

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Центр заочної, дистанційної та вечірньої форм навчання**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Інформаційна технологія формування повідомлення за шаблонами стандарту НАТО

Здобувача групи ІТ.мз-22с Нені Віктора Григоровича

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

\_\_\_\_\_

(підпис)

Віктор НЕНЯ

Керівник к.т.н., доц. доцент кафедри ІТ Світлана ВАЩЕНКО

\_\_\_\_\_

(підпис)

Суми – 2024

Сумський державний університет  
Центр заочної, дистанційної та вечірньої форм навчання  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІТ

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

*Нені Віктора Григоровича*

**1 Тема кваліфікаційної роботи** Інформаційна технологія формування повідомлення за шаблонами стандарту НАТО

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

**2. Термін здачі студентом кваліфікаційної роботи** «15» січня 2024 р.

**3 Вхідні дані до кваліфікаційної роботи** стандарти формування формалізованих повідомлень, затверджені НАТО: APP-11 NATO Message Catalogue Edition D Version 1ADatP-3 NATO Message Text Formatting System (Formets) – Concept of Formets (Conformets)

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** аналіз предметної області; планування проекту; вибір засобів реалізації; проектування інформаційної тенології; проектування бази даних шаблонів формалізованих повідомлень та бази даних сформованих повідомлень; реалізація інформаційної технології; опис порядку використання

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації)** актуальність задачі дослідження; постановка задачі; функціональні вимоги; нефункціональні увимоги; огляд структури шаблону повідомлення; діаграми бізнес-процесу на контекстному рівні та 1-ому рівні декомпозиції; діаграма варіантів використання; схеми моделей баз даних; приклади викорис-тання розробленої технології; висновки

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання

5 вересня 2023 р.

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
	Аналіз предметної області	Вересень 2023 р.	
	Постановка задачі	Вересень 2023 р.	
	Визначення вимог та інструментів реалізації	Вересень 2023 р.	
	Планування проекту	Жовтень 2023 р.	
	Проектування інформаційної технології	Жовтень 2023 р.	
	Проектування та реалізація об'єктів баз даних	Жовтень – Листопад 2023 р.	
	Реалізація програмного додатку	Листопад 2023 р.	
	Підготовка та оформлення документації	Грудень 2023 р.	
	Захист проекту	Січень 2024 р.	

Магістрант

\_\_\_\_\_

Віктор НЕНЯ

Керівник роботи

\_\_\_\_\_

к.т.н., доц. Світлана ВАЩЕНКО

## АННОТАЦІЯ

Тема кваліфікаційної роботи магістра «Інформаційна технологія формування повідомлення за шаблонами стандарту НАТО».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 31 найменування, 7 додатків. Загальний обсяг роботи – 126 сторінок, у тому числі 72 сторінки основного тексту, 2 сторінки списку використаних джерел, 53 сторінки додатків.

Кваліфікаційну роботу магістра присвячено розробці інформаційної технології формування військового повідомлення відповідно шаблонам формалізованих повідомлень, структура яких регламентується стандартом НАТО APP-11 Version 1 Edition D.

В роботі проведено аналіз протоколу інформаційного обміну, структури шаблонів формалізованих повідомлень та правил формування військових повідомлень на основі шаблонів.

У роботі виконано структурно-функціональне моделювання бізнес-процесів інформаційної технології, моделювання варіантів використання та моделювання даних для реалізації бази даних шаблонів формалізованих повідомлень та бази даних сформованих повідомлень на основі збережених шаблонів. Для реалізації інформаційної технології використані сучасні технології баз даних та середовища розробки.

Результатом проведеної роботи є впроваджена інформаційна технологія формування військових повідомлень на основі шаблонів стандарту НАТО.

Практичне значення роботи полягає у інтеграції підходів виконання військових доручень ЗСУ з підходами військ альянсу НАТО за рахунок використання єдиного стандарту і протоколу обміну військовими повідомленнями.

Ключові слова: шаблон військового повідомлення, формалізоване повідомлення, система баз даних, тригерна функція.

## ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Протокол інформаційного обміну	9
1.2 Перелік формалізованих повідомлень і сигналів бойового управління	11
1.3 Структура формалізованого повідомлення	12
2 ПОСТАНОВКА ЗАДАЧІ	26
2.1 Мета та задачі дослідження	26
2.2 Засоби реалізації	27
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	29
3.1 Структурно функціональне моделювання формування повідомлення за шаблоном стандарту НАТО	29
3.2 Моделювання варіантів використання	33
3.3 Проектування моделей баз даних	36
4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	43
4.1 Реалізація бази даних шаблонів повідомлень dbTemplates	43
4.2 Реалізація бази даних сформованих повідомлень dbMessage	49
4.4 Реалізація модуля редактора шаблонів повідомлень	54
4.5 Реалізація модуля редактора повідомлень	59
4.6 Ілюстрація впровадження інформаційної технології	63
ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
ДОДАТОК А	74
ДОДАТОК Б	78
ДОДАТОК В	82
ДОДАТОК Г	87
ДОДАТОК Д	94

ДОДАТОК Е	108
ДОДАТОК Ж	125

## ВСТУП

В сучасному світі за умов швидкого та динамічного розвитку інформаційних технологій цифровізація впроваджується в усі сфері людської діяльності. В умовах інтеграції Збройних Сил України та військових підрозділів альянсу НАТО виникає актуальне питання адаптації підходів управління військовими підрозділами до підходів, що застосовуються за стандартами НАТО, з врахуванням особливостей підготовки та застосування ракетних військ Збройних Сил України. Одним із ключовим питань є обмін повідомленнями при формування бойових наказів на виконання. Розроблення інформаційної технології автоматизованого формування повідомлень за шаблонами, що регулюються стандартом НАТО, є актуальною задачею.

Впровадження зазначеної інформаційної технології в роботу ракетних військ Збройних Сил України дозволить не тільки посилити інтеграцію з військовими підрозділами НАТО, але й підвищить ефективність виконання бойових задач українських військових за рахунок оперативного інформування та постановки задач.

**Мета.** Розробка інформаційної технології для автоматизованого формування повідомлень за шаблонами стандарту APP-11 NATO Message Catalogue Edition D Version 1. Практичною значимістю роботи є створення інструменту для оперативного формування повідомлення за встановленими шаблонами, які за структурою та змістом відповідають стандарту APP-11.

Для досягнення мети дослідження визначено такі задачі :

- аналіз та детальний опис предметної області;
- постановка задачі з детальним описом вимог до технології, яка озробляється;
- вибір програмних засобів реалізації;
- розроблення інформаційної технології для автоматизованого формування повідомлень;

- моделювання функціонування програмної реалізації інформаційної технології, що розробляється;
- розроблення моделі даних для відображення складеної структури шаблону повідомлення за стандартом APP-11;
- розроблення програмного додатку на основі бази даних; програмний додаток має реалізовувати запропоновану інформаційну технологію.



## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

Враховуючі специфічність предметної області дослідження віднайти програмні продукти чи інформаційні системи закордонного виробництва аналогічного функціонального призначення у відкритому доступі не є можливим. В Україні процес управління діями ракетних підрозділів Збройних Сил України знаходиться в досить активній, але ще далеко не фінальній, фазі інтеграції з інформаційними процесами підтримки комунікації між військовими підрозділами НАТО. Саме тому на час розробки інформаційної технології були відсутні аналогічні програмні продукти чи системи, які б можна було проаналізувати.

В зв'язку з цим розглянемо більш детально інформаційні процеси обміну повідомленнями та структуру шаблону повідомлення, які визначені стандартом НАТО в наступних підрозділах розділу.

### **1.1 Протокол інформаційного обміну**

Протокол інформаційного обміну призначений для відображення змісту автоматизованого обміну формалізованими повідомленнями (далі ФП) між пунктом управління ракетним підрозділом та вищою ланкою управління, а також між сигналом бойового управління та пунктом управління ракетним підрозділом.

