MagicLock
(MLock)

Nicklas Mortensen Hamang
Soheil Montaseri
Huy Ba Nguyen
Thach Khoi Pham

3. Vi ønsket å lage en lås som gjorde at eieren skulle slippe å være beskymret for å ha glemt å låst døren etter å ha dratt hjemmefra. Da også med å slippe å måtte bære en nøkkel, så da i dagens verden var det da enkelt erstattet med en RFID av formen til et vanlig kort. Vår løsning gjør da dette virkelig ved å ha døren låst. helt til en bestemt metode brukes for å åpne den, her er det da RFID som blir brukt. Vi har da også lagt til smartphone muligheter.

4. Videoen vår viser først en scenario som viser hvordan den brukes for å åpne en dør. Først med RFID(tag/brikke), så viser den bruken av applikasjonen. Etter det vises rask hvordan arduino'en brukes vi LCD'en. Til slutt har vi noen bilder av hvordan den ser ut både inn- og utvendig.

5. https://www.youtube.com/watch?v=th3AEW9aveA&feature=youtu.be

6.1 Utstyr:

Software/ Hardware:	**Beskrivelse:**
Arduino Mega 2560	Mega er den vi bruker til å styre komponenetene.
Annikken Andee shield	Kobles på arduino mega får å sende info over til enheten(GUI til for eks. mobil).
Kabler	Mange
TFT_320QVT (LCD)	Touchskjerm module.
MFRC-522 RFID sensor	Sensor som lesere opp de forskjellig tag/kort.
"lås"	Elektronisk lås.
"Video program"	Til å redigere video som skal leveres inn.
En iphone/ipad	Til kommunikasjon mellom arduino og den trådløsenheten).
Platikkboks	
usb strøm/batteri.	Går enten på usb strøm eller på DC

	strøm(batteri)
GoogleDocs	Rapport skriving.

6.2 Koden:

```
#include <Andee.h>
#include <UTFT.h>
#include <UTouch.h>
#include <SPI.h>
#include <RFID.h>
#include <avr/pgmspace.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>



#define SS_PIN 48
#define RST_PIN 49

//Defines the sccreen's max x & y
#define XMAX 319
#define YMAX 239

// Initialize display
// ------------------
// Set the pins to the correct ones for your development board
// -----------------------------------------------------------
// Standard Arduino Uno/2009 Shield            : <display model>,19,18,17,16
// Standard Arduino Mega/Due shield            : <display model>,38,39,40,41
// CTE TFT LCD/SD Shield for Arduino Due       : <display model>,25,26,27,28
// Teensy 3.x TFT Test Board                   : <display model>,23,22, 3, 4
// ElecHouse TFT LCD/SD Shield for Arduino Due : <display model>,22,23,31,33
//
// Remember to change the model parameter to suit your display module!
UTFT    myGLCD(SSD1289,38,39,40,41);



//Sets the icon images
extern unsigned int settings[0x400];
extern unsigned int addIcon[0x400];
extern unsigned int closeIcon[0x400];
extern unsigned int deleteIcon[0x400];
extern unsigned int lock[0x400];
extern unsigned int notes[0x400];
extern unsigned int timerIcon[0x400];

//Set's the RFID's pins
RFID rfid(48, 49);



//Info for all cards
typedef struct cards{
  int serNum0;
  int serNum1;
  int serNum2;
  int serNum3;
  int serNum4;
```

```
  int admin;
  int deleteUser;
  int timer;
  char name[20];

} card;



// Initialize touchscreen
// ----------------------
// Set the pins to the correct ones for your development board
// -----------------------------------------------------------
// Standard Arduino Uno/2009 Shield            : 15,10,14, 9, 8
