

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет _____ Магістерської та
_____ аспірантської підготовки

Кафедра інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Створення інтернет-радіомовлення на мові програмування
Liquidsoap

Виконав студент 2 року групи МК- 2
спеціальності 122 Комп'ютерні
науки та інформаційні технології

_____ Біляков Олександр Олександрович

Керівник д.х.н., професор
Кругляк Юрій Олексійович

Консультант _____

Рецензент _____

Одеса 2018

АНОТАЦІЯ

Дипломна робота на тему: «Створення інтернет-радіомовлення на мові програмування Liquidsoap», автор – студент групи МК2 Біляков Олександр Олександрович

Актуальність: актуальність дипломної роботи зумовило ріст популярності аудіо передач.

Мета: розробка радіостанції за допомогою мови liquidsoap.

Задачі: вивчити літературу, що стосується теми дослідження, розглянути існуючі аналоги, розробити радіостанцію liquidsoap і сайт для адміністрування.

Об'єкт дослідження: потокове мовлення у мережі.

Предмет: мережева радіостанція на мові liquidsoap.

Методи: системний аналіз поточкового мовлення, порівняння аналогів, розробка діючого проекту.

Результат: створення масштабованої моделі діючого проекту для запуску в учбовому закладі, функціональність якої можна розширювати.

Рекомендації: радіостанцію слід використовувати для створення поточкового мовлення організації.

Робота включає: 75 сторінки, 20 рисунків, 4 додатки, використаних джерел – 16.

Ключові слова: LIQUIDSOAP, ІНТЕРНЕТ-РАДІО, ПОТОКОВЕ МОВЛЕННЯ, ICECAST, ПОДКАСТ.

SUMMARY

Graduation work on the theme: "Creating Internet broadcasting in the programming language Liquidsoap", author –student of the group MK2 Alexander Bilyakov

Actuality: the relevance of the thesis led to the growth of popularity of audio streaming.

Objective: to develop a radio station using liquidsoap programming language.

Objectives: to study literature related to the topic of research, to consider existing analogues, to develop a liquidsoap radio station and a website for administration.

Object of research: streaming broadcast on the network.

Subject: network radio station in the liquidsoap programming language.

Methods: system analysis of streaming speech, comparison of analogues, development of the current project.

Result: creating a scalable model of an active project for launch in an institution whose functionality can be expanded.

Recommendations: The radio station should be used to create streaming broadcasting organization.

The work includes 75 pages, 20 images, 4 additions, used sources - 16.

Keywords: LIQUIDSOAP, INTERNET RADIO, STREAMING, ICECAST, PODCAST.

ЗМІСТ

| | |
|----------------------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ | 8 |
| ВСТУП | 10 |
| 1 ЗАГАЛЬНИЙ РОЗДІЛ | 12 |
| 1.1 Аналітичний огляд існуючих рішень | 12 |
| 1.1.1 Основні відомості по Інтернет-радіо..... | 12 |
| 1.1.2 Історія Інтернет-радіо | 13 |
| 1.1.3 Програмне забезпечення по організації серверу..... | 14 |
| 1.1.4 Програмне забезпечення для генерації аудіо потоку | 16 |
| 2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ | 18 |
| 2.1 Технічне завдання | 18 |
| 2.1.1 Призначення розробки..... | 18 |
| 2.1.2 Вимоги до програмної документації | 18 |
| 2.1.3 Стадії й етапи розробки програмного проекту | 19 |
| 2.1.4 Порядок контролю і прийому проекту..... | 20 |
| 2.2 Опис протоколу Icescast | 20 |
| 2.3 Основні інструменти при трансляванні | 22 |
| 2.3.1 Джерело-сервер..... | 22 |
| 2.3.2 Клієнт-сервер | 23 |
| 2.3.3 Установка з'єднання з сервером, передача mp3-даних..... | 24 |
| 2.3.4 Топологія мережі для роботи радіо | 25 |
| 2.3.5 Управління джерелом аудіопотоку за допомогою веб-сторінки..... | 26 |
| 2.4 Алгоритм програми..... | 29 |
| 2.4.1 Взаємодія користувача з системою | 29 |
| 2.4.2 Логіка роботи програми..... | 31 |
| 2.5 Тестування та налагодження системи..... | 33 |
| 2.6 Інструкція з експлуатації системи | 35 |
| 3 ОПИС ПРОЕКТНОГО І ТЕХНІЧНОГО РІШЕННЯ | 38 |
| 3.1 Логічне проектування | 38 |
| 3.1.1 Сервер Icescast2..... | 38 |
| 3.1.2 Джерело аудіопотоку Liquidsoap | 40 |
| 3.2 Зовнішнє проектування | 43 |
| 3.3 Трансляція з мікрофону | 45 |
| 3.4 Сайт радіостанції..... | 46 |
| 3.4.1 Налаштування веб-серверу..... | 46 |
| 3.4.2 Створення веб-сторінки для замовлення музики..... | 49 |

| | |
|-----------------------------------------|----|
| 4 ПРОСУВАННЯ САЙТУ | 52 |
| 5 УПРАВЛІННЯ ПРОЕКТАМИ | 62 |
| ВИСНОВКИ | 66 |
| ПЕРЕЛІК ПОСИЛАНЬ | 67 |
| ДОДАТОК А КОНФІГУРАЦІЯ LIQUIDSOAP | 69 |
| ДОДАТОК Б КОД СТОРІНКИ ЗАКАЗІВ | 71 |
| ДОДАТОК В ДІАГРАМА USE-CASE | 74 |
| ДОДАТОК Г ДІАГРАМА ДІЯЛЬНОСТІ | 75 |

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

AAC – Advanced Audio Coding
ACK SN – Acknowledgment Sequence Number
API – Application programming interface
DVB – Digital Video Broadcasting
GPL – General Public License
ISN – Initial Sequence Number (початковий порядковий номер)
LAMP – Linux Apache MySQL PHP
LAN – Local Area Network
MAST – Microsoft Application Stress Tool
MMS – Multimedia Messaging Service
NAT – Network Address Translation
NSV – Nullsoft Streaming Video
RTP – Real-time Transport Protocol
RTSP – Real Time Streaming Protocol
SDP – Session Description Protocol
TCP – Transmission Control Protocol
UDP – User Datagram Protocol
UML – Unified Modeling Language
VPN – Virtual Private Network
WAN – Wide Area Network
WCGI – Web Server Gateway Interface

Умовні позначення

кбіт/с – кількість Кб, переданих за секунду

Терміни

байт-орієнтований (byte-stream) протокол – комунікаційний протокол що забезпечує передачу даних у вигляді послідовностей байтів (byte) або символів
бітрейт – кількість даних в одиницю часу m для передачі аудіо потоку
контент (від англ. Content – «зміст») – інформаційно значущий вміст веб-сторінки

плейлист – упорядкований за тими чи іншими правилами, список аудіо, або відео файлів

трансляція – передача інформації на дальні відстані

ActionScript – це скриптова мова програмування

Ampache – вільне програмне забезпечення веб-аудіо серверу

edcast Standalone – програма для передачі звуку з мікрофону до серверу

Icecast – потоковий мультимедіа сервер

IceS – клієнт з вихідним кодом для потокового сервера

MP3 – формат файлу для зберігання аудіо-інформації

mp3PRO – алгоритм стиснення аудіо

Multicast – протоколи відсилають одну порцію даних цілій групі клієнтів

OCaml – одна із реалізацій мови програмування Caml

Ogg/Vorbis – вільний формат стиснення звуку

Opus – вільний та відкритий звуковий кодек

PHP – скриптова мова програмування

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією

RealAudio – запатентований аудіо формат, розроблений RealNetworks

Rich Internet application – це застосунок, доступний через Інтернет, і насичений функціональністю традиційною прикладних програм

Shoutcast – кроссплатформне безкоштовне серверне ПЗ компанії Nullsoft

Theora – відеокодек, розроблений Фондом Xiph.Org

Unicast – протоколи відправляють окрему копію даних кожному клієнту

WebM – вільний відкритий мультимедійний контейнер призначений для доставки аудіовізуальних даних в мережі Інтернет у межах стандарту HTML5.

ВСТУП

Сучасний світ неможливо уявити без постійного обміну інформацією, яка існує завдяки комп'ютеризації.

З моменту створення комп'ютерні мережі постійно вдосконалювались і зараз мають можливість передавати великі об'єми інформації. Завдяки цьому виникла можливість замінити аналогові методи передачі інформації на цифрові, наприклад можливість радіомовлення за допомогою комп'ютерних мереж чи інтернет-радіо.

Інтернет-радіо – передає звукові мовні передачі і музику за допомогою глобальної мережі Інтернет (WAN) в будь-яку точку Світу, де є доступ до Інтернет, або за допомогою локальної обчислювальної мережі (LAN) на будь-який комп'ютер, підключений до цієї мережі. У мережі Інтернет використовуються як провідні, так і бездротові технології передачі даних, у тому числі супутникові і стільникові мережі. Дротовими наземними лініями передачі даних є WAN / LAN-мережі, які застосовують технології Ethernet: VPN і звичайну телефонну лінію: DSL (ADSL), Dial Up. Супутниковий Інтернет використовує технологію DVB. Сьогодні поширюються бездротові технології передачі даних за технологією Wi-Fi і новому протоколу передачі даних WiMax, за ними йдуть мережі стільникових компаній за протоколами передачі даних GPRS і EDGE / 3G. Тобто, більш швидкісні бездротові інтернет-мережі, що передають дані на великі відстані – це WiMax та EDGE / 3G, які можуть служити відмінною платформою для передачі цифрового сигналу всіх інтернет-радіостанцій Світу з максимальною гарантією якості передачі звуку від джерела до споживача, абсолютно без виникнення шумових перешкод в дорозі. В принципі, нові бездротові мережі також називаються радіомережами (стільникові телефони теж є звичайними радіоприймачами), але їх відмінність полягає в тому, що вони передають цифрову, а не аналогову інформацію.

Інтернет радіомовлення передає потокові аудіо данні у якомусь форматі аудіоданих, наприклад: MP3, Ogg/Vorbis, RealAudio. У якості клієнта може виступати будь-який мультимедійний програвач, що підтримує потокове аудіо і здатний декодувати формат, в якому віщає радіо. Завдяки цьому забезпечується універсальність клієнтів, радіостанцію може програвати будь-які сучасні операційні системи, включаючи мобільні.

Вже сьогодні ми бачимо, що грань у радіокомпаній між поділом аналогового і цифрового радіомовлення стирається. Практично вже всі популярні

FM-радіостанції мають своє мовлення в Інтернеті і що найцікавіше: спочатку зародилися проекти, як інтернет-радіостанції, прагнуть вийти в FM-діапазон і у деяких це успішно виходить, крім того з переходом на цифрове мовлення в FM-діапазоні цей процес буде активно розвиватися й надалі: інтернет-радіостанції отримають можливість купити ліцензії на мовлення в FM-діапазоні[1].

На сьогоднішній час інтернет-радіо розвивається завдяки можливості швидко і з мінімальними витратами організувати свою радіостанцію.

Метою даного дипломного проекту є проектування локальної радіостанції, та сайту для адміністрування. Ця станція буде грати за програмою наперед створеного плейлісту, який включатиме в себе програвання музики, і записаних повідомлень у зазначений час.

Стаття за темою дипломного проекту подана до друку у збірник статей за матеріалами студентської конференції ОДЕКУ за 2017 рік.

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

1.1.1 Основні відомості по Інтернет-радіо

Інтернет-радіо – група технологій передачі потокових аудіоданих через комп’ютерну мережу для здійснення широкомовних передач. Також, як термін інтернет-радіо може розумітися радіостанція, яка використовує для мовлення технологію потокового мовлення в глобальній мережі Інтернет.

У технологічній основі системи лежить три елементи:

- станція – генерує аудіопоток (або зі списку звукових файлів, або прямий оцифруванням з аудіо карти, або копіюючи існуючий в мережі потік) і направляє його серверу (станція споживає мінімум трафіку, тому що створює один потік);
- сервер (повторювач потоку) – бере аудіопоток від клієнта і перенаправляє його копії всім підключеним до сервера клієнтам, по суті є реплікатором даних (сервер споживає трафік пропорційно кількості слухачів + 1, серверу потрібен хороший канал при великій кількості слухачів);
- клієнт – приймає аудіопоток від сервера і перетворює його в аудіосигнал, який і чує слухач інтернет-радіостанції. У деяких випадках в якості клієнта може виступати ще один повторювач, що дозволяє організувати каскадні системи радіомовлення (клієнт споживає мінімум трафіку, приблизно як і станція).

Крім потоку звукових даних також передаються і текстові дані, щоб в плеєрі відображалася інформація про станції і про поточну композиції.

Розглянемо поетапний спосіб підключення.

Unicast (Peer-to-peer) метод – Протоколи Unicast відправляють окрему копію даних кожному клієнту. Unicast підходить для більшості користувачів мережі Інтернет, але сильно утрудняє масштабування сервера для більшої кількості клієнтів.

Multicast – Протоколи Multicast розроблені для зниження навантаження з серверів на підключення / ширину каналу при отриманні потокового мультимедіа великою кількістю клієнтів. Ці протоколи відсилають одну порцію даних цілій групі клієнтів. Залежно від типу мережевої інфраструктури, групова передача даних може бути доступна, а може і не бути. Одним з потен-

ційних недоліків групової передачі є відсутність можливості реалізувати функцію відео за запитом. Безперервне мовлення потокової інформації також робить неможливим управління відтворенням користувачем. Проте, ця проблема може бути вирішена впровадженням у мережу передачі даних кешуючих серверів і буферизує приймається потік програмного забезпечення.

1.1.2 Історія Інтернет-радіо

Інтернет-радіо вперше було використано Карлом Маламудом. У 1993 році, Маламуд запустив «Internet Talk Radio», яке було «першим комп'ютерним радіо ток-шоу з щотижневим інтерв'ю з комп'ютерним спеціалістом.». Перший Інтернет-концерт трансливався на 24 червня 1993 року групою «Severe Tire Damage».

З 7 листопада 1994 року, WXYC (89,3 FM Chapel Hill, NC USA) стала першою традиційною радіостанцією, яка запустила мовлення в Інтернеті. WXYC використовувала FM-радіо, підключене до системи в SunSite, пізніше відому як Ibiblio, яка використовувала програмне забезпечення «Cornell's CU-SeeMe». WXYC почав тестові трансляції та тестування пропускну здатності вже в серпні 1994 року. WREK (91.1 FM, Atlanta, GA USA) почав транслявання в той же день, використовуючи своє власне програмне забезпечення під назвою CyberRadio1. Однак, на відміну від WXYC, це був бета запуск WREK і потік не рекламувався на більш пізній термін.

Журнал Time повідомив, що RealAudio прийняв «переваги останніх досягнень в області цифрового стиснення» і представив «АМ радіо-якість звуку в так званому режимі реального часу». Зрештою, такі компанії, як Nullsoft і Microsoft випустили плеєри потокового аудіо для вільного скачування. У міру того, як програми аудіоплеєри стали доступні, «почало з'являтися багато Веб-радіостанцій.»

У 1995 році Скотт Борн заснував NetRadio.com як першу в світі лише інтернет-радіомережу. NetRadio був піонером в інтернет-радіо. Це була перша лише Інтернет мережа, яка отримала ліцензію ASCAP. NetRadio в кінцевому підсумку пішли далі до первинної публічної пропозиції в жовтні 1999 р.

З появою потокового RealAudio через HTTP, транслявання стало більш доступним для багатьох радіо-шоу. Одним з таких шоу, TechEdge Radio в 1997 році трансливалося в 3-х форматах – ефір по радіо, ефір з сервером RealAudio і ефір з Інтернету по протоколу HTTP.

Починаючи з 2000 року, більшість радіостанцій в Інтернеті збільшили

якість потоку. На сьогоднішній день більшість станцій транслює потік між 64 кбіт/с і 320 кбіт/с забезпечуючи якість CD-аудіо[2, 3].

1.1.3 Програмне забезпечення по організації серверу

Shoutcast – кроссплатформне безкоштовне серверне ПЗ компанії Nullsoft. Призначено для організації потокового мовлення цифрового аудіо / відео сигналу у форматі MP3, AAC, AACplus, NSV, як в локальну мережу, так і в Інтернет. Простий в установці, налаштуванні та управлінні.

Мінімальний набір для організації мовлення складається з власне сервера Shoutcast (або демона), і джерела, в простому випадку реалізованого програмою Winamp або аналогічного програмного плеєра з відповідним плагіном, що кодує аудіосигнал з необхідною якістю, і передавального його в потоковому режимі сервера Shoutcast, для подальшої передачі ононого підключились до сервера клієнтам, в якості яких виступає будь-який програмний плеєр з підтримкою потокового мовлення.

Переваги:

- безкоштовний;
- простота налаштування;
- сумісність з основними ОС.

Недоліком є відсутність актуальних версій.

Apache є вільним програмним забезпеченням веб-аудіо файлового менеджера / веб-медіа серверу. Програма була спочатку написана, щоб скористатися Mod_mp3 Apache, але була адаптована для використання власного метода потокового медіа. Apache також забезпечує API для витягування метаданих у вигляді XML-документів. Apache дані можуть бути доступні за допомогою багатьох методів, підтримуваних в API, спочатку створеного для використання з Amrook2, але які також можуть бути використані для створення інших інструментів для Apache. Доступ до API контролюється внутрішніми списками контролю доступу; для підвищення продуктивності, всі запити в даний час мають межу 5000 результатів.

Переваги:

- безкоштовний;
- відмінна сумісність з операційними системами;
- наявність API і підтримка розширень.

Недоліком є складна установка і налаштування

QuickTime Streaming Server (QTSS) це сервер або сервісний демон,

який був побудований в Apple, Mac OS X Server до OS X Server 10.7. Він забезпечує відео і аудіо за запитом для користувачів через комп'ютерну мережу, в тому числі в Інтернеті. Його основний графічний інструмент конфігурації це QTSS Publisher і його порт є 1220. При використанні в поєднанні з QuickTime Broadcaster, можна поставляти в режимі реального часу відео та аудіо для декількох користувачів по мережі.

Переваги:

- простота налаштування і установки;
- наявність графічного інструменту конфігурації.

Недоліком є те, що працює лише на MAC OS.

Adobe Media Server (AMS) є медіа-сервер від Adobe Systems (спочатку продукт Macromedia). Цей сервер працює з використанням Flash Player для створення медіа, розрахована на багато багатофункціональних інтернет-додатків (Rich Internet Applications). Сервер використовує ActionScript 1, в основі ECMAScript скриптова мова, для серверної логіки. До версії 2, він був відомий як Flash Communication Server. До версії 5, він був відомий як Flash Media Server.

