

Programming Boot Camp

# Database design and data manipulation with Adalo

Tokyo Institute of Technology 2021/11/6

Ryo Imahashi

# Table of Contents

- Review previous lecture and check the goal of this lecture
- Introduction to databases
- Database design
- Data manipulation
- Improve sample application
- Exercises
- Summary

## Review previous lecture and check the goal of this lecture

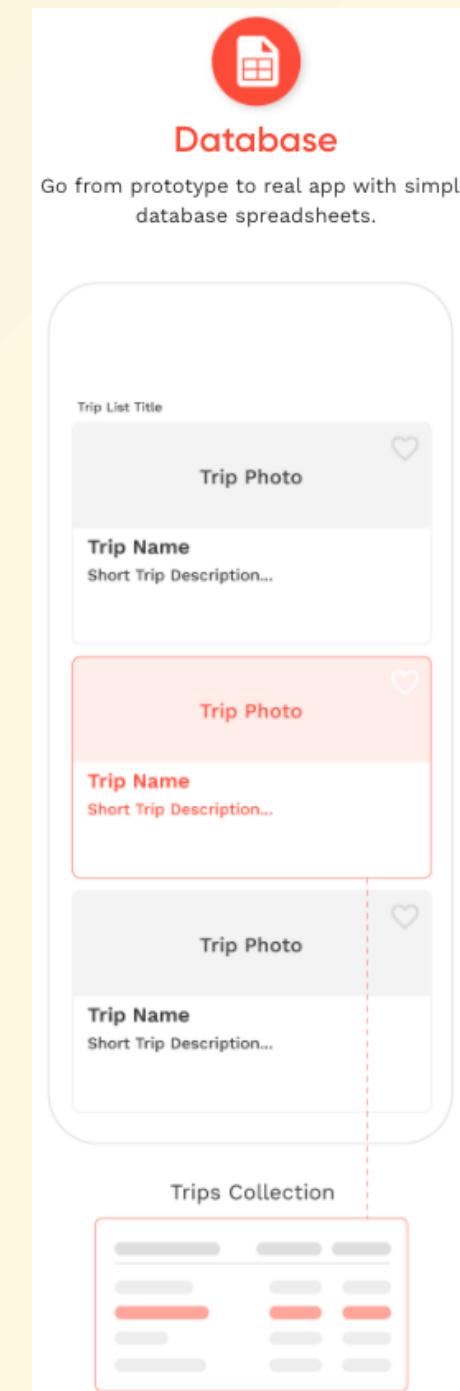
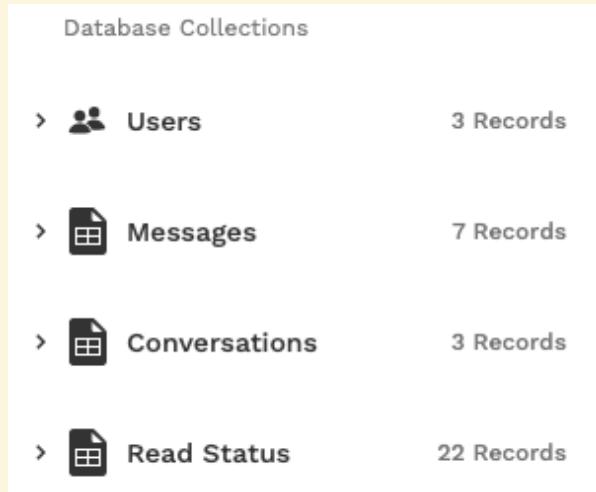
- In the previous lecture, we learned Adalo, a no-code tool, and created the UI of the pet health management app.
  - In the lecture, we used simple components that does not require a database. (Some of you may have used a database in the exercises).
- In this lecture, we will design a database that matches the UI we created in the previous lecture, so that we can manipulate the data from the app.
- After that, we will improve the sample app learning some features of Adalo, and finally, we will do app development exercises and presentation.

# Introduction to databases

First, let's learn about databases.

# Database(Previous lesson)

- A set of organized data.
- Data can be registered, read (displayed), updated, and deleted.
- Example: In the case of a Chat application



- Databases are often compared to "spreadsheet-like" software.
- A database can be used to CREATE, READ, UPDATE, and DELETE data. These operations are collectively called CRUD operations.

Users						<a href="#">+ ADD USER</a>				
	<a href="#">A</a> Email	Password	<a href="#">A</a> Username	<a href="#">A</a> Full Name	Profile Picture					
<input type="checkbox"/>	fuga@hoge.com	[hidden]		三人目のユーザー						
<input type="checkbox"/>	hoge@fuga.com	[hidden]		東 工大						
<input type="checkbox"/>	imahashi@example.com	[hidden]		今橋 謙						

# Basics of Adalo's database



You can access Adalo's database from this icon.  
There are three components of the Adalo database.

- Collection
- Property
- Record

# What is a Collection?

A collection of data that has the same property.

The screenshot shows a database interface with a sidebar on the left and a main table view on the right.

**Left Sidebar:**

- + Database Collections
- Database Collections
- Users (selected, 3 Records)
- ⋮
- Email
- Password
- Username
- Full Name
- Profile Picture
- Conversations (Creator)s
- Messages (Sender)s
- Read Statuses
- Conversations (Members)s
- Friends
- + ADD PROPERTY
- Messages (7 Records)
- Conversations (3 Records)

**Main View:**

**Table Headers:**

	A Email	>Password	A Username	A Full Name	Profile Picture
--	---------	-----------	------------	-------------	-----------------

**Data Rows:**

<input type="checkbox"/>	fuga@hoge.com	[hidden]		三人目のユーザー	
<input type="checkbox"/>	hoge@fuga.com	[hidden]		東 工大	
<input type="checkbox"/>	imahashi@example.com	[hidden]		今橋 陵	

**Top Right Buttons:**

- + ADD USER
- Upload icon
- Download icon
- Sync icon
- Search icon

**Bottom Right Buttons:**

- DONE

- Collection is used to divide and organize the various data.(An analogous term is table.)
- In many cases, a Collection is a group of data that a user can register, update, or delete in a single operation. <!-- (A collection is often said to be something that can be expressed as a noun.
- By default, Users is prepared as a Collection, and the rest can be added according to the application to be developed.

It is very difficult to decide what kind of collection to add. Let's practice and get used to it. (If you have any problems, consult with mentors.)

## What is Record?

- Record is a unit of information in a collection.
  - One row corresponds to one Record.
- In the example of Users Collection, the information belongs to one user is registered as one Record.

Users					
	<input type="checkbox"/> Email	<input type="checkbox"/> Password	<input type="checkbox"/> Username	<input type="checkbox"/> Full Name	<input type="checkbox"/> Profile Picture
<input type="checkbox"/>	fuga@hoge.com	[hidden]		三人目のユーザー	
<input type="checkbox"/>	hoge@fuga.com	[hidden]		東 工大	
<input type="checkbox"/>	imahashi@example.com	[hidden]		今橋 謙	

**+ ADD USER**    

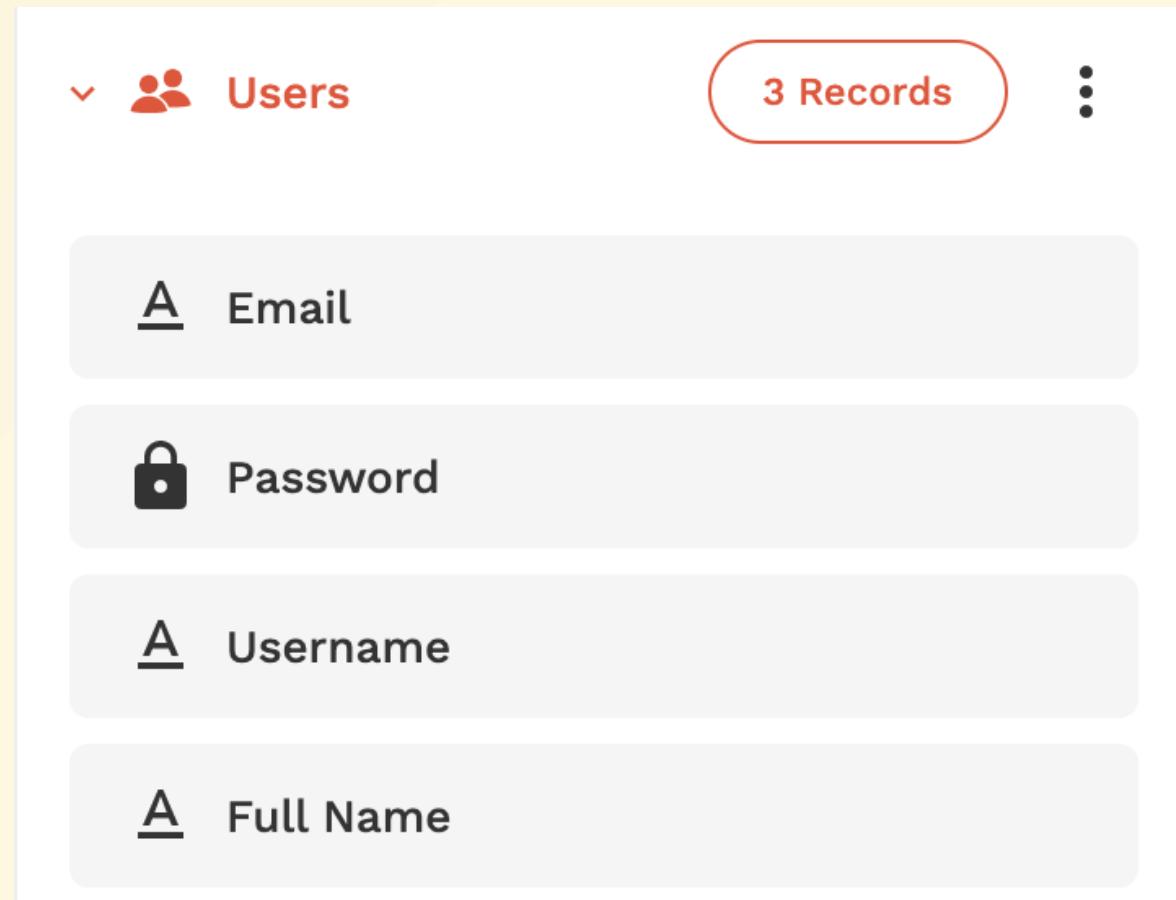
**DONE**

- Records are basically registered from the form on the screen of the app, but it is also possible to register from the form by pressing the "+Add xxxx" button on the upper right in Record View.
- You can also search for Records in the Collection, and upload (import) and download CSV files.

The screenshot shows a mobile application interface for creating a new user account. The title bar at the top says "New User". The form consists of five input fields: "Email" (with an "A" icon), "Password" (with a lock icon), "Username" (with an "A" icon), "Full Name" (with an "A" icon), and "Profile Picture" (with a camera icon). Below the profile picture field is a placeholder text "Choose image". At the bottom right are two buttons: "CANCEL" and a red "SAVE" button.

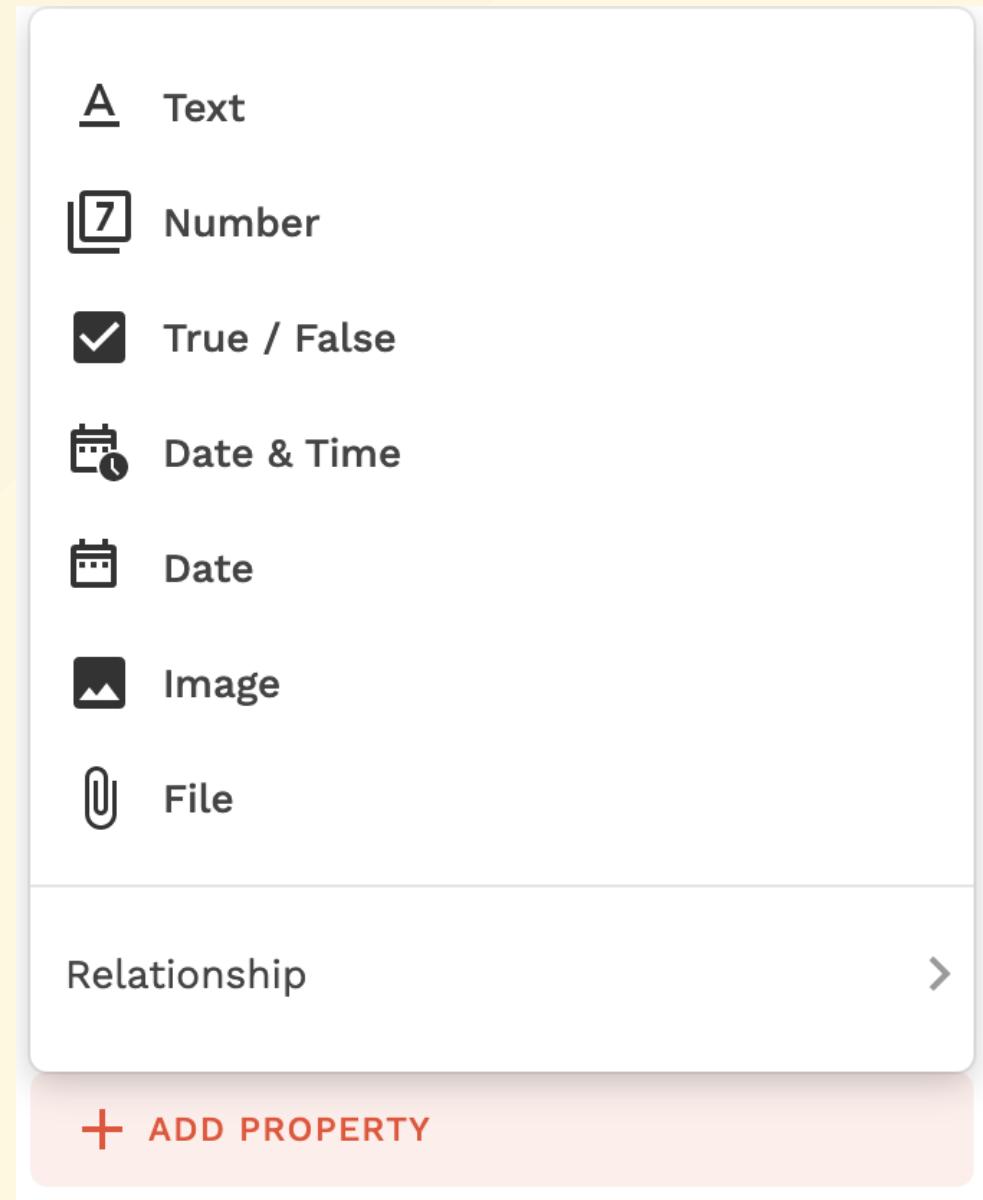
## What is Property?

- Property is each and every item that makes up a Record.
- The Users Collection consists of properties such as email, password, user name, and name.
- The value of Property can be empty.



To define what kind of data the Property is, select the Type when adding it.

- Text
- Number
- True/False
- Date/Time
- Date/Time
- Image
- File
- Relationship

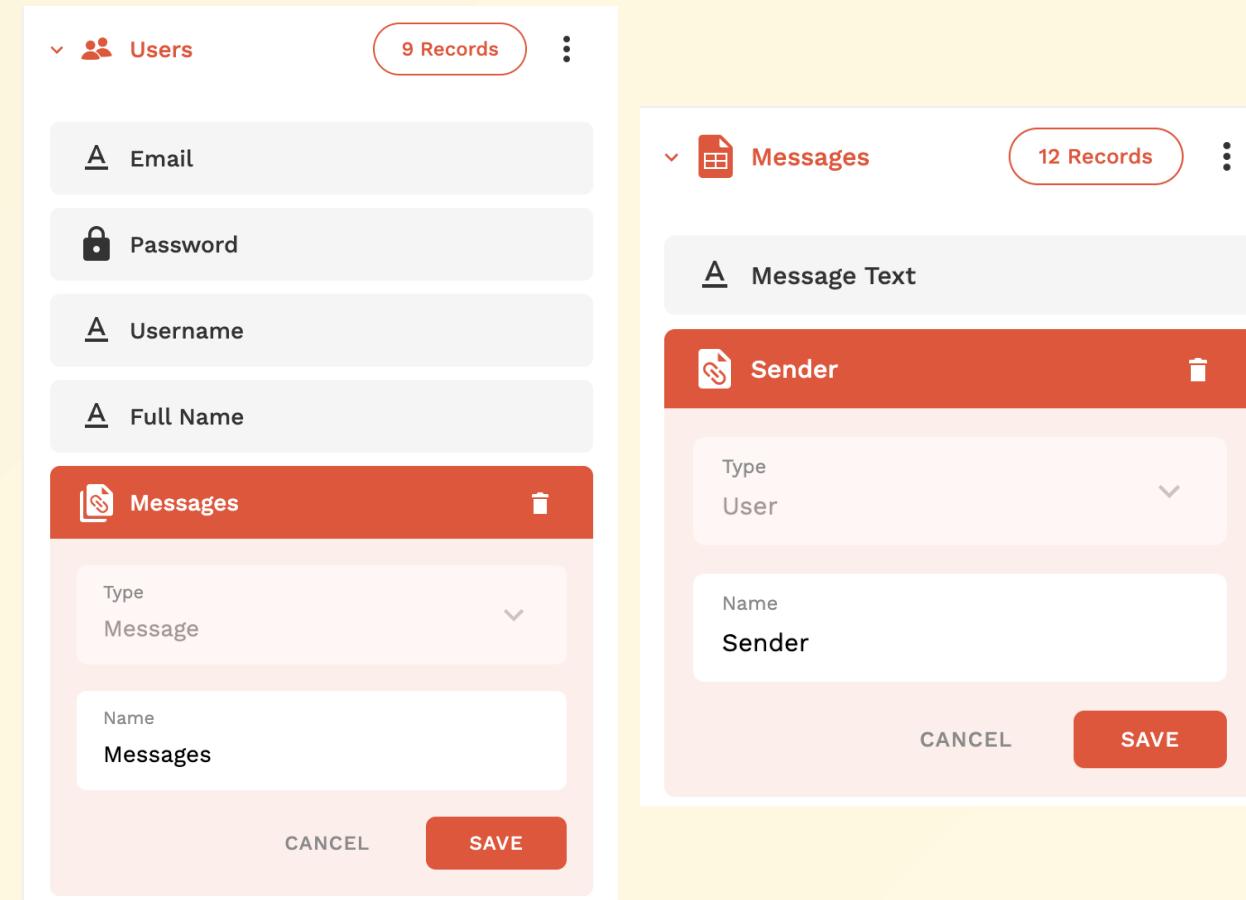


## What is Relationship?

- Instead of storing a large number of properties for a single Record, we can set a special property to relate multiple Collections, called Relationship. This allows you to divide a Collection into manageable pieces.

For example, a message sent by a user in the Chat app is stored in the Messages Collection, which is separate from the Users Collection, and these two collections are related by Relationship.

The Users side has a Relationship called Messages, and the Messages side has a Relationship called Sender (with Users).

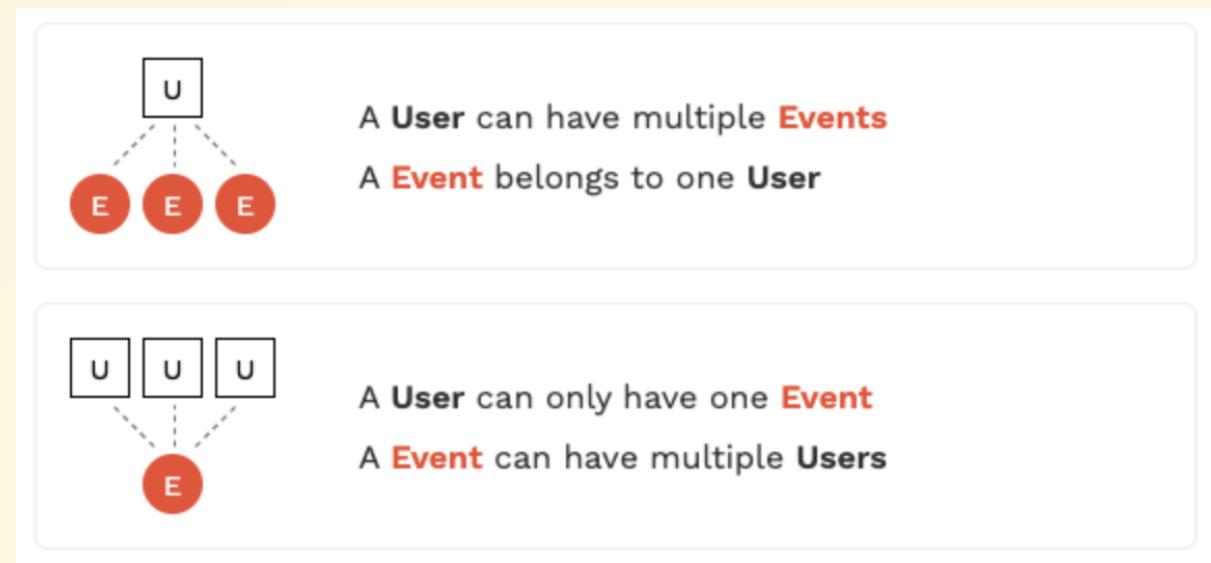


## Types of Relationship

- In Adalo's Relationship, you can choose one of two types, one-to-many or many-to-many, depending on the number of Records associated with the Collection.

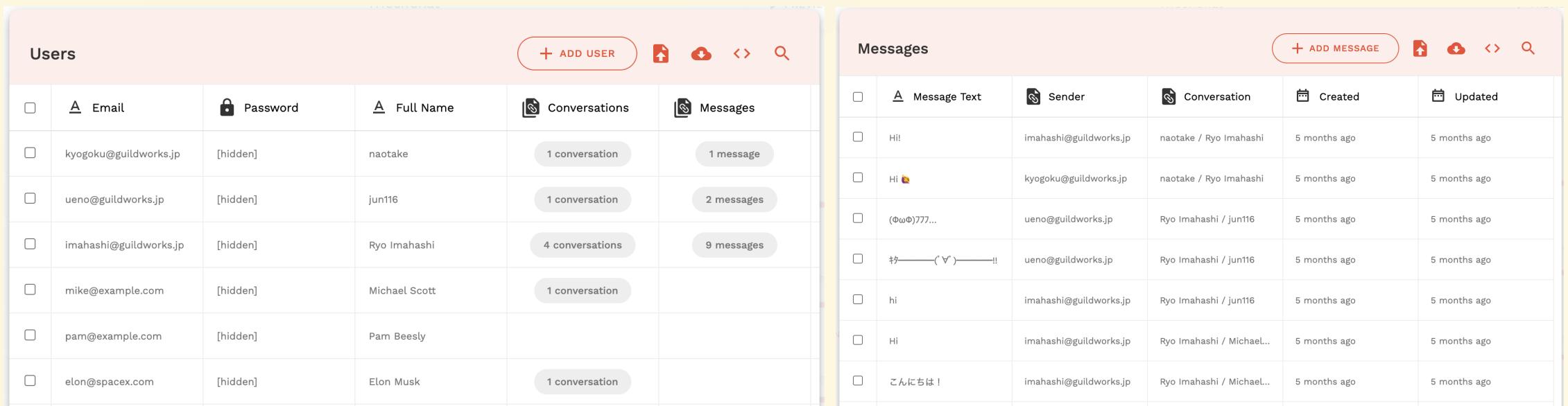
# One-to-many Relationship

- This means that one Record has a relationship with multiple Records in different collections.
- Depending on whether the Collection you are trying to set the Relationship to is one or many, two choices will appear.



# Example of a one-to-many Relationship

- In the Chat application, one user sends multiple messages, but the sender of the message is one user, so the Relationship in the Users Collection and Messages Collection is one-to-many.



The image shows two tables side-by-side, illustrating a one-to-many relationship between the 'Users' collection and the 'Messages' collection.

**Users Table:**

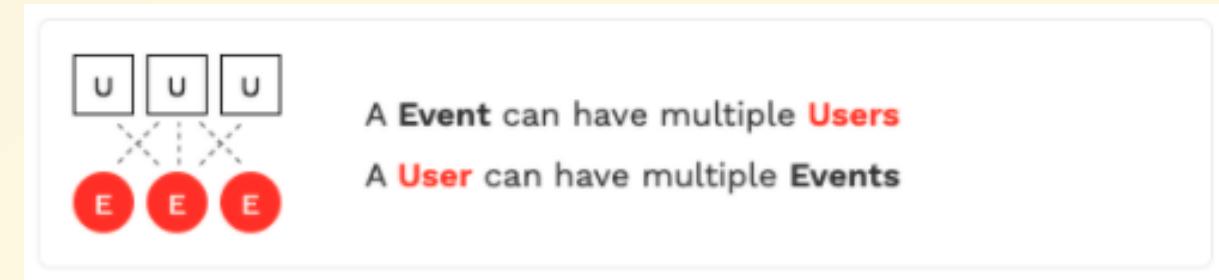
	Email	Password	Full Name	Conversations	Messages
<input type="checkbox"/>	kyogoku@guildworks.jp	[hidden]	naotake	1 conversation	1 message
<input type="checkbox"/>	ueno@guildworks.jp	[hidden]	jun116	1 conversation	2 messages
<input type="checkbox"/>	imahashi@guildworks.jp	[hidden]	Ryo Imahashi	4 conversations	9 messages
<input type="checkbox"/>	mike@example.com	[hidden]	Michael Scott	1 conversation	
<input type="checkbox"/>	pam@example.com	[hidden]	Pam Beesly		
<input type="checkbox"/>	elon@spacex.com	[hidden]	Elon Musk	1 conversation	

**Messages Table:**

	Message Text	Sender	Conversation	Created	Updated
<input type="checkbox"/>	Hi!	imahashi@guildworks.jp	naotake / Ryo Imahashi	5 months ago	5 months ago
<input type="checkbox"/>	Hi 🎉	kyogoku@guildworks.jp	naotake / Ryo Imahashi	5 months ago	5 months ago
<input type="checkbox"/>	(ΦωΦ)777...	ueno@guildworks.jp	Ryo Imahashi / jun116	5 months ago	5 months ago
<input type="checkbox"/>	♪————( ∀ )————!!	ueno@guildworks.jp	Ryo Imahashi / jun116	5 months ago	5 months ago
<input type="checkbox"/>	hi	imahashi@guildworks.jp	Ryo Imahashi / jun116	5 months ago	5 months ago
<input type="checkbox"/>	Hi	imahashi@guildworks.jp	Ryo Imahashi / Michael...	5 months ago	5 months ago
<input type="checkbox"/>	こんにちは！	imahashi@guildworks.jp	Ryo Imahashi / Michael...	5 months ago	5 months ago

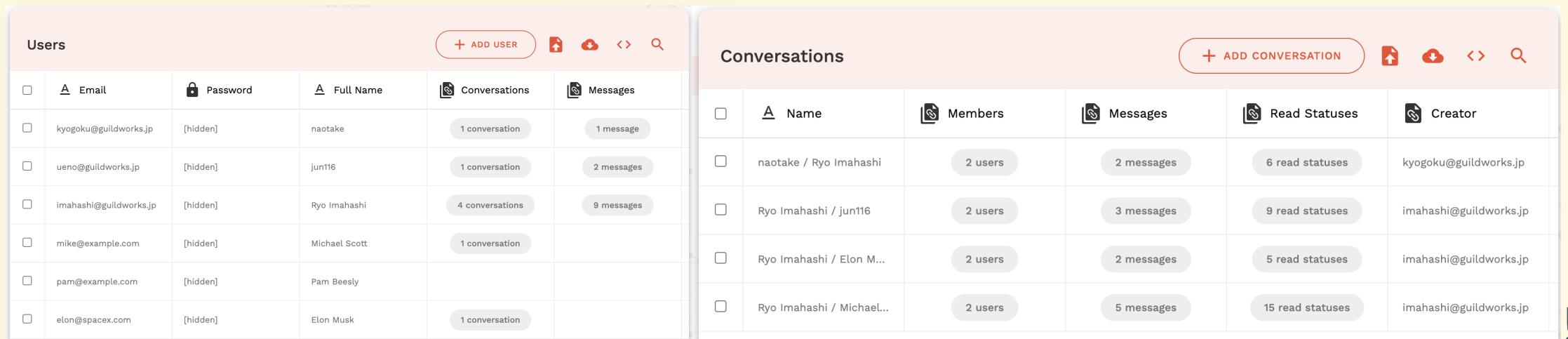
## Many-to-many Relationships

- This means that one Record in both Collections is tied to multiple Records in the other Collection.



# An example of a many-to-many Relationship

- In a Chat application, one user can have multiple conversations (to keep track of who and what messages were exchanged), and one conversation can have multiple members (users), so the relationship between the Users Collection and the Conversations Collection is many-to-many.



The screenshot shows two tables representing collections in a MongoDB interface:

Users					
	<input type="text"/> Email	<input type="password"/> Password	<input type="text"/> Full Name	<input type="button"/> Conversations	<input type="button"/> Messages
<input type="checkbox"/>	kyogoku@guildworks.jp	[hidden]	naotake	1 conversation	1 message
<input type="checkbox"/>	ueno@guildworks.jp	[hidden]	jun116	1 conversation	2 messages
<input type="checkbox"/>	imahashi@guildworks.jp	[hidden]	Ryo Imahashi	4 conversations	9 messages
<input type="checkbox"/>	mike@example.com	[hidden]	Michael Scott	1 conversation	
<input type="checkbox"/>	pam@example.com	[hidden]	Pam Beesly		
<input type="checkbox"/>	elon@spacex.com	[hidden]	Elon Musk	1 conversation	

Conversations					
	<input type="text"/> Name	<input type="button"/> Members	<input type="button"/> Messages	<input type="button"/> Read Statuses	<input type="button"/> Creator
<input type="checkbox"/>	naotake / Ryo Imahashi	2 users	2 messages	6 read statuses	kyogoku@guildworks.jp
<input type="checkbox"/>	Ryo Imahashi / jun116	2 users	3 messages	9 read statuses	imahashi@guildworks.jp
<input type="checkbox"/>	Ryo Imahashi / Elon M...	2 users	2 messages	5 read statuses	imahashi@guildworks.jp
<input type="checkbox"/>	Ryo Imahashi / Michael...	2 users	5 messages	15 read statuses	imahashi@guildworks.jp

!

# Database design

Let's design the database by looking at the UI of the sample app created in the previous lecture and thinking about what kind of data needs to be saved.

