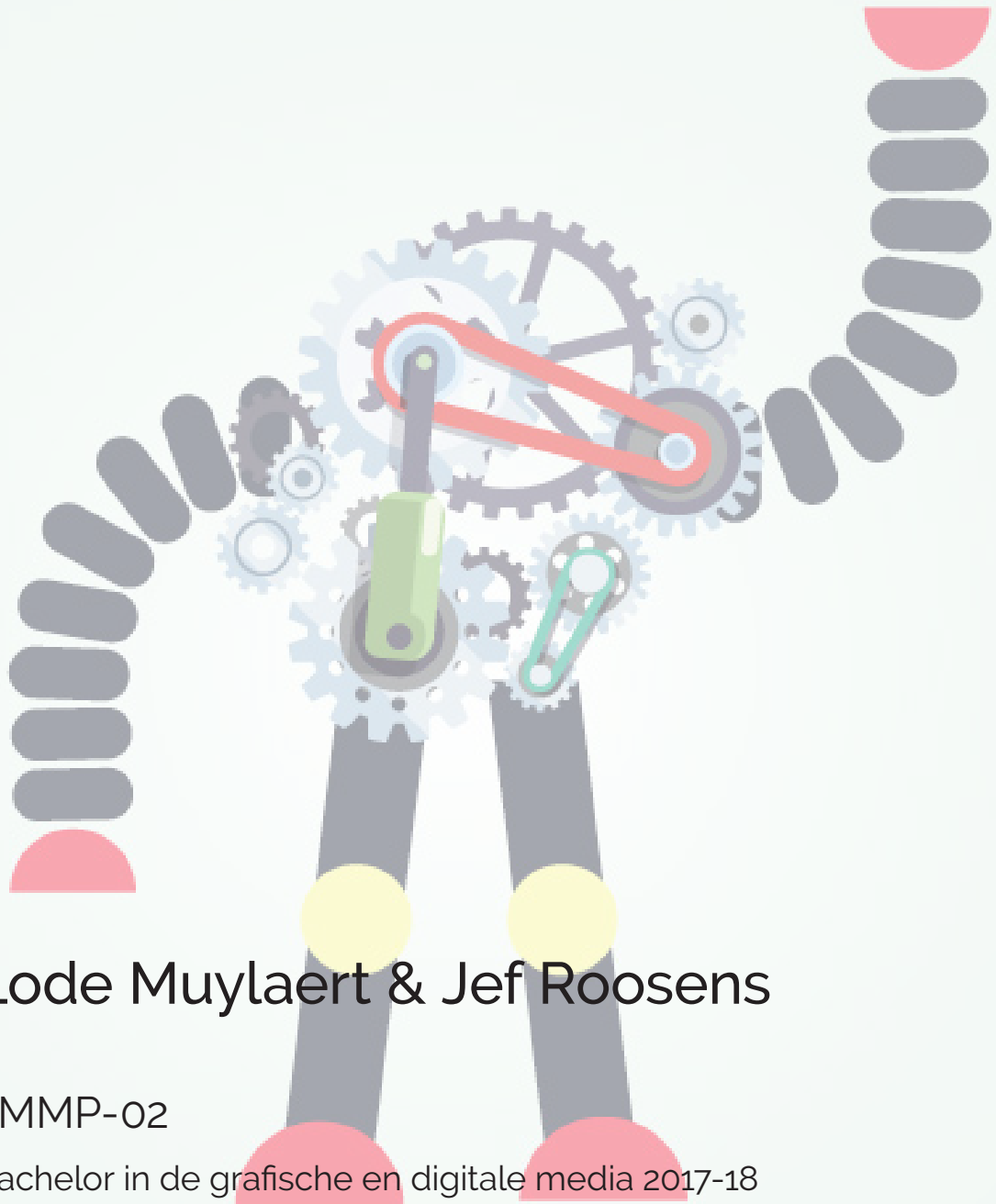


Roboplot



Lode Muylaert & Jef Roosens

3MMP-02

Bachelor in de grafische en digitale media 2017-18

Web Of Things

Arteveldehogeschool

INHOUDSTAFEL

1.	Discover	5
2.	Define	6
3.	Design	14
4.	Development	23
5.	Deliver	28
6.	Deploy	29
7.	Foto's	29

1. DISCOVER

BRIEFING

Algemene omschrijving van de opdracht

We ontwikkelen een softwaresysteem die realtime communicatie kan realiseren tussen gebruikers, software en hardware in de vorm van een robot.

Probleemstelling

realtime communicatie tussen gebruikers is vaak niet optimaal uitgewerkt.

Wat wil de opdrachtgever

De opdrachtgever is in dit geval de Arteveldehogeschool. Zij willen dat wij een systeem ontwikkelen die de realtime communicatie tussen robot en gebruiker weergeeft.

- Gesteld op privacy en integriteit(login)
- Geheel en zonder tekortkomingen.
- Gebruiker wil de robot kunnen aansturen
- Gebruiker wil opnames kunnen registreren van de afgelegde weg
- Gebruiker wil opnames kunnen afspelen
- Gebruiker wil opnames kunnen beheren aan de hand van een webapp

Wie behoort tot het doelpubliek?

Demografische profilering

- Wie
 - » ouderen(robots voor fysieke hulp) & jongeren(speelgoedautotjes)
- Leeftijd:
 - » Alle leeftijden



- Rollen:
 - » Gebruiker van de robot
 - » Visitor van de webapp(geen toegang)
 - » User van de webapp(toegan)

Wat is het budget

Er is geen budget voorzien voor deze opdracht . De opdracht is educatief.

Wat is de timing?

Tegen 12/01/2018 moet het volledige stelsysteem klaar zijn.

2. DEFINE

ANALYSE

Onze eerste taak bij het horen van deze opdracht was op zoek gaan naar het juiste materiaal. We hadden veel verschillende onderdelen nodig voor het laten rijden van onze robot, uitsturen van waarschuwingssignalen en het besturen.

Voor het besturen van de robot hebben besloten om een PS3 controller te gebruiken. Niet alleen werk deze draadloos over een bluetooth signaal, ook de mogelijkheden van de knoppen zijn eindeloos. Je hebt een variatie tussen buttons, triggers en joysticks die allemaal zowel analoog als digitaal kunnen worden uitgelezen. Door deze keuze zullen we tijdens het ontwikkelen niet te snel stoten op limieten die onze controller opstelt.

