ЗМІСТ

[ПЕРЕЛІК СКОРОЧЕНЬ 3](#_Toc514716134)

[ВСТУП 4](#_Toc514716135)

[РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТУ ТА ОСНОВНИХ ПОЛОЖЕНЬ 6](#_Toc514716136)

[РОЗДІЛ 2. АНАЛІЗ ІСНУЮЧИХ КОМЕРЦІЙНИХ ПРОГРАМНИХ РІШЕНЬ 9](#_Toc514716137)

[2.1 Аналіз програмного рішення «Skype» 9](#_Toc514716138)

[2.2 Аналіз програмного рішення «Microsoft Teams» 10](#_Toc514716139)

[2.3 Аналіз програмного рішення «Slack» 11](#_Toc514716140)

[2.4 Аналіз програмного рішення «Telegram» 11](#_Toc514716141)

[2.5 Аналіз програмного рішення «WhatsApp» 12](#_Toc514716142)

[2.6 Аналіз програмного рішення «Facebook Messenger» 13](#_Toc514716143)

[Висновки за розділом 16](#_Toc514716144)

[РОЗДІЛ 3. ФОРМУВАННЯ ВИМОГ ДО МАЙБУТНЬОЇ СИСТЕМИ 17](#_Toc514716145)

[3.1 Вибір архітектури та платформи розробки 17](#_Toc514716146)

[3.1.1 Клієнт-серверна архітектура 18](#_Toc514716147)

[3.1.2 Мова програмування C# та платформа Microsoft .NET Core 19](#_Toc514716148)

[3.1.3 СУБД Microsoft SQL Server 2017 20](#_Toc514716149)

[3.1.4 Бібліотека ASP.NET Core SignalR 22](#_Toc514716150)

[3.1.5 Фреймворк для створення веб-застосунків ASP.NET Core MVC 23](#_Toc514716151)

[3.1.6 Технологія Entity Framework Core 24](#_Toc514716152)

[3.1.7 Мова програмування JavaScript 25](#_Toc514716153)

[3.1.8 Мова розмітики HTML5 26](#_Toc514716154)

[3.1.9 Каскадні таблиці стилів CSS3 27](#_Toc514716155)

[3.1.10 Текстовий формат обміну даними JSON 28](#_Toc514716156)

[Висновки за розділом 28](#_Toc514716157)

[РОЗДІЛ 4. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ 28](#_Toc514716158)

[4.1 Проектування структури системи 28](#_Toc514716159)

[4.2 Проектування користувальницької частини системи 29](#_Toc514716160)

[4.3 Проектування моделі бази даних 29](#_Toc514716161)

[Висновки за розділом 29](#_Toc514716162)

[РОЗДІЛ 5. РЕАЛІЗАЦІЯ СПРОЕКТОВАНИХ ПІДСИСТЕМ 29](#_Toc514716163)

[5.1 Реалізація моделі бази даних 29](#_Toc514716164)

[5.2 Реалізація серверної частини системи 30](#_Toc514716165)

[5.3 Реалізація клієнтської частини системи 30](#_Toc514716166)

[Висновки за розділом 30](#_Toc514716167)

[РОЗДІЛ 6. ПЕРЕВІРКА РОБОТИ ТА ТЕСТУВАННЯ СИСТЕМИ 30](#_Toc514716168)

[6.1 Ручне тестування 30](#_Toc514716169)

[6.2 Модульне тестування 31](#_Toc514716170)

[Висновки за розділом 31](#_Toc514716171)

[РОЗДІЛ 7. КЕРІВНИЦТВО КОРИСТУВАЧА 31](#_Toc514716172)

[7.1 Керівництво для незареєстрованого оператора 31](#_Toc514716173)

[7.2 Керівництво для зареєстрованого оператора 32](#_Toc514716174)

[Висновки за розділом 32](#_Toc514716175)

[РОЗДІЛ 8. ОХОРОНА ПРАЦІ 32](#_Toc514716176)

[8.1 Вступ 32](#_Toc514716177)

[8.2 Характеристика робочого приміщення 33](#_Toc514716178)

[8.3 Аналіз шкідливих та небезпечних виробничих факторів 33](#_Toc514716179)

[8.3.1 Мікроклімат виробничих приміщень 33](#_Toc514716180)

[8.3.2 Освітлення 33](#_Toc514716181)

[8.3.3 Характеристика виробничого шуму 33](#_Toc514716182)

[8.3.4 Характеристика випромінювання 33](#_Toc514716183)

[8.3.5 Електробезпека 33](#_Toc514716184)

[8.3.6 Пожежна безпека 34](#_Toc514716185)

[8.4 Інструкція з техніки безпеки 34](#_Toc514716186)

[Висновки за розділом 34](#_Toc514716187)

[ВИСНОВКИ 35](#_Toc514716188)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 36](#_Toc514716189)

[Додаток А 38](#_Toc514716190)

[Додаток Б 39](#_Toc514716191)

ПЕРЕЛІК СКОРОЧЕНЬ

БД — База даних

СУБД – Система управління базами даних

ПЗ — Програмне забезпечення

HTTP — HyperText Transfer Protocol

HTML — HyperText Markup Language

URL — Uniform Resource Locator

JSON — JavaScript Object Notation

XML — Extensible Markup Language

VoIP — Voice over IP

IP — Internet Protocol

P2P — peer-to-peer

SMS — Short Message Service

ВСТУП

На сьогодні важко уявити своє життя без листування зі своїми друзями чи близькими. Не можливо організувати роботу як великих так і малих компаній без можливості миттєвого обміну інформацією та файлами. Чати стали невід’ємною частиною нашої повсякденної зайнятості.

