

Programming Boot Camp

Adaloでのデータベースの設計と操作

東京工業大学 2021/11/6

Ryo Imahashi

目次

- 前回のふりかえりと今回のゴールの確認
- データベースについて学ぼう
- データベース設計
- データベース操作
- サンプルアプリを改善しよう
- 演習
- まとめ

前回のふりかえりと今回のゴールの確認

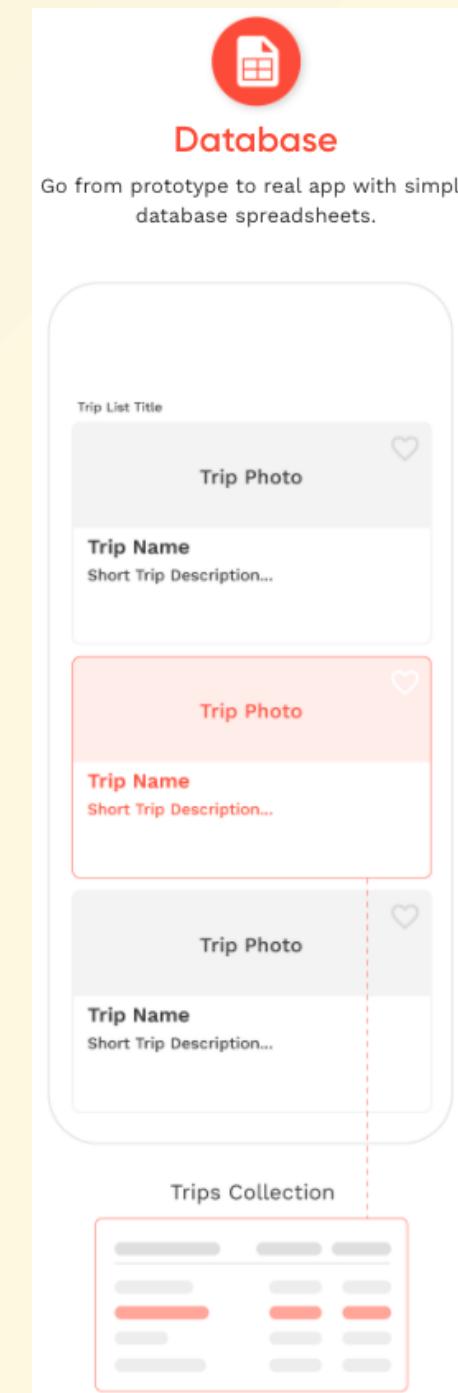
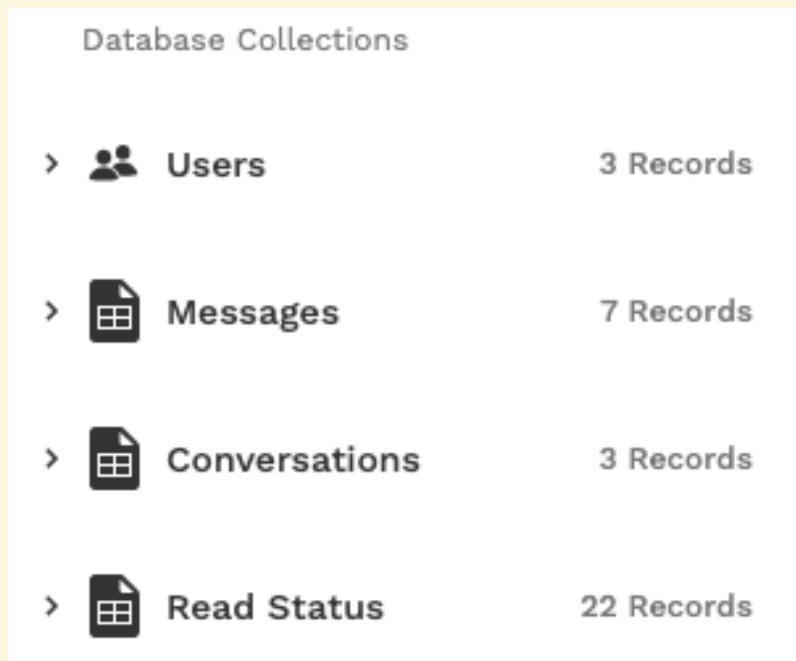
- 前回のレクチャーでは、ノーコードツールのAdaloについて紹介し、ペットの健康管理アプリを題材にアプリのUIを作成しました。
 - レクチャーでは、データベースを必要としない、シンプルなコンポーネントを使いました。(演習の中でデータベースを使った人もいるかもしれません)
- 今回のレクチャーでは、引き続きAdaloを使って、前回作ったUIに合わせたデータベースを設計し、アプリからデータを操作できるようにしていきましょう。
- その後はサンプルアプリの改善をしながらAdaloの機能を紹介し、最後にチームでのアプリ開発演習と発表をしていただきます。

データベースについて学ぼう

まず、これから扱うデータベースがどのようなものかを確認します。

Database(前回の復習)

- 整理されたデータの集合。
- データの登録、読み込み(表示)、更新、削除が行われる。
- 例: Chatアプリの場合



- データベースはよく「表計算ソフトのようなもの」と例えられます。
- データベースを使ってデータを作成(CREATE)、読み取り(READ)、更新(UPDATE)、削除(DELETE)することができます。これらの操作を総称してCRUDと呼びます。

Users						+ ADD USER				
	Email	Password	Username	Full Name						
<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						

Adaloのデータベースの基本



このアイコンからAdaloのデータベースにアクセスできます。
Adaloのデータベースの構成要素は、以下の3つです。

- Collection
- Property
- Record

Collectionとは

同じ属性(Property)を持ったデータの集まり

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:

Header: Users

Actions: + ADD USER, Up/Down arrows, Cloud icon, Refresh icon, Search icon

Table Headers: Email, Password, Username, Full Name, Profile Picture

Data Rows:

	Email	Password	Username	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 陵	

Bottom Right: DONE

- Collectionは、データベースで扱う様々なデータをデータの種類ごとに分割し、整理するためのものです。(類似の言葉として、テーブルがあります)
 - ユーザーが1度の操作で登録や更新、削除といった操作をするデータのまとめがCollectionになることが多いです。
 - デフォルトでは、UsersがCollectionとして用意されており、それ以外は開発するアプリに合わせて追加していきます。
- * どのようなCollectionを追加するかを決めるのはとても難しいです。練習しながら慣れていきましょう。(悩んだ時は、メンター陣に相談するのもおすすめです)

Recordとは

- Recordは、Collection内へ情報を保存する際の単位です。
 - 画像の1行が1つのRecordとなります。
- Users Collectionの例では、1人のユーザーが持つ情報をまとめて1Recordとして登録します。

The screenshot shows the MongoDB Compass interface with the 'Users' collection selected. The top navigation bar includes a red '+ ADD USER' button and several icons for file operations and search. The main area displays a table with four columns: Email, Password, Username, and Full Name. There are also columns for selecting rows and viewing profile pictures. The table contains four user records:

	Email	Password	Username	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	

A large yellow rectangular highlight covers the entire table area. At the bottom right of the table is a red 'DONE' button.

- Recordは基本的にアプリの画面上のフォームから登録できるようになりますが、Recordの表示中に右上の「+Add xxxx」ボタンを押して、右の画像のようなフォームから登録することも可能です。
- Collection内のRecordの検索や、CSVファイルのアップロード(インポート)・ダウンロードも可能です。

New User

Email

Password

Username

Full Name

Profile Picture

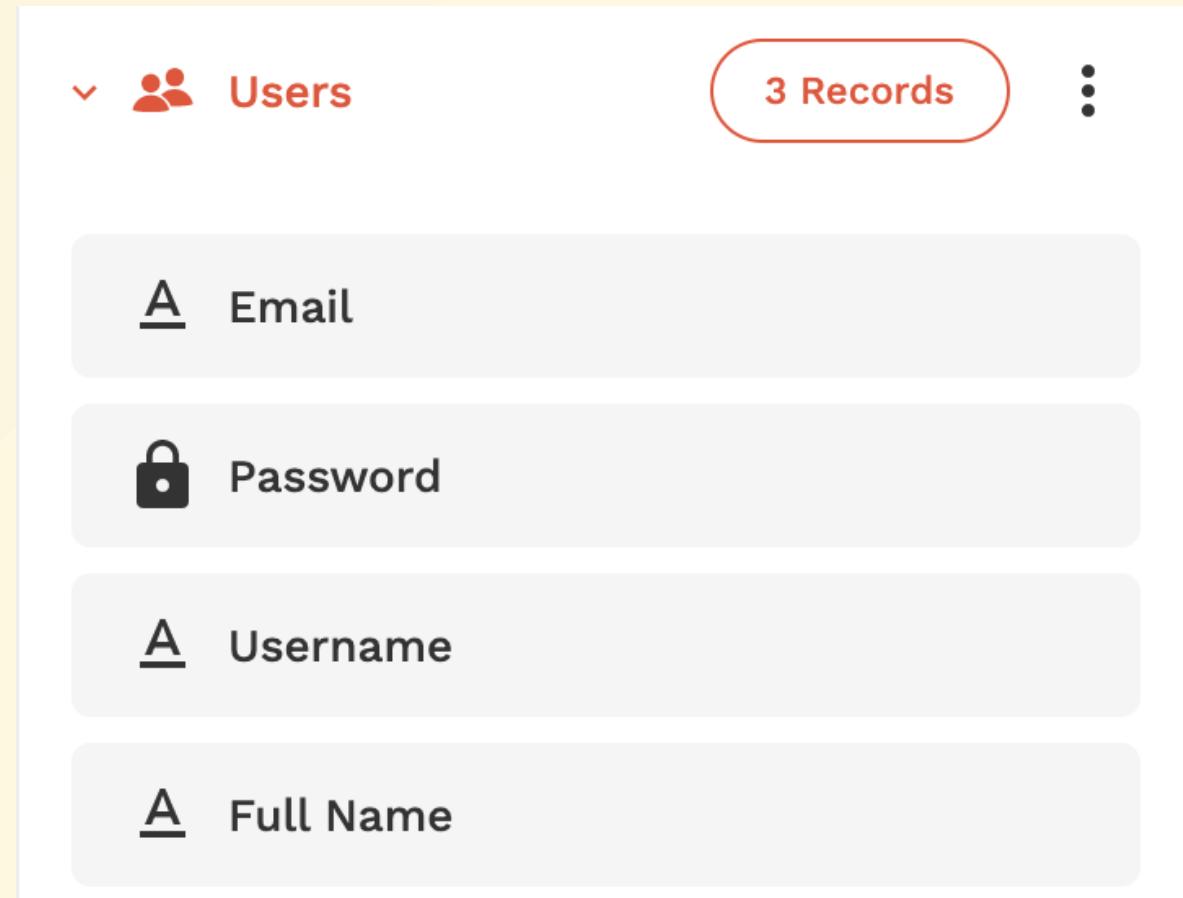
Choose image

CANCEL

SAVE

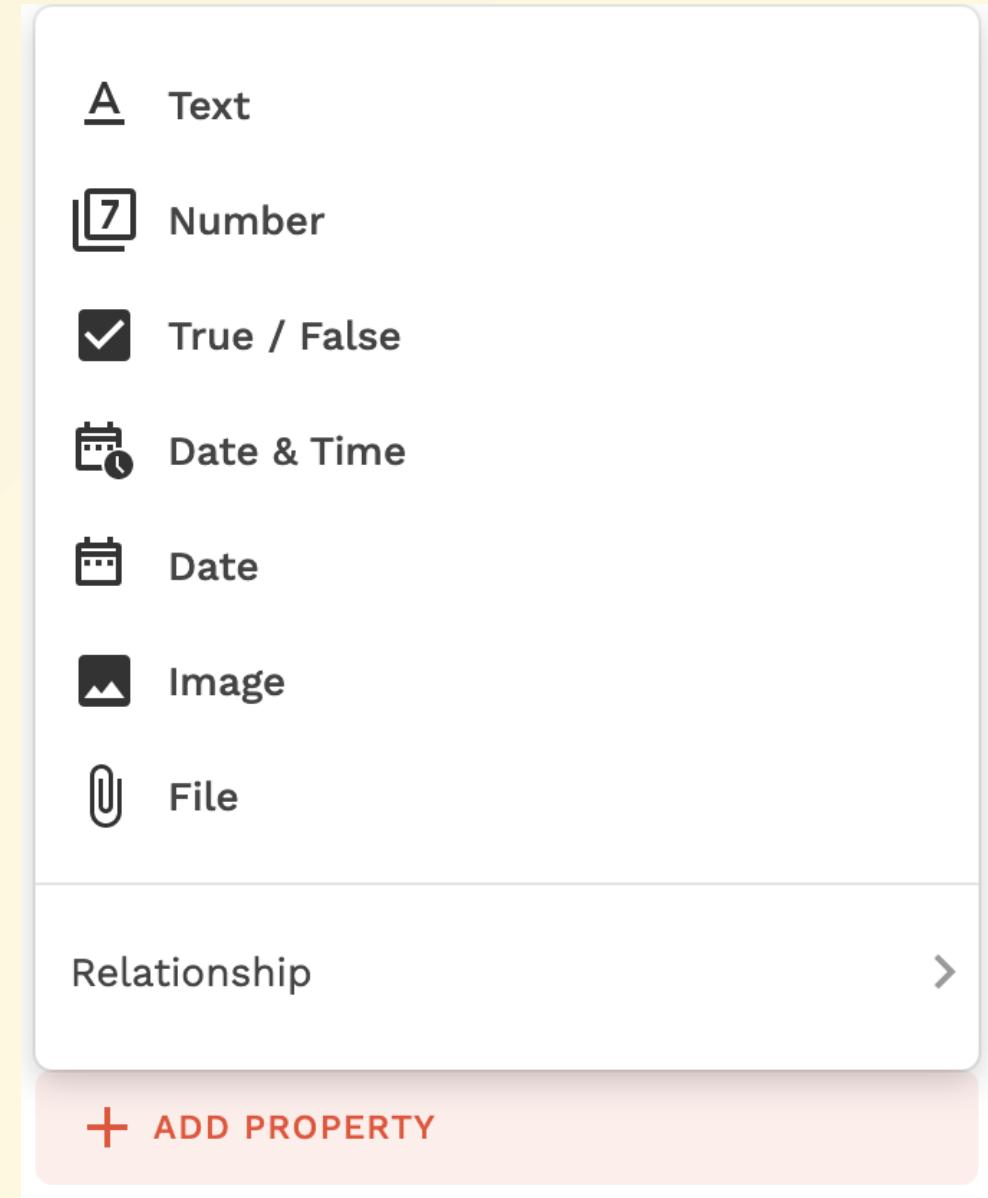
Propertyとは

- Propertyは、Recordを構成する一つ一つの項目です。
- Users Collectionは、Email、Password、用户名、氏名といったPropertyで構成されます。
- Propertyの値は空で登録される場合もあります。



Propertyがどのようなデータかを定義するため、Property追加時はTypeを選択します。

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



Relationshipとは

- 1つのRecordに対して多数のPropertyを保存する代わりに、Relationshipと呼ばれる複数のCollectionを関連づけるための特別なPropertyを設定します。これにより、Collectionを扱いやすい形に分割することができます。

- 例えば、Chatアプリでユーザーが送信したメッセージはUsers Collectionとは別のMessages Collectionに保存され、これら2つのCollectionはRelationshipで関連づけられます。
 - Users側にはMessagesというRelationshipが、Messages側にはSenderという(Usersとの)Relationshipが登録されています。

The screenshot displays a database interface with two main sections: 'Users' and 'Messages'.

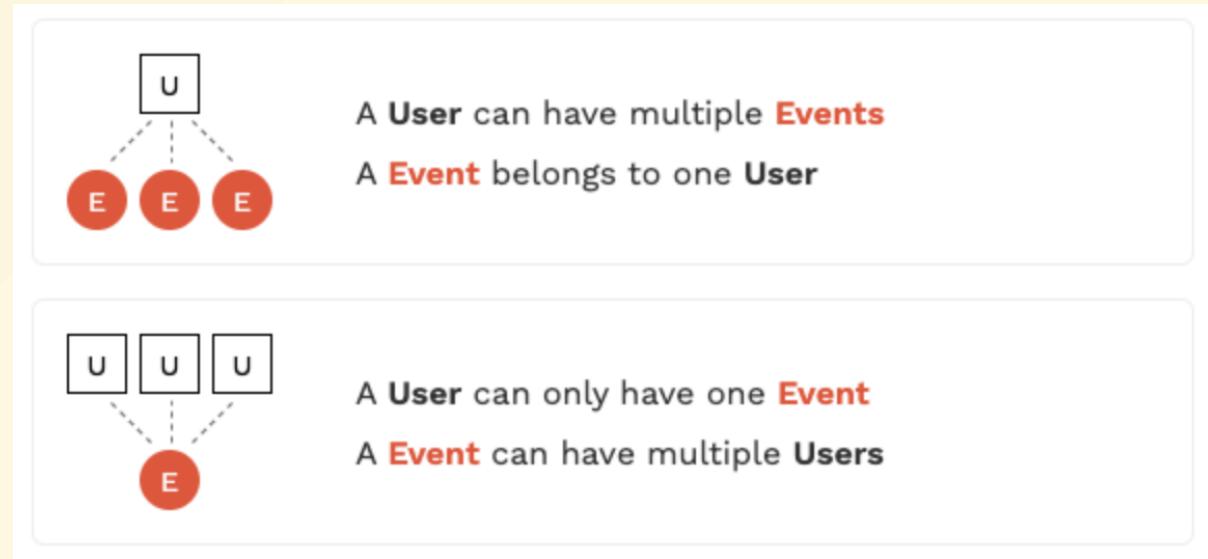
- Users Collection:** Contains 9 records. It includes fields: Email, Password, Username, and Full Name. A modal window is open, showing a relationship named "Messages" pointing to the "User" field in the "Users" collection.
- Messages Collection:** Contains 12 records. It includes fields: Message Text, Sender, and Type (User).

Relationshipの種類

- AdaloのRelationshipでは、Collection間で紐付けられるRecordの数に応じて、1対多と多対多という2つの種類のいずれかを選択します。

1対多のRelationship

- 1つのRecordが、別のコレクションにある複数のRecordと関係を持つことを意味します。
- Relationshipを設定しようとしているCollectionを1と多のどちらにするかで、2種類の選択肢が現れます。



1対多のRelationshipの例

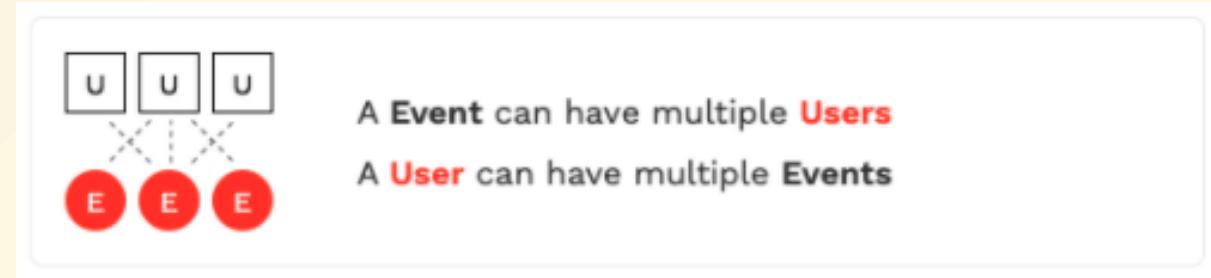
- Chatアプリでは1人のユーザーが複数のメッセージを送信しますが、メッセージの送信者は1人のユーザーなので、Users CollectionとMessages CollectionのRelationshipは1対多になります。

Users					
	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					
	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"/>	ヰ——(∀)——!!	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

多対多のRelationship

- 両方のCollectionの1Recordが、もう一方のCollectionの複数のRecordに紐付けされることを意味します。



多対多のRelationshipの例

- Chatアプリでは1人のユーザーが複数の会話(誰とどんなメッセージをやりとりしたかを管理するもの)を持ち、1つの会話には複数のメンバー(ユーザー)が所属するので、Users CollectionとConversations CollectionのRelationshipは多対多になります。

Users					
	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	

Conversations					
	Name	Members	Messages	Read Statuses	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

データベース設計

前回のレクチャーで作成したサンプルアプリのUIを見ながら、保存が必要なデータを考えて、データベースを設計しましょう。

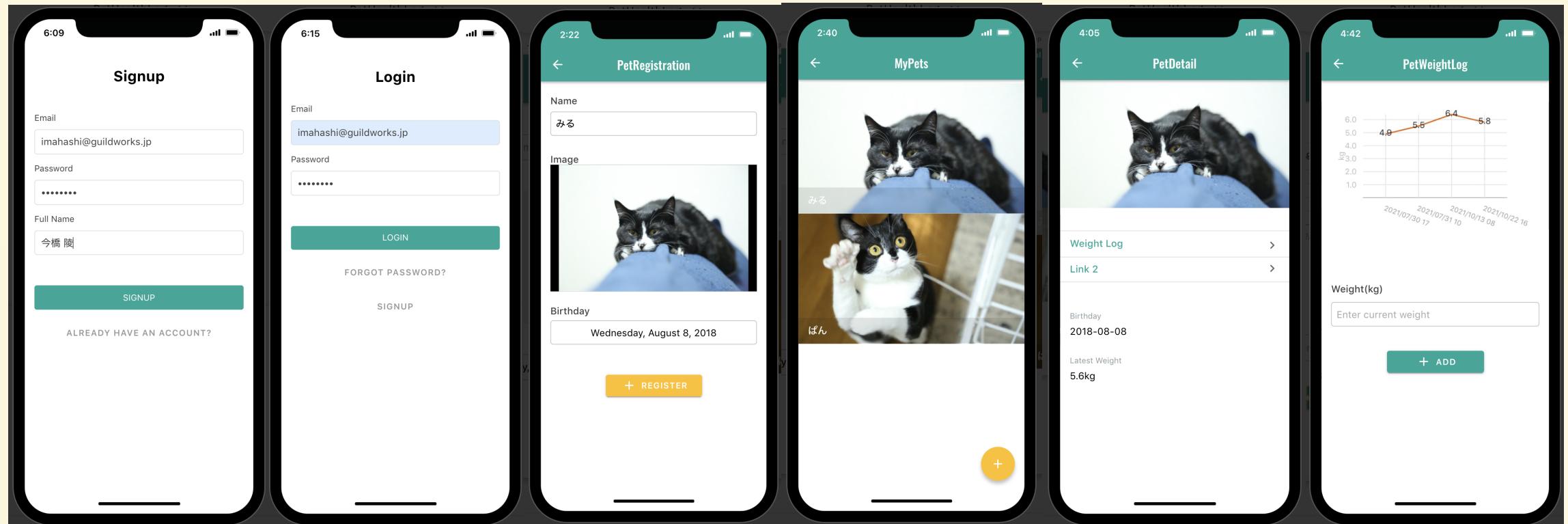
前回のレクチャーで作成したアプリのクローン用URL

- 以下のURLからアプリをクローンしてください。それを使ってここからのレクチャーを進めます。

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

データベースを設計してみよう

サンプルアプリのUIを見ながら、データベースを設計してみましょう。手順は次のページで紹介します。



データベース設計の手順

1. UIを見ながら、保存が必要になるデータをリストアップしましょう。テキストエディタ(メモ帳アプリ等)に書き起こしてください。
2. リストアップしたデータがどのようなCollectionに分類できるかを考えて、Adaloのデータベースに必要なCollectionを作成しましょう。
3. 1でリストアップしたデータを適切なCollectionにPropertyとして追加してください。その際、適切なTypeを選択してください。
4. 他のCollectionと関連を持つCollectionには、Relationship Propertyを設定しましょう。

* 次のスライド以降に解説がありますが、答えを見る前に一度自分で手を動かして考えてみることをおすすめします。

* 絶対的な正解はありません。悩んだら、直感に従って進めてみてください。

解説

UIを見ながら、保存が必要になるデータをリストアップすると、以下のようになりました。

- ユーザーのEmail
- ユーザーのパスワード
- ユーザーのFullName
- ペットの名前
- ペットの写真
- ペットの誕生日
- ペットの体重
- ペットの体重の登録日時

- その他のデータを挙げられた人がいれば、教えてください！

リストアップしたデータがどのようなCollectionに分類できるかを考えて、今回はこの3つに分類することにします。

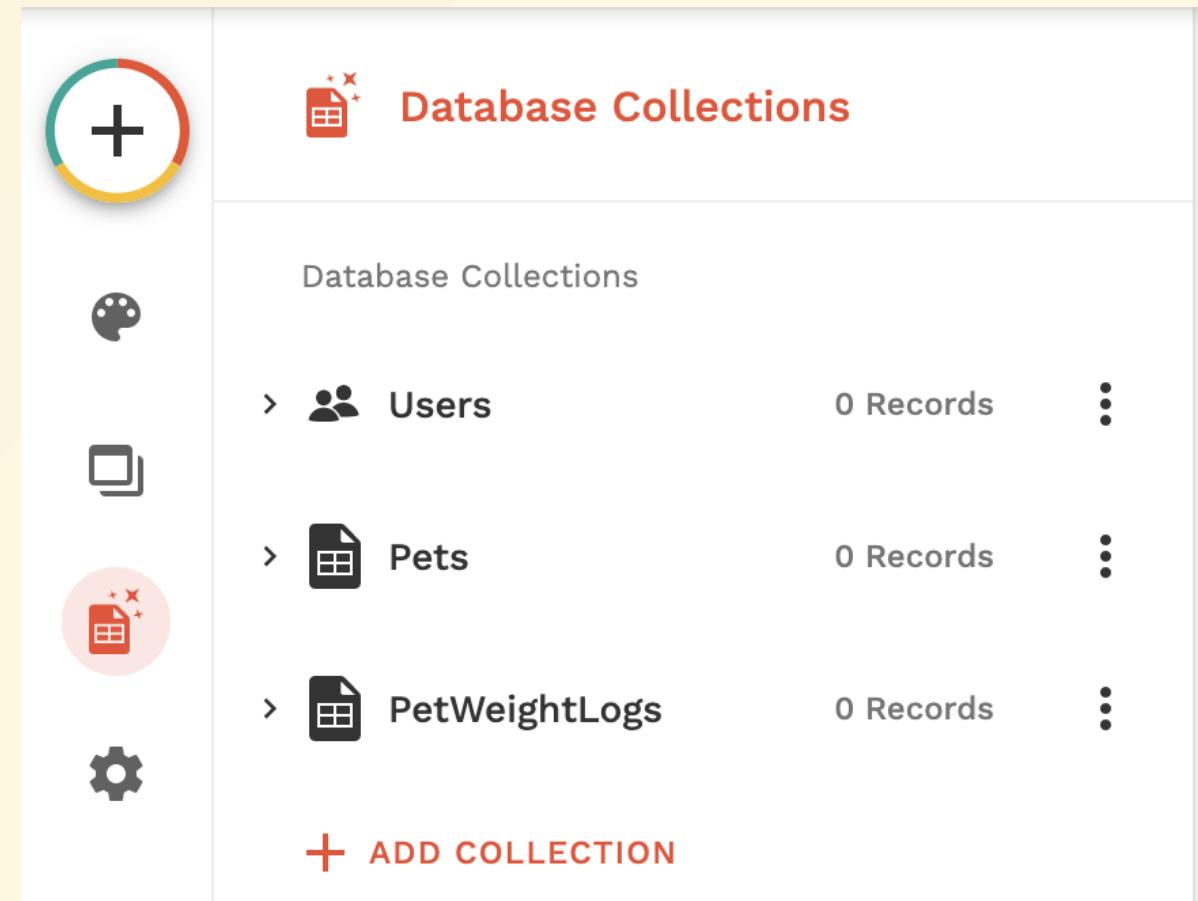
- Users
- Pets
- PetWeightLogs

- ユーザーのCollectionとペットのCollectionの2つを用意した人は多いのではないでしょうか？
- ペットの体重記録のCollectionは用意しなかった人もいるかもしれません。(ペットの体重とその登録日時をペットのCollectionに含める方法も間違いではありません。後ほど解説します。)
- その他のCollectionの分類をした人はいますか？

Collectionの分類に関する補足

- 「Aが決まればBが1つに決まる」という関係が成り立つ時、AをCollectionに、BをそのCollectionのPropertyにする場合が多いです。
 - ユーザーが決まれば、ユーザーのEmail、パスワード、FullNameがそれぞれ1つに決まります。
 - ペットが決まれば、ペットの名前、写真、誕生日がそれぞれ1つに決まります。
- 「Aに対してBが複数存在する」という関係が成り立つ時、AとBは別々のCollectionに分割することが多いです。
 - (1匹の)ペットに対して、ペットの体重とその登録日時は複数存在します。

