Interactive Stack with Text Editor Flutter Widget description

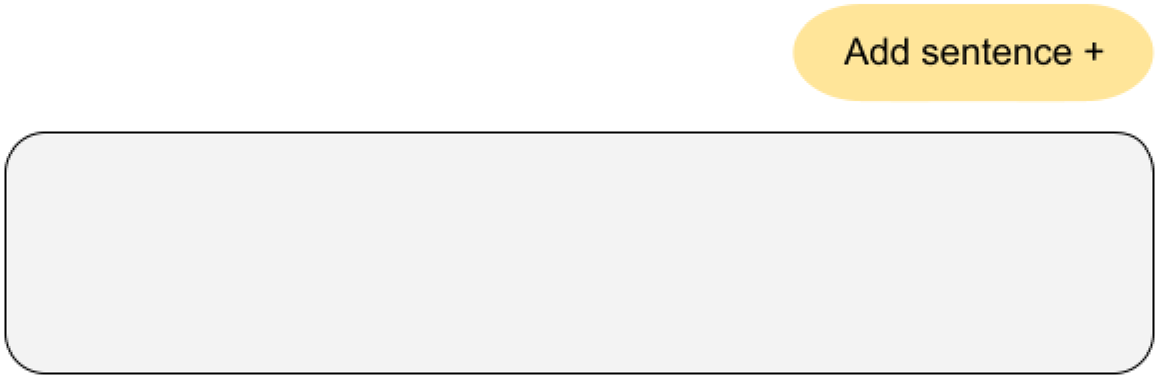
Background

This document describes a Flutter widget that allows users to create lists of sentences with highlighting capabilities. This widget builds on top of the previous one in which users where able to type in a TextField and keywords from a JSON list were highlighted interactively. In this case, users can add/edit/delete sentences that are kept in a list on top of the TextField editor, which is shared to edit existing sentences or input the text for new ones.

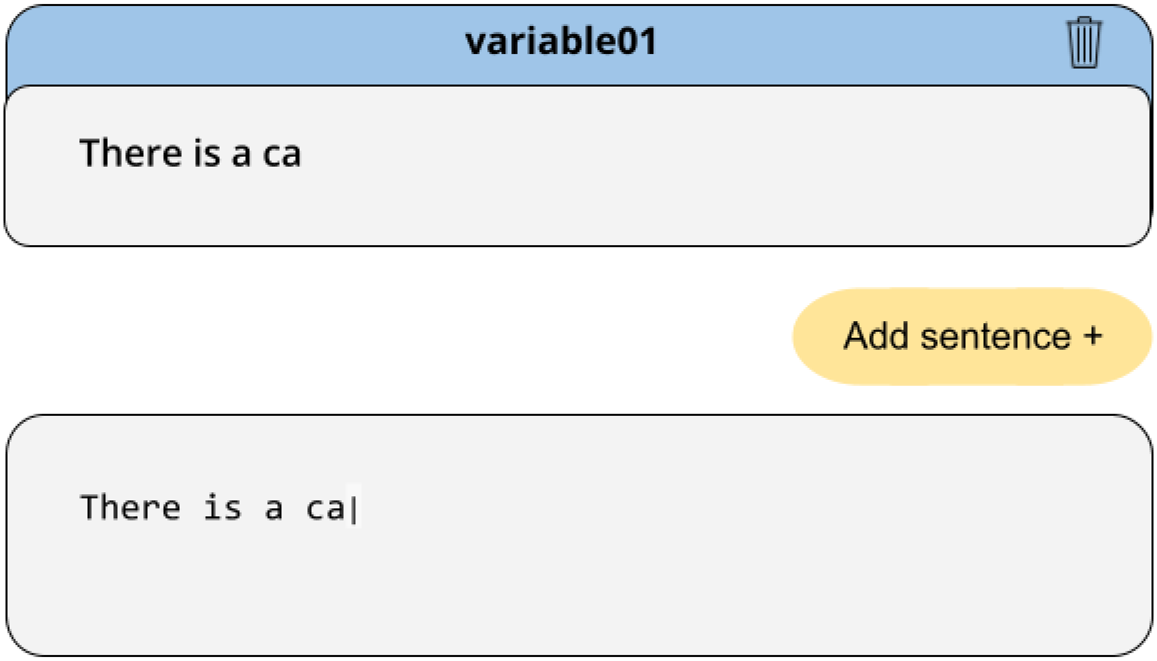
As in the previous example, the list of keywords is defined in a separate JSON document. Typed keywords are rendered as a clickable button. When the button is clicked, users can choose from a list of options also defined in the JSON document.