Протокол інформаційного обміну формується на основі наступних документів:

- APP-11 NATO Message Catalogue Edition D Version 1 November 2015 [1];

- ADatP-3 NATO Message Text Formatting System (Formets) – Concept of Formets (Conformets) [2];
- ACP 127 Communications Instructions – Tape Relay Procedures [3];
- STANAG 4406 Edition 2– 2006 Military Message Handling System [4];
- словник оперативно-тактичних понять та військово-технічних термінів;
- інформаційна модель управління ракетним підрозділом, озброєного оперативно-тактичним ракетним комплексом (ОТРК).

Схема зв'язності абонентів інформаційної взаємодії показано на рис. 1.1. Вона включає пункт управління (далі ПУ), основний та запасний канали інформаційної взаємодії.

Для ПУ вищої ланки управління (на схемі ПУ ВЛУ):

- основний канал взаємодії – з мобільним командним пунктом управління ракетним дивізіоном (на схемі МКПУ рдн);
- запасний канал взаємодії – з опорним командним пунктом (ОКП) МКПУ рдн.

Для мобільного командного пункту управління стартової батареї (МКПУ сбатер):

- основний канал взаємодії – з тактичним операційним центром (ТОЦ);
- запасний канал взаємодії – з основним командним пунктом (ОКП).

Для ПУ тактичної батаерії (ТБ) та підрозділів забезпечення:

- під час бойових дій основний канал взаємодії – з ТОЦ, запасний – з ОКП;

при плануванні та забезпеченні бойових дій основний канал взаємодії – з ОКП.

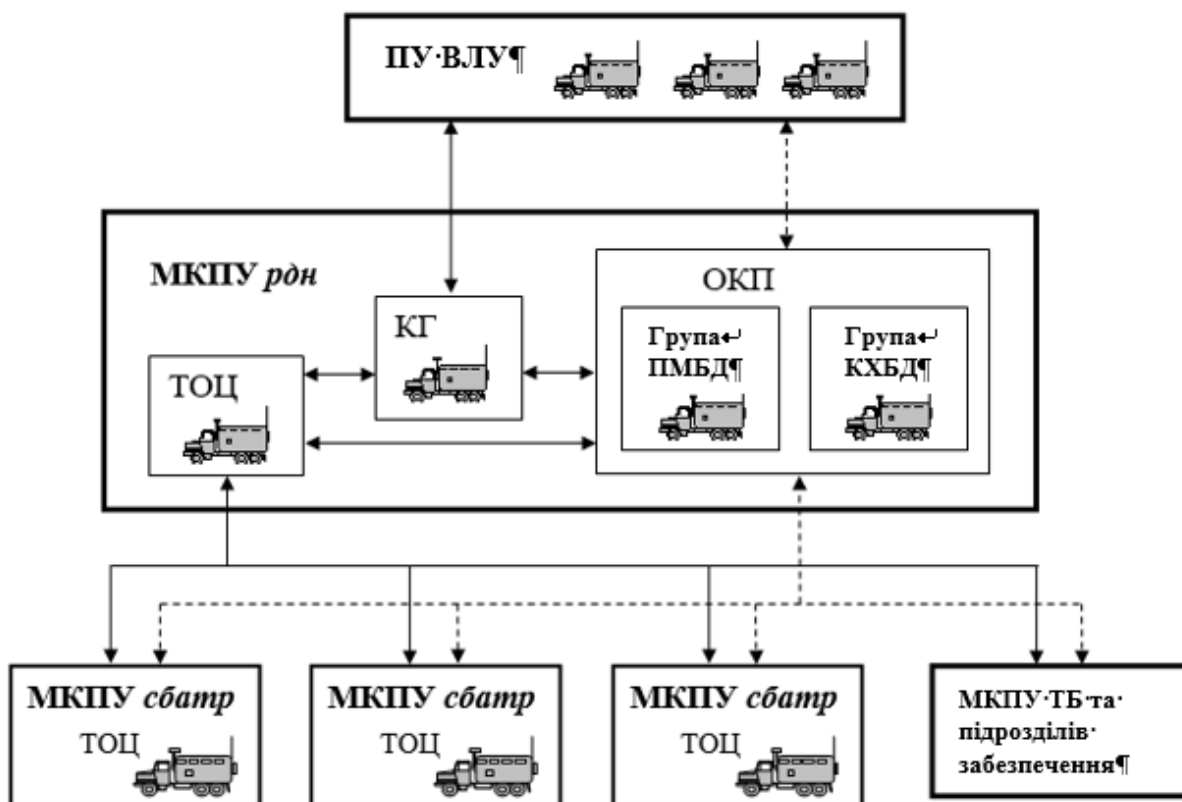


Рисунок 1.1– Схема зв'язності абонентів інформаційної взаємодії

*Джерело: розроблено автором*

## 1.2 Перелік формалізованих повідомлень і сигналів бойового управління

Перелік формалізованих повідомлень інформаційного обміну у ланках управління вищої ланки управління – ракетний дивізіон – стартова батарея (табл. 1.1) розроблено відповідно до APP-11 NATO Message Catalogue Edition D Version 1 November 2015 [1].

Перелік формалізованих повідомлень призначений для забезпечення процедур інформаційного обміну за стандартами НАТО при веденні бойових дій під час виконання завдань приведення ракетного підрозділу у бойову готовність, нанесення ракетного удару (проведення пусків ракет), здійснення

маневру силами та засобами ракетного підрозділу у ланці вищої ланки управління – ракетний дивізіон, а також за умови забезпечення бойових дій ракетног підрозділу та вирішення інших завдань у ланці вищої ланки управління – ракетний дивізіон – стартова батарея, що виникають при проведенні спільних операцій збройними силами кількох держав.

Повний перелік формалізованих повідомлень приведений у додатку А.

### 1.3 Структура формалізованого повідомлення

#### 1.3.1. Структурні компоненти формалізовано повідомлення

Повідомлення складається з таких складових частин (рис. 1.2):

1. Службової частини повідомлення (структура службової частини повідомлення не змінюється залежно від форми повідомлення з АРР 11) визначається двома документами [1,2];

Таблиця 1.1 – Перелік формалізованих повідомлень в ланці вищої ланки управління – ракетний дивізіон

№ п/п	Найменування формалізованого повідомлення в ланці вищої ланки управління – ракетний дивізіон	
1	FM.CFF FM.FMC NNFP.FP TARGETINTREC MSGCORRCANX	Вогневе завдання – виклик вогню Вогневе завдання – команда на виконання вогневого завдання Планування вогню неядерними боєприпасами – план вогню Дані про цілі Повідомлення про відміну попереднього повідомлення
2	FM.CFF FM.FMC NNFP.FP NNFP.COMFP TARGETINTREC MSGCORRCANX	Вогневе завдання – виклик вогню Вогневе завдання – команда на виконання вогневого завдання Планування вогню неядерними боєприпасами – план вогню План-розрахунок неядерного порядку вогневого ураження Дані про цілі Повідомлення про відміну попереднього повідомлення
3	FM.CFF FM.FMC MSGCORRCANX	Вогневе завдання – виклик вогню Вогневе завдання – команда на виконання вогневого завдання Повідомлення про відміну попереднього повідомлення
4	FRAGO	Окреме бойове розпорядження

## Продовження таблиці 1.1

№ п/п	Найменування формалізованого повідомлення в ланці вищої ланки управління – ракетний дивізіон	
5	FRAGO AFU.DCMD	Окреме бойове розпорядження Команда щодо розміщення позиційного району вогневого підрозділу
6	AIRATTACKWARN AIRRAIDWNG	Повітряна тривога Повітряна тривога
	CBRN HAZWARN	РХБЗ Попередження про небезпеку
	CBRN HAZWARN	РХБЗ Попередження про небезпеку

*Джерело: побудовано автором*

2. Формалізованого повідомлення, що включає:

- заголовок повідомлення (заголовок повідомлення не змінюється залежно від форми повідомлення з APP 11);
- текст повідомлення відповідно до форми з APP 11.

