

**БАКАЛАВРСЬКА РОБОТА**

**БР. ІІ - 18.00.00.000 ІІЗ**

**Група ІІ-21-4**

**Щербій Олександр**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Щербій Олександр Романович**

(прізвище, ім'я, по батькові)

УДК 004.4  
(індекс)

## **БАКАЛАВРСЬКА РОБОТА**

**Проектування та розробка системи обміну повідомленнями різного**

**рівня локалізації**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело**

Здобувач освітнього рівня Щербій О.Р.  
(підпис, ініціали та прізвище здобувача)

Науковий керівник Процюк Володимир Васильович, к.т.н., доцент  
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту  
Завідувач кафедри

доц. Бандура В.В.  
(посада) (підпис) (дата) (ініціали та прізвище)

**Івано-Франківськ – 2025**



## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз застосування інформаційних технологій для розробки систем обміну повідомленнями	05.05.2025	виконано
2	Розробка алгоритмічного забезпечення системи обміну повідомленнями про події	14.05.2025	виконано
3	Розробка архітектури системи обміну повідомленнями про події	29.05.2025	виконано
4	Проектування варіантів використання системи	01.06.2025	виконано
5	Програмна реалізація системи обміну повідомленнями різного рівня локалізації	16.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	11.06.2025	виконано

Студент – дипломник \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

Бакалаврська робота містить 77 сторінок, 35 рисунків, список використаних джерел із 35 найменуваннями.

**Метою даної роботи** є розробка архітектурної, алгоритмічної і програмної моделі системи обміну повідомленнями про події з урахуванням рівнів локалізації, категорій користувачів і динаміки подій.

**Об'єкт дослідження** - процеси передачі, обробки та керування повідомленнями у інформаційних системах.

**Предмет дослідження** - моделі, алгоритми та інструменти реалізації програмної системи обміну повідомленнями про події з урахуванням різних рівнів локалізації.

**В першому розділі** поведено детальний аналіз сучасних систем, таких як Svent, Eventbrite та Everbridge, що дозволив виявити ключові функціональні можливості й обмеження у сфері обміну повідомленнями про події.

**В другому розділі** розроблено архітектуру, сценарії використання та алгоритми функціонування системи, які враховують ролі користувачів та вимоги до безпеки й масштабованості.

**В третьому розділі** реалізовано структуровану модель бази даних, інтерфейс користувача для різних типів доступу, здійснено тестування функціональності, що підтверджує стабільність та ефективність системи.

**Висновок:** розроблено практичні аспекти проектування системи обміну повідомленнями про події з урахуванням їх географічної локалізації, ролей користувачів та характеру подій

**КЛЮЧОВІ СЛОВА:** ОБМІН ПОВІДОМЛЕННЯМИ, ЛОКАЛІЗАЦІЯ ПОДІЙ, УПРАВЛІННЯ ПОДІЯМИ, ПРОГРАМНА АРХІТЕКТУРА, МОДЕЛЬ БАЗИ ДАНИХ, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ТЕСТУВАННЯ СИСТЕМИ.

## ANNOTATION

The bachelor's thesis contains 77 pages, 35 figures, a list of used sources with 35 names.

**The purpose of this work** is to develop an architectural, algorithmic and software model of an event messaging system taking into account localization levels, user categories and event dynamics.

**The object of research** is the processes of transmission, processing and management of messages in information systems.

**The subject of research** is models, algorithms and tools for implementing a software event messaging system taking into account different localization levels.

**The first section** provides a detailed analysis of modern systems such as Cvent, Eventbrite and Everbridge, which allowed us to identify key functional capabilities and limitations in the field of event messaging.

**The second section** develops the architecture, usage scenarios and algorithms of the system, which take into account user roles and requirements for security and scalability.

**The third section** implements a structured database model, a user interface for different types of access, and functionality testing is carried out, confirming the stability and efficiency of the system.

**Conclusion:** practical aspects of designing an event messaging system have been developed, taking into account their geographical localization, user roles, and the nature of events. **KEYWORDS:** MESSAGING, EVENT LOCATION, EVENT MANAGEMENT, SOFTWARE ARCHITECTURE, DATABASE MODEL, USER INTERFACE, SYSTEM TESTING.

**KEYWORDS:** MESSAGING, EVENT LOCALIZATION, EVENT MANAGEMENT, SOFTWARE ARCHITECTURE, DATABASE MODEL, USER INTERFACE, SYSTEM TESTING.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМ ОБМІНУ ПОВІДОМЛЕННЯМИ .....	13
1.1. Завдання розробки системи інтегрованого управління подіями як системи обміну повідомленнями різного рівня локалізації .....	13
1.2. Мотивація розробки системи обміну повідомленнями про події .....	16
1.3. Аналіз існуючих програмних платформ і систем розповсюдження та обміну повідомлень про події .....	17
1.3.1. Комплексна хмарна програмна платформа Svent.....	17
1.3.2. Система Eventbrite .....	20
1.3.3. Платформа Everbridge.....	22
РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОБМІНУ ПОВІДОМЛЕННЯМИ ПРО ПОДІЇ .....	26
2.1. Архітектура системи обміну повідомленнями про події.....	26
2.2. Виклики управління подіями та умови ефективного використання розроблюваної.....	28
2.3. Аналіз вимог до системи. Модель випадків використання .....	30
2.3.1. Модель випадків використання загального користувача .....	31
2.3.2. Модель випадків використання адміністратора.....	31
2.3.3. Модель випадків використання авторизованого користувача .....	35
2.4. Проектування варіантів використання системи .....	37
2.4.1. Варіант використання: Реєстрація користувача.....	38
2.4.2. Варіант використання: Схвалення користувача.....	39
2.4.3. Варіант використання: Блокування користувача .....	40

					<b>БР.ІІ – 18.00.00.000 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Щербій О.Р.</i>			Проектування та розробка системи обміну повідомленнями різного рівня локалізації  <b>ПОЯСНЮВАЛЬНА ЗАПИСКА</b>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
<i>Перевір.</i>		<i>Процюк В.В.</i>					6	
<i>Реценз.</i>						<b>ІФНТУНГ Ш-21-4</b>		
<i>Н. Контр.</i>		<i>Піх М.М.</i>						
<i>Затверд.</i>		<i>Бандура В.В.</i>						

2.4.4. Варіант використання: Видалення користувача .....	40
2.4.5. Варіант використання: Додавання члена організації до внутрішнього списку .....	41
2.4.6. Варіант використання: Перегляд подій, створених конкретним користувачем .....	42
2.4.7. Варіант використання: Оновлення персональної інформації.....	43
2.4.8. Варіант використання: Створення події та планування повідомлення .....	44
2.4.9. Варіант використання: Перегляд власних попередніх подій .....	45
2.4.10. Варіант використання: Видалення власних подій .....	46
2.5. Технічні вимоги та характеристики системи.....	46
<b>РОЗДІЛ 3. ПРОЕКТУВАННЯ МОДЕЛІ БАЗИ ДАНИХ І ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ОБМІНУ ПОВІДОМЛЕННЯМИ РІЗНОГО РІВНЯ ЛОКАЛІЗАЦІЇ .....</b>	
3.1. Проектування моделі бази даних .....	49
3.2. Реалізація схеми бази даних .....	51
3.3. Модель взаємодії з графічним інтерфейсом користувача .....	54
3.3.1. Головна сторінка системи .....	54
3.3.2. Сторінка аутентифікації (входу) .....	55
3.3.3. Сторінка реєстрації користувача .....	57
3.4. Модель взаємодії з графічним інтерфейсом адміністратора .....	58
3.5. Модель взаємодії з графічним інтерфейсом авторизованого користувача .....	61
3.6. Вибір методології для реалізація графічного інтерфейсу користувача (GUI) .....	64
3.7. Тестування системи .....	66
3.7.1. Модульне тестування .....	66
3.7.2. Системне тестування .....	70
<b>ВИСНОВКИ .....</b>	<b>72</b>

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ..... 74

БІБЛІОГРАФІЧНА ДОВІДКА

					БР.ІІІ – 18.00.00.000 ІІЗ	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IEM - Integrated event messaging system

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

IDE - Integrated Development Environment

SRS - Software Requirement Specification

GUI - Graphical User Interface (Графічний інтерфейс користувача)

UI - User Interface

Admin - Administrator

MVC - Model-View-Controller

UT - Unit Testing

ST - System Testing

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Сучасні інформаційно-комунікаційні технології відіграють ключову роль у забезпеченні ефективної взаємодії між учасниками складних соціально-економічних та організаційних систем. З огляду на динамічність середовища, у якому діють державні, комерційні та громадські структури, виникає нагальна потреба в засобах, що дозволяють оперативно інформувати учасників про події, реагувати на інциденти, координувати дії в реальному часі. Одним із найважливіших напрямів у цьому контексті є створення систем обміну повідомленнями, здатних адаптуватися до змін, враховувати специфіку подій та забезпечувати їх доставку з урахуванням географічної локалізації та ієрархії користувачів.

Загальновідомо, що ефективність системи комунікації безпосередньо залежить від точності, актуальності та персоналізованості інформації, що передається. Багаторівнева структура подій — від локальних до глобальних — вимагає наявності механізмів фільтрації та пріоритезації повідомлень. При цьому важливим є також збереження контексту взаємодії, безперебійна робота у багатокористувацькому режимі та відповідність вимогам безпеки даних.

### **Актуальність роботи**

У сучасних умовах цифровізації суспільства ефективна організація обміну повідомленнями, зокрема під час управління подіями різного рівня локалізації, набуває особливої важливості. Існуючі інструменти часто не забезпечують достатнього рівня масштабованості, персоналізації та інтеграції між користувачами, адміністраторами та системами зовнішнього контролю. У зв'язку з цим розробка інноваційної програмної системи, що дозволяє здійснювати ефективний обмін повідомленнями з урахуванням ролей користувачів та особливостей подій, є важливим напрямом дослідження як у

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

сфері інформаційних технологій, так і в межах загальної стратегії цифрової трансформації управлінських процесів.

Актуальність теми зумовлена необхідністю створення універсального інструменту, здатного підтримувати адаптивний і масштабований обмін повідомленнями у гетерогенних інформаційних середовищах, включаючи кризові комунікації, управління подіями, корпоративну взаємодію та громадські сервіси.

Існуючі рішення, представлені на ринку, зокрема такі як Svent, Eventbrite, Everbridge, демонструють значний функціональний потенціал, але водночас мають певні обмеження щодо відкритості архітектури, можливостей інтеграції, локалізації та адаптації до специфіки конкретного середовища. Це створює підґрунтя для дослідження та розробки власної моделі системи обміну повідомленнями, яка поєднувала б переваги подієво-орієнтованого підходу, сучасні принципи розробки програмного забезпечення та гнучку архітектуру, що підлягає масштабуванню.

**Метою даної роботи** є аналіз теоретичних основ та розробка архітектурної, алгоритмічної і програмної моделі системи обміну повідомленнями про події з урахуванням рівнів локалізації, категорій користувачів і динаміки подій.

У межах поставленої мети вирішуються завдання аналізу вимог, побудови моделей використання, розробки структури бази даних, реалізації інтерфейсу та проведення тестування функціональності системи.

#### **Завдання дослідження**

Провести аналіз сучасних інформаційних технологій для побудови систем обміну повідомленнями.

Визначити функціональні та нефункціональні вимоги до системи.

Побудувати архітектуру системи та алгоритмічне забезпечення.

Спроекувати модель бази даних та інтерфейс взаємодії користувачів.

Реалізувати функціональні модулі системи.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Провести модульне та системне тестування розробленого рішення.

**Об'єкт дослідження** - процеси передачі, обробки та керування повідомленнями у інформаційних системах.

**Предмет дослідження** - моделі, алгоритми та інструменти реалізації програмної системи обміну повідомленнями про події з урахуванням різних рівнів локалізації.

#### **Методи дослідження**

- Системний аналіз предметної області;
- Об'єктно-орієнтоване проектування (UML-діаграми);
- Методи розробки програмного забезпечення (Agile-елементи);
- Методи баз даних (ER-моделювання);
- Методи модульного та системного тестування.

#### **Наукова новизна**

Запропоновано алгоритмічну архітектуру системи обміну повідомленнями, що підтримує багаторівневу локалізацію повідомлень та гнучке управління користувачами, а також реалізовано інтегрований підхід до побудови GUI з урахуванням типів доступу та рівнів взаємодії.

#### **Практичне застосування**

Результати цієї роботи мають практичну цінність для організацій, які потребують високоефективного каналу комунікації — зокрема в галузях безпеки, управління надзвичайними ситуаціями, медицини, освіти, а також в рамках корпоративного управління.

Бакалаврська робота містить 77 сторінок, 35 рисунків, 3 розділи список використаних джерел із 35 найменуваннями, 1 додаток.

					БР.ІІ – 18.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 1. АНАЛІЗ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМ ОБМІНУ ПОВІДОМЛЕННЯМИ

## 1.1. Завдання розробки системи інтегрованого управління подіями як системи обміну повідомленнями різного рівня локалізації

Сучасне організаційне середовище характеризується експоненційним зростанням обсягу та складності подій і завдань, що вимагає високого рівня ефективності управління часом та ресурсами. Традиційні методи організації та координації часто призводять до комунікаційних збоїв та операційних помилок. У відповідь на ці виклики, було розроблено прототип інформаційної системи для інтегрованого управління подіями, що функціонує як система обміну повідомленнями різного рівня локалізації. Ця система призначена для автоматизації процесів планування, організації та комунікації, пов'язаних з подіями, тим самим мінімізуючи людські помилки та підвищуючи загальну ефективність. Розроблений прототип реалізовано як веб-додаток з використанням технологій PHP для серверної логіки та інтерфейсу користувача, а також MySQL для управління базою даних. Система забезпечує групування учасників за подіями та цільове надсилання SMS-повідомлень відповідальним особам. Управління доступом та модерація запитів здійснюється адміністратором системи.

В умовах динамічного розвитку технологій та зростання інформаційних потоків, ефективне управління часом стає критично важливим фактором для забезпечення успіху будь-якої організації чи інституції. Збільшення кількості та складності подій і операційних завдань вимагає розробки та впровадження автоматизованих рішень для їх оптимальної організації та контролю. Недостатньо ефективна координація та комунікація між учасниками подій може призводити до суттєвих затримок,

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

непорозумінь та зниження продуктивності. Таким чином, існує нагальна потреба у створенні систем, що здатні забезпечити безперебійне управління подіями та налагодити ефективний інформаційний обмін.

Управління зростаючим обсягом організаційних подій із застосуванням переважно ручних методів є ресурсомістким та схильним до помилок процесом. Відсутність централізованої платформи для інтеграції інформації про події з даними про відповідальних осіб, а також неефективні канали комунікації створюють значні перешкоди для своєчасного та точного поширення необхідної інформації. Це призводить до низької операційної ефективності, втрати часу та потенційних фінансових втрат. Виникає необхідність у розробці автоматизованої системи, яка б дозволила оптимізувати ці процеси.

Розроблена система має забезпечити автоматизацію ключових етапів організації подій, покращити комунікацію між учасниками та мінімізувати ризики, пов'язані з ручною обробкою даних.



Рисунок 1.1 – Основні функціональні можливості розроблюваної системи

У рамках даної роботи було розроблено прототип інформаційної системи, що спроектованою для безперебійного управління різноманітними

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

організаційними подіями, діючи як система обміну повідомленнями різного рівня локалізації. Ця система дозволяє мінімізувати операційні помилки, що виникають при ручному виконанні відповідних завдань. Ефективне використання системи потенційно може значно підвищити пропускну здатність організації щодо управління завданнями у скорочені терміни.

Розроблений прототип реалізовано як веб-додаток. Для розробки логіки та інтерфейсу користувача було обрано мову програмування PHP, що забезпечує гнучкість та сумісність із веб-середовищем. Систему управління базами даних MySQL використано для структурованого зберігання всієї необхідної інформації щодо подій, учасників та повідомлень.

Система забезпечує наступні ключові функціональні можливості (рис. 1.1):

- Групування учасників за подіями. Автоматичне формування списків осіб, пов'язаних з конкретною подією, на основі введених даних.

- Цільове повідомлення (SMS). Можливість надсилання автоматизованих SMS-повідомлень з необхідною інформацією про подію безпосередньо відповідальним особам або групам учасників. Це є реалізацією функції обміну повідомленнями на локальному рівні (рівні групи або індивіда).

- Адміністративний контроль. Централізоване управління системою здійснюється адміністратором, який відповідає за модерацію запитів від членів організації.

