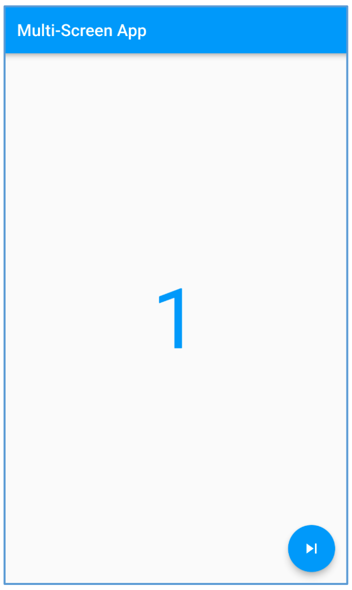
CROSS-PLATFORM MOBILE APP DEVELOPMENT

(503107)

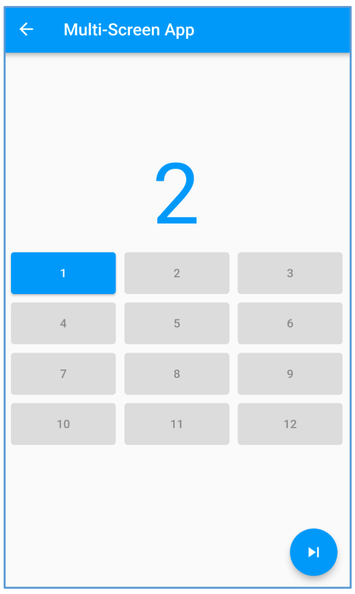
LAB 6

EXERCISE 1

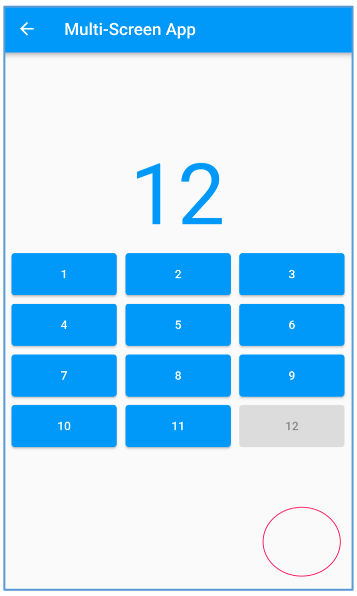
Design the MyApp widget in such a way that it can call itself to open a new screen every time the user clicks the floating action button. In the example image below, the maximum number of screens allowed is 12, but the user can completely customize it by changing the parameters when initializing the MyApp widget in the MaterialApp.



Step 1



Step 2



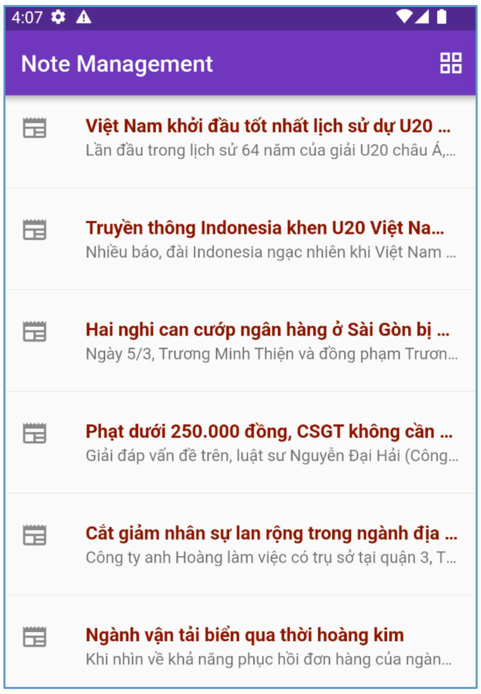
The last step

From the second step onwards, the screen will display a grid buttons with an amount equal to the total number of screens. In addition to using the automatically supported back button, users can also click on any active button to return to the corresponding screen. For example, while on screen 12, if the user clicks on button 8, the user will be taken to screen 8 and cannot go back to screen 9-12. When the user are on screen 8 and they press the back button, the user will be taken to screen 7. In addition, you must use only one class - MyApp widget with a single Scaffold widget inside. Do not make each screen a separate widget class.

```
void main() {
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyApp(totalScreen: 12,),
    ),
  );
}
class MyApp extends StatelessWidget {...}
```

EXERCISE 2

Given two files NoteListScreen.dart and AddNoteScreen.dart containing two custom widgets which are equivalent to screen showing the list of notes and creating/editing a note screen. Please work on these files to complete the basic note management application according to the specific requirements below.



