Birdry

System Requirements Specification

Team 1
Kyle Fritz
Laras Istiqomah
David Leiberg
Julian Sniffen
Nicholay Topin

Client
MAPLE Lab (point of contact: John Winder)

3/15/17

Birdry
System Requirements Specification

Table of Contents

1 Introduction

1.1 Purpose of This Document

The purpose of this document is to outline the functional and non-functional requirements for the software Birdry. Furthermore, it will identify the different system components and explain how they interact with each other. This document is primarily designed as a reference for development, however it will also be presented to the customer to ensure approval of the final design of the application.

1.2 References

Currently, this document does not reference any documents provided by the customer. It only references *UML Distilled*, by Martin Fowler, for the included figure.

1.3 Purpose of the Product

This software system will be an Android mobile application designed to help users identify and classify birds from pictures taken on their device. Additionally, the software will utilize the existing GPS technologies available on most devices to allow users to record the locations where birds have been previously identified. The application will be freely distributed and emphasize ease of use in order to reach a broad demographic.

Separate from the mobile application will be a server hosting a database of all taken photographs. The server will communicate with the mobile application over a network in order to populate the database and provide necessary support for the application. The resulting database will allow an administrator to monitor application use and extract useful information from gathered photographs.

1.4 Product Scope

The functionality of Birdry is expressed in Figure 1. Details for each use case can be found in Section 2, where the corresponding use case numbers and descriptions match the accordingly labelled use case UC X in the diagram (where X is the use case number).

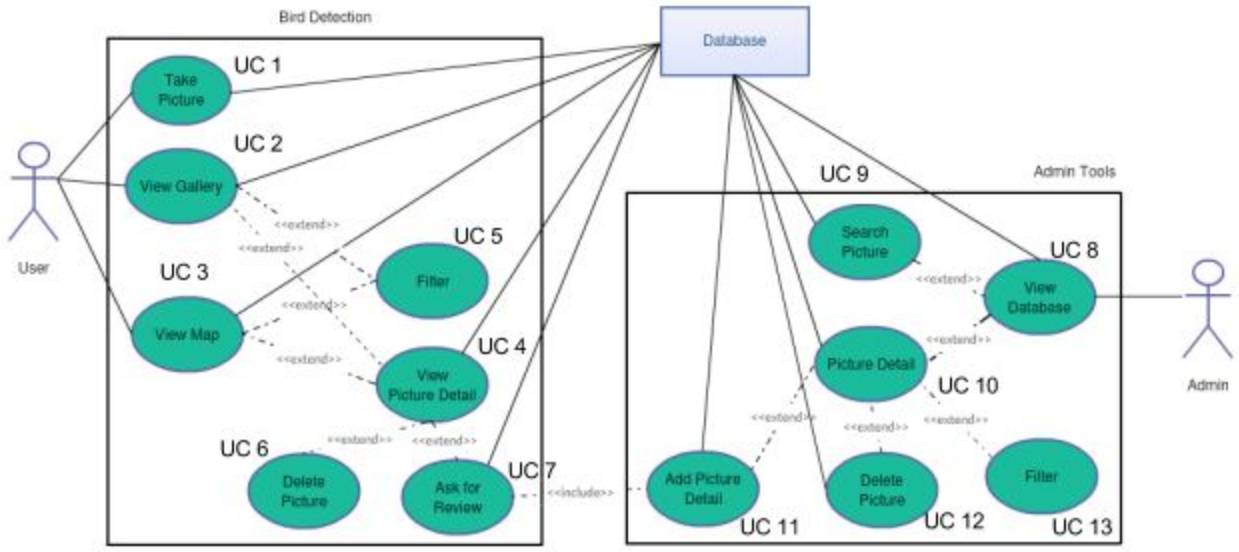

Figure 1. Use Case Diagram

2 Functional Requirements

Functional requirements are expressed in the form of use cases. Please refer to the use case diagram (Figure 1) for an overview of each use case. Priority ranking ranges from 1 (least importance) to 5 (greatest importance).