До переваг можна віднести універсальність завдяки технології Flash.

Недоліки:

- відсутність актуальної версії;
- потребує більше ресурсів ПК, ніж інші аналоги.

Icescast це проект потокового мультимедіа випущений як безкоштовне програмне забезпечення. Він також відноситься конкретно до програми сервера, який є частиною проекту. Сервер Icescast може транслювати потокове аудіо зміст, як Opus або Ogg/Vorbis у порівнянні зі стандартним HTTP, відео, як WebM або Theora по HTTP, MP3 по протоколу зв'язку, використовуюваного SHOUTcast, AAC, і NSV по протоколу SHOUTcast (Theora, AAC, і NSV підтримуються тільки у версії 2.2.0 і новіше).

Icescast потребує зовнішніх клієнтських програм для генерації потоків. Проект Icescast включає в себе клієнтську програму, відомий як Ices. Джерело аудіо працює, як правило, в місці, де генерується звук і Icescast сервер в місці, де доступно багато пропускної здатності.

Переваги:

- безкоштовний;
- підтримка багатьох форматів.

Недоліком є складність налаштування.

1.1.4 Програмне забезпечення для генерації аудіо потоку

SAM Broadcaster – професійне ПЗ для організації Інтернет-радіо. SAM – це потужне автоматизоване рішення для реалізації систем потокового мовлення, що розробляється протягом п'яти років спеціально для задоволення особливих запитів до систем онлайн-мовлення.

SAM3 в даний час підтримує формати MP3, mp3PRO, Windows Media 9 і потоковий Ogg Інтерфейс містить два пульта управління, розширену систему ротації плейліста, повну підтримку "drag & drop", статистику про слухачів, що відображається в реальному часі, автоматизовані запити, відповідає стандартам підтримку декількох серверів одночасно, включаючи Live365, Shoutcast, Iccast, Windows Media і ще масу можливостей.

Переваги:

- просте налаштування;
- зручний інтерфейс.

Недоліки:

- потребує платної підписки;
- працює лише на ОС Windows.

SimpleCast захоплює аудіо з вашого лінійний входу, мікрофону або будь якого звукового пристрою, а потім кодує цей звук в режимі реального часу і буде транслювати через Інтернет слухачам по всьому світу. Ідеально підходить для мовлення концертів, спортивних заходів або ток-шоу.

Переваги:

- просте налаштування;
- наявність API.

Недоліком є те, що програма потребує платної підписки.

IceS клієнт з вихідним кодом для потокового сервера. Метою даного клієнта є забезпечити потік аудіо до потокового сервера таким чином, що один або кілька слухачів могли отримати доступ до потоку. При такій схемі, цей клієнт може бути розташований віддалено з сервера IceCast. Основний приклад, потокового сервера, використовуваного в Iccast 2, хоча інші можуть бути використані, якщо дотримані певні умови. ICES 2.x підтримує відправку потоку Ogg Vorbis до сервера Iccast.

Переваги:

- безкоштовний;
- просте налаштування.

Недоліки:

- мала функціональність;
- невелика кількість підтримуваних форматів.

Liquidsoap – це мова програмування для аудіо, розроблена спочатку для створення аудіо- та відеопотоків, переданих з вихідного потоку на Icecast сервер. Різниця з будь-якими іншими доступними утилітами – це те, що Liquidsoap інтерпретує окрему скриптову мову, що робить його дуже універсальним і адаптуючим до різних вживань.

Останні релізи Liquidsoap також включають в себе можливість взаємодіяти з локальною звуковою картою або для виведення мультимедійних даних в локальні файли, використовуючи різні формати.

Таким чином, Liquidsoap не тільки клієнт-джерело аудіо- і відео потоку для Icecast, але общецільова спеціалізована мова аудіо, присвячена автоматизації обробки і трансляції аудіо і відео.

Liquidsoap опублікований під GNU General Public License (GPL), є частиною проекту Savonet, і розробляється в OCaml. Проект Savonet так само підтримує OCaml модулі, використовувані для різних додаткових можливостей, підтримуваних в Liquidsoap[4].

Переваги:

- безкоштовний;
- широка функціональність.

Недоліки:

- складне налаштування;
- відсутність графічного інтерфейсу.

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Технічне завдання

Даний розділ представляє собою технічні вимоги до програмного забезпечення радіомережі навчального закладу. Дані вимоги обов'язкові для врахування під час розроблення технічного завдання на програмний виріб.

При розробці технічних вимог було:

- 1) проаналізовано різні підходи до функціональності програмного забезпечення існуючих аналогів радіомережі, на різних технологіях та створених за допомогою різного програмного забезпечення;
- 2) визначено основні технічні вимоги до програмного забезпечення, необхідний склад технічних, функціональних та апаратних модулів.

2.1.1 Призначення розробки

Повне найменування системи: «Система мережевого радіомовлення».

Областю використання проекту є створення незалежної радіостанції з метою використовувати в якості інформувального каналу, трансляції музики, а також трансляції передач с ведучим.

Проект «Система мережевого радіомовлення» призначен для розміщення радіостанції у Інтернет просторі та локальної мережі закладу, спрямоване на інформування в прямому ефірі, а також організувати програми.

Для того, щоб усі бажаючі змогли прослухати радіостанцію навчального закладу, буде використовуватися програмне забезпечення Icescast у поєднанні з мовою програмування Liquidsoap.

Завдяки використанню протоколу http для трансляції, можливо охопити 90% користувачів мережі Інтернет, які можуть забажати ознайомитися з радіостанцією. В свою чергу Liquidsoap допоможе створити необхідні параметри радіостанції.

2.1.2 Вимоги до програмної документації

Кожна стадія проектування завершується складанням відповідних документів. Тому важливим елементом проектування програмних додатків

є оформлення програмної документації. Виняток може становити нескладних програм із коротким життєвим циклом і невеликою трудомісткістю.

Зміст програмних документів:

- специфікація – перелік і призначення всіх файлів програмного виробу, включаючи файли документації;
- текст програми – запис кодів програми та коментарі до них;
- програма і методика випробувань – перелік і опис вимог, які повинні бути перевірені в ході випробування програми, методи контролю;
- технічне завдання – документ, в якому викладаються призначення і область застосування програми, вимоги до програмного виробу, стадії і терміни розробки, види випробувань;
- пояснювальна записка – обґрунтування прийнятих і застосованих технічних і техніко-економічних рішень, схеми та опис алгоритмів, загальний опис роботи програмного виробу.

До програмних документів віднесено також документи, що забезпечують функціонування та експлуатацію програм – експлуатаційні документи:

- опис мови – містить синтаксис і семантику мови;
- керівництво з технічного обслуговування – містить відомості для застосування.
- тестових і діагностичних програм при обслуговуванні технічних засобів.

2.1.3 Стадії й етапи розробки програмного проекту

Розробка повинна бути виконана в три стадії:

- 1) розробка технічного завдання;
- 2) робочі проектування;
- 3) впровадження.

На першій стадії повинен бути виконаний етап розробки: погодження та затвердження цього технічного завдання.

На стадії робочого проектування повинні бути виконані перераховані нижче етапи робіт:

- 1) розробка програми;
- 2) розробка програмної документації;
- 3) випробування програми.

Зміст робіт по етапах.

На етапі розробки технічного завдання виконуються такі роботи:

- 1) постановка завдань;
- 2) визначення та уточнення вимог до технічних засобів;
- 3) визначення вимог до програми;
- 4) визначення стадій, етапів і термінів розробки програми та документації на неї;
- 5) вибір мов програмування;
- 6) узгодження та затвердження технічного завдання.

2.1.4 Порядок контролю і прийому проекту

Перевірка документації програми здійснюється самим замовником із залученням сторонніх експертів, здатних засвідчити факт відповідності створеного програмного продукту всім пунктам технічної документації, включаючи технічне завдання і технічний проект.

Іспити і тестування програми повинні проводитися в процесі створення програми самим розроблювачем:

- 1) з використанням контрольних тестів, що дозволяють домогтися перевірки працездатності;
- 2) із залученням сторонніх неофіційних бета-тестерів, що у процесі тестування програмного продукту повинні повідомляти розроблювачеві всі знайдені помилки і похибки в роботі програми.

Іспити і тестування відбуваються після завершення створення програми замовником:

- 1) з використанням перевірочних тестів;
- 2) в організації замовника із залученням сторонніх експертів.

2.2 Опис протоколу Iccast

Незважаючи на те, що було створено низку протоколів для трансляції мультимедійних файлів по мережі (UDP / RTP, RTSP, MMS, SDP та ін.), зараз для радіотрансляцій по мережі переважно використовуються протоколи TCP / HTTP з програмним протоколом Iccast.

TCP – надійний байт-орієнтований (byte-stream) протокол із установленням з'єднання. TCP знаходиться на транспортному рівні стека TCP / IP, між протоколом IP і власне додатком. Протокол IP займається пересиланням даних по мережі, неможливо гарантуючи доставку, цілісність, порядок

прибуття інформації і готовність одержувача до прийому даних; всі ці завдання покладені на протокол TCP.

Для прискорення та оптимізації процесу передачі великих обсягів даних протокол TCP визначає метод управління потоком, званий методом ковзного вікна, який дозволяє відправнику посилати черговий сегмент, не чекаючи підтвердження про отримання в пункті призначення попереднього сегмента.

Протокол TCP формує підтвердження не для кожного конкретного успішно отриманого пакета, а для всіх даних від початку посилки до деякого порядкового номера ACK SN (Acknowledge Sequence Number) виключно. Як підтвердження успішного прийому, наприклад, перших 2000 байт, висилається ACK SN = 2001: це означає, що всі дані в байтовому потоці під номерами від $ISN + 1 = 1$ до даного ACK SN -1 (2000) успішно отримані

Разом з посилкою відправнику ACK SN одержувач оголошує також "розмір вікна", наприклад – 6000. Це означає, що відправник може посилати дані з порядковими номерами від поточного ACK SN = 2001 до $(ACK SN + \text{розмір вікна} - 1) = 8000$, не чекаючи підтвердження з боку одержувача. Припустимо, в даний момент відправник посилає тисячеоктетний сегмент з порядковим номером даних SN = 4001. Якщо не буде отримано нове підтвердження (новий ACK SN), відправник буде посилати дані, поки він залишається в межах оголошеного вікна, тобто до номера 8001. Після цього посилка даних буде припинена до отримання чергового підтвердження і (можливо) нового розміру вікна. Однак розмір вікна вибирається таким чином, щоб підтвердження встигали приходити вчасно і зупинки передачі не відбувалося – для цього й призначений метод ковзного вікна. Розмір вікна може динамічно змінюватися одержувачем.

HTTP, підтримуваний протоколами IP і TCP (таким чином, HTTP сам не обробляє передачу пакетів даних) – це протокол, який дозволяє браузерам завантажувати HTML-сторінки. Цей протокол дозволяє не тільки пов'язувати документи між клієнтом і сервером, але також він допускає потокову трансляцію аудіо за допомогою передачі пакетів і виправляє помилки (у випадку, якщо пакет не буде доставлене з першого разу) за допомогою повторної передачі. Особливістю мережевих трансляцій є те, що для прослуховування аудіопотока не потрібно завантажувати аудіофайли цілком, а сама трансляція не зберігається на диску.

Iccast розробляється під ліцензією GPL (є проектом з відкритим вихідним кодом). Протокол у великій мірі заснований на HTTP / 1.0. Основна

відмінність, це група нових заголовків: x-audiocast-заголовків у Icecast.

URL типового Icecast-потoku має вигляд:

`http: // Server [/ path] [/ file]: port`

або

`http: //Server/path/file.pls`

Номер порту, як правило, лежить в діапазоні 8000-8999, в будь-якому випадку, він призначається сервером. Багато Icecast-серверів не мають власних доменних імен. Таким чином, URL зазвичай має вигляд:

`http: //nnn.nnn.nnn.nnn: XXXX`

де nnn.nnn.nnn.nnn – це IP-адресу сервера, а XXXX – номер порта[5].

2.3 Основні інструменти при трансляванні

Основними інструментами при трансляванні медіа є:

- джерело (генерує mp3-потік);
- сервер (поставляє mp3-потік джерела клієнту);
- клієнт (використовується для прослуховування аудіопотока, що йде з сервера).

2.3.1 Джерело-сервер

Щоб сервер міг зв'язуватися з клієнтом, йому потрібен джерело. Коли з'єднання з джерелом встановлено, сервер буде передавати дані клієнтам, коли вони будуть підключатися. Діалог відбувається так:

- 1) джерело створює з'єднання з портом сервера (службовим);
- 2) потім джерело посилає пароль: `password \ r \ n`;
- 3) якщо пароль правильний, сервер посилає у відповідь:

`OK2 \ r \ n`

`icy-caps: 11 \ r \ n \ r \ n`

що інформує джерело про те, що сервер авторизував dsp-модуль в якості джерела і готовий приймати дані. Якщо пароль неправильний, сервер відправить у відповідь неправильний пароль *password \r \n*.

Якщо джерело отримує у відповідь ОК2, він починає посилати інформацію про потоці сервера. Як правило, у формі:

```
icy-name: Unnamed
Server \r \n
icy-genre: Unknown
Genre \r \n
icy-pub: 1 \r \n
icy-br: 56 \r \n
icy-url: http://www.example.com \r \n
icy-irc:% 23shoutcast \r \n
icy-icq: 0 \r \n
icy-aim: N% 2FA \r \n
\r \n
```

Тут для передачі інформації і потоці використовуються заголовки:

icy-name – назва станції

icy-genre – музичний жанр станції

icy-pub – вказує чи допускає сервер публікацію себе у публічній директорії (1 – так, 0-ні)

icy-br – бітрейт потоку

icy-url – homepage потоку

icy-irc, icy-icq, icy-aim – контактна інформація для публікації в публічній директорії

Потім джерело починає відправляти mp3-потік.

2.3.2 Клієнт-сервер

Взаємодія клієнт-сервер відбувається способом, аналогічним тому, як взаємодіють браузер і веб-сервер – по протоколу HTTP. Однак Icecast має додаткові заголовки.

Клієнт підключається до сервера і, в додаток до звичайного HTTP-

заголовку, відправляє йому додаткове поле:

```
icy-metadata: val \r \n
```

Цей тег вказує на те, що якщо $val = 1$, то клієнт може обробляти назви пісень (метадані), передані в потоці, і, таким чином, сервер буде посилати додаткову інформацію про назву. Якщо $val = 0$, то метадані передаватися не будуть.

Потім сервер відправляє відповідь:

```
ICY 200 OK \r \n (означає, що сервер прийняв запит)
```

```
icy-notice1: <BR> This stream requires <a href = "http://www.Example.com/">
Example </a> <BR> (надлишкове зауваження)
```

```
icy-notice2: SHOUTcast Distributed Network Audio Server / posix v1.xx <BR>
(Повідомляє клієнту, який це сервер та його версію)
```

```
icy-name: Unnamed Server \r \n (ім'я сервера)
```

```
icy-genre: Unknown Genre \r \n (жанр сервера)
```

```
icy-url: http://www.example.com \r \n (homepage сервера)
```

```
icy-pub: 1 \r \n (публічний або непублічний сервер)
```

```
icy-br: 56 \r \n (бітрейт сервера)
```

```
icy-metaint: 8192 \r \n (див. далі)
```

```
\R \n (кінець заголовка)
```

З цього моменту сервер починає посилати аудіо-дані. На рис. 2.1 зображена структура цих даних.



Рисунок 2.1 – Потік в Icast

2.3.3 Установка з'єднання з сервером, передача mp3-даних

Передачею даних mp3-плеєра вони буферизуються для того, щоб уникнути спотворення аудіопотока. Розглянутий клієнт не обробляє назви пісень.

Icecast-клієнт використовує додатковий UDP-канал для оновлення метаданих. Для метаданих використовуються HTTP-заголовки x-audiocast.

Отже, процедура з боку клієнта виглядає наступним чином:

- 1) отримуємо і розбиваємо адресу трансляції на ім'я хоста і порт;
- 2) створюємо два сокета: один (TCP) для отримання mp3-поток, інший (UDP) для передачі користувальницьких датаграмм і метаданих;
- 3) відправляємо в сокет повідомлення виду (обмін повідомленнями йде через UDP-сокет).

GET / HTTP / 1.0

Host: ****.****.****.***

x-audiocast-udpport: 6000

Icy-MetaData: 0

Асцепт: * / *

Отримуємо з сокета повідомлення виду:

HTTP / 1.0 200 OK

Server: Icecast / VERSION

Content-Type: audio / mpeg

x-audiocast-name: Great Songs

x-audiocast-genre: Jazz

x-audiocast-url: http: //icecast.serv.dom/

x-audiocast-streamid:

x-audiocast-public: 0

x-audiocast-bitrate: 24

x-audiocast-description: served by Icecast

І далі читаємо з сокета в буфер дані, у разі icecast буде тільки mp3-потік – <data>. Mp3-дані передаються у вигляді так званих фреймів (frame, або кадр), в яких зберігаються аудіодані всередині mp3-файлу[6].

2.3.4 Топологія мережі для роботи радіо

На рис. 2.2 представлено приклад топології мережі для запуску інтернет-радіо.

Топологія мережі – це спосіб опису конфігурації мережі, схема розташування і з'єднання мережевих пристроїв. Топологія мережі дозволяє поба-

чити всю її структуру, мережеві пристрої, що входять в мережу, і їх зв'язок між собою. На схемі зображено основний комплекс для запуску інтернет-радіо(джерело, сервер) який передає дані в мережу. Маршрутизатор на схемі використовується в якості зв'язуючого пристрою в мережі.

В цілях безпеки джерело не підключено в мережу, вся взаємодія здійснюється через комп'ютер адміністратора. Це захищає джерело від прямих атак ззовні мережі, але залишається уразливість через адміністратора. Під мережею мається на увазі будь-яка зовнішня мережа з активними користувачами(локальна мережа, мережа Інтернет). Будь-який клієнт може слухати трансляцію, незважаючи на спосіб підключення і тип клієнту.

