

Exercise 1

1)

```
if (index >= _suggestions.length) {  
  _suggestions.addAll(generateWordPairs().take(10));  
}
```

If we were to remove the lines, we would receive a `RangeError`, as the next code line tries to build more tiles by accessing the index in `_suggestions`, which is limited to 20 now.

2)

We can use `ListView.separated()` to achieve that, however it assumes that the list is infinite, with a fixed item count, therefore the function `ListView.builder` is preferred for our infinite list.

3)

The `setState` is due to the fact that we are changing the state of tile (adding/removing from favorites, thus coloring the heart icon red/white) , and we need to update the UI, so `setState` triggers a call to the `build()` method for the `State` object, which results in an update to the UI.

Exercise 2

1)

I used the `Push()` method of `Navigator.of(context)` as follows:

```
void _loginScreen() {  
  Navigator.of(context).push(  
    MaterialPageRoute<void>(builder: (BuildContext context) {
```

Another way to achieve that would be to implement navigation right from the main screen, calling `Push()` right from the login icon's `onPressed` in the app bar from the build widget of the main screen.

2)

To show the `snackbar`, I used

```
ScaffoldFeatureController<SnackBar, SnackBarClosedReason>  
showSnackBar(SnackBar snackbar)
```

And it requires `Scaffold` widget (or specifically in our case, the “`Build`” widget, to provide child relation to the `Scaffold`, if I understood correctly).