**Cordova-project 2APPAI – 2WT - 3IoT**

**Brief description of the app**

**“about me”-pagina**

Dit is de startpagina van de applicatie. Hier wordt uitleg over mezelf gegeven en het doel van de applicatie. Hier is geen interactie tussen de applicatie en de gebruiker.

**Ledenlijst huidige zondag**

Op deze pagina worden de leden van de gekozen afdeling getoond. Hier kan men met behulp van de checkboxes aanduiden welke leden aanwezig zijn en welke afwezig. Helemaal onderaan de pagina staat ook een teller die aantoont hoeveel leden er aanwezig zijn.

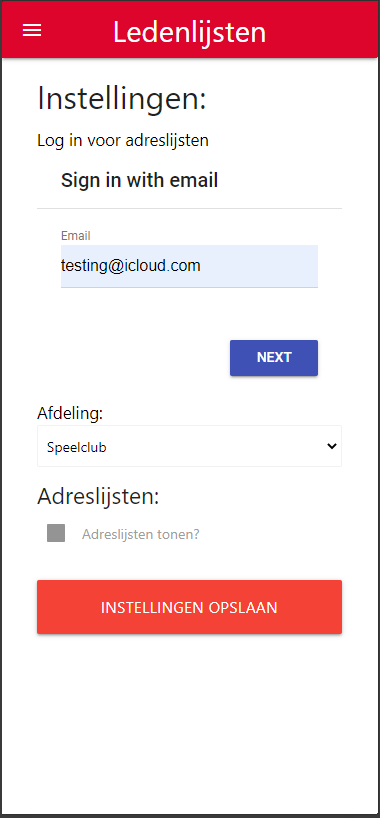
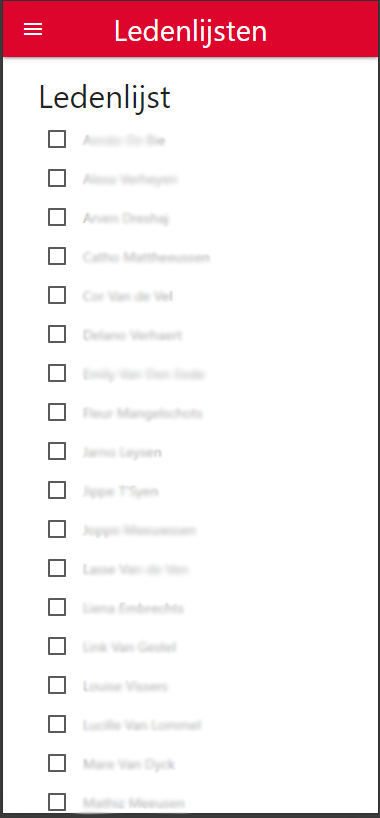
**Adreslijst leden**

Deze pagina is enkel gevuld wanneer een gebruiker ingelogd is via Firebase. Deze tab toont alle leden van de geselecteerde afdeling inclusief de bijhorende adressen.

**Instellingen**

Hier worden alle instellingen voor de applicatie getoond. Men kan kiezen van welke afdeling ze de lijsten willen bekijken, alsook of ze de adreslijsten willen opvragen per lid.

Hier kan men zich ook inloggen als de gebruiker is aangeduid als administratief verantwoordelijke van Chiro Vorselaar. Dit is per afdeling 1 leiding en deze logingegevens worden door mij en de volwassen begeleiding van Chiro Vorselaar aangeboden. Indien men zich inlogt als administratief verantwoordelijke dan krijgt men toegang tot extra informatie per lid, zoals bijvoorbeeld de adressen.

****

**Used Cordova plug-ins**

- cordova-plugin-whitelist

De whitelist plugin zorgt ervoor dat een gebruiker gelimiteerd wordt in welke URLs ze kunnen bereiken binnen een webview.

- cordova-plugin-androidx

- cordova-plugin-androidx-adapter

Deze plugins worden gebruikt om libraries in een Cordova/Phonegap project te enablen indien deze androidx gebruiken.

- cordova-plugin-browsertab

- cordova-plugin-inappbrowser

- cordova-universal-links-plugin

Deze plugins dienen voornamelijk voor het bouwen van de login voor Firebase die ingebouwd is. Het login element is een webview die door deze plugins ingeladen kan worden.

