МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук,
управління та адміністрування

Кафедра Інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка захищеного вебсервісу відеоконференцій

Виконав студент 2 курсу групи МІС-20
спеціальності 122 Комп'ютерні науки

Федченко Максим Сергійович

Керівник д.т.н., професор
Казакова Надія Феліксівна

Рецензент засновник Компанії «UALinux»
Попов Володимир Леонідович

Одеса 2021

АНОТАЦІЯ

на магістерську кваліфікаційну роботу
«Розробка захищеного вебсервісу відеоконференцій»,
студента Федченко Максима Сергійовича

Кваліфікаційна магістерська робота: 85 с., 3 табл., 29 рис., 5 дод., 23 с. дод, 30 джерел.

ПРОГРАМУВАННЯ, БАЗА ДАНИХ, ВЕБ, ПРОГРАМНА СИСТЕМА, ШИФРУВАННЯ, ЗАХИСТ, КОНФЕРЕНЦІЯ, WEBRTC, SFU, EF, MVC, ОНЛАЙН, КОМУНІКАЦІЯ, МЕДІА-ДАННІ.

Мета роботи – створення системи проведення захищених відеоконференцій, корпоративно-ділового напрямку, з можливостю подальшого масштабування.

Об’єкт дослідження – серверний та клієнтський додатки.

Метод дослідження – відкриті джерела з інформацією про організацію групових комунікацій, шифрування медіа-трафіку та особливості корпоративних комунікацій.

Перший розділ присвячений аналізу проблематики відеоконференцій у якості корпоративного інструменту, аналізу особливостей при бізнес комунікаціях та аналізу статистики використання та характеру використання відеоконференцій. Другий розділ присвячений засобам захисту системи відеоконференцій за допомогою протоколів TLS, SRTP, DTLS. Третій розділ присвячений основним технічним рішенням використаним при розробці системи відеоконференцій: WebRTC, SFU, EF, ASP.NET MVC. Четвертий розділ присвячений проектуванню та кодуванню системи відеоконференцій.

Основною сферою застосування даного додатку веб відеоконференцій є комунікації ділового характеру, з метою обговорення бізнес, технічних та проектних рішень різного характеру.

SUMMARY

for a master's degree

"Development of a Secure Video Conferencing Web Service",

of student Maksim Fedchenko

Qualifying master's thesis: 85 pages, 3 tables, 29 figures, 5 appendices, 23 appendices pages, 30 sources.

PROGRAMMING, DATABASE, WEB, SOFTWARE SYSTEM, ENCRYPTION, SECURITY, CONFERENCE, WEBRTC, SFU, EF, MVC, ONLINE, COMMUNICATION, MEDIA.

The purpose of the work is to create a system for secure video conferencing, corporate and business direction, with the possibility of further scaling.

The object of work – server and client applications.

Research method – open sources with information about the organization of group communications, encryption of media traffic and features of corporate communications.

The first section is devoted to the analysis of video conferencing as a corporate tool, analysis of features in business communications and analysis of statistics on the use and nature of the use of video conferencing. The second section is devoted to the means of protecting the video conferencing system using TLS, SRTP, DTLS. The third section is devoted to the main technical solutions used in the development of video conferencing: WebRTC, SFU, EF, ASP.NET MVC. The fourth section is devoted to the design and coding of video conferencing system.

The main sphere of this project of web video conferencing application is business communications, in order to discuss business, technical and design solutions of various kinds.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Центр сертифікації – сторона, чия чесність є незаперечною, а відкритий ключ широко відомий.

Сеансовий ключ – ключова інформація створювана між двома користувачами.

Simulcast – архітектура відеоконференцій, при якій процес обміну даними з сервером включає кілька медіа потоків різної роздільної здатності та якості.

VoIP – Voice over IP – IP телефонія.

SDP – Session Description Protocol – протокол опису сеансу зв’язку.

ICE – Interactive Connectivity Establishment – встановлення інтерактивного підключення.

NAT – Network Address Translation – перетворення мережевих адрес.

WebRTC – Web Real-Time Communication – веб-комунікація у реальному часі.

STUN – Session Traversal Utilities for NAT – утиліти проходження сесій для NAT.

TURN – Traversal Using Relay NAT – обхід за допомогою реле NAT.

SFU – Selective Forwarding Unit – одиниця селективного пересилання.

MAC – Message Authentication Code – код аутентифікації повідомлення.

SRTP – Secure Real-time Transport Protocol – сегментований цілий лічильник.

DTLS – Datagram Transport Layer Security – протокол датаграм безпеки транспортного рівня.

TLS – Transport Layer Security – захист на транспортному рівні.

SSRC – Synchronization source – джерело сихронізації; випадкове значення з метою синхронізації в сеансі RTP.

ORM – Object-Relational Mapping – об’єктно-реляційне зіставлення.

ВСТУП

Об’єктом розробки даного проекту є система веб відеоконференцій. Відеоконференція є потужним та найбільш ефективним засобом комунікацій між великою групою людей протягом тривалого часу, зі збереженням максимальної уваги до обговорюваної теми. Дані признаки є особливо важливими в бізнес-корпоративній середі, де важливість обговорюваних тем підкріплюються можливим добутком або витратами, а ефективність та лаконічність взаємодії – термінами часу та дедлайнами.

Відеоконференції відкривають незліченну кількість можливостей для бізнесу. Незалежно від того, чи це залучення віддалених працівників, співпраця між відділами та місцями, співбесіди з кандидатами на роботу чи управління постачальниками, ця технологія корпоративно ефективна, економічно ефективна та масштабована.

Для такої системи комунікацій, як і для більшості, є важливим її захищеність. Адже такий додаток необхідний бути розрахований на будь який характер даних що передаються, в тому числі важливий або особо таємній.

Не менш важливим також є використання системи відеоконференцій у якості веб додатку. Це гарантує більш велику підтримку користувальницького програмного забезпечення та несе в собі ряд інших переваг.

Також для даного характеру теми варто підкреслити її актуальність. Ріст частки відео-зв’язку як метода комунікації, в основі своєї, є постійною тенденцією. Однак крім того, в особливості, в сучасних реаліях пандемії є незаперечним лідером в спілкуванні будь-якого роду та характеру.

Таким чином, реалізація даної системи є більш ніж доцільною, та, саме, була здійснена, а її докладний опис було наведено у даній пояснювальній записці.

1 АНАЛІЗ ПРОБЛЕМАТИКИ ВІДЕОКОНФЕРЕНЦІЙ

При здійсненні діяльності організацій, істотну роль грає комунікація між співробітниками цієї організації. В умовах далекого взаємного розташування учасників або масштабованості організації, необхідним є використання відповідних програмних інструментів для комунікації. Інструментів що поліпшують взаємодію є безліч і кожен виконує свою відповідну функцію. Одну з таких функцій виконує інструмент відеоконференцій. При цьому такий інструмент повинен відповідати всім вимогам та специфіці таких корпоративних бізнес комунікацій. Що в веб-сервісі можуть виражатися: в унікальних полях і таблицях бази даних, що виражають організаційну структуру підприємства; в унікальних функціях при проведенні відеоконференцій; а також в деяких оптимізаційних рішеннях.

Таким чином аналіз проблематики відеоконференцій у якості корпоративного інструменту є важливим етапом підготовки до проектування.

1.1 Особливості корпоративних бізнес комунікацій

1.1.1 Основні відомості корпоративних комунікацій

Корпоративна бізнес комунікація пов’язана з взаємодією між фізичними особами та групами, що залучені в процеси адміністрації та управління в контексті ділової організації. Іншими словами, така комунікація є цілєорієнтованою, а саме базується на досягненні цілей бізнесу.

Метою комунікації є підвищення організаційної ефективності за рахунок зменшення помилок. Ділове спілкування включає різні аспекти, такі як маркетинг, зв’язки з громадськістю, відносини з клієнтами, корпоративне та міжособистісне спілкування тощо.

Основні елементи ділового спілкування [1][1)]:

– відправник;

– приймач;

– ділова інформація;

– фідбек.

Вищезазначені елементи вказують на ділове спілкування як на процес обміну інформацією чи новинами, пов'язаними з бізнесом, між різними діловими учасниками, такими як клієнти, постачальники, бізнес-клієнти, працівники тощо з метою ефективного адміністрування бізнесу.

Більше того, це передбачає регулярний потік інформації, а фідбек вважається вирішальним і важливим аспектом ділового спілкування. Завдяки різному рівню ієрархії та залученню величезної кількості людей, ділове спілкування відіграє важливу роль у різних функціях управління, тобто плануванні, координації, організації, керівництві та контролі.

1.1.2 Типи корпоративних бізнес комунікацій

1.1.2.1 Внутрішня комунікація

Внутрішня комунікація є спілкуванням, яке відбувається між членами організації. Це спілкування включає як офіційне, так і неформальне спілкування. Крім того, різні відділи, які передають комунікацію різними засобами, підпадають під внутрішнє спілкування. Внутрішнє спілкування має бути ефективним, оскільки воно є життєво важливим джерелом перегляду та представляє організаційні проблеми. Ефективне внутрішнє ділове спілкування може підвищити рівень задоволеності роботою, продуктивність праці, ефективність роботи працівників

[1)] [1] Microsoft Word - Methods Of Communication – [Електронний ресурс] URL: http://studymaterial.unipune.ac.in:8080/jspui/bitstream/123456789/4736/1/Methods%20of%20Communication.pdf (дата звернення: 21.04.2021)

за рахунок зменшення їхньої плинності та скарг та сприяє збільшенню прибутку [1][1)].

Внутрішня комунікація класифікується на внутрішню (висхідну) комунікацію та внутрішню (низхідну) комунікацію [2][2)].

Висхідна внутрішня комунікація являє собою внутрішню комунікацію що передбачає підхід від низу до верху управління. Інформація надходить від підлеглих до керівників або будь-якої особи, яка знаходиться на вищому рівні ієрархії.

Приклад висхідної внутрішньої комунікації зображено на рис. 1

Рисунок 1 – Приклад висхідної внутрішньої комунікації [1]

[1)] [1] Microsoft Word - Methods Of Communication – [Електронний ресурс] URL: http://studymaterial.unipune.ac.in:8080/jspui/bitstream/123456789/4736/1/Methods%20of%20Communication.pdf (дата звернення: 21.04.2021)

[2)] [2] Different Methods and Types of Business Communication – [Електронний ресурс] URL: https://wikifinancepedia.com/finance/business-planning/what-are-the-different-methods-modes-and-types-of-business-communication-systems (дата звернення: 21.04.2021)

До характеристик висхідної внутрішньої комунікації належать:

– він включає «bottom to top» підхід, тобто потік інформації відбувається від підлеглих до начальства.
– основною метою є надання своєчасного фідбеку, пропозицій, подання запитів, ескалації будь-яких питань чи проблем тощо начальству.

Низхідна внутрішня комунікація є спілкуванням при якому інформація надходить від вищого рівня, а саме керівництва, до працівників організації. Інформація пов'язана з передачею вказівок підлеглим або працівникам щодо виконання відповідних завдань. Комунікація донизу використовується менеджерами для повідомлення різних цілей, процедур та політики, керівних принципів, рішень, інструкцій тощо своїм підлеглим.

Процес низхідної комунікації в бізнесі включає передачу повідомлень від верхнього рівня до нижчого рівня за допомогою ланцюжка ієрархії. Цей тип спілкування може бути в усній або письмовій формі. Письмова форма включає різні повідомлення, посібники, виведення новин в електронній формі тощо, тоді як усна форма низхідної комунікації включає різні очні розмови, телефонне спілкування, зустрічі тощо.

До характеристик низхідної внутрішньої комунікації належать:

– включення підходу «top to bottom», тобто потік інформації відбувається від верхнього рівня до нижнього рівня.
– основна мета – донести організаційну мету, плани та процедури, інструкції тощо до підлеглих.

1.1.2.2 Горизонтальна комунікація

Горизонтальна або бічна комунікація пов'язана із спілкуванням між співробітниками, то є вербальне, або письмове спілкування. Це може включати міжвідомче спілкування або спілкування між міжвідомчими підрозділами, а також

може бути між людьми однакового або подібного рангу в компанії. Така комунікація є критично важливою для досягнення бажаних результатів. Комунікація, як вказано у [2][1)], відбувається серед працівників, що мають рівний рівень ієрархії. Для досягнення функціональної ефективності різних організаційних підрозділів, та взаємної кооперації використовується саме горизонтальна комунікація.

Приклад горизонтальної комунікації зображено на рис. 2.

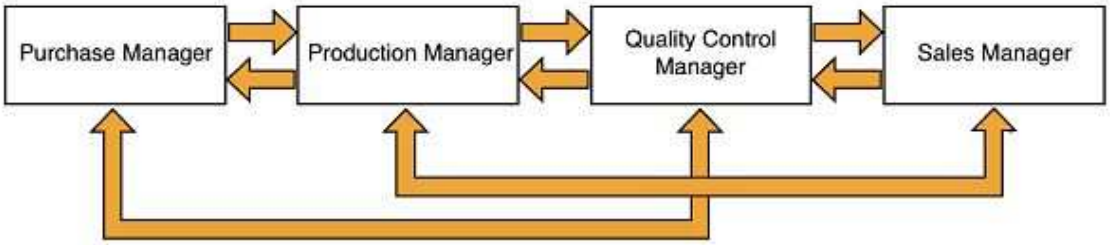

Рисунок 2 – Приклад горизонтальної комунікації

1.1.2.3 Зовнішня комунікація

Спілкування з людьми, що є зовнішніми для організації, відоме як зовнішня ділова комунікація. Ці люди можуть бути замовниками або акціонерами, постачальниками, партнерами чи регулюючими органами тощо.

1.1.3 Функції корпоративних бізнес комунікацій

Функція інформування о службових обов'язках є найважливішою ключовою функцією корпоративної комунікації. Члени команди, які мають ясність щодо очікуваних робочих завдань, і те, як вони можуть сприяти досягненню

[1)] [2] Different Methods and Types of Business Communication – [Електронний ресурс] URL: https://wikifinancepedia.com/finance/business-planning/what-are-the-different-methods-modes-and-types-of-business-communication-systems (дата звернення: 21.04.2021)

цілей організації, виконуючи свої робочі функції, вони можуть більше сприяти виконанню покладених на них завдань. За відсутності чіткості своїх ролей працівники можуть не завершити свою роботу, як очікувалося.

Функція забезпечення фідбеку працівникам та клієнтам також є важливою функцією ділового спілкування. Ефективність роботи співробітників може бути підвищена при регулярному фідбеку щодо їх роботи та компетенції. Це є важливим для розуміння поточних навичок та сильних сторін. Систематичний фідбек із замовниками та іншими зацікавленими сторонами щодо продуктів та послуг бізнесу сприяє поліпшенню виробничого процесу та якості.

Однією з функцій корпоративної комунікації є взаємодія з потенційними клієнтами, клієнтами та діловими партнерами для завершення ділової угоди чи операції. Довіра та емоції є важливим елементом цієї функції комунікації, тому найкращим є очна зустріч або використання веб-конференції, у разі неможливості зустрічі через відстань або інші обставини.

Функція побудові колективних зв’язків відіграє вирішальну роль у підтримці працівників та побудові соціального кола. Організація може мати відкрите культуре або робоче середовище, в якому працівники всіх рівнів можуть вільно спілкуватися між собою та начальством.

1.2 Аналіз відеоконференцій в бізнес комунікаціях

1.2.1 Особливості комунікації за допомогою відеоконференцій

Відеоконференції є основною частиною ведення бізнесу в 21 столітті. Це полегшує повністю візуальне спілкування з клієнтами та колегами. Відеоконференції у попередні роки були обмежені високими витратами та затримкою мережі. Однак зараз в Інтернеті проводиться більше зустрічей ніж у фізичних місцях.

Спочатку відеоконференції використовувались для зменшення дорожніх витрат. Зараз переваги відеоконференцій далекосяжні. Спільна робота ніколи не була простішою, і більше знань про переваги співпраці сприяє зростанню відеоконференцій.

Перспективою відеоконференцій є [3][1)]:

– покращена комунікація;
– кращі ділові відносини;
– більш ефективні зустрічі;
– задоволеність працівників;
– поліпшення конкурентних переваг.

1.2.2 Переваги відеоконференцій в корпоративних комунікаціях

Переваги відеоконференцій пов’язані з ключовим поняттям: одним з впливовіших факторів на продуктивність робочого процесу є ступінь взаємодії людей, що працюють в одному колективі.

Сьогодні зростаючий попит на відеоконференції робить її широко доступною. Це дуже важливо для стратегій співпраці багатьох компаній. Системи відеоконференцій працюють у діапазоні від базових домашніх ПК, розміщених у робочих кабінетах, до складних телепристроїв. Завдяки більш надійним технологіям та нижчим цінам відеоконференції стали основним інструментом корпоративної співпраці.

Тенденції витрат / попиту та простота використання роблять переваги відеоконференцій доступними для більшої кількості підприємств. Те, що відомо за терміном «співпраця», вплинуло на зростання відеоконференцій. Спільне робоче

[1)] [3] The Business Benefits of Video Conferencing | ViewSonic Library – [Електронний ресурс] URL: https://www.viewsonic.com/library/business/business-benefits-of-video-conferencing/ (дата звернення: 22.04.2021)

середовище, що орієнтоване на команду, забезпечує неймовірні ділові переваги. Серед багатьох переваг, продемонстрованих дослідженнями, було показано, що спільна робота за допомогою відео-конференцій [4][1)]:

- сприяє навчанню та виховуванню творчих здібностей працівників;
- формує довіру, заохочуючи рівну участь;
- розвиває почуття цілі;
- посилює розвиток співробітників;
- сприяє отриманню навичок вирішення конфліктів;
- сприяє більш широкому почуттю впливовості;
- заохочує партнерські або спільні зусилля;
- сприяє швидкому вирішуванні проблем та збільшуванню інновацій;
- створює ефективний темп роботи;
- поліпшує задоволеність роботою працівників, та сприяє лояльності.

Однією з великих переваг відеоконференцій є зменшення витраченого часу на зустрічі. Грошові витрати також є істотними. Проведення відеоконференцій усуває витрати на транспорт, готелі та харчування працівників.

Підсумком вигідних витрати щодо відеоконференцій є: зменшення витрат на відрядження лише для одного працівника значно перевищує витрати на систему дзвінків, якою може користуватися вся компанія.

Зменшені вимоги до поїздок допомагають ключовим працівникам максимізувати свій час. Проведення відео-дзвінків із клієнтами та партнерами по всьому світу підвищує продуктивність.

Відео-дзвінки зменшують непорозуміння, пов’язані з електронною поштою та телефонними дзвінками. За допомогою відеоконференцій вираз обличчя та мова тіла забезпечують більш повне повідомлення. Учасники конференцій також

[1)] [4] The Business Benefits of Video Conferencing | Parmetech – [Електронний ресурс] URL: https://www.parmetech.com/the-business-benefits-of-video-conferencing/ (дата звернення: 22.04.2021)

частіше висловлюються та уточнюють інформацію. Нарешті, учасники відеоконференцій, швидше за все, залишатимуться пильними та зосередженими, коли їх переглядають колеги.

Підсумком переваг продуктивності відеоконференцій є: відеоконференції максимізують людські ресурси; це підвищує ефективність і продуктивність; проекти можуть бути завершені швидше, і учасники почуватимуться більш синхронізованими між собою та цілями зустрічі.

Візуальні сигнали забезпечують рівень ефективності, неможливий при аудіо дзвінках. Через це учасники являються найбільш ефективними, працюючи за допомогою відеоконференції.