Việt Nam khởi đầu tốt nhất lịch sử dự U20 ...
Lần đầu trong lịch sử 64 năm của giải U20 châu Á,...

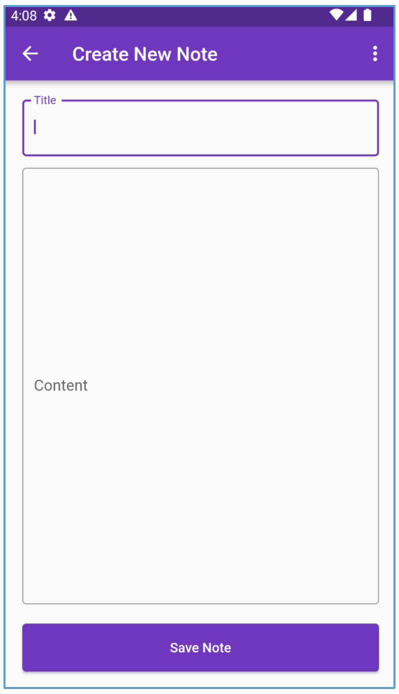
Truyền thông Indonesia khen U20 Việt Na...
Nhiều báo, đài Indonesia ngạc nhiên khi Việt Nam ...

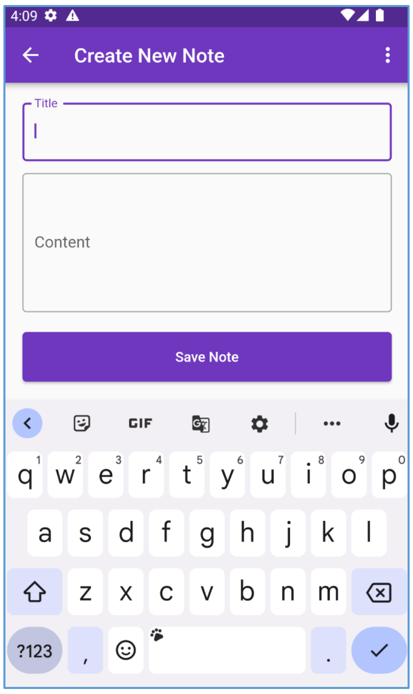
Hai nghi can cướp ngân hàng ở Sài Gòn bị ...
Ngày 5/3, Trương Minh Thiện và đồng phạm Trươn...

Phạt dưới 250.000 đồng, CSGT không cần ...
Giải đáp vấn đề trên, luật sư Nguyễn Đại Hải (Công...

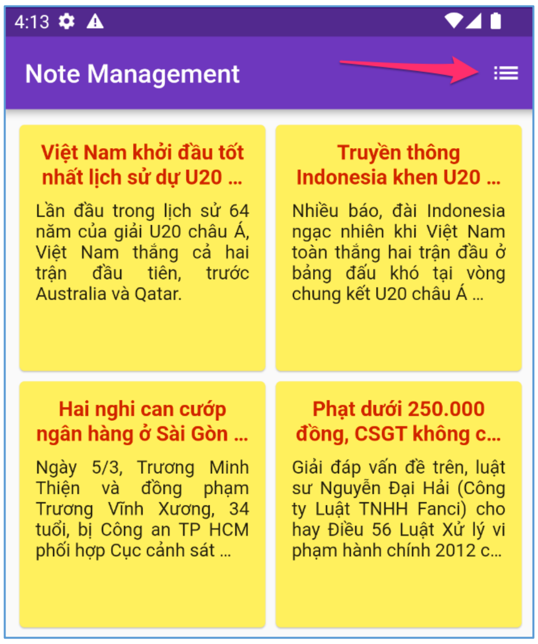
Cắt giảm nhân sự lan rộng trong ngành địa ...
Công ty anh Hoàng làm việc có trụ sở tại quận 3, T...

Ngành vận tải biển qua thời hoàng kim
Khi nhìn về khả năng phục hồi đơn hàng của ngàn...





In the main screen, when clicking on the Appbar action button in the upper right corner, users can change the way to display the note list between ListView and GridView. The **AddNoteScreen.dart** widget is used for both situations: adding a new note and updating an existing note. In each case, it is necessary to display the text of the title and button appropriately.