2.1 Use Case 1 - Take Picture

Number	1	
Name	Take_Picture	
Summary	User takes a picture with the application's camera	
Priority	5	
Preconditions	Application is turned on	
Postconditions	The photo is saved in the user's local storage and sent to the database	
Primary Actor	User	
Secondary Actors	Database	
Trigger	User chooses to take a photograph (e.g., clicks the camera button)	
Main Scenario	**Step**	**Action**
	1	User points the camera to the place he/she wants to take a picture
	2	User chooses a time to take a photo
	3	The photo is sent to the database
	4	The photo is saved in the user's local storage
Open Issues	None	

2.2 Use Case 2 - View Gallery

Number	2
Name	View_Gallery
Summary	User views the list of pictures he/she took

Priority	4	
Preconditions	User took photo(s) and application is in camera mode	
Postconditions	User is shown his/her gallery of photos	
Primary Actor	User	
Secondary Actors	Database	
Trigger	User chooses to transition to gallery (i.e., swipes in the direction of the gallery)	
Main Scenario	**Step**	**Action**
	1	Application transitions to gallery mode
	2	The screen shows the list of photos taken by the user
	3	User may stay or make application return to camera mode
Extensions	**Step**	**Branching Action**
	3a	The user chooses to filter: The user is directed to the use case 5
	3b	The user selects a photograph: The user is directed to the use case 4
Open Issues	None	

2.3 Use Case 3 - View Map

Number	3	
Name	View_Map	
Summary	User views the pins of photo(s) taken in the map	
Priority	4	
Preconditions	User took photo(s) and application is in camera mode	
Postconditions	User is shown location pins of the photo(s) taken	
Primary Actor	User	
Secondary Actors	Database	
Trigger	User chooses to transition to map (i.e, swipes in the direction of the map)	
Main Scenario	**Step**	**Action**
	1	Application transitions to map mode
	2	The screen shows the map and pins
	3	User may stay or make application return to camera mode
Extensions	**Step**	**Branching Action**
	3a	The user chooses to filter: The user is directed to the use case 5
	3b	The user selects a photograph: The user is directed to the use case 4
Open Issues	None	

2.4 Use Case 4 - Viewing Picture Detail

Number	4
Name	View_Picture_Detail
Summary	User views the detail of a photo
Priority	4
Preconditions	Application is in gallery or map mode

Postconditions	Detailed view of the picture is shown to the user	
Primary Actor	User	
Secondary Actors	Database	
Trigger	User selects a photo from the gallery or a pin on the map	
Main Scenario	**Step**	**Action**
	1	The screen shows an enlarged photo with the detail of the photo
Extensions	**Step**	**Branching Action**
	2a	The user chooses to delete the photo: The user is directed to the use case number 6
	2b	The user chooses to mark the photo for review: The user is directed to the use case number 7
Open Issues	None	

2.5 Use Case 5 - Filter Photos

Number	5	
Name	Filter_Photos	
Summary	User views the list of photos based on the desired filter type	
Priority	1	
Preconditions	Application is in gallery or map mode	
Postconditions	User views the gallery or map based on the filter type	
Primary Actor	User	
Secondary Actors		
Trigger	User clicks filter menu button	
Main Scenario	**Step**	**Action**
	1	User selects the type of filter he/she wants
	2	Stored photographs are filtered based on filter chosen
	3	Filtered photographs are shown in gallery or map
Open Issues	None	

2.6 Use Case 6 - Delete Picture

Number	6	
Name	Delete_Picture	
Summary	User deletes a picture	
Priority	2	
Preconditions	User is on the picture detail screen	
Postconditions	The picture is deleted on the user's local storage (picture is not deleted in database) and system returns to gallery/map mode (most recent)	
Primary Actor	User	
Secondary Actors		
Trigger	User clicks the delete button	
Main Scenario	**Step**	**Action**
	1	User is asked for confirmation of photo deletion
	2	User clicks yes button

	3	The photo is deleted from the user's local storage
	4	System returns to gallery/map mode (most recent)
Extensions	**Step**	**Branching Action**
	2a	User clicks no button: User is directed back to the photo detail view
Open Issues	None	