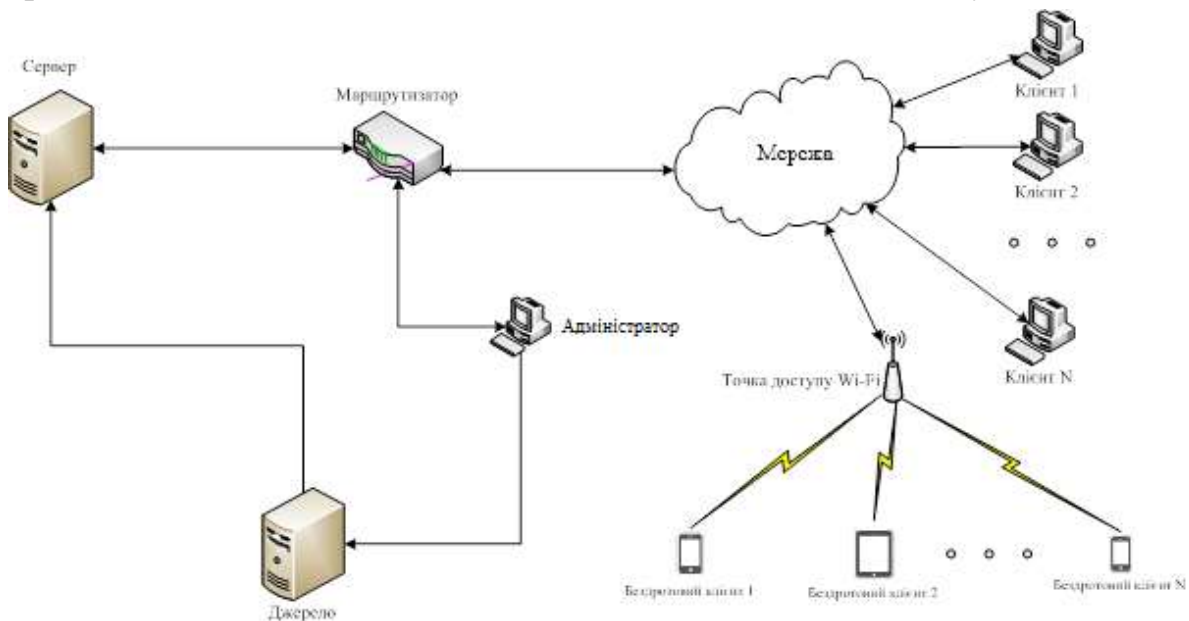


Рисунок 2.2 – Структурна модель програмного комплексу

2.3.5 Управління джерелом аудіопотоку за допомогою веб-сторінки

Джерело Liquidsoap не має стандартного графічного інтерфейсу для управління. Для того, щоб здійснювати управління з віддаленого комп'ютера, використовується протокол telnet.

Telnet (англ. TErminaL NETwork) – мережевий протокол для реалізації текстового інтерфейсу по мережі (у сучасній формі – за допомогою транспорту TCP). Назву «telnet» мають також деякі утиліти, що реалізують клієнтську частину протоколу. Призначення протоколу TELNET у наданні достатньо спільного, двонаправленого, восьмибітового байт-орієнтованого засобу зв'язку. Його основне завдання полягає в тому, щоб дозволити термінальним при-

строям і термінальним процесам взаємодіяти один з одним. Передбачається, що цей протокол може бути використаний для зв'язку виду термінал («зв'язування») або для зв'язку процес-процес («розподілені обчислення»).

Історично Telnet служив для віддаленого доступу до інтерфейсу командного рядка операційних систем. Згодом його стали використовувати для інших текстових інтерфейсів, аж до ігор MUD і анімованого ASCII-art. Теоретично, навіть обидві сторони протоколу можуть бути програмами, а не людиною. Іноді клієнти telnet використовуються для доступу до інших протоколів на основі транспорту TCP. Протокол telnet використовується в керуючому з'єднанні FTP, тобто заходити на сервер командою `telnet ftp.example.net ftp` для виконання налагодження і експериментів не тільки можливо, але і правильно (на відміну від застосування клієнтів telnet для доступу до HTTP, IRC і більшості інших протоколів).

У протоколі не передбачено використання ні шифрування, ні перевірки достовірності даних. Тому він вразливий для будь-якого виду атак, до яких вразливий його транспорт, тобто протокол TCP. Для функціональності віддаленого доступу до системи в наш час застосовується мережевий протокол SSH, основною причиною створення якого були питання безпеки. Так що варто мати на увазі, що сесія Telnet дуже незахищена, якщо тільки не здійснюється в повністю контрольованій мережі або з застосуванням захисту на мережевому рівні (різні реалізації віртуальних приватних мереж).

Telnet не має інтерфейсу, отож для зручності створюється веб-сторінка, яка слугує дружельюбним інтерфейсом для виконання telnet команд.

Веб-сторінка (англ. Web page) – документ або інформаційний ресурс Всесвітньої павутини, доступ до якого здійснюється за допомогою веб-браузера. Типова веб-сторінка являє собою текстовий файл у форматі HTML, який може містити посилання на файли в інших форматах (текст, графічні зображення, відео, аудіо, мультимедіа, аплети, прикладні програми, бази даних, веб-служби та інше), а також гіперпосилання для швидкого переходу на інші веб-сторінки або доступу до посилальних файлів. Багато сучасні браузери дозволяють перегляд змісту посилальних файлів безпосередньо на веб-сторінці, що містить посилання на даний файл. Сучасні браузери також дозволяють прямий перегляд змісту файлів певних форматів, у відриві від веб-сторінки, яка на них посилається.

Інформаційно значущий вміст веб-сторінки зазвичай називається контентом (від англ. Content – «зміст»).

Кілька веб-сторінок, об'єднаних загальною темою і дизайном, а також

пов'язаних між собою посиланнями, утворюють веб-сайт. При цьому утворюють веб-сайт сторінки можуть перебувати на одному або декількох веб-серверах, які можуть розташовуватися в одному дата-центрі або віддалено один від одного, часто в різних країнах.

Функціональність веб-сторінки залежить від потреби адміністраторів, також можна надати слухачам можливість замовляти музику, яка буде програватися далі. Для розміщення веб-сторінки в мережі необхідно встановити веб-сервер. Найпростіший варіант – це встановити комплекс LAMP.

LAMP – аббревіатура набору вільного ПЗ з відкритим кодом, в який входять ОС Linux, веб-сервер Apache, СКБД MySQL, та інтерпретатор Perl/PHP/Python – основні компоненти для побудови життєздатного багатозначного веб-сервера.

Існують аналогічні терміни (AMP) для подібних наборів, що працюють на інших системах, наприклад Microsoft Windows (WAMP), Mac OS (MAMP), Solaris (SAMP), чи OpenBSD (OAMP)[7].

Веб-фреймворк – це програмний засіб, який надає можливість створювати та запускати веб-додатки. У результаті вам не потрібно писати код самостійно і витратити час на пошук можливих помилок і помилок. У перші дні розробки веб-сайтів всі програми були написані вручну, і лише розробник певного додатка міг змінити його або застосувати його. Веб-фреймворк представив простий вихід із цієї пастки. Починаючи з 1995 року, всі труднощі, пов'язані зі зміною структури програми, були приведені в порядок через появу загальної продуктивності. І саме тоді з'явилися веб-мови. Їх різноманітність тепер добре працює як на статичних, так і на динамічних веб-сторінках. Ви можете обрати одну структуру, яка охоплює всі ваші потреби або об'єднує декілька, залежно від вашого завдання.

В ході виконання завдання буде використовуватися веб-фреймворк Django. Django – це безкоштовна веб-система з відкритим кодом на основі Python, яка відповідає архітектурній схемі model-view-template (MVT). Вона підтримується Django Software Foundation, незалежною організацією, яка заснована як некомерційна.

Основна мета Django – полегшити створення складних веб-сайтів, керуваних базами даних. Фреймворк забезпечує повторюваність і "підключення" компонентів, менший код, швидкий розвиток, і принцип не повторювати себе. Python використовується у всьому, навіть для файлів налаштувань та моделей даних. Django також забезпечує необов'язковий адміністративний інтерфейс створення, читання, оновлення та видалення, який генерується ди-

намічно через інтроспекцію та налаштовується за допомогою моделей адміністратора.

Деякі відомі сайти, що використовують Django, включають Instagram, Mozilla, The Washington Times, Disqus, Bitbucket.

2.4 Алгоритм програми

2.4.1 Взаємодія користувача з системою

Для відображення можливих дій користувача з системою, а також взаємодію компонентів системи між собою використаємо діаграму прецедентів.

Діаграма прецедентів (діаграма варіантів використання) в UML – діаграма, що відображає відносини між акторами і прецедентами і є складовою частиною моделі прецедентів, що дозволяє описати систему на концептуальному рівні. Прецедент – можливість модельованої системи (частина її функціональності), завдяки якій користувач може отримати конкретний, вимірний і потрібний йому результат. Прецедент відповідає окремому сервісу системи, визначає один з варіантів її використання і описує типовий спосіб взаємодії користувача з системою. Варіанти використання зазвичай застосовуються для специфікації зовнішніх вимог до системи.

Основне призначення діаграми – опис функціональності і поведінки, що дозволяє замовнику, кінцевому користувачеві і розробнику спільно обговорювати проєктовану або існуючу систему.

При моделюванні системи за допомогою діаграми прецедентів системний аналітик прагне:

- 1) чітко відокремити систему від її оточення;
- 2) визначити дійових осіб (акторів), їх взаємодія з системою і очікувану функціональність системи;
- 3) визначити в глосарії предметної області поняття, які стосуються детального опису функціональності системи (тобто, прецедентів).

Робота над діаграмою може початися з текстового опису, отриманого при роботі із замовником. При цьому нефункціональні вимоги (наприклад, конкретну мову або систему програмування) при складанні моделі прецедентів опускаються (для них складається інший документ).

Для відображення моделі прецедентів на діаграмі використовуються:

- 1) рамки системи (англ. system boundary) – прямокутник з назвою у верхній частині і еліпсами (прецедентами) всередині. Часто може

- бути опущений без втрати корисної інформації;
- 2) актор – стилізований чоловічок, що позначає набір ролей користувача (розуміється в широкому сенсі: людина, зовнішня сутність, клас, інша система), що взаємодіє з деякою сутністю (системою, підсистемою, класом). Актори не можуть бути пов'язані один з одним (за винятком відносин узагальнення / успадкування);
 - 3) прецедент – еліпс з написом, що позначає виконуваних системою дії (можуть включати можливі варіанти), що призводять до спостережуваних акторами результатами. Напис може бути ім'ям або описом (з точки зору акторів) того, «що» робить система (а не «як»). Ім'я прецеденту пов'язано з неперервним (атомарним) сценарієм – конкретної послідовністю дій, що ілюструє поведінку.

В ході сценарію актори обмінюються з системою повідомленнями. Сценарій може бути приведений на діаграмі прецедентів у вигляді UML-коментаря. З одним прецедентом може бути пов'язано кілька різних сценаріїв.

Частина дублюється інформації в моделі прецедентів можна усунути зазначенням зв'язків між прецедентами:

- 1) узагальнення прецеденту – стрілка з незафарбованим трикутником (трикутник ставиться у більш загального прецеденту);
- 2) включення прецеденту – пунктирна стрілка «include»;
- 3) розширення прецеденту – пунктирна стрілка «extend» (стрілка входить в розширюваний прецедент, в додатковому розділі якого може бути вказана точка розширення і, можливо у вигляді коментаря, умова розширення).

При роботі з варіантами використання важливо пам'ятати кілька простих правил:

- 1) кожен прецедент відноситься як мінімум до одного дійової особи;
- 2) кожен прецедент має ініціатора;
- 3) кожен прецедент призводить до відповідного результату.

На рис. 2.3 представлено діаграму прецедентів.

Основними акторами являються компоненти системи(сервер, джерело) та її користувачі(адміністратор, слухач). Із заданих прецедентів можна визначити роль кожного актора у системі. На рисунку чітко просліджується взаємодія між акторами. Адміністратор є також слухачем, тому він має можливості слухача, але слухач не має можливостей адміністратора.

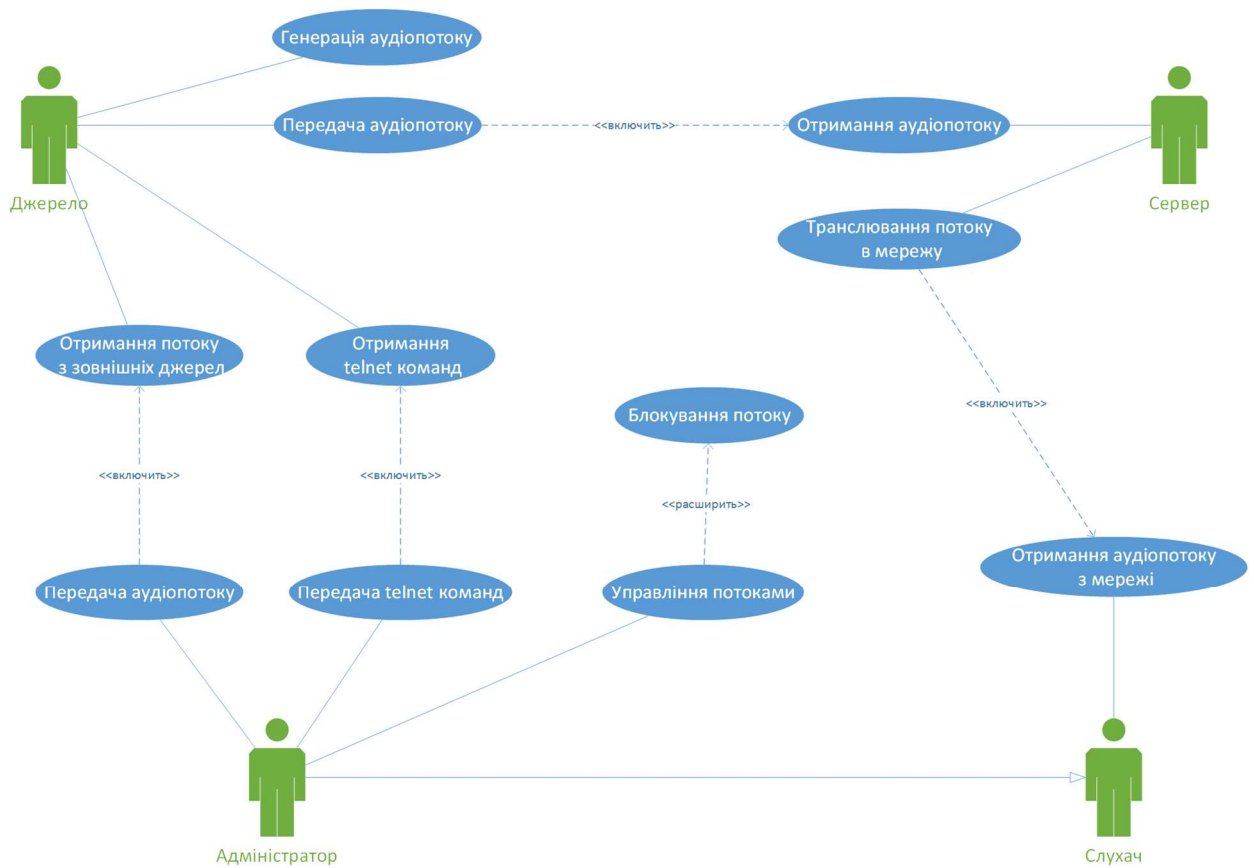


Рисунок 2.3 – Діаграма прецедентів для інтернет радіо

Слухач взаємодіє тільки з сервером, але адміністратор впливає також на джерело і може їм керувати.

Усе це робить систему взаємопов'язаною.

2.4.2 Логіка роботи програми

Для опису логіки програми використаємо діаграму діяльності, яка є своєрідною блок-схемою, що описує послідовність виконання операцій у часі. Їх можна використовувати для моделювання динамічних аспектів поведінки системи.

Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні цієї операції в попередньому стані.

У діаграмах діяльності використовуються піктограми "дія", "перехід", "вибір" і "лінії синхронізації". У мові UML дію зображується у вигляді прямокутника з закругленими кутами, переходи – у вигляді спрямованих стрілок, елементи вибору – у вигляді ромбів, лінії синхронізації – у вигляді гори-

зонтальних і вертикальних ліній. На рис. 2.4 приведені основні елементи.

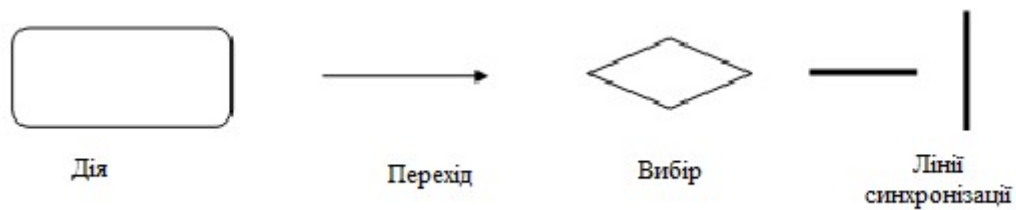


Рисунок 2.4 – Основні елементи діаграми діяльності

Стан дії є спеціальним випадком стану з деякою вхідною дією і, принаймні, одним вихідним зі стану переходом. Коли дія або діяльність в деякому стані завершується, потік управління відразу переходить в наступний стан дії або діяльність. Для опису цього потоку використовуються переходи, що показують шлях з одного стану дії або діяльності в інше.

Прості послідовні переходи зустрічаються найбільш часто, але їх одних недостатньо для моделювання будь-якого потоку управління. Як і в блок-схемі, ви можете включити в модель вибір, який описує різні шляхи виконання в залежності від значення деякого булевського вираження.

При моделюванні течії бізнес-процесів іноді буває корисно розбити стан діяльності на діаграмах діяльності на групи, кожна з яких представляє відділ компанії, що відповідає за ту чи іншу роботу. В UML такі групи називаються доріжками, оскільки візуально кожна група відокремлюється від сусідніх вертикальною рисою, як плавальні доріжки в басейні. Кожній присутній на діаграмі доріжці присвоюється унікальне ім'я. Кожна доріжка являє сферу відповідальності за частину всієї роботи, зображеної на діаграмі, і може бути реалізована одним або декількома класами. На рис. 2.5 зображена логіка роботи розроблюваної системи.

Діаграма діяльності даного виду добре відображає:

- послідовність дій;
- події, які ініціюють дії або є кінцевим результатом;
- умови розширення сценарію;

Якщо джерело не отримало команду завершення роботи, то в будь-якому разі воно буде генерувати аудіопотік, вміст якого залежить від заданих умов. Серверна частина налаштовано постійно транслювати потік у мережу, окрім випадків, коли потоки заблоковані адміністратором[8].

