

|| **PORTFOLIO MAD**

Opdrachtnemers

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

Document

Datum publicatie	19-09-2020
Documentversie	1.2

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 moeilijke om bij de camera te komen, maar verassend 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).

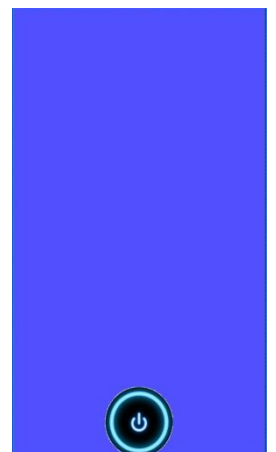
OnCreate

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, ook heb ik deze in 2 verschillende IDE's gemaakt, android studio en visual studio 2019.

OnCreate & OnStop

Preview

```
namespace clickergame
{
    [Activity(Label = "@string/app_name", Theme = "@style/AppTheme.NoActionBar", MainLauncher = true)]
    public class MainActivity : AppCompatActivity
    {
        int counter = 0;
        int plusclick = 1;
        TextView score;
        TextView multiplier;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            Xamarin.Essentials.Platform.Init(this, savedInstanceState);
            SetContentView(Resource.Layout.activity_main);

            counter = Preferences.Get("highscore", 0);
            score = FindViewById<TextView>(Resource.Id.score);
            score.Text = counter.ToString();
            multiplier = FindViewById<TextView>(Resource.Id.multiplier);
            Button btn = FindViewById<Button>(Resource.Id.add);
            Button increment = FindViewById<Button>(Resource.Id.increment);

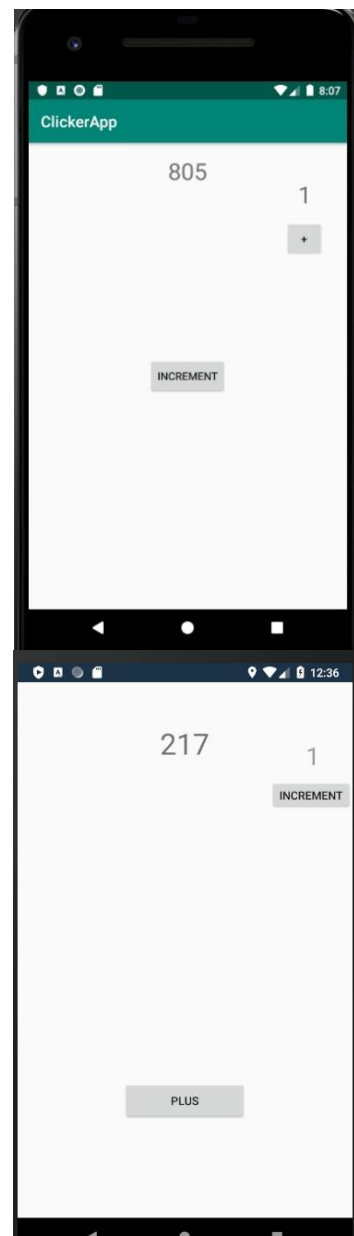
            increment.Click += Increment_Click;
            btn.Click += Btn_Click;
        }

        protected override void OnStop()
        {
            base.OnStop();
            Preferences.Set("highscore", counter);
        }

        protected override void OnPause()
        {
            base.OnPause();
            Preferences.Set("highscore", counter);
        }

        private void Increment_Click(object sender, EventArgs e)
        {
            plusclick = plusclick + 1;
            multiplier.Text = plusclick.ToString();
        }

        private void Btn_Click(object sender, EventArgs e)
        {
            counter = counter + plusclick;
            score.Text = counter.ToString();
        }
    }
}
```



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, en heb zelf geprobeerd het zo klein mogelijk te houden.

