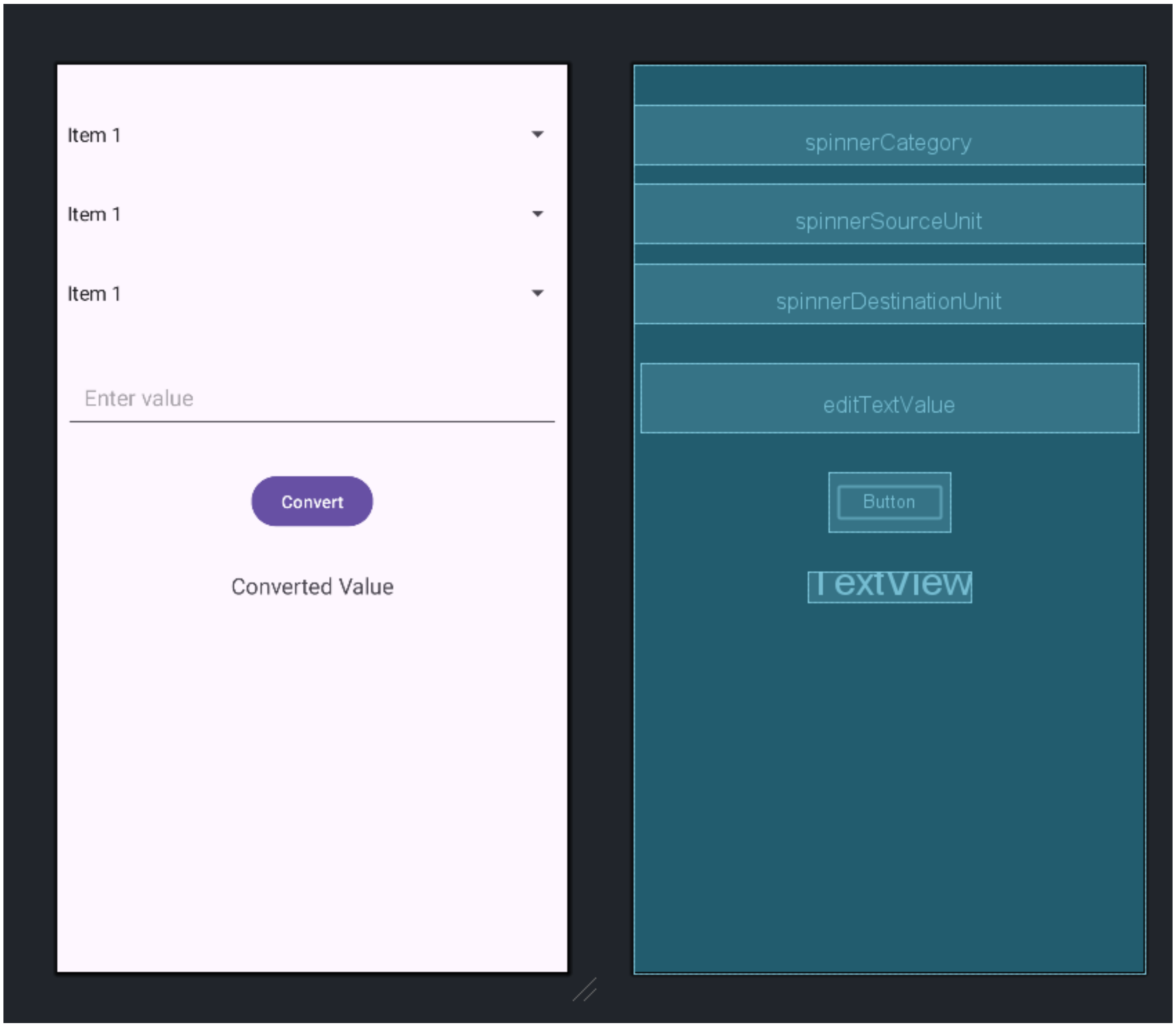
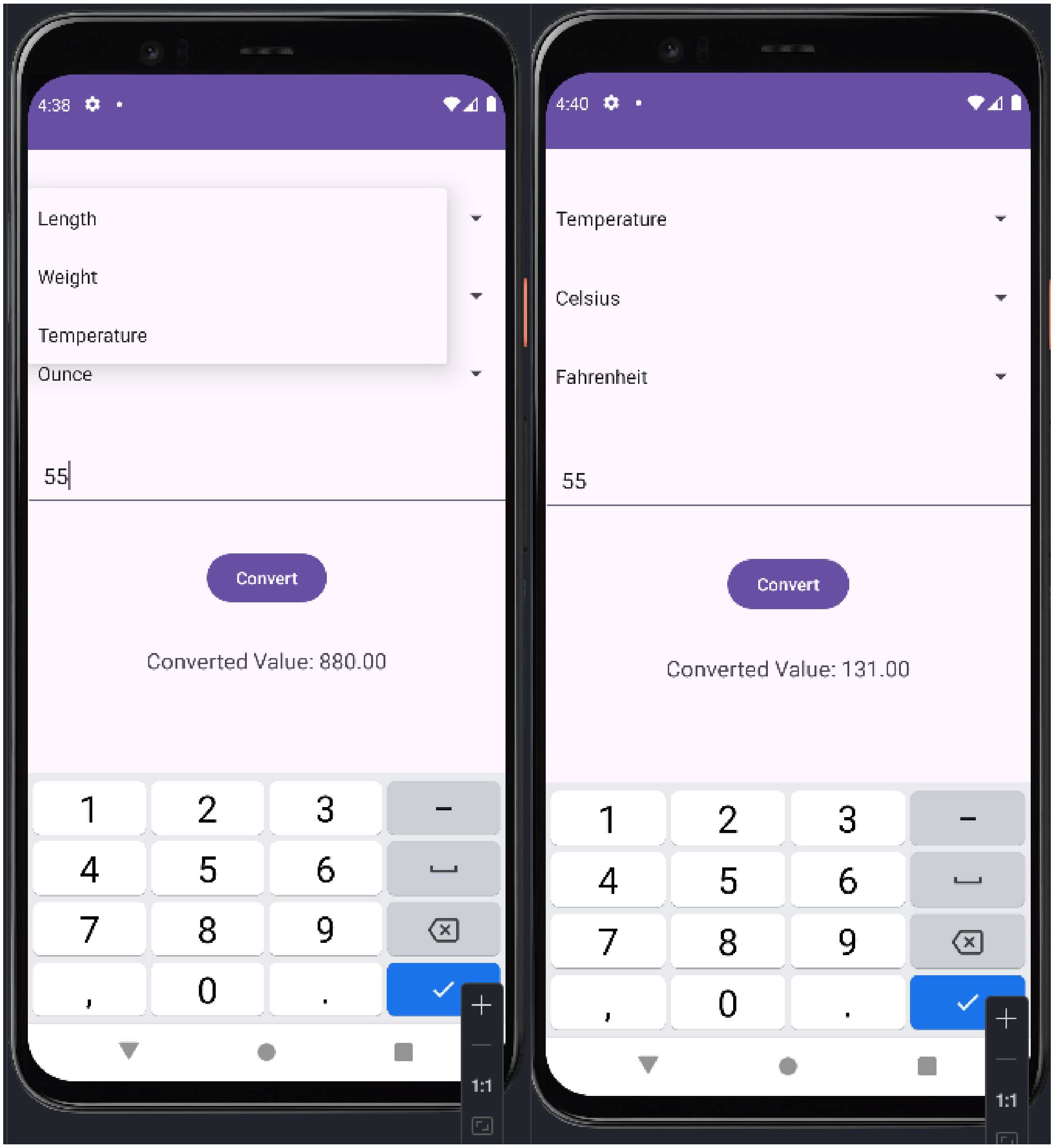
GitHub: https://github.com/KyleOr/SIT305-Task-2.1P