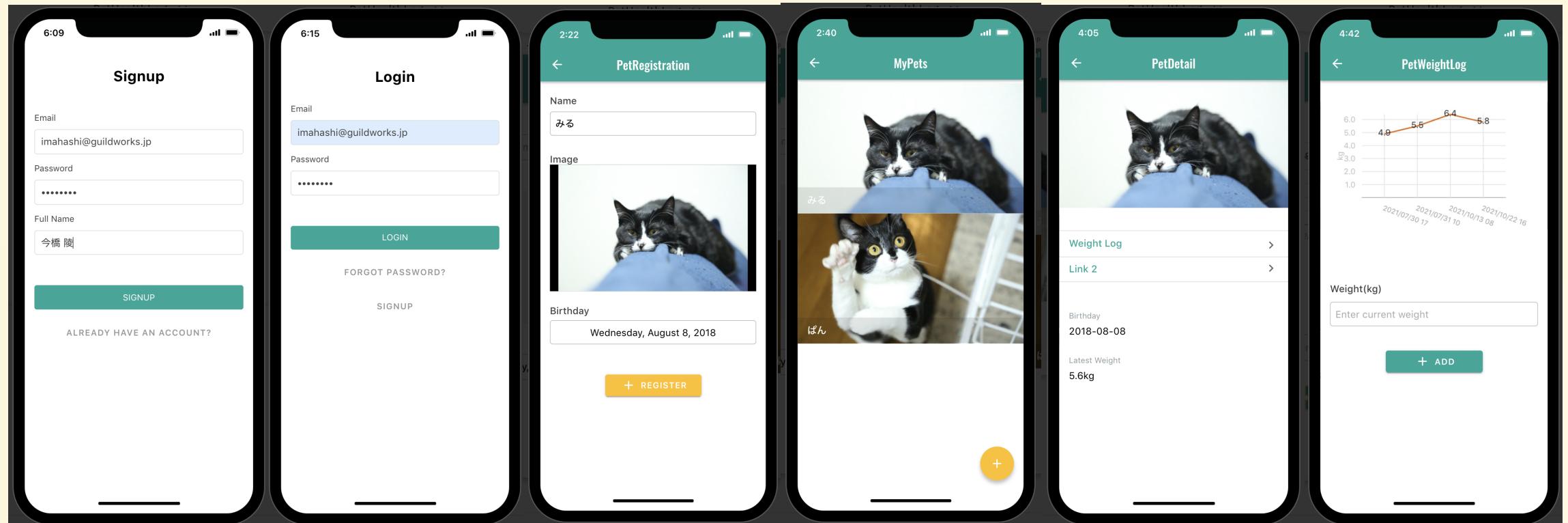
## URL for cloning the app created in the previous lecture

- Please clone the app from the following URL. We will use it to proceed with the lecture from here.

<https://previewer.adalo.com/014fd9d1-80c6-4325-899a-d943e778c865>

# Let's design the database.

Let's design the database by looking at the UI of the sample application. The steps are described in the next page.



## Database Design Steps

1. While looking at the UI, make a list of the data that will need to be saved. Write them down in a text editor (e.g. Notepad application).
2. Think about what kind of collections should be created to store the listed data, and create the collections in the Adalo database.
3. Add the data you listed in step 1 as a Property to the appropriate Collection and select the appropriate Type.
4. Set the Relationship Property to collections related to other collection.

In the next slide and onwards, there are explanations, but it is highly recommended that you try it by yourself before checking them.

There is no absolute right answer. When in doubt, follow your intuitions.

## Explanations

While looking at the UI, made a list of the data that needs to be saved, and it looked like this

- User's Email
- User's Password
- User's FullName
- Pet's Name
- Pet's Photo
- Pet's Birth Date
- Pet's weight
- Date and time the pet's weight was registered

- If anyone can name any other data, please let me know!

Thinking about what kind of collections should be created to store the listed data, listed these three Collections.

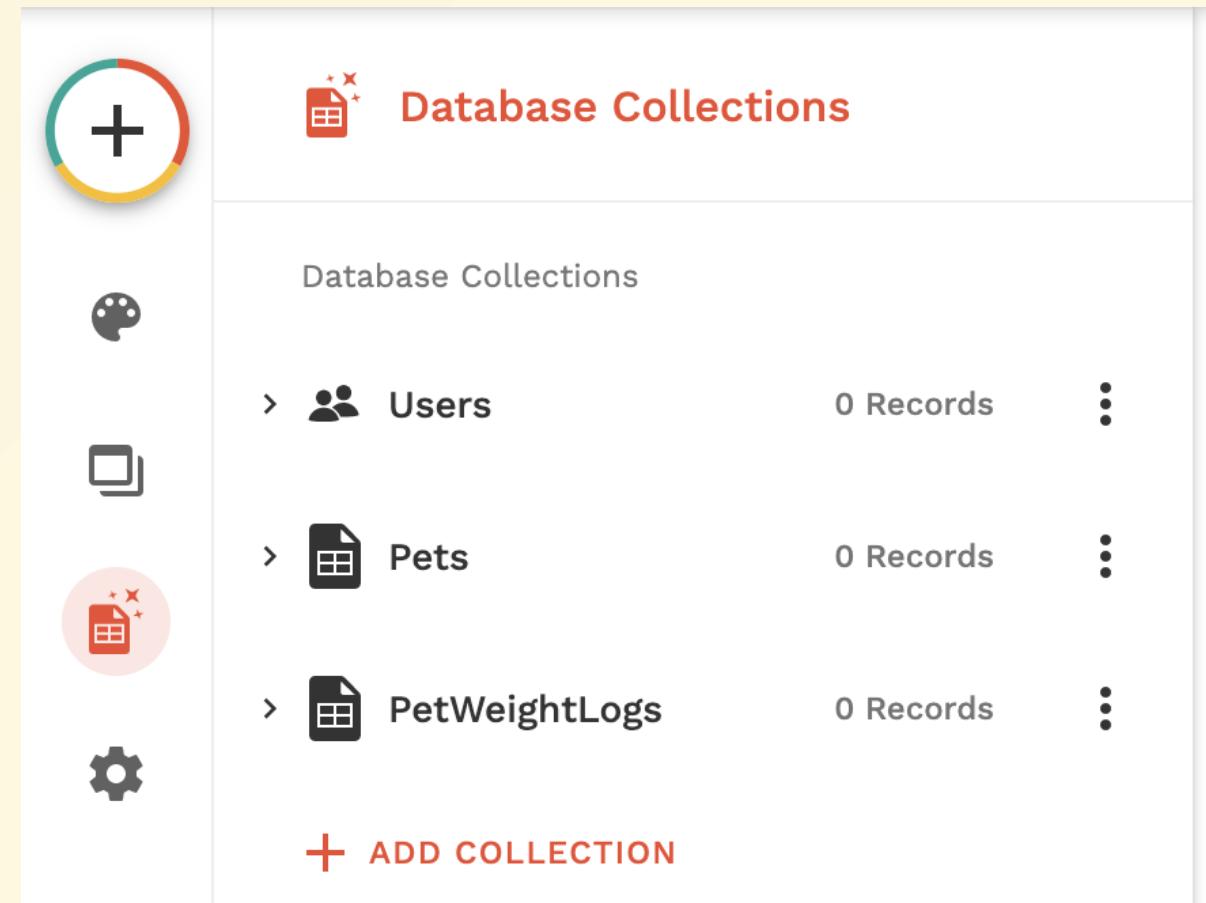
- Users
- Pets
- PetWeightLogs

- Many of you would have listed two collections, one for users and one for pets.
- Some people would not have listed a collection for pet weight records. (It is not wrong to include the pet's weight and its registration date in the pet's Collection. This will be explained later.)
- Have any of you listed other Collections?

## Additional information on Collection classification

- When the relationship "if A is identified, then B is identified" is true, "A" is often a Collection and "B" is a Property of that Collection.
  - If a user is identified, the user's email, password, and FullName are identified.
  - If a pet is identified, the pet's name, photo, and birthday are identified.
- When the relationship "there are multiple B's corresponding to A" is true, A and B are often split into two collections.
  - There are multiple pet weights and their registration times corresponding to a pet.

- Register the Collection into the Adalo database.
- Users is created by default.

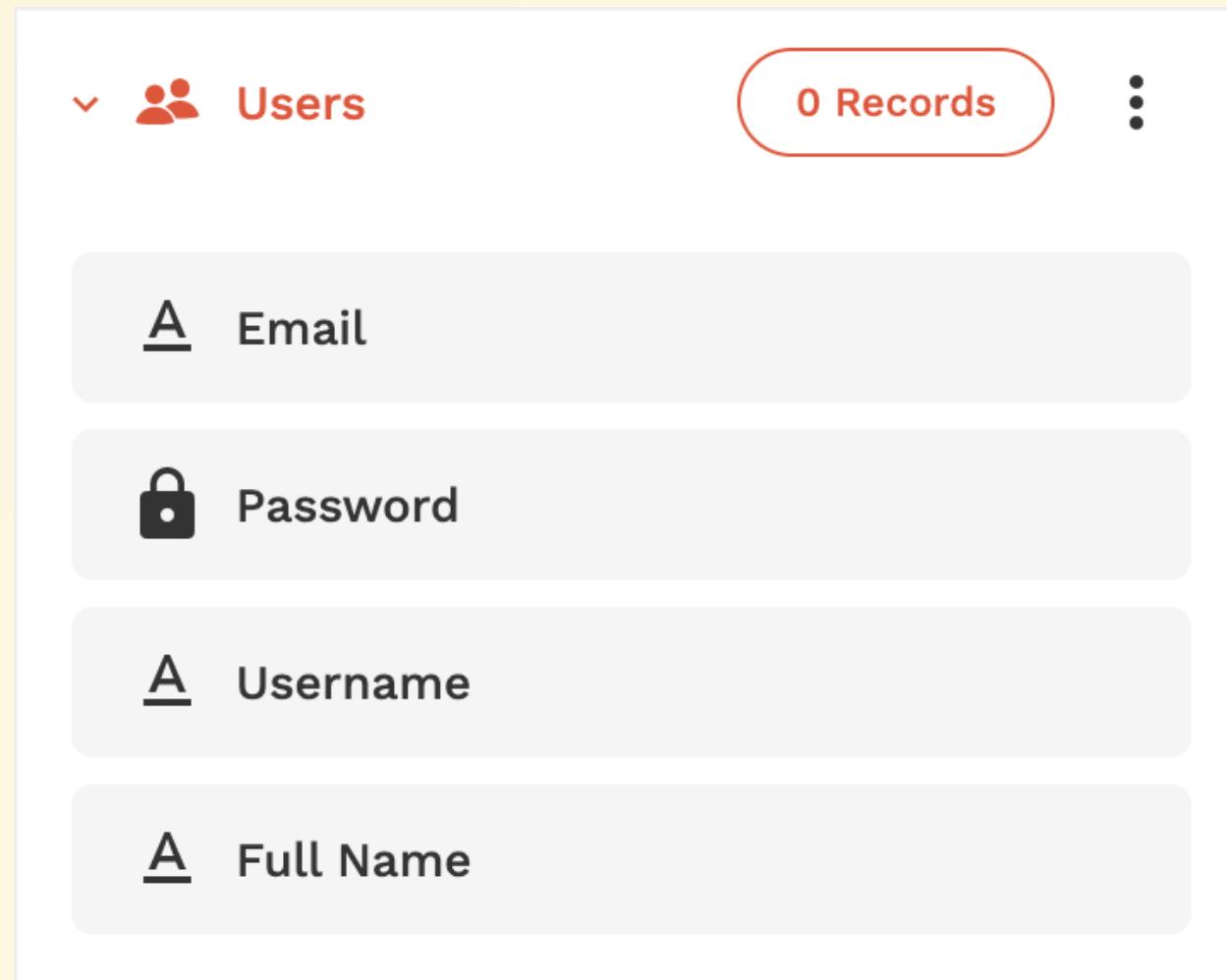


Next, I appended the data listed in 1 as a Property of the appropriate Collection. Types are placed in "()".

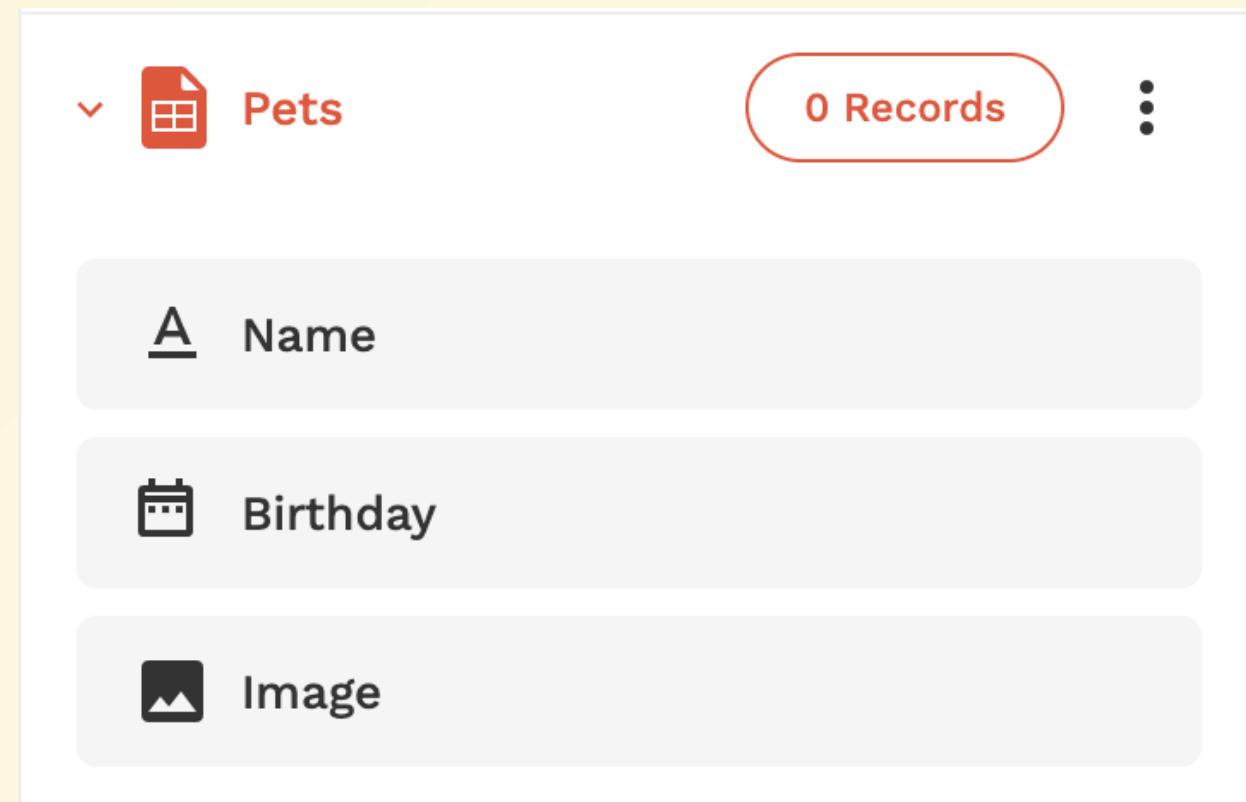
- Users
  - Email(Text)
  - Password(\*Password)
  - FullName(Text)
- Pets
  - Name(Text)
  - Image(Image)
  - Birthday(Date)
- PetWeightLogs
  - WeightKg(Number)
  - RegisteredTime(Date&Time)

Password is a special Type that is set to Password by default.

- In Adalo database, add the properties.
- The Users Collection is already set by default and contains all the necessary items.
- We don't need Username, but since we can't delete it, leave it as is.



- The Pets Collection Property looks like this.



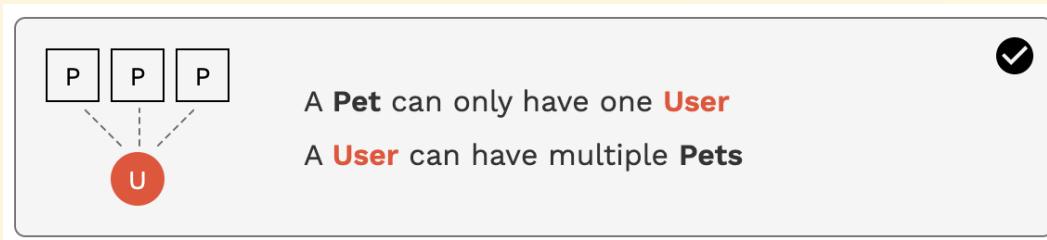
- The Property of the PetWeightLogs Collection looks like this.
- Delete the Name Property, which is set by default when you add a Collection.
  - You can delete it after dragging and dropping it so that the order is not at the top of the collection.

The screenshot shows a MongoDB interface with a collection named "PetWeightLogs". The collection has 0 records. It contains two properties: "WeightKg" and "WeightRegisteredTime".

- WeightKg:** Represented by a scale icon and the value 7.
- WeightRegisteredTime:** Represented by a clock icon.

Finally, for collections that are related to other collections, set the Relationship Property.

- Select the Users Collection to add a one-to-many Relationship with the Pet Collection.



▼ Users 0 Records ⋮

Email

Password

Username

Full Name

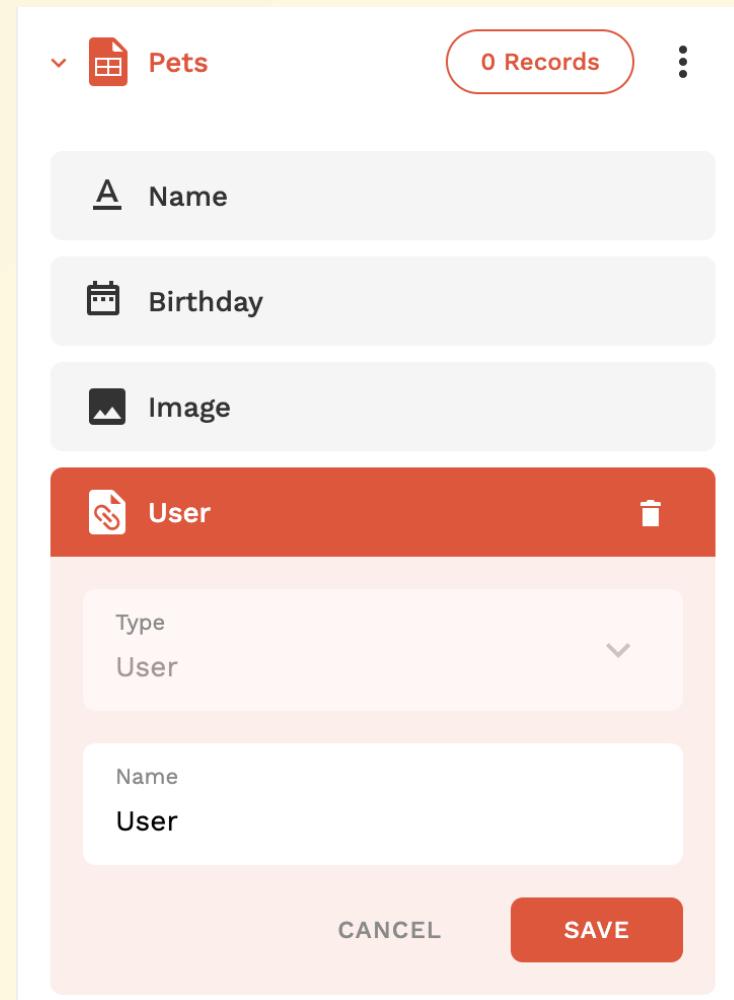
Pets trash icon

Type  
Pet

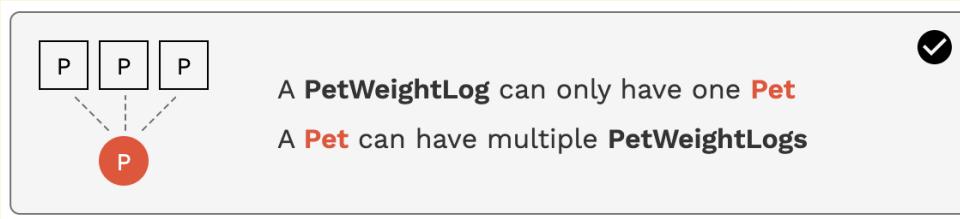
Name  
Pets

CANCEL SAVE

- If you check the Pets Collection, you will see that a Relationship with the Users Collection has been automatically added because the Relationship setting was made on the Users Collection side.
  - Since the Users Collection is one side, the "s" at the end is omitted and the Property name is "User".



- Add a Relationship to the Pets Collection with the PetWeightLogs Collection.
  - Select the Pets Collection to add a one-to-many Relationship with the PetWeightLogs Collection.



Pets 0 Records ⋮

A Name

Birthday

Image

User

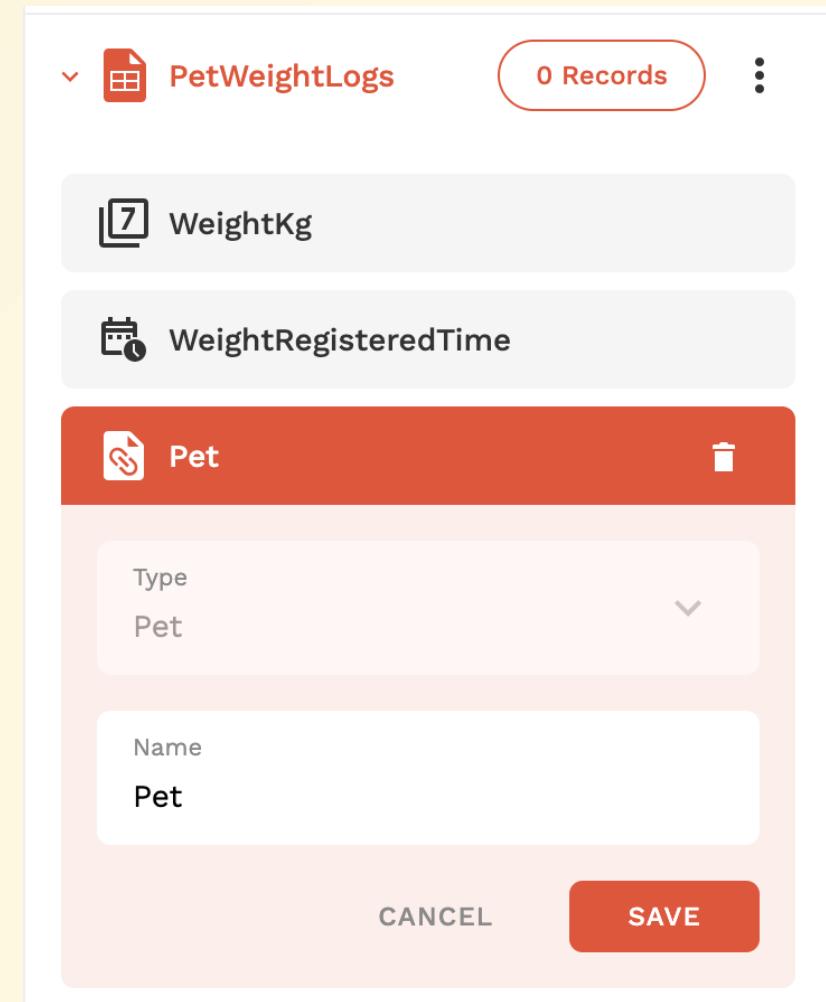
**PetWeightLogs** trash

Type  
PetWeightLog

Name  
PetWeightLogs

CANCEL SAVE

If you check the PetWeightLogs Collection, you will see that a Relationship with the Pets Collection has been automatically added because the Relationship setting was made on the Pets Collection side.



- Since the Pets Collection side is one side, the "s" at the end is omitted, and the Property name is "Pet".

# Reference: What happens if you include the pet weights and its registration times in the Pets Collection?

The record will be registered as follows, but in this case, you will have the troubles.

Pets						
	Name	Image	Birthday	Weight(kg)	WeightRegisteredTime	Created
<input type="checkbox"/>	みる		8/8/2018	5.2	November 3, 2021 12:00 AM	a few seconds ago
<input type="checkbox"/>	みる		8/8/2018	5.1	November 2, 2021 12:00 AM	a minute ago
<input type="checkbox"/>	みる		8/8/2018	5	November 1, 2021 12:00 AM	a minute ago

## The troubles

- Because multiple Records with different pet weights and their registration times are registered for one pet, the pet's information (Name, Image, Birthday) is registered in duplicate.
  - In order to change the information of one pet, we have to update all the duplicated records, which makes the process more complicated.
- Adalo has a convenient function to automatically generate a form to register a record in a single collection, but since the collection is not divided by the unit to register data, it cannot be used.

That's it for the sample app database design.

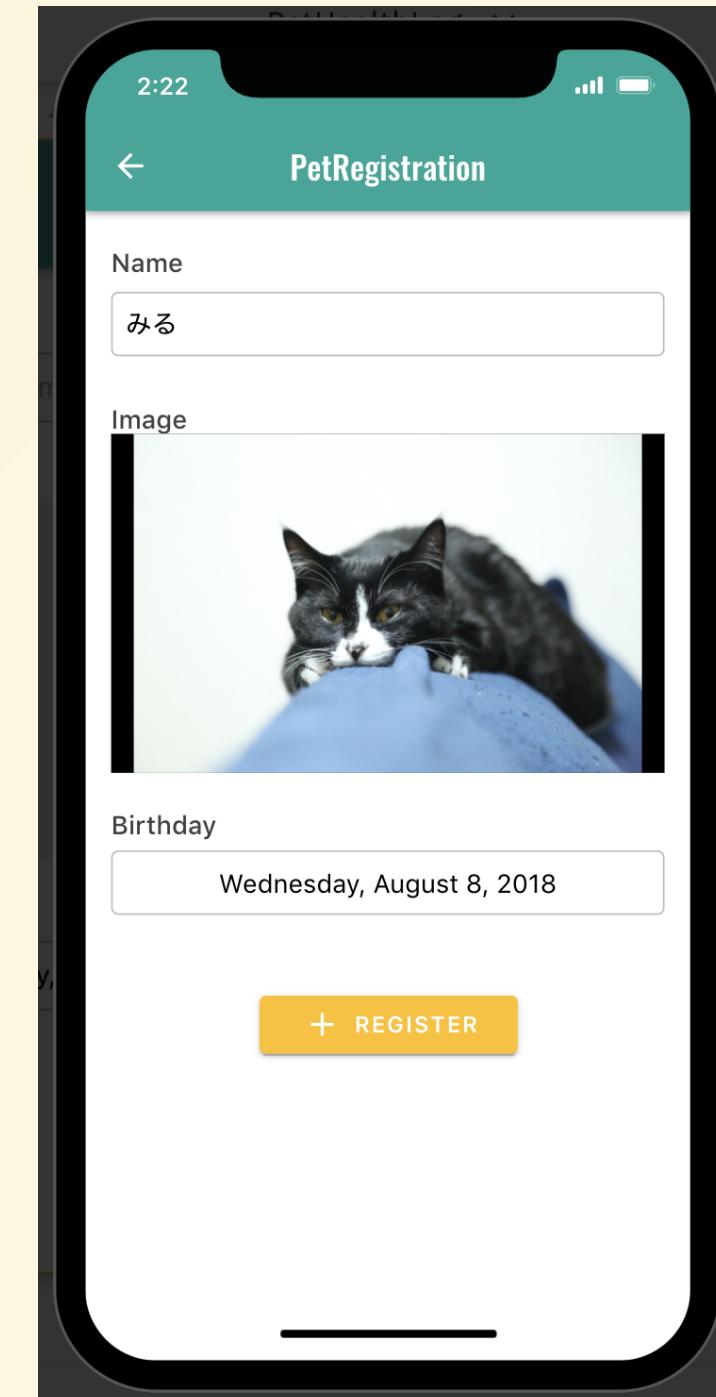
It is recommended that you set up Adalo's database in the same state as the document to avoid confusion in the later work.

# Data manipulation

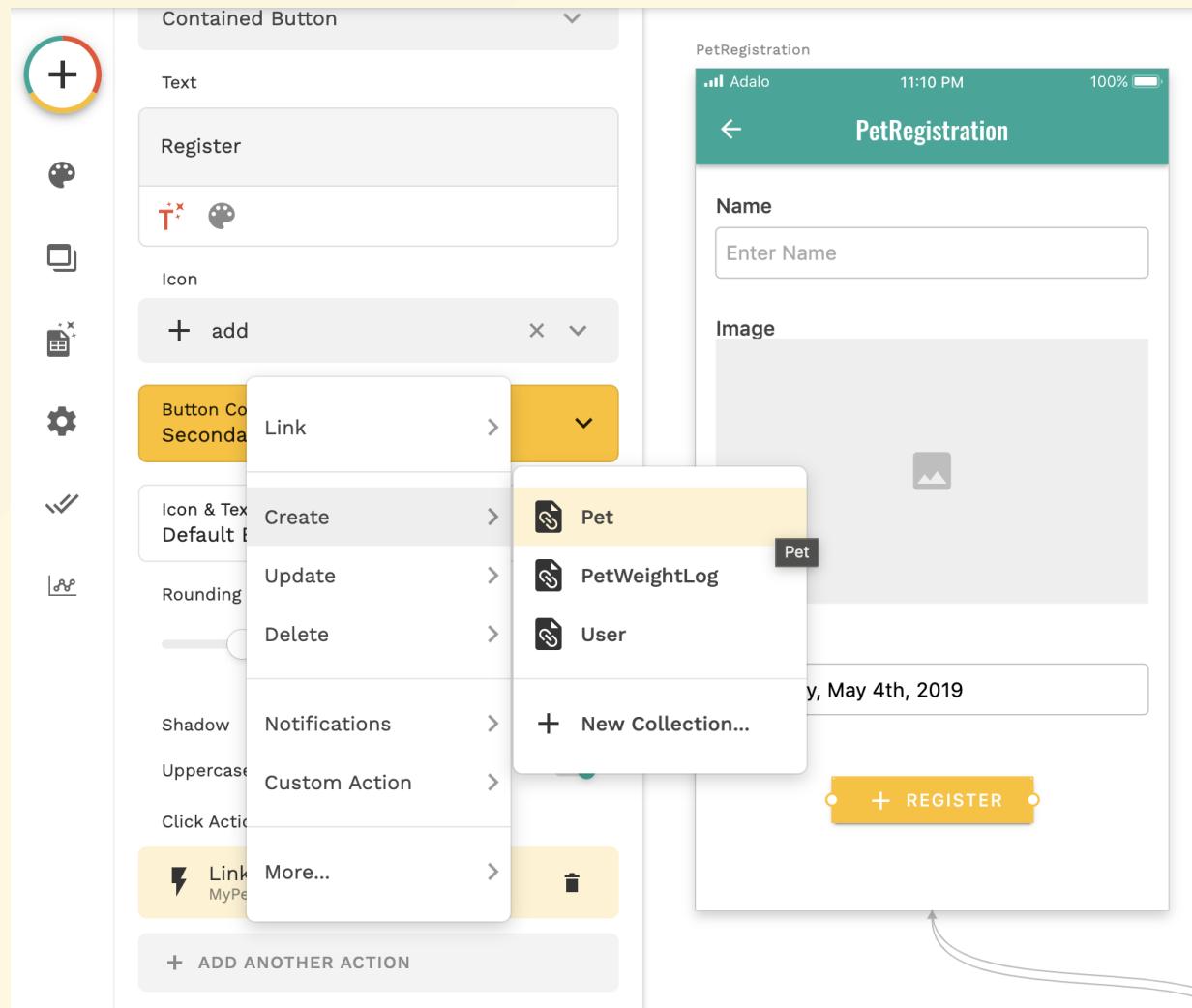
Let's use the database we designed so that we can perform CRUD(CREATE, READ, UPDATE, DELETE) operations in the sample application.