// Standard Arduino Mega/Due shield            :  6, 5, 4, 3, 2
// CTE TFT LCD/SD Shield for Arduino Due       :  6, 5, 4, 3, 2
// Teensy 3.x TFT Test Board                   : 26,31,27,28,29
// ElecHouse TFT LCD/SD Shield for Arduino Due : 25,26,27,29,30
//
UTouch  myTouch( 6, 5, 4, 3, 2);

// Declare which fonts we will be using
extern uint8_t BigFont[];

int x, y;
char stCurrent[20]="";
int stCurrentLen=0;
char stLast[20]="";
int n=1;

AndeeHelper button;
AndeeHelper slider;
AndeeHelper displayBox;

int setTimer=0;

int nC, nD, nS =0;

card Cards[20];



//lager et testkort
void testCard(){
  Cards[0].serNum0=187;
  Cards[0].serNum1=232;
  Cards[0].serNum2=93;
  Cards[0].serNum3=116;
  Cards[0].serNum4=122;
  strcpy(Cards[0].name, "Test");
  Cards[0].timer=5;
  Cards[0].admin=1;
}



/*****************
***Draw buttons***
*****************/
```

```
//Draws delete buttons
void drawDeleteButton(){
  myGLCD.clrScr();
  //Buttons
  myGLCD.setColor(0,0,255);
  myGLCD.fillRect(XMAX-100, YMAX-45, XMAX-5, YMAX-5);
  myGLCD.fillRect(XMAX-210, YMAX-45, XMAX-110, YMAX-5);
  myGLCD.drawBitmap(287, 5, 32, 32, closeIcon);

  //Arrow buttons
  myGLCD.setColor(155, 155, 155);
  myGLCD.fillRoundRect(40, 100, 80, 140);
  myGLCD.fillRoundRect(245, 100, 285, 140);

  //Name textbox
  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRect(85, 100, 240, 140);

  //Print buttons
  myGLCD.setBackColor(0,0,255);
  myGLCD.setColor(255, 255, 255);
  myGLCD.print("Done", XMAX-95, YMAX-35);
  myGLCD.print("Delete", XMAX-205, YMAX-35);

  myGLCD.setBackColor(155,155,155);
  myGLCD.print("<-", 45, 110);
  myGLCD.print("->", 250, 110);




}

//Draws Note buttons
void drawNoteButton(){

}



//Draws the keyboard when needed
void drawKeyboard(){
  //myGLCD.clrScr();
  int y1=120;
  int y2=153;
  int y3=186;
  int x1=30;
  int x2=58;
  int x3=86;
  int x4=114;
  int x5=142;
  int x6=170;
  int x7=198;
  int x8=226;
  int x9=254;
  int x10=282;
  myGLCD.drawBitmap(287, 5, 32, 32, closeIcon);
```

```
/*
  myGLCD.setColor(255, 255, 255);

  myGLCD.fillRect(25, 45, 240, 70);
  myGLCD.setColor(155, 155, 155);
  myGLCD.fillRoundRect(255, 45, 285, 70);
  */
  //Buttons
  myGLCD.setColor(0,100,255);

  myGLCD.fillRoundRect(5, y1, x1, 150);
  myGLCD.fillRoundRect(x1+3, y1, x2, 150);
  myGLCD.fillRoundRect(x2+3, y1, x3, 150);
  myGLCD.fillRoundRect(x3+3, y1, x4, 150);
  myGLCD.fillRoundRect(x4+3, y1, x5, 150);
  myGLCD.fillRoundRect(x5+3, y1, x6, 150);
  myGLCD.fillRoundRect(x6+3, y1, x7, 150);
  myGLCD.fillRoundRect(x7+3, y1, x8, 150);
  myGLCD.fillRoundRect(x8+3, y1, x9, 150);
  myGLCD.fillRoundRect(x9+3, y1, x10, 150);
  myGLCD.fillRoundRect(x10+3, y1, 310, 150);



  myGLCD.fillRoundRect(5+3, y2, x1+3, 183);
  myGLCD.fillRoundRect(x1+6, y2, x2+3, 183);
  myGLCD.fillRoundRect(x2+6, y2, x3+3, 183);
  myGLCD.fillRoundRect(x3+6, y2, x4+3, 183);
  myGLCD.fillRoundRect(x4+6, y2, x5+3, 183);
  myGLCD.fillRoundRect(x5+6, y2, x6+3, 183);
  myGLCD.fillRoundRect(x6+6, y2, x7+3, 183);
  myGLCD.fillRoundRect(x7+6, y2, x8+3, 183);
  myGLCD.fillRoundRect(x8+6, y2, x9+3, 183);
  myGLCD.fillRoundRect(x9+6, y2, x10+3, 183);
  myGLCD.fillRoundRect(x10+6, y2, 315+3, 183);

  myGLCD.fillRoundRect(5, y3, x1, 216);
  myGLCD.fillRoundRect(x1+3, y3, x2, 216);