Recording: https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=688b281c-2d57-4974-af8d-b2ac006e22e3

Subtasks

Subtask 1: Design UI for the App





Ounce

55

Converted Value: 880.00

Temperature

Celsius

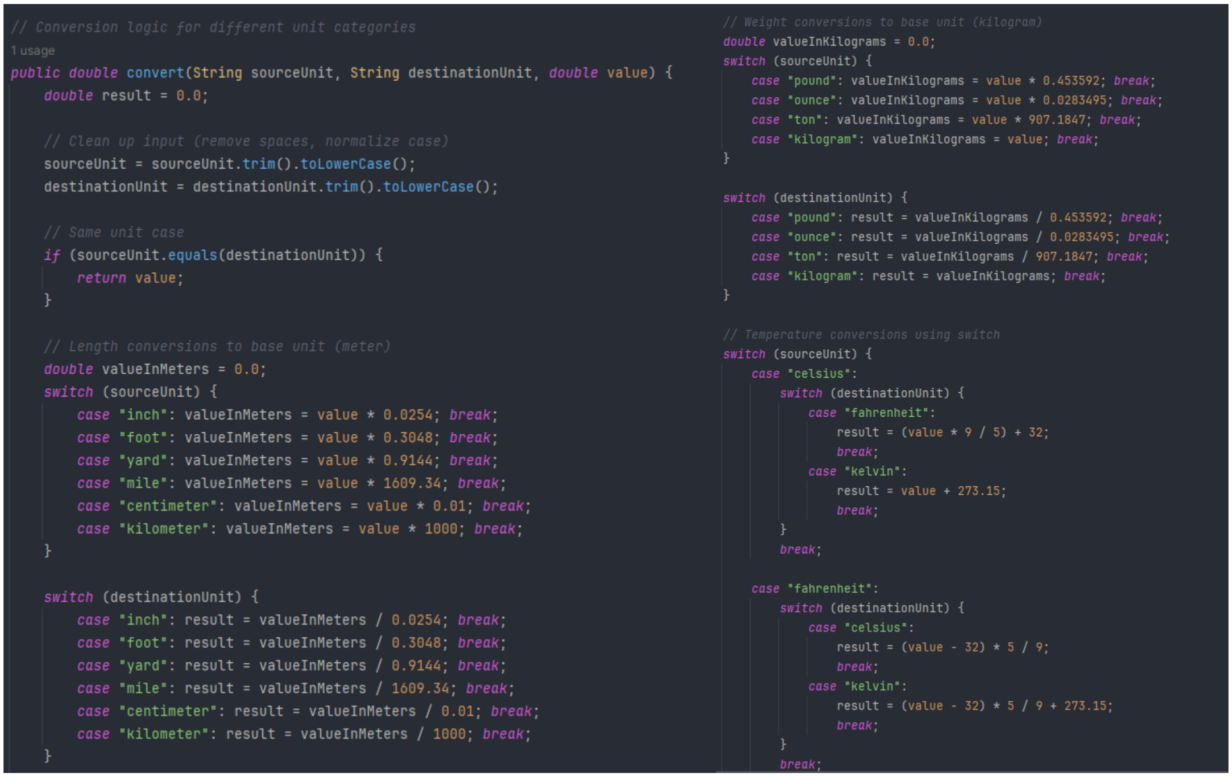
Fahrenheit

55

Converted Value: 131.00

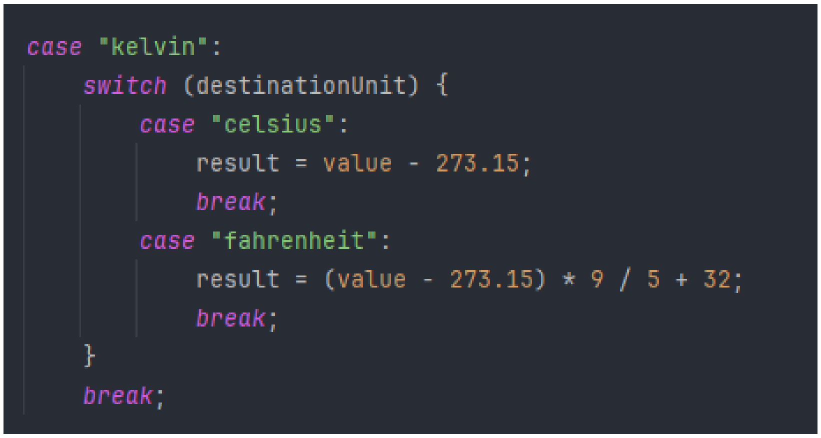
Subtask 2: Implement the Conversion Logic

