

# design documentation



relax, enjoy the birds

# contents

---

twitchin overview

use case 1 - identify birds sighted

use case 2 - create lists of birds sighted

the twitch action

viewing and editing a twitch record

working with lists

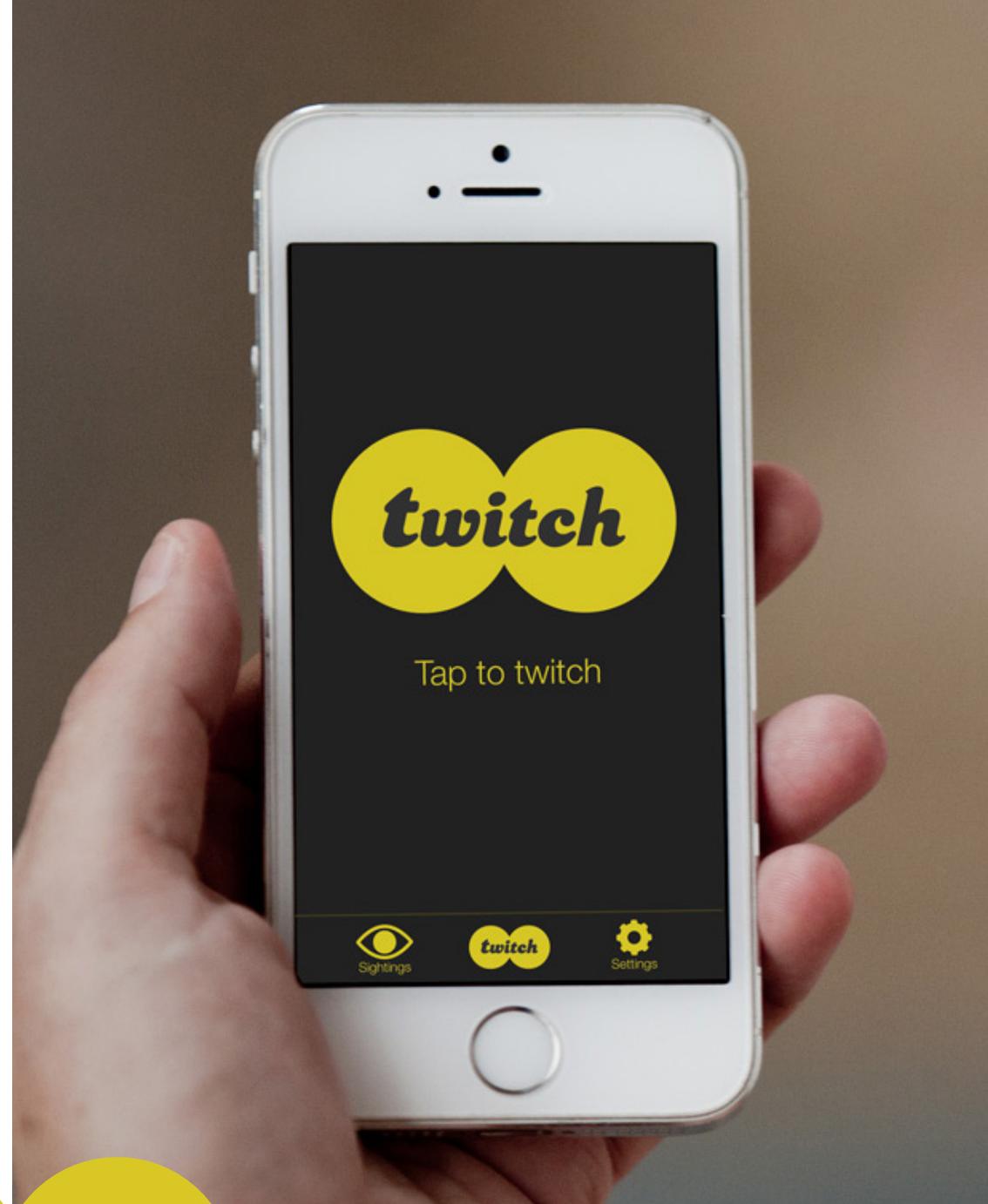
wireframes for primary screens

visual design and branding guidelines

functional specifications

technical considerations

UX design considerations



**twitchin**

# twitchin overview

---

Twitchin is an app for twitchers aka birdwatchers. It is designed to help twitchers spend less time fussing over books and notes, and more time enjoying the birds.

