

Requirements Document

2015-16

Project Name: MammalWeb

Team Number 5

Team Members

Freddie Keen

Quentin Lam

Will Taylor

Tom White

Thomas Wilshaw

|  |  |  |
| --- | --- | --- |
|  |  |  |

Document Information

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Project Name: | MammalWeb | | |
| Prepared By: | Team 5 | Preparation Date: | 24/11/2015 |
| Email / Phone: | cs-seg5@durham.ac.uk |  |  |
| Document Version No: | 1.0 | Document Version Date: 26/11/15 |  |
|  |  |  |  |

Version History

| Ver. No. | Ver. Date | Revised By | Description |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents:

1. Overview
2. Proposed goals (Scope):
3. Domain Analysis
4. Proposed deliverables of the system including a Project Plan
5. Identified Risks, Assumptions, Dependencies and Constraints

1. Solution requirements
2. Definition of terms
3. References

1. Overview:

Our brief is to develop a system that provides a conclusive classification of every image within a database using a series of pre-set details. We are provided with a large database of images taken from a wildlife reserve. They have all been assigned with multiple classifications describing the contents of the image. Our system will act as a function that takes the database as an input and outputs a new table with one, accurate classification for each image as well as search functionality. It is hoped that the project will serve as both an ecological education tool and a collection of useful scientific data to help better understand Britain’s wild mammals. Our task is to make the system more accessible for users and administrators.

A volunteer is provided with a motion triggered “camera trap” and is asked to go out into the reserve and attach it to a tree, recording all aspects of its positioning. They then return after a period of time to collect the camera and upload the photographs into the database.

The remainder of the document will outline the following points:

Our proposed goals –

Domain Analysis –

Proposed deliverables of the system –

Project Plan –

Identified Risks, Assumptions, Dependencies and Constraints –

Definition of terms –

Solution requirements –

**2. Proposed goals (Scope):**

Our aim is to develop a software which can distinguish different species within a photo (if there is any). We will not be able to classify a photo if there are not enough data (entries from 'spotters') provided. When there is sufficient data on a particular photo, our algorithm is to classify the species and the certainty of the classification. We store the classification and certainty in a new table in the data base. Once a photo has been correctly identified, they will be retired (not shown to spotters anymore). The algorithm is to be implemented with criteria similar to the Swanson et al (2015) and to have a threshold for correct classifications.

Our end users are going to be the scientist running the MammalWeb project. We are to provide a reliable platform for the scientists to upload new data and download relevant data from the data base.

We are also to provide a backend interface which allows the scientists to apply a filter and select data that meets the criteria (e.g. Date, site etc.), and they can either view or download the data (in .csv format) and do further analysis if they wish. An estimation of numbers of different species in a given area then can be derived from the processed data. This allows the scientists to monitor wild mammals in England much more efficiently without needing to go through thousands of photos manually.

\*\*(maybe)

We will also implement an algorithm which will decide the likelihood of a photo shown to the spotters, where the photos with more classifications will have a higher chance, increasing the retire rate.

\*\*

We will try to implement a user dashboard for the website. While the volunteers are classifying photos on their individual user accounts, they have an option to ‘like’ the photo if they find the photo particularly interesting. The dashboard will then be able to show the users their favourite photos. They can also view their uploaded photos and their classifications from the dashboard.

3. Domain Analysis:

3.1 MammalWeb: System currently in place

MammalWeb's purpose is to collect information about wildlife in the UK and allow biological scientists to form conclusions about the ecosystem on both the national and local scale. This information can then be used to inform decisions about conservation projects, culling action and similar activities that affect wildlife and the environment.

The front end of MammalWeb at the moment is the website MammalWeb.org on which users can upload images from camera traps that they have set up in their local area. Anyone can then sign on as a 'spotter' and classify images drawn from the all those that have been uploaded. The classifications (species present and age/gender/number if relevant) are then stored in a database.

Currently, extracting useful information from these classifications is not well implemented - the data dump we recieved contained data from a little over 20000 photos with nearly 90000 individual classifications and there is no automated system in place for removing 'bad' information and condensing the useful data into a form that is easier to use for the biologists. Any use of the collected data so far has been done by manually parsing the data.

