

Міністерство освіти і науки України
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА
Кафедра прикладних інформаційних систем

122 Комп'ютерні науки
Освітня програма «Прикладне програмування»

Кваліфікаційна робота бакалавра

на тему:

ПРОГРАМНИЙ ЗАСТОСУНОК ДОБИРАННЯ МУЗИЧНОГО КОНТЕНТУ

Виконав студент 4 курсу групи ПП-41

(Підпис)

Бачинський А. І.
(прізвище, ім'я, по батькові)

Керівник проф., д.т.н. Сайко В. Г.
(прізвище, ім'я, по батькові)

(Резолюція «До захисту»)

Попередній захист:

(Висновок: "До захисту в екзаменаційній комісії")

Завідувач кафедри _____ Плескач В.Л. _____
(Підпис) (Прізвище, ініціали) (Дата)

Засвідчую, що у цьому курсовому проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2021 року

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

Назва теми: «Програмний застосунок добирання музичного контенту»

Освітня програма: Прикладне програмування
Спеціальність: Комп'ютерні науки

ПІБ

Підпис

Бачинський Андрій Ігорович

Назва роботи українською та англійською мовами

Програмний застосунок добирання музичного контенту

Software application for selecting music content

Мета бакалаврської роботи, завдання

Мета бакалаврської роботи: Створення зручного медіаплеєру для відтворення та обирання музики.

План роботи:

1. Сучасні підходи до розроблення і реалізації програм для прослуховування музики.
2. Аналіз архітектурних рішень і вибір програмних засобів для реалізації системи.
3. Програмна реалізація застосунку добирання музичного контенту.

ПІБ, ступінь, звання наукового керівника роботи: _____

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№ п/п	Назва етапів бакалаврської роботи	Термін виконання етапів бакалаврської роботи	Відмітка про виконання
1.	Вибір теми та наукового керівника бакалаврської роботи	26.10.2020	Виконано
2.	Видача завдання бакалаврської роботи	23.11.2020	Виконано
3.	Настановча групова співбесіда з бакалаврської роботи	01.12.2020	Виконано
4.	Затвердження плану бакалаврської роботи	18.02.2021	Виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	25.02.2021	Виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	05.03.2021	Виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	09.04.2021	Виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	07.05.2021	Виконано
9.	Подання роботи у першому варіанті	11.05.2021	Виконано
10.	Оформлення пояснювальної записки бакалаврської роботи	12.05.2021	Виконано
11.	Подання бакалаврської роботи на попередній захист	24.05.2021	Виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедру	28.05.2020	Виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврську роботу)	11.06.2020	Виконано
14.	Захист бакалаврської роботи	22.06.2020	

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1
Завдання до дипломної роботи	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	2
Вступ	3
Розділ 1	14
Розділ 2	17
Розділ 3	9
Висновки	1
Перелік використаних джерел	2
Додатки	9

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата	Відомість дипломної роботи	Лист	Листів
Розробн.	Бачинський А. І.					
Керівн.	Сайко В. Г.					
Н/контр.	Макаренко С.А.					
Зав.каф.	Плескач В.Л.					

АНОТАЦІЯ

До бакалаврської дипломної роботи Бачинського Андрія Ігоровича на тему
«Програмний застосунок добирання музичних композицій»

Ця дипломна робота присвячена створенню програмного застосунку для відтворення музичних композицій з використанням динамічної та мультифункціонального фреймворку QT, а також використання SQLite для збереження прогресу використання програми.

Для розробки оптимальної програмної системи було проведено дослідження нині актуальних мультимедійних програвачів та технологій, які використовуються для відтворення музики. Також був проведений аналіз музичних файлів(безпосередній зміст файлів mp3, mp4).

Основним та найкращим, на мою думку, інструментом для створення програми було обрано QT5.

У цій роботі проведені дослідження найпопулярніших програмних застосунків, програмних інструментів для розробки, детальний опис встановлення та налаштування розробленої системи.

Загальний обсяг роботи: 69 сторінок.

Ключові слова: IDE, QT5, Cmake, програмний застосунок.

ANNOTATION

To the bachelor's thesis of Bachynsky Andriy Ihorovich on the topic "Software application for selecting Musical compositions"

This thesis is devoted to creating a software application for playing music tracks using the dynamic and multifunctional QT framework, as well as the use of SQLite to save the use of the program.

To develop the optimal software system, a study of current media players and technologies used to play music was conducted. There was also an analysis of music files (direct content of mp3 files, mp4).

The main and best, in my opinion, tool for creating a program was referring to QT5.

In this work researches of the most popular software applications, software TOOLS for development, the detailed description of installation and adjustment of the developed system are carried out.

TOTAL VOLUME OF WORK: 69 pages.

Keywords: IDE, QT5, CMake, software application, C++.

ЗМІСТ

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ	3
АНОТАЦІЯ	5
ANNOTATION	6
ВСТУП	8
РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО МУЗИЧНІ ЗАСТОСУНКИ. ТЕХНОЛОГІЇ РОЗРОБКИ МУЗИЧНИХ ЗАСТОСУНКІВ.	12
1.1 Передумови розвитку застосунків для прослуховування музики	12
1.2 Робочі платформи	14
1.3 Фреймворк Qt	16
1.3.1 Qt C++	17
1.3.2 Qt Jambia	18
1.3.3 Функціонал та склад Qt	18
1.3.4 Платформи, які підтримує фреймворк Qt	19
1.3.5 Компоненти Qt	21
1.3.6 Qt 3d візуалізація музики	23
Висновки	24
РОЗДІЛ 2 СТРУКТУРА МУЗИЧНИХ ФАЙЛІВ. АРХІТЕКТУРА ТА ІНСТРУМЕНТИ РОЗРОБКИ МУЗИЧНОГО ЗАСТОСУНКУ	26
2.1 Аналіз вхідних даних	26
2.2 Архітектура програмного застосунку	27
2.3 Структура музичних файлів	28
2.3.1 MP3 формат	28
2.3.2 MP4 формат	30
2.3.3 WAV формат	32
2.4 Опис технологій та додатків для розробки програмного застосунку	35
2.4.1 Веб-сервіс для хостингу ІТ-проектів GitHub	35
2.4.2 Кросплатформенна система автоматизації збірки CMake	37
2.4.3 IDE Qt creator	37
2.4.4 Система управління базами даних SQLite	38

	8
2.5 Реалізація програмного застосунку	39
2.6 Опис функціоналу	40
Висновки	41
РОЗДІЛ 3 АНАЛІЗ ФУНКЦІОНАЛУ, ОПИС ІНСТАЛЯЦІЇ.	43
ІНСТРУКЦІЯ З ЕКСПЛУАТАЦІЇ	43
3.1 Процес запуску програмного застосунку для відтворення музичних композицій	43
3.2 Опис та огляд основного функціоналу програмного застосунку	45
3.2.1 Головне вікно програмного застосунку	46
3.2.2 Доступ до дискового сховища та завантаження музичних композицій	47
3.2.3 Функція радіо	49
3.2.4 Можливості відтворення та налаштування тегів у композиціях	50
3.2.5 Налаштування черги музичних треків	50
ВИСНОВОК	52
ПЕРЕЛІК ПОСИЛАНЬ	53
ДОДАТКИ	55
Додаток А. Код реалізації радіо	55
Додаток Б. Реалізація головної сторінки	59
Додаток В. Реалізація налаштувань черги музичних композицій	63

ВСТУП

Музика завжди займала важливу роль у нашому житті. Кожного дня під час занять спортом, прогулянки вулицею, на шляху до роботи людина прослуховує музику за допомогою спеціалізованого програмного забезпечення (стрімінгових сервісів та програмних програвачів музики). Звичайному користувачу дуже важливо, щоб ця платформа була зручною та легкою у використанні, а також мала весь потрібний функціонал.

Щоденний стрес може швидко зламати людину, однак психологи давно стверджують, що хороша музика - найкраща профілактика депресії і може її зупинити. Але важливо розуміти, що вибір хорошої музики у всіх абсолютно різний. Так, наприклад, якщо вам подобається творчість Світлани Лободи, не слід сподіватися, що симфонії Баха позитивно вплинуть на вашу ситуацію.

Також пошук нових композицій, які вам подобаються, стимулює вивільнення ендорфінів та утворення нервових зв'язків. Це точно буде корисно в інших сферах життя.

Актуальність досліджень. Програмні застосунки для відтворення музики, інакше кажучи музичні плеєри, які були створені для прослуховування музичних композицій. У нашому світі є велика кількість модифікацій плеєрів, які використовуються на мобільних телефонах, планшетах та комп'ютерах. Я обрав саме комп'ютерний застосунок, адже є користувачем даної платформи, і відтворювачі музики, які існують зараз, мають ряд недоліків, на які я буду вказувати впродовж свого дослідження. Мій програмний застосунок є унікальним адже зібрав у собі весь функціонал, якого не вистачає у звичайному музичному відтворювачі.

Більшість плеєрів являються досить складними у використанні. До них залучено велику кількість непотрібного функціоналу для звичайного користувача. Деякі програми навіть відтворюють композиції з рекламою на початку. Звісно, всі розуміють, що розробники теж мають за щось жити, але потрібно створювати якісний контент, який би чесно купували на ринку, а не

завантажували до мережі безкоштовний продукт, з переліком платного контенту.

Весь процес відтворення контенту у моєму застосунку заключається у виборі музичної композиції, долучення цієї композиції у плейлист та безпосереднє прослуховування композиції за допомогою програмного застосунку. Так, звісно, перед цим треба завантажити композицію, яка цікавить, але користувач зможе дуже легко та без перешкод прослухати її після завантаження на персональний комп'ютер.

Об'єктом дослідження є особливості використання музичних плеєрів на персональних комп'ютерах користувачів, завантаженням контенту у програмний застосунок та маніпуляції з музичними треками.

Предметом дослідження є процес відтворення музичних доріжок за допомогою актуального програмного застосунку.

Мета дослідження полягає в ознайомленні з науковою літературою з обраного напрямку дослідження, розробка структури програмного застосунку та протоколів вивантаження даних, проведенні дослідження методів обробки музичних композицій та відтворення музичних треків.

Завдання нашого дослідження можна сформулювати наступним чином:

- опрацювати сучасну наукову літературу за обраним напрямком дослідження, привести визначення основних понять теми;
- провести аналіз предметної області та формалізацію основних понять;
- зробити огляд програмного забезпечення для розробки програмного застосунку
- проаналізувати концепцію системи програмного застосунку;
- описати програмне забезпечення, яке було використане для розробки застосунка: середовище розробки, бібліотеки для візуалізації, а також структуру музичних файлів(розібрати їх зміст);
- провести розробку програмного застосунку за допомогою QT creator, а також створити базу даних у SQLite

Новизна дослідження. Зроблено широкий літературний пошук з детальним аналізом наукової інформації. Проведено аналіз актуальних альтернативних технологій, та вказано на переваги нового програмного застосунку.

РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО МУЗИЧНІ ЗАСТОСУНКИ. ТЕХНОЛОГІЇ РОЗРОБКИ МУЗИЧНИХ ЗАСТОСУНКІВ.

Для того щоб створити застосунок з добирання та прослуховування музики потрібно проаналізувати діяльність інших музичних додатків (відтворювачів музики). Необхідно проаналізувати архітектуру побудови такого програмного забезпечення, алгоритми підбору, бібліотеки для відтворення звуку та обрати інструменти для створення програмного проекту. У цьому розділі буде згруповано основну інформацію щодо функціонування додатків, які використовуються для прослуховування музичних композицій.

1.1 Передумови розвитку застосунків для прослуховування музики

Кожне покоління відзначається власною музикою та носіями, за допомогою яких воно їх слухає. Приблизно 60 років тому молоді люди танцювали під хіти Елвіса Преслі та танці популярних джазових колективів під звуки патефонів. Але прогрес не зупиняється і щороку музичні оператори вдосконалюються та розвиваються.

Для багатьох людей музика є партнером у їх повсякденній діяльності. Хтось слухає улюблені пісні, гуляючи, подорожуючи, займаючись спортом і навіть працюючи. Я є одним з таких людей. Сьогодні для прослуховування музики вам потрібен лише один смартфон, а якщо бажаєте кращої якості - потрібні навушники або портативну акустику. Однак така музика як зараз була доступною не завжди. Раніше вам доводилося купувати багато дорогих інструментів, щоб насолоджуватися улюбленою піснею. Десятиліття тому люди ходили по магазинах та на ринках в пошуках касети. Сьогодні культура споживання цього мистецтва докорінно змінилася.

З появою потокових служб, які дозволяють отримувати пісні за передплатою, ми можемо слухати сотні пісень і забути більшість із них наступного дня. Це погано? Кожен повинен відповісти на це питання сам. На мій погляд, це не так, адже це лише результат нашого розвитку, появи цифрових технологій. Дивно, але кожен із нас сьогодні має в кишенях смартфон, який вміщує багато знань, незліченну кількість книг, фільмів та

музики різних епох. Але це було не завжди так. Повернемося на століття-два, коли перший музичний плеєр тільки зароджувався.

Почнемо з первинних записуючих пристроїв. Перший пристрій, який отримав назву фонографа, був виготовлений відомим американським ученим і винахідником Томасом Едісоном. Воскові диски використовувались як носії звуку, а тонка голка та тиснення - для перетворення тертя голки для читання у звук у наступних репродукціях. Якість таких записів бажала кращого, що не дивно, але чи не буде це подальшим розвитком беззвучного запису.

Однак наступним переломним моментом у розвитку аудіотехнологій став винахід магнітного методу запису та відтворення композицій.