App

```
public class MainActivity extends AppCompatActivity {
    //the EditText is getting set
    public EditText EditText;
    //these floats are for the first and second value entered
    public float mValueOne, mValueTwo, Result;
    //booleans for checking which one is active
    public boolean Addition, mSubtract, Multiplication, Division, Values, result;

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

    private void setGlobalVars() { EditText = findViewById(R.id.edt1); }

    public void setNumbers(View target)
    {
        Button but = (Button)target;
        String value = but.getText().toString();
        if(!value.equals("C")) {
            EditText.setText(EditText.getText() + value);
        }else{
            EditText.setText(null);
            mValueOne = 0.0f;
            mValueTwo = 0.0f;
            Result = 0.0f;
        }
    }

    public void calc(View target)
    {
        Button but = (Button)target;
        String value = but.getText().toString();
        if(!value.equals("=")){
            Values = mValueOne != 0.0f && !result;
            switch (value)
            {
                case "+":
                    Addition = true;
                    if(Values){mValueOne = mValueOne + Float.parseFloat(EditText.getText().toString());}
                    break;

                case "-":
                    mSubtract = true;
                    if(Values){mValueOne = mValueOne - Float.parseFloat(EditText.getText().toString());}
                    break;

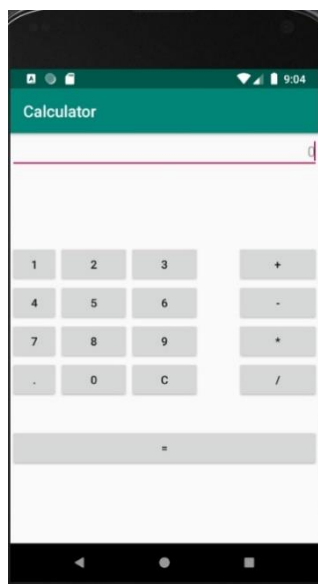
                case "*":
                    Multiplication = true;
                    if(Values){mValueOne = mValueOne * Float.parseFloat(EditText.getText().toString());}
                    break;

                case "/":
                    Division = true;
                    if(Values){mValueOne = mValueOne / Float.parseFloat(EditText.getText().toString());}
                    break;
            }
        }
        if(!Values) {
            mValueOne = Float.parseFloat(EditText.getText().toString());
        }
        EditText.setText(null);
        result = false;
    }

    else {
        mValueTwo = Float.parseFloat(EditText.getText().toString());

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

Preview



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;

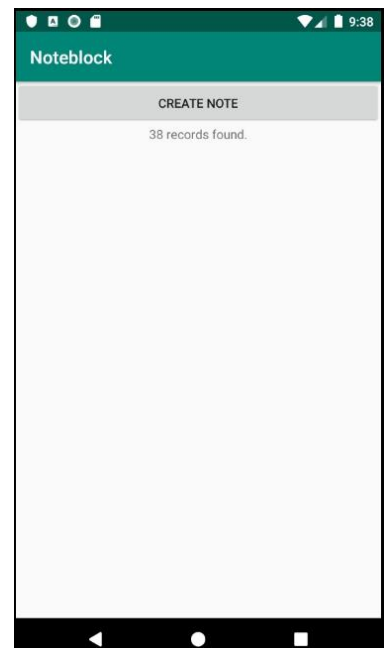
import android.content.Context;
import android.os.Bundle;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import java.util.List;

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) {
                    //
                }
            })
            .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>

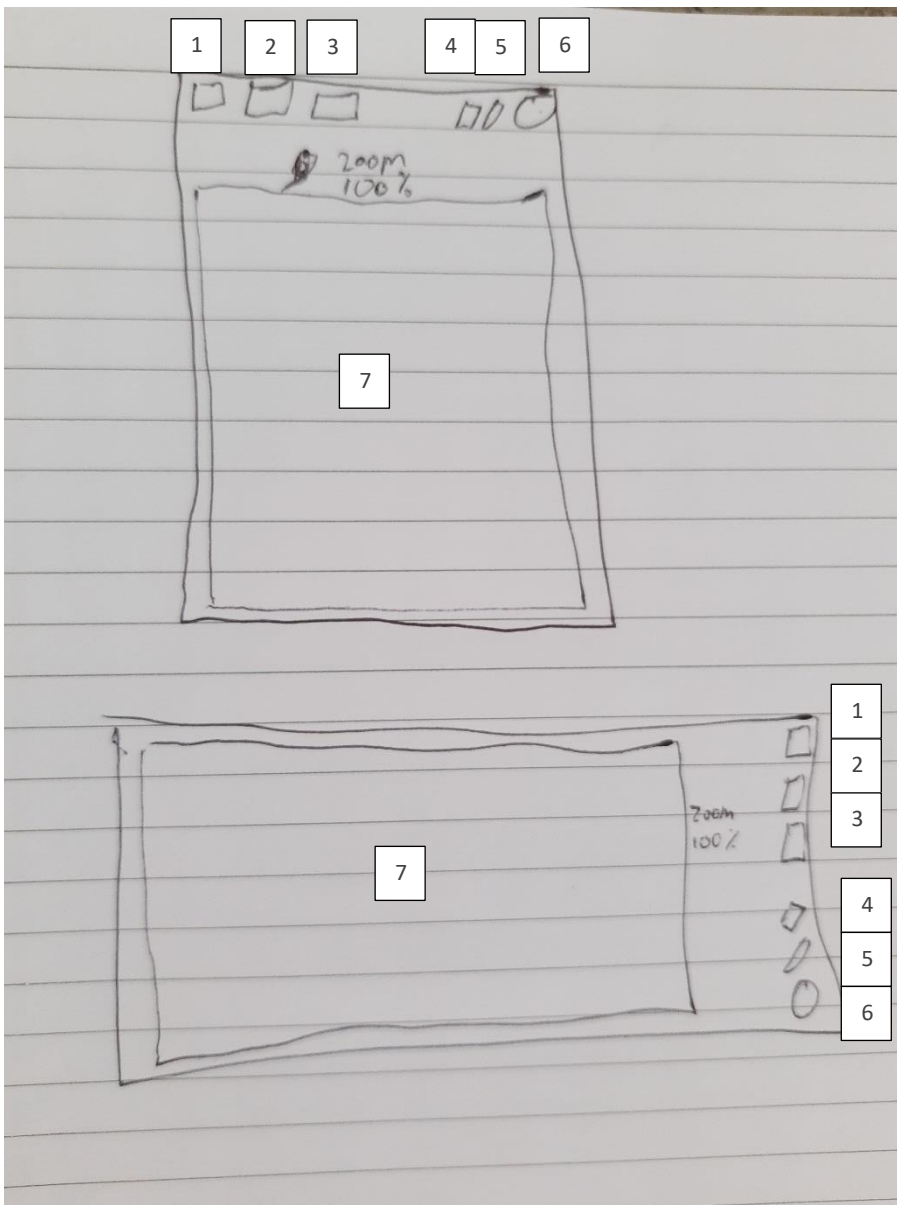
6.4 Software

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