

## Opdrachtnemers

Naam, Studentnummer, Klas	<b>Mike ten Klooster, 226751, DEV4</b>
Cohort	<b>2016-2017, DEV4</b>
Crebonummer	<b>25187</b>
Opleiding naam	<b>Applicatie- en mediaontwikkelaar</b>
Onderwijsinstelling	<b>Friesland College, Heerenveen</b>

## Document

Datum publicatie	<b>15-09-2020</b>
Documentversie	<b>1.1</b>

# 1. INHOUDSOPGAVE

1.	Inhoudsopgave.....	2
2.	Wat is mad voor mij? .....	3
3.	wat kan je met MAD?.....	3
4.	Welke Applicaties heb ik gemaakt .....	4
4.1	Flashlight .....	4
4.2	Clicker Game .....	5
4.3	Calculator .....	6
4.4	Noteblock .....	7
5.	Welke methodieken heb ik geleerd .....	8
5.1	Java.....	8
5.2	XML .....	8
6.	Project specificaties .....	9
6.1	Inleiding.....	9
6.2	Samenvatting .....	9
6.3	Link .....	9
6.4	Software .....	10
6.5	Eisen .....	10
7.	Schetsen .....	11
8.	Gebruikte documenten en Bronnen .....	12

MAD staat voor, Mobile Application Development.

## 2. WAT IS MAD VOOR MIJ?

Mobile Application Development is voor de apps die je op je telefoon hebt. Het maakt niet uit wat voor een smartphone je hebt aangezien voor alle smartphones apps beschikbaar zijn. De apps kunnen van alles, ze kunnen op zichzelf staan of gebruik maken van widgets van de telefoon zelf.

## 3. WAT KAN JE MET MAD?

Voordat ik met MAD begon kende ik geen Java en wist ik niks af van Mobiele applicaties.

Met MAD kan je veel dingen, kijk bijvoorbeeld naar Pokemon Go, naar Google Maps. En is dus veel te breed om iets te maken zonder onderzoek of duidelijk plan.

Dus ik besloot om me te verdiepen op basis van zo veel mogelijk verschillende projecten, ik heb gekeken wat mij het leukst leek om te doen. En op basis daarvan besloten welke projecten ik ga doen.

## 4. WELKE APPLICATIONS HEB IK GEMAAKT

Uit onderzoek bleek dat de ene opdracht die ik wou maken makkelijker bleek dan eerder gedacht en anderen moeilijker. De makkelijker was de Flashlight, en de moeilijker was de Noteblock.

### 4.1 Flashlight

Ik besloot te beginnen met het maken van de Flashlight aangezien het mij moeilijker leek om bij de camera te komen, maar verrassend genoeg was dat best wel makkelijk. Maar in retrospectief is dat best wel schokkend, aangezien je toestemming moet geven voor de camera (bij nieuwere software maakt het niet meer uit want dan kan je direct de zaklamp benaderen).

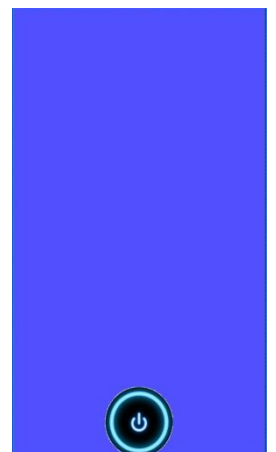
#### OnCreate

#### Preview

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // flash switch button (on / off)
    btn = (ImageButton) findViewById(R.id.btn);
    /*
     * First check if device is supporting flashlight or not
     */
    Flash = getApplicationContext().getPackageManager()
        .hasSystemFeature(PackageManager.FEATURE_CAMERA_FLASH);

    if (!Flash) {
        // if the device doesn't support flash
        // Show alert message and close the application
        AlertDialog alert = new AlertDialog.Builder(context: MainActivity.this)
            .create();
        alert.setTitle("Error");
        alert.setMessage("Sorry, your device doesn't support flash light!");
        alert.setButton(text: "OK", (OnClickListener) (dialog, which) -> {
            // closing the application
            finish();
        });
        alert.show();
        return;
    }
    // get the camera
    getCamera();