  myGLCD.fillRoundRect(x2+3, y3, x3, 216);
  myGLCD.fillRoundRect(x3+3, y3, x4, 216);
  myGLCD.fillRoundRect(x4+3, y3, x5, 216);
  myGLCD.fillRoundRect(x5+3, y3, x6, 216);
  myGLCD.fillRoundRect(x6+3, y3, x7, 216);
  myGLCD.fillRoundRect(x7+3, y3, x8, 216);
  myGLCD.fillRoundRect(x8+3, y3, x9, 216);
  //myGLCD.fillRoundRect(x9+3, y3, x10, 216);
  //myGLCD.fillRoundRect(x10+3, y3, 310, 216);
  //End buttons

  //Button letters
  myGLCD.setBackColor(0,100,255);
  myGLCD.setColor(255,255,255);
  myGLCD.print("q", 8, y1+5);
  myGLCD.print("w", x1+3, y1+5);
  myGLCD.print("e", x2+3, y1+5);
  myGLCD.print("r", x3+3, y1+5);
  myGLCD.print("t", x4+3, y1+5);
  myGLCD.print("y", x5+3, y1+5);
  myGLCD.print("u", x6+3, y1+5);
  myGLCD.print("i", x7+3, y1+5);
```

```
  myGLCD.print("o", x8+3, y1+5);
  myGLCD.print("p", x9+3, y1+5);
  myGLCD.print("", x10+3, y1+5);

  myGLCD.print("a", 11, y2+5);
  myGLCD.print("s", x1+6, y2+5);
  myGLCD.print("d", x2+6, y2+5);
  myGLCD.print("f", x3+6, y2+5);
  myGLCD.print("g", x4+6, y2+5);
  myGLCD.print("h", x5+6, y2+5);
  myGLCD.print("j", x6+6, y2+5);
  myGLCD.print("k", x7+6, y2+5);
  myGLCD.print("l", x8+6, y2+5);
  myGLCD.print("<-", x9+6, y2+5);
  myGLCD.print("", x10+6, y2+5);

  myGLCD.print("z", 8, y3+5);
  myGLCD.print("x", x1+3, y3+5);
  myGLCD.print("c", x2+3, y3+5);
  myGLCD.print("v", x3+3, y3+5);
  myGLCD.print("b", x4+3, y3+5);
  myGLCD.print("n", x5+3, y3+5);
  myGLCD.print("m", x6+3, y3+5);
  myGLCD.print("-", x7+3, y3+5);
  myGLCD.print("", x8+3, y3+5);
  myGLCD.print("", x9+3, y3+5);
  myGLCD.print("", x10+3, y3+5);
  //end button letters
}

//Draws homescreen
void drawButtons(){
  myGLCD.clrScr();
  Serial.println("Draw buttons");
  //myGLCD.setColor(0, 0, 0);
  //myGLCD.fillRect(0, 0, 319, 239);
  myGLCD.drawBitmap(100, 99, 64, 64, lock);
  myGLCD.drawBitmap(174, 99, 64, 64, settings);
  //myGLCD.drawBitmap(287, 5, 32, 32, closeIcon);
  myGLCD.setColor(255, 0, 0);
}

//Draws the buttons for timer screen
void drawTimerButtons()
{
  myGLCD.clrScr();
  myGLCD.drawBitmap(287, 5, 32, 32, closeIcon);

  myGLCD.setBackColor(0,0,0);
  myGLCD.setColor(255,255,255);

  myGLCD.fillRect(40, 100, 80, 130);
  myGLCD.fillRect(90, 100, 130, 130);
  myGLCD.fillRect(140, 100, 180, 130);

  myGLCD.print("Current: ", 5, 15);

  myGLCD.setColor(0,0,255);
  myGLCD.fillRoundRect(XMAX-45, YMAX-45, XMAX-5, YMAX-5);
```

```
  myGLCD.setBackColor(0,0,255);
  myGLCD.setColor(255,255,255);
  myGLCD.print("OK", XMAX-40, YMAX-40);

  myGLCD.setColor(0,0,0);
  myGLCD.setBackColor(255, 255, 255);

  /*
  myGLCD.print("1", 50, 105);
  myGLCD.print("2", 100, 105);
  myGLCD.print("3", 150, 105);
  */

  myGLCD.setColor(155, 155, 155);
  myGLCD.fillRect(40, 80, 80, 100);
  myGLCD.fillRect(90, 80, 130, 100);
  myGLCD.fillRect(140, 80, 180, 100);

  myGLCD.fillRect(40, 130, 80, 150);
  myGLCD.fillRect(90, 130, 130, 150);
  myGLCD.fillRect(140, 130, 180, 150);

  myGLCD.setBackColor(155,155,155);
  myGLCD.setColor(0,0,0);

  myGLCD.print("+", 50, 80);
  myGLCD.print("+", 100, 80);
  myGLCD.print("+", 150, 80);

  myGLCD.print("-", 50, 130);
  myGLCD.print("-", 100, 130);
  myGLCD.print("-", 150, 130);



 /* |----------|
 *  |    +     |
 *  |----------|
 * for opp, med - for ned.