I used a switch case so any source unit can be converted to any destination unit within their base unit (length, temperature, weight).



```
// Conversion logic for different unit categories
1 usage
public double convert(String sourceUnit, String destinationUnit, double value) {
    double result = 0.0;

    // Clean up input (remove spaces, normalize case)
    sourceUnit = sourceUnit.trim().toLowerCase();
    destinationUnit = destinationUnit.trim().toLowerCase();

    // Same unit case
    if (sourceUnit.equals(destinationUnit)) {
        return value;
    }

    // Length conversions to base unit (meter)
    double valueInMeters = 0.0;
    switch (sourceUnit) {
        case "inch": valueInMeters = value * 0.0254; break;
        case "foot": valueInMeters = value * 0.3048; break;
        case "yard": valueInMeters = value * 0.9144; break;
        case "mile": valueInMeters = value * 1609.34; break;
        case "centimeter": valueInMeters = value * 0.01; break;
        case "kilometer": valueInMeters = value * 1000; break;
    }

    switch (destinationUnit) {
        case "inch": result = valueInMeters / 0.0254; break;
        case "foot": result = valueInMeters / 0.3048; break;
        case "yard": result = valueInMeters / 0.9144; break;
        case "mile": result = valueInMeters / 1609.34; break;
        case "centimeter": result = valueInMeters / 0.01; break;
        case "kilometer": result = valueInMeters / 1000; break;
    }
```

```
    // Weight conversions to base unit (kilogram)
    double valueInKilograms = 0.0;
    switch (sourceUnit) {
        case "pound": valueInKilograms = value * 0.453592; break;
        case "ounce": valueInKilograms = value * 0.0283495; break;
        case "ton": valueInKilograms = value * 907.1847; break;
        case "kilogram": valueInKilograms = value; break;
    }

    switch (destinationUnit) {
        case "pound": result = valueInKilograms / 0.453592; break;
        case "ounce": result = valueInKilograms / 0.0283495; break;
        case "ton": result = valueInKilograms / 907.1847; break;
        case "kilogram": result = valueInKilograms; break;
    }

    // Temperature conversions using switch
    switch (sourceUnit) {
        case "celsius":
            switch (destinationUnit) {
                case "fahrenheit":
                    result = (value * 9 / 5) + 32;
                    break;
                case "kelvin":
                    result = value + 273.15;
                    break;
            }
            break;

        case "fahrenheit":
            switch (destinationUnit) {
                case "celsius":
                    result = (value - 32) * 5 / 9;
                    break;
                case "kelvin":
                    result = (value - 32) * 5 / 9 + 273.15;
                    break;
            }
            break;
```



```
case "kelvin":
    switch (destinationUnit) {
        case "celsius":
            result = value - 273.15;
            break;
        case "fahrenheit":
            result = (value - 273.15) * 9 / 5 + 32;
            break;
    }
    break;
```

Subtask 3: Research on Llama2

LLaMA 2 (Large Language Model Meta AI) is an advanced open-source large language model developed by Meta, designed to handle complex natural language understanding and generation tasks. Available in 7B, 13B, and 65B parameter sizes, LLaMA 2 is efficient enough to run on mobile devices, making it ideal for Android apps. Its ability to operate locally enhances privacy, reduces latency, and allows for customization based on specific app needs. LLaMA 2's versatility makes it suitable for various applications in mobile app development. (IBM, 2023)