# Creating Data

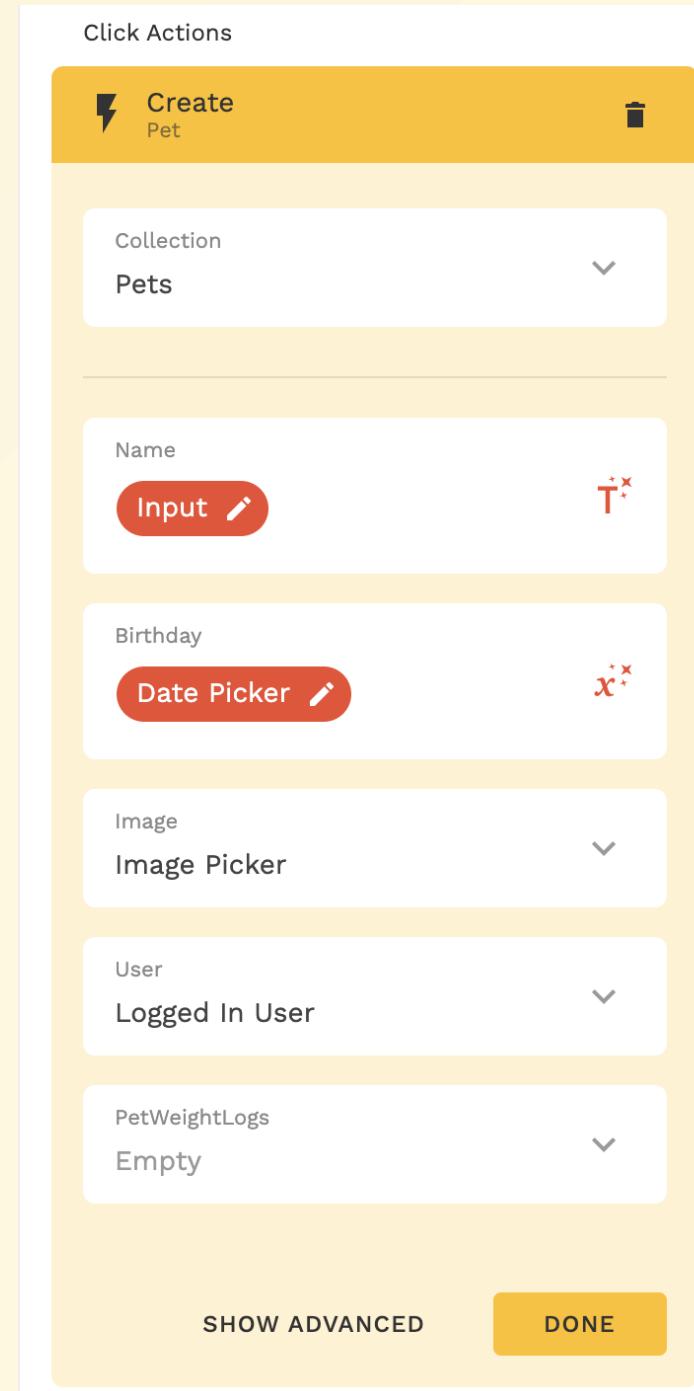
First, let's make it possible to actually register a record of a pet in the pet registration screen.



- Select the "REGISTER" button on the pet registration screen and click "ADD ANOTHER ACTION"
- Select "Create" > "Pet"



- For Name, select "Input" from Other Components.
- For Birthday, select "Date Picker" from Other Components.
- For Image, select "Image Picker" from Other Components.
- For User, select "Logged In User".
- Leave PetWeightLogs as Empty(not required here).



Let's try to register a pet with the Preview function.  
It is OK if you can register record in the Pets Collection

Pets

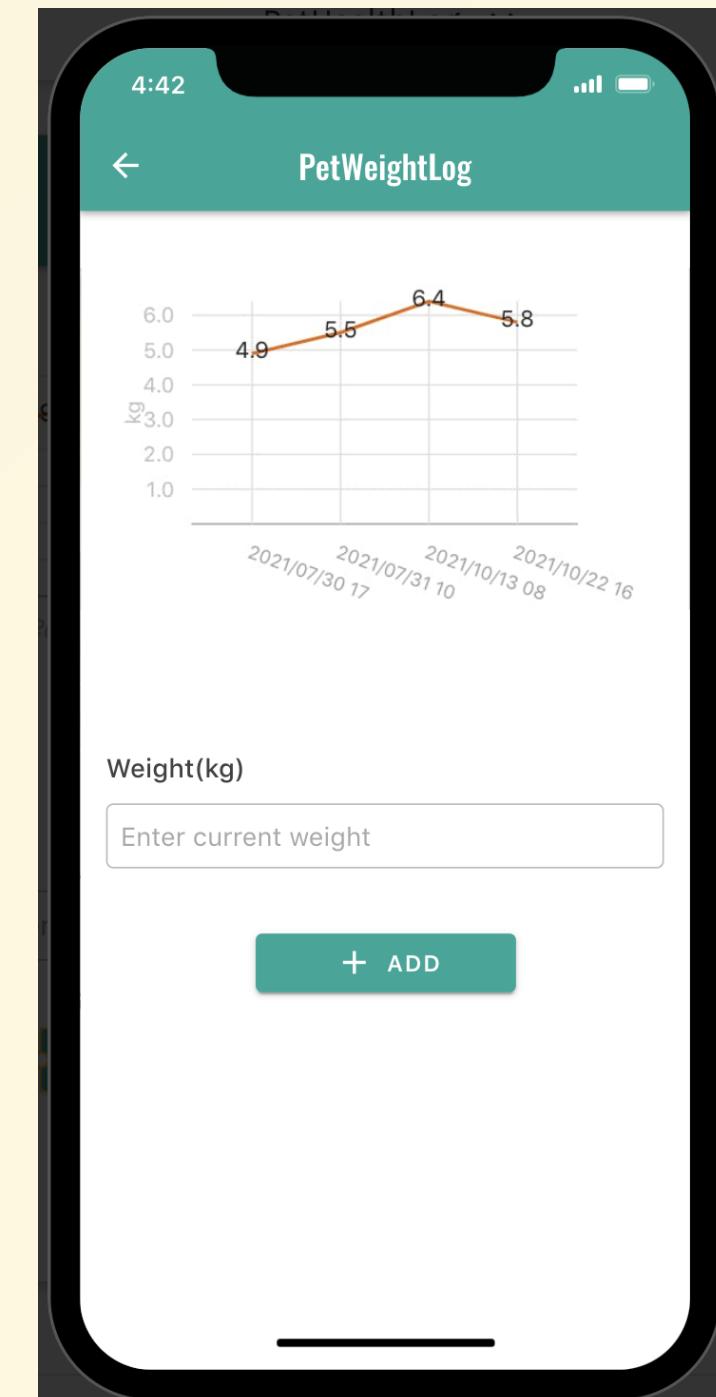
+ ADD PET

Upload Download Share Search

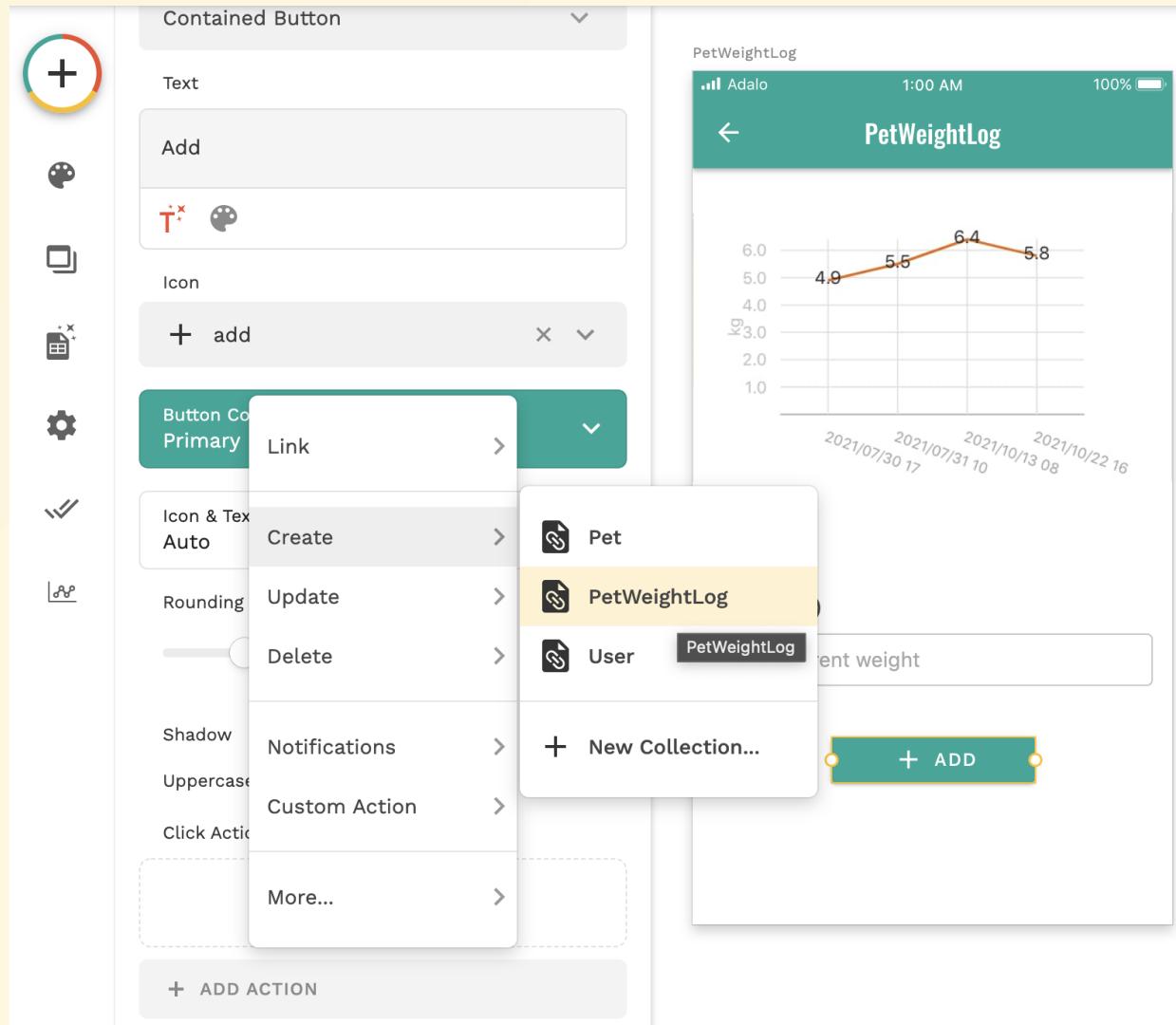
<input type="checkbox"/>	Name	Birthday	Image	User	PetWeightLogs	Created
<input type="checkbox"/>	みる(Database Record)	8/8/2018		imahashi@guildworks.jp		43 minutes ago

DONE

Next, let's make it possible to register the current weight in the pet weight management screen.

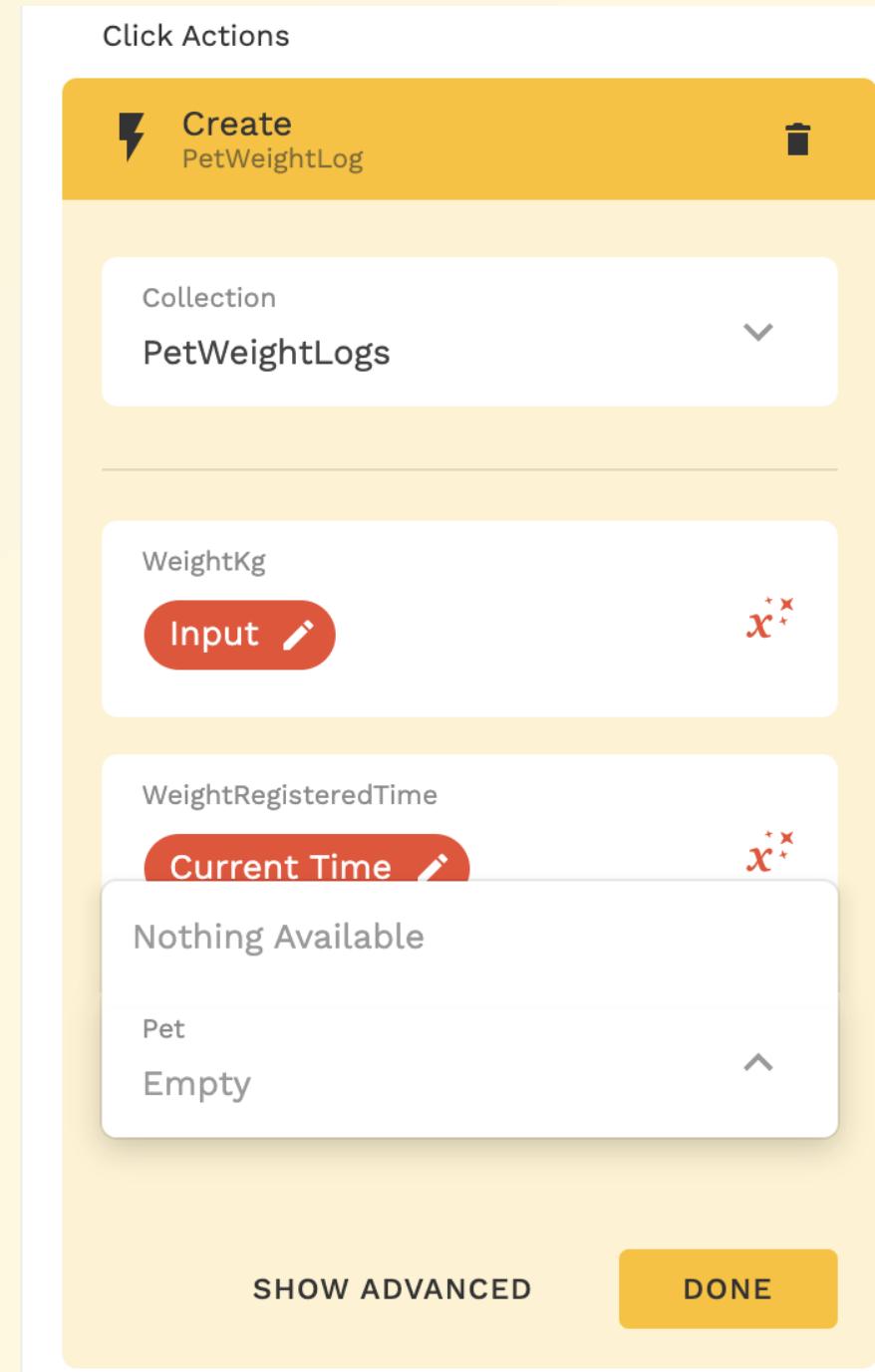


- Select the "ADD" button on the pet weight management screen and click "ADD ACTION".
- Select "Create" > "PetWeightLog"



Enter the following.

- For WeightKg, select "Input" from Other Components.
- For WeightRegisteredTime, select "Date&Time" > "Current Time".
- For Pet, Nothing is Available. Leave it as Empty for now.(We will configure this later)

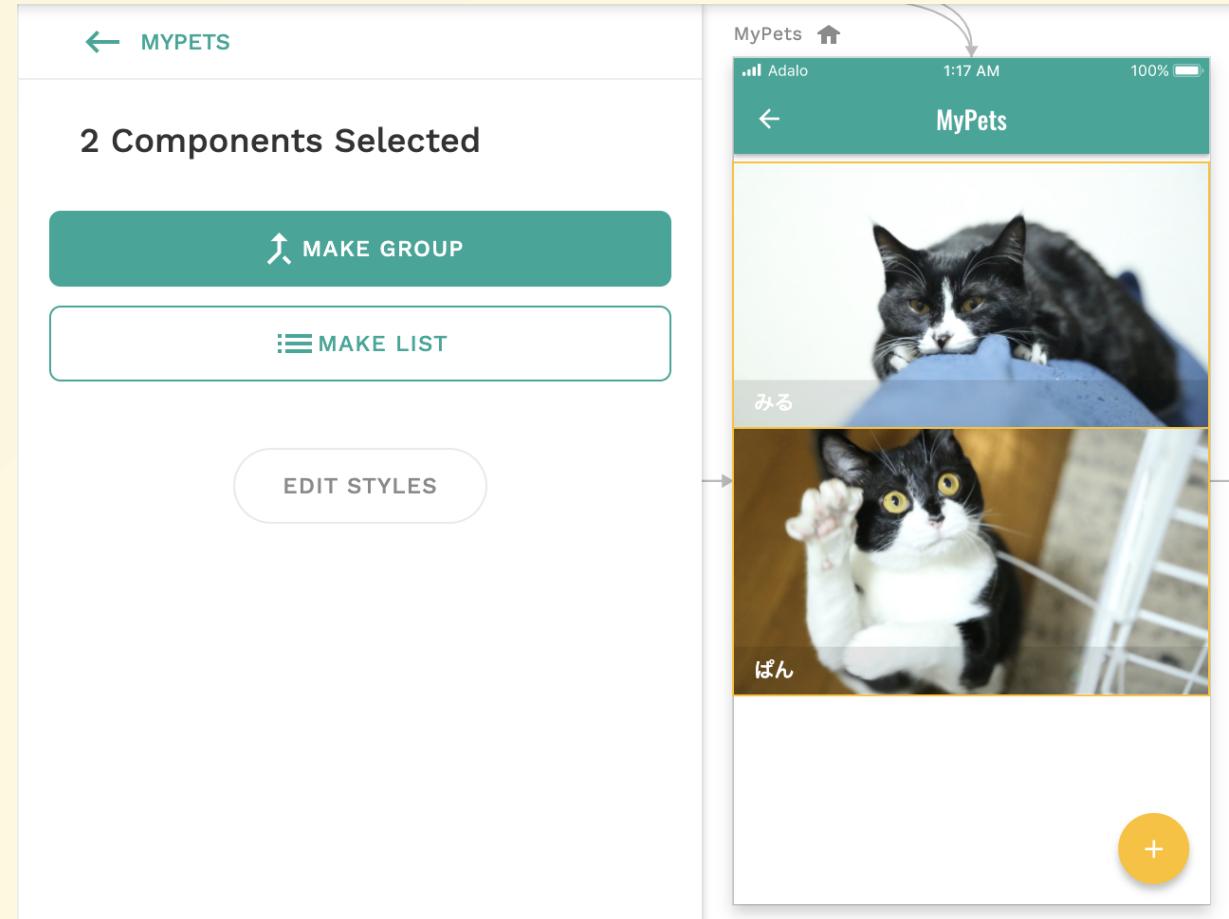


# Displaying Data (READ)

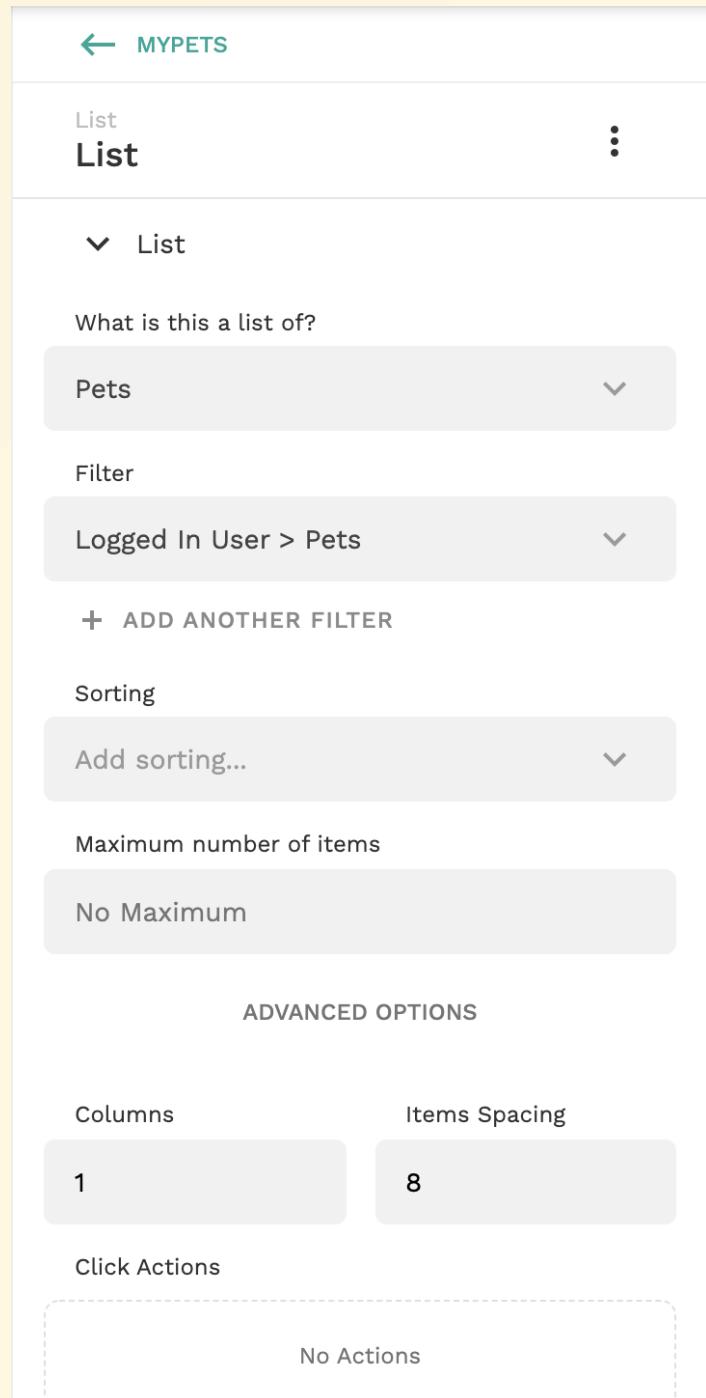
First, let's make registered pets displayed in pets list screen.



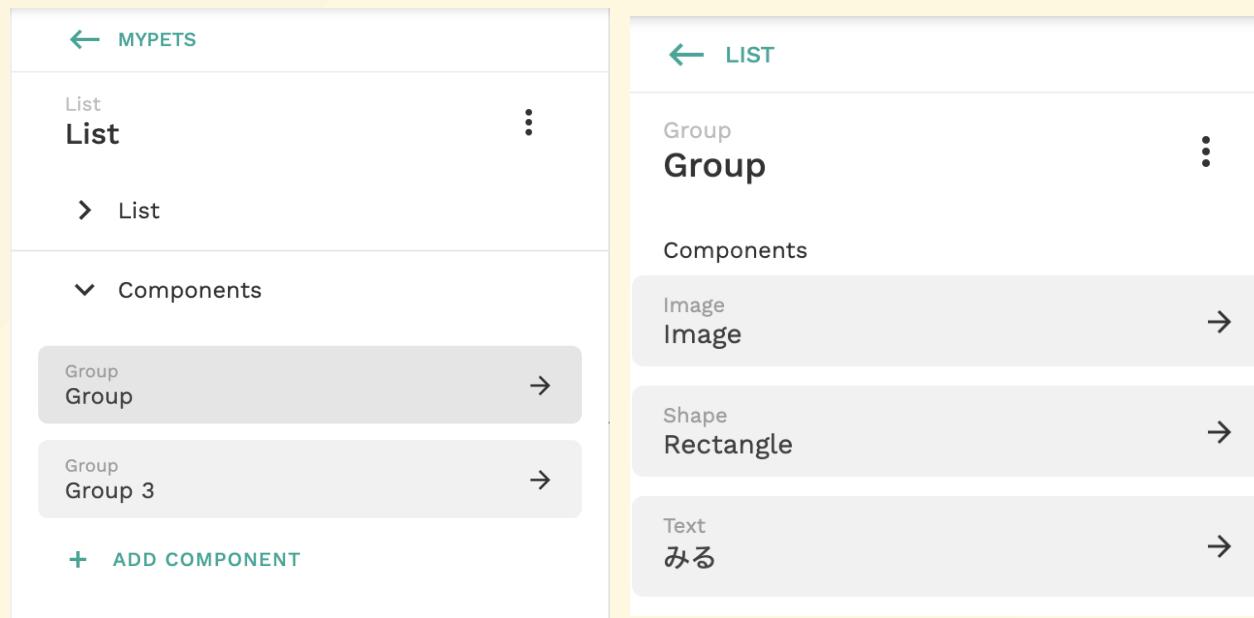
- Select the two components that display the pet's image and name, and click "MAKE LIST".



- Select "Pets" in What is this a list of?
- Select "Logged In User" > "Pets" in Filter

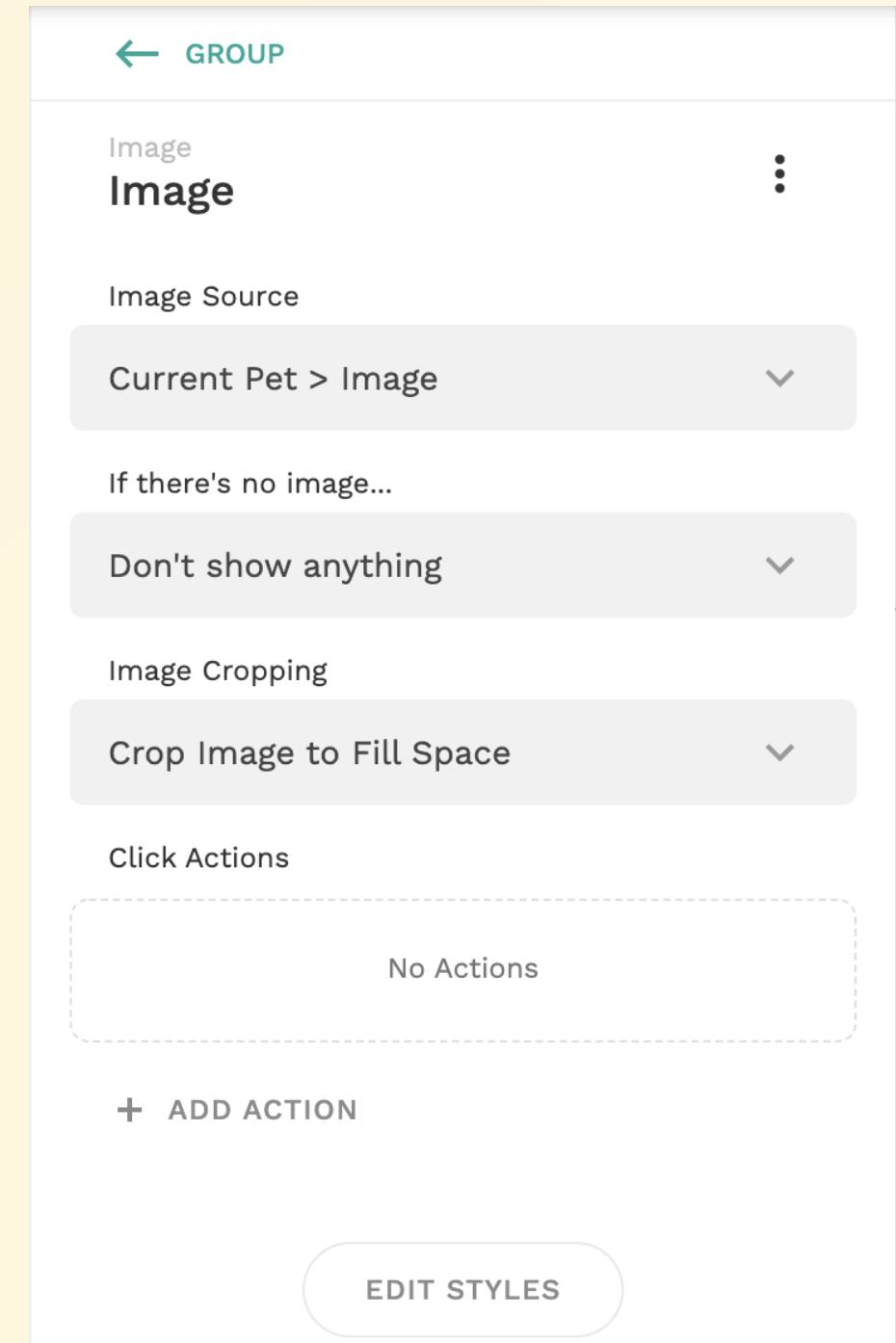


- Click the first Group which corresponds to first pet you manually created.
- Let's edit the Image component and the Pet Name component in this Group.



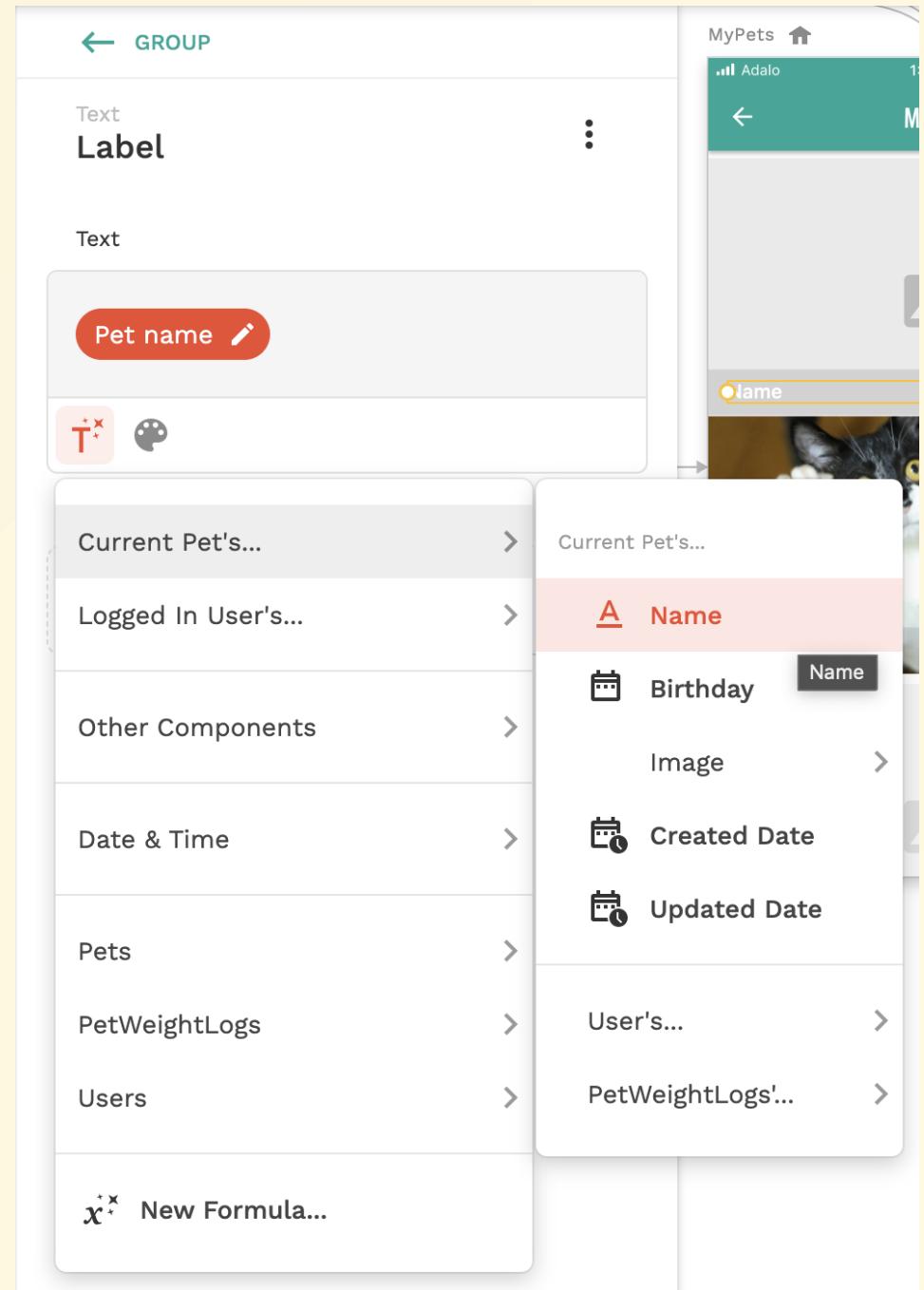
## Edit the Image component

- In Image Source, select "Database" > "Current Pet's" > "Image"
- In "If there's no image...", select "Don't show anything"
  - Alternatively, you can select "Show a place holder image" and set your [pet's silhouette image](#)