Example

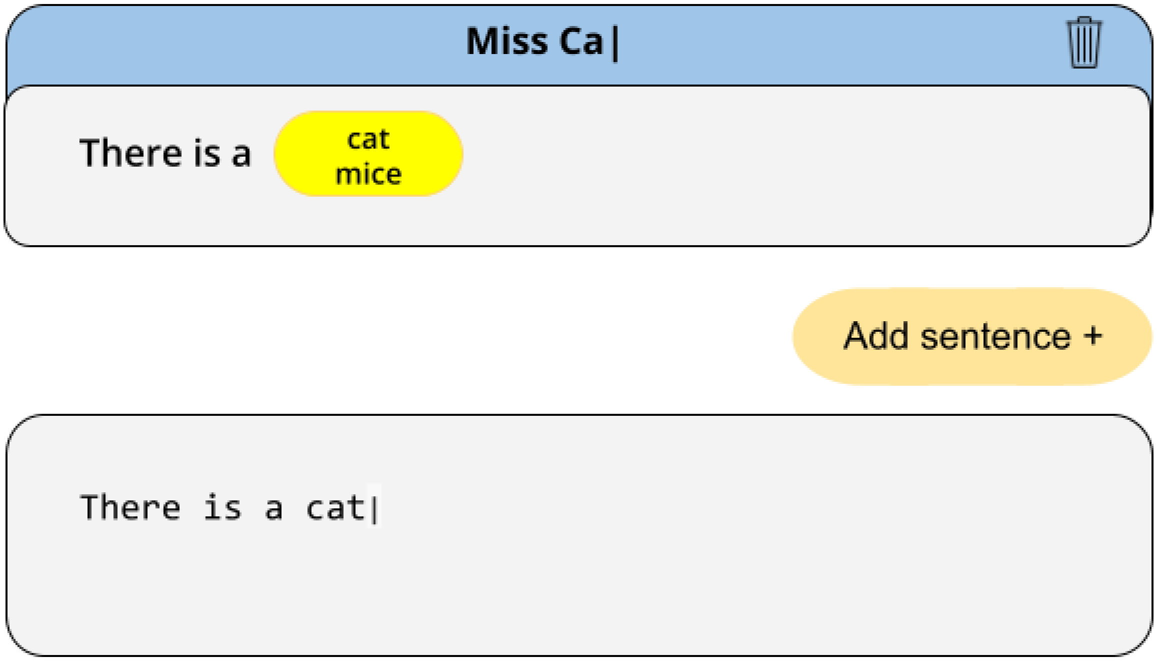
To better explain the desired functionality, we are going to see an example of how the widget would work. Initially there is a TextField and a button for adding a sentence.



