Process Documentation

**Automation of Number System Conversions**

Course Code: SPC 2201

Lecturer’s Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Introduction**

Number systems such as decimal, binary, octal, and hexadecimal are essential in computing. Manual conversion between these systems is time-consuming and prone to errors. This project automates the process, providing a mobile application that converts numbers efficiently and accurately.

**2. Objectives**

- To automate conversion between number systems (decimal, binary, octal, hexadecimal).  
- To provide a simple, user-friendly mobile app interface.  
- To deploy the application on Android and iOS platforms.  
- To apply teamwork in a software development project.

**3. Tools & Technologies**

- Programming Language: Dart  
- Framework: Flutter  
- IDE: VS Code / Android Studio  
- Version Control: GitHub (lecturer added as collaborator)  
- Platforms: Android & iOS

**4. System Design**

Algorithm (Decimal → Binary example):  
1. Divide the number by 2.  
2. Record the remainder.  
3. Repeat until quotient is 0.  
4. Read remainders in reverse order.  
  
Flowchart: Input → Processing (conversion) → Output

**5. Implementation Process**

Step 1: Set up Flutter environment.  
Step 2: Designed the app interface with a text input and output labels.  
Step 3: Implemented conversion functions.  
Step 4: Integrated logic with UI.  
Step 5: Tested app with multiple inputs.  
Step 6: Built APK and iOS package.  
Step 7: Uploaded project to GitHub and added lecturer as collaborator.

**6. Testing**

| Input | Expected Output (Binary, Octal, Hex) | Actual Output | Status |  
|-------|-------------------------------------|---------------|--------|  
| 25 | 11001, 31, 19 | 11001, 31, 19 | Pass |  
| 255 | 11111111, 377, FF | 11111111, 377, FF | Pass |  
| abc | Error message | Error message | Pass |

**7. Challenges & Solutions**

- Challenge: iOS deployment required MacOS.  
 Solution: Group tested on Android emulator and prepared APK.  
- Challenge: Invalid inputs crashed the app.  
 Solution: Added error handling.

**8. Conclusion**

The project successfully achieved its goal of automating number system conversions and deploying on Android and iOS. The app can be expanded in future to support additional bases and fractional numbers.