Існує багато аналогів таких систем з неймовірною кількістю функціональності та мільйонами користувачів. Здається, ніби вони покривають всі наші потреби в обміні повідомленнями по всьому світу. Але варто детальніше проаналізувати роботу підприємств та кожного працівника окремо. Не завжди універсальні додатки максимально зручні та корисні.

Уявімо менеджера чи керівника відділу, який постійно веде корпоративну переписку зі своїми начальниками та колегами. Виникають ситуації, коли потрібно підготувати звіти по роботі чи інші важливі матеріали та надіслати колегам в фіксований час. Тому було б корисно мати додаток, в якому ми могли б вибрати дату та час надсилання повідомлення, і запланувати відправку, навіть без підключення до інтернету, тобто месенджер онлайн та частково офлайн обміну даними. І хоча швидкість інтернету з використанням сучасних технологій досягає високих показників, іноді відбуваються збої в наданні даної послуги провайдерами. Запланувавши відправлення файлів чи інформації, користувач буде впевнений в тому, що інша особа отримає необхідні дані вчасно. Також, однією з особливостей програми може стати легка адаптація під потреби та вимоги кожного підприємства, а саме можливість налаштування роботи в локальній мережі компанії без доступу до інтернету та розгортання багатьох екземплярів додатку з наступною синхронізацією з головною системою.

Метою роботи є створення веб-додатку реального часу для покращення корпоративної комунікації з можливостями адаптації під потреби кожної компанії.

Для реалізації даної ідеї потрібно вирішити кілька завдань, серед них:

* написання функціональної частини застосування;
* розробка веб-інтерфейсу та мобільного інтерфейсу для користувачів;
* проектування структури бази даних;
* інтеграція з зовнішніми системами;

Проаналізувавши засоби розробки застосувань я обрав мову програмування C#, фреймворк для створення веб-застосунків ASP.NET Core та бібліотеку для створення застосунків реального часу ASP.NET Core SignalR як основні технології розробки.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТУ ТА ОСНОВНИХ ПОЛОЖЕНЬ

В цьому дипломному проекті виконується створення спеціалізованого програмного забезпечення для вирішення таких важливих задач кожної компанії як комунікація між усіма її працівниками.

Комунікація – це дуже складний процес, в який входять спілкування людей в групах, листування, надсилання звітів чи будь-які інші події обміну певними даними. До неї можна віднести обмін інформацією між особами за допомогою різних технічних засобів. За результатами багатьох досліджень виявлено, що керівники зазвичай витрачають більшу частину свого робочого часу на комунікацію. Адже їхніми головними задачами є організація, планування як внутрішніх так і зовнішніх бізнес-процесів та мотивація працівників для досягнення спільних цілей.

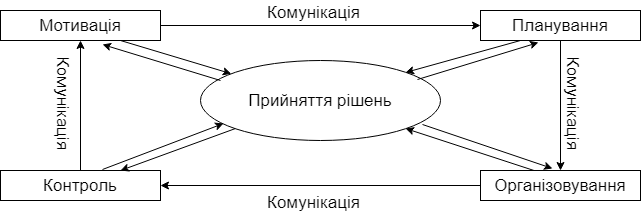


Рисунок 1.1 — Процес управління з використанням комунікації

Щоб вчасно та ефективно здійснювати управління, необхідно надсилати повідомлення у вигляді вказівок, звітів, наказів та іншого. Це може здійснюватися різним и способами, зокрема через електронну пошту, месенджери, різного роду загальнодоступні сховища даних. Якщо інформація вчасно передана іншій особі чи групі людей, вона можна бути цінною для обох сторін. Дані можуть передаватися не лише у вигляді тексту, а також і у вигляді файлів, картинок, архівів.

Проаналізувавши процеси, якііі відбуваються, можна виокремити чотири головних сутності:

1. Відправник – той, хто формує та передає інформацію.
2. Одержувач – той, кому направлено відсиланняданих.
3. Канал зв’язку – спосіб чи програмне забезпечення для передачі інформації.
4. Повідомлення – дані у будь-якому вигляді, підготовлені для надсилання.

Важливим моментом є те, що передача може бути перервана або в процесі передавання може відбутися пошкодження даних у разі використання сучасних технічних засобів.

Основними засобами комунікації в сучасних компаніях є електронні, які можуть представляти собою різного роду системи та програмне забезпечення для персональних комп’ютерів та смартфонів. Найбільш популярними системами є:

1. Адміністративні системи – використовуються менеджерами для оброблення інформації.
2. Системи інформаційного менеджменту – збирають та систематизують дані, відображають у зручному вигляді.
3. Електронні пошти
4. Месенджери та чати

Останні зайняли головну роль в сучасному житті кожної компанії, так як дають змогу швидко та з будь-яких пристроїв обмінюватися інформацією. Ці системи довзоляють миттєво відправляти повідомлення та переглядати їх в реальному часі.

Але існує декілька проблем, які можуть виникати у процесі листування. По-перше, потрібно постійний доступ до інтернету. У разі перебування в недоступній зоні або без технічних засобів зв’язку цього зробити не можна. Більше того, існує багато різноманітних систем з великим та корисним функціоналом, які пристосовані до сучаних вимог, але вирішити всі проблеми корпоративної комунікації вони не можуть.

Одним з найпоширеніших типів комунікації в середині компаніїї є відносини між керівником та підлеглим. Саме від цього залежить успіх як окремого відділу так і всього підприємства в цілому. Головними завданнями

керівника є поставлення задач до виконання, контроль виконнаня, допомога у вирішенні проблем та перевірка результатів. Підлеглий повинен швидко виконувати завдання, повідомляти керівника та підготовлювати звіти для підтвердження виконаної роботи.

Рішення проблем корпоративної комунікації, яка має велике значення в загальній системі роботи компаній будуть детально розглянуті в даному дипломному проекті.