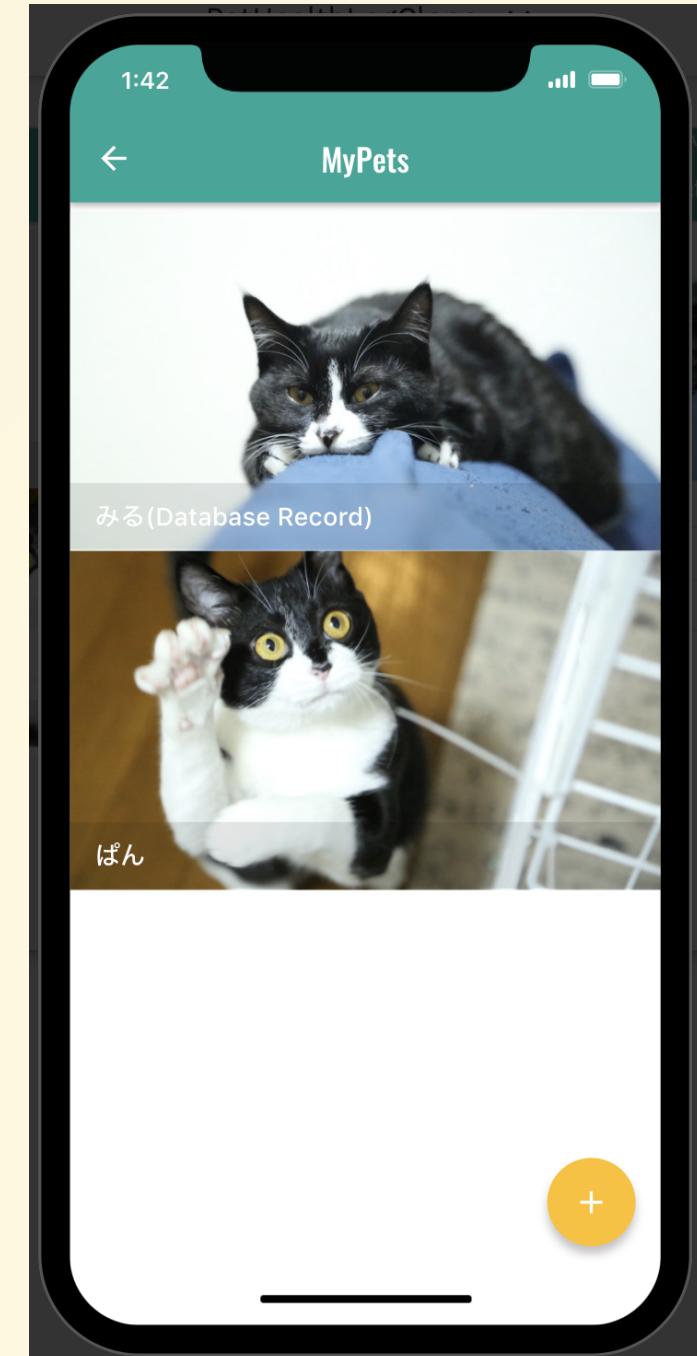


# Edit the pet name component

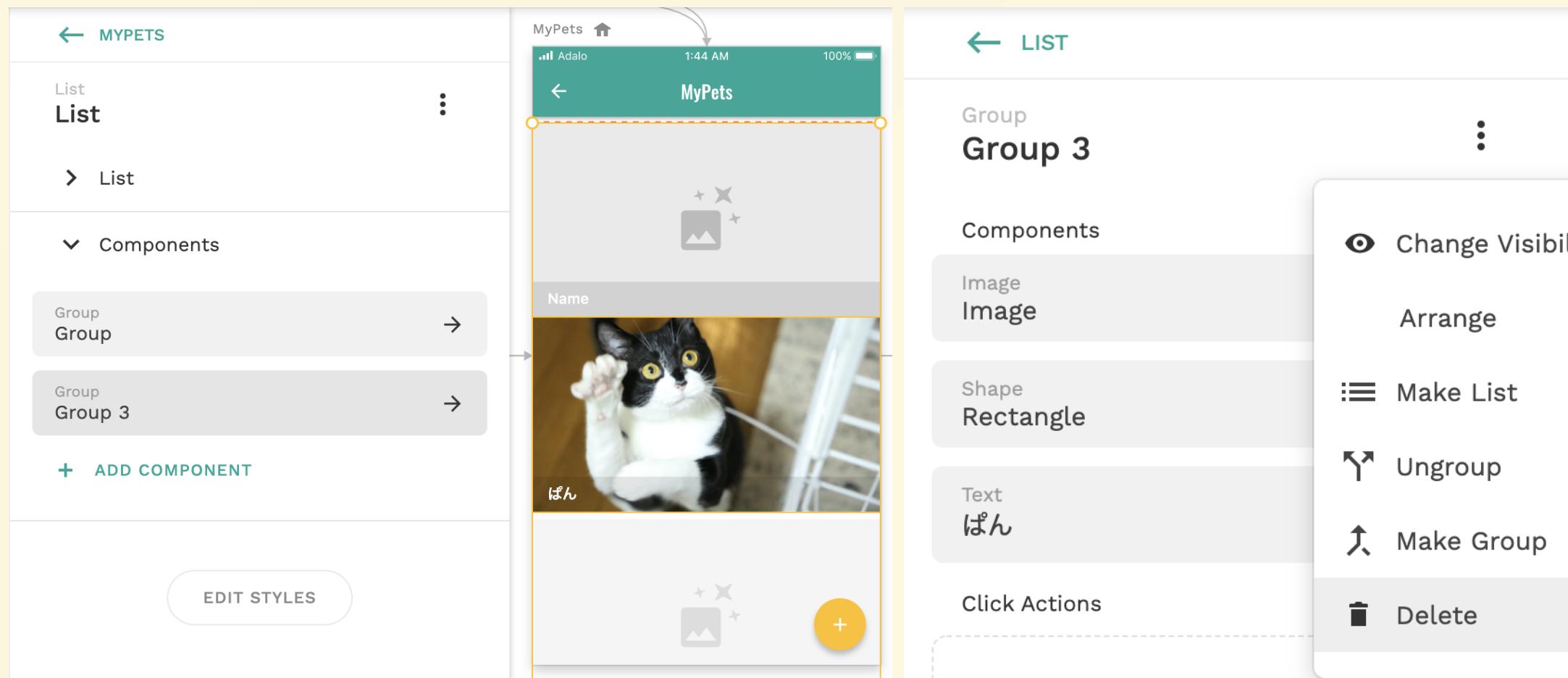
- Under Add Magic Text,  
select "Current Pet's" >  
"Name"



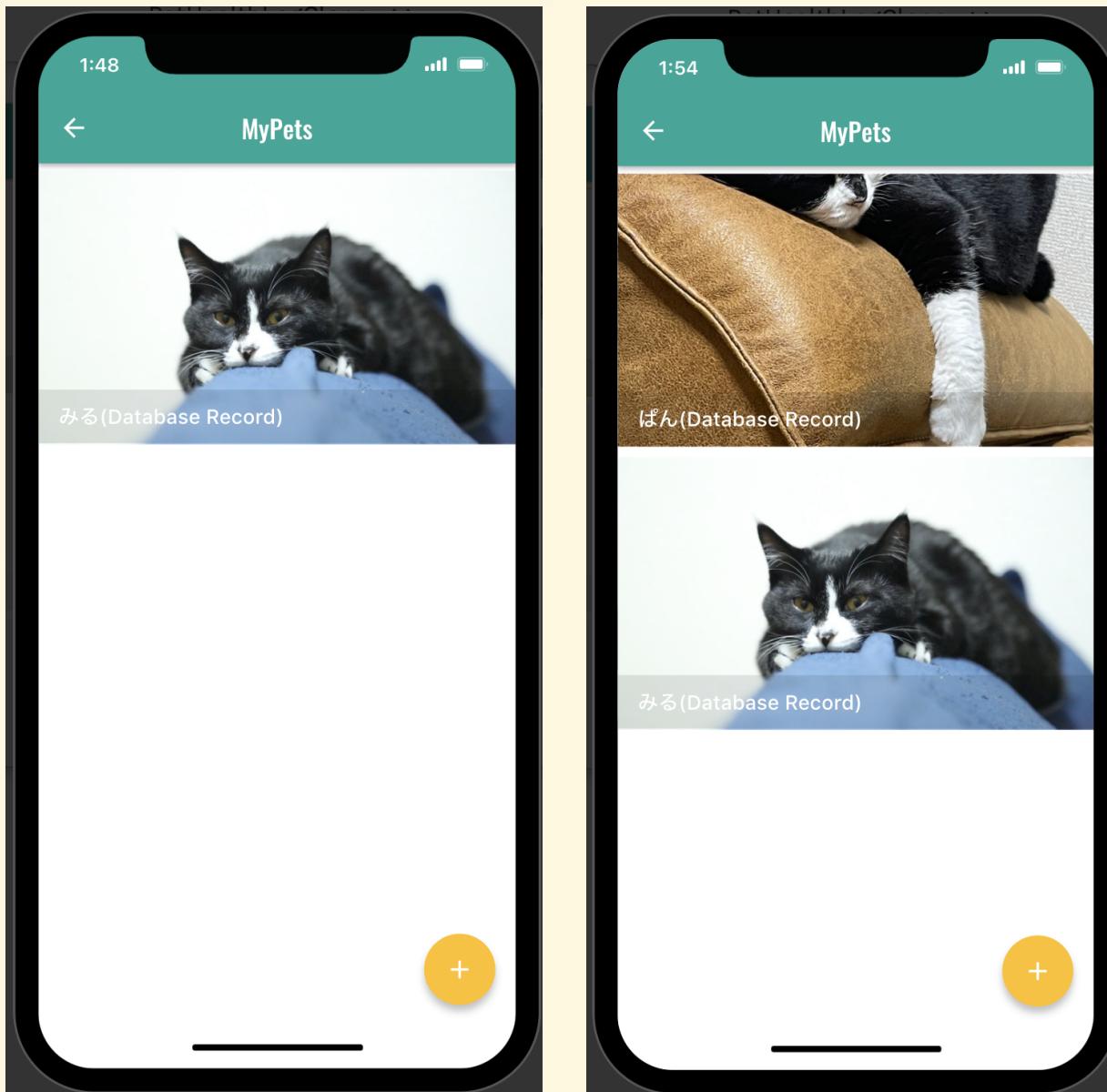
- When you check it by preview function, you will see the Record registered in the database as the first pet.



- Delete the second Group (which displays static second pet) in the component that makes up the List of pets, since it is not needed anymore.

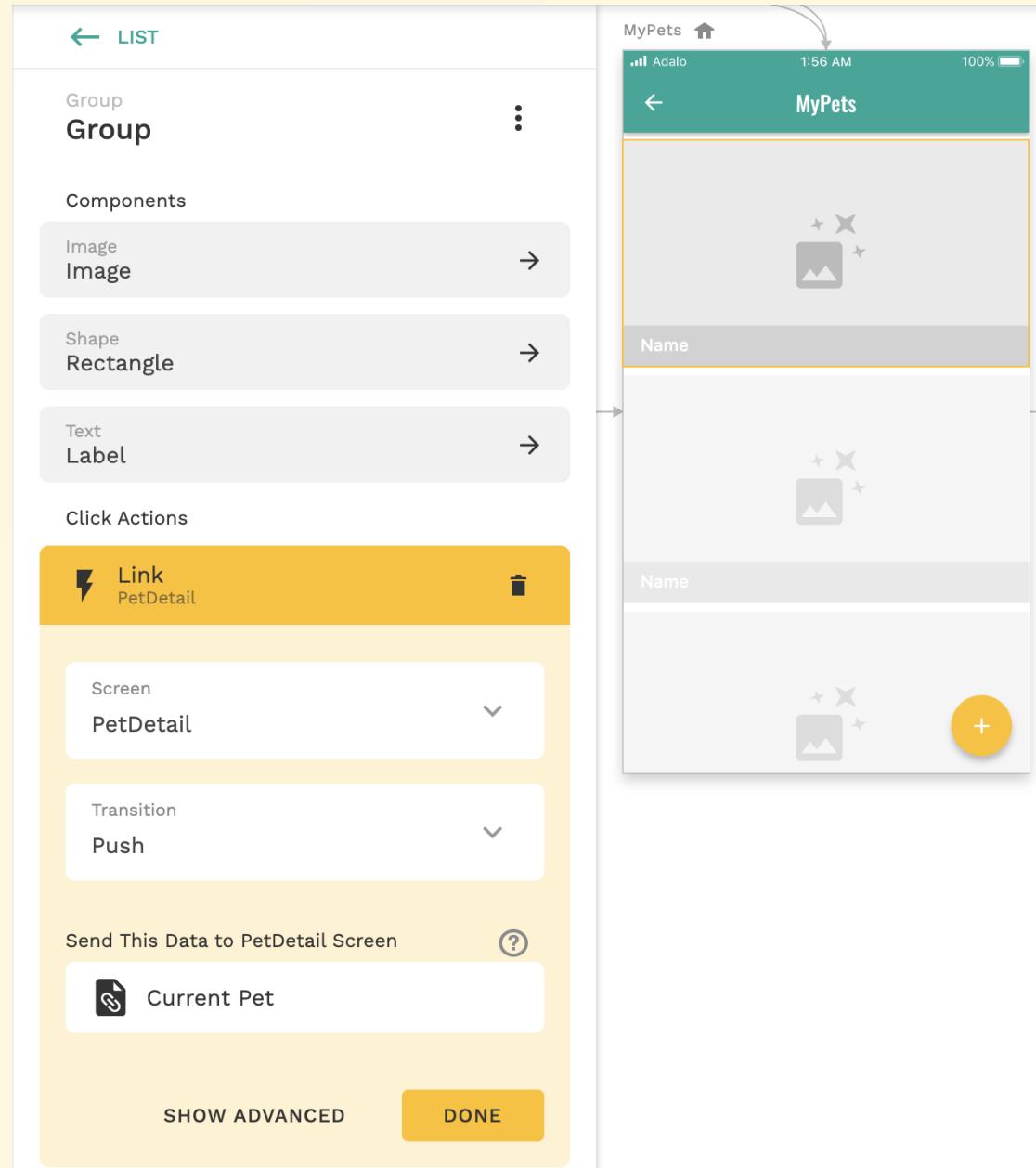


- When you check it by preview function, only the pets you have registered in the database are now displayed.
  - If you register additional pets, multiple pets will be displayed.

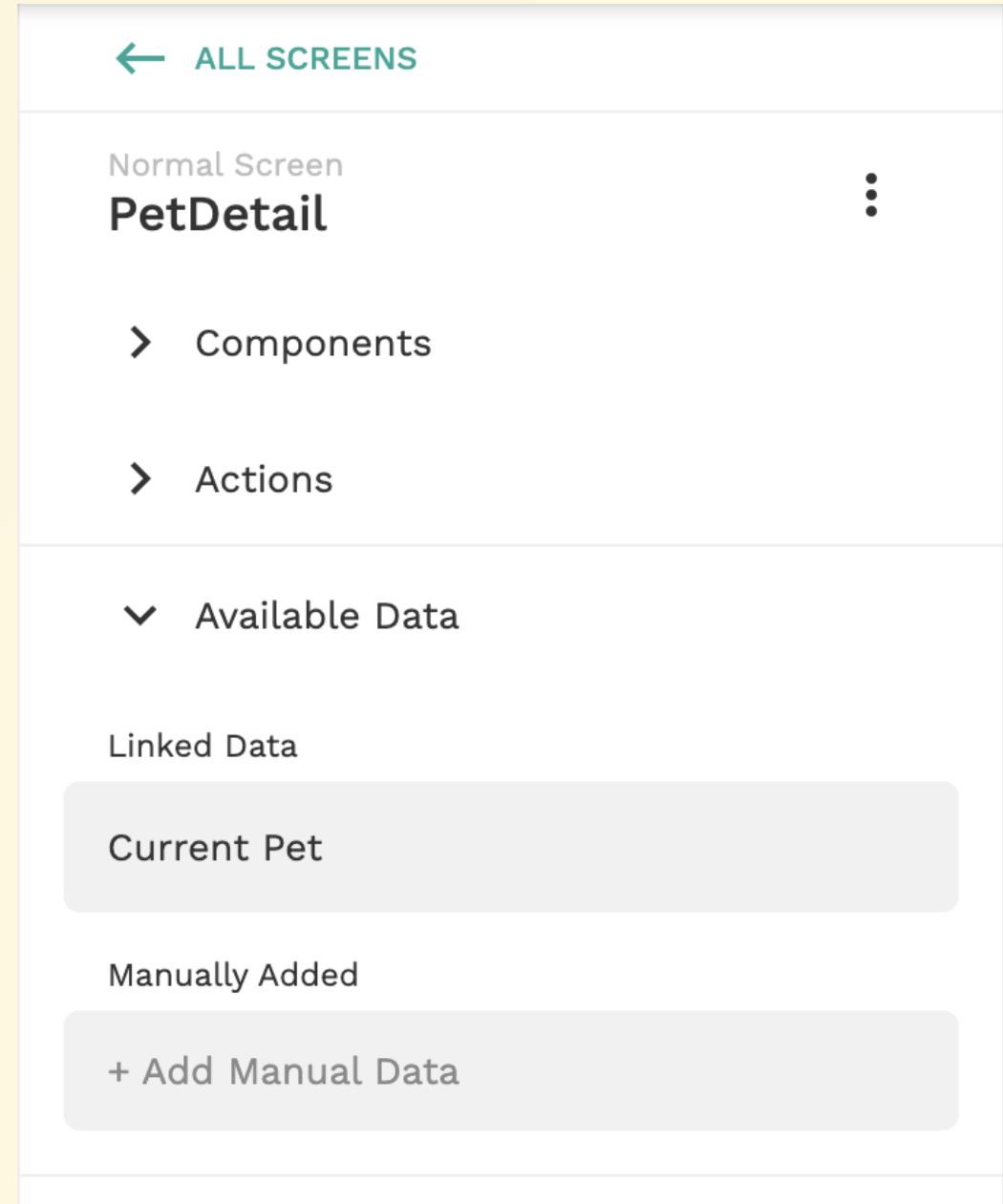


Make sure that when you click on a pet in the pet list, you can go to the detail screen of that pet.

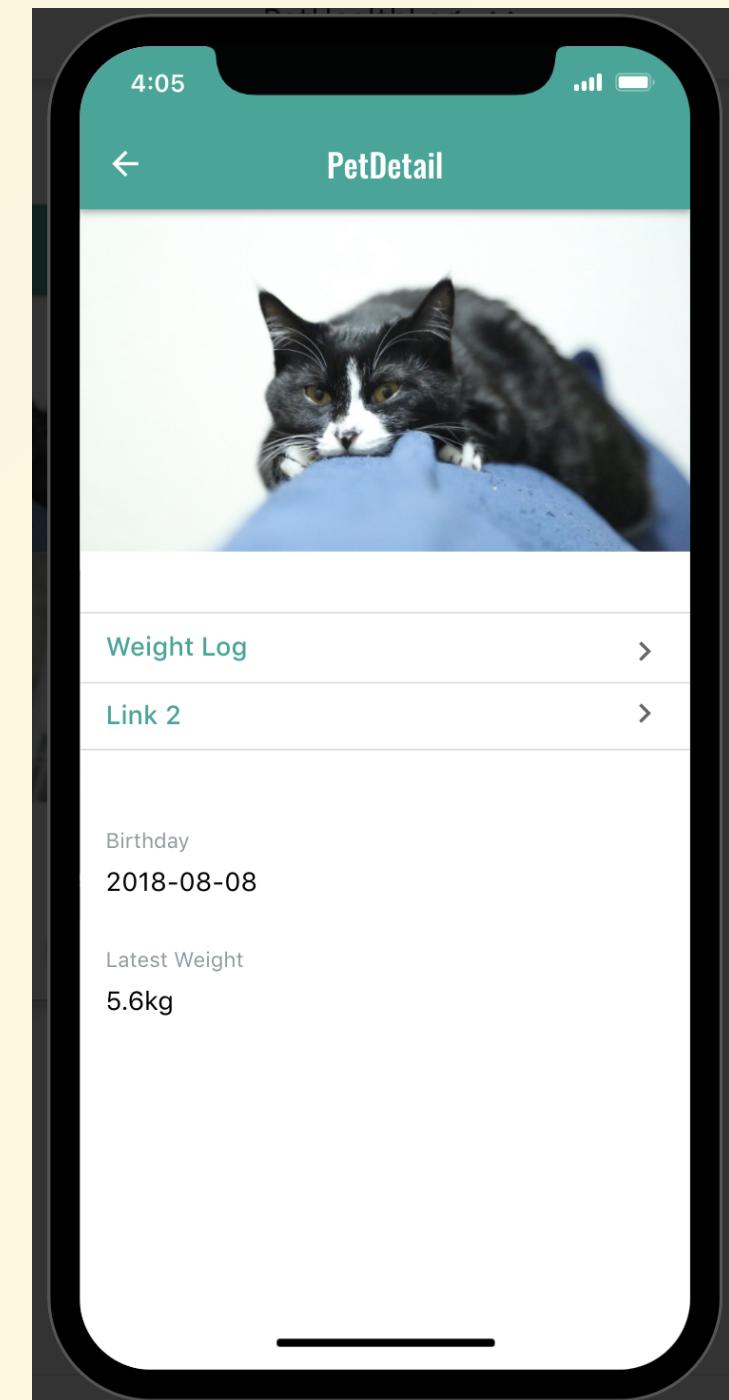
- "Current Pet" is automatically set in the "Send This Data to PetDetail Screen" of the Link set in the Group component of the pet.



- Since "Current Pet" was set in "Send This Data to PetDetail Screen" of Link from Pet List Screen, "Current Pet" is set as "Linked Data" in "Available Data" of Pet Detail Screen.
  - As a result, the pet selected in the Pet List screen (Current Pet) can be handled in the Pet Detail screen.

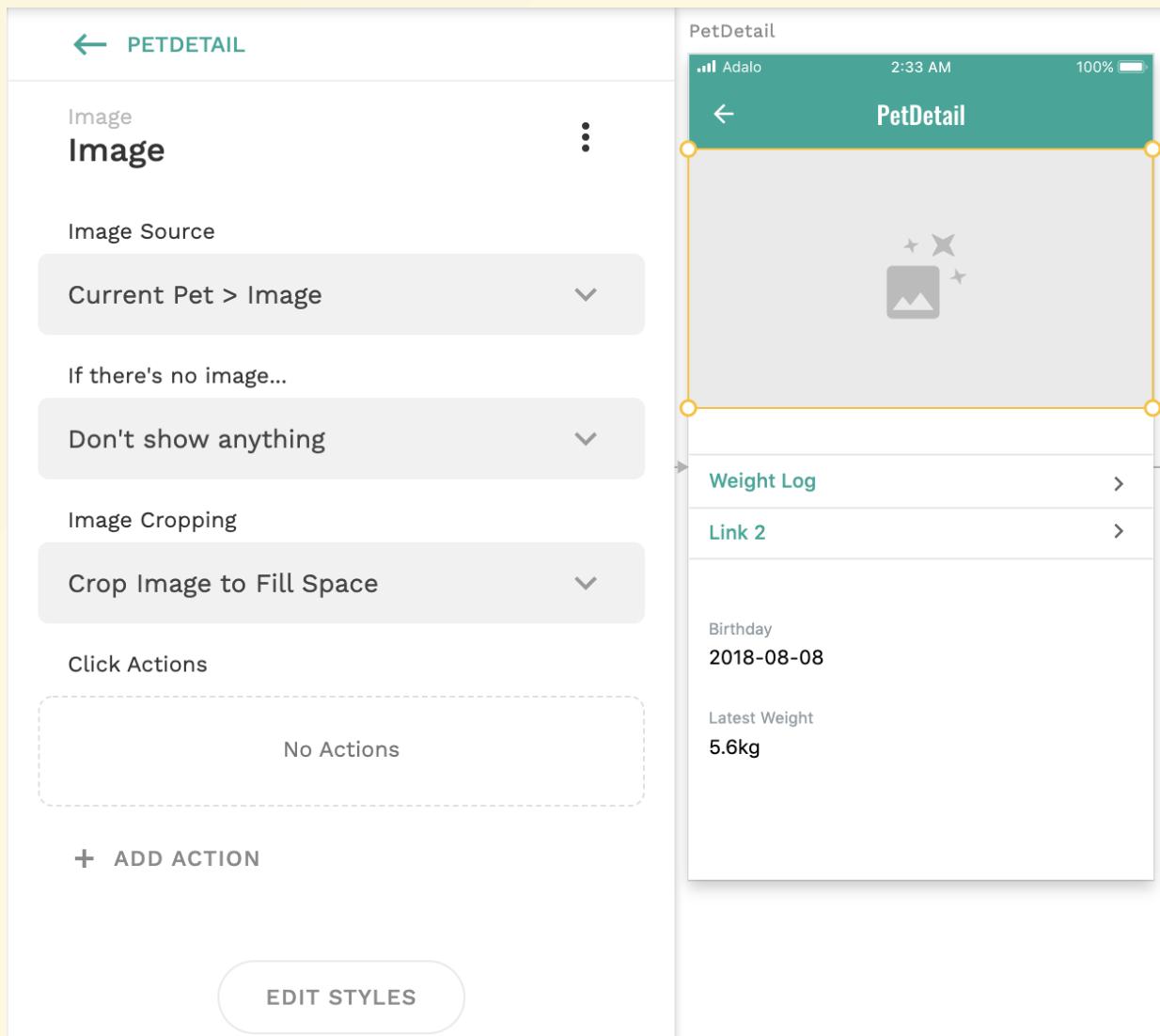


Next, we will make sure that the pet selected in the Pet List screen is displayed in the Pet Detail screen.



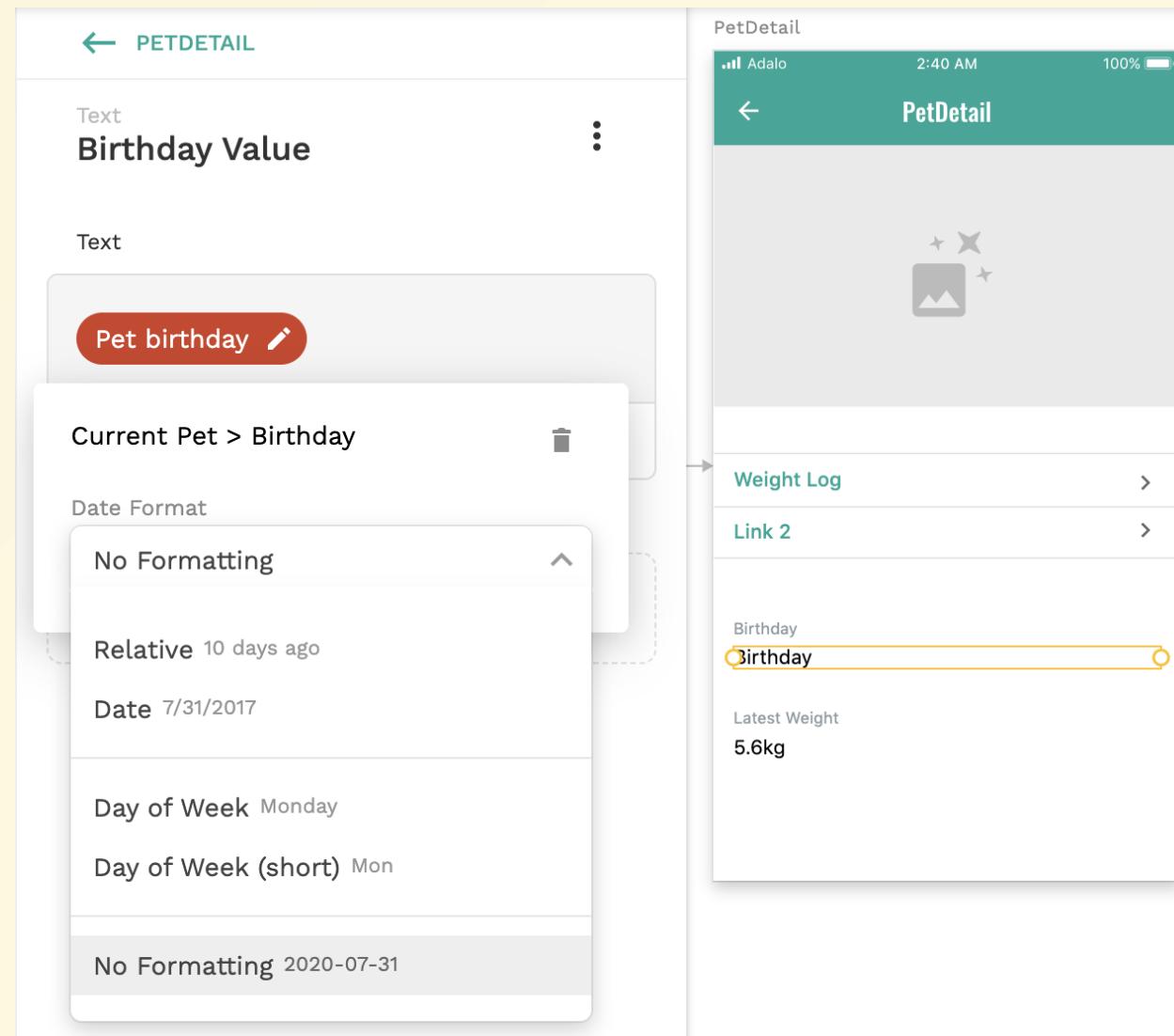
Click on the Image component and configure it as follows.

- In Image Source, select "Database" > "Current Pet's" > "Image".
- In "If there's no image...", select "Don't show anything"
  - Alternatively, you can select "Show a place holder image" and set your [image](#)



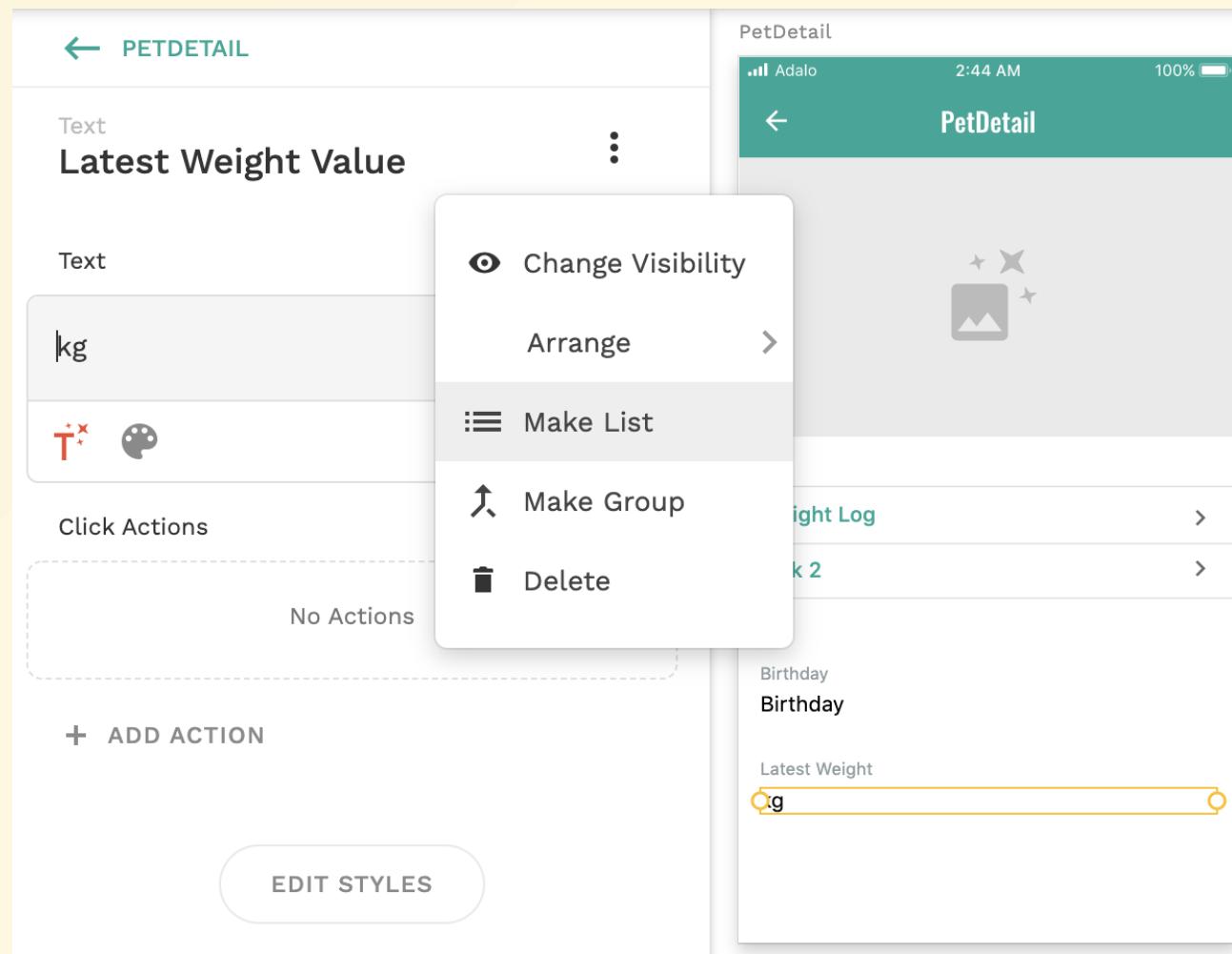
Click the Birthday Value component and configure it as follows.