- CollectionをAdaloのデータベースに登録しておきます。
- Usersはデフォルトで作成されています。

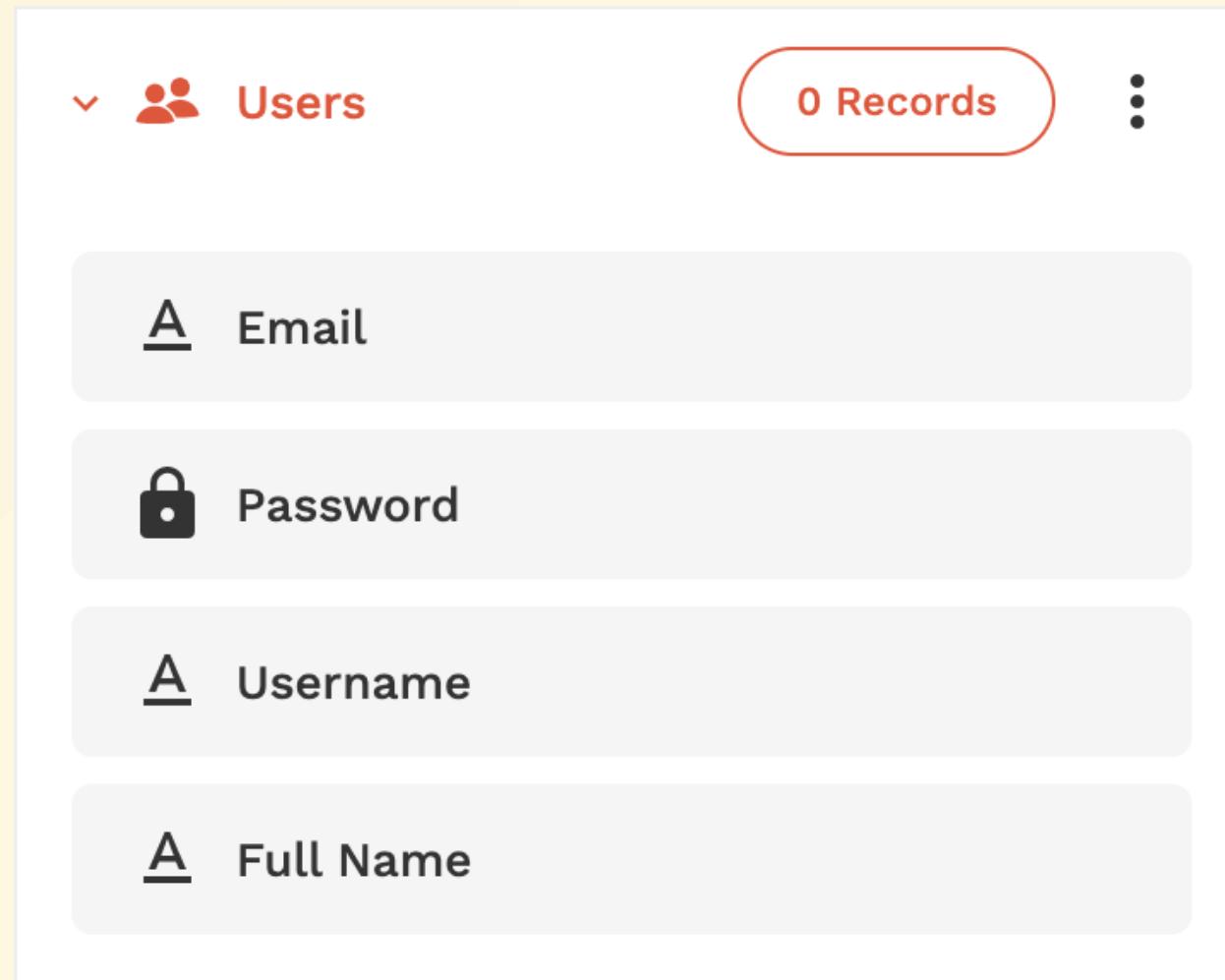


次に、1でリストアップしたデータを適切なCollectionの配下にPropertyとして追記すると、以下のようにになりました。()の中は選択するTypeです。

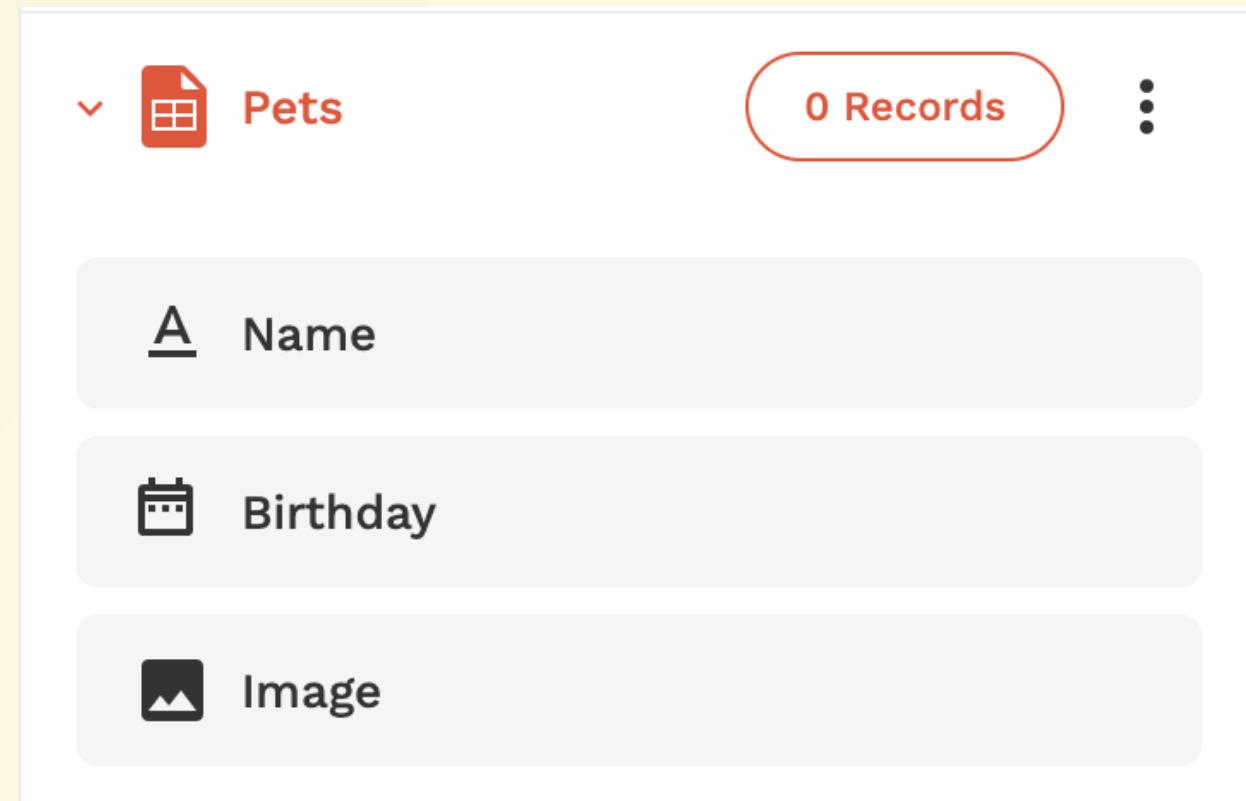
- Users
 - Email(Text)
 - Password(※Password)
 - FullName(Text)
- Pets
 - Name(Text)
 - Image(Image)
 - Birthday(Date)
- PetWeightLogs
 - WeightKg(Number)
 - RegisteredTime(Date&Time)

※ Passwordはデフォルトで設定される特殊なTypeです。

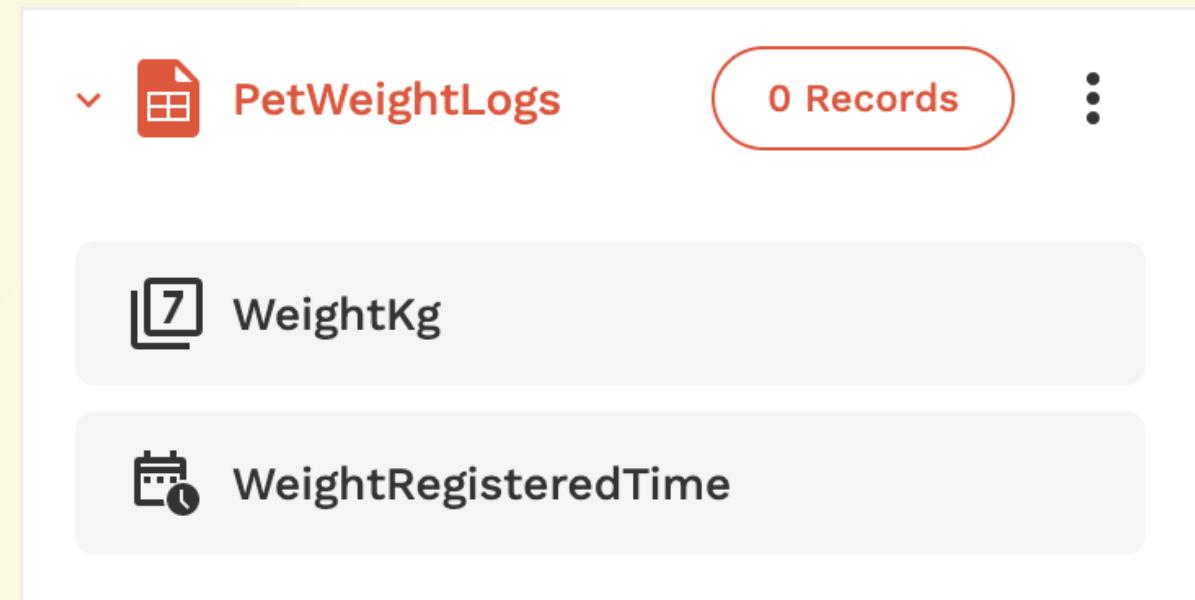
- Adaloで実際にPropertyを追加します。
- Users Collectionはデフォルトで設定済みで、必要な項目は含まれています。
- Usernameは不要ですが、削除できないのでそのままにしておきます。



- Pets CollectionのPropertyはこのようになります。

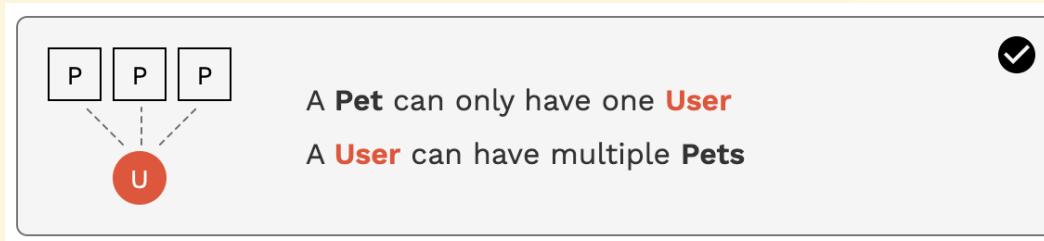


- PetWeightLogs Collection のPropertyはこのようになります。
- Collection追加時にデフォルトで設定されるName Propertyは不要なので、削除します。
 - ドラッグアンドドロップで順番がCollection内の一一番上でなくなるように移動すれば、削除できます。



最後に、他のCollectionと関連を持つCollectionには、Relationship Propertyを設定します。

- Users Collectionを選択して、Pet Collectionとの1対多のRelationshipを追加します。



▼ Users 0 Records ⋮

Email

Password

Username

Full Name

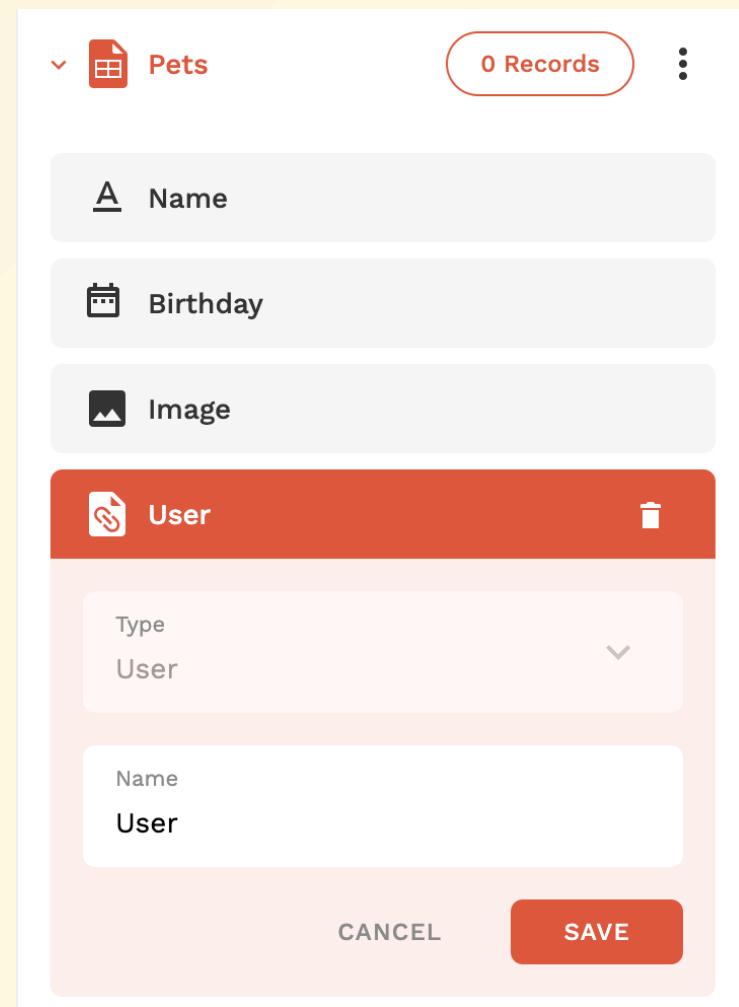
Pets trash icon

Type
Pet

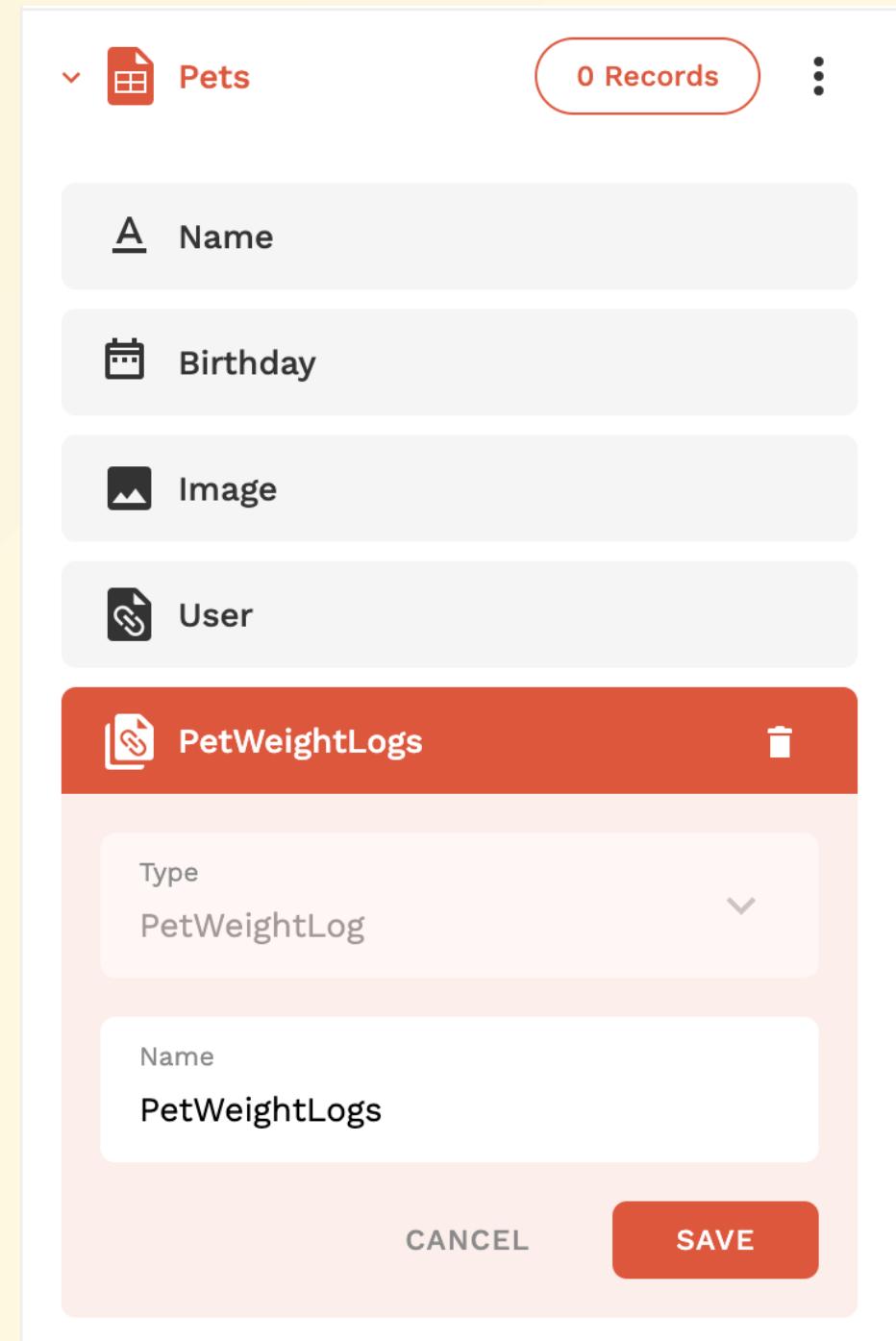
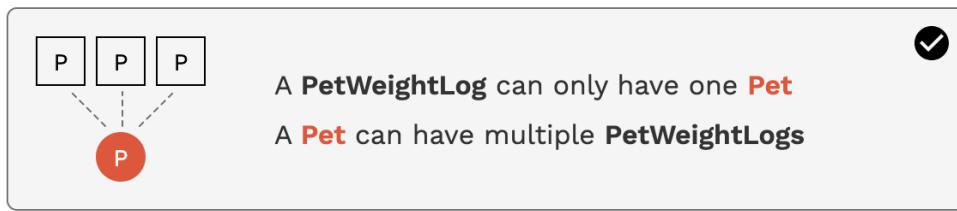
Name
Pets

CANCEL SAVE

- Pets Collectionを確認すると、Users Collection側でRelationshipの設定をしたので、自動でUsers CollectionとのRelationshipが追加されています。
 - Users Collection側が1なので、末尾のsが省略されて、UserというProperty名になっています。

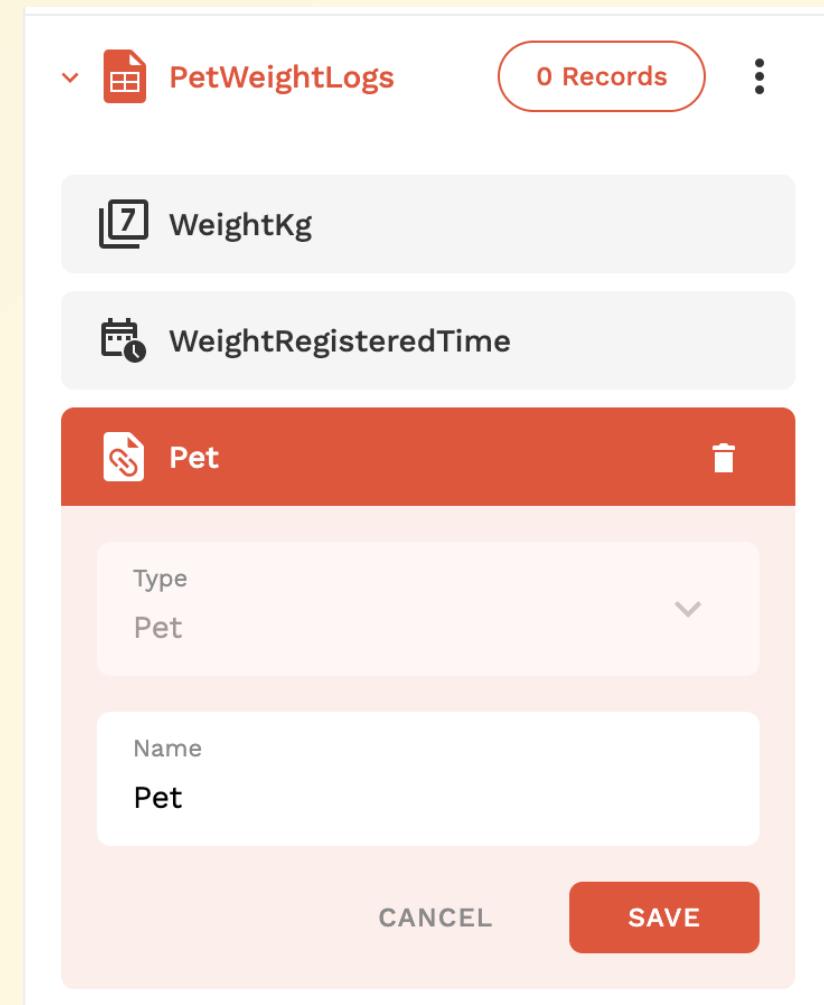


- Pets Collectionに、PetWeightLogs CollectionとのRelationshipを追加します。
 - Pets Collectionを選択して、PetWeightLogs Collectionとの1対多のRelationshipを追加します。



PetWeightLogs Collectionを確認すると、Pets Collection側でRelationshipの設定をしたので、自動でPets CollectionとのRelationshipが追加されています。

- Pets Collection側が1なので、末尾のsが省略されて、PetというProperty名になっています。



参考: Pets Collectionにペットの体重とその登録日時を含めた場合どうなるか

以下のようにレコードが登録されますが、この場合、少し困ることが出てきます。

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

困ること

- 1匹のペットに対して異なるペットの体重とその登録日時が結合されたRecordが複数登録されるため、ペットの情報 (Name,Image,Birthday)が重複して登録されてしまう。
 - 1匹のペットの情報を変更するために、重複して登録された Recordを全て更新しないといけなくなり、処理が複雑になる。
- Adaloには一つのCollectionを選んでそこにRecordを登録するためのフォームを自動生成する便利な機能があるが、データを登録する単位でCollectionが分かれていないので、それが使えなくなる。

データベース設計は以上です。

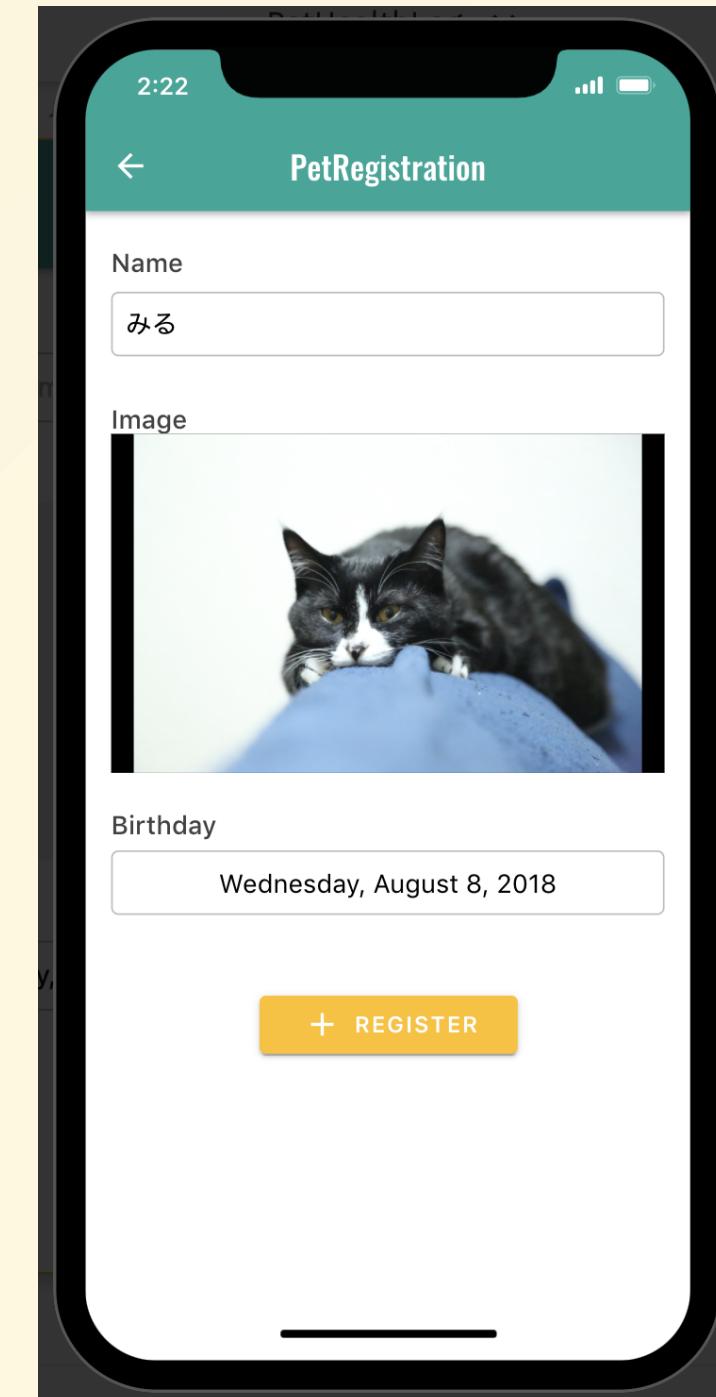
* この後の作業で混乱しないために、Adaloのデータベースを資料と同じ状態に設定しておくことをおすすめします。

データベース操作

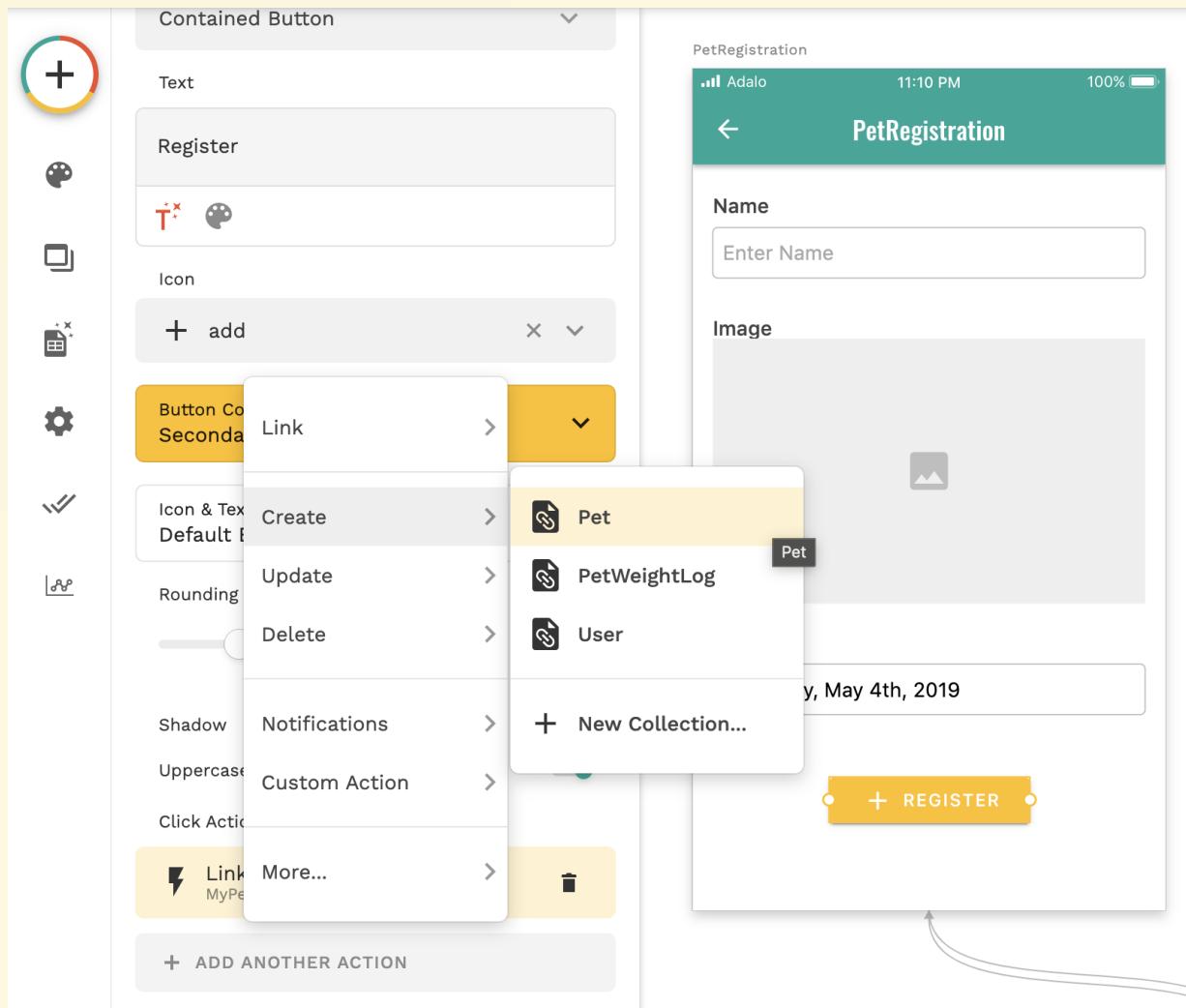
設計したデータベースを使って、サンプルアプリでデータのCRUD操作ができるようにしましょう。

データの作成(CREATE)

まず、作成済みのペット登録画面で実際にペットのレコードを登録できるようにします。

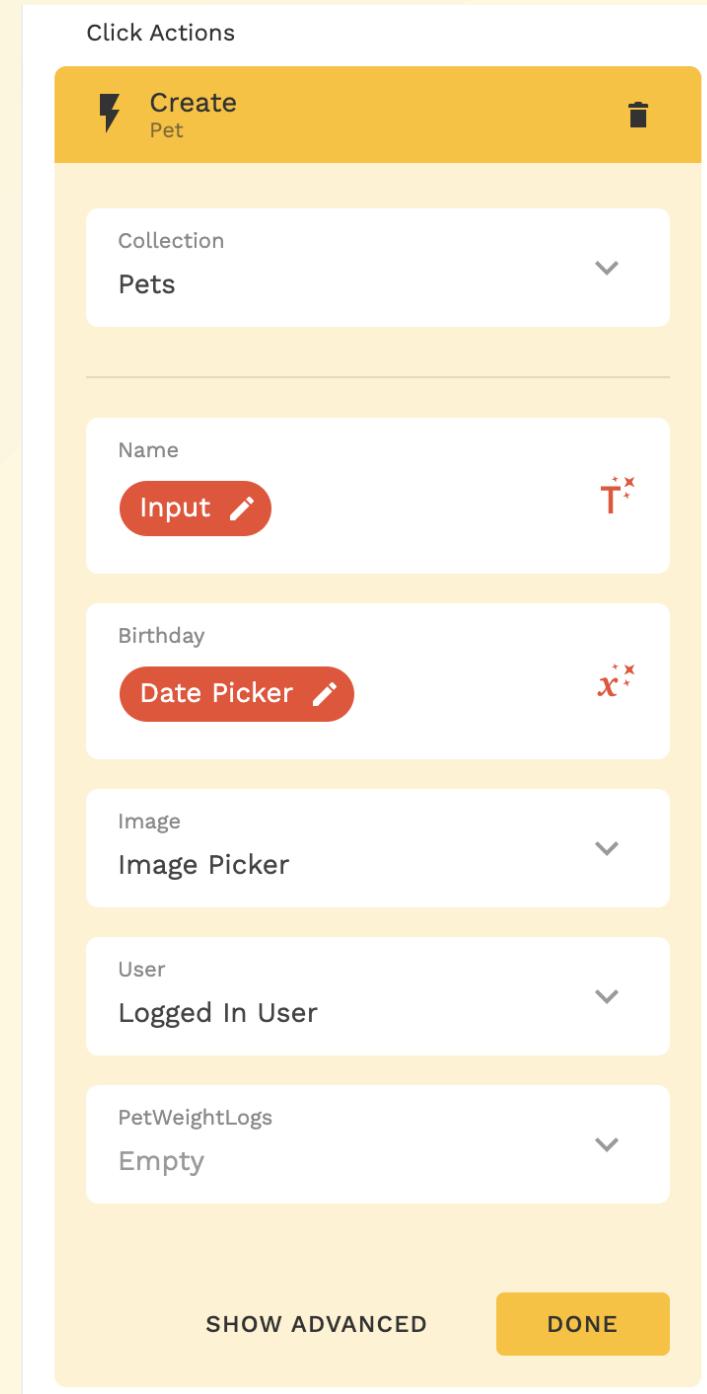


- ペット登録画面のREGISTERボタンを選択し、「ADD ANOTHER ACTION」をクリック
- Create > Pet を選択



以下を入力してDONE。

- NameはOther ComponentsのInputを選択
- BirthdayはOther ComponentsのDate Pickerを選択
- ImageはOther ComponentsのImage Pickerを選択
- UserはLogged In Userを選択
- PetWeightLogsはEmptyのまま(ペット登録時には不要)