РОЗДІЛ 2. АНАЛІЗ ІСНУЮЧИХ КОМЕРЦІЙНИХ ПРОГРАМНИХ РІШЕНЬ

Для того, щоб сформувати уявлення про систему, розглянемо вже існуючі комерційні програмні рішення.

* 1. Аналіз програмного рішення «Skype»

Skype – це ПЗ, в основному призначене для можливості робити телефонні та відеовиклики з використанням інтернету [1]. Продукт дає можливість зв’язуватися з абонентами мобільних телефонів та миттєвий обмін повідомленнями між користувачами системи. Він має свої важливі особливості:

а) Кросплатформенний додаток.

б) Довзоляє створення чату до 300 осіб.

в) Підтримує відеоконференції.

г) Є клієнтом інтернет-телефонії VoIP.

ґ) Синхронізація повідомлень після підключення до інтернету.

д) Безкоштовне ПЗ.

е) Дані передаються в зашифрованому вигляді.

є) Широкі можливості налаштування.

Переваги системи:

а) Система є кросплатформенною, що довзоляє її використання майже на будь-якій платформі.

б) Безкоштовна комунікація між користувачами додатку.

в) Можливість зв’язуватися з абонентами мобільних телефонів.

г) Весь трафік, який надсилання програмою, кодується без можливості це виключити.

д) Простий у налаштуванні та використанні.

е) Надсилання файлів будь-яких розмірів з можливістю відновити надсилання при проблемах з доступом до інтернету.

Додаток має велику кількість користувачів по всьому світу. P2P-архітектура

дозволяє системі легко масштабуватися. Наявний функціонал чудово підходить як для корпоративної комунікації, так і для використання в домашніх умовах.

Недоліки системи:

а) Закритий код.

б) Немає можливості перевірки безпечності з’єднань експертами.

в) Є прогалини в безпеці.

г) Потребує багато ресурсів при роботі на мобільних пристроях.

## 2.2 Аналіз програмного рішення «Microsoft Teams»

Ще одним цікавим та багатофункціональним додатком компанії Майкрософт є Microsoft Teams [2].

Сервіс надає наступні можливості:

а) Об’єднує в собі чат, зустрічі та замітки.

б) Можливість створення для груп для роботи над одним проектом.

в) Завантаження будь-яких документів у групи для спільної роботи над ними.

г) Додавання коментарів в самих документах під час редагування.

ґ) Створення мітингів та зустрічей із запрошенням необхідних користувачів.

д) Швидке приєднання до відеоконференцій.

Основні переваги системи:

а) Швидка інтеграція з іншими популярними системами, такі як JIRA, Facebook.

б) Інтуїтивно зрозумілий та простий інтерфейс з насиченим функціоналом.

в) Перегляд останніх файлів, з якими працював, в окремій вкладці.

г) Створення відеовикликів та конференцій.

д) Вбудована інтеграція зі Skype.

Основні недоліки:

а) Входить до пакету ПЗ Office 365, який є платним для використання.

б) Не дозволяє зареєструвати користувача без корпоративної підписки.

в) Підходить для використання лише на середніх та великих проектах.

## 2.3 Аналіз програмного рішення «Slack»

Наступним аналогічним програмним сервісом розглянемо систему, яка набула швидкої популярності, «Slack» [3].

Його особливостями є:

а) Власний хостинг.

б) Пошук інформації відразу серед усіх повідомлень.

в) Легко інтегрується з близько 100 сервісами та додатками.

г) Приватні групи.

д) Кросплатформенний додаток.

Переваги сервісу:

а) Має свою безкоштовну версію з досить широким функціоналом.

б) Можливість перегляду зображень в самому додатку.

в) Має великі перспективи розвитку, можлива поява нового функціоналу у вигляді відеодзвінків та іншого.

г) Платні тарифи є недорогими, але суттєво розширюють можливост.

Загалом додаток став відкриттям для світу, так як швидко набрав велику кількість шанувальників та відповідно користувачів по всьому світу, хоча спочатку розроблявся як гра. Внаслідок постійних та потужних інвестицій став додатком для бізнесу, який поступово зменшує важливість корпоративної електронної пошти та «незамінного додатку» Skype.

Недоліки:

а) Вразливості безпеки, через які одного разу був отриманий доступ до бази користувачів.

б) Платні тарифи все ж вносять певні обмеження для звичайних користувачів.

2.4 Аналіз програмного рішення «Telegram»

Telegram – програмне рішення, яке позиціонує себе перш за все як месенджер з високим рівнем безпеки та шифрування [4].

Система представляє наступні можливості:

а) Обмін текстовими повідомленнями та файлами різних форматів.

б) Можливість робити дзвінки іншим користувачам системи.

в) Авторизація за допомогою коду із SMS.

г) Використання складних алгоритмів шифрування під час авторизації та аутентифікації.

Переваги системи:

а) Створення секретних чатів для запобігання перехоплення повідомлень.

б) Можливість встановлення на найпопулярніших платформах.

в) Здійснення голосових дзвінків.

г) Дійсно швидка робота додатку на мобільних пристроях навіть при дуже поганій якості інтернет з’єднання.

д) Запрошення користувачів у групи та чати за допомогою прямих посилань.

Перш за все додаток повністю безкоштовний для використання та має багато інших перебаг, зокрема можливість відправки повідомлень самому собі, тобто їх збереження, автоматичне збереження повідомлення, яке не було надіслано, видалення повідомлень не тільки зі сторони того, хто надсилає, а і зі сторони отримувача.

Недоліки системи:

а) Відсутність відеовикликів та конференцій.

б) Немає можливості планування мітингів та зустрічей.

в) Пристосований для повсякденного використання, але не для комунікації в середині компаній.

г) Невелика кількість функціоналу порівняно з іншими сервісами комунікації.