 */
}


//Draws the buttons for Settings
void drawSettingButton(){

  Serial.println("Draw settings button");
  myGLCD.clrScr();
  myGLCD.drawBitmap(15, 90, 64, 64, timerIcon);
  myGLCD.drawBitmap((15+64+10), 90, 64, 64, addIcon);
  myGLCD.drawBitmap(15+(64+10)*2, 90, 64, 64, deleteIcon);
  //myGLCD.drawBitmap(15+(64+10)*3, 90, 64, 64, notes);
  myGLCD.drawBitmap(287, 5, 32, 32, closeIcon);

}
```

```
/*******************
***Work functions***
*******************/
//updates the numbers in timer
int updateTimer(int c, int d, int s){

  int tot;

  if(nC==9){
    if(c==1){
      nC=-1;
    }
  }
  if(nC==0){
    if(c==-1){
      nC=10;
    }
  }
  if(nD==9){
    if(d==1){
      nD=-1;
    }
  }
  if(nD==0){
    if(d==-1){
      nD=10;
    }
  }
  if(nS==9){
    if(s==1){
      nS=-1;
    }
  }
  if(nS==0){
    if(s==-1){
      nS=10;
    }
  }

  nC+=c;
  nD+=d;
  nS+=s;

  tot=(nC*100)+(nD*10)+nS;

  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRect(40, 100, 80, 130);
  myGLCD.fillRect(90, 100, 130, 130);
  myGLCD.fillRect(140, 100, 180, 130);

  myGLCD.setBackColor(255,255,255);
  myGLCD.setColor(0,0,0);

  myGLCD.printNumI(nC, 50, 105);
  myGLCD.printNumI(nD, 100, 105);
  myGLCD.printNumI(nS, 150, 105);
```

```
  Serial.println(tot);

  return tot;
}

//Funksjonen til timer
void timerState(int pos){
  drawTimerButtons();
  myGLCD.setColor(255, 255, 255);
  myGLCD.setBackColor(0,0,0);
  myGLCD.printNumI(Cards[pos].timer, 235, 15);
  int tmpTimer=0;
  while(true){
    if(myTouch.dataAvailable())
    {
      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();

      if(y>=80 && y<=100){
        if(x>=40 && x<=80){
          tmpTimer=updateTimer(1,0,0);
          delay(500);
        }else if(x>=90 && x<=130){
          tmpTimer=updateTimer(0,1,0);
          //tmpTimer+=10;
          delay(500);
        }else if(x>=140 && x<=180){
          tmpTimer=updateTimer(0,0,1);
          //tmpTimer+=1;
          delay(500);
        }

      }else if(y>=130 && y<=150){
        if(x>=40 && x<=80){
          tmpTimer=updateTimer(-1,0,0);
          //tmpTimer-=100;
          delay(500);
        }else if(x>=90 && x<=130){
          tmpTimer=updateTimer(0,-1,0);
          //tmpTimer-=10;
          delay(500);
        }else if(x>=140 && x<=180){
          tmpTimer=updateTimer(0,0,-1);
          //tmpTimer-=1;
          delay(500);
        }
      }else if(y>=5 && y<=39){
        if(x>=287 && x<=287+34){
          myGLCD.clrScr();
          drawSettingButton();
          break;
        }
      }else if(y>=YMAX-45 && y<=YMAX-5){
        if(x>XMAX-45 && XMAX-5){
          if(tmpTimer>0){
            Cards[pos].timer=tmpTimer;
            nC=0;
            nD=0;
```

```
            nS=0;
            Serial.print(tmpTimer);
            myGLCD.clrScr();
            drawSettingButton();
            break;
          }
        }
      }else if(y>=5 && y<=37){
        if(x>=287 && x<=287+32){
          break;
        }
      }
    }
  }
}

//Reads card for timer
void timerReadCard(){
  int j;
  myGLCD.clrScr();
  myGLCD.setColor(255,255,255);
  myGLCD.setBackColor(0,0,0);
  myGLCD.print("Read", 25, YMAX/2);
  int time, start = millis();
  while(time-start<=10000){
    if(rfid.isCard()){
      if(rfid.readCardSerial()){
        int k = checkCard(rfid.serNum[0], rfid.serNum[1], rfid.serNum[2], rfid.serNum[3],
rfid.serNum[4]);
        if(k==-1){
          delay(1000);
          drawSettingButton();
          break;
        }else if(k>=0){
          timerState(k);
          break;
        }
      }
    }
    time=millis();
  }
  myGLCD.clrScr();
  drawSettingButton();
}

//Neste eller forrige kort
void updateDelete(int i){
  Serial.println("In update delete");
  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRect(85, 100, 240, 140);
  myGLCD.setColor(255,255,255);
  //myGLCD.fillRect(x, y, x+140, y+40);
  myGLCD.setColor(0,0,0);
  myGLCD.setBackColor(255,255,255);
