

ДИПЛОМНА РОБОТА МАГІСТРА

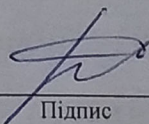
на тему Інформаційна система для соціальної мережі університету
Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Виконав: студент 2 курсу, група КНМ-19-1


Підпис

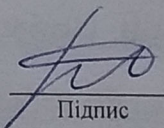
О.С. Ткачук
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри
КНІТ


Підпис

Р.О. Багрій
Ініціали, прізвище

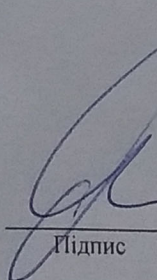
Нормоконтроль: к.т.н., доцент кафедри
КНІТ


Підпис

Р.О. Багрій
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор


Підпис

О.В. Бармак
Ініціали, прізвище

7 12 2020 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій


(підпис)

д.т.н., професор О.В. Бармак

« 7 » 9 2020 року

**ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ МАГІСТРА**

1. Тема дипломної роботи магістра: «Інформаційна система для соціальної мережі університету»

2. Завдання видано студенту Ткачуку Олександру Сергійовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи к.т.н., доцент Багрій Руслан Олександрович
(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від « 9 » 9 2020 р. № 22

5. Зміст пояснювальної записки (перелік питань, які треба розробити):

Метою магістерської роботи є дослідження та розробка інформаційної системи соціальної мережі університету для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання. Для цього необхідно дослідити та адаптувати існуючу систему онлайн-комунікації. На основі даної інформаційної системи необхідно створити або удосконалити додаток для онлайн-комунікації.

Реферат

Дипломна робота магістра присвячена розробці інформаційної системи соціальної мережі університету для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання.

Актуальність теми. В наш час невід'ємною частиною життя стали онлайн-комунікації в повсякденному житті. Спочатку вони з'явилися у виді електронних листів, чатів, форумів з розвитком інтернету з'явилися соціальні мережі різні месенджери, які згодом отримали можливість здійснювати аудіо/відео дзвінки. Зараз стало нормальним спілкування за допомогою комп'ютерної мережі інтернет, більшість спілкування і комунікацій перейшла в онлайн, значне місце займають соціальні мережі.

Зараз в зв'язку з пандемією на перший план виходять онлайн-комунікації зокрема відео конференції які використовуються наприклад для дистанційного навчання, що накладає на них певні вимоги щодо використання їх в сучасних умовах. Через що відбувається розвиток онлайн-комунікацій наприклад таких як Zoom, Google Meet та інші які надають змогу проводити дистанційне навчання, конференції тощо. Але на момент появи пандемії навчальні заклади не були готові до таких змін у навчанні, коли різні ресурси повністю не задовольняють потрібним нам вимогам, навчання стало менш продуктивним, викладачу важко відслідковувати присутність та активністю студентів, комунікація стала більш складною так як немає однієї загальної системи для усього.

Запропонована інформаційна система для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання.

Кінцевою метою застосування розроблено інформаційної системи, є використання її при дистанційному навчанні, що забезпечить зручність та вирішення проблем з проведенням його.

Мета і задачі роботи. Метою магістерської роботи є дослідження та розробка інформаційної системи соціальної мережі університету для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання.

Для досягнення поставленої мети визначено наступні задачі:

- дослідити та проаналізувати види комунікацій та онлайн-комунікацій;
- провести аналіз існуючих сервісів відео-конференцій;
- спроектувати та розробити інформаційну систему на основі існуючих аналогів відео-конференцій.

Також інформаційна система повинна забезпечити визначені вимоги для дистанційного навчання.

Об’єкт дослідження – онлайн-комунікація в дистанційному навчанні.

Предмет дослідження – інформаційна система для соціальної мережі університету.

Наукова новизна одержаних результатів. В результаті проведеної роботи були отримані такі результати:

Розглянуто особливості дистанційного навчання та онлайн-комунікації студента та викладача.

1. Досліджено особливості дистанційного навчання та онлайн-комунікацію вчитель-студент, що дало можливість сформулювати вимоги до засобів онлайн-комунікації в рамках соціальної мережі університету та визначити системи орієнтовані на таке навчання.

2. Розроблено інформаційну систему для соціальної мережі університету що дозволило забезпечити взаємодію між студентами та викладачем під час дистанційного навчання.

Практичне значення одержаних результатів. На основі розробленої інформаційної системи для соціальної мережі університету було створено веб-додаток з можливістю інтеграції в мережу університету для проведення дистанційної форми навчання. При розробці системи використовувалось API та SDK відео-конференцій Zoom.

Апробація результатів дипломної роботи магістра та публікації: Доповідь на тему «Інформаційна система онлайн-комунікації для дистанційного навчання» на XII Всеукраїнській науково-практичній конференції «Актуальні проблеми

комп'ютерних наук АПКН-2020» (Хмельницький, 9-10 листопада 2020 року, Хмельницький національний університет).

За темою дипломної роботи магістра автором виконана *наукова публікація* [27].

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 27 найменувань. Загальний обсяг дипломної роботи магістра становить 90 сторінки, з них 67 сторінок основного тексту та 22 сторінок додатків. У роботі наведено 34 рисунків.

Ключові слова: онлайн-комунікація, дистанційне навчання.

Зміст

Перелік скорочень	4
Вступ	5
Розділ 1	
Аналіз видів та технологій онлайн-комунікацій	8
1.1 Аналіз предметної області	8
1.2 Аналіз онлайн-комунікації вчитель-студент	10
1.3 Аналіз видів комунікації	12
1.4 Дослідження та порівняння технологій API/SDK	16
1.5 Постановка задачі	18
Висновки до розділу 1.	19
Розділ 2	
Розробка інформаційної системи онлайн-комунікацій для дистанційного навчання.	20
2.1 Загальний опис інформаційної системи онлайн-комунікацій	20
2.2 Сервіси відео конференцій	22
2.2.1 Google Meet	22
2.2.2 Zoom	25
2.2.3 Proficonf	29
2.3 Порівняння досліджуваних систем відео конференцій	31
2.4 Робота з SDK/API Zoom	32
Висновки до розділу 2	34
Розділ 3	
Проектування інформаційної системи для роботи з SDK/API Zoom	35
3.1 Проектування архітектури інформаційної системи	35
3.2 Проектування загальної схеми інформаційної системи	39
3.3 Проектування компонентів системи	40
Висновки до розділу 3	45
Розділ 4	

Програмна реалізація	46
4.1 Обґрунтування засобів реалізації	46
4.1.1 Мова програмування	46
4.1.2 Вибір фреймворка для верстки	48
4.2 Розробка модуля авторизації/реєстрації та особистого кабінету	51
4.3 Розробка модуля взаємодії із API Zoom	56
4.4 Розробка модуля для створення, планування заняття	57
4.5 Розробка модуля для створення списку відвідування заняття	59
Висновки до розділу 4	62
Загальні висновки	64
Перелік посилань	65
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
API	програмний інтерфейс програми
SDK	набір засобів розробки
MVC	Модель-Представлення-Контролер
БД	База даних

Вступ

Дипломна робота магістра присвячена розробці інформаційної системи соціальної мережі університету для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання.

Актуальність теми. В наш час невід'ємною частиною життя стали онлайн-комунікації в повсякденному житті. Спочатку вони з'явилися у виді електронних листів, чатів, форумів з розвитком інтернету з'явилися соціальні мережі різні месенджери, які згодом отримали можливість здійснювати аудіо/відео дзвінки. Зараз стало нормальним спілкування за допомогою комп'ютерної мережі інтернет, більшість спілкування і комунікацій перейшла в онлайн, значне місце займають соціальні мережі.

Зараз в зв'язку з пандемією на перший план виходять онлайн-комунікації зокрема відео конференції які використовуються наприклад для дистанційного навчання, що накладає на них певні вимоги щодо використання їх в сучасних умовах. Через що відбувається розвиток онлайн-комунікацій наприклад таких як Zoom, Google Meet та інші які надають змогу проводити дистанційне навчання, конференції тощо. Але на момент появи пандемії навчальні заклади не були готові до таких змін у навчанні, коли різні ресурси повністю не задовольняють потрібним нам вимогам, навчання стало менш продуктивним, викладачу важко відслідковувати присутність та активність студентів, комунікація стала більш складною так як немає однієї загальної системи для усього.

Запропонована інформаційна система для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання.

Кінцевою метою застосування розроблено інформаційної системи, є використання її при дистанційному навчанні, що забезпечить зручність та вирішення проблем з проведенням його.

Мета і задачі роботи. Метою магістерської роботи є дослідження та розробка інформаційної системи соціальної мережі університету для забезпечення взаємодії

між студентами та викладачем під час дистанційного навчання.

Для досягнення поставленої мети визначено наступні задачі:

- дослідити та проаналізувати види комунікацій та онлайн-комунікацій;
- провести аналіз існуючих сервісів відео-конференцій;
- спроектувати та розробити інформаційну систему на основі існуючих аналогів відео-конференцій.

Також інформаційна система повинна забезпечити визначені вимоги для дистанційного навчання.

Об'єкт дослідження – онлайн-комунікація в дистанційному навчанні.

Предмет дослідження – інформаційна система для соціальної мережі університету.

Наукова новизна одержаних результатів. В результаті проведеної роботи були отримані такі результати:

Розглянуто особливості дистанційного навчання та онлайн-комунікації студента та викладача.

3. Досліджено особливості дистанційного навчання та онлайн-комунікацію вчитель-студент, що дало можливість сформулювати вимоги до засобів онлайн-комунікації в рамках соціальної мережі університету та визначити системи орієнтовані на таке навчання.

4. Розроблено інформаційну систему для соціальної мережі університету що дозволило забезпечити взаємодію між студентами та викладачем під час дистанційного навчання.

Практичне значення одержаних результатів. На основі розробленої інформаційної системи для соціальної мережі університету було створено веб-додаток з можливістю інтеграції в мережу університету для проведення дистанційної форми навчання. При розробці системи використовувалось API та SDK відео-конференцій Zoom.

Апробація результатів дипломної роботи магістра та публікації: Доповідь на тему «Інформаційна система онлайн-комунікації для дистанційного навчання» на

XII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2020» (Хмельницький, 9-10 листопада 2020 року, Хмельницький національний університет).

За темою дипломної роботи магістра автором виконана *наукова публікація* [27].

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 27 найменувань. Загальний обсяг дипломної роботи магістра становить 90 сторінки, з них 67 сторінок основного тексту та 22 сторінок додатків. У роботі наведено 34 рисунків.

Ключові слова: онлайн-комунікація, дистанційне навчання.

Розділ 1

Аналіз видів та технологій онлайн-комунікацій

1.1 Аналіз предметної області

В наш час невід'ємною частиною життя людини є соціальні мережі, різні месенджери і тд, без яких ми вже не можемо уявити наше життя, інтернет-комунікації стали не від'ємним інструментом у нашому житті як для простого спілкування так і для роботи, розваг і тд.

Під Інтернет-комунікацією розуміється особлива форма комунікації, мережева взаємодія людей один з одним, а комп'ютерно-опосередкована комунікація описує взаємодію людей за допомогою комп'ютера, що об'єднані в мережу. Відповідно до каналу передачі і сприйняття інформації комп'ютерна комунікація поділяється на актуальну (спілкування з реальними людьми) та віртуальність (спілкування з уявними співрозмовниками) [1].

З моменту виникнення мережі Інтернет комунікація в мережевому співтоваристві зазнала змін. На початкових етапах суб'єктами комунікації в глобальних мережах були невідомий відправник і одержувачі інформації. Тепер тип комунікації можна визначити як "масова комунікація". На відміну від інтерактивної міжособистісної комунікації, масова комунікація може бути інтерактивною і односпрямованою. Масова комунікація в мережах (YouTube , блоги), на противагу традиційної, односпрямованої, з одного боку потенційно спрямована на спілкування з великою аудиторією, а з іншого - це само комунікація, так як суб'єкт усвідомлено вибирає потенційного одержувача, здійснює пошук конкретних повідомлень або контенту з глобальної мережі.

Класифікація ситуацій використання Інтернету (broad internet-using situations), класифікація жанрів, запропонована Девідом Крісталом [1]:

- електронна пошта;
- синхронні й асинхронні чати;

- віртуальні світи;
- відео конференції;
- ВЕБ-тексти, до яких належать електронні тексти з гіпер та лінійною структурою.

На даний момент в період пандемії, не від'ємним став перехід на дистанційне навчання або роботу, де люди зіштовхнулись з проблемами комунікацій між собою, особливо в навчанні стало важче працювати зі студентами, так як приходиться використовувати сторонні ресурси для взаємодії між студентами.

У зв'язку з актуальністю проблеми, виникла велика кількість завдань, які можна об'єднати в загальний клас «комунікації університету». У цей розділ входить величезне різноманіття різних завдань, які пов'язані тим, основні їх функції якф носять той чи інший комунікаційний сенс де використовується функціональна масштабність спілкування.

Функціональна масштабність, або соціально обумовлена масштабність спілкування. За цією ознакою розрізняють масову комунікацію (що має місце в суспільстві в цілому), обмежену (в рамках соціальних груп - компаній, організацій), локальну (у мікрогрупах типу асоціацій, професійних колективів), внутрішньо групову (у малих групах, в сім'ї), міжособистісну, або інтерперсональну (між двома чи кількома індивідами, які спілкуються одне з одним) і навіть внутрішньо особистісну, або інтраперсональну (між індивідом і електронними засобами передачі інформації) [2]. Тобто є можливість розпочати з груп студентів і розширитись до глобальної соціально комунікації всього університету що в наш час є дуже актуальним.

Так, наприклад, класична комунікація між людьми полягає в зручній онлайн-комунікації як просто повідомлення, передача файлів то що. А комунікація між всією групою або ж навіть цілим факультетом буде досить зручною наприклад для масових сповіщень як ціле направлених так і загальних.

Особливе місце в класі комунікації університету займає задача спілкування студентів з викладачем, яка полягає в зручності донесення до усіх потрібної інформації, для чого використовується онлайн чат.

Завдання онлайн-комунікації в зручній взаємодії студентів та викладачів між собою.

1.2 Аналіз онлайн-комунікації вчитель-студент

У випадку онлайн-комунікація вчитель-студент є дистанційним навчанням. Дистанційне навчання - новий освітній досвід, що з'явився в Україні зовсім недавно, але вже встиг отримати своє місце в педагогічному середовищі. Необхідність у такому методі навчання обумовлена різними факторами, серед яких можна назвати потребу в інтерактивній взаємодії учнів і викладачів в процесі навчання, надання учням можливості самостійної роботи з освоєння досліджуваного матеріалу. І, звичайно ж, цей спосіб навчання дає можливість навчатися тим, хто в силу певних причин не може це робити в масових установах.

При використанні дистанційного навчання можуть використовуватися різні методи донесення навчальної методичної інформації до учнів різних категорій (реферати та конспекти, інтерактивні лекції та семінари). Сьогодні вже змінилося декілька поколінь спеціальних технологій в галузі дистанційної освіти – починаючи від звичайних друкованих освітніх видань до сучасних комп'ютерних технологій: освітнє телебачення, аудіо та відеотрансляції, відеоконференції, E-Learning, різноманітні інтернет-трансляції, інтернет-конференції.

У нашому випадку онлайн-комунікація викладач-студент це є дистанційне навчання, через ситуацію з пандемією це стало як ніяк актуально в наш час до чого ми не були досить готові, переважна більшість додатків та рішень для навчання, не є зручні тому що з більшості програм на підходять тільки ті в яких можна створювати онлайн конференцію, таких як Zoom, Skype та Google Meet. Так як потрібна інтерактивність та контроль зі сторони викладача.

В ході проведеного дослідження хочеться виділити наступні переваги та недоліки дистанційного навчання [12]:

Переваги:

- можливість учнів до засвоєння максимального обсягу необхідних знань, які не можуть бути включені в звичні класно-урочні форми та форми підготовчих курсів;
- здатність до навчання осіб у віддалених важкодоступних регіонах;
- можливість батьківського контролю над знаннями учнів;
- здатність до централізації освітнього процесу різних регіонів;
- змішання кордонів між очною та заочною формами навчання.

Недоліки:

- недостатній контроль над засвоєнням учнями одержуваних знань;
- «теоретизація» навчання, зведення навчального процесу до механістичним прийомів і методів;
- неможливість формування повного уявлення про зміст навчального предмета;
- відсутність інтерактивності.

З цього дослідження видно, що в наш час не вистачає інтерактивності та недостаті контролю над учнями для одержування знань. Тому створено вимоги для реалізації онлайн-комунікацій викладач-студент:

- контроль студентів;
- інтерактивний зворотній зв'язок.

Контроль студентів, під цим розуміється відслідковування присутності учня, реальної взаємодії з вчителем, надання інформації для вчителя по присутності учня на місці за допомогою використання веб камери. Вести інформацію про проведення заняття, відвідуваність, тривалість, присутність також викладача і тд.

Інтерактивний зворотній зв'язок, можливість відслідковувати рухи або ж надати можливість зручно взаємодіяти з учнем, тобто відслідковувати підняту руку

чи іншу комунікацію так як при великій кількості студентів буде це майже неможливо.

Таким чином проведення дистанційного навчання передбачає виконання вимог, а саме: проведення онлайн занять, відслідковування присутності студента, активна взаємодія з ним, а також швидкий та зручний зворотній зв'язок з викладачем.

1.3 Аналіз видів комунікації

В наш час вже створено дуже багато різновидів комунікацій, за в яких основою є чат або аудіо, відео зв'язок або відео конференція.

Виділяють чотири основних види відео конференцій: один до одного, симетрична групова, селекторну нараду і веб-конференція [15].

Відео дзвінок - це найпростіший вид відео конференцій, що дозволяє двом учасникам бачити і чути один одного, а так же обмінюватися різною інформацією, використовуючи спеціальні засоби [15].

Групова відеоконференція - це відеоконференція, що проходить одночасно між трьома і більше учасниками. В даному типі конференцій кожен учасник може бачити і чути всіх учасників, а також його можуть бачити і чути всі учасники. Такі конференції найчастіше забезпечуються засобами відео магазину, що представляє собою міст, що забезпечує зв'язок між кількома відео джерелами, в якості яких можуть виступати термінали або шлюзи відео конференції. Сервера для відео конференцій можуть бути як програмними, так і програмно-апаратними [15].

Селекторна нарада - це відеоконференція, що проходить одночасно з великою аудиторією учасників. В даному типі конференцій присутні 3 ролі: Ведучий, Доповідач і Виступаючий із залу. При цьому будь-який учасник бачить і чує тільки Ведучого, Доповідача і виступає, при цьому сам може бути Доповідачем або Виступаючим [15].

Вебінар - це відеоконференція, використовувана для організації живих зустрічей або презентацій через мережу Інтернет. Кожен учасник такої відеоконференції сидить за своїм персональним комп'ютером, підключений до інших учасників конференції за технологією клієнт-сервер через мережу інтернет. Як клієнт використовується додаток, заздалегідь викачане на комп'ютер, або веб-додаток (наприклад Flash або Java) [15].

Для аналізу і проведення досліджень розглянемо такі програми як Discord, Telegram, Viber, Skype, Zoom, Google Meet.

Discord - пропрієтарний безкоштовний месенджер з підтримкою VoIP, відеоконференцій, призначений для використання різними спільнотами по інтересам, найбільш популярний у геймерів і учнів. Додаток здатний організувати голосові конференції з налаштуванням каналу зв'язку і працювати за принципом push-to-talk, створювати публічні і приватні чати для обміну текстовими повідомленнями. Програма має браузерну версію, відмінністю від настільної версії є те, що режим push-to-talk працює тільки якщо в браузері активна вкладка з додатком [3].

Telegram - багатоплатформовий месенджер, що дозволяє обмінюватися повідомленнями і медіафайлів багатьох форматів.

Клієнтські програми Telegram доступні для Android, iOS, Windows Phone, Windows, macOS і GNU / Linux. Користувачі можуть додавати і обмінюватися фотографіями, стікерами, голосовими повідомленнями, файлами будь-якого типу, а також робити аудіо- і відеодзвінки [4].

Viber - додаток-месенджер, яке дозволяє відправляти повідомлення, здійснювати відео- і голосові VoIP-дзвінки через інтернет. Здійснювати дзвінки між користувачами з встановленим Viber безкоштовні (оплачується тільки інтернет-трафік за тарифом оператора зв'язку). Viber має можливість відправляти текстові, голосові та відео повідомлення, документи, зображення, відеозаписи та файли, а також в автономному режимі [5].