2.7 Use Case 7 - Ask for Review

Number	7	
Name	Ask_for_Review	
Summary	User asks database to review the photo as he/she believes there is a bird in the photo	
Priority	2	
Preconditions	User is on the picture detail view	
Postconditions	The picture will be flagged and sent to the database for additional review	
Primary Actor	User	
Secondary Actors	Database	
Trigger	User clicks on the ask for review button	
Main Scenario	**Step**	**Action**
	1	User is asked to verify the bird existence
	2	The photo will be marked as "pending review"
Extensions	**Step**	**Branching Action**
	2a	The information is sent to the admin: Admin can review this in Use Case 11 (Admin Add Picture Detail)
Open Issues	None	

2.8 Use Case 8 - View Database

Number	8	
Name	View_Database	
Summary	Admin opens an application to view the contents of the database	
Priority	3	
Preconditions	Admin is logged in (as an administrator)	
Postconditions	The contents of the database are shown to the user	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	Admin opens database view	
Main Scenario	**Step**	**Action**
	1	The admin selects to view information in database
	2	System displays contained photographs with associated data
Open Issues	None	

2.9 **Use Case 9 - Admin Search Picture**

Number	9	
Name	Admin_Search_Picture	
Summary	Admin searches pictures in database	
Priority	4	
Preconditions	Admin is viewing the database of pictures	
Postconditions	Pictures taken by the searched user are shown to the admin	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	Admin chooses to search by user ID (e.g., clicks button)	
Main Scenario	**Step**	**Action**
	1	Admin types in the user's ID that he/she wants to search for
	2	Admin confirms search
	3	Database returns to admin a list of all of that user's photos in the database
Open Issues	None	

2.10 **Use Case 10 - Admin Picture Detail**

Number	10	
Name	Admin_Picture_Detail	
Summary	A selected photo's details are shown	
Priority	4	
Preconditions	Admin is viewing the database	
Postconditions	A detailed view of the picture is shown to the admin	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	Admin clicks on a picture	
Main Scenario	**Step**	**Action**
	1	Admin selects a photo from the database
	2	A detailed view of that selected picture is shown to the admin
Open Issues	None	

2.11 **Use Case 11 - Admin Add Picture Detail**

Number	11	
Name	Admin_Add_Picture_Detail	
Summary	Admin adds information to a picture from the database	
Priority	3	
Preconditions	Admin is in Admin Picture Detail view	
Postconditions	Admin completed all additions of new information on the selected photo	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	Admin clicks edit picture details button	
Main Scenario	**Step**	**Action**

	1	Admin edits the information in the selected picture's details
	2	System asks for confirmation
	3	Admin confirms changes
Extensions	**Step**	**Branching Action**
	3a	Admin cancels changes**:** Details are not changed, Admin returns to original detailed picture view
Open Issues	None	

2.12 **Use Case 12 - Admin Delete Picture**

Number	12	
Name	Admin_Delete_Picture	
Summary	Admin deletes a photo from the database	
Priority	4	
Preconditions	Admin is in Admin Picture Detail view	
Postconditions	The picture is deleted from the database	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the delete button	
Main Scenario	**Step**	**Action**
	1	Admin is asked to confirm photograph deletion
	2	Admin confirms deletion
	3	The photo is deleted from the database
Extensions	**Step**	**Branching Action**
	2a	Admin cancels deletion: Admin is directed back to photo detail view
Open Issues	None	

2.13 **Use Case 13 - Admin Filter**

Number	13	
Name	Admin_Filter	
Summary	Admin views the list of photos based on the desired sort type	
Priority	1	
Preconditions	Admin is viewing the photos in the database view	
Postconditions	Admin views the pictures based on the filter type	
Primary Actor	Admin	
Secondary Actors	Database	
Trigger	Admin clicks filter button	
Main Scenario	**Step**	**Action**
	1	Admin chooses the type of filtering he/she wants
	2	Stored photographs are filtered based on filter chosen
	3	Filtered photos are shown in database view
Open Issues		None

3 Use Case Tests

To ensure that client requirements are met, the final product shall be tested using the following tests. See Testing Report for further details.