Завдяки відеоконференцій жорсткі обмеження часу та віддалені дзвінки можуть сприяти більш цілеспрямованому обговоренню з меншою кількістю побічних розмов. Також, коли учасники можуть бачити колег, фокус і внесок вищі. Під час без-візуального контакту телефонних дзвінків учасники можуть втратити зосередженість та перейти на заняття іншими справами.

Підсумком переваг ефективності відеоконференцій є: візуальний контакт працівників підвищує ефективність; більше взаємодії; менше відволікання від цілей праці.

Спілкування людей складне. Мова тіла. Тон голосу. Часом на вимовлене слово припадає лише 7% тлумачення значення. Телефонні дзвінки виявляють тон, але залишають мову тіла та вираз обличчя. Це часто є критичним для повного розуміння.

Візуальне спілкування породжує ефективну співпрацю. Особисте розуміння та зв’язок, запорука успішних робочих відносин, розвиваються під час відеоконференцій. Як зазначається в Harvard Business Review, цей зв'язок «сприяє груповій лояльності, що призводить до спільної прихильності та дисципліни щодо роботи». Маючи почуття товариськості, справжнього знайомства з нашими співробітниками, «створює загальне почуття мети та менталітету».

Відеоконференції, як зазначено у [4][1)], формують відносини, довіру та товариськість серед співробітників. Керівники можуть проводити зустрічі в режимі реального часу, при находженні в глобально розташованих офісах. Команди почуваються більш зв'язаними з домашнім офісом. Керівники проектів мають можливість взаємодії з тими, з ким вони не зможуть зустрітися особисто. Постачальники доставляють інформативні, візуально ефективні презентації.

Підсумком переваг взаємозв'язку при використанні відеоконференцій є: відеоконференція пропонує всі аспекти спілкування з віддалених кабінетів для ефективніших розмов та кращих робочих відносин.

Кожна з перелічених переваг також посилює конкурентні переваги компаній. Команди, які спілкуються через відеоконференції, обмінюються знаннями швидше та ефективніше. Це скорочує час виходу на ринок. Відділи підтримки встановлюють особисті стосунки з клієнтами. Виробники можуть скоротити час та покращити якість за допомогою таких зустрічей. Можливим є ефективно перевіряти якість, пропонувати зміни та забезпечувати точність протягом усього життєвого циклу продукту.

Підсумком конкурентних переваг відеоконференцій є: використання відеоконференцій дозволяє компаніям бути більш ефективними; зустрічі більш продуктивні; цикли розробки більш продумані.

1.2.3 Статистика за використанням відеоконференцій

Зростання віддаленої, географічно розподіленої робочої сили є найбільшою причиною збільшення залежності від відеоконференцій як одного з найважливіших інструментів віддаленої роботи. У 2020 році використання відеоконференцій

[1)] [4] The Business Benefits of Video Conferencing | Parmetech – [Електронний ресурс] URL: https://www.parmetech.com/the-business-benefits-of-video-conferencing/ (дата звернення: 22.04.2021)

стрімко зросло через пандемію COVID-19 та наслідки блокування. Згідно з цим представлена статистика поширення використання відеоконференцій [5][1)]:

– 55% компаній дозволяють віддалену роботу;
– віддалена робота збільшиться на 77% з 2019-2022 років;
– віддалена робота концертів збільшиться на 19% з 2019-2022 років;
– 30% працівників - це віддалені працівники, які працюють повний робочий день;
– 62% працівників час від часу працюють вдома;
– з 2010-2020 рр. кількість працівників, які працюють вдома принаймні раз на тиждень, зросла на 400%.

Вигоди при віддаленій роботі поширюються на роботодавців так само, як і на працівників. Надання членам команди гнучкості роботи з будь-якого місця збільшує продуктивність праці та знижує плинність працівників.

Загальна статистика переваг засобу комунікацій за допомогою відеоконференцій зображено на рис. 3.

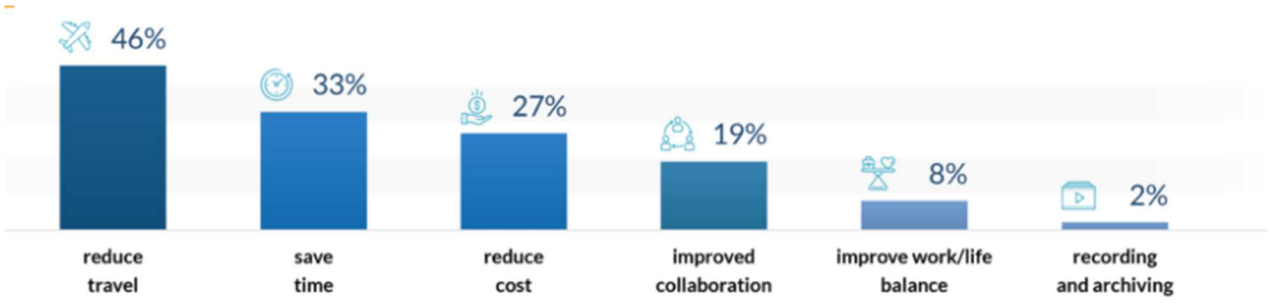

Рисунок 3 – Загальна статистика переваг відеоконференцій [5]

[1)] [5] 54 Basic Video and Web Conferencing Statistics: 2020/2021 Analysis of Data & Market Share - Financesonline.com – [Електронний ресурс] URL: https://financesonline.com/video-web-conferencing-statistics/ (дата звернення: 22.04.2021)

Докладніше о глобальній статистиці згідно з економічною вигодою представлено нижче:

– за допомогою програмного забезпечення для відеоконференцій компанії щороку заощаджують $ 11 000 на одного працівника;
– працівники заощаджують в середньому від 2000 $ до 7000 $ за допомогою інструментів веб-конференцій, що дозволяє їм дистанційно працювати;
– допомагаючи скоротити терміни проекту, інструменти відео дзвінків можуть заощадити компанії від 15% до 30% від загальної вартості проекту;
– відеоконференції можуть зменшити дорожні витрати до 30%;
– відеоконференції зменшують потребу у ділових поїздках на 47%.

Статистика основних перешкод впровадження засобу комунікацій за допомогою відеоконференцій зображено на рис. 4.

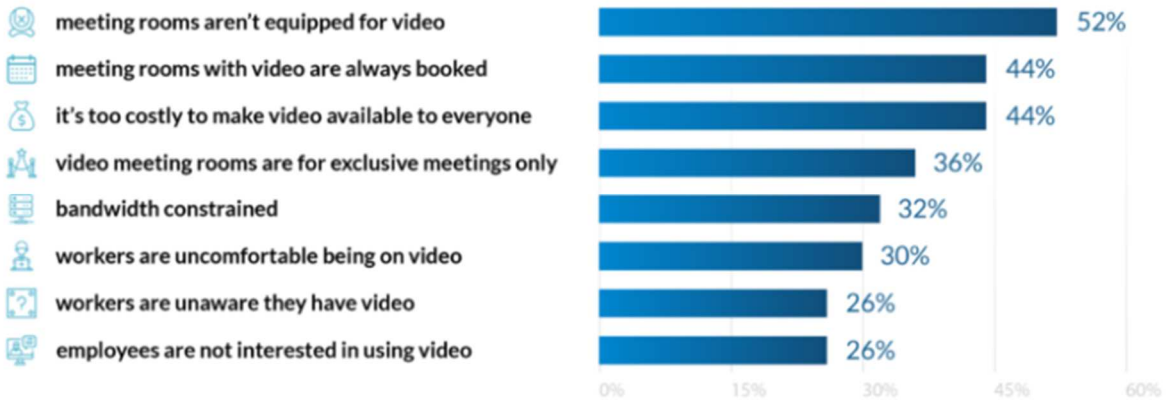

Рисунок 4 – Статистика основних перешкод впровадження відеоконференцій [5][1)]

[1)] [5] 54 Basic Video and Web Conferencing Statistics: 2020/2021 Analysis of Data & Market Share - Financesonline.com – [Електронний ресурс] URL: https://financesonline.com/video-web-conferencing-statistics/ (дата звернення: 22.04.2021)

Статистика поширеності технологій відеоконференцій зображено на рис. 5.

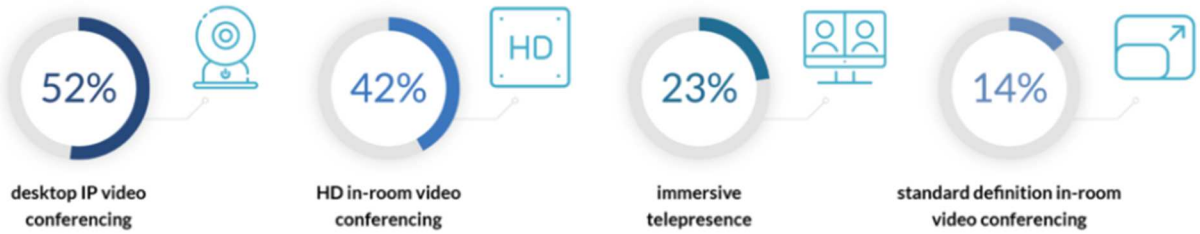

Рисунок 5 – Статистика поширеності технологій відеоконференцій [5][1)]

1.3 Аналіз технологій та програмних реалізацій

1.3.1 Аналіз технологій відео-зв'язку

Найчастіше використання настільних відеоконференцій спостерігається на настільних комп'ютерах або ноутбуках. Вони прості у використанні, будь-який користувач може встановити програмне забезпечення та отримати миттєвий доступ для спілкування. Все, що потрібно, це веб-камера та мікрофон, які, як правило, присутні в більшості ноутбуках. В таких десктопних додатках, як правило, використовується технологія VoIP.

VoIP [6][2)][7][3)] був побудований як передова телефонна технологія з метою зробити трансконтинентальний зв'язок економічно вигідним. По суті, VoIP – це набір цілого набору технологій, які працюють разом, щоб надати засобу повний

[1)] [5] 54 Basic Video and Web Conferencing Statistics: 2020/2021 Analysis of Data & Market Share - Financesonline.com – [Електронний ресурс] URL: https://financesonline.com/video-web-conferencing-statistics/ (дата звернення: 22.04.2021)

[2)] [6] Video Conferencing Tools – [Електронний ресурс] URL: http://www.ecommerce-digest.com/video-conferencing.html (дата звернення: 23.04.2021)

[3)] [7] Video Conferencing Technology – [Електронний ресурс] URL: https://www.vocal.com/video/video-conferencing-technology/ (дата звернення: 23.04.2021)

спектр послуг. Цей набір включає такі технології, як технології сигналізації, медіа-движок, протокол опису сеансів (SDP), протокол реального часу / протокол управління реальним часом (RTP / RTCP), переклад мережевих адрес (NAT), протоколи безпеки, якість обслуговування (QoS).) та інші технології телефонії.

Кожен із цих будівельних блоків має кілька варіантів. Наприклад, служба VoIP може використовувати будь-яку широко доступну технологію передачі сигналів, таку як протокол ініціювання сеансів (SIP), H.323 або протокол розширених повідомлень та присутності (XMPP). Подібним чином існує безліч різних медіа-двигунів та інших типів технологій. Кожен постачальник послуг VoIP використовує власний набір технологій, щоб представити найефективніші та найефективніші послуги.

Список поширених програмних рішень, що мають десктопні реалізації:

– Zoom;
– GoToMeeting;
– Google Meet;
– Microsoft Teams;
– Uberconference;
– Cisco Webex;
– Join.me;
– Bluejeans;
– ClickMeeting;
– GlobalMeet.

Однак VoIP є не єдиною технологію для конференцій, все більш розвивається та частіше використовується браузерно-орієнтована технологія WebRTC, на основі якого створюються веб-додатки для реалізації функцій відеоконференцій.

WebRTC (Web Real-Time Communication) – це API, який розробляється Консорціумом всесвітньої павутини (W3C). Простіше кажучи, це програмний

посередник, який дозволяє прикладним програмам взаємодіяти між собою та обмінюватися даними. WebRTC використовується, щоб увімкнути програми від браузера до браузера для голосових дзвінків, обміну файлами P2P та відеочату без плагінів. WebRTC – це нова технологія, доступ до якої здійснюється за допомогою API JavaScript.

Список поширених програмних рішень, що мають веб-реалізації:

– Skype for Business;

– BlueJeans;

– GoToMeeting;

– Google Meet.

Крім програмних засобів існує ряд апаратних можливостей що здатні розширювати можливості відеоконференцій або придати їм більш зручний характер, але при цьому потребують додаткової апаратури [7][1)]. Прикладом цього можуть служити «Room-based» відеоконференції або конференції з ефектом телеприсутності, для створення яких є потреба у великих моніторах відповідного дозволу, відеокамера що реагує на рух або 360-градусна відеокамера [8][2)]. Такі пристрої можуть бути унікальними для використання, та потребувати придбання разом з основними послугами, у компанії що надає послуги. Список програмних рішень що передбачають унікальні апаратні можливості:

– Zoom Rooms;

– Avatour;

– Cisco Webex.

---

1) [7] Video Conferencing Technology – [Електронний ресурс] URL: https://www.vocal.com/video/video-conferencing-technology/ (дата звернення: 23.04.2021)

2) [8] Video Conferencing Technology Trends Shaping The Future Of Communication – [Електронний ресурс] URL: https://www.shure.com/en-US/conferencing-meetings/ignite/video-conferencing-technology-trends-shaping-the-future-of-communication/ (дата звернення: 23.04.2021)

1.3.2 Аналіз недоліків існуючих програмних додатків

Залежність від інтернет з’єднання. Не завжди зв’язок може бути ідеальним, що іноді призводить до розмитих передач зображення та втрати важливих візуальних сигналів. Однак навіть при ідеальному підключенні затримка неминуча.

Навантаження серверу від медіа трафіку. Якщо сервіс використовують багата кількість людей, важкий медіа трафік може заподіяти перевантаженню сервера і в наслідок заморозити поточні відеоконференції.

Обмеження за кількістю підключених людей. Коли проводяться відеоконференції, неможливо оглянути кімнату, щоб зв’язатися з людьми так, як можливо, при знаходженні у конференц-залі. Якщо в засобі немає спеціальних функцій які б згладжували цей ефект, то є потреба регулювати кількість залучених в конференцію людей.

Проблема використання засобу та взаємодії з інтерфейсом. Не всім буде зручно користуватися платформою для відеоконференцій з самого початку. Компаніям та університетам потрібно виділити час, гроші та людські ресурси для навчання працівників. Однак навіть після ретельного навчання людям знадобляться дні чи навіть тижні, щоб звикнути до нового режиму спілкування.

Ціна. Якщо мова йде про організації, потрібно мати усіх у системі відеоконференцій, а також конференц-залах, обладнаних цими системами, що істотно впливає на вартість засобу.

Висновки до розділу

При аналізі проблематики відеоконференцій у якості корпоративного інструменту було:

- здійснено аналіз особливостей корпоративних бізнес комунікацій – це сприяло врахуванню специфіки корпоративних бізнес комунікацій при проектуванні веб-сервісу;
- здійснено аналіз відеоконференцій у контексті корпоративної бізнес комунікації – це сприяло врахуванню специфіки корпоративних відеоконференцій в межах підприємства при проектуванні веб-сервісу;
- здійснено аналіз технологій відеоконференцій та їх програмних реалізацій – це сприяло врахуванню специфіки недоліків існуючих програмних реалізацій та технологій на яких вони базуються.

2 АНАЛІЗ ЗАСОБІВ ЗАХИСТУ ОНЛАЙН КОНФЕРЕНЦІЙ

У той час як більшість користувальницьких додатків відео-чатів використовується для повсякденного досвіду, який може бути досить приватним та особистим, критично важливим є використання додатків для бізнес комунікацій, що може зосереджуватися навколо комерційної таємниці, деталей про товари, патенти та кадрові записи, що у разі витоку можуть мати серйозні економічні та юридичні наслідки. Тому при здійсненні відео-дзвінків, з оголошенням підвищено-чутливої інформації, як і для інших засобів зв'язку, є доцільним використання методів захисту приватної інформації.

Для забезпечення захисту веб-сервісу відеоконференцій необхідним є набір засобів (криптографічних протоколів) які будуть враховувати специфіку даних що передаються, а саме медіа даних які передаються в режимі реального часу в мережі з можливою втратою пакетів.

В даному розділі, власне, і будуть розглянуті та описанні дані засоби для захисту веб-сервісу відеоконференцій.

2.1 HTTPS як засіб захисту клієнт-серверного з'єднання

Перш ніж забезпечити безпеку передачі медіа даних між користувачами, потрібно забезпечити безпечне з'єднання користувачів з сервером, за допомогою якого, в тому числі, буде здійснюватися сигналізація для зв'язку клієнтів. Для цього існує вже інтегроване та підтримуєме браузерами і веб-серверами розширення протоколу HTTP на базі криптографічних протоколів SSL або TLS – HTTPS.

TLS є приймачем SSL і в наслідок цього більш надійним, саме він і буде використаний для забезпечення захисту клієнт-серверного з'єднання веб-сервісу відеоконференцій.

2.1.1 Загальний опис протоколу TLS

TLS дає можливість клієнт-серверним додаткам здійснювати зв’язок в мережі таким чином, що виключає можливість проводити прослуховування пакетів і здійснити несанкціонований доступ.

TLS використовує асиметричне шифрування для аутентифікації, симетричне шифрування для конфіденційності та коди автентичності повідомлень для збереження цілісності повідомлень.

Загальний опис процедури створення захищеного сеансу зв’язку [9][1)]:

- клієнт підключається до сервера, що підтримує TLS, і запитує захищене з’єднання;
- клієнт надає список підтримуваних алгоритмів шифрування і хеш-функцій;
- сервер вибирає зі списку, наданого клієнтом, найбільш надійні алгоритми серед тих, які підтримуються сервером, і повідомляє про свій вибір клієнта;
- сервер відправляє клієнту цифровий сертифікат для власної аутентифікації; цифровий сертифікат містить ім’я сервера, ім’я засвідчувального центру сертифікації і відкритий ключ сервера;
- клієнт, до початку передачі даних, перевіряє валідність отриманого серверного сертифіката щодо наявних у клієнта кореневих сертифікатів засвідчувальних центрів сертифікації;
- для шифрування сесії використовується сеансовий ключ; отримання загального секретного сеансового ключа клієнтом і сервером проводиться по протоколу Діффі-Хеллмана.

[1)] [9] Transport Layer Security - Wikipedia – [Електронний ресурс] URL: https://en.wikipedia.org/wiki/Transport_Layer_Security (дата звернення: 24.04.2021)

На цьому закінчується процедура підтвердження зв’язку. Між клієнтом і сервером встановлено безпечне з’єднання, дані, що передаються по ньому, шифруються і розшифровуються за використанням симетричної криптосистеми до тих пір, поки з’єднання не буде завершено.

2.1.2 Засоби безпеки протоколу

Засоби безпеки протоколу TLS [10][1)]:

- захист від зниження версії протоколу до попередньої (менш захищеною) версії або менш надійного алгоритму шифрування;
- нумерація послідовних записів додатки і використання порядкового номера в коді аутентифікації повідомлення (MAC);
- використання ключа в ідентифікатор повідомлення (тільки власник ключа може згенерувати код аутентифікації повідомлення); алгоритм обчислення коду аутентифікації (HMAC), який використовується у багатьох сесіях TLS;
- повідомлення, яким закінчується підтвердження зв’язку («Finished»), використовується для підтвердження автентичності раннє переданих повідомлень і, таким чином, обраних параметрів TLS-з’єднання.