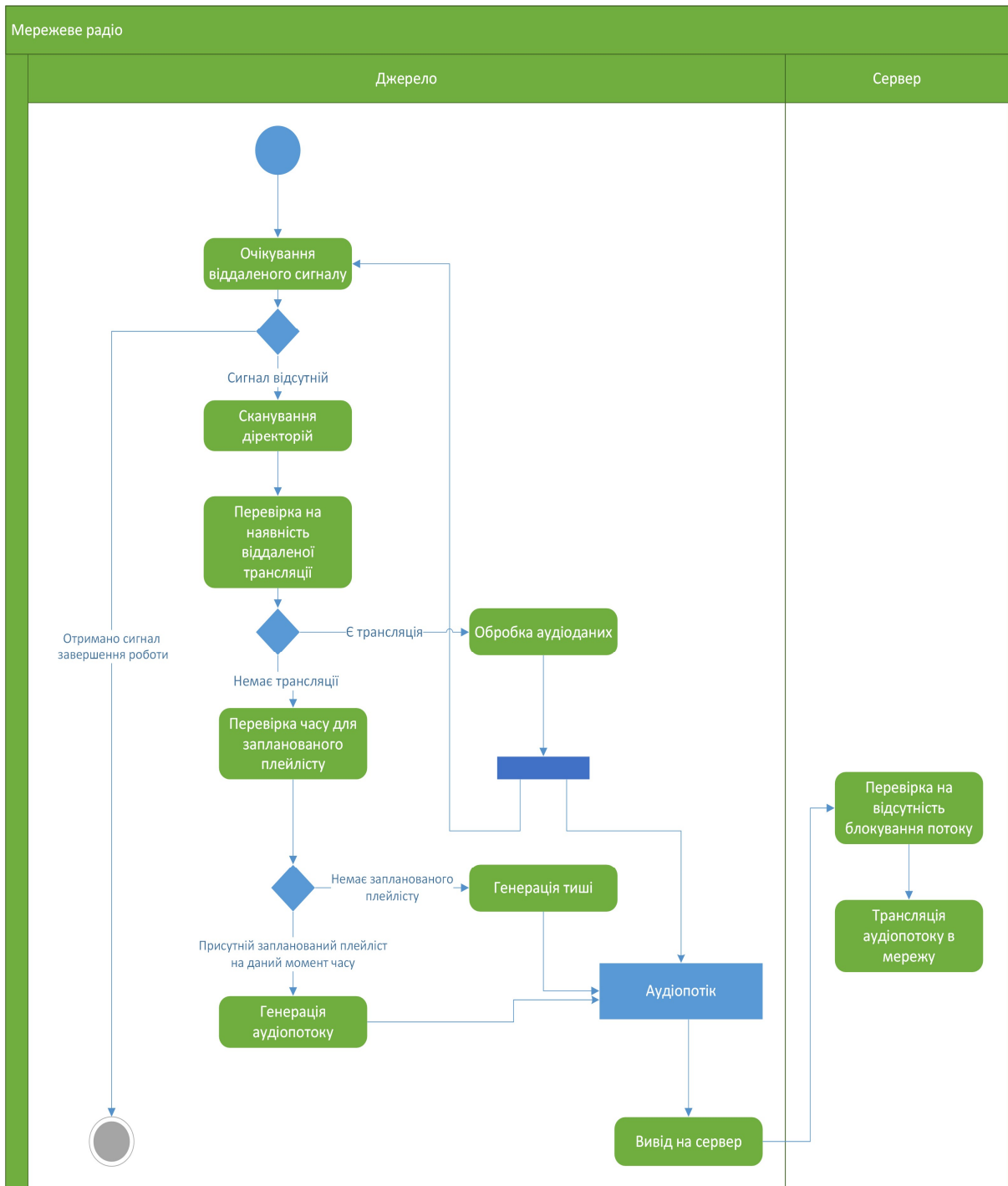


Рисунок 2.5 – Діаграма діяльності

2.5 Тестування та налагодження системи

Для тестування ми будемо використовувати продукт корпорація Microsoft, що дозволяють протестувати веб-сервер під навантаженням, MAST (Microsoft Application Stress Tool).

Тестування це процес визначення того, як програма відповідає певному набору умов. Кілька окремих сценаріїв тестування продуктивності (люкси,

випадки, скрипти) часто необхідні, щоб охопити всі умови та інтереси. Для більш якісного тестування, якщо це можливо, програма має бути розміщена на апаратну інфраструктуру, яка представляє живою середовищем.

Основна мета тестування полягає у визначенні для інтернет радіотого, наскільки добре програма виконується по відношенню до якогось параметру.

Ми провели тест на витримку навантаження, який дозволить визначити наступні показники: пропускна спроможність, кількість одночасно виконуваних запитів, середні час відгуку, пороги використання ресурсів (процесор, пам'ять).

Процес тестування навантаження складається з наступних етапів:

- 1) визначення ключових сценаріїв – визначити сценарії додатків, критичних для продуктивності;
- 2) визначте робочу задачу – розподілу навантаження на всі програми навантаження серед основних сценаріїв, зазначених у пункті 1;
- 3) визначити метрики – визначити показники, які ви хочете зібрати при виконанні тесту;
- 4) створення тестів – розробка тестових прикладів, де ви визначаєте кроки для виконання одного тесту поряд з очікуваними результатами;
- 5) моделювання навантаження – використання тестових інструментів для моделювання навантаження відповідно до тестів і захопити результати метрик;
- 6) проаналізуйте результати – аналіз метричних даних, отриманих в ході випробувань.

Почнемо з визначення ключових сценаріїв нашого сервера. Сценарії визначають поведінку користувачів, які зазвичай включають кілька заходів програми. Основні сценарії є ті, для яких у вас є конкретні цілі роботи або ті, які мають значний вплив на продуктивність. Наш ключовий сценарій включають в себе наступне:

- 1) увійти на сервер;
- 2) переглянути інформацію о радіостанції;
- 3) завантаження плейлісту;
- 4) пошук неіснуючого плейлісту, по закінченню вихід із системи.

Далі визначимо характеристики та навантаження, пов'язані з кожним із заданих сценаріїв. Для кожного сценарію необхідно визначити наступне:

- кількість учасників.
- інтенсивність запитів. Кількість запитів, отриманих від одночасного

навантаження користувачів в одиницю часу;

- шаблони запитів. Кількість паралельних користувачів, які звертаються до конкретної програми, і навантажують її.

Основні параметри тесту:

- кількість одночасних користувачів: 15 користувачів;
- кількість користувачів на ключових сценаріях;
- тривалість тесту: 20 хвилин;
- випадковий час: від 1 до 10 секунд на виконання тестового скрипту після кожної операції.

Підсумки:

- пропускна здатність: 34 запити в секунду;
- виконується запитів: 15 одночасно виконуваних запитів;
- середній час відгуку: час відгуку 2,0 секунди;
- пороги використання ресурсів: процесор: 80 відсотків, пам'ять: 61 відсотків від загального обсягу оперативної пам'яті.

2.6 Інструкція з експлуатації системи

Сервер Icescast містить веб-сайт, який використовується в якості панелі керування. Він призначений для контролю радіостанцій, що транслюються через сервер.

Головна сторінка містить три вкладки:

- «Administration»;
- «Server Status»;
- «Version».

Вкладка Version містить адресу даного сервера, а також посилання на офіційний сайт та форум.

Вкладка Status містить базову інформацію о трансльованих радіостанціях. Тут можна дізнатися ім'я та опис радіостанції, її бітрейт, кількість слухачів та поточну композицію.

Вкладка Administration доступна лише після вводу паролю адміністратора. Вона містить три вкладки:

- «Admin Home»;
- «Mountpoint List»;
- «Public Home».

Вкладка Public Home містить посилання на домашню сторінку. Після переходу в вкладку адміністрування знову, ввід паролю не потребується до

перезавантаження сторінки.

Вкладка Admin Home містить розширену о трансльованих радіостанціях інформацію порівняно з загальною вкладкою статусу. Також тут можна знайти інформацію про статус серверу..

Вкладка Mountpoint List містить список трансльованих радіостанцій, показуючи лише кількість слухачів, доступні для завантаження файли плей-лістів, а також кнопки інструментів для керування. Панель містить такі кнопки керування:

- «List Clients»;
- «Move Listeners»;
- «Update Metadata»;
- «Kill source».

List Clients виводить список усіх підключених до даної радіостанції клієнтів. Список містить IP адресу клієнту, час, який клієнт був підключений, у секундах, User Agent клієнта, а також дозволяє відключити даного клієнта від трансляції, натиснувши “Kick”. На рис. 2.6 приведений інтерфейс цієї вкладки.

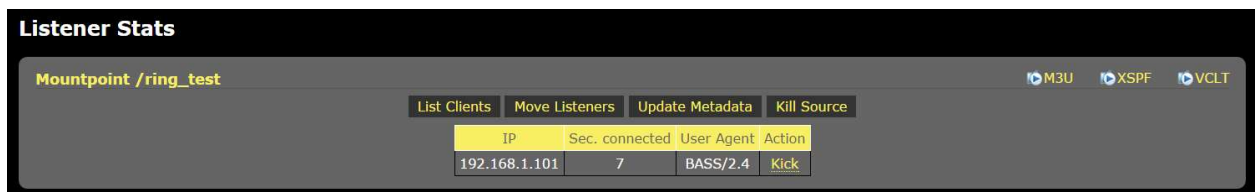


Рисунок 2.6 – Список клієнтів

Move Listeners доступна, якщо сервер транслює декілька радіостанцій. Ця функція дозволяє змінити потік однієї трансляції потоком іншої без перезавантаження клієнту. У нашому випадку ця функція дозволяє вимкнути музику на клієнтах, де вона присутня, або навпаки, ввімкнути на всіх клієнтах. Для цього необхідно натиснути “Move clients”. Для відміни цього необхідно виконати теж саме, перекинувши клієнтів з однієї радіостанції на іншу. На рис. 2.7 приведений інтерфейс цієї вкладки. Update Metadata дозволяє змінити заголовок, що відображується у клієнті під час програвання радіостанції. Щоб змінити заголовок необхідно ввести його у текстове поле і натиснути “Update”. На рис. 2.8 приведений інтерфейс цієї вкладки.

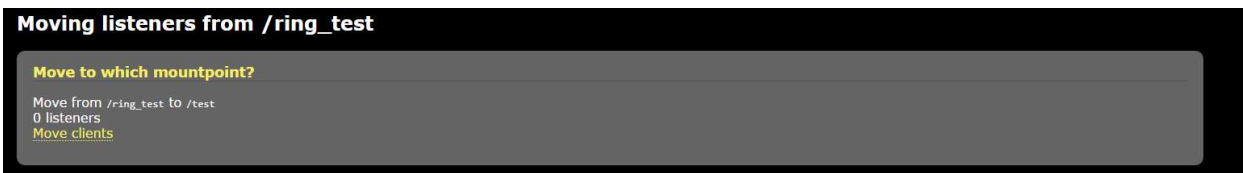


Рисунок 2.7 – Переміщення клієнтів

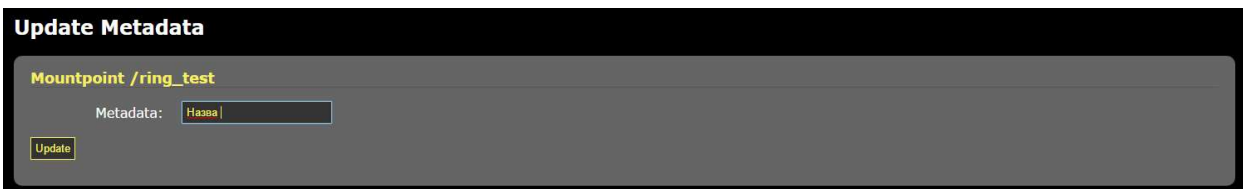


Рисунок 2.8 – Зміна заголовку

Kill source дозволяє тимчасово відключити трансляцію. При натисненні на кнопку трансляцію відключиться. Для відновлення трансляції достатньо повернутися на попередню сторінку у веб-браузері.

Для зміни музикальних композицій необхідно додати чи видалити музикальні композиції із папки, де вони розташовані. Перезавантажувати сервер не треба, програма джерело автоматично оновлює плейліст. На рис 2.9 зображено програвання радіостанції у безкоштовному плеєрі Winamp.



Рисунок 2.9 – Програвання радіостанції у Winamp

3 ОПИС ПРОЕКТНОГО І ТЕХНІЧНОГО РІШЕННЯ

3.1 Логічне проектування

Для виконання завдання з організації мережевого мовлення нам знадобиться машина під управлінням Linux, яка підключена до локальної мережі закладу. Для виконання завдання ми використовували стабільну версію Linux Debian 8.0.0.

Debian (Debian GNU/Linux) – комп'ютерна операційна система, основний дистрибутив якої складається тільки з вільного програмного забезпечення (main секції архіва Debian). Популярний та впливовий дистрибутив GNU/Linux. Багатоцільова операційна система, що використовується: настільними комп'ютерами, ноутбуками, серверами, вбудованими системами. Існують проекти на основі ядер: Debian GNU/Hurd, Debian GNU/kFreeBSD, Debian GNU/kNetBSD.

В якості доповнювальної послуги дозволяється встановлювати deb-пакунки, які не були включені у головний репозиторій, по причині надто обмеженої ліцензії або можливих законодавчих непорозумінь. А саме:

- додаткова (Contrib) – пакунки у цій частині розповсюджуються власником авторського права на умовах вільної ліцензії, але залежать від невільного ПО;
- невільна (Non-Free) – ліцензії пакетів у цій частині мають умови, що обмежують розповсюдження ПЗ.

Для установки і налаштування необхідного для роботи програмного забезпечення необхідно мати підключення до мережі Інтернет[9].

3.1.1 Сервер Icecast2

В першу чергу необхідно скачати і інсталивати пакет програми серверу, яка буде ретранслювати потік у мережу, дозволяючи клієнтам підключатися і прослуховувати радіо. Ми використовуємо Icecast2.

Програма є в репозиторіях Debian, тому інсталиємо її звідти. Викликаємо консоль і, використовуючи права суперкористувача, виконуємо команду:

```
apt-get install icecast2
```

У процесі інсталяції програма створить користувача і групу з ім'ям Iccast. Також користувачу буде запропоновано встановити необхідні паролі. Для зміни налаштувань необхідно редагувати конфігураційний файл, який знаходиться за адресою:

`/etc/icecast2/icecast.xml`

Повний текст цього файлу конфігурації буде приведений у додатку.

Строка `<clients>100</clients>` відповідає за максимальну кількість підключених клієнтів до серверу Iccast. Ця величина залежить від характеристик серверу та можливостями Інтернет каналу. В данному випадку число 100 значить, що максимальна кількість клієнтів, які можуть підключатися до серверу дорівнює 100 клієнтам. Ця кількість є достатньою для організації радіостанції в локальній мережі.

Блок `<authentication>` містить в собі необхідні паролі для трансляції на сервер з джерела, а також дані для входу в режим адміністратора у web інтерфейсі серверу.

Строка `<hostname> 127.0.0.1 </hostname>` відповідає за адресу серверу у мережі. При значенні 127.0.0.1 радіо можливо прослуховувати тільки з цієї машини, тому що це IP-адреса спеціального мережевого інтерфейсу внутрішньої петлі у мережевому протоколі TCP/IP. У цю строку необхідно вписати IP адресу машини у мережі, до цієї адреси будуть звертатися клієнти, на цю ж адресу буде йти потік з програми джерела.

Важливим параметром є порт. Вказується у строці `<port>8000</port>`. Діапазон портів Iccast зазвичай знаходиться між діапазоном 8000-8999, але можна указати любий вільний порт у системі. Після внесення необхідних змін необхідно відкрити файл за адресою `/etc/default/icecast2`. У цьому файлі вказан шлях до конфігураційного файлу, який ми змінювали, назва користувача та групи, від яких буде запускатися процес, а також опція, яка відповідає за ввімкнення Iccast.

Для того, щоб дозволити запуск, необхідно змінити `ENABLE=false` на `ENABLE=true` і зберегти файл. Після цього запускаємо Iccast за допомогою команди `/etc/init.d/icecast2`. На цьому налаштування Iccast завершено. Iccast має власну панель керування.

Ця панель містить інформацію про трансльовану в даний час радіостанцію. Крім технічної інформації, як бітрейт, адреси дати початку транлювання та інше, панель керування містить інформацію про клієнтів, прослуховую-

чи дану радіостанцію. На рис. 3.1 приведений інтерфейс цієї панелі.

Також цей інтерфейс дозволяє завантажити файл плейлісту, який дозволить запуснути прослуховування на клієнті в мінімум кроків.



Рисунок 3.1 – Інтерфейс панелі керування

Окрім цього можна відключити радіостанцію, чи перенести слухачів на іншу, яка транслюється через цей сервер.

3.1.2 Джерело аудіопотоку Liquidsoap

Наступним кроком нам необхідно встановити програму джерело. Ми використаємо Liquidsoap – мову програмування для аудіо, розроблену спочатку для створення аудіо- і відеопотоків, переданих з вихідного потоку на Icecast сервер. Різниця з будь-якими іншими доступними утилітами – це те, що Liquidsoap інтерпретує окрему скриптову мову, що робить його дуже універсальним і адаптуючим до різних вживанням.

Для інсталювання основного пакету програми, нам необхідно виконати в консолі від імені адміністратора команду:

```
apt-get install liquidsoap
```

Ця програма поширюється за ліцензією GNU GPL, тому у основний пакет не включено підтримку mp3, також багато модулів винесену у окремий інсталяційний пакет.

GPL – приклад сильної копілефт-ліцензії, яка вимагає, щоб усі похідні

роботи були доступні на тих же умовах, що й оригінал. GPL надає одержувачам комп'ютерної програми права відповідно до визначення вільного програмного забезпечення й використовує копілефт, щоб гарантувати, що ці права будуть збережені навіть тоді, коли робота буде значно змінена чи до неї будуть додані будь-які частини.

Для того, щоб встановити підтримку mp3 разом із усіма модулями виконаємо у консолі команду:

```
apt-get install liquidsoap-plugin-all
```

Перш ніж починати налаштування програми, необхідно спланувати режим роботи радіостанції. Ми будемо використовувати наступну схему: музика в назначений час, оповіщення о дзвінках, прямий ефір, заздалегідь записані повідомлення.

Конфігураційний файл, у якому знаходиться код радіостанції, повинен знаходитись у директорії `/etc/liquidsoap`. Ми створюємо файл і заповнюємо його:

```
set("log.file",true)
set("log.stdout",true)
set("log.level",4)
```

Вмикаємо логи, файли яких будуть знаходитися за адресою `/var/logs/liquidsoap set("harbor.bind_addr","0.0.0.0")`.