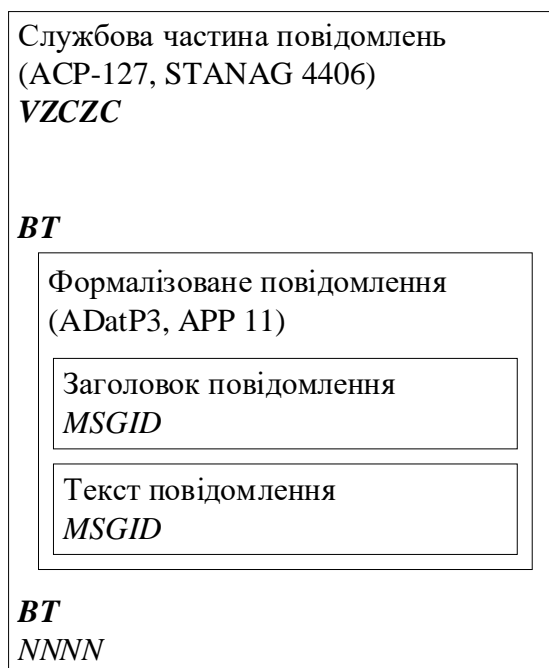


Рисунок 1.2– Складові частини повідомлення

*Джерело: розроблено автором*

Службова частина повідомлення має бути підготовлена відповідно до формату, відображеного в таблиці 1.2.

Таблиця 1.2 – Схематична діаграма службової частини повідомлення

Частина	Компонент	Формат лінії	Елемент	Зміст	Пояснення	Спосіб заповнення	Обов'язкові рядки
ЗАГОЛОВОК	Порядок дій	1	Інструкція обробки	Ідентифікація передаючих станцій. Условне позначення безпеки Напрямок – Направляюча інформація: – умовне позначення категорії терміновості; – індикатор способу маршрутизації; – умовне позначення в донесіннях; – адресні позначення за необхідності	Завжди містить ідентифікацію трансмісії (що включає "Старт - Повідомлення - Індикатор" за необхідності); також містить направляючу інформацію відповідно до вимог передачі конкретних інструкцій обробки повідомлень.	Автоматичний	+
		2	Станція, що викликається	Умовне позначення категорії терміновості. Індикатор(и) станції(й) маршрутизації, відповідальна(і) за доставку чи транзит.	Базова лінія маршрутизації. Якщо повідомлення має кілька категорій терміновості, вказує максимальний пріоритет.	Ручний (для визначення категорії терміновості)/ автоматичний	+

## Продовження таблиці 1.2

Частина	Компонент	Формат лінії	Елемент	Зміст	Пояснення	Спосіб заповнення	Обов'язкові рядки
		3	Станція відправника та час подачі повідомлення	Умовне позначення DE. Індикатор станції маршрутизація підготовки повідомлення передачі; серійний номер станції; час подачі повідомлення: дата розділені похилою лінією години та хвилини вираженими цифрами з наступним значенням суфіксом зони	Час подачі повідомлення є дата та час, коли повідомлення було подано до центру зв'язку.	Автоматичний	+
		4	Інструкції щодо передачі	Умовне позначення T. Категорія безпеки робочого сигналу (при використанні). Інші операційні сигнали; Спеціальна робоча група(и) (SOGs); умовне позначення адрес (ів); Індикатор(и) маршрутизації	Вказує категорію безпеки. Заголовки повідомлень. Не слід використовувати, якщо це не потрібно. Для визначення адреси не допускається кодована адреса.	Ручний (для визначення категорії безпеки)	+

## Продовження таблиці 1.2

Частина	Компонент	Формат лінії	Елемент	Зміст	Пояснення	Спосіб заповнення	Обов'язкові рядки
	Преамбула	5	Категорія терміновості; Дані про дату та час (DTG); Інструкції повідомлення.	Умовне позначення категорії терміновості; Дані про дату та час та суфікс зони (Z вказує по відношенню до Грінвіча). Операційний сигнал(и).	У разі подвійної категорії терміновості вказують обидва умовні позначення категорії терміновості розділених пробілом. Операційний сигнал (и) використовується лише тоді, коли потрібно передати інструкції з обробки повідомлень.	Ручний (для визначення категорії терміновості)/ автоматичний	+
	Адреса	6	Відправник	Умовне позначення FM. Позначення відправника.	Відправник повідомлення позначається простою мовою, індикатором маршрутизації, групою адрес або позивною.	Автоматичний	+
		7	Отримувач	Умовне позначення TO. Індикатор(и) маршрутизації. Робочий сигнал Адреса(и).	Адреса (а) одержувача позначається простою мовою, індикатори маршрутизації, адресна група (и) або позивний (і). У разі кількох адрес, коли адресати перераховані окремо, кожна адреса має бути на окремому рядку і може передувати або за допомогою операційного сигналу ZEN	Ручний	+



## Продовження таблиці 1.2

Частина	Компонент	Формат лінії	Елемент	Зміст	Пояснення	Спосіб заповнення	Обов'язкові рядки
		8	Інформація про отримувача	Умовне позначення INFO. Індикатор маршрутизатора (ів). Операційний сигнал (и). Умовне позначення адреси (ів).	Те саме, що і для рядка 7, за винятком того, що рядок 8 містить інформацію адресата (ів). Колективна адреса або адреси, що вказують у групі (AIG) у форматі рядка 7 може включати інформацію про одержувачів.	Автоматичний	
		9	Виключення	Умовне позначення ХМТ. Умовне позначення адрес (ів).	Використовується лише тоді, коли в рядку 7 колективна адреса, вказується адреса одержувача, який виключається для передачі повідомлення або група адрес AIG.	Ручний	+
		10	Облік інформації. Кількість груп.	Умовне позначення GR. Облік символів (за потреби). Кількість груп.	Кількість груп використовуватиметься лише тоді, коли текст складається із зашифрованих груп.	Не використовується	
Роздільник		11		Умовне позначення VT.		Автоматичний	+

## Продовження таблиці 1.2

Частина	Компонент	Формат лінії	Елемент	Зміст	Пояснення	Спосіб заповнення	Обов'язкові рядки
ТЕКСТ		12	Класифікація. Внутрішня інструкція. Інформація що передається відправником (порядок передачі).		Визначається АСР-121 []	Ручний	+
	Роздільник	13		Умовне позначення ВТ.		Автоматичний	+
ЗАВЕРШЕНІЕ СООБЩЕНИЯ		14	Підтвердження / Кількість груп.		Не використовується в реперфораторах	Не використовується	
		15	Виправлення	Умовне позначення С. Інші умовні позначення, операційні сигнали, які вказуються за необхідності.		Не використовується	
		16	Функція кінець повідомлення	2CR, 8LF, 4N's, 12LTRS	NNNN – є індикатор кінця повідомлення	Автоматичний	+

*Джерело: розроблено автором*

### 1.3.2. Алгоритм формування службового повідомлення:

#### 1. Ідентифікатор трансмісії (ТІ) (рядок 1):

ТІ потрібен у заголовку кожного повідомлення. При надсиланні за допомогою автоматичних пристроїв ТІ складається з наступних символів у зазначеній послідовності:

VZCZC\*\*\* ###

- V – потрібне для того, що перший символ інформації не втрачено або не спотворено;
- ZCZC – індикатор початку повідомлення;
- \*\*\* – назва станції або каналу (3 символи);
- \*\*\* – номер каналу (3 числа).

#### 2. Станція, що викликається (RI) (рядок 2):

RI складається з наступних символів у зазначеній послідовності:

\*\* ###

- \*\* – пріоритет повідомлення (2 символи).

Умовне позначення пріоритету повідомлення:

Z – Блискавка;

O – ТЕРМІНОВЕ;

P – ПРІОРИТЕТ;

R – РЕГЛАМЕНТНЕ.

\*\* формується як дублювання умовного позначення пріоритету повідомлення (ZZ, OO, PP, RR). Якщо повідомлення має подвійну категорію терміновості, цей рядок вказує максимальний пріоритет.

- ### – Індикатор(и) станції(й) маршрутизації, відповідальна(і) за доставку або транзит.

#### 3. Станція відправника та час подачі повідомлення (DERI) (рядок 3):

DE \*\*\* ###/#####Z

- \*\*\* – назва станції підготовки повідомлення;
- ### – код станції підготовки повідомлення;
- ##/#####Z – дата подачі повідомлення для відправки

Форма дати: День+«/»+Години+Хвилини+Код зони відносно Грінвіча.