De robot zelf is gebaseerd op een M-Bot frame met twee 9v DC motoren. Deze zijn krachtig genoeg om alle ballast te kunnen dragen en zijn ook perfect compatibel met de L293D driver die we gebruiken voor het aansturen van de motor.

Als waarschuwingssignalen hebben we gekozen voor twee led lampjes en een actieve 5v buzzer. Aangezien er tijdens het rijden geen display kan aangesloten worden door de beperking van de externe batterij moeten we op een andere manier signalen kunnen uitsturen. Bij het opstarten,



niet vinden van paden of afsluiten zullen deze in werking moeten treden.

Wat de applicatie betreft zijn we verder gegaan op de vereisten die zijn opgesteld in de opdracht. Firebase wordt door ons gebruikt als database en Vue als Javascript Framework. Firebase is nieuw voor ons maar het is altijd leuk om iets bij te leren. Met Vue hadden we beiden al ervaring dus was het niet nodig om veel tijd te steken in het volgen van tutorials. Ook Firebase is heel intuïtief en gebruiksvriendelijk. Hiervoor is er reeds voldoende documentatie te vinden op het net.

SPECIFICATIES

Keuze van technologie

- Webapp in vuejs
- Python voor op de robot

Wat gaan we doen ?

- Ontwikkeling robot en webapp
- geen promotie maken
- Permanente ondersteuning bieden

Technische Specificaties

- **Tools**
- Adobe Creative Cloud
 - » Adobe Illustrator CC
 - » Adobe InDesign CC
 - » Adobe Premiere pro
- VisualCode editor
- **Python libraries**
- Pygame



- names
- firebase-admin
- rpi.gpio
- Json
- io
- math
- time
- sys
- **Ontwikkelomgeving**
- Dotfiles
 - » Powershell
- Git
- python3
- Heroku
- Vue
- **Automatisatie**
- Yarn
- Webpack
- Sass
- **Serverside**
- Firebase database
- Firebase auth
- **Clientside**
- HTML
- Sass(SCSS)
- Webpack
- Vue.js 2



Functionele Specificaties

- Robot
 - » aansturing
 - » recorden
 - » Afspelen
 - » Lampjes
 - » Beeper
- Webapp
 - » Overzicht actief records in playlist
 - » Overzicht actieve record in de playlist
 - » Authenticatie
 - » Create, read & delete records

Niet-functionele Specificaties

- Alle scripting-code wordt gedocumenteerd met docblocks
- Webapp grafisch verantwoord, esthetisch verzorgd en gebruiksvriendelijk.
- Content moet echt representatief zijn

Sociale media integreren, toepassen of niet

Er zal geen gebruik gemaakt worden van sociale media.

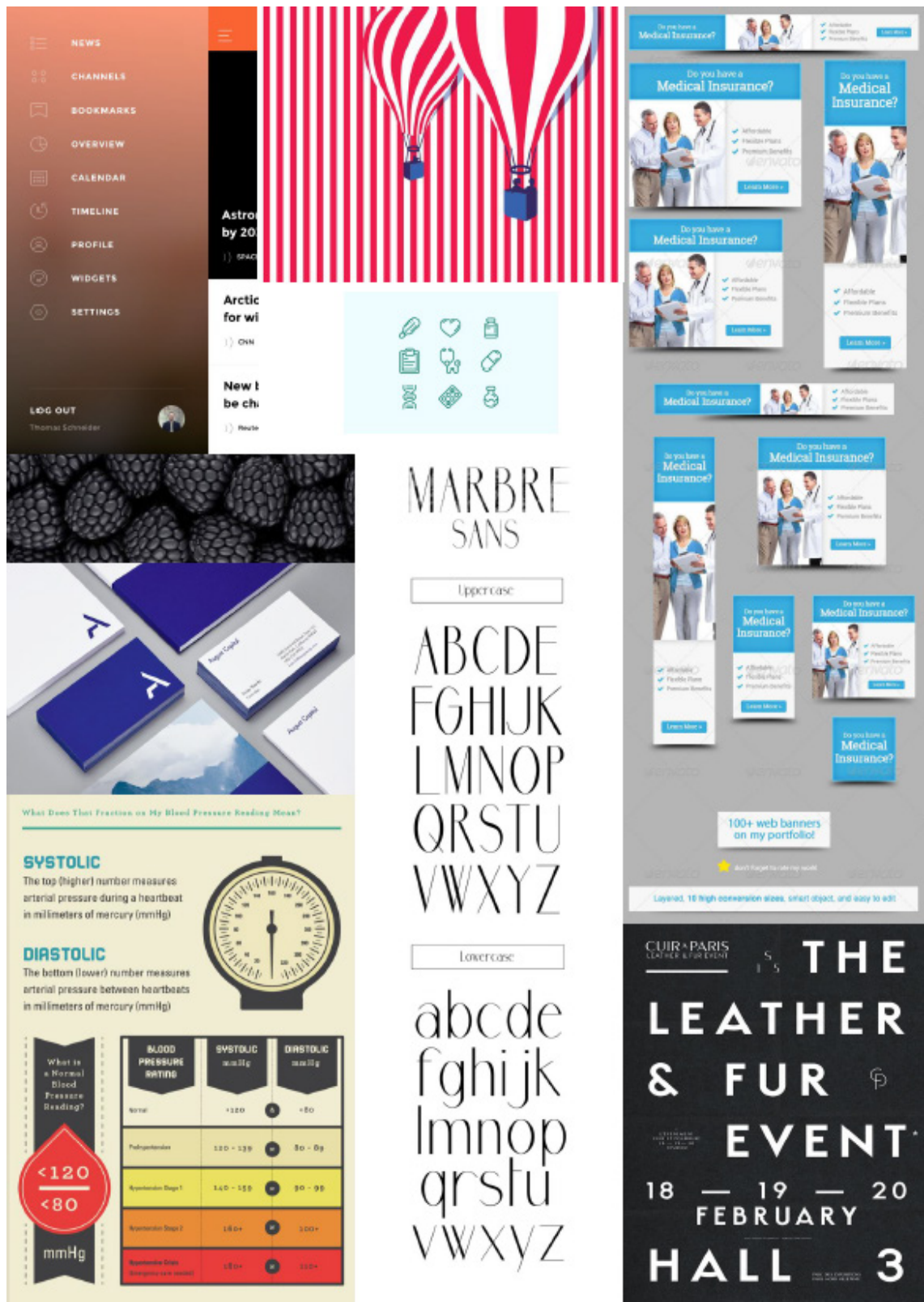


PLANNING

14/11 - 21/11	<ul style="list-style-type: none">- Bepalen van de hardware- Test opstelling maken- Controller PS3 uitlezen
22/11 - 19/12	<ul style="list-style-type: none">- Uitgewerkte testopstelling maken- Controleren van motoren met PS3 controller
20/12 - 26/12	<ul style="list-style-type: none">- Opbouwen van een concreet bestuurbaar model
27/12 - 2/12	<ul style="list-style-type: none">- Connectie maken met firebase en Vue framework
3/12 - 11/12	<ul style="list-style-type: none">- Afwerken project- Opmaken productiedossier



