```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>



int TRIG_PIN = 4;
int ECHO_PIN = 3;
int EXIT_BUTTON = 10;
int SERVO_PIN = 9;


LiquidCrystal_I2C lcd(0x27,16,2);
Servo gateServo;



const int TOTAL_SPOTS = 3;
int availableSpots = TOTAL_SPOTS;


const int OPEN_ANGLE = 0;
const int CLOSE_ANGLE = 90;

const int DETECT_DISTANCE = 20;

const int SERVO_DELAY = 15;

const int GATE_TIME = 3000;

const int DEBOUNCE_TIME = 300;


enum State {

  IDLE,

  ENTER_PREPARE,
  ENTER_OPEN,
  ENTER_WAIT,

  EXIT_OPEN,
```

```
  EXIT_WAIT,

  WAIT_CLOSE

};


State state = IDLE;

unsigned long stateTime = 0;
unsigned long lastButton = 0;

int distance = 999;

bool carDetected = false;



int getDistance()
{

  digitalWrite(TRIG_PIN,LOW);
  delayMicroseconds(2);


  digitalWrite(TRIG_PIN,HIGH);
  delayMicroseconds(10);


  digitalWrite(TRIG_PIN,LOW);



  long duration = pulseIn(ECHO_PIN,HIGH,30000);



  if(duration == 0)
  {
    return 999;
  }


  int cm = duration * 0.034 / 2;
```

```
  return cm;

}



void moveGate(int target)
{

  int current = gateServo.read();



  if(current < target)
  {

    for(int pos=current; pos<=target; pos++)
    {

      gateServo.write(pos);
      delay(SERVO_DELAY);

    }

  }


  else
  {

    for(int pos=current; pos>=target; pos--)
    {

      gateServo.write(pos);
      delay(SERVO_DELAY);

    }

  }

}
```

```
void openGate()
{
  moveGate(OPEN_ANGLE);
}



void closeGate()
{
  moveGate(CLOSE_ANGLE);
}




void updateLCD(String message)
{

  lcd.clear();


  lcd.setCursor(0,0);

  lcd.print("Spaces:");

  lcd.print(availableSpots);



  lcd.setCursor(0,1);

  lcd.print(message);

}




void setup()
{
```

```
  Serial.begin(9600);


  pinMode(TRIG_PIN,OUTPUT);
  pinMode(ECHO_PIN,INPUT);


  pinMode(EXIT_BUTTON,INPUT_PULLUP);



  gateServo.attach(SERVO_PIN);

  closeGate();



  lcd.init();
  lcd.backlight();



  updateLCD("Waiting");

}




void loop()
{

  unsigned long now = millis();



  distance = getDistance();


  carDetected = distance < DETECT_DISTANCE;
```

```
bool buttonPressed = false;



if(digitalRead(EXIT_BUTTON) == LOW)
{

  if(now - lastButton > DEBOUNCE_TIME)
  {

    buttonPressed = true;

    lastButton = now;

  }

}



switch(state)
{




  case IDLE:


    updateLCD("Waiting");



    if(carDetected && availableSpots > 0)
    {

      state = ENTER_PREPARE;

      stateTime = now;

    }
```

```
  if(buttonPressed && availableSpots < TOTAL_SPOTS)
  {

    state = EXIT_OPEN;

  }



  break;




case ENTER_PREPARE:


  if(now - stateTime >= 100)
  {

    state = ENTER_OPEN;

  }


  break;



case ENTER_OPEN:


  updateLCD("Opening");


  openGate();


  stateTime = now;
```

```
  state = ENTER_WAIT;

  break;



case ENTER_WAIT:

  if(now - stateTime >= GATE_TIME)
  {
    if(availableSpots > 0)
    {
      availableSpots--;
    }

    closeGate();

    state = WAIT_CLOSE;

  }

  break;



case EXIT_OPEN:

  updateLCD("Exit Gate");

  openGate();
```

```
  stateTime = now;

  state = EXIT_WAIT;

  break;



case EXIT_WAIT:

  if(now - stateTime >= GATE_TIME)
  {

    if(availableSpots < TOTAL_SPOTS)
    {
      availableSpots++;
    }


    closeGate();


    state = WAIT_CLOSE;

  }


  break;



case WAIT_CLOSE:

  updateLCD("Closing");
```

```
    if(!carDetected && digitalRead(EXIT_BUTTON) == HIGH)
    {

      state = IDLE;

    }


    break;



  }


  delay(50);

}
```