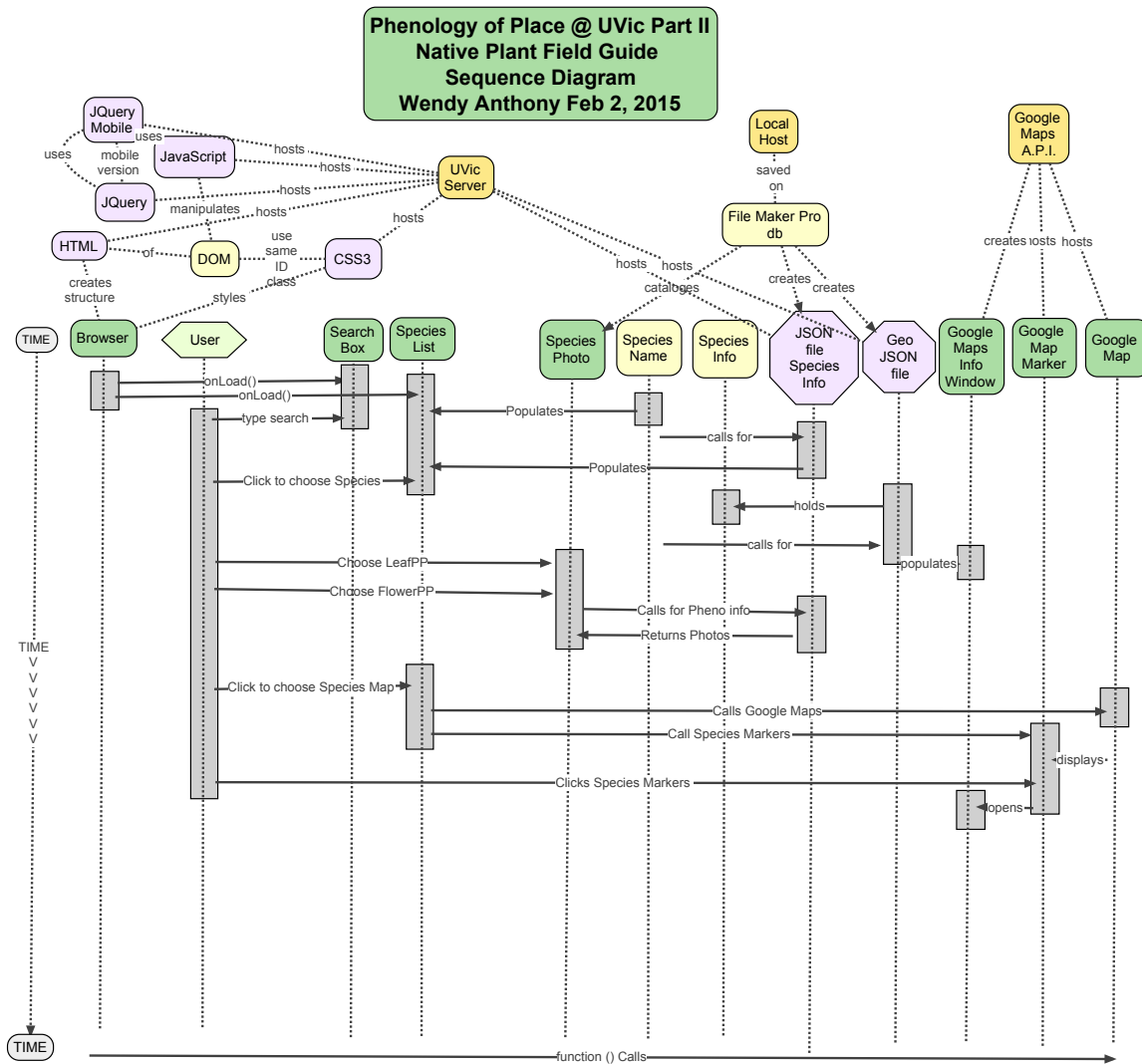


Figure 17: Sequence Diagram shows timing of interaction of objects and events.