Ера компакт-дисків настала після компакт-касет. Так, ці компакт-диски Sony і Philips вважаються популярними промоутерами цього формату. Однак відомо, що загальна концепція оптичного запису інформації була винайдена в 1965 році американським інженером Джеймсом Расселом. Це підтверджується заявками на патенти поданими ним у 1966 та 1969 роках. Однак радянські вчені Олександр Прохоров та Микола Басов розробили ці "холодні" лазери, які почали застосовувати в техніці за допомогою оптичних дисків. У 1964 р. обидва винахідники були номіновані на Нобелівську премію.

Завдяки новому принципу запису даних компакт-диски першого покоління записали до 650 МБ даних. Іншими словами, він міг записати близько години аудіо на одному диску, чого було достатньо для альбому. Першим аудіо-компакт-диском, який вийшов на ринок, став альбом ABBA "The Visitor". Пізніше великі ІТ-компанії зробили внесок у розповсюдження компакт-дисків та оптичних дисків. Наприклад, Джон Скаллі, колишній генеральний директор Apple, заявив журналістам у 1987 році, що оптичні носії з часом здійснять революцію в електронних пристроях. Загалом, так і було. Говорячи про музику, наступним переломним пунктом стала поява нового цифрового формату - MP3.

1.2 Робочі платформи

Розвиток музичних програвачів дійшов до початків цифрових технологій. Будь-який MP3 плеєр, комп'ютер, телефон, планшет потребують програмного забезпечення для відтворення аудіофайлів. Мій проект розроблявся для конкретних операційних систем, а саме:

1. Windows - це сімейство комерційних операційних систем Microsoft, орієнтованих на управління графікою. Спочатку Windows був графічним доповненням до популярної операційної системи MS-DOS у 1980-х та 1990-х.

Мій проект підтримує актуальні версії цього сімейства операційних систем, таких як:

- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 8.1
- Windows 10

Додаткова інформація про операційні системи надана у таблиці 1.1

Таблиця 1.1 – Версії операційних систем

Назва	Дата виходу	Остання версія
Windows Server	2003-2019	1809 (10.0.17763) (11 листопада 2018)
Windows XP	25 листопада 2001	5.1.2600.5512 SP3 (21 квітня 2008)

Продовження таблиці 1.1

Windows Vista	30 січня 2007	6.0.6001 / SP2 Build 6002 (25 травня 2009)
Windows 7	22 жовтня 2009	6.1.7601 / SP1 Build 7601 (22 січня 2011)
Windows 8	26 жовтня 2012	6.2.9200 (26 жовтня 2012)
Windows 8.1	17 жовтня 2013	6.3.9600 (17 жовтня 2013)
Windows 10	29 липня 2015	2009 (20H2) (1 жовтня 2020)

2. Linux - це сімейство Unix-подібних операційних систем на основі ядра Linux, яке включає різні утиліти, програми та інші компоненти проекту GNU. Для цієї операційної системи теж є актуальні версії:

- Ubuntu (18.04, 20.04, 21.04)
- Mint
- OpenSUSE
- Debian
- Arch Linux
- Deepin OS
- Elementary OS
- Manjaro Linux
- Fedora
- Kali Linux

3. MacOS - це сімейство власних операційних систем, випущених корпорацією Apple. Призначений для лінійки персональних комп'ютерів Macintosh. Широке використання графічного інтерфейсу користувача в сучасних операційних системах часто входить в MacOS. Перший

персональний комп'ютер Macintosh 128K був представлений в 1984 році.
Актуальні версії наведено у таблиці 1.1

Таблиця 1.2 – Актуальні версії операційних систем

Назва MacOS	Версія
MacOS X Cheetah	10.0.4
MacOS X Puma	10.1.5
MacOS X Jaguar	10.2.8
MacOS X Panther	10.3.9
MacOS X Tiger	10.4.11
MacOS X Leopard	10.5.8
MacOS X Snow Leopard	10.6.8
MacOS X Lion	10.7.5
MacOS X Mountain Lion	10.8.5
MacOS X Mavericks	10.9.5
MacOS X Yosemite	10.10.5
MacOS E Capitan	10.11.6
macOS Sierra	10.12.6
macOS High Sierra	10.13.6
macOS Mojave	10.14.6
macOS Catalina	10.15.7
macOS Big Sur	11.3

1.3 Фреймворк QT

Qt є основою для розробки програмного забезпечення платформи мовою програмування C ++. Qt для багатьох мов програмування: Python - PyQt, є

бібліотечні пакети, які дозволяють скористатися перевагами PySide; Рубін - QtRuby; Java - Qt Jambi; PHP - PHP-Qt та інші.

З часу свого створення в 1996 році бібліотека лягла в основу багатьох програмних проєктів. Qt також є основою популярного робочого столу KDE, який постачається з багатьма дистрибутивами Linux.

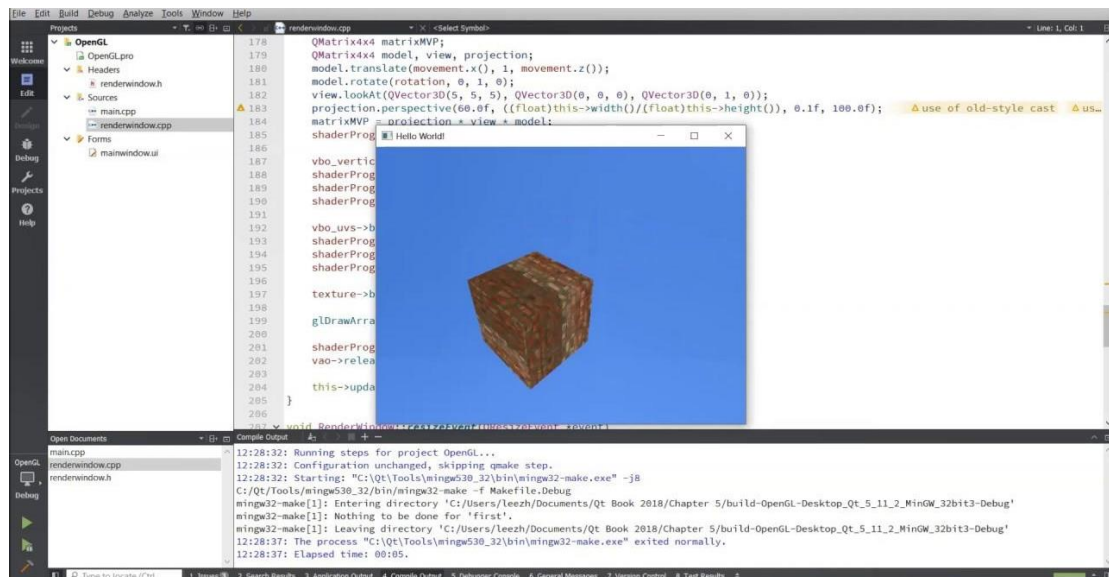


Рисунок 1.1 - Візуалізація за допомогою фреймворку Qt.

1.3.1 Qt C++

Сьогодні практично неможливо уявити програму без графічного інтерфейсу користувача (GUI). Windows API має необхідні інструменти для створення графічного інтерфейсу. Однак їх використання трудомістке і вимагає великого практичного досвіду. Навіть бібліотека MFC, розроблена для спрощення процесу написання графічних інтерфейсів у C++ для Windows, не забезпечує простоти створення програм, які ми хочемо. Найбільшим недоліком цих інструментів та бібліотек є їх залежність від платформи. На відміну від MFC, бібліотека Qt є крос-платформною. У той же час він надає можливість створювати інтерактивний графічний інтерфейс.

Бібліотека Qt - це сукупність класів (понад 500), визначених у модулях, наприклад - QtCore, QtGui, QtWidgets, QtOpenGL. Модуль QtCore в основному відповідає за зв'язок ОС та управління подіями. Модулі QtGui та QtWidgets

включають класи програмування графічного інтерфейсу. Модуль QtOpenGL дозволяє використовувати OpenGL.

1.3.2 Qt Jambi

Qt Jambi - це бібліотека Java, яка є графічним фреймворком для Qt. Це дозволяє розробникам Java використовувати Qt у своїх проектах.

Qt Jambi підтримує Linux, Unix, Mac OS X та Microsoft Windows.

Офіційна підтримка Nokia для Qt Jambi закінчилася в березні 2010 року і наразі проект підтримується спільнотою відкритих джерел.

1.3.3 Функціонал та склад Qt

Фреймворк Qt дозволяє програмному забезпеченню працювати на більшості сучасних операційних систем, а не просто модифікувати вихідний код та створювати програмне забезпечення для кожної системи. Це включає всі основні класи, які можуть знадобитися при розробці додатків, від елементів графічного інтерфейсу користувача до сіток, баз даних та класів XML. Він повністю об'єктно-орієнтований, розгортає та підтримує методи програмування компонентів.

Деякі особливості включають використання компілятора мета-об'єктів, системи розробки вихідного коду. Покращення забезпечує система плагінів, які можна розмістити безпосередньо на панелі візуального редактора. Користувач також може розширити звичайну функціональність віджетів залежно від розташування на екрані, відображення, розподілу при зміні розміру вікна.

Розташування на екрані, відображення, розподіл при зміні розміру вікна.

Цей графічний інтерфейс доповнений інтуїтивно зрозумілим середовищем розробки для Qt Designer, яке дозволяє створювати бесіди та форми в режимі WYSIWYG. Qt супроводжується Qt Linguist, утилітою Qt, яка спрощує локалізацію та переклад на багато мов, спрощує роботу з документами в бібліотеці та підтримує платформу створення програмного забезпечення на базі Qt. Починаючи з версії 4.5.0, цей пакет включає середовище розробки Qt Creator, яке включає редактор коду, довідку, графічні інструменти Qt Designer та можливості редагування. Qt Creator може використовувати GCC або

Microsoft VC ++ як компілятор, а GDB як компілятор. Ця бібліотека для версій Windows була доповнена компілятором, заголовком MinGW та об'єктними файлами.

Існують версії бібліотеки для Microsoft Windows, графічної системи UNIX Systems X11, Android, iOS, Mac OS X, Microsoft Windows CE, QNX, Linux та платформи S60. Триває перехід на Windows Phone та Windows RT. Також будуються порти до Хайку та Тизена.

Деякий час бібліотека також поширювалася у версії Qt / Embedded і розроблялася на мобільних пристроях, але з середини 2000-х вона була розділена на окремі продукти Qtoria.

Починаючи з версії 4.5, Qt поширюється за трьома ліцензіями:

- Qt Commercial - для розробки власного ліцензійного програмного забезпечення Qt допускає модифікації без змін;
- GNU GPL - для розробки розподілених відкритих джерел в умовах GNU GPL, а також для модифікації Qt;
- GNU LGPL - для розробки власного програмного забезпечення.

Вихідний код, однаковий для всіх варіантів ліцензії, знаходиться у вільному доступі у сховищі Git, розташованому на веб-сайті Github. На додаток до вихідного коду Qt є сховища суміжних бібліотек, розроблених авторами та спільнотою бібліотеки.

До версії 4.0.0 Qt / Mac, Qt / X11, Qt / Embedded випускалися лише за безкоштовною ліцензією, але з 4.0.0 (наприкінці червня 2005 р.) Програмне забезпечення Qt також випустило Qt / Windows. У той же час існували безкоштовні сторонні версії Qt / Windows на базі Qt / X11 до 4.0.0.

1.3.4 Платформи, які підтримує фреймворк Qt

Актуальні платформи наведені у таблиці 1.3.

Таблиця 1.3 – Актуальні платформи

Платформа	Опис
Linux/Unix	
X11	QT для віконного менеджера X(Linux, FreeBSD, HP-UX, Solaris, ALX та інші).
Wayland	Qt для Wayland. Додатки на Qt можуть перемикатися між графічними бекенда начебто X і Wayland під час завантаження, якщо додати опцію командного рядка - platform. Це дозволяє додаткам непомітно переходити з X11 на Wayland.
Вбудовані Linux-системи	Qt для вбудованих систем: КПК, смартфонів. Існує у вигляді декількох платформ, в залежності від технології відтворення. DirectFB, LinuxFB і EGLFS (EGL Full Screen).
Android	Qt для Android, Раніше відомий як Necessitas.
Платформи Apple	
OS X	Qt для Apple OS X; підтримка додатків на Cocoa.
IOS	Qt для IOS платформи (Iphone, Ipad).
Платформи Microsoft	
Windows	Qt для Microsoft Windows XP, Vista, 7, 8, 10.

