Step 3 – RESTful API

**Identify Participants:** There will be a single USER participant

**Identify Goal:** The goal of this web service is to return a list of data to a user based on their input.

1. The user should be able to enter a positive integer.
2. In a vertical list, print all numbers between 1 and the entered value.
3. Where the number is divisible by 3, you should instead print "walkers".
4. Where the number is divisible by 5, you should instead print "assessment".
5. Where the number is divisible by both 3 and 5, you should instead print "walkers assessment".
6. Update the logic, so that if today is Monday, "walkers" is substituted with "walkersm" and "assessment" is substituted with "assessment-m".
7. Add validation to ensure the entered value is an integer between 1 and 200.

**Identify Activities and change to verbs:**

1. The User **enters** an input = **Post** User input

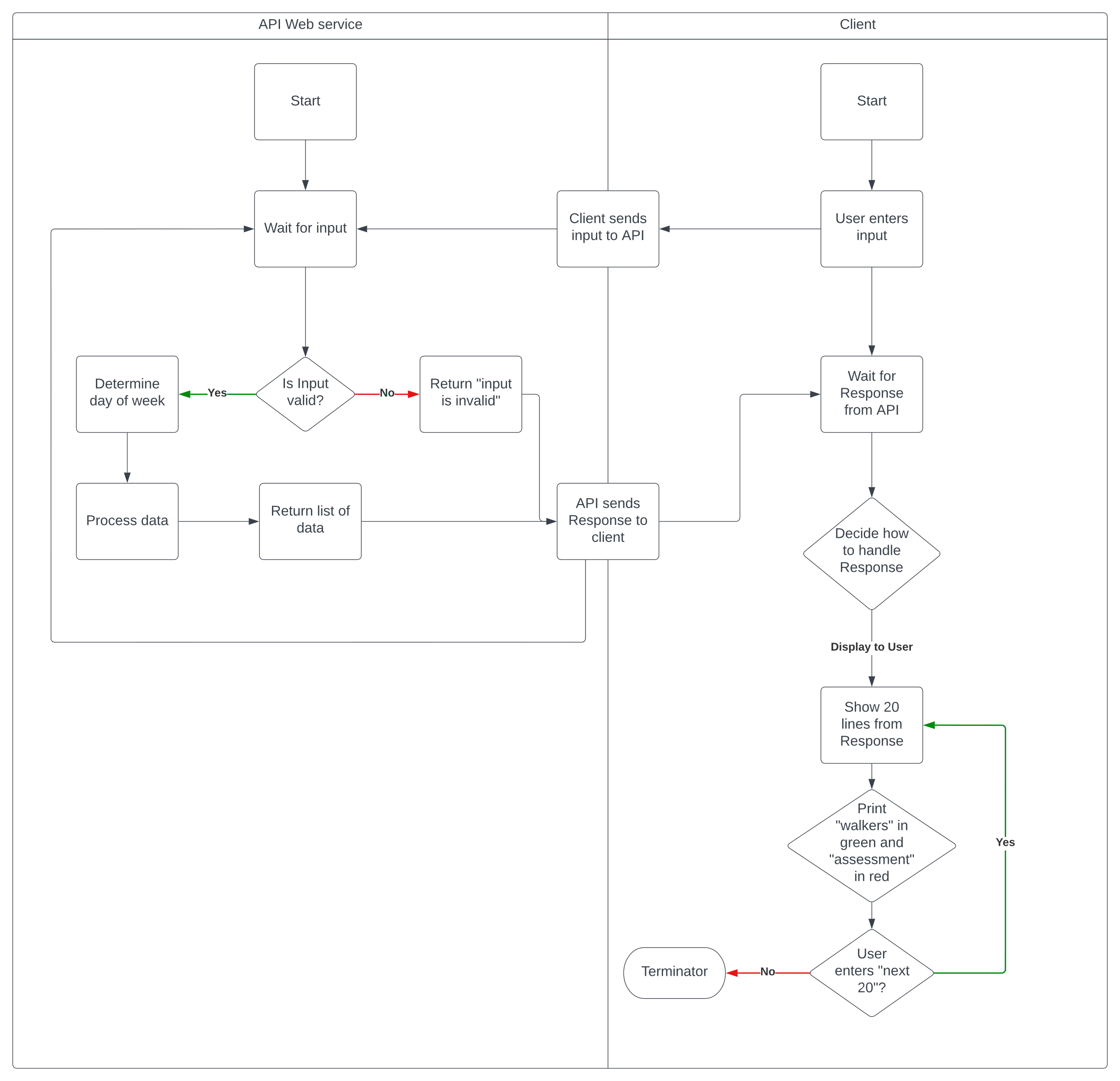
**Identify Steps:**

1. User enters an input on their client
2. Client sends input to API service
3. Validate if input is an integer between 1 and 200
4. Check day of the week
5. Loop from 1 to input and determine the correct output string for each value
6. Return all data to the client
7. Client determines what to do with this data (in our case for the assessment, does the below)
   1. Add styling so that "walkers" is printed in green, and "assessment" in red.
   2. Amend so only 20 values are displayed at a time.
   3. Implement "next 20" to display next 20 values and next 20 after that.
8. User views the client and either begins again from step 1 or enters “next 20” to view more in client

**Document data structures:**

**Input:** Data Posted to API should be a string.

**Response:** Returns data in a List<List<string>> (nested list of strings) in Json format.

**Flow chart:**

**Example API Controller:**

using Microsoft.AspNetCore.Mvc;

using System;

using WalkersAssessment.Models;

namespace WalkersAssessment.Controllers

{

    [ApiController]

    [Route("[controller]")]

    public class NumberController : ControllerBase

    {

        private readonly ILogger<NumberController> \_logger;

        public NumberController(ILogger<NumberController> logger)

        {

            \_logger = logger;

        }

        public static bool Validate(string input, out int number)

        {

            int num;

            bool canParse = int.TryParse(input, out num);

            number = canParse ? num : 0;

            if (canParse && (1 <= num && num <= 200)) return true;

            return false;

        }

        [HttpPost(Name = "PostNumber")]

        public Output Post(string input)

        {

            Output output = new Output();

            output.Data = new List<List<string>>();

            int num;

            if (Validate(input, out num))

            {

                var day = DateTime.Now.DayOfWeek;

                for (int i = 1; i <= num; i++)

                {

                    List<string> line = new List<string>();

                    if (i % 3 == 0)

                    {

                        line.Add(day.ToString() != "Monday" ? "walkers" : "walkersm");

                    }

                    if (i % 5 == 0)

                    {

                        line.Add(day.ToString() != "Monday" ? "assessment" : "assessment-m");

                    }

                    if (!(i % 3 == 0 || i % 5 == 0))

                    {

                        line.Add($"{i}");

                    }

                    output.Data.Add(line);

                }

                return output;

            }

            output.Data.Add(new List<string>() { "Invalid Input..." });

            return output;

        }

    }

}