- Управління доступом. Члени організації можуть подавати запити на використання системи. Після затвердження адміністратором, член отримує статус користувача та можливість ініціювати функціонал повідомлення.

- Обмеження та масштабованість. Розроблена система є функціональним прототипом, орієнтованим на задоволення базових потреб невеликої організації. Однак, її архітектура має закладену гнучкість, що

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє подальше розширення функціоналу та масштабування відповідно до зростаючих вимог та специфіки більш складних організаційних структур.

Розроблюваний прототип системи, що функціонує як система обміну повідомленнями різного рівня локалізації, демонструє потенціал автоматизації процесів управління подіями та комунікацій в організації. Впровадження подібних систем може суттєво підвищити операційну ефективність, знизити кількість помилок та забезпечити більш раціональне використання часових ресурсів в сучасному динамічному організаційному середовищі.

## **1.2. Мотивація розробки системи обміну повідомленнями про події**

Метою даної роботи є детальне представлення архітектури та функціоналу прототипу інформаційної системи обміну повідомленнями різного рівня локалізації. Робота має на меті надати вичерпний опис підходів та методів, використаних для розробки веб-додатку, який реалізує функціонал автоматизованого групового та цільового оповіщення. Буде продемонстровано, як реалізація системи як системи обміну повідомленнями різного рівня локалізації дозволяє оптимізувати процеси управління подіями, мінімізувати операційне навантаження на організаторів та забезпечити своєчасне і точне доведення інформації про майбутні події до відповідних груп учасників або відповідальних осіб.

Організаційна діяльність сучасних інституцій та підприємств пов'язана зі зростаючою складністю та кількістю подій, які вимагають ретельної підготовки та координації. Природа цих подій може суттєво варіюватися залежно від специфіки діяльності організації. Ключовим аспектом успішної організації події є ефективне інформування всіх залучених сторін. Традиційні підходи, що базуються на ручному розсиланні повідомлень значним групам адресатів, є неефективними, потребують значних часових ресурсів та схильні

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

до помилок. Можливість пропуску важливого повідомлення окремим організатором або учасником створює високий ризик комунікаційних збоїв, що може негативно позначитися на перебігу та результатах події.

Отже, виникає об'єктивна потреба в розробці автоматизованої системи, здатної взяти на себе функції масового та цільового оповіщення. Така система, функціонуючи як система обміну повідомленнями різного рівня локалізації, дозволить забезпечити автоматизоване розсилання необхідної інформації про події конкретним групам осіб або індивідуальним відповідальним сторонам, тим самим мінімізуючи людський фактор та пов'язані з ним ризики. Мотивацією для створення веб-орієнтованого порталу, що виступає в ролі автоматизованого відправника повідомлень про події, є прагнення надати організаціям інструмент для значного підвищення ефективності процесів інформування, звільнивши їх від рутинних операцій та забезпечивши надійну та своєчасну комунікацію з усіма зацікавленими сторонами.

### **1.3. Аналіз існуючих програмних платформ і систем розповсюдження та обміну повідомлень про події**

Існують різноманітні системи, які виконують функцію розповсюдження повідомлень про події, від спеціалізованих платформ управління подіями до систем масового сповіщення. Їх функціонал може відрізнитись за масштабом, цільовою аудиторією та основними можливостями (наприклад, лише оповіщення, або повний цикл управління подією з комунікаціями).

#### *1.3.1. Комплексна хмарна програмна платформа Svent*

Svent – це провідна комплексна хмарна програмна платформа для управління повним життєвим циклом подій (Event Management System, EMS)

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

та управління місцями зустрічей. Вона розроблена для надання організаціям інструментів, необхідних для ефективного планування, маркетингу, проведення та аналізу різноманітних заходів – від невеликих зустрічей до масштабних конференцій та виставок, як у фізичному, віртуальному, так і гібридному форматах.

Система Cvent призначена переважно для використання корпораціями, асоціаціями, урядовими установами та освітніми закладами, які регулярно проводять значну кількість подій.

Title	Code	Status	Start Date	Yes	No	Invited
Social Media	FMNCKJWSPY	Active	10/04/2016	0	0	1
Concur_OK	H9NK9B0X75P	Active	01/25/2016	1	0	1
Princeton Club Membership	KKN5RYG363	Pending	10/17/2015	0	0	52
New Frontiers in Interventional Cardiology 2015	FN0X0262PVM	Active	10/10/2015	175	13	215
Blue State University Campus Tours	Z2N634887PH	Active	09/01/2015	13	0	18
Blue State University Campus Tours	PCN00YLVPF	Pending	09/01/2015	0	0	0
Bluecorp 20th General Assembly	ZFNKQRO6PR	Active	08/24/2015	1	0	2
Division Managers Association Luncheon	4MN8UTPV68	Active	08/01/2015	45	5	53
ECO Science Camp	PVNT4CJ5Z8	Active	07/29/2015	12	1	10
UEFA Euro (auction)	PJNVMP7J4D9	Active	07/20/2015	5	0	5
UEFA Euro	HYXK096NRT8	Active	07/20/2015	0	0	1
Blue Corp's BIG Event	56NMSG78QLY	Active	07/12/2015	113	7	128
2013年18日牛展大会	VRN46LHG0W8	Active	07/11/2015	5	0	7
San Fermín: Running of the Bulls	K0NXLVN74D94	Active	07/06/2015	3	2	6

Рисунок 1.2 – Вигляд системи Cvent

Основні функціональні модулі та можливості системи Cvent включають:

- Надання інструментів для створення кастомізованих онлайн-форм реєстрації, управління даними учасників, обробки платежів, застосування знижок та промокодів.

- Конструктори для розробки професійних, брендovаних веб-сайтів для кожної події, що слугують центральним інформаційним ресурсом та платформою для реєстрації.

- Широкий спектр інструментів для просування подій та взаємодії з потенційними та зареєстрованими учасниками. Це включає:

- Модулі email-маркетингу для створення, сегментації, розсилки та відстеження ефективності запрошень, нагадувань, оновлень та іншої інформації.

- Функціонал управління соціальними мережами.

Можливості для створення мобільних сповіщень через додаток події.

- Інструменти для управління рекламою.

- Планування та організація сесій, управління спікерами, виставковими площами, розробка планів локацій, управління постачальниками та бюджетом події.

- Створення кастомізованих мобільних додатків для подій, що забезпечують учасників програмою, інформацією про спікерів та локації, можливостями для нетворкінгу, інтерактивними картами та отриманням push-сповіщень.

- Інструменти для підвищення залученості учасників під час події (наприклад, опитування, Q&A сесії, ігрові елементи, можливості для нетворкінгу).

- Надання детальних звітів та аналітичних даних щодо реєстрації, відвідуваності, ефективності комунікаційних кампаній, фінансових показників та загальної успішності події для оцінки ROI.

- Можливості для пошуку, порівняння та бронювання локацій для проведення подій.

Svent є не просто системою розповсюдження повідомлень, а комплексною платформою, де комунікації є інтегральною частиною більш широкого процесу управління подіями, забезпечуючи централізований контроль та автоматизацію на всіх етапах.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.3.2. Система Eventbrite

Система Eventbrite - популярна онлайн-платформа для створення, просування та управління подіями (особливо публічними), яка також надає можливості для розсилання повідомлень учасникам.

Основні можливості Eventbrite для організаторів:

- Можна швидко створити сторінку події, вказавши її назву, опис, дату, час та місце проведення.

- Дозволяє створювати різні типи квитків (платні, безкоштовні, з різними цінами), встановлювати їх кількість, терміни продажу та налаштовувати реєстраційні форми. Система автоматично обробляє платежі для платних квитків.

- Генерується професійна веб-сторінка для кожної події, яка містить всю необхідну інформацію для відвідувачів та є платформою для реєстрації/продажу квитків.

- Надає інструменти для маркетингу, включаючи розсилку запрошень електронною поштою, інтеграцію із соціальними мережами для поширення, а також видимість події на самому сайті Eventbrite (який працює як каталог подій).

- Дозволяє переглядати списки зареєстрованих учасників, керувати їхніми даними, а також спілкуватися з ними (наприклад, надсилати оновлення чи нагадування).

- Надає мобільні додатки як для організаторів (для відстеження продажів, керування учасниками та сканування квитків під час заходу), так і для відвідувачів (для зберігання квитків та пошуку подій).

- Надає доступ до звітів про продажі квитків, реєстрації, відвідуваність сторінки події тощо.

Для відвідувачів Eventbrite працює як:

- Дозволяє шукати події за інтересами, датою, місцем розташування.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

- Надає зручний інтерфейс для вибору квитків, заповнення даних та безпечної оплати.

- Зберігає всі куплені або отримані квитки в одному місці (на сайті та в мобільному додатку).

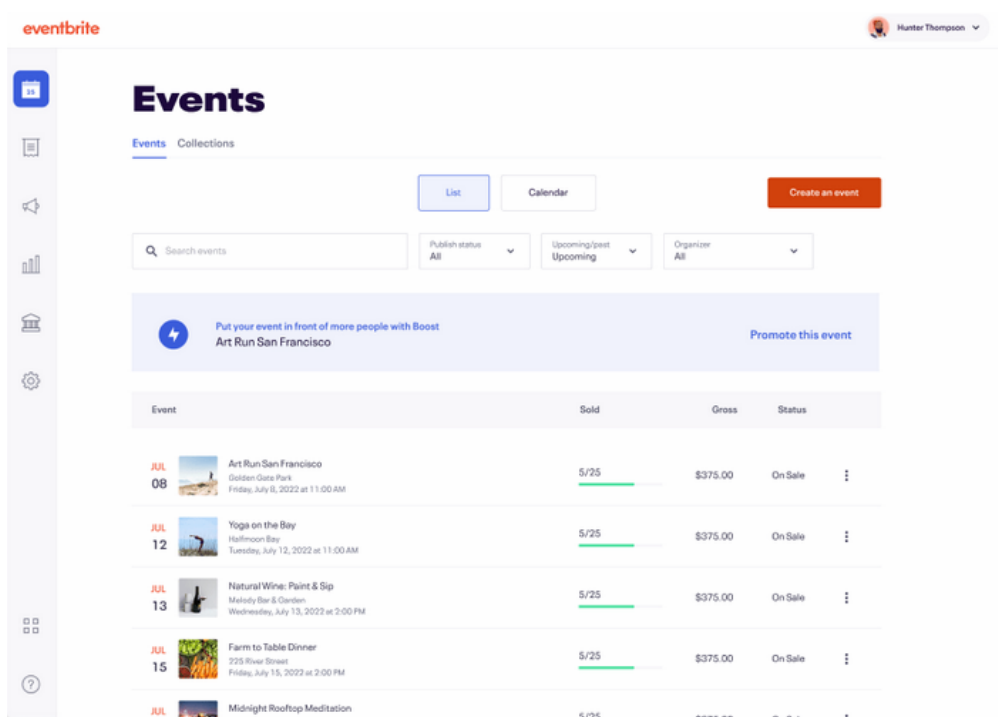


Рисунок 1.3 – Інтерфейс системи Eventbrite

Eventbrite є гнучким рішенням, яке підходить для широкого спектру подій – від невеликих майстер-класів та вебінарів до концертів, фестивалів та бізнес-заходів. Його бізнес-модель зазвичай передбачає стягнення комісії з проданих платних квитків, що робить його доступним варіантом для організаторів з обмеженим бюджетом.

З технічної точки зору, Eventbrite є хмарним SaaS-рішенням, що забезпечує масштабованість та доступність. Модель монетизації зазвичай базується на відсотках та/або фіксованих зборах з проданих платних квитків, що робить платформу доступною для організаторів без початкових інвестицій.

Таким чином, Eventbrite є прикладом успішної цифрової платформи, яка, поєднуючи функції онлайн-сервісу для управління подіями та маркетингу, значно спростила процеси організації, просування та участі у різноманітних заходах, надавши широкому колу користувачів потужні інструменти для реалізації їхніх подійних ініціатив.

### 1.3.3. Платформа Everbridge

Everbridge – це глобальна програмна платформа, яка спеціалізується на управлінні критичними подіями (Critical Event Management) та наданні рішень для масового сповіщення. Головне призначення системи – допомогти організаціям та державним установам швидко та надійно зв'язуватися з людьми під час надзвичайних ситуацій, кризових ситуацій або важливих подій, щоб забезпечити їхню безпеку та безперервність операційної діяльності.

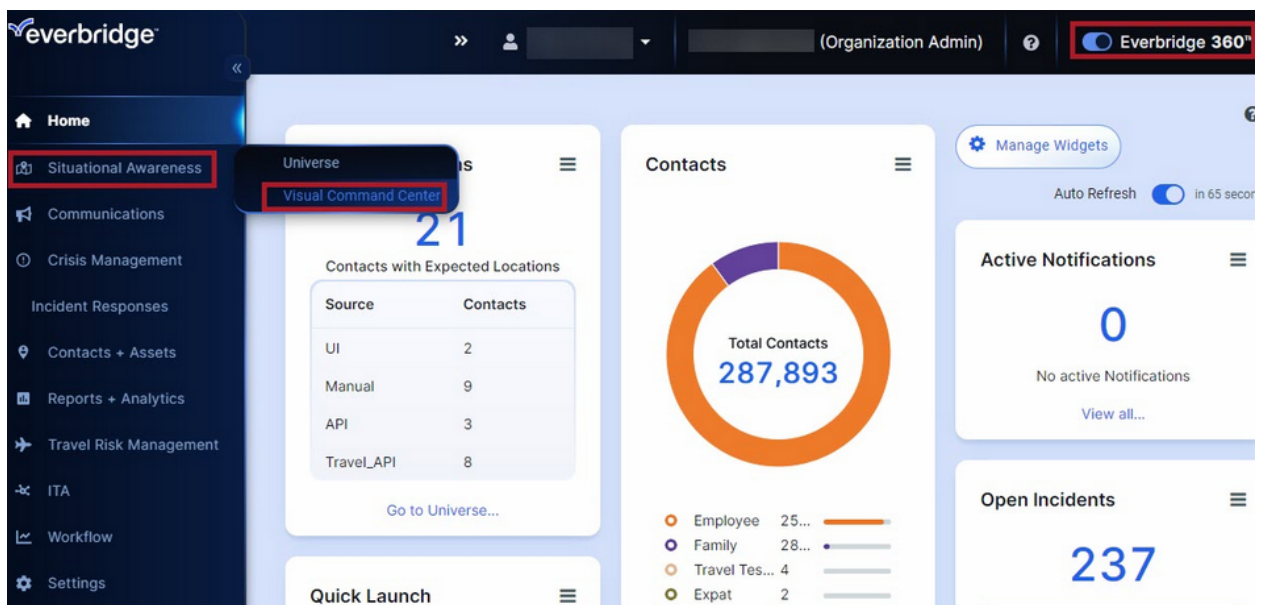


Рисунок 1.4 – Вигляд Everbridge

Основні функції та можливості Everbridge:

- Дозволяє одночасно надсилати повідомлення великій кількості людей.

Це основна функція системи.

						Арк.
					БР.ІІ – 18.00.00.000 ПЗ	22
Змн.	Арк.	№ докум.	Підпис	Дата		

- Система може надсилати повідомлення через різні канали зв'язку одночасно або послідовно, включаючи:

- SMS-повідомлення
- Електронну пошту (email)
- Голосові дзвінки (на стаціонарні та мобільні телефони)
- Мобільні push-сповіщення (через мобільний додаток Everbridge)
- Повідомлення у соціальні мережі
- Повідомлення на робочі столи комп'ютерів
- Системи публічного оповіщення

- Дозволяє надсилати повідомлення певним групам людей на основі їхнього місцезнаходження (географічне таргування), ролі в організації, приналежності до певних списків контактів або інших критеріїв.

- Ведення централізованої бази даних контактів з можливістю вказати їхні контактні дані та бажані канали зв'язку.

- Надання інструментів для координації дій під час критичної події – від її виявлення до комунікації та аналізу відповіді.

- Можливість інтеграції з іншими системами (наприклад, системами безпеки, ІТ-моніторингу) для автоматичного запуску сповіщень при певних подіях.

- Надання детальних звітів про доставку повідомлень, підтвердження отримання та відповіді від адресатів.

Типові сценарії використання Everbridge:

1. Оповіщення про надзвичайні ситуації (стихійні лиха, погодні попередження, загрози безпеці).
2. Інформування персоналу про збої в роботі (ІТ-аварії, відключення електроенергії, закриття будівель).
3. Координація дій аварійних та рятувальних служб.
4. Державні системи публічного оповіщення населення.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

5. Використовується для швидкого оповіщення відвідувачів конференцій, концертів, спортивних подій про зміни в розкладі, надзвичайні ситуації на локації або інші важливі оновлення.