3.1 Use Case 1 Test - Take Picture

After choosing to take a picture, confirm the phone's camera takes a photograph. The photograph must appear in the phone's local storage. See Testing Report for further details and results.

3.2 Use Case 2 Test - View Gallery

Upon choosing to view the gallery, confirm that all photographs taken through the application appear in the gallery. See Testing Report for further details and results.

3.3 Use Case 3 Test - View Map

Upon choosing to view the map, confirm that a map view is displayed with locations of taken photographs. See Testing Report for further details and results.

3.4 Use Case 4 Test - View Picture Details (from Gallery)

Upon selecting a photograph from the gallery, confirm that a "photo detail" view is shown for the selected photograph. See Testing Report for further details and results.

3.5 Use Case 4 Test - View Picture Details (from Map)

Upon selecting a photograph from the map, confirm that a "photo detail" view is shown for the selected photograph. See Testing Report for further details and results.

3.6 Use Case 5 Test - Filter Photos (can choose from Gallery)

Upon choosing to filter while viewing the gallery, confirm that the user is presented with different options for filtering the photographs. See Testing Report for further details and results.

3.7 Use Case 5 Test - Filter Photos (is applied in Gallery)

Upon choosing a filter option while viewing the gallery, confirm that the selection of displayed photographs is now restricted by the selection option. See Testing Report for further details and results.

3.8 Use Case 5 Test - Filter Photos (can choose from Map)

Upon choosing to filter while viewing the map, confirm that the user is presented with different options for filtering the photographs. See Testing Report for further details and results.

3.9 Use Case 5 Test - Filter Photos (is applied in Map)

Upon choosing a filter option while viewing the map, confirm that the selection of displayed photograph locations is now restricted by the selection option. See Testing Report for further details and results.

3.10 Use Case 6 Test - Delete Picture

Upon selected a picture for deletion, confirm that the user has proper credentials and that selected picture is removed from the database correctly. See Testing Report for further details and results.

3.11 Use Case 7 Test - Ask for Review

Upon selecting a picture for additional review, confirm that the user picture is then sent to the python server and reviewed an additional time. Also confirm that the results are sent back to the user correctly. See Testing Report for further details and results.

3.12 Use Case 8 Test - View Database

Upon a user logging in as admin, confirm that they are able to view the database correctly. See Testing Report for further details and results.

3.13 Use Case 9 Test - Admin Search

Upon a user logging in as admin and entering parameters into the search field, confirm that the only entries listed are the the ones corresponding to the parameters. See Testing Report for further details and results.

3.14 Use Case 10 Test - Admin Picture Detail

Upon a user logging in as admin and selecting a picture, confirm that the correct photo details are displayed for the selected picture. See Testing Report for further details and results.

3.15 Use Case 11 Test - Admin Add Picture Detail

Upon a user logging in as admin, selecting a photo, and adding details to it, confirm that the correct details are then added to the selected photo. See Testing Report for further details and results.

3.16 Use Case 12 Test - Admin Delete Picture

Upon a user logging in as admin selecting a picture to delete, confirm that the selected photo was removed from the database. See Testing Report for further details and results.

3.17 Use Case 13 Test - Admin Filter

Upon a user logging in as admin and selecting a sorting filter, confirm that the entries are restricted based on the selected filter. See Testing Report for further details and results.

4 Non-Functional Requirements

The client desires the product to have certain properties. Below is a prioritized list (1 is lowest, 5 is highest) of system constraints. See Testing Report for further details and results of testing.