INSPIRATION



Roboplot

Pagina 11

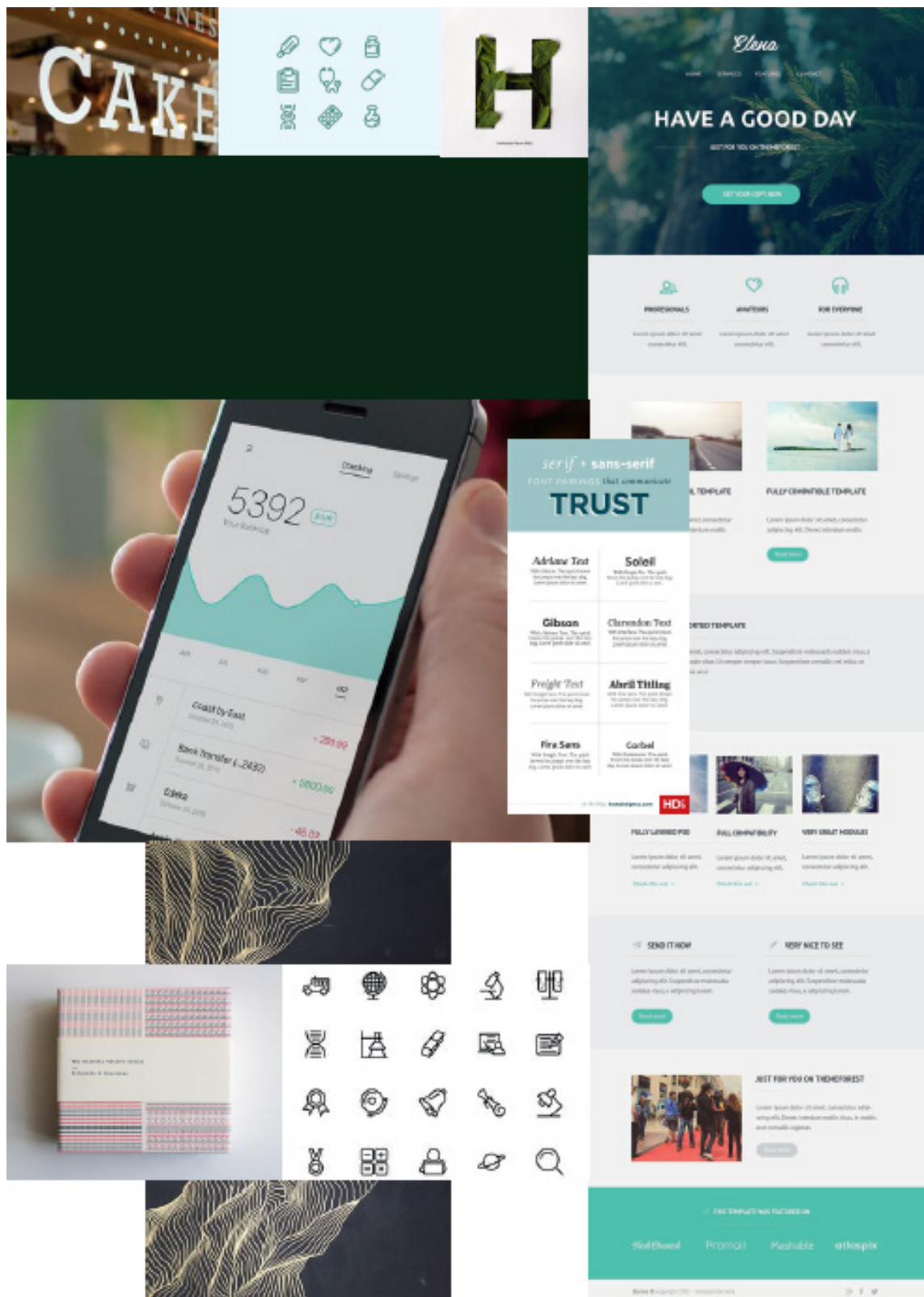




- 01 Gotham
- 02 **BEBAS NEUE**
- 03 BonvenoCF
- 04 **LOVELO**
- 05 OSTRICH SANS

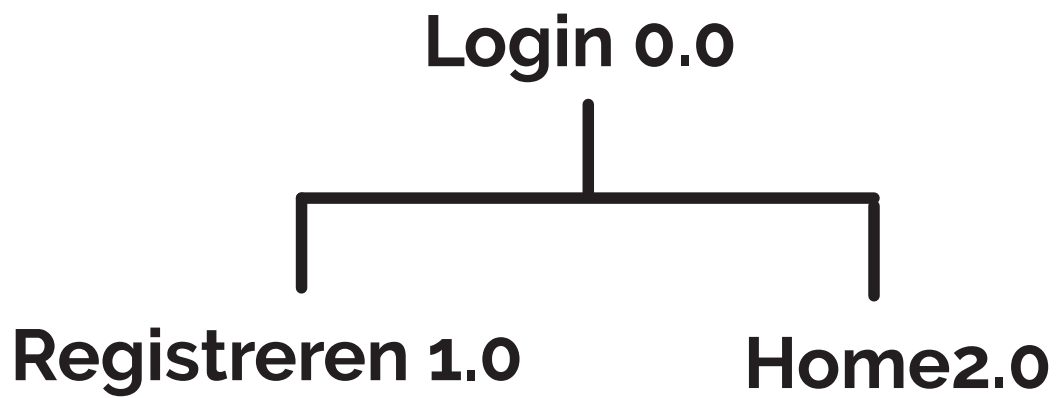
Kaladocopy

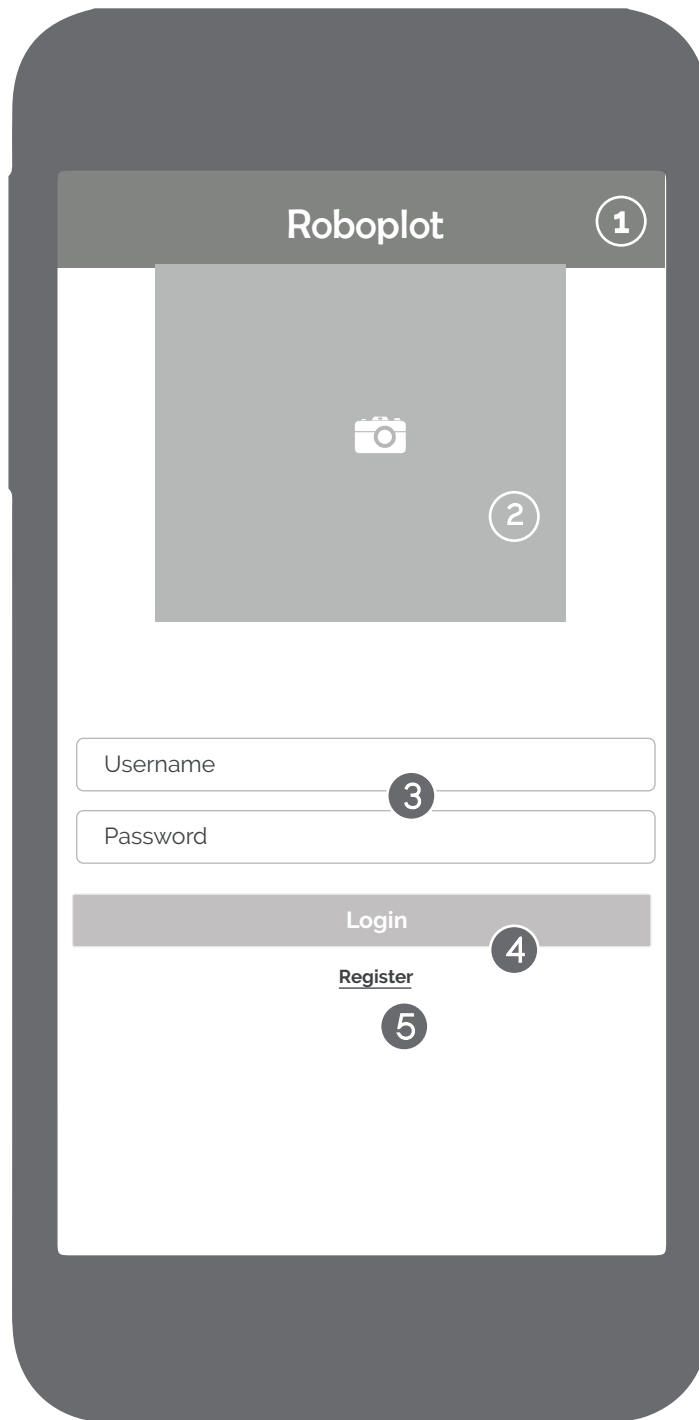




3. DESIGN

SITEMAP WEBSITE