Продовження таблиці 1.3

Windows CE	Qt для Windows CE 6, Windows Embedded Compact 7.
Windows RT	Підтримка для додатків розроблених на основі WinRt для Windows 8 та Windows Phone 8. Починаючи з версії 5.4: Windows Phone 8.1.
Інші платформи	
Integrity	Qt для Integrity
QNX	Qt для QNX
VxWorks	Qt для VxWorks

1.3.5 Компоненти Qt

Бібліотека розділена на ряд модулів:

- QtCore – основні класи бібліотеки, що використовуються іншими модулями;
- QtGui – компоненти графічного інтерфейсу;
- QtWidgets – включає класи для класичних віджетних програм, модуль витягується з QtGui в Qt 5;
- Qt QML – модуль для підтримки QML;
- QtNetwork – це набір класів мережевого програмування. Підтримка різних протоколів високого рівня може бути різною. У версії 4.2.x є класи для роботи з протоколами FTP та HTTP. Такі класи як QtcpServer, QtcpSocket для TCP та QudpSocket для UDP призначені для роботи з протоколами TCP / IP;
- QtOpenGL – набір класів для роботи з OpenGL;
- QSql – це набір класів для роботи з базами даних за допомогою SQL. Основними класами цього модуля у версії 4.2.x є: QSqlDatabase – клас, що забезпечує підключення до бази даних, об'єкт із класу QsqlDriver потрібен для роботи з будь-якою

конкретною базою даних – абстрактний клас, реалізований для конкретної бази даних і може вимагають створення бази даних SDK. Наприклад, для створення драйвера для СУБД Firebird або InterBase вам знадобляться .h-файли та бібліотеки статичних посилань, що входять до пакету доставки цього MBV;

- QtScript – класи для роботи зі сценаріями Qt;
- QtSvg – класи для відображення та роботи з даними масштабної векторної графіки (SVG);
- QtXml – модуль, що підтримується моделями XML, SAX та DOM;
- QtDesigner – класи для створення розширень для ваших особистих віджетів;
- QtUiTools – класи для розробки форм Qt Designer у додатку;
- QtAssistant – Довідкова система;
- Qt3Support – модуль з класами, необхідними для сумісності з бібліотекою Qt 3.x.x;
- QTest – класи, що підтримують модульне тестування;
- QtWebKit – це модуль WebKit, який інтегрований у Qt і доступний через класи. (Застарілий від 5,6-Qt);
- QtWebEngine доступний через інтегрований модуль Qt Chromium та його класи.
- QtXmlPatterns – модуль для підтримки Xquery 1.0 та Xpath 2.0;
- Phonon – модуль, що підтримує відтворення відео та аудіо та запис із локальних пристроїв та пристроїв та мереж (починаючи з Qt 5, замінюючи QtMultimedia);
- QtMultimedia – модуль, що підтримує відтворення відео та аудіо та запис із локальних та пристроїв та мереж;
- QtCLucene – повнотекстовий модуль підтримки пошуку, що використовується в новій версії Помічника в Qt 4.4;
- ActiveQt – це модуль для роботи з технологіями ActiveX та COM для розробників Qt під Windows.

- QtDeclarative – це модуль, який забезпечує декларативну структуру для створення динамічних, налаштування користувацьких інтерфейсів.

Також використовувалася технологія WoC – віджети на полотні, за допомогою плазми в KDE 4.1, віджети з бібліотеки Qt можна використовувати безпосередньо в аплетах. Забезпечує макет віджетів QGraphicsView із масштабованістю та різними графічними ефектами.

Бібліотека використовує власний формат проекту, який називається файлом .pro, що містить інформацію про те, які файли збирати, як шукати вихідні файли та багато іншого. Потім, використовуючи утиліту qmake, вони редагують утиліту для компілятора. Ви також можете працювати з інтеграторами з Microsoft Visual Studio 2003/2005/2008/2010. Інтеграція в Eclipse не існує з 2010 року.

Розробники Java можуть застосовувати Qt, використовуючи фреймворки Qt Jambi того самого постачальника (Qt офіційно припинив розробку цього фреймворку з 2009 року).

1.3.6 Qt 3d візуалізація музики

По мірі розробки модуля Qt 3D я хотів би використати його для тестування та перевірки як ми можемо відтворювати музику за допомогою цього модуля. Результатом цього експерименту став новий приклад аудіо-візуалізатора для модуля Qt 3D. Приклад зображено на рисунку 1.1

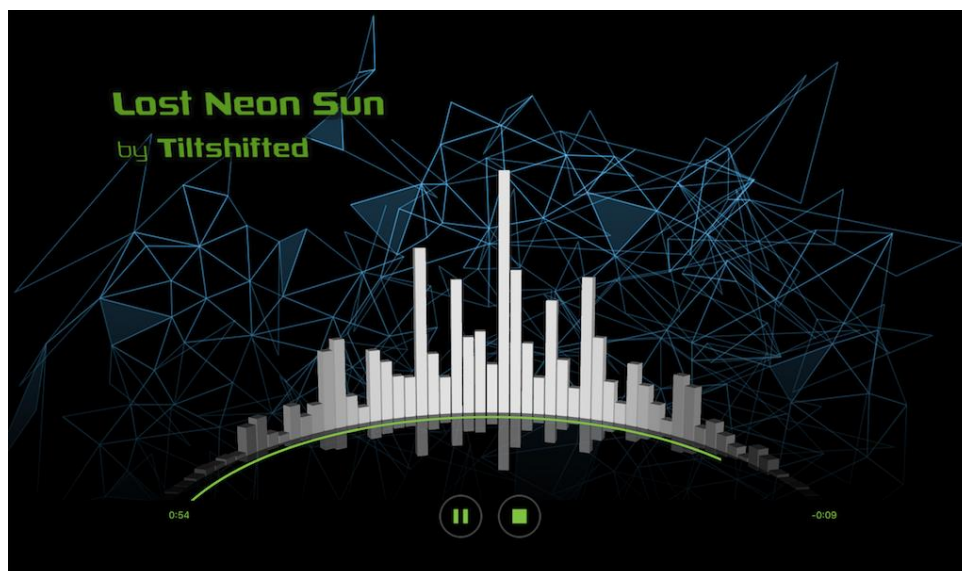


Рисунок 1.2 – Приклад 3D візуалізації музики Qt

Приклад показує як можна використовувати елементи Qt 3D з елементами Qt Quick. 3D-вміст відображається у форматі Scene3D. Колонки є об'єктами NodeInstantiator і анімовані шляхом візуалізації амплітуди частот музики, яку вони відтворюють. Кладемо кнопки паузи / відтворення та паузи на верхній шар сцени. Ці кнопки є елементами Qt Quick. Музика відтворюється через MediaPlayer, який постачається з модулем Qt Multimedia.

У цьому прикладі є вигнута лінія, яка огинає тривимірні об'єкти, щоб показати прогрес по довжині доріжки. Існує також 3D-призма, яка показує приклад та назву пісні. Заголовок пісні відображається на призмі, коли пісня відтворюється, а назва прикладу відображається при зупинці треку. Назви розміщуються по різні боки призми, а розворот призми анімується, коли потрібно змінити назву.

Якщо ви хочете спробувати приклад, ви можете знайти його на codereview.qt-project.org із прикладами \ qt3d \ audio-visualizer-qml. Звісно, ми знаємо, що у фоновому режимі ще є можливості для вдосконалення, щоб зробити анімацію стовпців більш плавною, але оскільки Qt 3D - це технологія, яка буде розглянута в майбутньому випуску Qt 5.6, ці покращення ймовірні.

Цей приклад розроблений, щоб дати нам уявлення про те, що ми можемо створити за допомогою Qt 3D.

Висновки

Провівши аналіз та опис інструментарію для розробки програмного застосунку з добирання музичних композицій, можна зробити висновок, що фреймворк Qt5 є досить актуальним для розробки такого програмного забезпечення. Він містить у собі всі необхідні засоби для реалізації та візуалізації програмного продукту.

Також було описано найбільш актуальні операційні системи для програмного застосунку та обрано мультиплатформенний варіант проекту, адже метою усіх програмних застосунків є охоплення великої цільової аудиторії.

РОЗДІЛ 2 СТРУКТУРА МУЗИЧНИХ ФАЙЛІВ. АРХІТЕКТУРА ТА ІНСТРУМЕНТИ РОЗРОБКИ МУЗИЧНОГО ЗАСТОСУНКУ

У цьому розділі проводиться опис роботи з розробки програмного застосунку для відтворення музичних композицій. Для якісної реалізації будуть проведені наступні кроки:

- аналіз вихідних даних;
- опис архітектури створення програмного застосунку;
- структура музичних файлів;
- опис технологій та додатків для розробки програмного застосунку;
- реалізація програмного застосунку;
- опис функціоналу;
- тестування веб-застосунку.

2.1 Аналіз вхідних даних

Програмний застосунок добирання музичних композицій являє собою інформаційну структуру, яка має доступ до мережі Інтернет. Доступ потрібен для додаткової функції – радіо. В основній цільовій аудиторії програмного застосунку можна виділити такі групи: школярі, студенти, та люди віком 30-60 років.

Для створення програмного застосунку було виведено наступну реалізацію програми:

- головне вікно програми, яке містить навігаційне меню, функціональні клавіші, додаткові налаштування, перехід до меню радіо;
- вікно для додавання бібліотек з музичними композиціями, яке має доступ до каталогу власного пристрою, щоб можна було знайти потрібні композиції;
- вікно для налаштування треків та зміни структури музичного файлу (назва, текст пісні, картинка альбому або треку);
- вікно для відтворення пісень, вибору треку, переключення на радіо.

Розглянувши інформацію вхідних ресурсів, можна зробити висновки, що програмний застосунок повинен відповідати таким вимогам:

- створення сучасного та зручного програмного застосунку;
- мультиплатформенність для комфортного використання додатку на різних операційних системах;
- для зручності використання винести на головне вікно основний інструментарій;
- для ширшого використання застосунку додати функціонал радіо;
- для текстової інформації обрати єдиний шрифт, стиль та розмір;
- реалізувати зручне керування додатком;
- додати інструкцію з використання додатку;
- реалізувати збереження прогресу користуванням додатком.

2.2 Архітектура програмного застосунку

Архітектура музичного додатку виконується за простою формулою: “Створюйте. Зберігайте. Відтворюйте.”. Для досягнення цієї мети буде надана платформа, яка практично не вимагає обслуговування, має мінімальний час відгуку та підтримує різні швидкості передачі даних. Архітектура музичного застосунку показана на рисунку 2.1. Вона надає всі необхідні інструменти для створення, розповсюдження та відтворення мультимедійної інформації.

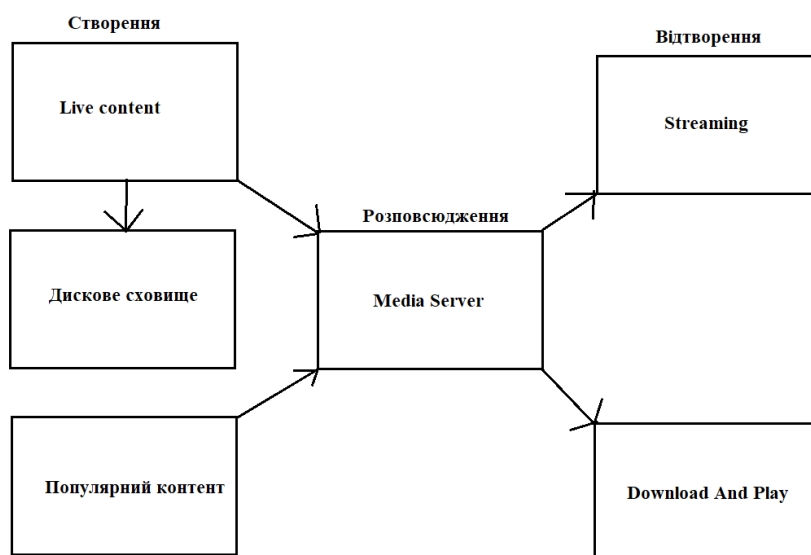


Рисунок 2.1 – Архітектура додатку

Платформа складається з низки важливих компонентів, які надає фреймворк Qt5, а саме:

- відтворення медіа-файлів формату mp3, mp4, wav;
- реструктуризація музичних файлів;
- відтворення музичних композицій з різноманітних каналів радіо;
- доступ до бібліотек персонального пристрою.

2.3 Структура музичних файлів

Так як у застосунку є можливість видозмінювати структуру медіа-файлів, то я хотів би детально розглянути будову mp3, mp4 та wav форматів. Вони є актуальними у наш час.

2.3.1 MP3 формат

MP3 - це формат файлу розроблений групою MPEG для зберігання аудіо-інформації. Формат був ліцензований, але 23 квітня 2017 року всі патенти закінчились, а плата за ліцензію була призупинена.

Також це один з найпопулярніших форматів цифрового кодування аудіо-інформації. Широко використовується у мережах обміну файлами для оцінки завантажень музики. Формат можна відтворювати на багатьох популярних операційних системах, більшості портативних аудіо-плеєрів і підтримується усіма сучасними моделями музичних центрів та DVD-програвачів.

MP3 використовує алгоритм стиснення втрат, призначений для різкого зменшення обсягу даних, необхідного для збільшення запису та забезпечення звуку, близького до оригіналу (на думку більшості слухачів), але якість звукової системи непомітна під час прослуховування. Принцип стиснення полягає у зменшенні точності певних частин звукового потоку, який практично не відрізнити від того, що сприймається людським слухом. Оскільки через вікові природні зміни слухового апарату в більшості випадків вони відрізняються у високоякісному звуковому обладнанні. Цей метод називається кодуванням сприйняття. У той же час, на першому етапі звукова діаграма будується у вигляді короткочасної послідовності, а потім видаляється інформація, непомітна для людського вуха, а решта даних є невеликою формою. Цей метод схожий на метод стиснення, який використовується для стиснення зображень у формат JPEG. Коли MP3 створюється із середньою

швидкістю 128 кбіт, в результаті виходить файл розміром приблизно 1/11 вихідного файлу CD-Audio (не стиснутий аудіоформат CD-Audio зі швидкістю 1411,2 кбіт / с).). Файли MP3 можна створювати на високій або низькій швидкості, що впливає на якість отриманого файлу.

Файл MP3 складається з декількох частин MP3 (фреймів), які, в свою чергу, складаються із заголовка та блоку інформації. Цю послідовність фрагментів називають елементарним потоком. Фрагменти не є незалежними елементами ("байтове сховище"), тому їх не можна довільно отримувати. Блок даних файлу MP3 містить стиснуту аудіоінформацію у вигляді частоти та амплітуди. На діаграмі нижче видно, що назва MP3 складається з маркера, який використовується для пошуку правильного фрагмента MP3. Потім додаються два біти, що вказують на те, що використовується MPEG і рівень 3; іншими словами, він виявляє MPEG-1 Audio Layer 3 або MP3. Наступні значення можуть відрізнятися залежно від типу файлу MP3. Стандарт ISO / IEC 11172-3 визначає набір значень для кожного відділу голови разом із загальним описом. В даний час більшість файлів MP3 мають метадані ID3 до або після фрагмента MP3; вони також показані на схемі.

