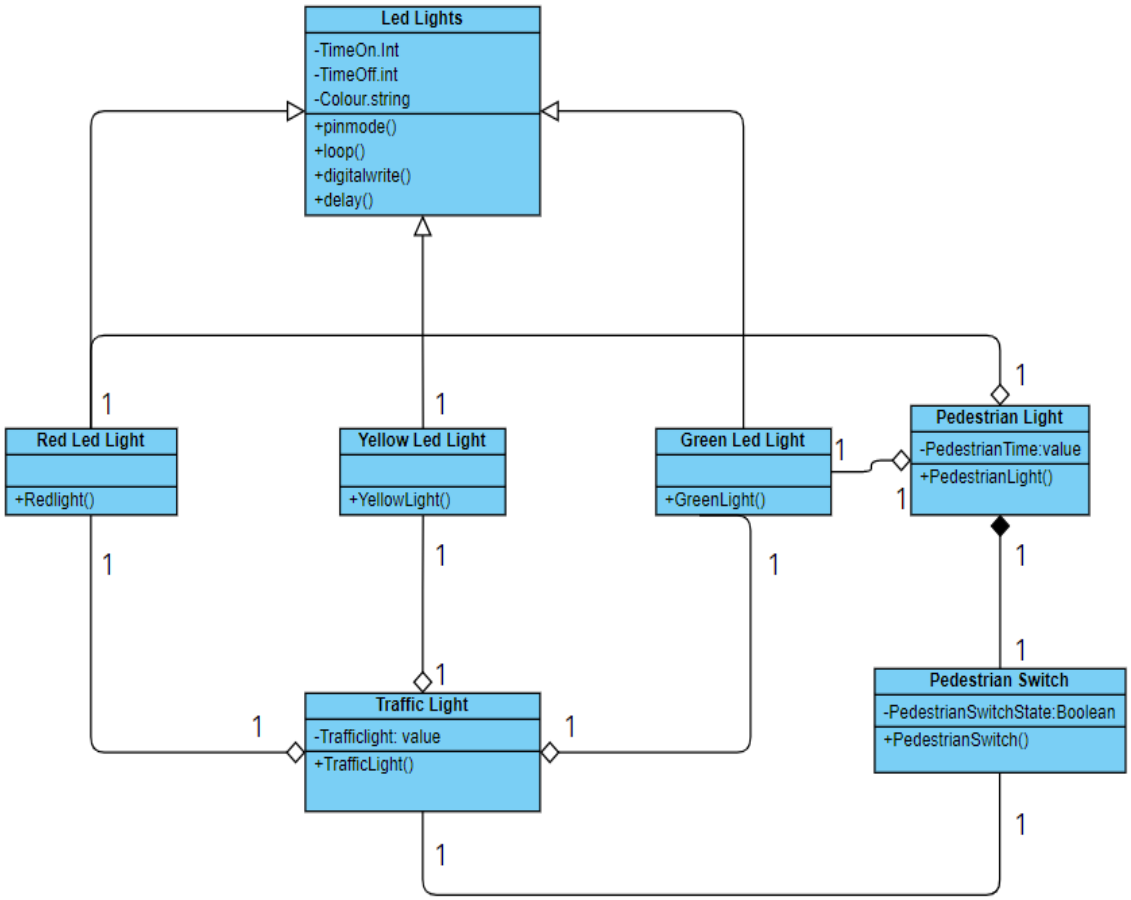
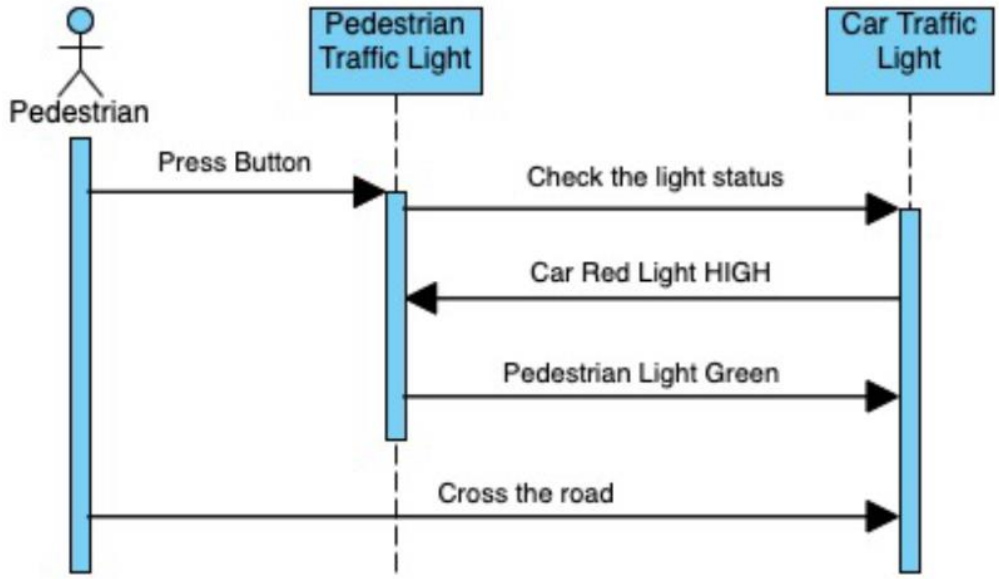