4. Інструкції про передачу (T) (рядок 4) визначають гриф секретності повідомлення.

Рядок T починається із символів за замовченням ZNR і ZNY, за яким слідує перша літера класифікації секретності повідомлень, що повторюється п'ять разів:

- ZNR UUUUU – Несекретне (UNCLASSIFIED);
- ZNY RRRRR – Для обмеженого користування (RESTRICTED);
- ZNY CCCCC – Конфіденційне (CONFIDENTIAL);
- ZNY SSSSS – Секретне (SECRET);
- ZNY TTTTT – Цілком таємно (TOP SECRET).

5. Дії першочерговості (AP) (рядок 5):

\* #####Z \*\*\* #####

- \* – пріоритет повідомлення (1 символ, див. п. 2);
- #####Z \*\*\* ##### – дані про дату та час (формат DTG).
- де ##### - День + Години + Хвилини;
- Z – код зони щодо Грінвіча;
- \*\*\* – скорочене позначення місяця;
- ##### - Рік.

6. Відправник (OD) (рядок 6):

FM \*\*\*

- \*\*\* – ім'я відправника.

7. Одержувач (AD) (рядок 7):

TO \*\*\*

- \*\*\* – ім'я одержувача або ім'я групи списку адрес (використовується якщо кілька одержувачів).

8. Одержувач (ID) (рядок 8):

INFO \*\*\*

- \*\*\* – інформація про одержувача (ій).

9. Виняток (ХМТ) використовується тільки в тому випадку, коли в рядку  
7 група адрес (рядок 9):

ХМТ \*\*\*

- \*\*\* – адреса одержувача, яка виключається з групи адрес у рядку 7;
- \*\*\* – група адрес одержувачів, які виключаються із групи адрес у рядку 7.

10. Кількість груп (GR) використовується лише тоді, коли повідомлення закодовано (рядок 10 – не використовується):

GR ##

- ## – кількість груп закодованого повідомлення.

11. Розділювач (BT) (рядок 11):

BT

12. Розділювач (BT) (рядок 12a, 12b, 12c, 12d, 12g):

- a) \*\*\*
- b) SIC\*\*\*
- c) SUBJ \*\*\*

ТЕХТ

a) гриф секретності повідомлення:

- U N C L A S – Несекретне (UNCLASSIFIED);
- R E S T R I C T E D – Для обмеженого користування (RESTRICTED);
- C O N F I D E N T I A L – Конфіденційна (CONFIDENTIAL);
- S E C R E T – Секретне (SECRET);
- T O P – S E C R E T – Цілком таємно (TOP SECRET).

Літери розділяються пробілами.

b) предметний показчик коду (SIC) є механізмом для формальної ідентифікації теми повідомлення (як правило, три символи, але можуть бути символи та цифри до 8 значень). Використовуються країнами НАТО. Якщо код невідомий, використовується аббревіатура відправника або значення за промовчанням AAA.

- c) \*\*\* – скорочення MSGID із APP-11.

d) TEXT - формалізоване повідомлення з APP-11.

13. Розділювач (BT) (рядок 13):

BT

14. Підтвердження кількості груп (рядок 14 – не використовується).

15. Виправлення (рядок 15 – не використовується).

16. Кінець повідомлення:

NNNN

1.3.3. Загальні правила формування формалізованого повідомлення.

Шаблон повідомлення має унікальний ідентифікатор (MTFID) та задану послідовність блоків. Унікальний MTFID – ключове слово, яке пов'язує інформацію з повідомленням. Він міститься в наборі MSGID, який міститься в заголовку кожного повідомлення. Блоки з ідентифікаторами EXER, OPER та REF також входять до заголовка повідомлення.

Вміст повідомлення є упорядкованими блоками, поділеними на поля, кожен з яких охоплює конкретний аспект повідомлення.

Блоки починаються з ідентифікатора (SETID – стовпець ІН Поля). SETID є унікальною групою символів, призначених кожному блоку, як ключ до структури і змісту цього блоку і забезпечує його ідентифікацію.

Існує два типи блоків: лінійні; табличні.

Кожному блоку (полю) у форматованому повідомленні надається важливість. Існує три типи категорій важливості:

– Обов'язкові. Блоки (поля), які мають бути заповнені, визначаються «+» у стовпцях «Обов'язковість блоку», «Обов'язковість значень у полі».

– Необов'язкові. Блоки (поля), використання яких визначається оператором.

– Умовні. Блоки, які заповнюються залежно від правил, передбачених шаблоном повідомлення.

Правила передбачені шаблоном повідомлення:

– встановлюють значення полів;

– забороняють або вимагають введення певної інформації залежно від іншого блоку (поля) у повідомленні.

Блоки всередині повідомлення можуть бути згруповані разом для формування сегментів, один сегмент може бути вкладений в інший сегмент. Сегмент може повторюватися залежно від правил шаблону повідомлення. Для кожного сегмента у повідомленні визначаються правила:

- початок сегменту;
- номери та назви блоків, включених у сегмент;
- кількість повторень;
- закінчення сегмента.

Для деяких блоків (полів) визначається альтернатива використання, тобто використання одного блоку забороняє використання іншого.

Умови вибору альтернативних блоків (полів):

– обов'язкові блоки (поля) є взаємовиключними, але вимагають обов'язкового заповнення одного з блоків (поля) позначаються цифрами у стовпці «Вибір із альтернатив»;

– необов'язкові Блоки (поля) є взаємовиключними, але не вимагають обов'язкового заповнення одного з блоків (поля), позначаються цифрами та «#» у стовпці «Вибір із альтернатив»;

– обов'язкові блоки (поля) вимагають обов'язкового заповнення одного з блоків (поля), але може використовуватися більше одного блоку (поля), що позначаються цифрами та «\*» у стовпці «Вибір з альтернатив».

Деякі блоки (поля) можуть повторюватися. Кількість повторень блоків (полів) визначаються коментарем у стовпці «Повтор блоку (поля)». У блоці може повторюватися останнєПоле або група суміжних полів, включаючи останнє Поле.

Існує такі формати полів:

- елементарний;
- складовий.

Елементарне Поле складається з одного фрагмента даних.

Складове Поле складається з кількох фрагментів даних. Порядок формування якого вказано у примітці до шаблону.

Довжина поля відображається у стовпці "Розмірність". Передбачені такі типи полів:

Char – символічне значення. При цьому вказується довжина поля та які значення використовуються: A-Z – великі літери; a-z – малі літери; SPC – спеціальні символи; 0-9 цифр;

DTG – міжнародний формат дати;

Figs – цілі числа із зазначенням довжини поля та діапазону значень;

Real – дійсні числа із зазначенням довжини поля, діапазоном значень та кількістю знаків після коми.

При використанні символічних полів заборонено використовувати такі символи: "/", "/". Винятком використання символів "/", "/" є поля "Free text" у блоках GENTEXT, AMPN, NARR та RMKS. При цьому використання "/" дозволяється лише спільно з ":" у наступному вигляді "://". У цьому блоках заборонено використовувати «:» перед символом закінчення блоку «//».

У форматі набору можуть вказуватися два або більше альтернативних форматів полів для одного і того ж формату поля у форматі лінійного або стовпчастого набору. Ці поля відомі як альтернативні поля вмісту. У цих випадках дескриптор поля може бути вказаний у конструкції набору, щоб допомогти читачеві у визначенні альтернативи, яку використовував автор повідомлення, та усунути ризик неоднозначності. Зазвичай при розробці повідомлення для альтернативи, що найчастіше використовується, не призначається дескриптор поля у форматі. Наприклад, на рис. 2-10 Alt A поля 2 розглядався як варіант, що найчастіше використовується, тому ніякий дескриптор поля не був призначений, де Alt B поля 2 має дескриптор поля «TYPE». В екземплярі повідомлення, якщо був використаний Alt B, він виглядатиме так: /TYPE:LTR/.