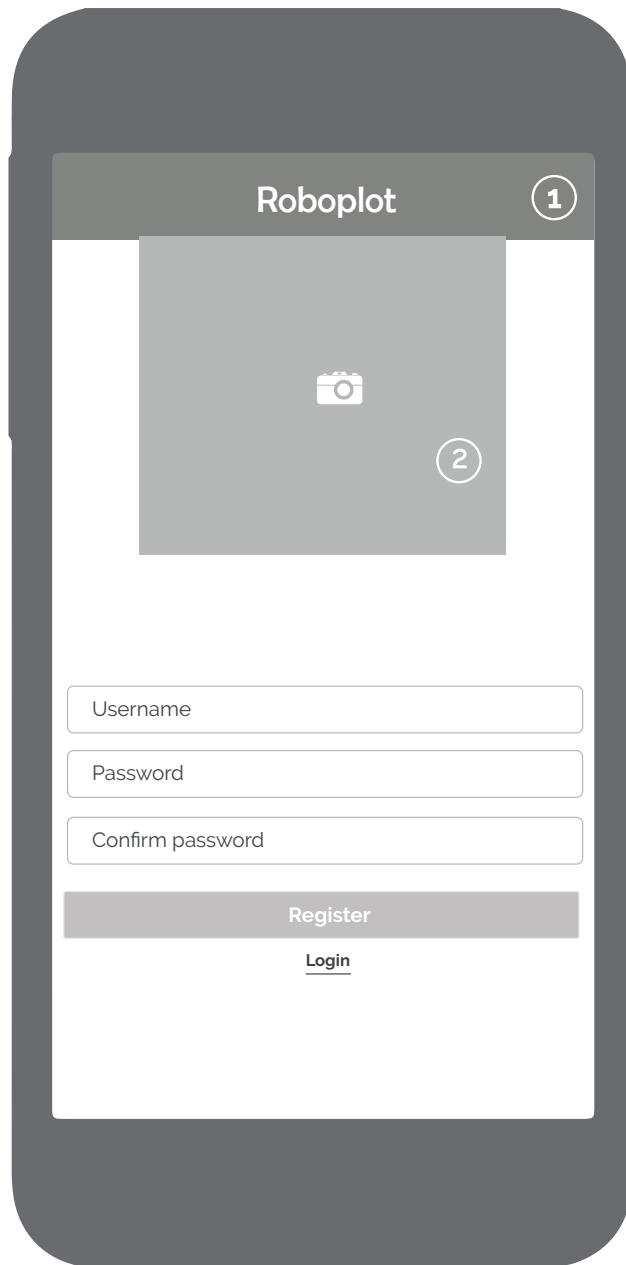




O.O LOGIN

- ① header
- ② Logo
- ③ Username en password
- ④ Login
- ⑤ Route naar register

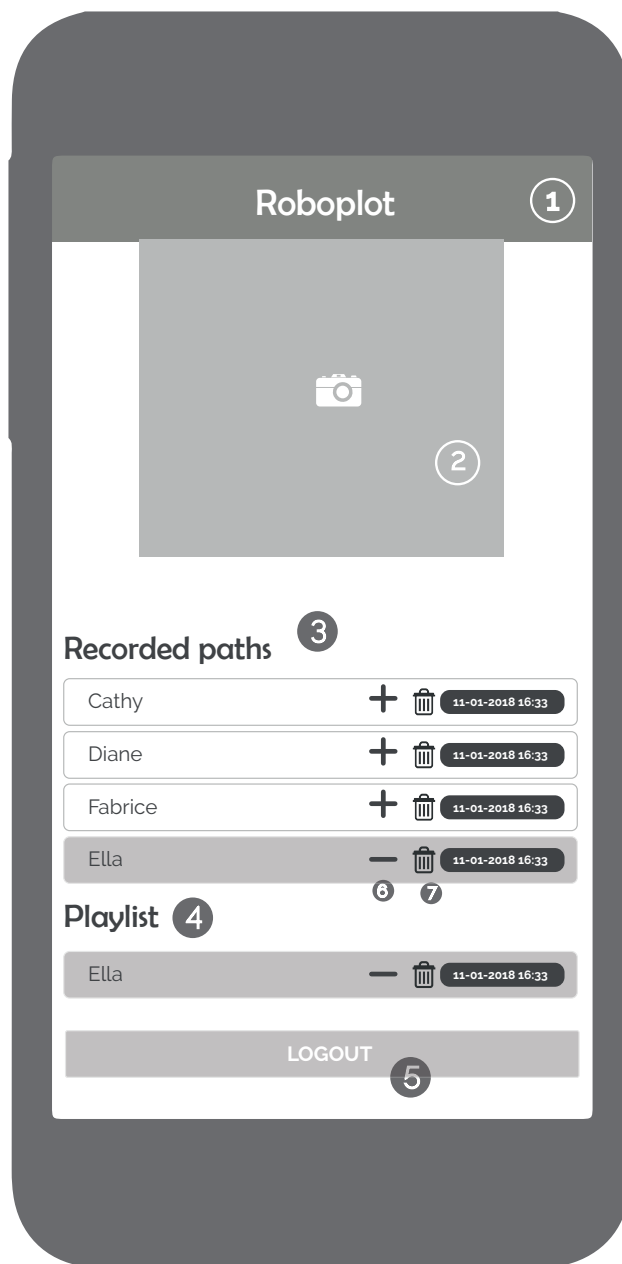




1.0 REGISTREREN

- ① header
- ② Logo
- ③ Opgenomen records
- ④ Playlist kan bestaan uit max 1 record
- ⑤ Logout