  myGLCD.print(Cards[i].name, 90, 110);
}

//Delete's position and move all one down
void deletePos(int pos){
  Serial.println("in delete posistion");
  n--;
```

```
  while(pos<n){
    Cards[pos].serNum0=Cards[pos+1].serNum0;
    Cards[pos].serNum1=Cards[pos+1].serNum1;
    Cards[pos].serNum2=Cards[pos+1].serNum2;
    Cards[pos].serNum3=Cards[pos+1].serNum3;
    Cards[pos].serNum4=Cards[pos+1].serNum4;
    strcpy(Cards[pos].name, Cards[pos+1].name);
    Cards[pos].admin=Cards[pos+1].admin;
    pos++;
  }
    Cards[n].serNum0=NULL;
    Cards[n].serNum1=NULL;
    Cards[n].serNum0=NULL;
    Cards[n].serNum3=NULL;
    Cards[n].serNum4=NULL;
    Cards[n].name[0]='\0';
    Cards[n].admin=NULL;
    updateDelete(0);
}

//Main for delete funksjon
void deleteState(){
  Serial.println("in delete state");
 drawDeleteButton();

 int i, j=0;

 updateDelete(i);
 Serial.println("After update");



 while(true){
  if(myTouch.dataAvailable()){
   myTouch.read();
   x=myTouch.getX();
   y=myTouch.getY();
   if(y>=100 && y<=140){
    if(x>=40 && x<=80){
      Serial.println("back key pressed");
      if(i==0){
        i=n;
      }
      i--;
      updateDelete(i);
      delay(500);
    }else if(x>=245 && x<=285){
      Serial.println("next key pressed");
      if(i==n-1){
       i=-1;
      }
      i++;
      updateDelete(i);
      delay(500);
    }
   }else if(y>=YMAX-45 && y<=YMAX-5){
    if(x>=XMAX-210 && x<=XMAX-110){
      Serial.println("Delete pressed");
      deletePos(i);
      i=0;
      delay(500);
```

```
      //Delete card. ikke sikker hvordan enda
    }else if(x>=XMAX-100 && x<=XMAX-5){
      Serial.println("Done pressed");
      myGLCD.clrScr();
      drawSettingButton();
      break;
    }
   }else if(y>=5 && y<=39){
    if(x>=XMAX-39 && x<=XMAX-5){
      Serial.println("X pressed");
      myGLCD.clrScr();
      drawSettingButton();
      break;
    }
   }
  }
  }

}

//åpner døren med timer fra telefon
int openDoorTimer(int timer){
  int time, start = millis();
  myGLCD.setColor(0, 255, 0);
  myGLCD.fillRect(0,0,XMAX, YMAX);

  myGLCD.setColor(255, 255, 255);
  myGLCD.setBackColor(0, 255, 0);
  myGLCD.print("Opened by phone", 10, YMAX/2);
  displayBox.setData("Door Open");
  displayBox.update();
  while((time-start)<(timer*1000)){

    time=millis();
  }
  displayBox.setData("Door Closed");
  myGLCD.clrScr();
  drawButtons();
}

//Opens up the door
void openDoor(int pos)
{

  Serial.println("open door");
  int time, start = millis();
  myGLCD.setColor(0, 255, 0);
  myGLCD.fillRect(0, 0, 319, 239);

  myGLCD.setColor(255, 255, 255);
  myGLCD.setBackColor(0, 255, 0);
  if(pos>=0){
    myGLCD.print(Cards[pos].name, 25, YMAX/2);
    delay(Cards[pos].timer*1000);
  }else{
    delay(15000);
  }
  myGLCD.clrScr();
  drawButtons();
```

```
}

//Adds next letter to name in addCardTwo
void updateStr(int val)
{
  char alpha[]={'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', 'a', 's', 'd', 'f', 'g', 'h',
'j', 'k', 'l', 'z', 'x', 'c', 'v', 'b', 'n', 'm', '-'};



  myGLCD.setBackColor(255, 255, 255);
  myGLCD.setColor(0, 0, 0);

  if(val==-1){
    if(stCurrentLen>=0){
      stCurrent[stCurrentLen-1]='\0';
      stCurrentLen--;
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRect(25, 45, 240, 70);
      myGLCD.setColor(0,0,0);
      myGLCD.print(stCurrent, 27, 50);
    }
  }else{
    if (stCurrentLen<20)
    {
      stCurrent[stCurrentLen]=alpha[val];
      stCurrent[stCurrentLen+1]='\0';
      stCurrentLen++;
      myGLCD.print(stCurrent, 27, 50);
    }
    else
    {
      /*
      myGLCD.setColor(255, 0, 0);
      myGLCD.print("BUFFER FULL!", CENTER, 192);
      delay(500);
      myGLCD.print("            ", CENTER, 192);
      delay(500);
      myGLCD.print("BUFFER FULL!", CENTER, 192);
      delay(500);
      myGLCD.print("            ", CENTER, 192);
      myGLCD.setColor(0, 255, 0);
      */
    }
  }
}

//Get position of card, return -1 if no card.