2.5 Аналіз програмного рішення «WhatsApp»

WhatsApp – сервіс, створений американськими програмістами, пізніше у 2014 році перекуплений компанією Facebook [5].

Система має наступні характеристики:

а) Безкоштовне встановлення та використання месенджера.

б) Пристосована для широкої лінійки платформ, в тому числі BlackBerry OS, Windows Phone, Series 40, iOS та інші.

в) Наявність можливості як відеодзвінків та і простого обміну повідомленнями.

Переваги сервісу:

а) щвидка побудова маршруту, час очікування збільшується при вказуванні 20 і більше точок пунктів

б) можливість кешування даних при втраті з’єднання

в) висока швидкість збору інформації та формування звітів

г) зручний та сучасний інтерфейс

Недоліки сервісу:

а) Достатня кількість багів та вразливостей, які знаходили звичайні користувачі системи.

б) Запуск нового функціоналу викликав підозри у його безперебійній та безпечній роботі на різних пристроях.

в) Відставання з випуском нових функцій від прямих конкурентів.

г) Можливі блокування зі сторони державних органів. Такий випадок виник у Бразилії, після чого декілька мільйонів користувачів зареєструвалося у додатку-конкуренті Telegram.

2.6 Аналіз програмного рішення «Facebook Messenger»

«Facebook Messenger» — система, призначена для миттєвого обміну повідомленнями та відео [6].

Дана система надає наступні можливості:

а) Вибір персональних тем та налаштувань.

б) Швидкий обмін повідомленнями та медіа файлами.

в) Надсилання звукових записів та сигналів.

г) Повністю безкоштовний застосунок.

ґ) Повідомлення користувачів про перегляд повідомлення отримувачем.

д) Доступність як на персональних комп’ютерах так і на мобільних платформах.

Переваги даного сервісу:

а) Не потрібно підтверджень реєстрації через надсилання кодів за допомогою SMS, що дозволяє використовувати додаток на планшетах та інших пристроях.

б) Підтримує велику кількість мов.

в) Досить простий у встановленні та використанні.

г) Має інтеграцію з телефонною книгою смартфону.

ґ) Доступна можливість надсилання відео.

е) Має галерею з усіма завантаженими файлами.

Недоліки системи:

а) Немає можливості групового голосового чату.

б) Надсилання музичних файлів неможливе.

в) Двостороннє шифрування повідомлень відсутнє.

Даний додаток не розрахований для корпоративної комунікації, так як не містить необхідного функціоналу та зручний у використанні для спілкування з друзями та близькими. Він має ряд своїх переваг та недоліків, дещо програє конкуренцію з іншими швидкими месенджерами, але пристосований під сучані вимоги і дозволяє зручно спілкуватися.

В результаті проведеного аналізу можна сформувати порівняльну таблицю 2.1. на якій представлені переваги та недоліки розглянутих продуктів.

Таблиця 2.1 – Порівняння розглянутих комерційних рішень

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерiй | Skype | Microsoft Teams | Slack | Telegram | WhatsApp | Facebook Messenger |
| **Кросплатформеність:** |  |  |  |  |  |  |
| Android | + | + | + | + | + | + |
| iOS | + | + | + | + | + | + |
| Windows | + | + | + | + | + | + |
| Web | + | + | + | + | + | + |
| Двосторонній голосовий чат | + | + | - | - | + | - |
| Не потрібна SIM-карта, щоб пройти авторизацію та аутентифікацію | + | + | + | - | - | + |
| Двостороннє шифрування даних | + | - | - | + | + | - |
| Можливість надсилати документи | + | + | + | + | + | - |
| Має функцію пошуку | + | + | + | + | + | - |
| Двосторонній відео чат | + | + | - | - | + | - |
| Сповіщення про перегляд повідомлення | + | - | - | + | + | + |
| Безкоштовний додаток | + | - | + | + | + | + |
| Функція «Залишатися невидимим» | + | - | + | - | + | + |
| Надсилання звукових голосових файлів | + | - | - | + | + | + |
| Налаштування доступності користувача | + | - | + | - | - | - |
| Надсилання музичних файлів | + | + | - | + | + | - |
| Двосторонній голосовий чат | + | + | - | - | + | - |
| Немає реклами | + | + | + | + | + | + |
| Видалення історії повідомлень | + | + | + | + | - | + |
| Підтримання віджетів | - | - | - | - | + | + |

Висновки за розділом

Отже, можна зробити висновок, що кожна система має свої переваги та недоліки у використанні для корпоративної комунікації. Основними умовами майбутнього проекту є вартість, надійність та швидкість надсилання повідомлень навіть за повільного з’єднання з інтернетом. В цьому випадку система буде повністю безкоштовною.

Також, планується підтримуватися тільки англомовний варіант графічного інтерфейсу, однак у ході розширення обов’язково буде додавання інших локалізацій та нової функціональності. Ще однією з переваг буде те, що додаток буде веб-застосунком і зможе працювати на різних пристроях. Але головною новинкою буде можливість планувати надсилання повідомлень та файлів заздалегідь. Таким чином дані будуть надіслані навіть у випадку, коли користувач не онлайн.

# РОЗДІЛ 3. ФОРМУВАННЯ ВИМОГ ДО МАЙБУТНЬОЇ СИСТЕМИ

Після ознайомлення з інформацією про готові рішення я вирішив створити систему, яка повинна задовольняти наступним вимогам:

а) Зберігання даних в одній базі даних для зменшення навантаження на складові системи.

б) Для забезпечення конфіденційності треба використати популярні алгоримти шифрування даних.

в) Інтуїтивно зрозумілий інтерфейс.

г) Можливість створення каналів, запрошення до них нових користувачів за допомогою надсилання запрошення на електронну пошту користувача.