We have been provided with an algorithm designed to aggregate data collected in citizen scientist projects. The algorithm, outlined in Swanson et al. (2015), creates an aggregate classification equal to the most common choice from all the individual classifications. For example, if 8 classification of deer, 3 of horse and 1 of nothing here have been recorded then the Swanson algorithm will say that the photo is of a deer. 3 metrics to show how likely the classification is to be correct are then calculated-

1. Evenness: All non-blank classifications are used in Pielou’s eveness index to calculate this. The formula is where S is the number of different species classified as being present (2 in the example given) and pi is the proportion of total classifications for species i (8/11 and 3/11 in the example- remember that blanks are ignored here). This comes out as 0.845 for the example. If only one species is classified, the result will be 0 and the highest possible result is 1 so a relatively high result like 0.845 can be interpreted as high uncertainty that the aggregate classification is wrong.
2. Fraction blanks: The fraction of “nothing here” classifications for an image that has an aggregate classification that is not “nothing”. The above example would have fractional blanks of 1/12
3. Fraction support: The fraction of classifications that support the aggregate answer (in the above example it would be 2/3)

In general, the Swanson paper says that the number of classifications required for an accurate aggregate answer is quite low. For easy to identify species, after only around 3 classifications the aggregate answer has an extremely high level of accuracy if the evenness is low. For more difficult species after around 10 classifications the aggregate is fairly likely to be accurate, but for these species even increasing the number of classifications does not improve the aggregate’s likelihood of being correct very much. We have not been given a ‘gold standard’ set of data so calculating which species are easy and which are difficult will not be possible. The paper focuses on images that only contain a single species of animal for ease of measuring how accurate the algorithm is, but does say that if two species are present then simply take both the most and second most identified species as the overall aggregate. Other specifics to consider are when there are ‘enough’ classifications on a picture to give satisfactorily accurate metrics and when to discard an image as having nothing in it (The suggested boundary is 5 blank classifications in a row).

3.2 Related systems

A closely related system would be Snapshot Serengeti (SS) where a very similar system of uploading and classifying camera trap photos by citizen scientists is in place. There are a series of blog posts by Margaret Kosmala (beginning here[[1]](#footnote-1)) that give good insight on how they went about improving the performance of their algorithm. SS went through a couple of slight variants for their plurality algorithm, but is overall very similar to what we are going to implement.

Initially, a species was made the aggregate classification if it had >=50% of the overall classifications. This gave a classification for 96% of the images captured where 57% were unanimous and 84% had at least ¾ fraction support. A later refinement of the algorithm[[2]](#footnote-2) made it closer to what we will be using by saying that the most picked animal was classified as the aggregate classification ie. If there were 10 Impala, 4 Thomson Gazelle and 7 Dik Dik classifications, the older version of the algorithm would not give an aggregate answer but the newer one (and Swanson’s) would say that the photo was of an impala. This meant that almost 97% of images received a classification but there were a few more errors- of the images that were not classified by the old algorithm but were by the new, 57% were correct when compared with the expert data set.

Important things to note about SS’ plurality algorithm is that they had a group of experts create a set of definite classifications and the algorithm agreed with these expert classifications on 95.8% of photos. This is similar to the certainty percentages given by Swanson for the results of his algorithm. Another potentially important point is that all of the images used for aggregate classifications had received at least 10 separate classifications. Initially photos with less than this were used however it led to very inaccurate results for some photos where animals that weren’t present were nevertheless identified as being there. The number of classifications for images on MammalWeb is significantly lower than for SS so defining boundaries of when a classification should be considered definite is probably something we need to experiment with.

4. Deliverables:

The main deliverables of the project are as follows, with any deadlines not detailed in the section for the relevant deliverable shown in the Gannt chart below:

4.1 Requirements Specification

This shall be a complete, well-structured and unambiguous document adhering to the template on duo. It will contain sections that will, as defined by the requirements specification document template on www.duo.dur.ac.uk, in order;

* Describe the purpose and context of the project, as well as the structure of the requirements specification document.
* Outline the scope of the project, along with its benefits, objectives and stakeholders.
* Explore the domain area of the project and provide references to similar systems.
* Show the project plan, with realistic, clearly defined activities attributed to team members.
* Discuss the potential risks of the project, enlisting all hardware and software used, the limitations of these software/hardware and issues concerning ongoing maintenance.
* Clearly entail and describe requirements, including non functional requirements, with a good numbering system.
* Define all terms used and contain references.