Є певні недоліки цього формату:

1. Технічні дефекти. Кількість звукових каналів обмежена двома, на відміну від AAC та Vorbis. Існує суворе обмеження швидкості відбору: неможливо встановити швидкість відбору, яку ви хочете. Максимальна швидкість вибору - 48 кГц для MP3, 192 кГц для Warbis та 96 кГц для AAC. MP3 можна зберігати лише на таких частотах вибору: 8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100 та 48000 Гц.

2. Правові обмеження. Патент MP3 належить компанії Alcatel-Lucent, для якої потрібна ліцензія на деякі способи використання цього формату (термін дії патентів на MP3 закінчився 23 квітня 2017 року).

Винаходи, опубліковані більше одного року в США, не запатентовані; однак для патентів, виданих до 8 червня 1995 р. (майже весь стандарт був опублікований 6 грудня 1991 р.), можна було продовжити термін їх дії. Патенти

на дешифрування MP3, відомі в США, втратили чинність у грудні 2012 року; Згідно з іншими джерелами, цього не було у вересні 2015 року, враховуючи лише патенти, видані до грудня 1992 року. У 2017 році всі патенти, пов'язані з цим форматом, втратили чинність, оскільки вони не були поновлені правовласниками.

2.3.2 MP4 формат

Формат MP4 / MPEG-4, випущений у 2001 році відповідно до ISO / IEC 14496-1: 2001, застосовує технологію кодування AAC, яка використовує стиснення втрат для запобігання користувачеві копіювати дані. Однак цифрові відео- та аудіо-дані, що містяться у файлі MP4, відтворюються в Інтернеті, а не надсилаються електронною поштою. Це протилежно формату MPEG (аудіо/відео), оскільки файли MP4 можуть містити іншу інформацію, таку як імена файлів, дані файлів та зображення. Слід зазначити, що файли MPEG-4 мають лише одне розширення - MP4. Окремі потоки зазвичай потрібні для введення даних у файл MPEG-4.

Файли MP4 зазвичай вважаються відео-файлами і містять цифрове відео, реалізоване зі специфікаціями кодування даних MPEG-4 та стандартами стиснення відео. У файлі MP4 аудіо- та відео- сегменти файлу стискаються окремо. Його аудіодані інтегровані зі стисненням аудіо AAC, а вбудовування відео здійснюється з характеристиками кодування відео MPEG-4. Як результат, ці файли MP4 можна зберігати на зовнішніх запам'ятовуючих пристроях та оптичних носіях та оптимізувати для портативних якісних відео для розповсюдження цих файлів MP4 в Інтернеті. Існує безліч програм Microsoft Windows, які можна використовувати для відкриття, створення та редагування цих файлів MP4, а також існує безліч програм Linux і Mac, які дозволяють користувачам створювати та переглядати вміст цих відеофайлів MP4.

Актуальні програми для операційної системи Windows відображено на рисунку 2.1. Для MacOS на рисунку 2.2. Для Linux на рисунку 2.3.

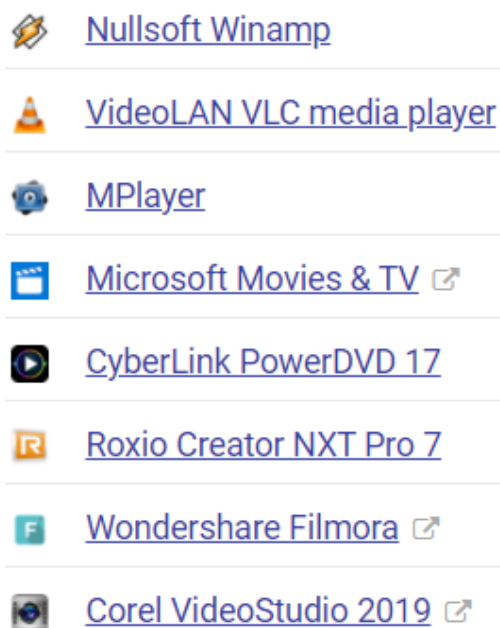


Рисунок 2.1 – Програмне заюзпечення для Windows

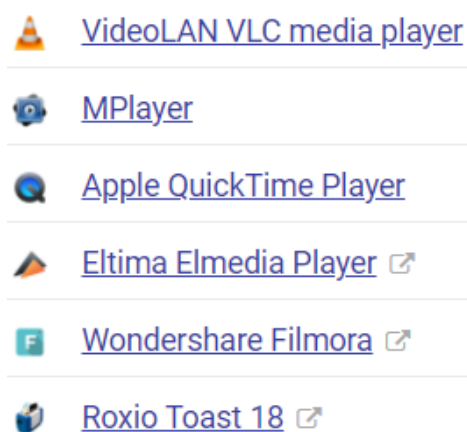


Рисунок 2.2 – Програмне заюзпечення для MacOS

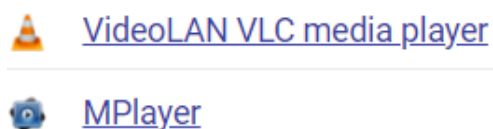


Рисунок 2.3 – Програмне заюзпечення для Linux

Переваги MP4 формату:

1. Якість джерела є дуже важливою. Наприклад, оригінальне відео має низьку роздільну здатність, а отримане погіршення не перетворюється у формат mpeg4, оскільки якість виводу значно покращується.

2. Під параметрами стиснення слід розуміти швидкість передачі даних, роздільну здатність зображення та частоту кадрів в секунду. За мінімальних налаштувань розмір файлу мінімальний, і не важко припустити, що якість огидна.

3. Апаратне забезпечення комп'ютера, який буде відтворювати відео, дуже важливе. Таким чином, на дуже слабких машинах помітні всі недоліки у відео (спотворення зображення, падіння кадрів).

Недоліки MP4 формату:

До недоліків формату Mpeg4 можна віднести низьку якість відео та звуку, порівняно з форматом DVD. Однак, DVD-відео більше, ніж mp4. Іноді відео у форматі mpeg4 може бути неприємним при появі квадратів, за наявності повільних переходів кольорів або послідовних "ламаних" кадрів.

Сьогодні медіа-файли у форматі mpeg4 ідеально підходять як демонстраційний матеріал для клієнтів. Працюючи зі стандартним пакетом Adobe Premier, користувачі не зможуть зберегти відредаговане відео у форматі mpeg4, але їх легко вилікувати, встановивши спеціальні плагіни. Наприклад, LSX-MPEG 2.01 для Adobe Premiere. Цей плагін дозволяє зберігати не тільки медіа у форматі mp4, але також mpeg 1 та mpeg 2, а при необхідності ви можете легко перекодувати медіа-файл у форматі mp4 в будь-який інший файл за допомогою інших програм.

2.3.3 WAV формат

Тож розглянемо найпоширеніший файл WAV (Windows PCM). У ньому чітко розмежовано дві частини. Перша - це початок файлу, інша - область даних. Ім'я файлу зберігає наступне:

- Розмір файлу.
- Кількість каналів.
- Швидкість відбору.

- Кількість бітів у вибірці (це значення ще називають глибиною звуку).

Але щоб краще зрозуміти значення заголовка, нам слід поговорити про інформаційне середовище та оцифровку звуку. Звук складається з вібрацій, які при оцифровці мають поступовий вигляд. Цей тип пов'язаний з тим, що комп'ютер здатний видавати звук певної амплітуди (гучності) за короткий проміжок часу, і що ця коротка хвилина нескінченно коротка. Довжина цього інтервалу визначає швидкість вибору. Наприклад, у нас є файл із швидкістю вибору 44,1 кГц, тому цей короткий час становить $1/44100$ за секунду (що впливає із значення $\text{Гц} = 1 / \text{с}$). Сучасні звукові карти підтримують швидкість дискретизації до 192 кГц. Отже, з часом ми це визначили.

Тепер, що стосується амплітуди (гучності звуку за короткий час). Амплітуда відображається у файлі як число 8, 16, 24, 32 біта (більш теоретично можливо). Точність звуку, я б сказав, залежить від точності амплітуди. Як відомо, 8 бітів = 1 байт, тому одне значення амплітуди за короткий час у файлі займає 1, 2, 3, 4 байта відповідно. Таким чином, чим більше місця займає файл у файлі, тим ширший діапазон можливих значень для цього числа, а отже, і вища точність амплітуди.

Для файлів PCM точність (або укуси) може бути наступною:

- 1 байт / 8 біт - -128 ... 127;
- 2 байти / 16 біт - -32 760 ... 32 760;
- 3 байти / 24 біта - -1 ... 1 (з плаваючою точкою);
- 4 байта / 32 біта - -1 ... 1 (з плаваючою точкою).

Однак, перелік можливих глибин насправді досить великий, показуючи лише найпопулярніші.

Поєднання амплітуди і короткого часу називається закономірністю.

Недоліками(труднощами) з роботою над такими файлами є:

1. Значення `ChunkSize`, звичайно, дуже велике. Це трапляється під час спроби зчитування даних у потоковому режимі. Наприклад, коли декодер LAME виводить результат декодування в `STDOUT`, у цьому полі `0x7FFFFFFF + 44 - 8`, у `subchunk2Size` `0x7FFFFFFF` (це максимальне значення 32-бітового

цілого числа зі знаком). Це пояснюється тим, що декодер у цьому режимі не видає результат повністю, а в малих наборах даних, і не може передбачити кінцевий обсяг інформації.

2. Може бути більше двох вкладок, наприклад, `ffmpeg 4.1.3 ffmpeg -i example.mp3 -f wav example.wav` При спробі декодувати аудіо за допомогою універсального декодера, окрім `fmt` та підкатегорій даних, які слід враховувати, декодований файл - це ще один СПИСОК перед областю даних, яку він включає. Таким чином, для отримання доступу до інформації необхідно передати непотрібні підсхеми на інформацію. Зробити це не складно, оскільки ви можете прочитати ідентифікатор підрядку та його розмір, а якщо це не інформація, ви передасте інформацію залежно від її розміру.

Ви можете використовувати різні типи інформації під час читання заголовка. Наприклад, в C (MSVS) ви можете використовувати `__int32` або `DWORD` замість масиву `char [4]`, але тоді це може бути менш незручним для порівняння з деякими рядковими константами. Я також хотів би попередити вас про 64-розрядні операційні системи. Зокрема, завжди важливо пам'ятати, що в C довжина змінної `int` у 64-бітовій системі дорівнює 8 байтам, а в 32-бітовій системі - 4 байти. У таких випадках ви можете використовувати змінну `__int32` або `__int64`, згадану вище, залежно від того, який обсяг змінної пам'яті вам потрібен. Існують `__Int8`, `__int16`, `__int32` та `__int64`, які доступні лише принаймні для версії 7 (Microsoft Visual Studio 2003.NET) на компіляторі `MSVC++`, але ви можете вибрати розмір типу даних.

2.4 Опис технологій та додатків для розробки програмного застосунку

Для розробки було використано наступне програмне забезпечення та веб-інструментарій:

- GitHub – використовувався для контролю версій розробленого програмного застосунку;
- CMake – використовувався для генерації файлів керування збірки;

- QT creator – використовувався як IDE для написання коду та візуалізації проекту;
- SQL Light – використовувався для баз даних, в які зберігається прогрес роботи програми.

2.4.1 Веб-сервіс для хостингу ІТ-проектів GitHub

GitHub - це найбільша служба хостингу та розробки ІТ-проектів.

Веб-сервіс базується на системі управління версіями Git і був розроблений GitHub, Inc. (раніше Логічний Awesome) Рубін на рейках та Ерланг. Послуга надає їм безкоштовний доступ до проектів з відкритим кодом та приватних малих проектів (починаючи з 2019 року) (включаючи SSL), а також різні платні тарифні плани для великих корпоративних проектів.

Девіз послуги - «Публічне кодування» можна перекласти українською як «Давайте напишемо код разом». На футболках зовсім інша фраза: "Я тебе люблю!" ("Відповідь!"). З одного боку, це відповідає англійському прокляттям і стосується неформальної атмосфери. З іншого боку, ці слова нагадують нам, що створювати нові форки за допомогою Git легко і безболісно - традиційно розробники з відкритим кодом негативно ставляться до побудови гілок, і однією особливістю GitHub є назва черги форків.

Талісман GitHub - восьмикутник, який, на відміну від багатьох помилкових уявлень, не має нічого спільного з короткометражним фільмом «Пригода Октокат», а був просто виявлений Томом Престоном-Вернером.

4 червня 2018 року Microsoft придбала GitHub за 7,5 мільярда доларів.

Я використовував сервіс GitHub для контролю та оновлення версій програмного продукту з добирання музичних композицій. Цей інструмент допоміг мені аналізувати помилки та контролювати видозміну проекту.

Можливості GitHub:

- Творці сайту називають GitHub "соціальною мережею для розробників".

- Окрім публікації коду, учасники можуть спілкуватися в чаті, коментувати правки один одного та стежити за новинами своїх друзів.
- Завдяки багатим можливостям Git програмісти можуть інтегрувати свої сховища - GitHub пропонує зручний для цього інтерфейс і може показати внесок кожного інвестора у вигляді дерева.
- Існують особисті сторінки для проектів, невеликі вікі та система відстеження помилок.
- Саме на цьому сайті ви можете переглядати файли проекту, виділяючи синтаксис для більшості мов програмування.
- Ви можете створювати особисті сховища, які бачите лише ви та обрані вами люди.