Вказуємо адресу, з якої можна приймати потік прямого ефіру. У випадку `0.0.0.0`, потік можна приймати з будь якої адреси. `def smooth_add (~delay=0.5,~p=0.2,~normal,~special)`. Після йде функція, яка дозволяє накладати заздалегідь записані повідомлення в музику, знизивши її гучність.

Необхідно задати шляхи, де буде знаходитись музика і повідомлення. Для `liquidsoap` можна вказувати як окремий файл, так і папку з музикою, чи окремий файл плейлісту:

```
radio_pl= playlist(reload=200, "/home/test/liquidsoap/radio")
```

В данному випадку музика знаходиться в папці `radio`, `reload=200` значить, що кожні 200 секунд плейліст буде обновлюватися. Це дозволяє доповнювати музикальні файли без необхідності перезапуску джерела.

```
radio30 = switch(track_sensitive=false, [({11h31m - 11h54m59s},
radio_pl]))
```

Дана строка вказує час, у який буде грати заданий плейліст `radio_pl`. Параметр `track_sensitive=false` значить, буде враховуватися при зміні плейлістів факт – дограла пісня до кінця чи ні. В даному випадку музика буде зупинятися в заданий час.

```
radio=smooth_add(normal=radio30, special=switch([(0m0s or 20m0s or
40m0s}, warning))))
```

Вставляємо в музику записані повідомлення, використовуючи використану раніше функцію `smooth_add`. В данному випадку повідомлення буде програватися кожні 20 хвилин. Тепер необхідно задати час програвання повідомлень. Функція не відрізняється від того, що ми використовували в заданні часу програвання музики, окрім того, що параметр `track_sensitive` буде дорівнювати `true`, тобто повідомлення програватиметься до кінця.

```
live = input.harbor("test-harbor", port=8001, password="hackme")
```

Тут ми вказуємо точку входу для прямого ефіру. `Liquidsoap` може приймати як готовий потік, який йде з `Icecast`, та і потік, який призначений для `Icecast`. Для організації прямого ефіру з комп'ютера під керуванням операційної системи `Windows` ми використаємо безкоштовну програму `edcastStandalone`.

```
f_radio=fallback(track_sensitive=false, [live, radio, time_ring, blank()])
```

Тепер ми формуємо ефірний потік. Формування ефірного потоку виробляється шляхом вибудовування спеціального джерела, який завжди повинен щось відтворювати і використовувати принцип низхідного пріоритету черги. Параметр `fallback` вказує на низхідний пріоритет, тобто джерело `live` має вище пріоритет ніж `radio`.

```
radio = smart_crossfade(f_radio)
```

Використовуємо кросфейд для ефірного потоку. Кросфейд – це музич-

ний прийом, коли два фрагменти композиції частково перекривають один одного, тобто один якийсь час звучить на тлі іншого, повільно затихаючи.

```
output.icecast (%mp3(bitrate=192), host = "192.168.1.200", port = 8000,
mount="test", radio)
```

Готовий ефірний потік виводимо на Icecast. Вказуємо необхідні параметри, в данному разі виводимо потік у форматі mp3 з бітрейтом 192. Хост і порт вказують на адресу серверу Icecast, mount вказує на назву даної радіостанції на сервері Icecast.

Після цього налаштування liquidsoap буде завершено.

3.2 Зовнішнє проектування

Основне місце ведення трансляції радіостанції – в будинку. Для цього нам необхідно визначити необхідну кількість клієнтів та метод їх підключення до серверу.

Для підключення клієнтів до серверу ми використаємо локальну мережу. Локальна мережа – це група з декількох комп'ютерів, з'єднаних за допомогою кабелів (іноді також телефонних ліній або радіоканалів), які використовуються для передачі інформації між комп'ютерами. Для з'єднання комп'ютерів в локальну мережу необхідно мережеве обладнання і програмне забезпечення.

Для охоплення всього закладу в локальну мережу, на кожному поверху знаходяться декілька маршрутизаторів фірми TP-LINK. Маршрутизатор – електронний пристрій, що використовується для поєднання двох або більше мереж і керує процесом маршрутизації, тобто на підставі інформації про топологію мережі та певних правил приймає рішення про пересилання пакетів мережевого рівня (рівень 3 моделі OSI) між різними сегментами мережі. Маршрутизатори працюють на мережному рівні моделі OSI: можуть пересилати пакети з однієї мережі до іншої. Для того, щоб надіслати пакети в потрібному напрямку, маршрутизатор використовує таблицю маршрутизації, що зберігається у пам'яті. Таблиця маршрутизації може складатися засобами статичної або динамічної маршрутизації. Крім того, маршрутизатори можуть здійснювати трансляцію адреси відправника й одержувача (англ. NAT, Network Address Translation), фільтрацію транзитного потоку даних на основі певних правил з метою обмеження доступу, шифрування чи дешифрування передаваних даних тощо. Маршрутизатори не можуть здійснювати передачу широкомовних повідомлень, таких як ARP-запит. Переваги маршрутизаторів даної

фірми в стабільності роботи та зручному інтерфейсі, що полегшує його налаштування і включення до загальної мережі.

Підключення маршрутизаторів до серверу здійснюється за допомогою стандарту 100BASE-T. Це стандарт FastEthernet, який використовує для передачі виту пару, обтиснену за методом TIA/EIA-568B. Швидкість такого з'єднання достатня для взаємодії із сервером.

Локальна мережа використовує змішану (гібридну) топологію. Змішана топологія – топологія, що переважає в великих мережах з довільними зв'язками між комп'ютерами. У таких мережах можна виділити окремі довільно пов'язані фрагменти, що мають типову топологію, тому їх називають мережами зі змішаною топологією.

Клієнтом для радіостанції може виступати будь-який комп'ютерний пристрій, маючий мережеву операційну систему.

Мережева операційна система – операційна система з вбудованими можливостями для роботи в комп'ютерних мережах. До таких можливостей можна віднести:

- підтримку мережевого обладнання;
- підтримку мережевих протоколів;
- підтримку протоколів маршрутизації;
- підтримку фільтрації зв'язку;
- підтримку доступу до віддалених ресурсів, таких як принтери, диски і т. п. по мережі;
- підтримку мережевих протоколів авторизації;
- наявність в системі мережних служб, які дозволяють віддаленим користувачам використовувати ресурси комп'ютера.

Майже всі сучасні операційні системи є мережевими, тобто у якості клієнта можна використовувати персональний комп'ютер із встановленою Windows XP і вище або ОС на основі Linux, сучасний смартфон на базі Android, IOS, Windows Phone, мікрокомп'ютери, наприклад Raspberry Pi із встановленою спеціальною ОС: Raspbian, Pidora, OpenELEC та інші, спеціальні пристрої для програвання мережевого радіо, наприклад Logitech Squeezebox Radio.

У нашому випадку візьмемо в якості клієнтів персональні комп'ютери із встановленою операційною системою Windows.

Комп'ютер повинен мати мережеву карту для підключення до локальної мережі і звукову карту для виведення звуку. У якості програвача використаємо програму Winamp. Це універсальний програвач медіа файлів, на да-

ний момент проект більше не розвивається і припинив підтримку.

Після встановлення і налаштування операційної системи та Winamp, необхідно прописати програвач в автозавантаження. Для цього створюємо ярлик на файл програми із параметром «`http://nnn.nnn.nnn.nnn:XXXX/mount`» і копіюємо його до папки автозавантаження. Також у головному вікні програвача необхідно ввімкнути функцію повторення, для того, щоб система автоматично пере підключалася до радіостанції у разі втрати з'єднання. Після виконання цих маніпуляцій клієнт працюватиме автономно.

Із-за буферизації потоку неможливо синхронізувати звучання декількох клієнтів, тому клієнти необхідно розмістити так, щоб один клієнт не перекривав звучання іншого. Дана система гнучка, тому при наявності можливості можна організувати клієнт у кожному кабінеті. Оптимальним буде розташувати клієнти біля маршрутизаторів. Такі клієнти будуть приймати з сервера трансляцію, яка містить лише оповіщення про дзвінки та прямий ефір. В вільний від ефіру час система транслюватиме тишу, імітуючи процес очікування. Другий тип трансляції, який містить у собі музику у назначений час та заздалегідь написані повідомлення, транслюватиметься на клієнти, які знаходяться у місцях скупчень народу.

При налаштуванні клієнтів рекомендується задати їм заздалегідь встановлені IP адреси. Це дозволить контролювати кожен клієнт за допомогою веб-інтерфейсу Icecast. У випадку, коли необхідно ввімкнути музику на всіх клієнтах(наприклад день відкритих дверей), достатньо у веб-інтерфейсі перекинути слухачів з однієї точки на іншу.

3.3 Трансляція з мікрофону

Радіостанція має можливість ведення прямого ефіру. Для цього ми будемо використовувати персональний комп'ютер під управлінням Windows. Цей комп'ютер повинен знаходитися у тій же мережі, де і сервер. Для трансляції звуку використаємо програму edcast Standalone. Це проста безкоштовна програма написана на мові Delphi з використанням бібліотеки BASS. Основне призначення даної програми, це передати звук з мікрофону до серверу мережевого радіомовлення.

Інсталюємо програму, та завантажимо бібліотеку для кодування аудіо потоку в mp3.

Необхідно налаштувати програму. Натискаємо “Add Encoder” і натискаємо два рази на створений елемент. Відкривається вікно налаштування.

Вибераємо спосіб кодування потоку в mp3, для цього змінимо “Encoder Type” на “MP3 Lame”. Після цього встановлюємо параметри кодування, такі як бітрейт та кількість каналів. Для передачі потоку в Liquidsoap, необхідно налаштувати як для передачі в Icast. Встановлюємо параметри, заздалегідь налаштовані в Liquidsoap. Після цього налаштування завершується. На рис. 3.2 приведений інтерфейс цього програмного продукту.

Для запуску трансляції вибираємо підключений мікрофон із списку і натискаємо “Connect”. Звук буде передаватися у програму джерело з затримкою. Для завершення трансляції необхідно натиснути “Disconnect”.

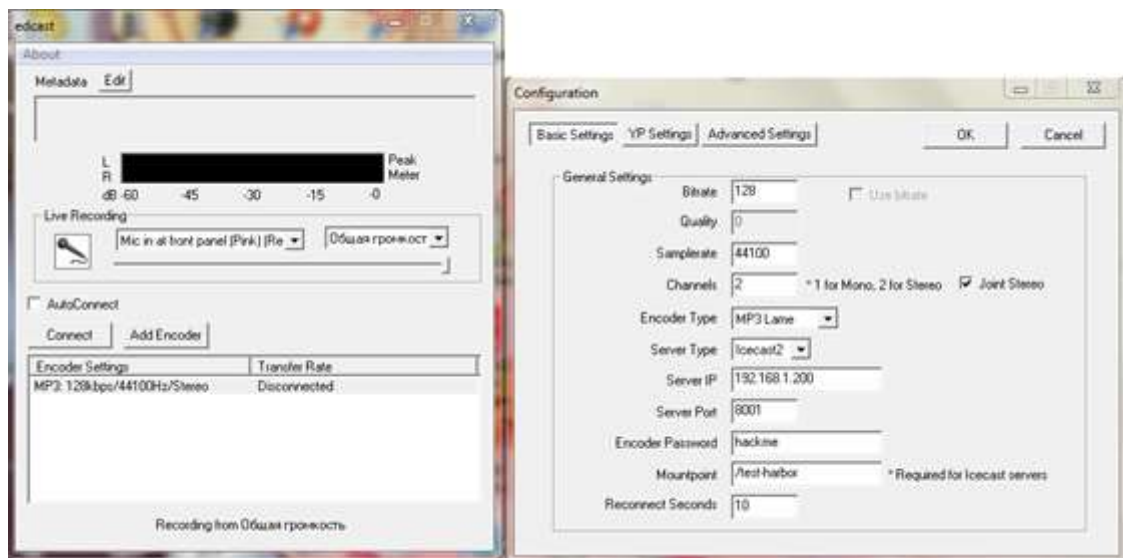


Рисунок 3.2 – Інтерфейс програми Edcast

3.4 Сайт радіостанції

3.4.1 Налаштування веб-серверу

Для встановлення веб-сервера Apache потрібно виконати команду:

```
apt-get install apache2
```

Після цього відкрийте в браузері <http://192.168.0.100>, і на екрані буде відображатися стандартна сторінка з написом "It works!".

За замовчуванням в веб-сервері Apache коренева директорія для сайтів (document root) розташована в директорії / var / www / html. Конфігураційний файл (файл з настройками) веб-сервера Apache знаходиться в файлі

etc/apache2/apache2.conf

Більш детальна документація по налаштуванню знаходиться в файлі /usr/share/doc/apache2/README.Debian.gz.

Для встановлення PHP необхідно виконати команду:

```
apt-get install php5 libapache2-mod-php5
```

Після установки потрібно перезавантажити веб-сервер apache:

```
service apache2 restart
```

На рис. 3.3 продемонстрована стандартна сторінка.

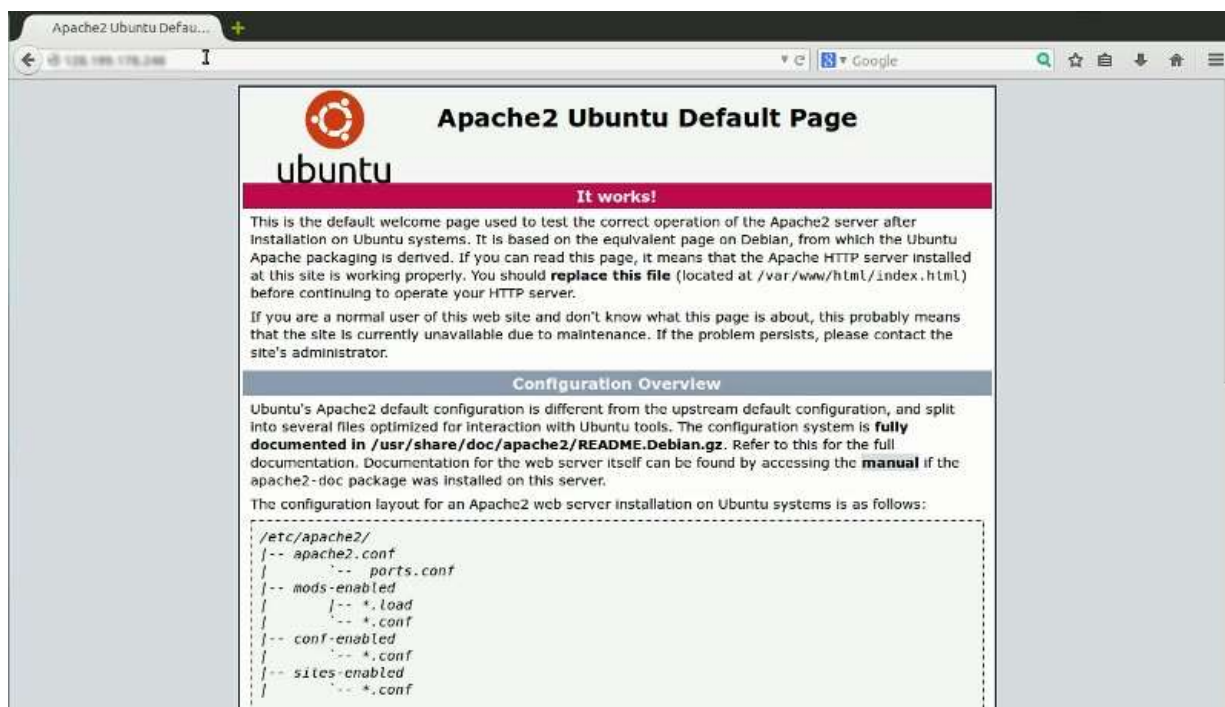


Рисунок 3.3 – Стандартна сторінка веб-серверу Apache

Для встановлення веб-фреймворку Django, нам необхідно встановити Python. Після його встановлення і налаштування виконуємо у консолі:

```
pip install django
```

Тепер необхідно створити проект. Структура Django складається з проекту, який містить у собі додатки. У консолі переходимо до каталога, де ми створимо проект і виконаємо команду

django-admin startproject mysite

Це створить каталог з проектом. Для керування проектом, в тому числі і створення додатків, використовується `manage.py`. Це утиліта командного рядка, яка дозволяє взаємодіяти з цим проектом Django різними способами.

Виконаємо команду

```
manage.py startapp radio
```

де `radio` – назва додатку. Для підключення додатку до проекту необхідно у файлі `settings.py` додати його до масиву `INSTALLED_APPS`. Django містить статичні файли, такі як CSS, JS у директорії `static`. Для задання цієї директорії у файлі налаштування необхідно додати

```
STATIC_URL = '/static/'
```

Для html файлім у каталозі проекту створюємо каталог `templates`. Для перевірки працездатності проекту необхідно запустити його за допомогою команди

```
manage.py runserver 127.0.0.1:8000
```

При цьому за вказаною адресою повинна відображатися стартова сторінка Django проекту (рис. 3.4).

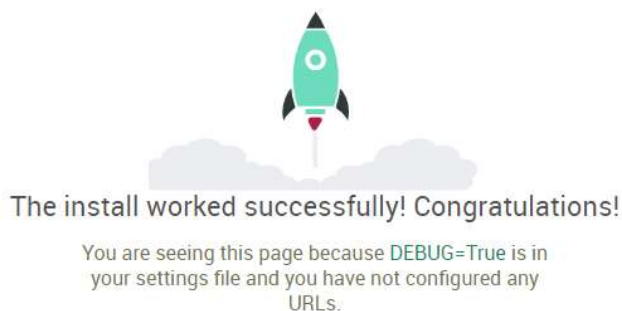
Для того, щоб Apache працював з Django, необхідно встановити `mod_wsgi`. Пакет `mod_wsgi` реалізує WSGI-сумісний інтерфейс для розміщення веб-додатків на базі Python на веб-сервері Apache. Після того, як ви встановите та активізуєте `mod_wsgi`, відредагуйте файл `httpd.conf` веб-сервера Apache, змінивши його наступним чином[10].

```
WSGIScriptAlias /path/to/mysite.com/mysite/wsgi.py
WSGIProxyPath /path/to/mysite.com
<Directory /path/to/mysite.com/mysite>
<Files wsgi.py>
Order deny,allow
Require all granted
```

```
</Files>
</Directory>
```

django

[View release notes for Django 2.1](#)



Django Documentation
Topics, references, & how-to's



Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

Рисунок 3.4 – Перевірка працездатності Django

3.4.2 Створення веб-сторінки для замовлення музики

Створимо веб-сторінку для представлення інтернет радіо Її можна розділити на дві частини: панель адміністратора і сторінка користувача. Хоча Icecast надає своя панель адміністратора, нам необхідна своя динамічна та розширювана панель адміністратора.