- cordova-plugin-buildinfo

De buildinfo plugin zorgt er voor dat allerlei data over het device beschikbaar zijn vanaf dat de deviceready event wordt afgevuurd wordt.

- cordova-plugin-compat

De compat plugin zorgt ervoor dat backwards compatibility met vroegere versies van Cordova behouden wordt.

- cordova-plugin-firestore

De firestore plugin is de plugin die het meeste gebruikt wordt in de applicatie. Het zorgt voor een realtime verbinding tussen de app en de cloud waar de data van de applicatie wordt bijgehouden.

- cordova-plugin-vibration

Deze simpele plugin zorgt voor een vibratie die x aantal seconden kan duren of met een patroon.

**Infopage about the app**

[Webruimte • SIN (sinners.be)](https://ledenlijstchirovorselaar.sinners.be/)

**Source code**

## App.js

//imports all the needed data

// import MyImage from '../../resources/me.jpg';

import {

    initializeApp

} from 'firebase/app';

import {

    getDatabase,

    ref,

    child,

    get

} from "firebase/database";

import {

    getAuth,

    signInWithEmailAndPassword,

    onAuthStateChanged,

    signInWithRedirect,

    getRedirectResult,

    GoogleAuthProvider

} from "firebase/auth";

import firebase from 'firebase/compat/app';

import \* as firebaseui from 'firebaseui';

import 'firebaseui/dist/firebaseui.css';

//all data needed for connecting to the database

const firebaseConfig = {

    apiKey: "AIzaSyD-NuJva4GTL\_2\_w81wdwhJDEgXhLWVrko",

    authDomain: "ledenlijst-chiro.firebaseapp.com",

    databaseURL: "https://ledenlijst-chiro-default-rtdb.europe-west1.firebasedatabase.app",

    projectId: "ledenlijst-chiro",

    storageBucket: "ledenlijst-chiro.appspot.com",

    messagingSenderId: "804184704488",

    appId: "1:804184704488:web:7248f6d785581317fe905d"

};

let ledenChecker = 0;

firebase.initializeApp(firebaseConfig);

const provider = new GoogleAuthProvider();

var ui = new firebaseui.auth.AuthUI(firebase.auth());

const auth = getAuth();

$(function () {

    //firebase login ui setup

    ui.start('#firebaseui-auth-container', {

        signInOptions: [

            firebase.auth.EmailAuthProvider.PROVIDER\_ID

        ],

        // Type other config options here...

    });

    var uiConfig = {

        callbacks: {

            signInSuccessWithAuthResult: function (authResult, redirectUrl) {

                // User successfully signed in.

                // Return type determines whether we continue the redirect automatically

                // or whether we leave that to developer to handle.

                $("#adrescheckbox").removeAttr("disabled");

                return false;

            },

            uiShown: function () {

                // The widget is rendered.

                // Hide the loader.

                document.getElementById('loader').style.display = 'none';

            }

        },

        // Will use popup for IDP Providers sign-in flow instead of the default, redirect.

        signInFlow: 'popup',

        signInSuccessUrl: '<url-to-redirect-to-on-success>',

        signInOptions: [

            firebase.auth.EmailAuthProvider.PROVIDER\_ID

        ]

    };

    ui.start('#firebaseui-auth-container', uiConfig);

    //set value of afdeling selector on settings page

    switch (localStorage.getItem('afd')) {

        case ("/Sloeber"):

            $("#afdselect option[value='1']").attr("selected", "selected");

            break;

        case ("/Speelclub"):

            $("#afdselect option[value='2']").attr("selected", "selected");

            break;

        case ("/Rakwi"):

            $("#afdselect option[value='3']").attr("selected", "selected");

            break;

        case ("/Tito"):

            $("#afdselect option[value='4']").attr("selected", "selected");

            break;

        case ("/Keti"):

            $("#afdselect option[value='5']").attr("selected", "selected");

            break;

        case ("/Aspi"):

            $("#afdselect option[value='6']").attr("selected", "selected");

            break;

    }

    if (localStorage.getItem('afdSet') === null) {

        localStorage.setItem('afd', "/Sloeber");

        localStorage.setItem('afdSet', 1);

    }

    // enables sidenav

    $('.sidenav').sidenav();