    // displaying button image
    toggleButtonImage();
    /*
     * Switch button click event to toggle flash on/off
     */
    btn.setOnClickListener((v) -> {
        if (FlashOn) {
            // turn off flash
            turnOffFlash();
        } else {
            // turn on flash
            turnOnFlash();
        }
    });
}
//endof onCreate()
```



## 4.2 Clicker Game

Na de Flashlight ging ik bezig met een clicker game, de reden waarom ik deze heb gekozen is omdat ik wou spelen met de storage(Shared preference), en hoe dat werkte als je je scherm zou draaien. Het vergeet het aangezien het eigenlijk de app opnieuw opstart, dus om daar om heen te gaan is een force save maken wanneer je één op de knop klikt die +1 doet, of wanneer je de app op afsluit / op pauze zet.

### OnCreate & OnStop

### Preview

```
package com.example.clickerapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    TextView tvCounter;
    TextView plusCounter;
    Button btnIncrement;
    Button btnPlusClick;
    int counter = 0;
    int plusClick = 1;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // de saved data van de counter wordt opgevangen en terug op counter gezet (als er geen gesavede data is = counter 0)
    counter = Integer.valueOf(this.getSharedPreferences( name: "highScore", Context.MODE_PRIVATE).getString( key: "highScore", defValue: "0"));

    plusCounter = findViewById(R.id.plusCounter);
    tvCounter = findViewById(R.id.tvCounter);
    btnIncrement = findViewById(R.id.btnIncrement);
    btnPlusClick = findViewById(R.id.btnPlusClick);

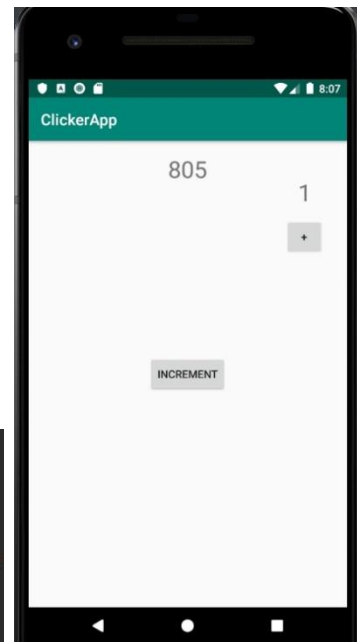
    //set de counter om naar text zodat het gedisplayed kan worden
    tvCounter.setText(String.valueOf(counter));
    plusCounter.setText(String.valueOf(plusClick));

    btnPlusClick.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            plusClick = plusClick + 1;
            plusCounter.setText(String.valueOf(plusClick));
        }
    });

    //bij elke click op de button wordt dit uitgevoerd
    btnIncrement.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            counter = counter + plusClick;
            tvCounter.setText(String.valueOf(counter));
        }
    });

    //hier wordt de counter opgeslagen als "highScore"
    protected void onStop(){
        this.getSharedPreferences( name: "highScore", Context.MODE_PRIVATE).edit().putString("highScore", String.valueOf(counter)).apply();

        super.onStop();
    }
}
```



## 4.3 Calculator

Als derde ging ik bezig met het maken van een Rekenmachine, maar waarom zou je een rekenmachine maken? Nou ik wou kijken hoe Java het oppakt om met values achter knoppen te werken, ik heb deze begin jaar 3 gemaakt dus de code is slecht en outdated er op terugkijkend.

### App

### Preview

```
package com.example.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    //all Buttons are getting set
    protected Button button0, button1, button2, button3, button4, button5, button6,
        button7, button8, button9, buttonAdd, buttonSub, buttonDivision,
        buttonMul, button10, buttonC, buttonEqual;
    //the EditText is getting set
    public EditText EditText;
    //these floats are for the first and second value entered
    public float mValueOne, mValueTwo;
    //booleans for checking which one is active
    public boolean Addition, mSubtract, Multiplication, Division;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Retrieves GlobalVars
        this.setGlobalVars();
        // Retrieves EventListeners
        this.setEventListeners();
    }
}
```

```
button5.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + "5");
});

button6.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + "6");
});

button7.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + "7");
});

button8.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + "8");
});

button9.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + "9");
});

button0.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + "0");
});