ґ) Митєвий обмін повідомленнями, можливість завантаження файлів будь-яких форматів.

д) Планування відправлення повідомлень на певну дату та час.

е) Налаштування профілю користувача.

Тепер визначимо структуру додатку в деталях. Спочатку потрібно визначити архітектуру, згідно з якою буде проектуватися система.

Для даної системи я вибрав клієнт-серверну архітектуру.

Для побудови такої системи перерахуємо ряд вимог, які потрібно вирішити:

а) функціональність реєстрації нового користувача та авторизованого входу в систему;

б) створення каналів, запрошення користувачів до них;

в) надсилання повідомлень, прикріплення файлів;

г) визначення користувачів, які знаходяться онлайн та їх відображення;

ґ) обмін повідомленнями в реальному часі;

3.1 Вибір архітектури та платформи розробки

Для реалізації необхідного функціоналу була обрана клієнт-серверна архітектура додатку та наступні технології:

Платформа .NET Core, мова програмування C#, JavaScript та бібліотека JQuery, система управління базами даних Microsoft SQL Server 2017, бібліотека для розробки застосунків реального часу – ASP.NET Core SignalR, фреймворк для створення веб-застосунків ASP.NET Core 2.0, технологія доступу до даних Entity Framework Core, стандарт для розмітки веб сторінок HTML5 та каскадні таблиці стилів CSS3, текстовий формат обміну даними JSON.

3.1.1 Клієнт-серверна архітектура

При плануванні системи з великою кількістю користувачів найбільш доцільно використовувати архітектуру взаємодії типу «клієнт-сервер» [7]. При такому типі архітектури відбувається обмін даними в мережі між клієнтами та серверами, які надають їм певні сервіси.

Сервер – це комп’ютер або програма, який є незалежною складовою мережі та забезпечує доступ до загальних ресурсів .

Клієнт – це користувач або отримувач ресурсів та сервісів, які надає сервер.

На серверах в основному є мережева операційна система та апаратне забезпечення.

Головними завданнями сервера є управління трафіком, забезпечення централізованого захисту, надання клієнтам інформації, доступу до спільних пристроїв і файлів.

Клієнт-серверна архітектура є одним з наважливіших принципів при проектуванні та створенні розподілених додатків. Вона перш за все визначає рівні операцій та довзоляє розподілити обов’язки між клієнтом та сервером [8].

Процес взаємодії представлений на рисунку 3.1.

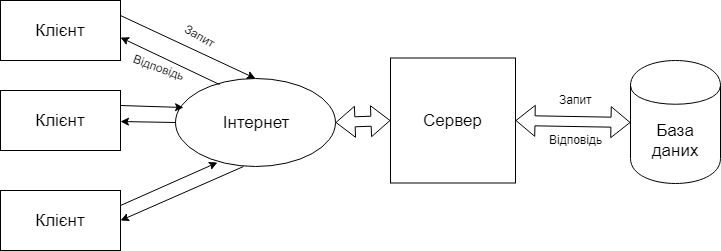


Рисунок 3.1 — Процес взаємодії між клієнтом та сервером

Виділяють три логічних рівні операцій:

1. Представлення даних – це інтерфейс користувача, який дає йому змогу виконувати операції та переглядати результати у вигляді відображення необхідних даних.
2. Прикладний рівень – обробка команд та всієї інформації, головна логіка додатків.
3. Управління даними – зберігання та доступ до даних.

Дволанкова архітектура досить проста у використанні, але не завжди є надійною та вимагає високої продуктивності сервера, так як він обслуговує всі запити клієнтів.

* + 1. Мова програмування C# та платформа Microsoft .NET Core

.NET Core – це фреймворк від корпорації Microsoft для створення різного виду застосунків, починаючи від звичайних веб-застосунків і закінчуючи мобільними платформами [9].

Його головними перевагами є те, що він безкоштовний та кросплатформний. Порівнюючи з .NET Framework, він оптимізований під сучасні потреби, весь код є відкритим та доступним для перегляду. Особливістю даної платформи є модульність, так як розробники можуть легко використовувати NuGet-пакети

замість збірок. Написані на підтримуваних платформою мовами програми

перетворюються в незалежний від платформи байт-код, який ще називають CIL (Common Intermediate Language) обробкою якого займається спеціальна віртуальна машина CoreCLR (Common Lirary Runtime), а потім в кінцевому рахунку перетворюється на машинний код, який виконується на цільовому процесорі. До таких мов відноситься велика кількість сучасних популярних мов програмування серед яких: Visual Basic, J#, C++/CLI та C#, яка буде використана для виконання поставлених в даній роботі задач.

.NET runtime надає головні сервіси: систему типів, збірку та збірник сміття. Платформа містить інтерфейс командного рядка CLI (Command Line Interface). Він є основою для інструментів вищих рівнів, а також є кросплатформенним, що довзоляє запускати його на будь-які підтримуваній платформі.

C# - це легка та потужна об’єктно-орієнтована мова, яка використовується в розробці комерційних рішень [10]. Простий та, в той же час, елегантний синтаксис, схожий з популярними та часто використовуваними мовами, висока швидкість обробки, захищеність та багато інших значних і важливих переваг. Постійні оновлення та оптимізація дозволяють швидко і якісно створювати програмне забезпечення, що є дуже важливим в умовах сучасного ринку.

* + 1. СУБД Microsoft SQL Server 2017

Microsoft SQL Server – новітня система управління базами даних, яка представляє засоби по створенню та управлінню реляційною моделлю базами даних [11]. Постійний розвиток та оновлення довзоляють використовувати чудову масштабованість і продуктивність.

До найбільш ефективних та визначних переваг даної системи відносять:

а) висока степінь безпеки та відповідність вимогам;

б) аналітика в режимі реального часу;

в) масштабованість;

г) продуктивність.