    $("#ElementName").css("pointer-events", "none");

    // methods that activate on click of one of the sidenav links

    $('.sidenav a').on('click', function () {

        $('.spa').hide();

        $('#' + $(this).data('show')).show();

        switch (this.innerText) {

            case "ledenlijsten":

                getMemberData();

                $('#tabLists').css({

                    display: "block"

                });

                break;

        }

        $('.sidenav').sidenav('close');

    });

    // starts app on about me page

    $('.tabInformation').show();

    //saves settings to localstorage

    $('#settingsbtn').on('click', function () {

        if ($('#adrescheckbox').is(":checked") == true) {

            fillTable();

        }

        const choice = $('#afdselect').val();

        switch (choice) {

            case ("1"):

                localStorage.setItem('afd', "/Sloeber");

                break;

            case ("2"):

                localStorage.setItem('afd', '/Speelclub');

                break;

            case ("3"):

                localStorage.setItem('afd', '/Rakwi');

                break;

            case ("4"):

                localStorage.setItem('afd', '/Tito');

                break;

            case ("5"):

                localStorage.setItem('afd', '/Keti');

                break;

            case ("6"):

                localStorage.setItem('afd', "/Aspi");

                break;

        }

        ledenChecker = 0;

        $('#aanwLeden').text(ledenChecker);

        navigator.vibrate(500)

        alert("Uw instellingen zijn opgeslagen!");

    });

    //login methods

    $('#loginBtn').on('click', function () {

        const email = $('#email').val();

        const password = $('#password').val();

        signInWithEmailAndPassword(auth, email, password);

    });

    // method for grabbing data out of firebase db

    function getMemberData() {

        const dbRef = ref(getDatabase());

        const afdeling = localStorage.getItem('afd');

        $('#afdTitel').innerText = afdeling;

        get(child(dbRef, 'Members/' + afdeling + '/')).then((snapshot) => {

                if (snapshot.exists()) {

                    $('#memberList').empty();

                    snapshot.forEach((child) => {

                        $('#memberList')

                            .append($('<p></p>')

                                .append($('<label></label>')

                                    .append($('<input type="checkbox" />'))

                                    .append($('<span>' + child.key + '</span>'))));

                    });

                } else {

                    console.log("No data available");

                }

            })

            .catch((error) => {

                console.error(error);

            });

    }

    // on checkbox in ledenlijst checked, procs this function:

    $('#memberList').on('change', "[type='checkbox']", function () {

        if (this.checked) {

            ledenChecker++;

        } else {

            ledenChecker--;

        }

        $('#aanwLeden').text(ledenChecker);

    });

    // fills the table with afdelingdata + adressen

    function fillTable() {

        const dbRef = ref(getDatabase());

        const afdeling = localStorage.getItem('afd');

        get(child(dbRef, 'Members/' + afdeling + '/')).then((snapshot) => {

                if (snapshot.exists()) {

                    $('#adresstable').empty();

                    $('#adresstable')

                    .append($("<tr id='tableheader'></tr>")

                    .append('<th>Naam</th>')

                    .append('<th>Dorp</th>')

                    .append('<th>Straat</th>'));

                    snapshot.forEach((child) => {

                        const childKeyString = child.key.replace("'", "\'");

                        let childKeyId = child.key.replaceAll(" ", "\_");

                        childKeyId = childKeyId.replaceAll("'", "\_");

                        $('#adresstable')

                            .append($('<tr id="' + childKeyId + '"></tr>'));

                        $("#" + childKeyId)

                        .append($('<td>' + childKeyString + '</td>'));

                        var memberSnap = snapshot.child(child.key);

                        memberSnap.forEach((childData) => {

                            $("#" + childKeyId)

                            .append($('<td>' + childData.val() + '</td>'));

                        })

                    });

                } else {

                    console.log("No data available");

                }

            })

            .catch((error) => {

                console.error(error);

            });

    }

});

//firebase authentication methods

//These methods offer possibilities for expanding the app

signInWithEmailAndPassword(auth, email, password)

    .then((userCredential) => {

        // Signed in

        const user = userCredential.user;

    })

    .catch((error) => {

    });

onAuthStateChanged(auth, (user) => {

    if (user) {

        // User is signed in, see docs for a list of available properties

        // https://firebase.google.com/docs/reference/js/firebase.User

        const uid = user.uid;

        // ...