- Select "Current Pet's" > "Birthday" in Text
- Select "No Formatting" in Date Format



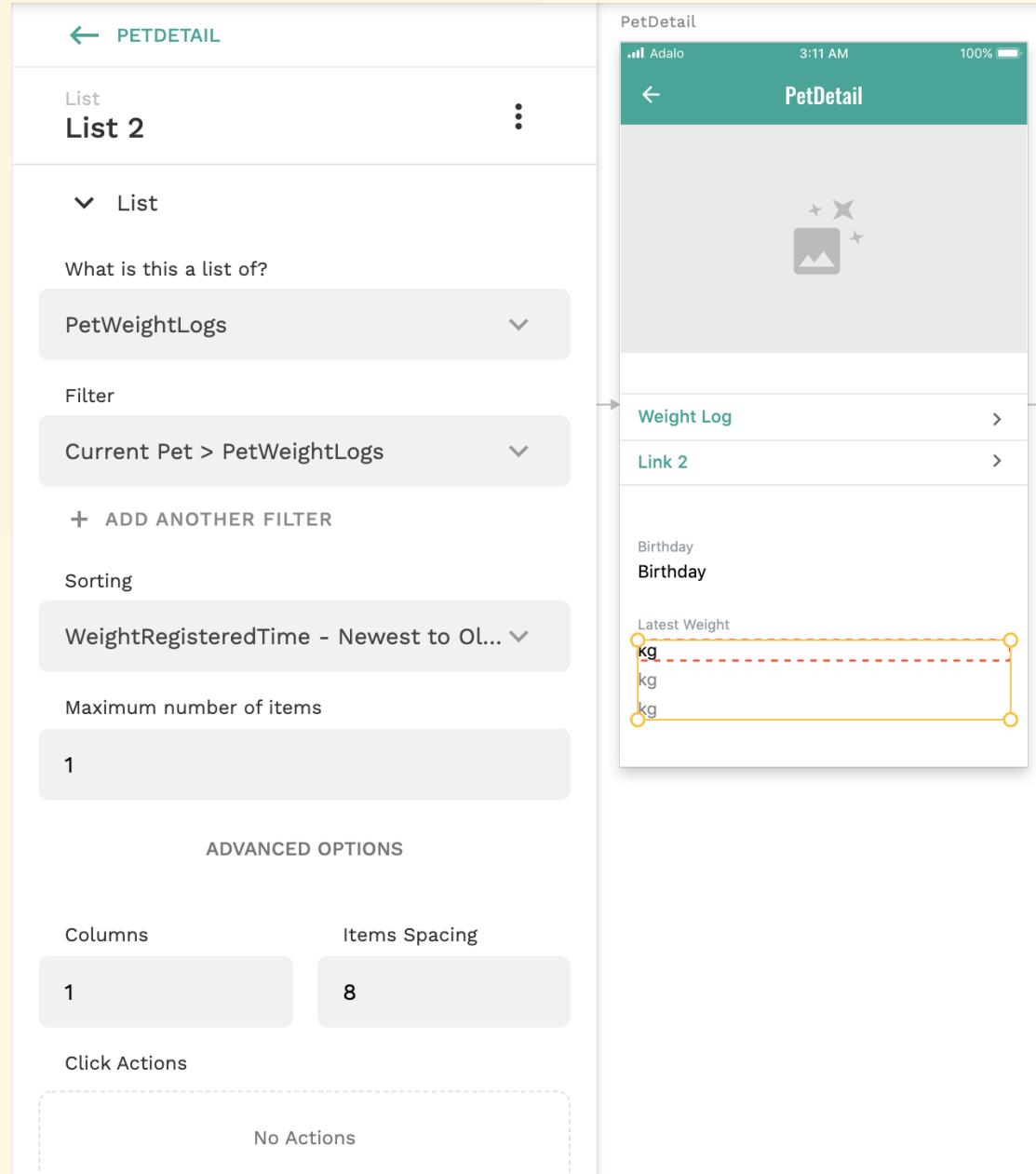
- Click the Latest Weight Value component and "MAKE LIST".

This is required to displaying the latest weight.  
(The settings on the next page will narrow down the list to the most recent one.)

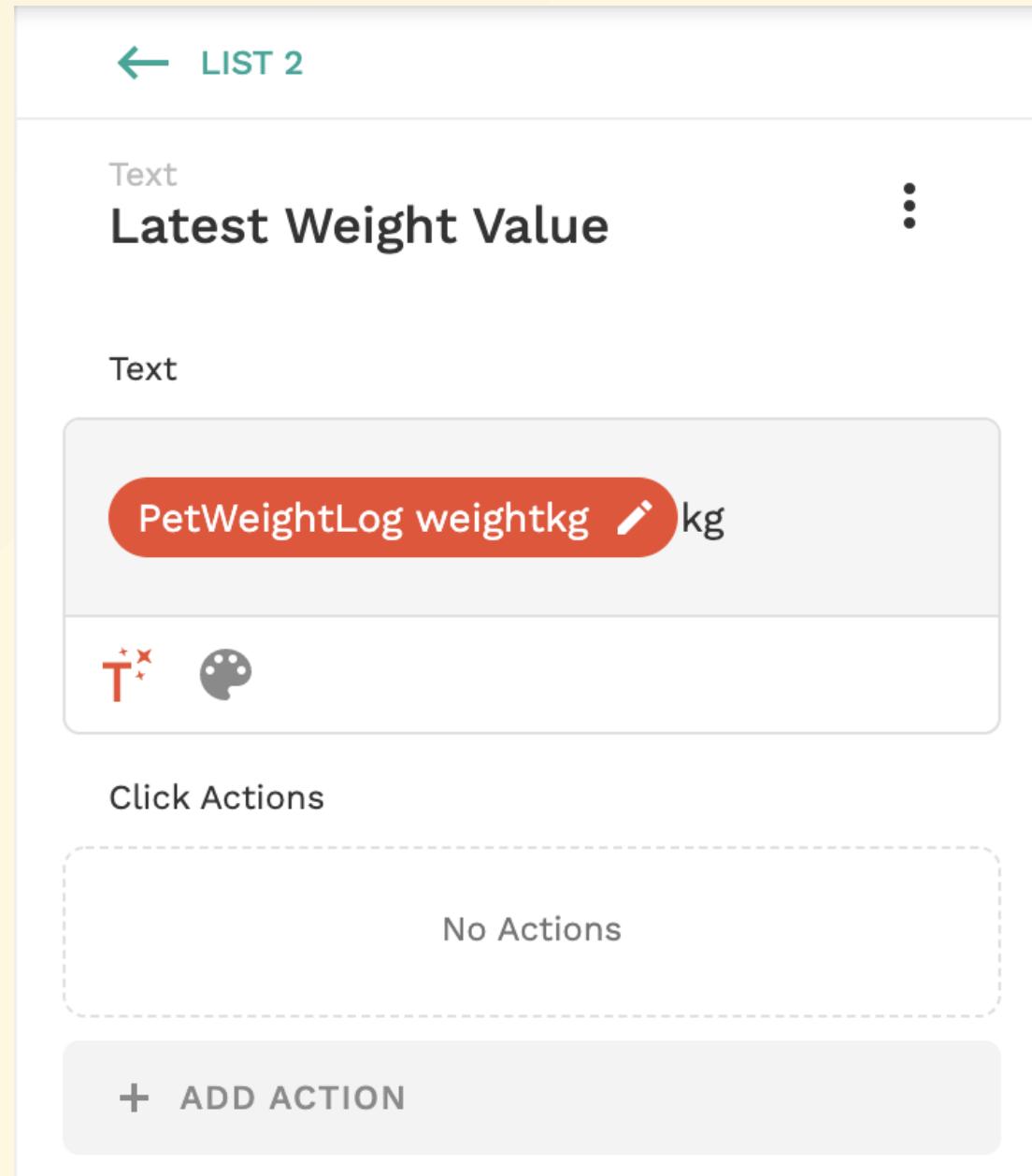


- Select "etWeightLogs" in "What is this a list of?"
- Select "Current Pet" > "PetWeightLogs" in Filter
- In Sorting, select "WeightRegisteredTime - Newest to Oldest"
- Set "Maximum number of items" to 1

This will narrow down the list to only the latest one.

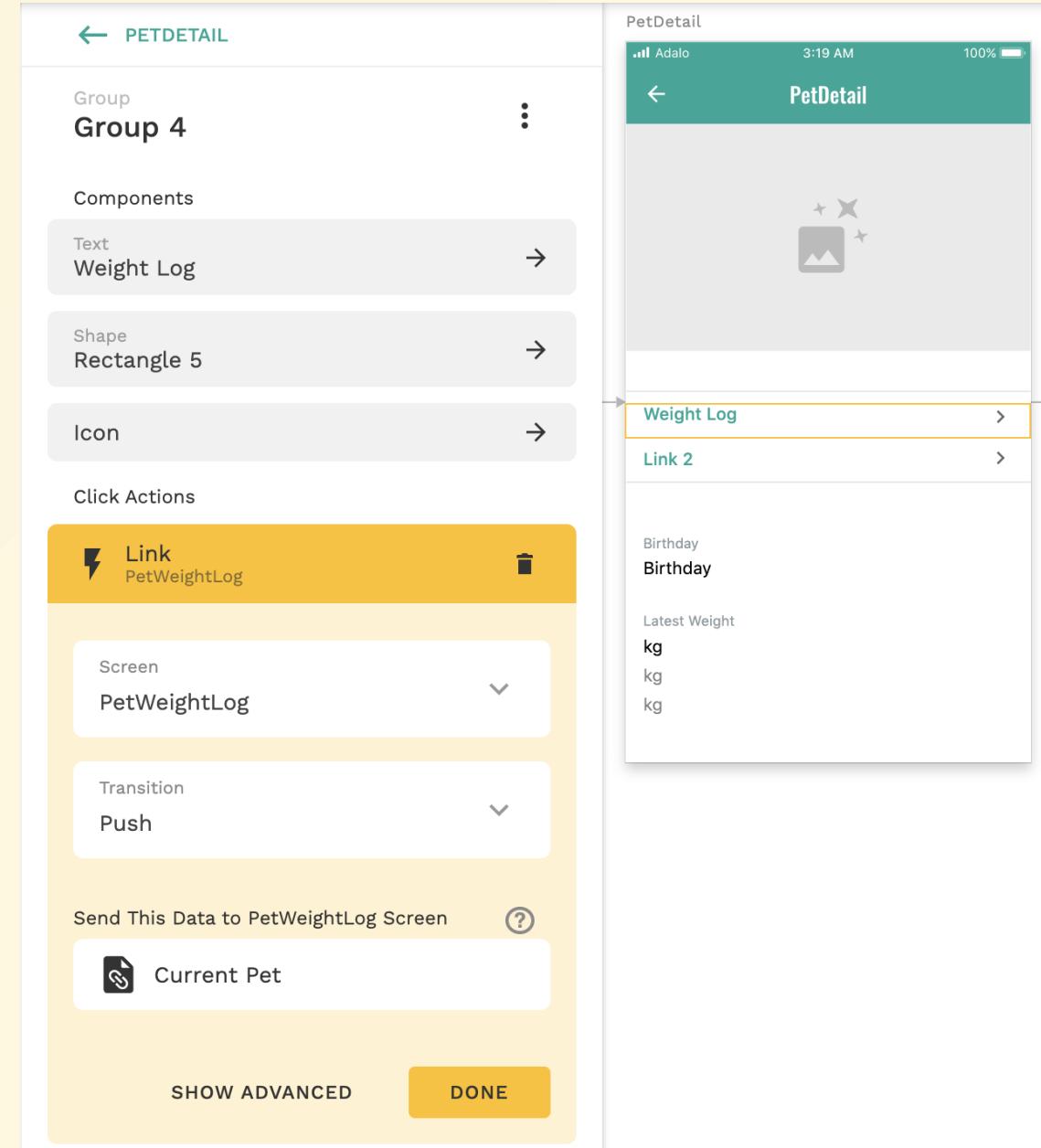


- Click the Latest Weight Value component, add select "Current PetWeightLog's" > "WeightKg" to the Text, then add "kg" to the end.



Make sure that the selected pets can be passed on when moving from the Pet Detail screen to the Pet Weight Management screen.

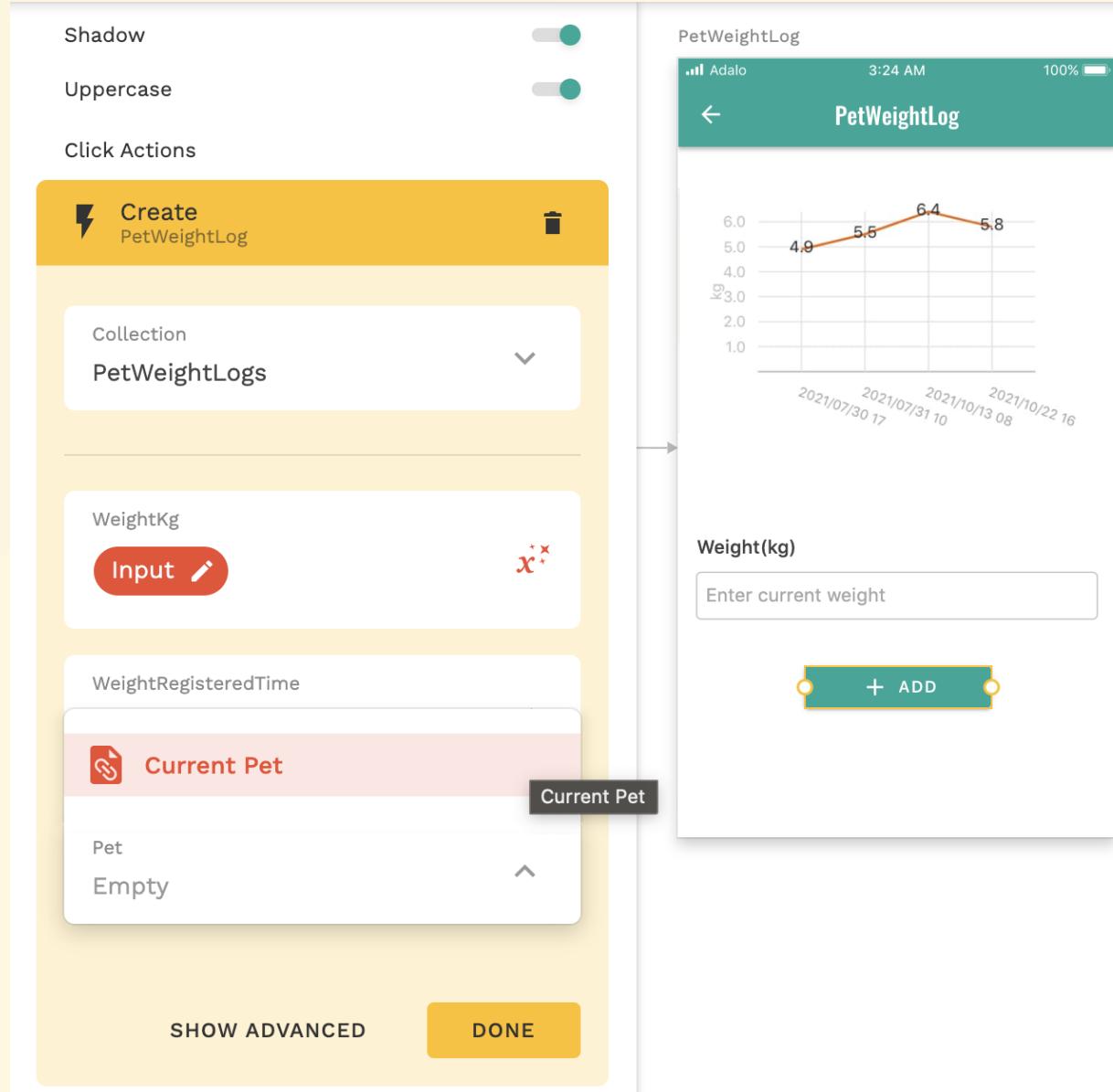
- Select the component of the link "Weight Log", and confirm that "Current Pet" is automatically set in "Send This Data to PetDetail Screen".



(The remaining work of creating data)

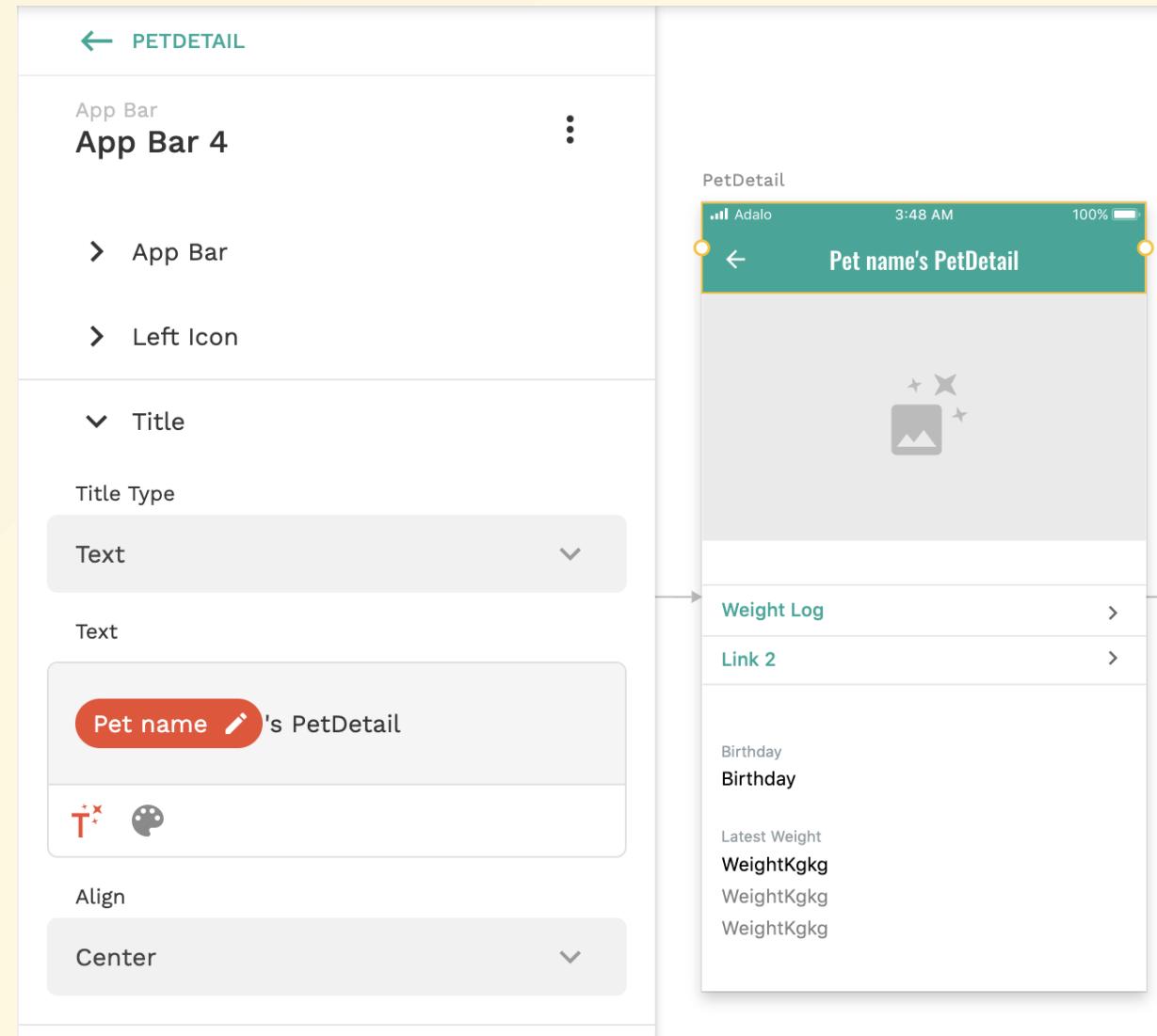
Allow added weight in the Pet Weight Management screen to be registered in the database.

- Select the "ADD" button, and set "Current Pet" to the "Pet" (that was once left as Empty).



Display the name of the selected pet in the header of the Pet Details screen and the Pet Weight Management screen. On each screen,

- Select the App Bar component at the top of the screen and add "Current Pet's" > "Name" to the Title Text.
- Add the string "'s" after it.

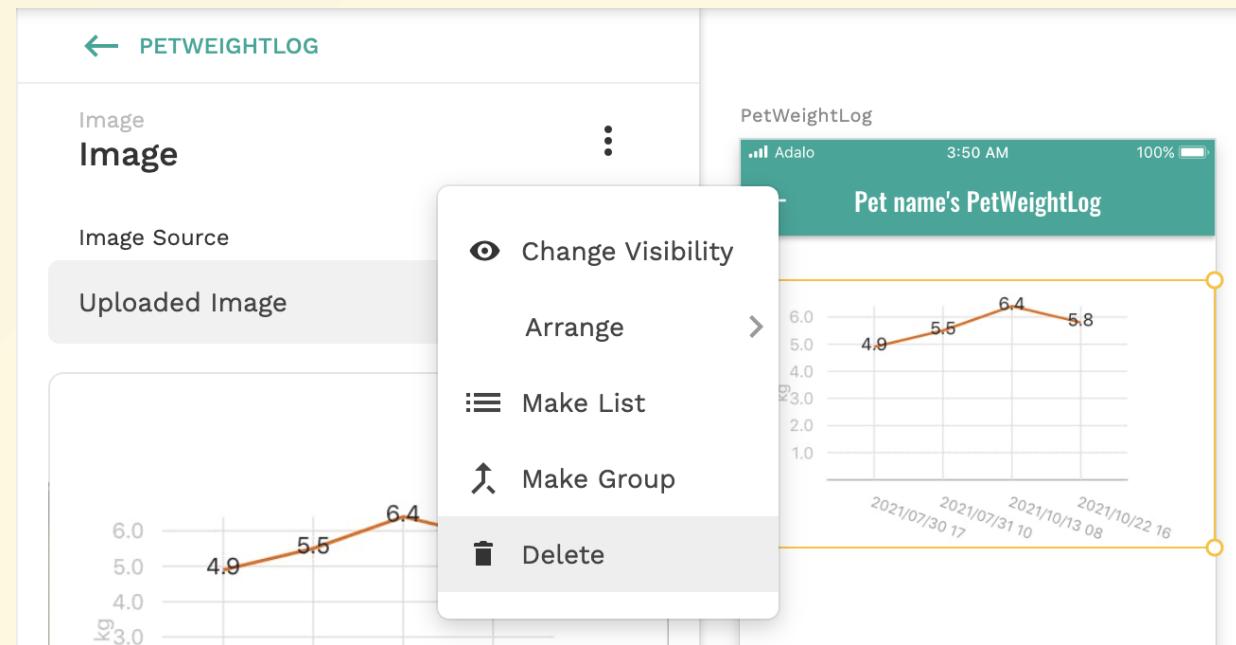


In the Preview function, check the following.

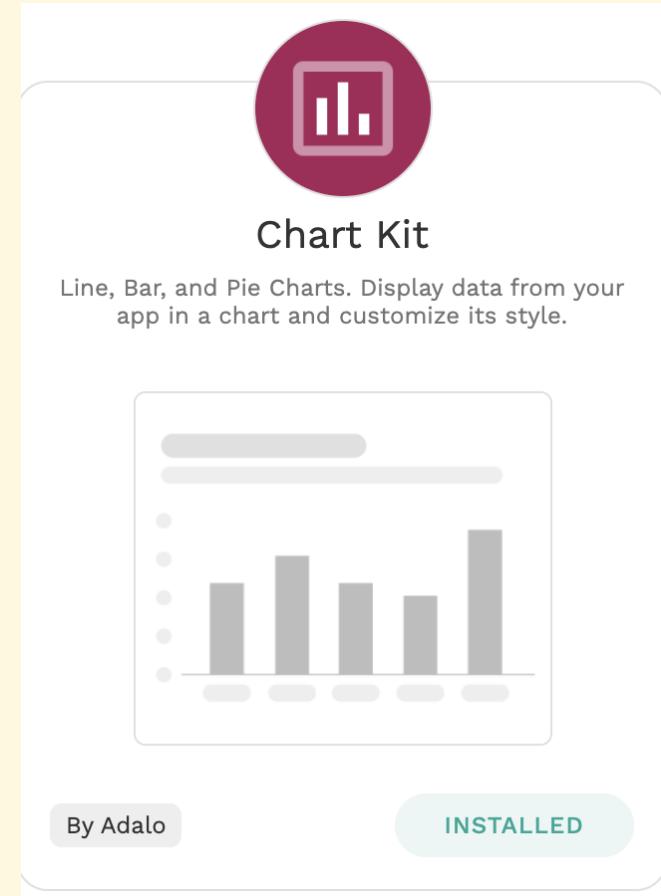
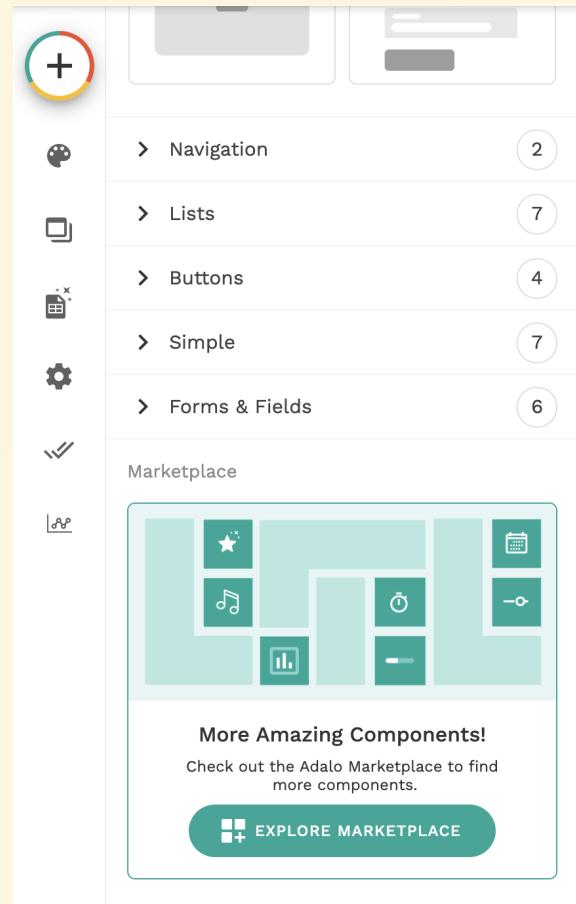
- You can move to the detail screen of the selected Pet from the Pet List screen.
- It is possible to move to the weight management screen of the selected pet from the pet detail screen.
- When the pet weight is registered, the record is registered in the PetWeightLogs Collection of the database.
- The latest weight is displayed on the pet detail screen.

Next, we will display registered pet's weight logs as a graph in the pet weight management screen.

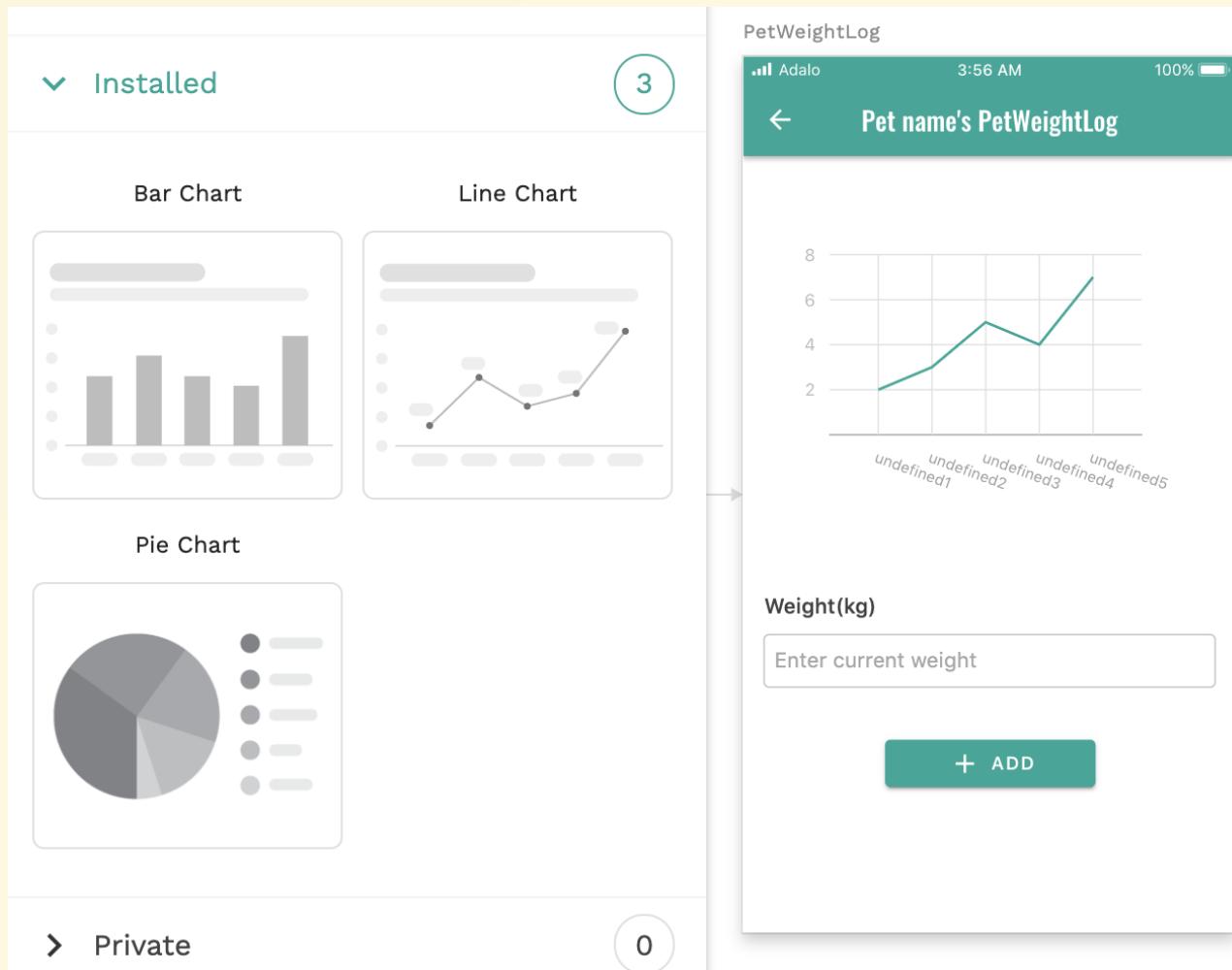
- First, delete the graph that was pasted as an image



- Select "EXPLORE MARKETPLACE" from "ADD COMPONENT"
- Install "Chart Kit"



- Add "Line Chart" to the screen



## Set up a Line Chart.

- Select "PetWeightLogs" in "What is this a chart of?"
- Select "Current Pet" > "PetWeightLogs" in Filter
- In Custom Filter, set "WeightRegisteredTime", "Is after", "30 days ago" to specify display period.
- In Sorting, select "WeightRegisteredTime - Oldest to Newest"

Line Chart  
Line Chart

▼ Line Chart

What is this a chart of?