Для авторизації користувачів і пошуку контактів додаток використовує номер телефону і передає вміст телефонної адресної книги (імена і телефони всіх контактів)

Skype - безкоштовне власницьке програмне забезпечення з закритим кодом, що забезпечує текстову, голосовий та відеозв'язок через Інтернет між комп'ютерами (IP-телефонія), опціонально використовуючи технології пірінгових мереж, а також платні послуги для дзвінків на мобільні і стаціонарні телефони [6].

Програма також дозволяє здійснювати конференц-дзвінки, відеодзвінки (до 50 абонентів, включаючи ініціатора), а також забезпечує передачу текстових повідомлень (чат) і передачу файлів. Є можливість разом із зображенням з веб-камери передавати зображення з екрану монітора, а також створювати і відправляти відеоповідомлення користувачам настільних версій програми. Програмні клієнти Skype випущені для macOS, iOS, Windows, Linux, Windows Phone, Open webOS, Android, PSP, Maemo, Xbox 360, PlayStation Vita, BlackBerry [6].

Zoom - програма для організації відеоконференцій, розроблена компанією Zoom Video Communications. Вона надає сервіс відеотелефонії, який дозволяє підключати одночасно до 100 пристроїв безкоштовно, з 40-хвилинним обмеженням для безкоштовних акаунтів. Користувачі мають можливість підвищити рівень обслуговування, використовуючи один з тарифних планів, з максимальним числом підключень до 500 чоловік одночасно, без обмежень за часом. Під час пандемії COVID-19 стався найбільший сплеск популярності Zoom для віддаленої роботи, дистанційного навчання і соціального спілкування з використанням інтернету [7].

Google Meet - це програмне забезпечення для миттєвого обміну повідомленнями і відеоконференцій, розроблене компанією Google. Вона замінює собою одразу три системи для миттєвого обміну повідомленнями: Google Talk, Google+ Чати і сервіс відеочатів Google+ Відеозустрічі, а також онлайн-трансляцію через YouTube [8].

Proficonf - це професійна платформа для відеоконференцій і вебінарів в реальному часі, що працює прямо в браузері. Головні переваги Proficonf - це

простота використання, стабільну якість відеосигналу при великій кількості учасників (до 250) і безпрецедентно плавна анімація. Зручний інтерфейс користувача забезпечує комфортне використання сервісу навіть малодосвідчених користувачам [19].

Виходячи з інформації про аналоги стає зрозумілою мета програм та сфера у якій вони здебільшого використовуються. Такі програми як Discord, TeamSpeak які здебільшого направленні на комунікацію для геймерів в різних іграх і тд.

А програми Instagram, Facebook Messenger, Viber, Telegram є менеджери які мають можливість робити відео та аудіо дзвінки та передавати різні види файлів , здебільшого застосовуються для повсякденного спілкування і також для різних інших ділових і тд. комунікацій з яких Instagram більш соціальна мережа ніж просто месенджер. Zoom заточений під відео та аудіо конференції, в яких дуже зручним є можливість відображати свій екран що допомагає при навчанні. Skype більш практичний для всіх видів комунікацій як ділових так і для навчання, спілкування з друзями і тд. В ньому є можливість створювати конференції аудіо та відео, групові чати, передавати майже усі види файлів. Усі ці програми об'єднує одне комунікація людей, здебільшого через чат, принцип роботи якого зображено на рис 1.1. Також вони мають можливість проводити аудіо та відео дзвінки або конференціях що в період карантину стає дуже зручним для комунікації в навчанні , проведенні лекцій і тд.

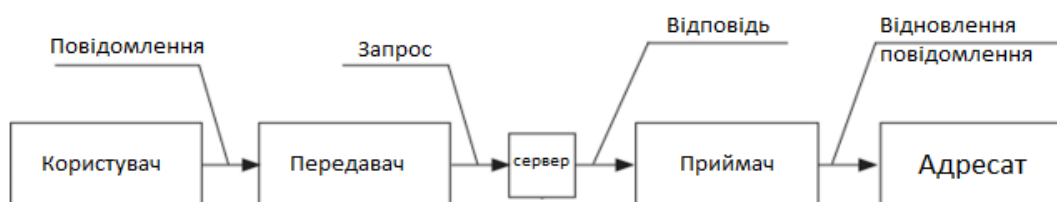


Рисунок 1.1 Принцип роботи месенджерів

1.4 Дослідження та порівняння технологій API/SDK

Терміни API і SDK часто використовуються разом, але вони не є синонімами. SDK реалізують незалежний від мови REST API певною мовою, такому як Java або C ++. API REST самі по собі не прив'язані до якогось конкретного мови; зазвичай ми демонструємо API, здійснюючи запити, використовуючи cURL, інструмент командного рядка для відправки веб-запитів і отримання відповідей. Але розробники не будуть використовувати запити cURL при реалізації API. Замість цього вони будуть реалізовувати запити API, використовуючи мову, на якому написано їх додаток [9].

SDK - набір засобів розробки, який дозволяє фахівцям з програмного забезпечення створювати додатки для певного пакету програм, програмного забезпечення базових засобів розробки, апаратної платформи, комп'ютерної системи, ігрових консолей, операційних систем і інших платформ (рис 1.2) [10].



Рисунок 1.2 – SDK та приклади додатків

Інструменти SDK включатимуть цілий ряд речей, включаючи бібліотеки, документацію, зразки коду, процеси та керівництва, які розробники можуть використовувати та інтегрувати у свої власні програми. SDK призначені для використання для певних платформ або мов програмування [14].

API - це інтерфейс, що дозволяє двом незалежним компонентів програмного забезпечення обмінюватися інформацією. API грає роль посередника між внутрішніми і зовнішніми програмними функціями, забезпечуючи настільки ефективний обмін інформацією, що кінцеві користувачі зазвичай його просто не помічають. [11].

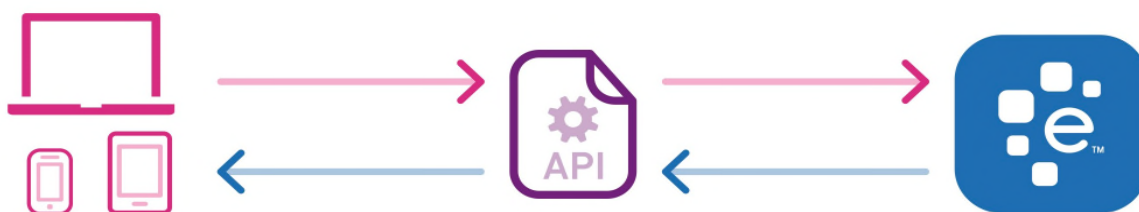


Рисунок 1.3 – Робота API

API діє як віртуальний посередник і передає інформацію з одного інтерфейсу, наприклад мобільного додатка, в інший. API пов'язує різні частини програмної платформи, щоб передана інформація дійшла до місця призначення [11].

Ці сполучні вузли не тільки виконують роль внутрішніх каналів зв'язку, а й дозволяють зовнішнім інструментам отримувати доступ до цієї ж інформації. Таким чином API-інтерфейси можуть належати до однієї з двох категорій [11]:

- внутрішні / приватні API;
- зовнішні / відкриті API.

Приватні API доступні тільки розробникам і користувачам з числа співробітників організації. Такі API зазвичай пов'язують внутрішні процеси для зменшення розрізненості робочих даних і оптимізації спільної роботи [11].

Відкриті API, в свою чергу, дозволяють зовнішнім розробникам отримувати доступ до інформації та інтегрувати інформацію, яка передається з одного програмного інструменту в інший. Відкриті або приватні API економлять час розробників, дозволяючи їм об'єднувати платформи з наявними інструментами і усуваючи необхідність в створенні нового функціоналу з нуля.

Існує два основних типи API, міжсерверний або клієнт-до-сервера та компонент-до-компонента. API між серверами та клієнтом до сервера зазвичай є інтерфейсом без доступу до віддаленого сервера для встановлення або отримання даних. Як правило, ці типи API використовують протокол RESTful. Протоколи вимагають, щоб будь-який виклик API був повністю незалежним від будь-яких минулих або майбутніх викликів API. API визначає протокол зв'язку та очікувану поведінку, що дозволяє клієнту робити запити та отримувати відповіді на сервер та з нього [13].

REST - підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів. В основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP.

1.5 Постановка задачі

Метою магістерської роботи є дослідження та розробка інформаційної системи соціальної мережі університету для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання. Для цього необхідно дослідити та адаптувати існуючу систему онлайн-комунікації. На основі даної інформаційної системи необхідно створити або удосконалити додаток для онлайн-комунікації.

Отже, можемо зробити висновок, що для досягнення поставленої мети необхідно :

- проаналізувати особливості онлайн-комунікації вчитель-студент та сформулювати вимоги до дистанційного навчання;
- дослідити технології існуючих системи онлайн-комунікації та визначити відповідність вимогам дистанційного навчання в рамках інформаційної системи;

- розробити інформаційну систему соціальної мережі університету на основі існуючої платформи онлайн-конференцій з метою покращення взаємодії між студентами та викладачем під час дистанційного навчання.

Висновки до розділу 1.

Розглянуто особливості дистанційного навчання та онлайн-комунікації студента та викладача. Показано, що на даний момент немає однієї системи для навчання та комунікації студентів з вчителем, в якій би можна було зручно відслідковувати присутність студентів, вести онлайн конференції, та мати швидку комунікацію між студентом та викладачем находячись в одній системі.

Розглянуто види комунікацій. Можна зробити висновок, що онлайн-комунікації розвиваються досить швидко і їх численність дуже велика і збільшується з кожним роком, які заміняють нам звичний спосіб комунікації, майже усе перейшло в онлайн, спираючись на це виникає потреба в системі орієнованій на навчання та комунікацію студентів з викладачем.

Дослідженні та порівнянні технології API/SDK , говорять про те що за основу розробки інформаційної системи можна взяти уже існуючу та вдосконалити її використавши можливості API/SDK.

Таким чином вирішено спроектувати та розробити інформаційну систему на основі уже існуючої з потрібною функціональністю.

Розділ 2

Розробка інформаційної системи онлайн-комунікацій для дистанційного навчання.

2.1 Загальний опис інформаційної системи онлайн-комунікацій

Недоліком розглянутих інформаційних систем онлайн-комунікацій є обмеження функціоналу, так як, вони мають не один напрямок застосування, тому і виникає нестача функціональності яка відсутня повністю або частино, котра потрібна для дистанційного навчання. Так як проведення дистанційного навчання передбачає виконання вимог, а саме: проведення онлайн занять, відслідковування присутності студента, активна взаємодія з ним, а також швидкий та зручний зворотній зв'язок з викладачем. Виходячи із зазначених вище недоліків, розроблено інформаційну систему, яка надає можливість виконувати основні вимоги дистанційного навчання. Запропонована інформаційна система дозволить реалізувати в собі виконання вимог для дистанційного навчання.

Інформаційна система складається з наступних кроків (рис 2.1)

1. Вхідна інформація: дані учасників (студентів, викладачів) та вимоги до дис. навчання.
2. Система онлайн-комунікації .
3. Збір та обробка мета даних системи онлайн-комунікацій.
4. Вихідна інформація: звіт онлайн-комунікацій.

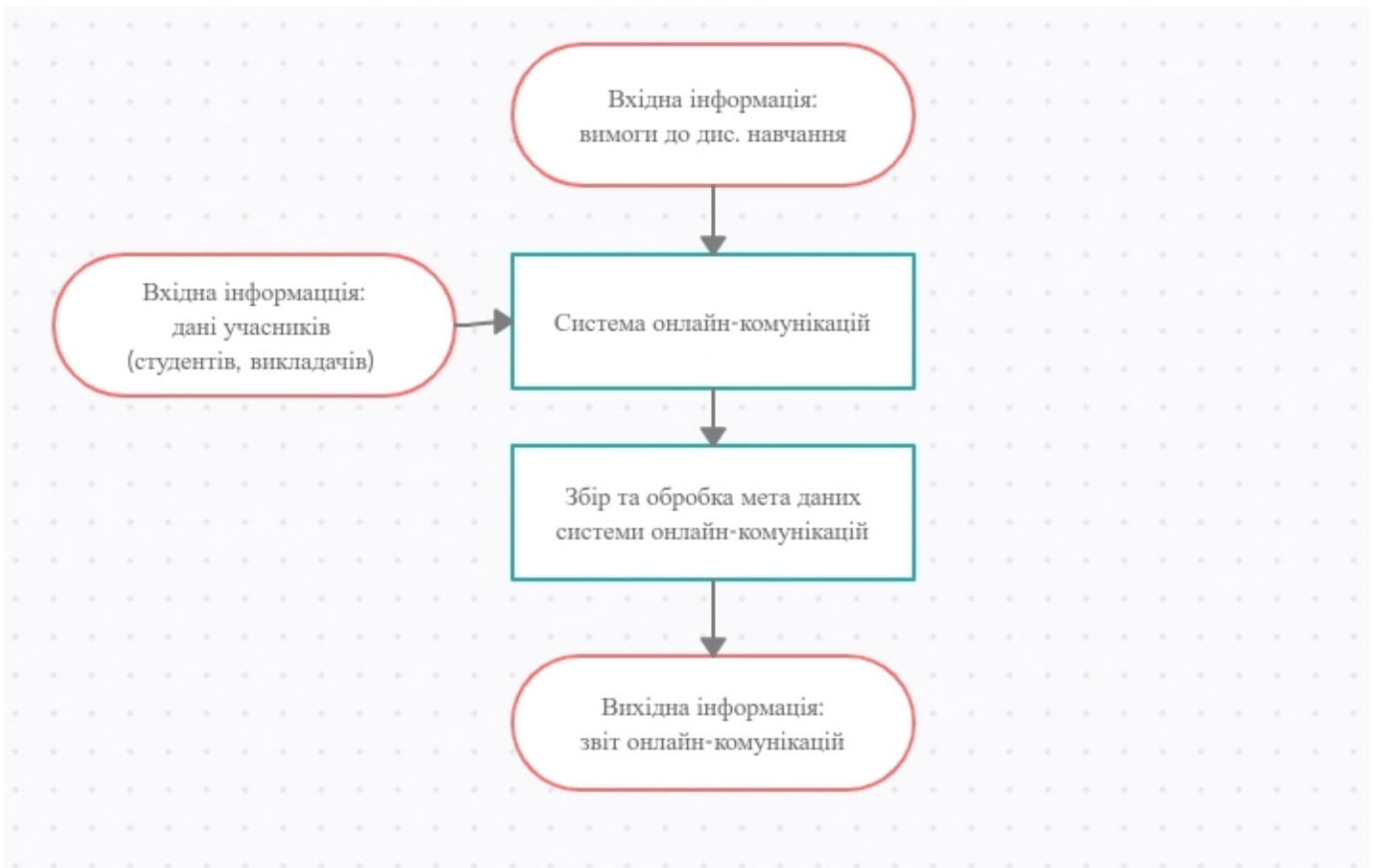


Рисунок 2.1 Схема інформаційної системи онлайн-комунікацій

Вхідною інформацією є два об'єкти, перший з яких це дані про учасників тобто студентів та викладачів, такі дані як ім'я, прізвище, група та інші. Ця інформація потрібна для того щоб система онлайн-комунікацій виконувала свої задачі, також другим об'єктом є вимоги до дистанційного навчання які реалізовані в системі.

Система онлайн-комунікацій має в собі виконання усіх вхідних вимог, та має можливість створювати конференції, вести списки проведення конференцій, планувати їх за допомогою чого проходить дистанційне навчання.

Збір та обробка мета даних надає можливість обробити усі дані отримані після конференції для подальшої роботи з ними з чого отримуються звіти про конференцію

2.2 Сервіси відео конференцій

В наш час де панує пандемія невід’ємним стали онлайн відео конференції за допомогою яких проводиться навчання, з розглянутих вище систем комунікації визначено, що системи Google Meet та Zoom надають відкритий доступ до прикладного програмного інтерфейсу – API/SDK. Це дає можливість по перше інтегрувати відео конференцію на будь-який сайт, по друге працювати з різними даними конференції, такими як, статистикою відвідувань, робота з чатами, записи у хмару, зустрічі та інші, що відповідає нашим вимогам.

Ще однією платформою є Proficonf який надає можливість як і попередні сервіси створювати онлайн-конференції. На від мінусу від інших має велику перевагу при поганому з’єднанні що дає можливість примати участь у конференції при стабільному з’єднанні в 256 кб/с. Це дає йому велику перевагу над іншими сервісами онлайн-конференцій.

Розглянемо найбільш відомі платформи відео конференцій більше детально.

2.2.1 Google Meet

Google Meet - це програмне забезпечення для миттєвого обміну повідомленнями і відеоконференцій, розроблене компанією Google.

Можливості:

- Відео зустрічі з кількістю учасників до 250;
- Прямі трансляції на 100 000 користувачів;
- Запрошення до конференції за допомогою Meet, Gmail, по телефону;
- Демонстрація екрану;
- Запис і автоматичне збереження відеозустрічей на Google Диску і в подію Google Календаря;

– Обмін файлами та документами.

Відео зустрічі – це зв’язок двох або більше одночасних дзвінків до однієї і тієї ж наради (рис 2.2). Показники звітування враховують лише користувачів які були присутні на відео зустрічі.

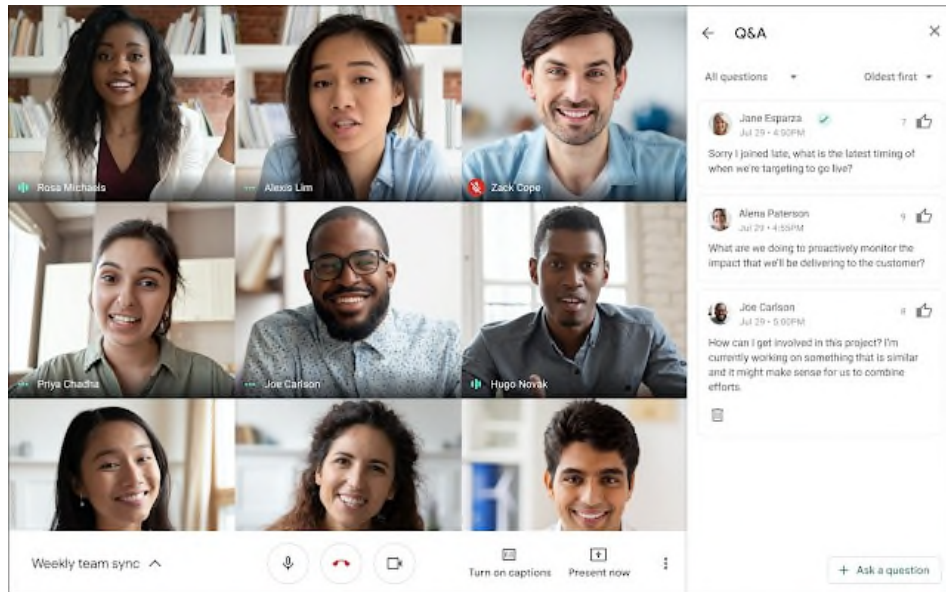


Рисунок 2.2 Вигляд відео зустрічі Google Meet

API розраховує тривалість зустрічі як суму всіх інтервалів із накладанням двох або більше дзвінків (рис 2.4.1). На діаграмі нижче показано, як розраховується тривалість зустрічі.

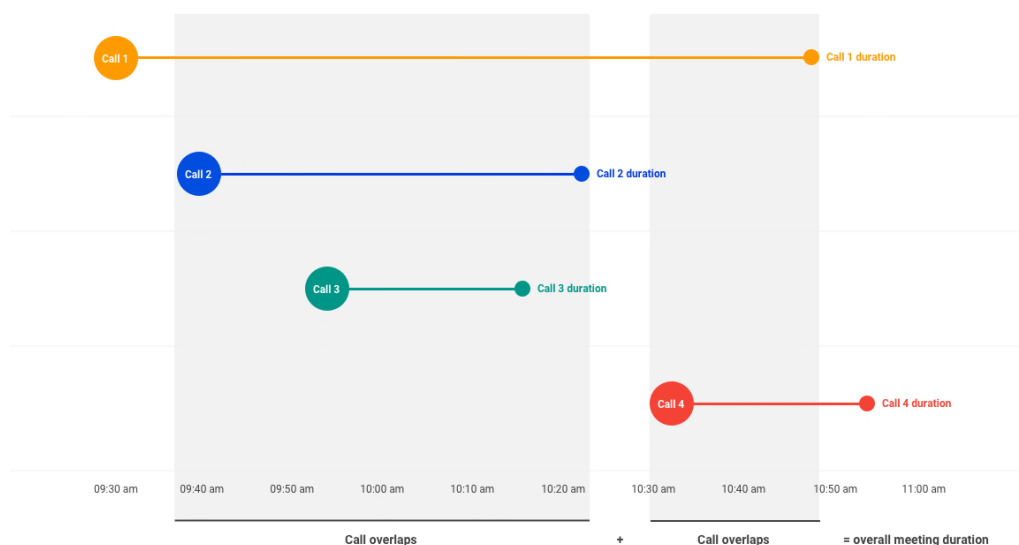


Рисунок 2.3 Графік тривалості зустрічі [7]

Також сервіс Google Meet має таку можливість як зберігання запису відео конференції та зберігання його на такій платформі як Google Диск, де є можливість скачати або переглянути її. Google Диск - сховище даних, яке дозволяє користувачам зберігати свої дані на серверах у хмарі та ділитися ними з іншими користувачами в Інтернеті.

