**System Structure**

This Document aims to provide the basic knowledge needed to navigate and understand the folder structure of the system, it consists of sections for each area of the system details the contributing files and some paragraphs explaining key features of the program that allow it to run in general.

There are typically 5 types of files responsible per page and consist of:

 1 view containing the front-end html page that the user will see

 3 Controllers –

1 manages front-end JavaScript

1 for back-end Server / database interactions (mapping to database handled by Model files)

1 routing controller that handles directing to a page

There are cases where a larger page uses more than 3 controllers, but this will be explained when they appear.

The files can typically be found / identified by the following features:

**Views:** (api/views/pages [files ending with “.ejs“])

**Models:** Model (api/models)

**Routing Controller:** Controller Routing (api/controllers/ [files with prefix of “view-“ and ending “.js”])

**Front-end Controller:** Controller Front-end (api/controllers/ [files with suffix of “.page.js“])

**Back-end Controller:** Controller Back-end (api/controllers/ [all other files ending with “.js“])

**Routing**

The URL mapping to controllers in handled by sails’ routes.js file found under the config files, this allows user friendly url’s to be mapped to file names and allows easy linking in the view files in cases of buttons linking to another page (e.g. the export button is mapped to /export instead of having to type api/controller/bird/export-date-bird-simple.js)

**Utility Files**

These files are functions which we have used across multiple pages, sails allow them to written into their own folder () which can then be accessed globally by the rest of the system. The utilities we created include:

1. The AutoComplete function in search boxes (js/components/autocomplete.component.js)
2. The time converters (api/services/TimeUtils.js) which take unix timestamps and return formatted versions (i.e. in the Age field it returns number of years, month, days but for hatch date it can return the actual date string as DD/MM/YYYY)
3. The locking functionality to prevent concurrent changes on the same piece of information (api/controllers/try-acquire-renew-lock.js)