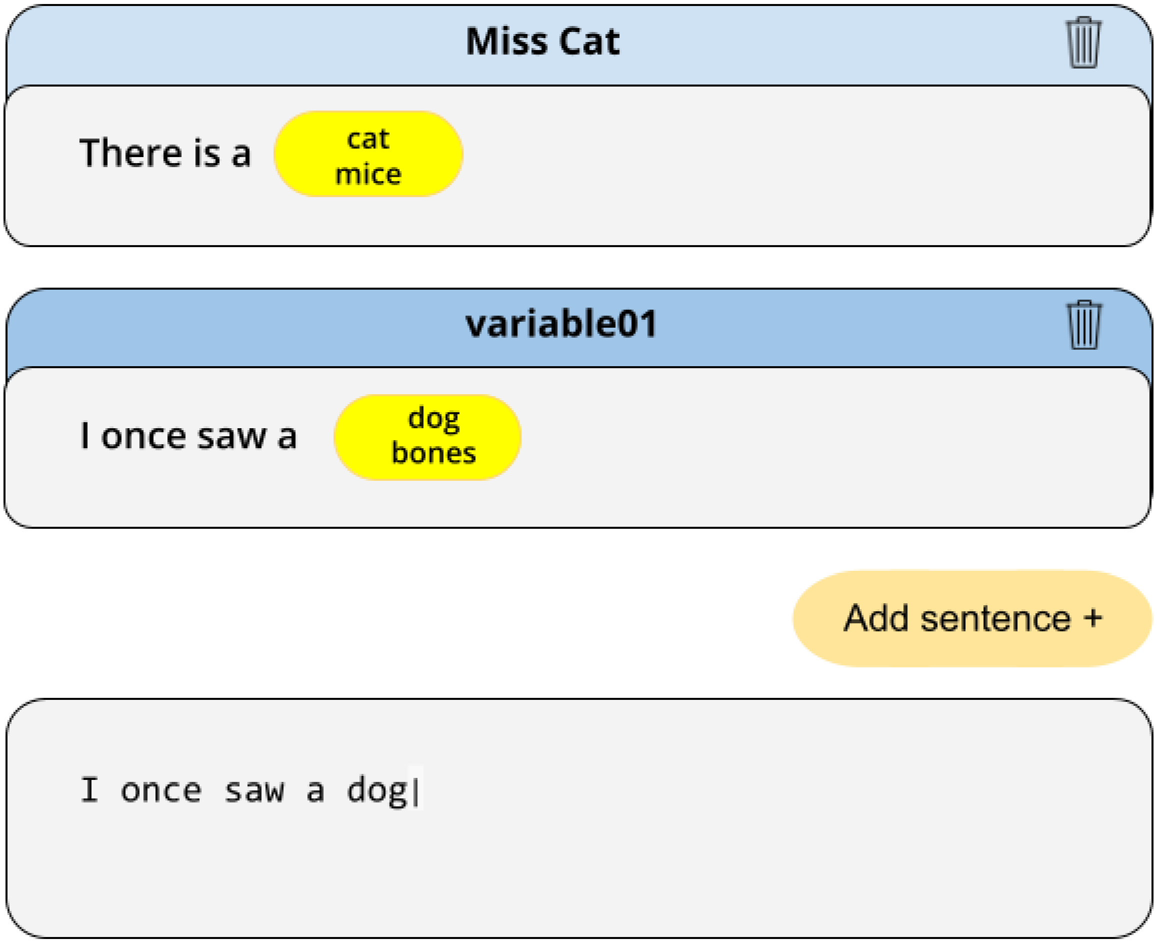
The TextField is initially deactivated and when a user clicks on the button a new widget gets added on top of the button. The TextField is now connected to this widget and the text replicated. The same functionality for highlighting text applies as in the previous widget.



Apart from the functionality to sync between the Textfield and the widget's text, the widget gets a default name that can be edited later. Also, the widget has an icon that the user can click for deleting any time.



Users can add new widgets which stack in a list. Only one widget can be active (connected to the TextField) at once. The active widget is highlighted and the other widgets go grey.