With one tap of the twitch button, Twitchin records sound, geolocation and the time of the twitch. The idea is that one twitch equals one sighting.

Automating this data recording makes it easier for a twitcher to identify birds, because they don't have to rely on memory alone.



**twitchin**

# use case 1

---

## identify birds sighted

Birdwatchers like to identify birds that they have seen. They usually consult a bird guide either while in the field or when they return home from a twitching session.

To correctly identify a bird using a bird guide, a twitcher requires certain data to compare with the guide e.g. shape, colour, sound, behaviour, etc. The twitcher must remember all of this information and then compare it to records in a bird guide to identify the bird. Trying to record this information in the field can be a distraction from the enjoyment of watching the birds.

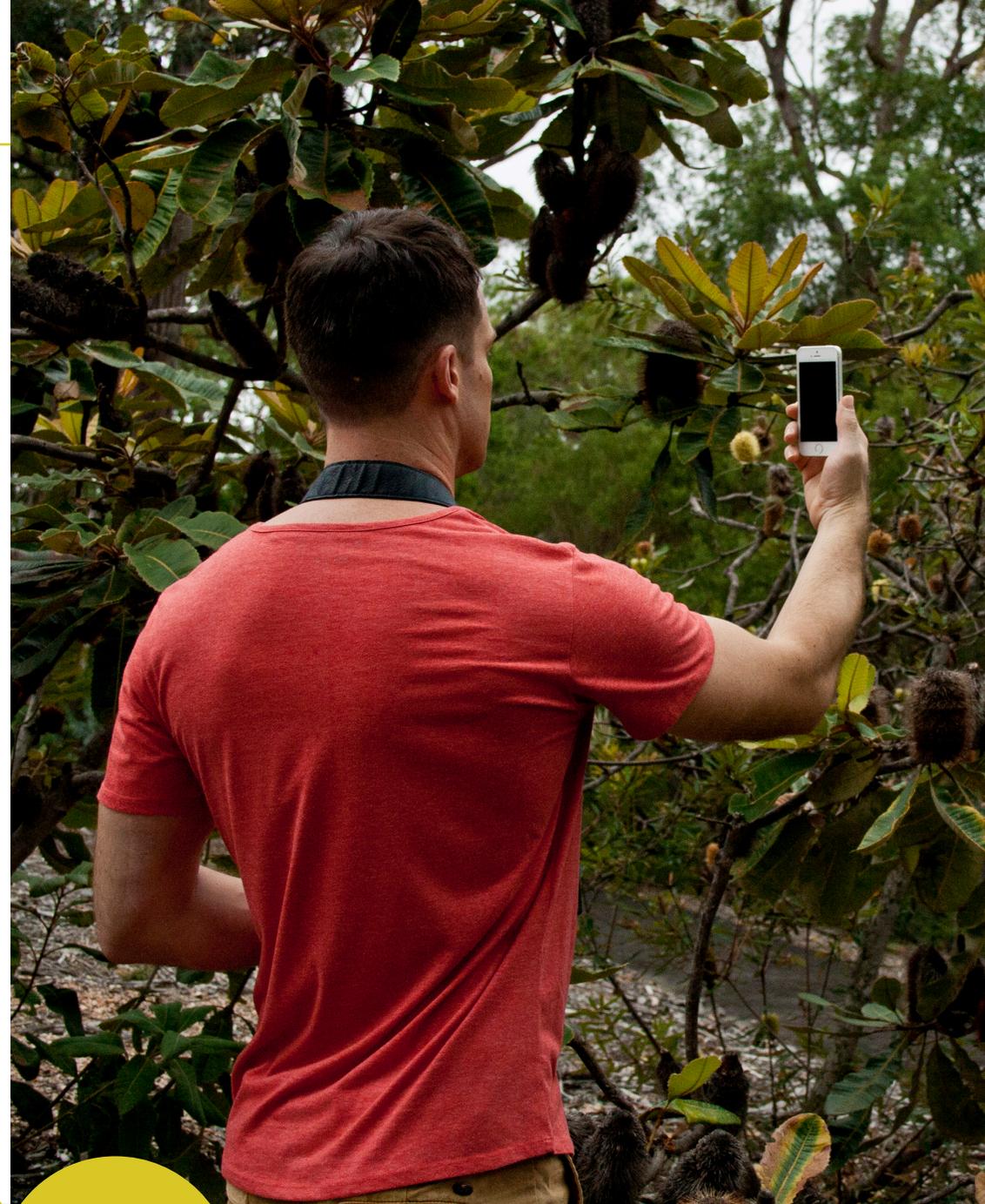
### Context

In the field:

Twitchin records data - sound, geolocation, time and notes.

At home:

Compare twitchin data records with bird guide to identify birds.



**twitchin**

## use case 2

### create lists of birds sighted

Birdwatchers like to make lists of the birds they identify, to keep a record of all the birds they've seen and where they've seen them. These lists may be sorted by birds seen at a location, across multiple locations according to a specific field trip, or even by bird species.

The twitcher needs to go through their book or books to find each identified bird and add them to the list. They may end up with lots of disparate lists, sometimes more than one for a single location. Because these lists are usually created in a notebook they won't include sounds or images and will often never see the light of day after they have been created.