PetWeightLogs

Filter

Current Pet > PetWeightLogs

Custom Filter

WeightRegisteredTime

Is after

30 days ago

+ ADD ANOTHER FILTER

Sorting

WeightRegisteredTime - Oldest to Ne...

Maximum number of items

No Maximum

ADVANCED OPTIONS

- Set X Axis Value to "PetWeightLog" > "WeightRegisteredTime"
  - Set Date Format to "Date / Time"
- Set Y Axis Value to "PetWeightLog" > "WeightKg"

The screenshot shows a configuration interface for a visualization, likely a line chart. It includes two main panels: one for the X-axis and one for the Y-axis.

**X Axis Value:** Set to "PetWeightLog weightregisteredtime".

**Y Axis Value:** Set to "PetWeightLog weightkg".

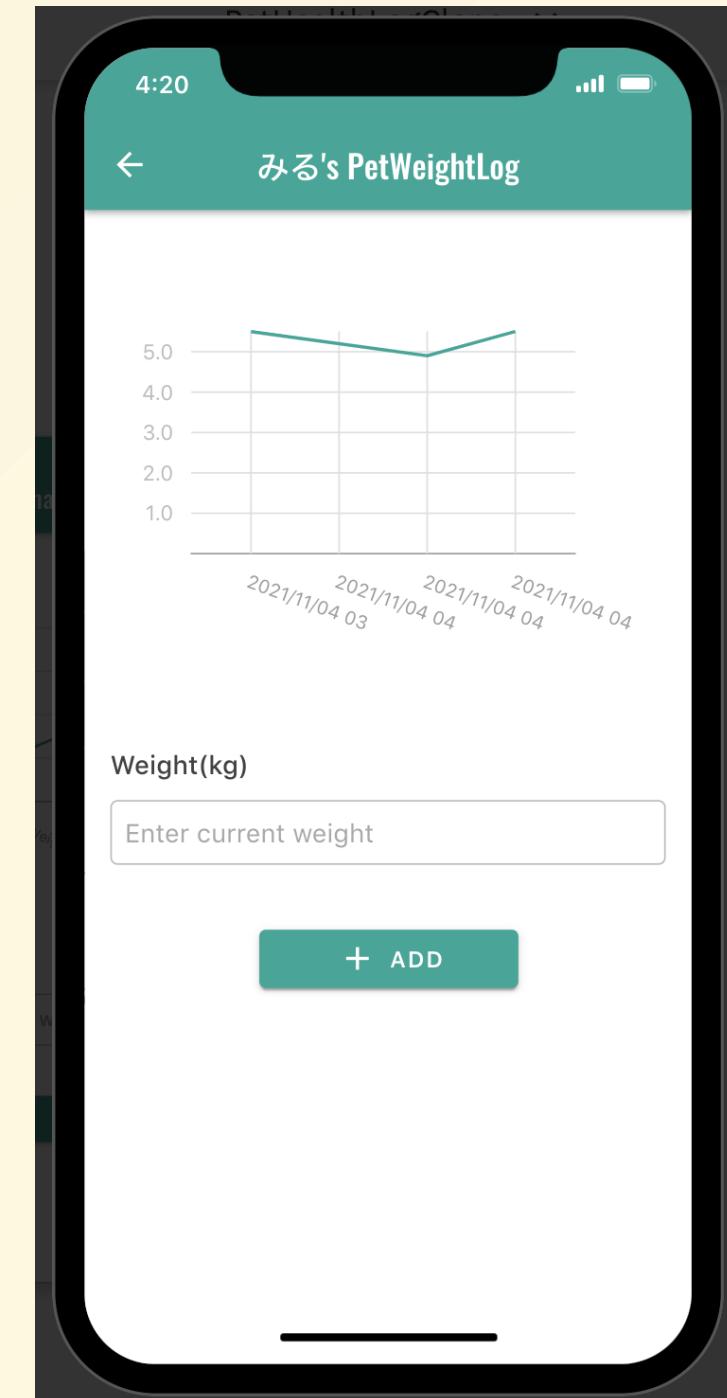
**Point Action:** Set to "No Actions".

**Date Format:** Set to "Date / Time".

Let's preview the graph. If you add multiple weights, the graph will be drawn.

(The registration time of the weight is too long and will be displayed abbreviated...)

We used the Date&Time type to test it by registering multiple weights on the same day, but it would be better to use the Date type and restrict multiple registration on the same day.)

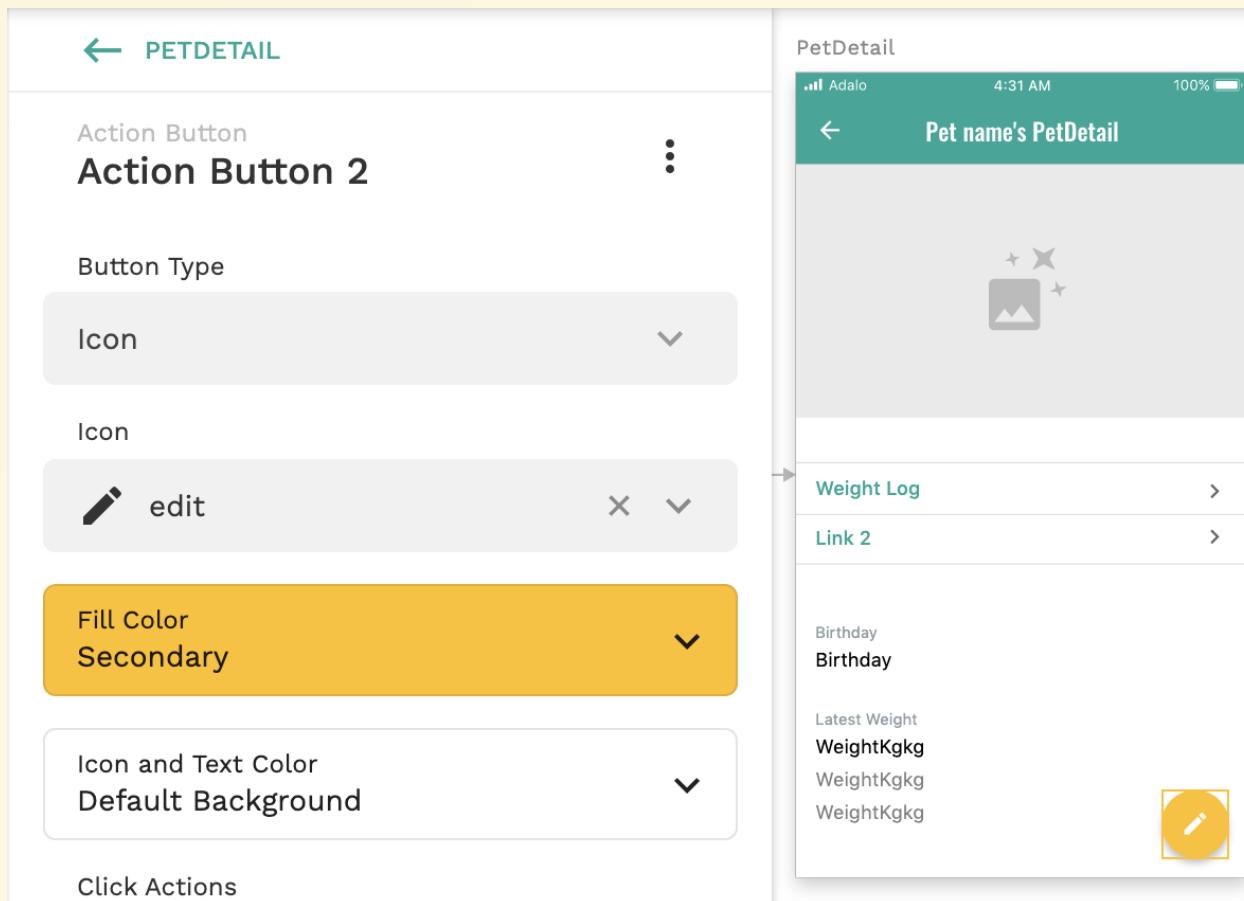


## Updating Data

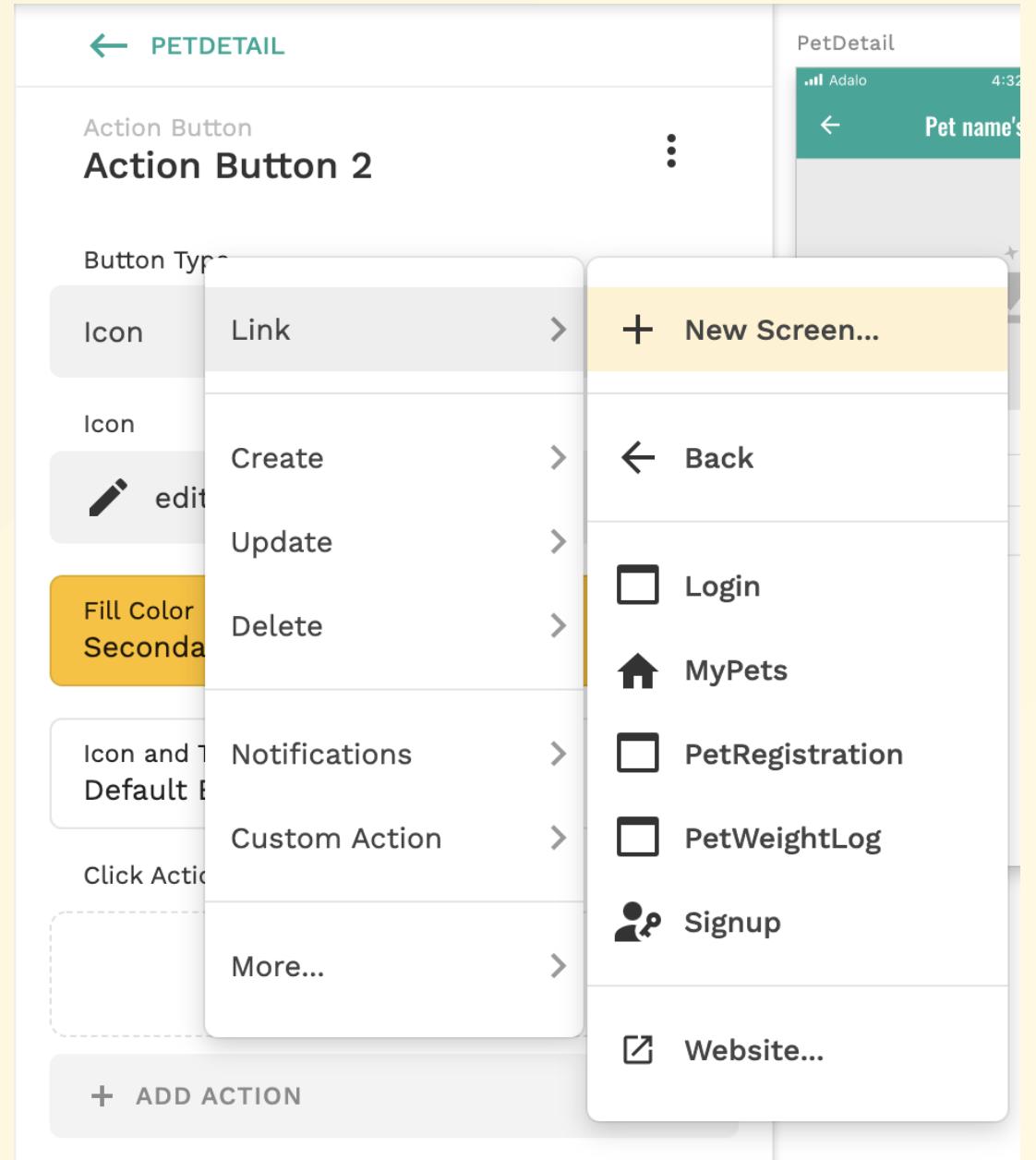
Create a new "EditPetInfo" screen where you can update the information of registered pet.

First, add a link to the Edit Pet Info screen in the Pet Detail screen.

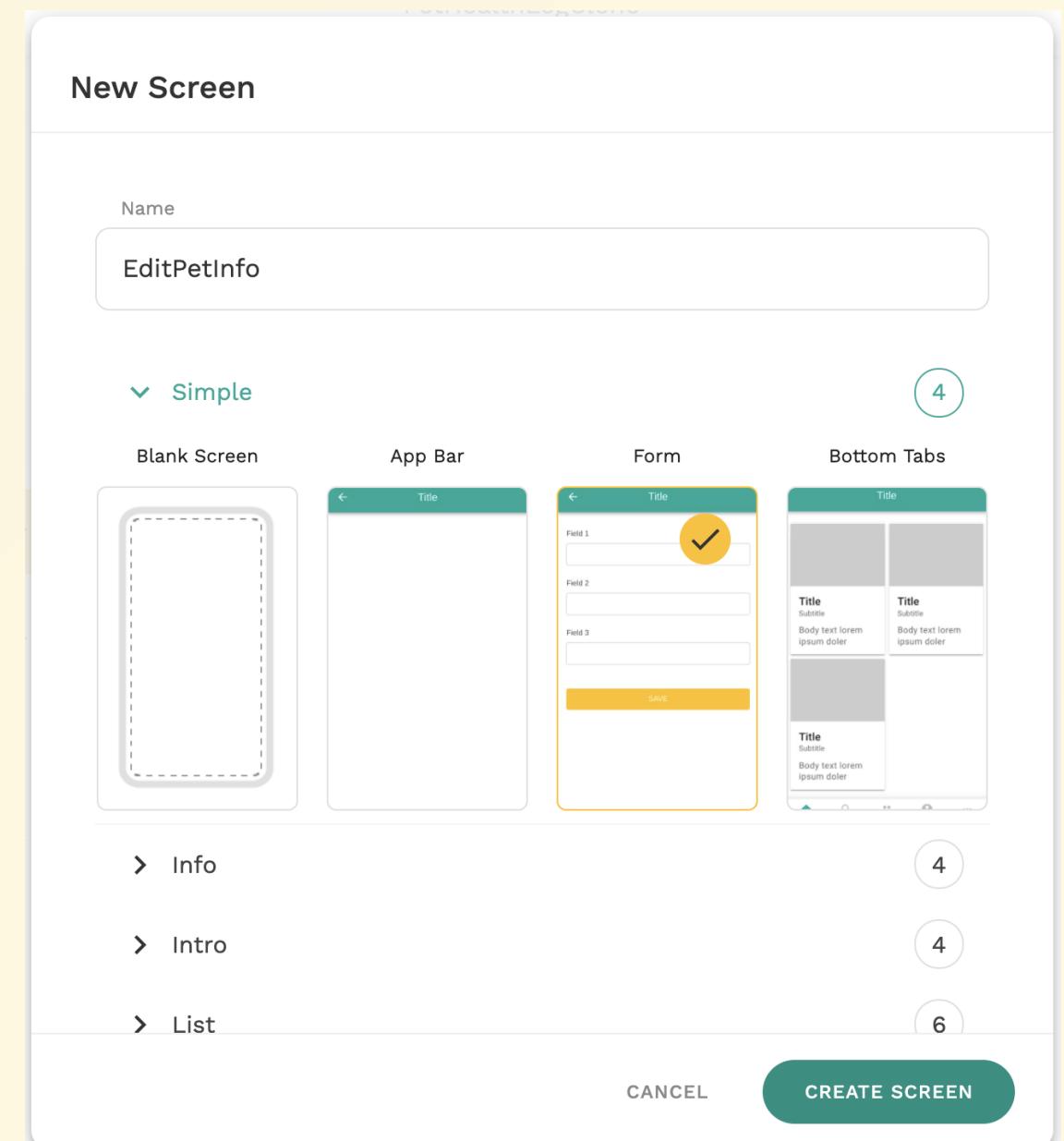
- Add "Action Button" from "ADD COMPONENT"
- Change the Icon to "edit"
- Change Icon and Text Color to "Default Background" (white)



- Select "Link" > "New Screen" from ADD ACTION



- Enter "EditPetInfo" in a Screen Name
- Select "Form"
- Click "CREATE SCREEN"

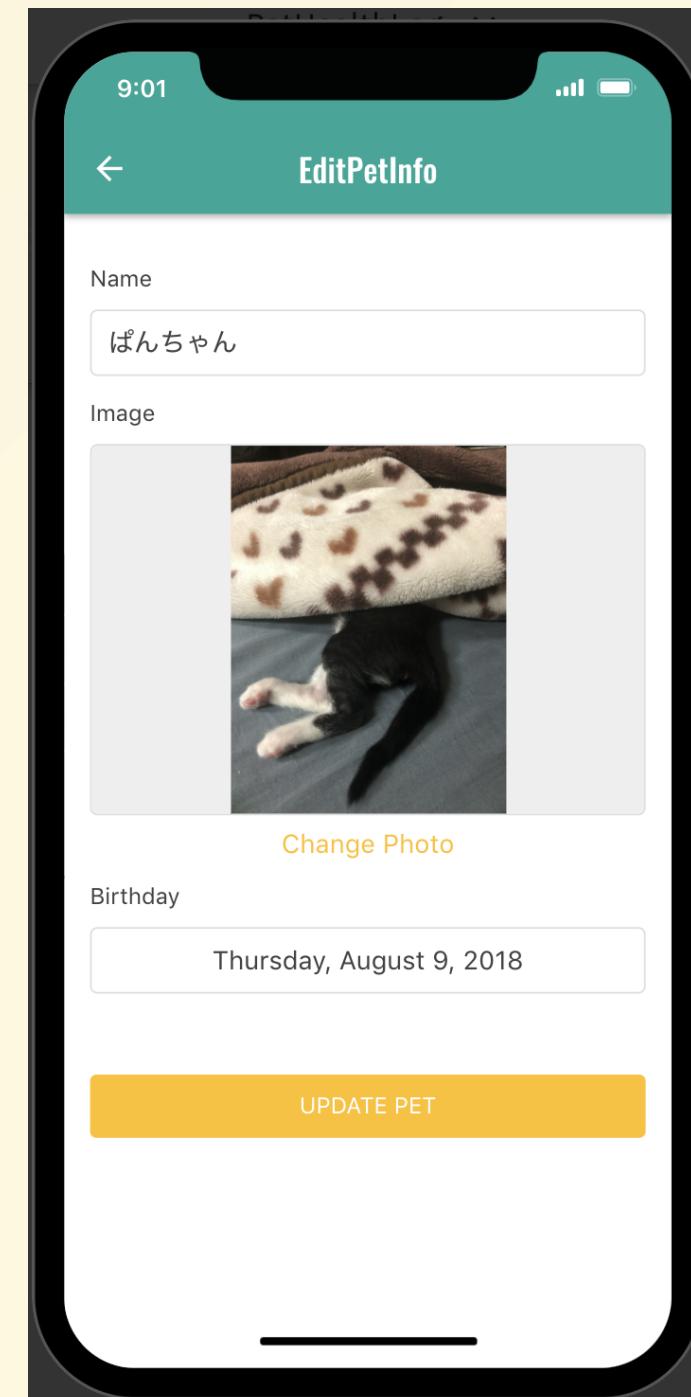


- Select "Pets" in "Which data collection?"
  - The form is automatically generated according to the collection.
- Select "Update Current Pet" in "What do you want the form to do?"
- Change the order of "Birthday" and "Image" in Fields

The screenshot displays the 'EditPetInfo' screen from the Adalo mobile application. At the top, the navigation bar shows 'EditPetInfo' and the status bar indicates 'Adalo', '4:37 AM', and '100%'. The main content area features three input fields: 'Name' (placeholder 'Enter name...'), 'Birthday' (placeholder 'Thursday, May 4th, 2019'), and 'Image' (with a 'Choose Photo' button). Below these fields is a large orange 'UPDATE PET' button. To the left, a sidebar titled 'Form' provides configuration options: 'Which data collection?' is set to 'Pets', 'What do you want the form to do?' is set to 'Update Current Pet', and under 'Fields', 'Birthday' and 'Image' are listed. The overall interface is clean and modern, designed for mobile devices.

EditPetInfo screen is ready.

Make sure that you can edit the pet information.



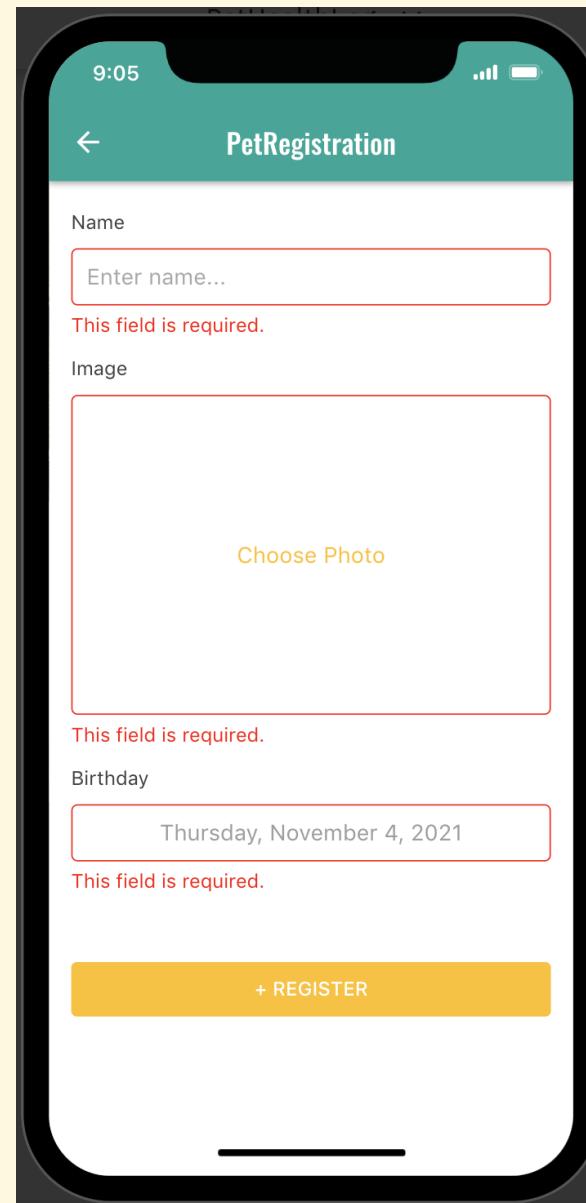
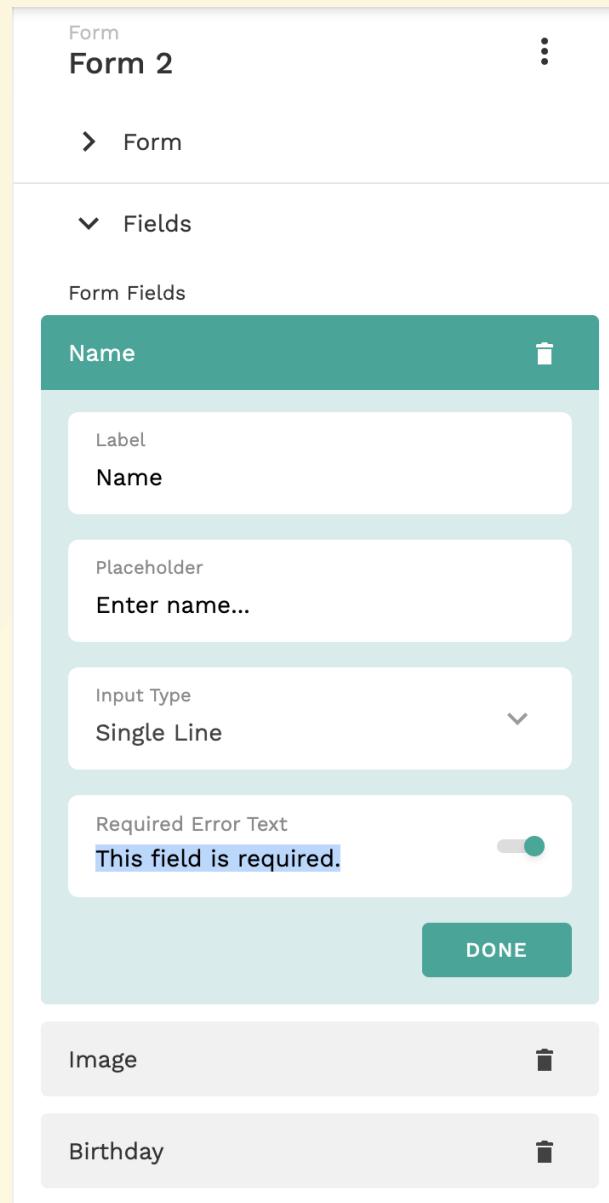
## Supplement

- In the previous article, we created the input form for the pet registration screen by adding components one by one, but it can be automatically generated in the same way as we created the pet edit screen.

- Let's delete all components except "AppBar" in the pet registration screen, add a "Form" component, and specify "Pet Collection" to generate the form automatically.
  - Then, manually add link to the pet list screen after registration by adding action to the Submit Button.

# Additional Information

- Using the Form component, you can check if the required fields are filled in by just checking "the Required Error Text" of the field item.
- Use the Form component as much as possible when creating input forms.



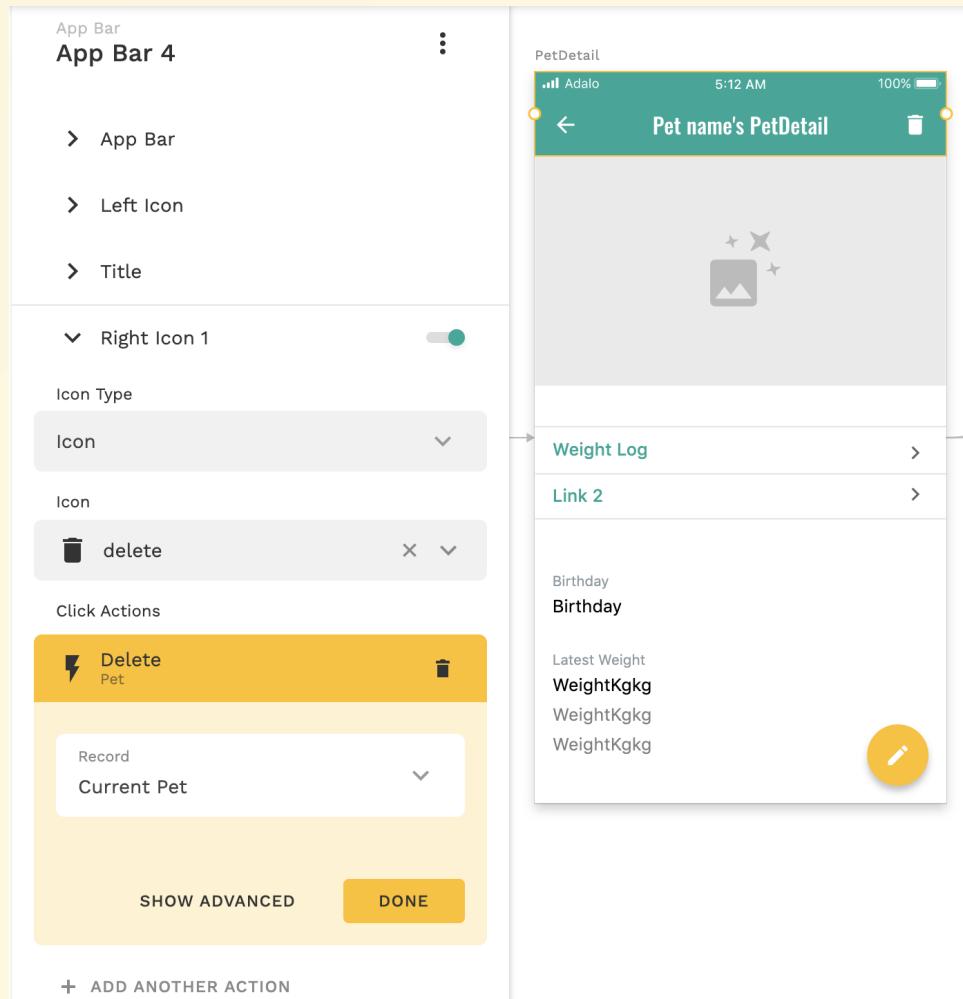
## Reference: The actual flow of application development in Adalo

- In the sample application, we have proceeded with the following flow: UI creation > database design > linkage between them.
  - This is because it seems difficult to design the database without knowing what kind of screen and data we need.
- When actually developing an application with Adalo, It is recommended that first draw a UI sketch and identify the necessary data, and then design the database. By doing so, you can take advantage of automatic generation for UI creation.
  - However, the actual development would not be a one-way flow, but a trial-and-error process, alternating between database design and UI design.

## Deleting Data

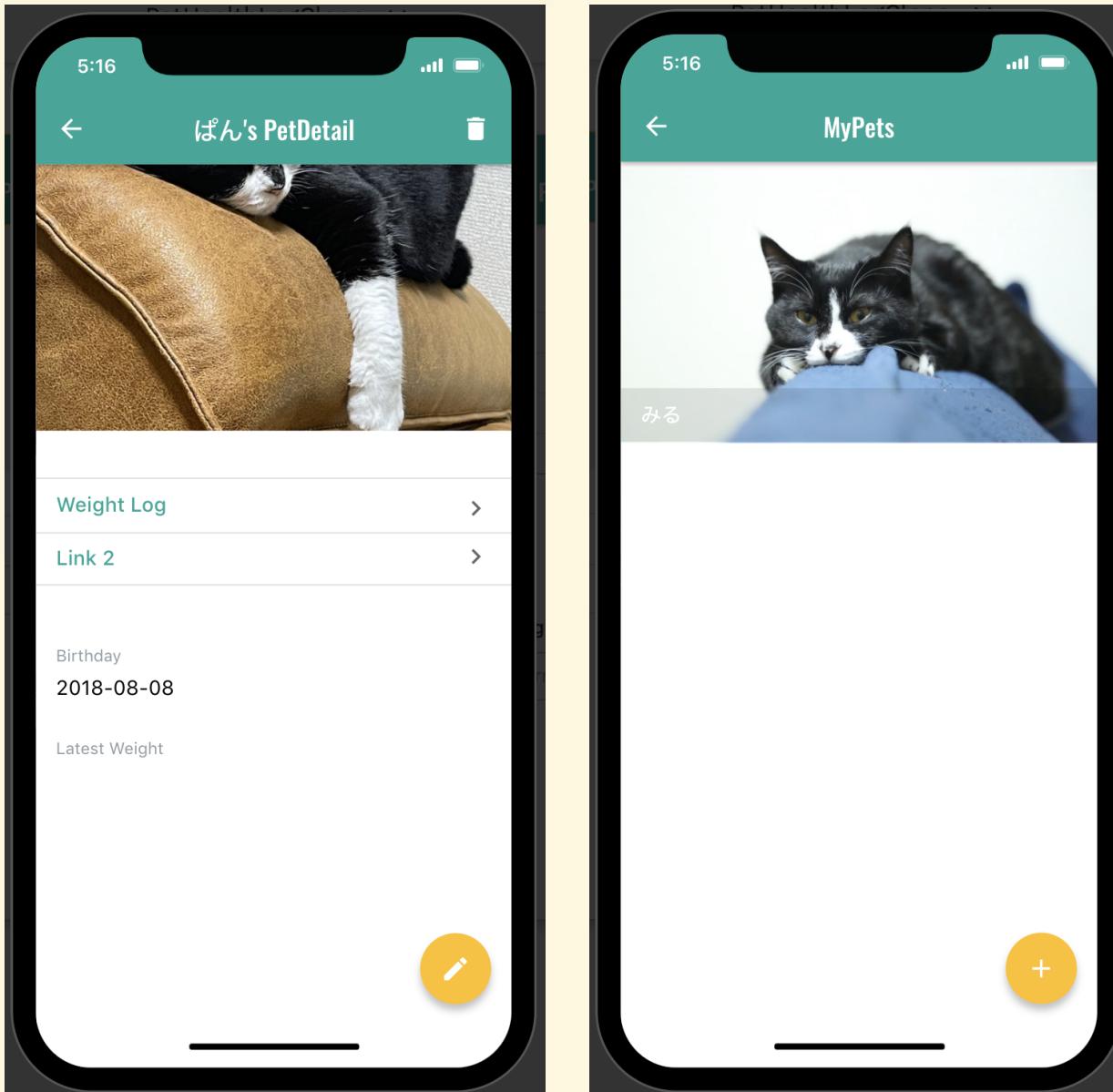
Create a button to delete a registered pet in the pet details screen.