2.1.3 Опис процедури встановлення з’єднання

Згідно з протоколом TLS, додатки обмінюються записами, що інкапсулюють інформацію, яка повинна бути передана. Кожна із записів може бути стиснута, доповнена, зашифрована або ідентифікована MAC (код аутентифікації повідомлення) в залежності від поточного стану з’єднання (стану протоколу).

[1)] [10] RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc5246 (дата звернення: 24.04.2021)

Кожен запис в TLS містить наступні поля: «Content Type» (визначає тип вмісту записи), «Version» (поле, яке вказує версію протоколу TLS) і «Length» (поле, яке вказує довжину пакета) [10][1)][11][2)].

Коли з'єднання встановлюється, взаємодія йде по протоколу рукостискання (TLS handshake).

Алгоритм з'єднання:

а) Фаза переговорів

1) клієнт посилає повідомлення «ClientHello», вказуючи останню версію підтримуваного TLS-протоколу, випадкове число і список підтримуваних методів шифрування і стиснення, придатних для роботи з TLS;
2) сервер відповідає повідомленням «ServerHello», що містить: обрану сервером версію протоколу, випадкове число, надіслане клієнтом, відповідний алгоритм шифрування і стиснення зі списку наданого клієнтом;
3) сервер посилає повідомлення «Certificate», яке містить цифровий сертифікат сервера (в залежності від алгоритму шифрування цей етап може бути пропущений);
4) сервер посилає повідомлення «CertificateRequest», яке містить запит сертифіката клієнта для взаємної перевірки автентичності;
5) клієнт посилає повідомлення «Certificate», яке містить цифровий сертифікат клієнта;
6) сервер відсилає повідомлення «ServerHelloDone», що ідентифікує закінчення підтвердження зв'язку;

---

1) [10] RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc5246 (дата звернення: 24.04.2021)

2) [11] Transport Layer Security protocol | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/windows-server/security/tls/transport-layer-security-protocol (дата звернення: 24.04.2021)

7) клієнт відповідає повідомленням «ClientKeyExchange», яке містить відкритий ключ «PreMasterSecret» або нічого (залежить від алгоритму шифрування);
8) клієнт і сервер, використовуючи ключ «PreMasterSecret» і випадково згенеровані числа, обчислюють загальний секретний ключ. Вся інша інформація про ключі буде отримана із загального секретного ключа (згенерованих клієнтом і сервером випадкових значень).

b) Клієнт посилає повідомлення «ChangeCipherSpec», яке вказує на те, що вся подальша інформація буде зашифрована встановленим в процесі підтвердження зв'язку алгоритмом, використовуючи загальний секретний ключ
   1) клієнт посилає повідомлення «Finished», яке містить хеш і MAC, згенеровані на основі попередніх повідомлень процедури підтвердження зв’язку;
   2) сервер намагається розшифрувати «Finished» повідомлення клієнта і перевірити хеш і MAC. Якщо процес розшифровки або перевірки не вдається, підтвердження зв’язку вважається невдалим, і з’єднання повинно бути обірвано.
c) Сервер посилає «ChangeCipherSpec» і зашифроване повідомлення «Finished», і в свою чергу клієнт теж виконує розшифровку і перевірку.

З цього моменту підтвердження зв’язку вважається завершеним, протокол встановленим. Весь подальший вміст пакетів та всі дані будуть зашифровані.

2.2 SRTP як засіб захисту медіа з'єднання

Захищений транспортний протокол режиму реального часу (SRTP), профіль транспортного протоколу реального часу (RTP), здатен забезпечувати конфіденційність, автентифікацію повідомлення, контролю трафіку та захист від повторного відтворення пакетів. Завдяки даними характеристикам та його базуванні на дейтаграмах, SRTP є найбільш відповідним криптографічним протоколом для організації шифрування даних відео-дзвінків зі з'єднанням «peer to peer».

2.2.1 Опис, особливості на переваги

SRTP забезпечує основу для шифрування та автентифікації потоків повідомлень RTP та RTCP. SRTP визначає набір за дефолтом криптографічних перетворень, що дозволяє новим трансформаціям бути введеними в майбутньому. Завдяки відповідному менеджменту ключів, SRTP є безпечним для RTP-додатків з юнікаст та мультикаст віщанням [12][1)].

SRTP може досягти високої пропускної здатності та низького розширення пакетів. SRTP виявляється придатним для захисту неоднорідних середовищ (поєднання дротових та бездротових мереж). Для досягнення надійності, за замовчуванням описані перетворення базуються на адитивному потоковому шифрі, хеш-функції для аутентифікації повідомлень, та «неявному» індексі для синхронізації пакетів.

Цілями безпеки для SRTP є забезпечення:

- конфіденційності трафіку RTP;
- цілісності усіх пакетів RTP разом з захистом від повторно відтворених пакетів.

[1)] [12] Secure Real-time Transport Protocol - Wikipedia – [Електронний ресурс] URL: https://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol (дата звернення: 25.04.2021)

Ці служби безпеки є опціальними і незалежними одна від одної, за винятком того, що захист цілісності SRTCP є обов'язковим (зловмисний або помилкова зміна повідомлень RTCP може в іншому випадку порушити обробку потоку RTP).

Іншими, функціональними, цілями протоколу є [13][1)]:

- фреймворк, який дозволяє модернізувати нові криптографічні перетворення;
- низька вартість пропускної здатності, тобто фреймворк, що забезпечує ефективність стиснення заголовку RTP;
- низька обчислювальна вартість;
- невеликий розмір (тобто, невеликий розмір коду та пам'ять даних для введення інформації та списки відтворення);
- обмежене розширення пакетів для підтримки цілі економії пропускної здатності;
- незалежність від основного методу доставки, мережі та фізичного шару, що використовуються RTP, зокрема висока стійкість до втрати пакетів та повторне замовлення.

Ці властивості гарантують, що SRTP є відповідною схемою захисту для RTP як в дротових, так і в бездротових сценаріях.

Окрім вищезазначених прямих цілей, SRTP передбачає деякі додаткові характеристики. Вони були введені для полегшення управлінням ключами та подальшого підвищення безпеки. Вони включають [14][2)]:

- єдиний «майстер ключів» що забезпечує необхідними ключами сесій, що надає конфіденційність та захист цілісності для потоку SRTP;

[1)] [13] RFC 3711 - The Secure Real-time Transport Protocol (SRTP) – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc3711 (дата звернення: 25.04.2021)

[2)] [14] [MS-SRTP]: Secure Real-time Transport Protocol (SRTP) Profile | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/openspecs/office_protocols/ms-srtp/d9641c95-b152-4cc7-8311-d178f3241f1f (дата звернення: 25.04.2021)

- крім того, генерація сеансових ключів може бути налаштована на періодичне оновлення, що обмежує кількість зашифрованого трафіку виробленого фіксованим ключем, доступним для крипто-аналізу зловмисника;
- «соляні клавіші» використовуються для захисту від попередньо обчислених атак та атак з використанням тимчасової пам’яті.

2.2.2 Шифрування потоку даних

Для шифрування даних у ненадійних мережах з можливою втратою пакетів, SRTP використовує AES з режимом лічильника.

Сегментований цілочисельний режим лічильника (AES-CTR) складається з шифрування послідовних цілих чисел. Щоб рандомізувати початкову точку цілочисельної послідовності. Кожен пакет є зашифрованим окремим сегментом потоку ключів, який повинен обчислюватися наступним чином.

Сегментом ключового потоку повинно бути об’єднання 128-бітових вихідних блоків шифру AES у напрямку шифрування, використовуючи ключ k = k_e, в якому індекси блоків зростають. Кожен сегмент ключових потоків виглядає як:

E (k, IV) || E (k, IV + 1 mod 2 ^ 128) || E (k, IV + 2 mod 2 ^ 128)

де 128-бітове ціле значення IV що повинно визначатися SSRC, індексом пакета SRTP i та ключовим засобом k_s сеансу SRTP, як показано нижче.

IV = (k_s * 2 ^ 16) XOR (SSRC * 2 ^ 64) XOR (i * 2 ^ 16)

Кожен із трьох термінів у XOR-сумі вище заповнений такою кількістю провідних нулів, скільки потрібно для чіткого визначення операції, розглянуте як 128-бітове значення.

Включення SSRC дозволяє використовувати той самий ключ для захисту різних потоків SRTP в межах одного сеансу RTP.

Важливим моментом є що початкове значення IV є фіксованим для кожного пакета і формується шляхом «резервування» 16 нулів у найменш значущих бітах для цілі лічильника. Кількість блоків сформованого потоку ключів для будь-якого фіксованого значення IV не повинна перевищувати 2^16, щоб уникнути повторного використання потоку ключів. AES має розмір блоку 128 біт, тому 2^16 вихідних блоків достатньо для генерування 2^23 біт ключового потоку, необхідного для шифрування найбільшого можливого пакета RTP (за винятком «jumbogram» IPv6, які, явно не будуть використовуватися для мультимедійного трафіку на основі RTP). Це обмеження на максимальний бітовий розмір пакету, який можна зашифрувати, забезпечує безпеку методу шифрування, обмежуючи ефективність імовірнісних атак (BDJR).

Для певного ключа режиму лічильника кожне значення IV, яке використовується як вхід, повинно бути різним. Щоб задовольнити це обмеження, реалізація повинна забезпечити, щоб комбінація пакета SRTP індексів ROC || SEQ і SSRC, що використовуються при побудові IV, відрізняються для будь-якого конкретного ключа. Неможливість забезпечити цю унікальність може бути катастрофічною для безпечного RTP. Це на відміну від ситуації з самим RTP, який може терпіти такі збої. Рекомендується, що, якщо присутній спеціальний модуль безпеки, номери послідовності RTP та SSRC або генеруються, або перевіряються цим модулем (тобто, послідовність номерів та обробка SSRC в системі SRTP повинні захищатися, таким же чином як і ключ).

2.2.3 Захист за допомогою автентифікації та повірки цілісності

Реалізації SRTP використовують «неявний» пакетний індекс для послідовності, тобто, не весь індекс явно передається в пакеті SRTP. Для попередньо визначених перетворень індекс і використовується у відтворенні захист, шифрування, повідомлення автентифікації, а також для виведення ключів.

Коли сесія розпочнеться, сторона відправника встановлює лічильник, ROC, до нуля. Кожен раз, коли послідовний номер RTP, SEQ, обгортається за модулем 2^16, сторона відправника збільшує ROC на одиницю, модуль 2^32. Тоді індекс пакета відправника визначається як:

i = 2 ^ 16 * ROC + SEQ.

Реалізації на стороні приймача використовують порядковий номер RTP для визначити правильний індекс пакета, який є місцем пакет у послідовності всіх пакетів SRTP. Надійний підхід для правильне використання лічильника для перекидання вимагає обробки та використання бути чітко визначеними. Зокрема, непрацюючі пакети RTP, з порядковими номерами близькіми до 2^16 або до нулю, повинні бути належним чином оброблені.

Безпечний захист від відтворення можливий лише за умови присутньому захисту цілісності. Пакет «відтворюється», коли його зберігає зловмисник, а потім повторно вводиться в мережу. Коли аутентифікація повідомлення SRTP захищає від таких атак за допомогою списку відтворення, кожен приймач SRTP підтримує список, який концептуально містить індекси всіх отриманих засвідчених пакетів. На практиці список може використовуватися «sliding window» підхід, так що фіксованого обсягу пам’яті вистачає для захисту від відтворення.

Приймач перевіряє індекс вхідного пакета з списку відтворення та вікна. Тільки пакети з індексом перед вікном, або у вікні, але ще не отримані, можуть бути прийняті.

Після автентифікації пакета список відтворення оновлюється новим індексом.

Повідомлення RTP піддаються атакам на їх цілісність та ідентифікації джерела. Для захисту від цих атак, кожен потік SRTP є захищеним за допомогою HMAC-SHA1.

Алгоритм хешування HMAC-SHA1 є алгоритмом аутентифікації повідомлення за замовчуванням. Довжина ключа аутентифікації сеансу за замовчуванням

(n_a) дорівнюється 160 біт, довжина тегу автентифікації за замовчуванням (n_tag) дорівнюється 80 біт, а SRTP_PREFIX_LENGTH, для HMAC-SHA1, дорівнюється нулю. Для SRTP менші значення не рекомендуються.

2.2.4 Генерація ключів

Незалежно від шифрування або автентифікації повідомлень що застосовуються, сумісні SRTP реалізації повинні використовувати генерацію сеансового ключа SRTP для подальшого його використання. Як тільки синхронізація генерації ключів буде правильно сигналізована на початку сесії, немає необхідності в додаткових сигналах між сторонами, які використовують майстер ключів SRTP.

Щонайменше одну початкову генерацію ключа повинно виконувати SRTP. Подальші програми генерації ключів можуть бути виконано відповідно до значення key_derivation_rate у криптографічному контексті. Функція генерації ключів викликається до відправки першого пакету, потім, коли r> 0, генерація ключа виконується щоразу коли індекс mod r дорівнює нулю. Значення key_derivation_rate зберігається фіксованим протягом терміну служби відповідного головного ключа.

2.3 DTLS як засіб встановлення з’єднання для SRTP

Для забезпечення ключами SRTP з’єднання необхідна робота ще одного протоколу, що буде ці ключі між мережевими вузлами обмінювати. Так як відеозв’язок повинен мати можливість бути здійсненим між клієнтами на пряму, TLS не є відповідним варіантом, в зв’язку з його базуванні на стійкому з’єднанні TCP. DTLS в свою чергу базується на дейтаграмах і є видозміненим протоколом TLS. Таким чином дана комбінація протоколів DTLS-SRTP буде забезпечувати необхідну надійність каналу переданих медіа даних.

2.3.1 Особливості та переваги над TLS

У багатьох випадках найбільш надійним засобом захисту клієнтських та серверних додатків було б використання TLS; однак вимога до семантики дейтаграм, що виникає за необхідністю роботи протоколу в незнадійних мережах, автоматично забороняє використання TLS. DTLS [15][1)] навмисно розроблений, щоб бути найбільш схожим до TLS, як для мінімізації нововведень до засобів безпеки, так і для максимізувати кількість повторного використання коду та інфраструктури.

Основною ідеєю проектування DTLS є побудова «TLS над транспортом у вигляді дейтаграм». Причиною того, що TLS не можна використовувати безпосередньо в середовищах дейтаграм, є те, що пакети можуть бути втрачені або їх порядок змінений. TLS не має внутрішніх засобів для усунення такого роду ненадійності; тому реалізації TLS ламаються при повторному розміщенні на транспорті дейтаграм. Метою DTLS є внесення лише мінімальних змін до TLS, необхідних для вирішення цієї проблеми.

У рівні шифрування трафіку TLS (так званий рівень запису TLS), записи не є незалежними. Існує два види залежностей між записами:

1. Криптографічний контекст (потік ключа шифру потоку) зберігається між записами.

2. Захист від повторного відтворення та переупорядкування повідомлень забезпечується завдяки MAC, що включає порядковий номер, але порядковий номер є неявними в записах.

DTLS вирішує першу проблему забороною потокових шифрів. DTLS вирішує другу проблему додаванням явних порядкових номерів.

[1)] [15] Datagram Transport Layer Security - Wikipedia – [Електронний ресурс] URL: https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security (дата звернення: 26.04.2021

Для обробки втрачених пакетів DTLS використовує простий таймер повторної передачі [16][1)].

Кожному повідомленні протоколу рукостискання присвоюється певний порядковий номер. Коли мережевий вузол отримує повідомлення протоколу рукостискання, він може швидко визначити, чи є це повідомлення наступним повідомленням, яке він очікує. Якщо воно є, то він його обробляє. Якщо ні, то ставить це повідомлення у чергу для подальшої обробки, як тільки всі попередні повідомлення будуть отриманні.

2.3.2 Протокол рукостискання DTLS

DTLS використовує такий ж самий формат повідомленнь протоколу рукостискання, що і TLS, але за винятком трьох основних змін [16][1)]:

- додано обмін файлами «stateless cookie», щоб запобігти атакам відмови сервісну.
- модифікації заголовка рукостискання для усунення втрати повідомлення, переупорядкування та фрагментація повідомлень DTLS (щоб уникнути фрагментації IP).
- таймери повторної передачі для обробки втрат повідомлень.

За цими винятками, формати, потоки та логіка повідомлень DTLS є такі ж, як у TLS.

Протоколи безпеки дейтаграм надзвичайно сприйнятливі до різноманітних DoS-атак. Для протидії цим атакам DTLS позичає «stateless cookie» алгоритм, що використовується у Photuris та IKE протоколах. Коли клієнт надсилає своє повідомлення «ClientHello» на сервер, сервер може відповісти повідомленням «HelloVerifyRequest». Це повідомлення містить «stateless cookie», створений за

[1)] [16] RFC 6347 - Datagram Transport Layer Security Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc6347 (дата звернення: 26.04.2021)

допомогою алгоритму Photuris. Клієнт повторно передає «ClientHello» з доданим файлом cookie. Потім сервер перевіряє файл cookie та продовжує рукостискання лише якщо він дійсний. Цей механізм змушує зловмисника або клієнта мати можливість отримати файл cookie, який робить DoS-атаки з підробленою IP-адресою важко здійснюваними.

Фактичний обмін ключами здійснюються за чотирьома повідомленнями: сертифікат серверу, «ServerKeyExchange», сертифікат клієнту та «ClientKeyExchange».

Після повідомлення «Hello» сервер надішле свій сертифікат в повідомленні сертифікату, якщо воно підлягає автентифікації. Крім того, повідомлення «ServerKeyExchange» може бути надіслане, якщо це потрібно (наприклад, якщо сервер не має сертифікату, або якщо його сертифікат призначений лише для підписання). Якщо сервер автентифікований, він може запитати сертифікат від клієнта, якщо це відповідає вибраному набору шифрів. Далі сервер надсилає повідомлення «ServerHelloDone» із зазначенням що фаза привітання-повідомлення рукостисканням завершена. Потім сервер чекає відповіді клієнта. Якщо сервер надіслав повідомлення «CertificateRequest», клієнт надішле сертифікат повідомлення. Надалі надіслане повідомлення «ClientKeyExchange» та вміст цього повідомлення буде залежати від обраного алгоритму відкритого ключа між «ClientHello» та «ServerHello». Якщо клієнт надіслав сертифікат із можливістю підпису, цифровий підпис повідомлення «CertificateVerify» надсилається для явної перевірки закритого ключа що міститься у сертифікаті. В цей час клієнт надсилає повідомлення «ChangeCipherSpec» і копіює очікувану специфікацію шифру в поточну специфікацію шифру. Далі клієнт надсилає готове повідомлення за новими алгоритмами, ключами та секретами. У відповідь сервер надсилає власне повідомлення «ChangeCipherSpec», переносить очікування на поточну специфікацію шифру та надсилає готове повідомлення під новою специфікацією

шифру. На цьому, за допомогою повідомлення ACK, рукостискання завершено, і клієнт і сервер можуть почати обмінюватися даними рівня додатків.

Схема роботи протоколу DTLS зображена на рис. 6.