Підключення клієнта до зустрічі Meet являється дзвінком. Клієнтом може бути мобільний додаток, веб-додаток або пристрій Chrome. Кожне з'єднання вважається окремим дзвінком. Наприклад, один користувач, який приєднується як з телефону, так і з робочого столу, зараховується як два окремі дзвінки. Або перерване з'єднання, яке повторно ініціює той самий користувач, зараховується як два окремі дзвінки. Тривалість дзвінка відноситься до загальної тривалості даного з'єднання, навіть якщо до зустрічі підключений лише один клієнт. Звіти Meet також розглядають кожен тип клієнта незалежно, щоб показати використання на різних пристроях. У наступній таблиці перелічені підтримувані типи клієнтів.

Підключення з будь-яких пристроїв. Запрошені учасники можуть приєднуватися до онлайн-відеоконференцій зі своїх комп'ютерів за допомогою будь-якого сучасного браузера - без необхідності встановлювати додаткове програмне

забезпечення. А на мобільних пристроях можна використовувати додаток Google Meet.

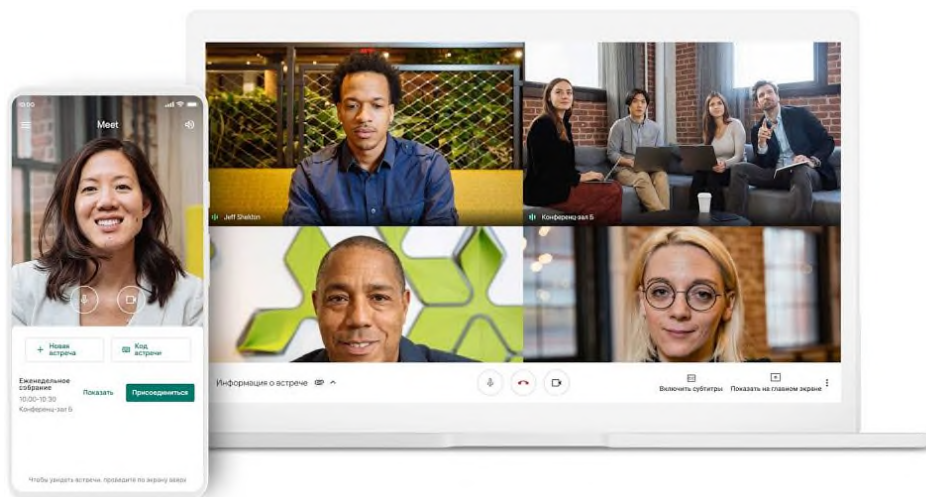


Рисунок 2.4 Кросплатформа Google meet

2.2.2 Zoom

Zoom – це сервіс для проведення відеоконференцій та онлайн-зустрічей. Організувати зустріч може будь-який користувач, що створив обліковий запис. Безкоштовна версія програми дозволяє проводити відеоконференцію тривалістю 40 хвилин, однак на період пандемії сервіс прибрав це обмеження.



Рисунок 2.5 Вигляд відео зустрічі Zoom

Переваги Zoom при використанні у дистанційному навчанні:

- Відмінний зв'язок. Платформа працює стабільно. Попри велике навантаження, яке зараз спричинене великою кількістю користувачів та шаленою їх активністю, Zoom чудово справляється з обслуговуванням усіх відеоконференцій.
- Відео та аудіо зв'язок з кожним учасником. Організатор конференції має можливість вимикати і вмикати мікрофон, а також вимикати відео та запитувати ввімкнення відео в усіх учасників. Можна увійти в конференцію як учасник з правами тільки для перегляду.
- Демонстрація екрану зі звуком. Демонстрацію екрану можна поставити на паузу. Крім того, можна ділитися не всім екраном, а лише окремими додатками, наприклад, вмикати демонстрацію браузера. В налаштуваннях можна дати всім учасникам можливість ділитися екранами, або ж ввімкнути обмеження, щоб робити це міг лише організатор

- Інтерактивна дошка. В платформу вбудована інтерактивна дошка яку можна демонструвати всім учасникам відеоконференції. така опція буде зручною та незамінною для навчання та й під час обговорення робочих моментів. Крім того є можливість легко і швидко перемикається з демонстрації екрану на інтерактивну дошку.
- Запис відеоконференції. Можна проводити запис відеосеансу як на комп'ютер, так і на хмарне сховище. Зручно, що можна налаштувати автоввімкнення запису, а також ставити його на паузу.
- Адміністрування. Під час конференції можна призначити співорганізатора, у якого будуть такі ж можливості як і у організатора відеоконференції: вмикати і вимикати мікрофон у окремих користувачів, перейменовувати відеочати та ділити їх на кімнати.

Zoom Client SDK дозволяють новим та існуючим програмам інтегрувати багатий набір функцій , доступних у програмі Zoom Client. Пропоновані на всіх основних власних платформах, Zoom Client SDK призначені для:

- Простий у використанні: прості імпортовані бібліотеки та пакети дозволяють швидко впровадити платформу зустрічі Zoom у нові та існуючі програми.
- Локалізація: Природна підтримка семи основних мов та відкрита розширюваність перекладу дозволяє збільшити використання програм на міжнародному рівні.
- Індивідуальні: Спеціальні функції інтерфейсу дозволяють розробникам розширювати та пристосовувати інтерфейс наради для прямої інтеграції з вашим додатком.

Веб-модулі, побудовані для забезпечення можливості проведення зустрічей та вебінарів у веб-браузері.

Також сервіс надає функціонал “Білої дошки” він надає можливості при онлайн-конференції в реальному часі показувати схематично інформацію за допомогою панелі для малювання.

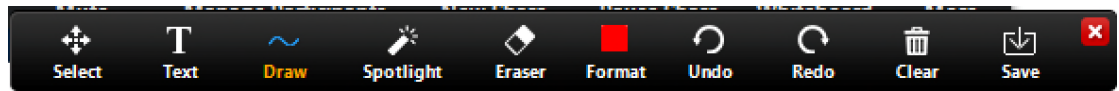


Рисунок 2.6 Панель для малювання Zoom

Zoom Web SDK . Веб-SDK дозволяє розробляти відеопрограми, що працюють на основі основного фреймворку Zoom, всередині веб-клієнта HTML5 за допомогою високооптимізованого модуля WebAssembly. Як розширення клієнта браузера Zoom, цей SDK призначений для реалізацій, коли кінцевий користувач має середовище з низькою пропускнуою здатністю, знаходиться поза мережевим брандмауером або має обмеження на свою машину, що заважає їм встановлювати Zoom Desktop або Mobile Clients .Підтримка браузера Web SDK безпосередньо відображає підтримку Zoom Web Client.

Сервіс забезпечує кросплатформеність і однаково працює як на ПК, так і на мобільному телефоні. Користувачу лише потрібно завантажити програму на комп'ютер чи додаток на телефон.



Рисунок 2.7 Кросплатформеність Zoom

2.2.3 Proficonf

Proficonf - це платформа для віддаленого спілкування між професіоналами та їх клієнтами. Являє собою сервіс, який дозволяє збирати до 250 користувачів в одній віртуальній переговорній кімнаті, причому кожен з відвідувачів має можливість говорити - на відміну від більшості вебінарних майданчиків, у яких кількість спікерів обмежена [16].

Proficonf відрізняється адаптивним зв'язком, який забезпечує стабільну трансляцію відео та звуку навіть при швидкості інтернет-з'єднання в 256 Кбіт / сек. Сервіс адаптується під швидкість вашого з'єднання і підбирає оптимальну якість трансляції. Так, при поганому з'єднанні в першу чергу буде знижуватися якість картинки, для підтримки хорошої якості звуку.

Сервіс працює в браузері і не вимагає установки сторонніх додатків на ваш ПК або смартфон.

Безкоштовний тариф Proficonf володіє привабливими умовами для клієнта на ринку відеоконференцій:

- 500 МБ файл-хостингу;
- до 25 учасників;
- немає обмежень за часом проведення заходів;
- спільний доступ до файлів і центру демонстрації.



Рисунок 2.8 Інтерфейс Proficonf

Основними перевагами використання цього сервісу є:

- необмежений час проведення заходів;
- до 250 осіб у кожній конференції;
- управління учасниками - призначення ролей модератора, спікера, глядача;
- автоматичний запис конференції в момент старту заходу;
- центр демонстрації будь-яких файлів, у тому числі відео з YouTube;

– хмарне простір для зберігання файлів від 500 Мб в безкоштовному тарифі до 5 Гб в платних. Дозволяє завантажувати і демонструвати файли і зберігати відеозаписи проведених заходів.

Також сервіс має корисну функцію віртуальну дошку, яка нагадує програму Paint та дає можливість відобразити схеми і зробити інші помітки (рис. 2.8).

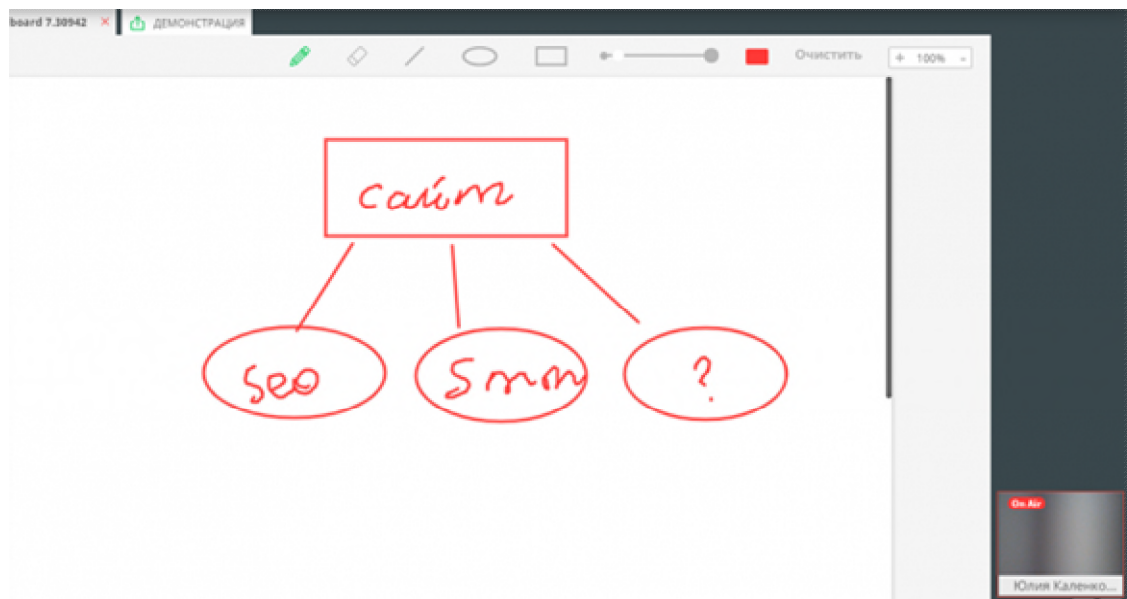


Рисунок 2.9 Віртуальна дошка Proficonf

2.3 Порівняння досліджуваних систем відео конференцій

В результаті проведеного дослідження вдалося з’ясувати, що серед різних сервісів відео конференцій, користувачам надаються схожі основні функції, не дивлячись на індивідуальність кожної, кожен з них надає одну і ту ж можливість відео конференцій, що є базовим для кожної з програм.

Далі наводяться загальні функції та властивості систем, розглянутих в дослідженні:

- кросплатформеність для проведення відео конференцій;
- стартовий безкоштовний тариф;

- SDK для мобільних платформ Android, iOS, тощо;
- вбудований чат;
- запис конференції;
- демонстрація матеріалів;
- планування конференцій наперед;
- робота з даними про проведену конференцію.

Виходячи з проведеного дослідження сервісів онлайн-конференцій, найбільш відповідною для дистанційного навчання є система Zoom, в якому є увесь необхідний функціонал, який буде використаний в створенні інформаційної системи для соціальної мережі університету.

2.4 Робота з SDK/API Zoom

API Zoom є основним засобом для розробників, щоб отримати доступ до ресурсів Zoom. API надає можливість працювати з такими найпопулярнішими функціями як створення нової зустрічі, додавання та видалення користувачів, перегляд звітів та інформаційних панелей тощо.

Усі API під Zoom API базуються на архітектурі REST та отримують доступ через HTTP за вказаними URL-адресами. Базова URL-адреса для всіх запитів - <https://api.zoom.us/v2/> . Повна URL-адреса змінюється залежно від кінцевої точки ресурсу, до якого здійснюється доступ [20].

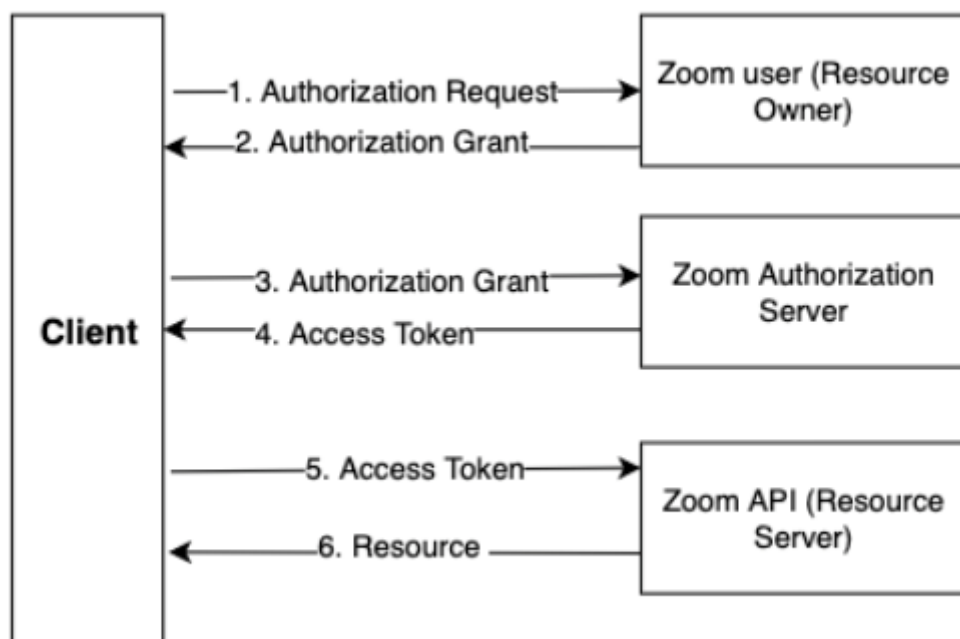


Рисунок 2.10 Робота з SDK/API Zoom

Кожен HTTP-запит, зроблений до Zoom API, повинен бути автентифікований Zoom. Zoom підтримує наступні два основних способи автентифікації запиту:

- OAuth 2.0;
- JWT.

Веб-маркер JSON (JWT) пропонує метод генерації токенів, які забезпечують безпечну передачу даних за допомогою акуратного та компактного об'єкта JSON. JWT містять підписане корисне навантаження, яке допомагає встановити автентифікацію між серверами.

OAuth 2.0 дозволяє програмам отримувати доступ до ресурсів Zoom (таких як інформація про профіль користувача), які надаються через Zoom API [20].

Протокол OAuth визначає три конкретні ролі, і ці ролі беруть активну участь у процесі автентифікації за допомогою Zoom API:

Власник ресурсу: власник ресурсу - це користувач облікового запису Zoom, який може авторизувати або заборонити клієнту доступ до інформації, пов'язаної з обліковим записом користувача Zoom [20].

Сервер ресурсів : Сервер ресурсів - це сервер, на якому розміщується ресурс. Якщо програма інтегрується із Zoom API для отримання інформації, пов'язаної з користувачем, сервер Zoom API вважається сервером ресурсів [20].

Клієнт : Клієнт - це програма, яка вимагає доступу до інформації користувача. Якщо програма подає запити на доступ до Zoom API, вона вважається клієнтом [20].

Висновки до розділу 2

В даному розділі розглянуто основні сервіси відео конференцій, їх можливості та основні характеристики. Оскільки вони реалізують більшу частину функціоналу для створення та керування конференцією за допомогою SDK та API.

Проведено порівняння систем відео конференцій та визначені загальні функції їх властивості. Та обрано систему на основі якої створено інформаційну систему де обрана система буде надавати можливість проводити онлайн конференції.

Розглянута робота API та SDK, яка дає можливість користувачеві взаємодіяти з програмною системою Zoom та використовувати її функціонал.

Розділ 3

Проектування інформаційної системи для роботи з SDK/API Zoom

3.1 Проектування архітектури інформаційної системи

Проектування – це процес визначення компонентів, архітектури, інтерфейсів та інших характеристик системи. Результатом проектування є сам проект – цілісна сукупність моделей, властивостей або характеристик, описаних у формі, придатній для реалізації системи.

Проектування системи направлено на представлення системи, яка відповідає поставленій меті та вимогам. Проектування включає в себе оцінку і прийняття рішень для вибору таких компонентів системи, які відповідають її архітектурі.

На даний момент існує тенденція розглядати архітектурне і детальне проектування як різні види діяльності. Архітектурні рішення в порівнянні із «звичайними» проектними рішеннями розглядаються як більш абстрактні, концептуальні та глобальні. Вони направлені на найбільш високо рівневі структури системи.

Детальне проектування, в свою чергу, визначається як процес деталізації та розширення початкового проекту до такої міри, при якій проект повністю готовий до реалізації.

В проектуванні виділяються функціональні, оптимальні та системні види проектування.

Функціональне проектування направлене, перш за все, на створення ефективного робочого об'єкта. Виконання необхідних функцій – головна мета і основа розробки об'єкта. До уваги приймаються функціональні показники якості і показники надійності.

Проектування, метою якого є не тільки пошук функціонально ефективних рішень, суперечливих потреб людини, обґрунтований вибір остаточного варіанта називають оптимальним проектуванням. Таке проектування ефективно стало

використовуватися на початку 20 століття завдяки великим досягненням теорії прийняття рішень і теорії дослідження операцій, а також широкому використанню та поширенню обчислювальної техніки, що дозволило розробити відповідні методи в доступні для огляду терміни, прорахувати чисельні варіанти і вирішувати складні математичні завдання.

Велике значення в оптимальному проектуванні відводиться підготовці на етапі технічного завдання, повного переліку вимог, щодо розроблюваного об'єкта, виділення серед них показників якості і перетворенню найбільш важливих із них в критерії оптимізації.

До типових вимог науково-технічної продукції відносять вимоги функціональності, надійності, технологічності, стандартизації та уніфікації, естетичність, економічність, тощо.

Системне проектування – основний принцип базується на системному підході.

Системне проектування комплексно вирішує поставлені завдання, бере до уваги взаємодію та взаємозв'язок окремих об'єктів системи і їх частин як між собою, так і в зовнішньому середовищі, враховує соціально-економічні та екологічні наслідки їх функціонування. Системне проектування ґрунтується на ретельному спільному розгляді об'єкта проектування і процесу проектування.

Основна ідея будь-якого з патернів MVP, MVC, MVVM полягає в поділі логіки і UI-частини програми так, щоб їх можна тестувати окремо. При цьому самі патерни досить сильно відрізняються між собою.

Для проектування інформаційної системи вибрано архітектурний шаблон MVC.

MVC (Модель-Представлення-Контролер) - схема поділу даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, уявлення і контролер - таким чином, що модифікація кожного компонента може здійснюватися незалежно [17].

