Code for Library Management system

```
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <Servo.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>

#define SS_PIN 10
#define RST_PIN 9


Servo servo1;
Servo servo2;
Servo servo3;
int angle;

MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class

int code[] = {233,147,123,90}; //This is the stored UID (Unlock Card)
int codeRead = 0;
String uidString;


const byte rows = 4;
const byte cols = 4;


char keyMap [rows] [cols] = {

  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins [rows] = {5}; //pins of the keypad
byte colPins [cols] = {6,7};

Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, rows, cols);
```

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);


void setup()
{
   SPI.begin();       // Init SPI bus
   mfrc522.PCD_Init(); // Init MFRC522
   Serial.println("Arduino RFID reading UID");

   lcd.begin(16,2);
   servo1.attach(2 );
   servo2.attach(3 );
   servo3.attach(4);
   Serial.begin(9600);
}



void loop()
{ servo1.write(160);
  char Key = myKeypad.getKey();
 Serial.print(Key);
  if(Key == '1')
  {
  lcd.setCursor(0, 0);
  lcd.print("Game of Thrones");
  servo2.write(100);
 delay(1550);
  servo3.write(150);
  delay(1000);
  servo2.write(95);
 delay(100);
  servo2.write(90);
 delay(100);
  servo2.write(85);
 delay(100);
    servo2.write(80);
 delay(100);
  servo2.write(75);
 delay(100);
  servo2.write(70);
 delay(1550);
```

```
if ( mfrc522.PICC_IsNewCardPresent())
  {
      if ( mfrc522.PICC_ReadCardSerial())
      {
         lcd.clear();
         Serial.print("Tag UID:");
         lcd.setCursor(0,0);
         lcd.print("Tag UID:");
         lcd.setCursor(0,1);
         for (byte i = 0; i < mfrc522.uid.size; i++) {
                Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                Serial.print(mfrc522.uid.uidByte[i], HEX);

                lcd.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                lcd.print(mfrc522.uid.uidByte[i], HEX);
            //  lcd.print(" ");
          }

          Serial.println();

          int i = 0;
          boolean match = true;
          while(i<mfrc522.uid.size)
          {

             if(!(int(mfrc522.uid.uidByte[i]) == int(code[i])))
             {
                match = false;
             }
            i++;
          }

          delay(3000);
         lcd.clear();
         lcd.setCursor(0,0);


         if(match)
         {

            lcd.print("Book found");

            servo1.write(160);
```

```
              delay(1000);
              servo1.write(125);
              delay(1000);
              servo2.write(100);
              delay(1000);
              servo3.write(180);
              delay(1000);
              servo2.write(30);
              delay(1000);
              servo1.write(160);
              delay(1000);

           }

           else{

              lcd.print(" Book not Found  ");
             Serial.println("\Book not Found");
           }

             Serial.println("============================");

             mfrc522.PICC_HaltA();

             delay(3000);
        }
}










  }


  if(Key == '2')
  {
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
  lcd.print("Book Return");


            servo1.write(160);
            delay(1000);
            servo1.write(125);
            delay(1000);
            servo2.write(100);
            delay(1000);
            servo3.write(150);
            delay(1000);
            servo2.write(70);
            delay(1000);
            servo1.write(160);
            delay(1000);
/*



  if ( mfrc522.PICC_IsNewCardPresent())
    {
       if ( mfrc522.PICC_ReadCardSerial())
       {
         lcd.clear();
         Serial.print("Tag UID:");
         lcd.setCursor(0,0);
         lcd.print("Tag UID:");
         lcd.setCursor(0,1);
         for (byte i = 0; i < mfrc522.uid.size; i++) {
               Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
               Serial.print(mfrc522.uid.uidByte[i], HEX);

               lcd.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
               lcd.print(mfrc522.uid.uidByte[i], HEX);
             //  lcd.print(" ");
          }

          Serial.println();

          int i = 0;
          boolean match = true;
          while(i<mfrc522.uid.size)
          {
```

```
            if(!(int(mfrc522.uid.uidByte[i]) == int(code[i])))
            {
              match = false;
            }
           i++;
         }

         delay(3000);
        lcd.clear();
        lcd.setCursor(0,0);


        if(match)
        {

          lcd.print("Book Returned");

          servo1.write(160);
          delay(1000);
          servo1.write(125);
          delay(1000);
          servo3.write(20);
          delay(1000);
          servo2.write(30);
          delay(1000);
          servo1.write(160);
          delay(1000);

        }

        else{

          lcd.print(" Access denied  ");
         Serial.println("\nUnknown Card");
        }

         Serial.println("============================");

         mfrc522.PICC_HaltA();

         delay(3000);
      }
}
```

```
*/





  }


}
```