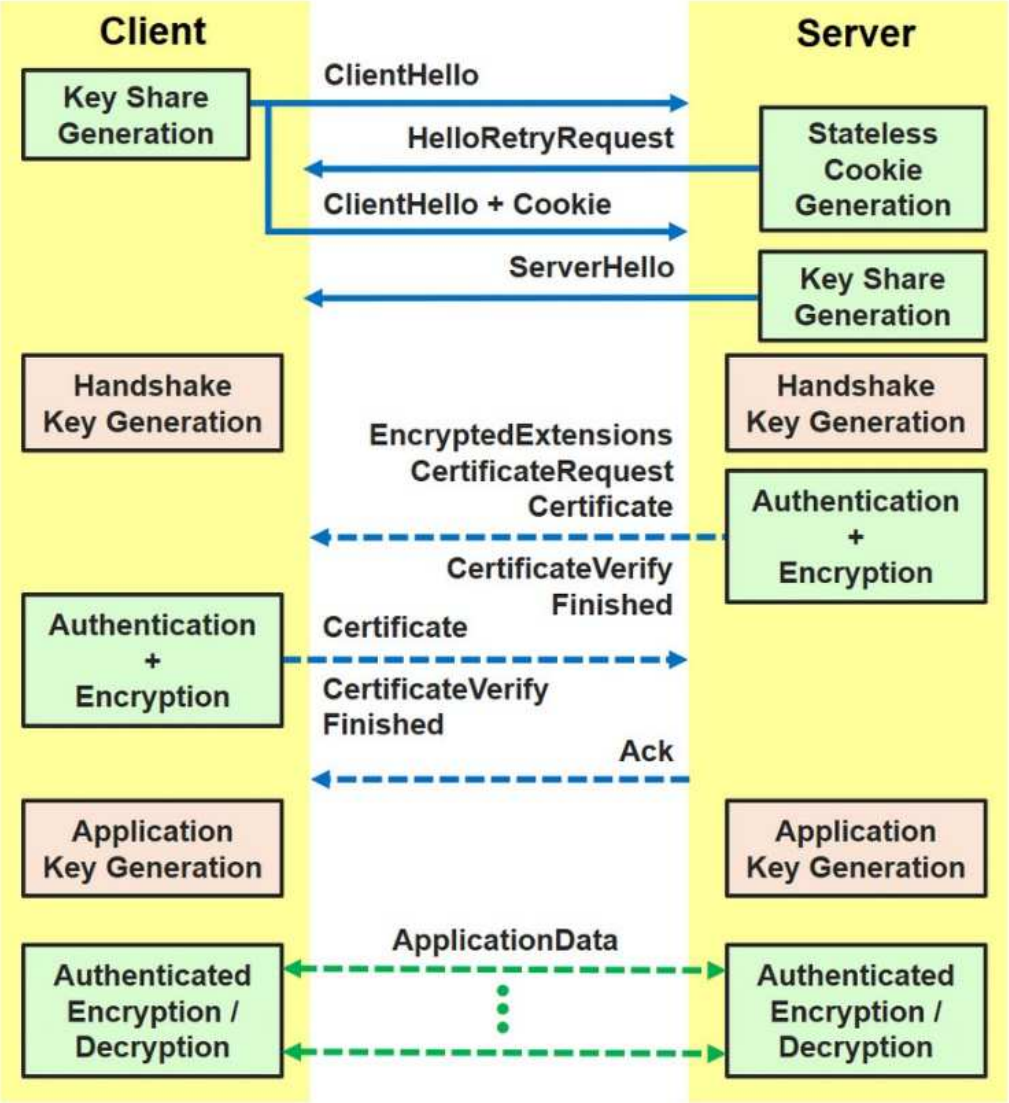

Рисунок 6 – Схема роботи протоколу DTLS [16][1)]

[1)] [16] RFC 6347 - Datagram Transport Layer Security Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc6347 (дата звернення: 26.04.2021)

Повідомлення DTLS групуються у серію повідомлень, як зазначено на рисунку вище. Хоча кожен політ повідомлень може складатися з ряду повідомлень, вони повинні розглядатися в якості моноліту на благо тайм-ауту та ретрансляції.

Формат повідомлення рукостискання [17][1)]:

```
struct {
     HandshakeType msg_type;
     uint24 length;
     uint16 message_seq;
     uint24 fragment_offset;
     uint24 fragment_length;
     select (HandshakeType) {
       case hello_request: HelloRequest;
       case client_hello:  ClientHello;
       case hello_verify_request: HelloVerifyRequest;
       case server_hello:  ServerHello;
       case certificate:Certificate;
       case server_key_exchange: ServerKeyExchange;
       case certificate_request: CertificateRequest;
       case server_hello_done:ServerHelloDone;
       case certificate_verify:  CertificateVerify;
       case client_key_exchange: ClientKeyExchange;
       case finished: Finished;
     } body;
   } Handshake;
```

Кожне повідомлення DTLS повинно вміщуватися в одну дейтаграму транспортного рівня. Однак повідомлення рукостискання є потенційно більшим за максимальний розмір запису. Тому DTLS забезпечує механізм фрагментації повідомлення рукостискання над кількість записів, кожен з яких може передаватися окремо, таким чином уникаючи фрагментації IP.

[1)] [17] Support for DTLS protocol | SSL offload and acceleration – [Електронний ресурс] URL: https://docs.citrix.com/en-us/citrix-adc/current-release/ssl/support-for-dtls-protocol.html (дата звернення: 26.04.2021)

2.3.3 Особливості використання DTLS-SRTP

DTLS-SRTP є розширенням SRTP для DTLS що поєднує в собі переваги продуктивності та гнучкості шифрування SRTP завдяки гнучкості та зручності інтегрованого DTLS ключа та управління асоціаціями [18][1)].

Ключовими моментами DTLS-SRTP є:

- дані програми захищені за допомогою SRTP;
- рукостискання DTLS використовується для встановлення ключа, алгоритмів та параметрів для SRTP;
- розширення DTLS використовується для узгодження алгоритмів SRTP;
- інші типи вмісту рівня запису DTLS захищені за допомогою звичайного формату запису DTLS.

DTLS-SRTP визначений для медіа-сеансів «peer-peer», в яких є рівно двоє учасників. Кожен сеанс DTLS-SRTP містить одинарна асоціація DTLS або два контексти SRTP (якщо медіа-трафік протікає в обох вказівки щодо того самого хосту) або одного контексту SRTP (якщо медіа-трафік рухається лише в одному напрямку). Весь SRTP-трафік перетікаючи цю пару в заданому напрямку, використовує один SRTP контекст. Один сеанс DTLS-SRTP захищає лише передані дані одна пара портів джерела та пункту призначення UDP.

Загальний шаблон DTLS-SRTP [16][2)] такий. Для кожного RTP потік мережевих вузлів робить DTLS рукостискання для того самого джерела та пункту призначення пари портів для встановлення асоціації DTLS. З якої сторони знаходиться DTLS клієнта, а стороною якого є сервер DTLS, потрібно встановити через деякий позасмуговий механізм, такий як SDP. Ключовий матеріал із цього

---

1) [16] RFC 6347 - Datagram Transport Layer Security Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc6347 (дата звернення: 26.04.2021)

2) [18] RFC 5764 - Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP) – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc5764 (дата звернення: 26.04.2021)

рукостискання подається в стек SRTP. Як тільки ця асоціація є встановлені, пакети RTP захищаються (стаючи SRTP), використовуючи це ключовий матеріал.

Між однією парою учасників може бути кілька медіа сесій. Для кожного повинен бути окремий сеанс DTLS-SRTP для кожної чіткої пари джерел та портів призначення, використовуваних медіа-сеансом (хоча сеанси можуть використовувати один сеанс DTLS і, отже, амортизуйте початкове рукостискання з відкритим ключем).

У реалізації, коли є кілька сеансів медіа, є нова установа сеансу DTLS (виконана за допомогою криптографічного алгоритму відкритого ключа) для кожного медіа-каналу. В якості оптимізації використовується DTLS-SRTP реалізація використовує наступну стратегію: єдина DTLS асоціація створена, а всі інші асоціації DTLS чекають поки цей зв'язок не буде встановлений перед тим, як продовжувати їх рукостискання. Ця стратегія дозволяє пізнішим сеансам використовувати DTLS відновлення сесії, що дозволяє амортизувати дорогі операції криптографії з відкритим ключем за допомогою декількох рукостискань DTLS.

Ключі SRTP, що використовуються для захисту пакетів, породжених клієнтом, є на відміну від ключів SRTP, що використовуються для захисту пакетів, створених сервером. Всі джерела RTP, що походять від клієнта для одного і того ж каналу використовує однакові ключи SRTP, і, аналогічно, всі RTP джерела, що надходять на сервер для того самого каналу, використовують ці ж ключи SRTP. Реалізація SRTP забезпечує, щоб усі значення джерела синхронізації (SSRC) для всіх джерел RTP що походять з одного пристрою за одним і тим же каналом, різні, для того, щоб уникнути проблеми «two-time pad».

Висновки до розділу

Обрані, в результаті аналізу предметної області, криптографічні засоби забезпечують необхідну надійність та конфіденційність каналу зв’язку для передачі медіа-інформації та зв’язку з сервером.

Схема захисту веб-сервісу відеоконференцій зображена на рис. 2.

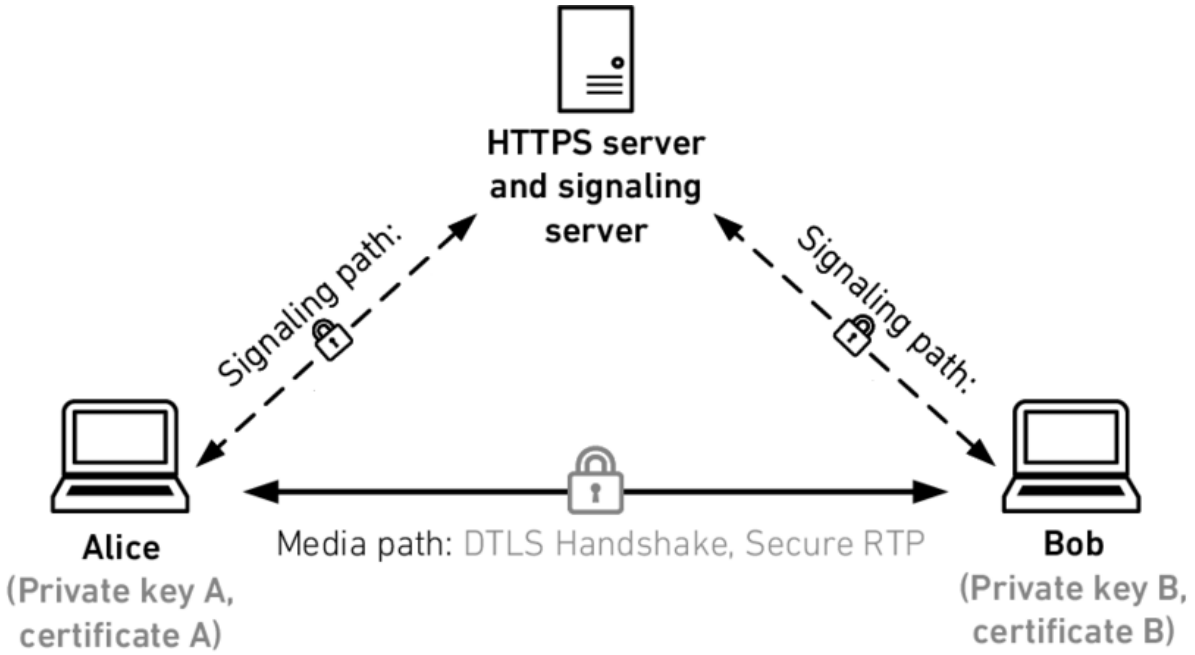

Рисунок 7 – Схема захисту веб-сервісу відеоконференцій

Таким чином веб-сервіс онлайн відеоконференцій є захищеним.

3 ВПРОВАДЖЕННЯ ТЕХНІЧНИХ РІШЕНЬ

У випадку даного проекту, особливо істотним є засоби та інструменти що використовуються та на яких базується додаток. Саме, важливими є наступні фактори, яким засоби повинні задовольняти:

- продуктивність – у випадку додатку відеоконференцій, медіа трафік сприяє важкій напрузі на сервер;
- масштабованість – необхідність даного додатку мати властивість корпоративності, передбачає підтримку особливих властивостей, у випадку підприємств різного характеру, що задовольняють заявленим вимогам;
- простота та доступність – робітники різноманітних підприємств можуть не мати відповідного програмного забезпечення, або навичок для його встановлення та використання, необхідних для використання засобу;
- безпека – додаток передбачає досягнення безпеки переданих та збережених даних шляхом їх шифрування.

3.1 Технологія WebRTC у якості p2p медіа з’єднання між браузерами

У якості технології передачі медіа інформації було використано WebRTC [19][1)]. Даний засіб має в своєму розпорядженні повну підтримку протоколів DTLS та SRTP, що надає бажану безпеку зв’язку. Також, WebRTC підтримується браузерами за замовченням не вимагаючи від користувача встановлення плагінів або будь-якого іншого стороннього програмного забезпечення, що надає доступність для користувачів, їх ОС та програмного забезпечення.

---

1) [19] WebRTC API - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API (дата звернення: 22.11.2021)

3.1.1 Формат обміну медіа інформації SDP

Для опису мультимедійного вмісту з'єднання, такого як роздільна здатність, формати, кодеки, шифрування тощо, в технології WebRTC використовується стандарт SDP (Session Description Protocol) [20][1)][21][2)], таким чином дозволяючи двом одноранговим партнерам розуміти один одного після передачі даних SDP повідомлень. Іншими словами, це метадані, що описують формат медіа даних.

Таким чином, технічно, SDP насправді не є протоколом, а форматом даних, який використовується для опису з'єднання, яке поділяє медіа між пристроями.

SDP складається з одного або кількох рядків тексту UTF-8, кожен із яких починається з одного символу, за яким слідує знак рівності, а за ним структурований текст, що містить значення або опис, формат якого залежить від типу. Рядки тексту, які починаються з даної літери, зазвичай називаються «letter-lines». Наприклад, рядки, що надають опис медіа, мають тип «m», тому ці рядки називаються «m-lines».

Встановлення з'єднання WebRTC між двома пристроями вимагає використання сервера сигналізації, для встановлення зв'язку між терміналами. Саме, завдання сервера сигналізації полягає в передачі опису формату медіа даних – SDP, та опису шляху до терміналу за допомогою ICE, який буде розглянутий пізніше, щоб дозволити двом одноранговим користувачам знайти та встановити з'єднання.

При запуску процесу сигналізації користувач, який ініціює виклик, створює пропозицію. Ця пропозиція містить опис сеансу у форматі SDP і має бути

---

1) [20] WebRTC connectivity - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/ Connectivity (дата звернення: 22.11.2021)

2) [21] Introduction to WebRTC protocols - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/ WebRTC_API/Protocols (дата звернення: 23.11.2021)

доставлена користувачеві-отримувачу, якого можливо назвати абонентом. Абонент відповідає на пропозицію повідомленням відповіді, яке також містить опис SDP. Сервер сигналізації використовуватиме WebSocket для передачі повідомлень про пропозиції типу «video-offer» та відповідей на повідомлення типу «video-answer». Повідомлення складаються з наступних полів [22][1)]:

- type – тип повідомлення; або «video-offer», або «video-answer»;
- name – ім’я користувача відправника;
- target – ім’я користувача для отримання опису;
- sdp – рядок SDP, що описує локальний кінець з’єднання з точки зору відправника (або віддалений кінець з’єднання з точки зору одержувача).

3.1.2 Формат обміну інформації встановлення шляху ICE

На етапі обміну SDP повідомлень обидва учасники знають, які кодеки та параметри кодека мають використовуватися для цього виклику. Але все ще не має можливості передавати самі медіа-дані. Для цієї задачі, саме, і використовується ICE.

ICE (Interactive Connectivity Establishment) [21][2)] – це структура, яка дозволяє веб-браузеру з’єднуватися з одноранговими користувачами. Існує багато причин, чому пряме з’єднання від однорангового узла A до однорангового B не працюватиме. Він повинен обійти брандмауери, які заважають відкривати з’єднання, надати унікальну адресу, якщо, як і більшість ситуацій, пристрій не має загальнодоступної IP-адреси, і передавати дані через сервер, якщо маршрутизатор не

---

1) [22] Signaling and video calling - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/ Signaling_and_video_calling (дата звернення: 23.11.2021)

2) [21] Introduction to WebRTC protocols - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/ WebRTC_API/Protocols (дата звернення: 23.11.2021)

дозволяє безпосередньо з'єднуватися з іншими терміналами. ICE використовує для цього сервери STUN та TURN (які будуть розглянуті в наступному пункті).

Двом терміналам потрібно обмінятися кандидатами ICE, щоб домовитися про фактичне з'єднання між ними. Кожен кандидат у ICE описує метод, який може використовувати для спілкування одноранговий пристрій-відправник. Кожен партнер надсилає кандидатів (кожен кандидат репрезентує унікальний шлях взаємодії через мережу) у тому порядку, в якому вони були виявлені, і продовжує надсилати кандидатів, доки не закінчіться пропозиції, навіть якщо медіа-файли вже почали трансляцію.

Після того, як обидва однорангові термінали домовляться про сумісного кандидата, SDP цього кандидата використовується кожним терміналом для створення та відкриття з'єднання, через яке потім починає надходити медіа. Якщо пізніше вони домовляться про кращого (зазвичай більш продуктивного) кандидата, потік може змінити формат за потребою.

Кожен ICE-кандидат надсилається іншому однорангові шляхом надсилання повідомлення JSON типу «new-ice-candidate» через сервер сигналізації до віддаленого однорангового пристрою. Кожне повідомлення-кандидат містить такі поля [22][1)]:

- type – тип повідомлення: «new-ice-candidate»;
- target – ім'я користувача особи, з якою ведуться переговори; сервер направить повідомлення лише цьому користувачеві;
- candidate – рядок кандидата ICE, що описує запропонований метод підключення.

Кожне повідомлення ICE включає данні про протокол зв'язку (TCP або UDP), IP-адресу, номер порту, тип з'єднання (наприклад, чи є вказаний IP-адрес

[1)] [22] Signaling and video calling - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/ Signaling_and_video_calling (дата звернення: 23.11.2021)

самим терміналом або сервером ретрансляції), а також іншу інформацію, необхідну для з'єднання двох комп'ютерів разом. Це включає NAT або інші складності мережі.

3.1.3 Встановлення p2p зв'язку за допомогою серверів STUN та TURN

Сервери STUN та TURN використовуються протоколом ICE для визначення найкращого маршруту та протоколів для спілкування між одноранговими терміналами, навіть якщо вони знаходяться за брандмауером або використовують NAT.

STUN (Session Traversal Utilities for NAT) – це протокол для виявлення маршруту та визначення будь-яких обмежень у маршрутизаторі, які заважають прямому з'єднанню з терміналом.

Демонстрація роботи серверу STUN зображено на рис.8.

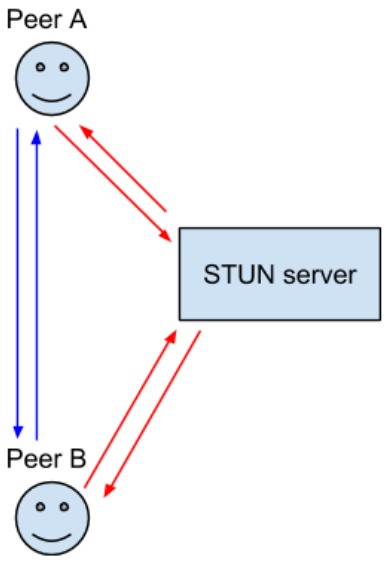

Рисунок 8 – Встановлення зв'язку з залученням сервера STUN

Але навіть з використанням серверу STUN встановлення peer-to-peer зв'язку не є ста процентним. Для цього існує кілька причин, одна з яких полягає в тому, що використовувані пристрої NAT і брандмауер з підвищеною безпекою не дозволяють здійснювати такий прямий трафік. У таких випадках ICE встановлює зв'язок передачі медіа трафіку через проміжний загальнодоступний сервер під назвою TURN (Traversal Using Relays around NAT). Очевидним є те, що такий зв'язок пов'язан з деякими накладними витратами (також таке з'єднання не можна розцінювати як peer-to-peer [23][1)]), тому він використовується лише в тому випадку, якщо немає інших альтернатив.