Клієнтами Everbridge є великі підприємства, муніципалітети, державні установи, заклади охорони здоров'я та освіти по всьому світу, які потребують надійного інструменту для ефективної комунікації у критично важливих ситуаціях. По суті, Everbridge – це інфраструктура для швидкого, широкого та націленого розповсюдження повідомлень, що є життєво важливим у кризових умовах.

#### **1.4. Архітектура та основний функціонал системи обміну повідомленнями про події**

Пропонована система являє собою веб-орієнтовану інформаційну платформу, спроектовану для автоматизації процесів управління подіями та оптимізації пов'язаних комунікацій. За своєю суттю, система функціонує як система обміну повідомленнями різного рівня локалізації, що дозволяє здійснювати цільове оповіщення визначених груп користувачів або окремих осіб щодо майбутніх подій.

Доступ до функціональних можливостей системи здійснюється через веб-інтерфейс. Для взаємодії із системою користувач повинен пройти процедуру аутентифікації (увійти до системи), використовуючи попередньо створений обліковий запис.

Центральним елементом функціоналу є модуль створення та планування подій з подальшим автоматизованим розсиланням повідомлень. Після успішного входу, авторизовані користувачі отримують можливість ініціювати створення нової події в системі. Процес створення події передбачає внесення наступних ключових даних:

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

- Зміст повідомлення, що деталізує інформацію про подію (наприклад, дата, час, місце, порядок денний тощо).

- Планована дата та точний час автоматичного надсилання повідомлення. Це дозволяє планувати комунікацію заздалегідь.

- Формування списку отримувачів — визначення цільової групи осіб або індивідуальних користувачів, які мають отримати сповіщення. Можливість точного визначення списку адресатів реалізує принцип обміну повідомленнями різного рівня локалізації, забезпечуючи релевантність та адресність інформаційних потоків.

Після введення всіх необхідних даних та підтвердження, система реєструє нову подію та планує автоматичне надсилання відповідного повідомлення всім зазначеним отримувачам у визначений час.

Для отримання доступу до функціоналу ініціювання повідомлень (статус члена системи), користувачі повинні пройти первинну процедуру реєстрації. Реєстрація дозволяє створити унікальний обліковий запис та отримати відповідні права доступу.

В рамках подальшого розвитку системи передбачається розробка мобільного додатку. Мобільний клієнт матиме на меті забезпечити зручний альтернативний інтерфейс для взаємодії з веб-платформою. Функціонал мобільного додатку включатиме можливості реєстрації нових користувачів, аутентифікації (входу), а також потенційно доступ до ключових операцій з управління подіями та перегляду повідомлень, забезпечуючи доступність системи з мобільних пристроїв.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОБМІНУ ПОВІДОМЛЕННЯМИ ПРО ПОДІЇ

### 2.1. Архітектура системи обміну повідомленнями про події

На рисунку 2.1 представлена високорівнева архітектуру системи, яка реалізує функціонал обміну повідомленнями різного рівня локалізації для ефективного управління організаційними подіями.

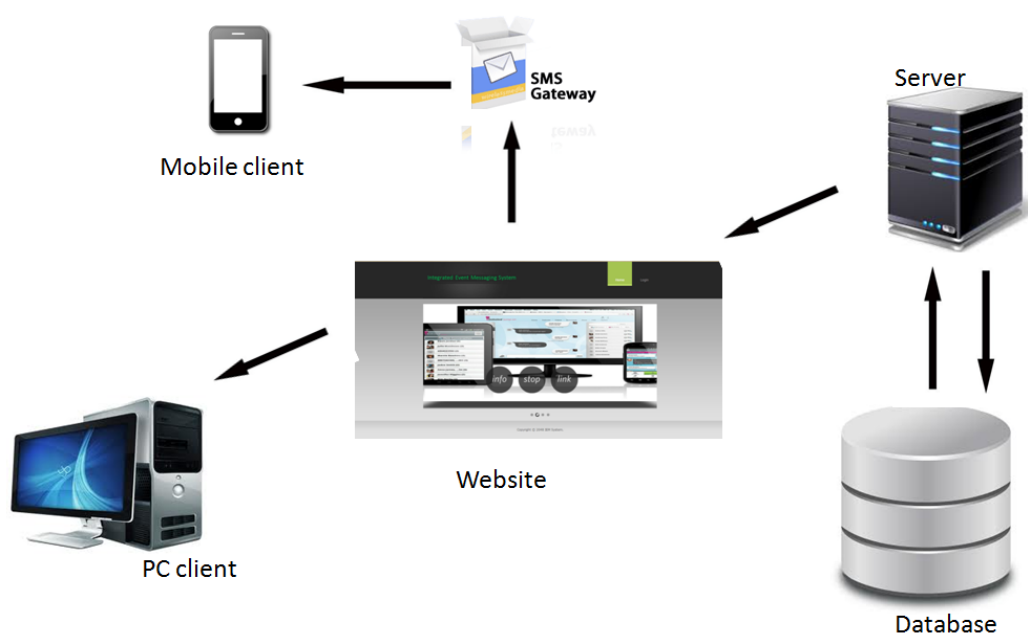


Рисунок 2.1 – Архітектура системи обміну повідомленнями про події

Система складається з декількох ключових компонентів, що взаємодіють між собою:

#### 1. Клієнтські інтерфейси (Client Interfaces)

ПК-клієнт (PC client): Представляє робочі станції користувачів, які отримують доступ до системи за допомогою веб-браузера. Це типовий інтерфейс для доступу через настільні або портативні комп'ютери.

					БР.ІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Мобільний клієнт (Mobile client): Відображає доступ до системи за допомогою мобільних пристроїв (смартфонів, планшетів). Доступ може здійснюватися через мобільний веб-браузер або спеціалізований мобільний додаток, що планується до розробки. Мобільний клієнт також є кінцевою точкою доставки SMS-повідомлень.

## 2. Веб-сайт (Website)

Є центральним компонентом користувацького інтерфейсу системи. Веб-сайт виступає як єдина точка доступу для обох типів клієнтів, забезпечуючи можливості для аутентифікації користувачів, створення та редагування подій, формування списків отримувачів та ініціювання надсилання повідомлень. Він взаємодіє із серверною частиною системи.

## 3. Сервер (Server)

Серверний компонент, на якому виконується основна бізнес-логіка системи ІПП. Сервер обробляє запити, що надходять від веб-сайту (наприклад, запити на створення події, вхід користувача, надсилання повідомлення), керує даними та координує взаємодію з іншими компонентами, зокрема, з базою даних та SMS-шлюзом.

## 4. База даних (Database)

Сховище даних системи. У базі даних зберігається вся релевантна інформація, включаючи облікові записи користувачів, деталі створених подій, контент повідомлень, списки отримувачів, графіки розсилки та іншу операційну інформацію. Сервер здійснює операції читання та запису даних у базу даних для забезпечення функціонування системи.

## 5. SMS-шлюз (SMS Gateway)

Зовнішній інтегрований сервіс, призначений для автоматизованого надсилання SMS-повідомлень на мобільні номери отримувачів. Сервер системи ІПП взаємодіє з SMS-шлюзом, передаючи йому запити на відправлення повідомлень із зазначеним текстом та номерами телефонів

					БР.ІП – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

цільової аудиторії. SMS-шлюз забезпечує безпосередню доставку цих повідомлень на мобільні пристрої користувачів.

Користувачі взаємодіють із системою через клієнтські інтерфейси (ПК або мобільний клієнт), отримуючи доступ до веб-сайту. Веб-сайт передає запити на сервер, який обробляє їх, звертаючись до бази даних за необхідними даними або для їх збереження. У випадку, коли потрібно надіслати повідомлення про подію, сервер формує відповідний запит і надсилає його до SMS-шлюзу. SMS-шлюз здійснює фізичну доставку SMS-повідомлення на мобільні пристрої користувачів, визначених у списку отримувачів для даної події. Ця інтегрована архітектура забезпечує функціональність цільового обміну повідомленнями, що є фундаментальною для системи управління подіями.

## **2.2. Виклики управління подіями та умови ефективного використання розроблюваної**

Організаційна діяльність у більшості установ передбачає проведення множинних подій різного масштабу та складності, що вимагає ефективної взаємодії та координації між залученими суб'єктами – членами команди організаторів, учасниками, постачальниками тощо. При застосуванні ручних методів координації, зокрема для розповсюдження інформації, виникає значний ризик втрати повідомлень, що може призвести до комунікаційних збоїв, непорозумінь та операційних проблем. Неотримання критично важливого повідомлення одним із ключових членів команди, відповідальних за організацію події, може мати каскадні наслідки та поставити під загрозу успішну реалізацію всього заходу.

Крім того, ручне формування та управління робочими групами для конкретних подій "на вимогу" є процесом, що потребує значних часових та людських ресурсів, і є менш гнучким у порівнянні з попереднім, системним

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

групуванням. Натомість, попереднє визначення та структуризація команд для організації подій у рамках централізованої системи сприяє чіткому розподілу обов'язків, підвищує прозорість процесів та забезпечує кращу підготовленість усіх учасників, що позитивно впливає на загальну атмосферу та ефективність роботи.

Саме для подолання цих системних викликів, пов'язаних з неефективністю ручної координації та ризиками втрати інформації, виникає нагальна потреба у впровадженні автоматизованих систем управління комунікаціями. Система повідомлень про події, що виконує функції автоматизованої координації, є необхідним інструментом для забезпечення надійності, швидкості та точності обміну інформацією в процесі організації заходів.

Ефективність використання системи надання повідомлень про події як інструменту управління подіями та обміну повідомленнями різного рівня локалізації значною мірою залежить від дотримання певних організаційних та операційних умов. Потенціал системи буде реалізовано повною мірою лише за умови:

1. Якісного попереднього планування. Майбутні події організації мають бути ретельно сплановані та передбачені заздалегідь.
2. Оптимального попереднього формування груп. Для кожної запланованої події необхідно ефективно та завчасно здійснити групування осіб, відповідальних за її організацію або участь. Точність та актуальність цих груп є критично важливою для реалізації функціоналу цільового обміну повідомленнями.
3. Надійної роботи системи. Повинна бути забезпечена безперебійна та коректна робота модуля генерації повідомлень у веб-інтерфейсі, а також надійність функціоналу автоматизованої розсилки.
4. Системного адміністративного контролю. Життєво важливою є наявність відповідального адміністратора, який здійснює централізоване

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

управління системою, включаючи модерацію запитів від членів організації (прийняття, відхилення, оновлення статусу), управління правами доступу та забезпечення цілісності даних.

За умови правильного та послідовного виконання цих передумов, ефективно використання розробленої системи обміну повідомленнями про події дозволить значно спростити та автоматизувати процеси координації, мінімізувати ймовірність помилок, пов'язаних з людським фактором, та забезпечити організацію подій у найкоротші терміни з високим ступенем надійності. Таким чином, досягається суттєве підвищення операційної ефективності управління подіями в організації.

### **2.3. Аналіз вимог до системи. Модель випадків використання**

У цьому розділі наведено всі необхідні вимоги для розробки проекту. Для визначення вимог потрібно надати чітке і всебічне розуміння продуктів, які повинні бути розроблені. Крім того, тут описано всі дані, випадки використання та сценарії, модель даних, функціональні та нефункціональні вимоги, інтерфейси та моделі поведінки, обмеження та критерії перевірки, а також необхідну додаткову інформацію, яка є специфічною для системи, що реалізовується. Функціональні вимоги (Functional Requirements) - це вимоги, які описують функції або дії, які система повинна виконувати. Вони визначають, що система робить для користувача або для іншої системи. По суті, це опис поведінки системи в конкретних умовах

Аналіз вимог до системи — це повний опис поведінки системи, яка розробляється, і може включати набір випадків використання, які описують взаємодії користувачів із програмним забезпеченням. Крім того, він також містить нефункціональні вимоги.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Моделі випадків використання та проектування випадків використання для кожної ролі або актора описані нижче.

### 2.3.1. Модель випадків використання загального користувача

Модель випадків використання загального користувача показана на рисунку 2.1.



Рисунок 2.1 - Модель випадків використання загального користувача

Випадок використання "Реєстрація" загального користувача.

Масові люди можуть надсилати запит адміністратору на отримання статусу користувача системи.

Початковий покроковий опис:

- Якщо користувач не є авторизованим адміністратором, то він є загальним користувачем. Йому потрібно надіслати запит адміністратору для отримання статусу авторизованого користувача.
- Система надасть форму користувачеві для заповнення.
- Користувач заповнить форму, і адміністратору буде надіслано сповіщення.
- Коли адміністратор прийме запит, загальний користувач стане авторизованим користувачем.

### 2.3.2. Модель випадків використання адміністратора

Модель випадків використання адміністратора в системі показана на рисунку 2.2.

Випадок використання "Схвалення користувача" адміністратора:

Короткий опис

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Адміністратор схвалить або відхилить запити, які надходять від загальних користувачів.

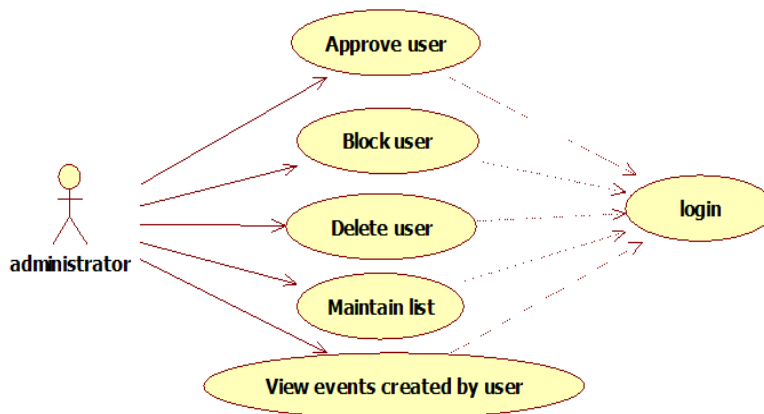


Рисунок 2.2 - Модель випадків використання адміністратора користувача

Початковий покроковий опис:

- Адміністратор може схвалити або відхилити будь-який запит, який надходить від загального користувача. Коли він схвалює або відхиляє запит, користувачеві надсилається лист із повідомленням про схвалення або відхилення.

- Адміністратор вибирає сповіщення для перегляду необроблених запитів.

- Адміністратору надається можливість або прийняти запит, або відхилити його.

- Адміністратор або приймає, або відхиляє запит. Після того, як адміністратор прийме запит, загальний користувач стане авторизованим користувачем системи.

*Випадок використання "Ведення списку" адміністратора*

Короткий опис

Адміністратор зможе додати будь-якого нового члена організації на основі груп. Він також має право видалити будь-якого члена зі списку, якщо цей член більше не є членом організації, або відновити когось із видалених.

Початковий покроковий опис:

- Адміністратор підтримує членів організації через цей випадок використання.

- Адміністратор вибирає кнопку "Додати нового учасника".

- Система надає адміністратору форму для заповнення.

- Адміністратор заповнює форму, надаючи інформацію про учасників.

Таким чином система зберігає цю інформацію для учасника.

Адміністратор може видалити будь-якого учасника, якщо він цього бажає. Видалені учасники зберігаються в опції "Попередній учасник". Вибравши цю опцію, адміністратор зможе відновити учасника за своїм бажанням.

*Випадок використання "Блокування користувача" адміністратора*

Короткий опис

Адміністратор зможе заблокувати будь-якого поточного користувача і також зможе розблокувати їх.

Початковий покроковий опис:

- Цей випадок використання надає адміністратору можливість заблокувати користувача, якщо його дії є шкідливими для організації.

- Адміністратор вибирає опцію "Поточні користувачі".

- Йому буде надано список поточних користувачів системи з можливістю їх заблокувати.

- Після вибору опції блокування користувач буде заблокований.

- Користувач буде збережений у списку заблокованих користувачів.

- Вибравши заблокованого користувача, адміністратору буде надано можливість розблокувати їх.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

- Коли адміністратор вибирає опцію розблокування, користувач знову стане поточним користувачем.

*Випадок використання "Видалення користувача" адміністратора*

Короткий опис

Адміністратор зможе видалити будь-якого поточного користувача або члена, якщо він більше не перебуває в організації, і також зможе відновити їх, якщо адміністратор видалив його помилково.

Початковий покроковий опис:

- Якщо користувач більше не перебуває в організації, його також буде видалено зі списку користувачів. Цей випадок використання надає адміністратору таку можливість.

- Адміністратор вибирає опцію "Поточні користувачі".

