```cpp
#include "filefinderaudio.h"

FileFinderAudio::FileFinderAudio(QObject *parent) :
    DatabaseOperationsAudio(parent)
{
}

void FileFinderAudio::getDirTree()
{
    if(mPath.front()==NULL)
    {
        return;
    }
    mDir= new QDir(mPath.front());

    //Assuming the directory mentioned in the mPath exist

    if(mDir->exists()==true)
    {
        mFileQueue.push(mDir->absolutePath());
    }
    else
    {
//        QMessageBox::information(NULL,"Error","Directory :: "+mPath.front()+" doesn't
exist");
        return;
    }

    QFileInfoList vFileList;

    QStringList vNameFilter,vDefaultNameFilter=mDir->nameFilters();
    vNameFilter<<"*.mp3"<<"*.ogg"<<"*.flac"<<"*.midi"<<"*.wav"<<"*.wma";

    mDir->setSorting(QDir::Name);

    while(mFileQueue.empty()==false)
    {
        mDir->setPath(mFileQueue.front());

        mDir->setNameFilters(vNameFilter);
        mDir->setFilter(QDir::NoDotAndDotDot|QDir::Readable|QDir::Files);


        //        vFileList=mDir->entryInfoList();
        //        emiting the current directory as mDir -> absolutePath()
        //        and the files in the directory as mDir-> entryInfoList()


        updateDB(mPath.front(),QFileInfo(mDir->absolutePath()),mDir->entryInfoList());
        //        emit(updateDirTreeView(QFileInfo(mDir->absolutePath()),mDir-
>entryInfoList()));

        mDir->setNameFilters(vDefaultNameFilter);
        mDir->setFilter(QDir::NoDotAndDotDot|QDir::Readable|QDir::Dirs);
        vFileList=mDir->entryInfoList();


        for(QFileInfoList::size_type i=0; i <vFileList.count(); ++i)
        {
            mFileQueue.push(vFileList[i].absoluteFilePath());
        }


        mFileQueue.pop();

    }

    delete mDir;
}

void FileFinderAudio::setPath(QString vPath)
{
    mPath.push(vPath);
}

void FileFinderAudio::initiator()
{
        if(mPath.front()!=NULL)
    {
        updateSource(mPath.front());
        getDirTree();
        updateTreeView();
        mPath.pop();
    }

}
```