//when the button is clicked Add is set to true
buttonAdd.setOnClickListener((v) -> {
    mValueOne = Float.parseFloat(EditText.getText() + "");
    Addition = true;
    EditText.setText(null);
});

//when the button is clicked Subtract is set to true
buttonSub.setOnClickListener((v) -> {
    mValueOne = Float.parseFloat(EditText.getText() + "");
    mSubtract = true;
    EditText.setText(null);
});

//when the button is clicked Multiplication is set to true
buttonMul.setOnClickListener((v) -> {
    mValueOne = Float.parseFloat(EditText.getText() + "");
    Multiplication = true;
    EditText.setText(null);
});
```

```
private void setGlobalVars()
{
    //all fields findViewById(initializing)
    button0 = (Button) findViewById(R.id.button0);
    button1 = (Button) findViewById(R.id.button1);
    button2 = (Button) findViewById(R.id.button2);
    button3 = (Button) findViewById(R.id.button3);
    button4 = (Button) findViewById(R.id.button4);
    button5 = (Button) findViewById(R.id.button5);
    button6 = (Button) findViewById(R.id.button6);
    button7 = (Button) findViewById(R.id.button7);
    button8 = (Button) findViewById(R.id.button8);
    button9 = (Button) findViewById(R.id.button9);
    button10 = (Button) findViewById(R.id.button10);
    buttonAdd = (Button) findViewById(R.id.buttonadd);
    buttonSub = (Button) findViewById(R.id.buttonsub);
    buttonMul = (Button) findViewById(R.id.buttonmul);
    buttonDivision = (Button) findViewById(R.id.buttondiv);
    buttonC = (Button) findViewById(R.id.buttonC);
    buttonEqual = (Button) findViewById(R.id.buttoneq1);
    EditText = (EditText) findViewById(R.id.edt1);
}

public void setEventListeners()
{
    //if a button with this id is clicked, set EditText.
    button1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText.setText(EditText.getText() + "1");
        }
    });

    button2.setOnClickListener((v) -> {
        EditText.setText(EditText.getText() + "2");
    });

    button3.setOnClickListener((v) -> {
        EditText.setText(EditText.getText() + "3");
    });

    button4.setOnClickListener((v) -> {
        EditText.setText(EditText.getText() + "4");
    });
}
```

```
//when the button is clicked Division is set to true
buttonDivision.setOnClickListener((v) -> {
    mValueOne = Float.parseFloat(EditText.getText() + "");
    Division = true;
    EditText.setText(null);
});

//when the = button is pressed check which one of the if statements is true.
buttonEqual.setOnClickListener((v) -> {
    mValueTwo = Float.parseFloat(EditText.getText() + "");

    if (Addition) {
        EditText.setText(mValueOne + mValueTwo + "");
        Addition = false;
    } else if (mSubtract) {
        EditText.setText(mValueOne - mValueTwo + "");
        mSubtract = false;
    } else if (Multiplication) {
        EditText.setText(mValueOne * mValueTwo + "");
        Multiplication = false;
    } else if (Division) {
        EditText.setText(mValueOne / mValueTwo + "");
        Division = false;
    }
});

//when the button is clicked make the EditText empty
buttonC.setOnClickListener((v) -> { EditText.setText(""); });

//when the button is clicked, add a . (dot)
button10.setOnClickListener((v) -> {
    EditText.setText(EditText.getText() + ".");
});
```



## 4.4 Noteblock

Als vierde ging ik aan de slag met het maken van een Noteblock zodat ik er boodschappenlijstjes in kon zetten en makkelijk kon bijhouden. Hierin maak ik ook gebruik van de storage (shared preference & database).

### OnCreate

```
package com.example.noteblock;

import androidx.appcompat.app.AppCompatActivity;