Передача трафіку зі залученням серверу TURN зображено на рис.9.

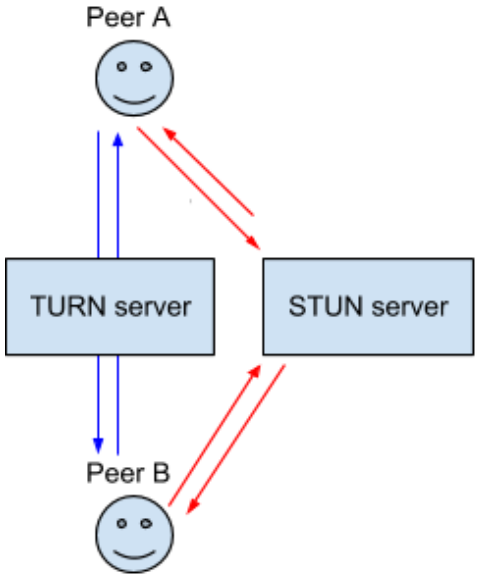

Рисунок 9 – Передача трафіку зі залученням серверу TURN

Таким чином технологія WebRTC досягає необхідного зв'язку між клієнтами мережі за використанням веб браузерів.

[1)] [23] Lifetime of a WebRTC session - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/ Session_lifetime (дата звернення: 23.11.2021)

Діаграма взаємодії між двома користувачами WebRTC зображено на рис.10.

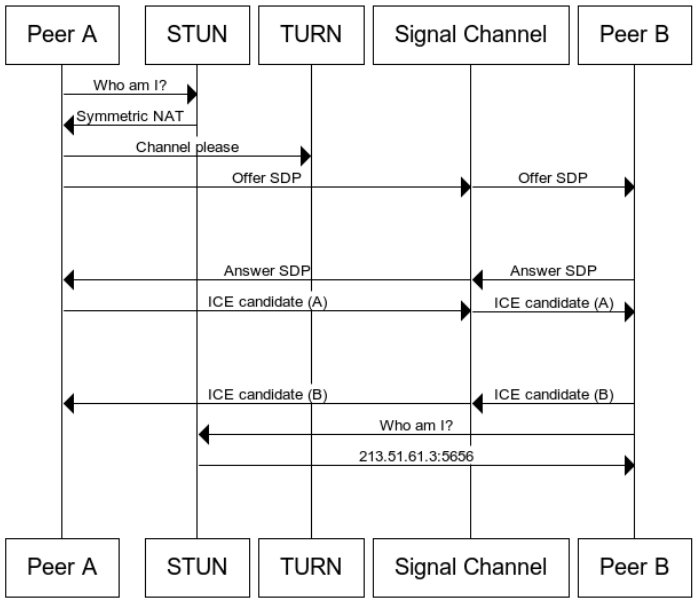

Рисунок 10 – Діаграма взаємодії між двома користувачами WebRTC

3.2 Архітектури групових з'єднань для медіа-зв'язку

WebRTC технологія представляє засіб з'єднання p2p. З метою ж створення групових комунікацій існує ряд рішень що служать архітектурою для забезпечення такого зв'язку.

3.2.1 Архітектура групових комунікацій Mesh

Архітектура Mesh [24][1)] передбачає з'єднання терміналів один з одним, для утворення сітчастої структури. Наприклад, термінали A, B і C здійснюють зв'язок

[1)] [24] WebRTC implementation method(Mesh, SFU, MCU) – [Електронний ресурс] URL: https://millo-l.github.io/WebRTC-implementation-method-Mesh-SFU-MCU/ (дата звернення: 23.11.2021)

багато до багатьох. Коли термінал А хоче поділитися медіа, термінал А повинен надіслати дані на термінал В і термінал С відповідно. Аналогічно, якщо В хоче поділитися медіа, В має надіслати дані А та С тощо. Це рішення має високі вимоги до пропускної здатності кожного терміналу.

Архітектура Mesh виглядає наступним чином (рис.11.):

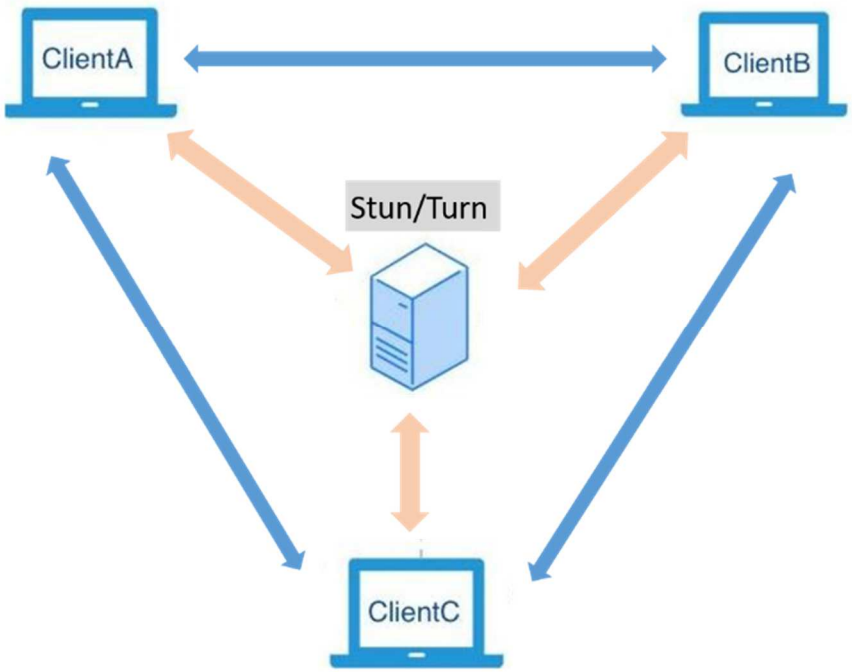

Рисунок 11 – Архітектура Mesh [25][1)]

Як показано на рис. 11., клієнти А, В і С підключаються один до одного і підключаються до сервера STUN/TURN відповідно. Таким чином формується топологія сітки.

Переваги та недоліки архітектури Mesh наведені у табл. 1.

[1)] [25] WebRTC multi-party communication architecture: Mesh, MCU and SFU - Huawei Enterprise Support Community – [Електронний ресурс] URL: https://forum.huawei.com/enterprise/en/webrtc-multi-party-communication-architecture-mesh-mcu-and-sfu/thread/780655-881 (дата звернення: 30.11.2021)

Таблиця 1 – Переваги та недоліки архітектури Mesh

Переваги	Недоліки
Серверу не потрібно пересилати дані. STUN / TURN відповідає тільки за NAT. Може бути реалізована на основі існуючої моделі WebRTC без необхідності розробки медіа-серверу.	При спільному доступі до медіа-потоків кінець спільного доступу повинен надіслати медіа-потік кожному учаснику, який споживає велику кількість пропускної здатності висхідного зв'язку.
Використовуються ресурси пропускної здатності клієнта. Зберігання ресурсів серверу (пропускна здатність сервера зазвичай орендується)	У багатосторонньому спілкуванні, кожен клієнт повинен підтримувати NAT обхід.
	Високі вимоги до ЦП і пам'яті. Рекомендується спілкуватися з менш ніж 4 людьми.

3.2.2 Архітектура групових комунікацій MCU

MCU (Multi-point Control Unit) [24][1)] передбачає групу терміналів що утворюють топологію зірка. Кожен термінал надсилає аудіо- та відео-потоки для спільного використання за допомогою серверу MCU. Сервер MCU змішує аудіо- та відео-потоки всіх терміналів в одній групі користувачів, генерує змішаний аудіо- та відео-потік і надсилає змішаний аудіо- та відео-потік на кожен термінал. Таким чином, кожен термінал може переглядати та чути відео та аудіо інших терміналів.

[1)] [24] WebRTC implementation method(Mesh, SFU, MCU) – [Електронний ресурс] URL: https://millo-l.github.io/WebRTC-implementation-method-Mesh-SFU-MCU/ (дата звернення: 23.11.2021)

Фактично сервер є аудіо- та відео-мікшером. Це рішення створює великий тиск на сервер.

MCU отримує аудіо- та відео-потоки з кожного розділу, декодує потоки, змішує потоки з іншими декодованими аудіо- та відео-потоками та повторно кодує потоки, а потім надсилає змішані потоки всім людям у групі користувачів.

Технологія MCU з'явилася дуже рано в області відеоконференцій, і зараз дуже зріла, в основному використовується в області апаратних відеоконференцій. Модель рішення MCU являє собою зірчасту структуру, як зображено на рис.12.

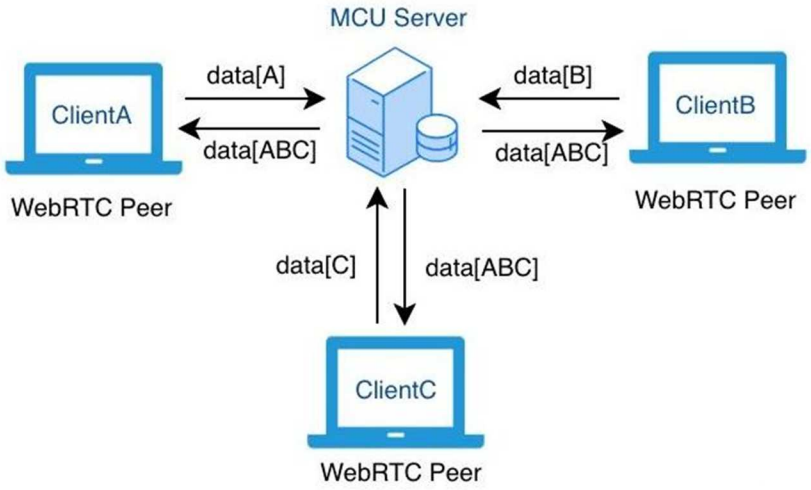

Рисунок 12 – Архітектура MCU [25][1)]

Коли термінал хоче поділитися аудіо- та відео-потоками, термінал надсилає потоки на MCU. Після отримання потоків MCU декодує, змішує, кодує потоки та повертає потоки на термінал. Демонстрація цього процесу зображена на рис.13.

[1)] [25] WebRTC multi-party communication architecture: Mesh, MCU and SFU - Huawei Enterprise Support Community – [Електронний ресурс] URL: https://forum.huawei.com/enterprise/en/webrtc-multi-party-communication-architecture-mesh-mcu-and-sfu/thread/780655-881 (дата звернення: 30.11.2021)

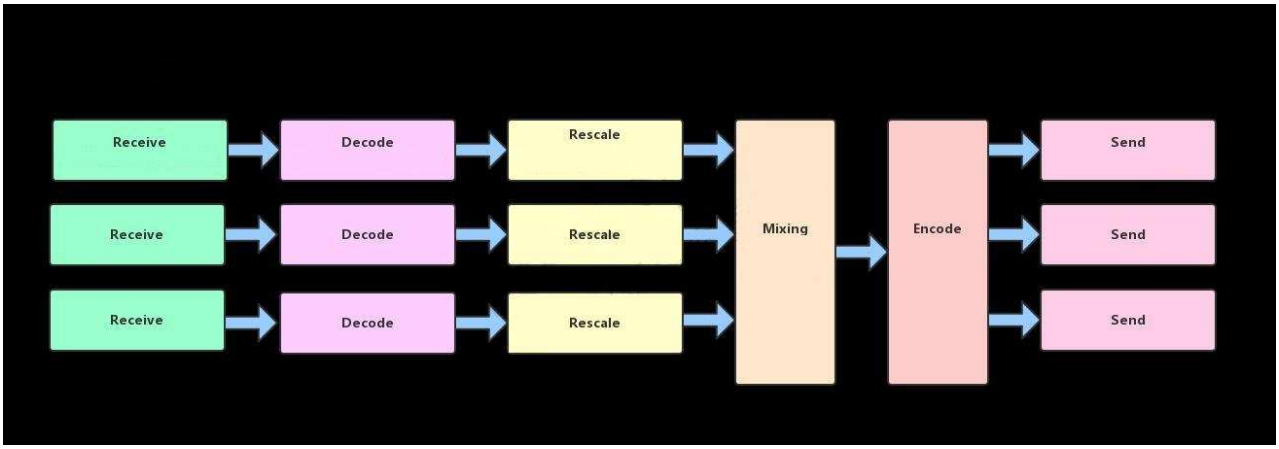

Рисунок 13 – Зображення процесу обробки медіа потоків сервером MCU

Переваги та недоліки архітектури MCU наведені у табл. 2.

Таблиця 2 – Переваги та недоліки архітектури MCU

Переваги	Недоліки
Декодування та перекодування можуть захистити відмінності між різними кодеками, відповідати вимогам інтеграції більшої кількості клієнтів, а також підвищити користувальницький досвід та конкурентоспроможність продукту.	Повторне декодування, кодування та змішані потоки вимагають великої кількості операцій і споживають велику кількість ресурсів процесора.
Багатоканальні відео змішуються в один канал, і всі учасники бачать одну і ту ж картину.	Повторне декодування, кодування та змішування також вводять затримки.
	Потужність, надана MCU, обмежена через високе споживання ресурсів. Як правило, максимальна кількість відеоканалів – десять.

3.2.3 Архітектура групових комунікацій SFU

SFU (Selective Forwarding Unit) [26][1)] передбачає склад з одного сервера та кількох терміналів. На відміну від MCU, SFU не змішує аудіо- та відео-потоки. Після отримання аудіо- та відео-потоків, якими ділиться термінал, аудіо- та відео-потоки безпосередньо пересилаються на інші термінали в групі. Таким чином SFU сервер це аудіо та відео маршрутизатор, або транспондер.

SFU функціонує як маршрутизатор медіа-потоку. Він отримує аудіо- та відео-потоки з терміналів і пересилає потоки на інші термінали, якщо потрібно. SFU широко використовується в аудіо та відеоконференціях, особливо після популяризації WebRTC. Більшість медіа-серверів, які підтримують багатостороннє спілкування WebRTC, використовують структуру SFU.

Модель рішення SFU зображено на рис. 14.

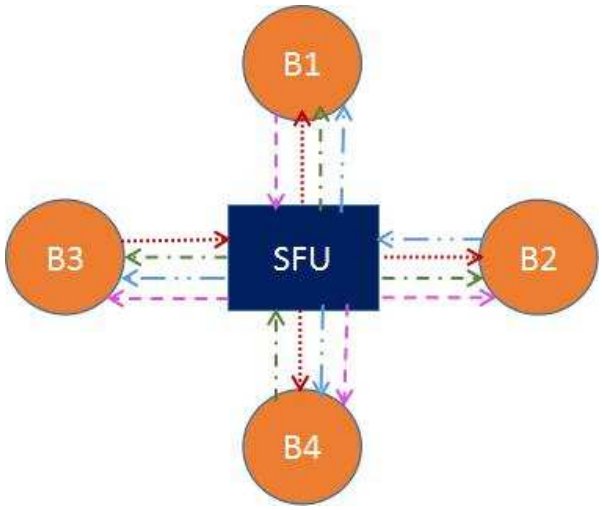

Рисунок 14 – Зображення процесу обробки медіа потоків сервером MCU [25][2)]

[1)] [26] SFU (Selective Forwarding Unit) — Video Conferencing Blog – [Електронний ресурс] URL: https://trueconf.com/blog/wiki/sfu (дата звернення: 30.11.2021)

[2)] [25] WebRTC multi-party communication architecture: Mesh, MCU and SFU - Huawei Enterprise Support Community – [Електронний ресурс] URL: https://forum.huawei.com/enterprise/en/webrtc-multi-party-communication-architecture-mesh-mcu-and-sfu/thread/780655-881 (дата звернення: 30.11.2021)

На рис.14. зображено B1, B2, B3 і B4 представляють чотири браузери. Кожен браузер надає спільний потік SFU, і SFU пересилає кожен потік трьом браузерам, відмінним від спільного користувача.

У порівнянні з MCU, SFU (докладніше за ресурсом [24][1)]) набагато простіше за своєю структурою. Він лише отримує потоки та пересилає їх іншим. Однак ця проста структура сприяє швидкому та гнучкому процесу передачі аудіо та відео. Наприклад, SFU може виконувати управління потоком на основі статусу мережі низхідної лінії зв'язку терміналу і може вибірково відкидати деякі медіадані на основі поточної пропускної здатності та затримки мережі, щоб забезпечити безперервність зв'язку.

Зараз багато SFU підтримують режими SVC і Simulcast для адаптації до різних мережевих умов, таких як мережі Wi-Fi і 4G, а також різні термінали, такі як телефони, планшети та ПК.

Переваги та недоліки архітектури SFU наведені у табл. 3.

Таблиця 3 – Переваги та недоліки архітектури SFU

Переваги	Недоліки
Кодування або декодування не потрібно, що споживає мало ресурсів процесора.	Час перегляду відео учасників не синхронізується або відеозображення несумісні.
Пряма переадресація також значно знижує затримку і покращує продуктивність в режимі реального часу.	
Велика гнучкість для адаптації до різних мережевих умов і типів терміналів	

[1)] [24] WebRTC implementation method(Mesh, SFU, MCU) – [Електронний ресурс] URL: https://millo-l.github.io/WebRTC-implementation-method-Mesh-SFU-MCU/ (дата звернення: 23.11.2021)

3.2.4 Загальний аналіз архітектур

Таким чином через різні обмеження архітектури Mesh, її не є можливим використовувати в реальних сценаріях застосування. Архітектура MCU є зрілою технологією і широко використовується в апаратних відеоконференціях, але використання в якості програмного засобу не є оптимальним засобом. В випадку архітектури SFU, що є популярною та відносно новою архітектурою, в даний час усі багатосторонні комунікаційні медіа-сервери WebRTC використовують саме цю архітектуру. Архітектура SFU гнучка і забезпечує високу продуктивність, та саме вона була використана у якості засобу комунікації групи мережевих клієнтів.

3.3 Програмні рішення серверного додатку

Для розробки серверного додатку були використані наступні застосування: Entity Framework як засіб менеджменту даних, завдяки підвищеної безпеки запитів від SQL ін’єкцій, за допомогою LINQ запитів; ASP.NET MVC як засіб створення серверної частини, завдяки архітектури MVC та масштабованості що вона передбачає.

3.3.1 Entity Framework

Entity Framework [27][1)] – це платформа ORM (object-relational mapping) з відкритим вихідним кодом для додатків .NET, що підтримується Microsoft. EF дозволяє розробникам працювати з даними, використовуючи об’єкти спеціальних класів, не зосереджуючи повну увагу на базових таблицях і стовпцях бази

[1)] [27] Entity Framework Overview - ADO.NET | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ ef/overview (дата звернення: 02.12.2021)

даних, де ці дані зберігаються. За допомогою Entity Framework розробники можуть працювати на більш високому рівні абстракції, при розробці додатків що використовують БД для зберігання даних. Також можливо створювати та підтримувати орієнтовані на дані програми з меншою кількістю коду в порівнянні з традиційними програмами що використовують ADO.NET.