- ⑥ Toevoegen of verwijderen uit playlist
- ⑦ Permanent verwijderen van de record





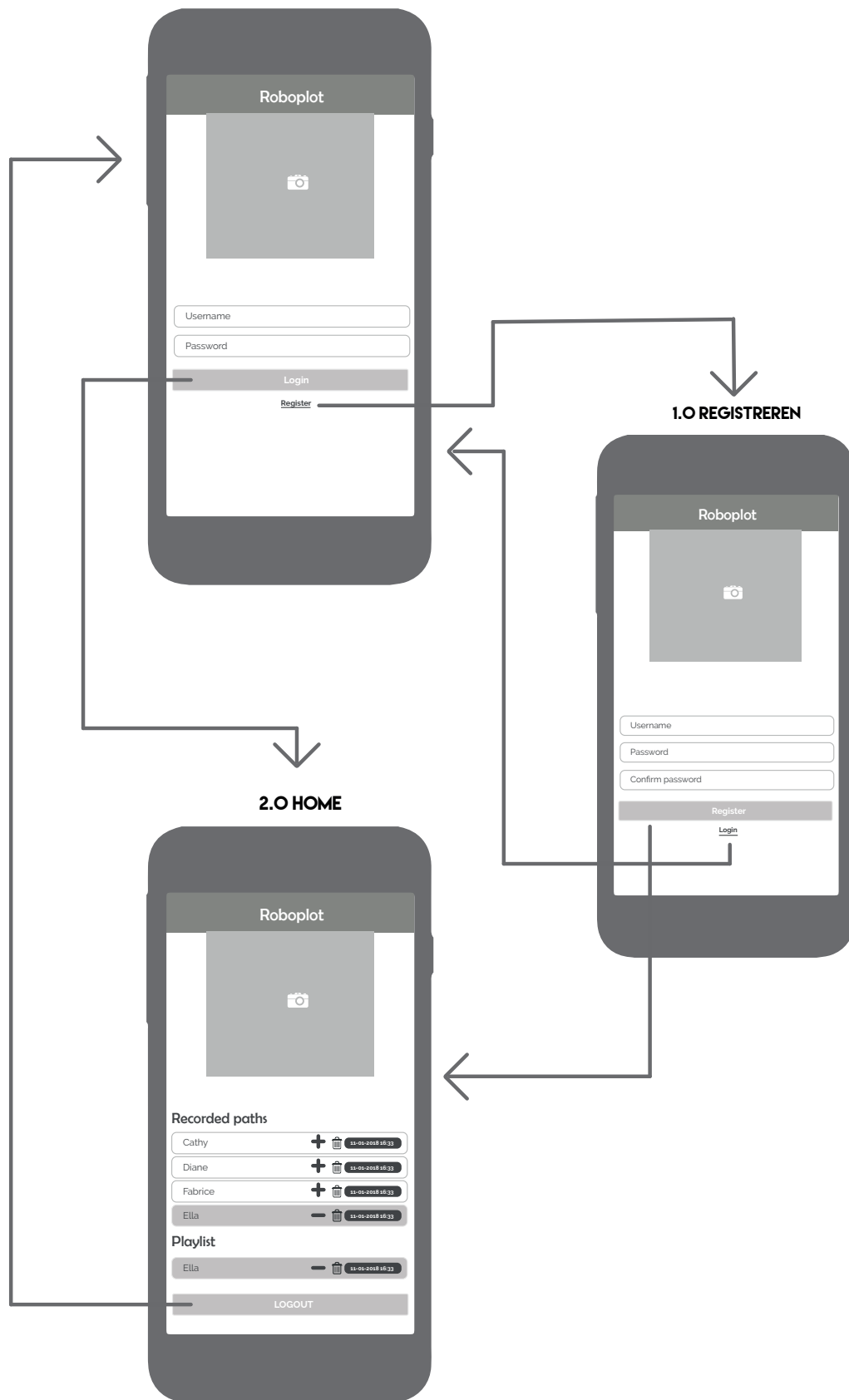
2.0 HOME

- ① header
- ② Logo
- ③ Opgenomen records
- ④ Playlist kan bestaan uit max 1 record
- ⑤ Logout
- ⑥ Toevoegen of verwijderen uit playlist
- ⑦ Permanent verwijderen van de record