Реляційна модель баз даних зараз використовується в багатьох програмних

рішеннях, але вона має свої переваги та недоліки порівняно з іншими. За рахунок того, що головним об’єктом є «таблиця», користувачу легко працювати з нею. По-друге, після внесення змін у реляційну базу даних, прикладна програма потребує лише невеликих змін. Проектування ж здійснюється за певними правилами. Недоліками такої моделі є низька швидкість обробки запитів до доступу до даних порівняно з нереляційними моделями. При використанні у великих системах, схема таблиць розширюється і призводить до ускладнення розуміння структури даних.

Для приведення такої моделі до реляційної потрібно виконати процес, який називається нормалізацією бази даних. Вона полягає в приведенні структури бази даних до шести нормальних форм, які призначені для оптимізації виконання запитів до таблиць [13].

Розберемо три основні нормальні форми:

1. Відношення знаходиться в першій нормальній формі, якщо таблиця має ключ та кожне поле містить тільки одне значення, що відповідає прицнипі атомарності даних.
2. Відношення знаходиться в другій нормальній формі при виконанні умов першої нормальної форми та дані не повинні повторно з’являтися в декількох рядках. Інакше їх треба виносити в коремі таблиці.
3. Відношення знаходиться в третій нормальній формі, якщо виконані умови другої нормальної форми та будь-яке поле залежить лише від основного ключа. Якщо з’являються інші залежності, дані потрібно виносити в окремі таблиці.

Четверта, п’ята та шоста номальні форми використовуються досить рідко, тому що надмірна нормалізація в багатьох випадках може призвести до погіршення продуктивності виконання запитів. Через те іноді проводять зворотній процес, який називається денормалізацією.

Виділяють 3 типи зв’язків між об’єктами:

Один до одного – будь-якому об’єкту А може відповідати тільки один об’єкт Б, і навпаки.

Один до багатьох – будь-якому об’єкту А може відповідати кілька об’єктів Б, і, в той же час, будь якому об’єкту Б може відповідати тільки один об’єкт А.

Багато до багатьох – будь-якому об’єкту А може відповідати багато об’єктів типу Б, і, в той же час, будь якому об’єкту Б може відповідати кілька об’єктів типу А.

Для раціональної організації бази даних по реляційній моделі потрібно задовольняти наступні вимоги:

а) повинні задовольнятися вимоги третьої нормальної форми;

б) додавати нове відношення треба тільки тоді, коли в ньому є необхідність відповідно до вимог предметної області;

в) атрибут потрбіно додавати тільки в тоді, коли є впевненість в тому, що він буде використовуватися.

* + 1. Бібліотека ASP.NET Core SignalR

ASP.NET Core SignalR є популярною бібліотекою компанії Microsoft, яка створена для розробки застосунків реального часу. Вона дає змогу серверу працювати з багатьма клієнтами та одночасно відправляти їм дані [14]. Цей тип моделі називається «publisher-subscriber», адже клієнти фактично підписуються на хаб, який може відправляти всім оновлення у вигляді нових даних.

Бібліотека повністю написана з «нуля» та має суттєві переваги, порівняно з старішою версією [15]. Вона надійніша та простіша у використанні. Розробники приклали максимум зусиль, щоб інтерфейс взаємодії зробити максимально схожим до минулої версію. Це дає змогу без великих проблем оновлювати старі застосунки. Далі розглянемо головні переваги та оновлення бібліотеки:

1. Підтримка двійкових протоколів – надсилання повідомлень у двійковому форматі суттєво зменшує їх розміри і відповідно підвищує швидкість роботи додатків.
2. Можливість підключення користувацькихх реалізацій протоколів.
3. Тепер клієнти для сервера можна створювати як на JavaScript так і на

TypeScript.

1. Спрощено модель підключення клієнтів до сервера. Тепер, якщо клієнт втратив зв’язок з сервером, йому потрібно явно встановити нове підключення. Хоча в попередній версії відбувалися автоматичні спроби перепідключення, які вимагали зберігати на сервері 1000 останніх повідомлень для повторного надсилання клієнту.

Всі ці оновлення не залишають вагань у виборі бібліотеки. Хоча в ній наявні певні проблеми з технічою реалізацією, фактичних конкурентів з таким простим інтерфейсом використання немає.

* + 1. Фреймворк для створення веб-застосунків ASP.NET Core MVC

ASP.NET Core – це фреймворк та технологія, яка застосовується для створення різних типів веб-застосунків [16]. Вона має повністю відкритий код та може працювати поверх кросплатформного середовища .NET Core. Фреймворк повінстю побудований з незалежних компонентів, що дає їй змогу легко розширюватися. Відповідно можна використовувати або вбудовану реалізацію цих компонентів, або ж створити свою. Головними відмінностями ASP.NET Core від попередніх версій ASP.NET є:

а) Кросплатформеність.

б) Відкритий код, а значить доступність до щмін та покращень.

в) Використання платформи .NET Core з її функціональністю

г) Підтримка для легкого створення і використання пакетів NuGet.

Завдяки цим та іншим перевагам набуває популярності і використовується при розробці все більшої кількості програмних систем.

Фреймворк ASP.NET Core MVC працює поверх платформи ASP.NET Core, і призначений для того, щоб спростити створення веб-застосунків [17]. Концепція самого патерна полягає у використанні такого шаблону:

1. Модель – це клас, який може описувати логіку, яка зв’язана з даними, наприклад валідацію, а також, самі дані застосунку.
2. Представлення – це користувацький інтерфейс, через який користувач може взаємодіяти із застосунком. Цей рівень не повинен містити логіки керування даними.
3. Контролер – клас-посередник, який забезпечує зв’язок між представленням та сховищем даних. Він обробляє запит від користувача і відправляє необхідні дані.