Preview機能でペットを登録してみましょう。
Pets CollectionにRecordが登録されたらOKです。

Pets

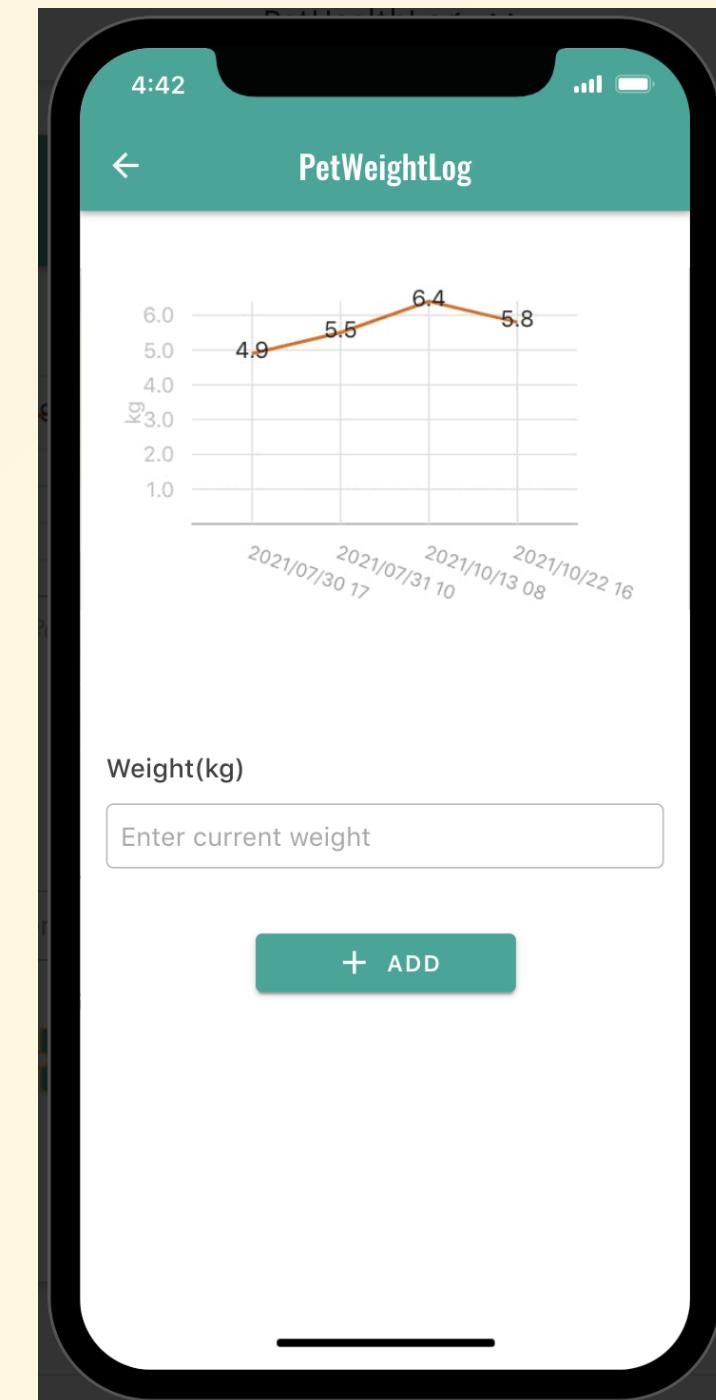
+ ADD PET

↑ ↓ <> 🔎

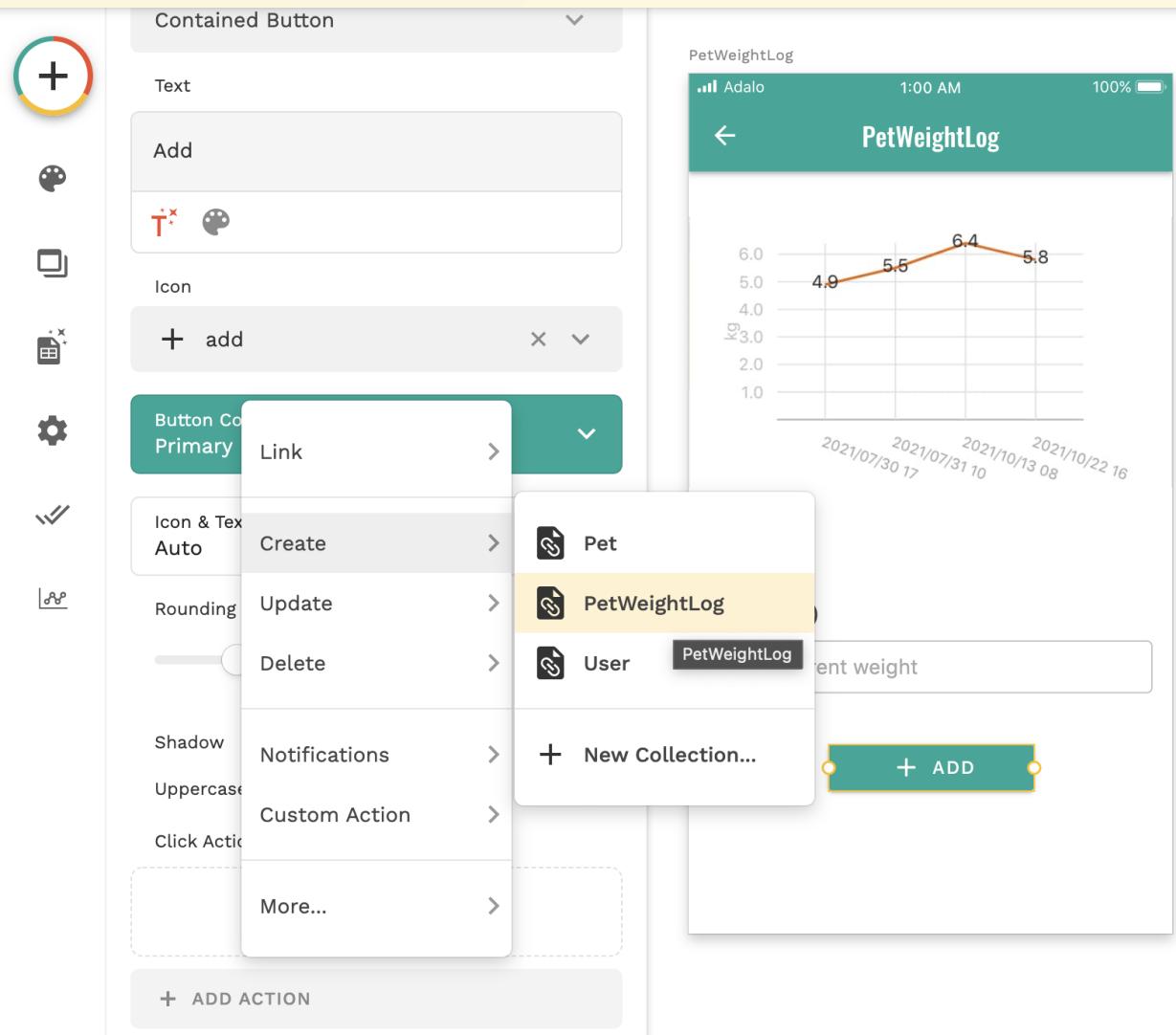
<input type="checkbox"/>	Name	Birthday	Image	User	PetWeightLogs	Created
<input type="checkbox"/>	みる(Database Record)	8/8/2018		imahashi@guildworks.jp		43 minutes ago

DONE

次に、ペットの体重管理画面で現在の体重を登録できるようになります。

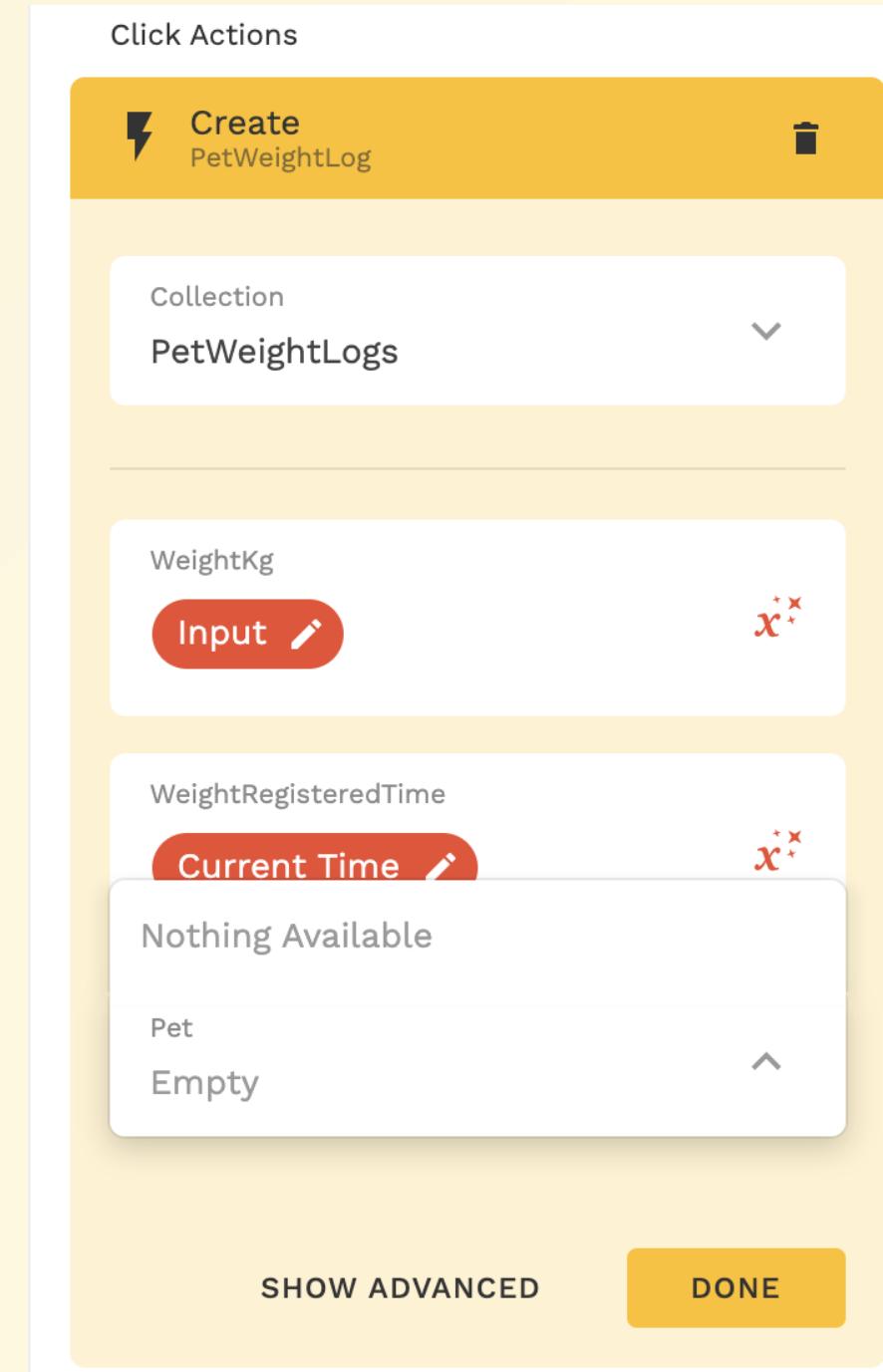


- ペットの体重管理画面で ADDボタンを選択し、「ADD ACTION」をクリック
- Create > PetWeightLog を選択



以下を入力してDONE。

- WeightKgはOther ComponentsのInputを選択
- WeightRegisteredTimeは Date&Time > Current Time を選択
- PetはNothing Availableなので、一旦Emptyのままにする
(後ほど、選択したペットの体重を登録できるように設定します)

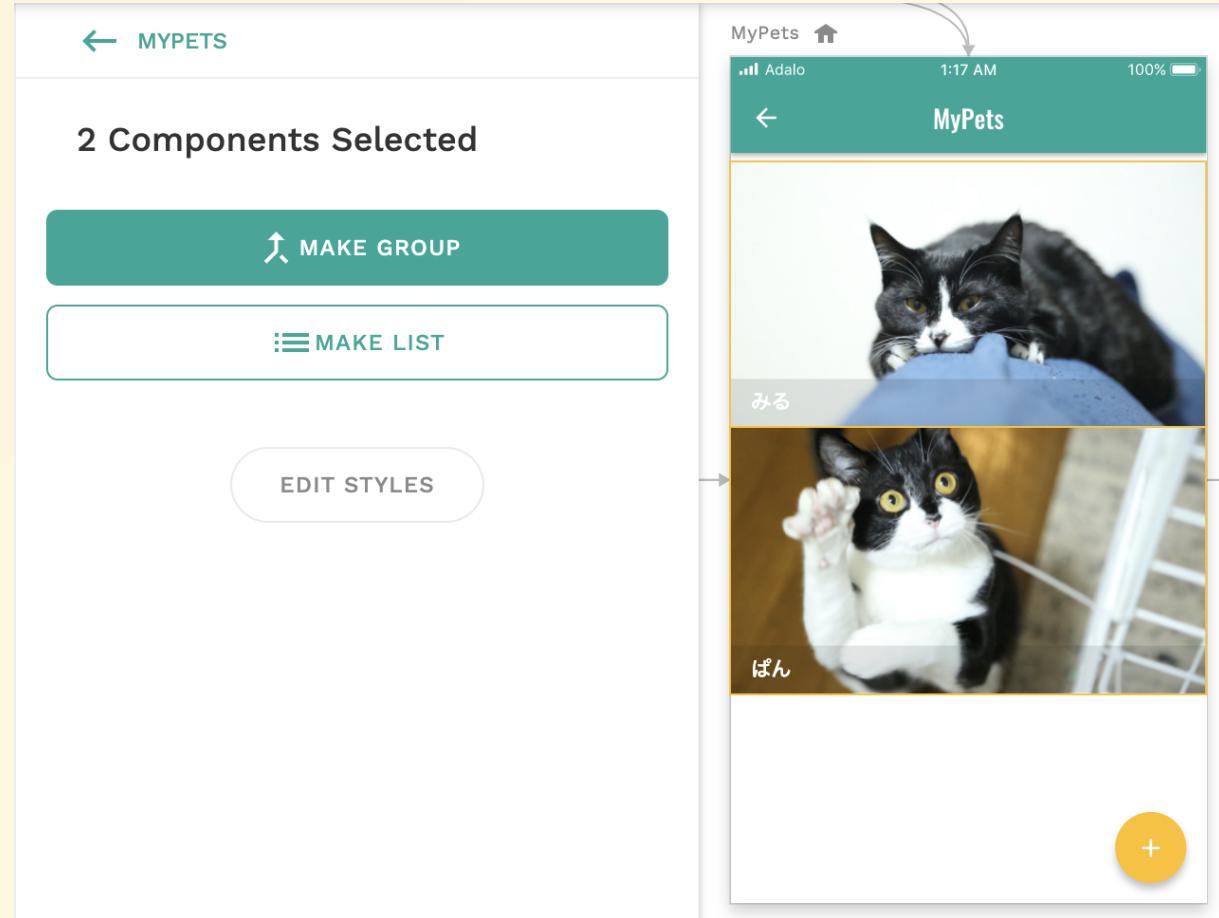


データの表示(READ)

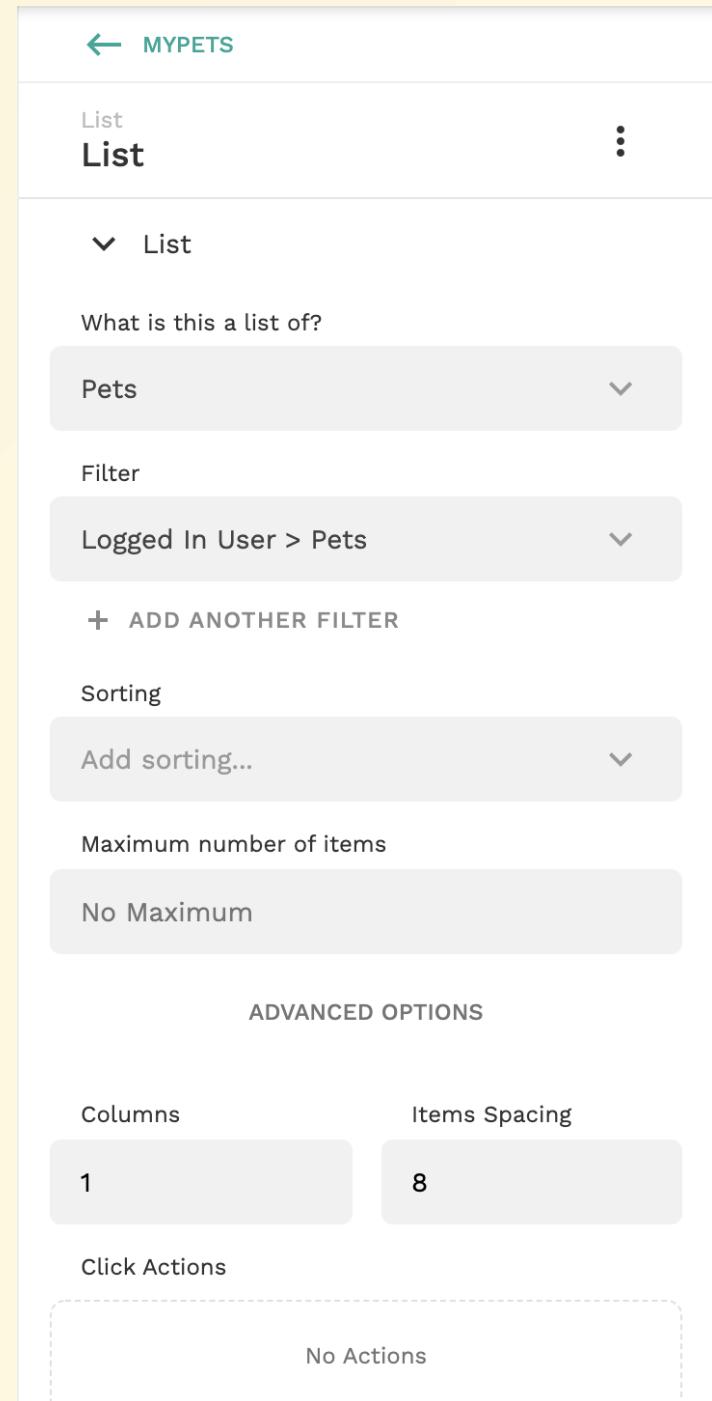
まず、作成済みのペット一覧画面に実際に登録したペットが表示されるようにします。



- ペットの画像と名前を表示している2つのコンポーネントを選択し、MAKE LISTをクリック



- What is this a list of?で Pets を選択
- Filterで Logged In User > Petsを選択



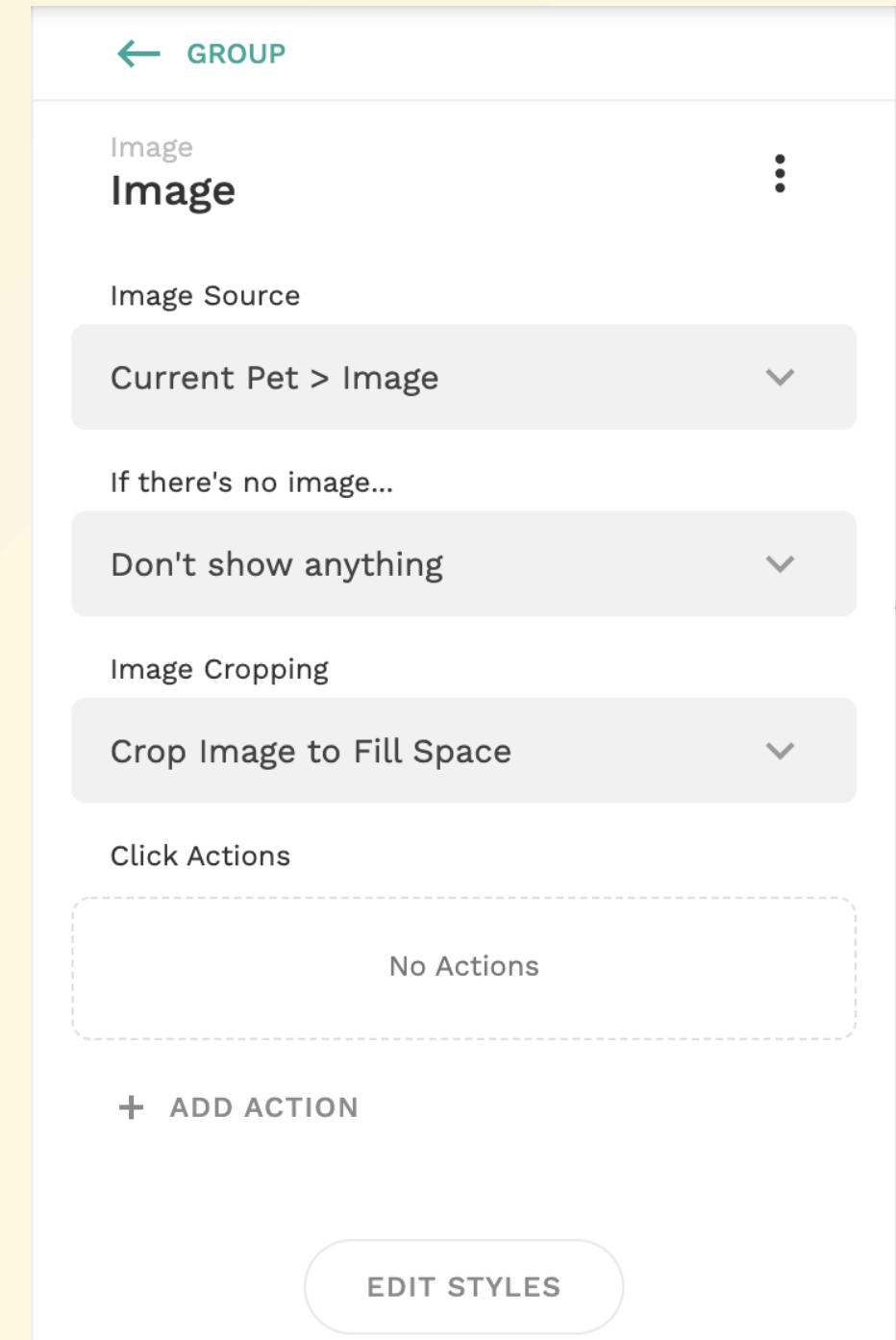
- 作成したListを構成するコンポーネントであるGroupの1つ目をクリック。
- Group内のImageコンポーネント、ペット名のコンポーネントをそれぞれ編集していきます。

The screenshot shows the Figma interface with two main panels:

- Components Panel (Left):** Shows a tree structure under the heading "Components". The first item is a "Group" component labeled "Group". Below it is another "Group" component labeled "Group 3". At the bottom of the list is a button labeled "+ ADD COMPONENT".
- Preview Panel (Right):** Shows a preview of the "List" component. It displays three items: "Image" (with a thumbnail icon), "Group" (with a thumbnail icon), and "Text" (with the text "みる"). Each item has a right-pointing arrow indicating it can be edited.

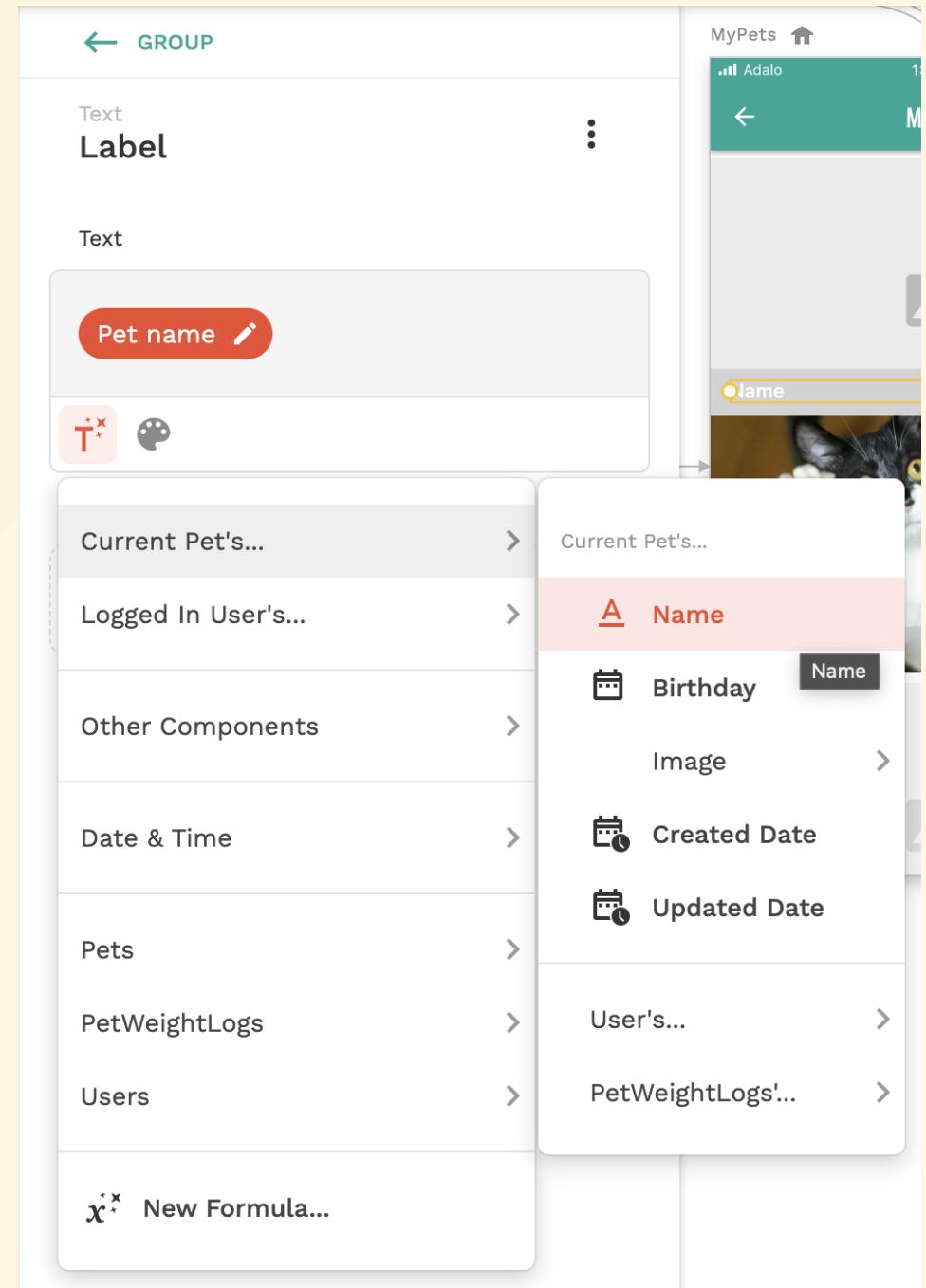
Imageコンポーネントを編集

- Image Sourceで、Database > Current Pet's > Imageを選択
- If there's no image...でDon't show anythingを選択
 - あるいは、Show a place holder imageを選択して好きなペットのシルエット画像を設定してもOKです

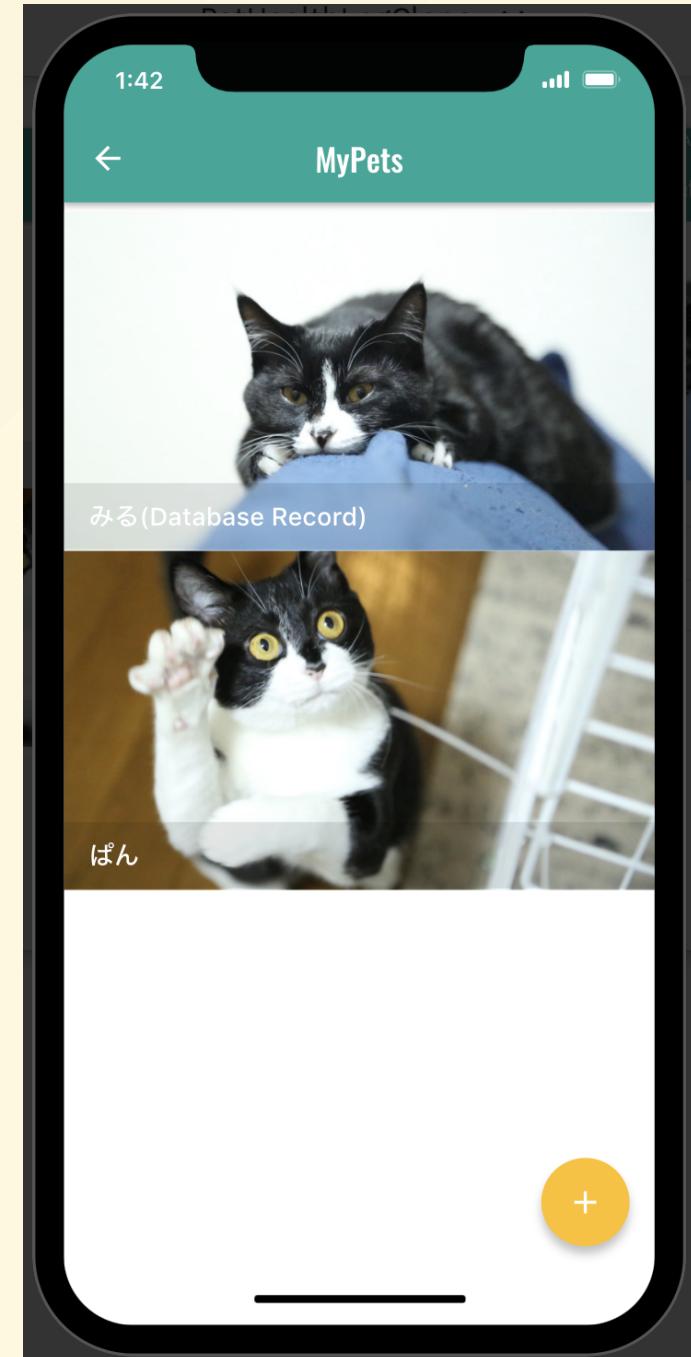


ペット名のコンポーネントを編集

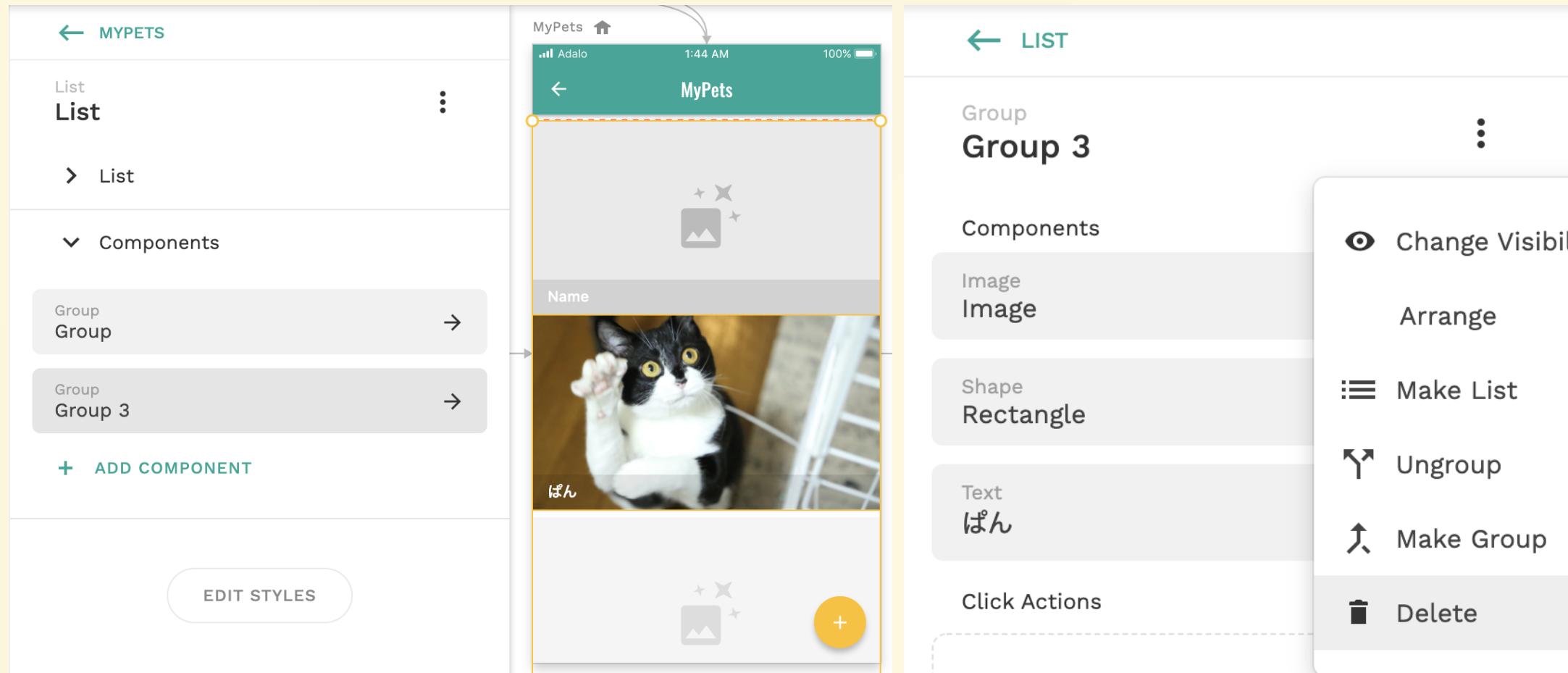
- Add Magic TextでCurrent Pet's > Nameを選択



