App 17.1

Ich habe das gesamte Projekt mit allen Dateien (Java, Layout, Manifest, etc.) auf Github (https://github.com/RobinHelbig/RadioAppDB) hochgeladen. So ist es vermutlich einfacher die Projektstruktur zu verstehen.

Nichtsdestotrotz habe ich hier nochmal alle Java Klassen aufgelistet.

AppDatabase.java

```
package de.helbigrobin.app13.database;

import android.content.Context;

import androidx.room.Room;
import androidx.room.RoomDatabase;

@androidx.room.Database(entities = {RadioStation.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase{
    public abstract RadioStationDao radioStationDao();

    private static AppDatabase instance;

    public static synchronized AppDatabase getInstance(Context context){
        if(instance == null){
            instance = Room.databaseBuilder(context.getApplicationContext(),
AppDatabase.class, "radioAppDB").createFromAsset("databases/radioAppDB").build();
        }
        return instance;
    }
}
```

RadioStation.java

```
package de.helbigrobin.app13.database;

import androidx.annotation.NonNull;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

import java.io.Serializable;

@Entity
public class RadioStation implements Serializable {
    @PrimaryKey(autoGenerate = true)
    public int uid;

    @ColumnInfo(name = "name")
    public String name;

    @ColumnInfo(name = "stream_url")
    public String streamUrl;
    @ColumnInfo(name = "website_url")
    public String websiteUrl;

    @ColumnInfo(name = "logo_url")
    public String logoUrl;

    @ColumnInfo(name = "favourite")
    public Boolean favourite;

    public RadioStation(String name, String streamUrl, String websiteUrl, String
logoUrl){
        this.name = name;
        this.streamUrl = streamUrl;
        this.websiteUrl = websiteUrl;
        this.logoUrl = logoUrl;
        this.favourite = false;
    }

    @NonNull
    @Override
    public String toString(){
        return name;
    }
}
```

RadioStationDao.java

```
package de.helbigrobin.app13.database;

import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.Query;
import androidx.room.Update;

import java.util.List;

@Dao
public interface RadioStationDao {
    @Query("SELECT uid, name, stream_url, website_url, logo_url, favourite FROM
radiostation")
    List<RadioStation> getAll();

    @Query("SELECT uid, name, stream_url, website_url, logo_url, favourite FROM
radiostation WHERE uid = :uid")
    RadioStation getById(Long uid);

    @Insert
    long insertRadioStation(RadioStation radioStation);

    @Update
    void updateRadioStation(RadioStation radioStation);

    @Delete
    void deleteRadioStation(RadioStation radioStation);
}
```

Configuration.java

```
package de.helbigrobin.app13.mainActivity.fragments;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;

import androidx.annotation.NonNull;
import androidx.appcompat.widget.SwitchCompat;
import androidx.fragment.app.Fragment;

import de.helbigrobin.app13.R;

public class Configuration extends Fragment {
    public Configuration() {
        super(R.layout.fragment_configuration);
    }

    @Override
    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        setupLayout();
    }

    private void setupLayout(){
        String playLastStationKey =
getString(R.string.sharedPreferences_key_playLastStation);
        String playLastStationUIdKey =
getString(R.string.sharedPreferences_key_playLastStation_uid);
        String onlyShowFavouritesKey =
getString(R.string.sharedPreferences_key_onlyShowFavourites);
        SwitchCompat playLastStationSwitch =
getView().findViewById(R.id.playLastStationSwitch);
        SwitchCompat onlyShowFavouritesSwitch =
getView().findViewById(R.id.onlyShowFavouritesSwitch);

        SharedPreferences prefs = getActivity().getSharedPreferences(
                "de.helbigrobin.app13", Context.MODE_PRIVATE);

        boolean playLastStation = prefs.getBoolean(playLastStationKey, false);
        boolean onlyShowFavourites = prefs.getBoolean(onlyShowFavouritesKey,
false);

        playLastStationSwitch.setChecked(playLastStation);
        onlyShowFavouritesSwitch.setChecked(onlyShowFavourites);

        playLastStationSwitch.setOnCheckedChangeListener((buttonView, isChecked) -
> {
            prefs.edit().putBoolean(playLastStationKey, isChecked).apply();

            /*Wenn die Option zum direkten Anzeigen der letzten Station beim
Appstart akiviert oder deaktiviert wird, lösche ich die letzte offene Station, da
der User glaube ich nicht erwartet,
            dass eine Station geöffnet wird, die vor dem Treffen dieser
Einstellung mal offen war.