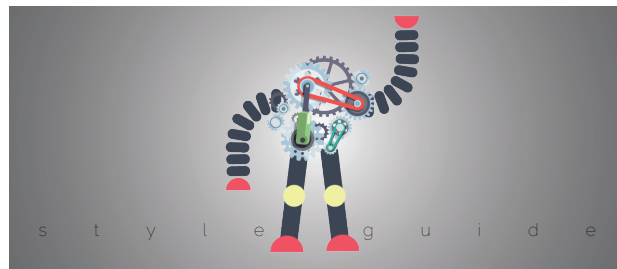


WIREFLOW MOBILE WEB APP

0.0 LOGIN



STYLE GUIDE

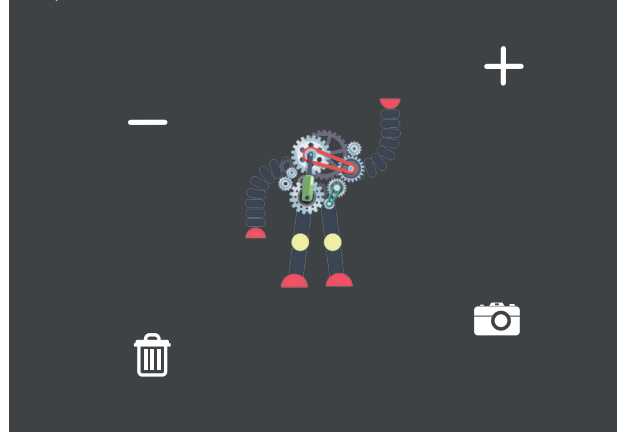


Colors



ICONS

<http://fontawesome.io/icons/>



Fonts

TITLE
Font-Family: Arial
Font-size: 24px
Color: #35495E

REGULAR
Font-Family: Arial
Font-size: 14px
Color: #35495E

Roboplot

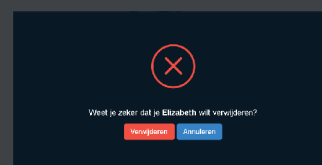
Roboplot

UI ELEMENTS

BUTTONS



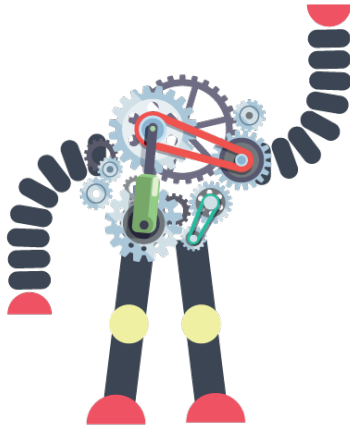
POPUP



VISUAL DESIGNS

Login

Roboplot

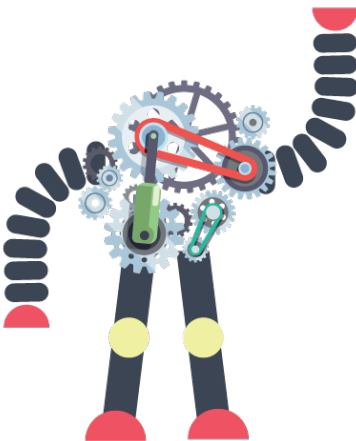


LOGIN

[Register](#)



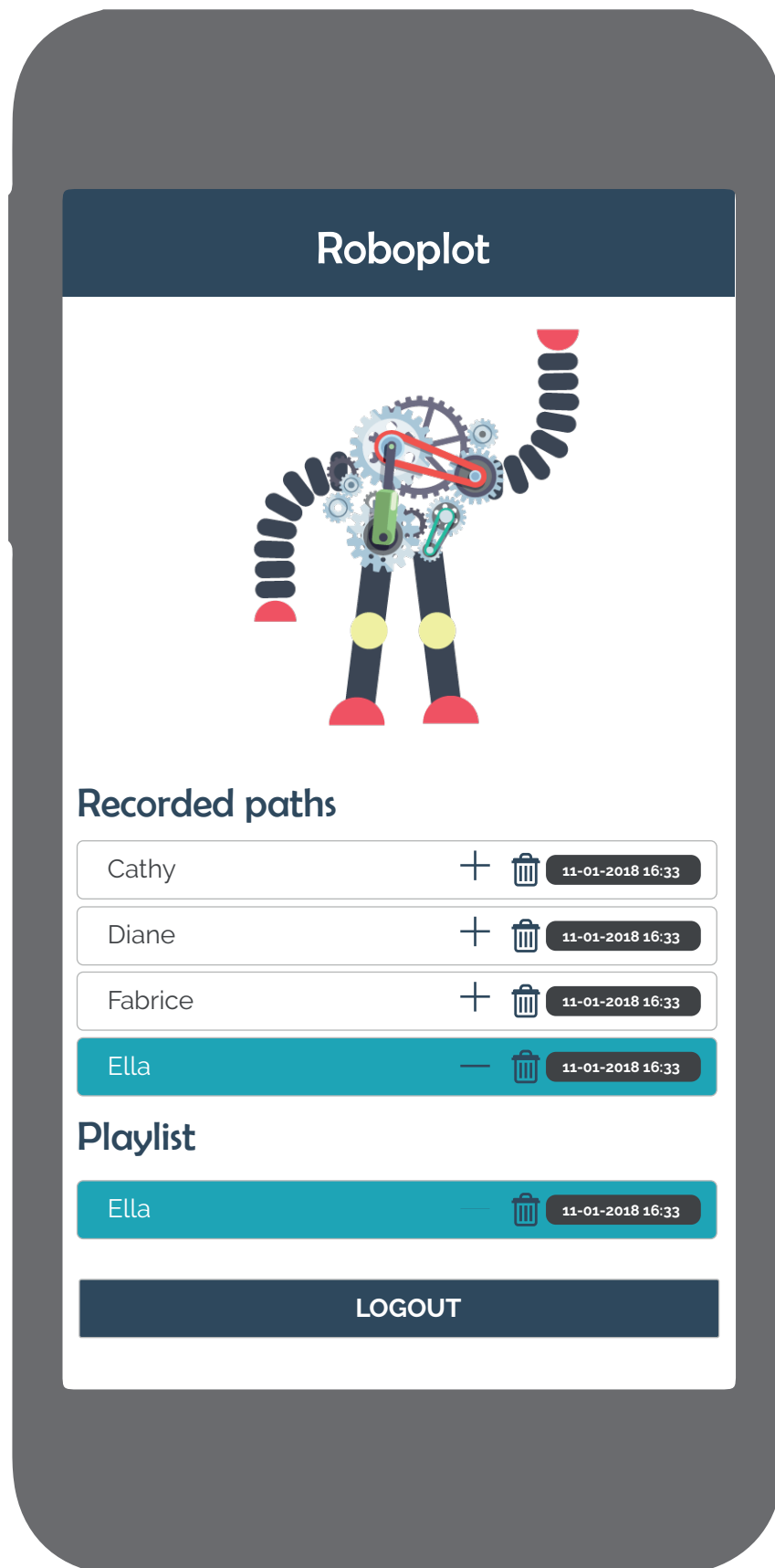
Roboplot



Register

[Login](#)