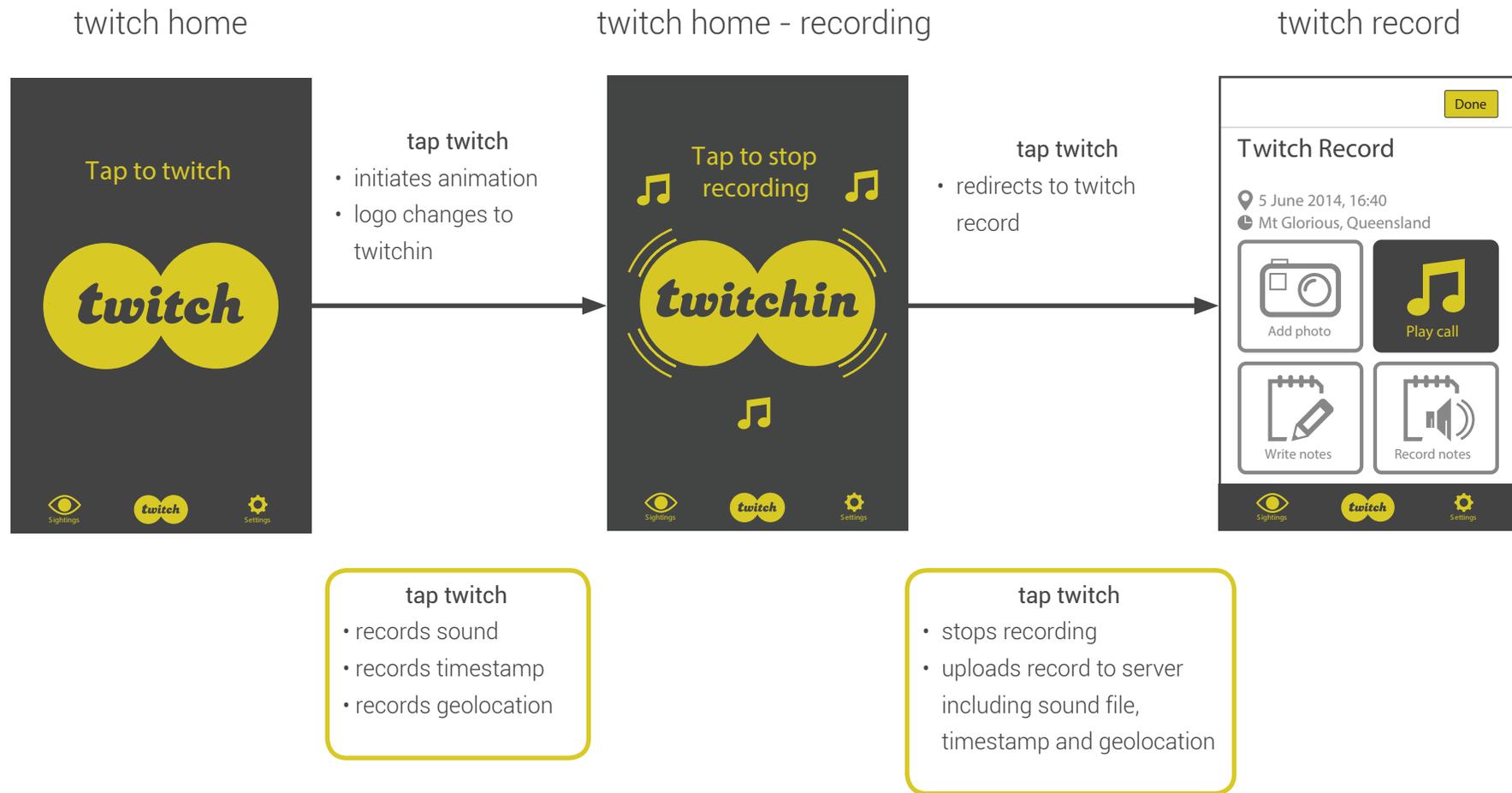
### Context

This is generally done at home, or back at the hotel. With twitchin many of these lists will be sorted and created automatically, but custom lists can also be created, that will allow list making to become more mobile.



**twitchin**

# the twitch action



# viewing and editing a twitch record

initial record  
- unedited

complete record  
- all fields edited

- seamless inline editing - tap the record title and edit directly



- Done button returns to twitch home