The first five sections of this document shall be drafted by Freddie Keen, Quentin Lam, Will Taylor, Tom White and Tom Willshaw respectively with a soft deadline of 24/11/2015, before being proof read, discussed and edited by the team in a meeting on 24/11/2015. The remaining two sections will then be written collaboratively before the draft deadline for this document on 02/12/15, and edited between this date and the final deadline for the requirements specification on 28/01/2015.

4.2 A Classification and filtering web system

This shall be a webpage or set of webpages that use an implementation of an algorithm to ‘decide’ when a photo has been classified with certainty, per Swanson et al. (2015) and to associate species with images where this has been decided.”(quotation from [www.duo.dur.ac.uk](http://www.duo.dur.ac.uk)). This will be worked on by the entire team with a soft deadline of 01/02/2016, beginning after the completion of the draft requirements specification.

This shall involve a method to store the results from the basic classification algorithm such that the algorithm is run only when additional classifications are added by users of the system, (paraphrased, www.duo.dur.ac.uk). This will be worked on by the whole team between 01/01/2016 and 10/02/2016. This shall allow “users [of the system] to select images based on filter criteria including species, favourite status, date/time, habitat, type … site” ([www.duo.dur.ac.uk](http://www.duo.dur.ac.uk)).

This shall also contain an interface for the scientists behind the MammalWeb project to search through and filter the images based on criteria such as species, whether an image has been classified by the algorithm yet, the location in which the image was taken or the time when it as taken. The scientists shall be able to download classified data in .csv format. This shall be compatible for analysis by programs written in Python or R.

The final part of this deliverable shall be a web dashboard visible to a user of the system implementing as many as possible of the following:

- Showing the user their favourite photos

-Showing the user a timeline of their uploads and classifications a chord diagram of relationships.

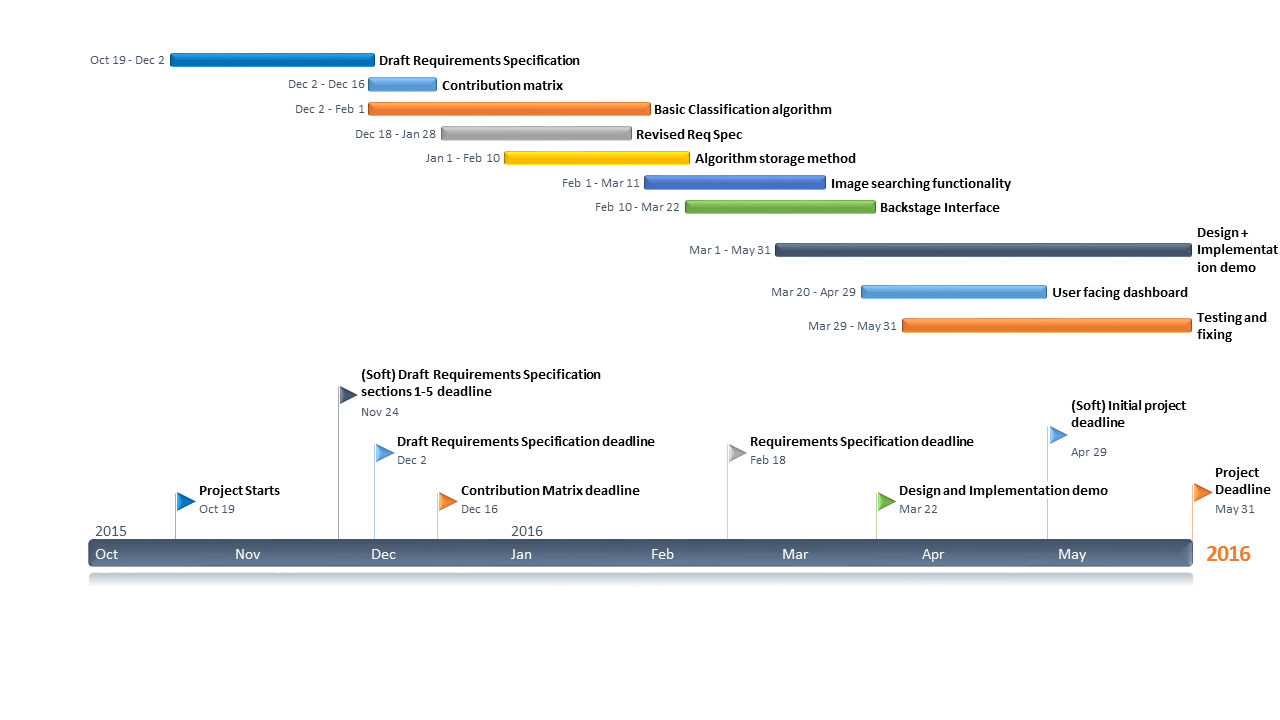
-Showing the user an interactive map based on data selected using the filter detailed in ‘functionality for users of the system to select and filter images’.