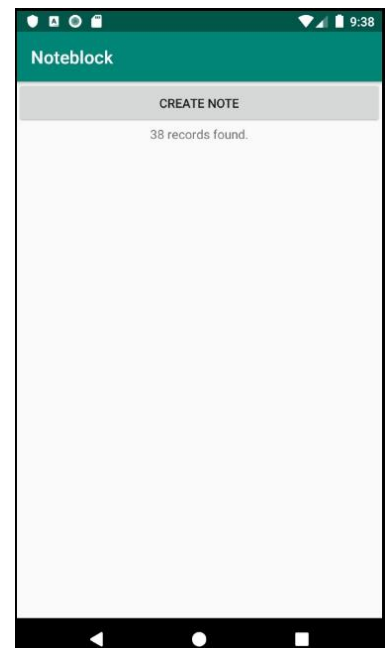
import android.content.Context;
import android.os.Bundle;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnAddNote = findViewById(R.id.btnAddNote);
        btnAddNote.setOnClickListener(new OnClickListenerCreateNote());
        countRecords();
        // readRecords();
    }

    public void countRecords() {
        int recordCount = new TableControllerNote( context: this).count();
        TextView textViewRecordCount = (TextView) findViewById(R.id.textViewRecordCount);
        textViewRecordCount.setText(recordCount + " records found.");
    }
}
```

### Preview



```
class OnClickListenerCreateNote extends MainActivity implements View.OnClickListener {

    // extends MainActivity
    @Override
    public void onClick(View view) {
        final Context context = view.getRootView().getContext();
        // final Context context = this;

        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        final View formElementsView = inflater.inflate(R.layout.note_input_form, root null, attachToRoot false);
        final EditText editNote = formElementsView.findViewById(R.id.editNote);
        new AlertDialog.Builder(context)
            .setView(formElementsView)
            .setTitle("Create Note")
            .setPositiveButton( Text "Add",
                (dialog, id) -> {
                    dialog.dismiss();
                    Context context = getApplicationContext();
                    String userNote = editNote.getText().toString();
                    ObjectNote objectNote = new ObjectNote();
                    objectNote.note= userNote;