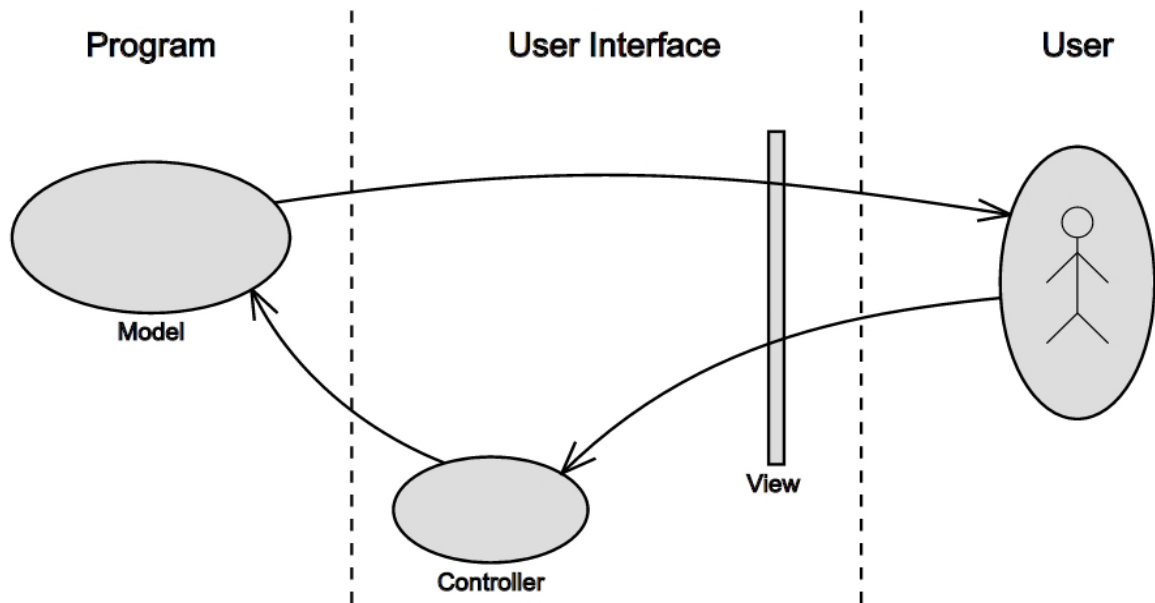


Рисунок 3.1 Схема роботи MVC

Концепція паттерна MVC передбачає поділ додатка на три компоненти [18]:

Модель (model): описує використовувані в додатку дані, а також логіку, яка пов'язана безпосередньо з даними, наприклад, логіку валідації даних. Як правило, об'єкти моделей зберігаються в базі даних.

У MVC моделі представлені двома основними типами: моделі уявлень, які використовуються уявленнями для відображення і передачі даних, і моделі домену, які описують логіку управління даними.

Модель може містити дані, зберігати логіку управління цими даними. У той же час модель не повинна містити логіку взаємодії з користувачем і не має визначати механізм обробки запиту. Крім того, модель не повинна містити логіку відображення даних в поданні .

Подання (view): відповідають за візуальну частину або призначений для користувача інтерфейс, нерідко html-сторінка, через який користувач взаємодіє з додатком. Також уявлення може містити логіку, пов'язану з відображенням даних. У той же час уявлення не повинно містити логіку обробки запиту користувача або управління даними.

Контролер (controller): представляє центральний компонент MVC, який забезпечує зв'язок між користувачем та програмою, поданням і сховищем даних. Він містить логіку обробки запиту користувача. Контролер отримує вводяться користувачем дані і обробляє їх. І в залежності від результатів обробки відправляє користувачеві певний висновок, наприклад, у вигляді подання, наповненого даними моделей.

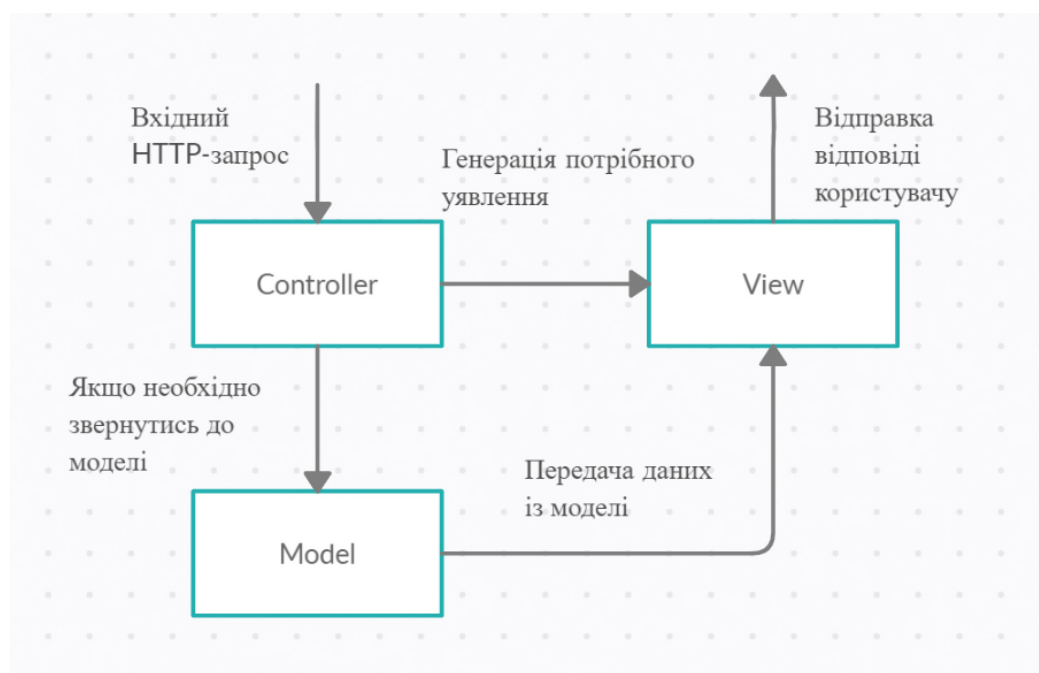


Рисунок 3.2 Взаємодія компонентів MVC між собою

Такий поділ компонентів інформаційної системи дозволяє реалізувати концепцію розділення відповідальності, при якій кожен компонент відповідає за свою строго окреслену сферу. У зв'язку з чим легше побудувати роботу над окремими компонентами. І завдяки цьому систему легше розробляти, підтримувати і тестувати окремі компоненти.

3.2 Проектування загальної схеми інформаційної системи

Для вирішення поставлених задач, розроблено інформаційну систему для онлайн комунікацій із взаємодією з SDK/API Zoom.

Запропонована система дозволяє проводити онлайн-конференції, працювати з API Zoom за допомогою якої ми отримуємо мета дані які обробляємо та формуємо список відвідування пар.

Для того щоб створити онлайн-конференцію потрібно бути зареєстрованим в інформаційній системі, в якій після авторизації є можливість створити або ж приєднатись до онлайн конференції.

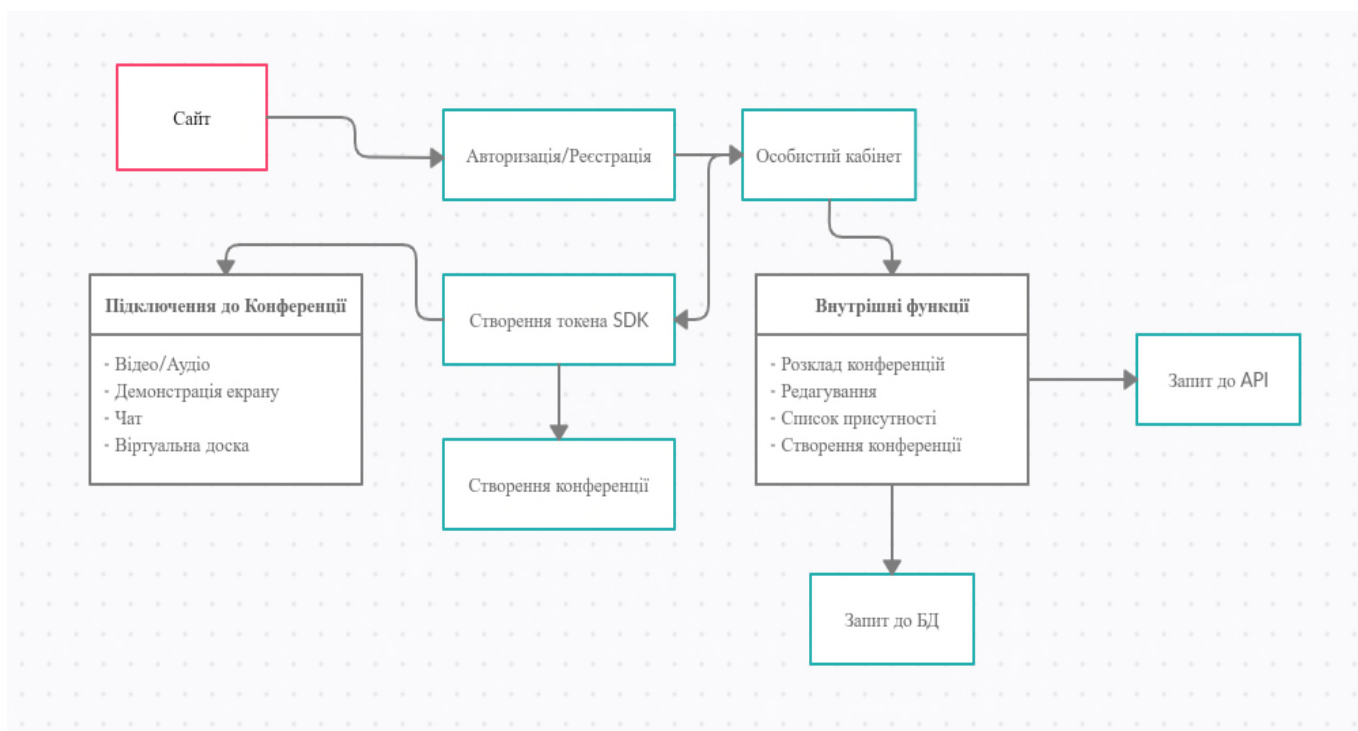


Рисунок 3.3 Взаємодія компонентів MVC між собою

Використання API надає змогу використовувати функції для взаємодії з сервером Zoom , надає можливість створювати конференції та отримувати мета дані про конференцію.

Кінцевим результатом являється інтеграція та робота з SDK, створення інформаційної системи як оболонки для онлайн-конференцій та додаткового функціоналу.

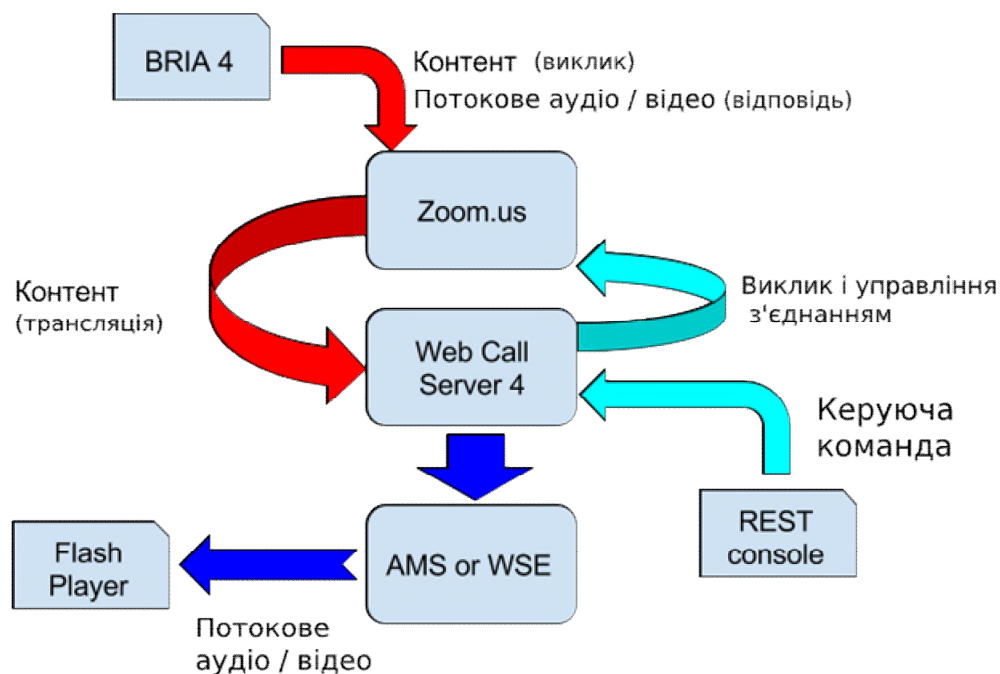


Рисунок 3.4 Схеми роботи системи онлайн-комунікації Zoom

При підключенні до онлайн-конференції по мережі протоколу BRIA 4 передаються дані аудіо/відео які потрапляють у конференцію та відтворюються для усіх присутніх в ній користувачів, використовуючи WCS4 (Web Call Server 4) аудіо та відео надає можливість відтворювати відео в реальному часі і для того щоб його відтворити, дані з WCS4 передаються потоком у Flash Player та відображаються у всі користувачів.

3.3 Проектування компонентів системи

Інформаційна система складається з наступних модулів:

- модуль авторизації/реєстрації та особистий кабінет;
- модуль взаємодії з API Zoom;

- модуль для створення, планування заняття;
- модуль для створення списку відвідування заняття.

Для авторизації в інформаційній системі використовується модуль авторизації (рис 3.5) суть якого полягає в тому, що система відправляє запит до бази даних з метою перевірки наявності облікового запису та співпадіння введених даних користувача тобто логіну та паролю.

В разі наявності логіна, відправляється запит на авторизацію в якому відбувається перевірка на збіг логіну з його паролем, де після успішної авторизації користувач потрапляє в особистий кабінет.

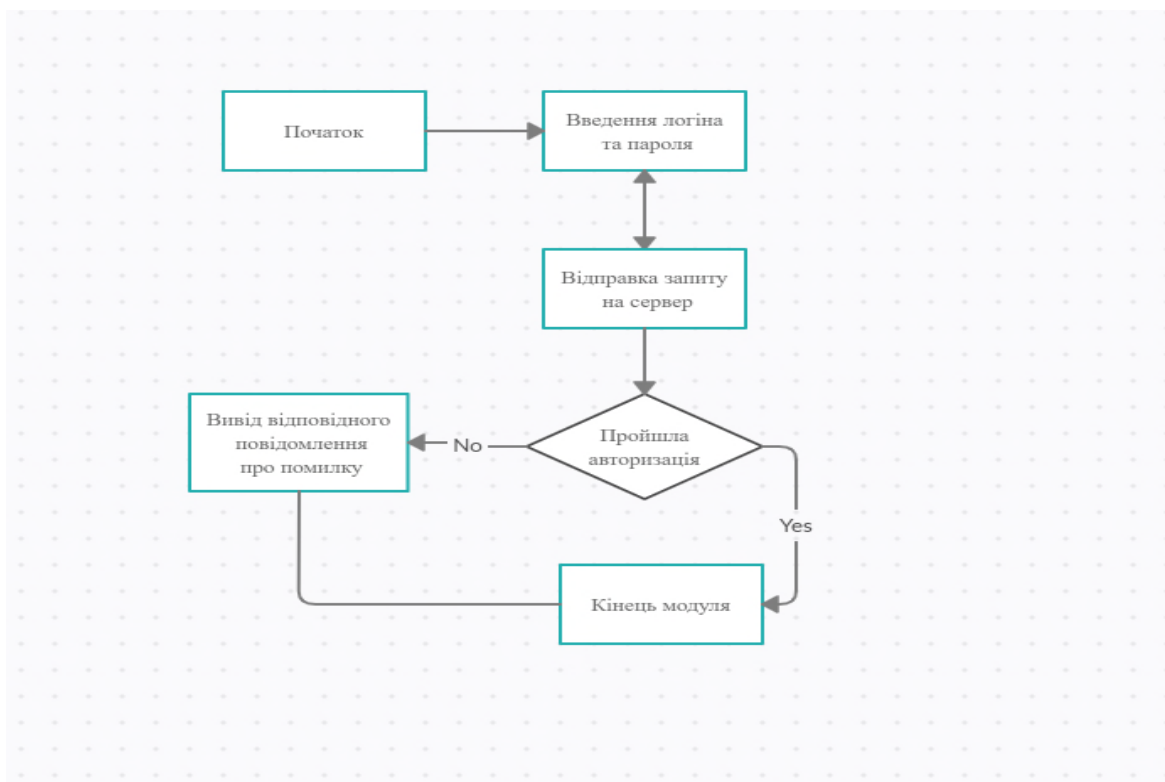


Рисунок 3.5 – Схема роботи модуля авторизації

Також якщо користувач не має свого облікового запису він має можливість зареєструватись (рис 3.6). Після вводу даних відправляється запит на сервер для перевірки уже існуючого облікового запису з таким самим email/логіном, де після успішного проходження перевірки відбувається реєстрація, тобто створення нового

облікового запису, після чого користувач отримає повідомлення про успішну реєстрацію або ж про помилку реєстрації.

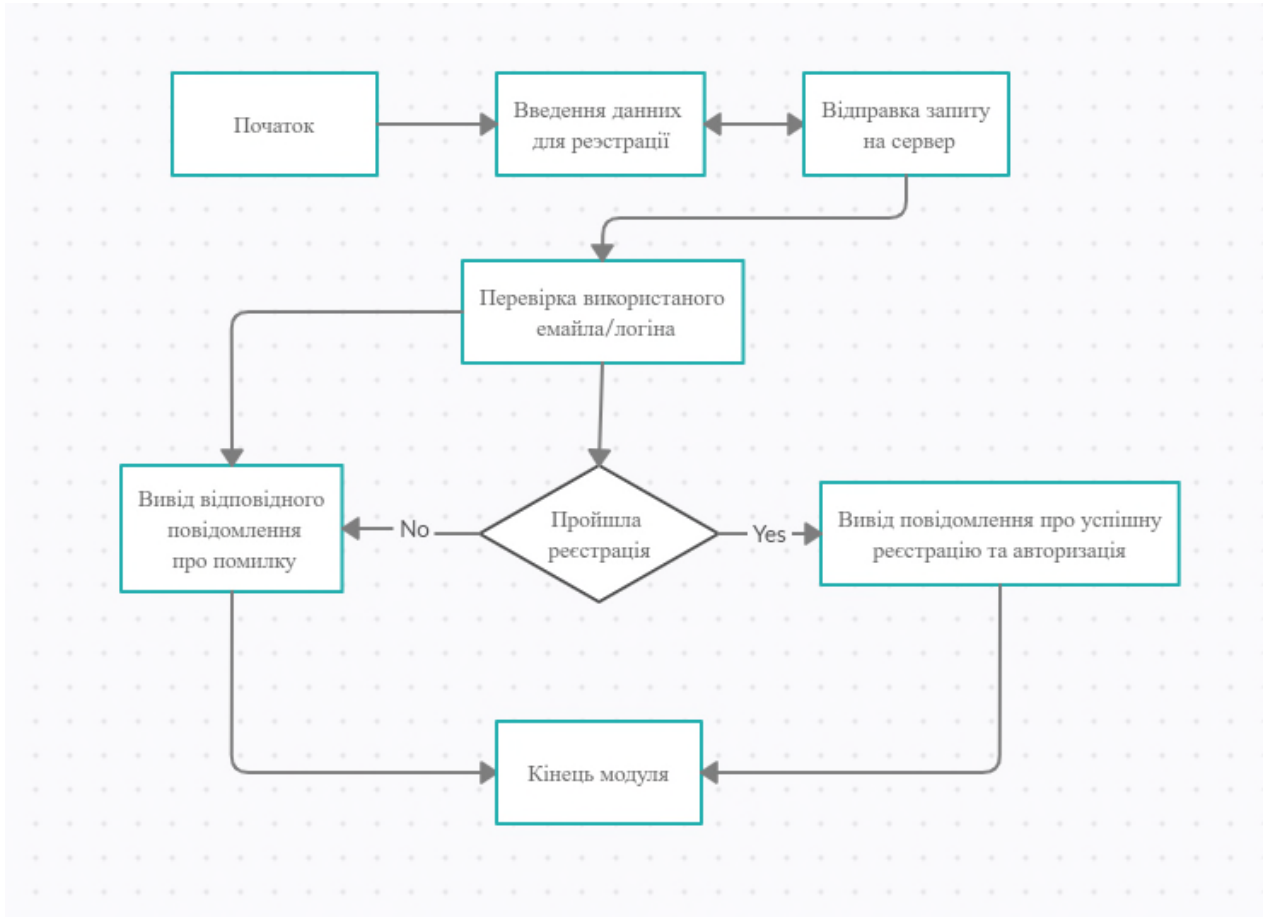


Рисунок 3.6 – Схема роботи модуля реєстрації

Для того щоб працювати з API та SDK нам потрібен токен. Для створення якого необхідні наступні параметри:

- ApiKey - Ключ API вашого облікового запису;
- ApiSecret - API Секрет вашого облікового запису;
- meetingNumber - номер наради, до якого приєднується користувач;
- Role - 1 для ведучого зустрічі, 0 для учасників.

На основі цих параметрів сервіс Zoom створює підпис тобто токен у форматі JSON, який використовується для приєднання до конференції або ж використання API. Роботу модуля зображено на рис 3.7 .