4.3 Design and Implementation Demo

This shall be a short presentation on the design and implementation of the project, worked on by the whole team and completed between 01/05/2016 and 31/05/2016.

An initial soft deadline for all of the above deliverables except the design and implementation demo to have been implemented has been set at 29/04/2016, to leave time for testing the system, making sure that it works, and finishing the presentation about implementation and design. The overall project plan, with hard and soft deadlines, is shown graphically in the Gantt chart below.

Gannt chart depicting the planned order of tasks as well as hard and soft deadlines for team 5’s MammalWeb Group project, Durham University 2nd year computer science Group Project Module, 2015-2016.



5. Risks, Assumptions, Dependencies and Constraints:

**5.1 Risks**

As this piece of software is entirely a data gathering program there is no risk of harm to any user. That being said there are risks of generating erroneous or misleading data. Hopefully through good database design and careful query setup these risks can be mitigated.

The main risk faced is incorrectly implementing the algorithm. Clearly successful completion of the project hinges on this being correct. The likelihood of this happening is fairly low as the paper provided (*reference to paper*) clearly outline the algorithm and testing should be fairly straightforward. That being said the impact of this risk is obviously very high and so due care must be taken.

The second risk of generating misleading data stems from the usage and design of the search/query page of the website. This needs to be carefully designed with input from the client to make sure it is simple and easy to extract the sort of data they want. It should also export it in a way that is easy for them to analyses in an error free way. If the clients properly involved in the planning of this step the risk should be reduced. Also as the data is going to be used by people trained and experienced in data handling and analysis the impact of this is not particularly high.

Another risk consideration is security. As a web page accessible by anyone the page must be made secure. This will include making sure the pages are only available over HTTPS and modelling the database query page to avoid the possibility of SQL injections and other attacks.

Finally, there is the data protection risk. As the camera traps are left in public places the database may contain many images of members of the public. These images should be removed from the database to avoid any privacy issues. Although this is not directly under the scope of our project it is still worth consideration and due care.

**5.2 Assumptions**

We assume that the user is able to use a simple and hopefully straightforward search interface. We also assume they are able to handle the resulting CSV (*define*) files themselves.

**5.3 Dependencies**

The only dependencies for this project are as follows. The users continue to classify the images and upload new ones, without this the client has nothing to gather data from. The client has enough web hosting space to cope with an increasingly large data base.

**5.4 Constraints**

The main constraint on this project is time. With a deadline in March there is only a short amount of time to get the system up and running. Due to this constraint some of the less core features may have to be set aside. So long as a working system is set up the project will have been successful. However some of the other features would greatly enhance the usability and user friendliness of the system so are well worth trying to implement if enough time is allocated.

6. Solution requirements:

6.1 Functional Requirements:

|  |  |
| --- | --- |
| ID, type and title | FR1.1 - Server side - Classify images based on Swanson algorithm |
| Description | Given the selection of classifications provided by volunteers, determine, to a defined level of certainty, the most accurate classification of the image. This entails factors such as the number of animals in the photo and the species of the animals. |
| Priority | High |
| Dependencies | N/A |
| Expected results | An output stating the number of animals in every photo and the species of the animals, accompanied by a level of certainty. |
| Exception handling | Given an uncertain output, flag image as uncertain. TBC |

|  |  |
| --- | --- |
| ID, type and title | FR 2.1 – Server Side - Storing results of classification of documents in SQL database |
| Description | Creates a new table in the database to store data generated from our classification algorithm. |
| Priority | High |
| Dependencies | FR1.1 |
| Expected results | When filtering results for classified data, only photos that are associated with the new table are returned. It should return the species and certainty for each entries. |
| Exception handling | The user may abandon search anytime |

|  |  |
| --- | --- |
| ID, type and title | FR2.2 – Client Side – Create downloadable CVS files |
| Description | An admin should be able to download the results as a CSV file for external processing and analysis. |
| Priority | High |
| Dependencies | FR1.1, FR2.1 |
| Expected results | Once the data has been searched and filtered a button should become active that will download the CSV file. |
| Exception handling | Button only becomes available if data has been searched and filtered. |