Деякі типи інформації в полі можуть бути різними способами. Наприклад, розташування може бути вказано як географічні координати,



пеленг, діапазон або назву місця. У цих випадках використовується таблиця дескрипторів, які вказані у примітці шаблону набору. У цих випадках дескриптор вказується в конструкції поля та визначає спосіб введення інформації обраного оператором. При використанні дескрипторів кількість символів не впливає на розмір поля. Між дескриптором і Поле, якщо дескриптор не порожнє місце, ставиться символ «:».

Деякі поля використовують таблиці вибору значень, які вказані в примітці до шаблону.

У структурі тексту повідомлення можна використовувати додаткові блоки: AMPN, NARR та RMKS. Ці блоки містять лише одне текстове Поле "Free text". Ці блоки не можуть повторюватися.

Блок AMPN використовується для надання додаткової інформації щодо попереднього блоку.

Блок NARR використовується для надання інформації перед сегментом.

Блок RMKS використовується для надання додаткової інформації до тексту повідомлення. При використанні він буде останнім блоком у тексті повідомлення.

Залежно від структури шаблону повідомлення та необхідної додаткової інформації блоком AMPN може йти блок NARR або RMKS, також за блоком NARR може йти блок RMKS. При цьому блок AMPN не може йти за блоком NARR або GENTEXT.

Додаткові блоки можна використовувати лише між блоками. При використанні блоку шаблону у вигляді таблиці додаткові блоки можуть вставлятися тільки після закінчення таблиці.

Шаблон и кодограма заголовка повідомлення і додаткових блоків приведені у додатку Б. Приклад шаблону повідомлення відповідно стандарту APP-11 приведений у додатку В пояснювальної записки.

## 2 ПОСТАНОВКА ЗАДАЧІ

### 2.1 Мета та задачі дослідження

На основі аналізу предметної області сформульована мета дослідження: розробка інформаційної технології для автоматизованого формування повідомлень за шаблонами стандарту APP-11 NATO Message Catalogue Edition D Version 1. Розроблена та впроваджена інформаційна технологія надасть програмний інструменту для оперативного формування повідомлення за встановленими шаблонами, які за структурою та змістом відповідають стандарту НАТО APP-11.

Для досягнення поставленої мети необхідно виконати такі задачі:

- розробити інформаційну технологію технології для автоматизованого формування повідомлень;
- обрати програмні засоби реалізації запропонованої інформаційної технології;
- провести моделювання функціонування відповідної програмної реалізації інформаційної технології, що розробляється;
- розробити модель даних для відображення складеної структури шаблону повідомлення за стандартом APP-11;
- розробити програмний додаток на основі бази даних; програмний додаток має реалізовувати запропоновану інформаційну технологію.

Інформаційна технологія має бути реалізована у форматі програмного додатку.

Функціональні вимоги до програмної реалізації:

- імпорт шаблонів повідомлення, визначених стандартом APP-11, до бази даних інформаційної технології;
- вибір шаблонів повідомлення з переліку збережених в базі даних шаблонів для створення повідомлення;

- редагування обраного шаблону повідомлення для створення формалізованого повідомлення;
- збереження готового формалізованого повідомлення в пов'язаних об'єктах бази даних;
- відкриття готового повідомлення для редагування;
- надсилання готового формалізованого повідомлення адресу (ам);
- валідація отриманого формалізованого повідомлення з метою визначення відповідності шаблону повідомлення, визначеного стандартом APP-11, та збереження результатів валідації і отриманого повідомлення;
- перегляд отриманого повідомлення.

Нефункціональні вимоги до додатку:

- адаптивність бази даних до змін в структурі шаблону повідомлення, що можуть бути спричинені змінами в стандартах НАТО для формування формалізованих повідомлень;
- темний колір оформлення візуальних форм інтерфейсу користувача (з врахуванням умов використання додатків, в які впроваджені інформаційна технологія);
- можливість швидкої зміни локалізації даних та інтерфейсу користувача;
- підтримка безпеки доступу до даних на рівні авторизації в системі, що реалізує інформаційну технологію;
- система повинна працювати на базі операційної системи Linux з використанням кросплатформної системи управління базою даних.

## 2.2 Засоби реалізації

Огляд публікацій в предметній області визначив відсутність подібних аналогів на момент розроблення інформаційної технології. Тому здобувачем прийнято рішення самостійно вибрати засоби реалізації, які б дозволили

реалізувати комплексну модель даних для відображення універсальної структури шаблону повідомлення відповідно стандарту APP-11, враховуючі всі особливості формування повідомлення.

Для реалізації моделі даних була обрана об'єктно-реляційна система управління базами даних PostgreSQL [5, 6], використання якої дозволить імпортувати шаблони повідомлень в форматі XML та розкласти останні на структурні блоки шаблону в базу даних. Для побудови моделі даних шаблону повідомлень була обрана комбінація реляційної технології бази даних та об'єктно-реляційний паттерн проектування Entity-Attribute-Value.

Потужність реляційної моделі даних [7, 8, 9] дозволить реалізувати цілісне відображення структурних компонентів формалізованого повідомлення, а технологія паттерну метамодельювання EAV [10, 11, 12, 13, 14] дозволить реалізувати блочно-модульну структуру формування повідомлення на основі шаблонів формалізованих повідомлень, структура і алгоритм формування яких визначається стандартом НАТО APP-11.

Прийнято рішення реалізувати логіку функціонування інформаційної технології на рівні логіки бази даних в максимально можливому обсязі.

Для практичної реалізації інформаційної технології буде використане середовище розробки QT 4.8, gcc та інтегроване середовище розробки Qt Creator [15, 16, 17]. Базові мови програмування C++ [18, 19, 20, 21], pg/SQL [6].

### **3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ**

#### **3.1 Структурно функціональне моделювання формування повідомлення за шаблоном стандарту НАТО**

Метод структурованого системного аналізу та проектування (SSADM) є широко використовуваним методом розробки комп'ютерних програм та інформаційних систем та технологій. SSADM ділить проект розробки програми на модулі, етапи, етапи та завдання, а також забезпечує структуру для опису проектів у спосіб, який підходить для керування проектом. SSADM – це водоспадний метод для аналізу та проектування інформаційних систем. Він реалізує каскадний погляд на розвиток системи, у якому є ряд кроків, кожен з яких веде до наступних кроків:

У SSADM використовуються декілька технологій, одна з яких реалізована в стандарті функціонального моделювання IDEF0, що підтримується Національним інститутом стандартів і технологій США (NIST) [22].

Модель IDEF0 використовує два основні графічні елементи: прямокутний блок та стрілки. Прямокутний блок містить назву та номер і використовується для подання функції або діяльності. Стрілка – це спрямована лінія, що складається з одного або кількох сегментів стрілки, яка моделює відкритий канал або канал, що передає дані чи об'єкти від джерела (без наконечника стрілки) до використання (зі наконечником стрілки). Існує чотири класи стріл: вхідні дані, вихідні дані, управління (контроль) та механізми.

Результати функціонального проектування інформаційної технології формування формалізованого повідомлення за шаблоном стандарту НАТО представлені двома діаграмами на контексному рівні та першому рівні декомпозиції для процесу формування повідомлення на основі збереженого

шаблону формалізованого повідомлення. Функціональне призначення інформаційної технології полягає у забезпеченні підготовки, прийому та перегляду військових повідомлень відповідно до стандартів НАТО; формуванні вихідних повідомлень, обробці вхідних повідомлень (ідентифікацію та парсинг) та перетворенні військових повідомлень. Також програма забезпечує доступ до бази даних шаблонів військових повідомлень і до бази даних повідомлень.

Контексна діаграма представлена на рис. 3.1 та відображає основний процес технології – формування повідомлення за шаблоном НАТО. Вхідними даними технології є:

- назва шаблону повідомлення за стандартом НАТО, перелік який описаний в п.1.2 пояснювальної записки;
- змістовна частина повідомлення, яка має бути передана формалізованим повідомленням;
- нове отримане формалізоване повідомлення за шаблоном стандарту НАТО.

Елементами керування є:

- Стандарт APP-11 NATO Message Catalogue Edition D Version 1 November 2015, який визначає шаблони формалізованих повідомлень;
- Стандарт ADatP-3 NATO Message Text Formatting System (Formets) – Concept of Formets (Conformets);
- База даних шаблонів формалізованих повідомлень.