            */
            prefs.edit().remove(playLastStationUIdKey).apply();
        });
        onlyShowFavouritesSwitch.setOnCheckedChangeListener((buttonView, 
isChecked) -> {
            prefs.edit().putBoolean(onlyShowFavouritesKey, isChecked).apply();
        });
    }
}
```

ConfigureRadioStations.java

```
package de.helbigrobin.app13.mainActivity.fragments;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.ContextMenu;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import java.util.List;

import de.helbigrobin.app13.R;
import de.helbigrobin.app13.RadioStationActivity;
import de.helbigrobin.app13.RadioStationEditActivity;
import de.helbigrobin.app13.database.AppDatabase;
import de.helbigrobin.app13.database.RadioStation;
import de.helbigrobin.app13.database.RadioStationDao;
import de.helbigrobin.app13.mainActivity.MainActivity;

public class ConfigureRadioStations extends Fragment {
    List<RadioStation> radioStations;
    ArrayAdapter<RadioStation> arrayAdapter;

    public ConfigureRadioStations() {
        super(R.layout.fragment_configureradiostations);
    }

    @Override
    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        radioStations = ((MainActivity) getActivity()).radioStations;
        setupRadioStationList();
    }

    /*
    Verändertes RadioStation Objekt wird hier nach dem Bearbeiten zurückgegeben, 
um die Daten in der Liste der Radiostationen zu aktualisieren
```

```
     */
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent intent) {
        super.onActivityResult(requestCode, resultCode, intent);

        if (requestCode == 1) {
            if (resultCode == Activity.RESULT_OK) {
                RadioStation updatedStationData = (RadioStation)
intent.getSerializableExtra("radioStation");
                if (updatedStationData != null){
                    for (RadioStation element : radioStations){
                        if (element.uid == updatedStationData.uid){
                            element.name = updatedStationData.name;
                            element.streamUrl = updatedStationData.streamUrl;
                            element.websiteUrl = updatedStationData.websiteUrl;
                            element.logoUrl = updatedStationData.logoUrl;
                        }
                    }
                    arrayAdapter.notifyDataSetChanged();
                }
            }
        }
    }

    @Override
    public void onCreateContextMenu(@NonNull ContextMenu menu, @NonNull View v,
ContextMenu.ContextMenuInfo menuInfo){
        super.onCreateContextMenu(menu, v, menuInfo);
        getActivity().getMenuInflater().inflate(R.menu.context_menu, menu);
    }

    @Override
    public boolean onContextItemSelected(MenuItem item){
        int position =
((AdapterView.AdapterContextMenuInfo)item.getMenuInfo()).position;
        RadioStation station = radioStations.get(position);

        if(item.getItemId() == R.id.edit) {
            Intent intent = new Intent(getActivity().getApplicationContext(),
RadioStationEditActivity.class);
            intent.putExtra("radioStation", station);
            startActivityForResult(intent, 1);
        } else if(item.getItemId() == R.id.delete){
            AsyncTask.execute(() -> {
                //DB Update
                AppDatabase db = AppDatabase.getInstance(getContext());
                RadioStationDao radioStationDao = db.radioStationDao();
                radioStationDao.deleteRadioStation(station);

                //Synchron im MainThread UI updaten
                new Handler(Looper.getMainLooper()).post(() -> {
                    radioStations.remove(station);
                    arrayAdapter.notifyDataSetChanged();
                });
            });
        } else {
            return super.onContextItemSelected(item);
        }

        return true;
    }
    private void setupRadioStationList(){
        ListView lv = getView().findViewById(R.id.radioEdit_menu);
        arrayAdapter = new
RadioConfigAdapter(getActivity().getApplicationContext(),
R.layout.listviewelement_radioedit,
                radioStations);
        lv.setAdapter(arrayAdapter);
        registerForContextMenu(lv);
    }

    static class RadioConfigAdapter extends ArrayAdapter<RadioStation> {
        private final List<RadioStation> items;
        private final Context context;

        public RadioConfigAdapter(Context context, int layoutResourceId,
List<RadioStation> items) {
            super(context, layoutResourceId, items);
            this.context = context;
            this.items = items;
        }

        @Override
        public View getView(int position, View convertView, ViewGroup parent) {
            View listItem = convertView;
            if(listItem == null)
                listItem =
LayoutInflater.from(context).inflate(R.layout.listviewelement_radioedit,parent,fal
se);

            RadioStation currentStation = items.get(position);