Для початку реалізуємо отримання статусу інтернет радіостанції. Необхідно отримати інформацію у вигляді json об'єкта. Він доступний за адресою

`http://адреса_icecast/status-json.xsl,`

де адреса_icecast – url адреса серверу icecast. Інформація доступна без авторизації. Оскільки інформація часто оновлюється, отримуємо її у момент завантаження сторінки за допомогою jquery

```
getJSON('http://192.168.1.207:8000/status-json.xsl'
```

Об'єкт містить як загальну інформацію о сервері, таку як ім'я та час початку роботи, так і інформацію об усіх точках трансляції, у тому числі їх ім'я, адресу та назву поточної композиції. З об'єктом можна працювати як з масивом, тому, наприклад, для отримання часу початку роботи сервера достатньо звернутися до об'єкта `server_start` таким чином:

```
data['icestats']['server_start'].
```

Ще однією функцією панелі адміністратора є додавання пісні до черги. Для цього необхідно за допомогою Django серверу отримати список доступних композицій. Скануємо директорію з музикою і відображуємо при завантаженні сторінки замовлень. Для відправки команди на сервер використовуємо telnet підключення. Telnet не шифрується, тому, для забезпечення безпеки, у файлі конфігурації `liquidsoap` додаємо строку `set("server.telnet.bind_addr", "localhost")`, яка дозволяє підключатися лише з одного ip-адресу. Задаємо адресу, на якому розташований скрипт заказу музики.

Для з'єднання з сервером telnet у python необхідно використати модуль `telnetlib`. Цей модуль відкриває з'єднання з інтернет сокетом або доменним сокетом Unix. Функція приймає адресу серверу, порт. Відправляємо до джерела команду `request.push` і повний шлях до замовленої музики. Наприклад:

```
request.push /home/test/liquidsoap/radio/test.mp3
```

Не забудемо закрити з'єднання командою `quit`. Після цього наступною у плейлісті буде програватися замовлена музика.

Важливою функцією у панелі адміністратора є можливість підключитися і вести пряму трансляцію напряму з інтерфейсу сайту. Зазвичай для прямих трансляцій використовують професійні і зачату платні програми, які можуть обробляти звук у реальному часі і надають додаткові інструменти, такі як програвання семплів, зміна голосу, мікшер і тд.

Для забезпечення цієї функціональності використовуємо javascript бібліотеку `Webcast.js`. Це бібліотека веб-трансляції призначена для написання браузерних клієнтів для потокового передавання локальних файлів та медіаданих (веб-камера, аудіо мікрофона). З її допомогою можна не тільки вести пряму трансляцію, але й організувати сервер радіостанції у браузері, який буде за-

міняти потік основної радіостанції. Інтерфейс представлений на рис. 3.5.

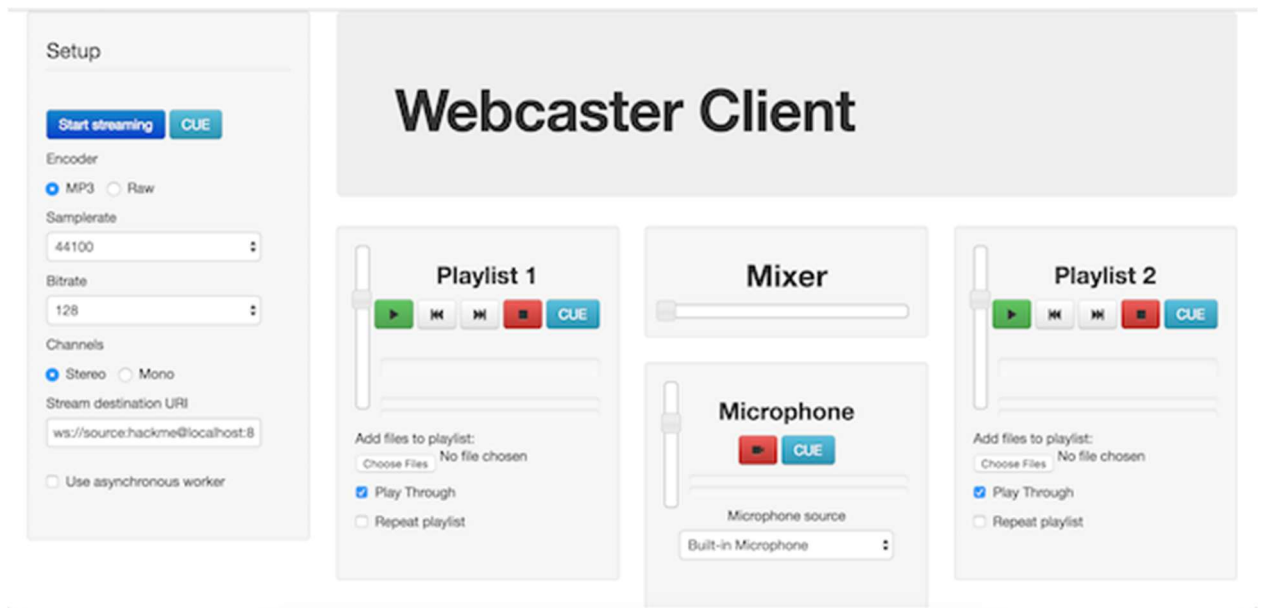


Рисунок 3.5 – Інтерфейс Webcast.js

Для сторінки користувача реалізуємо перегляд статусу аналогічно панелі адміністратора. Щоб уникнути колізій, замовлення музики необхідно пустити замовлення через Django. Для цього використаємо модуль `django-queue-manager`, який дозволить створювати чергу пісень, з якої буде братися композиція і відправлятися telnet командою до радіостанції.

Реалізуємо доступ до прослуховування радіостанції двома шляхами. По-перше, це скачування плейлісту у вигляді файлу `m3u`. Цей формат є стандартом і його розуміють більшість програвачів. По-друге реалізуємо онлайн програвання на сайті. Для цього необхідно створити інтерфейс плеєру на сайті. Найпростіший плеєр створюється за допомогою тега `<audio>` у HTML5. Окрім цього, для початку трансляції необхідно відправити заголовок `Accept`.

```
$http({
  method: 'GET',
  url: 'https://radio.burgerrecords.com:8001/status-json.xml',
  headers: {
    'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
    'Origin': undefined })
```

4 ПРОСУВАННЯ САЙТУ

Сайт, який представляє якусь організацію, потребує користувачів. Користувачі забезпечують популярність організації, що спроваджує задоволенню її цілей. Незалежно від якості та наповненості, сайт без користувачів некорисний. Тому важливу роль у сучасному інтернеті грає просування сайту.

Просування сайту – це комплекс заходів, які забезпечують відвідуваність сайту користувачами. Ці заходи охоплюють багато каналів залучення користувачів, але не обов'язково охоплювати їх усі. Вибір необхідної стратегії залежить від цільової аудиторії, на яку спрямовано контент сайту.

Сайт інтернет радіо учбового закладу є некомерційним, тому йому підходять пасивні методи розповсюдження, які не потребують великий вклад фінансових коштів. Цільовою аудиторією сайту є студенти, абітурієнти, а також люди, які шукають додаткову освіту, тому основний напрямок розповсюдження – це реклама освітніх послуг закладу.

Одним із найпоширеніших каналів просування сайту є SEO (search engine optimization) – пошукова оптимізація. Це комплекс заходів по внутрішньої і зовнішньої оптимізації для підняття позицій сайту в результатах видачі пошукових систем за певними запитами користувачів.

Пошукова система враховує такі параметри сайту при обчисленні його релевантності:

- частота ключових слів: складні алгоритми сучасних пошукачів дозволяють проводити семантичний аналіз тексту, щоб відсіяти пошуковий спам, коли ключове слово зустрічається дуже часто (більше 7-8 відсотків від всього вмісту) або дуже рідко (1-3 %) на сторінці. Оптимальна щільність ключових слів: 5-7 % від текстового вмісту веб-сторінки, при цьому ключові слова (фрази) повинні зустрічатися в тексті як мінімум 3-4 рази;
- індекс цитування сайту, або кількість веб-ресурсів, що посилаються на даний сайт; багатьма пошукачами не враховуються взаємні посилання (один на одного), також важливо, щоб посилання були з сайтів схожої тематики, що і розкручуваний (оптимізовуваний) сайт.

Відповідно, всі чинники, що впливають на положення сайту у видачі пошукової системи, можна розбити на зовнішні і внутрішні. Робота по оптимізації включає роботу:

- з внутрішніми чинниками які знаходяться під контролем власника веб-сайту) – приведення тексту і розмітки сторінок у відповідність з

вибраними запитами, поліпшення якості і кількості тексту на сайті, стилістичне оформлення тексту (заголовки, жирний шрифт), поліпшення структури і навігації, використання внутрішніх посилань;

- зовнішніми чинниками – обмін посиланнями, реєстрація в каталогах та інші заходи для підвищення і стимулювання кількості і частоти посилання на ресурс.

Пошукові системи організують дані, які вони збирають, відповідно до критеріїв індексації та критеріїв рейтингу. Критерії індексації визначають інформацію, яку буде зберігати пошукова система, і як ця інформація буде оброблятися. Приклади критеріїв індексації включають в себе: індексування лише сторінок, які проходять один або декілька тестів якості, пропущені частини вмісту сторінки тощо. Приклади критеріїв рейтингу включають: надання додаткової ваги для використання термінів запиту у заголовках, URL-адресах сторінок та жирним шрифтом; надання кредиту за використання термінів запиту в текстовому каналі зв'язку, що вказує на сторінку; тощо[11].

Пошукова система впливає на результати, які вона обслуговує користувачам, коригуючи або критерії індексації, або критерії рейтингу, або обидва. Разом з тим інші фактори можуть вплинути на результати пошукової системи. Наприклад, пошукова система, яка поширює свою базу даних в різних центрах обробки даних, може не завжди мати синхронних результатів у своїй мережі. А алгоритми балансування завантаження пошукової системи можуть призвести до того, що деякі завдання вирішення запитів закінчуються передчасно, перш ніж оцінювати всі доступні відповідні результати в періоди високого трафіку.

З точки зору пошукової системи, вміст результатів пошуку визначається інфраструктурою, критеріями індексації та критеріями рейтингу. Метою пошукової системи (в ідеалі) є надання найбільш доступних та задовольняючих результатів для своїх користувачів. Результати не можуть бути просто актуальними, вони повинні бути задовільними, щоб користувачі відчували впевненість у використанні служби пошукової системи. Звичайно, задовольняючі результати повинні бути релевантними, але вони не повинні бути найбільш відповідними, ані навіть найбільш авторитетними чи вичерпними результатами. Тому пошукова система вимірює якість результатів пошуку відповідно до задоволеності користувачів, а не за об'єктивними критеріями.

Маючи вибір між пошуковою системою Девід та пошуковою системою Едвард, користувач буде вибирати одну з двох на основі якості результатів пошуку (з точки зору користувача) або швидкості доставки результатів, якщо

помітна якість результатів приблизно рівний. Тобто чим менш експертом є користувач у темі, тим менша ймовірність того, що користувач буде розрізняти якість вмісту, який надає пошукова система. Отже, час вирішення запитів буде більш важливим, ніж деталі вмісту, якщо обидва пошукові системи надають приблизно однакове значення у своїх результатах.

Якщо Девід надає більш швидкий процес обробки запитів, більша кількість користувачів, як правило, використовує Девід, ніж Едвард. Однак, якщо Едвард значно покращить помітну якість своїх результатів за результатами Давида, користувачі будуть схильні використовувати Едвард більше, ніж Давид, незважаючи на різницю у часі вирішення запитів. Користувачі хочуть витратити якнайменше часу на пошук, але на практиці більшість людей відчувають втому від пошуку (або втоми пошукової системи) на певному рівні.

Це означає, що користувач має повноваження оптимізувати свій пошуковий досвід, змінюючи запити або змінюючи пошукові системи (і користувачі фактично роблять обидва).

Пошукова оптимізація відбувається трьома способами: через зміни, внесені пошуковою системою, через зміни, внесені пошуковим записом, і через зміни, внесені контент-провайдерами. Тому, говорячи про оптимізацію пошукової системи, треба дивитися на все, що робить кожен спосіб. Існує чотири причини, через які змінюються результати пошуку:

- 1) ви щось робите зі своїм сайтом;
- 2) хтось інше робить щось із своїм сайтом;
- 3) пошукові системи роблять щось із своїми даними;
- 4) люди шукають щось інше.

Останню причину можна сприймати як "Люди змінюють спосіб пошуку" – це означає, що може змінюватися те, що вони шукають, або де вони шукають, або коли вони шукають. SearchScare не є основним ресурсом. Вона розвивається по годинно, якщо не частіше. Пошукові системи, доступні вам сьогодні вранці, відрізняються від пошукових систем, доступних вам сьогодні ввечері.

Дієвим способом дізнатися, чи оптимально сайт працює з користувачами є веб аналітика. Веб-аналітика – це вимірювання, збір, аналіз та звітність веб-даних з метою розуміння та оптимізації використання веб-сайтів. Тим не менше, веб-аналітика – це не просто процес вимірювання веб-трафіку, але його можна використовувати як інструмент для дослідження бізнесу та ринку, а також для оцінки та підвищення ефективності веб-сайту. Веб-аналітичні додатки також можуть допомогти компаніям вимірювати результати тради-

ційних друкованих або транслюючих рекламних кампаній. Це допомагає оцінити, як змінюється трафік на веб-сайті після запуску нової рекламної кампанії. Веб-аналітика надає інформацію про кількість відвідувачів веб-сайту та кількість переглядів сторінок. Це допомагає оцінити тенденції трафіку та популярності, що є корисним для дослідження ринку[12]. Однією з найпопулярніших безкоштовних веб-аналітик є Google Analytics

Google Analytics – це абсолютно безкоштовний інструмент для збору веб-аналітики, наданий Google, який допомагає вимірювати трафік веб-сайту та збирати важливу інформацію про відвідувачів вашого веб-сайту. Цей інструмент може дати вам відповіді на запитання звідки відвідувачі вашого веб-сайту, звідки вони приходять, які сторінки вони відвідують, скільки вони залишаються на вашому сайті та чому і коли вони залишають ваш сайт, а також, які сторінки є найпопулярнішими, і, таким чином, дозволить дізнатись, як ваш веб-сайт дійсно працює.

Коли ви знаєте, що працює, а що не працює, ви точно знатимете, які кроки вам потрібно зробити, щоб оптимізувати ваш веб-сайт і кожен вміст, який ви надаєте, з метою залучення відвідувачів та надання їм інформації вони повинні.

Google Analytics пропонується в двох додаткових версіях: Google Analytics на основі підписки 360, орієнтована на корпоративних користувачів і Google Analytics для мобільних додатків, SDK, який дозволяє збирати дані про використання з iOS та додатків Android

Оскільки Google Analytics надає чітке уявлення про те, які зміни вам потрібно внести для покращення вашого сайту та збільшення кількості конверсій, то зрозуміло, що ваш наступний крок на шляху до вдосконалення вашого сайту обов'язково повинен використовувати цей інструмент.

Це абсолютно безкоштовно, і, з огляду на інформацію, яку вона може надати вам, ви можете бути впевнені, що це допоможе вам зробити кожен онлайн-маркетингову кампанію реальними перевагами, які призведуть до покращення вашої суті справи. Приклад інтерфейсу зображений на рис 4.1. Зручність і простота використання інтерфейсу інструменту веб-аналітики безумовно одна з найважливіших складових при щоденній роботі. Інтерфейс експериментів Analytics включає в себе три основних розділа: мастер настройки, список експериментів и отдельные отчеты по каждому из них.

Данные об эксперименте также можно просмотреть в представлении Analytics. Сервіс інтегрований з Google AdWords.

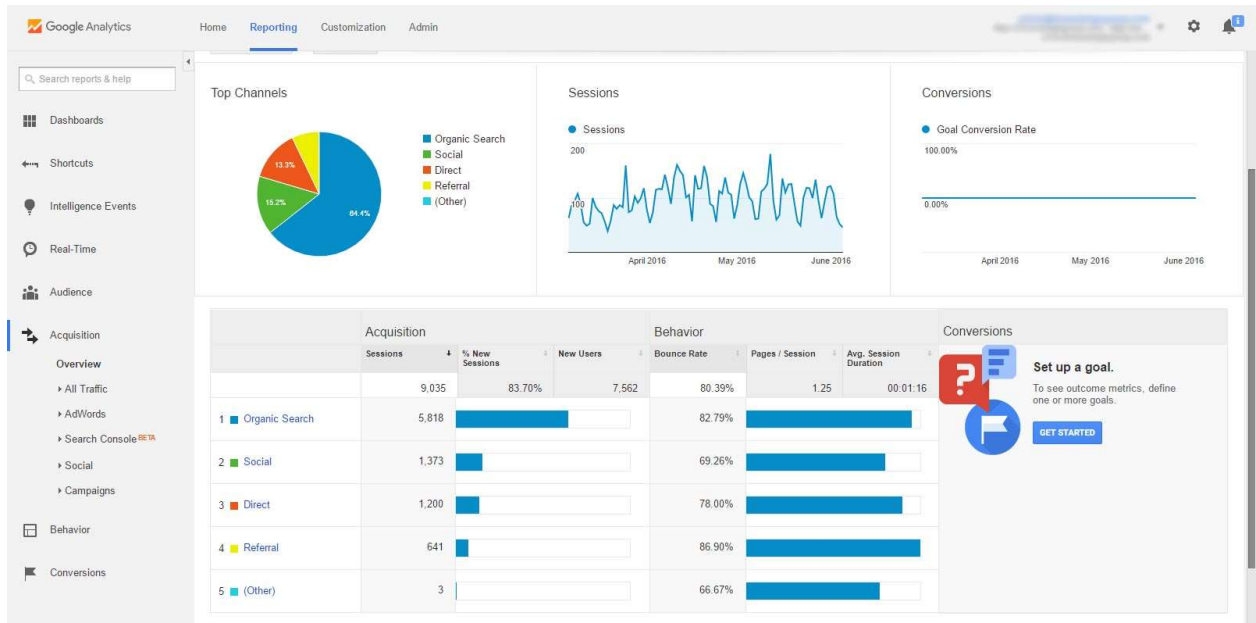


Рисунок 4.1 – Інтерфейс Google AdWords

Особливістю сервісу є те, що веб-майстер може оптимізувати рекламні та маркетингові кампанії Google AdWords за допомогою аналізу даних, отриманих за допомогою сервісу Google Analytics, про те звідки приходять відвідувачі, як довго вони залишаються на сайті і де вони знаходяться географічно. Користувачі бачать групи оголошень і віддачу від ключових слів в звітах. Також доступні додаткові можливості включаючи поділ відвідувачів на групи.