На рисунку 3.2 представлена роль шаблону MVC в загальній архітектурі веб застосувань на основі фреймворку.

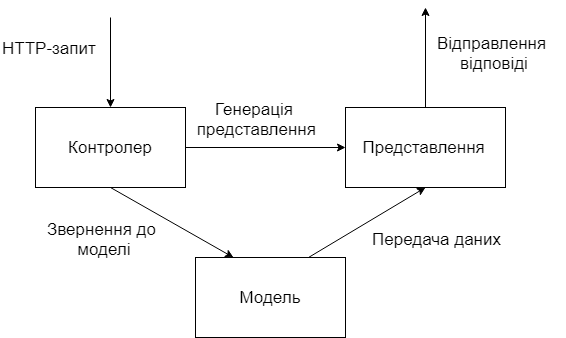


Рисунок 3.2 — Шаблон MVC в архітектурі веб застосувань

Переваги фреймворку ASP.NET Core MVC:

а) відкритий код;

б) система маршрутизації;

в) легкість створення застосунків;

г) тестованість;

ґ) розширюваність;

* + 1. Технологія Entity Framework Core

Entity Framework Core – це об’єктно-орієнтована технологія роботи з даними, яка представляє собою Object-relational mapping (ORM) бібліотеку [19]. Дана технологія дає більш високий рівень абстрагованості ніж інші бібліотеки, і це в свою чергу дозволяє розробнику абстрагуватися від бази даних та маніпулювати даними незалежно від типу бази даних.

Основною особливістю бібліотеки, та і всіх ORM бібліотек є те, що розробник вже не працює з таблицями, внутрішніми або зовнішніми ключами. Натомість робота виконується з об’єктами [20].

Основним елементом роботи в Entity Framework Core є поняття сутності, який представляє дані, які асоційовані з деяким об’єктом.

Є кілька типів взаємодії з базою даних за допомогою даної бібліотеки:

а) Code First – при реалізації даного типу спочатку створюється клас моделі, а потім бібліотека за допомогою цієї моделі повністю генерує базу даних, створює в ній таблиці та інші необхідні складові;

б) Database first – по створеній базі даних зі всім необхідним функціоналом бібліотека Entity Framework створює класи та звязки взаємодії між ними;

в) Model first – спочатку генерується модель бази даних, і потім бібліотека Entity Framework по створеній моделі створює базу даних, таблиці і звязки між ними.

Для даного проекту дану технологію я вибрав тому, що з її допомогою можна легко створювати сутності, вибирати, змінювати та видаляти їх з бази даних. Крім того вона має високу швидкість в порівнянні зі своїми аналогами.

* + 1. Мова програмування JavaScript

JavaScript – об’єктно-орієнтована мова програмування з нестрогою типізацією.

Найчастіше використовується як засіб, що надає можливість взаємодії коду на стороні веб переглядача з користувачем, керувати структурою веб сторінок та переглядачем користувача, виконувати асинхронний обмін даними з сервером.

Останнім часом мова набула шаленої популярності, тому її вже почали використовувати для розробки різних типів ігор, всередині різних типів інших застосувань, мобільних додатків та віконних застосувань, на серверній стороні тощо.

JavaScript підтримує об’єктно-орієнтовану, прототипну, імперативну та функціональну парадигми програмування та деякі особливі архітектурні властивості до яких відносять:

а) Слабка типізація;

б) Прототипне наслідування;

в) Автоматичне керування пам’яттю та інші.

В даному дипломному проекті дана мова була вибрана для виконання асинхронних запитів, роботи з клієнтом і також для роботи з мапами Google Maps, які в проекті використовуються на стороні оператора для перегляду працюючих на даний момент кур’єрів та можливості задання їм маршруту за допомогою маркерів.

* + 1. Мова розмітики HTML5

HTML5 (HyperText Markup Language) – це стандартизована мова для розмітики веб документів.

Дана розмітка використовується для інтерпретації її веб переглядачами клієнта і отриманий в результаті цього текст використовується для відображення на екрані монітора веб або мобільного пристрою.

Передача документів від сервера до клієнта відбувається як правило за допомогою протоколів HTTP або HTTPS у вигляді простого тексту та з використанням шифрування.

HTML-документ – це по суті звичайний текстовий документ, який можна створити в спеціалізованому або звичайному текстовому редакторі. Документ складається з дерева HTML-елементів, кожен з яких визначається відкриваючим та закриваючим тегом. Також в документі є обов’язкові два розділи, до яких

відносять: заголовок (head) – для опису допоміжної інформації, та змістовна частина сторінки або як ще кажуть тіло сторінки (body). Процес інтерпретації починається до повного завантаження документа в веб переглядач.

Оскільки в даному проекті операторська підсистема представляє собою веб застосування, то використання даного стандарту є доцільним оскільки альтернативи йому на сьогоднішній день немає.

* + 1. Каскадні таблиці стилів CSS3

CSS3 – це мова стилів, яка відображає та взаємодіє з HTML-документами і яка використовується в основному для опису зовнішнього вигляду документу. На сьогоднішній день без цього стандарту практично неможливо собі уявити сучасну веб розробку. Таблиця стилів складається з набору правил, які визначають, як буде відображатися той чи інший елемент на сторінці, тобто описує, які стилі будуть для нього використовуватися.

Переваги використання CSS3:

а) підвищення сумісності з різними платформами за рахунок використання веб стандартів;

б) один документ CSS3 може керувати великою кількістю HTML-сторінок, за рахунок чого зміни виконуються швидше;

в) файл з оформленням завантажується один раз і при цьому кешується, внаслідок чого навантаження на сервер, час завантаження та трафік зменшується;

г) код розділяється на вмістовну частину та оформлення, внаслідок чого отримуємо більш зрозумілий та легкий код;

ґ) є можливість під’єднувати різні CSS файли для різних типів пристроїв, які будуть відображати закодовану логіку відображення.