Раніше можливість створення приватних сховищ була платною.

- Ви можете додавати нові файли безпосередньо до свого сховища через веб-інтерфейс служби.
- Код проекту можна завантажити з сайту у вигляді звичайних архівів, не тільки через Git.
- На додаток до Git, служба підтримує пошук та редагування коду через SVN та Mercurial.
- На сайті є прохідний сервіс gist.github.com для швидкої публікації фрагментів коду.

Раніше проекти Ruby автоматично публікувались у сховищі RubyGems служби, але в жовтні 2009 року GitHub припинив роботу служби.

2.4.2 Кросплатформенна система автоматизації збірки CMake

CMake - це міжплатформна автоматизована система побудови проектів. Він лише створює Makefile, не будуючи його безпосередньо, а потім виконує утиліту Make.

CMake може перевірити наявність необхідних бібліотек, підключити їх та будувати проекти під різними компіляторами та операційними системами. У вас є багато коду та файл, що містить інформацію для CMake, і щоб зібрати цю

річ в іншому місці, ви можете запустити там CMake, і він зробить це сам по собі. Зручно, корисно, просто.

Починаючи з версії 2.4.2, CMake CPack включає автоматизовану систему створення програмного пакету та модуль CMake для роботи з нею. Система дозволяє створювати програмні пакети для менеджерів популярних пакетів (DEB, RPM, DMG), засоби встановлення програмного забезпечення (NSIS для Microsoft Windows), а також архіви (TGZ, TBZ2, ZIP, саморозпаковується TGZ).

2.4.3 IDE Qt creator

Qt Creator - це повноцінне інтегроване середовище розробки (IDE) для програм C ++, QML та JavaScript. Це універсальна крос-платформенна IDE, яка дозволяє користувачам масштабувати код вище та створювати підключені інструменти, програми та користувацькі інтерфейси.

Програмне забезпечення є єдиною складною консоллю, яка дозволяє розробникам створювати додатки для :

- Багатьох платформ;
- настільних систем;
- мобільних пристроїв, таких як iOS та Android.

Додаток включає редактор коду, який легко інтегрується з іншими інструментами:

- кодування;
- дизайн;
- поширення;
- тестування;
- моніторинг.

Простий та інтуїтивно зрозумілий інтерфейс Qt Creator робить процес навчання плавним.

Інші функції, які роблять програму швидкою та простою, включають:

- створення користувацького інтерфейсу;
- інструменти для профілювання;
- візуальне налагодження;

- інтегрований графічний інтерфейс;
- конструктор форми.

Можливості Qt creator IDE:

- редактор коду з автозаповненням та підсвічуванням синтаксису;
- перетягнення дизайну інтерфейсу користувача;
- інструменти для профілювання;
- візуальне налагодження;
- вбудований графічний інтерфейс;
- дизайнер форм;
- управління проектами;
- Qt Assistant.

Переваги:

- Редактор коду. Ще однією особливістю є редактор вихідного коду. Ця функція супроводжується функціями автоматичного заповнення та підсвічування синтаксису. Крім того, Qt Creator підтримує розробку декларативного інтерфейсу користувача (UI) за допомогою модуля Qt Quick.
- Відповідність. Однією з хороших речей Qt Creator є сумісність цих платформ. Користувачі можуть створити додаток у Windows, скопіювати його на Linux або Mac та відкрити з локального Qt Creator. Це дозволить вам отримати максимум від цього і змусити це працювати так, як можуть уявити початківці та досвідчені розробники.
- Розробка та впровадження. Програма пропонує комплексний та інтуїтивно зрозумілий API. Як результат, перехресна компіляція дозволяє пришвидшити розробку, створення прототипів та реалізацію проекту.
- Надійна і стабільна платформа. Qt Creator пропонує швидку та безпроблемну інтеграцію апаратного забезпечення з повною

оптимізацією та без додаткових витрат. Крім того, ви маєте повний контроль над своїм кодом.

- Індивідуальний UX. Це програмне забезпечення дозволяє налаштувати та створити сучасний інтерфейс, надаючи кінцевому користувачеві природний досвід користування у всіх сферах. Qt Creator пропонує створити масштабовані інтерфейси для різних розмірів екрану.
- Масштабування. Програма забезпечує оптимізацію апаратних ресурсів за допомогою C ++, QML або інших мов програмування. Ви також можете зосередити увагу на менших пристроях, збільшуючи та зменшуючи екран на більшому екрані.
- Легко. Програмне забезпечення є гнучким і простим у використанні, а також має інструменти та функції для спрощення процесу розробки. Крім того, програмне забезпечення платформи пропонує бібліотеки та API. Qt Creator дозволяє вам змінювати вимоги, оскільки ви впевнені у завтрашньому дні.
- В результаті користувацький інтерфейс забезпечує відчуття та зовнішній вигляд робочого столу. Крім того, легко змінити та налаштувати стиль інтерфейсу користувача.
- Інтуїтивно зрозумілий користувацький інтерфейс. Дизайн, сценарії та програмування цієї програми дозволяють створювати гнучкі користувацькі інтерфейси. Крім того, ви можете вибрати найкращий метод проектування.

Недоліки:

- Відключення. Користувачі незадоволені несподіваним вимкненням. Таким чином, це явище негативно впливає на робочі процеси та продуктивність праці.
- Розташування пошукової системи. Користувач витрачає багато часу на пошук цього інструменту.

- Бібліотека рідко оновлюється, тому користувачі обмежуються наявними матеріалами та ресурсами. Qt Creator повинен регулярно оновлювати бібліотеку, щоб покращити робочий процес користувача.

2.4.4 Система управління базами даних SQLite

SQLite - це вбудована бібліотека, яка реалізує автономний, безсерверний, транзакційний механізм СУБД SQL. Це означає, що база даних не потребує налаштування у вашій системі.

SQLite не є самостійним процесом, як інші бази даних. Ви можете прив'язати її статично або динамічно відповідно до вашої програми. SQLite отримує файли безпосередньо зі свого сховища.

Особливості SQLite:

- SQLite не вимагає окремого сервера або системного процесу для запуску (немає сервера).
- SQLite поставляється з нульовою конфігурацією, що означає, що конфігурація та адміністрування не потрібні.
- Вся база даних SQLite зберігається на міжплатформенному диску.
- SQLite дуже малий і компактний, повністю налаштований менше 400 КБ або менше 250 КБ з видаленими додатковими функціями.
- SQLite є автономним і тому не має зовнішніх залежностей.
- Транзакції SQLite повністю сумісні з ACCL, забезпечуючи безпечний доступ до багатьох процесів або потоків.
- SQLite підтримує більшість функцій мови запитів, знайдених у стандарті SQL92 (SQL2).
- SQLite написаний на ANSI-C і забезпечує простий у використанні API.
- SQLite доступний в системах UNIX (Linux, Mac OS-X, Android, iOS) та Windows (Win32, WinCE, WinRT).

2.5 Реалізація програмного застосунку

Для реалізації програмного застосунку були використані наступні інструменти:

- CMake - сімейство інструментів, розроблених для створення, тестування та пакування програмного забезпечення з відкритим кодом.
- Qt - безкоштовний набір інструментів віджетів з відкритим кодом для створення графічних користувацьких інтерфейсів, а також крос-платформенних додатків, що працюють на різних програмних та апаратних платформах.
- TagLib - бібліотека для читання та редагування мета-даних декількох популярних аудіо-форматів.
- Мова програмування C++ - компільована строго типізована мова програмування загального призначення. Підтримує різні парадигми програмування: процедурну, узагальнену, функціональну; найбільшу увагу приділено підтримці об'єктно-орієнтованого програмування.

Велика частина функціоналу реалізується за допомогою бібліотеки Qt5, яка містить у собі значний функціонал та кількість інструментів.

2.6 Опис функціоналу

Проаналізувавши можливості декількох музичних плеєрів, я реалізував у своєму програмному застосунку наступний функціонал:

- підтримує кілька популярних форматів аудіо-файлів(mp3, mp4, wav);
- підтримка таких стандартних опцій:
 - відтворення, пауза, зупинка, перемотування вперед, перемотування назад, наступний та попередній трек.
 - регулювати гучність звуку;
 - автоматично відтворити наступну пісню в черзі;
 - створювати, читати, оновлювати та видаляти особисті списки відтворення;
 - імпорт, експорт списків відтворення у стандартному форматі списку відтворення m3u;

- додавати та вилучати пісні з черги;
- цикл (повторення) однієї пісні або черги;
- перемішати пісні, перелічені в черзі;
- сортувати пісні, перелічені в черзі;
- редагувати обкладинки пісень за допомогою зображень на комп'ютері;
- відображати та редагувати аудіо-тегів;
- можливість завантажувати обкладинки альбомів через Інтернет;
- відображення та редагування тексту пісень;
- підтримка автоматичного вимкнення програми через деякий час;
- підтримка радіо;
- додавати та видаляти з комп'ютера картинки для списків відтворення.

На рисунку 2.4 можна побачити головну сторінку програмного застосунку.

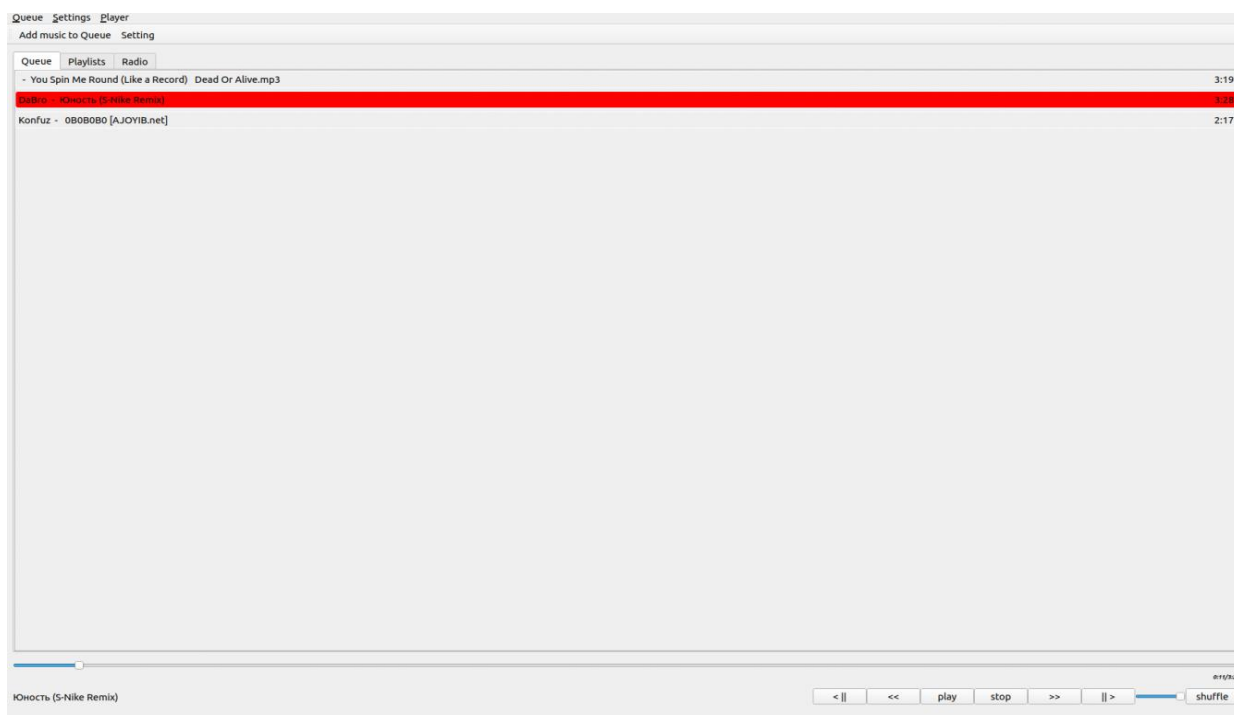


Рисунок 2.4 – Головна сторінка програмного застосунку

Висновки

У цьому розділі ми проаналізували інструменти, які у подальшому будуть використані для розробки програмного застосунку з добирання музичних композицій. Також було обрано мову програмування C++ для створення проекту.

Для розробки було обрано та описано архітектуру програмного застосунку. Вона являє собою звичайну клієнтську архітектуру.

Було обрано наступні інструменти для розробки: SQLite, TaffLib, CMake, Qt.

Виконавши аналіз вхідної інформації, стало відомо цільову аудиторію застосунку – школярі, студенти та люди віком 30-60 років. Зважаючи на це, створено прототип та дизайн програмного застосунку, який має зручний та простий інтерфейс.

Для покращення функціоналу додатку, було проаналізовано декілька музичних застосунків таких як: Windows Media Player та VLC.

Для перевірки виконаних кроків було проведено тестування веб-застосунку за допомогою перевірки повного функціоналу програмного застосунку.

Розробка програмного застосунку з власним дизайном потребує якісної підготовки та чіткого плану з виконання всіх етапів розробки, які стосуються створення прототипної версії.

РОЗДІЛ 3 АНАЛІЗ ФУНКЦІОНАЛУ, ОПИС ІНСТАЛЯЦІЇ. ІНСТРУКЦІЯ З ЕКСПЛУАТАЦІЇ

У розділі 3 ми проведемо аналіз та опис основного функціоналу програмного застосунку з добирання музичних композицій. Запуск та перевірка програмного застосунку відбудеться на операційній системі Linux.

Також проведемо розбір адміністрування програмного застосунку, що дозволить оцінити доступність та простоту його використання для користувача, що не має досвіду роботи в таких додатках.

3.1 Процес запуску програмного застосунку для відтворення музичних композицій

Так як основною операційною для якої розроблялося дане програмне забезпечення є Linux, то запуск програми буде відбуватися саме на ній.