- click to toggle play-pause of the recorded bird call

- once a field has been edited, it's box and label changes to it's edited state - image or dark version of icon

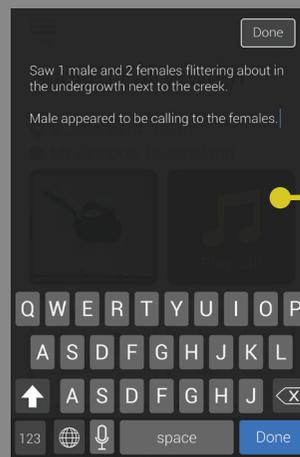
add a photo

add notes

add sound notes



- photo selection via ios native selection
- choose photo library or native camera



- text notes added using ios native notes skinned to match the twitchin look and feel

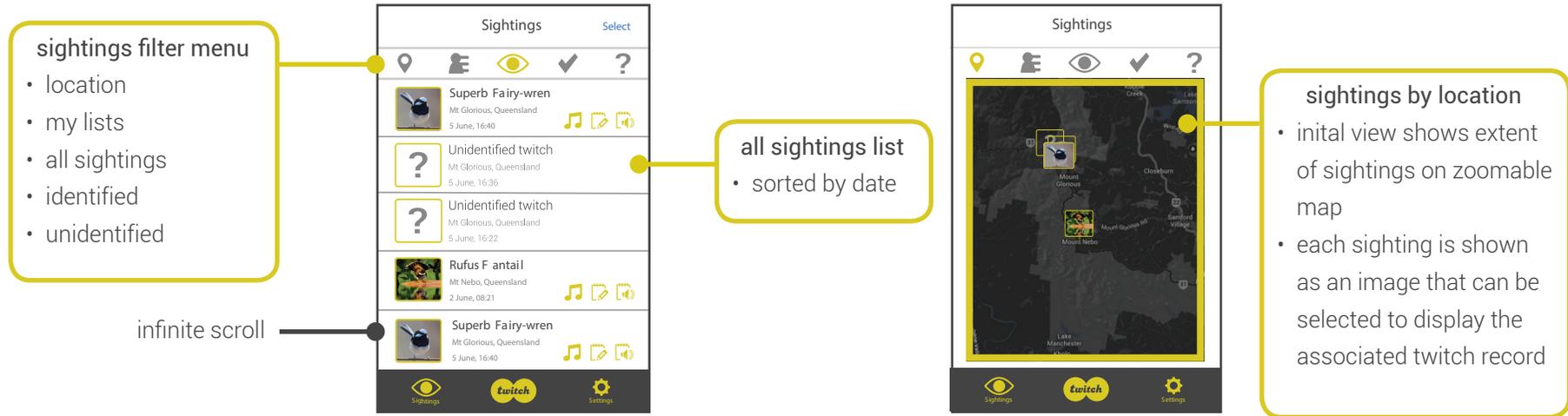


- add sound notes using the ios native recorder (same as for siri)