- Йому буде надано список поточних користувачів системи з можливістю їх видалити.

- Після вибору опції видалення користувач буде видалений.

- Користувач буде збережений у списку видалених користувачів.

- Вибравши видаленого користувача, адміністратору буде надано можливість відновити їх.

- Коли адміністратор вибирає опцію відновлення, користувач знову стане поточним користувачем.

*Випадок використання "Перегляду подій, створених користувачами" адміністратора*

Короткий опис

Адміністратор також може відстежувати, який користувач створює які події. Тому якщо він вважає, що будь-які події можуть зашкодити організації, він може заблокувати або видалити користувача.

Початковий покроковий опис:

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

- Адміністратору потрібно вибрати опцію користувача.
- Потім адміністратору потрібно вибрати опцію "Показати деталі".
- Система надасть інформацію про події, створені конкретним користувачем.

### 2.3.3. Модель випадків використання авторизованого користувача

Модель випадків використання авторизованого користувача показана на рисунку 2.3.

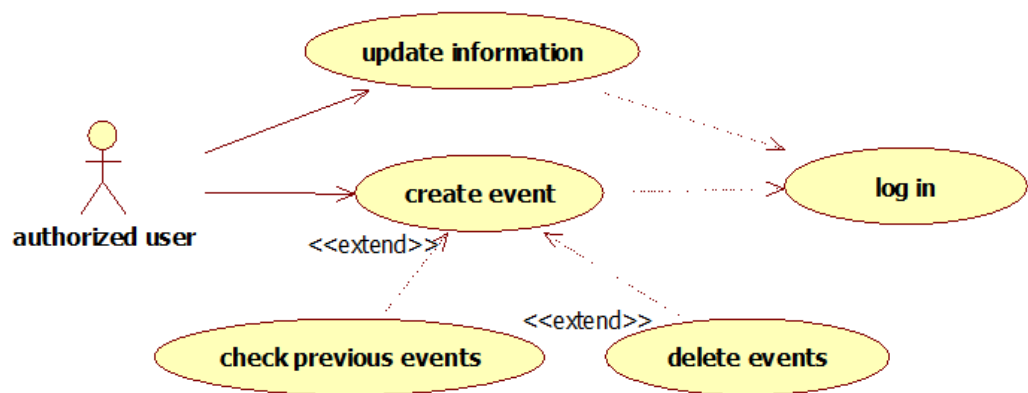


Рисунок 2.3 - Модель випадків використання авторизованого користувача

*Випадок використання "Оновлення інформації" авторизованого користувача*

Короткий опис

Оскільки користувач є членом організації, його інформація (ім'я, електронна пошта, номер контакту тощо) зберігається в базі даних. Він може оновлювати цю інформацію, оскільки його будуть повідомляти про будь-яку подію через його електронну пошту та номер контакту. Тому йому надається можливість змінювати цю інформацію, коли він цього потребує.

Початковий покроковий опис:

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

- Авторизований користувач вибирає опцію оновлення після входу в систему, надавши правильне ім'я користувача та пароль.
- Потім йому буде надано його інформація, яка зберігається в базі даних.
- Він може оновити будь-яку свою інформацію, якщо бажає.
- Після натискання кнопки "Надіслати" оновлена інформація буде збережена в базі даних.

*Випадок використання "Створення події" авторизованого користувача*

Короткий опис

Будь-який авторизований користувач може створити будь-яку подію в будь-який час.

Початковий покроковий опис:

- Авторизований користувач натискає кнопку "Створити подію".
- Потім з'являється сторінка, яка запитує інформацію про подію.
- Йому потрібно ввести інформацію про назву події, повідомлення, яке він хоче надіслати, а також дату та час, коли він хоче надіслати повідомлення.
- Після надіслання форми система запитає про отримувачів, показавши членів організації.
- Після вибору членів авторизованому користувачеві потрібно натиснути кнопку "Надіслати".
- Він також може змінити будь-яку інформацію про подію, якщо бажає, навіть після надіслання отримувачів.
- Після того, як він переконується в події, йому потрібно натиснути кнопку "Створити подію".

Таким чином буде створено подію, і повідомлення буде надіслано отримувачам у вказаний час і дату.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

*Випадок використання "Перевірки попередніх подій" авторизованого користувача*

Короткий опис

Користувач може перевірити попередні події, які він створив.

Початковий покроковий опис:

- Користувач натискає опцію подій після входу в систему.
- Йому буде показано події, які він створив раніше, з інформацією про те, що він надіслав і кому він надіслав повідомлення.

*Випадок використання "Видалення подій" авторизованого користувача*

Короткий опис

Користувачам надається можливість видалити будь-яку подію, якщо вони цього бажають.

Початковий покроковий опис:

- Авторизований користувач вибирає опцію подій.
- Система надає користувачеві події, які він вже створив, які вже були доставлені або запитані для доставки, з можливістю видалити будь-яку.
- Користувач може видалити будь-яку подію в будь-який час, коли він цього бажає.

## **2.4. Проектування варіантів використання системи**

Цей розділ представляє ключові варіанти використання (Use Cases) системи, описуючи взаємодію між основними акторами системи (неавторизованим користувачем, авторизованим користувачем, адміністратором) та самою системою. Опис кожного варіанта використання включає його назву, актора, передумови для виконання, основний потік подій та альтернативні сценарії.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

#### 2.4.1. Варіант використання: Реєстрація користувача

Актор: Неавторизований користувач (потенційний користувач, що є членом організації).

Передумова: Користувач не має активного облікового запису в системі та ініціює процес створення нового профілю.

Опис: Користувач надає необхідні персональні та контактні дані для створення облікового запису з метою отримання доступу до функціоналу системи після схвалення адміністратором.

Основний потік подій:

1. Користувач отримує доступ до інтерфейсу реєстрації.
2. Користувач вводить ідентифікатор користувача (логін).
3. Користувач вводить інформацію про відділ.
4. Користувач вводить посаду.
5. Користувач вводить контактний номер телефону.
6. Користувач вводить адресу електронної пошти (e-mail).
7. Користувач вводить пароль.
8. Користувач повторно вводить пароль для підтвердження.
9. Користувач вводить відповідь на секретне запитання (для відновлення доступу).
10. Користувач ініціює процес реєстрації, натискаючи кнопку "Зареєструватися".
11. Система валідує введені дані та створює запит на реєстрацію нового користувача.
12. Система інформує користувача про успішне відправлення запиту.

Альтернативні сценарії:

1. Некоректний формат даних: Система виявляє некоректний формат введеної інформації (наприклад, невалідний e-mail або номер телефону). Система генерує повідомлення про помилку та вимагає повторного введення даних.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Неспівпадіння паролів: Введені паролі у полях не співпадають. Система генерує повідомлення про помилку та вимагає повторного введення пароля.

3. Відсутність обов'язкових даних: Користувач не надав дані для одного або декількох обов'язкових полів (ідентифікатор користувача, відділ, посада, контактний номер, e-mail). Система генерує повідомлення про помилку з переліком відсутніх даних та вимагає їх надання.

#### *2.4.2. Варіант використання: Схвалення користувача*

Актор: Адміністратор.

Передумова: Користувач має статус Адміністратора та успішно аутентифікований в системі. У системі є запити на реєстрацію від нових користувачів.

Опис: Адміністратор переглядає запити на реєстрацію нових користувачів та приймає рішення про їх схвалення, надаючи їм статус авторизованого користувача.

Основний потік подій:

1. Адміністратор отримує сповіщення про наявність нових запитів на реєстрацію.
2. Адміністратор обирає сповіщення або переходить до відповідного розділу управління користувачами.
3. Адміністратор переглядає деталі запиту на реєстрацію.
4. Адміністратор ініціює схвалення облікового запису, натискаючи кнопку "Схвалити користувача".
5. Система змінює статус користувача на "авторизований" та надає йому відповідні права доступу.
6. Система інформує користувача, який надіслав запит, про його схвалення (наприклад, автоматичним email).

Альтернативний сценарій:

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Відсутність нових запитів: У системі відсутні нерозглянуті запити на реєстрацію. Система не відображає відповідних сповіщень або розділ запитів є порожнім.

#### *2.4.3. Варіант використання: Блокування користувача*

Актор: Адміністратор.

Передумова: Користувач має статус Адміністратора та успішно аутентифікований в системі.

Опис: Адміністратор обмежує доступ авторизованого користувача до системи шляхом його блокування.

Основний потік подій:

1. Адміністратор отримує доступ до списку активних користувачів системи (натискаючи кнопку "Поточні користувачі" або аналогічний елемент інтерфейсу).

2. Адміністратор обирає зі списку користувача, якого необхідно заблокувати.

3. Адміністратор ініціює дію блокування, натискаючи кнопку "Заблокувати користувача".

4. Система змінює статус користувача на "заблокований" та обмежує його можливість входу та використання функціоналу.

#### *2.4.4. Варіант використання: Видалення користувача*

Акто: Адміністратор.

Передумова: Користувач має статус Адміністратора та успішно аутентифікований в системі.

Опис: Адміністратор остаточно видаляє обліковий запис користувача із системи.

Основний потік подій:

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Адміністратор отримує доступ до списку користувачів системи (натискаючи кнопку "Поточні користувачі" або аналогічний елемент інтерфейсу).

2. Адміністратор обирає зі списку користувача, якого необхідно видалити.

3. Адміністратор ініціює дію видалення, натискаючи кнопку "Видалити користувача".

4. Система видаляє обліковий запис користувача та пов'язані з ним дані (з урахуванням політик цілісності даних та архівів).

#### *2.4.5. Варіант використання: Додавання члена організації до внутрішнього списку*

Актор: Адміністратор.

Передумова: Користувач має статус Адміністратора та успішно аутентифікований в системі. Цей сценарій може використовуватися для ручного додавання членів до бази даних системи, наприклад, для формування списків розсилки, незалежно від їх статусу реєстрації як користувачів системи.

Опис: Адміністратор вручну вносить дані про члена організації до внутрішньої бази даних системи.

Основний потік подій:

1. Адміністратор отримує доступ до інтерфейсу управління членами організації (натискаючи кнопку "Члени" або аналогічний елемент інтерфейсу).

2. Адміністратор ініціює операцію додавання нового члена, натискаючи кнопку "Вставити членів" або аналогічний елемент.

3. Адміністратор вводить Ім'я члена.

4. Адміністратор вводить Прізвище члена.

5. Адміністратор вводить Відділ члена.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

6. Адміністратор вводить Посаду члена.
7. Адміністратор вводить Контактний номер телефону члена.
8. Адміністратор вводить адресу електронної пошти (e-mail) члена.
9. Адміністратор ініціює збереження даних, натискаючи кнопку "Вставити" або аналогічний елемент.
10. Система валідує введені дані та додає нового члена до внутрішнього списку.

Альтернативні сценарії:

1. Некоректний формат електронної пошти: Система виявляє некоректний формат введеної адреси електронної пошти. Система генерує повідомлення про помилку та вимагає повторного введення коректних даних.
2. Відсутність обов'язкових даних: Адміністратор не надав дані для одного або декількох обов'язкових полів (Ім'я, Відділ, Посада, Контактний номер, e-mail). Система генерує повідомлення про помилку з переліком відсутніх даних та вимагає їх надання.

*2.4.6. Варіант використання: Перегляд подій, створених конкретним користувачем*

Актор: Адміністратор.

Передумова: Користувач має статус Адміністратора та успішно аутентифікований в системі.

Опис: Адміністратор переглядає список усіх подій, які були створені певним авторизованим користувачем.

Основний потік подій:

1. Адміністратор отримує доступ до списку користувачів системи (натискаючи кнопку "Користувачі" або аналогічний елемент інтерфейсу).
2. Адміністратор обирає конкретного користувача зі списку.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

3. Адміністратор ініціює запит на перегляд подій, створених цим користувачем (натискаючи посилання "Показати деталі" або аналогічний елемент).

4. Система отримує дані про події, пов'язані з обраним користувачем, та відображає їх перелік.

Альтернативний сценарій:

1. Відсутність створених подій: Обраний користувач не створив жодної події в системі. Система генерує відповідне інформаційне повідомлення (наприклад, у спливаючому вікні), вказуючи на відсутність створених подій для даного користувача.

#### *2.4.7. Варіант використання: Оновлення персональної інформації*

Актор: Авторизований користувач (користувач, схвалений адміністратором).

Передумова: Користувач успішно аутентифікований в системі та має статус авторизованого користувача.

Опис: Авторизований користувач переглядає та, за необхідності, редагує власні персональні дані, що зберігаються в системі.

Основний потік подій:

1. Користувач обирає пункт меню або елемент інтерфейсу, пов'язаний з управлінням власним обліковим записом або профілем ("опція облікового запису" або аналогічно).

2. Система отримує запит на доступ до персональних даних та відображає поточну інформацію користувача.

3. Користувач вносить зміни до полів, доступних для редагування (наприклад, контактний номер, відділ, посада, e-mail).

4. Користувач ініціює збереження внесених змін, натискаючи кнопку "Оновити" або аналогічний елемент.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

5. Система валідує змінені дані та оновлює інформацію про користувача в базі даних.

#### 2.4.8. *Варіант використання: Створення події та планування повідомлення*

Актор: Авторизований користувач.

Передумова: Користувач успішно аутентифікований в системі та має статус авторизованого користувача, що надає йому право ініціювати створення подій.

Опис: Авторизований користувач створює нову подію в системі, вказує деталі повідомлення та визначає цільову групу отримувачів для автоматизованої розсилки.

Основний потік подій:

1. Користувач ініціює процес створення нової події (натискаючи кнопку "Створити подію" або аналогічний елемент інтерфейсу).
2. Система надає інтерфейс для введення даних про подію.
3. Користувач вводить назву події.
4. Користувач вводить контент повідомлення, що буде розіслано.
5. Користувач вводить дату запланованої розсилки повідомлення.
6. Користувач вводить час запланованої розсилки повідомлення.
7. Користувач переходить до наступного етапу конфігурації (натискаючи кнопку "Далі" або аналогічний елемент).
8. Система відображає список доступних відділів організації.
9. Користувач вибирає один або кілька відділів зі списку для визначення потенційних отримувачів.
10. Користувач підтверджує вибір відділів (натискаючи кнопку "Вибрати" або аналогічний елемент).
11. Система відображає перелік осіб, що належать до обраних відділів, як потенційних отримувачів.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Користувач вибирає конкретних отримувачів повідомлення зі списку (реалізація цільової локалізації повідомлення).

13. Користувач підтверджує вибір отримувачів (натискаючи кнопку "Вибрано" або аналогічний елемент).

14. Користувач завершує конфігурацію події та ініціює її створення в системі (натискаючи кнопку "Створити подію" або аналогічний елемент).

15. Система валідує введені дані, зберігає інформацію про подію та планує автоматизовану розсилку повідомлення у визначений час визначеним отримувачам.

Альтернативні сценарії:

1. Відсутність обов'язкових даних події: Користувач не надав дані для одного або декількох обов'язкових полів (назва події, повідомлення, дата або час). Система генерує повідомлення про помилку, що вимагає надання повної інформації.

2. Відсутність вибору відділу: Користувач не вибрав жодного відділу на етапі 9. Система генерує повідомлення про помилку та вимагає зробити вибір.

3. Відсутність вибору отримувача: Користувач не вибрав жодного конкретного отримувача зі списку на етапі 12. Система генерує повідомлення про помилку та вимагає зробити вибір.

#### *2.4.9. Варіант використання: Перегляд власних попередніх подій*

Актор: Авторизований користувач.

Передумова: Користувач успішно аутентифікований в системі та має статус авторизованого користувача.

Опис: Авторизований користувач отримує доступ до переліку подій, які він створив у системі раніше.

Основний потік подій:

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

1. Користувач обирає пункт меню або елемент інтерфейсу, пов'язаний з доступом до списку подій ("опція подій" або аналогічно).

2. Система отримує запит та відображає перелік усіх подій, створених даним користувачем, з основними деталями.

Альтернативний сценарій:

2.1. Відсутність створених подій: Даний користувач ще не створив жодної події в системі. Система відображає відповідне інформаційне повідомлення, що вказує на відсутність створених ним подій.

#### *2.4.10. Варіант використання: Видалення власних подій*

Актор: Авторизований користувач.

Передумова: Користувач успішно аутентифікований в системі та має статус авторизованого користувача. Користувач має створені події в системі.

Опис: Авторизований користувач видаляє одну або кілька подій, які він створив.

Основний потік подій:

1. Користувач отримує доступ до переліку своїх створених подій (як у варіанті використання "Перегляд власних попередніх подій").