Smart Virtual Assistant – LLaMA 2 can power in-app assistants to provide real-time, context-aware responses, such as personalized workout advice in a fitness app. Beyond simple question-answering, it can analyze user preferences, past activity, and health data to offer dynamic recommendations tailored to individual fitness goals. For example, it can suggest adjusting workout intensity based on user fatigue levels or recommending meal plans aligned with fitness targets. Its natural language understanding capabilities allow it to handle follow-up questions and maintain context over time, creating a conversational and intuitive experience. (Mahayana, 2023)

Content Generation and Summarization – It can create articles, social media posts, and summaries of long texts, improving user productivity in news and note-taking apps. LLaMA 2 can generate content in various tones and styles, adapting to the user's preferences and platform requirements. For example, it can write engaging social media captions based on trending topics or craft professional emails by analyzing the context of previous conversations. Its summarization capabilities allow it to condense lengthy articles or research papers into concise, informative summaries, helping users quickly grasp key points without reading the entire text. (Dheiryatalks, 2023)

Language Translation and Multilingual Chat – LLaMA 2 can enable real-time translation and multilingual communication in chat and travel apps. It can detect the user's language automatically and provide seamless translations during conversations, allowing for more natural and accurate communication. Beyond simple translation, it can also adjust for cultural nuances and context, ensuring that the tone and meaning remain consistent across languages. For travel apps, it can suggest local phrases, provide instant translations for menus or signs using text recognition, and offer culturally appropriate responses, enhancing the overall travel experience. (Pakhale, 2023)

Code Generation and Debugging Assistance – LLaMA 2 can generate code snippets, suggest fixes, and explain programming errors in Android development tools. It can analyze incomplete or buggy code and recommend efficient solutions, helping developers reduce debugging time. Additionally, it can offer context-aware code completion and suggest best practices based on the project structure and coding style. LLaMA 2 can also provide explanations for error messages, guiding developers through the debugging process and improving their overall coding efficiency. (Goel, 2024)

Emotion and Sentiment Analysis – LLaMA 2 can be used to analyze and understand user emotions or sentiment in real-time through text input in apps. For example, it can detect if a user's message in a chat or social media post expresses frustration, happiness, or concern. This can be valuable in customer support apps, where LLaMA 2 can prioritize urgent tickets based on sentiment or suggest appropriate responses for support agents. In wellness apps, it can help track mood patterns over time and provide insights or suggestions to improve mental health. By understanding the emotional tone behind interactions, LLaMA 2 can enable more empathetic and responsive user experiences, fostering a stronger connection between the user and the app. (Massaron, 2024)

Dheiryatalks (2023). Llama 2: The Next Generation of AI Tools That Will Change the World. [online] Medium. Available at: https://medium.com/@dheiryatalks/llama-2-the-next-generation-of-ai-tools-that-will-change-the-world-5245c862ff1d.

Dmitri Yanno Mahayana (2023). Creating Virtual Assistance using with Llama-2 7B Chat Model. [online] Medium. Available at: https://medium.com/@dmitri.mahayana/creating-virtual-assistance-using-with-llama-2-7b-chat-model-9f693c8250ee.

Goel, P. (2024). Llama models for code generation and Safety - Pradeep Goel - Medium. [online] Medium. Available at: https://medium.com/@pradeepgoel/llama-models-for-code-generation-and-safety-55995717a3f1.

IBM (2023). Llana2. [online] Ibm.com. Available at: https://www.ibm.com/think/topics/llama-2.

Massaron, L. (2024). Fine-tune Llama 2 for sentiment analysis. [online] kaggle.com. Available at: https://www.kaggle.com/code/lucamassaron/fine-tune-llama-2-for-sentiment-analysis.

Pakhale, T. (2023). Text Translation and Summarization Using LLaMa-2. [online] www.e2enetworks.com. Available at: https://www.e2enetworks.com/blog/step-by-step-approach-to-text-translation-and-summarization-using-llama-2.