            TextView textView = (TextView)
listItem.findViewById(R.id.radioEditTextView);
            textView.setText(currentStation.name);

            ImageView imageView = (ImageView)
listItem.findViewById(R.id.radioEditImageView);
            if(currentStation.favourite){
                imageView.setImageResource(R.mipmap.favorite_yes);
            } else {
                imageView.setImageResource(R.mipmap.favorite_no);
            }

            imageView.setOnClickListener(view -> {
                //Favoritenstatus togglen
                currentStation.favourite = !currentStation.favourite;

                AsyncTask.execute(() -> {
                    //DB Update
                    AppDatabase db = AppDatabase.getInstance(getContext());
                    RadioStationDao radioStationDao = db.radioStationDao();
                    radioStationDao.updateRadioStation(currentStation);

                    //Synchron im MainThread UI updaten
                    //Neuen Favoritenstatus in der UI setzen
                    new
Handler(Looper.getMainLooper()).post(this::notifyDataSetChanged);
                });
            });
```

```java
            return listItem;
        }
    }
}
```

RadioStationList.java

```java
package de.helbigrobin.app13.mainActivity.fragments;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import java.util.ArrayList;
import java.util.List;

import de.helbigrobin.app13.R;
import de.helbigrobin.app13.RadioStationActivity;
import de.helbigrobin.app13.database.RadioStation;
import de.helbigrobin.app13.mainActivity.MainActivity;

public class RadioStationList extends Fragment {
    List<RadioStation> radioStations;
    ArrayAdapter<RadioStation> arrayAdapter;

    public RadioStationList() {
        super(R.layout.fragment_radiostationlist);
    }

    @Override
    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        radioStations = ((MainActivity) getActivity()).radioStations;
        setupRadioStationList();
    }

    private void setupRadioStationList(){
        String onlyShowFavouritesKey =
getString(R.string.sharedPreferences_key_onlyShowFavourites);
        String playLastStationUIdKey =
getString(R.string.sharedPreferences_key_playLastStation_uid);

        SharedPreferences prefs = getActivity().getSharedPreferences(
                "de.helbigrobin.app13", Context.MODE_PRIVATE);

        boolean onlyShowFavourites = prefs.getBoolean(onlyShowFavouritesKey,
false);

        List<RadioStation> shownRadioStations = new ArrayList<>();
        for(RadioStation station : radioStations){
            if(station.favourite || !onlyShowFavourites){ //Wenn Sender Favorit
oder Einstellung onlyShowFavourites = false
                shownRadioStations.add(station);
            }
        }

        ListView lv = getView().findViewById(R.id.radio_menu);
        arrayAdapter = new ArrayAdapter<>(getActivity().getApplicationContext(),
                android.R.layout.simple_list_item_1,
                shownRadioStations);
        lv.setAdapter(arrayAdapter);
        lv.setOnItemClickListener((parent, view, position, id) -> {
            RadioStation selectedStation = shownRadioStations.get(position);

            //Wenn eine RadioStation geöffnet wird, wird die uid gespeichert, um
sie beim Appstart direkt öffnen zu können
            prefs.edit().putLong(playLastStationUIdKey,
selectedStation.uid).apply();

            Intent intent = new Intent(getActivity().getApplicationContext(),
RadioStationActivity.class);
            intent.putExtra("radioStation", selectedStation);
            startActivity(intent);
        });
        registerForContextMenu(lv);
    }
}
```

MainActivity.java

```
package de.helbigrobin.app13.mainActivity;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.fragment.app.Fragment;

import com.google.android.material.navigation.NavigationView;

import java.util.ArrayList;
import java.util.List;

import de.helbigrobin.app13.R;
import de.helbigrobin.app13.RadioStationActivity;
import de.helbigrobin.app13.RadioStationAddActivity;
import de.helbigrobin.app13.database.AppDatabase;
import de.helbigrobin.app13.database.RadioStation;
import de.helbigrobin.app13.database.RadioStationDao;
import de.helbigrobin.app13.mainActivity.fragments.Configuration;
import de.helbigrobin.app13.mainActivity.fragments.ConfigureRadioStations;
import de.helbigrobin.app13.mainActivity.fragments.RadioStationList;

public class MainActivity extends AppCompatActivity {
    private DrawerLayout drawerLayout;
    private ActionBarDrawerToggle actionBarDrawerToggle;
    private NavigationView navigationView;
    private Class<? extends Fragment>  currentlyShownFragmentClass;

    public List<RadioStation> radioStations = new ArrayList<>();

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

        setupDatabase();
        setupBurgerMenu();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        getMenuInflater().inflate(R.menu.options_menu, menu);