2. Користувач обирає зі списку подію, яку необхідно видалити.

3. Користувач ініціює дію видалення, натискаючи кнопку "Видалити подію" або аналогічний елемент, пов'язаний з конкретною подією.

4. Система запитує підтвердження дії (опціонально).

5. Система видаляє обрану подію та пов'язані з нею дані.

### **2.5. Технічні вимоги та характеристики системи**

Система призначена для взаємодії з користувачами, які можуть мати різні ролі та рівні доступу: Адміністратор, Авторизований користувач та Неавторизований користувач (потенційний користувач). Відповідно до

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

архітектури системи та обраного способу реалізації (веб-додаток), до всіх категорій користувачів висувається мінімальний рівень вимог щодо цифрової грамотності.

Очікується, що користувачі системи володітимуть базовими навичками роботи з персональним комп'ютером або мобільним пристроєм та матимуть досвід використання стандартного функціоналу веб-браузерів. Це включає здатність здійснювати навігацію веб-сторінками, вводити дані у форми, використовувати інтерактивні елементи інтерфейсу (кнопки, посилання, випадаючі списки тощо). Інтерфейс системи розроблено таким чином, щоб бути інтуїтивно зрозумілим для осіб з цим мінімальним рівнем компетенції, не вимагаючи спеціалізованих технічних знань.

Проектування та розробка інтегрованої системи управління подіями з функціоналом обміну повідомленнями базується на використанні певного стеку технологій та інструментів, обраних з урахуванням вимог до функціональності, масштабованості (в перспективі) та зручності розробки. Технічні вимоги до реалізації включають використання наступних ключових компонентів:

Технології розробки:

- Фреймворк: Codeigniter (на базі PHP). Використання даного фреймворку забезпечує структурування коду згідно з патерном Модель-Вид-Контролер (MVC), що сприяє модульній організації проекту, полегшує його підтримку та розширення.

- Фронтенд-технології: HTML, CSS, JavaScript. Ці технології застосовуються для розробки клієнтської частини веб-додатку, створення інтерфейсу користувача та реалізації інтерактивних елементів.

Інтегроване середовище розробки (IDE):

- NetBeans: Використано як основний інструмент для написання, відлагодження та управління програмним кодом проекту.

Система управління базами даних (СУБД):

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

- MySQL: Обрано для забезпечення постійного зберігання, структурування та управління всіма даними системи (інформація про користувачів, події, повідомлення, списки отримувачів тощо).

Середовище веб-сервера для розробки та тестування:

- WampServer: Використовується як локальне середовище розробки, що інтегрує веб-сервер Apache, СУБД MySQL та інтерпретатор PHP, необхідні для роботи веб-додатку.

Отже, в розділі було представлено ключові аспекти проектування системи. Детально обговорено системні вимоги до функціоналу, описано модель даних, яка ляже в основу структури бази даних, а також визначено типи та характеристики користувачів, що будуть взаємодіяти із системою. Крім того, в заключній частині розділу були описані технічні засоби та інструменти розробки, обрані для реалізації даного програмного продукту. Викладені положення формують концептуальну та технічну основу для подальших етапів розробки та впровадження системи.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

## РОЗДІЛ 3. ПРОЕКТУВАННЯ МОДЕЛІ БАЗИ ДАНИХ І ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ОБМІНУ ПОВІДОМЛЕННЯМИ РІЗНОГО РІВНЯ ЛОКАЛІЗАЦІЇ

### 3.1. Проектування моделі бази даних

Етап проектування (Design Phase) у процесі розробки програмного забезпечення є критично важливим для формування архітектурних рішень, які визначають загальну структуру програмного комплексу, модель організації даних та специфікацію користувацьких інтерфейсів. На цій стадії особлива увага приділяється проектуванню структури бази даних, оскільки структурні недоліки або помилки, допущені на етапі моделювання даних, можуть виявитися надзвичайно витратними для виправлення на подальших стадіях реалізації проекту та суттєво обмежувати масштабованість і продуктивність системи.

В рамках проектування системи обміну повідомленнями різного рівня локалізації була розроблена модель даних, яка визначає структуру зберігання всієї інформації, необхідної для функціонування системи. Ця модель представлена у вигляді ER-діаграми (Сутність-Зв'язок), зображеної на рисунку 3.1.

Відповідно до моделі, представленої на рисунку 3.1, база даних системи ІІІ побудована навколо трьох основних сутностей, реалізованих у вигляді реляційних таблиць: members, requests та events.

Сутність members (Члени) призначена для зберігання вичерпної інформації про членів організації, які є потенційними користувачами системи або можуть виступати в ролі отримувачів повідомлень. Таблиця members містить наступні ключові атрибути: унікальний ідентифікатор члена (member\_id), ім'я (first\_name), прізвище (last\_name), пароль (password – ймовірно, у хешованому форматі), контактний номер телефону

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

(contact\_number), адреса електронної пошти (email), посада (position), відділ (department), поточний статус члена в організації (status) та статус користувача в системі (user\_status – наприклад, неавторизований, авторизований, заблокований). Записи до цієї таблиці додаються адміністратором при ручному веденні списку членів організації у системі.

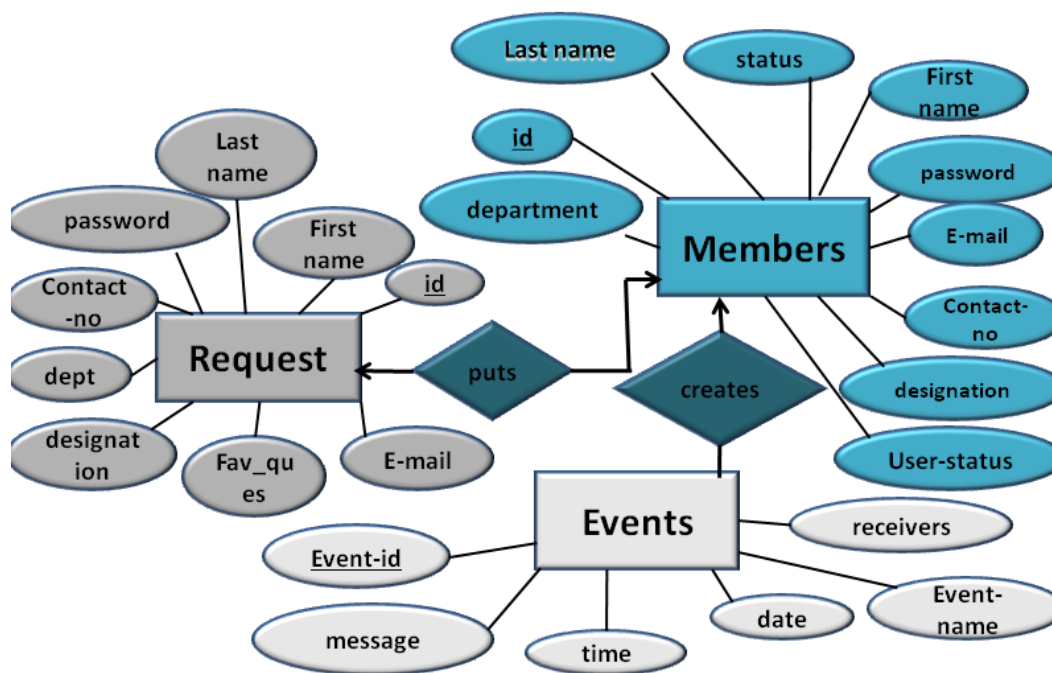


Рисунок 3.1 – ER діаграма

Сутність requests (Запити) використовується як тимчасове сховище даних для користувачів, які ініціювали процес реєстрації в системі, але їхні запити ще не були розглянуті та схвалені адміністратором. Таблиця requests включає атрибути: ідентифікатор запиту (id), ім'я (first\_name), прізвище (last\_name), пароль (password), контактний номер телефону (contact\_number), відділ (department), посада (position), відповідь на секретне запитання для відновлення доступу (favorite\_question), та адреса електронної пошти (email). Нові записи додаються до цієї таблиці щоразу, коли неавторизований користувач надсилає запит на отримання статусу авторизованого користувача через відповідний інтерфейс реєстрації.

Сутність events (Події) призначена для зберігання структурованої інформації про заплановані події, створені авторизованими користувачами системи, та асоційовані з ними повідомлення для розсилки. Таблиця events містить атрибути: унікальний ідентифікатор події (event\_id), назва події (event\_name), контент повідомлення, що має бути розіслано (message), запланований час (time) та дата (date) автоматизованої розсилки, а також інформація про отримувачів (recipients). Остання атрибутика, ймовірно, зберігає посилання або ідентифікатори користувачів чи груп, що мають отримати повідомлення, що є критично важливим для реалізації функціоналу обміну повідомленнями різного рівня локалізації. Новий запис додається до цієї таблиці щоразу, коли авторизований користувач успішно створює нову подію через інтерфейс системи.

Описана модель даних формує основу для організації інформаційних потоків та зберігання даних у системі, забезпечуючи підтримку її основного функціоналу.

### 3.2. Реалізація схеми бази даних

Етап реалізації (Implementation Phase) є наступним після проектування у життєвому циклі розробки програмного забезпечення. На цій стадії відбувається безпосереднє втілення розробленої архітектури та дизайну у функціонуючий програмний продукт. Етап реалізації охоплює процеси кодування, тестування, інтеграції компонентів, а також розгортання системи на відповідному програмно-апаратному середовищі.

На основі логічної моделі даних, розробленої на етапі проектування (представленої у вигляді ER-діаграми), була реалізована фізична схема бази даних. Ця схема визначає конкретні таблиці, їхню структуру (імена полів, типи даних, розміри, обмеження цілісності) у вибраній системі управління базами даних (MySQL).

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Схема бази даних системи представлена на рисунку 3.2.

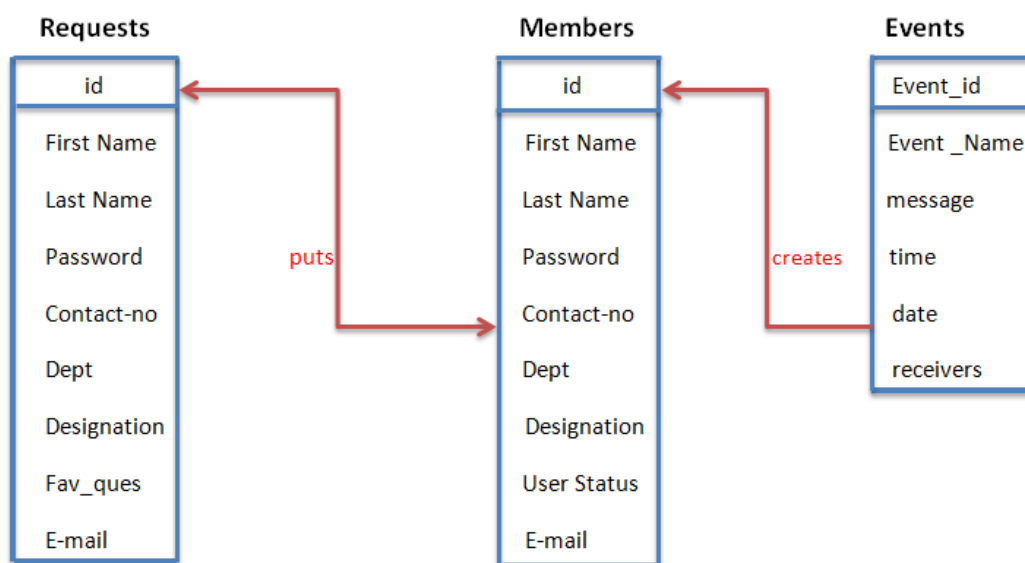


Рисунок 3.2 - Схема бази даних

Згідно з реалізованою схемою, база даних системи складається з трьох основних таблиць: events, members та requests. Структура кожної з цих таблиць детально описана нижче.

Таблиця members (рисунок 3.3) призначена для зберігання інформації про членів організації.

FIELD	TYPE	SIZE	CONSTRAINTS
<b>ID</b>	INT	1000	PRIMARY KEY
<b>FIRSTNAME</b>	VARCHAR	30	
<b>LASNAME</b>	VARCHAR	20	
<b>DEPARTMENT</b>	VARCHAR	20	
<b>DESIGNATION</b>	VARCHAR	20	
<b>CONTACT-NO</b>	VARCHAR	11	
<b>E-MAIL</b>	VARCHAR	20	
<b>USER STATUS</b>	INT	1	
<b>STATUS</b>	INT	1	
<b>PASSWORD</b>	VARCHAR	32	

Рисунок 3.3 - Структура таблиці members

Таблиця requests (рисунок 3.4) використовується для тимчасового зберігання інформації про запити користувачів на реєстрацію.

FIELD	TYPE	SIZE	CONSTRAINTS
ID	INT	1000	PRIMARY KEY
FIRSTNAME	VARCHAR	30	
LASNAME	VARCHAR	20	
DEPARTMENT	VARCHAR	20	
DESIGNATION	VARCHAR	20	
CONTACT-NO	VARCHAR	11	
E-MAIL	VARCHAR	20	
FAVORITE QUESTION	VARCHAR	30	
PASSWORD	VARCHAR	32	

Рисунок 3.4 - Структура таблиці requests

Таблиця events (рисунок 3.5) призначена для зберігання інформації про створені події та дані, необхідні для розсилки повідомлень.

FIELD	TYPE	SIZE	CONSTRAINTS
EVENT ID	INT	1000	PRIMARY KEY
EVENT NAME	VARCHAR	30	
MESSAGE	VARCHAR	200	
TIME	TIME		
DATE	DATE		
RECEIVERS	VARCHAR	200	
CREATOR	VARCHAR	30	

Рисунок 3.5 - Структура таблиці events

Реалізована схема бази даних є основою для функціонування системи, забезпечуючи зберігання та доступ до всіх необхідних даних, що використовуються серверною логікою та клієнтським інтерфейсом.

### 3.3. Модель взаємодії з графічним інтерфейсом користувача

Розробка ефективного та інтуїтивно зрозумілого графічного інтерфейсу користувача (GUI) є важливим етапом проектування системи, оскільки саме через GUI відбувається вся взаємодія користувачів із програмним продуктом. У даному підрозділі описано ключові елементи інтерфейсу та сценарії взаємодії користувачів на початкових етапах роботи із системою, зокрема, головну сторінку, інтерфейси аутентифікації та реєстрації. Загальна модель взаємодії з GUI може бути представлена схематично, як показано на Рисунку 3.6.

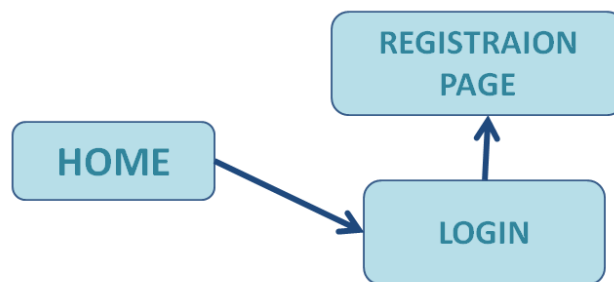


Рисунок 3.6 - Модель взаємодії графічного інтерфейсу користувача (GUI) головної сторінки

#### 3.3.1. Головна сторінка системи

Ініціація взаємодії користувача із системою відбувається шляхом доступу до веб-адреси додатку, що призводить до завантаження головної сторінки. Ця сторінка слугує точкою входу до системи та надає первинні навігаційні опції. Як проілюстровано на рисунку 3.7, типова структура головної сторінки включає верхню панель навігації, де розміщені посилання на основні розділи.

Згідно з дизайном, на головній сторінці візуально представлені ключові елементи навігації: посилання на саму "Головну" сторінку (зазвичай використовується для повернення на стартову сторінку) та опція "Вхід"

(Login). Опція "Вхід" призначена для ініціації процесу аутентифікації користувача. Важливо зазначити, що після успішного проходження процедури аутентифікації, користувачі з різними ролями (Адміністратор та Авторизований користувач) отримують доступ до різних адаптованих інтерфейсів та наборів функціональних можливостей, що відповідають їхнім правам доступу.