    } else {

        // User is signed out

        // ...

    }

});

signInWithRedirect(auth, new GoogleAuthProvider())

    .then(() => {

        return getRedirectResult(auth);

    })

    .then((result) => {

        const credential = GoogleAuthProvider.credentialFromResult(result);

        // This gives you a Google Access Token.

        // You can use it to access the Google API.

        const token = credential.accessToken;

        // The signed-in user info.

        const user = result.user;

        // ...

    }).catch((error) => {

        // Handle Errors here.

        const errorCode = error.code;

        const errorMessage = error.message;

    });

getRedirectResult(auth)

    .then((result) => {

        const credential = GoogleAuthProvider.credentialFromResult(result);

        if (credential) {

            // This gives you a Google Access Token.

            // You can use it to access the Google API.

            const token = credential.accessToken;

            // The signed-in user info.

            const user = result.user;

            // ...

        }

    })

    .catch((error) => {

        // Handle Errors here.

        const errorCode = error.code;

        const errorMessage = error.message;

    });

## Index.html

<!doctype html>

<html lang="nl">

<head>

    <meta charset="UTF-8">

    <meta name="viewport"

        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

    <link rel="stylesheet" href="css/materialize.1.0.0.min.css">

    <link rel="stylesheet" href="css/app.css">

    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

    <script src="https://kit.fontawesome.com/0806b0bc25.js" crossorigin="anonymous"></script>

    <title>Project Cordova</title>

</head>

<body>

    <header>

        <nav>

            <div class="nav-wrapper">

                <a href="#!" class="brand-logo">Ledenlijsten</a>

                <a href="#" data-target="mobile-demo" class="sidenav-trigger"><i class="material-icons">menu</i></a>

                <ul class="right hide-on-med-and-down">

                    <li><a href="#!">About Me</a></li>

                    <li><a href="#!">Ledenlijsten</a></li>

                    <li><a href="#!">Adressen</a></li>

                    <li><a href="#!">Instellingen</a></li>

                </ul>

            </div>

        </nav>

        <ul class="sidenav" id="mobile-demo">

            <li><a href="#!" data-show="tabInformation">About me</a></li>

            <li><a href="#!" data-show="tabLists">ledenlijsten</a></li>

            <li><a href="#!" data-show="tabAdressen">Adressen</a></li>

            <li><a href="#!" data-show="tabSettings">Instellingen</a></li>

        </ul>

    </header>

    <main class="container">

        <div class="row spa tabInformation" id="tabInformation">

            <div class="col">

                <div class="about-section">

                    <h2>Over deze app</h2>

                    Deze specifieke applicatie dient als een tool om ledenlijsten bij te houden voor Chiro Vorselaar. Elke zondag kan de leiding op de applicatie aanduiden welke leden aanwezig zijn en deze ook terug opvragen. Indien u een administratief verantwoordelijke bent van de chiro kan u ook inloggen met uw credentials en verder informatie per lid opvragen.

                </div>

                <h3 class="centeredTitle">Over mezelf</h3>

                <div class="row">

                    <div class="column">

                        <div class="card">

                            <div class="container">

                                <h3>Luca Van Genechten</h3>

                                <p class="title">Creator</p>

                                <p>Ik ben een tweedejaarsstudent aan Thomas More Geel in de richting Application Development van de IT Factory. Ik heb al 2 jaar eerder gestuurd aan de Karel De Grote Hogeschool te Antwerpen. Mijn doel voor dit jaar is om zonder herexamens af te studeren zodat ik volgend jaar in het buitenland kan studeren.

                                </p>

                                <p>Mail:<br>lucavg@icloud.com</p>

                                <p>Gsm:<br><a href="sms:+0471016602&body=hello!">Stuur mij een bericht!</a></p>

                                <p>Telefoon:<br><a href="tel:+0471016602">Bel mij!</a></p>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <div class="row spa" id="tabLists">