Механізмами є:

- Технічні засоби;
- База даних повідомлень;
- Модуль формування повідомлення;
- Модуль перегляду та редагування повідомлення.

На виході системи отримуємо:

- Збережене формалізоване повідомлення;
- Відображене повідомлення.



Рисунок 3.1– Контекстна діаграма інформаційної технології

*Джерело: розроблено автором*

На рисунку 3.2 наведена діаграма декомпозиції 1-го рівня для процесу «Формування повідомлення за шаблоном стандарту НАТО» (A0), що розкладається на такі процеси:

- Вибір шаблону формалізованого повідомлення (A0.1)
- Заповнення полів шаблону повідомлення (A0.2)
- Збереження повідомлення (A0.3)
- Відкриття повідомлення (A0.4)
- Редагування повідомлення (A0.5)
- Збереження отриманого повідомлення (A0.6)

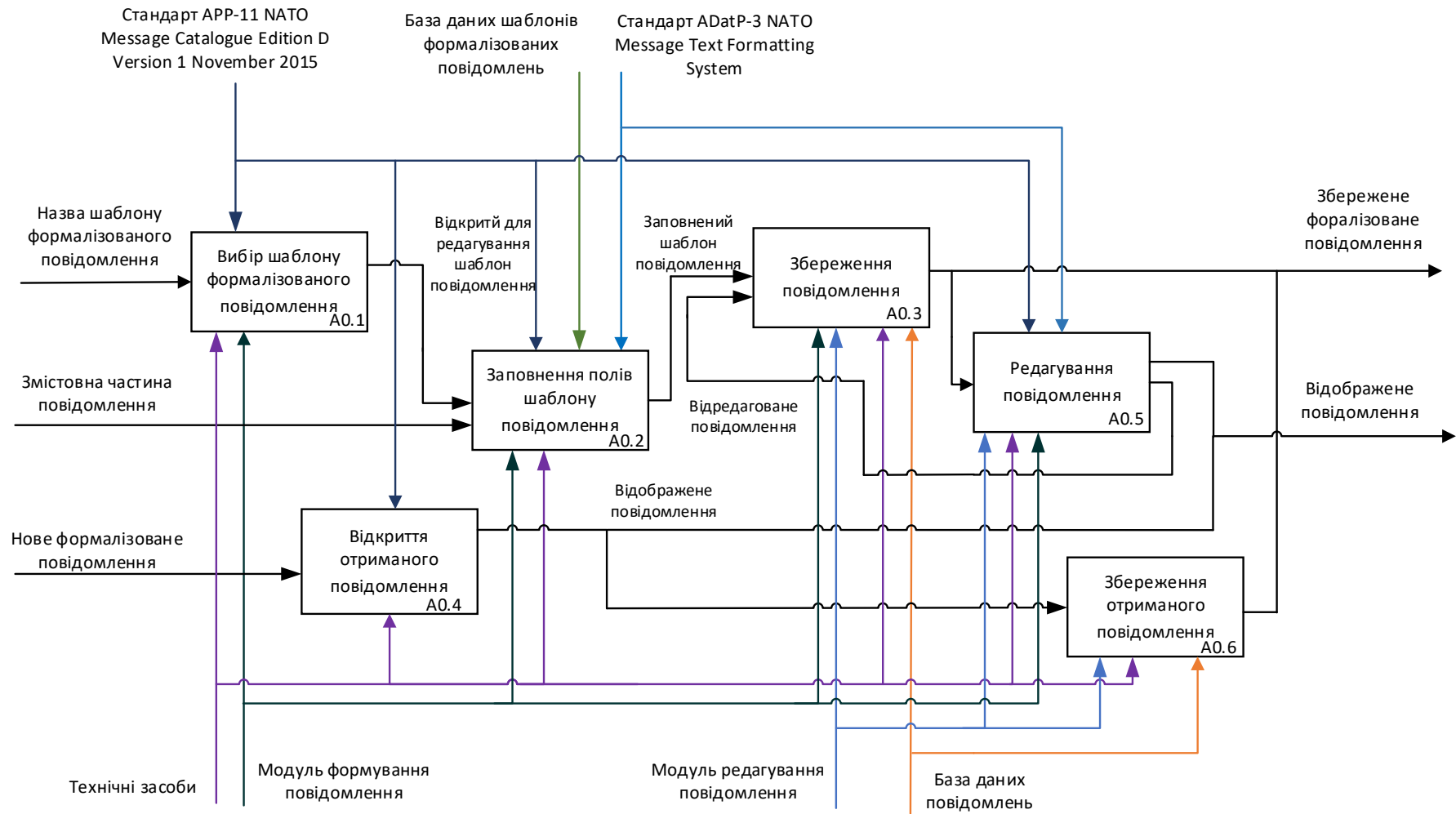


Рисунок 3.2– Діаграма декомпозиції 1-го рівня

*Джерело: розроблено автором*



## 3.2 Моделювання варіантів використання

Уніфікована мова моделювання (UML) – це мова моделювання загального призначення в галузі розробки програмного забезпечення, засоби якої надають стандартний спосіб візуалізації дизайну системи. UML є, безумовно, найбільш широко використовуваною нотацією як для розробки вимог, так і для проектування. [23].

Use Case diagram (діаграма варіантів використання) описує набір дій (варіантів використання), які система повинна виконувати у співпраці з одним або декількома зовнішніми користувачами або системами (акторами), щоб забезпечити певні спостережувані та цінні результати для суб'єктів або інших зацікавлених сторін системи.

На рис. 3.3 проілюстрована діаграма варіантів використання інформаційної технології формування повідомлення за шаблонами повідомлень стандарту НАТО APP-11. Для досліджуваної технології визначено таких акторів (два актори):

- Оператор: користувач, який може вибрати потрібні шаблони повідомлень з бази даних шаблонів формалізованих повідомлень стандарту НАТО, редагувати поля даних для параметрів шаблону повідомлень, зберігати проміжний екземпляр напівготового повідомлення в базі даних повідомлень та відкривати його на повторне редагування, збереження фінального екземпляру формалізованого повідомлення в базі даних повідомлень;
- PostgreSQL: система керування базою даних, яка використовується для всіх варіантів використання, що передбачають роботу із записами бази даних шаблонів формалізованих повідомлень та базою даних сформованих повідомлень.

Наступні варіанти використання (шість варіантів):

- Пошук шаблону формалізованого повідомлення в базі даних шаблонів формалізованих повідомлень;
- Перегляд структури шаблону формалізованого повідомлення;
- Редагування шаблону повідомлення;
- Збереження повідомлення в базі даних повідомлень;
- Відкриття збереженого повідомлення в базі даних повідомлень.

Такі артефакти (два артефакти):

- Шаблони формалізованих повідомлень за стандартом НАТО;
- Повідомлення.

