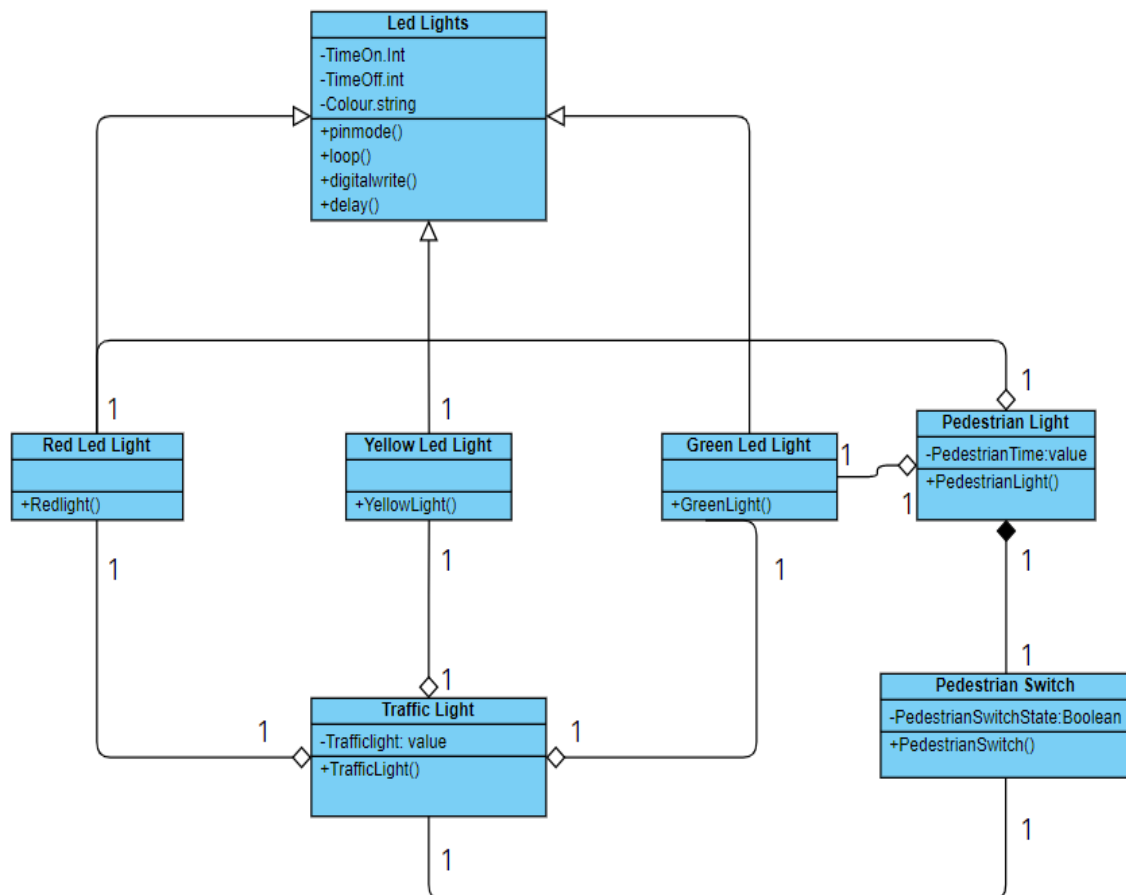
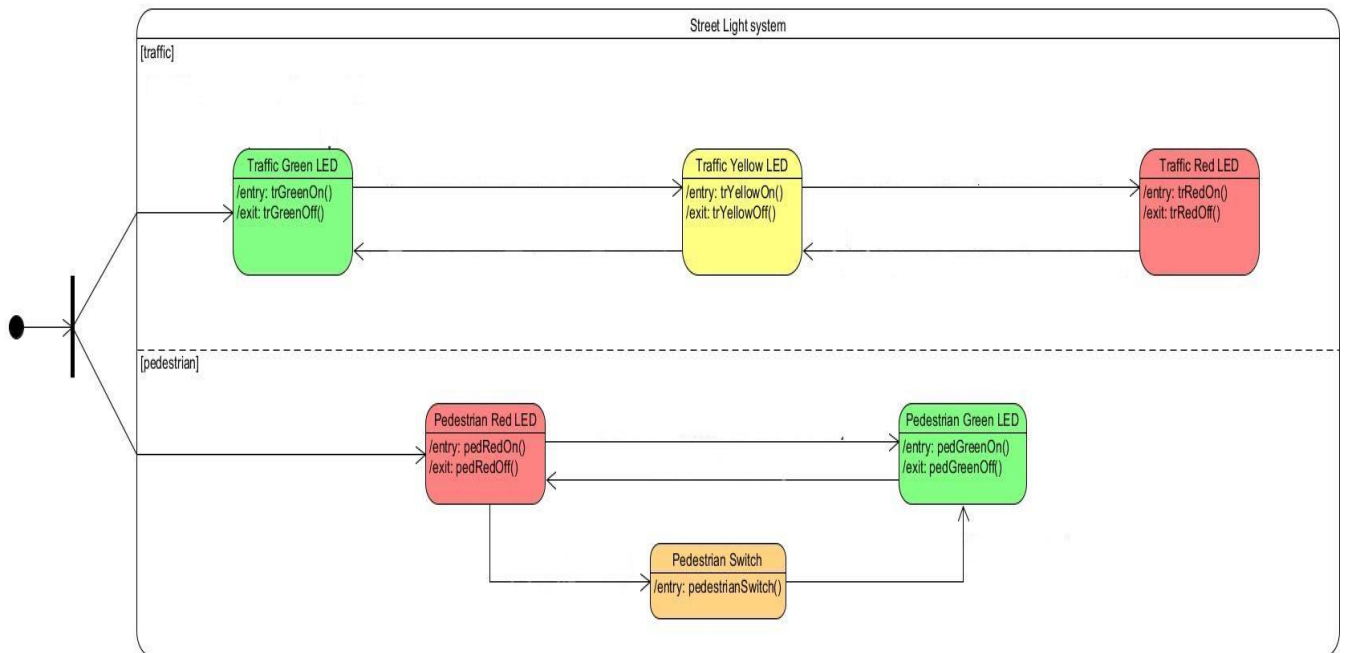
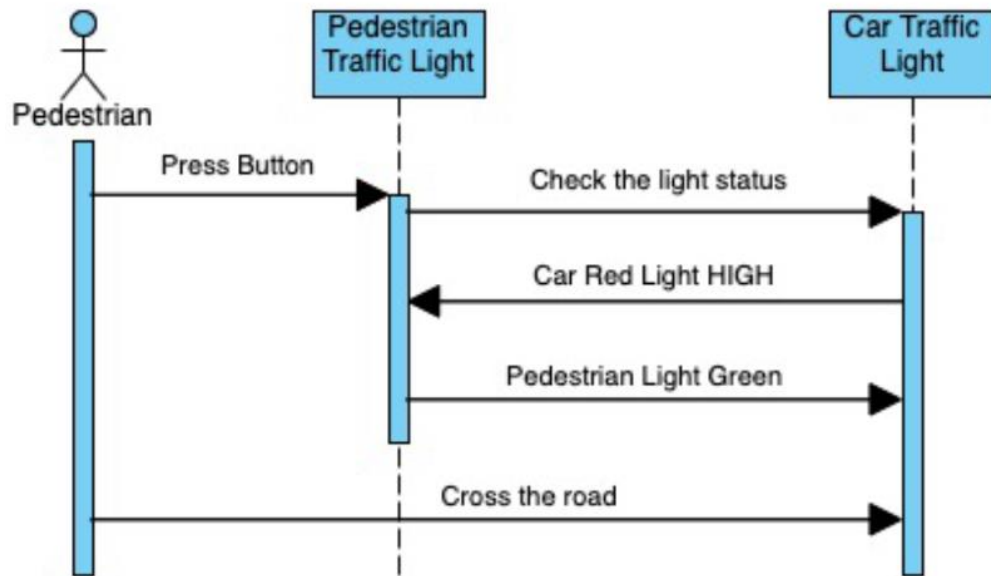