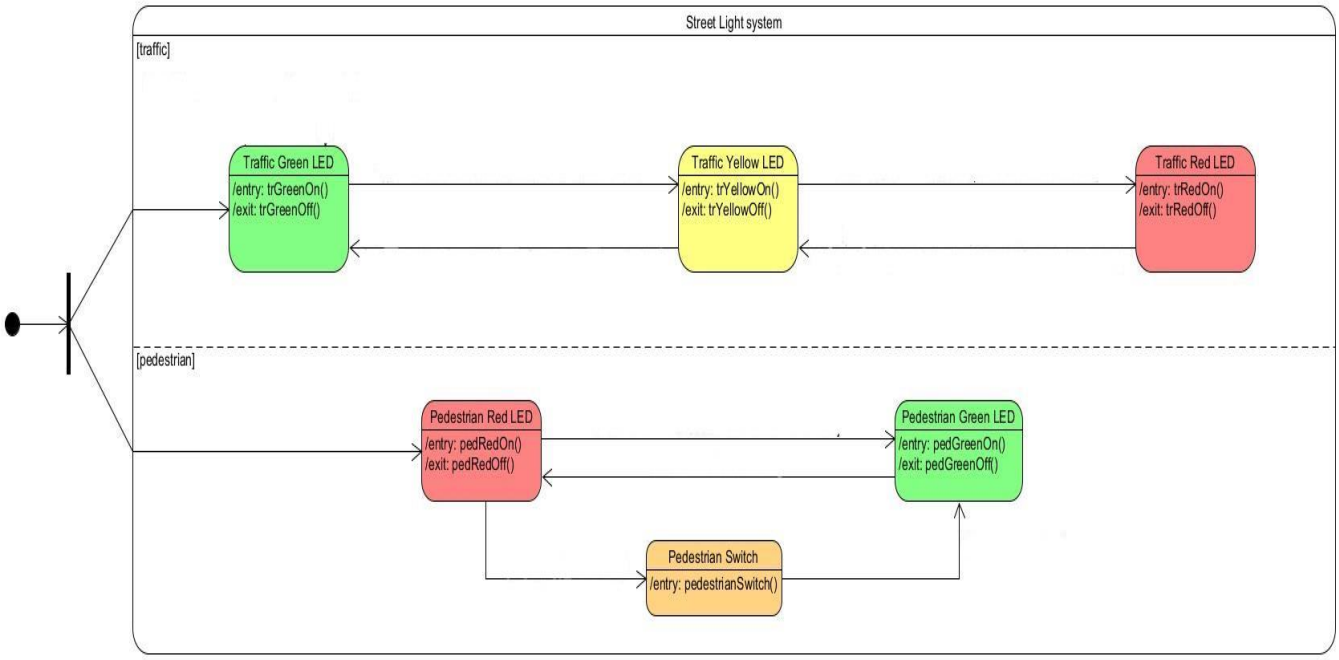
Task 2 a