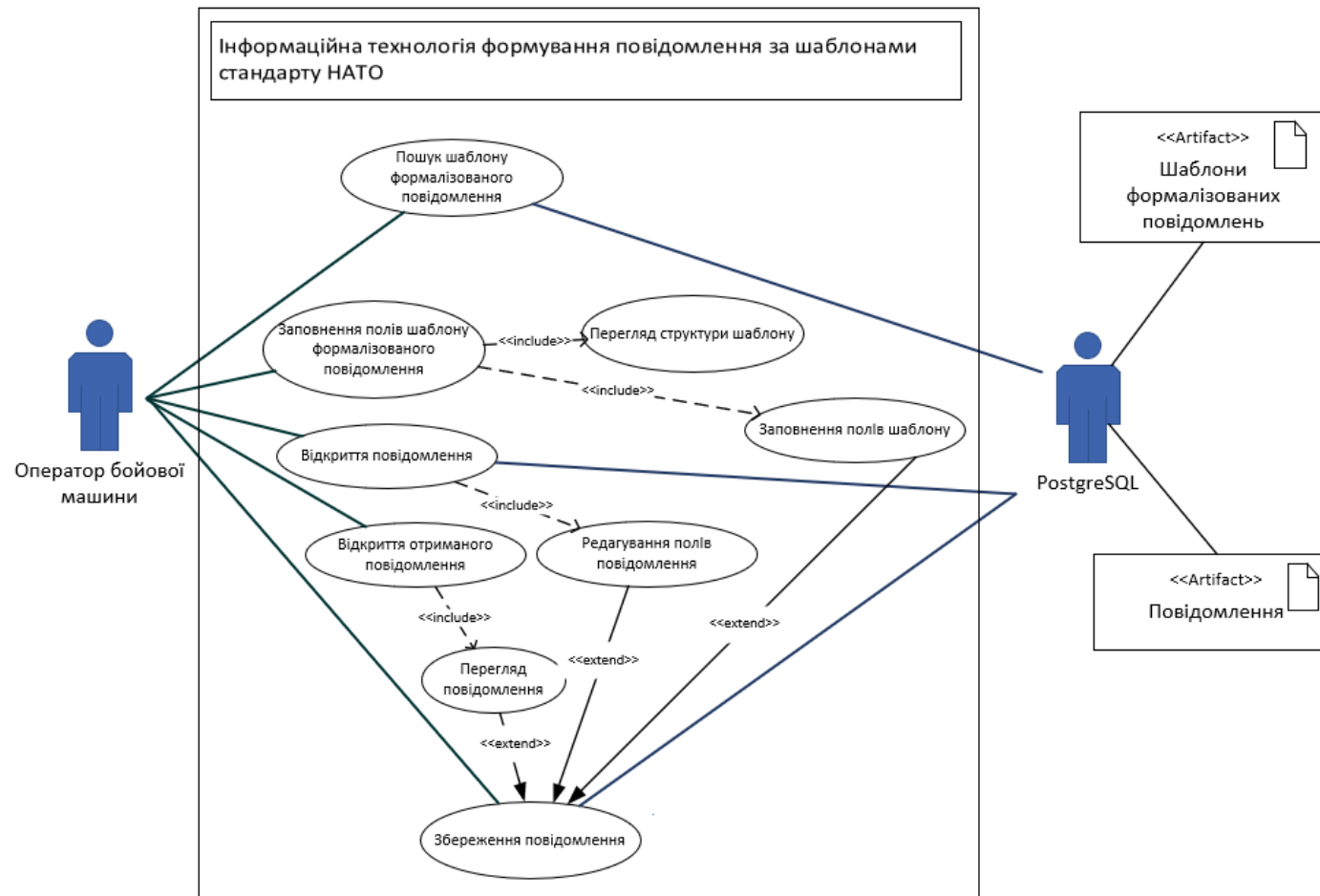


Рисунок 3.3– Діаграма варіантів використання інформаційної технології

Джерело: розроблено автором

### 3.3 Проектування моделей баз даних

База даних призначена для зберігання інформації про структуру шаблонів повідомлень, блоків та параметрів блоків для кожного шаблону повідомлення відповідно до стандарту версії APP-11 НАТО, параметрів та підпараметрів відповідно до реалізованих мов у системі.

#### 3.3.1. Перелік і призначення таблиць бази даних dbtemplates.

Список таблиць бази даних dbTemplates представлений у таблиці 3.1.

Таблиця 3.1 – Список таблиць бази даних dbTemplates

№	Назва таблиці	Призначення таблиці
1	languages	Таблиця зберігає інформацію про перелік мов інтерфейсу системи
2	msgid	Таблиця зберігає інформацію про перелік шаблонів повідомлень
3	heads	Таблиця зберігає інформацію про структуру заголовків шаблонів повідомлення
4	templates	Таблиця зберігає інформацію про структуру шаблонів повідомлення
5	blocks	Таблиця зберігає інформацію про блоки шаблонів повідомлення
6	parametrs	Таблиця зберігає інформацію про параметри блоків шаблонів повідомлення
7	options	Таблиця зберігає інформацію про перелік варіантів значень, допустимих для параметрів блоків шаблонів повідомлень
8	suboptions	Таблиця зберігає інформацію про перелік внутрішніх варіантів значень допустимих для параметрів блоків повідомлень, які можуть мати значення лише з допустимого переліку
9	data_type	Таблиця зберігає інформацію про типи даних, допустимі для значень параметрів блоків шаблонів повідомлень
10	relation_bloks_parametrs	Таблиця зберігає інформацію про залежність переліку параметрів у блоці від значення параметра іншого параметра у поточному шаблоні повідомлення
11	relation_parametrs_parametrs	Таблиця зберігає інформацію про залежність значення параметра від параметрів поточного або інших блоків поточного шаблону повідомлення

## Продовження таблиці 3.1

№	Назва таблиці	Призначення таблиці
12	trnslt_msgid	Таблиця зберігає переклади назв та аббревіатур шаблонів повідомлень
13	trnslt_heads	Таблиця зберігає переклади структурних елементів заголовків шаблонів повідомлень
14	trnslt_tmpl	Таблиця зберігає переклади структурних елементів шаблонів повідомлень
15	trnslt_blocks	Таблиця зберігає переклади назв та структурних елементів блоків шаблонів повідомлень
16	trnslt_parametrs	Таблиця зберігає переклади назв структурних елементів параметрів блоків шаблонів повідомлень
17	trnslt_options	Таблиця зберігає переклади назв допустимих значень параметрів блоків шаблонів повідомлень
18	trnslt_suboptions	Таблиця зберігає переклади назв внутрішніх допустимих значень для допустимих значень параметрів блоків шаблонів повідомлень
19	trnslt_data_type	Таблиця зберігає переклади найменувань допустимих даних типів для значень параметрів блоків шаблонів повідомлень
20	templates_in_xml	Таблиця зберігає дані про шаблон у форматі xml
21	block_teg_xml	Таблиця зберігає дані про теги блоків для xml-документа
22	parametr_teg_xml	Таблиця зберігає дані про теги параметрів для xml-документа
23	xml_version	Таблиця зберігає дані про версію xml-документа
24	mtf_config	Таблиця зберігає дані про конфігурацію заголовка xml-документа
25	classification_groups	Таблиця зберігає групи класифікацій шаблонів повідомлень
26	classifier	Таблиця зберігає належність шаблонів до груп класифікацій

*Джерело: побудовано автором*

### 3.3.2 Структура таблиць бази даних dbtemplates

База даних шаблонів повідомлень зберігає інформацію в 26 таблицях, структура яких представлена в додатку Д пояснювальної таблиці.

При додаванні нової мови інтерфейсу до таблиць Д.12 – Д.19 додається нове поле з аббревіатурою мови та переклад значень описаних параметрів.

### 3.3.3. Зв'язки між таблицями бази даних dbtemplates.

У базі даних основні зв'язки між таблицями реалізовані за принципом цілісності посилання згідно таблиці 3.2. Зовнішній ключ підпорядкованої таблиці посилається на первинний ключ головної таблиці. Зв'язок налаштований таким чином, що при видаленні запису з головної таблиці відбувається каскадне видалення всіх рядків підпорядкованої таблиці, в яких зовнішній ключ посилається на первинний ключ запису головної таблиці, що видаляється. Аналогічно налаштовано каскадне оновлення даних при зміні даних у зв'язаних таблицях.

Таблиця 3.2 – Опис зв'язків між основними таблицями бази даних dbTemplates

Головна таблиця	Підлегла таблиця	Зовнішній ключ підлеглої таблиці
msgid	heads	msgid_id
msgid	templates	msgid_id
blocks	templates	block_id
blocks	parametrs	block_id
data_type	parametrs	type_id
parametrs	options	parameter_id
options	suboptions	option_id
blocks	relation_blocks_parametrs	block_id
parametrs	relation_blocks_parametrs	parametr_id
parametrs	relation_parametrs_parametrs	parametr_id
parametrs	relation_parametrs_parametrs	relay_id
msgid	templates_in_xml	msgid_id
msgid	mtf_config	msgid_id
parametrs	parametr_teg_xml	parametr_id
blocks	block_teg_xml	block_id
msgid	classifier	msgid_id
classification_groups	classifier	classification_group_id

*Джерело: побудовано автором*

Зв'язки між основними таблицями та таблицями перекладів реалізовані за допомогою тригерних функцій за принципом цілісності посилання: при видаленні запису з основної таблиці відбувається видалення запису з таблиці перекладу, де значення поля key посилається на відповідні значення полів основної таблиці.

