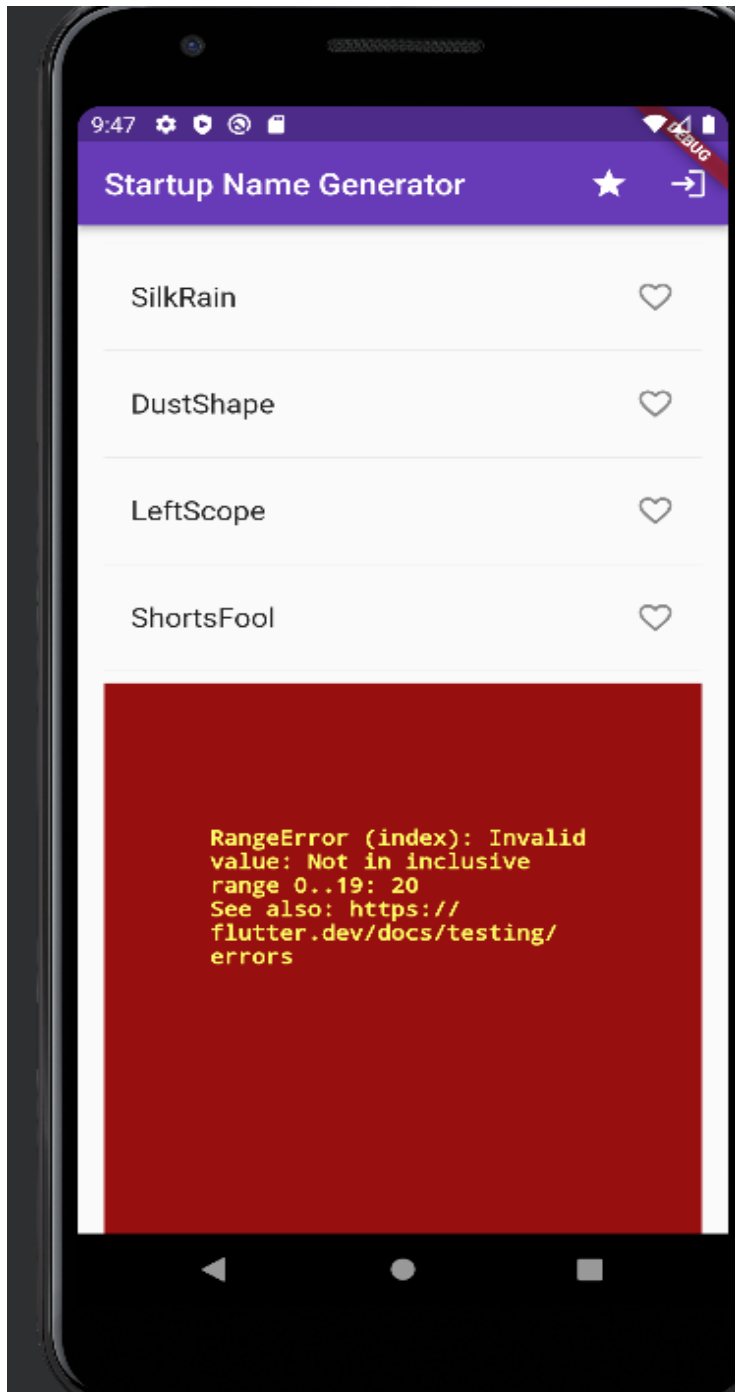


Part1:

1. The lines that enable infinite-scrolling are:

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

If we remove these lines, the index will get bigger and when it hits the number of pairs that have been generated we will get this error because there are no more pairs to show :



2. An additional option for constructing a list is a default constructor that receives an explicit `list<Widget>` of children. This method eagerly constructs the entire list, not only the visible items like the option used in our app, but this constructor is only appropriate for list views with a fixed number of children . this method is preferred in this case because it has better complitaion time.
3. When we tap on the “favourite” icon, we remove/ add new items (paired-words) to the saved list, therefore the internal state must be changed/updated. So we use “setState” to notify that RandomWords widget’s state has changed which cause the framework to call `_RandomWordsState’s build()` method to update the widget.

## Part2:

1. `MaterialApp` widget wraps several widgets that are commonly required for Material Design applications. Three of its properties are:
  1. It configures the Navigator which is used to traverse the different routes of the app.
  2. It defines “ home” The widget for the default route of the app .
  3. It can set a shared theme across the entire app
2. If the `Dismissible` is a list item, the key is used to distinguish and identify it from the other items and remove or update its node in the app’s widget tree.