- Select the App Bar and turn on the "Right Icon 1"'s toggle
- Change Icon to "delete"
- Select "Delete" > "Current Pet" from "ADD ACTION"
- Select "Link" > "Mypets" from "ADD ANOTHER ACTION"



Let's try deleting with the Preview function.

When the deletion is completed, you will be redirected to the Pet List screen. The deleted pets will not be displayed.



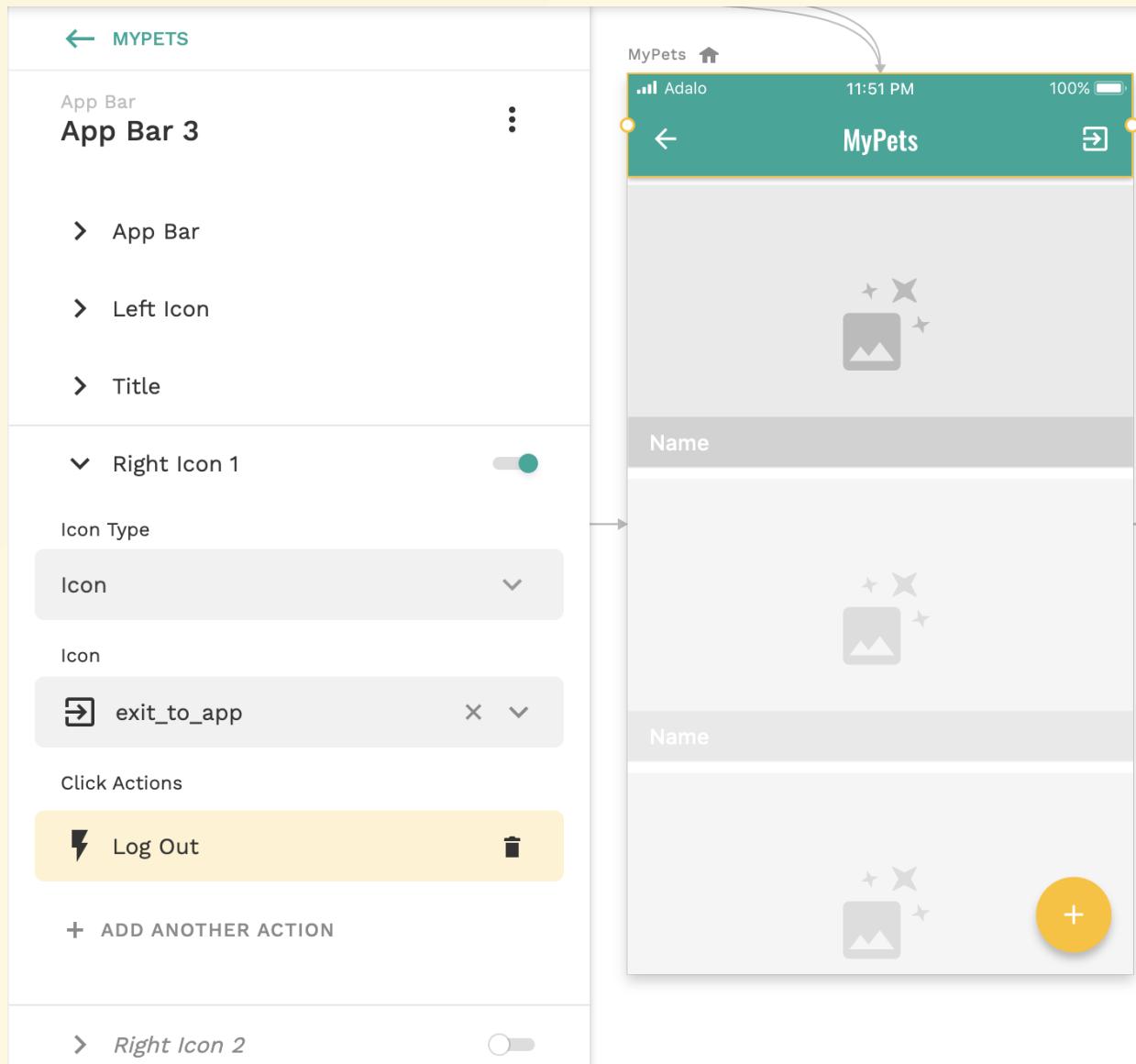
We have now implemented all kinds of CRUD(CREATE, READ, UPDATE, DELETE) operations.

# Improve sample application

Let's improve the sample application by using the features of Adalo that we haven't learned yet.

# Logout

- Select the AppBar in the pet list screen, and activate "Right Icon 1"
- Change Icon to "exit\_to\_app"
- Select "ADD ACTION" > "More..." > "User Login" > "Log Out"
- Select "ADD ANOTHER ACTION" > "Link" > "Login"

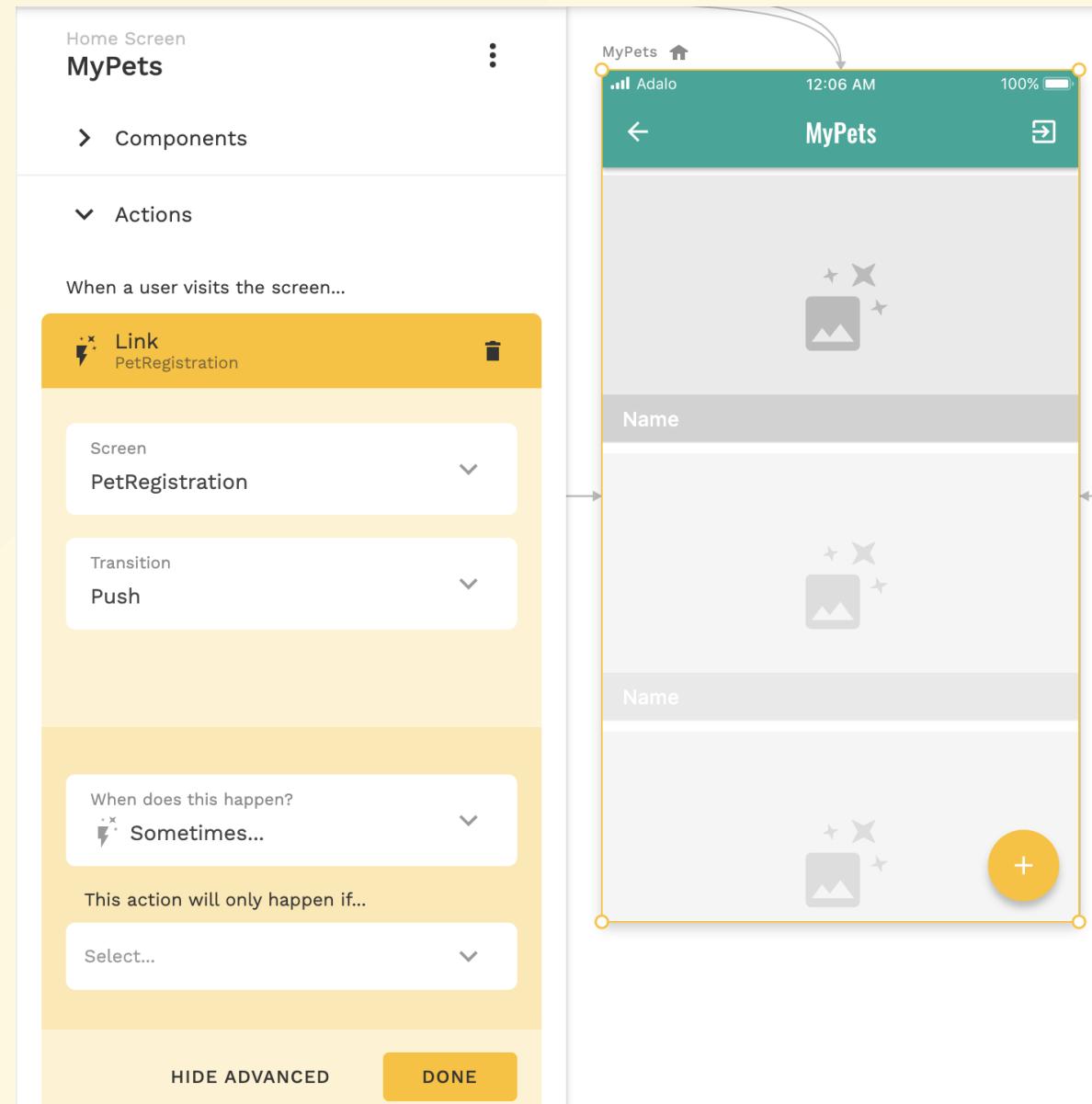


Check it with the Preview function.

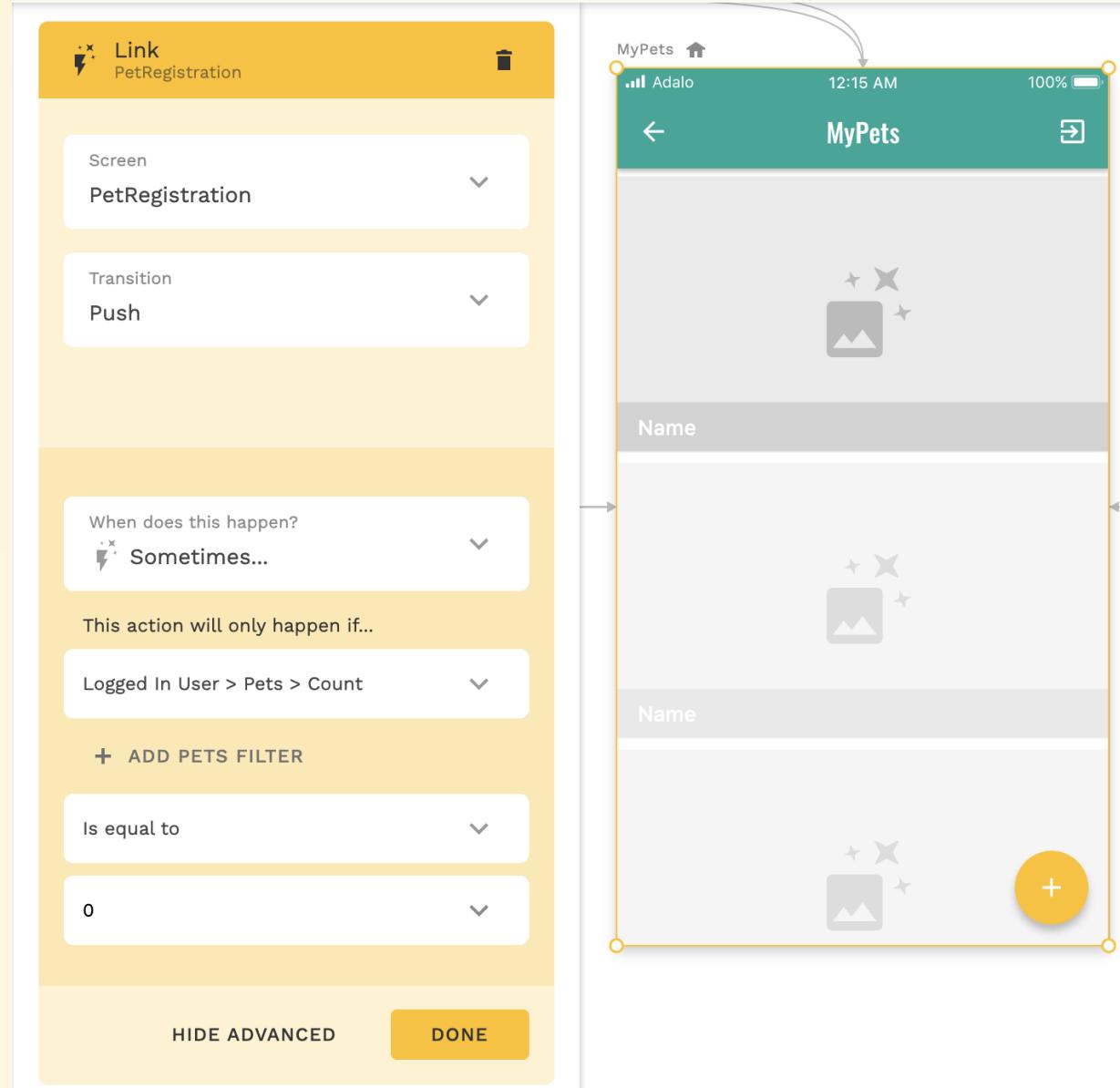
# Action execution condition setting

If no pets have been registered, redirect users from Pet List screen to Pet Registration screen.

- In the Pet List screen, select "Actions" > "ADD ACTION" > "Link" > "Pet Registration"
- Click "SHOW ADVANCED" and change "When does



- Set "This action will only happen if..." "More" > "Logged In User's" > "Pets" > "Count"
- Change the number under "Is equal to" to 0.



Check it out with the Preview function.

When you sign up as a new user, you will be redirected from the Pet List screen to the Pet Registration screen.

## Selective Input Form

Add gender to the pet's information and allow selective input in the input form.

The choices used in the selective input form are prepared as records by adding a collection to the database.

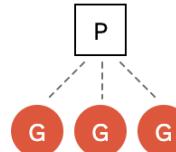
- Add a Genders Collection to the database (leave the Property as default)
- Click "0 Records" > "ADD GENDER" and add two Records, "Male" and "Female".

Genders				
<input type="checkbox"/>	A Name	<span>Created</span>	<span>Updated</span>	
<input type="checkbox"/>	Female	8 minutes ago	8 minutes ago	<span>+</span> ADD GENDER <span>Upload</span> <span>Download</span> <span>Compare</span> <span>Search</span>
<input type="checkbox"/>	Male	8 minutes ago	8 minutes ago	

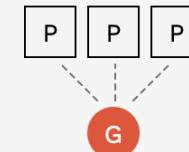
- Add a one-to-many Relationship to the Genders Collection with the Pets Collection
  - Since one pet has one gender, and one gender is set for multiple pets

How are Pets & Genders related?

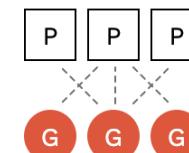
P = Pets    G = Genders



A Pet can have multiple Genders  
A Gender belongs to one Pet



A Pet can only have one Gender  
A Gender can have multiple Pets

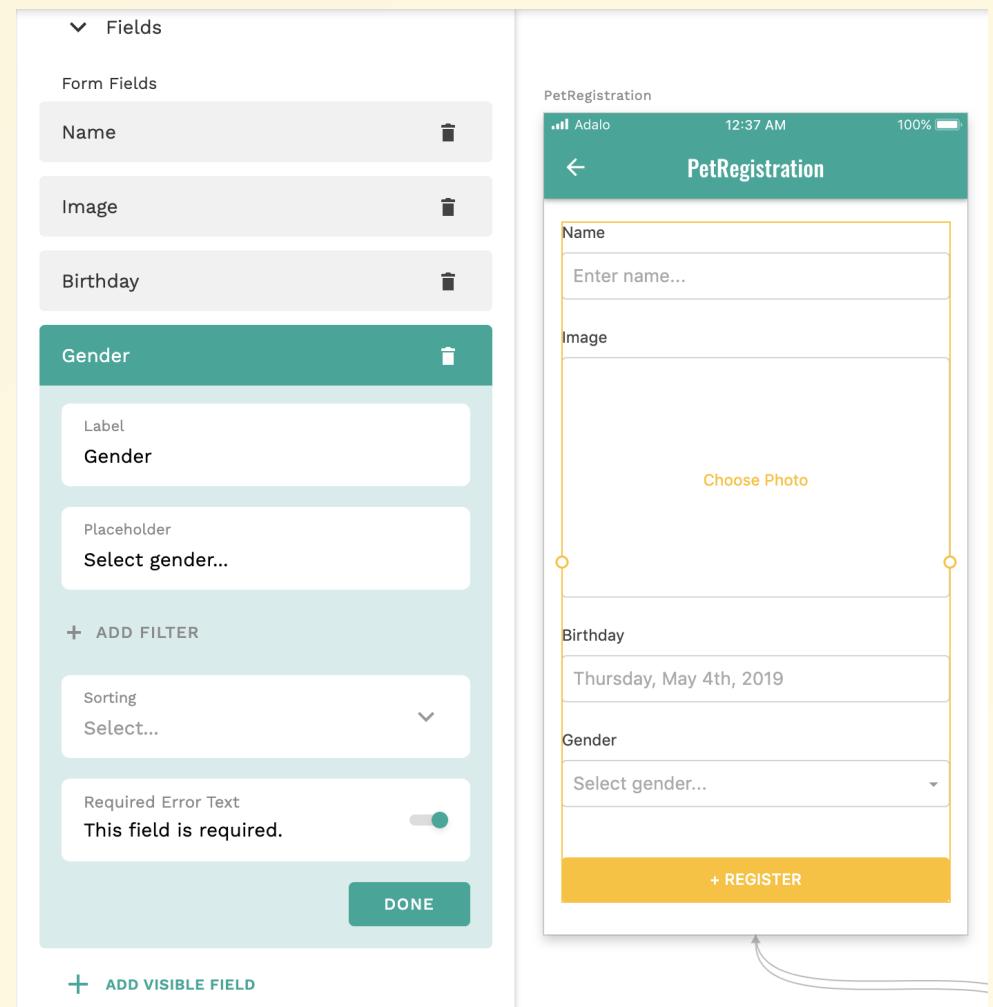


A Gender can have multiple Pets  
A Pet can have multiple Genders

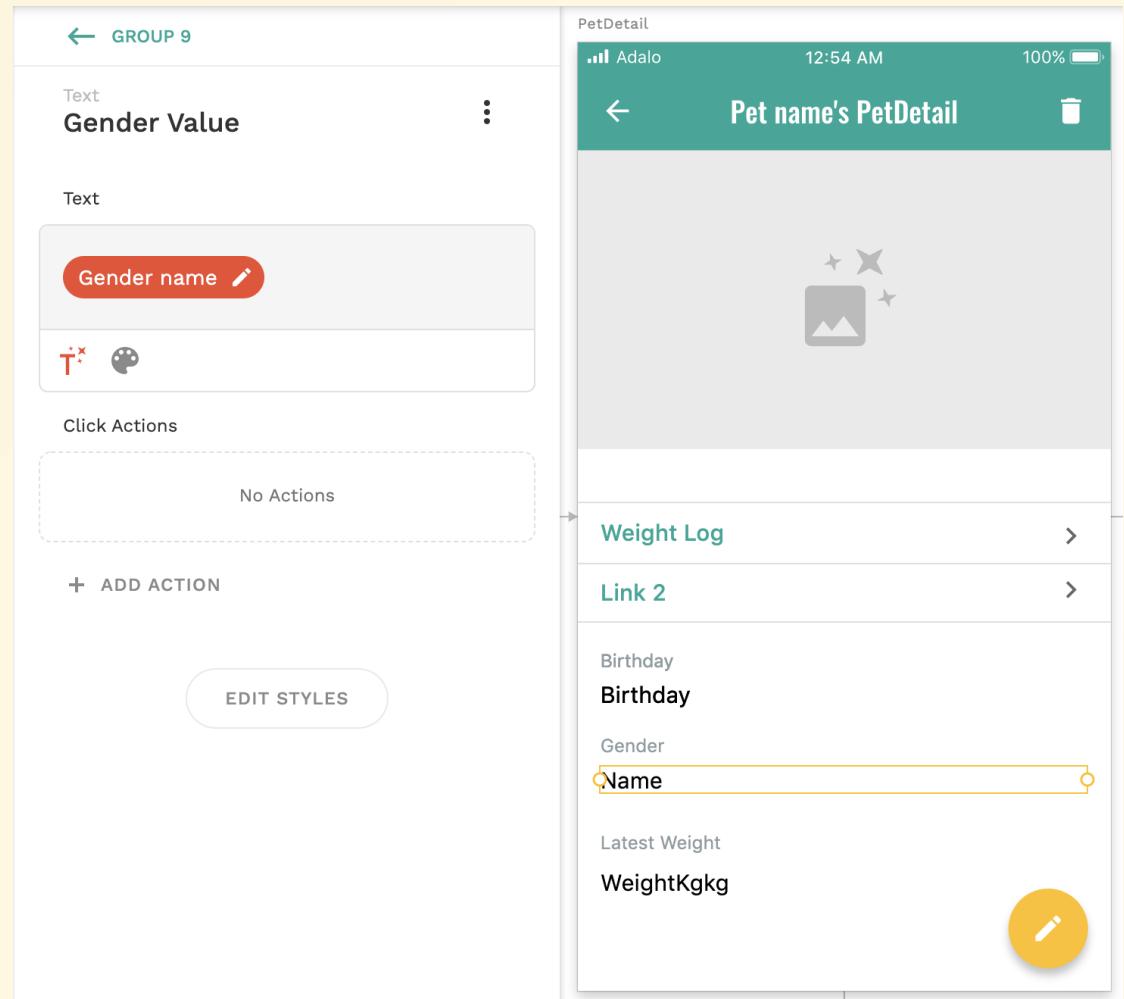
CANCEL

DONE

- Select the form on the pet registration screen
- Select "Fields" > "ADD VISIBLE FIELD" > "Gender"

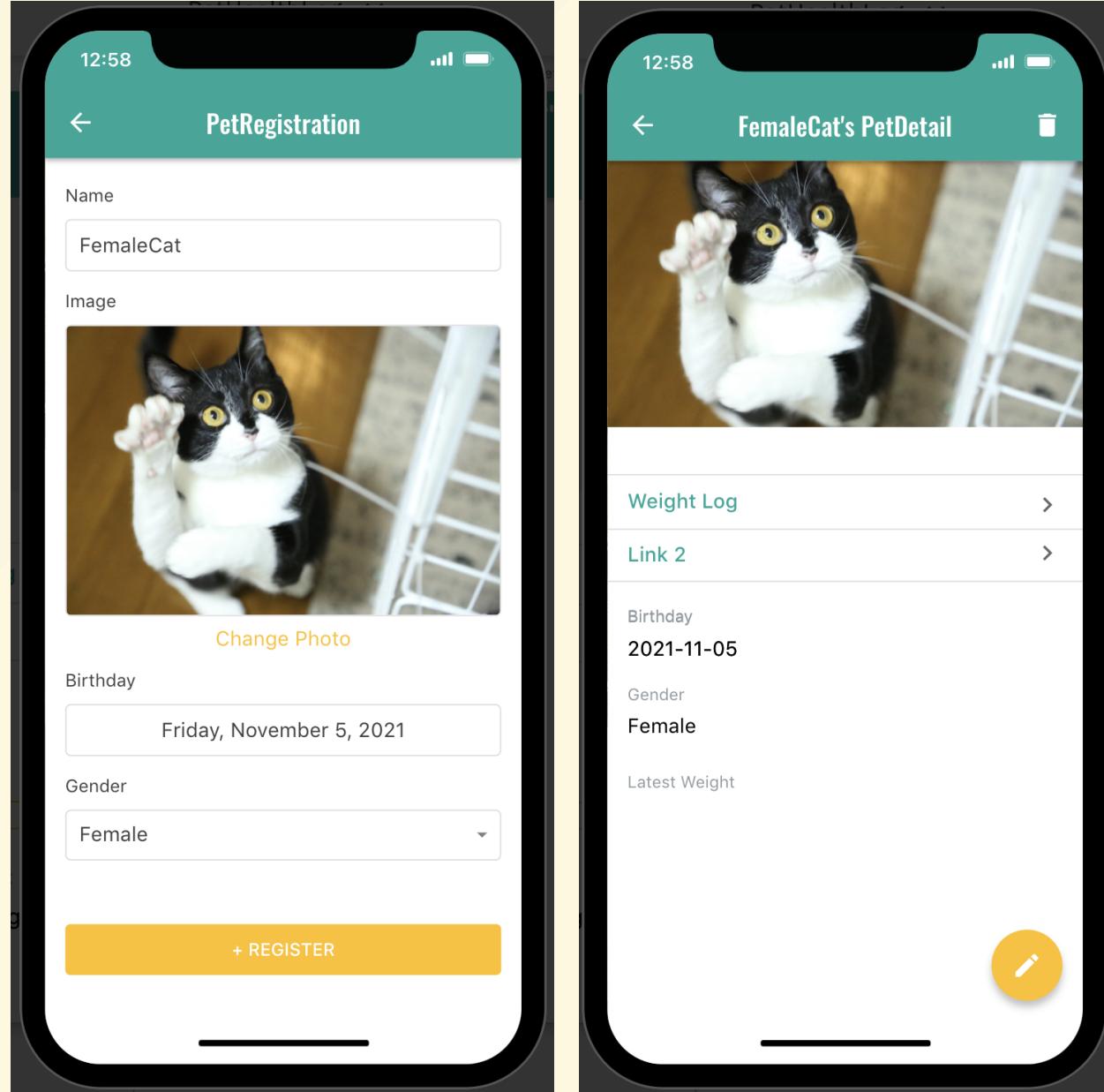


- Add a Gender field to the Pet Detail screen.



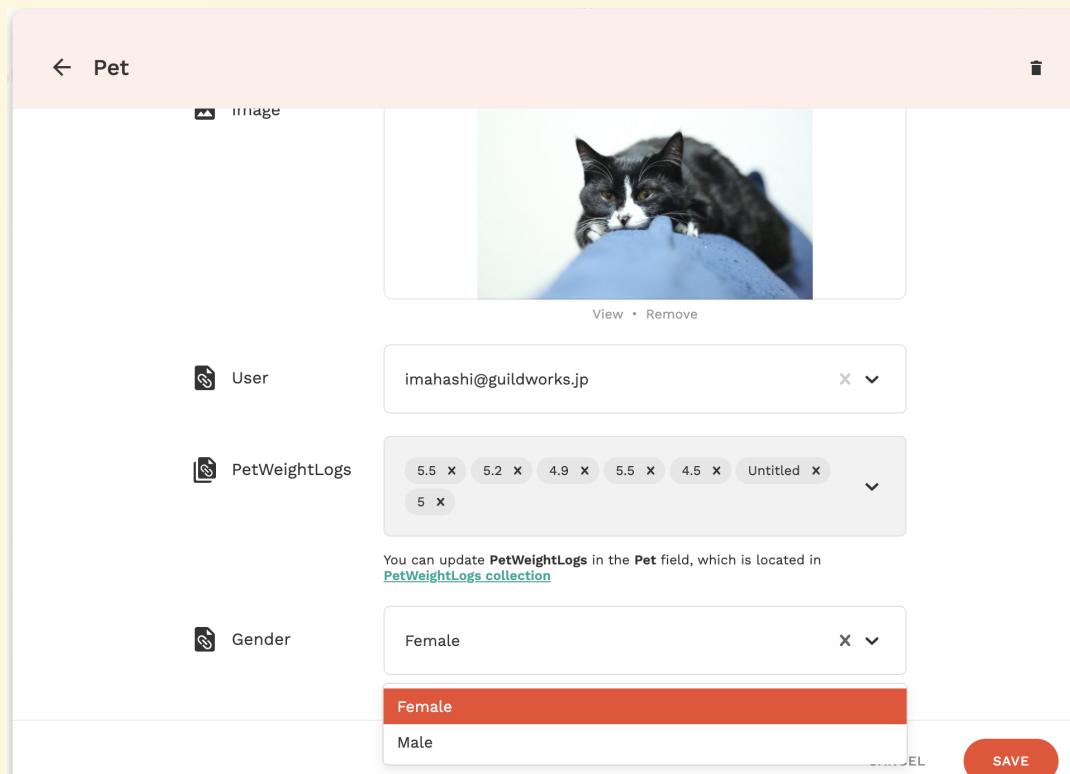
Check it out in the Preview function.

- You can select the gender on the pet registration screen.
- The selected gender will be displayed on the Pet Details screen.



# Supplement

- If you are bothered by the empty fields in the pet you registered before you were able to select the gender, you can go to the Pet Collection Record, click on the pet, and set the gender manually.



## Reference

You can create a multiple-choice input form using Marketplace's MultiselectDropdown.

If you need it, try it out.

 Marketplace



Multiselect Dropdown

Select multiple items from a dynamically-generated list.