        super.onCreateOptionsMenu(menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item){
        if(item.getItemId() == R.id.finishOption){
            finish();
        } else if (actionBarDrawerToggle.onOptionsItemSelected(item)){
            return true;
        }
        else {
            return super.onContextItemSelected(item);
        }

        return true;
    }

    /*
    Neues RadioStation Objekt wird hier nach dem Erstellen zurückgegeben, um die
Daten in die Datenbank und in die Liste der Radiostationen hinzuzufügen
     */
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent intent) {
        super.onActivityResult(requestCode, resultCode, intent);

        if (requestCode == 1) {
            if (resultCode == Activity.RESULT_OK) {
                //Neue RadioStation
                RadioStation newStation = (RadioStation)
intent.getSerializableExtra("radioStation");
                if (newStation != null){
                    AsyncTask.execute(() -> {
                        AppDatabase db = AppDatabase.getInstance(this);
                        RadioStationDao radioStationDao = db.radioStationDao();
                        long newStationUid =
radioStationDao.insertRadioStation(newStation);

                        //Ich hole den Eintrag, den ich gerade in die Datenbank
geschrieben habe, um sicherzustellen, dass er dort nun auch vorhanden ist
                        RadioStation stationFromDatabase =
radioStationDao.getById(newStationUid);
```

```
                        radioStations.add(stationFromDatabase);

                        //Derzeit angezeigtes Fragment neu laden
                        runOnUiThread(()-
>showFragment(currentlyShownFragmentClass));
                    });
                }
            }
        }
    }

    private void checkIfStationNeedsToBeOpened(){
        String playLastStationKey =
getString(R.string.sharedPreferences_key_playLastStation);
        String playLastStationUIdKey =
getString(R.string.sharedPreferences_key_playLastStation_uid);
        SharedPreferences prefs = getSharedPreferences(
                "de.helbigrobin.app13", Context.MODE_PRIVATE);

        boolean playLastStation = prefs.getBoolean(playLastStationKey, false);

        //Wenn Einstellung zum automatischen Öffnen der letzten Station aktiviert
ist
        if (playLastStation){
            long playLastStationUId = prefs.getLong(playLastStationUIdKey, -1);
            //Wenn beim letzten Beenden der App eine Station offen war
            if(playLastStationUId > -1){
                for(RadioStation station : radioStations){
                    if(station.uid == playLastStationUId){
                        //Öffne Station automatisch
                        Intent intent = new Intent(getApplicationContext(),
RadioStationActivity.class);
                        intent.putExtra("radioStation", station);
                        startActivity(intent);
                    }
                }
            }
        }
    }

    private void setupBurgerMenu(){
        drawerLayout = findViewById(R.id.activity_main);
        actionBarDrawerToggle = new ActionBarDrawerToggle(this,
drawerLayout,R.string.Open, R.string.Close);

        drawerLayout.addDrawerListener(actionBarDrawerToggle);
        actionBarDrawerToggle.syncState();

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        /*
        Die "Radiosender anhören", "Radiosender editieren" und "Allgemeine
Einstellungen" Layouts werden als Fragmente geladen, da so das Burger Menu nicht
jedes mal neu erstellt werden muss.
        "Radiosender hinzufügen" ist eine eigene Activity, da ich dabei kein
Burger Menu anzeigen wollte.
         */
        navigationView = findViewById(R.id.nv);
        navigationView.setNavigationItemSelectedListener(item -> {
            int id = item.getItemId();
            switch(id)
            {
                case R.id.radioStations:
                    showFragment(RadioStationList.class);
                    break;
                case R.id.configureRadioStations:
                    showFragment(ConfigureRadioStations.class);
                    break;
                case R.id.configuration:
                    showFragment(Configuration.class);
                    break;
                case R.id.addRadioStation:
                    Intent intent = new Intent(getApplicationContext(),
RadioStationAddActivity.class);
                    startActivityForResult(intent, 1);
                    break;
                default:
                    return true;
            }

            return true;
        });
    }

    private void setupDatabase(){
        //Asynchron Datenbank anfragen
        AsyncTask.execute(() -> {
            AppDatabase db = AppDatabase.getInstance(this);
            RadioStationDao radioStationDao = db.radioStationDao();
            radioStations.addAll(radioStationDao.getAll());