int getPos(){
  myGLCD.clrScr();
  myGLCD.print("Reading card", 25, YMAX/2);
  int ser0, ser1, ser2, ser3, ser4, tmp0, tmp1, tmp2, tmp3, tmp4;
  int i=0;
  while(true){
    if(rfid.isCard()){
      if(rfid.readCardSerial()){
        ser0 = rfid.serNum[0]; ser1 = rfid.serNum[1]; ser2 = rfid.serNum[2]; ser3 =
rfid.serNum[3]; ser4 = rfid.serNum[4];

        while(i<n){
```

```
          ser0 = Cards[i].serNum0; ser1 = Cards[i].serNum1; ser2 = Cards[i].serNum2; ser3 =
Cards[i].serNum3; ser4 = Cards[i].serNum4;
          if((ser0 == tmp0) && (ser1 == tmp1) && (ser2 == tmp2) && (ser3 == tmp3) && (ser4 ==
tmp4)){
            return i;
          }
          i++;
        }
        return -1;
      }
    }
 }
}

//Edits the card read
void addCard(int ser0, int ser1, int ser2, int ser3, int ser4){

  int tmpAdmin=0;

  myGLCD.clrScr();

  myGLCD.drawBitmap(287, 5, 32, 32, closeIcon);

  myGLCD.setBackColor(0,0,0);

  myGLCD.setColor(255,255,255);

  myGLCD.printNumI(ser0, 10, 60);
  myGLCD.printNumI(ser1, 70, 60);
  myGLCD.printNumI(ser2, 130, 60);
  myGLCD.printNumI(ser3, 190, 60);
  myGLCD.printNumI(ser4, 250, 60);

  myGLCD.print("Admin;", 10, 100);

  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRect(10, 120, 30, 140);

  myGLCD.setColor(0, 100, 255);
  myGLCD.fillRoundRect(XMAX-90, YMAX-40, XMAX- 10, YMAX-10);

  myGLCD.setColor(255, 255, 255);
  myGLCD.setBackColor(0, 100, 255);
  myGLCD.print("OK", XMAX-70, YMAX-30);



  while(true){
    if (myTouch.dataAvailable()){
      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();
      if(y>=120 && y<=140){
       if(x>=10 && x<=30){
         if (tmpAdmin==0){
           myGLCD.setColor(0,0,0);
           myGLCD.fillRect(15, 125, 25, 135);
           Serial.println("Admin on");
```

```
          tmpAdmin=1;
          delay(500);
        }else if(tmpAdmin==1){
          myGLCD.setColor(255,255,255);
          myGLCD.fillRect(15, 125, 25, 135);
          tmpAdmin=0;
          Serial.println("Admin off");
          delay(500);
        }

       }
      }else if(y>=5&&y<=37){
       if(x>= 287 && x<=287+32){
         break;
       }
     }
      if(y>=YMAX-40 && y<=YMAX-10){
        if(x>=XMAX-90 && x<=XMAX-10){
          addCardTwo(ser0, ser1, ser2, ser3, ser4, tmpAdmin);
          break;
        }
      }
      x=NULL;
      y=NULL;
    }
  }

}

//Add's name to the card
void addCardTwo(int ser0, int ser1, int ser2, int ser3, int ser4, int tmpAdmin){
  int y1=120;
  int y2=153;
  int y3=186;
  int x1=30+3;
  int x2=58+3;
  int x3=86+3;
  int x4=114+3;
  int x5=142+3;
  int x6=170+3;
  int x7=198+3;
  int x8=226+3;
  int x9=254+3;
  int x10=282+3;



  myGLCD.clrScr();
  myGLCD.setColor(255, 255, 255);

  myGLCD.fillRect(25, 45, 240, 70);
  //myGLCD.print(stCurrent, 27, 50);
  myGLCD.setColor(155, 155, 155);
  myGLCD.fillRoundRect(255, 45, 305, 90);
  myGLCD.setBackColor(155, 155, 155);
  myGLCD.setColor(0,0,0);
  myGLCD.print("OK", 265, 65);

  drawKeyboard();
```

```
  char alpha[]={'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', 'a', 's', 'd', 'f', 'g', 'h',
'j', 'k', 'l', 'z', 'x', 'c', 'v', 'b', 'n', 'm'};
  while(true){
   if(myTouch.dataAvailable()){
     myTouch.read();
     x=myTouch.getX();
     y=myTouch.getY();
     if(y>=45 && y<=90){
       if(x>=255 && x<=305){
         if(stCurrentLen){
           Serial.println("Adding card");
           Cards[n].serNum0 = ser0;
           Cards[n].serNum1 = ser1;
           Cards[n].serNum2 = ser2;
           Cards[n].serNum3 = ser3;
           Cards[n].serNum4 = ser4;
           Cards[n].admin=tmpAdmin;
           strcpy(Cards[n].name,stCurrent);
           Cards[n].timer=15;
           stCurrent[0]='\0';
           stCurrentLen=0;
           n++;
           //drawButtons();
           break;
         }
       }
     }else if(y>=5&&y<=37){
       if(x>= 287 && x<=287+32){
         break;
       }
     }else if(y>=y1 && y<=150){
       if(x>=5 && x<=x1 ){
         updateStr(0);
         delay(500);
       }else if(x>=x1+3 && x<=x2){