- 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

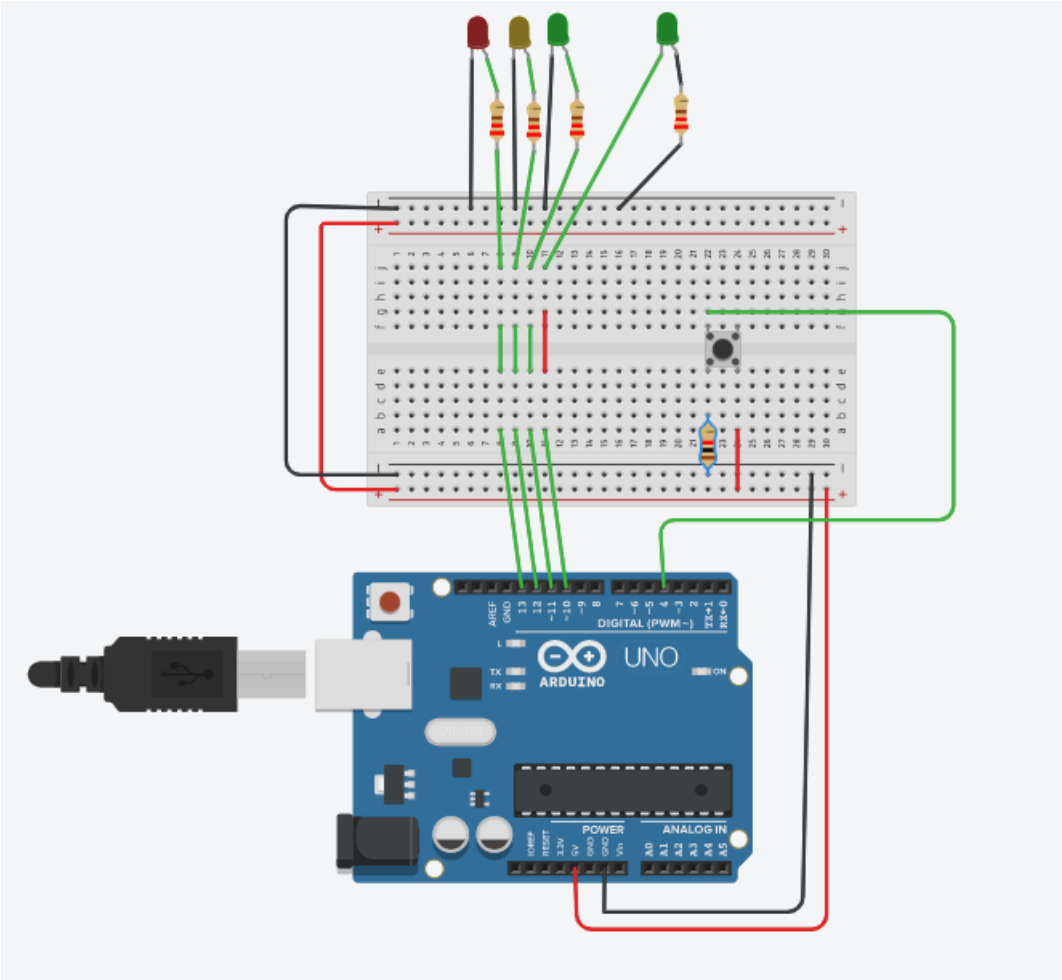
Class Diagram




Pedestrian
Pedestrian Traffic Light
Car Traffic Light
Press Button
Check the light status
Car Red Light HIGH
Pedestrian Light Green
Cross the road


Street Light system
[traffic]
Traffic Green LED
/entry: trGreenOn()
/exit: trGreenOff()
Traffic Yellow LED
/entry: trYellowOn()
/exit: trYellowOff()
Traffic Red LED
/entry: trRedOn()
/exit: trRedOff()
[pedestrian]
Pedestrian Red LED
/entry: pedRedOn()
/exit: pedRedOff()
Pedestrian Green LED
/entry: pedGreenOn()
/exit: pedGreenOff()
Pedestrian Switch
/entry: pedestrianSwitch()

```
int red_led = 13;
int yellow_led = 12;
int green_led = 11;
int pedestrian_led = 10;
int button = 4;
const int DELAY_RED = 3000;
const int DELAY_YELLOW = 1000;
const int DELAY_GREEN = 3000;
const int DELAY_PEDESTRIAN = 3000;
int light = 0;
void setup()
{
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(1,INPUT);
}
void loop()
{
  for(int x=0; x>3; x++)
  {
   light++ ;
  }
switch(light)
{
  case 0:
  red_light();
  break;
  case 1:
  yellow_light();
  break;
  case 2:
  green_light();
  break;
}
}
  void red_light()
  {
    digitalWrite(red_led, HIGH);
    digitalWrite(yellow_led, LOW);
    digitalWrite(green_led, LOW);
    digitalWrite(pedestrian_led, LOW);
    delay(DELAY_RED);
    light++;
  }
void yellow_light()
  {
    digitalWrite(red_led, LOW);
    digitalWrite(yellow_led, HIGH);
    digitalWrite(green_led, LOW);
    digitalWrite(pedestrian_led, LOW);
    delay(DELAY_YELLOW);
  light++;
}
void green_light()
  {
    digitalWrite(red_led, LOW);
    digitalWrite(yellow_led, LOW);
    digitalWrite(green_led, HIGH);
    digitalWrite(pedestrian_led, LOW);
    delay(DELAY_GREEN);
      int state = digitalRead(button);
      if(state==0)
      {
      digitalWrite(red_led, LOW);
      digitalWrite(yellow_led, HIGH);
      digitalWrite(green_led, LOW);
      digitalWrite(pedestrian_led, LOW);
      delay(DELAY_YELLOW);
      light=0;
      }
  else
  {
    digitalWrite(red_led, HIGH);
    digitalWrite(yellow_led, LOW);
    digitalWrite(green_led, LOW);
    digitalWrite(pedestrian_led, HIGH);
    delay(DELAY_PEDESTRIAN);
    light=0;
  }
return;
}
```


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN