PART 1.

Code Implementation

UI Elements

```
dart
Copy code
import 'package:flutter/material.dart';

class SendMoneyPage extends StatefulWidget {
  @override
  _SendMoneyPageState createState() => _SendMoneyPageState();
}

class _SendMoneyPageState extends State<SendMoneyPage> {
  final _formKey = GlobalKey<FormState>();
  final _recipientController = TextEditingController();
  final _amountController = TextEditingController();

  String? _selectedPaymentMethod;
  bool _isFavorite = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Send Money"),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Recipient Name TextField
              TextFormField(
                controller: _recipientController,
                decoration: InputDecoration(
                  labelText: "Recipient Name",
                  border: OutlineInputBorder(),
                ),
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return "Recipient name cannot be empty.";
                  }
                  return null;
                },
              ),
              SizedBox(height: 16.0),

              // Amount TextField
              TextFormField(
                controller: _amountController,
                keyboardType: TextInputType.number,
                decoration: InputDecoration(
                  labelText: "Amount",
                  border: OutlineInputBorder(),
                ),
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return "Amount cannot be empty.";
                  }
                  if (double.tryParse(value) == null || double.parse(value)
<= 0) {
                    return "Amount must be a positive number.";
                  }
                  return null;
                },
              ),
              SizedBox(height: 16.0),

              // Payment Method Dropdown
              DropdownButtonFormField<String>(
                value: _selectedPaymentMethod,
                decoration: InputDecoration(
                  labelText: "Payment Method",
                  border: OutlineInputBorder(),
                ),
                items: ["Credit Card", "Bank Transfer", "Wallet"]
                    .map((method) => DropdownMenuItem<String>(
                          value: method,
                          child: Text(method),
                        ))
                    .toList(),
                onChanged: (value) {
                  setState(() {
                    _selectedPaymentMethod = value;
                  });
                },
                validator: (value) {
                  if (value == null) {
                    return "Please select a payment method.";
                  }
                  return null;
                },
              ),
              SizedBox(height: 16.0),

              // Favorite Switch
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Text("Mark as Favorite"),
                  Switch(
                    value: _isFavorite,
                    onChanged: (value) {
                      setState(() {
                        _isFavorite = value;
                      });
                    },
                  ),
                ],
              ),
              SizedBox(height: 24.0),

              // Submit Button
              ElevatedButton(
                onPressed: () {
                  if (_formKey.currentState!.validate()) {
                    // Process the transaction
                    ScaffoldMessenger.of(context).showSnackBar(
                      SnackBar(content: Text("Transaction Submitted!")),
                    );
                  }
                },
                child: Text("Send Money"),
              ),
            ],
          ),
        ),
      ),
    );
  }

  @override
  void dispose() {
    _recipientController.dispose();
    _amountController.dispose();
    super.dispose();
  }
}
```

Key Features Implemented

1. **Recipient Name and Amount Fields**:
   - TextFields with proper input validation and error feedback.
   - Ensure the recipient's name is not empty and the amount is a positive number.
2. **DropdownButton for Payment Method**:
   - Allows users to select a payment method from predefined options.
   - Includes validation to ensure a selection is made.
3. **Switch for Marking as Favorite**:
   - Toggle to mark the transaction as a favorite, with a default value of `false`.
4. **Styling**:
   - Used consistent margins, paddings, and styled `OutlineInputBorder` for all input fields.
   - Applied Material Design principles for a clean layout.

Validation

1. **TextField Validation**:
   - Uses `validator` property in `TextFormField` to display error messages.
   - Ensures inputs meet specified conditions before submission.
2. **Dropdown Validation**:
   - Ensures the user selects a payment method.
3. **Feedback**:
   - Error messages are shown below the corresponding fields for clear guidance.

PART 2.

Reusable Button Widget

Definition:

A reusable widget means it can be called multiple times across the app with minimal customization, maintaining the same design and behavior.