         updateStr(1);
         delay(500);
       }else if(x>=x2+3 && x<=x3){
         updateStr(2);
         delay(500);
       }else if(x>=x3+3 && x<=x4){
         updateStr(3);
         delay(500);
       }else if(x>=x4+3 && x<=x5){
         updateStr(4);
         delay(500);
       }else if(x>=x5+3 && x<=x6){
         updateStr(5);
         delay(500);
       }else if(x>=x6+3 && x<=x7){
         updateStr(6);
         delay(500);
       }else if(x>=x7+3 && x<=x8){
         updateStr(7);
         delay(500);
       }else if(x>=x8+3 && x<=x9){
         updateStr(8);
         delay(500);
       }else if(x>=x9+3 && x<=x10){
         updateStr(9);
         delay(500);
       }
```

```
}else if(y>=y2 && y<=183){
  if(x>=8 && x<=x1+3 ){
    updateStr(10);
    delay(500);
  }else if(x>=x1+6 && x<=x2+3){
    updateStr(11);
    delay(500);
  }else if(x>=x2+6 && x<=x3+3){
    updateStr(12);
    delay(500);
  }else if(x>=x3+6 && x<=x4+3){
    updateStr(13);
    delay(500);
  }else if(x>=x4+6 && x<=x5+3){
    updateStr(14);
    delay(500);
  }else if(x>=x5+6 && x<=x6+3){
    updateStr(15);
    delay(500);
  }else if(x>=x6+6 && x<=x7+3){
    updateStr(16);
    delay(500);
  }else if(x>=x7+6 && x<=x8+3){
    updateStr(17);
    delay(500);
  }else if(x>=x8+6 && x<=x9+3){
    updateStr(18);
    delay(500);
  }else if(x>=x9+6 && x<=x10+3){
    //Backstep
    updateStr(-1);
    delay(500);
  }
}else if(y>=y3 && y<=216){
  if(x>=8 && x<=x1+3 ){
    updateStr(19);
    delay(500);
  }else if(x>=x1+6 && x<=x2+3){
    updateStr(20);
    delay(500);
  }else if(x>=x2+6 && x<=x3+3){
    updateStr(21);
    delay(500);
  }else if(x>=x3+6 && x<=x4+3){
    updateStr(22);
    delay(500);
  }else if(x>=x4+6 && x<=x5+3){
    updateStr(23);
    delay(500);
  }else if(x>=x5+6 && x<=x6+3){
    updateStr(24);
    delay(500);
  }else if(x>=x6+6 && x<=x7+3){
    updateStr(25);
    delay(500);
  }else if(x>=x7+6 && x<=x8+3){
    updateStr(26);
    delay(500);
  }else if(x>=x8+6 && x<=x9+3){
    updateStr(27);
    delay(500);
  }else if(x>=x9+6 && x<=x10+3){
```

```
          //Backstep
          //updateStr();
          delay(500);
        }
      }
    }
  }
}

//Reads a ne card
void readAddCard()
{
  myGLCD.clrScr();
  myGLCD.setColor(255,255,255);
  myGLCD.setBackColor(0,0,0);
  myGLCD.print("Read new card", 25, 100);
  Serial.println("in addCard");

  if(n>=20)
  {
    myGLCD.clrScr();
    myGLCD.print("Max user's", 50, 219/2);
  }



  int time, start = millis();
  int i=0;
  while(time-start<=15000 && i==0){
   if(rfid.isCard()){
     if(rfid.readCardSerial()){
       int r=checkCard(rfid.serNum[0], rfid.serNum[1], rfid.serNum[2], rfid.serNum[3],
rfid.serNum[4]);
       if(r<0){
         Serial.println("Added card");

         addCard(rfid.serNum[0], rfid.serNum[1], rfid.serNum[2], rfid.serNum[3], rfid.serNum[4]);

         /*
         Cards[n-1].serNum0=rfid.serNum[0];
         Cards[n-1].serNum1=rfid.serNum[1];
         Cards[n-1].serNum2=rfid.serNum[2];
         Cards[n-1].serNum3=rfid.serNum[3];
         Cards[n-1].serNum4=rfid.serNum[4];

         Serial.println("");
         Serial.print("Card: ");
         Serial.print(n);
         Serial.print(" - RFID: ");
         Serial.print(rfid.serNum[0]);
         Serial.print(" - Store: ");
         Serial.print(Cards[n-1].serNum0);
         */

         myGLCD.clrScr();
         drawSettingButton();
         i=1;
         break;
       }else if (r>=0){
         myGLCD.clrScr();
```

```
          myGLCD.print("Card used", 25, YMAX/2);
          i=i;
          delay(1000);
          myGLCD.clrScr();
          drawSettingButton();
          break;
        }
      }
    }
    rfid.halt();
    //Serial.println("Out of loop");
    time=millis();
  }
}

/*
*Return 1 if admin, 0 if user, -1 if no cards
*/
int checkAdmin(int serNum0, int serNum1, int serNum2, int serNum3, int serNum4)
{

}

int checkCard(int ser0, int ser1, int ser2, int ser3, int ser4)
{
  Serial.println("in checkCard");
  int i=0;