Особливості Entity Framework:

- моделювання – EF створює EDM (Entity Data Model) на основі сутностей POCO (Plain Old CLR Object) із властивостями отримання/встановлення різних типів даних; EDM використовується під час запиту або збереження даних сутності в базі даних;
- запити – EF дозволяє використовувати запити LINQ (C#/VB.NET) для отримання даних із бази даних; постачальник бази даних перекладе ці запити LINQ на мову запитів, специфічну для бази даних (наприклад, SQL для реляційної бази даних); такий підхід також дозволяє вирішити деякі проблеми безпеки, такі як атаки ін’єкцій SQL; але при необхідності EF також дозволяє виконувати прямі запити SQL безпосередньо до бази даних;
- відстеження змін – EF відстежує зміни, що відбулися в екземплярах сутностей (а саме їх значень властивостей), які потрібні для операції до бази даних;
- паралелізм – EF за замовчуванням використовує модель модифікації БД, що передбачає захист змін що були внесені іншим користувачем після отримання даних з бази даних;
- транзакції – EF виконує автоматичне керування транзакціями під час запиту або збереження даних; також э можливість для налаштування управління транзакціями;
- кешування – повторні запити будуть повернуті з кешу, замість того, щоб звертатися до бази даних;

– конфігурація – EF дозволяє налаштовувати модель EF за допомогою атрибутів анотації даних або Fluent API для заміни встановлення умов;
– міграції – EF надає набір команд міграції, які можна виконати на консолі диспетчера пакетів NuGet або в інтерфейсі командного рядка для створення або керування базовою схемою бази даних.

Компоненти Entity Framework [28][1)]:

– EDM (Entity Data Model) – EDM складається з трьох основних частин: концептуальної моделі, моделі відображення та моделі зберігання;
– концептуальна модель – містить класи моделі та їх зв'язки незалежно від дизайну таблиці бази даних;
– модель зберігання – модель проектування бази даних, яка включає таблиці, подання, збережені процедури, а також їхні зв'язки та ключі;
– відображення – складається з інформації про те, як концептуальна модель зіставляється з моделлю зберігання;
– LINQ-to-Entities (L2E) – мова запитів, яка використовується для написання запитів до об'єктної моделі; взаємодія виконується за допомогою сутностей які визначені в концептуальній моделі;
– Entity SQL – мова запитів (лише для EF 6);
– служба об'єктів – є основною точкою входу для доступу до даних із бази даних та повернення їх; служба об'єктів відповідає за матеріалізацію, яка є процесом перетворення даних, повернених від постачальника даних клієнта сутності (наступного рівня), у структуру об'єкта сутності;
– постачальник даних клієнта Entity – основна відповідальність цього рівня полягає в перетворенні запитів LINQ-to-Entities або Entity SQL у запит SQL, який розуміє база даних; він спілкується з постачальником

[1)] [28] Advantages of Entity Framework - Chubby Developer – [Електронний ресурс] URL: https://www.chubbydeveloper.com/advantages-of-entity-framework/ (дата звернення: 02.12.2021)

даних ADO.Net, який, у свою чергу, надсилає або отримує дані з бази даних;

– постачальник даних ADO.Net – рівень зв'язується з базою даних за допомогою стандартного ADO.Net.

Детальна схема архітектури Entity Framework зображено на рис.15.

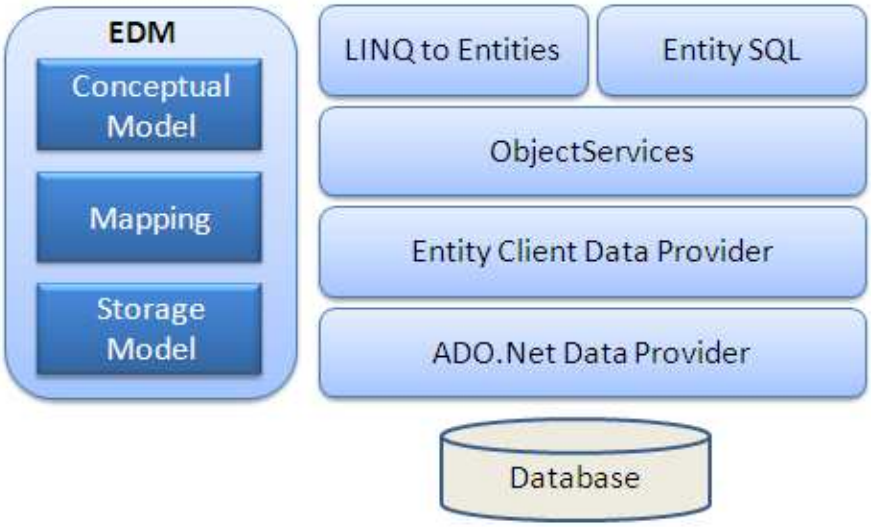

Рисунок 15 – Схема архітектури Entity Framework [27][1)]

3.3.2 ASP.NET MVC

ASP.NET MVC [29][2)] – це фреймворк веб-розробки від Microsoft, який поєднує в собі особливості архітектури MVC (Model-View-Controller), найсучасніші ідеї та методи Agile (гнучка методологія розробки) та найкращі частини існуючі ASP.NET. платформи.

[1)] [27] Entity Framework Overview - ADO.NET | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ ef/overview (дата звернення: 02.12.2021)

[2)] [29] ASP.NET MVC | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/aspnet/mvc/ (дата звернення: 03.12.2021)

Шаблон MVC розділяє введення, обробку та вихід програми. Ця модель розділена на три взаємопов'язані частини, які називаються моделлю (Model), представленням (View) і контролером (Controller).

При розробці програми MVC, контролер отримує всі запити до програми, а потім інструктує модель підготувати будь-яку інформацію, необхідну для представлення. Представлення використовує ці дані, підготовлені контролером, для отримання вихідного результату.

Мета шаблону MVC полягає в тому, щоб кожну з цих частин можна було розробити, протестувати у відносній ізоляції, а також об'єднати для створення надійної програми.

Три рівні моделі MVC:

- Model – це частина програми, яка реалізує логіку для області програми що відповідає за менеджмент даних; вона реалізує отримання і зберігання стану моделі в базі даних; наприклад, об'єкт продукту може отримувати інформацію з бази даних, оперувати ними та записувати інформацію назад у таблицю продуктів;
- View – це компонент, який використовується для відображення інтерфейсу користувача (UI) програми, також званого моделлю перегляду в MVC; він відображає клієнтські додатки ASP.NET MVC, створені за допомогою даних моделі; поширеним прикладом може бути перегляд редагування таблиці, при якому відображається текстові поля, спливаючі вікна та прапорці на основі поточного стану об'єктів;
- Controller – компонент контролер обробляє взаємодію користувача, працює з моделлю та вибирає представлення для відображення Ul; у додатку ASP.NET MVC, в представленні відображається лише інформація яку контролер керує та реагує на введення та взаємодію користувача за допомогою фільтрів дій у MVC; наприклад, контролер керує значеннями рядка запиту і передає ці значення моделі.

Схема патерну MVC, у випадку ASP.NET, зображена на рис.16.

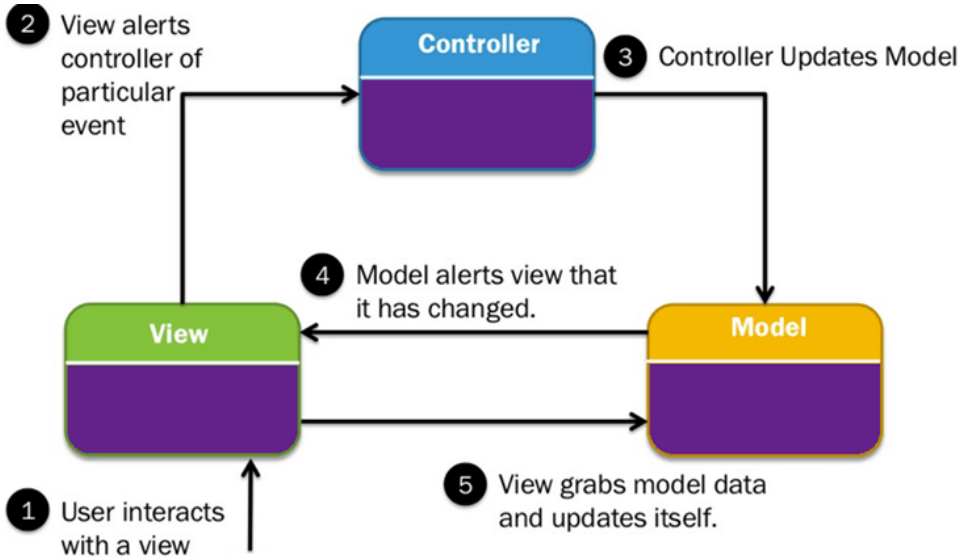

Рисунок 16 – Схема патерну MVC

Переваги ASP.NET MVC [30][1)]:

- спрощене керування структурою програми, завдяки поділу додатка на компоненти; а саме, на модель, подання та контролер;
- забезпечення повного контролю над відтвореним HTML та чіткого поділу проблем;
- забезпечення підтримки розробки на основі тестування (TDD);
- узгоджена розробка додатків, що базується на слабко зв'язаних компонентах; команда розробників може працювати одночасно над моделлю, представленнями та контролером;
- гнучка підтримка кілька механізмів перегляду, таких як Razor, ASPX або користувацький засіб;
- високий рівень підтримки розробником.

[1)] [30] ASP.NET MVC - Overview – [Електронний ресурс] URL: https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_overview.htm (дата звернення: 03.12.2021)

Висновки до розділу

В результаті, у процесі створення додатку були використані наступні основні засоби:

- WebRTC, як засіб медіа з’єднання;
- SFU сервер, як засіб управління груповими з’єднаннями через транслюючий сервер;
- Entity Framework, як засіб розробки бази даних та реалізації нижнього рівня доступу до неї;
- ASP.NET MVC, як засіб створення серверного додатку, що надає функції управління та адміністрації груп, користувачів та засідань за допомогою медіа-з’єднання.

Таким чином дані засоби задовольняють наступним вимогам до системи:

- продуктивність;
- гнучкість;
- масштабованість;
- простота та доступність;
- безпека.

4 РОЗРОБКА СИСТЕМИ ВІДЕОКОНФЕРЕНЦІЙ

В даному розділі наданий опис структури додатку та пояснення до нього. Саме, опис стосується: бази даних; клієнтського додатку, за допомогою якого здійснюється медіа з'єднання; серверного додатку, а точніше надання діаграми класів, використання та послідовності дій.

4.1 Проектування бази даних

Діаграма бази даних додатку зображена на рис.17.

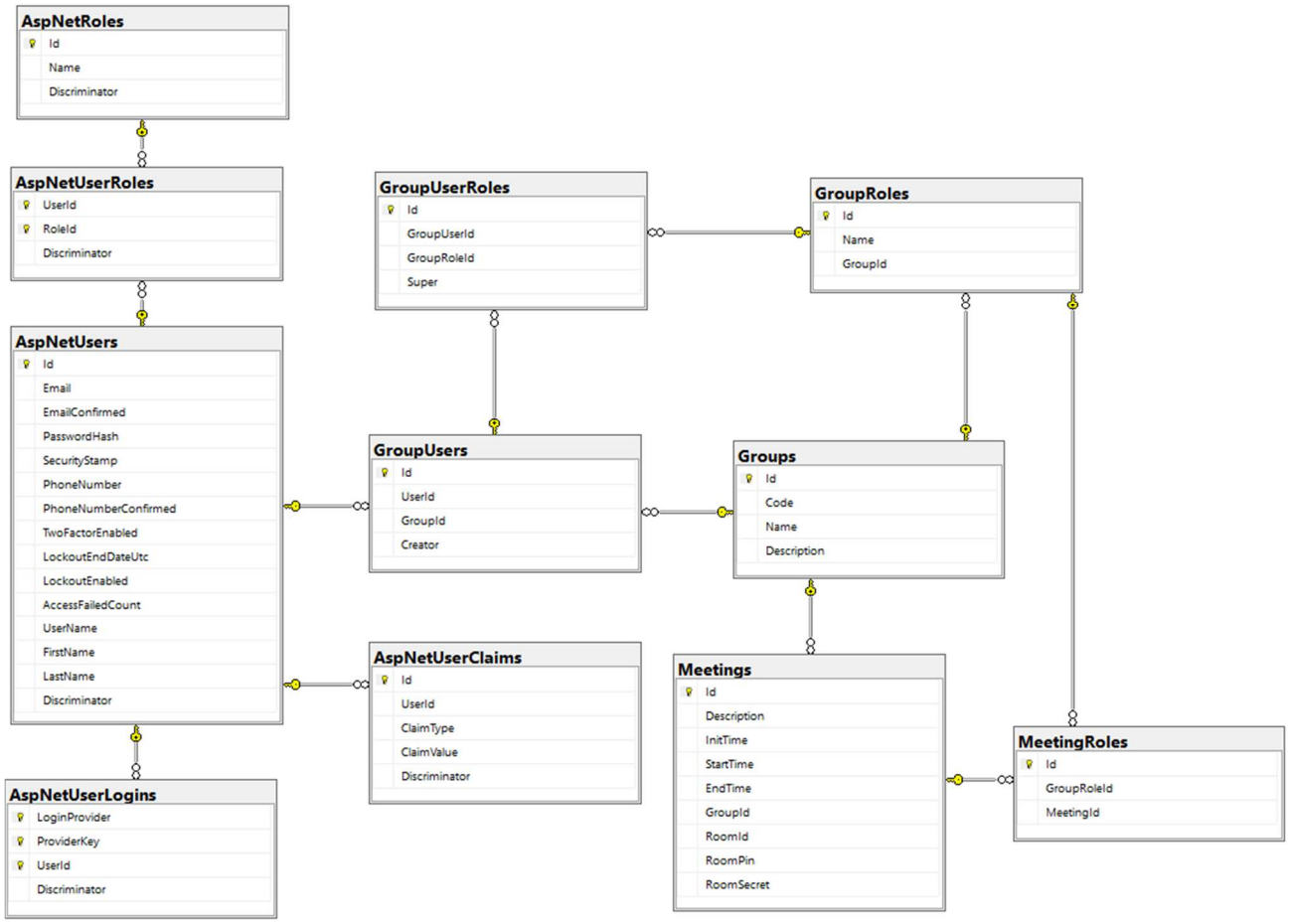

Рисунок 17 – Діаграма бази даних додатку

AspNetUsers – таблиця згенерована системою авторизації та аутентифікації ASP.NET Identity, що репрезентує авторизованих користувачів додатку;

AspNetUserLogins – таблиця згенерована системою авторизації та аутентифікації ASP.NET Identity, що репрезентує облікові записи стосовно користувачів;

AspNetUserClaims – таблиця згенерована системою авторизації та аутентифікації ASP.NET Identity, що репрезентує твердження стосовно корисувачів;

AspNetRoles – таблиця згенерована системою авторизації та аутентифікації ASP.NET Identity, що репрезентує повноваження користування додатком;

AspNetUserRoles – таблиця згенерована системою авторизації та аутентифікації ASP.NET Identity, що репрезентує повноваження користувачів;

Groups – таблиця що репрезентує організацію або групу;

GroupUsers – таблиця що репрезентує учасника організації або групи;

GroupRoles – таблиця що репрезентує посади або характер участі у групі;

GroupUserRoles – таблиця що репрезентує посади стосовно учасників групи;

Meetings – таблиця що репрезентує засідання або зустрічі групи;

MeetingRoles – таблиця що репрезентує посади стосовно запланованих зустріч.

4.2 Розробки серверного додатку

UML діаграма класів, що управляють доступом до джерела даних, зображено на рис.18. (докладніше див. додаток А)

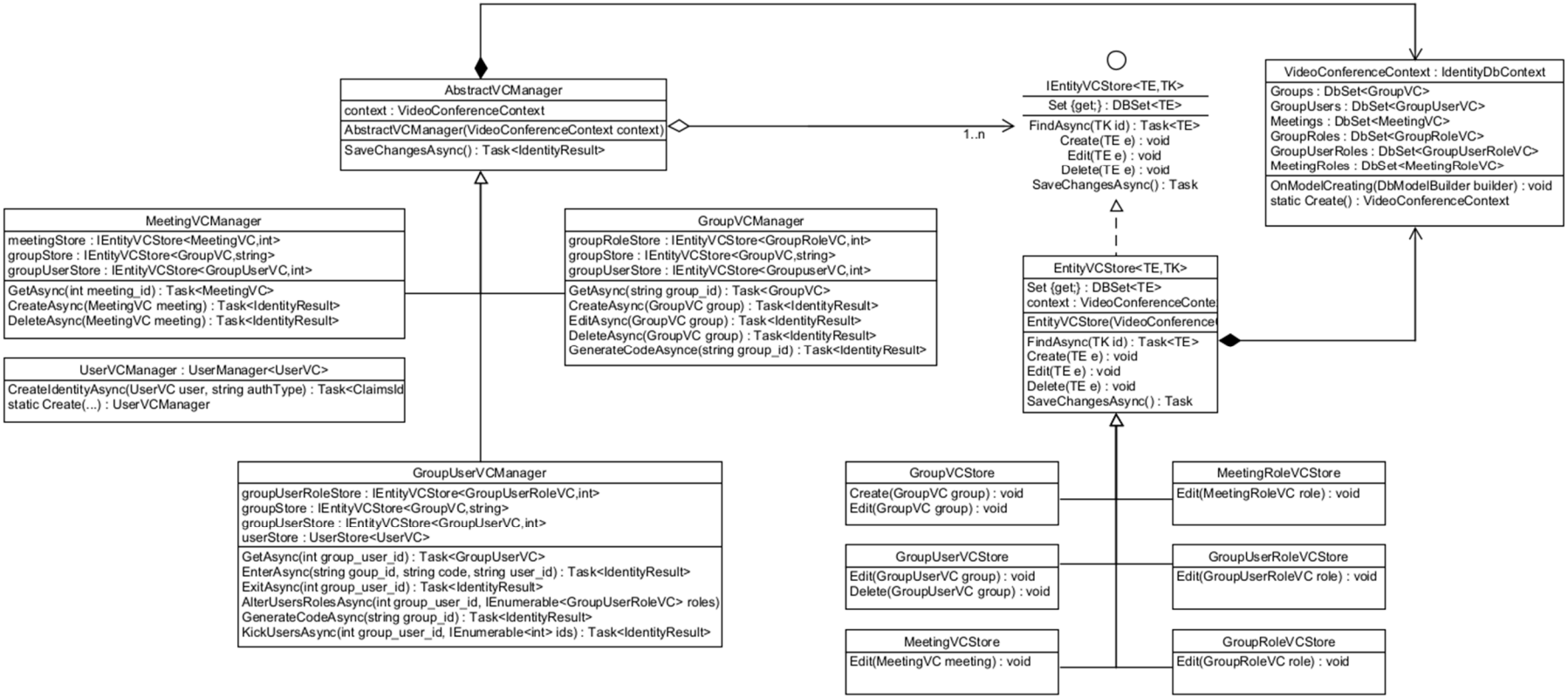

Рисунок 18 – Діаграма класів управління доступу до джерела даних

VideoConferenceContext – клас контексту даних, за допомогою якого здійснюється взаємодія з базою даних.

GroupVCStore, GroupUserVCStore, MeetingVCStore, MeetingRoleVCStore, GroupUserRolesVCStore, GroupRoleVCStore – класи що реалізують основні операції над даними (CRUD).

EntityVCStore – абстрактний клас, від якого унаслідуванні класи CRUD операцій.

IEntityVCStore – інтерфейс, який реалізує абстрактний клас EntityVCStore.