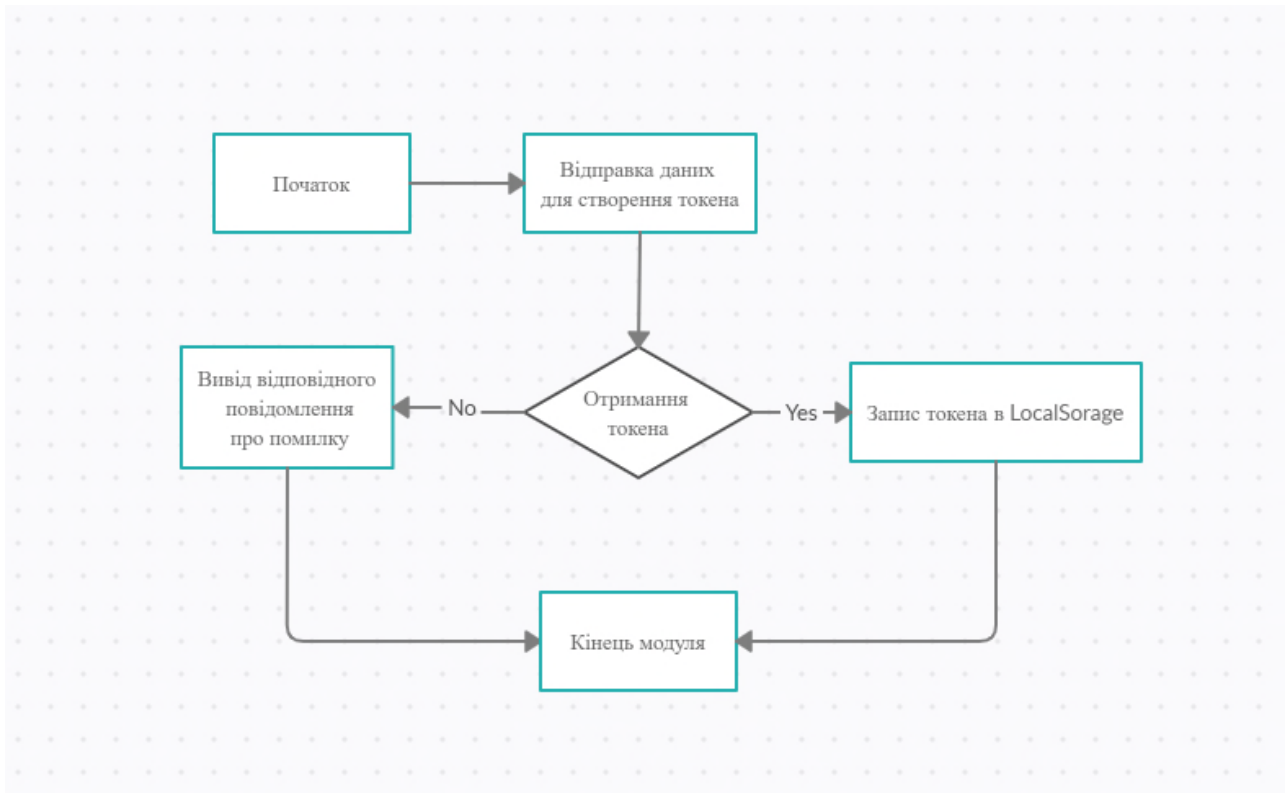


Рисунок 3.7 – Схема створення токена для роботи з API

Для того щоб отримати дані про проведену конференцію або ж створити її, потрібна API Zoom. Для цього розроблено модуль (рис 3.8), який надає змогу отримати потрібні мета дані з API та записати їх до БД для майбутнього використання їх.

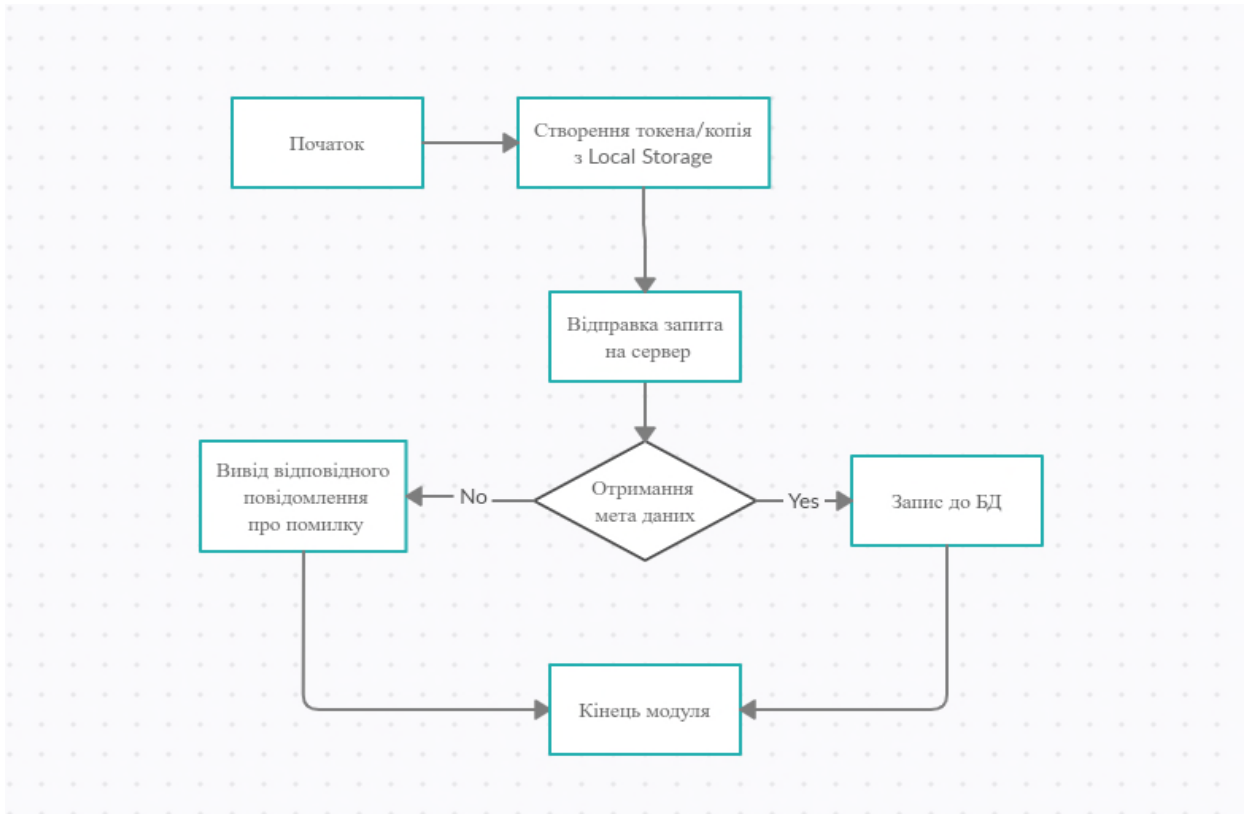


Рисунок 3.8 – Схема взаємодії з API

Для того щоб підключитись до API, потрібно відправити токен який створений і записаний у Local Storage, якщо його немає то створити за допомогою методу створення токена. Далі ми відправляємо запит на перевірку токена де при проходженні перевірки ми отримуємо потрібні мета дані онлайн-конференції які далі записуємо до бази даних для певної онлайн-конференції.

Для того щоб обробити отримані мета дані з API, є модуль для їх обробки (рис 3.9). Щоб отримати дані для обробки створюється запит на сервер для перевірки доступу до них, якщо доступ надано то отримані дані ми обробляємо та записуємо їх до БД для користувача і виводимо на екран , якщо ні то виводиться помилка про нестачу прав на отримання цієї інформації.

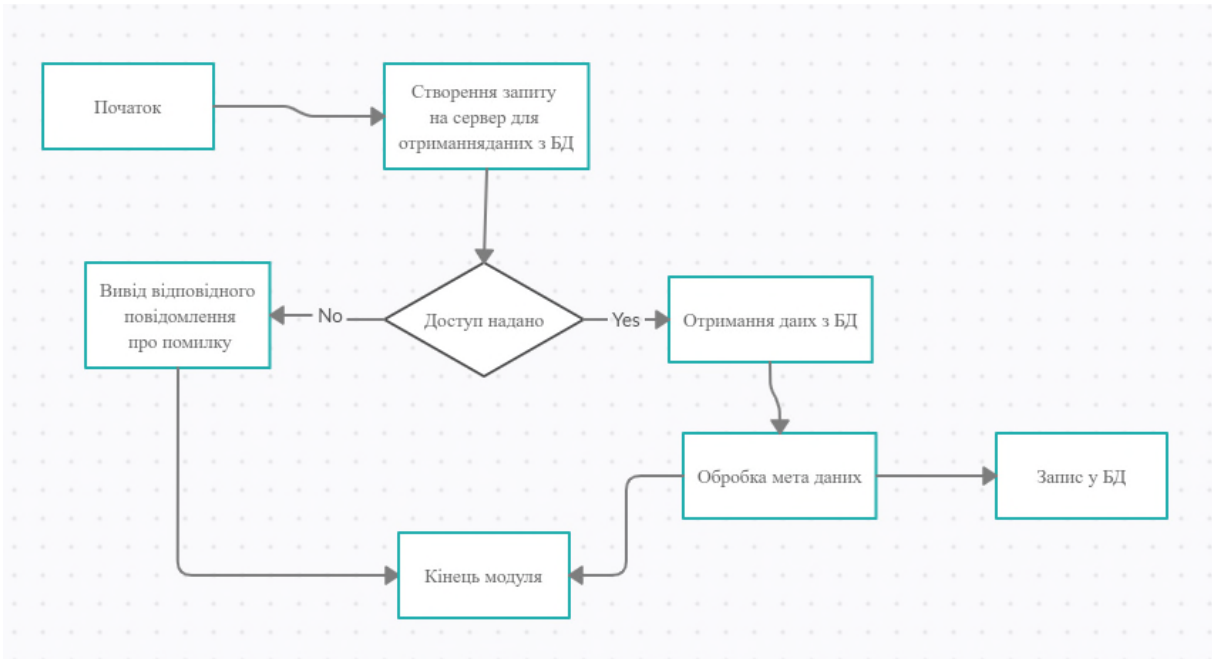


Рисунок 3.9 – Схема роботи модуля обробки мета даних

Висновки до розділу 3

В даному розділі спроектовану архітектуру інформаційної системи для онлайн-комунікації з інтеграцією SDK Zoom та роботою його API.

Розглянуто шаблони проектування. Було обрано патер проектування MVC так як він найкраще підходить для розробки інформаційної системи та подальшого удосконалення функціоналу.

Розробленні програмні компоненти інформаційної системи та створенні схеми функціонування системи.

Розділ 4

Програмна реалізація

4.1 Обґрунтування засобів реалізації

4.1.1 Мова програмування

Мова програмування – це система позначень для опису алгоритмів та структур даних, певна штучна формальна система, засобами якої можна виражати алгоритми. Мову програмування визначає набір лексичних, синтаксичних і семантичних правил, що задають зовнішній вигляд програми і дії, які виконує виконавець (комп'ютер) під її управлінням [21].

Для реалізацію інформаційної системи потрібно щоб вона була онлайн для того щоб у будьякий час нею можна було користуватись, для цього потрібен веб-сайт, також має бути можливість працювати з API/SDK Zoom та базою даних. API/SDK Zoom підтримує такі мови програмування як: Node.js, PHP, Pheton, Java, C#. Для інформаційної системи більш зручною мовою буде PHP так як вона виконує усі потрібні функції та легка у роботі з інтерфейсом а також працює з API/SDK Zoom, що надає перевагу у виборі їй.

PHP - це скриптова мова на стороні сервера. що використовується для розробки статичних веб-сайтів або динамічних веб-сайтів або веб-програм [22]. Скрипти PHP можна інтерпретувати лише на сервері, де є встановлений PHP. А для клієнтські комп'ютери які отримують доступ до скриптів PHP, потребують тільки веб-браузера так як PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта тобто у браузер.

Скриптова мова - це мова яка інтерпретує сценарії під час виконання.



Рисунок 4.1 – Схема архітектури веб-програми PHP

Існують три основні області, де використовується PHP :

- написання скриптів для виконання їх на сервері;
- написання скриптів для виконання їх у командному рядку;
- створення додатків GUI, які виконуються на стороні клієнта.

PHP також доступний для більшості операційних систем таких як Linux, Windows, Mac OS та інші. Також включена підтримка більшості сучасних веб-серверів, таких, як Apache, Microsoft Internet Information Server, Personal Web Server, серверів Netscape і iPlanet, сервера O'Reilly Website Pro, Caudium, Xitami, OmniHTTPd та багатьох інших. Для більшості серверів PHP поставляється в якості модуля, для інших, що підтримують стандарт CGI, PHP може функціонувати як процесор CGI.

Таким чином, PHP, дає свободу вибору операційної системи і веб-сервера. Крім того, з'являється вибір між використанням процедурного або об'єктно-орієнтованого програмування або ж їх поєднання.

Процедурне програмування — парадигма програмування, заснована на концепції виклику процедури. Процедури, також відомі як підпрограми, методи, або функції. Процедури містять певну послідовність кроків для виконання. В ході

виконання програми будь-яка процедура може бути викликана з будь-якого місця програми, включно з самої процедури, яка викликається [23].

Об'єктно-орієнтоване програмування — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають чотири основні концепції: інкапсуляція, успадкування, поліморфізм та абстракція. Однією з переваг ООП є краща модульність програмного забезпечення. Попри те, що ця парадигма з'явилась в 1960-тих роках, вона не мала широкого застосування до 1990-тих, коли розвиток комп'ютерів та комп'ютерних мереж дав змогу писати надзвичайно об'ємне і складне програмне забезпечення, що змусило переглянути підходи до написання програм. Сьогодні багато мов програмування або підтримують ООП або ж є цілком об'єктно-орієнтованими [24].

4.1.2 Вибір фреймворка для верстки

Макет веб-сайту - це зразок, який визначає структуру веб-сайту. Він виконує функцію структурування інформації, присутньої на сайті, як для власника веб-сайту, так і для користувачів. Він забезпечує чіткі шляхи навігації в межах веб-сторінок і ставить найважливіші елементи веб-сайту та центру. Процес створення макету і є так званою версткою.

Хороше компонування елементів на сайті, швидка загрузка та зрозуміло описані компоненти роблять важливу інформацію легко та інтуїтивно доступною для пошуку та роботи з нею, це дає можливість утримувати користувача в системі, так як при незрозумілому дизайні користувач губиться та швидко покидає його.

Тому потрібно використати якісний дизайн для інформаційної системи для того щоб користувач не мав проблем з використанням її. Також велику роль грає адаптивна верстка, яка дає можливість використовувати систему на різних платформах та приладах. Що є дуже зручним в наш час коли є багато різних гаджетів.

Верстка – це створення структури гіпертекстового документа на основі HTML-розмітки. Для стилізації верстки використовують CSS – це спеціальна мова для стилізування сторінок який використовується для їхнього зовнішнього вигляду. Для зручної та більш швидкої роботи з CSS використовують його препроцесори.

Препроцесор CSS - це надбудова CSS, яка додає раніше недоступні функції до CSS за допомогою нових конструкцій синтаксису. Препроцесори CSS перетворюють написаний за допомогою пре процесорної мови код, у чистий і дійсний код CSS.

За допомогою препроцесорів пишеться код котрий націлений на :

- читабельність для користувача;
- структурованість і логічність;
- продуктивність.

Також за допомогою JS верстка стає динамічною там має змогу працювати з серверною частиною сайту в нашому випадку інформаційної системи.

JavaScript (JS) — динамічна, об'єктно-орієнтована, прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [25].

JS надає можливість збільшити інтерактивність в інформаційній системі за допомогою динамічної обробки запитів, це надає нам можливість не перезавантажувати сторінку для отримання результату запиту за допомогою AJAX.

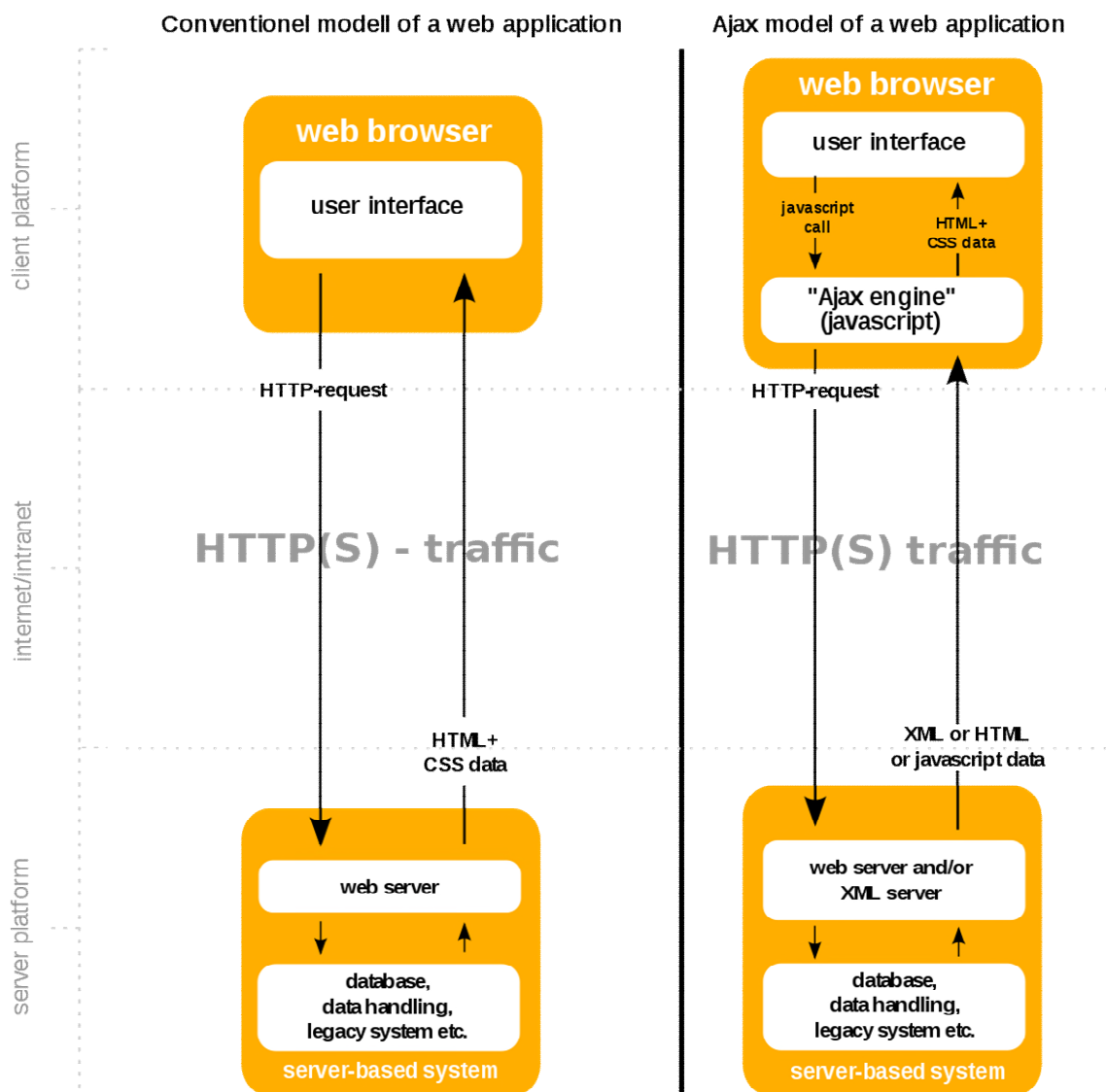


Рисунок 4.2 – Схема принципу роботи AJAX [26]

AJAX – це підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані принцип роботи зображено на рисунку 4.2 [26]. З наведеної схеми (рис 4.2) зрозуміло як працює AJAX, користувач за допомогою взаємодії з інтерфейсом звертається до JS функції яка в свій час звертаються до AJAX, який робить запит на веб сервер, у відповідь отримується відповідь в XML/HTML яка передається у функцію JS що опрацьовує

мета дані та в залежності від них виконує потрібні дії наприклад виводить сповіщення або ж проводить авторизацію користувача та інше.

Для більш зручної розробки використана відкрита версія фреймворка Bootstrap для створення веб-додатків. Його структура побудована на HTML, CSS та JavaScript (JS), щоб полегшити розробку адаптивних, мобільних веб-додатків та програм. Bootstrap включає в себе компоненти інтерфейсу користувача, макети та інструменти JS, а також рамки для реалізації.