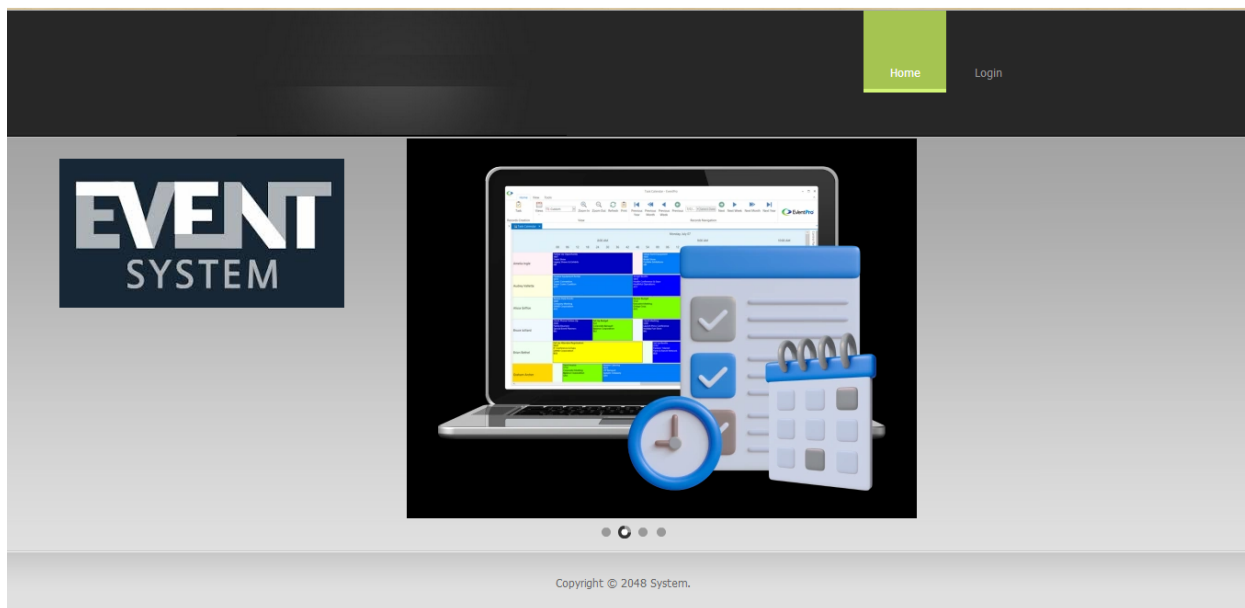


Рисунок 3.7 – Головна сторінка

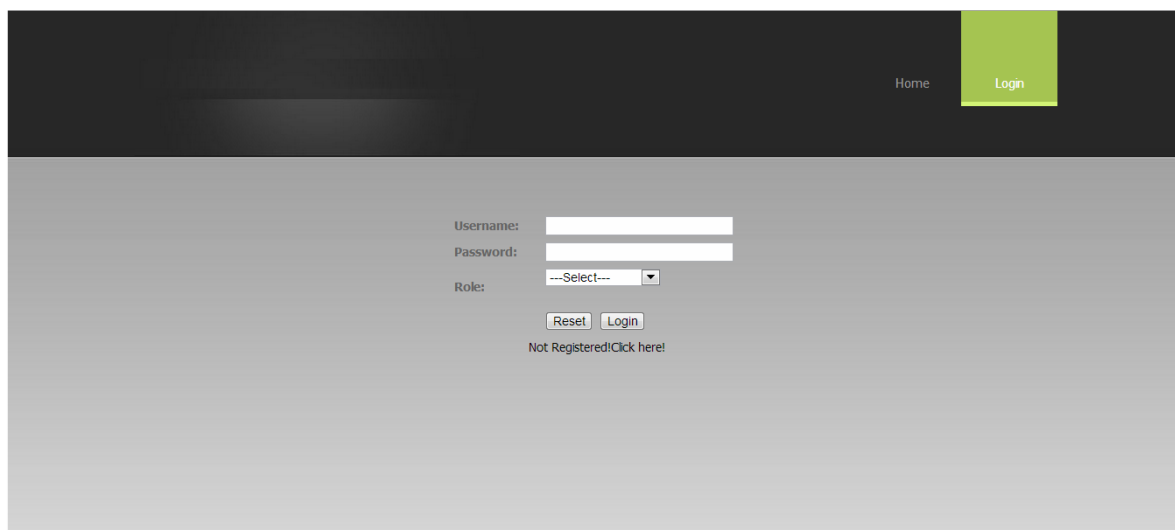
### 3.3.2. Сторінка аутентифікації (входу)

Сторінка аутентифікації є ключовим інтерфейсом для доступу до захищених функціональних можливостей системи. Перехід на цю сторінку здійснюється шляхом вибору опції "Вхід" на Головній сторінці. Інтерфейс Сторінки входу (рисунок 3.8) містить стандартизовану форму для збору облікових даних від користувача. Ця форма, як правило, вимагає введення Імені користувача (Username), Пароля (Password) та, як вказано у даній моделі, вибору Ролі користувача перед спробою аутентифікації.

Крім полів для введення облікових даних, Сторінка входу також надає посилання для користувачів, які ще не мають облікового запису в системі,

					БР.ІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

скеровуючи їх до інтерфейсу реєстрації. Процес аутентифікації включає валідацію введених даних на сервері.



The screenshot shows a login form on a dark-themed website. At the top right, there are two navigation links: "Home" and "Login". The "Login" link is highlighted with a green background. The login form consists of three input fields: "Username:" with a white text box, "Password:" with a white text box, and "Role:" with a dropdown menu showing "--Select--". Below the input fields are two buttons: "Reset" and "Login". At the bottom of the form, there is a link that says "Not Registered! Click here!".

Рисунок 3.8 – Сторінка входу

У випадку введення некоректної комбінації Імені користувача та Пароля, система реагує генеруванням відповідного повідомлення про помилку (наприклад, "Неправильний пароль або ім'я користувача"), яке може відобразитися користувачеві у вигляді спливаючого вікна або діалогового вікна, як ілюстровано на рисунку 3.9.

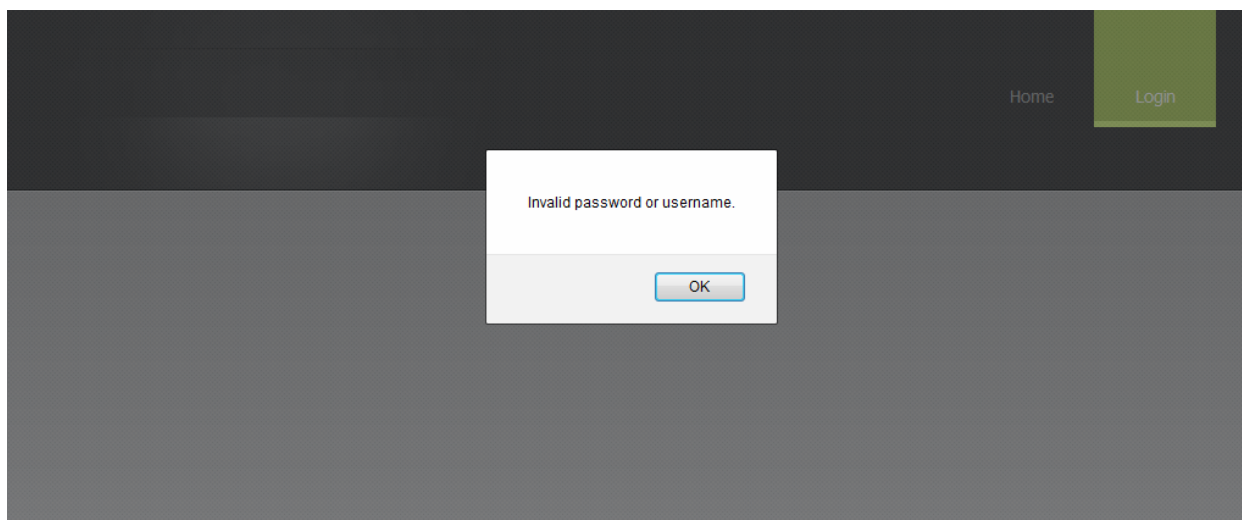


Рисунок 3.9 - Сторінка входу з діалоговим вікном сповіщення

### 3.3.3. Сторінка реєстрації користувача

Інтерфейс реєстрації призначений для збору інформації від нових користувачів, які бажають отримати доступ до функціоналу системи. Перехід до Сторінки реєстрації здійснюється зазвичай за відповідним посиланням зі Сторінки входу. Як показано на рисунку 3.10, ця сторінка містить розширену форму для введення персональних, контактних даних та інформації для створення облікового запису.

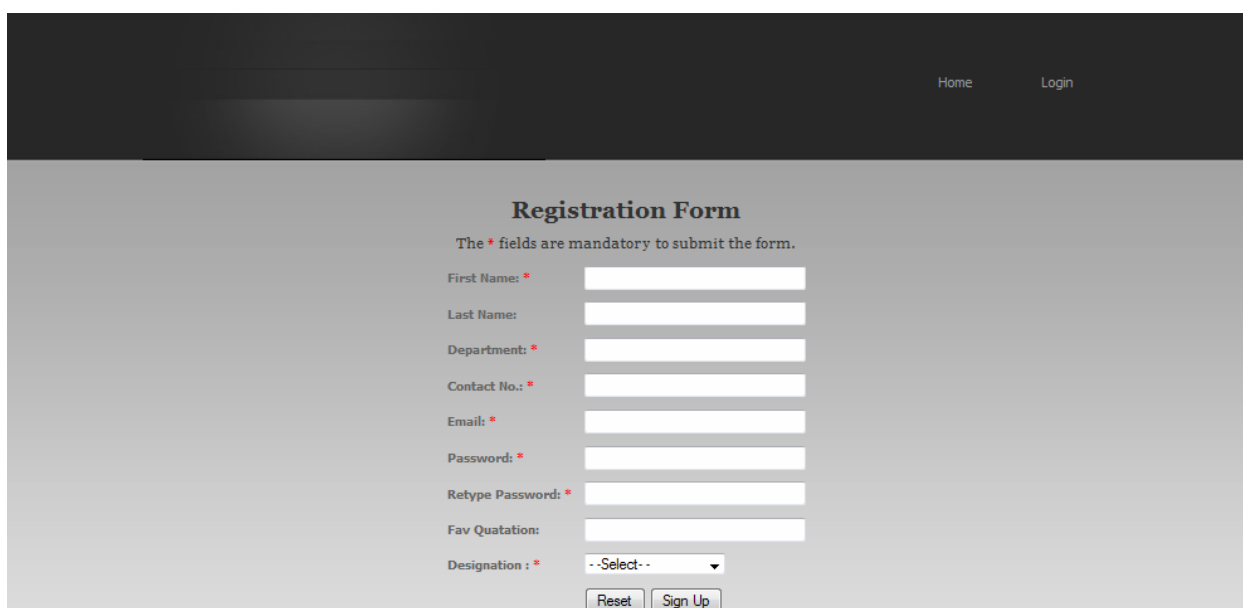


Рисунок 3.9 – Сторінка реєстрації користувача

Форма реєстрації включає поля для введення: імені, відділу, посади, контактному номеру телефону, адреси електронної пошти та пароля. Усі ці поля визначені як обов'язкові для заповнення, що забезпечує збір мінімально необхідного набору даних про потенційного користувача. Після заповнення всіх необхідних полів, користувач ініціює відправлення форми шляхом натискання кнопки "Надіслати". Надіслана інформація про нового користувача передається до системи та надходить до адміністратора для подальшого розгляду та схвалення запиту на реєстрацію, як це описано у варіанті використання "Схвалення користувача". Система здійснює валідацію заповнення обов'язкових полів на клієнтській або серверній стороні; у

					БР.ІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

випадку пропуску одного або декількох обов'язкових полів, відправлення форми блокується або користувачеві надається відповідне повідомлення з вимогою надати повну інформацію.

### 3.4. Модель взаємодії з графічним інтерфейсом адміністратора

Архітектура адміністративного інтерфейсу користувача (GUI) для взаємодії з моделлю системи представлена на рисунку 3.10. Нижче наведено детальний опис компонентів та функціональних можливостей кожного графічного інтерфейсу, доступного адміністратору.

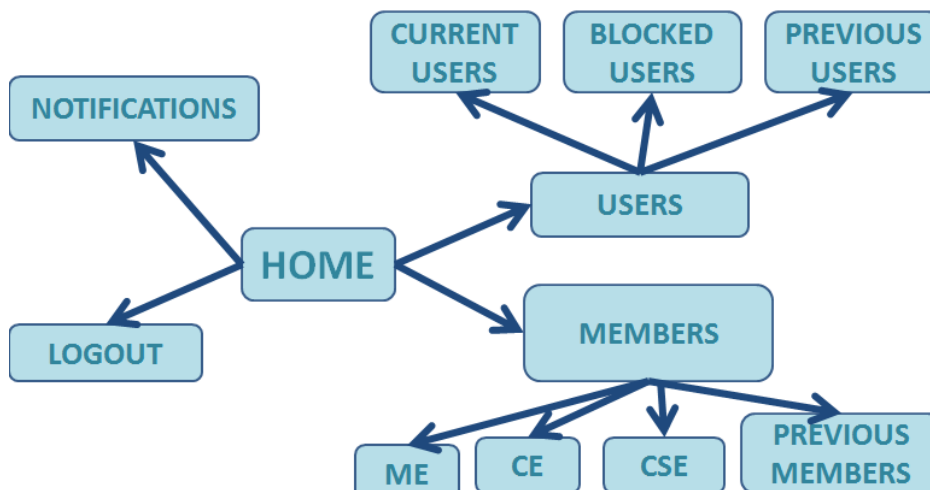


Рисунок 3.10 - Взаємодія моделі GUI головної сторінки адміністратора

#### *Головна сторінка адміністратора*

Основний інтерфейс адміністратора при вході в систему. Він забезпечує доступ до чотирьох ключових розділів: управління членами (Members), управління користувачами (Users), перегляд сповіщень (Notifications) та функція виходу із системи (Logout). Цей вид є початковою точкою взаємодії для адміністративного персоналу.

### Сторінка користувачів (Users Page)

Ця сторінка надає адміністратору функціонал для моніторингу та керування обліковими записами користувачів системи. Вона структурована на три підрозділи: "Поточні користувачі" (Current Users), "Заблоковані користувачі" (Blocked Users) та "Попередні користувачі" (Previous Users). Розділ "Поточні користувачі" відображає список активних облікових записів із відповідною інформацією. На цій вкладці адміністратор має можливість ініціювати блокування або видалення обраного користувача. Розділ "Заблоковані користувачі" містить перелік облікових записів, які були заблоковані адміністратором. У випадку відсутності заблокованих користувачів система відображає відповідне інформаційне повідомлення. Розділ "Попередні користувачі" містить дані облікових записів, які були видалені адміністратором.

Графічний інтерфейс сторінки користувачів проілюстровано на рисунку 3.11.

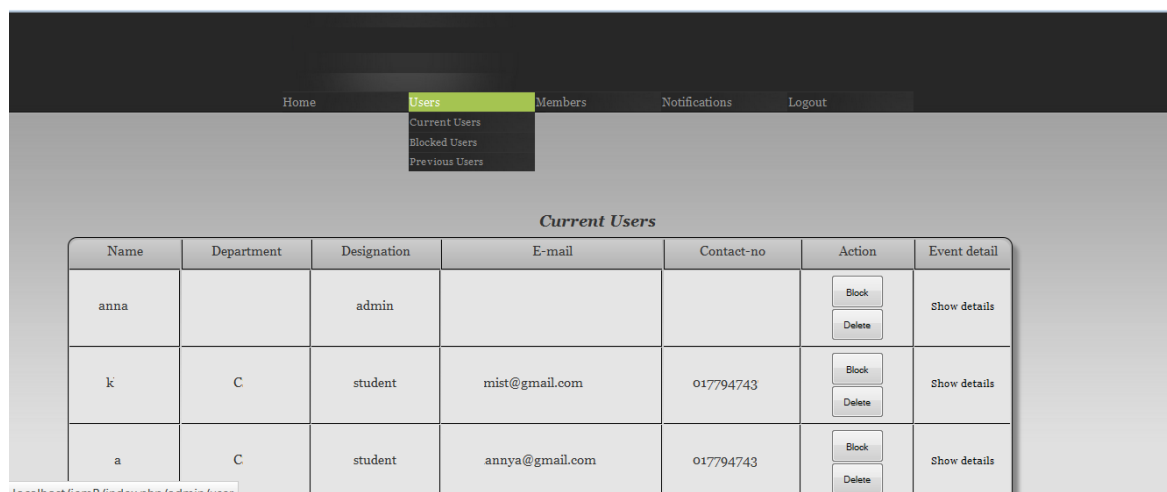


Рисунок 3.11 - Сторінка користувачів

### Сторінка членів (Members Page)

Ця сторінка призначена для управління списком членів системи. Адміністратору доступні функції додавання нових членів, перегляду членів, організованих за відділами, а також видалення існуючих членів зі списку.

