**Online Wildlife Library (OWL) Architecture Design Document V1.0**

**Table of Contents**

[1 Overview](#h.30j0zll)

[1.1 System Purpose](#h.1fob9te)

[1.2 System Architecture](#h.3znysh7)

[2 OWL Client](#h.tyjcwt)

[2.1 General Use Interface](#h.3dy6vkm)

[2.1.1 Question Display](#h.1t3h5sf)

[2.1.2 Keyword Collection Module](#h.4d34og8)

[2.1.3 Result Display](#h.2s8eyo1)

[2.2 Client Communication Module](#h.17dp8vu)

[2.3 Implementation Technology](#h.3rdcrjn)

[3 OWL Server](#h.26in1rg)

[3.1 Server Communication Module](#h.lnxbz9)

[3.2 Query Processing Module](#h.35nkun2)

[3.3 Implementation Technology](#h.1ksv4uv)

[4 OWL Database Server](#h.44sinio)

[4.1 Implementation Technology](#h.2jxsxqh)

[5 Data Storage and Communication](#h.z337ya)

[5.1 Sample OWL Database Schema](#h.3j2qqm3)

[5.2 OWL Communication Protocol](#h.4i7ojhp)

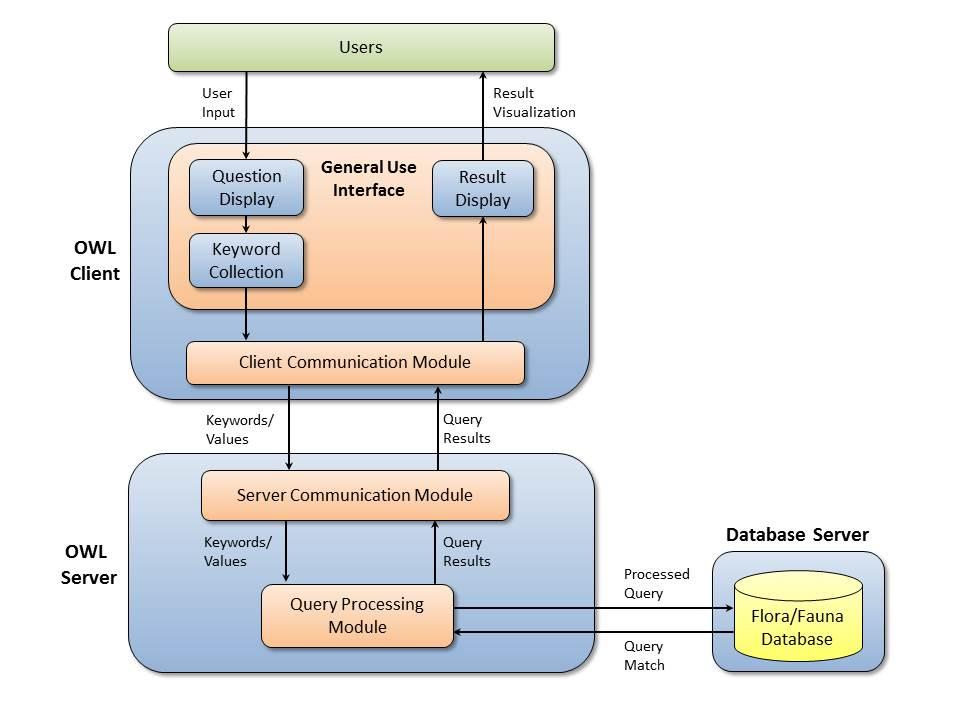
# Overview

## System Purpose

OWL is a web-based tool for identifying wildlife. Users are asked a series of questions about a plant or animal they want to identify. These questions generate a collection of keywords, which are matched to keys in a database.

## System Architecture

Figure 1 shows a high-level view of the OWL system architecture:



**Figure 1: OWL system architecture**

The *OWL Client* is a web-based user interface that enables users to answer questions about plants and animals, and see results returned by the system. The *OWL Server* processes user queries, matches them against information in a database, and returns any matches it discovers. The *Database Server* stores databases containing plant and animal information available to OWL. Each of these components is described in greater detail below.

# OWL Client

The OWL client provides users with an interface into the system. It is designed to be accessible by web browser. Key components include a *General Use Interface*, and a *Client Communication Module*.