            //Synchron im MainThread UI updaten
            runOnUiThread(() -> {
                showFragment(RadioStationList.class);
                checkIfStationNeedsToBeOpened();
            });
        });
    }

    private void showFragment(Class<? extends Fragment> fragmentClass){
        currentlyShownFragmentClass = fragmentClass;
        getSupportFragmentManager().beginTransaction()
                .setReorderingAllowed(true)
                .replace(R.id.mainActivity_fragment, fragmentClass, null)
                .commit();
        drawerLayout.closeDrawer(GravityCompat.START);
    }
}
```

IRadioService.java

```
package de.helbigrobin.app13;

public interface IRadioService{
    RadioService.RadioState getState();
    void startRadio();
    void stopRadio();
    void pauseRadio();
    void continueRadio();
}
```

RadioService.java

```
package de.helbigrobin.app13;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Binder;
import android.os.Build;
import android.os.IBinder;
import android.os.PowerManager;
import android.util.Log;
import android.widget.RemoteViews;

import androidx.annotation.Nullable;
import androidx.core.app.NotificationCompat;

import com.squareup.picasso.MemoryPolicy;
import com.squareup.picasso.Picasso;

import de.helbigrobin.app13.database.RadioStation;

public class RadioService extends Service implements IRadioService {
    RadioStation radioStation = null;
    MediaPlayer mediaPlayer;
    RadioBinder binder = new RadioBinder();
    RadioState state = RadioState.Playing;
    RemoteViews notificationContentView;

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        String action = intent.getAction();
        if(action != null){
            //Hier werden die Button OnClicks der ForegroundService Notification
verarbeitet

            if(action.equals("start")){
                if(getState() == RadioState.Stopped){
                    startRadio();
                } else if (getState() == RadioState.Paused){
                    continueRadio();
                } else if(getState() == RadioState.Playing){
                    pauseRadio();
                }
            } else if (action.equals("stop")) {
                if(getState() == RadioState.Paused || getState() ==
RadioState.Playing){
                    stopRadio();
                }
            }
        }
       return START_STICKY;
    }

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        RadioStation radioStation = (RadioStation)
intent.getSerializableExtra("radioStation");
        if (radioStation != null) {
            this.radioStation = radioStation;
            mediaPlayer = new MediaPlayer();
            mediaPlayer.setWakeMode(getApplicationContext(),
PowerManager.PARTIAL_WAKE_LOCK);

            String source = this.radioStation.streamUrl;

            try {
                mediaPlayer.setDataSource(source);
            } catch(Exception e){
                Log.i("statusInfo", "Error: " + e.getMessage());
            }

            startRadio();
        }
        return binder;
    }

    @Override
    public boolean onUnbind(Intent intent) {
        this.radioStation = null;
        setState(RadioState.Stopped);
        mediaPlayer.release();
        stopForeground(true);
        return true;
    }

    @Override
    public RadioState getState() {
        return state;
    }

    public void setState(RadioState state){
        this.state = state;

        //Beim Pausieren und Weiterspielen des Streams den Pause/Weiterspielen
Text im Button der ForegroundService Notification passend setzen
        if(notificationContentView != null && (getState() == RadioState.Paused ||
getState() == RadioState.Playing)){
            if (getState() == RadioState.Paused){

notificationContentView.setTextViewText(R.id.notifcationStartButton,
getString(R.string.weiterspielen));
            } else if(getState() == RadioState.Playing){

notificationContentView.setTextViewText(R.id.notifcationStartButton,
getString(R.string.pause));
            }

            NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, "1")
                    .setContent(notificationContentView)
                    .setSmallIcon(R.drawable.ic_launcher_foreground)
                    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
            startForeground(1, builder.build()); //Buttons der Notification
Updaten
        }
        //RadioStationActivity informieren über neuen Zustand
        Intent sendLevel = new Intent();
        sendLevel.setAction("RADIO_STATE_UPDATE");
        sendLevel.putExtra( "state", getState().ordinal());
        sendBroadcast(sendLevel);
    }

    @Override
    public void startRadio() {
        Log.i("statusInfo", "Start " + radioStation.uid);

        setState(RadioState.Playing);

        displayForegroundMessage();

        try{
            mediaPlayer.setOnPreparedListener(MediaPlayer::start);
            mediaPlayer.prepareAsync();