GroupVCManager, GroupUserVCManager, MeetingVCManager – класи що інкапсулюють доступ до даних та операції над ними.

AbstractVCManager – абстрактний клас, від якого унаслідуванні інкапсулюючи класи.

UserVCManager – клас що інкапсулює данні акаунту.

UML діаграма класів, що управляють взаємодією з акаунтом, зображено на рис.19. (докладніше див. додаток Б)

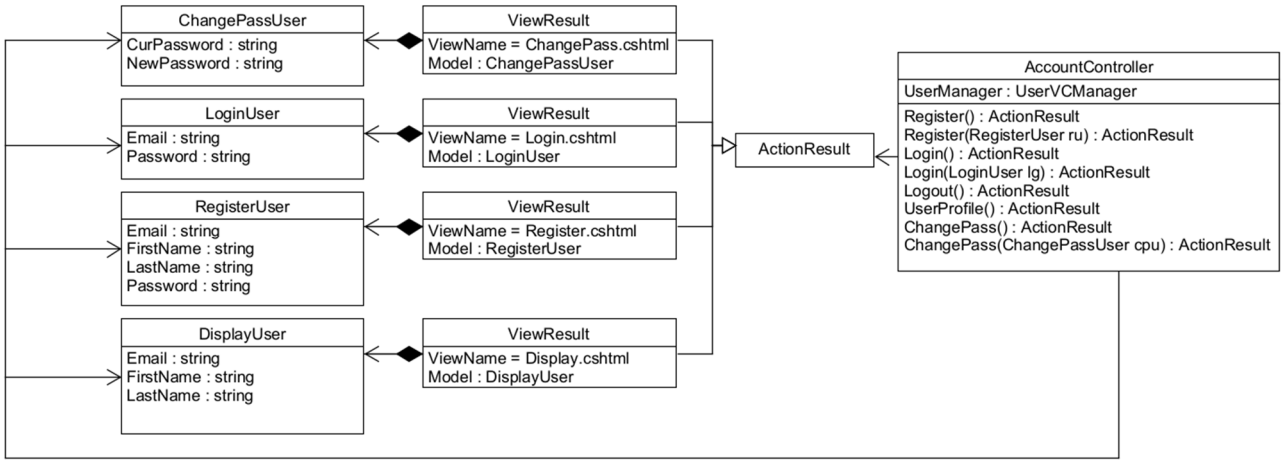

Рисунок 19 – Діаграма класів управління акаунтом

AccountController – клас контролер для взаємодії користувача з акаунтом.

Login.cshtml, Register.cshtml, Display.cshtml, ChangePass.cshtml – сторінки для компіляції у HTML.

LoginUser, RegisterUser, DisplayUser, ChangePassUser – класи, моделі представлень.

UML діаграма класів, що відповідають за менеджмент груп, зображено на рис.20. (докладніше див. додаток В)

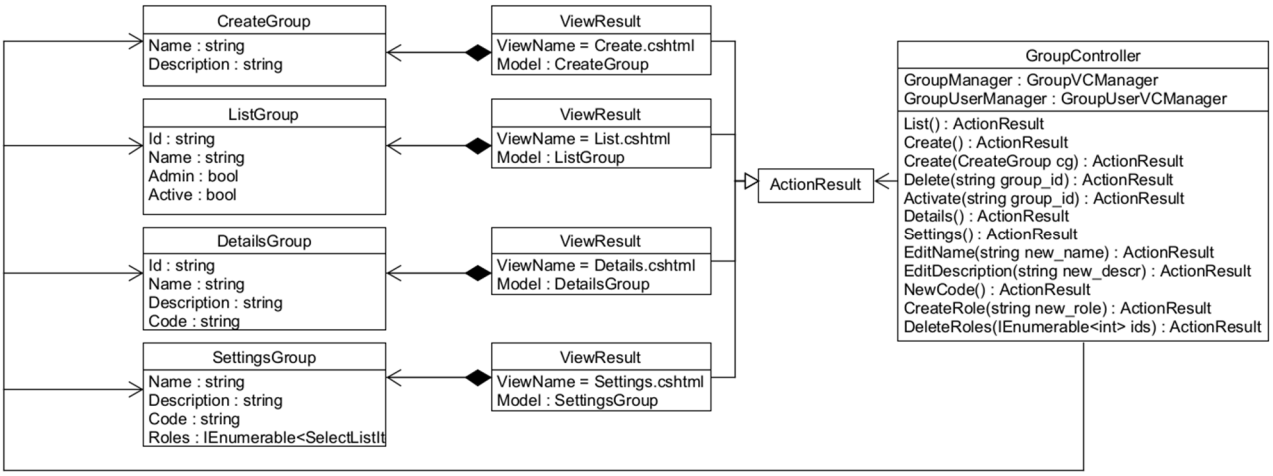

Рисунок 20 – Діаграма класів менеджменту груп

GroupController – клас контролер для взаємодії користувача з налаштуваннями груп.

Create.cshtml, List.cshtml, Details.cshtml, Settings.cshtml – сторінки для компіляції у HTML.

CreateGroup, ListGroup, DetailsGroup, SettingsGroup – класи, моделі представлень.

UML діаграма класів, що відповідають за менеджмент участі в групі, зображено на рис.21. (докладніше див. додаток Г)

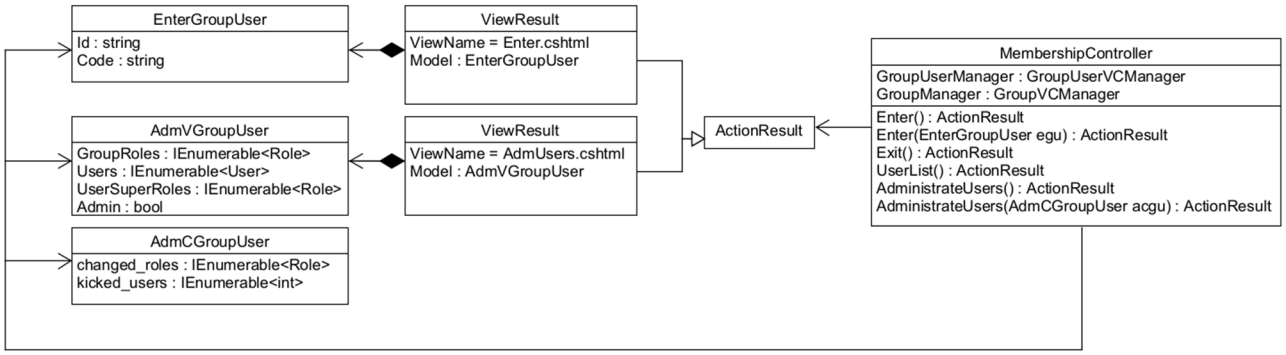

Рисунок 21 – Діаграма класів менеджменту участі у групі

MembershipController – клас контролер для взаємодії користувача з налаштуваннями участі у групі (своєї та інших).

Enter.cshtml, AdmUsers.cshtml – сторінки для компіляції у HTML.

EnterGroupUser, AdmVGroupUser – класи, моделі представлень.

AdmCGroupUser – клас прив’язки даних.

UML діаграма класів, що відповідають за управління засіданнями, зображено на рис.22. (докладніше див. додаток Д)

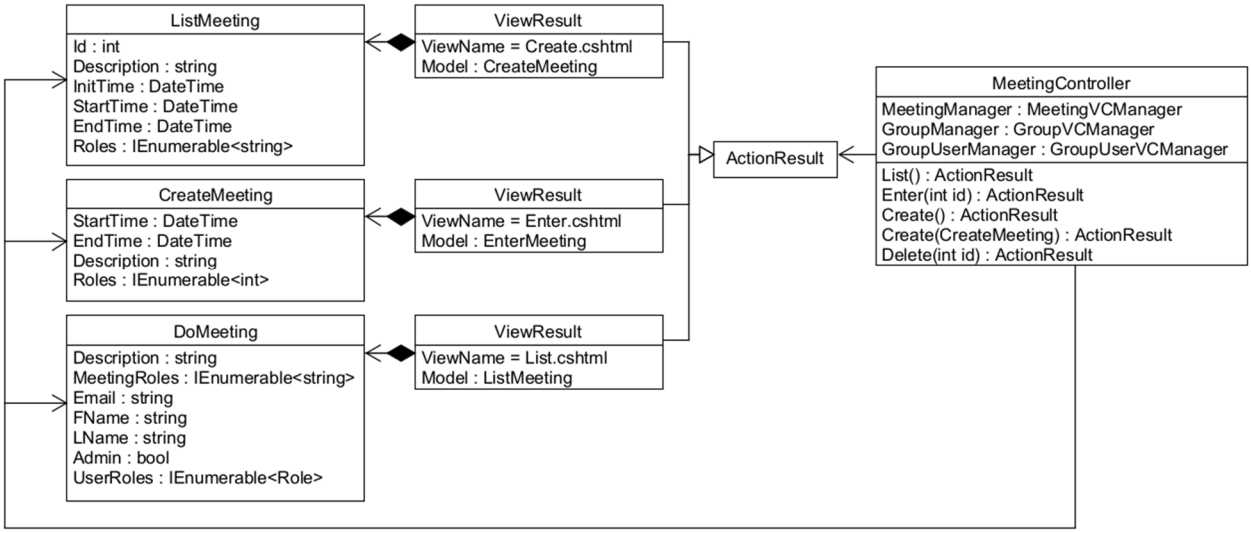

Рисунок 22 – Діаграма класів менеджменту засідань

MeetingController – клас контролер для взаємодії користувача з використанням засідань.

Create.cshtml, List.cshtml, Enter.cshtml – сторінки для компіляції у HTML.

CreateMeeting, ListMeeting, DoMeeting – класи, моделі представлень.

UML діаграма використання, що описує взаємодію з додатком, зображено на рис.23.

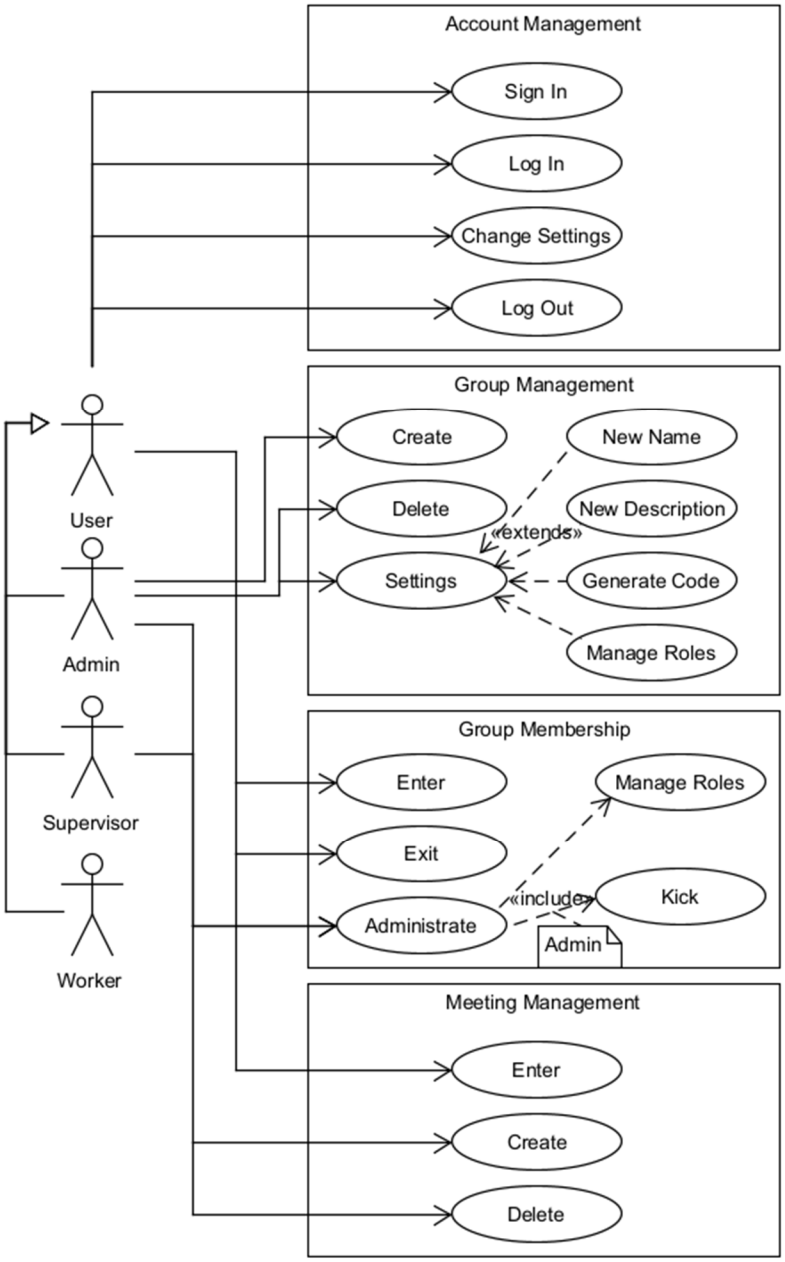

Рисунок 23 – Діаграма взаємодії з додатком

UML діаграма послідовності дій, що описує процес від створення групи до проведення зустрічі за допомогою відео-зв’язку, зображено на рис.24.

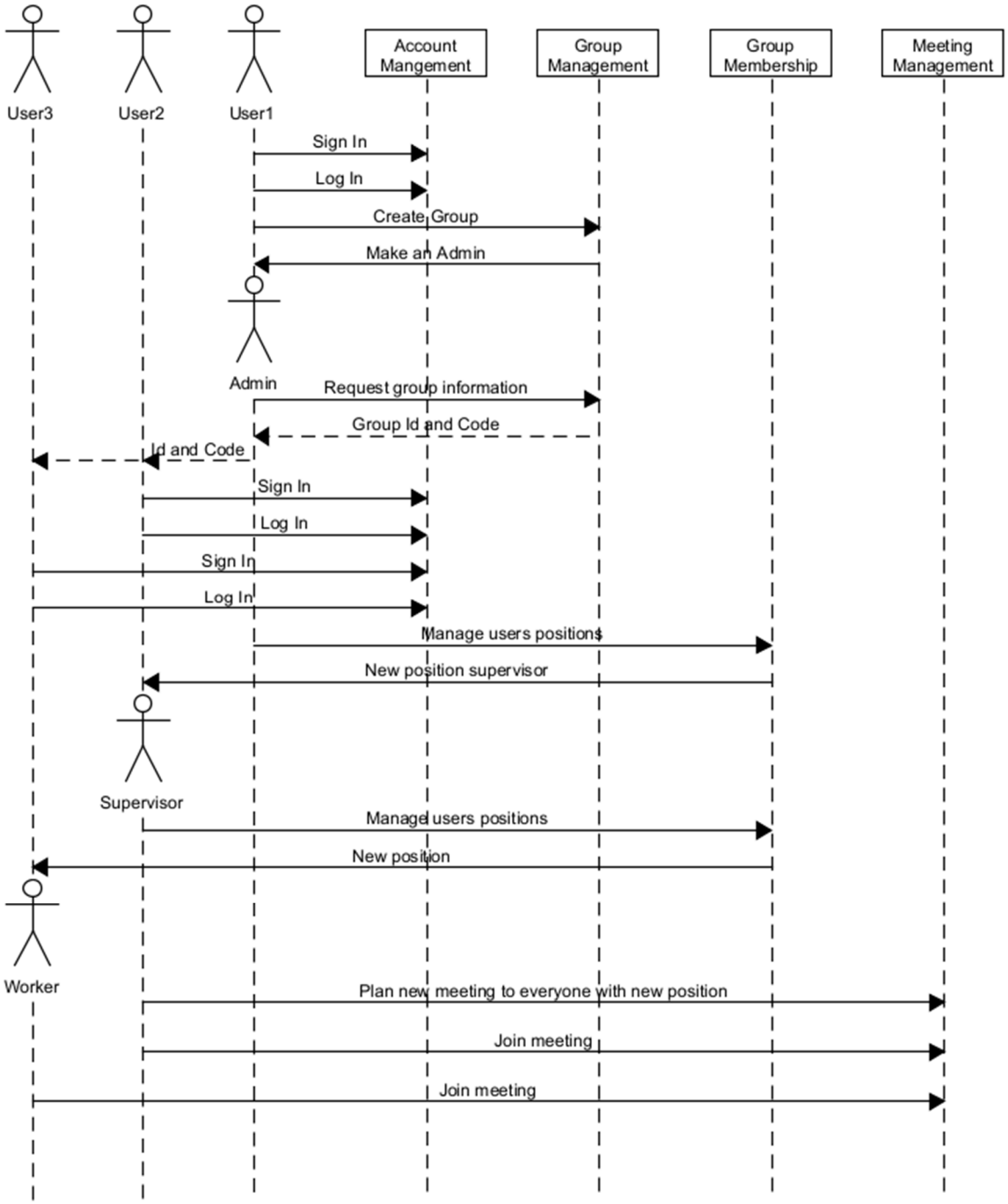

Рисунок 24 – Діаграма послідовності дій

4.3 Розробка клієнтського API медіа-зв’язку

UML діаграма класів, що описує структуру класів взаємодії з SFU сервером, на стороні клієнта, для медіа-зв’язку, зображено на рис.25.

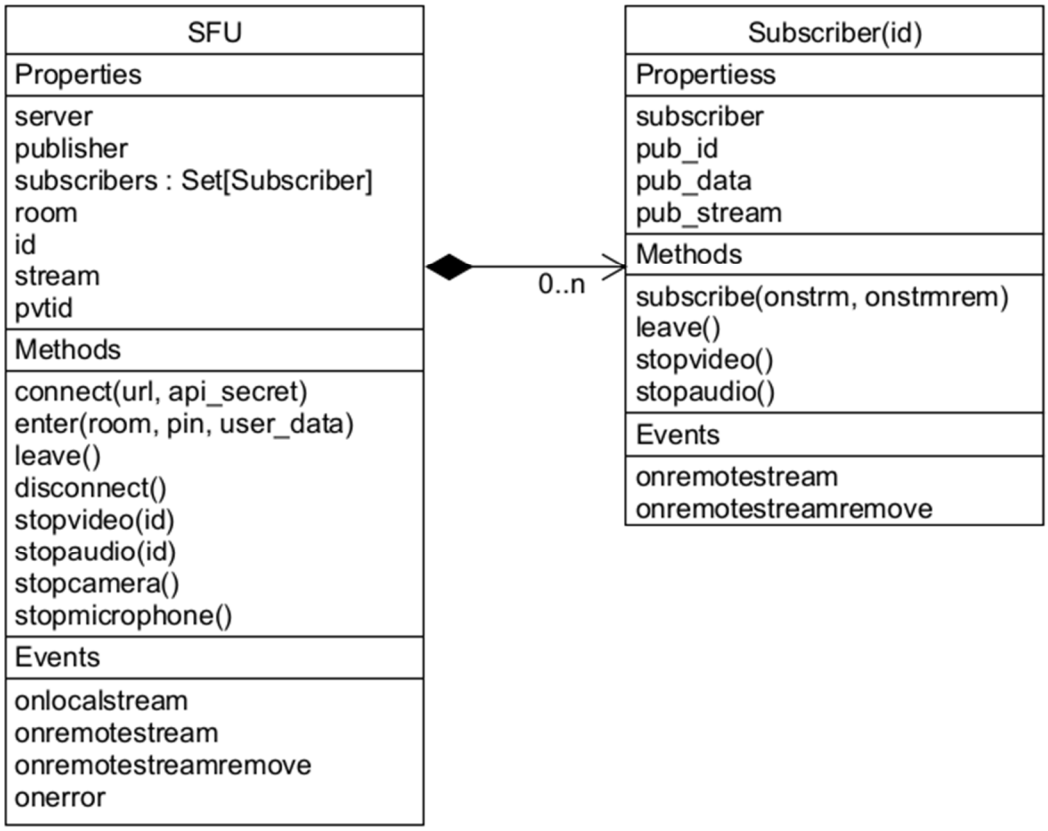

Рисунок 25 – Діаграма послідовності дій

SFU – об’єкт для взаємодії з SFU сервером.