Miss Cat

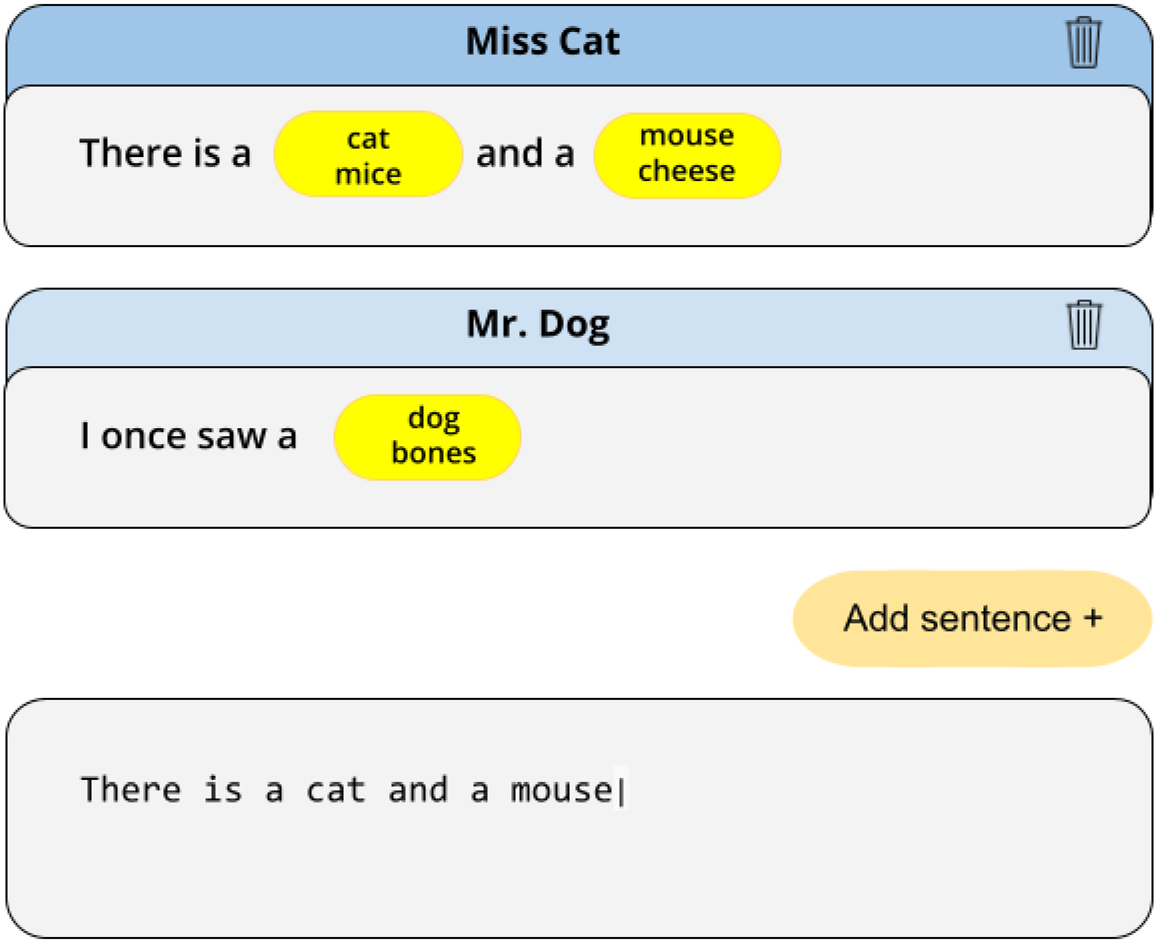
There is a cat
mice

variable01

I once saw a dog
bones

```
I once saw a dog
```

Users can add new widgets that are added to the lower part of the list. If a user clicks on any non active widgets in the list, this becomes active and the TextField replicates the text for making edits.