Реклама являє собою ефективний канал просування сайту. Хоча сайт є некомерційним, невеликий вклад в рекламу спроможний підняти популярність учбового закладу. Можна виділити декілька типів реклами.

Контекстна реклама показується користувачам на основі їх інтересів. Це гарантує, що основна частина людей, які побачили цю рекламу, є потенційними користувачами сайту. Для виявлення інтересів користувачів на сьогоднішній день використовуються рекламні мережі, наприклад Google AdWords. Також, окрім інтересів, можна вибрати тематику сайтів для показу. Таким чином, вибравши сайти для підготовки до здачі шкільних іспитів і сайти для абітурієнтів, отримаємо широке охоплення аудиторії. Представлення рекламного повідомлення повинно зацікавити цільову аудиторію.

Ads (раніше відомий як AdWords) – сервіс контекстної, в основному, пошукової реклами від компанії Google, що надає зручний інтерфейс і безліч інструментів для створення ефективних рекламних повідомлень. Ads – флаг-

манський рекламний проект Google і основне джерело доходів компанії. 24 липня бренд AdWords змінив назву на Google Ads.

Створення стандартного рекламного оголошення в Google Ads займає, як правило, менше 30 хвилин. Процес розміщення реклами відбувається в 8 кроків.

У першу чергу рекламодавцю необхідно створити обліковий запис в Google Ads, для чого достатньо вказати адресу електронної пошти та пароль, які використовуються для доступу до інших сервісів Google. При відсутності таких можна зареєструвати аккаунт Ads на будь-який інший адресу електронної пошти.

Потім рекламодавцю пропонується вибрати цільову аудиторію (тобто зробити таргетинг) шляхом вибору місця розташування (країна, регіон, місто), мови потенційних клієнтів, а також демографічних показників (вік і стать).

На третьому етапі необхідно вказати бюджет рекламної кампанії.

На четвертому встановлюється ціна за клік по рекламному повідомленню. При виборі рекомендованого варіанту «Автоматичне встановлення ставок» вказується верхня межа ціни за клік, наприклад, 10 руб., А система автоматично підбирає мінімально необхідну ціну за клік для показу рекламного оголошення на кращій в кожному конкретному випадку позиції. При цьому фактична ціна за переходи в більшості випадків менше максимальної ціни і ніколи її не перевищує з групи, виявляючи при цьому найбільш ефективні.

П'ятий етап присвячений створенню безпосередньо рекламного оголошення, для чого необхідно ввести 2 заголовка (по 30 символів кожен), коротко описати рекламований продукт в рядку опису (90 символів) і вказати адресу веб-сторінки, на яку перейде користувач, що натиснув на контекстне оголошення. Внести зміни в текст оголошення можливо в будь-який момент.

На шостому етапі рекламодавцю пропонується вибрати ключові слова, при використанні яких в пошуковому запиті користувача буде показано розміщується оголошення. Пропонований на цьому етапі інструмент добору ключових слів дозволяє підібрати ключові слова, перевірити їх популярність (кількість запитів користувачів по ним) і дізнатися рівень конкуренції за даними словами серед рекламодавців.

Потім можна створити додаткові варіанти оголошень, з яких згодом Ads емпіричним шляхом відбере найбільш ефективні.

На заключному етапі рекламодавцю необхідно вибрати спосіб оплати і ввести платіжні дані. Серед способів оплати є: банківський переказ, кредитні

карти Visa і MasterCard, Webmoney.

Визначивши, на кого буде спрямована реклама і вибравши відповідні ключові слова, веб-видавець створює текстові оголошення, які інформують його потенційних клієнтів (цільову аудиторію) про пропоновані їм продукти чи послуги. Для створення рекламного повідомлення існують певні редакційні правила Google Ads, які стосуються опису сайту в рекламному повідомленні і унікальною вигоді пропонованого продукту або послуги. Всі оголошення проходять попередню модерацію.

Рекламні оголошення, що розміщуються в системі Ads, короткі і ідентичні за структурою: назва-посилання, адреса сайту і текстовий блок розміром не більше трьох рядків. Перший рядок допускає до 25 символів, друга і третя – по 35 символів. 4-й рядок – відображає цільової URL. При націленні на деякі країни Східної Європи і Азії (включаючи Росію і Україну) дозволяється використовувати подовжені текстові оголошення. Тема – 30 символів, друга і третя рядок – по 38 символів.

По завершенні перевірки оголошення модератором проводиться оплата відповідно до вимог Google Ads, і інтернет-реклама з'являється в мережі[13].

Одним із способів зекономити є обмін рекламою з компанією, тематика якої схожа, в випадку учбового закладу це тематика, направлена на учнів старшої школи та абітурієнтів. Кількість користувачів у такої компанії повинен бути рівнозначним або більшим. Але таке співробітництво несе за собою певні ризики. Якщо у партнера виникне конфлікт з їх користувачами, які прийшли по рекламному оголошенню, компанія, яка розмістила це оголошення несе репутаційні ризики.

Контент маркетинг – сукупність маркетингових прийомів, заснованих на створенні і / або розповсюдженні корисною для споживача інформації з метою завоювання довіри і залучення потенційних клієнтів. В випадку інтернет радіо це, по-перше наповнення цікавим контентом радіостанцію, по-друге це супроводження основного контенту додатковим. Наприклад створення ілюстрацій, коміксів та історій, які заохочують прослухати радіостанцію.

Контент самої радіостанції повинен відповідати тематиці закладу, також повинен мати оригінальну частину. У абітурієнта вона повинна викликати інтерес до закладу, а у студента повинна виникати бажання слухати її на постійній основі. Основним оригінальним контентом мають бути передачі, які потім зберігаються на сайті у вигляді подкастів.

Подкаст – епізодична серія цифрових аудіо- та відеофайлів, які корис-

тувач може завантажити, щоб слухати. Він часто доступний для підписки, так що нові епізоди автоматично завантажуються через веб на власний локальний комп'ютер, мобільний додаток або портативний мультимедійний плеєр.

Спочатку це слово було запропоновано Беном Хаммерлі як злиття "iPod" (бренд медіаплеєра) та "трансляція".

Розподілені файли знаходяться в аудіоформаті, але іноді можуть містити інші формати файлів, такі як PDF або EPUB. Відео, які використовуються під час перегляду моделі подкастів, іноді називають відео-подкастами або відеокадрами.

Генератор подкаста підтримує центральний список файлів на сервері як веб-канал, доступ до якого можна отримати через Інтернет. Слухач або переглядач використовує спеціальне клієнтське прикладне програмне забезпечення на комп'ютері або мультимедійному програвачі, який отримує доступ до цього веб-каналу, перевіряє його для оновлень та завантажує нові файли в серії. Цей процес може бути автоматизований, щоб автоматично завантажувати нові файли. Файли локально зберігаються на пристрої користувача, готові до використання в автономному режимі.

Існує безліч різних мобільних програм, доступних людям для підписки та прослуховування подкастів. Багато хто з цих програм дозволяють користувачам завантажувати подкасти або транслювати їх за вимогою як альтернативу завантаженню. Багато плеєрів, які підтримують подкасти (програми, а також спеціальні пристрої), дозволяють слухачам пропускати подкаст та контролювати швидкість відтворення.

Деякі називають подкастинг конвертованим середовищем, що об'єднує аудіо-, веб-та портативні медіапрогравачі, а також як руйнівну технологію, яка змусила деяких осіб у радіо-бізнесі переглянути встановлені практики та уявлення про аудиторію, споживання, виробництво та розподіл. Подкасти, як правило, безкоштовні для слухачів, і їх часто можна створювати за незначні витрати, що відрізняє їх від традиційної моделі засобів захисту інформації та засобів виробництва.

Творці подкастів можуть монетизувати свої подкасти, дозволяючи компаніям купувати час реклами, а також через такі сайти, як Patreon, що надає додаткові послуги та контент для слухачів за окрему плату. Podcasting – це дуже горизонтальна форма: виробники є споживачами, споживачі можуть стати виробниками, і обидва можуть спілкуватися один з одним.

Популярність подкастів значно зросла за останній час. На графіку (рис 4.2) показано дослідження Reuters Digital News Report, де зображено процент

користувачів інтернету у 11 країнах, які за перший квартал 2018 року слухали хоча б один подкаст. На підвищення популярності інтернет радіостанцій і зокрема подкастів впливають багато факторів, основним з яких є широке розповсюдження швидкісного інтернету, медіа пристроїв, які завжди під рукою, таких як смартфони, та дешевого звукового обладнання.

Відповідь на те, чому подкасти стали популярні можна знайти в статті журналу Forbes від 12 листопада: «Однією з причин, з якої подкасти стають настільки популярними, є те, що формат є унікальним, який відповідає нашим зайнятим життям. Будь-яка тема, яка вас цікавить? Ви знайдете подкаст, присвячений цьому. Тільки п'ятнадцять хвилин, щоб заощадити? Ви знайдете подкаст, який ви зможете слухати в той час. Віддавати перевагу історіям? Є подкасти, написані, щоб подряпати цей сюжетний свербіж. Бажаєте новин? Наукову-фантастику? Науку? Політику? Ви знайдете подкаст, який підходить»[14].

Обов'язково створіть профілі в соціальних мережах. Як правило, компанії роблять це відразу після запуску свого веб-сайту. Багато компаній на сьогоднішній день мають представників та представництво у соціальних мережах. Це допомагає їм спілкуватися з користувачами, найчастіше в неформальному стилі.

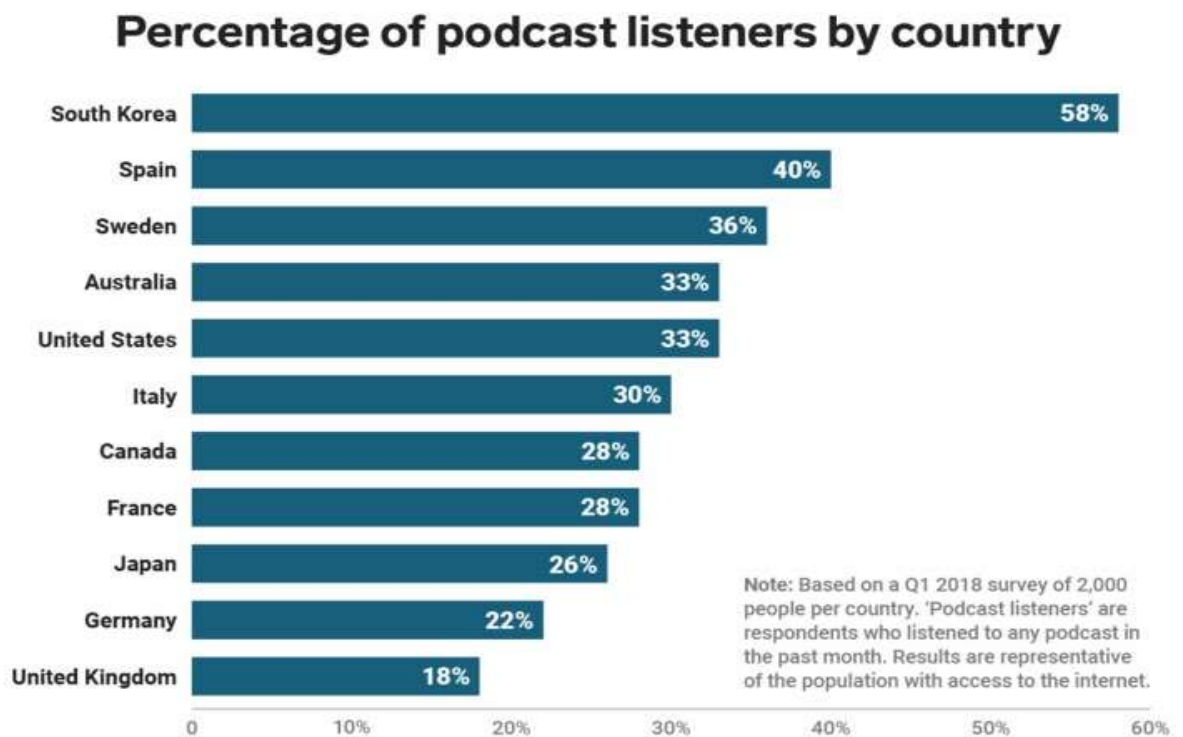


Рисунок 4.2 – Популярність подкастів

Таким чином зростає лояльність користувачів, які мають пряму взаємодію з компанією у комфортній для них обстановці. Ще однією перевагою є те, що сторінки профілів в соціальних мережах мають високі позиції в результатах пошуку. Тому для формування і просування певного бренду це може мати значення.

Такий вид просування сайту називається SMM (Social media marketing) або маркетинг у соціальних мережах.

Маркетинг у соціальних мережах – це використання платформ та веб-сайтів соціальних мереж для просування товару чи послуги. Незважаючи на те, що терміни електронного маркетингу та цифрового маркетингу досі є домінуючими в академічних сферах, маркетинг соціальних мереж стає все більш популярним як для практиків, так і для дослідників. Більшість соціальних медіа-платформ мають вбудовані інструменти аналізу даних, які дозволяють компаніям відслідковувати прогрес, успіх та взаємодію рекламних кампаній. Компанії звертаються до широкого кола зацікавлених сторін через маркетинг соціальних мереж, включаючи поточних та потенційних клієнтів, поточних та потенційних працівників, журналістів, блогерів та широкої громадськості. На стратегічному рівні маркетинг соціальних мереж включає управління маркетинговою кампанією, управління, встановлення масштабу (наприклад, більш активне або пасивне використання) та створення бажаної фірми "культури" та "тону" соціальних мереж.

При використанні маркетингу у соціальних мережах фірми можуть дозволити клієнтам та користувачам Інтернету публікувати вміст, створений користувачами (наприклад, онлайнві коментарі, огляди продуктів тощо), також відомий як "зароблені медіа", а не використовувати маркетингову копію[15].

5 УПРАВЛІННЯ ПРОЕКТАМИ

Управління проектами – це практика ініціювання, планування, виконання, контролю та припинення роботи команди для досягнення конкретних цілей та задоволення певних критеріїв успіху у визначений час. Проект є тимчасовим заходом, призначеним для створення унікального продукту, послуги або результату з певним початком і кінцем (зазвичай обмеженим у часі та часто обмеженим фінансуванням або штатним розписом), що здійснюються для досягнення унікальних цілей і завдань, як правило, для досягнення вигідних змін або додана вартість. Тимчасовий характер проектів стоїть на відміну від звичайного бізнесу (або операцій), що є постійними, або напівпостійними функціональними заходами для виробництва товарів чи послуг. На практиці управління такими відмінними виробничими підходами вимагає розробки чітких технічних навичок та стратегій управління.

Планування IT проекту можна розділити на декілька частин. Перша частина – це постановка цілей. Проекти створені для досягнення конкретних цілей. Необхідно визначити, яку функцію буде виконувати проект, і у якому напрямку він буде розвиватися.

По-друге визначимо задачі проекту. Задача – це те, що потрібно зробити, щоб проект виконував поставлені цілі. Розробку кожної задачі можна представити як процес, який складається з чіткого алгоритму дій. На рис. 5.1 представлено алгоритм виконання процесу. На початку виконання процесу необхідно зібрати інформацію, яка знадобиться при виконанні задачі. На етапі аналізу необхідно обробити отриману інформацію та визначити сильні та слабкі сторони задачі, та можливі шляхи вирішення. Після необхідно спланувати виконання задачі. Це включає в себе створення упорядкованого плану, та створення чеклісту, за допомогою якого буде перевірятися готовий результат.

По-третє визначаємо доступні ресурси. В управлінні проектом необхідні ресурси для виконання завдань проекту. Ресурси можуть бути людьми, обладнанням, засобами, фінансуванням або іншим, яке можна визначити (звичайно, крім праці), необхідних для завершення проектної діяльності. Отже, брак ресурсу буде обмеженим для завершення діяльності проекту. Ресурси можуть бути такими, які зберігатися або не зберігаються.

Ресурси, які зберігаються, залишаються доступними, якщо вони не вичерпані за рахунок використання, і можуть поповнюватися з проектних завдань, які їх створюють.



Рисунок 5.1 – Алгоритм виконання процесу

Ресурси, які не зберігаються, повинні бути поновлені для кожного періоду часу, навіть якщо вони не використовуються в попередні періоди часу. Планування ресурсів, доступність та оптимізація вважаються ключем до успішного управління проектами.

Розподіл обмежених ресурсів базується на пріоритеті, наданому кожній із заходів проекту. Їх пріоритет розраховується за допомогою методу критичного шляху та евристичного аналізу. Для випадку із обмеженням на кіль-

кість ресурсів метою є створення найбільш ефективного графіка – мінімізація тривалості проекту та максимальне використання наявних ресурсів.

Ризик проекту – це ефект, що дозволяє накопичити ймовірності настання ряду подій, які позитивно або негативно вплинуть на цілі самого проекту. Вони діляться на два типи: відомі і невідомі. Як правило, відомі ризики можна розпізнати на початку проекту, що дозволяє ними керувати – створити резервні плани дій, що передбачають можливі втрати. А невідомі ризики визначити заздалегідь не можна, тому спрогнозувати подальші дії неможливо[15].

Мета проекту інтернет радіо – це транслявання радіостанції в мережі. Для забезпечення повного опису доповнимо мету, що трансляція повинна вестися цілодобово, а також покращувати взаємодію студентів з учбовим закладом.

Цю мету слід розбити на декілька задач: створити інтернет станцію і створити сайт для її представлення. Задачі пов'язані одна з одною, але виконуються окремо. Ці задачі і є планом проекту, кожен задачу розбиваємо на підпункти. Створення інтернет станції включає в себе налаштування серверу Icescast, налаштування джерела аудіопотоку. Підпункти є послідовними, неможливо виконувати їх одночасно. Створення сайту включає в себе: написання логіки, створення адміністративного і клієнтського представлення. Ці задачі є паралельними, є можливість виконувати їх одночасно. Схема виконання проекту представлена на рис 5.2.