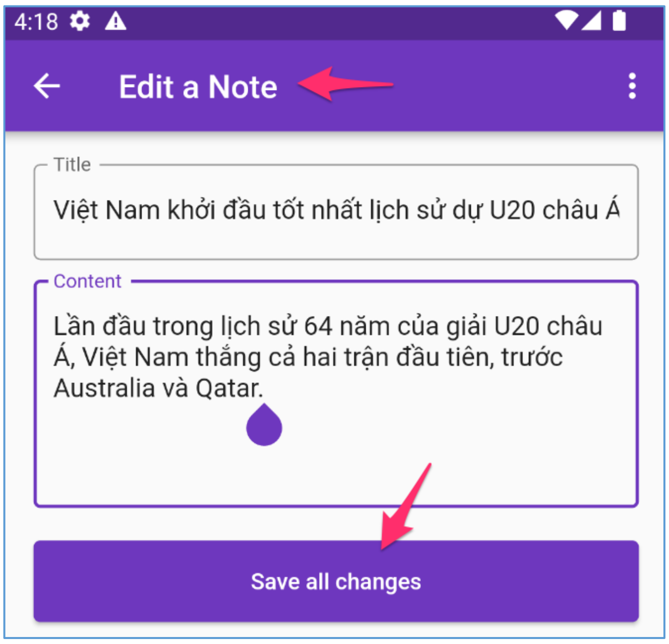


Việt Nam khởi đầu tốt nhất lịch sử dự U20 ...
Lần đầu trong lịch sử 64 năm của giải U20 châu Á, Việt Nam thắng cả hai trận đầu tiên, trước Australia và Qatar.

Truyền thông Indonesia khen U20 ...
Nhiều báo, đài Indonesia ngạc nhiên khi Việt Nam toàn thắng hai trận đầu ở bảng đấu khó tại vòng chung kết U20 châu Á ...

Hai nghi can cướp ngân hàng ở Sài Gòn ...
Ngày 5/3, Trương Minh Thiện và đồng phạm Trương Vĩnh Xương, 34 tuổi, bị Công an TP HCM phối hợp Cục cảnh sát ...

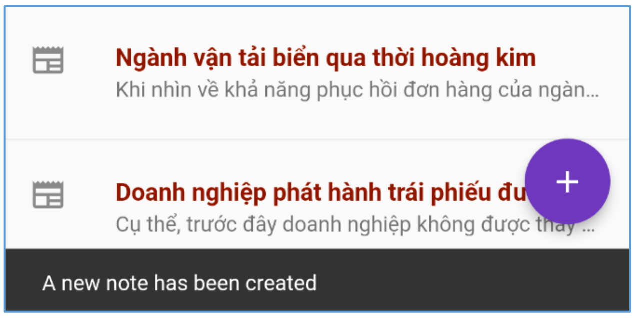
Phạt dưới 250.000 đồng, CSGT không c...
Giải đáp vấn đề trên, luật sư Nguyễn Đại Hải (Công ty Luật TNHH Fanci) cho hay Điều 56 Luật Xử lý vi phạm hành chính 2012 c...



Title: Việt Nam khởi đầu tốt nhất lịch sử dự U20 châu Á

Content: Lần đầu trong lịch sử 64 năm của giải U20 châu Á, Việt Nam thắng cả hai trận đầu tiên, trước Australia và Qatar.

To add a new note, the user clicks the Floating action button. After successfully adding or editing a note, the app will display a snackbar with the appropriate message for each case. If the user presses save while not actually making any changes, the snackbar mesasge will not be displayed.


Ngành vận tải biển qua thời hoàng kim
Khi nhìn về khả năng phục hồi đơn hàng của ngàn...

Doanh nghiệp phát hành trái phiếu đư
Cụ thể, trước đây doanh nghiệp không được thay ...

A new note has been created

EXERCISE 3

Create a mobile app that allows users to browse and purchase products. The app should consist of the following pages:

- **Home page**: This page should display a list of products that users can browse through. Each product should have a name, image, and price.
- **Product detail page**: When the user taps on a product from the home page, they should be taken to a product detail page that displays more information about the product, such as a detailed description, reviews, and additional images.
- **Cart page**: When the user adds a product to their cart from the product detail page, they should be taken to a cart page that displays a list of the products they've added to their cart, along with the total cost.
- **Checkout page**: When the user is ready to checkout, they should be taken to a checkout page where they can enter their payment and shipping information.

To complete this exercise, you'll need to use various Flutter widgets, including ListView, ListTile, Navigator, and AppBar, as well as state management techniques such as Provider or Bloc to manage the state of the user's cart. Additionally, you could also connect the app to a backend API to retrieve product data and process payments.