Недоліки:

а) відсутня можливість підлаштовування зовнішнього вигляду під різні типи відображення на пристроях (монітор, планшет, голосовий веб переглядач тощо.);

б) неможливість налаштування багатьох тегів потрібним способом за допомогою одних тільки атрибутів, внаслідок чого обмежуються можливості форматування;

в) велика кількість конструкцій, які не відносяться безпосередньо до розмітики вмісту, внаслідок чого виходить великий розмір файлу, підвищується час завантаження сторінки та трафік;

г) код втрачає свою логічну розмітку і як наслідок погано індексується пошуковими роботами, тому що тепер їм потрібно обробити велику кількість непотрібної для них інформації.

В проекті даний стандарт використовується для керування виглядом веб сторінки.

* + 1. Текстовий формат обміну даними JSON

JSON (JavaScript Object Notation) – це формат обміну даними між комп’ютерами, який представлений у текстовому вигляді заснований на мові програмування JavaScript. Не дивлячись на походження від JavaScript, даний формат є універсальним та може використовуватися практично з будь якою мовою програмування.

JSON-текст являє собою одну з двох типів структур:

а) набір пар ключ-значення (об’єкт, структура, словник, список, хеш-таблиця, запис);

б) впорядкований набір значень (вектор, масив або послідовність).

Висновки за розділом

РОЗДІЛ 4. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Проектування структури системи

## 4.2 Проектування користувальницької частини системи

4.3 Проектування моделі бази даних

Висновки за розділом

РОЗДІЛ 5. РЕАЛІЗАЦІЯ СПРОЕКТОВАНИХ ПІДСИСТЕМ

## 5.1 Реалізація моделі бази даних

## 5.2 Реалізація серверної частини системи

## 5.3 Реалізація клієнтської частини системи

Висновки за розділом

РОЗДІЛ 6. ПЕРЕВІРКА РОБОТИ ТА ТЕСТУВАННЯ СИСТЕМИ

## 6.1 Ручне тестування

## 6.2 Модульне тестування

Висновки за розділом

РОЗДІЛ 7. КЕРІВНИЦТВО КОРИСТУВАЧА

## 7.1 Керівництво для незареєстрованого оператора

## 7.2 Керівництво для зареєстрованого оператора

Висновки за розділом

РОЗДІЛ 8. ОХОРОНА ПРАЦІ

## 8.1 Вступ

## 8.2 Характеристика робочого приміщення

## 8.3 Аналіз шкідливих та небезпечних виробничих факторів

## 8.3.1 Мікроклімат виробничих приміщень

## 8.3.2 Освітлення

## 8.3.3 Характеристика виробничого шуму

## 8.3.4 Характеристика випромінювання

## 8.3.5 Електробезпека

## 8.3.6 Пожежна безпека

## 8.4 Інструкція з техніки безпеки

Висновки за розділом

ВИСНОВКИ

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вікіпедія. Skype [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Skype

2. Microsoft Teams – сервис для совместной работы над проектами. [Електронний ресурс] – Режим доступу: https://tearay.net/microsoft-teams-servis-dlya-sovmestnoj-raboty-nad-proektami/

3. Вікіпедія. Slack [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Slack

4. Вікіпедія. Telegram [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Telegram

5. Вікіпедія. WhatsApp [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/WhatsApp

6. Facebook Messenger vs WhatsApp: 48 фактов в сравнении [Електронний ресурс] – Режим доступу: https://versus.com/ru/facebook-messenger-vs-whatsapp

7. Клієнт-серверна архітектура [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Клієнт-серверна\_архітектура

8. Клієнт-серверна архітектура та ролі серверів. [Електронний ресурс] – Режим доступу: https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229

9. Вікіпедія. .NET Framework [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/.NET\_Framework#.NET\_Core

10. Вікіпедія. C Sharp [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/C\_Sharp

11. Достоинства SQL Server 2017 [Електронний ресурс] – Режим доступу: https://itpro.ua/product/microsoft-sql-server-2017/?tab=description

12. Вікіпедія. Реляційна модель даних [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Реляційна\_модель\_даних

13. Вікіпедія. Нормалізація баз даних [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Нормалізація\_баз\_даних

14. SignalR Core. «Hello Habr!» [Електронний ресурс] – Режим доступу: https://habr.com/company/jugru/blog/349096/

15. Вышла первая версия SignalR для ASP.Net Core 2.0 [Електронний ресурс] – Режим доступу: https://habr.com/post/338490/

16. Введение в ASP.NET Core [Електронний ресурс] – Режим https://metanit.com/sharp/aspnet5/1.1.php

17. ASP.NET Core MVC [Електронний ресурс] – Режим доступу: https://metanit.com/sharp/aspnet5/3.1.php

18. Metanit. Руководство по ASP.NET MVC 5 [Електронний ресурс] – Режим доступу: https://metanit.com/sharp/mvc5/

19. Введение в Entity Framework Core [Електронний ресурс] – Режим доступу: https://metanit.com/sharp/entityframeworkcore/1.1.php

20. Metanit. Entity Framework [Електронний ресурс] – Режим доступу: https://metanit.com/sharp/entityframework/

21. Вікіпедія. ASP.NET Web API [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/ASP.NET\_Web\_API

22. Вікіпедія. Модель «сутність — зв'язок» [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Модель\_сутність\_—\_зв%27язок

23. Воробьев С.В. Начало работы с базами данных: Учебно-методическое пособие. -М.: МИЭМ, 2008. - 60 с.

# Додаток А

# Додаток Б

Вихідні коди модульного тествання