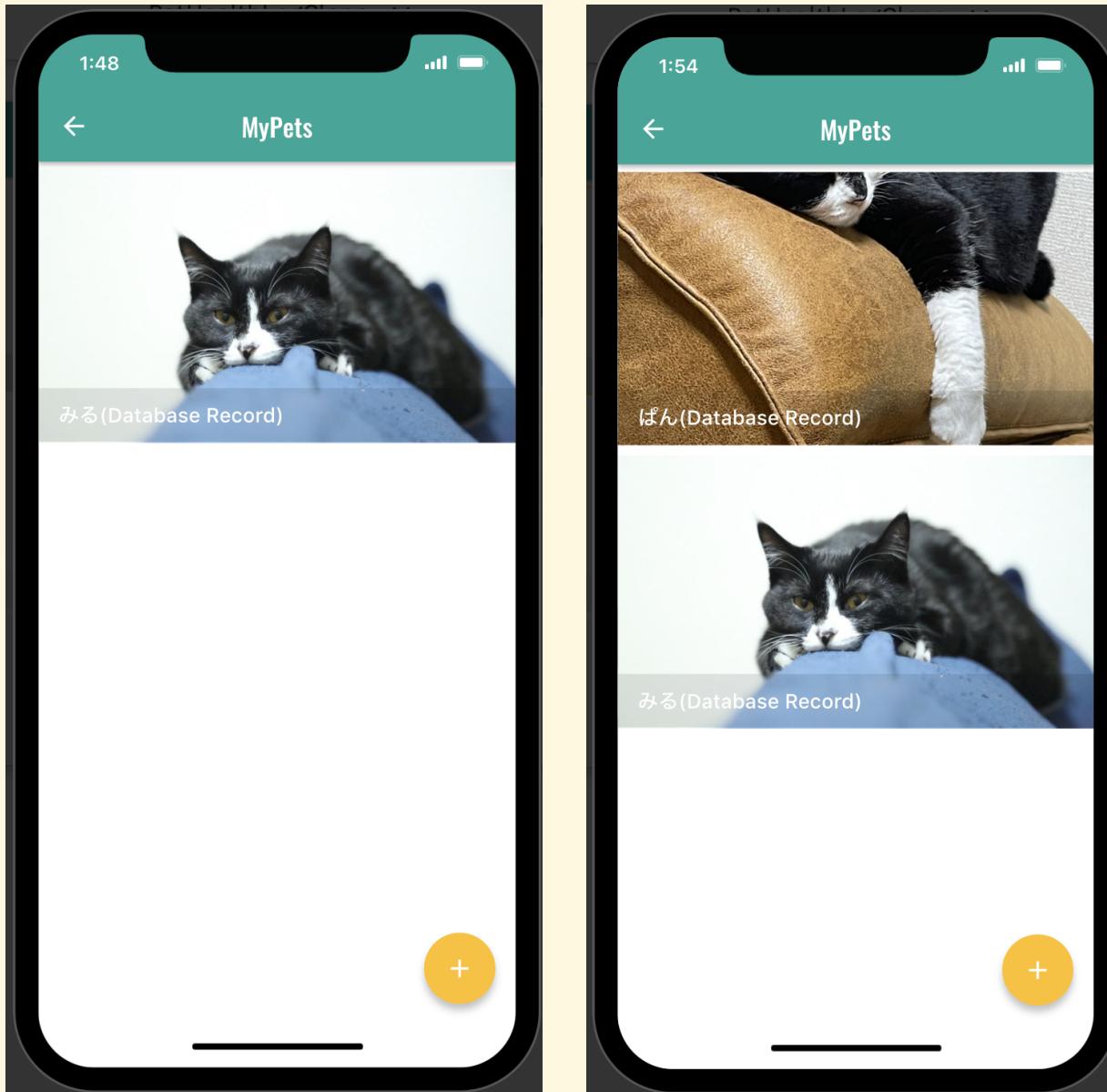
- Preview機能で確認すると、1匹目のペットとしてデータベースに登録したRecordが表示されます。



- ペットのListを構成するコンポーネントの中の2つ目のGroup(固定で表示していた2匹目のペット)は不要なので、削除しましょう

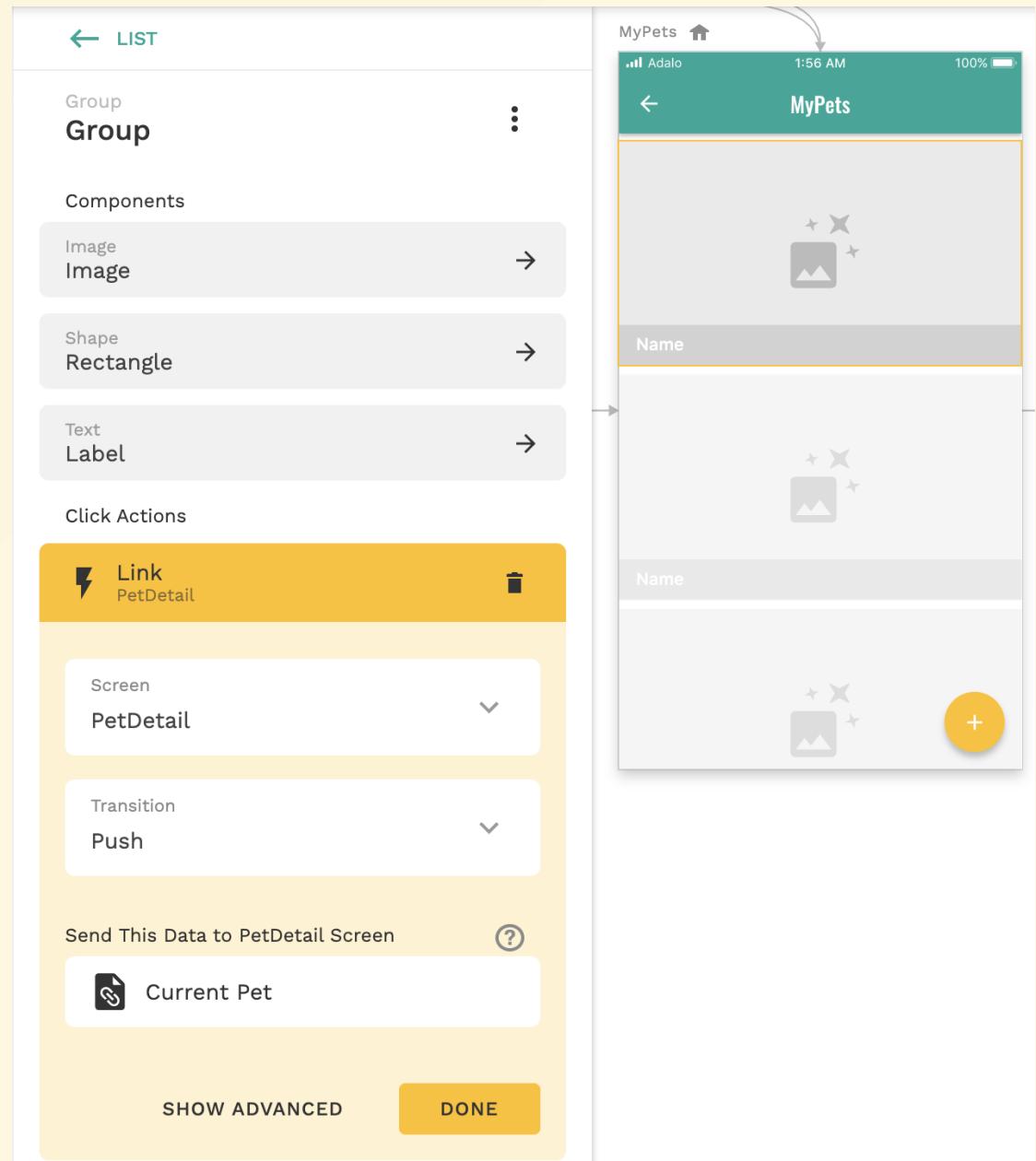


- Preview機能で確認すると、データベースに登録したペットだけが表示されるようになりました。
 - ペットを追加で登録すれば、複数のペットが表示されます。

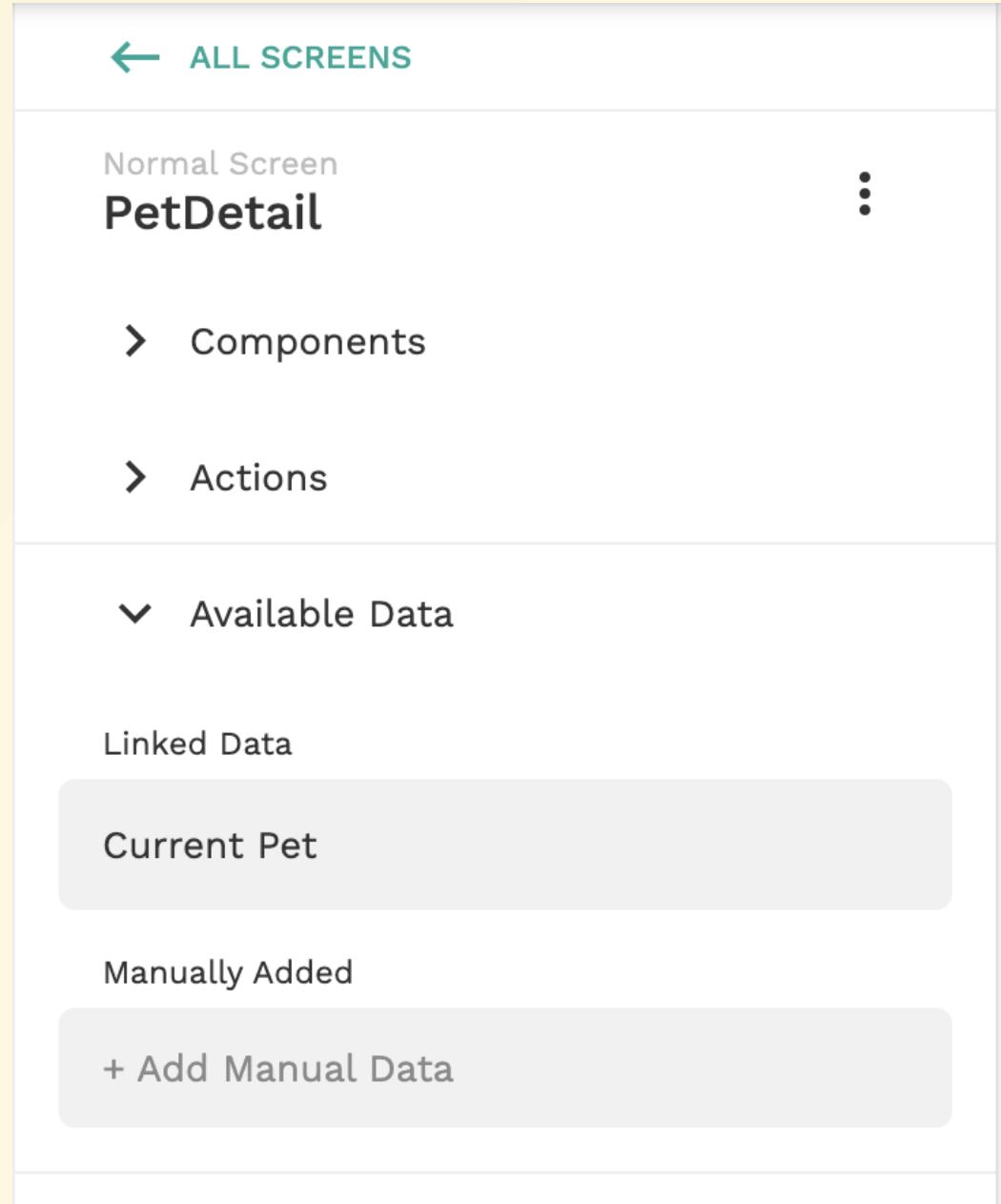


ペット一覧でペットをクリックした時に、そのペットの詳細画面に遷移できるようになっていることを確認します。

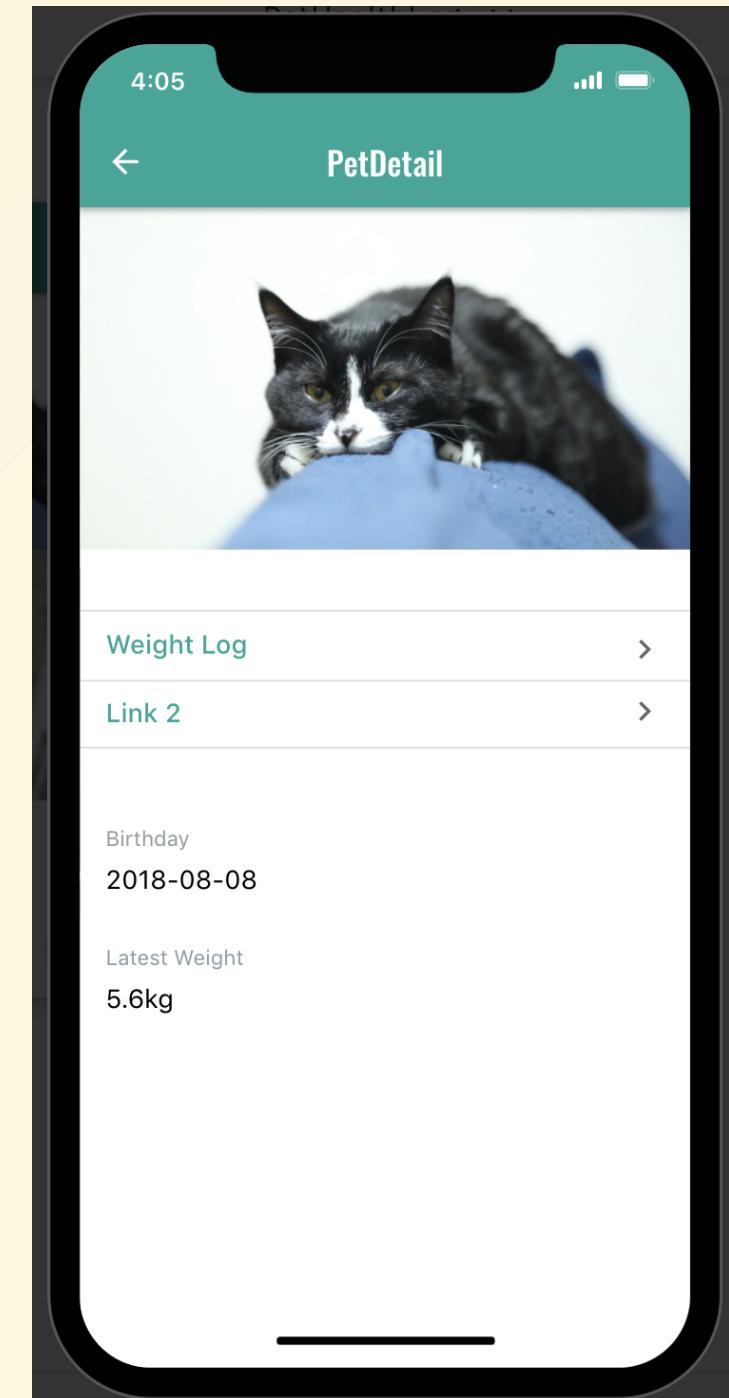
- ペットのGroupコンポーネントに設定されているLinkのSend This Data to PetDetail ScreenにCurrent Petが自動で設定されています。



- ペット一覧画面からのLinkのSend This Data to PetDetail ScreenにCurrent Petが設定されたため、ペット詳細画面のAvailable Data内にLinked DataとしてCurrent Petが設定されています。
 - これにより、ペット詳細画面でペット一覧画面で選択したペット(Current Pet)を扱えるようになります。

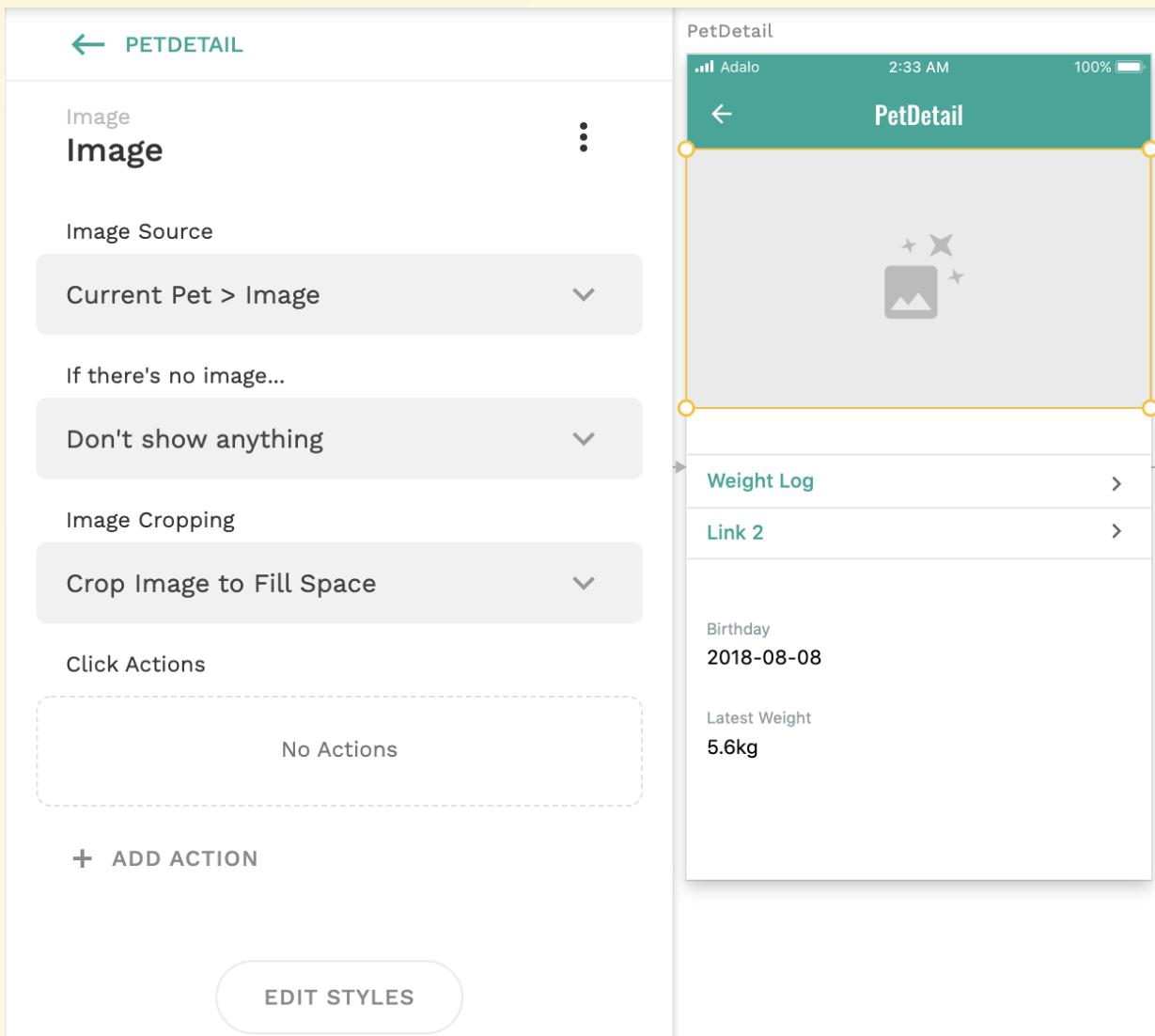


次に、ペット詳細画面にペット一覧画面で選択したペットが表示されるようにします。



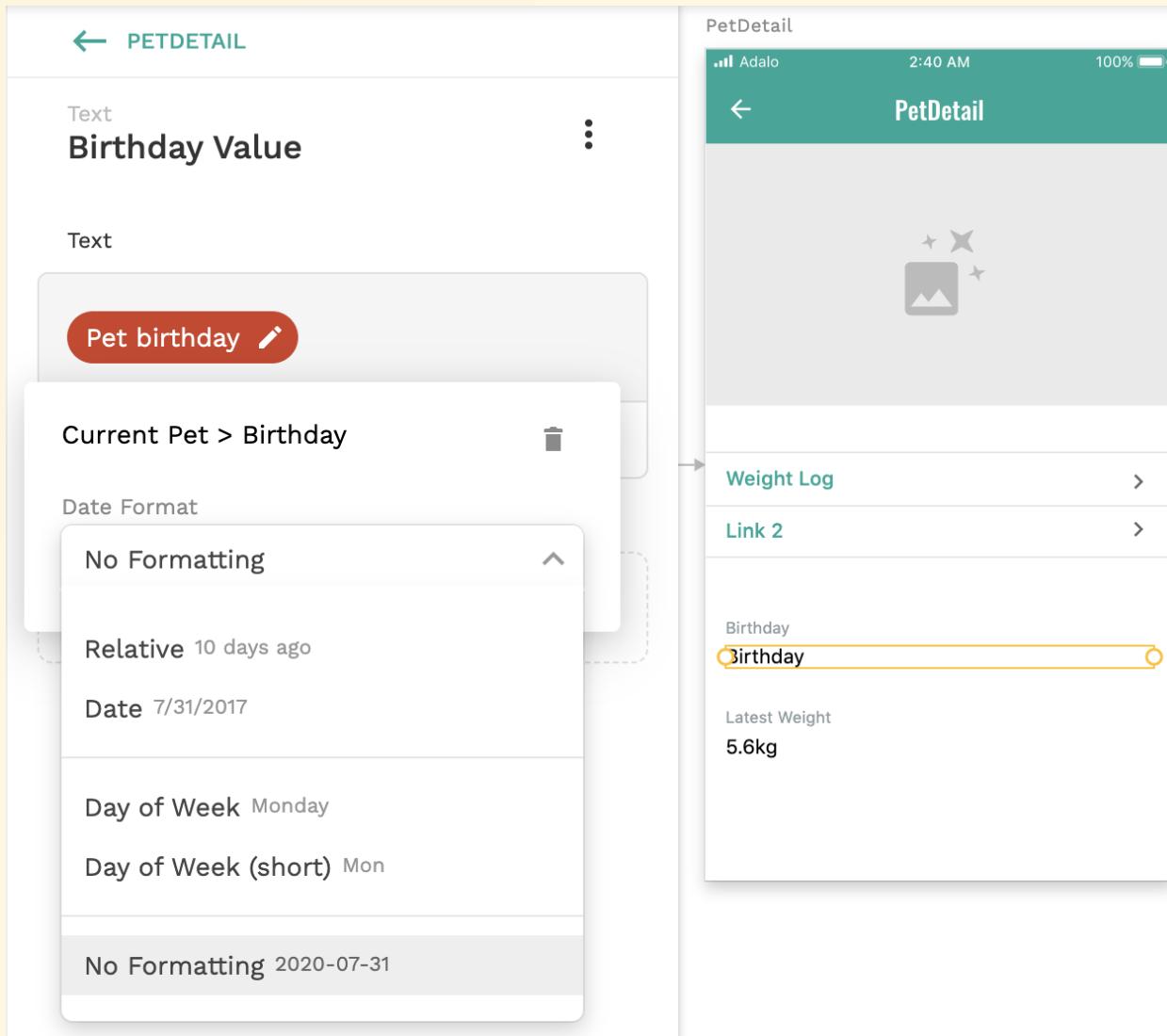
Imageコンポーネントをクリックし、

- Image SourceでDatabase > Current Pet's > Imageを選択
- If there's no image...でDon't show anythingを選択
 - あるいは、Show a place holder image を選択して好きなペットのシルエット画像を設定してもOKです



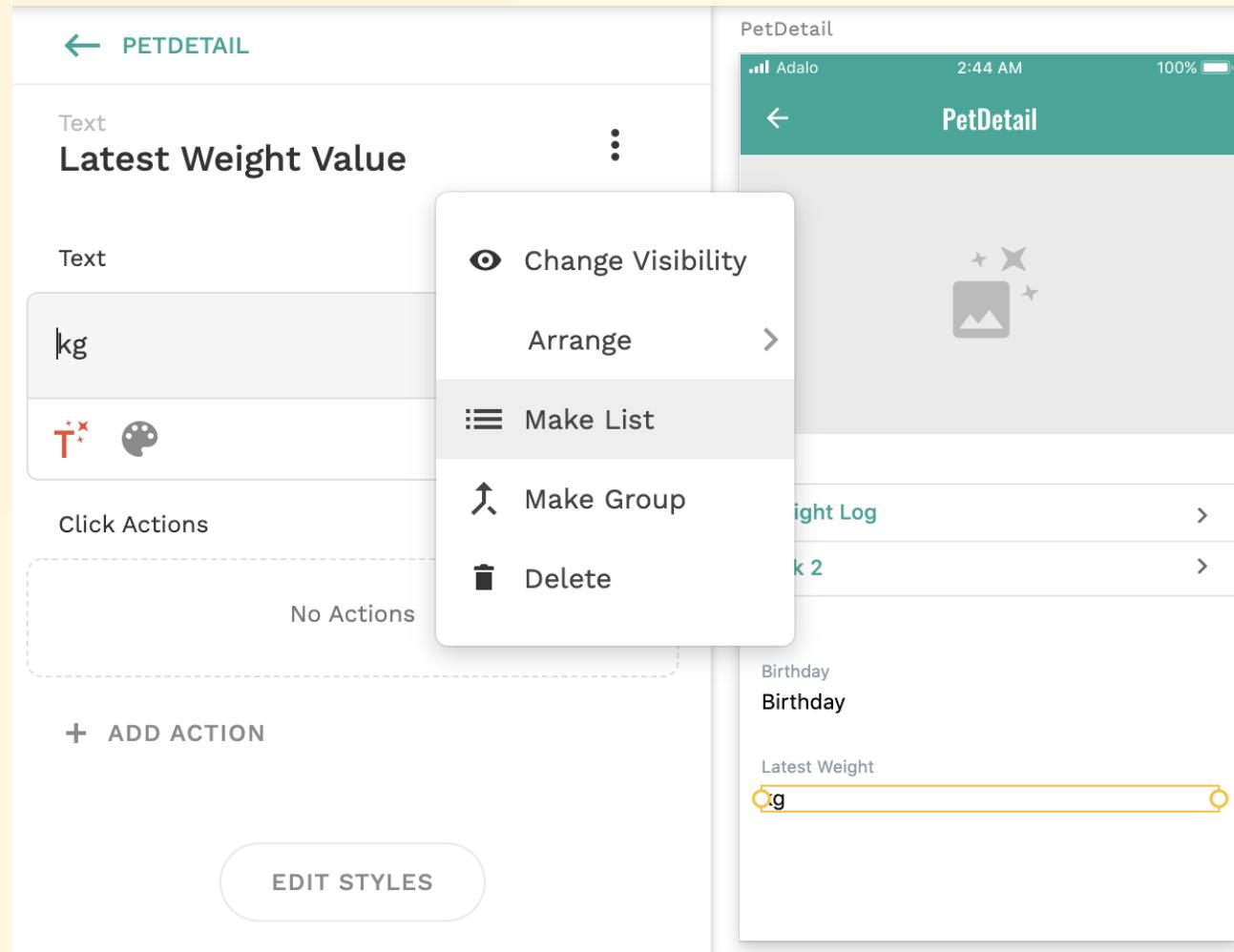
Birthday Valueコンポーネント をクリックし、

- TextでCurrent Pet's > Birthdayを選択
- Date FormatでNo Formattingを選択



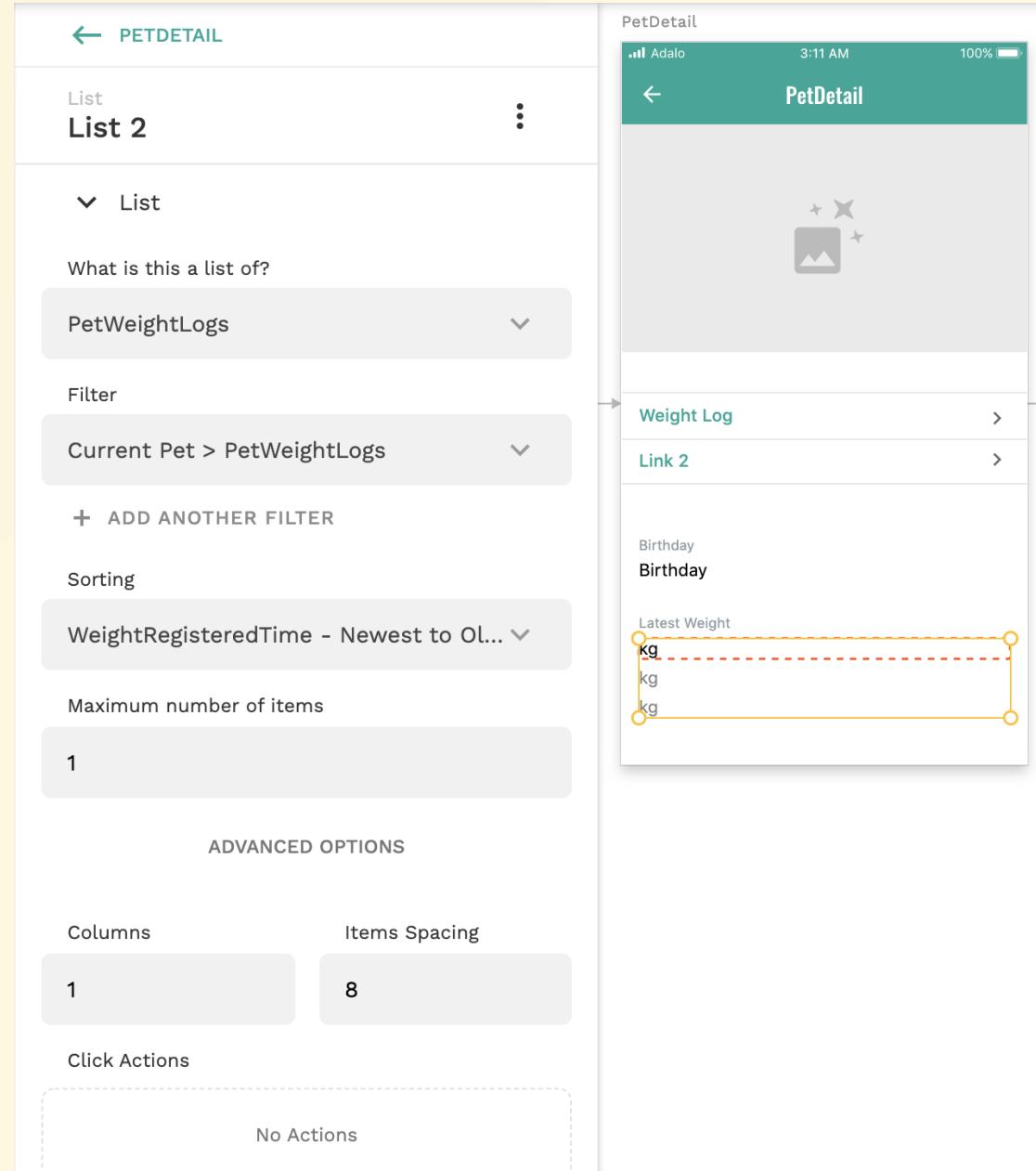
- Latest Weight Valueコンポーネントをクリックし、MAKE LISTでリストにする