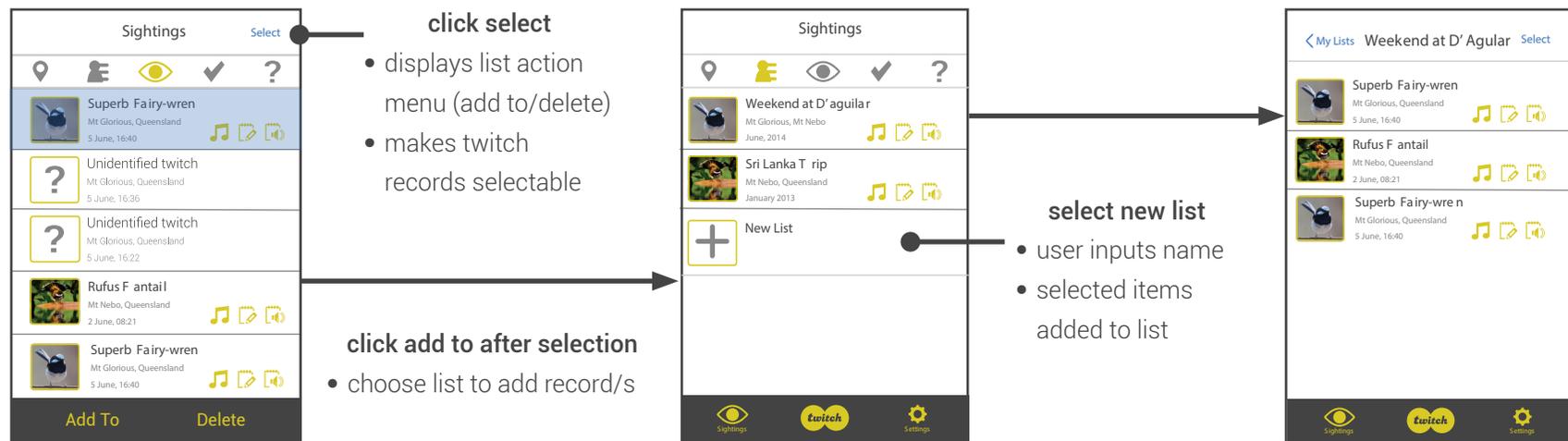


# working with lists

## sorting lists



## custom lists



# wireframes for primary screens

## main twitch

portrait only: w320px x h480px  
margin: 10px 20px 50px 20px



## twitch record

portrait only: w320px x h480px  
margin: 10px 20px 50px 20px



## twitch list

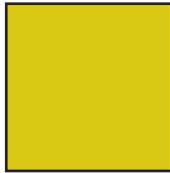
portrait only: w320px x h480px  
margin: 10px 20px 50px 20px



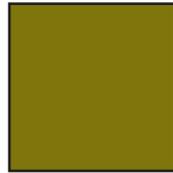
# visual design and branding guidelines

---

## colour palette



#dac914



#7e750c



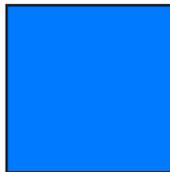
#444444



#888888



#cccccc



#007aff

## branding



Can be used on any dark background, including images.



Can be used on any light or bright coloured background e.g. red



This is the action button. To be used only on the twitchin home screen.

**Note:** it is possible that the two very similar logos could cause confusion in the brand. This will require testing.

# functional specifications

---

## twitch action

- Initial tap of the twitch button triggers recording of timestamp, geolocation and initiates sound recording. Visually, the twitch changes to twitchin and some form of animation will be displayed, to provide feedback that the app is recording.
- Second tap of the twitch button must stop the recording, write the sound to a file and forward the entire record (sound, time, location) to the server, then redirect to the new record's page.