            <div id="currentMembers">

                <h4 id="afdTitel">Ledenlijst</h4>

                <form id="memberList" action="#">

                </form>

                <h4>Aanwezige leden: <span id="aanwLeden">0</span></h4>

            </div>

        </div>

        <div class="row spa" id="tabAdressen">

            <h4 class="centeredTitle">Adreslijst</h4>

            <table id="adresstable">

            </table>

        </div>

        <div class="row spa" id="tabSettings">

            <div>

                <h4>Instellingen:</h4>

                <label for="loginForm" class="settingsLabel">Log in voor adreslijsten</label>

                <div id="firebaseui-auth-container"></div>

                <div id="loader">Loading...</div>

                <label class="settingsLabel" for="afdeling">Afdeling:</label>

                <form action="#">

                    <select name="afdelingselect" id="afdselect" class="show">

                        <option value="0" selected>Maak een keuze</option>

                        <option value="1">Sloeber</option>

                        <option value="2">Speelclub</option>

                        <option value="3">Rakwi</option>

                        <option value="4">Tito</option>

                        <option value="5">Keti</option>

                        <option value="6">Aspi</option>

                    </select>

                    <h5>Adreslijsten:</h5>

                    <p>

                        <label>

                            <input id="adrescheckbox" type="checkbox" disabled="true"/>

                            <span>Adreslijsten tonen?</span>

                        </label>

                    </p>

                    <a href="#" id="settingsbtn" class='col s12 btn btn-large m-md red settingsbutton'>

                        Instellingen opslaan

                    </a>

                </form>

            </div>

        </div>

    </main>

    <script src="https://code.jquery.com/jquery-3.1.0.js" integrity="sha256-slogkvB1K3VOkzAI8QITxV3VzpOnkeNVsKvtkYLMjfk=" crossorigin="anonymous"></script>

    <script src="js/materialize.1.0.0.min.js"></script>

    <script defer src="outputdir/app.bundle.js"></script>

</body>

</html>

## App.cs

$color-alpha: #1772FF;

$color-form-highlight: #EEEEEE;

a.brand-logo {

    font-size: 1.6rem;

    margin: 25;

}

p {

    text-align: justify;

    margin-left: 10px;

}

.disp {

    margin: 15px;

    display: flex;

    justify-content: space-evenly;

    height: 50px;

    position: relative;

}

.photocontainer {

    max-width: 100%;

    max-height: 100%;

}

.card {

    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

    margin: 8px;

}

html {

    box-sizing: border-box;

}

\*,

\*:before,

\*:after {

    box-sizing: inherit;

}

.container {

    padding: 0 16px;

}

.about-section {

    padding: 15px;

    text-align: center;

    background-color: #DD052C;

    color: white;

    width: 100%;

}

.centeredTitle {

    text-align: center;

}

.container::after,

.row::after {

    content: "";

    clear: both;

    display: table;

}

.title {

    color: grey;

}

.button {

    border: none;

    outline: 0;

    display: inline-block;

    padding: 8px;

    color: white;

    background-color: #000;

    text-align: center;

    cursor: pointer;

    width: 100%;

}

.center-vertically {

    margin: 0;

    position: relative;

    top: 50%;

    -ms-transform: translateY(-50%);

    transform: translateY(-50%);

}

.nav-wrapper {

    background-color: #DD052C;

}

.spa {

    display: none;

}

#tabDevice {

    display: block;

}

#compassView {

    position: relative;

    text-align: center;

}

#compassImg {

    position: relative;

    top: -22px;

    left: 0;

}

.settingsLabel {

    font-size: larger;

    color: black;

}

.show {

    display: block;

}

.show option {

    font-size: xx-large;

}

.settingsbutton {

    margin: 20px;

}

table {

    width:100%;

    border:1px solid #EEEEEE;

}

.tableheader {

    display:flex;

    width:100%;

    background:#000;

    padding:(24px \* 1.5) 0;

}

tr {

    display:flex;

    width:100%;

    padding:(24px \* 1.5) 0;

}

tr:nth-of-type(odd) {

    display:flex;

    width:100%;

    padding:(24px \* 1.5) 0;

    background: #EEEEEE;

}

td, th {

    flex: 1 1 20%;

    text-align:center;

}

th {

    text-transform:uppercase;

}

## Webpack.config.cjs

const path = require('path');

const webpack = require('webpack');

const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {

    mode: 'development',

    entry: ['@babel/polyfill', '/www/js/app.js'],

    output: {

        filename: 'app.bundle.js',

        path: path.resolve(\_\_dirname, 'www/outputdir'),

    },

    devtool: "source-map",

    module: {

        rules: [{

                test: /\.txt$/,

                use: 'raw-loader',

            }, {

                test: /\.(jpe?g|png|gif|svg)$/i,

                loader: 'url-loader'