* 最新の1件を表示するためには、そのコンポーネントをListにします
(次のページの設定で、最新の1件に絞り込みます)

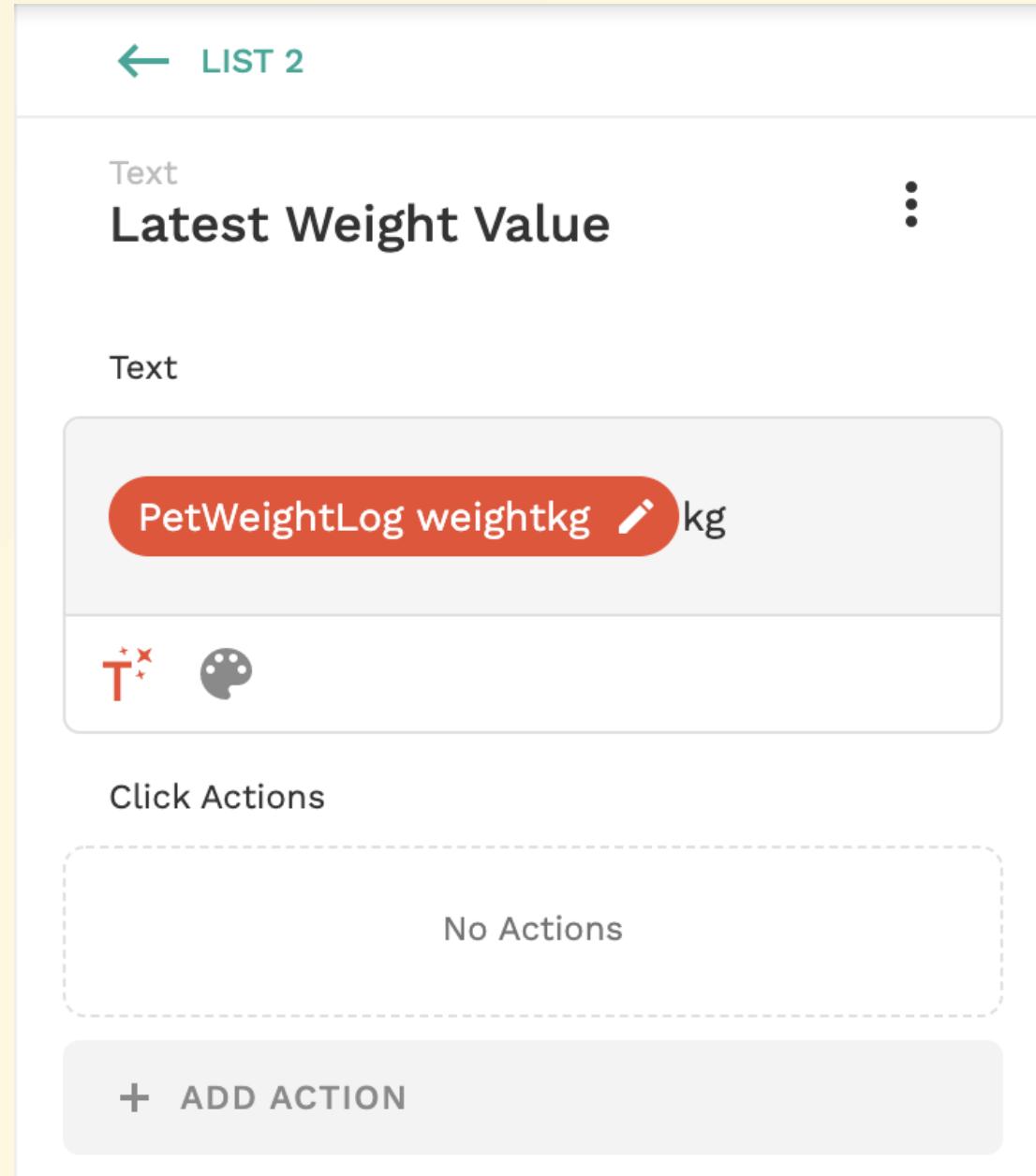


- What is this a list of?で PetWeightLogsを選択
- Filterで Current Pet > PetWeightLogsを選択
- Sortingで WeightRegisteredTime - Newest to Oldestを選択
- Maximum number of itemsに1を設定

これにより、最新の1件だけに絞り込まれます。

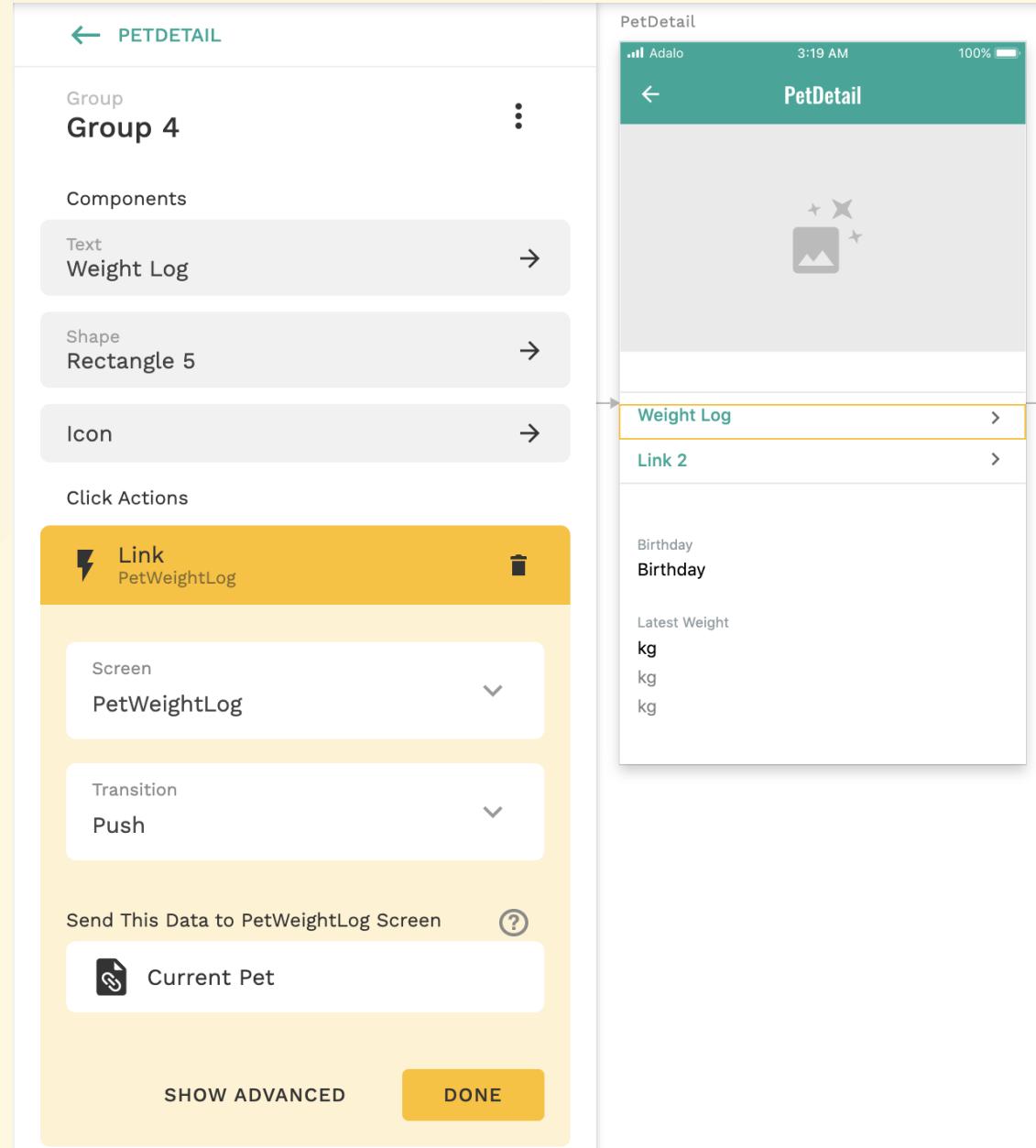


- Latest Weight Valueコンポーネントをクリックし、TextにCurrent PetWeightLog's > WeightKgを追加した後、末尾に"kg"をつける



ペット詳細画面からペットの体重管理画面に遷移する際に選択したペットが受け渡されることを確認します。

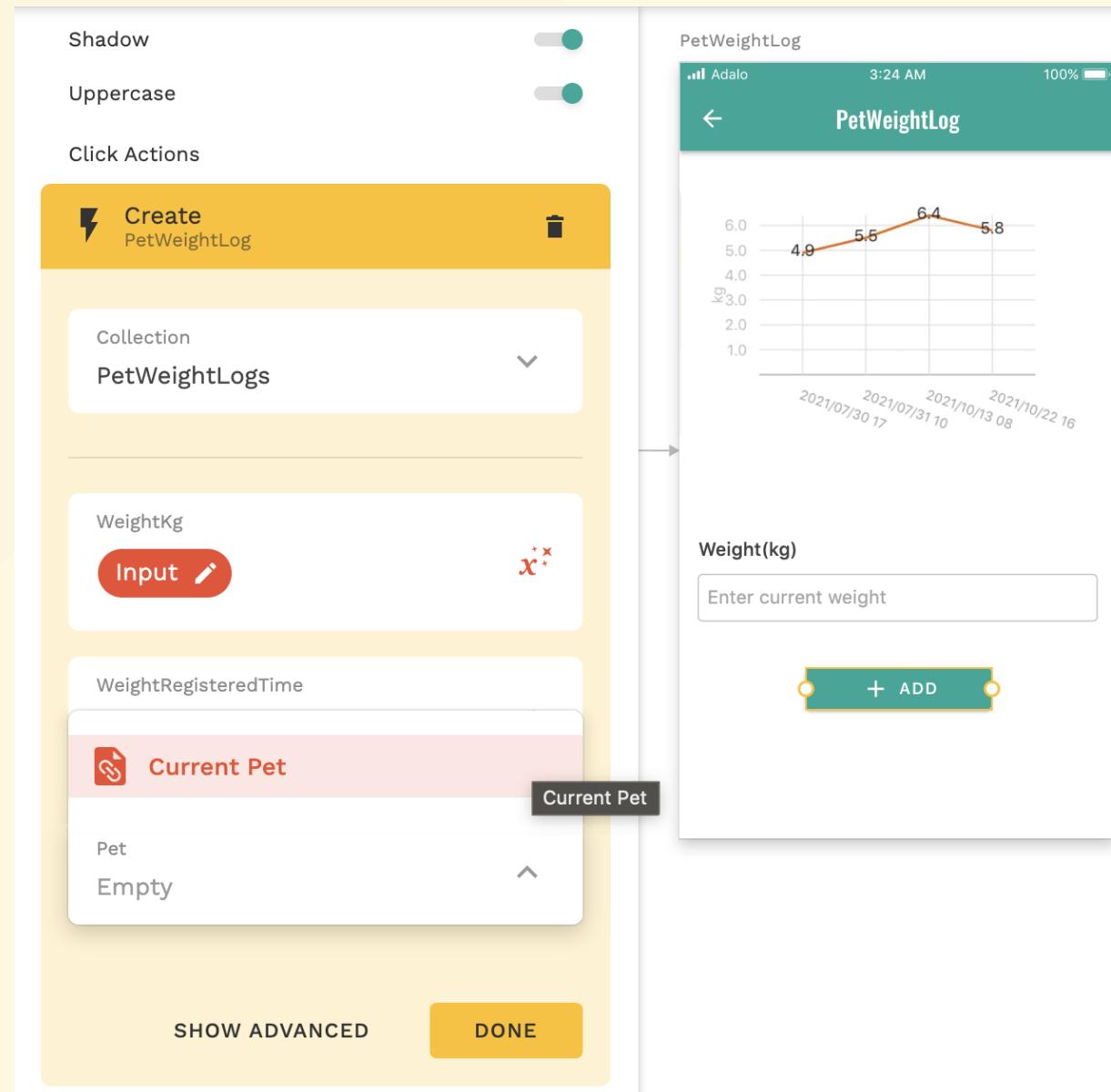
- Weight Logというリンクのコンポーネントを選択し、LinkのSend This Data to PetDetail ScreenにCurrent Petが自動で設定されています。



* データの作成(CREATE)の残作業

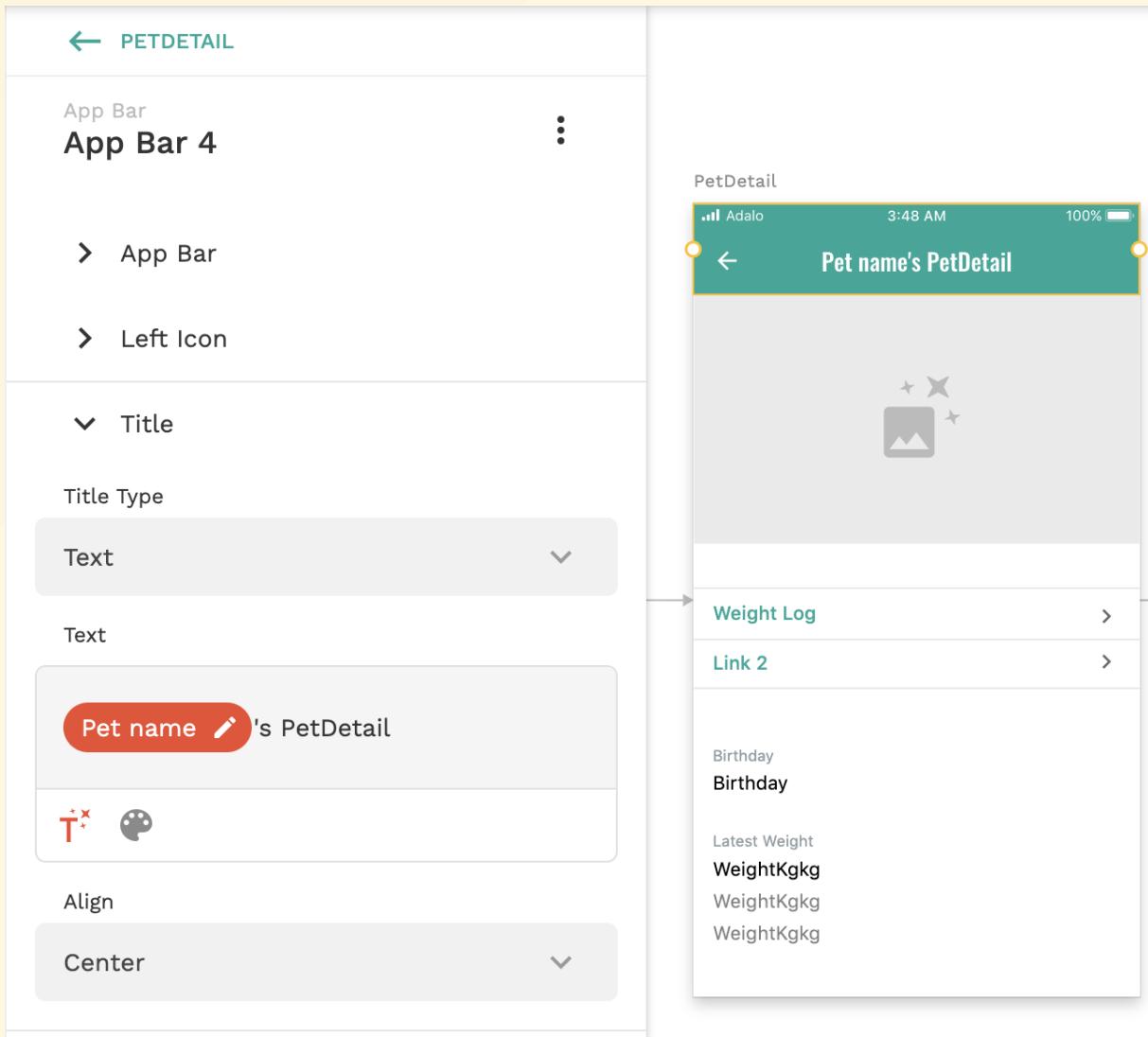
ペットの体重管理画面で選択したペットの体重をデータベースに登録できるようにします。

- ADDボタンを選択し、CreateのActionで一旦EmptyのままにしていたPetにCurrent Petを設定



ペット詳細画面、ペットの体重管理画面のヘッダーに選択したペットの名前を表示しましょう。それぞれの画面で、

- 画面上部のApp Barコンポーネントを選択し、TitleのTextに Current Pet's > Name を追加
- その後ろに、"'s" という文字列を追加

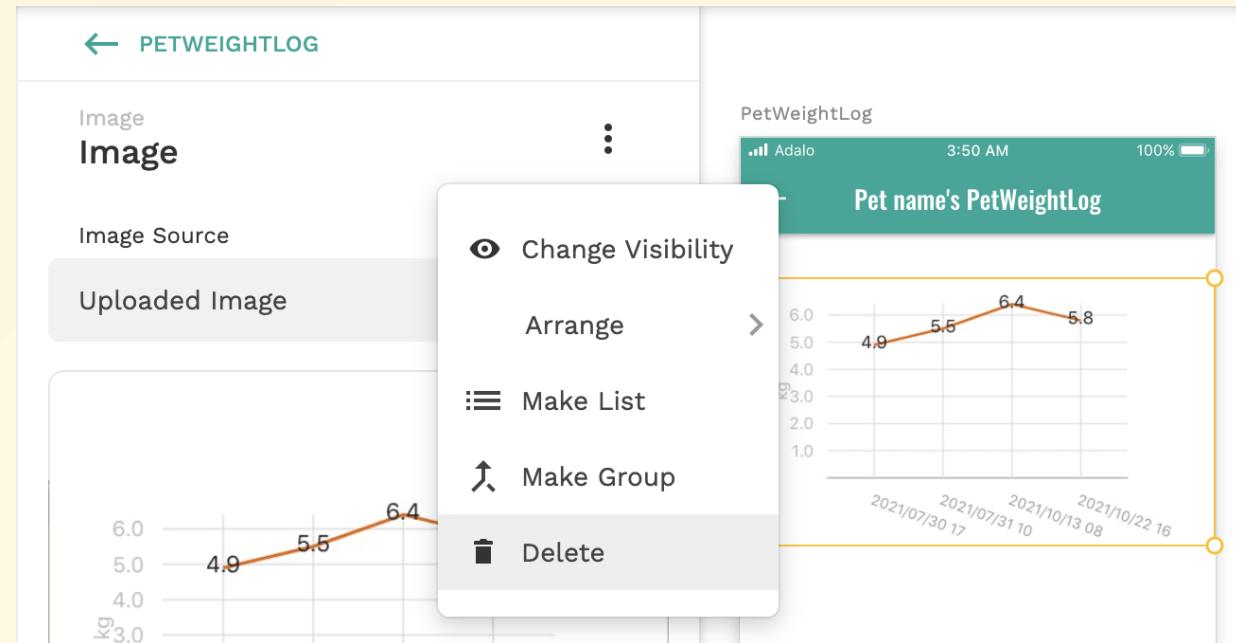


Preview機能で、以下を確認しましょう。

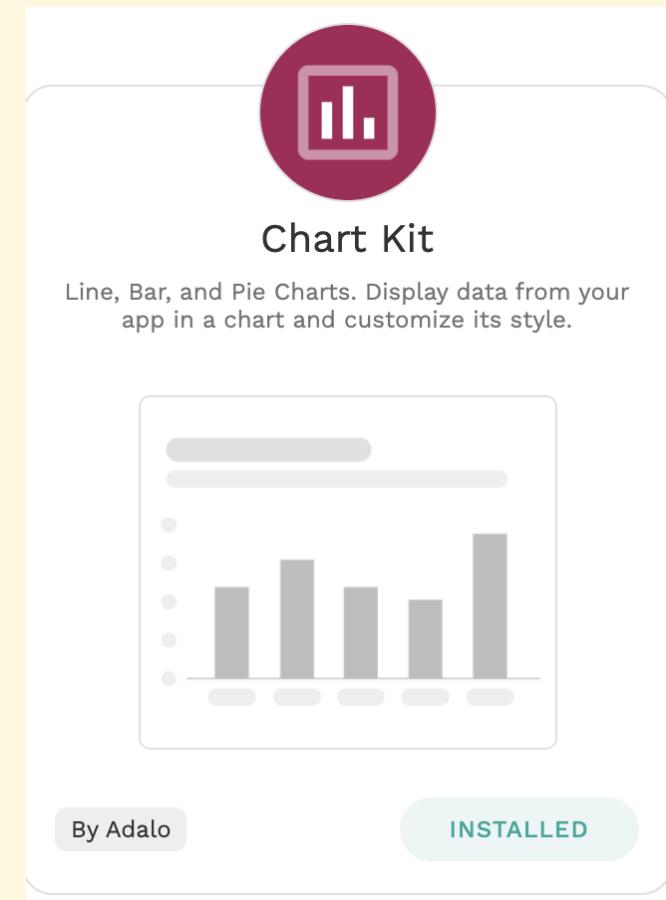
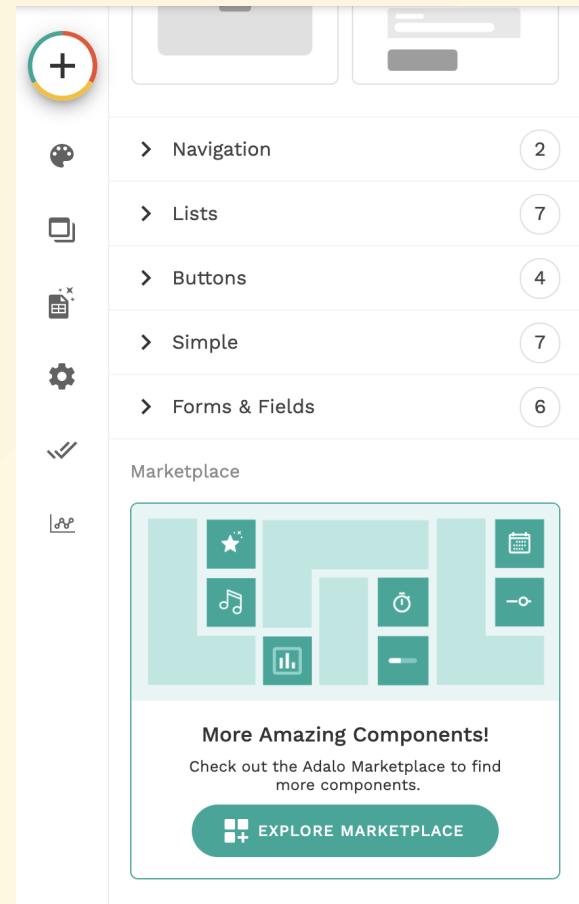
- ペット一覧画面から選択したペットの詳細画面に遷移できること
- ペット詳細画面から選択したペットの体重管理画面に遷移できること
- ペットの体重を登録すると、データベースのPetWeightLogs CollectionにRecordが登録されること
- ペット詳細画面に最新の体重が表示されること

次に、ペットの体重管理画面で過去に登録したペットの体重がグラフとして表示されるようになります。

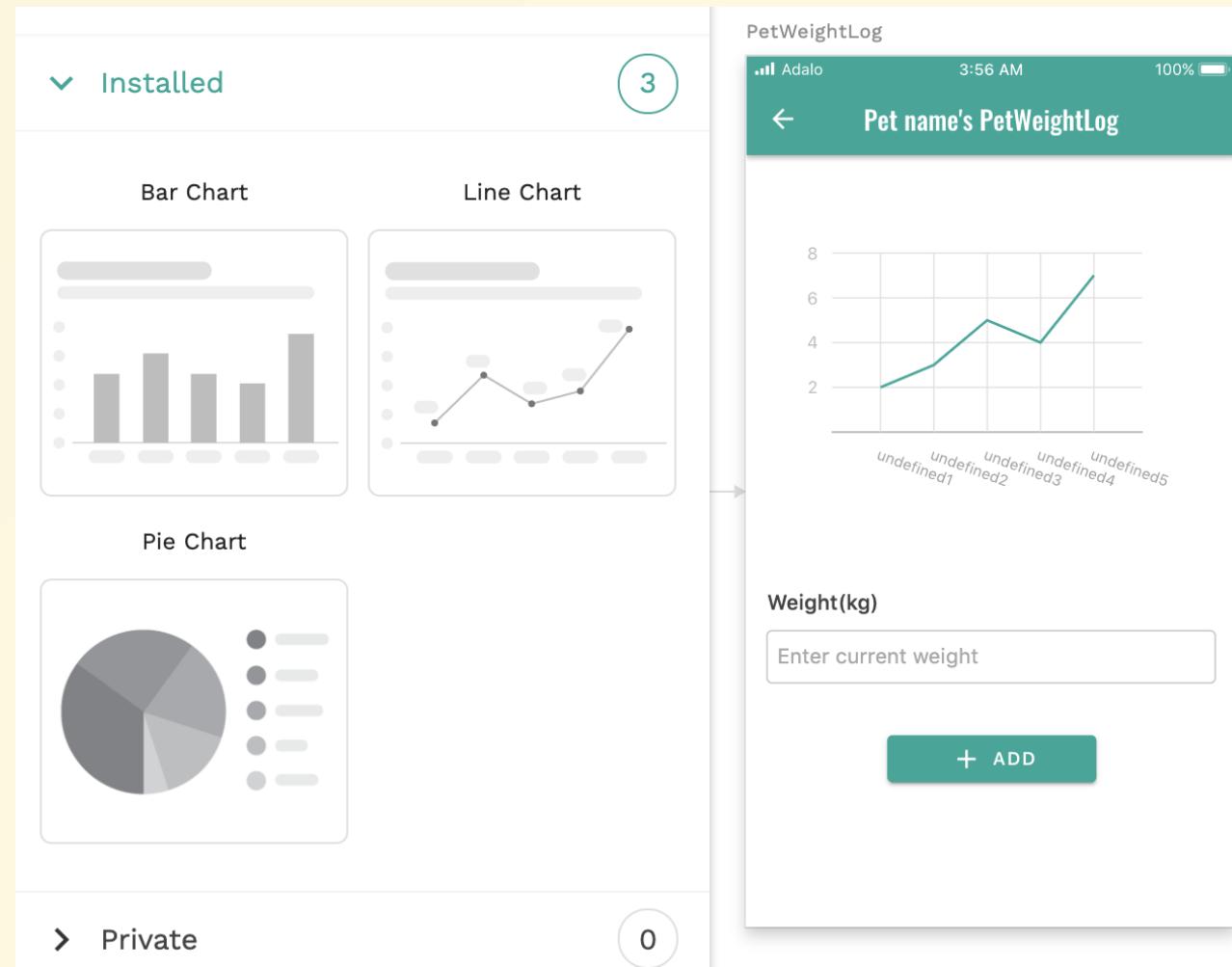
- まず、画像で貼り付けていたグラフを削除



- ADD COMPONENTから
EXPLORE MARKETPLACE
を選択
- Chart KitをINSTALL



- Line Chartを画面に追加

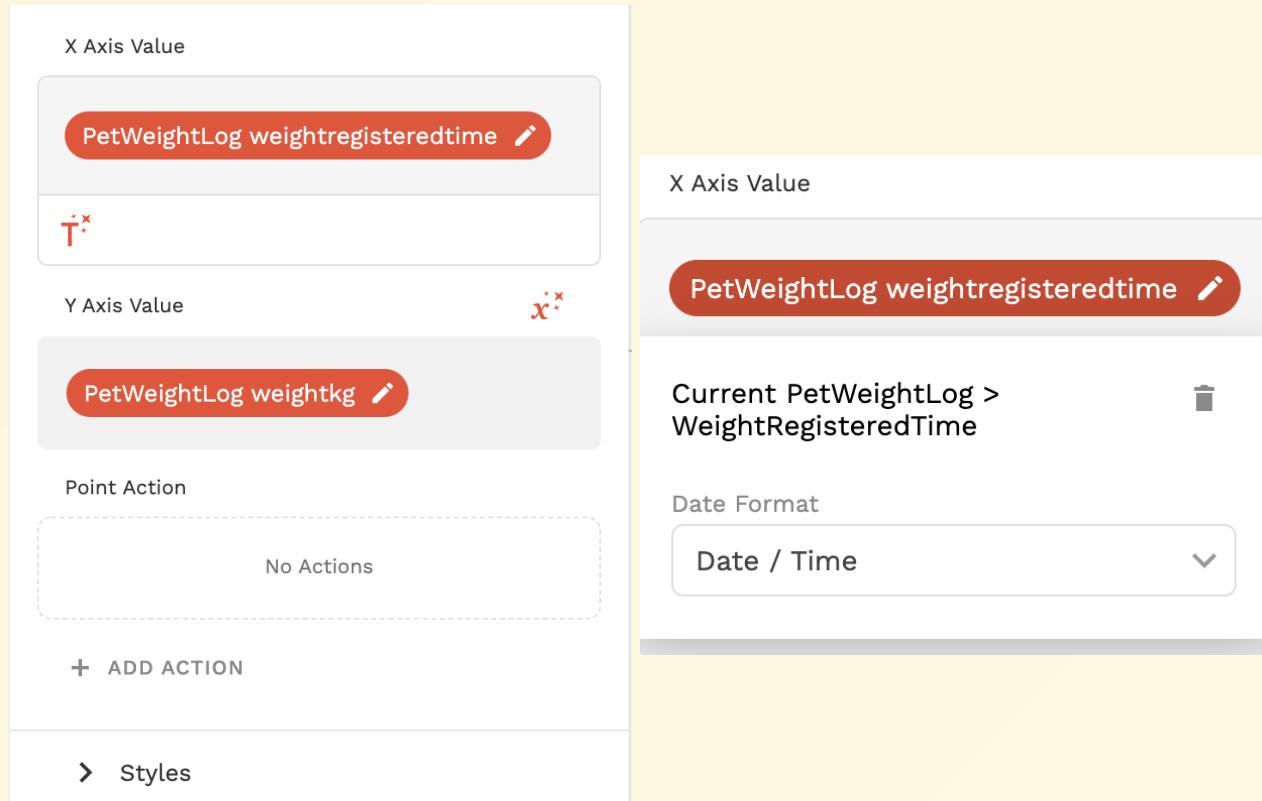


Line Chartを設定します。

- What is this a chart of?で PetWeightLogsを選択
- FilterでCurrent Pet > PetWeightLogsを選択
- Custom Filterに WeightRegisteredTime Is after 30 days ago を設定し、表示期間を指定
- Sortingで WeightRegisteredTime - Oldest to Newestを選択

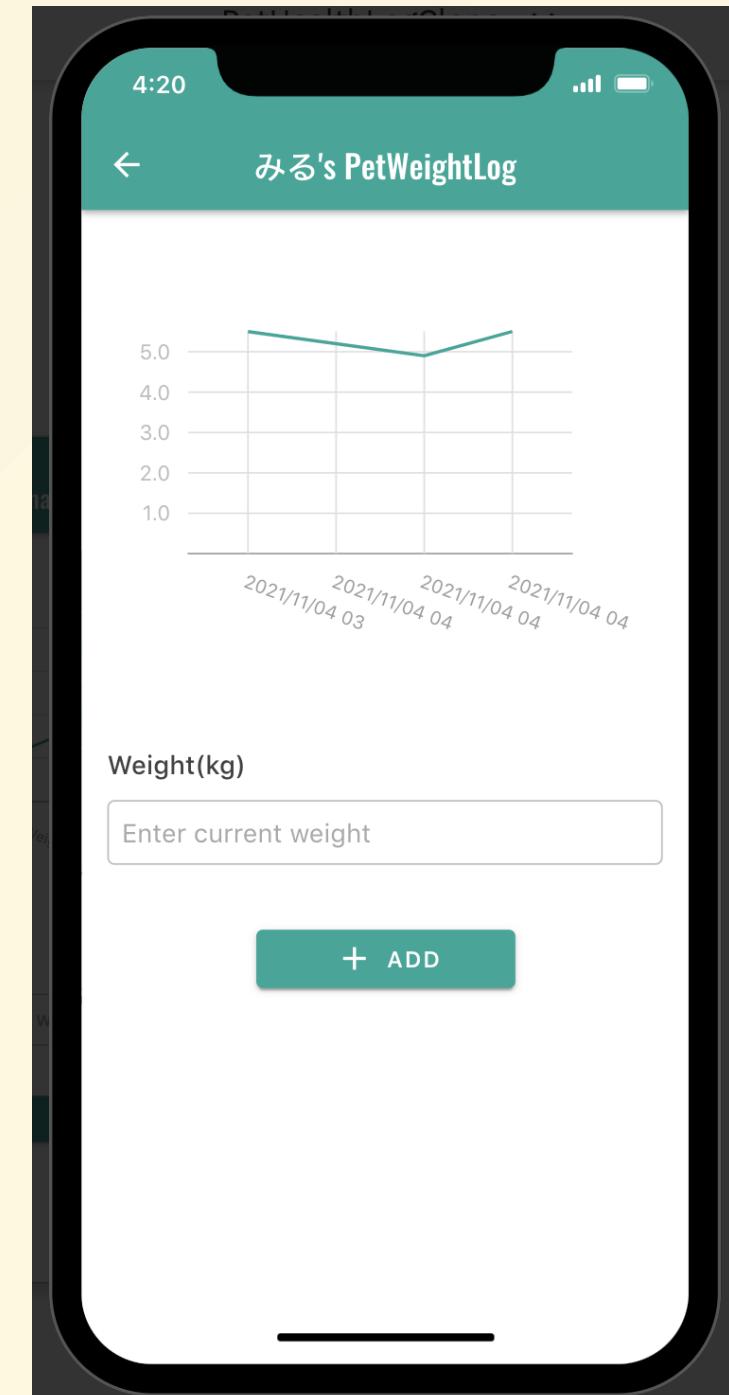
The screenshot shows the configuration interface for a Line Chart. At the top, it says "Line Chart" and "Line Chart". Below that, there's a section titled "What is this a chart of?" with a dropdown menu set to "PetWeightLogs". Underneath is a "Filter" section with a dropdown menu set to "Current Pet > PetWeightLogs". A "Custom Filter" section follows, containing a dropdown menu for "WeightRegisteredTime" with "Is after" selected and "30 days ago" chosen. There's also a red "X" button next to the filter input. Below the filter section is a "+ ADD ANOTHER FILTER" button. The next section is "Sorting" with a dropdown menu set to "WeightRegisteredTime - Oldest to Ne...". At the bottom, there's a "Maximum number of items" section with a dropdown menu set to "No Maximum". At the very bottom is a "ADVANCED OPTIONS" button.