Code Example (Flutter):

```
dart
Copy code
import 'package:flutter/material.dart';

class SendMoneyButton extends StatelessWidget {
  final String label; // Button label text
  final VoidCallback onPressed; // Callback function for button action

  const SendMoneyButton({
    Key? key,
    required this.label,
    required this.onPressed,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      onPressed: onPressed,
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.green, // Consistent primary color
        padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 15),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(12), // Consistent corner
radius
        ),
        textStyle: const TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.bold,
          color: Colors.white, // Consistent font and color
        ),
      ),
      child: Text(label),
    );
  }
}
```

---

2. Applying UI Consistency

Consistency is vital for maintaining a professional and user-friendly interface. Here's how to ensure uniformity across the app:

Design Tokens

- **Colors**: Define a primary color palette in a separate constants file.

  ```
  dart
  Copy code
  class AppColors {
    static const Color primary = Colors.green;
    static const Color textColor = Colors.white;
    static const Color borderColor = Colors.grey;
  }
  ```

- **Fonts**: Define a consistent typography style using `TextTheme`.

  ```
  dart
  Copy code
  class AppTypography {
    static const TextStyle buttonText = TextStyle(
      fontSize: 16,
      fontWeight: FontWeight.bold,
      color: AppColors.textColor,
    );
  }
  ```

- **Spacing**: Use consistent padding and margins throughout the app.

  ```
  dart
  Copy code
  ```

```
      class AppSpacing {
        static const double buttonHorizontal = 20.0;
        static const double buttonVertical = 15.0;
      }
```

Example of Using the Button in Screens

```
dart
Copy code
import 'package:flutter/material.dart';
import 'send_money_button.dart'; // Import custom button

class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Home Screen'),
      ),
      body: Center(
        child: SendMoneyButton(
          label: 'Send Money',
          onPressed: () {
            // Navigate to send money screen
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => const
SendMoneyScreen()),
            );
          },
        ),
      ),
    );
  }
}
```

---

3. Ensuring Reusability

- **Parameterization**: Add parameters like `color`, `icon`, or `isDisabled` to increase flexibility.
- **Theming**: Integrate with app themes for centralized updates.

Extended Button Example

```
dart
Copy code
class SendMoneyButton extends StatelessWidget {
  final String label;
  final VoidCallback onPressed;
  final Color? backgroundColor;
  final Widget? icon;

  const SendMoneyButton({
    Key? key,
    required this.label,
    required this.onPressed,
    this.backgroundColor,
    this.icon,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return ElevatedButton.icon(
      onPressed: onPressed,
      style: ElevatedButton.styleFrom(
        backgroundColor: backgroundColor ?? AppColors.primary,
        padding: const EdgeInsets.symmetric(
          horizontal: AppSpacing.buttonHorizontal,
          vertical: AppSpacing.buttonVertical,
        ),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(12),
        ),
        textStyle: AppTypography.buttonText,
      ),
      icon: icon ?? const SizedBox.shrink(),
      label: Text(label),
    );
  }
}
```

PART 3.

Smooth UI Animations

Using AnimatedContainer

AnimatedContainer is ideal for animating size, color, or shape transitions smoothly.

Example: Success Message After Transaction

Dart

Copy code

Import ‘package:flutter/material.dart’;

```
Class SuccessMessage extends StatefulWidget {
 Const SuccessMessage({Key? Key}) : super(key: key);

 @override
 State<SuccessMessage> createState() => _SuccessMessageState();
}

Class _SuccessMessageState extends State<SuccessMessage> {
 Bool _isVisible = false;

 @override
 Void initState() {
  Super.initState();
  // Show the success message after a delay
  Future.delayed(const Duration(seconds: 1), () {
   setState(() {
    _isVisible = true;
   });
  });
 }

 @override
 Widget build(BuildContext context) {
  Return Scaffold(
   Body: Center(
    Child: AnimatedContainer(
     Duration: const Duration(seconds: 1),
     Curve: Curves.easeInOut,
     Height: _isVisible ? 100 : 0,
     Width: _isVisible ? 300 : 0,
     Decoration: BoxDecoration(
      Color: Colors.green,
      borderRadius: BorderRadius.circular(12),
     ),
     Alignment: Alignment.center,
     Child: _isVisible
       ? const Text(
         'Transaction Successful!',
         Style: TextStyle(
          Color: Colors.white,
          fontSize: 18,
          fontWeight: FontWeight.bold,
         ),
        )
       : null,
    ),
   ),
  );
 }
}
```