4. DEVELOPMENT

CODE SNIPPETS WEBAPP

- ***Login met firebase***

```
login: function(){
  firebase.auth().signInWithEmailAndPassword(this.email, this.password).then(
    function(){
      console.log('ingelogd')
    },
    function(err){
      console.log('niet gelukt message: ' + err.message)
    }
  )
},
```

- ***Registreatie met fierbase***

```
registreer: function(){
  firebase.auth().createUserWithEmailAndPassword(this.email, this.password).then(
    function(user){
      console.log('aangemaakt')
    },
    function(err){
      console.log('niet gelukt error message: ' + err.message)
    }
  )
  console.log('registreer')
},
```

- ***Modaltoggle***

```
toggle: function(record){
  this.error = null;
  if(this.modal == false){
    this.deleterec.key = record['.key'];
    this.deleterec.naam = record['naam'];
    this.$refs.modal.open();
    this.modal = true
  }else{
    this.deleterec.key = null;
    this.deleterec.naam = null;
    this.$refs.modal.close()
    this.modal = false
  }
},
```



- **Verwijderen**

```
permanent: function(){
  ref2.child(this.deleterec.key).remove()
  this.$refs.modal.close()
  this.modal = false
  this.deleterec.key = null;
  this.deleterec.naam = null;
},
```

- **Router check auth**

```
router.beforeEach((to, from, next) => {
  let currentUser = firebase.auth().currentUser;
  let requiresAuth = to.matched.some(record => record.meta.requiresAuth);

  if(requiresAuth && !currentUser) next('login')
  else if (!requiresAuth && currentUser) next('home')
  else next()
})
```

- **add to playlist**

```
toevoegen: function(record){
  let key = record['.key'];
  this.click = key;
  this.error = null;
  let naam = record['naam'];
  if(this.actief.length == 0){
    let datum = record['datum'];
    let waarden = record['waarden'];
    let rec = {
      'datum': datum,
      'naam': naam,
      'waarden': waarden,
      'key': key
    }
    ref.set(rec);
    ref2.child(key).update({ actief: true});
    this.vorigerecord = naam;
  }else{
    this.error = "Je kan maar 1 record tegelijk in de afspeellijst plaatsen."
  }
},
```



CODE SNIPPETS PYTHON

- **Libraries import**

```
# imports
#import dependencies
import math, time
import datetime
import RPi.GPIO as GPIO
from lib import PS3
import json
import io
import names
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
```

- **Variabelen**

```
#global variabelen
counter = 0
toggle = False
data = []
active = True
buttonTriangleDelay = 0
```

- **Motor pins**

```
#GPIO mode selecteren
GPIO.setmode(GPIO.BCM)

#motor1
Motor1A = 23 # Set GPIO-23 as Input 1 of the controller IC
Motor1B = 24 # Set GPIO-24 as Input 2 of the Controller IC
Motor1E = 25 # Set GPIO-25 as Enable pin 1 of the controller IC

#motor2
Motor2A = 22 # Set GPIO-23 as Input 1 of the controller IC
Motor2B = 27 # Set GPIO-24 as Input 2 of the Controller IC
Motor2E = 17 # Set GPIO-25 as Enable pin 1 of the controller IC

#motoren
GPIO.setup(Motor1A,GPIO.OUT)
GPIO.setup(Motor1B,GPIO.OUT)
GPIO.setup(Motor1E,GPIO.OUT)

GPIO.setup(Motor2A,GPIO.OUT)
GPIO.setup(Motor2B,GPIO.OUT)
GPIO.setup(Motor2E,GPIO.OUT)

pwmA=GPIO.PWM(25,100) #configuring Enable pin means GPIO-25 for PWM
pwmB=GPIO.PWM(17,100) #configuring Enable pin means GPIO-17 for PWM
```



- **Recorden van path**

```
#logs to array
if toggle==True:
    if counter%50 == 0:
        item = {"l": str(left),"r": str(right)}
        data.append(item)

#start met recorden
if (ps3.r1):
    toggle = True
    GPIO.output(lamp1, GPIO.HIGH)
    #print('Start opname')
#Stop met recorden
if (ps3.r2):
    if toggle==True:
        if data:
            #write
            GPIO.output(lamp1, GPIO.LOW)
            rec['waarden'] = data
            newrecord = write.push(rec)
            data = []
            rec['naam'] = names.get_first_name(gender='female')
        toggle = False
        #print('Stop opname')
    counter += 1
```

