

УДК 004.457

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ АКТУАЛИЗАЦИИ МУЗЫКАЛЬНОЙ БИБЛИОТЕКИ НА РАЗЛИЧНЫХ УСТРОЙСТВАХ

Набережнев К.Е., студент гр. ИТм-151, II курс
Научный руководитель: Ванеев О.Н., к.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева
г. Кемерово

Каждый, кто имеет дело с музыкальными файлами, может столкнуться с проблемой актуализации музыкальной библиотеки на различных устройствах. Актуальность данной проблемы обусловлена нарастающим количеством устройств, таких как настольный или портативный ПК, смартфон, музыкальный плеер и прочие устройства. Пользователь может хранить музыкальную коллекцию на стационарном устройстве, и получать из нее актуальные музыкальные композиции на смартфон или другое переносимое устройство. При этом появляется проблема актуализации. Актуализация музыкальной библиотеки заключается в обновлении информации о наличных треках, их характеристик на различных устройствах, воспроизводящих музыкальные файлы. Файлы на устройствах могут различаться метаданными, либо даже качеством. Для решения данной проблемы пользователю приходится пользоваться различными интернет-сервисами, которые могут не всегда удовлетворять требования. Например, пользователь может:

- Слушать музыку по интернету на специализированных сайтах в ущерб качеству музыки и отсутствие ручного выбора песен. Требуется интернет соединение, что может быть достаточно проблематично.
- Продолжать слушать музыку непосредственно со стационарного хранилища с использованием специального программного обеспечения (например, по протоколу SMB: «ES Проводник» и «Black Player» на устройствах под управлением ОС Android). Требуется постоянное соединение с устройством-сервером, что достаточно проблематично.
- Продолжать слушать музыку со стационарного хранилища путем ручного копирования новых файлов с помощью приложений, использующих такие протоколы, как SMB, FTP, MTP, HTTP и другие (например, подключить к компьютеру устройство под управлением ОС Android по USB и копировать на него файлы с помощью проводника по протоколу MTP). Требуется актуализация. Производить придется вручную. Также, использование метаданных музыкальных файлов минимально.

Метаданными музыкального файла называются данные, которые содержат описательную информацию о музыкальном файле, такую как испол-

нитель, альбом, название песни, номер трека в альбоме, обложка альбома, и прочую информацию для идентификации файла. Такие метаданные также называют музыкальными тегами. Они заполняются в преобладающем большинстве музыкальных файлов, и это позволяет использовать метаданные для структуризации музыкальной библиотеки.

Цель данной работы состоит в обеспечении автоматизированных возможностей актуализации музыкальных библиотек на различных устройствах.

Основными задачами данной работы являются:

- Обеспечить возможность получения новых музыкальных файлов с другого устройства;
- Обеспечить возможность получения измененных метаданных музыкальных файлов с другого устройства, если идентичные файлы уже имеются;
- Обеспечить возможность удаления отсутствующих на другом устройстве музыкальных файлов;
- Дополнительными задачами работы являются:
- Обеспечить базовые возможности для работы с метаданными музыкальных файлов;
- Обеспечить возможность создания различных шаблонов для автоматизации актуализации музыкальной библиотеки;
- Обеспечение кроссплатформенности.

Для разработки информационной системы были выбраны язык C# и среда разработки Visual Studio. Для обеспечения кроссплатформенности все библиотеки для бизнес-логики пишутся с использованием технологии .NET Standard. Это позволит использовать один код для всех платформ, что ускорит разработку и последующую модификацию. Для написания кода, который специфичен на каждой платформе, но используется в портативных библиотеках классов, будет использоваться паттерн «Внедрение зависимости» («Dependency Injection»).

Базовый возможности для работы с метаданными музыкальных файлов включают в себя такие задачи:

- Изменение метаданных музыкального файла;
- Вложение обложки музыкального файла в метаданные;

Технология .NET Standard предоставляет API для работы с файлами и потоками (Stream), что значительно упростит задачу. Также для непосредственной работы с метаданными используется библиотека TagLib#.

Для реализации возможности создания различных шаблонов необходимо хранить метаданные. Для этого используется библиотека EntityFrameworkCore. Библиотека представляет широкие возможности для удобной работы с базами данных. Также библиотека поддерживает .NET Standard, что идеально подходит для решения задачи.

Представлены шаблоны следующих видов:

- Шаблоны для перемещения – позволит организовать структуру папок и название файла, используя для этого хранимые метаданные;

- Шаблоны для текста – позволит обрабатывать тестовые поля метаданных так, как требует этого пользователь;
- Шаблоны для дубликатов – позволит установить приоритет качества музыкальных файлов удаления дубликатных файлов.

Для обеспечения возможности получения новых музыкальных файлов с другого устройства, необходимо идентифицировать песни. Если реализовать идентификацию песен, то получить новые файлы не составит сложностей. Новых файлов может быть большое количества, а также там могут находиться файлы, которые пользователь не захочет копировать. Для этого нужно реализовать различного вида фильтрации, для того, чтобы можно было выбрать те файлы, которые нужно копировать, а какие нужно игнорировать.

Для обеспечения возможности удаления отсутствующих на другом устройстве музыкальных файлов также необходимо идентифицировать песни. Также, как и при получении новых файлов, файлов может быть много, а также среди удаляемых файлов могут быть те, которые пользователь хочет оставить. Аналогично, нужно реализовать фильтрации и файлы для игнорирования.

Для обеспечения возможности получения измененных метаданных музыкальных файлов также необходимо идентифицировать песни. Наряду с идентификацией песен нужно определить, изменилась ли песня.

Для реализации основных задач требуется идентифицировать песни. Для этого было решено использовать ключ, который создается из метаданных. Ключи хранятся в базе данных. Создаются при добавлении музыкальных файлов в базу данных. Это позволит идентифицировать музыкальный файл. Наряду с ключом нужно хранить информацию для получения данных об изменениях. Это может быть либо полная информация о музыкальном файле, либо просто дата последнего изменения. Для возможности получения измененных метаданных даты должно быть достаточно.

Исходя из вышесказанного, нужно создать сущность, которая будет связана с песней, иметь ключ и дату последнего изменения. Созданная сущность названа SongSymbol.



Рисунок 1. Сущность для идентификации песни

При изменении метаданных песни также может измениться и ключ. Это может создать проблемы в идентификации песен. Для того, чтобы избежать проблем нужно создавать новый ключ с обновленной датой, а старый ключ не удалять и дату оставить. Это позволит в лучшем виде реализовать задачу по изменении метаданных музыкальных файлов. Пример хранения ключей представлен в приложенном файле.

Из всего вышесказанного можно сделать следующий вывод: разработанная информационная система позволит:

- Получать новые музыкальные файлы с другого устройства;
- Получать измененные метаданные музыкальных файлов с другого устройства, если идентичные файлы уже имеются;
- Удалять отсутствующие на другом устройстве музыкальные файлы;
- Работать с метаданными музыкальных файлов;
- Использовать шаблоны для автоматизации актуализации музыкальной библиотеки.

Список литературы:

1. Server Message Block [Электронный ресурс] – Электрон. текстовые данные. – 2017. – Режим доступа: https://ru.wikipedia.org/wiki/Server_Message_Block