## viewing and editing twitch records

- This process must be as simple as possible, with the number of touches kept to a minimum.
- As well as the recorded sound file, users can add images, written and sound notes, as evidence to assist with bird identification.
- An evidence type's icon will change from an empty icon to a full (or dark) icon, depending on whether there is any evidence recorded for it.
- Listening to the recorded sound should be a tap to start and to stop. No need for an overlay screen like in the prototype, just toggle play/pause in the 'play call' box.
- Adding of imagery, written and sound notes will all follow ios native functionality e.g. camera, photo library, notes and keyboard, etc.

## working with lists

- twitch records will be displayed in a number of pre-sorted lists that can be accessed via 'sightings' in the main nav menu
- pre-sorted lists are displayed using the icon control menu at the top of the sightings page and include:



**All sightings** - all twitch records displayed in chronological order



**Identified** - all twitch records that have been identified i.e. bird name has been changed



**Unidentified** - all twitch records that have not been identified i.e. no name change



**Location** - all twitch records shown on a map. Map begins zoomed out to the extent of all locations. Users can zoom in. Twitch records are displayed as their associated images and selecting them will redirect the user to that record's page.



**My Lists** - displays custom lists created by the user. Lists are created by selecting records from the other lists, using the select command and then selecting the adding them to an existing list or creating a new list. Will work the same as list creation in ios photo library.

**twitchin**

# technical considerations

---

## native vs browser based app

Initial investigations of technological specifications were focused on creating twitchin as a browser based app that would utilise HTML5 audio recording. However, due to a current lack of browser support for this method. It was decided that twitchin should be created as a native app, so that the device's native recording, geolocation and other data sources may be accessed more reliably. Initial development would be for iPhone only. SoundCloud's native iOS app is a good example of how a sound recorder can work natively.

## SoundCloud vs dedicated server

SoundCloud has been investigated as a potential place to store sounds and metadata and alleviate the need for a dedicated server. It would also provide ready access to a community for sharing bird sighting data.



SoundCloud allows individuals to store up to 120 minutes of audio on a free account. The average bird call audio file would most likely be around 20 seconds and 300KB. So,

users could store around 360 twitchin records for free, then have an option to purchase more storage space.

As things currently stand, it is difficult to upload prerecorded audio files to soundcloud directly using their HTML5 API. However, it is possible to forge a partnership with SoundCloud, to allow this to happen.

A partnership with SoundCloud could be beneficial to twitchin to provide an existing, successful community for twitchers to interact in. This could also be sold to SoundCloud as a means for them to increase their userbase and expand into a non-musical audio market.

## working offline

This prototype assumes that an internet connection is always available. However, a functional version could use the HTML5 offline application cache to allow it to run without a network connection. Data could be stored using HTML5 local storage which allows data to persist across future sessions. When connectivity is restored the data can synchronize with the remote server. Using local storage is a good idea anyway, as it provides data backup in case of upload failures or system crashes.

**twitchin**

# UX design considerations

---

## starting and stopping recording

Three methods for starting and stopping recording during the twitch action were investigated. Each method had benefits and pitfalls:

- **tap to start and tap again to stop** - this method was thought to decrease the integrity of the single tap twitch, however it would give the twitcher control over when they record and for how long.
- **timer based recording** - Using a timer would hold the integrity of the single touch twitch, however it would give the twitcher very little control over the recording - they could miss a call.
- **time-lapse style (e.g. Vine)** - this method is really novel and provides some interesting results for video, but for trying to capture bird calls it would be problematic as you need to concentrate on when to touch the screen.

For the purpose of this prototype it was decided that the tap to start method was most in keeping with capturing an audio file of a bird call.

## usability testing

No usability testing has been performed to date on this application design. Once the prototype is complete, testing in the field and laboratory should be performed to ensure that the UX design is optimal.

It is recommended that each of the methods of starting and stopping recording be tested.

The logo for 'twitchin' is located at the bottom center of the page. It consists of the word 'twitchin' in a bold, lowercase, sans-serif font, colored yellow. The text is set against a dark grey background that is shaped like two overlapping circles, creating a stylized 'O' or 't' shape behind the letters.

**twitchin**