  int tmp0;
  int tmp1;
  int tmp2;
  int tmp3;
  int tmp4;

  while(i<n){
    Serial.println("");
    Serial.print("Card: ");
    Serial.print(i);
    Serial.print(" - RFID: ");
    Serial.print(rfid.serNum[0]);
    Serial.print(" - Store: ");

    tmp0 = Cards[i].serNum0;
    tmp1 = Cards[i].serNum1;
    tmp2 = Cards[i].serNum2;
    tmp3 = Cards[i].serNum3;
    tmp4 = Cards[i].serNum4;

    Serial.print(Cards[i].serNum0);
    if(tmp0==ser0
    && tmp1==ser1
    && tmp2==ser2
    && tmp3==ser3
    && tmp4==ser4){
      Serial.println("\nOpen door\n");

      return i;
   }
   i+=1;
  }
```

```
  return -1;

}


//Opens up Settings
void settingState(){
  drawSettingButton();
  while(true)
  {
    if(myTouch.dataAvailable())
    {
      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();

     if(y>=90 && y<=154)
     {
       if(x>=15 && x<=(15+64)){
        timerReadCard();
       }else if(x>=(15+64+10) && x<=(15+10+(64)*2)){
        readAddCard();
       }else if(x>=(15+10+(64)*2)+10 && x<= (15+10+(64)*2)+10+64){
         deleteState();
       }/*else if(x>=((15+10+(64)*2)+10+64)+10 && x<= (15+10+(64)*2)+10+64+10+64){
        //notes
       }*/

     }
    if(y>=5 && y<=39)
    {
      if(x>=287 && x<=(287+34)){
        myGLCD.clrScr();
        drawButtons();
        break;
      }
    }
  }
  if(rfid.isCard())
    {
      if (rfid.readCardSerial())
      {
        //Serial.println("Lest kort");
        int r = checkCard(rfid.serNum[0], rfid.serNum[1], rfid.serNum[2], rfid.serNum[3],
rfid.serNum[4]);
        if(r>=0){
          openDoor(r);
        }
        if(r==0){
         Serial.println("Card not found");
        //Lest kort, send til sjekk
      }


    }
  }
  rfid.halt();
  }
}
```

```
void setInitialData(){

  displayBox.setId(0);
  displayBox.setType(DATA_OUT);
  displayBox.setLocation(0,0,FULL);
  displayBox.setTitle("Status");
  displayBox.setData("Door Closed");
  displayBox.setTitleColor(WHITE);
  displayBox.setTitleTextColor(TEXT_DARK);

  slider.setId(1);
  slider.setType(SLIDER_IN);
  slider.setLocation(1,0,FULL);
  slider.setTitle("Timer");
  slider.setSliderMinMax(0, 120);
  slider.setSliderInitialValue(setTimer);
  slider.setSliderNumIntervals(121);
  slider.setSliderReportMode(ON_VALUE_CHANGE);
  slider.setSliderColor(THEME_RED);
  slider.setColor(THEME_RED_DARK);

  button.setId(2);
  button.setType(BUTTON_IN);
  button.setLocation(2,0, FULL);
  button.setTitle("Open Door");
  button.requireAck(true);

}

/************************
**  Required functions  **
************************/
void setup()
{
// Initial setup
  SPI.begin();
  rfid.init();
  Andee.begin();
  Andee.clear();

  setInitialData();

  Serial.begin(9600);
  Serial.println("Setup");
  myGLCD.InitLCD();
  myGLCD.clrScr();

  testCard();

  myTouch.InitTouch();
  myTouch.setPrecision(PREC_MEDIUM);

  myGLCD.setFont(BigFont);
  myGLCD.setBackColor(255, 255, 255);
  drawButtons();
}
```

```
void loop()
{
  Serial.println("loop");
  while(true){

    setTimer=slider.getSliderValue(INT);

    if(button.isPressed()){
      button.ack();
      openDoorTimer(setTimer);
    }
    //Serial.println("before touch");
    if(myTouch.dataAvailable())
    {
      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();

      if(x>=100 && x<=164)
      {
        if(y>=99 && y<=163)
        {
          //drawKeyboard();
          openDoor(-1);
        }

      }
      if(x>=174 && x<=238){
       if(y>=99 && y<=163){
         settingState();
       }
      }
    }

    //Serial.println("before card");
    if(rfid.isCard())
    {
      if (rfid.readCardSerial())
      {
        //Serial.println("Lest kort");
        int r = checkCard(rfid.serNum[0], rfid.serNum[1], rfid.serNum[2], rfid.serNum[3],
rfid.serNum[4]);
        if(r>=0){
          openDoor(r);
        }
        if(r==0){
         Serial.println("Card not found");
        //Lest kort, send til sjekk
      }

    }
  }
  rfid.halt();
  slider.update();
  button.update();
  displayBox.update();
}
}
```