Для початку треба відкрити основний каталог, його можна побачити на рисунку 3.1 з проектом. Далі, натиснувши правою кнопкою мишки, відкрити термінал з цього каталогу. Цю дію ви можете побачити на рисунку 3.2.

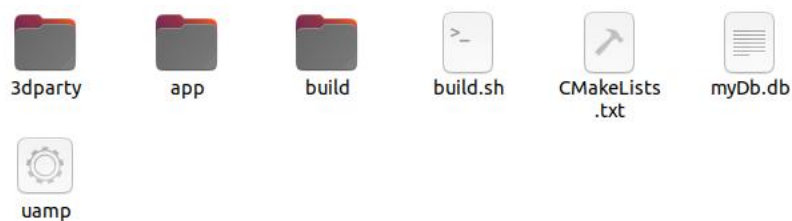


Рисунок 3.1 - Кореневий каталог проекту

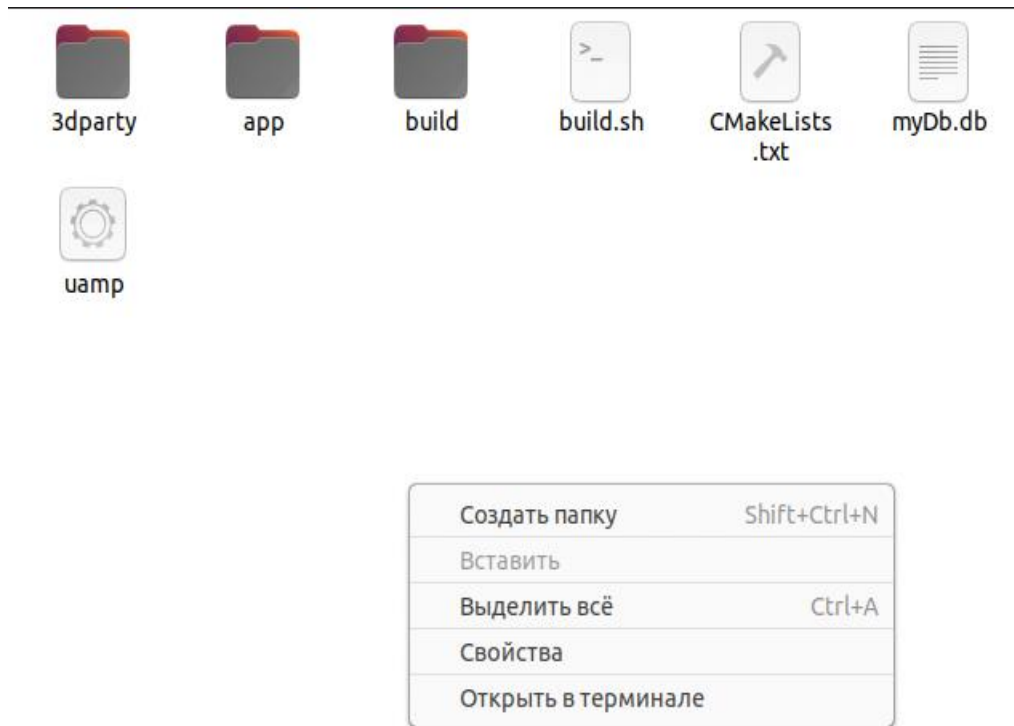


Рисунок 3.2 - Відкриття проекту в терміналі

Після відкриття терміналу потрібно зібрати проект у одну цільну програму. Для цього необхідний повний перелік бібліотек та програм, які були використані для розробки програмного застосунку: Qt, CMake, Clang. Для операційної системи MacOS також потрібен наступний інструмент – Homebrew. Після інсталяції всіх пакетів, можна збирати проект. Для цього потрібно у терміналі кореневого каталогу ввести наступну команду: `./build.sh`. Після виконання команди, у кореновому каталозі з'явиться файл "UAMP". За допомогою нього ми і будемо запускати наш програмний застосунок. Процес збирання проекту показано на рисунку 3.3. Файл для запуску програмного застосунку зображено на рисунку 3.4.

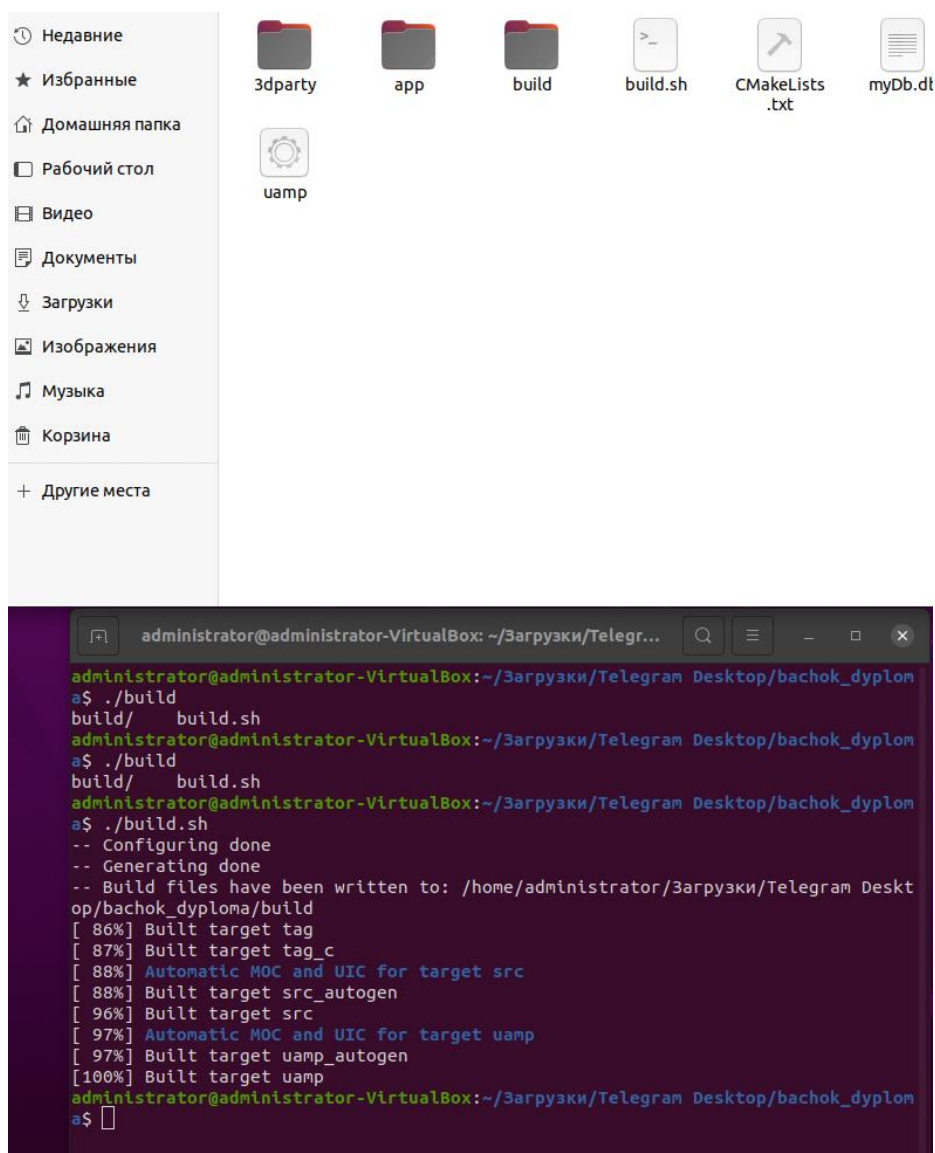


Рисунок 3.3 - Процес збірки проекту



Рисунок 3.4 - Файл для запуску проекту

3.2 Опис та огляд основного функціоналу програмного застосунку

Цей програмний застосунок має широкий та актуальний функціонал, який є досить простим у використанні та має досить привабливий інтерфейс.

Користувачу доступна велика кількість необхідних у використанні функцій, таких як:

- можливість відтворювати та зупиняти музику;

- зміна позиції відтворення треку;
- перехід до попереднього або наступного треку;
- зміна гучності звуку;
- автоматичне відтворення наступної пісні в черзі;
- переглядати та оновлювати бібліотеки зі списками музики;
- імпорт, експорт бібліотек з музикою у стандартному форматі списку відтворення m3u;
- корегування черги відтворення композицій;
- зміна порядку відтворення треків;
- випадкове відтворення пісень;
- сортування пісень у черзі;
- редагування обкладинок пісень за допомогою зображень на комп'ютері;
- відображення та зміна тегів у треках;
- можливість завантажувати обкладинки альбомів через мережу Інтернет;
- відображення та редагування тексту пісень;
- підтримка автоматичного вимкнення програми через деякий час;
- радіо;
- додавання та видалення з комп'ютера картинки для списків відтворення.

3.2.1 Головне вікно програмного застосунку

Після запуску додатку з добирання музичних композицій ми зможемо побачити головну сторінку застосунку. Головне вікно має досить простий та приємний інтерфейс, що відображено на рисунку 3.5.

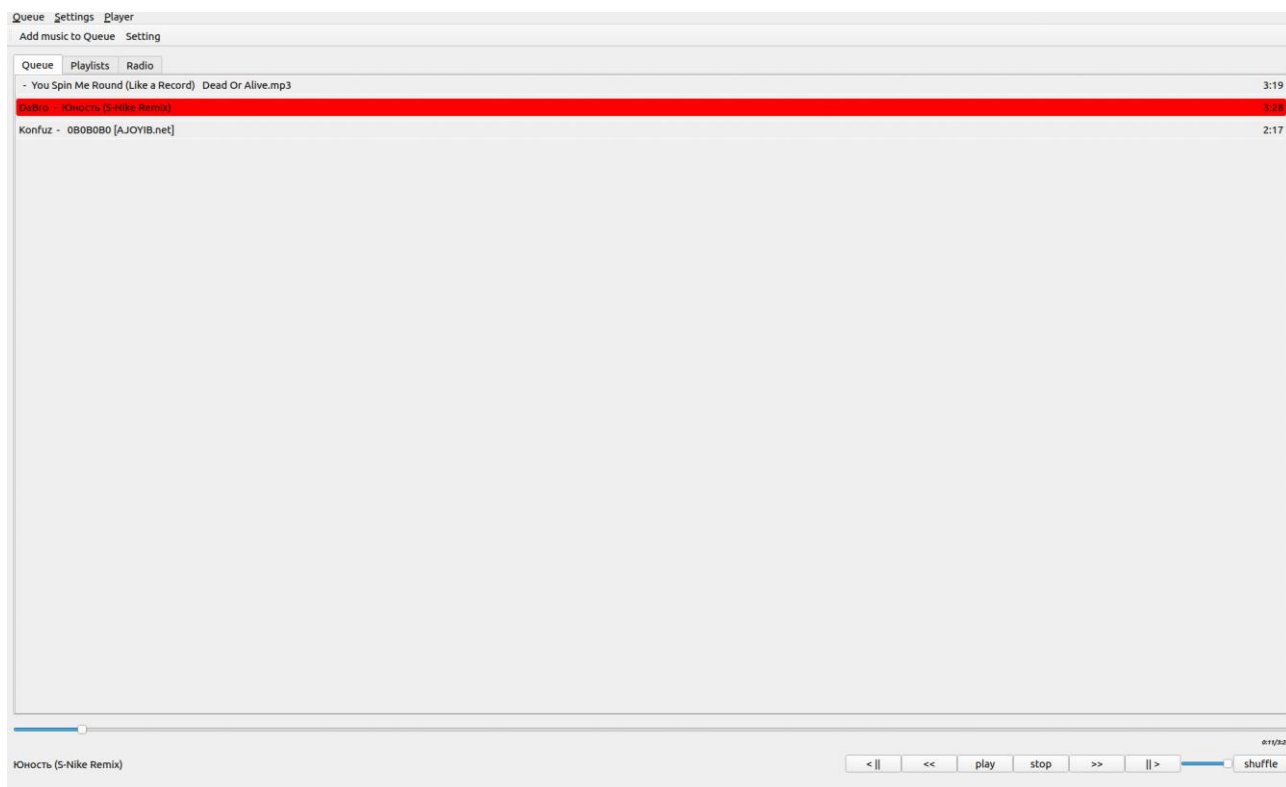


Рисунок 3.5 - Головне вікно програмного застосунку

З цієї сторінки є переходи до всього функціоналу додатку. Звідси можна додавати та видаляти треки, завантажувати бібліотеки з музикою, здійснювати налаштування альбомів, змінювати теги у музичних файлах.

3.2.2 Доступ до дискового сховища та завантаження музичних композицій

У програмний застосунок можна долучати лише файли, які є локально на вашому пристрої. У нього є доступ до файлової системи, він одразу розпізнає формати, які підтримує. Функціонал відображення музичних композицій у дисковому сховищі відображено на рисунку 3.5.