Властивості SFU об’єкту:

– server – об’єкт з’єднання до SFU серверу;
– publisher – об’єкт плагін що публікує медіа-потік на SFU сервер;
– subscribers – об’єкти конструктору Subscriber(id);
– room – ідентифікатор відео-кімнати;

– id – ідентифікатор користувача в поточному сеансі;
– stream – публікований користувачем медіа-потік;
– pvtid – приватний ідентифікатор користувача.

Методи SFU об’єкту:

– connect(url, api_secret) – з’єднання з SFU сервером;
– enter(room, pin, user_data) – вхід у кімнату;
– leave() – вихід з кімнати;
– disconnect() – роз’єднання з SFU сервером;
– stopvideo(id) – припинення завантаження відео-потоку;
– stopaudio(id) – припинення завантаження аудіо-потоку;
– stopcamera() – відключення камери;
– stopmicrophone() – відключення мікрофону.

Події SFU об’єкту:

– onlocalstream – подія додавання нового локального медіа-потоку;
– onremotestream – подія додавання нового віддаленого медіа-потоку;
– onremotestreamremove – подія видалення віддаленого медіа-потоку;
– onerror – подія помилки при взаємодії з SFU сервером.

Subscriber(id) – конструктор для створення об’єктів, що завантажують опубліковані потоки з SFU сервера.

Властивості об’єкту створеного за допомогою конструктора Subscriber(id):

– subscriber – об’єкт плагін що завантажує медіа-потік з SFU серверу;
– pub_id – ідентифікатор віддаленого користувача в поточному сеансі;
– pub_data – данні віддаленого користувача;
– pub_stream – медіа-потік віддаленого користувача.

Методи об’єкту створеного за допомогою конструктора Subscriber(id):

– subscribe(onstrm, onstrmrem) – початок завантаження віддаленого медіа-потоку;

– leave() – кінець завантаження віддаленого медіа-потоку;
– stopvideo() – припинення завантаження віддаленого відео-потоку;
– stopaudio() – припинення завантаження віддаленого аудіо-потоку.

Події об’єкту створеного за допомогою конструктора Subscriber(id):

– onremotestream – подія додавання нового віддаленого медіа-потоку;
– onremotestreamremove – подія видалення віддаленого медіа-потоку;

4.4 Тестування системи відеоконференцій

Для кінцевого тестування, в якості прикладу, було створено групу що представляє проект по розробці гри. Також було зареєстровано та включено в групу п’ять учасників. Результат цих дій можливо побачити на рис.26.

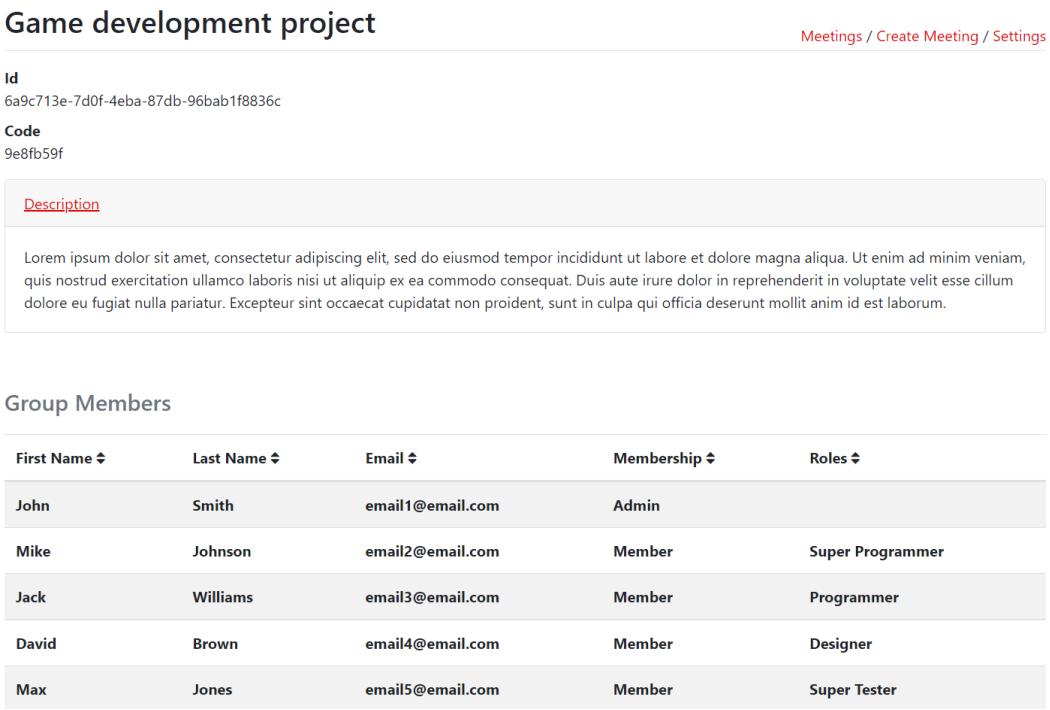

Рисунок 26 – Інформаційна сторінка групи

Наступним етапом кінцевого тестування було, у відношенні групи, створення ролей та добавлення відповідних до учасників цієї групи. Результатом виконання кроків є налаштування що зображенні на рис.27.

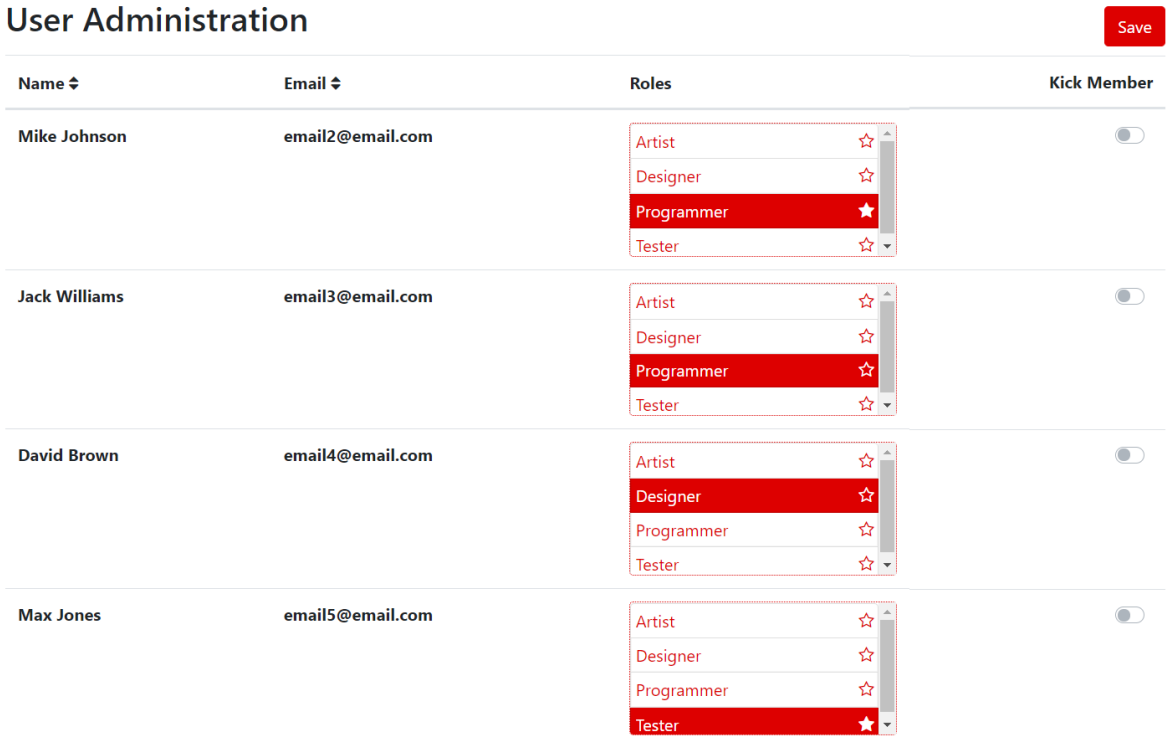

Рисунок 27 – Сторінка адміністрування учасників групи

Слідом було протестоване створення конференцій та їх адміністрування. Підсумок поточних дій зображено на рис.28.

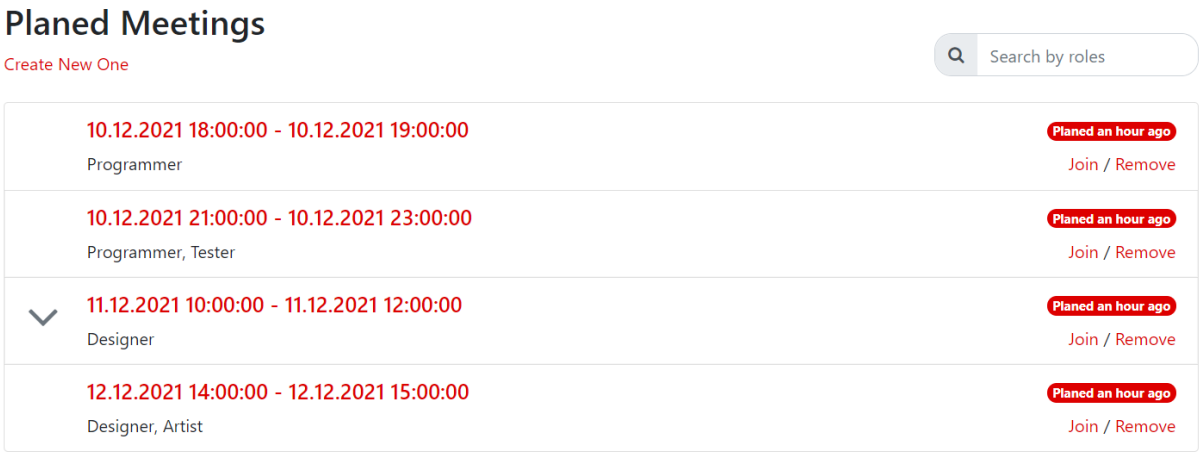

Рисунок 28 – Сторінка з поточними конференціями групи

Наступним був протестований процес проводження конференцій, налаштування поточної конференції та впровадженні рішення стосовно конференцій. Докладніше, стосовно даних кроків тестування зображено на рис.29.

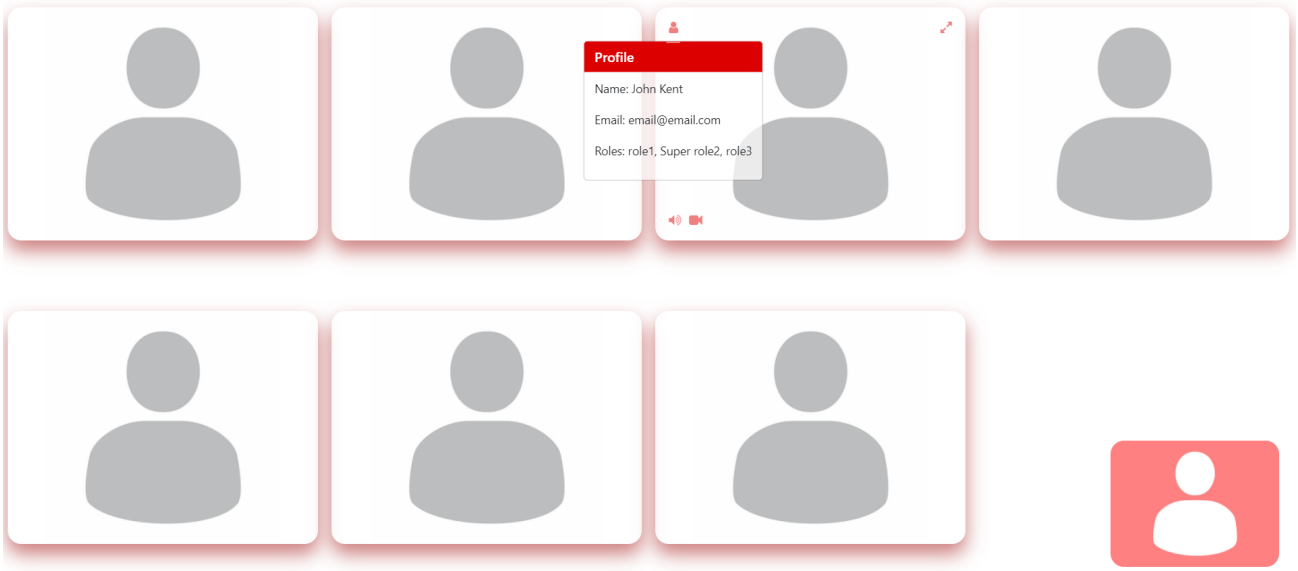

Рисунок 29 – Сторінка проводження відео-конференцій

Також були протестовані деякі інші функції стосовно додатку. Такі як налаштування групи, уповноваження користувачів щодо дій що виконуються, зв’язок з SFU сервером клієнтського та серверного додатків, стабільність медіа зв’язку та деякі додаткові функції.

Висновки до розділу

При розробці додатку були виконані наступні дії:

- проектування системи в загалом, с позиції її функціоналу та користувальницького досвіду;
- проектування бази даних;
- реалізація спроектованої бази даних та її налаштувань у серверному додатку, засобами Entity Framework;
- проектування та кодування системи доступу до сховища даних;
- проектування та кодування системи класів та представлень взаємодії з користувачем;
- проектування та кодування клієнтського додатку групового відеозв’язку за допомогою SFU сервера;
- тестування додатку.

В результаті було розроблено захищену систему відеоконференцій, що задовольняє всім поставленим вимогам.

ВИСНОВКИ

При виконанні даного проекту було проведено:

- аналіз проблематики відеоконференцій;
- аналіз засобів захисту онлайн конференцій;
- впровадження технічних рішень;
- розробка системи відеоконференцій.

Таким чином, як результат, для даного проекту були досягнуті необхідні задачі з безпеки, корпоративно-ділової направленості, можливої масштабованості та продуктивності зв’язку медіа типу.

Даний проект представляє ефективний інструмент ведення корпоративних комунікацій та є перспективним зачатком конкурентоспроможного веб-застосування за рахунок малочисельної кількості аналогів подібного характеру.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Microsoft Word - Methods Of Communication – [Електронний ресурс] URL: http://studymaterial.unipune.ac.in:8080/jspui/bitstream/123456789/4736/1/Methods%20of%20Communication.pdf (дата звернення: 21.04.2021)
2. Different Methods and Types of Business Communication – [Електронний ресурс] URL: https://wikifinancepedia.com/finance/business-planning/what-are-the-different-methods-modes-and-types-of-business-communication-systems (дата звернення: 21.04.2021)
3. The Business Benefits of Video Conferencing | ViewSonic Library – [Електронний ресурс] URL: https://www.viewsonic.com/library/business/ business-benefits-of-video-conferencing/ (дата звернення: 22.04.2021)
4. The Business Benefits of Video Conferencing | Parmetech – [Електронний ресурс] URL: https://www.parmetech.com/the-business-benefits-of-video-conferencing/ (дата звернення: 22.04.2021)
5. 54 Basic Video and Web Conferencing Statistics: 2020/2021 Analysis of Data & Market Share - Financesonline.com – [Електронний ресурс] URL: https://financesonline.com/video-web-conferencing-statistics/ (дата звернення: 22.04.2021)
6. Video Conferencing Tools – [Електронний ресурс] URL: http://www.ecommerce-digest.com/video-conferencing.html (дата звернення: 23.04.2021)
7. Video Conferencing Technology – [Електронний ресурс] URL: https://www.vocal.com/video/video-conferencing-technology/ (дата звернення: 23.04.2021)
8. Video Conferencing Technology Trends Shaping The Future Of Communication – [Електронний ресурс] URL: https://www.shure.com/en-US/conferencing-

meetings/ignite/video-conferencing-technology-trends-shaping-the-future-of-communication/ (дата звернення: 23.04.2021)

9. Transport Layer Security - Wikipedia – [Електронний ресурс] URL: https://en.wikipedia.org/wiki/Transport_Layer_Security (дата звернення: 24.04.2021)
10. RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc5246 (дата звернення: 24.04.2021)
11. Transport Layer Security protocol | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/windows-server/security/tls/transport-layer-security-protocol (дата звернення: 24.04.2021)
12. Secure Real-time Transport Protocol - Wikipedia – [Електронний ресурс] URL: https://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol (дата звернення: 25.04.2021)
13. RFC 3711 - The Secure Real-time Transport Protocol (SRTP) – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc3711 (дата звернення: 25.04.2021)
14. [MS-SRTP]: Secure Real-time Transport Protocol (SRTP) Profile | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/openspecs/office_protocols/ms-srtp/d9641c95-b152-4cc7-8311-d178f3241f1f (дата звернення: 25.04.2021)
15. Datagram Transport Layer Security - Wikipedia – [Електронний ресурс] URL: https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security (дата звернення: 26.04.2021)
16. RFC 6347 - Datagram Transport Layer Security Version 1.2 – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc6347 (дата звернення: 26.04.2021)
17. Support for DTLS protocol | SSL offload and acceleration – [Електронний ресурс] URL: https://docs.citrix.com/en-us/citrix-adc/current-release/ssl/support-for-dtls-protocol.html (дата звернення: 26.04.2021)

18. RFC 5764 - Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP) – [Електронний ресурс] URL: https://tools.ietf.org/html/rfc5764 (дата звернення: 26.04.2021)
19. WebRTC API - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API (дата звернення: 22.11.2021)
20. WebRTC connectivity - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Connectivity (дата звернення: 22.11.2021)
21. Introduction to WebRTC protocols - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Protocols (дата звернення: 23.11.2021)
22. Signaling and video calling - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Signaling_and_video_calling (дата звернення: 23.11.2021)
23. Lifetime of a WebRTC session - Web APIs | MDN – [Електронний ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Session_lifetime (дата звернення: 23.11.2021)
24. WebRTC implementation method(Mesh, SFU, MCU) – [Електронний ресурс] URL: https://millo-l.github.io/WebRTC-implementation-method-Mesh-SFU-MCU/ (дата звернення: 23.11.2021)
25. WebRTC multi-party communication architecture: Mesh, MCU and SFU - Huawei Enterprise Support Community – [Електронний ресурс] URL: https://forum.huawei.com/enterprise/en/webrtc-multi-party-communication-architecture-mesh-mcu-and-sfu/thread/780655-881 (дата звернення: 30.11.2021)
26. SFU (Selective Forwarding Unit) — Video Conferencing Blog – [Електронний ресурс] URL: https://trueconf.com/blog/wiki/sfu (дата звернення: 30.11.2021)

27. Entity Framework Overview - ADO.NET | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview (дата звернення: 02.12.2021)
28. Advantages of Entity Framework - Chubby Developer – [Електронний ресурс] URL: https://www.chubbydeveloper.com/advantages-of-entity-framework/ (дата звернення: 02.12.2021)
29. ASP.NET MVC | Microsoft Docs – [Електронний ресурс] URL: https://docs.microsoft.com/en-us/aspnet/mvc/ (дата звернення: 03.12.2021)
30. ASP.NET MVC - Overview – [Електронний ресурс] URL: https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_overview.htm (дата звернення: 03.12.2021)