Using AnimatedOpacity

AnimatedOpacity is great for fade-in or fade-out effects.

Example: Fade-in Success Message

Dart

Copy code

```
Import ‘package:flutter/material.dart’;

Class FadeInMessage extends StatefulWidget {
  Const FadeInMessage({Key? Key}) : super(key: key);

  @override
  State<FadeInMessage> createState() => _FadeInMessageState();
}

Class _FadeInMessageState extends State<FadeInMessage> {
  Bool _isVisible = false;

  @override
  Void initState() {
    Super.initState();
    Future.delayed(const Duration(seconds: 1), () {
      setState(() {
        _isVisible = true;
      });
    });
```

```
}

@override
Widget build(BuildContext context) {
  Return Scaffold(
    Body: Center(
      Child: AnimatedOpacity(
        Duration: const Duration(seconds: 1),
        Opacity: _isVisible ? 1.0 : 0.0,
        Child: Container(
          Padding: const EdgeInsets.all(16),
          Decoration: BoxDecoration(
            Color: Colors.green,
            borderRadius: BorderRadius.circular(12),
          ),
          Child: const Text(
            'Transaction Successful!',
            Style: TextStyle(
              Color: Colors.white,
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ),
    ),
    );
  }
}
```

2. Page Transitions

Flutter provides the PageRouteBuilder class to create custom page transitions. Combine it with animations for a polished experience.

Custom Slide Transition

Dart

Copy code

```
Import ‘package:flutter/material.dart’;

Class LoginPage extends StatelessWidget {
  Const LoginPage({Key? Key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Return Scaffold(
      Body: Center(
        Child: ElevatedButton(
          onPressed: () {
            Navigator.of(context).push(_createRoute());
          },
          Child: const Text(‘Go to Dashboard’),
        ),
      ),
    );
  }
}

Route _createRoute() {
  Return PageRouteBuilder(
    pageBuilder: (context, animation, secondaryAnimation) => const DashboardPage(),
    transitionsBuilder: (context, animation, secondaryAnimation, child) {
      const begin = Offset(1.0, 0.0); // Start from the right
      const end = Offset.zero; // End at the current position
      const curve = Curves.easeInOut;

      var tween = Tween(begin: begin, end: end).chain(CurveTween(curve: curve));
      var offsetAnimation = animation.drive(tween);

      return SlideTransition(
        position: offsetAnimation,
        child: child,
      );
    },
  );
}

Class DashboardPage extends StatelessWidget {
  Const DashboardPage({Key? Key}) : super(key: key);
```

```
 @override
 Widget build(BuildContext context) {
  Return Scaffold(
   appBar: AppBar(title: const Text('Dashboard')),
   body: const Center(child: Text('Welcome to the Dashboard!')),
  );
 }
}
```

Custom Fade Transition

Dart

Copy code

```
Route _createFadeRoute() {
 Return PageRouteBuilder(
  pageBuilder: (context, animation, secondaryAnimation) => const DashboardPage(),
  transitionsBuilder: (context, animation, secondaryAnimation, child) {
   return FadeTransition(
    opacity: animation,
    child: child,
   );
  },
 );
}
```

Tips for Effective Animations

Duration: Keep transitions short (e.g., 300ms to 1s) to maintain responsiveness.

Curves: Use easing curves like Curves.easeInOut or Curves.decelerate for natural motion.

Consistency: Apply similar animations across the app for a cohesive look.