- X Axis Valueに
PetWeightLog >
WeightRegisteredTimeを設
定
 - Date FormatにDate /
Timeを設定
- Y Axis Valueに
PetWeightLog > WeightKg
を設定



Preview機能でグラフの表示を確認しましょう。体重を複数追加すると、グラフが描画されます。

* 体重の登録日時が長く、省略して表示されてしまいます。同じ日に複数の体重を登録してテストをしたかったのでDate&Time型にしていますが、本来はDate型にして、1日に1回しか登録できないよう制御する方が良さそうです。

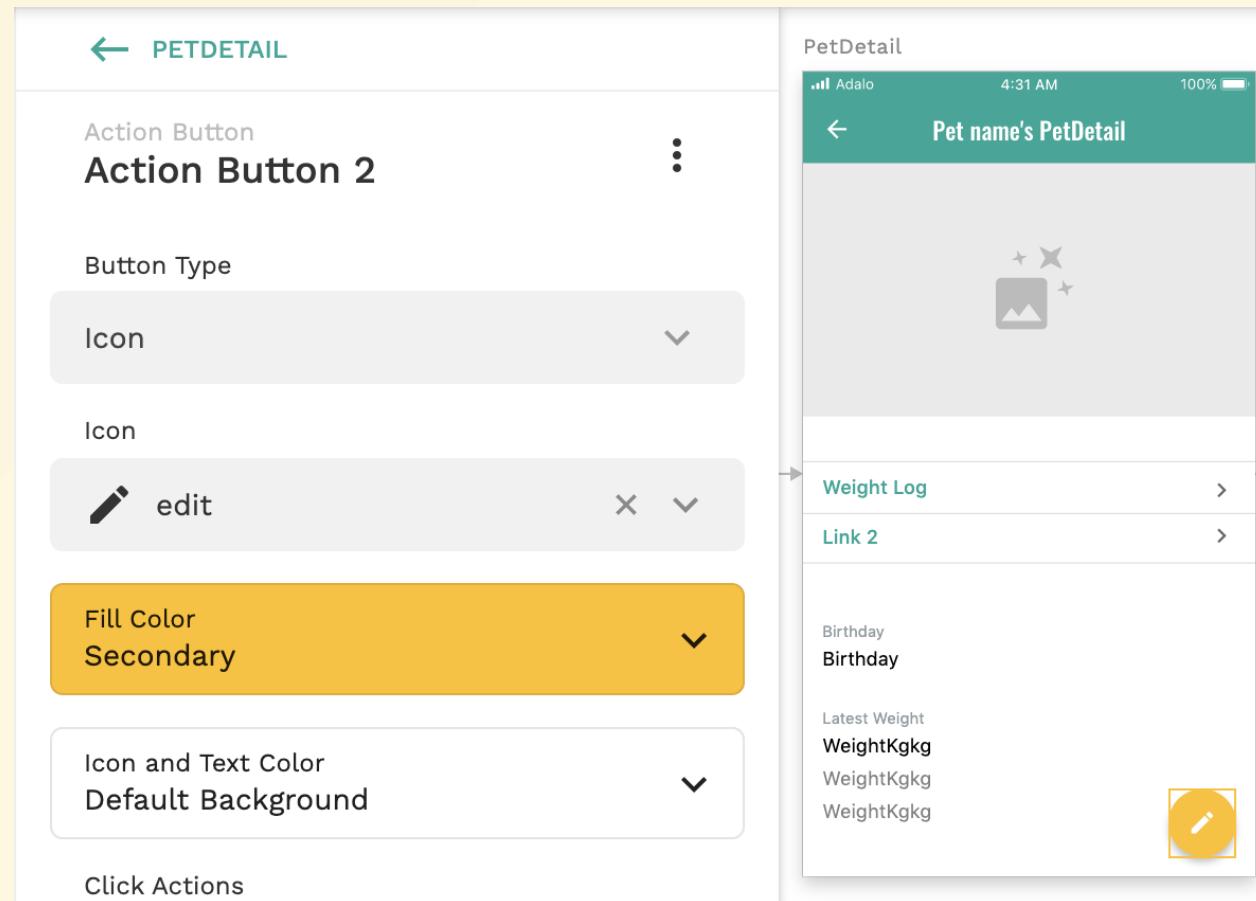


データの更新(UPDATE)

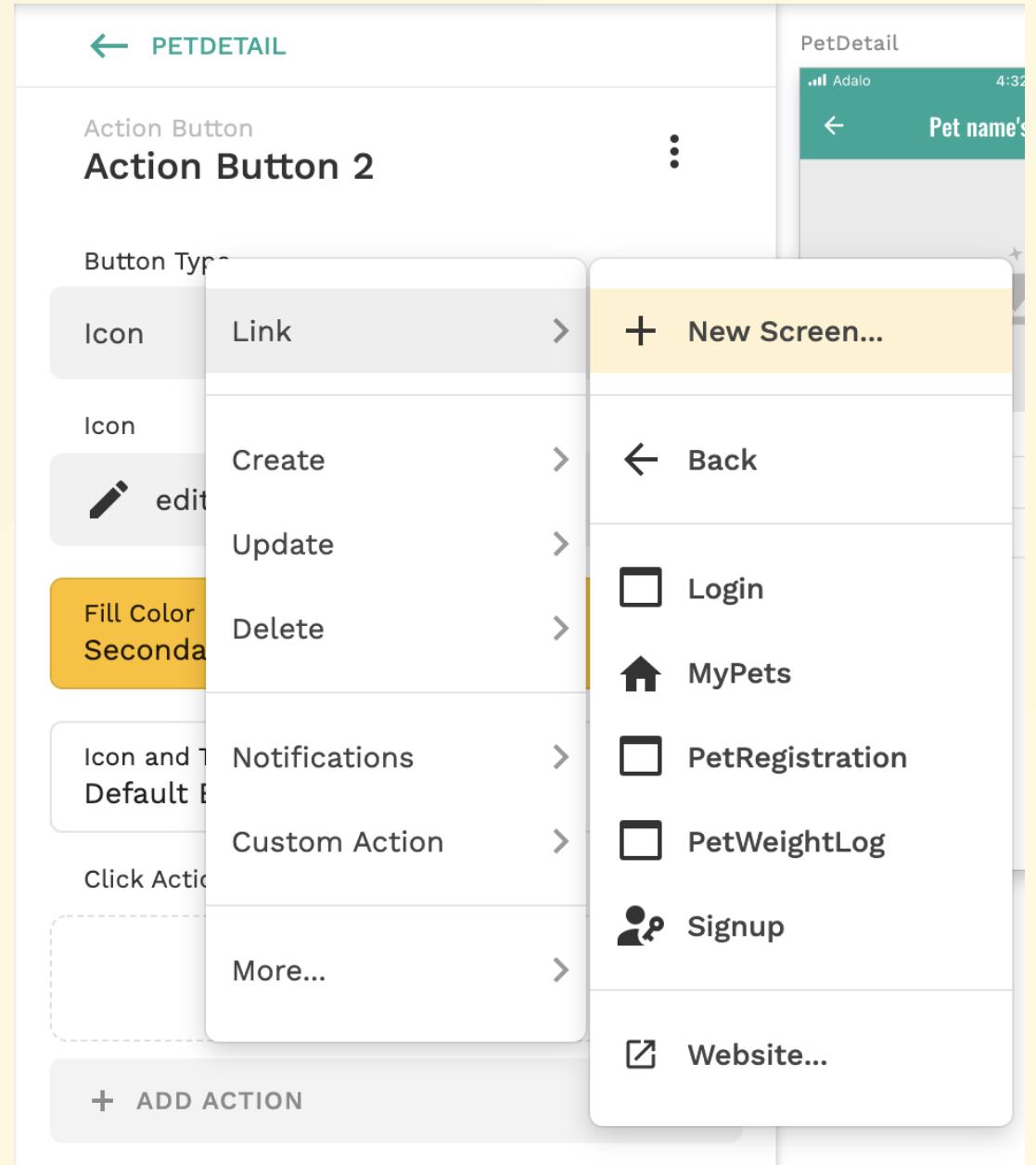
登録したペットの情報を更新できるペット情報編集画面を新たに作ります。

まず、ペット詳細画面にペット情報編集画面への導線を追加します。

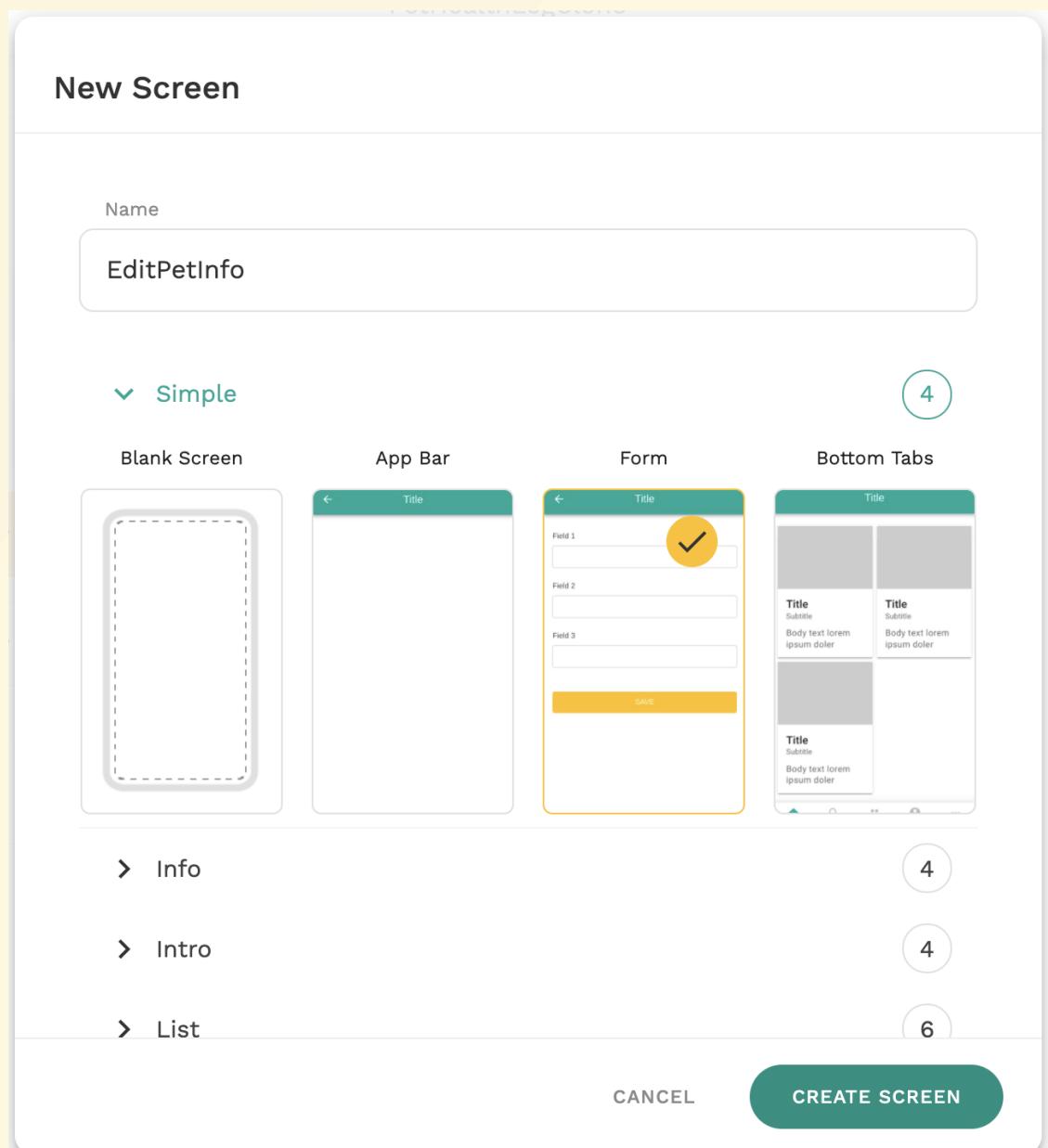
- ADD COMPONENTから Action Buttonを追加
- Iconをeditに変更
- Icon and Text Colorを Default Background(white) に変更



- ADD ACTIONからLink > New Screenを選択



- Screen Nameを入力
- Formを選択
- CREATE SCREENをクリック

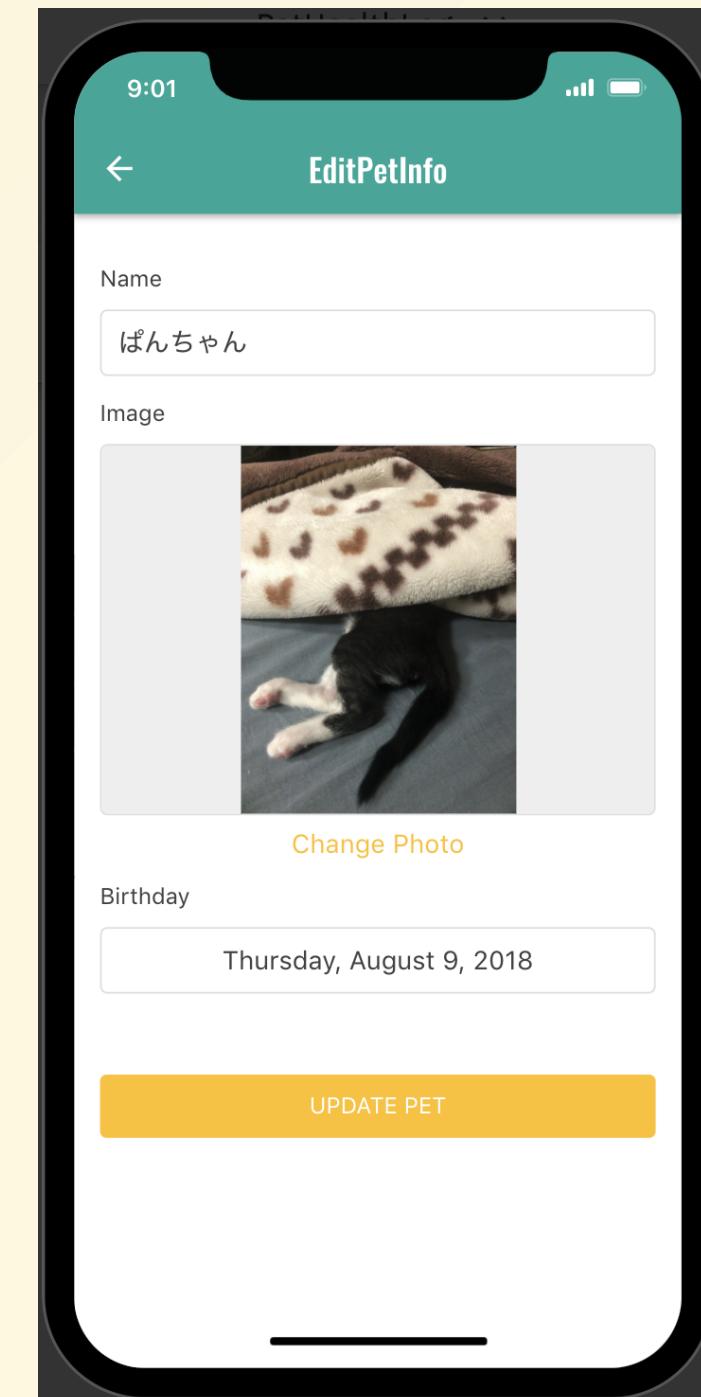


以下の設定をするだけでペット情報編集画面は完成です。

- Which data collection?で Petsを選択
 - 選択したCollectionに合わせたフォームを自動生成してくれる
- What do you want the form to do?でUpdate Current Petを選択
- FieldsでBirthdayとImageの順番を入れ替え

The image shows two screens of the Adalo mobile application. The left screen is a configuration interface titled 'EDITPETINFO' with a subtitle 'Form'. It displays a dropdown menu for 'Which data collection?' set to 'Pets', another dropdown for 'What do you want the form to do?' set to 'Update Current Pet', and two expandable sections for 'Fields' and 'Submit Button'. The right screen is a preview of the 'EditPetInfo' screen on an iPhone. This screen has a green header with the title 'EditPetInfo'. It contains three input fields: 'Name' (placeholder 'Enter name...'), 'Birthday' (placeholder 'Thursday, May 4th, 2019'), and 'Image' (with a 'Choose Photo' button). At the bottom is a large orange 'UPDATE PET' button.

Preview機能でペット情報編集
画面が使えることを確認しま
しょう。



補足

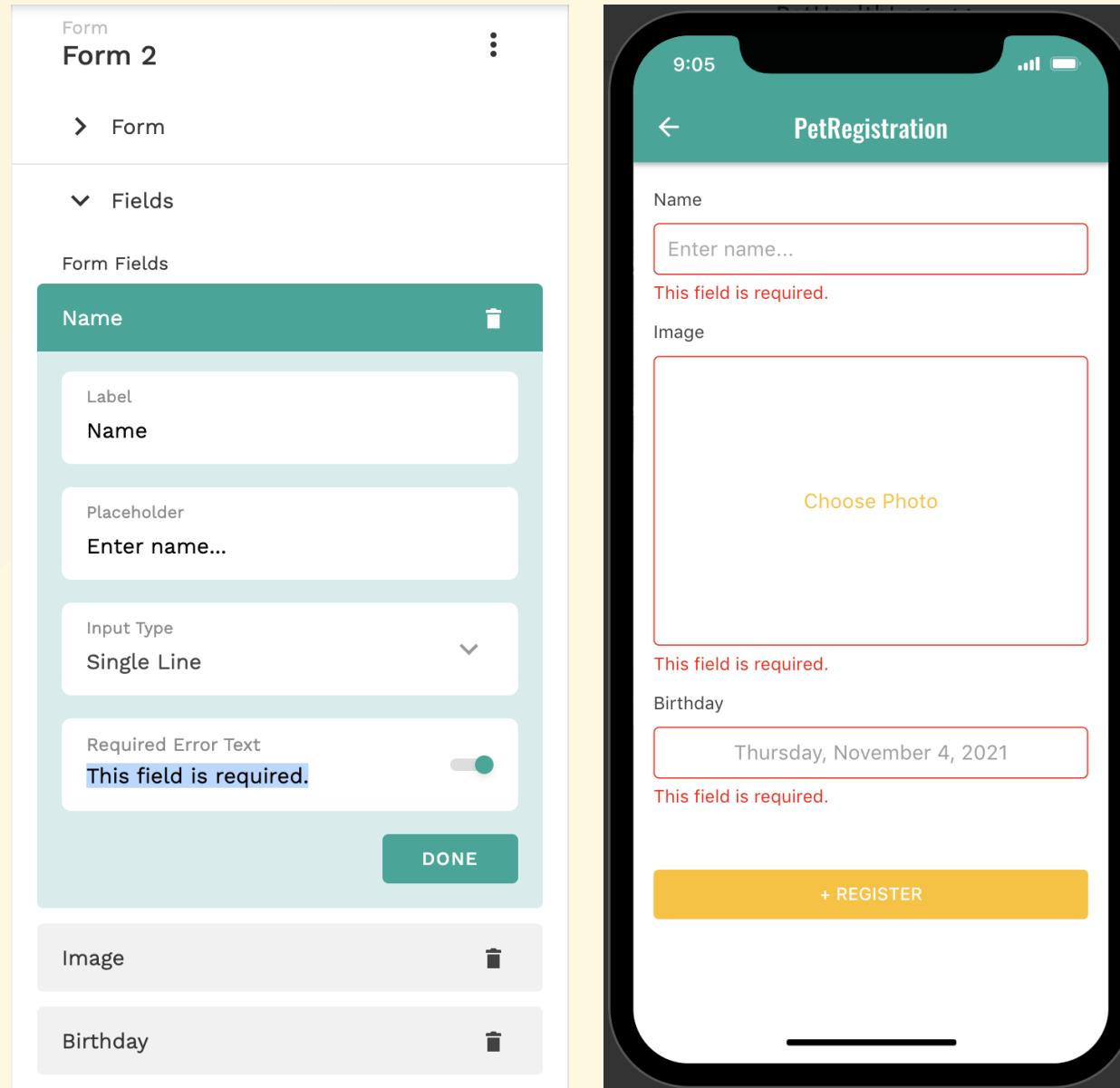
- 前回はペット登録画面の入力フォームをコンポーネントを一つずつ追加して作成しましたが、今回ペット編集画面を作ったのと同様の手順で自動生成できます。

やってみよう

- ペット登録画面のAppBar以外のコンポーネントを削除して、Formコンポーネントを追加し、Pet Collectionを指定してフォームを自動生成してみましょう。
 - 登録後のペット一覧画面への遷移だけは、手動でSubmit ButtonにActionを追加する必要があるので、注意してください。

補足

- Formコンポーネントを使えば、FieldのRequired Error Textにチェックを入れるだけで、必須項目が入力されているかどうかをチェックすることができます。
- 入力フォームを作る際は、できるだけFormコンポーネントを使いましょう。



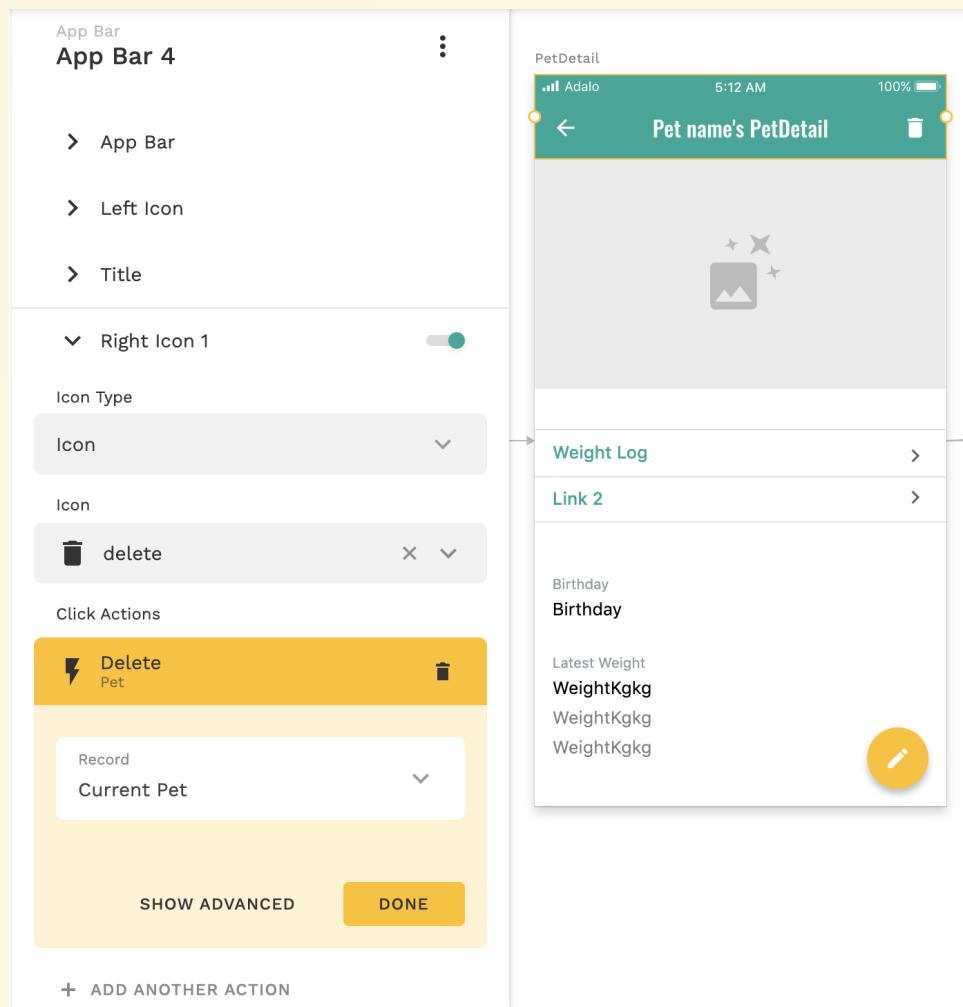
参考: 実際のAdaloでのアプリ開発の流れについて

- サンプルアプリでは、UI作成 > データベース設計 > それらの連携という流れで作成を進めてきました。
 - これは、どのような画面かが分からないと必要なデータも分からず、データベースの設計も難しいと考えたためです。
- 実際にAdaloでアプリ開発をする際は、まずUIスケッチ等を描いて必要なデータを洗い出し、その次にデータベースの設計を行うことをおすすめします。そうすることで、その後のUIの作成に自動生成を活用できます。
 - とはいっても、実際の開発は一方的な流れではなく、データベースとUIを交互に変更しながら試行錯誤することになると思います。

データの削除(DELETE)

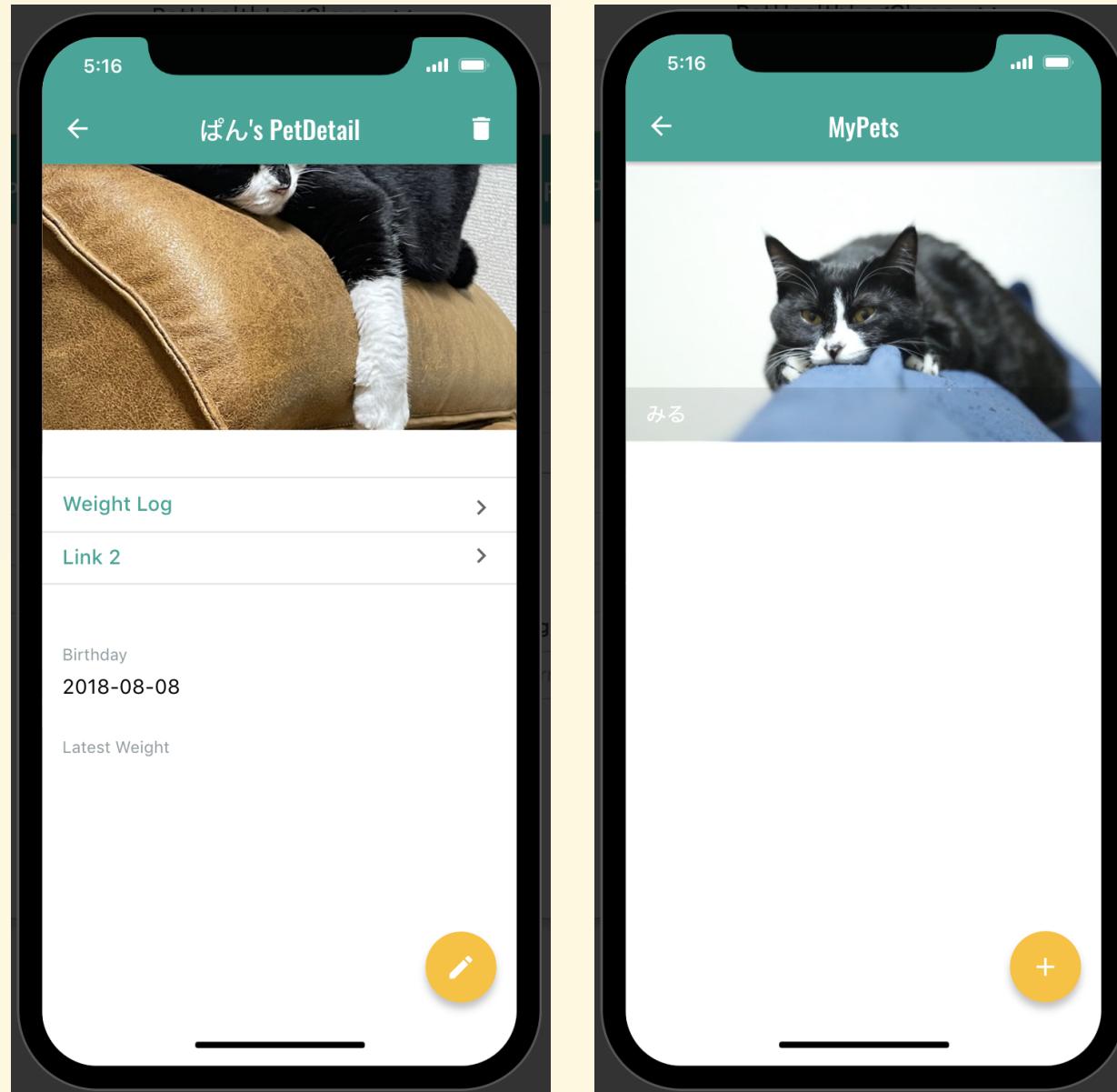
ペット詳細画面に、登録したペットを削除できるボタンを作ります。

- App Barを選択し、Right Icon 1のトグルをONにする
- Iconをdeleteに変更
- ADD ACTIONからDelete > Current Petを選択
- ADD ANOTHER ACTIONからLink > Mypetsを選択