        } catch(Exception e){
            Log.i("statusInfo", "Error: " + e.getMessage());
        }
    }

    @Override
    public void stopRadio() {
        Log.i("statusInfo", "Stop");
        stopForeground(true);
        setState(RadioState.Stopped);
        mediaPlayer.stop();
    }

    @Override
    public void pauseRadio() {
        Log.i("statusInfo", "Pause");
        setState(RadioState.Paused);
        mediaPlayer.pause();
    }

    @Override
    public void continueRadio() {
        Log.i("statusInfo", "Continue");
        setState(RadioState.Playing);
        mediaPlayer.start();
    }

    void displayForegroundMessage(){
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            //Für neue Android Versionen muss ein Channel erstellt werden
            CharSequence name = "RadioChannel";
            String description = "Für Radio Messages";
            int importance = NotificationManager.IMPORTANCE_LOW;
            NotificationChannel channel = new NotificationChannel("1", name,
importance);
            channel.setDescription(description);
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }


        //Remote View erstellen und mit Infos füllen
        notificationContentView = new RemoteViews(getPackageName(),
R.layout.notification);
        notificationContentView.setTextViewText(R.id.notifcationStartButton,
getString(R.string.pause));
        notificationContentView.setTextViewText(R.id.notifcationStopButton,
getString(R.string.stop));

        //Intent zum Weiterspielen/Pausieren des Streams (ruft den Service mit
Action auf)
        Intent startIntent = new Intent(getApplicationContext(),
RadioService.class);
        startIntent.setAction("start");
        PendingIntent startPendingIntent =
PendingIntent.getService(getApplicationContext(), 0, startIntent, 0);

notificationContentView.setOnClickPendingIntent(R.id.notifcationStartButton,
startPendingIntent);

        //Intent zum Stoppen des Streams (ruft den Service mit Action auf)
        Intent stopIntent = new Intent(getApplicationContext(),
RadioService.class);
        stopIntent.setAction("stop");
        PendingIntent stopPendingIntent =
PendingIntent.getService(getApplicationContext(), 0, stopIntent, 0);

notificationContentView.setOnClickPendingIntent(R.id.notifcationStopButton,
stopPendingIntent);

        //Intent zum Öffnen der RadioStationActivity (wenn neben die Buttons
geklickt wird)
        Intent openIntent = new Intent(getApplicationContext(),
RadioStationActivity.class);
        PendingIntent openPendingIntent =
PendingIntent.getActivity(getApplicationContext(), 0, openIntent, 0);
        notificationContentView.setOnClickPendingIntent(R.id.notificationMain,
openPendingIntent);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
"1")
                .setContent(notificationContentView)
                .setSmallIcon(R.drawable.ic_launcher_foreground)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT);

        Notification notification = builder.build();

        //Bild der Station setzen

Picasso.get().load(radioStation.logoUrl).memoryPolicy(MemoryPolicy.NO_CACHE,
MemoryPolicy.NO_STORE).into(notificationContentView, R.id.notifcationImageView, 1,
notification);

        startForeground(1, notification);
    }

    public class RadioBinder extends Binder {
        public IRadioService getRadioService(){
            return RadioService.this;
        }
    }

    public enum RadioState{
        Stopped, Playing, Paused
    }
}
```

RadioStationActivity

```
package de.helbigrobin.app13;

import android.content.BroadcastReceiver;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.ServiceConnection;
import android.content.SharedPreferences;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.os.IBinder;
import android.util.Log;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

import de.helbigrobin.app13.database.RadioStation;

public class RadioStationActivity extends AppCompatActivity {
    public IRadioService radioService;

    Button startButton, stopButton;
    WebView webView;
    RadioServiceConnection con;
    RadioStateUpdateReceiver radioStateUpdateReceiver;
    WifiStateUpdateReceiver wifiStateUpdateReceiver;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Intent intent = getIntent();

        RadioStation radioStation = (RadioStation)
intent.getSerializableExtra("radioStation");
        if (radioStation != null){
            Intent serviceIntent = new Intent(this, RadioService.class);
            serviceIntent.putExtra("radioStation", radioStation);
            startService(serviceIntent);
            con = new RadioServiceConnection();
            bindService(serviceIntent, con, Context.BIND_AUTO_CREATE);
            setContentView(R.layout.activity_radio);

            //BroadcastReceiver registrieren
            radioStateUpdateReceiver = new RadioStateUpdateReceiver();
            registerReceiver(radioStateUpdateReceiver, new