            },

            {

                test: /.css$/,

                use: ['style-loader', 'css-loader'],

            }

        ],

    },

    plugins: [

        new webpack.ProvidePlugin({

            process: 'process/browser',

        }),

        new webpack.ProvidePlugin({

            Buffer: ['buffer', 'Buffer'],

        }),

        new HtmlWebpackPlugin({

            title: 'Production',

            template: 'www/index.html',

        }),

    ],

    externals: {

        "jquery": "jQuery",

    },

    resolve: {

        fallback: {

            assert: require.resolve('assert'),

            buffer: require.resolve('buffer'),

            console: require.resolve('console-browserify'),

            constants: require.resolve('constants-browserify'),

            crypto: require.resolve('crypto-browserify'),

            domain: require.resolve('domain-browser'),

            events: require.resolve('events'),

            fs: false,

            http: require.resolve('stream-http'),

            https: require.resolve('https-browserify'),

            os: require.resolve('os-browserify/browser'),

            path: require.resolve('path-browserify'),

            punycode: require.resolve('punycode'),

            process: require.resolve('process/browser'),

            querystring: require.resolve('querystring-es3'),

            stream: require.resolve('stream-browserify'),

            string\_decoder: require.resolve('string\_decoder'),

            sys: require.resolve('util'),

            timers: require.resolve('timers-browserify'),

            tls: false,

            tty: require.resolve('tty-browserify'),

            url: require.resolve('url'),

            util: require.resolve('util'),

            vm: require.resolve('vm-browserify'),

            zlib: require.resolve('browserify-zlib'),

            child\_process: false,

            buffer: require.resolve("buffer"),

        },

        modules: [

            path.join(\_\_dirname, "js/helpers"),

            "node\_modules"

        ],

    },

};

## Config.xml

<?xml version='1.0' encoding='utf-8'?>

<widget id="be.lucavg.plugins" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">

    <name>Ledenlijsten Chiro Vorselaar</name>

    <description>

        An application which serves to create lists for every member of chiro vorselaar, divided in their respective departments.

    </description>

    <author email="r0849476@student.thomasmore.be" href="https://ledenlijstchirovorselaar.sinners.be/">

        Luca Van Genechten

    </author>

    <content src="index.html" />

    <access origin="\*" />

    <allow-intent href="http://\*/\*" />

    <allow-intent href="https://\*/\*" />

    <allow-intent href="tel:\*" />

    <allow-intent href="sms:\*" />

    <allow-intent href="mailto:\*" />

    <allow-intent href="geo:\*" />

    <platform name="android">

        <allow-intent href="market:\*" />

        <icon density="ldpi" src="resources/android/icon/drawable-ldpi-icon.png" />

        <icon density="mdpi" src="resources/android/icon/drawable-mdpi-icon.png" />

        <icon density="hdpi" src="resources/android/icon/drawable-hdpi-icon.png" />

        <icon density="xhdpi" src="resources/android/icon/drawable-xhdpi-icon.png" />

        <icon density="xxhdpi" src="resources/android/icon/drawable-xxhdpi-icon.png" />

        <icon density="xxxhdpi" src="resources/android/icon/drawable-xxxhdpi-icon.png" />

        <preference name="AndroidXEnabled" value="true" />

        <preference name="android-minSdkVersion" value="22" />

        <preference name="GradlePluginKotlinEnabled" value="true" />

        <preference name="GradlePluginKotlinCodeStyle" value="official" />

        <preference name="GradlePluginKotlinVersion" value="1.3.50" />

    </platform>

    <universal-links>

        <host name="ledenlijstchirovorselaar.page.link" scheme="https" />

        <host name="ledenlijst-chiro.firebaseapp.com" scheme="https">

        <path url="/\_\_/auth/callback"/>

        </host>

    </universal-links>

    <config-file target="config.xml" parent="/\*">

        <preference name="GradlePluginGoogleServicesEnabled" value="true" />

        <preference name="GradlePluginGoogleServicesVersion" value="4.2.0" />

    </config-file>

</widget>