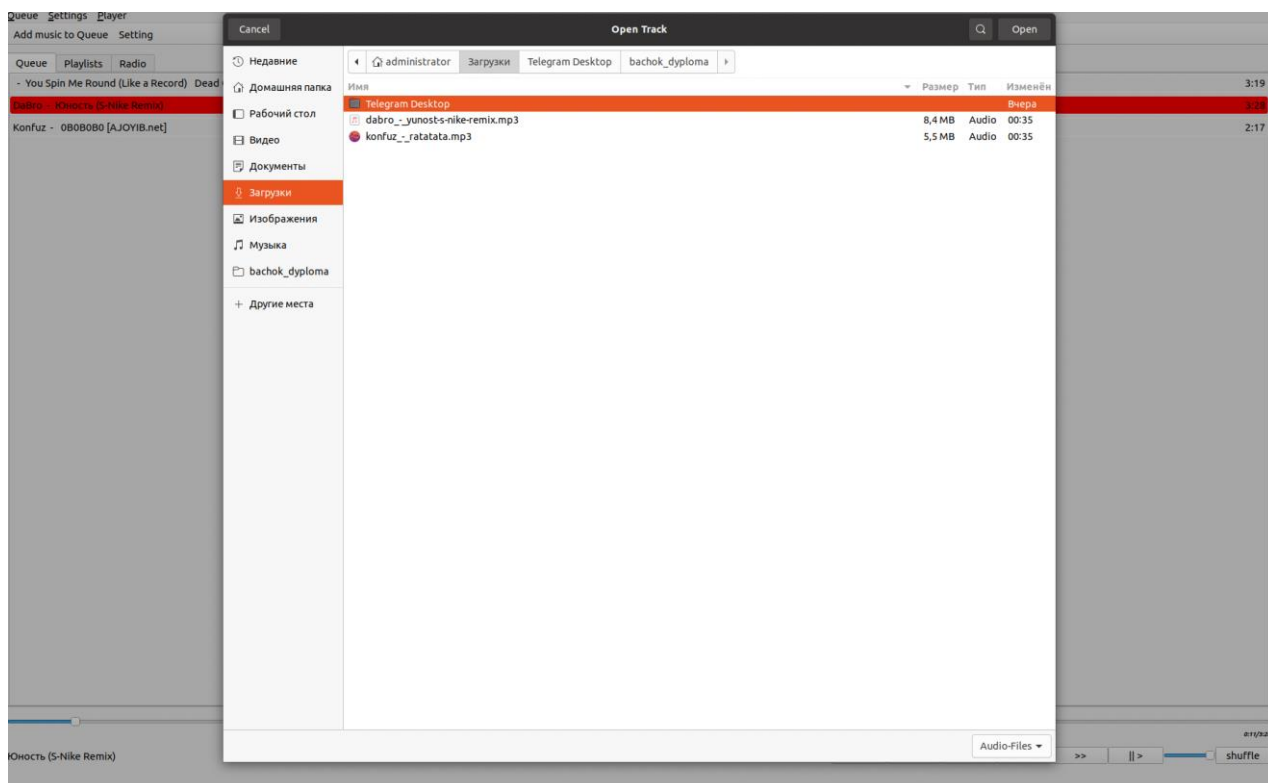


Рисунок 3.5 - Розпізнавання музичних композицій

Меню додавання файлів реалізовано досить зручно і містить у собі весь необхідний перелік дій, яких потребує користувач. Меню долучення нових композицій відображено на рисунку 3.6.

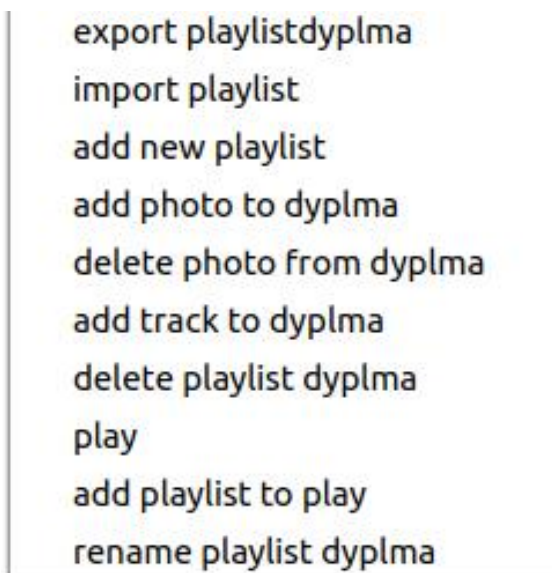


Рисунок 3.6 - Меню долучення нових бібліотек та окремих композицій

3.2.3 Функція радіо

У сьогоднішні радіо є досить важливою річчю, адже цільова аудиторія не завжди має бажання витратити час на пошук великої кількості музичних композицій та формування бібліотек з музикою. Ця функція реалізована у музичному додатку в окремому вікні. Радіо має чотири станції відтворення:

- Radio ROKS;
- Hit FM;
- Radio Jazz;
- Kiss FM.

Вікно знаходиться на панелі керування музичним додатком, воно зображено на рисунку 3.7.

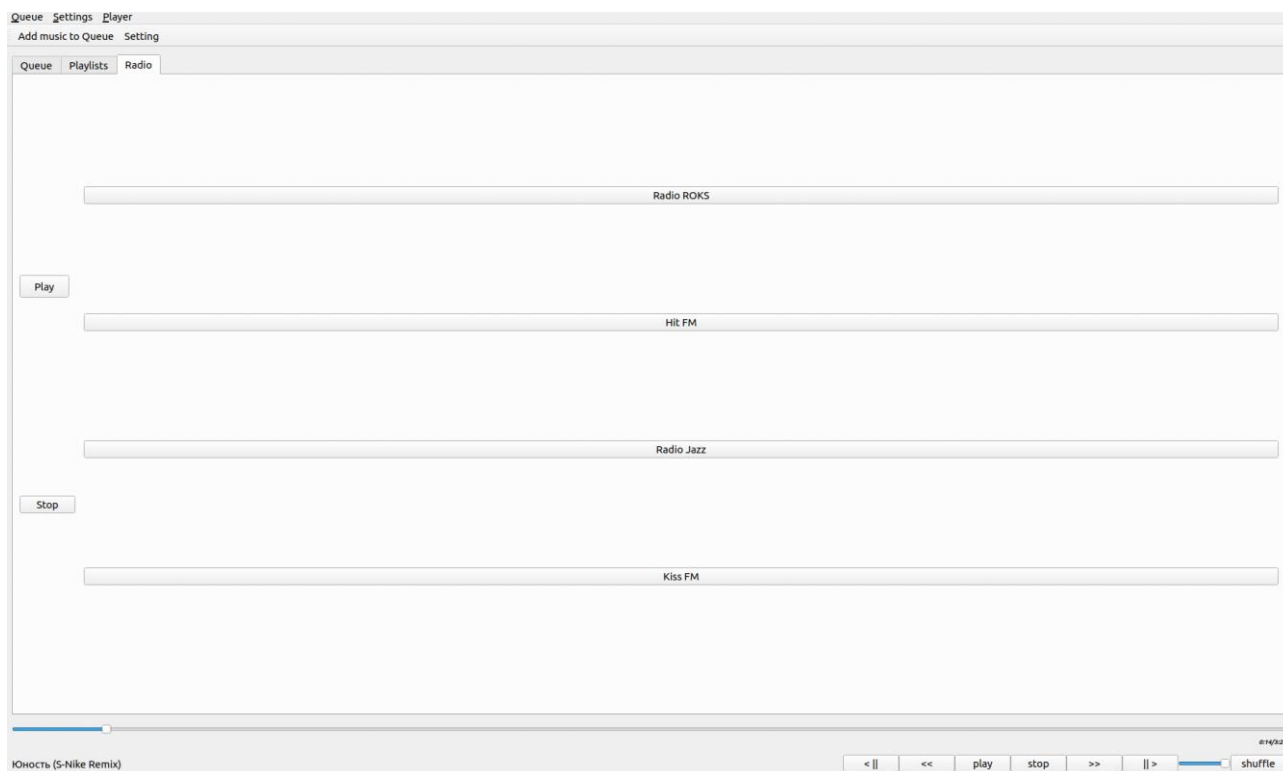


Рисунок 3.7 - Функція радіо

Тож не обов'язково витратити свій час на пошук музичних композицій, вони одразу є у додатку. Таким функціоналом не може похизуватись жоден із застосунків відомих мені.

3.2.4 Можливості відтворення та налаштування тегів у композиціях

Для керування відтворенням музики (перемотки, налаштування черги, видалення треків з черги та налаштування звуку) було створено зручну панель

інструментів, на якій є функціональні кнопки із зображенням символів для розуміння дій цих клавiш. Панель інструментів відображено на рисунку 3.8.

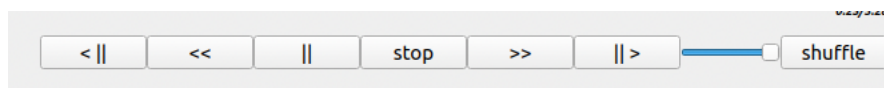


Рисунок 3.8 - Панель інструментів

У кожній музичній композиції є свої теги: назва альбому, картинка треку, текст та назва пісні, виконавач та інші. Усі ці теги можна змінювати у нашому додатку. Такого функціоналу немає у стандартних програвачах, що дає певну унікальність програмному застосунку, розробленому мною. Налаштування тегів зображено на рисунку 3.9.

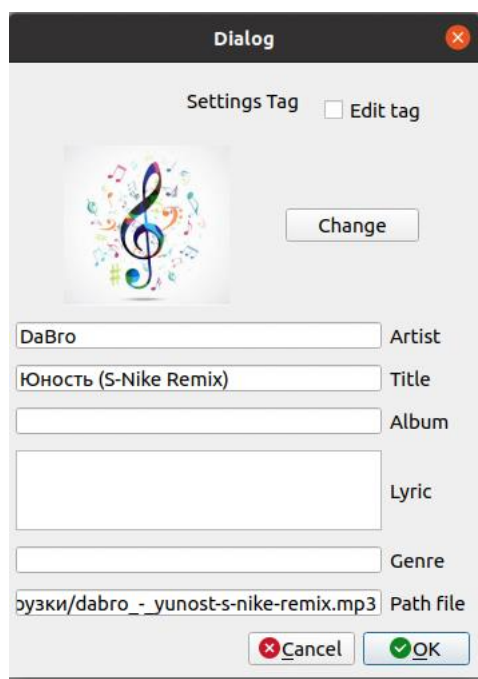


Рисунок 3.9 - Налаштування тегів

3.2.5 Налаштування черги музичних треків

У цьому додатку є різні операції над чергою відтворення музичних композицій:

- сортування композицій;
- повтор композиції;
- довільне відтворення.

Меню налаштувань зображене на рисунку 3.10.

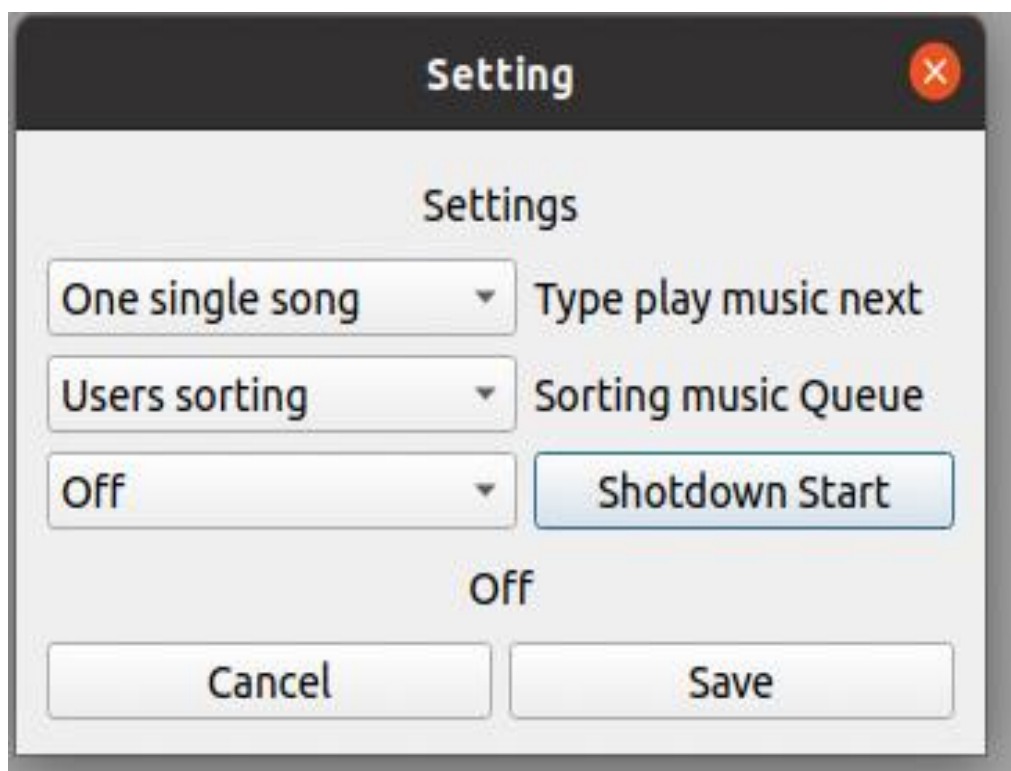


Рисунок 3.10 - Меню налаштувань

Висновки

У розділі 3 ми провели аналіз та опис основного функціоналу програмного застосунку з добирання музичних композицій. Запуск та перевірка програмного застосунку відбулася на операційній системі Linux.

Також було проведено розбір адміністрування програмного застосунку, що дозволить оцінити доступність та простоту його використання для користувача, що не має досвіду роботи в таких додатках.

Також були наведені переваги цього програмного застосунку, а також недоліки аналогів.

ВИСНОВОК

У цій дипломній роботі було проаналізовано та розглянуто різноманітні способи і технології розробки програмного застосунку. Було описано основні переваги над альтернативними засобами відтворення музичних композицій.

Досліджено, які технології є актуальними для створення програмного застосунку для різного функціоналу та потреби. Також були наведені основні інструменти для розробки та реалізації цього проекту.

Здійснений аналіз основних засобів розробки, таких як: Qt creator, CMake, Clang, SQLite.

Було розроблено застосунок з рядом корисного функціоналу, який є досить простим і актуальним у використанні. Якщо порівнювати даний продукт з альтернативами, то він, звісно, має велику кількість переваг, написаних вище.