Preview機能で削除を試してみましょう。

削除が完了してペット一覧画面に遷移すると、削除したペットは表示されません。



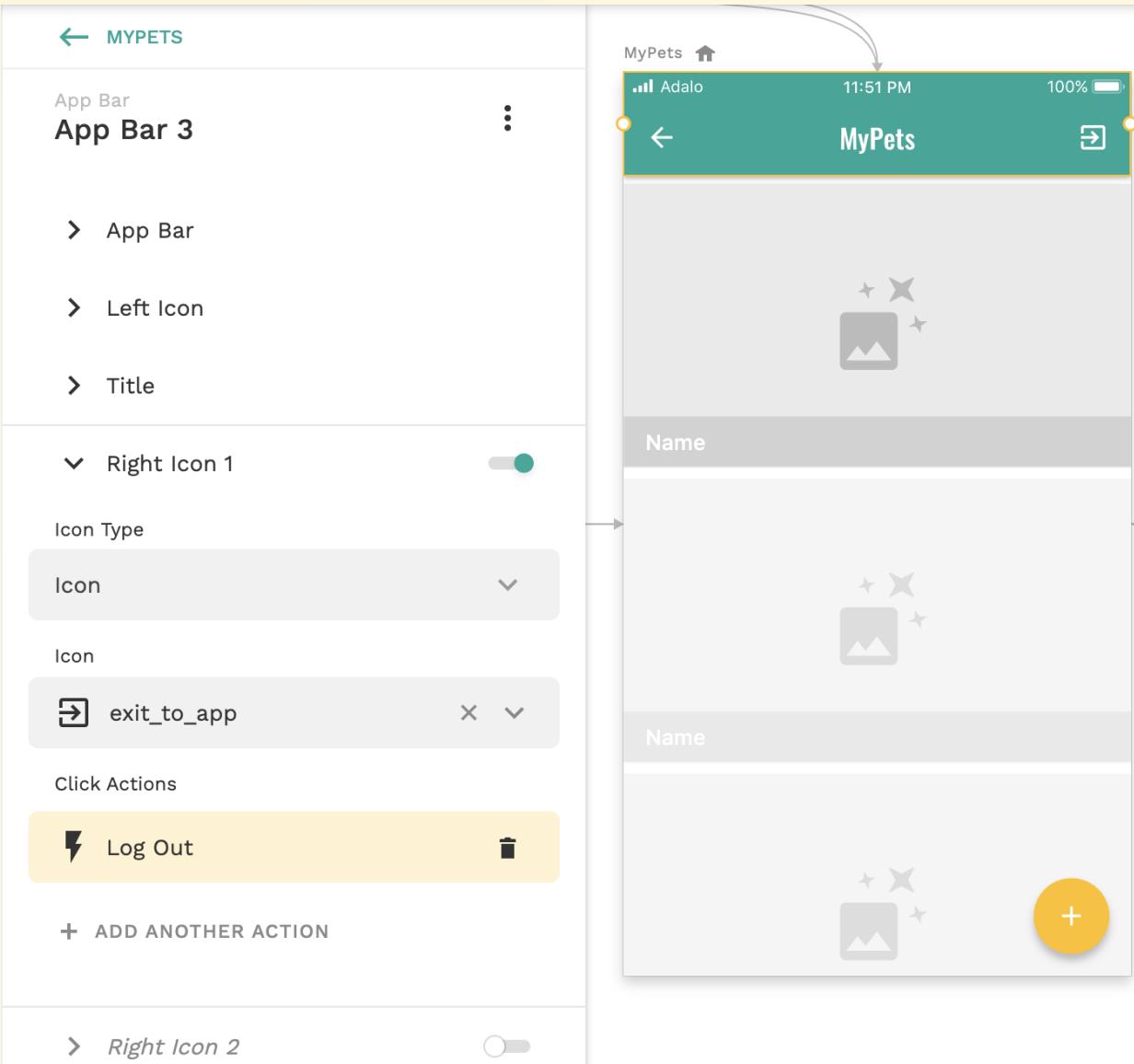
以上で、データベースに対するCRUD操作(CREATE, READ, UPDATE, DELETE)を全種類実装することができました。

サンプルアプリを改善しよう

まだ紹介していないAdaloの機能を使いながら、サンプルアプリを改善していきます。

ログアウト

- ペット一覧画面のAppBarを選択し、Right Icon 1を有効化
- Iconをexit_to_appに変更する
- ADD ACTION > More... > User Login > Log Outを選択
- ADD ANOTHER ACTION > Link > Loginを選択



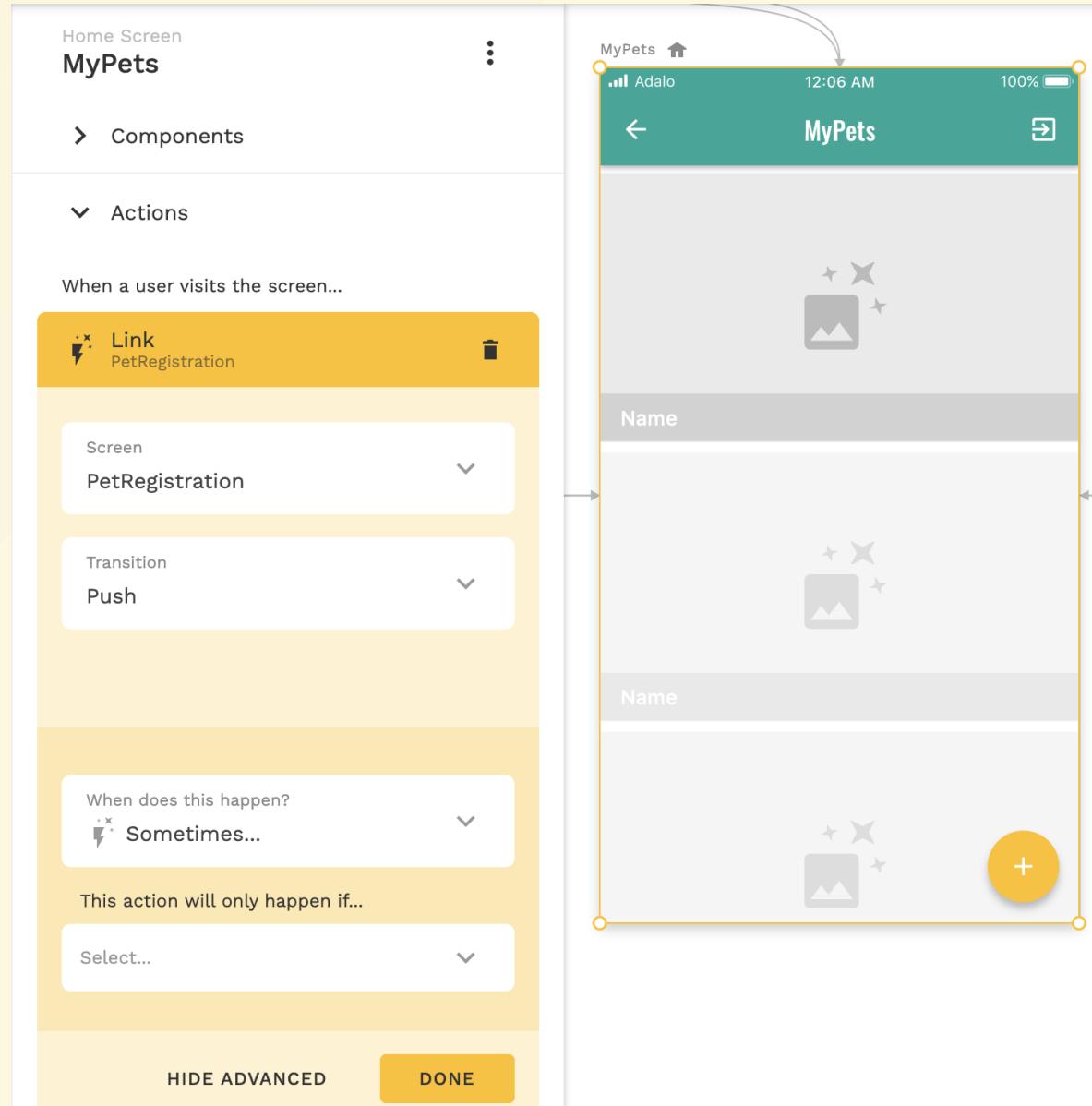
Preview機能で確認しましょう。

追加したアイコンをクリックするとログアウトし、ログイン画面に遷移するようになりました。

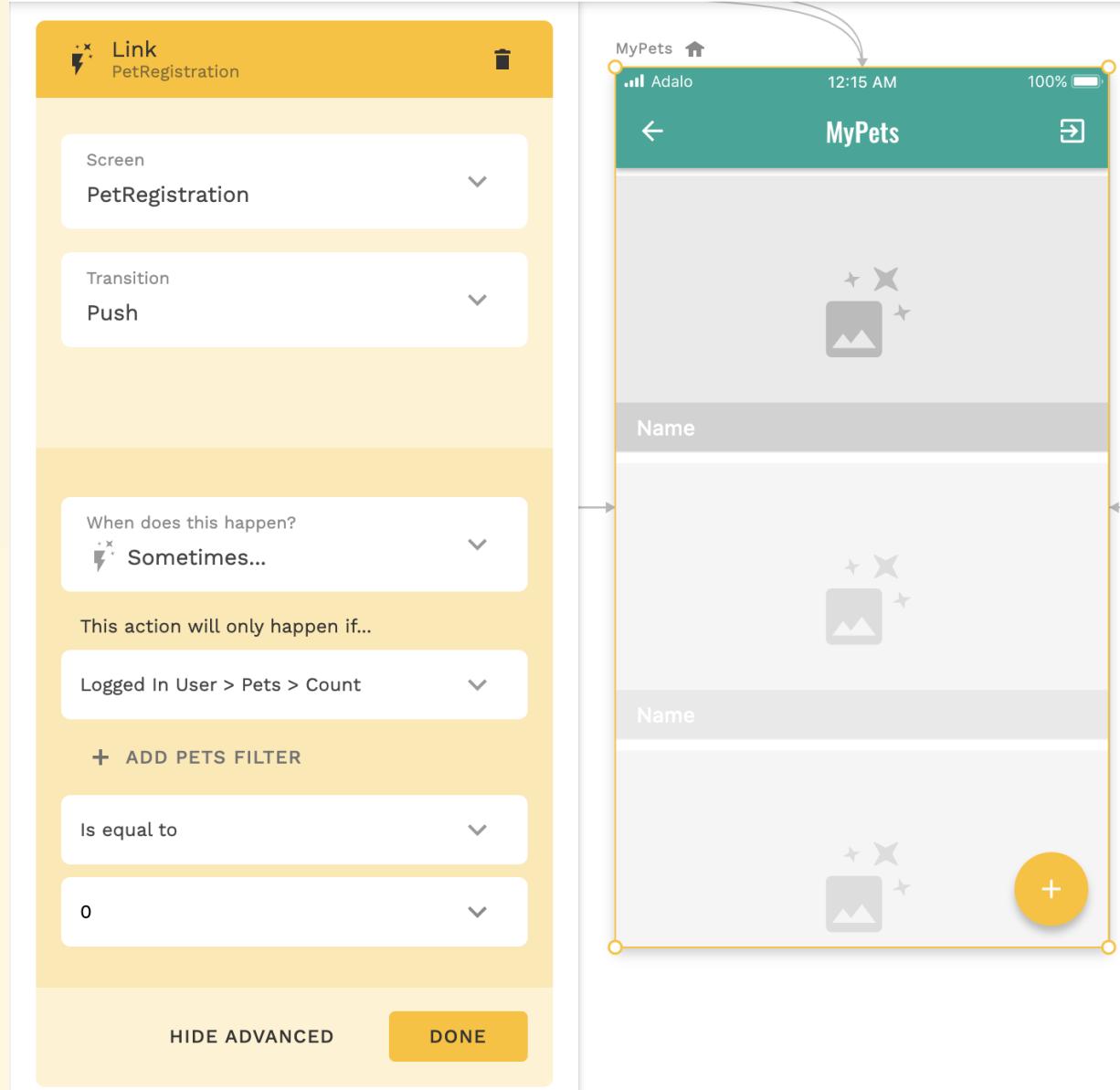
Actionの実行条件設定

ペットが未登録の場合にペット一覧画面を表示したらペット登録画面へ遷移させます。

- ペット一覧画面でActions > ADD ACTION > Link > Pet Registrationを選択
- SHOW ADVANCEDをクリックし、When does this happen?をSometimesに変更



- This action will only happen if...でMore > Logged In User's > Pets' > Count を選択
- Is equal to の下の数値を0に変更



Preview機能で確認しましょう。

新しいユーザーでSignupすると、ペット一覧画面からペット登録画面に遷移します。

選択式入力フォーム

ペットの情報に性別を追加し、入力フォームで選択式入力ができるようになります。

選択式入力フォームで使う選択肢は、データベースにCollectionを追加してレコードとして用意します。

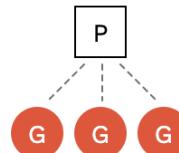
- データベースにGenders Collectionを追加(PropertyはデフォルトのままでOK)
- 0 Records > ADD GENDERとクリックし、MaleとFemaleの2つのRecordを追加

Genders				+ ADD GENDER				
	Name	Created	Updated					
<input type="checkbox"/>	Female	8 minutes ago	8 minutes ago					
<input type="checkbox"/>	Male	8 minutes ago	8 minutes ago					

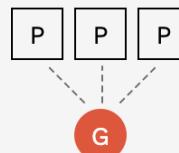
- Genders CollectionにPets Collectionとの1対多のRelationshipを追加する
 - 1匹のペットは1つの性別を持ち、1つの性別は複数のペットに対して設定されるため

How are Pets & Genders related?

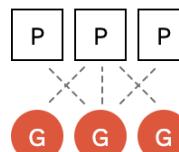
P = Pets G = Genders



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

- ペット登録画面のフォームを選択
- Fields > ADD VISIBLE FIELD > Genderを選択

The image displays two screenshots illustrating the configuration and preview of a form field.

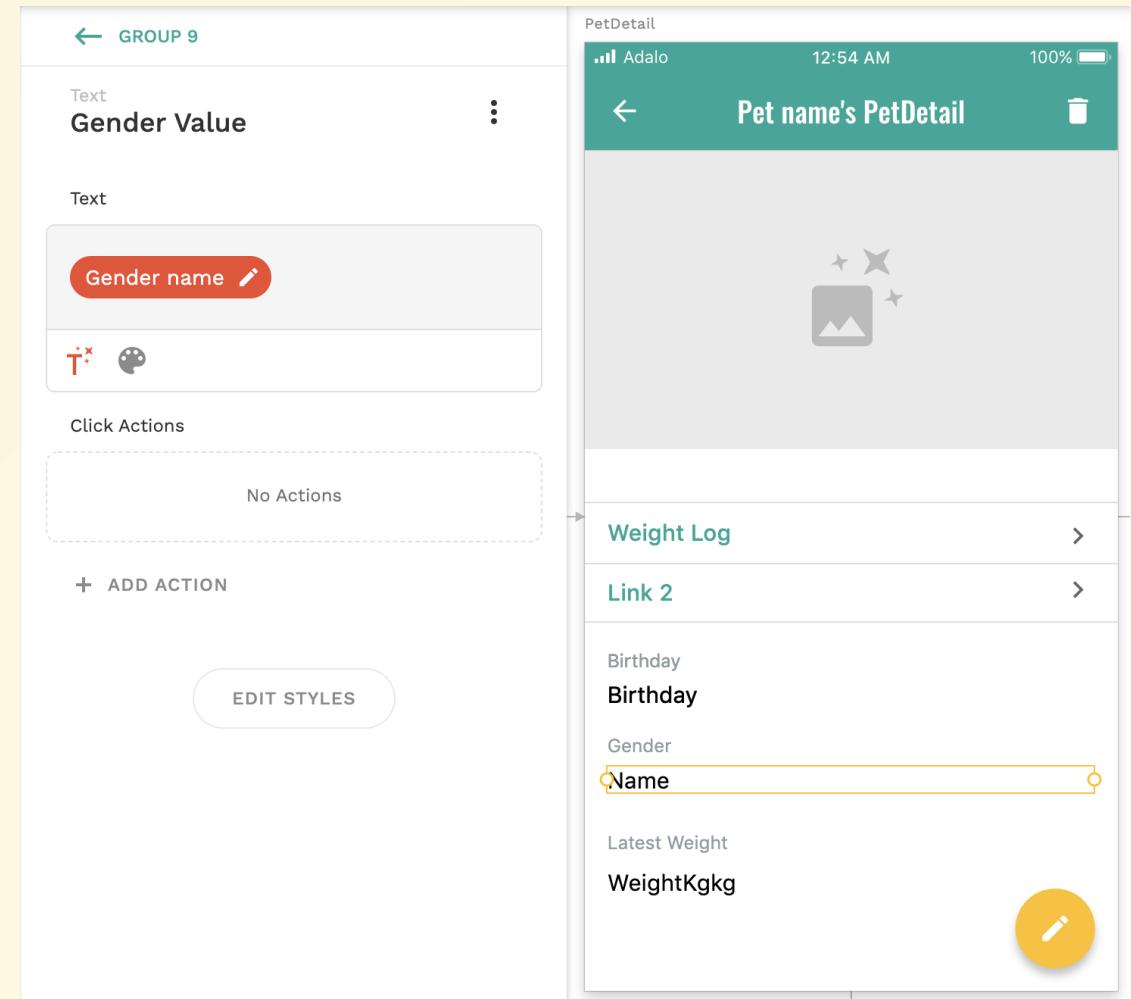
Builder Interface (Left):

- The top section shows the "Fields" configuration for a "Form Fields" section, listing "Name", "Image", "Birthday", and "Gender".
- The "Gender" field is highlighted with a green background, indicating it is selected for configuration.
- Below the field, settings include:
 - Label:** Gender
 - Placeholder:** Select gender...
 - + ADD FILTER**
 - Sorting:** Select...
 - Required Error Text:** This field is required.
- A "DONE" button is at the bottom right of the configuration panel.
- A "+ ADD VISIBLE FIELD" button is located at the bottom left of the configuration panel.

Mobile Application Preview (Right):

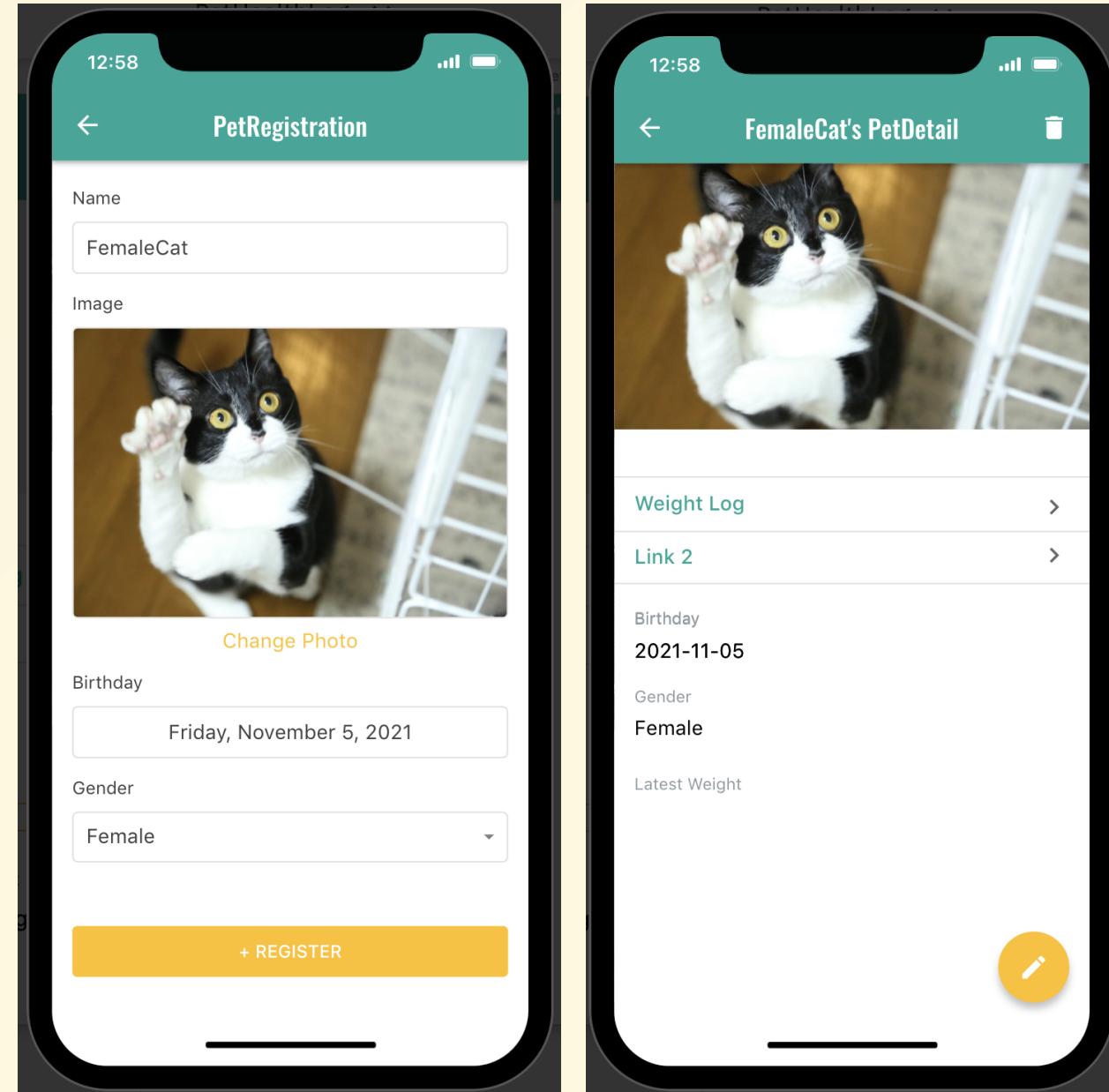
- The app title is "PetRegistration".
- The screen shows the "PetRegistration" form with fields:
 - Name: Enter name...
 - Image: Choose Photo
 - Birthday: Thursday, May 4th, 2019
 - Gender: Select gender...
- A large yellow "REGISTER" button is at the bottom right.
- An arrow points from the "Gender" field in the builder interface to its corresponding field in the mobile app preview.

- ペット詳細画面に性別の表示欄を追加



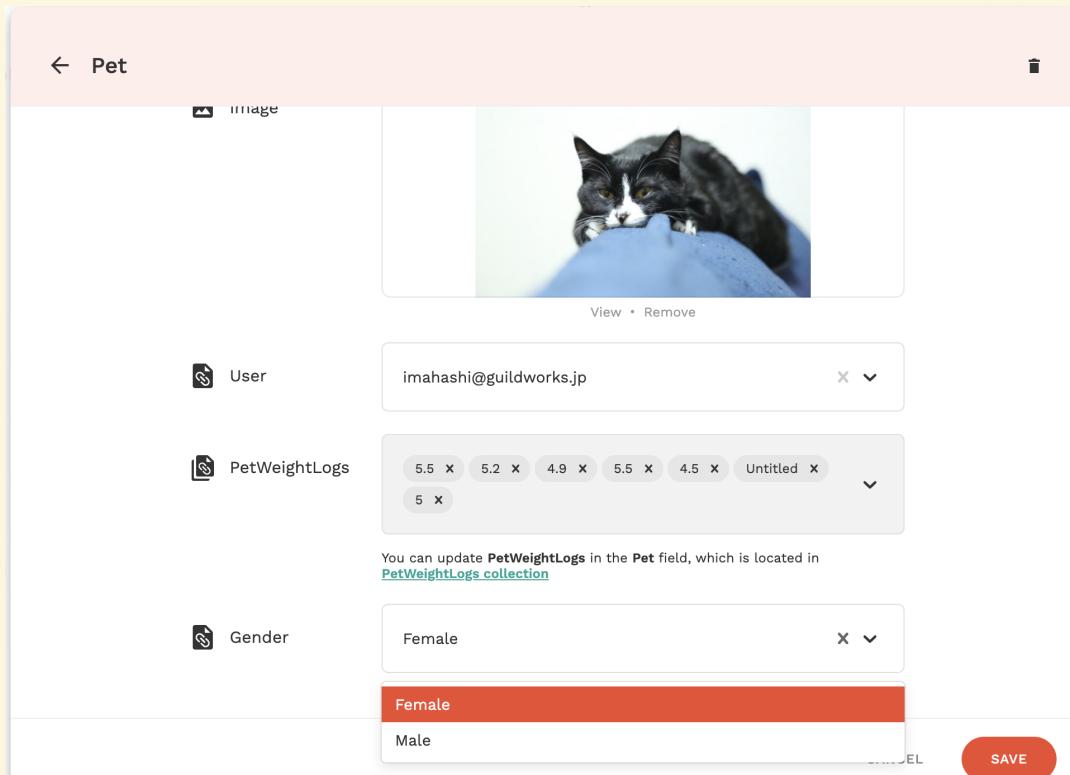
Preview機能で確認しましょう。

- ・ペット登録画面で性別が選択できます。
- ・選択した性別がペット詳細画面に表示されます。



補足

- 性別を選択できるようになる前に登録したペットで空欄が表示されることが気になるようであれば、Pet CollectionのRecordを表示してそのペットをクリックし、性別を手動で設定してもOKです。



参考

複数選択式の入力フォーム
は、Marketplaceの
MultiselectDropdownを使って
作れます。

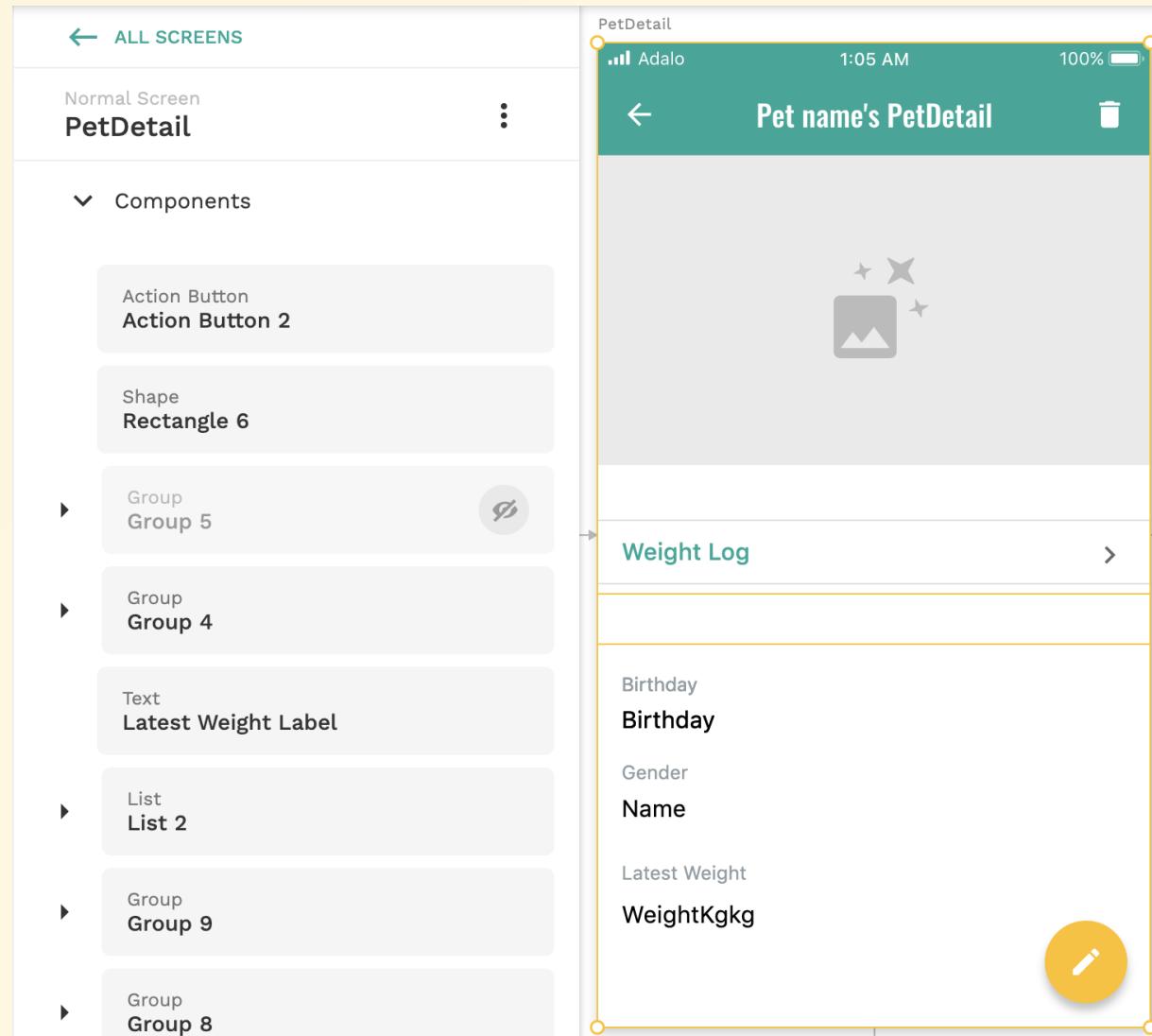
必要になったら、試してみて
ください。

The screenshot shows a card for the "Multiselect Dropdown" component from the Marketplace. At the top left is the Marketplace logo, which consists of a green square with four white squares inside. To its right is the text "Marketplace". Below this is a circular icon containing a black icon of a dropdown menu with multiple horizontal bars. To the right of the icon is the component's name, "Multiselect Dropdown", followed by a brief description: "Select multiple items from a dynamically-generated list." Below this description is a large preview window showing a user interface element. This element features a search bar at the top, followed by a list of five items. Each item has a checkbox next to it; the first and third items have checked boxes, while the others are empty. The preview is enclosed in a light gray rounded rectangle. At the bottom left of the card is the text "By Conroy Whitney" with a small profile icon. At the bottom right is a green button with the word "INSTALLED" in white capital letters.