#	Item	Priority	Specific Constraints
1	Application shall run on Android phones	5	Remaining tests (functional and nonfunctional) will be passed on at least two Android devices
2	Application must show status of photo classification	5	Customer must be able to distinguish between different classification statuses
3	Map will be made with GoogleMaps API	5	GoogleMaps API will be used for map portion

4	Classification must be done with a neural network	5	A server-side convolutional neural network will be used for classifying images
5	Application shall be able to take photos in real time	4	Average forced delay between photos must be less than 2 seconds
6	Application shall transition between views smoothly	4	Average delay between view changes must be less than 2 seconds; no tearing of animation
7	User can return to map/gallery from detailed view	4	From detail view of a specific photo, will be method to return to view from which the photo was selected
8	All code shall be documented	3	Each method's pre- and post-conditions and purpose of each class must be documented
9	Picture classification shall fail gracefully	3	Application will not fail if classification fails; a suitable classification status will be set
10	User can filter by classification	3	Filter options will allow user to see photos of a chosen status (pending, confirmed bird, etc.)
11	Database images must be tagged with User ID	3	Each image in the database will be tagged based on submitting user
12	User ID will not compromise user privacy	3	User IDs will be securely hashed to prevent storage of device/personal information
13	Map pins will be colored based on photo classification	2	Map pin color will be determined by consistent classification status coloring
14	Gallery will display newest photos first	2	Photos will appear in chronological order (newest first) within the gallery view
15	Application will be color-blind friendly	1	No red-green color-coding will be used

5 User Interface

See "User Interface Design Document for Birdry."

6 Deliverables

Deliverables include:

Individual Electronic (PDF) Copies of Each of the Following:

- System Requirement Specification (March 15, 2017)
- System Design Document (April 5, 2017)
- User Interface Design Document (April 5, 2017)
- Test Report (May 3, 2017)
- Code Inspection Report (May 3, 2017)
- Administrator Manual (May 10, 2017)

An Electronic Copy in a ZIP File Containing the Following (May 15, 2017):

- System Requirement Specification
- System Design Document
- User Interface Design Document
- Test Report
- Code Inspection Report
- Administrator Manual
- All source code
- An executable APK file
- All software and code required to run the Python server

7 Open Issues

There have not yet been any issues which have been raised and do not yet have a conclusion. Should any appear in the future, they will be included in this section in a future draft.

Appendix A **– Agreement Between Customer and Contractor**

We, the clients and members of the development team, undersign here to indicate our agreement to the development and delivery of a mobile application for the identification and classification of pictures depicting birds, with classification functionality provided by an artificial intelligence framework running on a remote server.

If future changes are necessary, a revised hard copy will be presented to the client. Once approved by the client and signed off by all team members, the revisions will come into effect.

Clients

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Team

Name: _____ Signature: _____ Date: _____

Name: _____ Signature: _____ Date: _____

Name: _____ Signature: _____ Date: _____

Name: _____ Signature: _____ Date: _____

Name: _____ Signature: _____ Date: _____

Appendix B **– Team Review Sign-off**

We, the members of the development team, undersign here to indicate that every member of the team has reviewed and approved the contents and format of this system requirements specification, and that their comments and/or concerns (if any) are listed here.

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Name: _____ Signature: _____ Date: _____

Comments:

_____

_____

Appendix C **– Document Contributions**

Below is an estimate of the contribution from each team member, including specific work done and percentage of total document completed.

Kyle Fritz
Contributions:
- Use Cases 2.9-2.13
- Grammar check of entire document

Est. Contribution: 17%

Laras Istiqomah
Contributions:
- Figure 1 creation
- Use Cases 2.1-2.8

Est. Contribution: 23%

David Leiberg
Contributions:
- Intent clauses for Appendices A, B, C
- Helped organize deliverables in section 6
- Reviewed priority ranking on use cases 2.1 - 2.6
- Labelled use case diagram in section 1 and finished section 1.4

Est. Contribution: 15%

Julian Sniffen
Contributions:
- Finished 1.1 (Purpose of This Document)
- Finished (1.3 Purpose of the Product)
- Use Case Tests 3.10-3.16
- Initial sketch of 6.1 (Deliverables)
- Grammar check of entire document

Est. Contribution: 20%

Nicholay Topin
Contributions:
- Coordination with clients (initial meetings)
- Revision of Use Cases 2.1-2.13
- Use Case Tests 3.1-3.9
- Nonfunctional Requirements 1-15
- Final polishing
- Coordination with clients (review sign-off)

Est. Contribution: 25%