- **Motoren aansturen**

```
def setMotor(l, r):
    leftMotor = float(l)
    rightMotor = float(r)
    if (leftMotor == 0):
        pwmA.stop()
    elif (leftMotor > 0):
        pwmA.start(0)
        pwmA.ChangeDutyCycle(leftMotor)
        GPIO.output(Motor1A,GPIO.LOW)
        GPIO.output(Motor1B,GPIO.HIGH)
    elif (leftMotor < 0):
        pwmA.start(0)
        pwmA.ChangeDutyCycle(leftMotor * -1)
        GPIO.output(Motor1A,GPIO.HIGH)
        GPIO.output(Motor1B,GPIO.LOW)

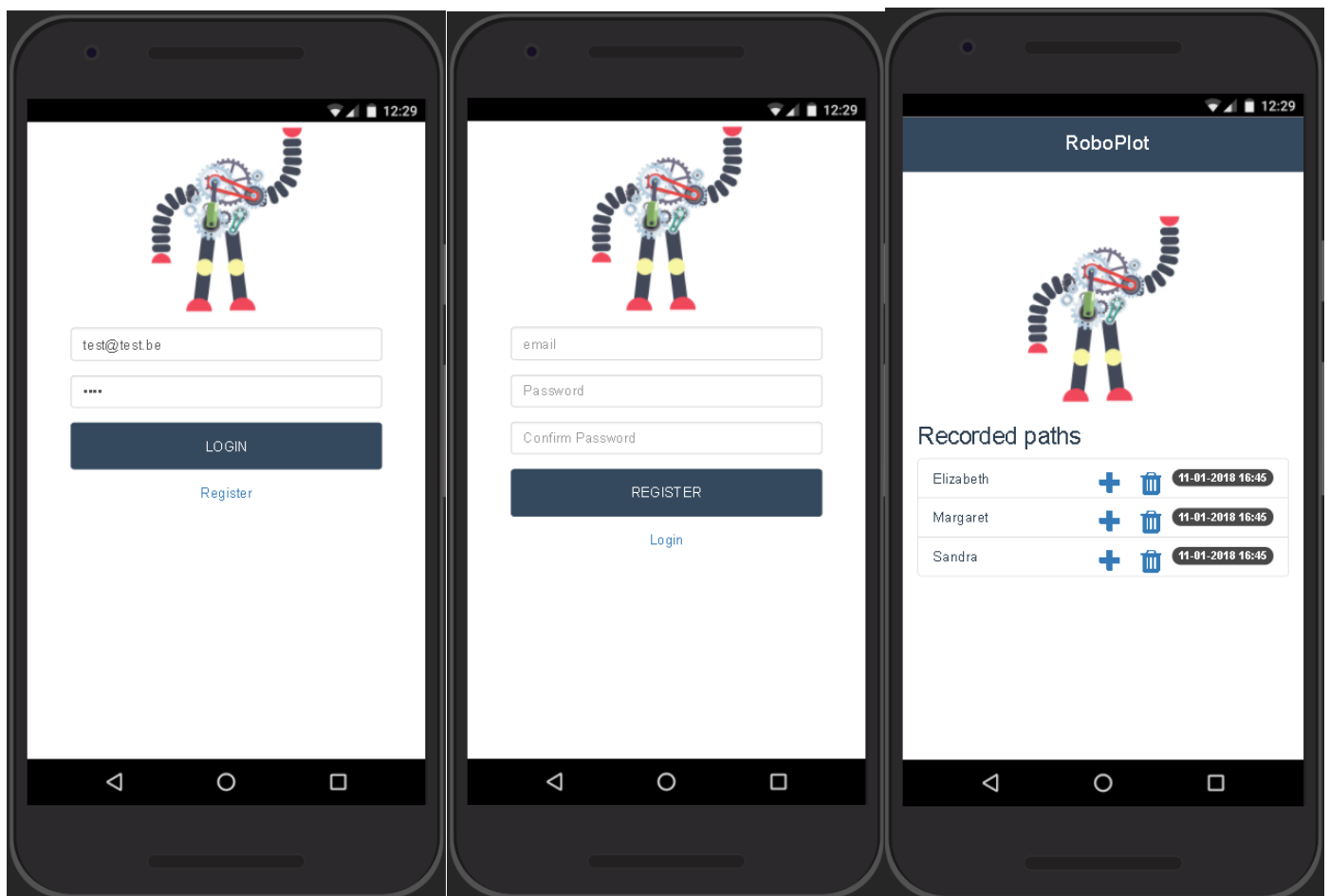
    if (rightMotor == 0):
        pwmB.stop()
    elif (rightMotor > 0):
        pwmB.start(0)
        pwmB.ChangeDutyCycle(rightMotor)
        GPIO.output(Motor2A,GPIO.LOW)
        GPIO.output(Motor2B,GPIO.HIGH)
    elif (rightMotor < 0):
        pwmB.start(0)
        pwmB.ChangeDutyCycle(rightMotor * -1)
        GPIO.output(Motor2A,GPIO.HIGH)
        GPIO.output(Motor2B,GPIO.LOW)
```



- *Afspelen record*

```
buttonTriangleDelay += 1
if (buttonTriangleDelay > 1500):
    if (ps3.a_triangle > 0):
        records_active = db.reference('actief').get()
        if records_active:
            array = records_active['waarden']
            for coord in array:
                fbLeft = float(coord['l'][0:7])
                fbRight = float(coord['r'][0:7])
                setMotor(fbLeft, fbRight)
                time.sleep(0.019)
        else:
            for x in range(0, 3):
                GPIO.output(lamp1, GPIO.HIGH)
                GPIO.output(buzzerE, GPIO.HIGH)
                time.sleep(0.1)
                GPIO.output(lamp1, GPIO.LOW)
                GPIO.output(buzzerE, GPIO.LOW)
                time.sleep(0.1)
            buttonTriangleDelay = 0
```

SCHERMAFBEELDINGEN



5. DELIVER

HANDLEIDING

- Sluit de raspberry pi aan op de powerbank
- Plug de gv batterij in de houder aan de onderkant van de robot
- Navigeer naar de map ./code/roboplot/src
- Open app.py "python3 app.py"



- Start recording



- Stop recording



- Playlist afspelen



- Linker motor aansturen



- Rechter motor aansturen



6. DEPLOY

DEPLOYMENT GUIDE WEBAPP

LOCAAL

"Yarn dev" of "Yarn start" staat gelijk aan node build/dev-server.js die ervoor zal zorgen dat er in de map van het project een server gestart zal worden op poort 8080

ONLINE

- yarn build
- ga in de "dist" folder van de app
- "heroku login"
- "git init"
- "heroku git:remote -a roboplot"
- git commit naar heroku master

DEPLOYMENT GUIDE PYTHON APP ROBOT

clone

- cd ~/Code/
- git clone <https://github.com/gdmgent-1718-wot/roboplot.git>

Install required packages

- sudo apt-get update
- sudo apt-get -y install libusb-dev joystick python-pygame

Install Sixpair

- cd ~
- wget <http://www.pabr.org/sixlinux/sixpair.c>
- gcc -o sixpair sixpair.c -lusb



Reboot your PI

- `sudo reboot`

Connect PS3 controller

- Connect your controller via USB to the PI

Getting the hardware address from your controller

- `$ sudo ~/sixpair`

Disconnect your controller and try connecting via bluetooth. If the controller asks for a pin. Just leave the field empty. (Try this a couple of times if it's not working)

Test the controller

- `jstest /dev/input/js0`

This script delivers all value's of the buttons. You will see changes when you press a button

Execute

- `cd roboplot/src`
- `python3 app.py`



7. EXTRA INFO

WEBAPP

<https://www.roboplot.herokuapp.com>

PROMOVIDEO

<https://www.youtube.com/watch?v=e1qKE4O-l1A&feature=youtu.be>

PRESENTATIE

<https://github.com/gdmgent-1718-wot/roboplot/blob/master/docs/presentatie.pdf>



8. FOTO'S