IntentFilter("RADIO_STATE_UPDATE"));

            wifiStateUpdateReceiver = new WifiStateUpdateReceiver();
            registerReceiver(wifiStateUpdateReceiver, new
IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION));

            startButton = findViewById(R.id.startButton);
            stopButton = findViewById(R.id.stopButton);
            webView = findViewById(R.id.webview);

            showRadioWebsite(radioStation);
        }
    }

    @Override
    protected void onDestroy(){
        super.onDestroy();

        //Broadcast Receiver freigeben
        unregisterReceiver(radioStateUpdateReceiver);
        unregisterReceiver(wifiStateUpdateReceiver);

        if (radioService != null){
            Intent serviceIntent = new Intent(this, RadioService.class);
            stopService(serviceIntent);
            unbindService(con);
        }
    }

    @Override
    public void onBackPressed() {
        if (webView.canGoBack()) { //Wenn geht in Webview zurück navigieren
            webView.goBack();
        } else {
            String playLastStationUIdKey =
getString(R.string.sharedPreferences_key_playLastStation_uid);
            SharedPreferences prefs = getSharedPreferences(
                    "de.helbigrobin.app13", Context.MODE_PRIVATE);

            //Uid der geöffneten RadioStation wird wieder entfernt, da die Station
bewusst geschlossen wurde und somit nicht automatisch beim Appstart geöffnet
werden sollte
            prefs.edit().remove(playLastStationUIdKey).apply();
            super.onBackPressed(); //Ansonsten Action verlassen
        }
    }

    public void onStartButtonClick(View view){
        if(radioService != null){
            if(radioService.getState() == RadioService.RadioState.Stopped){
                radioService.startRadio();
            } else if (radioService.getState() == RadioService.RadioState.Paused){
                radioService.continueRadio();
            } else if(radioService.getState() == RadioService.RadioState.Playing){
                radioService.pauseRadio();
            }
        }
    }

    public void onStopButtonClick(View view){
        if(radioService != null){
            //Im State Stopped kann der Stop Button nicht geklickt werden. Deshalb
wird dieser Fall hier nicht betrachtet
```

```
            if(radioService.getState() == RadioService.RadioState.Paused ||
radioService.getState() == RadioService.RadioState.Playing){
                radioService.stopRadio();
            }
        }
    }


    private void showRadioWebsite(RadioStation radioStation){
        Log.i("statusInfo", "Website " + radioStation.uid);
        webView.setWebViewClient(new WebViewClient());
        webView.getSettings().setJavaScriptEnabled(true);
        webView.loadUrl(radioStation.websiteUrl);
    }

    class RadioServiceConnection implements ServiceConnection {
         boolean bound = false;
         @Override
         public void onServiceConnected (ComponentName arg0, IBinder arg1) {
             radioService  = ((RadioService.RadioBinder) arg1).getRadioService();
             bound = true;
         }

        @Override
         public void onServiceDisconnected(ComponentName arg0) {
             bound = false;
         }
    }

    public class RadioStateUpdateReceiver extends BroadcastReceiver {

        /*
        Der RadioStateUpdateReceiver wird informiert, wenn der Status des
RadioService verändert wird (Stopped, Playing, Paused).
        Das muss passieren, da das Pausieren/Weiterspielen und Stoppen über die
ForegroundService Notification direkt den Service ansteuert. Ohne den
anschließenden Broadcast würde die RadioStationActivity das nicht mitbekommen und
ihre Buttons nicht anpassen.
         */

        @Override
        public void onReceive(Context context, Intent intent) {

            if(intent.getAction().equals("RADIO_STATE_UPDATE"))
            {
                int state = intent.getIntExtra("state",-1);
                if(state >= 0 && state <= 2){
                    RadioService.RadioState radioState =
RadioService.RadioState.values()[state];
                    if(radioState == RadioService.RadioState.Stopped){
                        stopButton.setEnabled(false);
                        startButton.setText(getString(R.string.start));
                    } else if(radioState == RadioService.RadioState.Paused){
                        stopButton.setEnabled(true);
                        startButton.setText(getString(R.string.weiterspielen));
                    } else if(radioState == RadioService.RadioState.Playing){
                        stopButton.setEnabled(true);
                        startButton.setText(getString(R.string.pause));
                    }
                }
            }
        }
    }

    public class WifiStateUpdateReceiver extends BroadcastReceiver {

        /*
        Der WifiStateUpdateReceiver wird informiert wenn sich der Wifi Status
aktualisiert.
