

```

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/foundation.dart';
void main() {
  runApp(

    Chat(),

  );
}

final ThemeData kIOSTheme = ThemeData(
  primarySwatch: Colors.lightBlue,
  primaryColor: Colors.grey[100],
  primaryColorBrightness: Brightness.light,
);

final ThemeData kDefaultTheme = ThemeData(
  primarySwatch: Colors.cyan,
  accentColor: Colors.green,
);

String _name = 'Your Name';

class Chat extends StatelessWidget {
  const Chat({
    Key key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Chat',
      theme: defaultTargetPlatform == TargetPlatform.iOS
        ? kIOSTheme
        : kDefaultTheme,
      home: ChatScreen(),
    );
  }
}

class ChatMessage extends StatelessWidget {
  ChatMessage({this.text, this.animationController});
  final String text;
  final AnimationController animationController;

  @override
  Widget build(BuildContext context) {
    return SizeTransition(
      sizeFactor:
        CurvedAnimation(parent: animationController, curve: Curves.easeOut),
      axisAlignment: 0.0,
      child: Container(
        margin: EdgeInsets.symmetric(vertical: 10.0),
        child: Row(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(

```

```

        margin: const EdgeInsets.only(right: 16.0),
        child: CircleAvatar(child: Text(_name[0])),
      ),
      Expanded(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(_name, style: Theme.of(context).textTheme.headline4),
            Container(
              margin: EdgeInsets.only(top: 5.0),
              child: Text(text),
            ),
          ],
        ),
      ),
    ],
  ),
),
],
),
);
}
}

class ChatScreen extends StatefulWidget {
  @override
  _ChatScreenState createState() => _ChatScreenState();
}

class _ChatScreenState extends State<ChatScreen> with TickerProviderStateMixin {
  final List<ChatMessage> _messages = [];
  final _textController = TextEditingController();
  final FocusNode _focusNode = FocusNode();
  bool _isComposing = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Chat'),
        elevation:
          Theme.of(context).platform == TargetPlatform.iOS ? 0.0 : 4.0,
      ),
      body: Container(
        child: Column(
          children: [
            Flexible(
              child: ListView.builder(
                padding: EdgeInsets.all(8.0),
                reverse: true,
                itemBuilder: (_, int index) => _messages[index],
                itemCount: _messages.length,
              ),
            ),
            Divider(height: 1.0),
            Container(
              decoration: BoxDecoration(color: Theme.of(context).cardColor),
              child: _buildTextComposer(),
            ),
          ],
        ),
      ),
    );
  }
}

```

```

        decoration: Theme.of(context).platform == TargetPlatform.iOS //new
            ? BoxDecoration(
                border: Border(
                    top: BorderSide(color: Colors.grey[200]),
                ),
            )
            : null),
    );
}

Widget _buildTextComposer() {
    return IconTheme(
        data: IconThemeData(color: Theme.of(context).accentColor),
        child: Container(
            margin: EdgeInsets.symmetric(horizontal: 8.0),
            child: Row(
                children: [
                    Flexible(
                        child: TextField(
                            controller: _textController,
                            onChanged: (String text) {
                                setState(() {
                                    _isComposing = text.length > 0;
                                });
                            },
                            onSubmitted: _isComposing ? _handleSubmitted : null,
                            decoration:
                                InputDecoration.collapsed(hintText: 'Send a message'),
                            focusNode: _focusNode,
                        ),
                    ),
                    Container(
                        margin: EdgeInsets.symmetric(horizontal: 4.0),
                        child: Theme.of(context).platform == TargetPlatform.iOS
                            ? CupertinoButton(
                                child: Text('Send'),
                                onPressed: _isComposing
                                    ? () => _handleSubmitted(_textController.text)
                                    : null,
                            )
                            : IconButton(
                                icon: const Icon(Icons.send),
                                onPressed: _isComposing
                                    ? () => _handleSubmitted(_textController.text)
                                    : null,
                            )
                    ),
                ],
            ),
        ),
    );
}

void _handleSubmitted(String text) {
    _textController.clear();
    setState(() {
        _isComposing = false;
    });
    ChatMessage message = ChatMessage(
        text: text,

```

```
        animationController: AnimationController(  
          duration: const Duration(milliseconds: 700),  
          vsync: this,  
        ),  
      );  
      setState(() {  
        _messages.insert(0, message);  
      });  
      _focusNode.requestFocus();  
      message.animationController.forward();  
    }  
  
    @override  
    void dispose() {  
      for (ChatMessage message in _messages)  
        message.animationController.dispose();  
      super.dispose();  
    }  
  }  
}
```