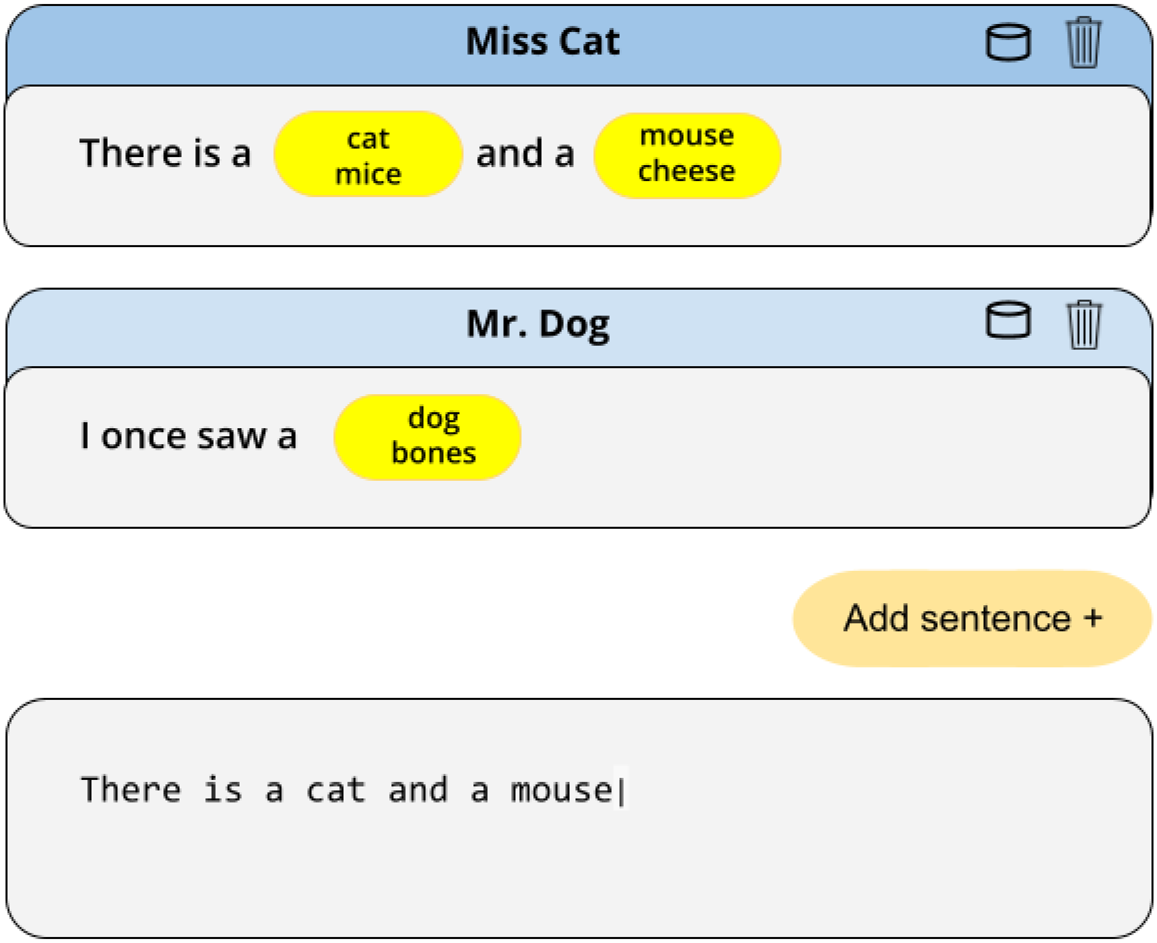
There is a cat and a mouse

Extras

This are extra functionality that can be made part of the same deliverable or done at a larter starge as a continuation.

1. Keywords are stored in a Firebase Firestore database. Instead of using a local JSON document, the list of animals is stored as a collection using the Firestore database backend. In this case, the widget needs to dynamically load the list of keywords (animals) from this database. We would provide a project and create this database giving you a user to connect to this database.
2. Give users the ability to store the sentences as new entries in the database adding a new icon to store next to the delete one. The contents of the new

document added to the database will be the name given to the widget and the preferences selected for the animals used in the sentence. For example, is a user clicks on the save icon of Miss Cat:



There is a cat mice and a mouse cheese

I once saw a dog bones

There is a cat and a mouse|

The following document will get added to the database collection:

```
{
   "species" : "Miss Cat",
   "foods": ["mice", "cheese"]
}
```

Note: I know this examples don't make much sense from the point of view of storing animal information related to sentences. In reality, we want to use this functionality to build more complex mathematical expressions and define variables. This is a very simplified scenario but, if this goes well, there will be new deliverables and things will make more sense.