Структура цієї сторінки показана на рисунку 3.12.

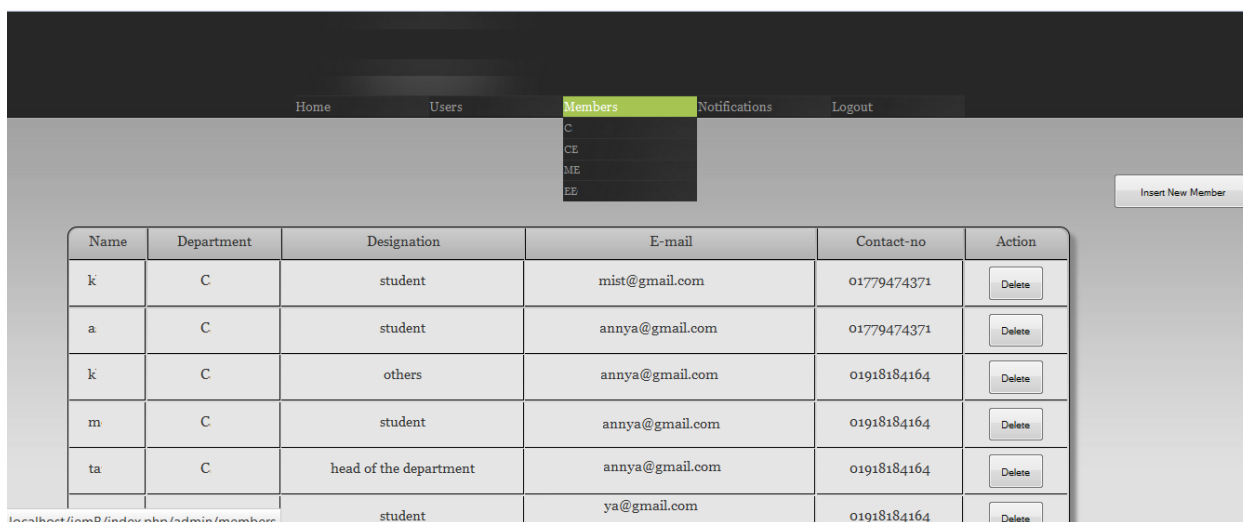


Рисунок 3.12 - Сторінка з переліком членів

Як видно на рисунку 3.12, у правому верхньому куті сторінки розташована кнопка "Вставити нового члена" (Insert New Member). Активація цієї кнопки ініціює відображення форми для введення даних нового члена системи.

Графічний інтерфейс форми для додавання нових членів представлено на рисунку 3.13.

Рисунок 3.13 - Сторінка вставки нових членів

### Сторінка сповіщень (Notifications Page)

Цей розділ акумулює сповіщення, що надходять від загальних користувачів системи. Графічний інтерфейс сторінки сповіщень відображено на рисунку 3.14.

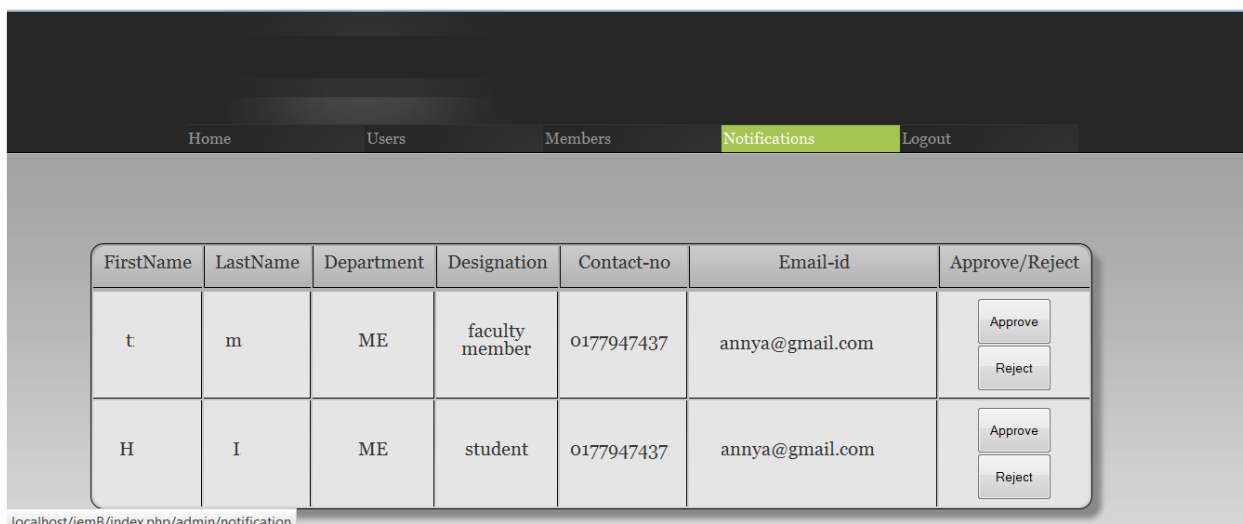


Рисунок 3.14 - Сторінка сповіщень

### 3.5. Модель взаємодії з графічним інтерфейсом авторизованого користувача

Модель графічного інтерфейсу користувача (GUI) для авторизованого користувача представлена на рисунку 3.15. Цей інтерфейс забезпечує доступ до функціоналу системи, доступного зареєстрованим користувачам.

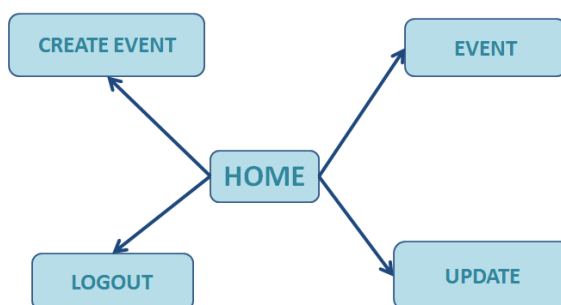


Рисунок 3.15 - Взаємодія моделі GUI головної сторінки авторизованого користувача

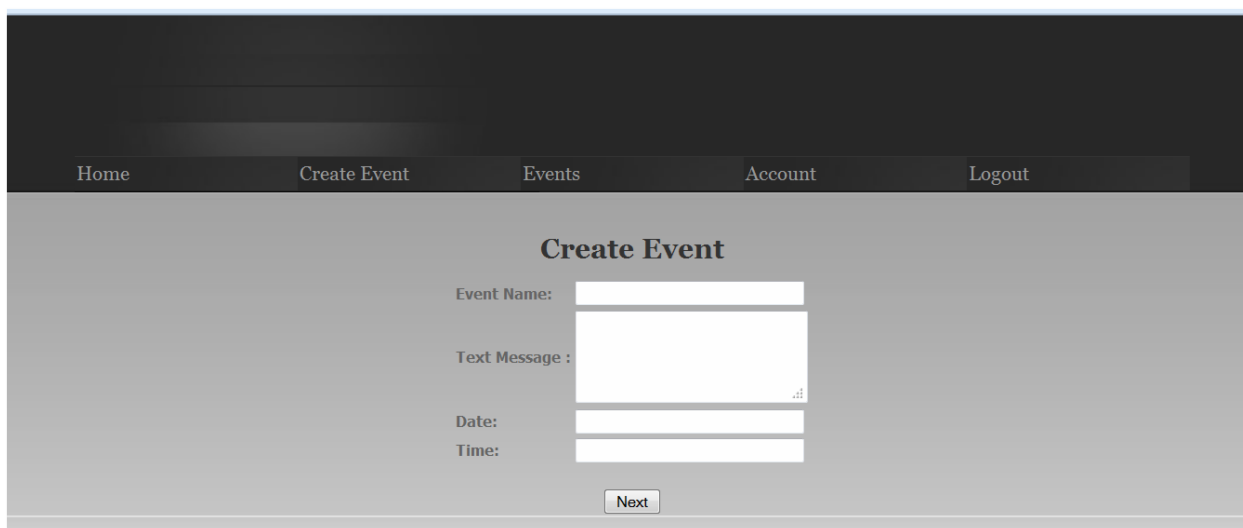
### *Головна сторінка авторизованого користувача*

Цей інтерфейс є основною точкою входу для авторизованого користувача та надає доступ до трьох ключових функціональних розділів: створення подій (Event Creation), перегляд створених подій (Events) та управління обліковим записом (Account Update). Опис кожного з цих розділів наведено нижче.

### *Сторінка створення події (Event Creation Page)*

Цей інтерфейс дозволяє авторизованому користувачу ініціювати створення нової події. Для цього користувач повинен ввести необхідні атрибути події, які включають назву події, текстове повідомлення, час, дату та перелік отримувачів, яким має бути надіслано сповіщення.

Структура сторінки створення події показана на рисунку 3.16.



The screenshot shows a web application interface for creating an event. At the top, there is a dark navigation bar with the following links: Home, Create Event, Events, Account, and Logout. Below the navigation bar, the main content area is titled "Create Event". It contains a form with the following fields: "Event Name:" with a text input field, "Text Message:" with a larger text area, "Date:" with a date input field, and "Time:" with a time input field. Below these fields is a "Next" button.

Рисунок 3.16 - Сторінка створення події

### *Сторінка подій (Events Page)*

Цей розділ системи містить перелік усіх подій, створених поточним авторизованим користувачем. Графічний інтерфейс сторінки подій відображено на рисунку 3.17.

Event Name	Message	Date	Time	Receivers	Action
Cultural function	Please be on time.	2025-03-20	06:50:00	Show details	Delete
Sports	You all are invited.	2025-03-20	10:03:00	Show details	Delete
programming contest	We are going to held a programming contest next week.	2025-03-20	10:30:00	Show details	Delete
Debate competition	You all are requested to enjoy the show. Thank you.	2025-03-20	10:50:00	Show details	Delete

Рисунок 3.17 - Сторінка подій

На цій сторінці для кожної створеної події відображається її назва, відповідне повідомлення, час, дата та загальна інформація про отримувачів. Для перегляду детальної інформації про отримувачів конкретної події користувач може активувати посилання "Показати деталі" (Show Details), розташоване у відповідному полі.

Інтерфейс перегляду деталей отримувачів проілюстрований на рисунку 3.18.

First Name	Last Name	Department	Designation	Contact-no	E-mail
k	h	C	student	0177947437	mist@gmail.com

**Reciever 2 :**

First Name	Last Name	Department	Designation	Contact-no	E-mail
a	r	C	student	0177947437	annya@gmail.com

Рисунок 3.18 - Деталі сторінки отримувачів

### Сторінка облікового запису (Account Page)

Цей інтерфейс надає доступ до персональної інформації авторизованого користувача. Функціонал сторінки передбачає можливість редагування та оновлення цієї інформації користувачем.

Графічний інтерфейс сторінки облікового запису показано на рисунку 3.19.

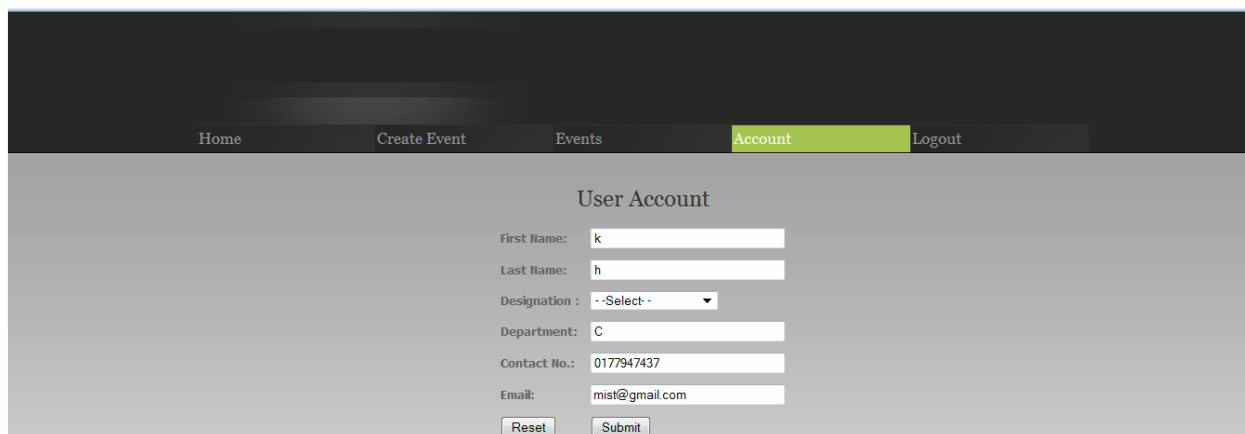


Рисунок 3.19 - Сторінка облікового запису

Ця частина розділу представляє систематичну методологію проектування, розроблену для вирішення системних задач значного масштабу та складності. Зазначений підхід спрямований на задоволення потреб та досягнення цілей кінцевих користувачів. В рамках глави розглянуто системний погляд, який передбачає аналіз користувачів у їхньому операційному контексті.

### 3.6. Вибір методології для реалізація графічного інтерфейсу користувача (GUI)

Графічний інтерфейс користувача (GUI) являє собою інтерактивний програмний компонент, що опосередковує взаємодію користувача з електронними пристроями та системами за допомогою графічних елементів,

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

таких як іконки, кнопки, вікна та інші візуальні індикатори, на протигагу командним інтерфейсам або текстовій навігації. Взаємодія в середовищі GUI реалізується переважно шляхом прямої маніпуляції цими графічними об'єктами.

Процес розробки GUI здійснювався відповідно до систематичного підходу, який розпочався з етапу глибокого аналізу потреб кінцевих користувачів. Це дозволило сформуваи чіткі вимоги до функціональності та ергономіки інтерфейсу, забезпечуючи його орієнтованість на цілі користувачів, як було зазначено у попередніх розділах.

Наступним кроком стало вивчення та застосування відповідних шаблонів проектування GUI та архітектурних патернів. На початковій стадії було проведено ідентифікацію та структурування даних, необхідних для коректного функціонування всіх компонентів графічного інтерфейсу. Ці дані були організовані та збережені у відповідному середовищі, що забезпечило ефективний доступ до них з боку компонентів GUI. Для полегшення доступу та обробки даних з вихідних (сирих) файлів було реалізовано шар абстракції або "обгортку" (wrapper), яка трансформувала сирі дані у формат, придатний для використання в системі та відображення в інтерфейсі.

Виходячи з визначених вимог та аналізу даних, було обрано та прийнято за основу відповідний шаблон або архітектурний підхід. Дотримуючись принципів цього шаблону, було розпочато процес безпосередньої реалізації компонентів GUI, що включав створення візуальних елементів, визначення їхньої поведінки та логіки взаємодії.

Для імплементації користувацького інтерфейсу було обрано архітектурний фреймворк Model-View-Controller (MVC), зокрема його реалізацію на базі CodeIgniter. Використання MVC забезпечило чіткий поділ відповідальності між моделлю даних (Model), візуальним представленням (View) та логікою керування (Controller), що сприяє підвищенню

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

модульності, полегшенню розробки, тестування та подальшого супроводу системи.

На початкових етапах реалізації виникли певні труднощі, пов'язані з необхідністю адаптації структури моделі даних. Згідно з ітераційним підходом та зворотним зв'язком від аналізу потреб користувачів, було здійснено модифікацію моделі даних, щоб вона максимально відповідала вимогам та сценаріям використання, підтримуючи тим самим ефективну взаємодію з GUI. Успішне подолання цих труднощів дозволило завершити формування повноцінного графічного інтерфейсу користувача.

### 3.7. Тестування системи

Тестування системи є критично важливим етапом життєвого циклу розробки програмного забезпечення, спрямованим на верифікацію та валідацію функціональності та продуктивності розробленої системи. Процес тестування було структуровано відповідно до загальноприйнятих методологій, що включали модульне тестування та системне тестування.

#### 3.7.1. Модульне тестування

Модульне тестування є рівнем тестування програмного забезпечення, який фокусується на верифікації окремих, найменших тестованих компонентів або "модулів" системи в ізоляції від решти коду. Основною метою модульного тестування є підтвердження того, що кожен розроблений модуль функціонує відповідно до своїх специфікацій та проектних вимог. У контексті процедурного програмування модулем може виступати окрема функція або процедура, тоді як в об'єктно-орієнтованому програмуванні модулем часто є окремий метод класу.

Методологія модульного тестування, застосована в даному проєкті, передбачала аналіз коду реалізації для кожного конкретного компонента

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

(наприклад, окремого поля введення або функції обробки даних). На основі цього аналізу визначалися допустимі діапазони вхідних даних, що включали коректні (валідні), некоректні (невалідні) та граничні значення. Для кожного тестового входу було визначено очікуваний результат, що базувався на логіці реалізації та функціональних специфікаціях модуля. Проводилось виконання тестів шляхом подачі визначених вхідних даних та порівняння фактичних виходів з очікуваними результатами.