Робота виконана з використанням фреймворку Qt, а також мови програмування C++. База даних реалізована на основі SQLite.

Після реалізації проекту було проведено тестування додатку за допомогою повної перевірки всіх функцій. Програмне забезпечення тестувалося на трьох операційних системах: MacOS, Windows 10 та Linux(Ubuntu).

Усі модулі та блоки програмного застосунку відображались коректно та повністю функціонували.

ПЕРЕЛІК ПОСИЛАНЬ

1. История музыкальных плееров: веб-сайт. URL: <https://trashbox.ru/link/history-of-music-players> (дата звернення 01.05.2021);
2. Файл формата MP3, что это? : веб сайт URL: <https://filesreview.com/ru/info/mp3> (дата звернення 03.05.2021);
3. Что такое MP3? : веб-сайт. URL: <https://ru.macblurayplayer.com/audio/what-is-mp3.htm> (дата звернення 08.05.2021);
4. Почему я пользуюсь Qt : веб-сайт. URL: <https://habr.com/ru/post/221523/> (дата звернення 08.05.2021);
5. Qt: веб-сайт. URL: <https://ru.wikipedia.org/wiki/Qt> (дата звернення 15.05.2021);
6. C++ : веб-сайт. URL: <http://progopedia.ru/language/c-plus-plus/> (дата звернення 20.05.2021);
7. MP4(MPAG-4 Video Stream): веб-сайт. URL: <https://www.online-convert.com/ru/file-format/mp4> (дата звернення 21.05.2021);
8. Что такое MP4? Разница между MP3 и MP4 : веб-сайт. URL: <http://helpexe.ru/tehnologija-objasnila/chto-takoe-mp4-raznica-mezhdu-mp3-i-mp4-2> (дата звернення 03.04.2021);
9. MP3, AAC, WAV. Все форматы аудиофайлов: веб-сайт. URL: <https://www.audiomania.ru/content/art-7314.html> (дата звернення 10.04.2021);
10. Що таке операційна система Windows. Операційні системи: веб-сайт. URL: <http://ipkey.com.ua/faq/956-os-windows.html> (дата звернення 13.04.2021);
11. Операційні системи MacOS: веб-сайт. URL: <http://www.linux-info.ru/macosx/opisanie.html> (дата звернення 01.05.2021);
12. Операційні системи Linux під різні задачі: веб-сайт. URL: <https://habr.com/ru/company/ua-hosting/blog/342826/> (дата звернення 23.04.2021);
13. Проигрыватель Windows Media: веб-сайт. URL: <https://open-file.ru/programs/microsoft-windows-media-player> (дата звернення 11.05.2021);
14. Знайомство з Virtualbox. Огляд програми: веб-сайт. URL: <https://techlist.top/knowning-virtualbox/> (дата звернення 17.05.2021);

15. Обзор Windows Media Player 12: веб-сайт. URL: <https://windata.ru/windows-7/programmy-7/obzor-windows-media-player-12/> (дата звернення 28.04.2021);
16. Полное руководство по CMake. Система сборки: веб-сайт. URL: <https://habr.com/ru/post/432096/> (дата звернення 25.04.2021);
17. VLC медіа плеєр. Загальна інформація: веб-сайт. URL: <https://remontka.pro/vlc-media-player/> (дата звернення 02.04.2021);
18. Задача знаходження інформаційного боку: веб-сайт. URL: https://ela.kpi.ua/bitstream/123456789/37402/1/Yakubenko_bakalavr.pdf;
19. Організація баз даних: веб-сайт. URL: <https://dspace.chmnu.edu.ua/jspui/bitstream/123456789/305/1/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F%20%D0%B1%D0%B0%D0%B7%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85.%20%D0%A6%D0%B8%D0%BA%D0%BB%20%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82.%20%D1%80%D0%BE%D0%B1%D1%96%D1%82.%20%D0%92%D0%B8%D0%BF.%20269.pdf> (дата звернення 08.04.2021);;
20. GitHub: веб-сайт. URL: <https://uk.wikipedia.org/wiki/GitHub> (дата звернення 11.04.2021);;
21. Qt: веб-сайт. URL: <https://ru.wikipedia.org/wiki/Qt> (дата звернення 21.05.2021);;
22. Фреймворк Qt: веб-сайт. URL: <http://wikipedia.ru.nina.az/wiki/Qt> (дата звернення 23.04.2021);.

ДОДАТКИ

Додаток А. Код реалізації радіо

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>radio</class>
  <widget class="QWidget" name="radio">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>848</width>
        <height>158</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>Form</string>
    </property>
    <layout class="QHBoxLayout" name="horizontalLayout">
      <item>
        <layout class="QVBoxLayout" name="verticalLayout">
          <item>
            <widget class="QPushButton" name="Play">
              <property name="minimumSize">
                <size>
                  <width>0</width>
                  <height>32</height>
                </size>
              </property>
              <property name="maximumSize">
                <size>
                  <width>71</width>
                  <height>32</height>
                </size>
              </property>
            </widget>
          </item>
        </layout>
      </item>
    </layout>
  </widget>
</ui>

```

```

</property>
<property name="text">
  <string>Play</string>
</property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="Stop">
    <property name="text">
      <string>Stop</string>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <widget class="QSplitter" name="splitter_2">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <widget class="QSplitter" name="splitter">
      <property name="orientation">
        <enum>Qt::Vertical</enum>
      </property>
    </widget>
  </widget>
</item>
<item>
  <layout class="QVBoxLayout" name="verticalLayout_2">
    <item>
      <widget class="QPushButton" name="RadioROKS">
        <property name="text">
          <string>Radio ROKS</string>
        </property>

```



```

</widget>
</item>
<item>
  <widget class="QPushButton" name="HitFM">
    <property name="text">
      <string>Hit FM</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="RadioJazz">
    <property name="text">
      <string>Radio Jazz</string>
    </property>
  </widget>
</item>
<item>
  <widget class="QPushButton" name="KissFM">
    <property name="text">
      <string>Kiss FM</string>
    </property>
  </widget>
</item>
<item>
  <spacer name="horizontalSpacer">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>40</width>
        <height>20</height>
      </size>
    </property>
  </item>

```

```
</spacer>  
</item>  
</layout>  
</item>  
</layout>  
</widget>  
<resources/>  
<connections/>  
</ui>
```

Додаток Б. Реалізація головної сторінки

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>generalWindow</class>
  <widget class="QMainWindow" name="generalWindow">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>454</width>
        <height>300</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>generalWindow</string>
    </property>
    <widget class="QWidget" name="centralWidget">
      <layout class="QVBoxLayout" name="verticalLayout_2">
        <item>
          <layout class="QVBoxLayout" name="generalLayout">
            <item>
              <layout class="QHBoxLayout" name="horizontalLayout"/>
            </item>
          </layout>
        </item>
      </layout>
    </widget>
    <widget class="QMenuBar" name="menuBar">
      <property name="geometry">
        <rect>
          <x>0</x>
          <y>0</y>
          <width>454</width>

```

```

    <height>22</height>
</rect>
</property>
<widget class="QMenu" name="menu_Queue">
    <property name="title">
        <string>&Queue</string>
    </property>
    <addaction name="action_Add_music_to_Queue"/>
</widget>
<widget class="QMenu" name="menu_Settings">
    <property name="title">
        <string>&Settings</string>
    </property>
    <addaction name="action_Setting"/>
</widget>
<widget class="QMenu" name="menu_Player">
    <property name="title">
        <string>&Player</string>
    </property>
    <addaction name="action_Stop"/>
    <addaction name="action_Next"/>
    <addaction name="action_Previous"/>
    <addaction name="action_Rewind"/>
    <addaction name="action_Fast_forwsrd"/>
    <addaction name="action_Play_Pause"/>
</widget>
<addaction name="menu_Queue"/>
<addaction name="menu_Settings"/>
<addaction name="menu_Player"/>
</widget>
<widget class="QToolBar" name="mainToolBar">
    <attribute name="toolBarArea">
        <enum>TopToolBarArea</enum>
    </attribute>

```

```

<attribute name="toolBarBreak">
  <bool>false</bool>
</attribute>
<addaction name="action_Add_music_to_Queue"/>
<addaction name="action_Setting"/>
</widget>
<widget class="QStatusBar" name="statusBar"/>
<action name="action_Add_music_to_Queue">
  <property name="text">
    <string>&Add music to Queue</string>
  </property>
</action>
<action name="action_Setting">
  <property name="text">
    <string>&Setting</string>
  </property>
</action>
<action name="action_Stop">
  <property name="text">
    <string>&Stop</string>
  </property>
</action>
<action name="action_Next">
  <property name="text">
    <string>&Next</string>
  </property>
  <property name="shortcut">
    <string>Down</string>
  </property>
</action>
<action name="action_Previous">
  <property name="text">
    <string>&Previous</string>
  </property>

```

```

<property name="shortcut">
  <string>Up</string>
</property>
</action>
<action name="action_Rewind">
  <property name="text">
    <string>&Rarr;Rewind</string>
  </property>
  <property name="shortcut">
    <string>Left</string>
  </property>
</action>
<action name="action_Fast_forwsrd">
  <property name="text">
    <string>&Rarr;Fast forwsrd</string>
  </property>
  <property name="shortcut">
    <string>Right</string>
  </property>
</action>
<action name="action_Play_Pause">
  <property name="text">
    <string>&Rarr;Play/Pause</string>
  </property>
  <property name="shortcut">
    <string>Space</string>
  </property>
</action>
</widget>
<layoutdefault spacing="6" margin="11"/>
<resources/>
<connections/>
</ui>

```

Додаток В. Реалізація налаштувань черги музичних композицій

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>WindowSetting</class>
  <widget class="QDialog" name="WindowSetting">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>350</width>
        <height>200</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>350</width>
        <height>200</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>350</width>
        <height>200</height>
      </size>
    </property>
    <property name="windowTitle">
      <string>Setting</string>
    </property>
    <layout class="QVBoxLayout" name="verticalLayout">
      <item>
        <widget class="QLabel" name="label_3">
          <property name="text">
            <string>Settings</string>

```

```

</property>
<property name="alignment">
  <set>Qt::AlignCenter</set>
</property>
</widget>
</item>
<item>
  <layout class="QFormLayout" name="formLayout">
    <item row="1" column="1">
      <widget class="QLabel" name="label">
        <property name="text">
          <string>Type play music next</string>
        </property>
      </widget>
    </item>
    <item row="2" column="0">
      <widget class="QComboBox" name="TypeSort">
        <property name="sizePolicy">
          <sizepolicy hsiptype="Expanding" vsizetype="Fixed">
            <horstretch>0</horstretch>
            <verstretch>0</verstretch>
          </sizepolicy>
        </property>
        <property name="minimumSize">
          <size>
            <width>170</width>
            <height>0</height>
          </size>
        </property>
        <property name="maximumSize">
          <size>
            <width>170</width>
            <height>16777215</height>
          </size>

```



```

</property>
<item>
  <property name="text">
    <string>Users sorting</string>
  </property>
</item>
<item>
  <property name="text">
    <string>Artist</string>
  </property>
</item>
<item>
  <property name="text">
    <string>Title</string>
  </property>
</item>
<item>
  <property name="text">
    <string>Album</string>
  </property>
</item>
<item>
  <property name="text">
    <string>Genre</string>
  </property>
</item>
</widget>
</item>
<item row="2" column="1">
  <widget class="QLabel" name="label_2">
    <property name="text">
      <string>Sorting music Queue</string>
    </property>
  </widget>
</item>

```

```

</item>
<item row="1" column="0">
  <widget class="QComboBox" name="TypePlay">
    <property name="minimumSize">
      <size>
        <width>170</width>
        <height>0</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>170</width>
        <height>16777215</height>
      </size>
    </property>
    <item>
      <property name="text">
        <string>One single song</string>
      </property>
    </item>
    <item>
      <property name="text">
        <string>Loop Queue</string>
      </property>
    </item>
    <item>
      <property name="text">
        <string>Repeat a single song</string>
      </property>
    </item>
    <item>
      <property name="text">
        <string>Random music</string>
      </property>
    </item>
  </widget>
</item>

```

```

</item>
</widget>
</item>
<item row="3" column="0">
  <widget class="QComboBox" name="Times">
    <property name="minimumSize">
      <size>
        <width>170</width>
        <height>0</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>16777215</width>
        <height>170</height>
      </size>
    </property>
    <item>
      <property name="text">
        <string>Off</string>
      </property>
    </item>
    <item>
      <property name="text">
        <string>1 minute</string>
      </property>
    </item>
    <item>
      <property name="text">
        <string>5 minutes</string>
      </property>
    </item>
    <item>
      <property name="text">

```

```

    <string>10 minutes</string>
  </property>
</item>
<item>
  <property name="text">
    <string>15 minutes</string>
  </property>
</item>
</widget>
</item>
<item row="3" column="1">
  <widget class="QPushButton" name="pushButtonStart">
    <property name="text">
      <string>Shutdown Start</string>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <widget class="QLabel" name="TimeToShutdown">
    <property name="text">
      <string>Off</string>
    </property>
    <property name="alignment">
      <set>Qt::AlignCenter</set>
    </property>
  </widget>
</item>
<item>
  <layout class="QHBoxLayout" name="horizontalLayout_3">
    <item>
      <widget class="QPushButton" name="pushButton_Cancel">
        <property name="text">

```

```
<string>Cancel</string>
</property>
</widget>
</item>
<item>
  <widget class="QPushButton" name="pushButton_Save">
    <property name="text">
      <string>Save</string>
    </property>
  </widget>
</item>
</layout>
</item>
</layout>
</widget>
<resources/>
<connections/>
</ui>
```