コンポーネントの非表示設定

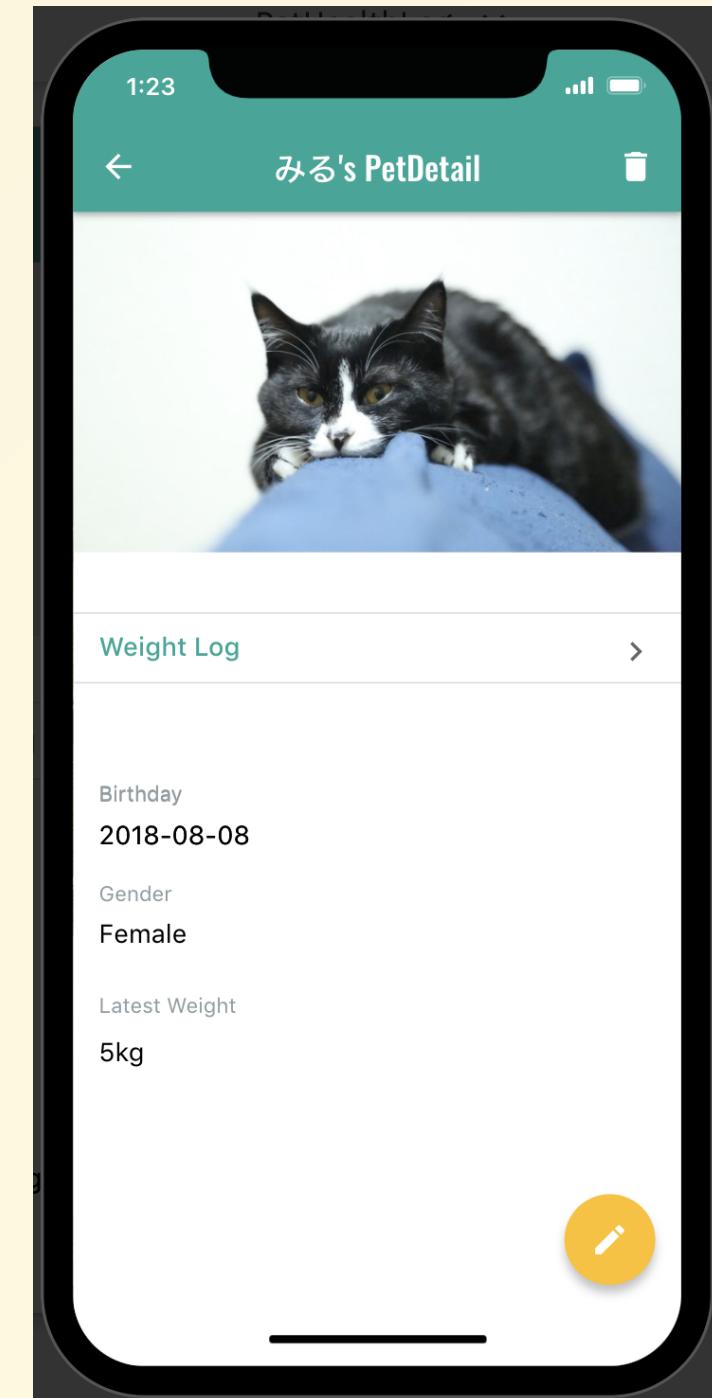
表示しないコンポーネントを削除せずに残しておくことができます。

- ペット詳細画面のコンポーネント一覧で、Link 2を含むGroupにマウスオーバーして、右側の目のアイコンをクリック



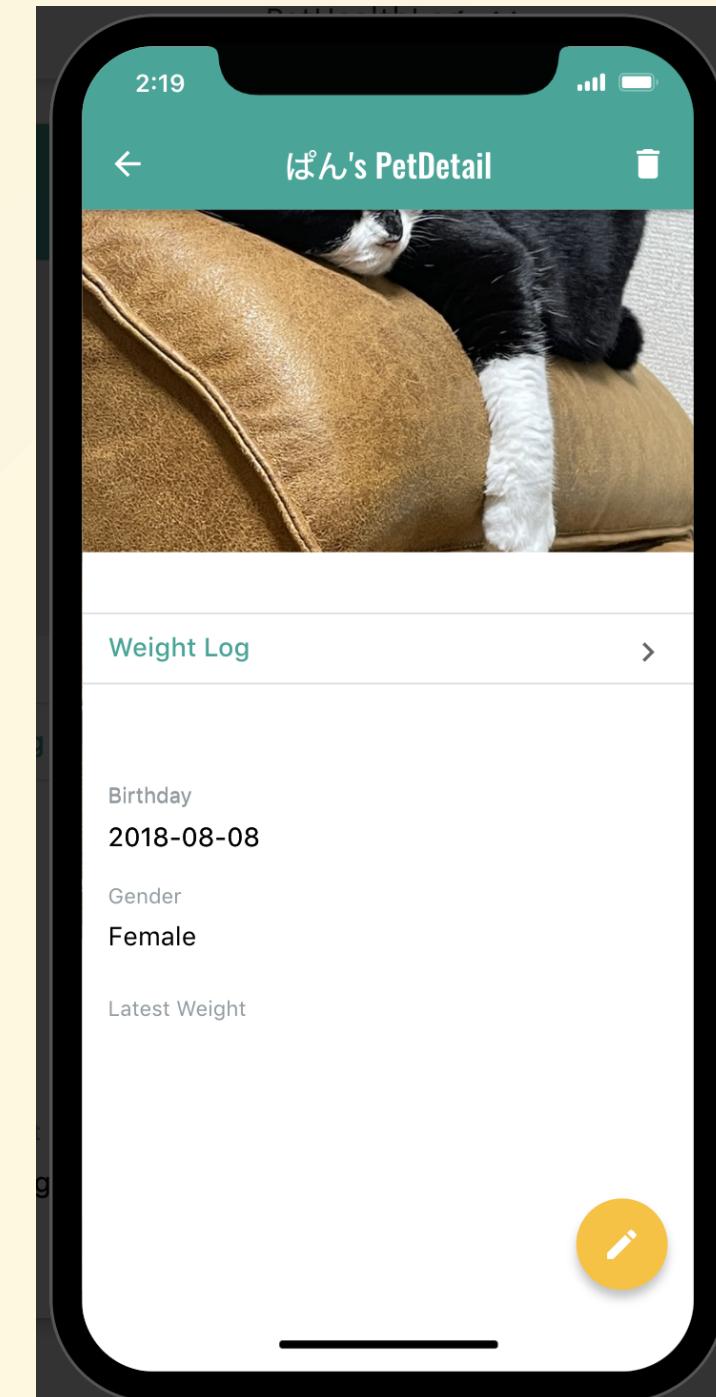
Preview機能で確認すると、
Link 2が消えています。

もう一度目のアイコンをクリ
ックすれば、再度表示させら
れます。

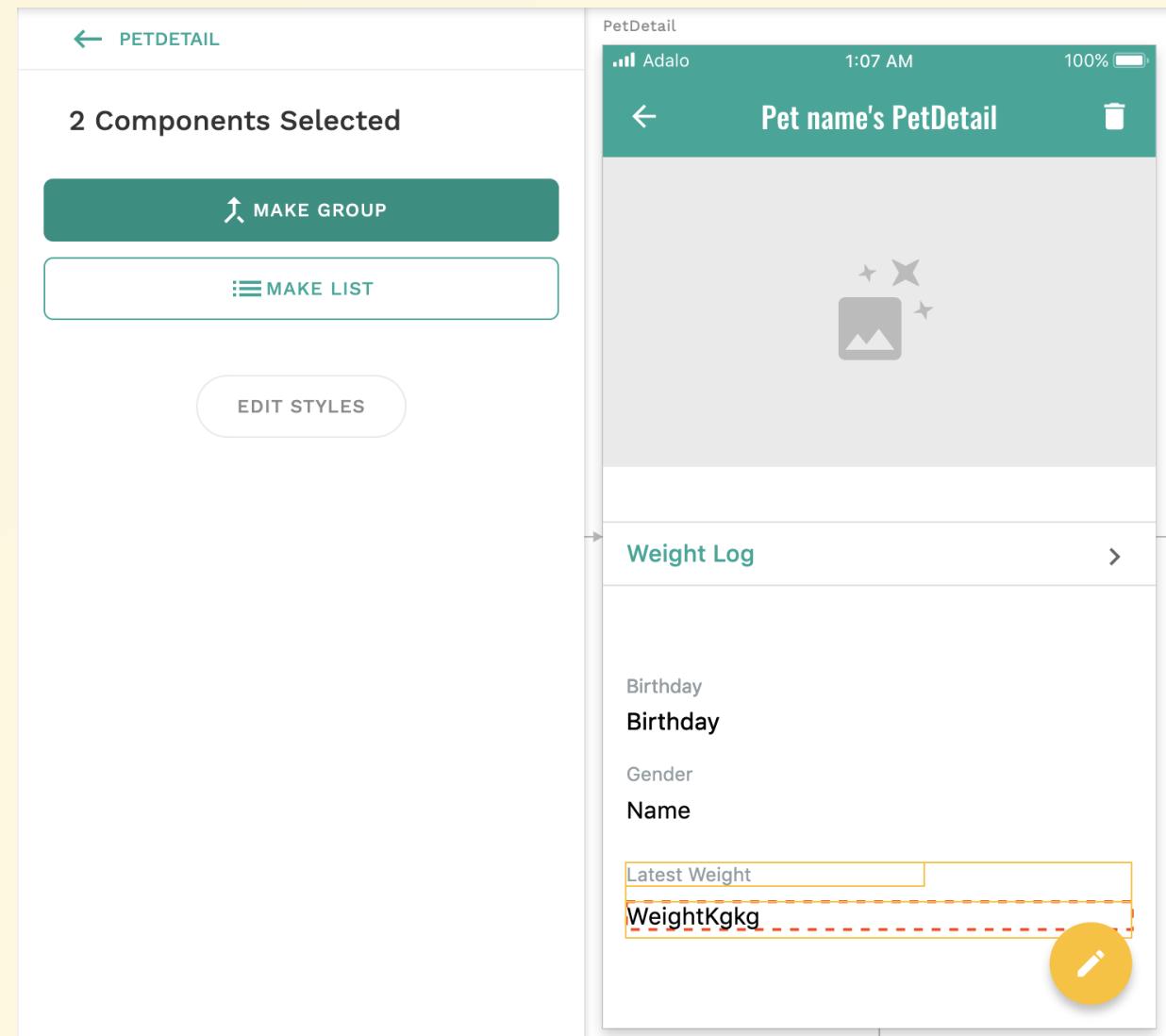


条件によってコンポーネントの表示or非表示を切り替える

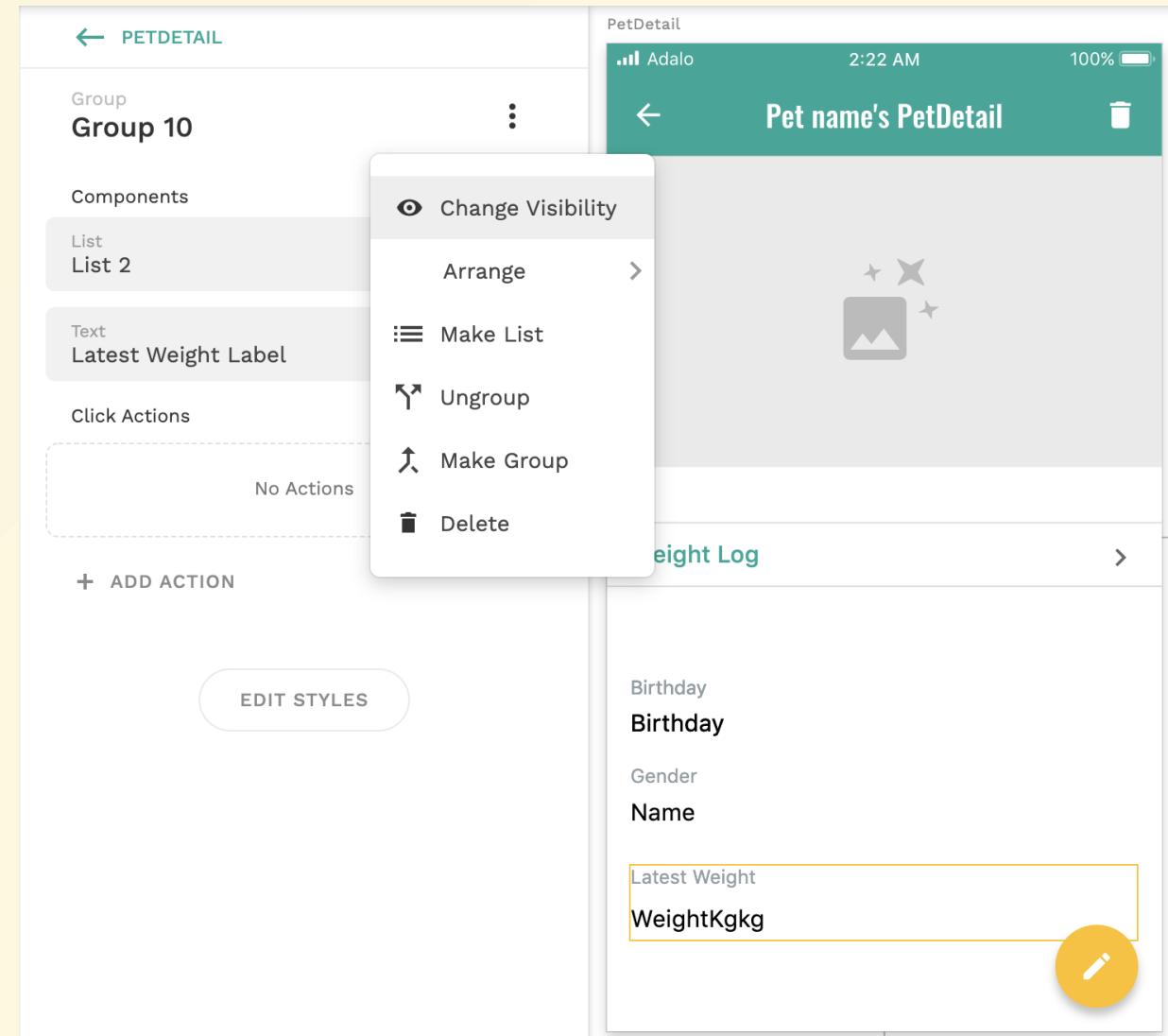
体重が未登録であれば、ペット詳細画面のLatest Weightは非表示にします。



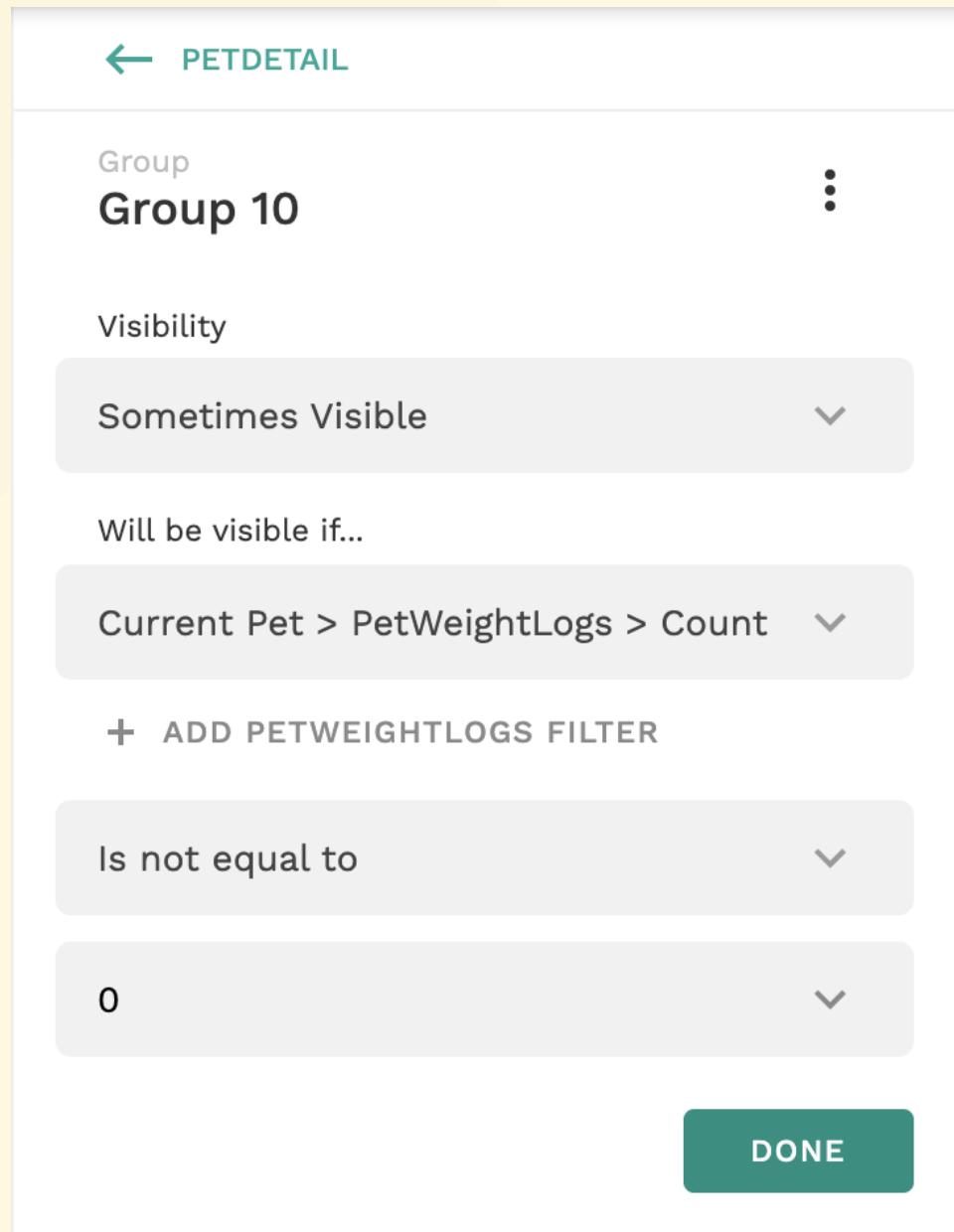
- Latest Weightというラベルとその値を選択し、グループ化します



- Change Visibilityを選択します

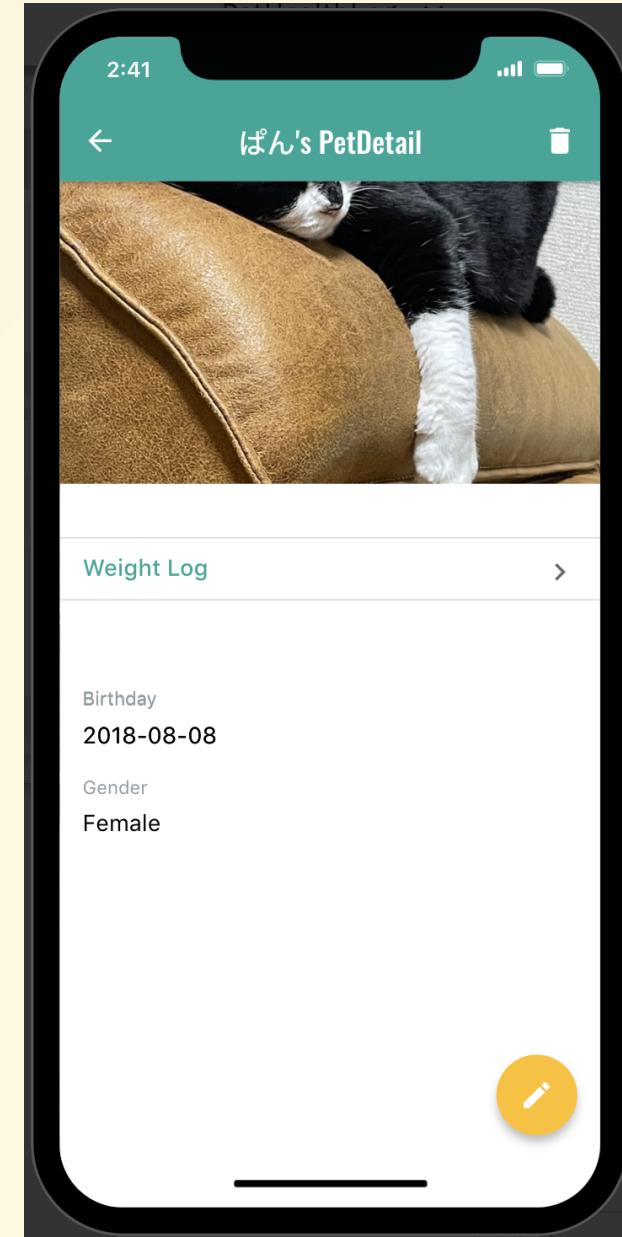


- VisibilityをSometimes Visibleに変更
- Will be visible if...でCurrent Pet > PetWeightLogs > Countを選択
- Is not equal to 0と設定



Preview機能で確認すると、体重が未登録であれば、Latest Weightが非表示になりました。

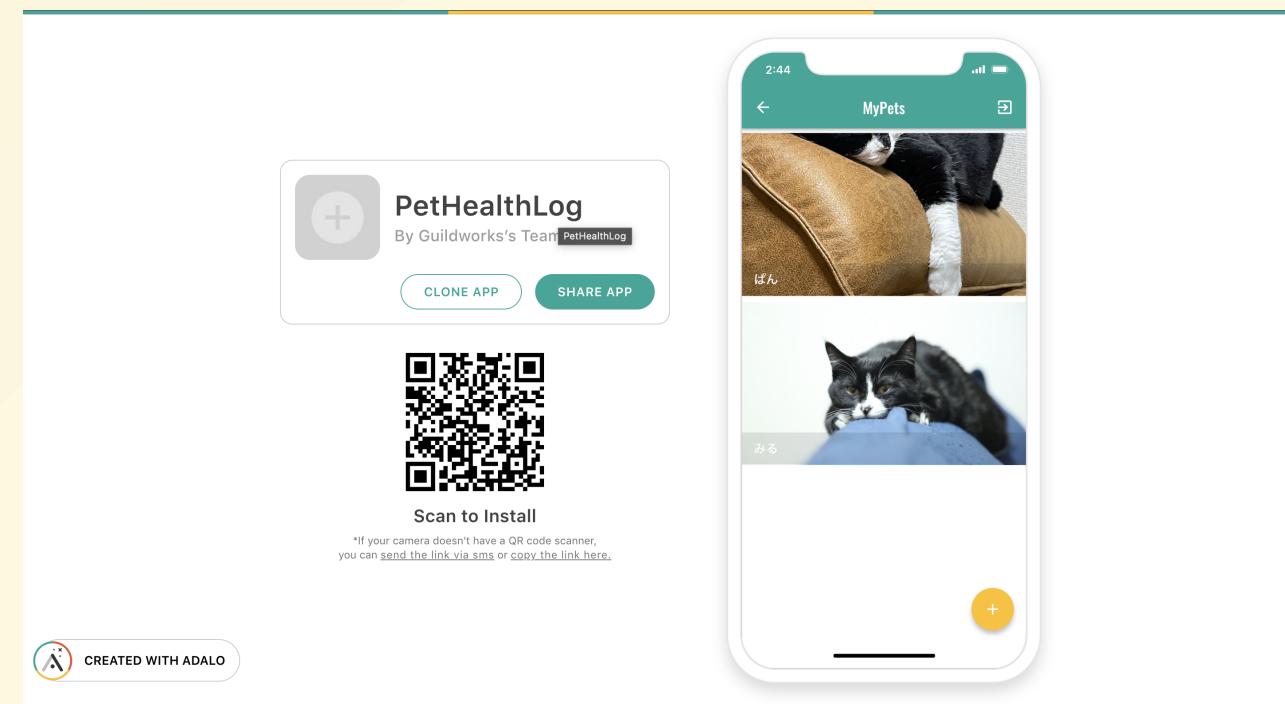
サンプルアプリの改善は一旦ここまでとします。



クローン用URL

- 以下のURLからここまで
の作業を反映したアプリをク
ローンできますので、答
え合わせに使ってください。

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



演習

Development Phaseのチームメンバーと一緒に取り組みましょう。

1. 以下の機能を持つチームメンバー管理アプリを開発してください。

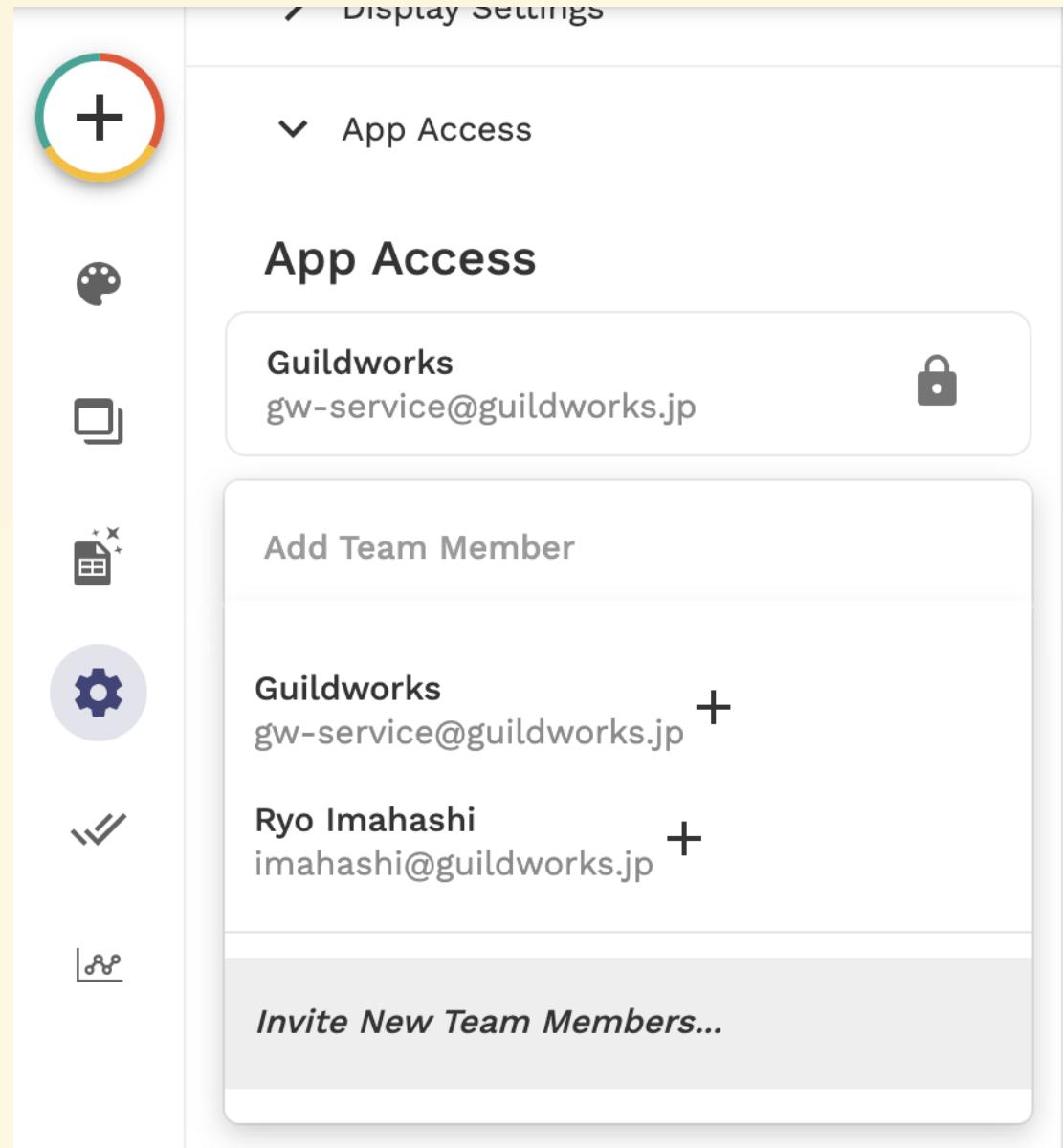
- チームメンバーの登録
- メンバー一覧の表示
- メンバー詳細の表示
- メンバー情報の更新
- メンバー削除
- 自分たちで考えたオリジナル機能(いくつあっても良い)

2. (演習1で時間が余れば)アプリを1つ自由に開発してください。

- * 今日のレクチャーに自分以外のチームメンバーが参加していない場合は、他のチームに加わってください。
- * リモート参加の人は、チームメンバーとZoomを繋いで会話や画面共有をしながら作業に加わってください。
- * 最後は全チームに発表していただきます。
- * アプリが使えるようになったら、SlackでURLを共有して、みんなに見てもらいましょう。

演習についての補足

- チームメンバーと一つのアプリを共同編集するには、
Settings > AppAccess > Add Team Member > Invite New Team Memberを選択してチームメンバーのメールアドレスを入力しましょう。



- ※他のメンバーと一緒に一つのアプリを編集する際、他の人が行った編集は自分の画面にリアルタイムで反映されません。(リロードが必要なようです)
- ※同時に複数人が同じ画面を編集すると、先に行われた編集が後から行われた編集に上書きされてしまいます。それぞれが別の画面を編集するか、みんなで同じPCを見ながら一緒に編集しましょう。

参考

- クローンできるAdaloのアプリが公開されているので、やりたいことに近いものがないか探してみると良いかもしれません。
 - App Templates
<https://www.adalo.com/app-templates>
 - UI & Functional Kits
<https://www.adalo.com/cloneable-kits>

クローンできるアプリの例

- Eventカレンダー <https://www.adalo.com/cloneables/event-calendar>
- スワイプで回答するクイズ
<https://www.adalo.com/cloneables/quiz-app>
- SNSのフォロー機能 <https://www.adalo.com/cloneables/follow-function>
- Facebookのクローン <https://www.adalo.com/cloneables/facebook-clone>
- ブログアプリ <https://www.adalo.com/cloneables/minimal-blog-app>
- 商品販売アプリ <https://www.adalo.com/cloneables/ecommerce-app>

参考

- Adaloだけで実現できないことがある場合、外部サービスと連携することでそれを実現できるかもしれません。
- Zapierというサービスを使えば、案内に従って操作することで、簡単にAdaloと外部サービスを連携させられます。興味があれば、試してみてください。

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

ZapierでAdaloと連携させることが可能な外部サービスの例

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

演習結果の発表

チームごとに演習で作ったアプリについて発表してください。

まとめ

- ・ 今回のレクチャーでは、データベースを設計し、アプリからそのデータベースに対してCRUD操作を行えるようにしました。
- ・ また、アプリの改善をしながら、Adaloの機能をいくつか紹介しました。
- ・ 以下のような、外部サービスとの連携に関するAdaloの機能はまだ紹介していないので、5回目のレクチャーかDevelopment Phaseで時間が取れたら、改めて紹介したいと思います。
 - Custom Action(Adaloから外部サービスのAPIを呼び出す機能)
 - External Collections(外部サービスのAPIから取得したデータをAdaloのコレクションとして扱う機能)

- ここまで内容を踏まえて、Development Phaseで自分たちが作りたいアプリがAdaloで実現できそうかは、チームで考えてみると良いと思います。
- 次回はノーコードツールBubbleのレクチャーです。お楽しみに！

以上です！

お疲れさまでした！