Отже маючи фреймворк для HTML – Bootstrap , який використано для створення інтерфейсу, отримано максимально комфортний інтерфейс для взаємодії користувачу з ним.

4.2 Розробка модуля авторизації/реєстрації та особистого кабінету

Для того щоб мати доступ до особистого кабінету потрібна авторизація та реєстрація, для яких було створено підчас проектування схеми роботи авторизації та реєстрації, за допомогою яких реалізовано доступ до особистого кабінету. Для цього модуля створено 2 модальні вікна, які викликаються за допомогою двох кнопок зображених на рисунку 4.3, при авторизації вони змінюються на кнопку для переходу в особистий кабінет (рис 4.4).

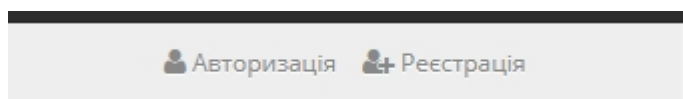


Рисунок 4.3 – Кнопки виклику модальних вікон

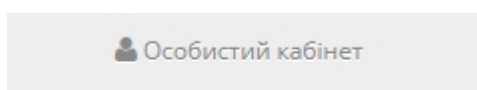
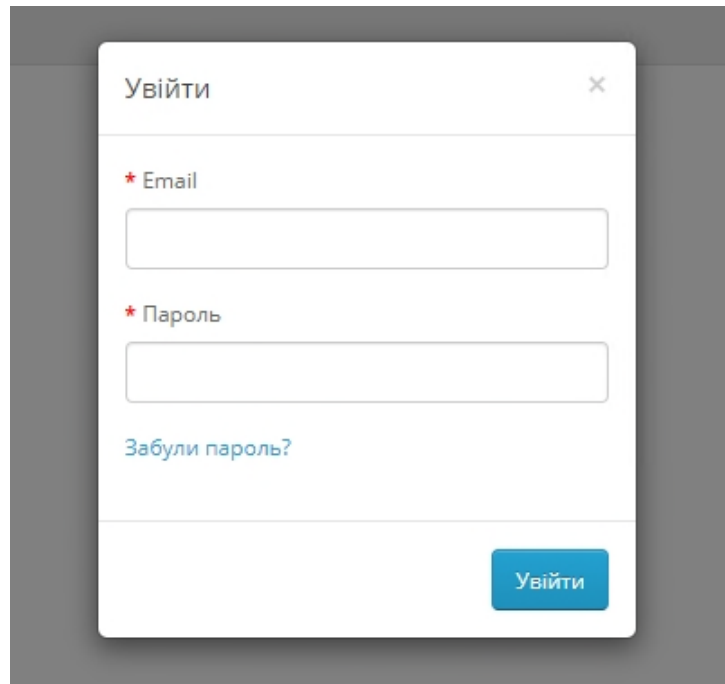


Рисунок 4.4 – Кнопка переходу в особистий кабінет

В створених модальних вікнах знаходяться форми авторизації та реєстрації, в авторизації (рис 4.5) знаходяться такі поля як e-mail та пароль для того щоб увійти в систему. Попередньо модальні вікна відкриваються при кліку за допомогою JS коду.

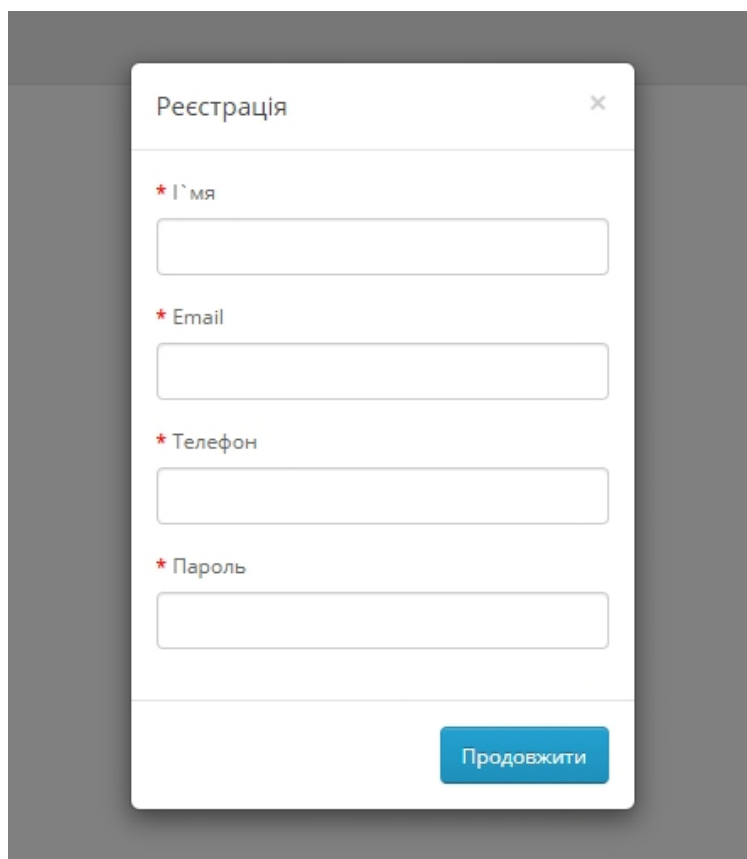
```
<script type="text/javascript">
  $(document).delegate('.quick_login', 'click', function(e) {
    $('#modal-login').modal('show');
  });
  $(document).delegate('.quick_register', 'click', function(e) {
    $('#modal-register').modal('show');
  });
</script>
```



The image shows a modal window for login. The title is "Увійти" (Login). It has a close button (X) in the top right corner. There are two input fields: "Email" and "Пароль" (Password), both with a red asterisk indicating they are required. Below the password field is a blue link "Забули пароль?" (Forgot password?). At the bottom right is a blue button labeled "Увійти" (Login).

Рисунок 4.5 – Модальне вікно авторизації

В реєстрації (рис 4.6) знаходяться такі поля як ім'я, e-mail, телефон та пароль.



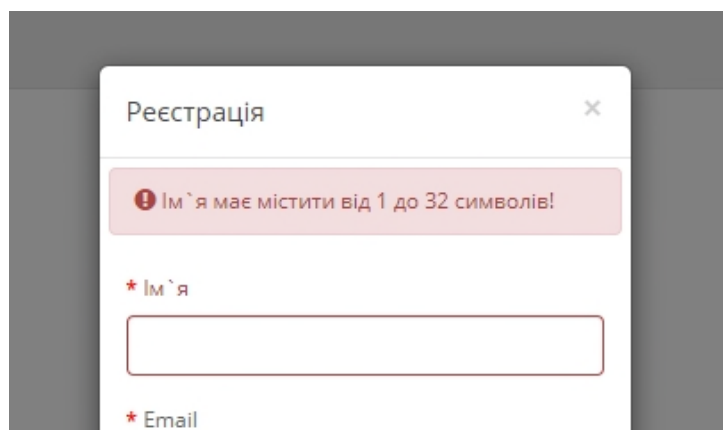
The image shows a registration modal window titled "Реєстрація" (Registration) with a close button (X) in the top right corner. The form contains four input fields, each with a red asterisk indicating a required field:

- * Ім'я (Name)
- * Email
- * Телефон (Phone)
- * Пароль (Password)

At the bottom right of the form is a blue button labeled "Продовжити" (Continue).

Рисунок 4.6 – Модальне вікно реєстрації

Після відправки запиту на реєстрацію здійснюється перевірка на уже використаний e-mail та на правильність введених полів, якщо запит відбувся успішно то відразу відбудеться авторизація, при невдалій реєстрації будуть виведені помилки в такому виді як зображено на рисунку 4.7.



The image shows the same registration modal window as in Figure 4.6, but with an error message displayed at the top. The error message is: "Ім'я має містити від 1 до 32 символів!" (Name must contain 1 to 32 characters!). The "Ім'я" (Name) input field is highlighted with a red border, indicating the source of the error. The "Email" field is also visible below it.

Рисунок 4.7 – Вивід помилок при реєстрації

Перевірка правильності введених даних виконується в контролері quicksignup.php.

```

if ((utf8_strlen(trim($this->request->post['name'])) < 1) ||
(utf8_strlen(trim($this->request->post['name'])) > 32)) {
    $json['error'] = $json['error_name'] = $this->language-
    >get('error_name');
}
if ((utf8_strlen($this->request->post['email']) > 96) ||
!preg_match('/^[^\@]+\@.*\.[a-z]{2,6}$/i', $this->request-
>post['email'])) {
    $json['error'] = $json['error_email'] = $this->language-
    >get('error_email');
}
if ($this->model_account_customer->getTotalCustomersByEmail($this->request-
>post['email'])) {
    $json['error'] = $json['error_email'] = $this->language-
    >get('error_exists');
}
if ((utf8_strlen($this->request->post['telephone']) < 3) ||
(utf8_strlen($this->request->post['telephone']) > 32)) {
    $json['error'] = $json['error_telephone'] = $this->language-
    >get('error_telephone');
}
if ((utf8_strlen($this->request->post['password']) < 4) ||
(utf8_strlen($this->request->post['password']) > 20)) {
    $json['error'] = $json['error_password'] = $this->language-
    >get('error_password');
}
if ($this->config->get('config_account_id')) {
    $this->load->model('catalog/information');
    $information_info = $this->model_catalog_information-
    >getInformation($this->config->get('config_account_id'));
    if ($information_info && !isset($this->request->post['agree'])) {
        json['error'] = sprintf($this->language->get('error_agree'),
    $information_info['title']);
    }
}
}

```

Після запиту дані потрапляють в контролер в якому проходять через перевірки для кожного поля де при виявленні її відправляються до темплейта звідки був здійснений запит, де за допомогою js приймається відповідь (рис 4.8). Яка обробляється та виводиться на екран користувачу (рис 4.7).

```

$('#quick-register .createaccount').click(function() {
$.ajax({
url: 'index.php?route=common/quicksignup/register',
type: 'post',
data: $('#quick-register input[type=\text\'], #quick-register input[type=\password\'], #quick-register input[type=\checkbox\']:checked'),
dataType: 'json',
beforeSend: function() {
$('#quick-register .createaccount').button('loading');
$('#modal-register .alert-danger').remove();
},
complete: function() {
$('#quick-register .createaccount').button('reset');
},
success: function(json) {
$('#modal-register .form-group').removeClass('has-error');

if(json['islogged']){
window.location.href="index.php?route=account/account";
}
if (json['error_name']) {...}
if (json['error_email']) {...}
if (json['error_telephone']) {...}
if (json['error_password']) {...}
if (json['error']) {
$('#modal-register .modal-header').after('<div class="alert alert-danger" style="margin:5px;"><i class="fa fa-exclamation-circle"></i> '
}
if (json['now_login']) {
$('.quick-login').before('<li class="dropdown"><a href="<?php echo $account; ?>" title="<?php echo $text_account; ?>" class="dropdown-tog
$('.quick-login').remove();
}
if (json['success']) {
$('#modal-register .main-heading').html(json['heading_title']);
success = json['text_message'];
success += '<div class="buttons"><div class="text-right"><a onclick="location();" class="btn btn-primary"> + json['button_continue'] + '<
$('#modal-register .modal-body').html(success);
}
}
});
});

```

Рисунок 4.8 – Обробка відповіді на передані дані форми реєстрації

При успішній авторизації користувачу надається доступ до особистого кабінету зображеному на рисунку 4.9.

Особистий кабінет

Обліковий запис

* email

* Імя

* Прізвище

* Тел.

Користувач Студент

Обліковий запис

Зміна пароля

Заняття

Історія відвідування занять

Вийти

Продовжити

Рисунок 4.9 – Обробка відповіді на передані дані форми реєстрації

В котрому користувач може виправити особисті дані, змінити пароль, перейти до списку занять, та переглянути історію відвідування занять. В заняттях користувач може переглянути інформацію про проведені раніш заняття, перейти до запланованих занять або індивідуально запланованих конференцій. Таким чином увесь процес проведення дистанційного навчання переводиться у одну загальну інформаційну систему, вирішуючи проблему проведення та комунікації студентів з викладачем.

4.3 Розробка модуля взаємодії із API Zoom

Для того щоб користуватись Zoom API потрібен згенерований ключ. Ключ генерується за допомогою запиту на сервер Zoom. Для цього потрібні такі дані як :

- Api_key – ключ API облікового запису Zoom;
- Api_sercet – API секрет облікового запису Zoom;
- Meating_number – номер конференції до якої піденуємося;
- Role – роль 1 для викладача конференції, 0 для студента.

При використанні функціоналу API Zoom відправляється запит на сервер з даними для створення ключа, які в свою чергу передаються до серверу Zoom та у відповідь отримуємо ключ для роботи з API.

```
public function generate_signature ( $api_key, $api_sercet, $meeting_number, $role){
    $time = time() * 1000 - 30000;
    $data = base64_encode($api_key . $meeting_number . $time . $role);
    $hash = hash_hmac('sha256', $data, $api_sercet, true);
    $_sig = $api_key . "." . $meeting_number . "." . $time . "." . $role . "." .
    base64_encode($hash);
    return rtrim(strtr(base64_encode($_sig), '+/', '-_'), '=');
}
```

У відповідь отримуються дані в форматі JSON, в якому знаходяться ключ для підключення до конференції (рис 4.10).

```
{
  "signature": "eNUzSlBhQV9SSlcyLTlsNV9IQWFMQS4xMjM0NTY3ODkuMTU4MzE2OTUzODc3My4wLkJMNETi"
}
```

Рисунок 4.10 – Ключ для під'єднання до онлайн-конференції

Згенерований ключ надає можливість працювати з API Zoom, основні можливості це робота з конференцією, звіти, чати та можливість реєструвати користувачів в системі Zoom. За допомогою ключа також здійснюється підключення до онлайн-конференцію вигляд якої зображено на рисунку 4.12

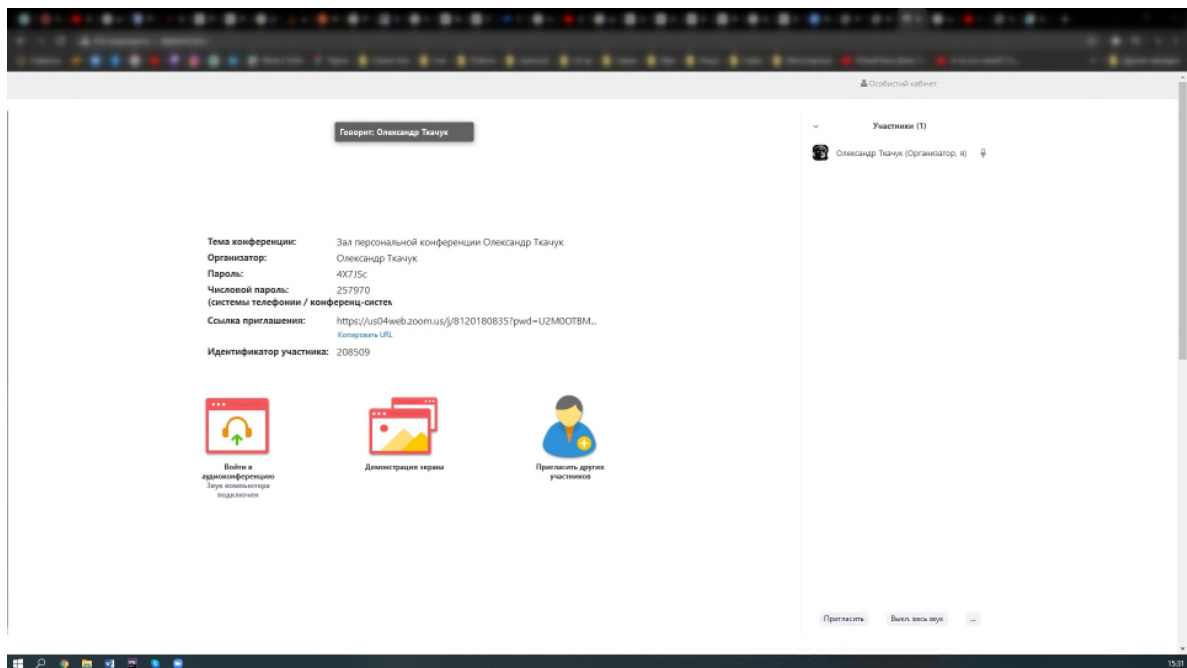


Рисунок 4.12 – Ключ для під'єднання до онлайн-конференції

4.4 Розробка модуля для створення, планування заняття

Для створення або планування пари створено модуль createMeeting, який робить запит до API передаючи параметри для створення її.

```

function createMeeting (userId,oauth_access_token){
  let data = {
    userId: userId,
    oauth_access_token: oauth_access_token,
    body:[
      {
        "topic": "string",
        "type": "integer",
        "start_time": "string [date-time]",
        "duration": "integer",
        "schedule_for": "string",
        "timezone": "string",
        "password": "string",
        "agenda": "string",
        "recurrence": {
          "type": "integer",
          "repeat_interval": "integer",
          "weekly_days": "string",
          "monthly_day": "integer",
          "monthly_week": "integer",
          "monthly_week_day": "integer",
          "end_times": "integer",
          "end_date_time": "string [date-time]"
        },
        "settings": {...}
      }
    ]
  }
  $.ajax({
    url: `https://api.zoom.us/v2/report/users/${userId}/meetings`,
    type: 'post',
    data: data,
    dataType: 'json',
    success: function(json) {
      return json
    },
    error: function (json){
      return json
    }
  });
}

```

В об'єкті settings вказуються параметри конференції такі як тема, опис, час проведення, тривалість конференції також надається можливість створити пароль для підключення до конференції, налаштування відео для користувачів де надається можливість відключити трансляцію відео для усіх учасників окрім організатора та

навпаки. Додатково інформаційна система надає можливість вказати групу, або ж конкретно студентів які будуть запрошені на заняття. Далі після успішного виконаного запиту отримані дані записуються у базу даних для усіх користувачів яких відзначили при плануванні або створенні заняття. Після чого ці дані відображаються у вказаних студентів та викладачів на сторінці занять, за допомогою чого в подальшому можна під'єднатися до заняття.

4.5 Розробка модуля для створення списку відвідування заняття

Для отримання історії відвідуваних занять студента, створена функція отримання мета даних про відвідані зустрічі.

```
function confHistoryUser (userId,oauth_access_token){
  let data = {
    userId: userId,
    oauth_access_token: oauth_access_token
  }
  $.ajax({
    url: `https://api.zoom.us/v2/report/users/${userId}/meetings`,
    type: 'get',
    data: data,
    dataType: 'json',
    success: function(json) {
      return json
    },
    error: function (json){
      return json
    }
  });
}
```

У відповідь на успішний запит буде отримано об'єкт даних з відвідуваними конференціями (рис 4.13). Після чого отримані мета дані відправляються на обробку та записуються у базу даних.

```

{
  "from": "2019-07-14",
  "to": "2019-08-14",
  "page_count": 1,
  "page_size": 30,
  "total_records": 9,
  "next_page_token": "",
  "meetings": [
    {
      "uuid": "/lqsAAA5V88Bw==",
      "id": 12345,
      "host_id": "gdfg3434dsssfss",
      "type": 2,
      "topic": "MyTestMeetings",
      "user_name": "Prashant Bhandari",
      "user_email": "pb@hjdskfhds.fdkhfhds",
      "start_time": "2019-07-15T23:24:52Z",
      "end_time": "2019-07-15T23:30:19Z",
      "duration": 6,
      "total_minutes": 11,
      "participants_count": 2
    },
    {
      "uuid": "411bbbbbcVyw==",
      "id": 00000,
      "host_id": "gdfg3434dsssfss",
      "type": 2,
      "topic": "My Meeting",
      "user_name": "Raju Khanal",
      "user_email": "srhkdjfh@jh.fsadghjm",
      "start_time": "2019-07-16T17:14:39Z",
      "end_time": "2019-07-16T17:26:20Z",
      "duration": 12,
      "total_minutes": 17,
      "participants_count": 2
    }
  ]
}

```

Рисунок 4.13 – Отримані мета дані студента

Мета дані відправляються на обробку для створення списку відвідування, що записуються для кожного користувача, який був обраний при створенні заняття.

```

public function userHistoryMeeting ( $data ) {
    $listHistory = array();

    foreach ($data as $item) {
        $listHistory[] = array(
            'id'           => $item['id'],
            'topic'        => $item['topic'],
            'name'         => $item['user_name'],
            'start_time'   => $item['start_time'],
            'end_time'     => $item['end_time'],
            'total_minutes' => $item['total_minutes']
        );
    }
}

```

```

$this->model_account_zoom->getHistory($listHistory);

$this->response->addHeader('Content-Type: application/json');
$this->response->setOutput(true);
}

```

Для створення списку відвідування проведеної пари, створений модуль для відправлення запиту на отримання мета даних про проведення пари.

```

function confHistoryInfo (meetingId,oauth_access_token){
    let data = {
        userId: meetingId,
        oauth_access_token: oauth_access_token
    }
    $.ajax({
        url:
`https://api.zoom.us/v2/report/meetings/${meetingId}/participants`,
        type: 'get',
        data: data,
        dataType: 'json',
        success: function(json) {
            return json
        },
        error: function (json){
            return json
        }
    });
}

```

Історія відвідування занять




Назва	Дата проведення	Тривалість Заняття	Тривалість перебування на занятті	
Алгоритмізація програмування	29.11.2020	00:25	00:00	
Алгоритмізація програмування	28.11.2020	00:36	00:30	
Алгоритмізація програмування	27.11.2020	00:15	00:10	

Рисунок 4.14 – Відображення звіту про відвідування занять

Алгоритмізація програмування

Дата проведення : 27.11.2020 | Тривалість: 00:15





І.П.Б	Перебування	Відсоток перебування на занятті	
Кравчук Станіслав Олександрович	00:12	80%	
Ткачук Олександр Сергійович	00:10	66%	
Яков Ярослав Петрович	00:05	33%	
Цвітков Максим Васильович	00:01	6%	

Рисунок 4.15 – Відображення детального звіту про проведені заняття

Після чого отримані мета дані передаються до модуля обробки та записуються у базу даних до певного заняття та для присутніх котрі були на ній, та відображаються на сторінці самого заняття або ж у особистому кабінеті в історії відвідувань приклад відображення зображено на рисунку 4.14. Також надається можливість переглянути звіт про окремий предмет перейшовши за допомогою кнопки навпроти предмета (рис 4.14). На сторінці предмета відображається список усіх хто був присутній на занятті, та відображається час який користувач перебував на занятті, який вираховується у відсотках та в залежності від них фарбується кольором для більш зручного розуміння відвідування. Таким чином вирішується проблема відслідковування присутності студентів при дистанційному навчанні, за допомогою чого викладачі контролюють відвідування занять.