|  |  |
| --- | --- |
| ID, type and title | FR3.1 – Client Side – Remove classified images from display list |
| Description | Once an image has received enough classifications to be confidently classified it should be removed from the list of images that are shown to the users. |
| Priority | Medium |
| Dependencies | FR1.1 |
| Expected results | Once images are classified they no longer appear to normal users. |
| Exception handling | N/A an image is either classified or not. |

|  |  |
| --- | --- |
| ID, type and title | FR4.1– Client Side – A table showing the images that meet certain filter criteria selected by the user. |
| Description | The system should be able to show the user a table of images filtered by as many as possible of the following characteristics:  animal (species), user(s), gender, age, time period (down to seconds if possible), site (preferably being able to define an arbitrary geographical area, or at least by sites defined in the database), sequence, habitat type, presence of humans, blank images, number of classifications, and whether the algorithm has produced a consensus classification or if the photo needs to be manually checked. |
| Priority | Medium |
| Dependencies | FR1.1, FR2.1 |
| Expected results | Once the selected images are shown to the user they should be given an option to refine their filter criteria. |
| Exception handling | If a certain attribute of an image is unknown, it is not shown when this attribute is selected by the user as a filter criterion. |

|  |  |
| --- | --- |
| ID, type and title | FR4.2 – Client Side – A table showing the images taken at sites where the sites meet certain filter criteria selected by the user. |
| Description | The system should be able to show the user a table of images taken at sites where the sites where the sites have been filtered by any combination of as many as possible of the following characteristics: arbitrary geographical area, users, time period during which photos were taken, species of animals spotted at the site, number of photos submitted, number of sequences submitted, habitat types, presence of humans. |
| Priority | Medium |
| Dependencies | FR1.1, FR2.1 |
| Expected results | Once the relevant images are shown to the user they should be given an option to refine their filter criteria. |
| Exception handling | If a certain attribute of a site is unknown, images taken at this site are not shown when this attribute is selected by the user as a filter criterion. |

|  |  |
| --- | --- |
| ID, type and title | FR 5.1 –Client side - Favorite image slideshow |
| Description | Users can mark images as favorites (already implemented) which can then be viewed as a slideshow. |
| Priority | Low |
| Dependencies | FR2.1 |
| Expected results | A signed in user can go from their spotter status page to a slideshow of all favorited images. |
| Exception handling | Can only be accessed by signed in users.  If no images favorited a message saying so should be displayed. |

|  |  |
| --- | --- |
| ID, type and title | FR 5.2 –Client side- Timeline of uploads/classifications |
| Description | Users can access timelines that show when and how many images they have uploaded/classified |
| Priority | Very low |
| Dependencies | FR2.1 |
| Expected results | A signed in user can go from their spotter status page to a timeline page where they are presented with timelines for uploaded and classified images |
| Exception handling | Can only be accessed by signed in users.  If no images uploaded/classified a message saying so should be displayed. |

**7. Definition of Terms:**

**.CSV format -** Comma separated values. A file format where the individual data are separated with commas.

**Dashboard -** A web page or part of a web page that collects and presents information to a user based on specific criteria.

**Deliverable** - A physical or non-physical object presented to the customer as part of the project.

**DUO**- **Durham University Online**. The online blackboard learning system for Durham University.

**Domain (of a software project)** - Other systems and software providing similar functionality to the project.

**Functional Requirements** - Things that the project system must do.

**Gannt chart** - A timeline showing the planned start and end dates for work on each separate task involved in the project.

**Gold standard data**- Classifications provided by experts and are considered 100% accurate. Can be used to assess algorithm accuracy

**MW**- **MammalWeb** the organisation and project which we are working on: http://www.mammalweb.org/

**Non Functional Requirements** - Intended characteristics of the project other than its functionality.

**Normal User** - A non admin users, i.e. someone who only uploads and classifies images.

**Python** - A programming language.

**R** - A programming language.

**Scope (of a software project)** - The boundaries of what a project will deliver.

**SS**- **Snapshot Serengeti,** a similar citizen science project from which we drew inspiration

**8. References:**

1. http://blog.snapshotserengeti.org/2013/01/30/some-results-from-season-4/ [↑](#footnote-ref-1)
2. http://blog.snapshotserengeti.org/2013/06/07/majority-rules-algorithm/ [↑](#footnote-ref-2)