                    boolean createSuccessful = new TableControllerNote(context).create(objectNote);
                    if(createSuccessful){
                        Toast.makeText(context, Text "Note information was saved.", Toast.LENGTH_SHORT).show();
                    }else{
                        Toast.makeText(context, Text "Unable to save note information.", Toast.LENGTH_SHORT).show();
                    }
                    MainActivity.countRecords();
                    readRecords();
                    finish();
                    Intent intent = new Intent(context,MainActivity.class);
                    context.startActivity(intent);
                    // ((MainActivity) context).readRecords();
                })
            .setNegativeButton( Text "Cancel", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                    // Do nothing
                }
            })
            .show();
    }
}
```

## 5. WELKE METHODIEKEN HEB IK GELEERD

### 5.1 Java

Voor Java heb ik veel verschillende dingen opgedaan, hoe ik dingen binnen de telefoon kan benaderen, databases binnen mobile en shared preferences.

### 5.2 XML

Binnen XML heb ik verschillende type layouts geleerd, zoals linear, relative en constraint. En de type tags die er bij nodig zijn.



## 6. PROJECT SPECIFICATIES

### 6.1 Inleiding

Dit project is mijn eindproject voor het keuzedeel Mobile application development.

De doelgroep is alle leeftijdsklassen

### 6.2 Samenvatting

Het project is het maken van een app waarmee je op basis van vakjes aanklikken met een kleuren pallet je een afbeelding maakt die je kan opslaan op je telefoon.

Zo kan je op je telefoon alsnog je creatieve ingevingen kwijt.

### 6.3 Link

Github

<https://github.com/Miketenklooster/MAD>

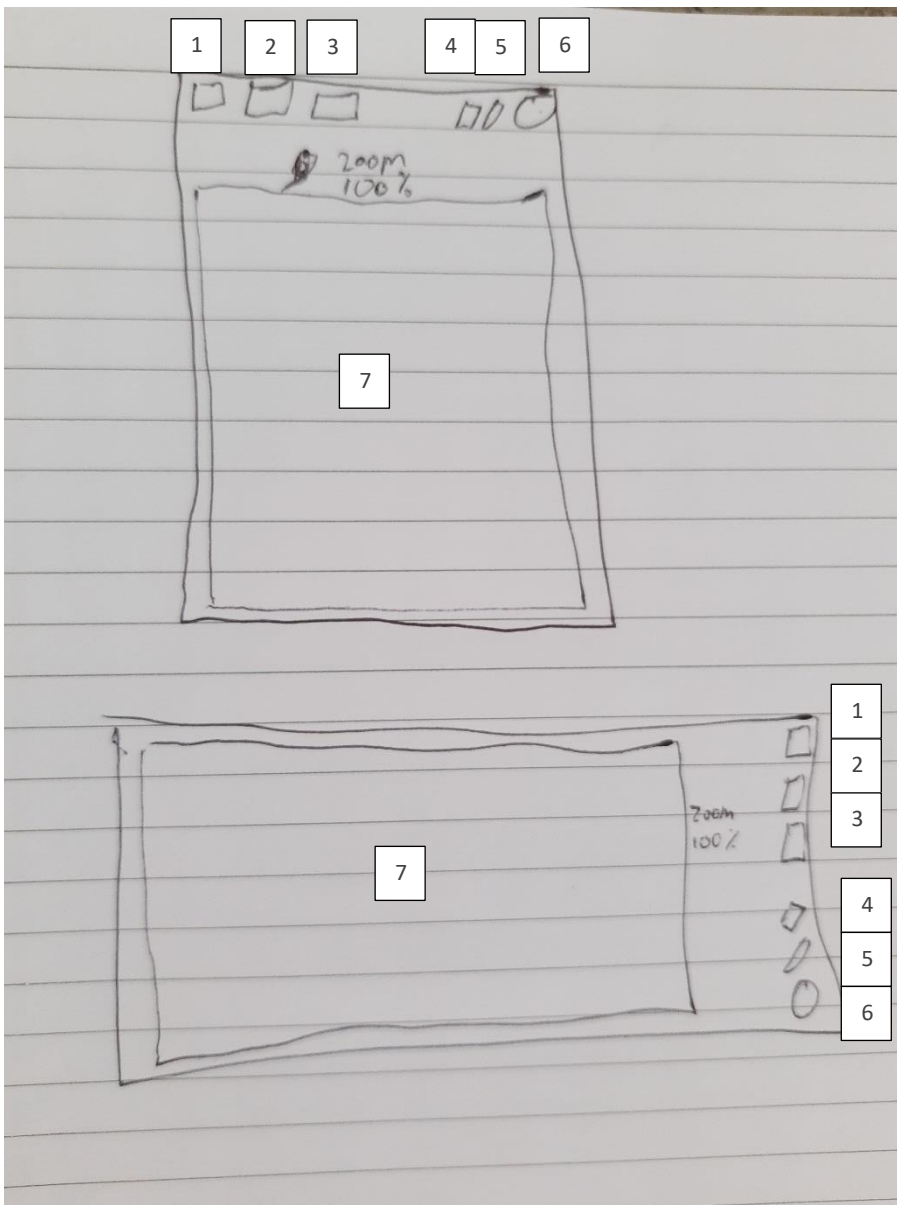
## 6.4 Software

Microsoft Teams	Chat applicatie die we gebruiken voor onderlinge communicatie.
Github	Voor versie beheer, kan je iedereen zijn input bijhouden.
Java 13	Java versie die ik ga gebruiken.
Android Studio	Code editor die ik ga gebruiken.
Microsoft Word	Voor het maken van documentatie.

## 6.5 Eisen

1. Mobile / Tablet
2. Moet meerdere projecten kunnen hebben.
3. Kunnen opslaan.
4. Groote van het veld aanpassen.
5. Verschillende kleuren kiezen op basis van een color picker.
6. Een color met de eyedropper tool kunnen pakken.
7. Keizen of je de afbeelding opslaat met de grid lijnen erbij of met een transparante achtergrond.

## 7. SCHETSEN



### Page

1. Save
2. Save as
3. Reset
4. Gum
5. Eyedrop
6. Color picker
7. Grid

## 8. GEBRUIKTE DOCUMENTEN EN BRONNEN

Google translate: <https://www.google.com/search?client=firefox-b-d&q=google+translate>

Stackoverflow: <https://stackoverflow.com/>

Android studio: <https://developer.android.com/docs>

Feedback

- Romeo Beun