Висновки до розділу 4

В даному розділі обґрунтовано засоби реалізації інформаційної системи за допомогою мови програмування PHP та використання фреймворку для верстки Bootstrap.

Розроблено модуль авторизації та реєстрації яка надає доступ до особистого кабінету та дає можливість працювати з функціоналом інформаційної системи.

Розроблено модуль взаємодії з Zoom API за допомогою чого надається доступ до роботи з API та можливість використовувати функціонал Zoom.

Розроблено модуль для взаємодії інформаційної системи з API Zoom для створення або планування заняття.

Розроблено модуль створення списку відвідування занять, який надає можливість відслідковувати присутність студентів на парі та кількість проведеного часу на занятті.

В результаті виконання даного розділу було створено програмну реалізацію інформаційної системи для соціальної мережі університету.

Загальні висновки

Запропонована інформаційна система призначена для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання. Інтеграція з API Zoom та SDK Zoom надає можливість проводити онлайн-конференції.

Проведено аналіз онлайн-комунікацій вчитель-студент. Детально розглянуто види комунікацій також проведено дослідження та порівняння технологій API та SDK.

Розглянуто та проведено порівняння існуючих сервісів відео конференцій та сформовано вимоги для інформаційної системи. Запропоновано інформаційну систему для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання на основі сформованих вимог.

Спроектовано архітектуру інформаційної систем та загальну схему інформаційної системи, а також компоненти системи основний з яких робота з API/SDK Zoom.

Розроблено програмний продукт, в якому реалізовано всі функціональні вимоги інформаційної системи для забезпечення взаємодії між студентами та викладачем під час дистанційного навчання.

Перелік посилань

1. Мережеве спілкування [електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мережеве_спілкування
2. Інтернет як засіб соціальної комунікації та соціального впливу [електронний ресурс]. – Режим доступу: http://nbuv.gov.ua/j-pdf/PoMe_2008_4_14.pdf
3. Discord [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Discord>
4. Telegram [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Telegram>
5. Viber [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Viber>
6. Skype [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Skype>
7. Zoom [електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Zoom_Video_Communications
8. Google Meet [електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Google_Hangouts
9. SDK і приклади додатків [електронний ресурс]. – Режим доступу: <https://starkovden.github.io/sdks-sample-apps.html>
10. SDK [електронний ресурс]. – Режим доступу: <https://wiki.moda/wikipedia/SDK>
11. Що таке програмний інтерфейс додатків (API) [електронний ресурс]. – Режим доступу: <https://www.wrike.com/ru/blog/chto-takoe-programmnyj-interfejs-prilozhenij-api/>
12. Переваги та недоліки дистанційного навчання [електронний ресурс]. – Режим доступу: <https://ru.osvita.ua/vnz/reports/adv/46958/>
13. What is a SDK and an API? [електронний ресурс]. – Режим доступу: <https://www.skyhook.com/blog/what-is-an-sdk-and-an-api>

14. Що таке SDK? Все, що потрібно знати [електронний ресурс]. – Режим доступу: <https://clevertap.com/blog/what-is-an-sdk/>
15. Веб конференції [електронний ресурс]. – Режим доступу: <https://roi4cio.com/categories/category/veb-konferencii/>
16. Стаття про Proficonf [електронний ресурс]. – Режим доступу: <https://lmslist.ru/webinar-platform/obzor-proficonf/>
17. Model-View-Controller [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Model-View-Controller>
18. Введення в MVC [електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/aspnet5/3.1.php>
19. Proficonf [електронний ресурс]. – Режим доступу: <https://conf.apiway.ai/ru/software/proficonf>
20. API Zoom [електронний ресурс]. – Режим доступу: <https://marketplace.zoom.us/docs/api-reference/using-zoom-apis>
21. Мова програмування [електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F
22. What is PHP? Write your first PHP Program [електронний ресурс]. – Режим доступу: <https://www.guru99.com/what-is-php-first-php-program.html>
23. Процедурне програмування [електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F
24. Об'єктно-орієнтоване програмування [електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0>

%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%
BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F

25. JavaScript [електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/JavaScript>

26. AJAX [електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/AJAX>

27. НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ «АПКН-2020» –
[Електронний ресурс]. – Режим доступу: <https://kn.khnu.km.ua/page.aspx?r=3&p=7>;

ДОДАТКИ

Додаток А

Програмні коди

```

<?php
class ControllerCommonHeader extends Controller {
    public function index() {
        // Analytics
        $this->load->model('extension/extension');

        $data['analytics'] = array();

        $analytics = $this->model_extension_extension->getExtensions('analytics');

        foreach ($analytics as $analytic) {
            if ($this->config->get($analytic['code'] . '_status')) {
                $data['analytics'][] = $this->load->controller('analytics/' .
$analytic['code']);
            }
        }

        if ($this->request->server['HTTPS']) {
            $server = $this->config->get('config_ssl');
        } else {
            $server = $this->config->get('config_url');
        }

        if (is_file(DIR_IMAGE . $this->config->get('config_icon'))) {
            $this->document->addLink($server . 'image/' . $this->config-
>get('config_icon'), 'icon');
        }

        $data['title'] = $this->document->getTitle();

        $data['base'] = $server;
        $data['description'] = $this->document->getDescription();
        $data['keywords'] = $this->document->getKeywords();
    }
}

```

```

    $data['links'] = $this->document->getLinks();
    $data['styles'] = $this->document->getStyles();
    $data['scripts'] = $this->document->getScripts();
/**
 * задаємо мета robots
 * можна в необхідному контролері додати $this->document->addMetaRobots
 * також через vqmod добавляється $data['robots'] для кастомних сторінок, можна
добавити через адмінку в модулі "Seo кастомных страниц"
 */
    $data['meta'] = $this->document->getMetaRobots();
    $data['lang'] = $this->language->get('code');
    $data['direction'] = $this->language->get('direction');

    $data['alter_lang'] = $this->getAlterLanguageLinks($this->document-
>getLinks(),$data['lang'],$this->config->get('config_language_default'));
    $data['name'] = $this->config->get('config_name');
    $data['gtm'] = trim($this->config->get('config_gtm'));

    if (is_file(DIR_IMAGE . $this->config->get('config_logo'))) {
        $data['logo'] = $server . 'image/' . $this->config->get('config_logo');
    } else {
        $data['logo'] = '';
    }

    $this->load->language('common/header');
    $data['og_url'] = ((isset($this->request->server['HTTPS']) && !empty($this-
>request->server['HTTPS'])) ? HTTPS_SERVER : HTTP_SERVER) . substr($this->request-
>server['REQUEST_URI'], 1, (strlen($this->request->server['REQUEST_URI'])-1));
    $data['og_image'] = $this->document->getOgImage();

    $data['text_home'] = $this->language->get('text_home');

    // Wishlist
    if ($this->customer->isLoggedIn()) {
        $this->load->model('account/wishlist');

        $data['text_wishlist'] = sprintf($this->language->get('text_wishlist'),
$this->model_account_wishlist->getTotalWishlist());
    } else {

```

```

        $data['text_wishlist'] = sprintf($this->language->get('text_wishlist'),
(isset($this->session->data['wishlist']) ? count($this->session->data['wishlist']) : 0));
    }

```

```

    $data['text_shopping_cart'] = $this->language->get('text_shopping_cart');
    $data['text_logged'] = sprintf($this->language->get('text_logged'), $this->url-
>link('account/account', '', 'SSL'), $this->customer->getFirstName(), $this->url-
>link('account/logout', '', 'SSL'));

```

```

    $data['text_account'] = $this->language->get('text_account');
    $data['text_register'] = $this->language->get('text_register');
    $data['text_login'] = $this->language->get('text_login');
    $data['text_order'] = $this->language->get('text_order');
    $data['text_transaction'] = $this->language->get('text_transaction');
    $data['text_download'] = $this->language->get('text_download');
    $data['text_logout'] = $this->language->get('text_logout');
    $data['text_checkout'] = $this->language->get('text_checkout');
    $data['text_page'] = $this->language->get('text_page');
    $data['text_category'] = $this->language->get('text_category');
    $data['text_all'] = $this->language->get('text_all');

```

```

    $data['home'] = $this->url->link('common/home');
    $data['wishlist'] = $this->url->link('account/wishlist', '', 'SSL');
    $data['logged'] = $this->customer->isLoggedIn();
    $data['account'] = $this->url->link('account/account', '', 'SSL');
    $data['register'] = $this->url->link('account/register', '', 'SSL');
    $data['login'] = $this->url->link('account/login', '', 'SSL');
    $data['order'] = $this->url->link('account/order', '', 'SSL');
    $data['transaction'] = $this->url->link('account/transaction', '', 'SSL');
    $data['download'] = $this->url->link('account/download', '', 'SSL');
    $data['logout'] = $this->url->link('account/logout', '', 'SSL');
    $data['shopping_cart'] = $this->url->link('checkout/cart');
    $data['checkout'] = $this->url->link('checkout/checkout', '', 'SSL');
    $data['contact'] = $this->url->link('information/contact');
    $data['telephone'] = $this->config->get('config_telephone');

```

```

    $status = true;

```

```

    if (isset($this->request->server['HTTP_USER_AGENT'])) {

```

```

        $robots = explode("\n", str_replace(array("\r\n", "\r"), "\n",
trim($this->config->get('config_robots'))));

        foreach ($robots as $robot) {
            if ($robot && strpos($this->request->server['HTTP_USER_AGENT'],
trim($robot)) !== false) {
                $status = false;

                break;
            }
        }
    }

    // Menu
    $this->load->model('catalog/category');

    $this->load->model('catalog/product');

    $data['categories'] = array();

    $categories = $this->model_catalog_category->getCategories(0);

    foreach ($categories as $category) {
        if ($category['top']) {
            // Level 2
            $children_data = array();

            $children = $this->model_catalog_category-
>getCategories($category['category_id']);

            foreach ($children as $child) {
                $filter_data = array(
                    'filter_category_id' => $child['category_id'],
                    'filter_sub_category' => true
                );

                $children_data[] = array(
                    'name' => $child['name'] . ($this->config-
>get('config_product_count') ? ' (' . $this->model_catalog_product-
>getTotalProducts($filter_data) . ')' : ''),

```

```

                'href'      => $this->url->link('product/category',
'path=' . $category['category_id'] . '-' . $child['category_id'])
            );
        }

        // Level 1
        $data['categories'][] = array(
            'name'      => $category['name'],
            'children' => $children_data,
            'column'   => $category['column'] ? $category['column'] : 1,
            'href'     => $this->url->link('product/category', 'path=' .
$category['category_id'])
        );
    }
}

$data['language'] = $this->load->controller('common/language');
$data['currency'] = $this->load->controller('common/currency');
$data['search'] = $this->load->controller('common/search');
$data['cart'] = $this->load->controller('common/cart');

// For page specific css
if (isset($this->request->get['route'])) {
    if (isset($this->request->get['product_id'])) {
        $class = '-' . $this->request->get['product_id'] . ' product-page';
    } elseif (isset($this->request->get['path'])) {
        $class = '-' . $this->request->get['path'] . ' category-page';
    } elseif (isset($this->request->get['manufacturer_id'])) {
        $class = '-' . $this->request->get['manufacturer_id'] . '
manufacturer-page';
    } else {
        $class = '';
    }

    $data['class'] = str_replace('/', '-', $this->request->get['route']) .
$class;
} else {
    $data['class'] = 'common-home';
}

```

```

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template')
'/template/common/header.tpl')) {
            return $this->load->view($this->config->get('config_template')
'/template/common/header.tpl', $data);
        } else {
            return $this->load->view('default/template/common/header.tpl', $data);
        }
    }
}

```

```

protected function getAlterLanguageLinks($links,$lang_cur,$lang_def) {
    $result = array();
    if ($this->config->get('config_seo_url')) {
        foreach($links as $link) {
            if($link['rel']=='canonical') {
                $url=$link['href'];
                $schema = parse_url($url,PHP_URL_SCHEME);
                $server = strtolower($schema)=='https' ? HTTPS_SERVER : HTTP_SERVER;
                $cur_lang = substr($url, strlen($server),2);
                if($lang_cur == $cur_lang){
                    $query = substr($url, strlen($server)+2);
                }else{
                    $cur_lang = $lang_def;
                    $query = '/' .str_replace($server,'',$url);
                }
                $this->load->model('localisation/language');
                $languages = $this->model_localisation_language->getLanguages();
                $active_langs = array();
                foreach($languages as $lang) {
                    if($lang['status']) {
                        $active_langs[]=$lang['code'];
                    }
                }
                if(in_array($cur_lang, $active_langs)) {
                    foreach($active_langs as $lang) {
                        if($lang_def != $lang){
                            $result[$lang] = $server.$lang.($query ? $query : '');
                        }else{
                            $pos = strpos($query, '/');
                            if($pos == 0){

```



```

$data['text_phone'] = $this->language->get('text_phone');
$data['entry_name'] = $this->language->get('entry_name');
$data['entry_phone'] = $this->language->get('entry_phone');
$data['button_continue'] = $this->language->get('button_continue');
$data['text_loading'] = $this->language->get('text_loading');

$this->load->model('catalog/information');

$data['informations'] = array();

foreach ($this->model_catalog_information->getInformations() as $result) {
    if ($result['bottom']) {
        $data['informations'][] = array(
            'title' => $result['title'],
            'href' => $this->url->link('information/information',
'information_id=' . $result['information_id'])
        );
    }
}

$data['contact'] = $this->url->link('information/contact');
$data['return'] = $this->url->link('account/return/add', '', 'SSL');
$data['sitemap'] = $this->url->link('information/sitemap');
$data['manufacturer'] = $this->url->link('product/manufacturer');
$data['voucher'] = $this->url->link('account/voucher', '', 'SSL');
$data['affiliate'] = $this->url->link('affiliate/account', '', 'SSL');
$data['special'] = $this->url->link('product/special');
$data['account'] = $this->url->link('account/account', '', 'SSL');
$data['order'] = $this->url->link('account/order', '', 'SSL');
$data['wishlist'] = $this->url->link('account/wishlist', '', 'SSL');
$data['newsletter'] = $this->url->link('account/newsletter', '', 'SSL');

$data['powered'] = sprintf($this->language->get('text_powered'), $this->config-
>get('config_name'), date('Y', time()));

$data['user_name'] = $this->customer->getFirstName();
$data['user_phone'] = $this->customer->getTelephone();

// Whos Online

```

```

if ($this->config->get('config_customer_online')) {
    $this->load->model('tool/online');

    if (isset($this->request->server['REMOTE_ADDR'])) {
        $ip = $this->request->server['REMOTE_ADDR'];
    } else {
        $ip = '';
    }

    if (isset($this->request->server['HTTP_HOST']) && isset($this->request->server['REQUEST_URI'])) {
        $url = 'http://' . $this->request->server['HTTP_HOST'] . $this->request->server['REQUEST_URI'];
    } else {
        $url = '';
    }

    if (isset($this->request->server['HTTP_REFERER'])) {
        $referer = $this->request->server['HTTP_REFERER'];
    } else {
        $referer = '';
    }

    $this->model_tool_online->addOnline($ip, $this->customer->getId(), $url, $referer);
}

if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/common/footer.tpl')) {
    return $this->load->view($this->config->get('config_template') . '/template/common/footer.tpl', $data);
} else {
    return $this->load->view('default/template/common/footer.tpl', $data);
}
}
}
<?php
class ControllerAccountAccount extends Controller {
    public function index() {
        if (!$this->customer->isLoggedIn()) {

```

```

        $this->session->data['redirect'] = $this->url->link('account/account',
'', 'SSL');

        $this->response->redirect($this->url->link('account/login', '', 'SSL'));
    }

    $this->load->language('account/account');

    $this->document->setTitle($this->language->get('heading_title'));

    $data['breadcrumbs'] = array();

    $data['breadcrumbs'][] = array(
        'text' => $this->language->get('text_home'),
        'href' => $this->url->link('common/home')
    );

    $data['breadcrumbs'][] = array(
        'text' => $this->language->get('text_account'),
        'href' => $this->url->link('account/account', '', 'SSL')
    );

    if (isset($this->session->data['success'])) {
        $data['success'] = $this->session->data['success'];

        unset($this->session->data['success']);
    } else {
        $data['success'] = '';
    }

    $data['heading_title'] = $this->language->get('heading_title');

    $data['text_my_account'] = $this->language->get('text_my_account');
    $data['text_my_orders'] = $this->language->get('text_my_orders');
    $data['text_my_newsletter'] = $this->language->get('text_my_newsletter');
    $data['text_edit'] = $this->language->get('text_edit');
    $data['text_password'] = $this->language->get('text_password');
    $data['text_address'] = $this->language->get('text_address');
    $data['text_wishlist'] = $this->language->get('text_wishlist');
    $data['text_order'] = $this->language->get('text_order');

```

```

$data['text_download'] = $this->language->get('text_download');
$data['text_reward'] = $this->language->get('text_reward');
$data['text_return'] = $this->language->get('text_return');
$data['text_transaction'] = $this->language->get('text_transaction');
$data['text_newsletter'] = $this->language->get('text_newsletter');
$data['text_recurring'] = $this->language->get('text_recurring');

$data['edit'] = $this->url->link('account/edit', '', 'SSL');
$data['password'] = $this->url->link('account/password', '', 'SSL');
$data['address'] = $this->url->link('account/address', '', 'SSL');
$data['wishlist'] = $this->url->link('account/wishlist');
$data['order'] = $this->url->link('account/order', '', 'SSL');
$data['download'] = $this->url->link('account/download', '', 'SSL');
$data['return'] = $this->url->link('account/return', '', 'SSL');
$data['transaction'] = $this->url->link('account/transaction', '', 'SSL');
$data['newsletter'] = $this->url->link('account/newsletter', '', 'SSL');
$data['recurring'] = $this->url->link('account/recurring', '', 'SSL');

if ($this->config->get('reward_status')) {
    $data['reward'] = $this->url->link('account/reward', '', 'SSL');
} else {
    $data['reward'] = '';
}

$data['column_left'] = $this->load->controller('common/column_left');
$data['column_right'] = $this->load->controller('common/column_right');
$data['content_top'] = $this->load->controller('common/content_top');
$data['content_bottom'] = $this->load->controller('common/content_bottom');
$data['footer'] = $this->load->controller('common/footer');
$data['header'] = $this->load->controller('common/header');

$this->response->redirect($this->url->link('account/simpleedit', '', 'SSL'));

if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') .
'/template/account/account.tpl')) {
    $this->response->setOutput($this->load->view($this->config-
>get('config_template') . '/template/account/account.tpl', $data));
} else {
    $this->response->setOutput($this->load-
>view('default/template/account/account.tpl', $data));
}

