**Введение**

Данный курсовой проект является проектом по разработке программного обеспечения, направленного на простое взаимодействие с треками на вашем устройстве. Зачастую для проигрывания треков не нужно подключение к интернету, но также есть и обратная сторона: синхронизация с базой данной для конкретного пользователя. Для обеспечения полного опыта в проигрывании музыки на различных персональных компьютерах было принято решение разрабатывать данное программное средство.

Цель курсового проекта – это создание программного обеспечения с графическим интерфейсом, направленного на простое, быстрое и удобное управление музыкальными композициями. Данное программное средство должно в малой степени нагружать систему, на котором используется приложение, и должно обладать удобным и понятным графическим интерфейсом для пользователей.

Перед началом выполнения проекта следует выделить наиболее важные задачи, которые будут заложены в основу данного программного обеспечения. Наиболее важной подзадачей является проектирование архитектуры программного средства, т. к. это является определяющим фактором дальнейшего развития и функционирования приложения. Следующий по степени влияния на дальнейшую разработку проекта это логика работы приложения, т.е. следующий этап – проектирование логики работы программного средства. Выбор языка программирования является ключевым аспектом данного проекта, т.к. нам потребуется высокоуровневый язык программирования, который сможет поддерживать интеграцию с графическим интерфейсом, а также позволит не сильно нагружать систему. Следующие этапы перечислю в виде списка, так как их аспекты будут раскрыты немного позже:

* Выбор технологии для создания графического интерфейса
* Проектирования эргономичного графического интерфейса
* Создание кода программного средства
* Тестирование программного средства

Минимальный функционал программы должен включать в себя возможность проигрывания треков с устройства и из базы данных, переключение между треками, систему аккаунтов, возможность добавлять или убирать треки, редактировать персональную информацию пользователя. Данные пункты должны быть использованы с помощью графического интерфейса.

Приняв во внимание все факты, изложенные выше, было принято решение использовать парадигму объектно-ориентированного программирования (ООП), что позволит сделать код более структурированным и обеспечить более легкую поддержку кода в будущем для сторонних разработчиков, если таковые появятся. Для разработки архитектуры приложения принято решение использовать архитектурный паттерн проектирования Model-View-Controller (MVC), который позволит разделить разрабатываемое приложение на слой представлений, которые будут отвечать за графическое отображение интерфейса, слой контроллера, который будет принимать на себя роль управления информацией от модели к представлению и наоборот, и модели, которая будет производить основную обработку информации приложения.

Выбор языка программирования оказывает существенное влияние на дальнейший процесс разработки, использованные технологии и скорость выполнения поставленных задач. Так как для нас будет важна скорость работы программы, широкая поддержка на различных устройствах, а также возможность использовать парадигму ООП, то стоит выбрать язык, созданный для работы с учетом всех этих условий. Основываясь на вышеперечисленных фактах был выбран язык программирования “C++”. Данный язык имеет поддержку ООП, является малотребовательным к возможностям персонального компьютера, например, если сравнивать с языком «Python», а также является гибким инструментом разработки за счет многолетнего существования на рынке и имеет многочисленное количество фреймворков.

В качестве среды разработки были выбраны следующие программы:

* CLion – для разработки основной части программы
* QT Creator – для разработки графического интерфейса, так как данная среда разработки предоставляет встроенный графический редактор.

В качестве основного фреймворка для разработки программы была выбрана технология QT, которая является кроссплатформенным инструментарием для разработки прикладного программного обеспечения, широко используемой для создания графических интерфейсов. Предоставляет быструю многоуровневую разработку, кроссплатформенность и имеет хорошую документацию.