Модульне тестування було систематично проведено для кожного випадку використання, визначеного в рамках проекту. Нижче наведено опис ключових тестових сценаріїв для різних груп користувачів.

#### *Тестування випадків використання загального користувача*

Модульні тести для загального користувача сфокусовані на функціональності, доступній до авторизації.

#### Випадок використання "Реєстрація":

Тестування спрямоване на перевірку коректності процесу подання запиту на реєстрацію загальним користувачем. Перевірялася успішність надсилання запиту після заповнення форми реєстрації. Ключові сценарії включали:

Верифікація валідаційних правил форми: перевірка реакції системи на спробу надсилання форми з пропущеними обов'язковими полями (ім'я, прізвище, контактний номер, електронна пошта, пароль). Очікуваний результат – відхилення надсилання форми.

Перевірка формату електронної пошти: тестування введення невалідного формату електронної пошти. Очікуваний результат – відхилення введення та вимога надання коректної адреси електронної пошти.

Перевірка успішного надсилання форми з усіма валідними даними. Очікуваний результат – успішне надсилання запиту та генерація відповідного сповіщення для адміністратора.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

### *Тестування випадків використання адміністратора*

Модульні тести для адміністратора охоплювали функції управління користувачами та членами системи.

#### *Випадок використання "Схвалення/Відхилення користувача":*

Тестування функціоналу обробки адміністратором запитів на реєстрацію. Перевірялася коректність дій при схваленні або відхиленні запиту та генерація відповідного сповіщення (електронного листа) користувачеві. Ключовий сценарій включав:

Перевірка залежності статусу запиту від успішності надсилання електронного листа.

Очікуваний результат – запит залишається активним до моменту підтвердження успішного надсилання комунікації користувачеві.

#### *Випадок використання "Блокування користувача":*

Тестування функції блокування облікових записів користувачів адміністратором. Перевірялося коректне переміщення даних користувача між списками "поточних" та "заблокованих" користувачів, а також неможливість авторизації для заблокованого облікового запису.

#### *Випадок використання "Видалення користувача":*

Тестування функції видалення облікових записів користувачів адміністратором. Перевірялося коректне переміщення даних користувача зі списку "поточних" до списку "попередніх" користувачів після видалення.

Випадок використання "Ведення списку членів": Тестування функціоналу додавання нових членів організації. Перевірялася валідація даних, що вводяться у форму додавання члена. Сценарії включали:

- Перевірка неможливості надсилання форми з пропущеними обов'язковими полями (ім'я, посада, відділ, контактний номер).
- Перевірка валідності формату електронної пошти.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

### *Тестування випадків використання авторизованого користувача*

Модульні тести для авторизованого користувача покривали функції, пов'язані зі створенням та управлінням подіями та власним профілем.

#### *Випадок використання "Створення події":*

Тестування процесу створення нової події. Перевірялася коректність введення та збереження даних події (назва, повідомлення, час, дата). Сценарії включали:

Перевірка генерації повідомлення про помилку у випадку відсутності обов'язкових даних (назви, повідомлення, дати або часу).

Верифікація логічної послідовності часу та дати (неможливість створення події з минулою датою/часом). Очікуваний результат – відхилення створення події та вимога введення коректних даних.

Перевірка валідації вибору отримувачів: вимога вибору хоча б одного отримувача для надсилання сповіщення.

Перевірка успішного створення події з усіма валідними даними та обраними отримувачами.

#### *Випадок використання "Оновлення інформації":*

Тестування функціоналу оновлення персональних даних користувача. Перевірялася успішність збереження змінених даних після натискання кнопки "Оновити" та функціональність скидання форми до початкового стану при натисканні кнопки "Скидання".

Випадок використання "Перегляд попередніх подій": Тестування відображення списку подій, створених авторизованим користувачем. Перевірялася коректність відображення основної інформації про подію та можливість перегляду детальної інформації про отримувачів, яким було надіслано сповіщення (SMS/електронні листи).

Випадок використання "Видалення подій": Тестування функції видалення створених подій авторизованим користувачем. Перевірялася успішність видалення обраної події зі списку.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.7.2. Системне тестування

Системне тестування є рівнем тестування, що виконується на повністю інтегрованому програмному продукті. Основною метою системного тестування є верифікація поведінки системи в цілому на відповідність документації специфікації вимог до програмного забезпечення (SRS). Цей вид тестування фокусується на оцінці системи з точки зору бізнес-вимог, функціональних вимог та вимог кінцевого користувача.

Системне тестування відноситься до категорії "чорноскринькового" тестування (black-box testing), оскільки оцінка зовнішньої функціональності програмного забезпечення проводиться без врахування внутрішньої структури коду. Це перший етап комплексного (end-to-end) тестування інтегрованого додатку перед його розгортанням або випуском.

В рамках проекту було проведено системне тестування усієї веб-системи після завершення її інтеграції. Тестування охоплювало ключові наскрізні процеси та функціональні області, такі як повний цикл реєстрації користувача (включаючи подання запиту та його обробку адміністратором), процедури входу та аутентифікації для різних ролей користувачів, взаємодія між різними типами користувачів, а також функціонал створення подій. Особлива увага приділялася перевірці коректності роботи критично важливих підсистем, зокрема системи надсилання SMS-повідомлень. Успішне виконання визначених тестових сценаріїв системного тестування підтвердило відповідність інтегрованої системи висунутим до неї вимогам.

В умовах функціонування інституцій та організацій часто виникає необхідність у поширенні уніфікованих повідомлень серед значних груп осіб з метою інформування про події або надання важливої інформації. Ручне виконання цієї задачі, особливо при великій кількості отримувачів, є надзвичайно трудомістким та схильним до помилок процесом. Пропуск критично важливих повідомлень окремими отримувачами становить суттєвий ризик, що вимагає від ініціатора комунікації підвищеної уваги та

						Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата	БР.ІІІ – 18.00.00.000 ПЗ	

контролю. Для вирішення цієї проблеми була розроблена автоматизована система, призначена для ефективного масового сповіщення. Ця система усуває адміністративне навантаження з організацій щодо інформування цільових аудиторій та забезпечує автоматизоване розсилання повідомлень, що є складним для реалізації вручну.

В процесі розробки проекту було використано архітектурний фреймворк Model-View-Controller (MVC), зокрема його реалізацію CodeIgniter. Впровадження фреймворка, що було першим досвідом для команди розробників, виявило певні початкові труднощі, пов'язані з освоєнням парадигми MVC. Подолання цих труднощів дозволило набути навичок ефективного використання фреймворка для веб-розробки.

Крім того, значним технічним викликом стала реалізація функціоналу масового надсилання SMS-повідомлень користувачам на основі створеної події. Було досліджено та апробовано низку методичних підходів та технічних рішень для забезпечення надійної доставки SMS, частина з яких виявилася неефективною.

Розроблена система повідомлень про події є сучасною веб-орієнтованою платформою, що має потенціал суттєво покращити комунікаційні процеси в інституціях та організаціях, забезпечуючи ефективно та надійне донесення важливої інформації до цільових груп. Система може бути корисною для широкого кола користувачів, які потребують оптимізації масових комунікацій. Таким чином, впровадження системи повідомлень про події сприятиме підвищенню ефективності діяльності організацій та інституцій.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

## ВИСНОВКИ

В дипломній роботі було виконано комплексний аналіз теоретичних і прикладних аспектів проєктування та реалізації системи обміну повідомленнями про події різного рівня локалізації. Інформаційні технології є ключовим інструментом для побудови інтегрованих систем управління подіями, що функціонують у мультиагентних, розподілених або географічно розосереджених середовищах. Проаналізовані сучасні хмарні платформи (Cvent, Eventbrite, Everbridge) свідчать про наявність високого попиту на рішення, здатні забезпечити надійне, масштабоване та локалізоване сповіщення користувачів про події.

Мотивація до розробки системи обумовлена потребою у високофункціональному інструменті для оперативного обміну повідомленнями з урахуванням ролей користувачів, рівнів доступу, місцезнаходження та характеру подій. Це дозволяє не лише підвищити ефективність комунікацій, а й оптимізувати інформаційне навантаження на користувача.

У процесі розробки архітектури системи було враховано основні виклики, пов'язані з масштабованістю, асинхронною обробкою подій, потребою у локалізації повідомлень та забезпеченням захисту персональних даних. Було запропоновано модульну архітектуру, яка дозволяє гнучко адаптувати систему під потреби організацій із різною структурою.

Побудовано модель вимог та сценаріїв використання, що охоплюють різні типи взаємодії з системою — від базового користувача до адміністратора. Це забезпечило повне охоплення функціональності й дозволило ефективно реалізувати логіку авторизації, управління користувачами та подіями.

На основі моделювання варіантів використання було спроектовано алгоритмічне забезпечення ключових процесів системи, включаючи

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

реєстрацію, перевірку, створення та обробку подій, що забезпечує чітке розмежування повноважень і мінімізує ризики помилок при обміні повідомленнями.

Було розроблено модель бази даних, яка реалізує підтримку зберігання інформації про користувачів, події, їх статуси, локалізацію та хронологію. Це створює підґрунтя для забезпечення цілісності, доступності та актуальності даних.

Реалізовано графічний інтерфейс користувача з урахуванням ролей системи, що забезпечує інтуїтивну взаємодію, зменшення кількості помилок і пришвидшення виконання ключових операцій. Використання сучасних методологій проектування GUI сприяло підвищенню зручності користування.

Проведене модульне й системне тестування підтвердило функціональну коректність та стабільність роботи системи, а також її здатність адаптуватися до навантажень і масштабуватися в умовах реального використання.

Загалом, результати дослідження доводять доцільність застосування поєднання подієво-орієнтованого підходу та сучасних інформаційних технологій для створення систем обміну повідомленнями, які враховують контекст подій, рівень локалізації та особливості користувацьких ролей.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. B. Donley and J. Offutt, “Web application tesing challanges,” Software Engineering, George Mason University, September 2009.
2. SmartDraw and LLC, <http://www.smartdraw.com/resources/tutorials/entityrelationship-diagrams/>.
3. New Ways Eventbrite is Powering Your Growth - <https://www.eventbrite.co.uk/blog/new-ways-eventbrite-powering-growth-ds00/>
4. C. P. Ltd., <http://creately.com/diagram-type/article/simple-guidelines-drawing-umlclass-diagrams/>
5. Association Event Management Software | Cvent - <https://www.cvent.com/en/event-marketing-management/association-solutions>
6. Gyllstrom, D., Wu, E., Chae, H.-J., Diao, Y., Stahlberg, P., & Anderson, G. (2006). SASE: Complex Event Processing over Streams. arXiv.
7. Sharvari, T., & Sowmya Nag, K. (2019). A study on Modern Messaging Systems- Kafka, RabbitMQ and NATS Streaming. arXiv.
8. Özyar, U. C., & Yurdakul, A. (2022). A Decentralized Framework with Dynamic and Event-Driven Container Orchestration at the Edge. arXiv.
9. John, V., & Liu, X. (2017). A Survey of Distributed Message Broker Queues. arXiv.
10. Zhang, Y., & Zhang, M. (2024). Design of Event Management System for Space Big Data Secure Transmission Message Middleware Based on G/S mode. PMBDA 2023.
11. Miriyala, N. S. (2024). Event Driven System Design with High Availability. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 10(6), 2470-2477.
12. Subach, Y. (2020). Structuring events in a messaging system. Retrieved from <https://yurisubach.com/2020/04/10/structuring-events/>

					БР.ІІІ – 18.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

13. Mühl, G., Schröter, A., Parzyjegl, H., Kounev, S., & Richling, J. (2012). Performance modeling and analysis of message-oriented event-driven systems. *Software and Systems Modeling*.
14. Aravind, M., Wiklander, G., Palmheden, J., & Dobrin, R. (2017). An Event-Based Messaging Architecture for Vehicular Internet of Things (IoT) Platforms. In *ICT Innovations 2017* (pp. 37–46). Springer.
15. Kuusela, A. (2009). Design of an Event Based Messaging System. LUTPub.
16. AltexSoft Editorial Team. (2021). Event-Driven Architecture and Pub/Sub Pattern Explained. Retrieved from <https://www.altexsoft.com/blog/event-driven-architecture-pub-sub/>
17. GeeksforGeeks. (2024). Message Queues vs Event Streams in System Design. Retrieved from <https://www.geeksforgeeks.org/message-queues-vs-event-streams-in-system-design/>
18. Particle. (n.d.). Messaging Architecture for Scale. Retrieved from <https://docs.particle.io/scaling/best-practices/messaging-architecture-for-scale/>
19. Chanaka Fernando. (2019). A futuristic view of building distributed systems with messaging. Medium. Retrieved from <https://chanakaudaya.medium.com/a-futuristic-view-of-building-distributed-systems-with-messaging-560d0652513a>
20. He, F., Baresi, L., Ghezzi, C., & Spoletini, P. (2007). Formal analysis of publish-subscribe systems by probabilistic timed automata. In *Formal Techniques for Networked and Distributed Systems (FORTE 2007)*, Lecture Notes in Computer Science, 4574.
21. Kounev, S., Sachs, K., Bacon, J., & Buchmann, A. P. (2008). A methodology for performance modeling of distributed event-based systems. In *Proceedings of 11th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC'08)*.

					БР.ІІІ – 18.00.00.000 ІІЗ	Арк. 75
Змн.	Арк.	№ докум.	Підпис	Дата		

22. Naedele, M., & Janneck, J. W. (1998). Design patterns in Petri net system modeling. In Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'98).
23. Mulyar, N., & von der Aalst, W. (2005). Towards a pattern language for colored Petri nets. In Proceedings of the Sixth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools (CPN '05).
24. Kounev, S., & Sachs, K. (2009). Benchmarking and performance modeling of event-based systems. *Informatik Spektrum*, 51(5), 262–269.
25. Mühl, G., Schröter, A., Parzyjegla, H., Kounev, S., & Richling, J. (2009). Stochastic analysis of hierarchical publish/subscribe systems. In Euro-Par 2009 Parallel Processing, Lecture Notes in Computer Science, 5704. Springer.
26. Gyllstrom, D., Wu, E., Chae, H.-J., Diao, Y., Stahlberg, P., & Anderson, G. (2006). SASE: Complex Event Processing over Streams. arXiv preprint arXiv:cs/0612128.
27. Sharvari, T., & Sowmya Nag, K. (2019). A study on Modern Messaging Systems - Kafka, RabbitMQ and NATS Streaming. arXiv preprint arXiv:1912.03715.
28. Özyar, U. C., & Yurdakul, A. (2022). A Decentralized Framework with Dynamic and Event-Driven Container Orchestration at the Edge. arXiv preprint arXiv:2206.11663.
29. John, V., & Liu, X. (2017). A Survey of Distributed Message Broker Queues. arXiv preprint arXiv:1704.00411.
30. Zhang, Y., & Zhang, M. (2024). Design of Event Management System for Space Big Data Secure Transmission Message Middleware Based on G/S mode. In Proceedings of PMBDA 2023.
31. Miriyala, N. S. (2024). Event Driven System Design with High Availability. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10(6), 2470–2477.

					БР.ІІІ – 18.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

32. Mühl, G., Schröter, A., Parzyjegla, H., Kounev, S., & Richling, J. (2012). Performance modeling and analysis of message-oriented event-driven systems. *Software and Systems Modeling*, 11(4), 495–511.
33. Aravind, M., Wiklander, G., Palmheden, J., & Dobrin, R. (2017). An Event-Based Messaging Architecture for Vehicular Internet of Things (IoT) Platforms. In *ICT Innovations 2017* (pp. 37–46). Springer.
34. He, F., Baresi, L., Ghezzi, C., & Spoletini, P. (2007). Formal analysis of publish-subscribe systems by probabilistic timed automata. In *Formal Techniques for Networked and Distributed Systems* (pp. 250–265). Springer.
35. Kounev, S., Sachs, K., Bacon, J., & Buchmann, A. P. (2008). A methodology for performance modeling of distributed event-based systems. In *Proceedings of ISORC'08* (pp. 13–20). IEEE.

					БР.ІІІ – 18.00.00.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

## БІБЛІОГРАФІЧНА ДОВІДКА

**Тема дипломної роботи:** “ Проектування та розробка системи обміну повідомленнями різного рівня локалізації ”

Обсяг пояснювальної записки: 77 аркушів.

Дата закінчення роботи: 11 червня 2025 р.

Підпис студента \_\_\_\_\_