         */

        @Override
        public void onReceive(Context context, Intent intent) {

            if
(intent.getAction().equals(ConnectivityManager.CONNECTIVITY_ACTION)) {
                boolean noConnectivity =
intent.getBooleanExtra(ConnectivityManager.EXTRA_NO_CONNECTIVITY, false);
                if(noConnectivity){ //Wenn keine Verbindung, dann Activity beenden
(Radio Stop in onDestroy)
                    finish();
                }
            }
        }
    }
}
```

RadioStationAddActivity

```
package de.helbigrobin.app13;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import de.helbigrobin.app13.database.RadioStation;

public class RadioStationAddActivity extends AppCompatActivity {
    EditText name, stream, website, logo;
    Button confirmButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setupLayout();
    }

    private void setupLayout(){
        setContentView(R.layout.radiostation_form);

        TextView title = findViewById(R.id.title_textview);
        confirmButton = findViewById(R.id.confirm_button);
        name = findViewById(R.id.nameEdit);
        stream = findViewById(R.id.streamEdit);
        website = findViewById(R.id.websiteEdit);
        logo = findViewById(R.id.logoEdit);

        title.setText(R.string.titleAdd);
        confirmButton.setText(R.string.addStation);
        //Button zuerst disabled, weil alle Felder noch leer sind
        confirmButton.setEnabled(false);

        TextChangeListener textChangeListener = new TextChangeListener();
        name.addTextChangedListener(textChangeListener);
        stream.addTextChangedListener(textChangeListener);
        website.addTextChangedListener(textChangeListener);
        logo.addTextChangedListener(textChangeListener);

        confirmButton.setOnClickListener((view -> {
            String nameString = name.getText().toString();
            String streamUrl = stream.getText().toString();
            String websiteUrl = website.getText().toString();
            String logoUrl = logo.getText().toString();

            RadioStation newStation = new RadioStation(nameString, streamUrl,
websiteUrl, logoUrl);

            Intent returnIntent = new Intent();
            returnIntent.putExtra("radioStation", newStation);
            setResult(Activity.RESULT_OK, returnIntent);

            //Synchron im MainThread UI updaten
            finish();
        }));
    }

    private class TextChangeListener implements TextWatcher{
        public void afterTextChanged(Editable s) {
            //Jedes Feld der RadioStation muss einen Wert haben, um sie
abspeichern zu können
            confirmButton.setEnabled(!name.getText().toString().isEmpty() &&
                    !stream.getText().toString().isEmpty() &&
                    !website.getText().toString().isEmpty() &&
                    !logo.getText().toString().isEmpty());
        }

        public void beforeTextChanged(CharSequence s, int start, int count, int
after) {}

        public void onTextChanged(CharSequence s, int start, int before, int
count) {}
    }
}
```

RadioStationEditActivity

```
package de.helbigrobin.app13;

import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import de.helbigrobin.app13.database.AppDatabase;
import de.helbigrobin.app13.database.RadioStation;
import de.helbigrobin.app13.database.RadioStationDao;

public class RadioStationEditActivity extends AppCompatActivity {
    EditText name, stream, website, logo;
    Button confirmButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Intent intent = getIntent();

        RadioStation radioStation = (RadioStation)
intent.getSerializableExtra("radioStation");
        if (radioStation != null){
            setupLayout(radioStation);
        }
    }

    private void setupLayout(RadioStation station){
        setContentView(R.layout.radiostation_form);

        TextView title = findViewById(R.id.title_textview);
        confirmButton = findViewById(R.id.confirm_button);
        name = findViewById(R.id.nameEdit);
        stream = findViewById(R.id.streamEdit);
        website = findViewById(R.id.websiteEdit);
        logo = findViewById(R.id.logoEdit);

        title.setText(getString(R.string.titleEdit).replace("#id#",
""+station.uid));
        confirmButton.setText(R.string.saveChanges);
        name.setText(station.name);
        stream.setText(station.streamUrl);
        website.setText(station.websiteUrl);
        logo.setText(station.logoUrl);

        TextChangeListener textChangeListener = new TextChangeListener();
        name.addTextChangedListener(textChangeListener);
        stream.addTextChangedListener(textChangeListener);
        website.addTextChangedListener(textChangeListener);
        logo.addTextChangedListener(textChangeListener);

        confirmButton.setOnClickListener((view -> {
            station.name = name.getText().toString();
            station.streamUrl = stream.getText().toString();
            station.websiteUrl = website.getText().toString();
            station.logoUrl = logo.getText().toString();

            AsyncTask.execute(() -> {
                AppDatabase db = AppDatabase.getInstance(this);
                RadioStationDao radioStationDao = db.radioStationDao();
                radioStationDao.updateRadioStation(station);

                Intent returnIntent = new Intent();
                returnIntent.putExtra("radioStation", station);
                setResult(Activity.RESULT_OK, returnIntent);

                //Synchron im MainThread UI updaten
                runOnUiThread(this::finish);
            });
        }));
    }

    private class TextChangeListener implements TextWatcher {
        public void afterTextChanged(Editable s) {
            //Jedes Feld der RadioStation muss einen Wert haben, um sie
abspeichern zu können
            confirmButton.setEnabled(!name.getText().toString().isEmpty() &&
                    !stream.getText().toString().isEmpty() &&
                    !website.getText().toString().isEmpty() &&
                    !logo.getText().toString().isEmpty());
        }

        public void beforeTextChanged(CharSequence s, int start, int count, int
after) {}

        public void onTextChanged(CharSequence s, int start, int before, int
count) {}
    }
}
```