```

```

    }
}

```

```

public function userHistoryMeeting ( $data ) {
    $listHistory = array();

    foreach ($data as $item) {
        $listHistory[] = array(
            'id'          => $item['id'],
            'topic'       => $item['topic'],
            'name'        => $item['user_name'],
            'start_time'  => $item['start_time'],
            'end_time'    => $item['end_time'],
            'total_minutes' => $item['total_minutes']
        );
    }

    $this->model_account_zoom->getHistory($listHistory);

    $this->response->addHeader('Content-Type: application/json');
    $this->response->setOutput(true);
}

```

```

public function generate_signature ( $api_key, $api_sercet, $meeting_number, $role){

```

```

    $time = time() * 1000 - 30000;

```

```

    $data = base64_encode($api_key . $meeting_number . $time . $role);

```

```

    $hash = hash_hmac('sha256', $data, $api_sercet, true);

```

```

    $_sig = $api_key . "." . $meeting_number . "." . $time . "." . $role . "." .
base64_encode($hash);

```

```

    return rtrim(strtr(base64_encode($_sig), '+/', '-_'), '=');
}

```

```

}

```

```

<?php

```

```

class ControllerAccountAccount extends Controller {

```

```

    public function index() {

```

```

        if (!$this->customer->isLoggedIn()) {

```

```

        $this->session->data['redirect'] = $this->url->link('account/account',
'', 'SSL');

        $this->response->redirect($this->url->link('account/login', '', 'SSL'));
    }

    $this->load->language('account/account');

    $this->document->setTitle($this->language->get('heading_title'));

    $data['breadcrumbs'] = array();

    $data['breadcrumbs'][] = array(
        'text' => $this->language->get('text_home'),
        'href' => $this->url->link('common/home')
    );

    $data['breadcrumbs'][] = array(
        'text' => $this->language->get('text_account'),
        'href' => $this->url->link('account/account', '', 'SSL')
    );

    if (isset($this->session->data['success'])) {
        $data['success'] = $this->session->data['success'];

        unset($this->session->data['success']);
    } else {
        $data['success'] = '';
    }

    $data['heading_title'] = $this->language->get('heading_title');

    $data['text_my_account'] = $this->language->get('text_my_account');
    $data['text_my_orders'] = $this->language->get('text_my_orders');
    $data['text_my_newsletter'] = $this->language->get('text_my_newsletter');
    $data['text_edit'] = $this->language->get('text_edit');
    $data['text_password'] = $this->language->get('text_password');
    $data['text_address'] = $this->language->get('text_address');
    $data['text_wishlist'] = $this->language->get('text_wishlist');
    $data['text_order'] = $this->language->get('text_order');

```

```

$data['text_download'] = $this->language->get('text_download');
$data['text_reward'] = $this->language->get('text_reward');
$data['text_return'] = $this->language->get('text_return');
$data['text_transaction'] = $this->language->get('text_transaction');
$data['text_newsletter'] = $this->language->get('text_newsletter');
$data['text_recurring'] = $this->language->get('text_recurring');

$data['edit'] = $this->url->link('account/edit', '', 'SSL');
$data['password'] = $this->url->link('account/password', '', 'SSL');
$data['address'] = $this->url->link('account/address', '', 'SSL');
$data['wishlist'] = $this->url->link('account/wishlist');
$data['order'] = $this->url->link('account/order', '', 'SSL');
$data['download'] = $this->url->link('account/download', '', 'SSL');
$data['return'] = $this->url->link('account/return', '', 'SSL');
$data['transaction'] = $this->url->link('account/transaction', '', 'SSL');
$data['newsletter'] = $this->url->link('account/newsletter', '', 'SSL');
$data['recurring'] = $this->url->link('account/recurring', '', 'SSL');

if ($this->config->get('reward_status')) {
    $data['reward'] = $this->url->link('account/reward', '', 'SSL');
} else {
    $data['reward'] = '';
}

$data['column_left'] = $this->load->controller('common/column_left');
$data['column_right'] = $this->load->controller('common/column_right');
$data['content_top'] = $this->load->controller('common/content_top');
$data['content_bottom'] = $this->load->controller('common/content_bottom');
$data['footer'] = $this->load->controller('common/footer');
$data['header'] = $this->load->controller('common/header');

$this->response->redirect($this->url->link('account/simpleedit', '', 'SSL'));

if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') .
'/template/account/account.tpl')) {
    $this->response->setOutput($this->load->view($this->config-
>get('config_template') . '/template/account/account.tpl', $data));
} else {
    $this->response->setOutput($this->load-
>view('default/template/account/account.tpl', $data));
}

```

```

    }
}

public function country() {
    $json = array();

    $this->load->model('localisation/country');

    $country_info = $this->model_localisation_country->getCountry($this->request->get['country_id']);

    if ($country_info) {
        $this->load->model('localisation/zone');

        $json = array(
            'country_id'      => $country_info['country_id'],
            'name'            => $country_info['name'],
            'iso_code_2'     => $country_info['iso_code_2'],
            'iso_code_3'     => $country_info['iso_code_3'],
            'address_format' => $country_info['address_format'],
            'postcode_required' => $country_info['postcode_required'],
            'zone'           => $this->model_localisation_zone->getZonesByCountryId($this->request->get['country_id']),
            'status'         => $country_info['status']
        );
    }

    $this->response->addHeader('Content-Type: application/json');
    $this->response->setOutput(json_encode($json));
}
}

```

Додаток Б
Ксерокопії наукових публікацій, виконаних при роботі над
дипломною роботою магістра

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XII всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2020»

9-10 листопада 2020

Хмельницький 2020

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК - 2020*XII Всеукраїнська науково-практична конференція*

Метою конференції є висвітлення актуальних проблем комп'ютерних наук, інформатики та інформаційних технологій.

СЕКЦІЇ КОНФЕРЕНЦІЇ:

1. Комп'ютерні науки та прикладні інформаційні технології.
2. Комп'ютерна інженерія та системи захисту інформації.
3. Математичне моделювання та інженерія програмного забезпечення
4. Телерадіокомунікації, медійні та комунікаційні системи.
5. Проблеми впровадження інформаційних технологій у виробництво та управління.

Робочі мови конференції: українська, англійська

ОРГКОМІТЕТ:

СИНЮК О. М. голова оргкомітету, проректор Хмельницького національного університету з наукової роботи, доктор технічних наук, професор
СОРОКАТИЙ Р. В. заступник голови оргкомітету, завідувач кафедри Комп'ютерних наук та інформаційних технологій ХНУ, доктор технічних наук, професор

БАРМАК О. В. заступник голови оргкомітету, доктор технічних наук, професор

САВЕНКО О. С. декан Факультету програмування та комп'ютерних і телекомунікаційних систем ХНУ, доктор технічних наук, професор

ВИСОЦЬКА О. В. доктор технічних наук, завідувач кафедри радіоелектронних та біомедичних комп'ютеризованих засобів і технологій Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», професор

ЛАВРОВ Є. А. доктор технічних наук, професор (Сумський державний університет)

ТИМОФЄЄВА Л. В. відповідальна за студентську науково-дослідну роботу ХНУ

МАЗУРЕЦЬ О. В. секретар конференції, старший викладач кафедри комп'ютерних наук та інформаційних технологій ХНУ

КОНТАКТНА ІНФОРМАЦІЯ:

e-mail для листування: apkt.khnu@gmail.com

Сова О. Я., Дука О. В., Назаренко І. М. Методи автоматизованого розгортання та налаштування мережевої та серверної інфраструктури з контролем версій	278
Ставінська І. В., Григорова А. А. Віртуальні асистенти в сфері HR-менеджменту	281
Старанчук З. І., Табеньський С. М. Багатокомп'ютерна система виявлення комп'ютерних атак на основі штучних імунних систем та нейронних мереж	285
Стецюк М. В., Стецюк В. М., Савенко О. С. Модель архітектури автоматизованих інформаційних систем супроводу фінансово-господарських процесів у корпоративних мережах в умовах впливу зловмисних дій	288
Табунов А. А., Шевченко В. Л. Програмне забезпечення для визначення координат за допомогою сенсорів смартфона без використання GPS	292
Тимоцюк С. В., Пономаренко Р. М. Дослідження та розробка програмного забезпечення підтримки освітнього процесу у вищих навчальних закладах	295
Тіторов І. Д., Скрипник Т. К. Аналітична система рекомендацій закладів харчування на основі відгуків та рейтингу	300
Ткачук Є. А., Багрій Р. О., Скрипник Т. К. Методи оптимізації доставки замовлень	303
Ткачук О. С., Багрій Р. О., Скрипник Т. К. Інформаційна система онлайн-комунікації для дистанційного навчання	307
Тригуб І. Є., Гайчук С. В. Особливості розробки корпоративного порталу для міжнародного туроператора на базі CRM-системи	311
Тузенко О. О., Кулішова К. О. Інформаційна система оцінки екологічної стійкості транспортних систем	316
Федорова А. В., Ніколаско В. В., Лавров Є. А. Метод побудови адаптивної інформаційної системи	320

УДК 004.77

Ткачук О. С., Багрій Р. О., Скрипник Т. К.

*Хмельницький національний університет***ІНФОРМАЦІЙНА СИСТЕМА ОНЛАЙН-КОМУНІКАЦІЇ ДЛЯ
ДИСТАНЦІЙНОГО НАВЧАННЯ**

В роботі розглянуто онлайн-комунікацію між студентом та викладачем на основі існуючих видів комунікації, таких як чати, форуми, відео конференції та електронна пошта. Проведено дослідження видів комунікації та різновиди їх систем. На основі чого запропоновано інформаційну систему для вирішення проблем дистанційного навчання та покращення комунікації між студентами та викладачем, а також здійснено програму реалізації системи на основі існуючих систем.

The robots are viewed by an online commentator and tutor based on informational messages such as chat, forums, video conferencing and email. A study of the types of communications and varieties of their systems. On the basis of which the system of informing on problems of remote control and improvement of communication by students and the teacher is offered, and also program realization of system on the basis of existing systems is carried out.

В наш час невід'ємною частиною життя людини є соціальні мережі, різні месенджери і тд. Інтернет-комунікації стали не від'ємним інструментом у нашому житті як для простого спілкування так і для роботи, розваг, навчання і тд. На даний момент, в період пандемії, не від'ємним став перехід на дистанційне навчання та віддалену роботу. Людство зіштовхнулось з проблемами комунікації між собою, особливо це проявилось у навчанні.

Проблема онлайн-комунікації між студентом та викладачем полягає в тому що, на даний момент немає однієї зручної системи в якій є все необхідне для як для спілкування так і для навчання. Проведення дистанційного навчання передбачає виконання вимог, а саме: проведення онлайн пар, відслідковування присутності студента, активна взаємодія з ним, а також швидкий та зручний зворотній зв'язок з викладачем.

Отже, метою роботи є розробка інформаційної системи, яка б дала можливість покращити взаємодії між студентами та викладачем під час дистанційного навчання. Для цього необхідно дослідити та адаптувати існуючу систему онлайн-комунікації. На основі даної інформаційної системи необхідно створити або удосконалити додаток для онлайн-комунікації.

В рамках проведених досліджень проаналізовано існуючі способи онлайн-комунікації:

1. Електронна пошта за складом елементів та принципу роботи практично повторює систему звичайної (паперової) пошти, запозичуючи як терміни, так і характерні особливості – простоту використання, затримки передачі повідомлень, достатню надійність і, в той же час, відсутність гарантії доставки.
2. Форум – найпоширеніша форма спілкування вчителя/ вчительки та учнів у дистанційному навчанні.
3. Чат – мережевий засіб для швидкого обміну текстовими повідомленнями між користувачами інтернету в системі реального часу. Зазвичай, під словом «чат» мається на увазі інтернет-ресурс з можливостями чату, чат-програма, рідше – сам процес обміну текстовими повідомленнями.
4. Відеоконференція – це конференція в режимі реального часу онлайн. Вона проводиться у визначений день і час

Отже на основі аналізу способів комунікацій та сформульованих вимог до дистанційного навчання можна зробити висновок, що інформаційна система має бути основана на платформі онлайн конференцій. Реалізація інформаційної системи у вигляді додатку має бути безкоштовною, простою і водночас зручною у користуванні.

Враховуючи зазначені вище вимоги, візьмемо за основу інформаційної системи, систему онлайн-комунікації Zoom. Дана система має ряд переваг, а саме: можливість виконувати основні функції системи онлайн-комунікації, доступний програмний інтерфейс (API) (рис.1).

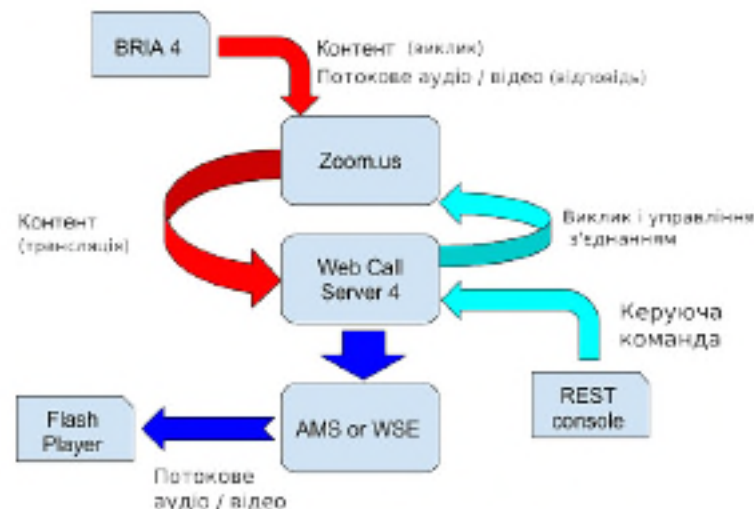


Рисунок 1 – Схема роботи систему онлайн-комунікації Zoom

Робота інформаційної системи проходить у прості послідовності, спочатку відбувається виклик від користувача який заходить в онлайн конференцію, дані якого аудіо/відео відправляються Zoom.us який далі передає їх в трансляцію після чого ці дані передаються для усіх користувачів конференції в реальному часі.

Також у керуючого конференції є команди для керування конференцією, які також відпрацьовують в реальному часі таким чином він може відключати аудіо/відео або ж видалити з конференції.

Однією з основних частин інформаційної системи є чат який також працює у реальному часі рисунку 2.

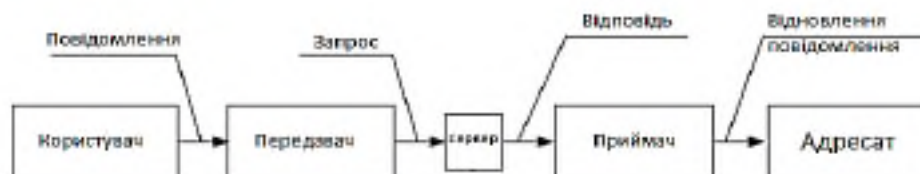


Рисунок 2 – Схема роботи чату

При підключенні до конференції у користувача є можливість перевірити своє обладнання аудіо/відео а також керувати ними.

Виходячи з аналізу схеми роботи чату та інформаційної системи стає зрозуміло, що вона виконує свою ціль та включає в себе всі поставлені умови.

Для створення інформаційної системи на основі Zoom використаємо API/SDK Zoom. Після встановлення додатку та створення облікового запису, система створює зашифрований електронний підпис, який необхідний для створення або приєднання до онлайн конференції [1].

Для створення підключення необхідні наступні параметри:

1. ApiKey - Ключ API вашого облікового запису;
2. ApiSecret - API Секрет вашого облікового запису;
3. meetingNumber - номер наради, до якого приєднується користувач
4. role - 1 для ведучого зустрічі, 0 для учасників.

На основі цих параметрів сервіс Zoom створює підпис у форматі JSON, який використовується для приєднання до конференції.

Для створення журналу відвідування розроблено метод, що відслідковує учасників конференції через API Zoom з детальною інформацією про час початку та кінця сеансу.

Також активність студента визначається за допомогою модуля сповіщень, що викликається за допомогою натискання кнопки «Підняти руку».

Для інтеграції інформаційної системи в модульне середовище університету використано Zoom API для зручної взаємодії студента та викладача, за допомогою яких надана можливість створювати приватну конференцію для декількох осіб.

Таким чином, результатом є створення гнучкої інформаційної системи, яка дасть змогу вбудувати її в інформаційні системи навчальних закладів, що дозволить вирішити основні проблеми дистанційного навчання, такі як проведення онлайн занять в режимі відеоконференції з відслідковуванням присутності студентів, активної взаємодії з ними, а також швидким та зручним зворотнім зв'язком.

Отже, на основі проведеного дослідження видів комунікацій було створено інформаційну систему, що дозволяє опрацьовувати метадані з SDK/API Zoom. Це дало можливість застосувати отриману інформацію для вирішення проблем дистанційного навчання та покращення комунікації між студентами та викладачем. Розроблена інформаційна система також має можливості розширення функціоналу, такого як, інтеграція з різними соціальними месенджерами для більш ефективного оповіщення про різні заходи.

Перелік посилань

1. Zoom SDK [Електронний ресурс]. – Режим доступу: <https://marketplace.zoom.us/docs/sdk/native-sdks/web/build/signature>
2. Мережеве спілкування [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мережеве_спілкування
3. Zoom API [Електронний ресурс]. – Режим доступу: <https://developers.google.com/admin-sdk/reports/v1/appendix/usage/customer/meet>

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилوک в документах: 16%

ID: 82348 Назва: Інформаційна система для соціальної мережі університету Додано в БД: 2020-12-03 Автора: Ткачук Олександр Сергійович Керівники: Багрий Р.О. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	71841	567	7617 (11%)	61 (11%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
82348			

82348

Автора: Ткачук Олександр Сергійович

Керівники: Багрий Р.О.

Консультанти:

Опоненти:

РІШЕННЯ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система для соціальної мережі університету

Автор: Ткачук О.С.

Спеціальність: 122 Комп'ютерні науки

Науковий керівник: к.т.н., доц. Багрій Р.О.

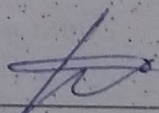
Після аналізу звіту подібності зроблено такий висновок:

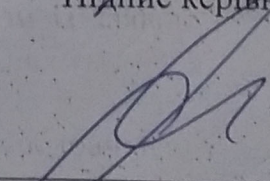
№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	-
3	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-
4	Інше:	-

Підтвердження: Виявлені запозичення не є плагіатом т.я. розміщені в розділах, які не описують безпосередньо авторське дослідження, складають 19% та мають посилання на приведені список літературних джерел.

01.11.2020

Дата


Підпис керівника


Підпис завідувача кафедри

ВІДГУК ОПОНЕНТА
на дипломну роботу магістра

Магістра гр. КНМ-19-1 Ткачука Олександра Сергійовича

На тему: Інформаційна система для соціальної мережі університету.

1. Актуальність і значення теми.

Завдяки швидкому розвитку інформаційних технологій зокрема інтернет ресурсів, частиною життя стало невід'ємним перебування у соціальних мережах, форумах, месенджерах. Так в період пандемії на перший план виходять онлайн-комунікації зокрема відео конференції які використовуються для налагодження комунікацій в різних сферах життя зокрема в навчанні. Таким чином з'явилась проблема у відсутності однієї загальної системи для онлайн-комунікацій студентів з викладачем з можливістю проведення дистанційного навчання. Інформаційна система для соціальної мережі університету дозволяє проводити онлайн заняття, слідкувати та формувати списки відвідування студентів та надає активну взаємодію між студентами та викладачем.

2. Оцінка якості та достовірності проведених досліджень.

Отримані результати добре співвідносяться з результатами, наведеними в наукових роботах. Проведені в роботі дослідження підтвердили достовірність одержаних результатів.

3. Оцінка запропонованих заходів та пропозицій, практичної цінності та ефективності.

Проведені дослідження представляють науково-технічну цінність, є ефективним дослідженням в галузі інформаційних систем, їх можна використати з метою інтеграції дистанційного навчання в соціальну мережу університету.

4. Загальний висновок та оцінка

Робота виконана в повному обсязі. Пояснювальна записка оформлена в відповідності з нормами. Відмічені недоліки не знижують цінності дипломної роботи. За своєю структурою, практичними цінностями, поставленій меті та вирішеними задачами робота відповідає вимогам вищої школи і вимогам, що пред'являються до освітньо-кваліфікаційного рівня «магістр», а її автор О. С. Ткачук заслуговує присвоєння кваліфікації магістра з комп'ютерних наук.

Робота заслуговує на оцінку « добре ».

Опонент Мазурчук Н.П., ктн, доцент кафедри
касієвської інформатико-математичної та
аграрно-економічної