## General Use Interface

The OWL client’s core user interface consists of three subcomponents: a *Question Display*, *Keyword Collection Module*, and *Result Display*.

### Question Display

The Question Display presents OWL users with a series of questions about the wildlife they want to identify. Each question is associated with some keyword and some value (e.g. number of leaves on a stem). Keywords are temporarily collected in the Keyword Collection Module.

<10/01 Modification: use dynamic approach to iteratively query results>

### Keyword Collection Module

This module temporarily stores keywords and values generated by the Question Display. Once all of the system’s questions have been answered, the module’s contents are passed to the OWL Server using the Client Communication Module.

### Result Display

The Result Display presents OWL users with the results of a database query executed using their question-generated keyword/value collections. Result contents may include a text description of the wildlife in question, and possibly a link to a related image. If no database match was found, the display will inform the user of this failure, and reset the Question Display.

## Client Communication Module

This module is responsible for mediating communication with an OWL Server. It takes in keyword/value collections generated by the General Use Interface, wraps them in the OWL communication protocol, and transmits them to the Server. It is also responsible for unwrapping query results sent from the server, and posting them to the Result Display.

## Implementation Technology

The OWL Client will be built using a combination of several open-source JavaScript libraries. The General Use Interface’s subcomponents will be built using the React JavaScript framework. The Client Communication Module will be built using the SuperAgent AJAX API.

# OWL Server

The OWL Server is responsible for mediating information requests from the OWL Client (represented as collections of keywords and values). Raw requests are transformed into database queries and executed. The results of the queries are then forwarded back to the client.

## Server Communication Module

This module is responsible for managing communication with OWL Clients. It unwraps keyword/value collections sent by a client via the OWL communication protocol, and passes them to the Query Processing Module. It is also responsible for wrapping and transmitting query results from Processing Module.

## Query Processing Module

The Query Processing Module transforms client-generated keyword/value collections into database queries, and executes them. It then passes the results of those queries to the Server Communication Module for transmission back to the client.

## Implementation Technology

The OWL server will be implemented using C# running on Microsoft’s IIS.

# OWL Database Server

The OWL Database Server will host a relational database containing information about plants and animals. Fields in the database will correspond to keywords collected from the OWL Client. Keys will uniquely identify records containing descriptive text and images of wildlife. Values associated with each input keyword will be matched against fields in the database; if one or more matches for the given set of values exist, the database server will return the appropriate records to the OWL Server. If no records match the given keys, the database server’s return message must clearly indicate no record was found.

## Implementation Technology

OWL’s databases will be built and hosted using Microsoft’s SQL server.

# Data Storage and Communication

## Sample OWL Database Schema

Table 1 shows a sample database table OWL may use. The “Fields” column indicates fields in the plant description table. The “Valid Values” column lists the valid values a field may hold.

**Table 1: Sample plant description database table**

|  |  |
| --- | --- |
| **Fields** | **Valid Values** |
| Plant\_Name (PK) | <any String> |
| Plant\_Type | Shrub, Shrub/Tree, Shrub/Vine, Tree, Vine |
| Leaf\_Shape | Pointed, Rounded |
| Leaf\_Color | Green, Yellow/Green, Yellow/Red, Red, None |
| Leaf\_Edge | Sawtooth, Scalloped, Smooth |
| Leaf\_Clustering | Yes, No |
| Cluster\_Size | 0,3,4,5 |
| Leaflets | 0,5,6,7,8,9,10,11,12,13 |
| Flower\_Present | Yes, No |
| Flower\_Color | Green, Yellow/Green, White, None |
| Habitat | Woodland, Open\_Field, Swamp, Any |
| Geographic\_Region | Eastern\_US, Western\_US, Any |
| Season | Spring, Summer, Autumn, Winter |
| Light\_Preference | Sunlight, Shade, Any |
| Description | <any String> |
| Image | <link to an image file> |

## OWL Communication Protocol

OWL uses JSON to transmit information between Client and Server. OWL’s JSON format uses a collection of key-value pairs, with keys corresponding to fields in the Flora/Fauna database and values corresponding to valid field values. (See Table 1 for an example set of fields and valid values).