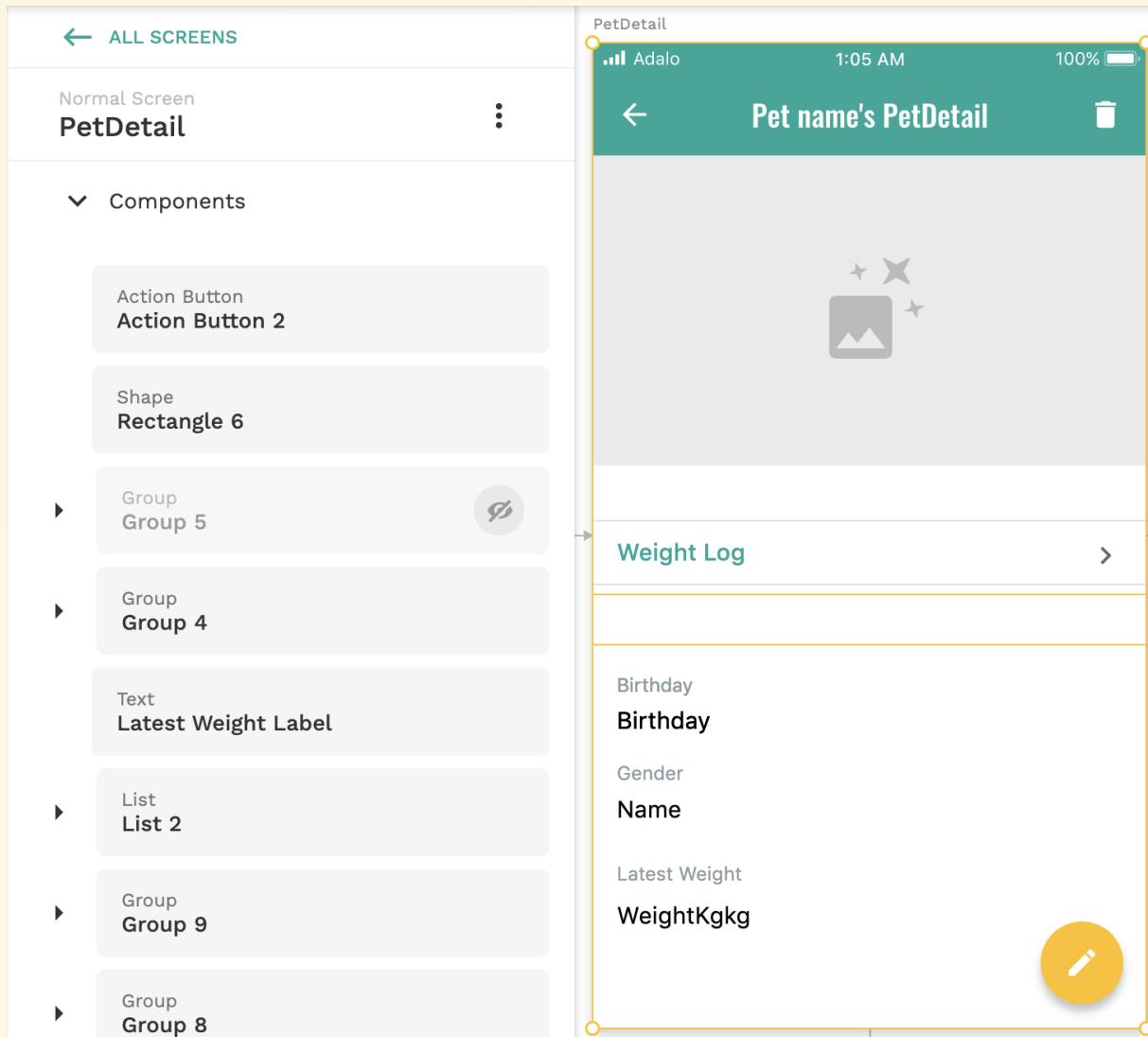
By Conroy Whitney 

INSTALLED

# Hiding components

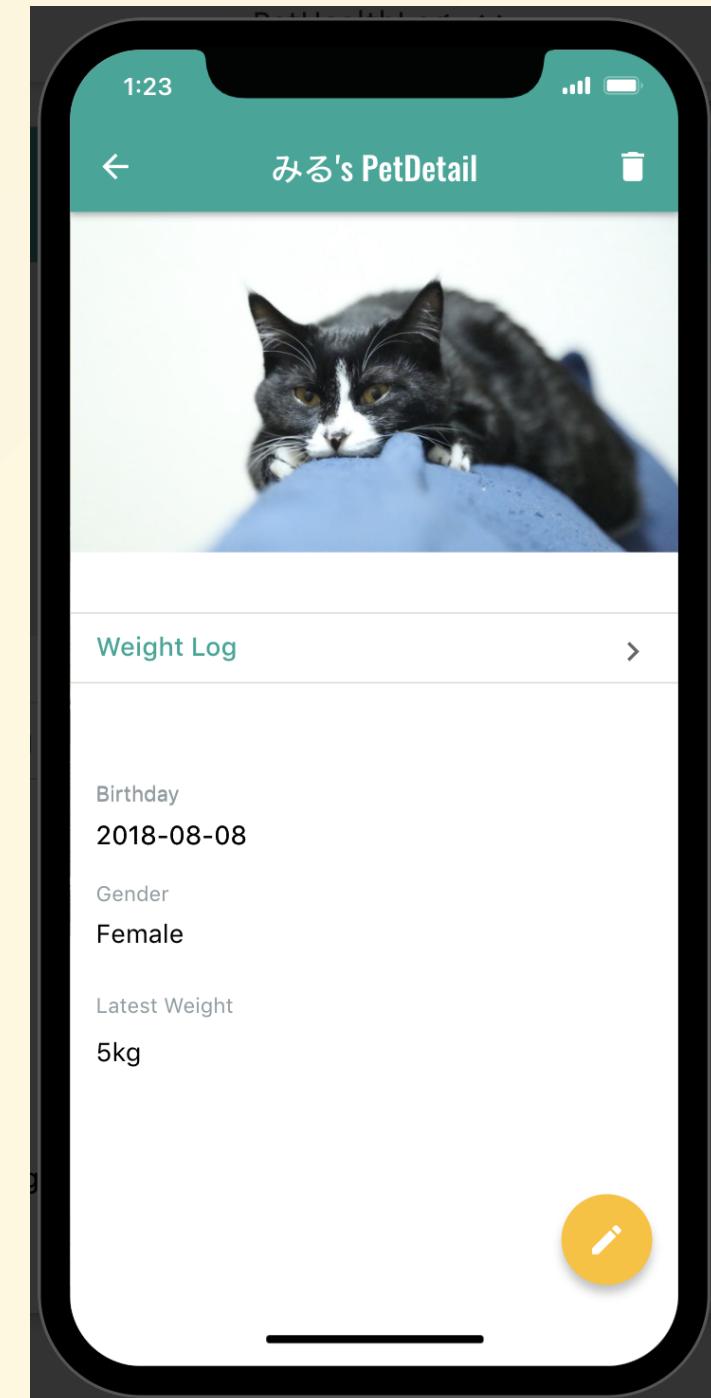
You can keep the components which you do not want to display without deleting them.

- In the component list of the Pet Details screen, mouse over the Group that contains "Link 2", and click the eye icon on the right side



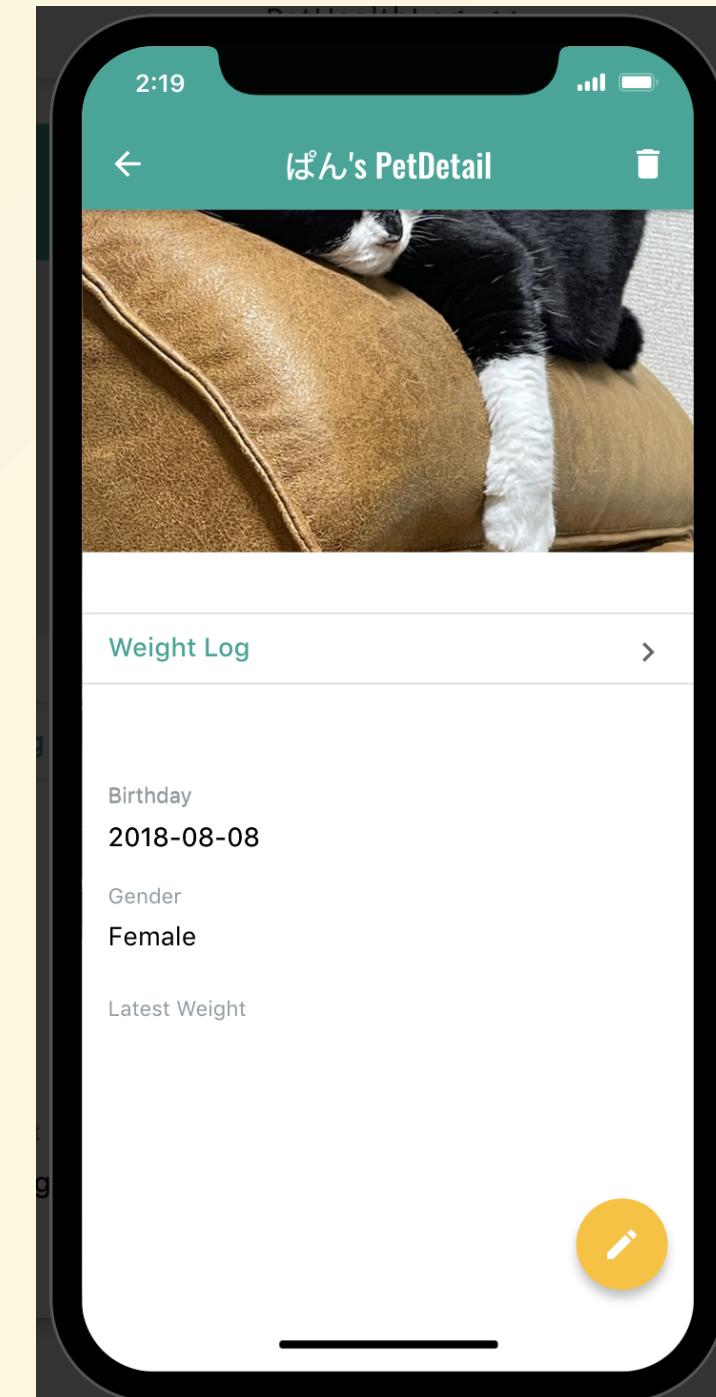
If you check the Preview function, you will see that "Link 2" has disappeared.

You can make it appear again by clicking on the icon a second time.

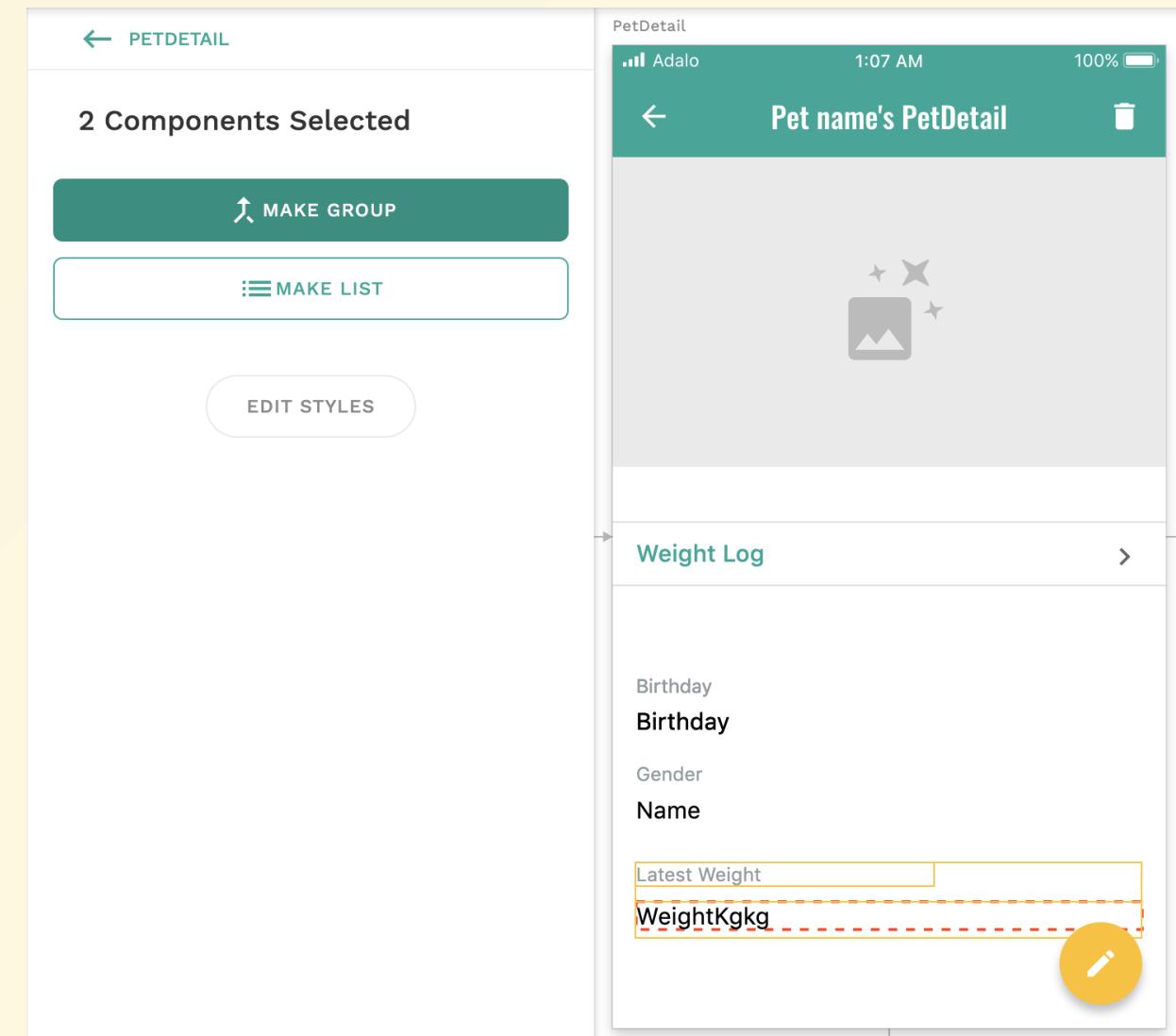


# Show or hide components depending on conditions

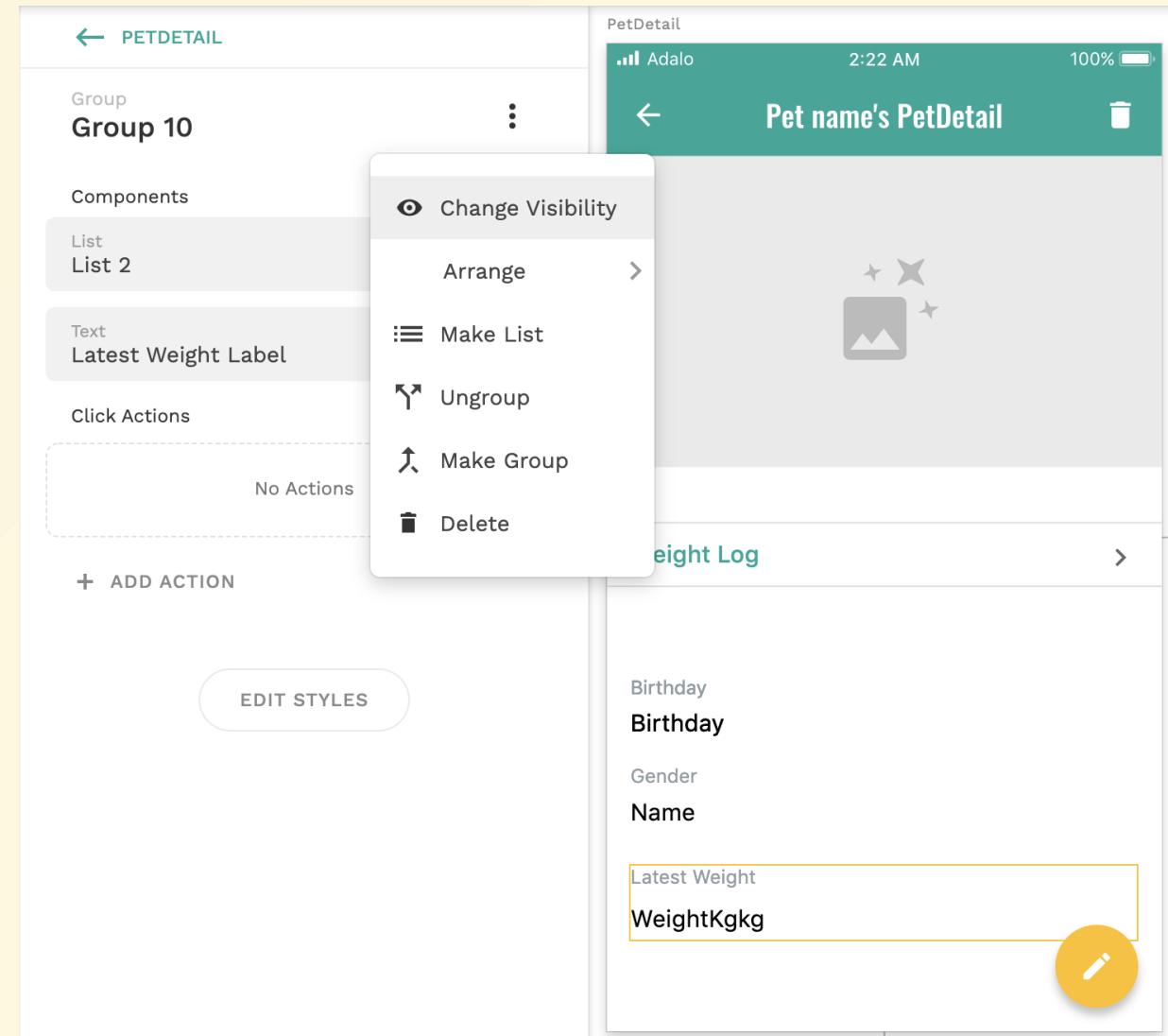
If the weight is not registered, the Latest Weight in the Pet Details screen should be hidden. Let's configure this.



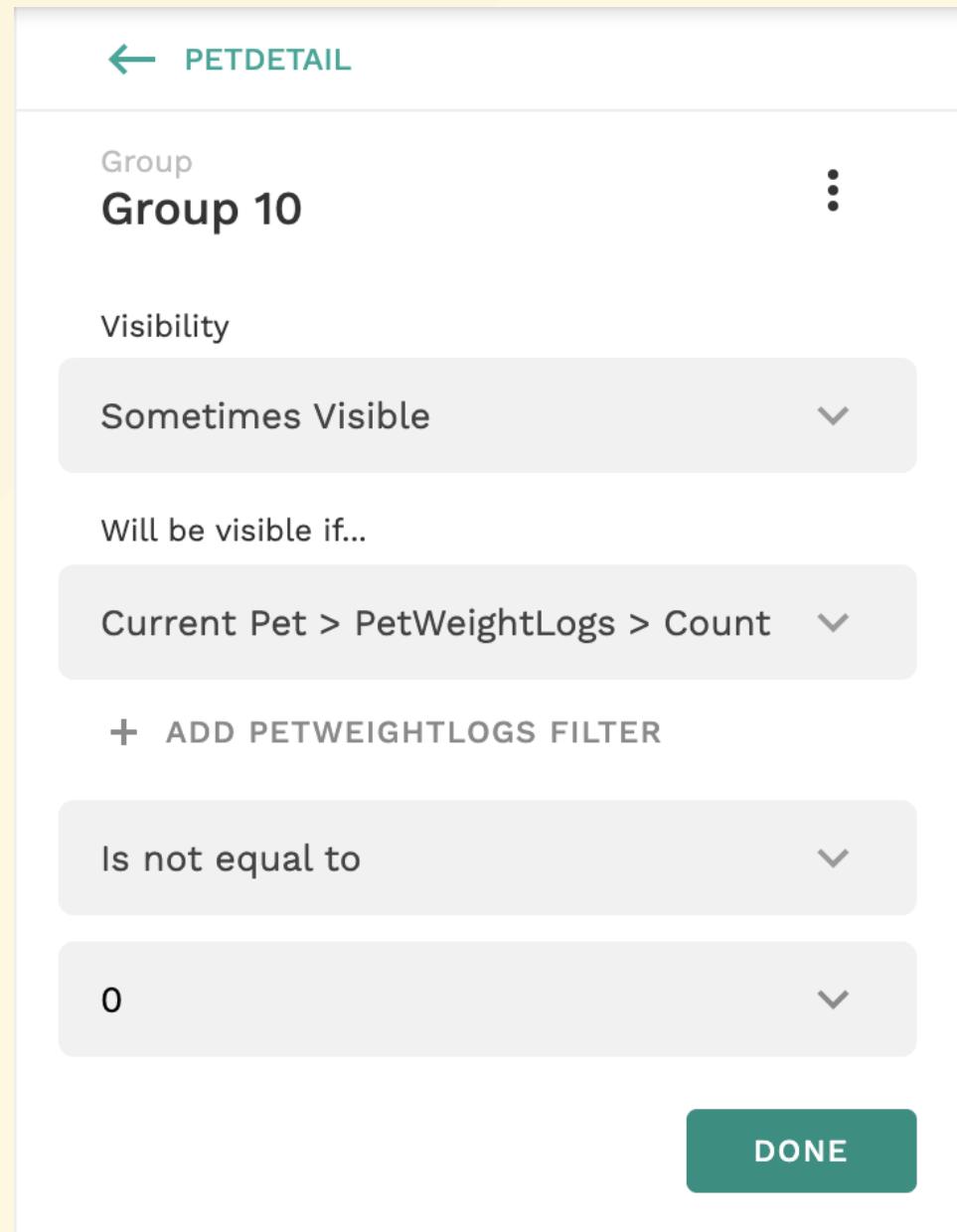
- Select the label "Latest Weight" and its value, and group them together



- Select Change Visibility

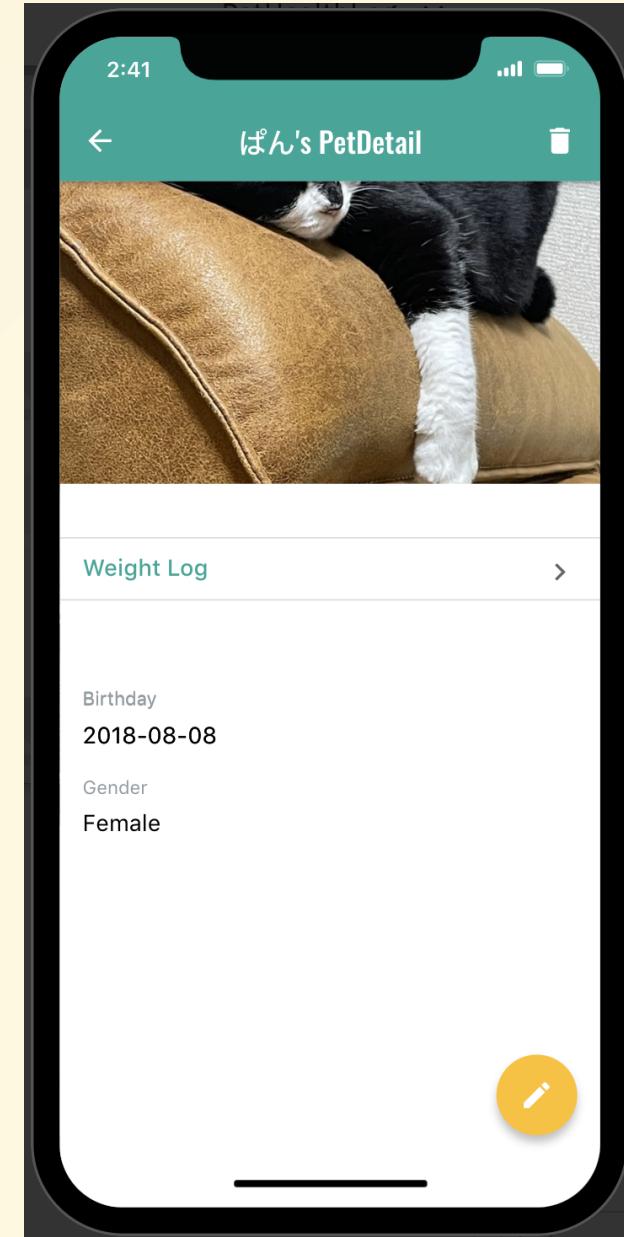


- Change Visibility to "Sometimes Visible"
- Set "Will be visible if..." as "Current Pet" > "PetWeightLogs" > "Count"
- Set "Is not equal to 0"



Checking with the Preview function, if the weight is unregistered, the Latest Weight is now hidden.

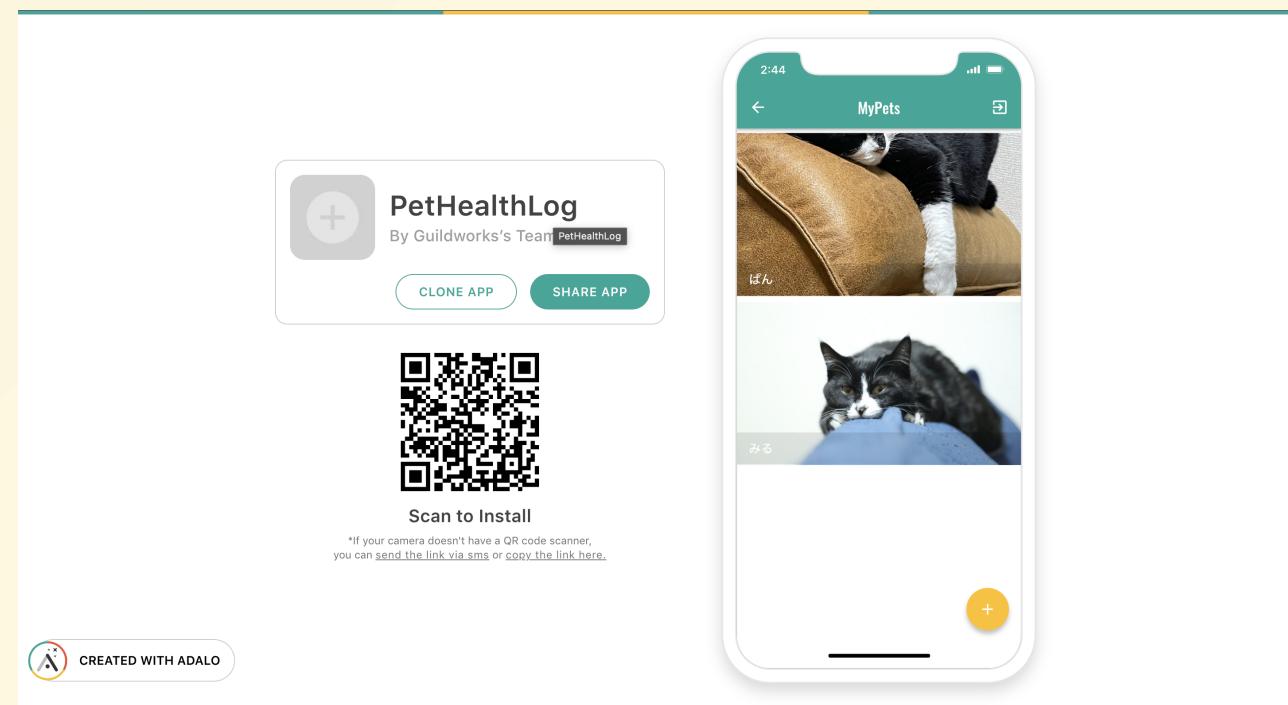
This is the end of the improvement of the sample application.



# URL for cloning

- You can use the following URL to clone the application that reflects the work done so far, and use it to check your answers.

<https://previewer.adalo.com/f1324ea8-ec47-4c22-a3a9-3258044eb754>



# Exercises

Work on this with your Development Phase team members.

1. Develop a team member management app with following features.

- Registration of team members
- Display of member list
- Display member details
- Update member information
- Delete a member
- Your original functions

2. Develop new app freely (if there is enough time after exercise 1). 112

If no other team members are participating in today's lecture, please join another team.

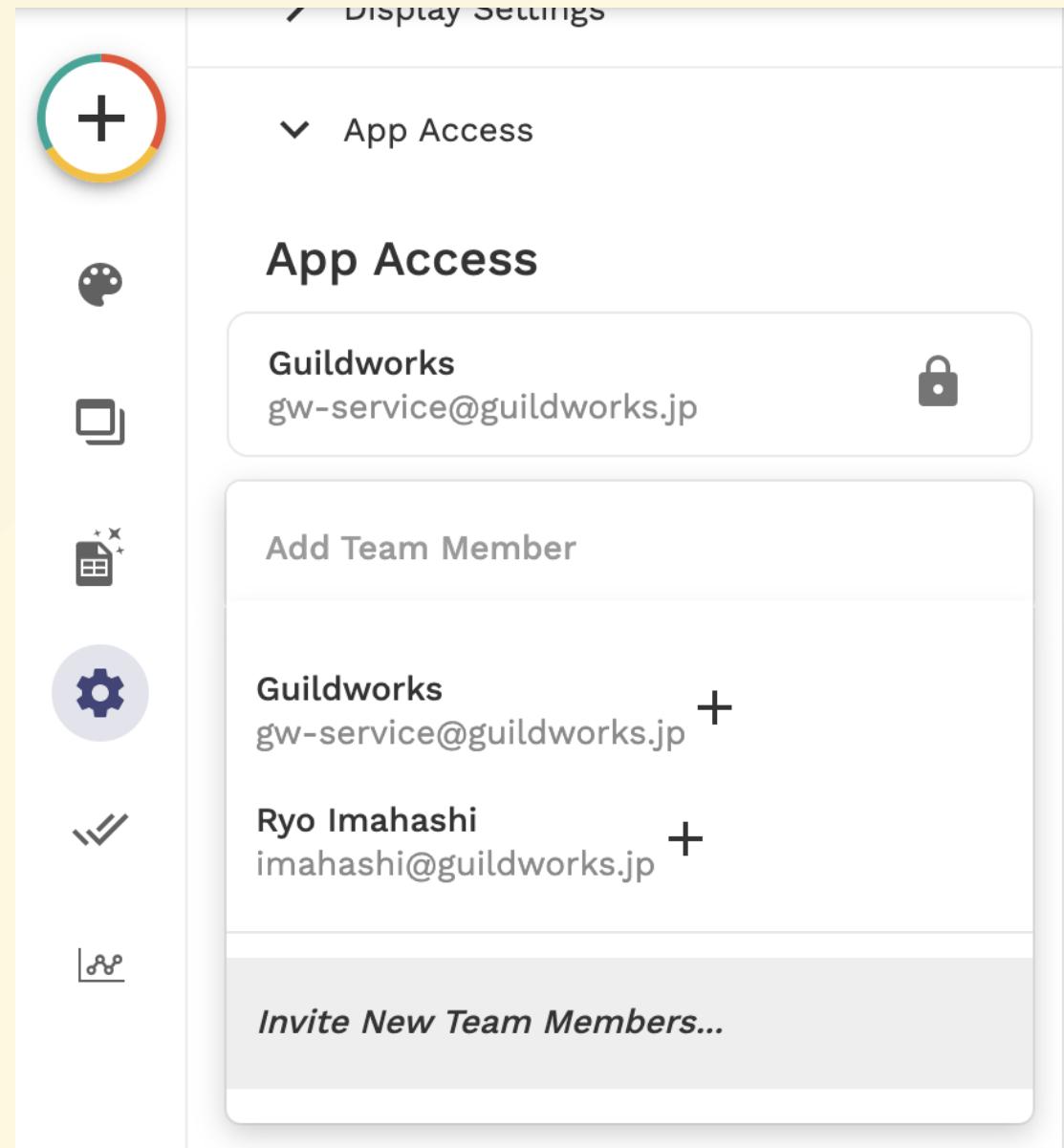
If you are a remote participant, please join the other team members by connecting to Zoom and having a conversation or sharing your screen with them.

At the end of the lecture, all teams will be asked to make a presentation.

When the app is ready to use, share the URL on Slack for everyone to see.

# Additional information about the exercise

- To co-edit one app with a team member, select Settings > AppAccess > Add Team Member > Invite New Team Member and enter the team member's email address.



When editing one application at the same time with other members, the edits made by others will not be reflected on your screen in real time. (Reloading seems to be necessary).

When multiple people edit the same screen at the same time, the edits made earlier are overwritten by the edits made later. Each person should edit a different screen, or edit together while viewing the same computer.

## Reference

- There are a number of Adalo apps available that can be cloned, so you may want to look for one that is similar to what you want to do.
  - App Templates  
<https://www.adalo.com/app-templates>
  - UI & Functional Kits  
<https://www.adalo.com/cloneable-kits>

## Examples of apps that can be cloned

- Event Calendar <https://www.adalo.com/cloneables/event-calendar>
- Swipe-to-answer app quiz <https://www.adalo.com/cloneables/quiz-app>
- SNS follow function <https://www.adalo.com/cloneables/follow-function>
- Facebook clone <https://www.adalo.com/cloneables/facebook-clone>
- Blog app <https://www.adalo.com/cloneables/minimal-blog-app>
- Ecommerce app <https://www.adalo.com/cloneables/ecommerce-app>

## Reference

- If there is something you cannot achieve with Adalo alone, you may be able to achieve it by integrating with external services.
- If you use a service called Zapier, you can easily integrate Adalo with external services by following the instructions. If you are interested, please give it a try.

<https://zapier.com/apps/adalo/integrations>

## Examples of external services that can be linked to Adalo with Zapier

- Google Spreadsheet
- Google Calender
- Slack
- Zoom
- Twitter
- Instagram
- Spotify
- Bubble
- Google Meet
- Strava

## Presentation of the exercise results

Each team should present the application that they made in the exercise.

# Summary

- In this lecture, we designed a database and made it possible to perform CRUD operations on the database from the application.
- We also introduced some of Adalo's features while improving the app.
- I have not yet introduced the following Adalo features related to integration with external services, so I would like to introduce them again when I have time in the 5th lecture or Development Phase.
  - Custom Action(Calling APIs of external services from Adalo)
  - External Collections(function to handle data acquired from APIs of external services as Adalo's collections)

- Based on what we've learned so far, I think it's a good idea for teams to think about whether the application they want to create in the Development Phase can be realized with Adalo.
- Next time, I will give a lecture on Bubble, a no-code tool. Look forward to it!

**That's all!**

**Thank you for your hard work!**