# Task 2

- ▶ Extend the (car) traffic light system of the first sprint by a pedestrian light
  - ▶ Extend a button for the pedestrian. If pressed, the pedestrian lights switches as fast as possible to green.
  - ▶ Guarantee that
    - ▶ The pedestrian lights signal is red if the traffic light signal is green
    - ▶ If the pedestrian lights signal is green the traffic light is red
    - ▶ At least the green traffic light is required to stay for 10 time units in green and 5 time units in yellow.
- ▶ First, model your solution
  - ▶ Extend the class diagram
  - ▶ Add a sequence diagram for showing the interaction with a pedestrian pressing a button including the switching of the lights
  - ▶ Update the state machine model accordingly
  - ▶ Map the state machine model according to the switch case pattern to Arduino code
  - ▶ Extend the tinkercad simulation by
    - ▶ A pedestrian button
    - ▶ Green and red light for the pedestrian light

## Class Diagram





```

1  int red_led = 13;
2  int yellow_led = 12;
3  int green_led = 11;
4  int pedestrian_led = 10;
5  int button = 4;
6  const int DELAY_RED = 3000;
7  const int DELAY_YELLOW = 1000;
8  const int DELAY_GREEN = 3000;
9  const int DELAY_PEDESTRIAN = 3000;
10 int light = 0;
11 void setup()
12 {
13     pinMode(13, OUTPUT);
14     pinMode(12, OUTPUT);
15     pinMode(11, OUTPUT);
16     pinMode(3, OUTPUT);
17     pinMode(1, INPUT);
18 }
19 void loop()
20 {
21     for(int x=0; x>3; x++)
22     {
23         light++ ;
24     }
25     switch(light)
26     {
27         case 0:
28             red_light();
29             break;
30         case 1:
31             yellow_light();
32             break;
33         case 2:
34             green_light();
35             break;
36     }
37 }
38 void red_light()
39 {
40     digitalWrite(red_led, HIGH);
41     digitalWrite(yellow_led, LOW);
42     digitalWrite(green_led, LOW);
43     digitalWrite(pedestrian_led, LOW);
44     delay(DELAY_RED);
45     light++;
46 }
47 void yellow_light()
48 {
49     digitalWrite(red_led, LOW);
50     digitalWrite(yellow_led, HIGH);
51     digitalWrite(green_led, LOW);
52     digitalWrite(pedestrian_led, LOW);
53     delay(DELAY_YELLOW);
54     light++;
55 }
56 void green_light()
57 {
58     digitalWrite(red_led, LOW);
59     digitalWrite(yellow_led, LOW);
60     digitalWrite(green_led, HIGH);
61     digitalWrite(pedestrian_led, LOW);
62     delay(DELAY_GREEN);
63     int state = digitalRead(button);
64     if(state==0)
65     {
66         digitalWrite(red_led, LOW);
67         digitalWrite(yellow_led, HIGH);
68         digitalWrite(green_led, LOW);
69         digitalWrite(pedestrian_led, LOW);
70         delay(DELAY_YELLOW);
71         light=0;
72     }
73     else
74     {
75         digitalWrite(red_led, HIGH);
76         digitalWrite(yellow_led, LOW);
77         digitalWrite(green_led, LOW);
78         digitalWrite(pedestrian_led, HIGH);
79         delay(DELAY_PEDESTRIAN);
80         light=0;
81     }
82     return;
83 }

```