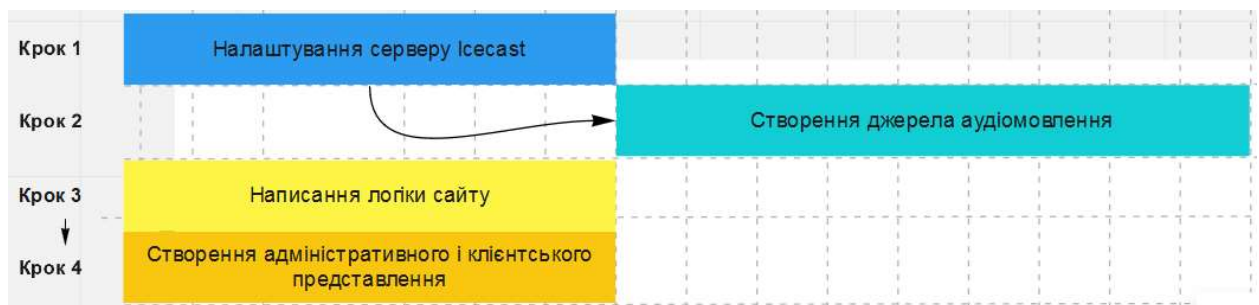


Рисунок 5.2 – Схема виконання проекту

У випадку інтернет радіо відомими ризиками є втрата аудиторії і неспроможність підтримувати якість контенту. У випадку негативної ситуації проект не буде нести суттєвих фінансових втрат. Ніяких гарантій перед користувачами проект не несе, тому припинити підтримку можна у будь який момент. Підтримка проекту з технічної сторони може здійснюватися однією

людиною, яка знайома зі структурою проекту і володіє начальними навичками мови liquidsoap для інтернет радіо, та python і javascript для сайту.

ВИСНОВКИ

В даній роботі ставилась мета спроектувати радіомережу на основі комп'ютерної мережі для подальшого використання в закладі, використовуючи ОС Windows чи Linux. Мета була цілком досягнута та спроектована радіомережа відповідає усім вимогам відмовостійкості.

В ході виконання дипломного проекту були розроблені технічний та робочий проекти системи, її топологія та інформаційна структура. Дану радіомережу було спроектовано на основі ОС Linux, перевагами якого є безкоштовність, відкритість та гнучкість системи.

В ході проектування були використані програма сервер Iccast, джерело Liquidsoap, також для передачі звуку з мікрофону була використана безкоштовна програма Edcast Standalone. Для створення сайту була використана мова програмування Python і фреймворк Django.

Можливості даної радіомережі:

- 1) програвання музики у заданий час, а також система, яка дублює дзвінки, можна встановити будь-яку мелодію на дзвінок;
- 2) система оповіщень, яка дозволяє програти файл із заданої папки не зупиняючи музику, а зменшуючи її гучність і використовуючи її в якості фону;
- 3) передача прямих голосових трансляцій з ОС Windows за допомогою програми Edcast Standalone.
- 4) Керування з панелі адміністратора

У даному дипломному проекті було досягнуто наступних результатів, які були отримані в ході тестування:

- 1) сервер може витримувати велику кількість одночасних слухачів;
- 2) при великому навантаженні на сервер трансляція не переривається.

Функції радіостанції можна змінити чи розширити у будь-який момент, для цього необхідно внести зміни до скриптового файлу. Функціонал панелі адміністратора можна розширити, для цього необхідно внести додаткову логіку до серверу Django.

При розробці даного дипломного проекту були отримані та закріплені знання в різних областях науки і техніки. Було розглянуте і реалізоване широке коло задач, що виникли в ході виконання дипломного проектування. Були враховані всі помилки і неточності, що з'являлися на етапах розробки. Також, при розробці дипломного проекту були розглянуті і вивчені основні принципи проектування в Linux.

ПЕРЕЛІК ПОСИЛАНЬ

1. Білл Росс, Джой Ленскі. Нескінченний набір: Цифрові платформи радіостанції: Пер. с англ. – СПб: Питер, 2008. – 660 с.
2. Вольфганг Хоег, Томас Лаутербах К. Цифрове радіомовлення: принципи та програми DAB, DAB + і DMB: Пер. с англ. – СПб: Питер, 2009. – 426 с.
3. Джош Квітнер. Вільний кіберпростір радіо: Пер. с англ. – СПб: Символ-Плюс, 2009. – 216с.
4. Док Сеарлс. Чому так багато інтернет-радіостанцій в ефірі // Linux Journal. – № 2. – С. 49.
5. Ольга Хариф. Останні дні інтернет-радіо? : Пер. с англ. – СПб: Питер, 2007. – 410 с.
6. Рэнд Альфред. Цей день в Tech: Пер. с англ. – М.: Издательский дом Вильямс, 2013. – 154 с.
7. В. Олифер, Н. Олифер. Компьютерные сети. Принципы, технологии, протоколы. – СПб: Питер, 2013. – 944 с.
8. Фаулер М., Скотт К. UML. Основы: Пер. с англ. – СПб: Символ-Плюс, 2002. – 192 с.
9. Debian [Електронний ресурс] – Режим доступу:
<https://ru.wikipedia.org/wiki/Debian>
10. Baumgartner Peter, Malet Yann. High Performance Django (1st ed.) – 2015. – 184 с.
11. Игорь Ашманов, Андрей Иванов. Оптимизация и продвижение сайтов в поисковых системах. – СПб.: Питер, 2009. – 400 с.
12. Раскрутка. Секреты эффективного продвижения сайтов / А. Бабаев, Н. Евдокимов, М. Боді і ін. – СПб.: Питер, 2013. – 272 с.
13. Брайан Клифтон. Google Analytics для профессионалов. – М.: «Диалектика», 2013. – 608 с.
14. Forbes [Електронний ресурс] Режим доступу:
<https://www.forbes.com/sites/quora/2018/10/12/why-are-podcasts-gaining-in-popularity>
15. SEMBOOK. Энциклопедия поискового продвижения сайтов Ingate / Н. Неелова, Г. Загребельный, Е. Фролкина. – Москва.: Питер, 2014. – 520 с.
16. Лапыгин Ю. Н. Управление проектами: от планирования до оценки эффективности. – М.: Омега-Л, 2008. – 252 с.

Д О Д А Т К И

ДОДАТОК А КОНФІГУРАЦІЯ LIQUIDSOAP

```

#ЛОГИ
set("log.file",false)
set("log.stdout",true)
set("log.level",4)
set("harbor.bind_addr","0.0.0.0")
set("tag.encodings",["UTF-8"])
# ТЕЛНЕТ
set("server.telnet", true)

def smooth_add(~delay=0.5,~p=0.2,~normal,~special)
  d = delay
  fade.final = fade.final(duration=d*2.)
  fade.initial = fade.initial(duration=d*2.)
  q = 1. - p
  c = amplify
  fallback(track_sensitive=false,
    [special,normal],
    transitions=[
      fun(normal,special)->
        add(normalize=false,
          [c(p,normal),
            c(q,fade.final(type="sin",normal)),
            sequence([blank(duration=d),c(q,special)])]),
      fun(special,normal)->
        add(normalize=false,
          [c(p,normal),
            c(q,fade.initial(type="sin",normal))])
    ]
  )
end

radio_pl= playlist(reload=200, "/home/test/liquidsoap/radio")
ring_in= playlist("/home/test/liquidsoap/ring/ring_in.mp3")
ring_out=playlist("/home/test/liquidsoap/ring/ring_out.mp3")
ring_war=playlist("/home/test/liquidsoap/ring/ring_war.mp3")
gimn=playlist("/home/test/liquidsoap/ring/gimn.mp3")

```

```

ring_mor=playlist("/home/test/liquidsoap/ring/ring_mor.mp3")
warning=playlist(reload=100, "/home/test/liquidsoap/warning")

radio30 = switch(track_sensitive=false, [({ 0h31m - 23h54m59s},
radio_pl)])

radio=smooth_add(normal=radio30, special=switch([(0m0s or 20m0s or
40m0s}, warning])))

time_ring=switch(track_sensitive=true,
[
    ({ (10h00m-10h00m01s) or (11h30m-11h30m01s) or (13h20m-
13h20m01s) or (14h50m-14h50m01s)}, ring_in),
    ({ (8h40m-8h40m01s) or (10h10m-10h10m01s) or (12h00m-
12h00m01s) or (13h30m-13h30m01s) }, ring_out),
    ({ (10h8m-10h8m01s) or (11h55m-11h55m30s) or (13h28m-
13h28m01s)}, ring_war),
    ({ 8h20m}, gimn),
    ({ (8h35m-8h35m01s)}, ring_mor)
])

live = input.harbor("test-harbor", port=8001, password="hackme")
f_radio=fallback(track_sensitive=false, [live, radio, time_ring, blank()])
radio = smart_crossfade(f_radio)
ring_radio = fallback(track_sensitive=false, [live, time_ring, blank()])
output.icecast (%mp3(bitrate=192),
                host = "192.168.1.200", port = 8000,
                fallible=true,
                mount="test",
                radio)
output.icecast (%mp3(bitrate=192),
                host = "192.168.1.200", port = 8000,
                fallible=true,
                mount="ring_test",
                ring_radio)

```

ДОДАТОК Б

КОД СТОРІНКИ ЗАКАЗІВ

Main.html

```

<body>
<div class="container">
  <div class="jumbotron jumbotron-fluid bg-info text-white">
    <div class="container">
      <p class="lead" id="json_host">host: </p>
      <p class="lead" id="json_admin">admin: </p>
      <p class="lead" id="json_id">server id: </p>
      <p class="lead" id="json_start">server start: </p>
    </div>
  </div>
  <div id="points">
    </div>
  </div>
  <div class="modal fade" id="exampleModalCenter" tabindex="-1" role="dialog"
  aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalCenterTitle">Modal title</h5>
          <button type="button" class="close" data-dismiss="modal" aria-
  label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
      </div>
    </div>
  </div>
</body>

```

admin.js

```

$('#auth_submit').on('click', function () {
  var login = $('#authInputLogin1').val();
  var pass = $('#authInputPassword1').val();
  var b64 = window.btoa(login+pass);

```

```

document.cookie = 'Auth=' + b64;
console.log(getCookie('Auth'))
});
$('#points').on('click','button',function () {
    console.log("FINE");
    var mount = 'http://192.168.1.207:8000/admin/listclients?mount=/test'; //+
$(this).attr('id_mount');
    console.log(mount);
    $.ajax({
        type:'get',
        beforeSend: function(request) {
request.setRequestHeader("Authorization", 'Basic' + getCookie('Auth'));
console.log('set')
        },
        crossDomain: true,
        url: mount,
        dataType: "xml",
        xhrFields: {
withCredentials: true
        },
        processData: true,
        // headers: {
        //   'Authorization': 'Basic' + getCookie('Auth')
        // },
        // error: function (){
        //   $('#modal_list').html('ERROR 401')
        // },
        success: function (data) {
            $(data).find("listener").each(function () {

```

```
        console.log($(this).find("IP"));
        console.log("TEST_MODAL");
    })
}
});
})
```

ДОДАТОК В ДІАГРАМА USE-CASE

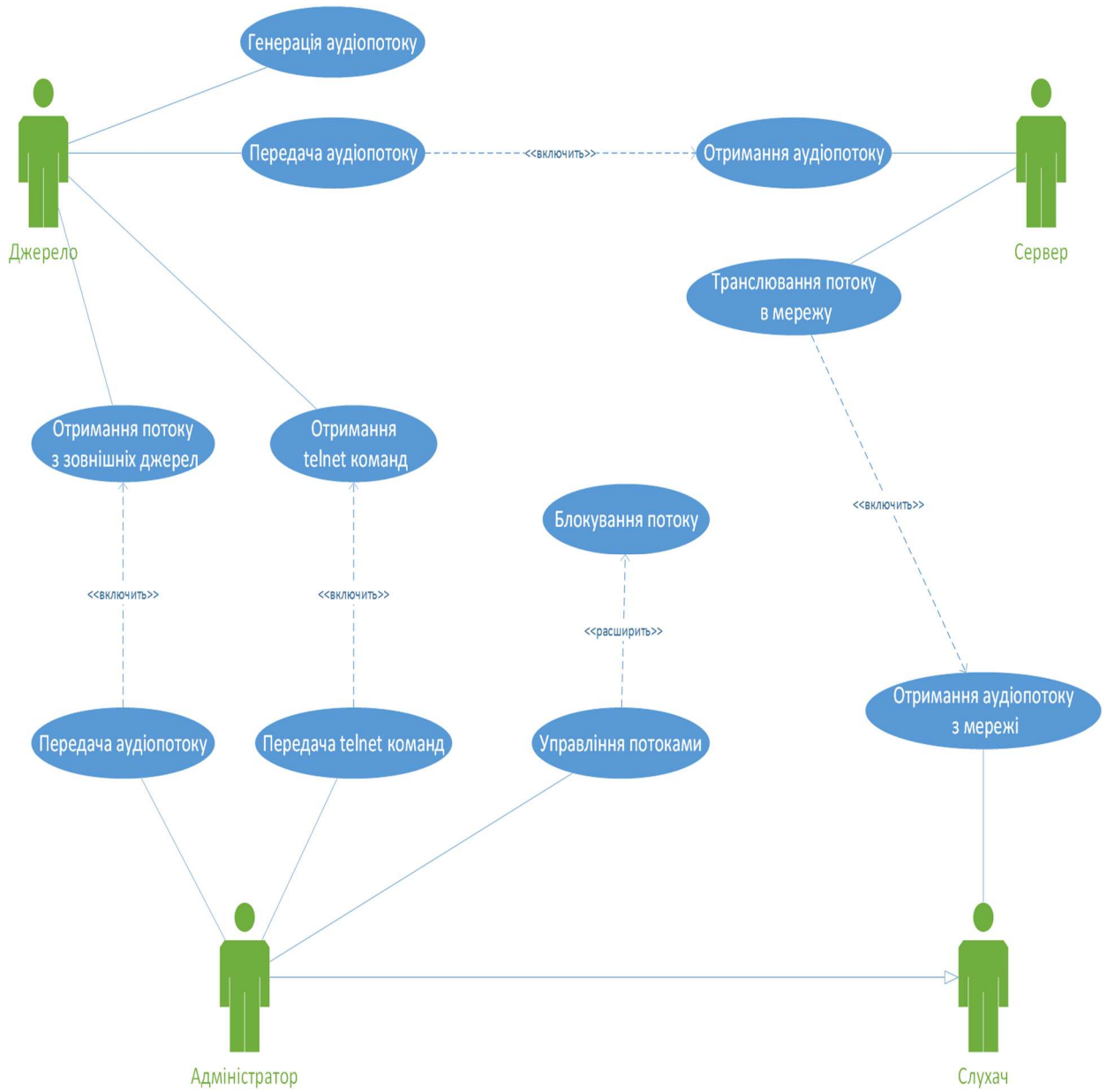


Рисунок В.1 – Діаграма прецедентів для інтернет радіо

ДОДАТОК Г ДІАГРАМА ДІЯЛЬНОСТІ

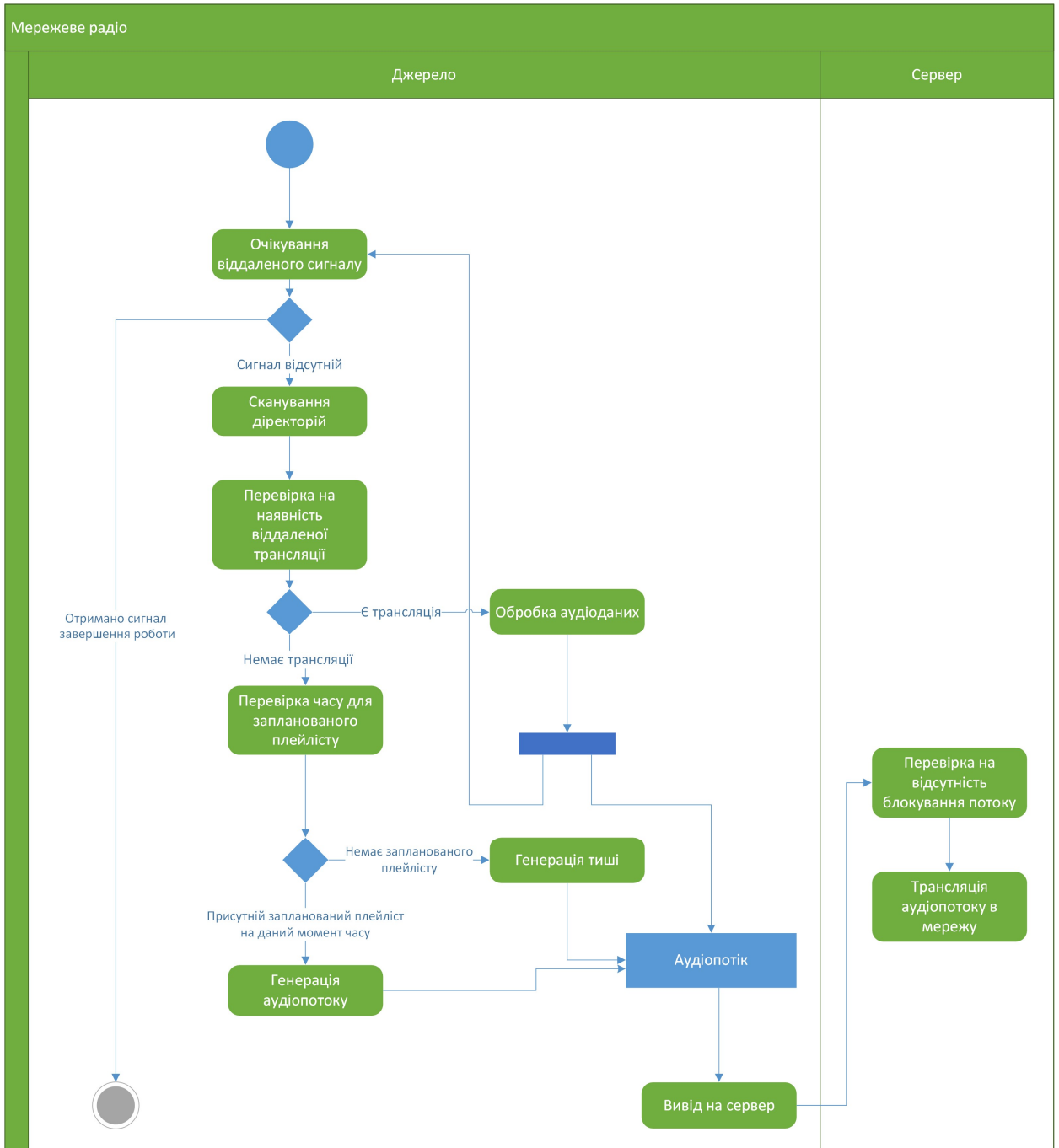


Рисунок Г.1 – Діаграма діяльності для радіомережі