### 3.3.4. Перелік і призначення таблиць бази даних dbmessages.

База даних dbMessage призначена для зберігання інформації про вхідні та вихідні повідомлення, структура яких відповідає шаблонам повідомлень, затверджених стандартом версії APP-11 NATO Message Catalogue Edition Version 1. November 2015.

Дані до бази даних заносяться автоматично в програмному режимі після перевірки відповідності стандартним шаблонам. Після успішної перевірки повідомлення структурується по блоках згідно з шаблоном та записується в таблиці бази даних. Одночасно відбувається запис результатів перевірки повідомлення.

Список таблиць бази даних dbMessages представлений у таблиці 3.3.

Таблиця 3.3 – Список таблиць бази даних dbMessages

№	Назва таблиці	Призначення таблиці
1	messages	Таблиця зберігає загальну інформацію про повідомлення та саме повідомлення у повному обсязі
2	service_heads	Таблиця зберігає інформацію про службову частину повідомлення
3	adress	Таблиця зберігає інформацію про перелік адрес, на які може бути надіслано або з яких може бути отримано повідомлення
4	adress_groups	Таблиця зберігає інформацію про перелік адресних груп, на які може бути відправлено або з яких може бути отримане повідомлення
5	adress_book	Таблиця зберігає інформацію про входження адрес до адресних груп
6	messages_edited	Таблиця зберігає інформацію повідомлення та самі повідомлення, які були відредаговані після отримання
7	blocks	Таблиця зберігає інформацію про блоки повідомлень
8	parameter_value	Таблиця зберігає інформацію про значення параметрів блоків повідомлень
9	call_station	Таблиця зберігає пріоритети повідомлень для станції передачі
10	category_message	Таблиця зберігає інформацію про категорії безпеки повідомлень
11	priority	Таблиця зберігає інформацію про класифікацію таємності та пріоритет повідомлень
12	type_message	Таблиця зберігає інформацію про типи повідомлень

## Продовження таблиці 3.3

№	Назва таблиці	Призначення таблиці
13	trnslt_call_station	Таблиця зберігає переклад станцій, що викликаються, на мови інтерфейсу
14	trnslt_category_message	Таблиця зберігає переклад категорій безпеки повідомлень мовами інтерфейсу
15	trnslt_priority	Таблиця зберігає переклад рівнів секретності та пріоритету мовами інтерфейсу
16	trnslt_type_message	Таблиця зберігає переклад типів повідомлень мовами інтерфейсу

*Джерело: побудовано автором*

База даних шаблонів повідомлень зберігає інформацію у 16 таблицях, структура яких представлена у таблицях Е.1 – Е.16 додатку Е. При додаванні нової мови інтерфейсу до таблиць додається нове поле з аббревіатурою мови та переклад значень описаних параметрів.

#### 3.3.5. Зв'язки між таблицями бази даних dbmessages.

У базі даних основні зв'язки між таблицями реалізовані за принципом цілісності посилення згідно таблиці 3.4. Зовнішній ключ підпорядкованої таблиці посиляється первинний ключ головної таблиці. Зв'язок налаштований таким чином, що при видаленні запису з головної таблиці відбувається каскадне видалення всіх рядків підпорядкованої таблиці, в яких зовнішній ключ посиляється на первинний ключ запису головної таблиці, що видалається. Аналогічно налаштовано каскадне оновлення даних при зміні даних у зв'язаних таблицях.

3.3.6. Зв'язки між таблицями допустимих значень параметрів сервісного заголовка і таблицями перекладів бази даних dbmessages.

Зв'язки між таблицями допустимих значень параметрів сервісного заголовка та таблицями перекладів реалізовані за допомогою тригерних функцій за принципом цілісності посилення: при видаленні запису з головної таблиці відбувається видалення запису з таблиці перекладу, де значення поля key посиляється на відповідні значення полів головної таблиці.



Таблиця 3.4 – Опис зв'язків між основними таблицями бази даних dbMessages

Головна таблиця	Підпорядкована таблиця	Зовнішній ключ таблиці
messages	service_heads	message_id
messages	messages_edited	message_id
messages	blocks	message_id
service_heads	adres_groups	shead_id
adres	adres_book	adres_id
adres_groups	adres_book	group_id
blocks	parameter_value	block_id

*Джерело: побудовано автором*

Перелік тригерів для реалізації описаних зв'язків подано у таблиці 3.5.

Таблиця 3.5 – Опис тригерів і тригерних функцій реалізації зв'язків між допустимих значень параметрів сервісного заголовка і таблицями переклада

Назва тригера	Назва тригерної функції	Основна таблиця	Таблиця перекладу, пов'язана з основною
call_station_delete	trigger_call_station_delete_translate	call_station	trnslt_call_station
category_message_delete	trigger_category_message_delete_translate	category_message	trnslt_category_message
priority_delete	trigger_priority_delete_translate	priority	trnslt_priority
type_message_delete	trigger_type_message_delete_translate	type_message	trnslt_type_message

*Джерело: побудовано автором*

Проілюструємо код тригерної функції на прикладі тригерної функції `trigger_call_station_delete_translate` для реалізації зв'язку між таблицями `call_station` та `trnslt_call_station`:

```
CREATE OR REPLACE FUNCTION public.trigger_call_station_delete_translate()
  RETURNS trigger AS
  $BODY$
begin
delete from trnslt_call_station where trim(trnslt_call_station.key)=trim(OLD.mearning);
return OLD;
end;
```

```
$BODY$  
LANGUAGE plpgsql VOLATILE  
COST 100;  
ALTER FUNCTION public.trigger_call_station_delete_translate()  
OWNER TO postgres;
```

Тригер `call_station_delete` спрацьовує перед видаленням запису з таблиці `call_station`, викликає функцію `trigger_call_station_delete_translate`. Код тригера:

```
CREATE TRIGGER call_station_delete  
BEFORE DELETE  
ON public.call_station  
FOR EACH ROW  
EXECUTE PROCEDURE public.trigger_call_station_delete_translate();
```

## 4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

Операційна система, необхідна для функціонування програми редагування та створення шаблону військових повідомлень – Rad Hat Enterprise Linux 7 (як така, що використовується в автоматизованих робочих місцях різних віськових підрозділів) [24].

В операційній системі має бути встановлений драйвер QPSQL driver.

1. Якщо програмне забезпечення редактора шаблонів військових повідомлень функціонує на комп'ютері, який використовується як клієнтська консоль, вимоги до іншого програмного забезпечення не висуваються.

2. Якщо програмне забезпечення редактора шаблонів військових повідомлень функціонує на комп'ютері, який використовується як сервер бази даних, необхідно, щоб в операційній системі було встановлено PostgreSQL 9.4

### 4.1 Реалізація бази даних шаблонів повідомлень dbTemplates

Модель даних, описана в підрозділі 3.3, реалізована засобами системи управління базами даних postgresql. Доступ до засобів СУБД забезпечений за допомогою програми pgAdmin III - кроссплатформенної системи управління СУБД PostgreSQL для ОС Unix та Windows.

pgAdmin III вільно доступний відповідно до умов ліцензії PostgreSQL. PgAdmin III написаний на C++ і використовує відмінний кроссплатформовий інструментарій wxWidgets (раніше wxWindows). З'єднання з PostgreSQL виконується за допомогою власної бібліотеки libpq [25].

Схема бази даних представлена на рисунку 4.1.

#### 4.1.1. Порядок заповнення бази даних dbTemplates.

База даних повинна заповнюватись у певній послідовності від головної таблиці до підлеглих згідно зі схемою, наведеною на рис.4.1, використовуючи ідентифікатори для формування зв'язків.

Порядок заповнення таблиць бази даних описаний далі на прикладі запитів на вставку даних до таблиць msgid и trnslt\_msgid.

При заповненні значень первинних ідентифікаторів використовується стандартне значення (default), яке автоматично генерується створеною послідовністю (sequences). Послідовності є унікальними для кожного первинного ідентифікатора таблиці.

При заповненні значень посилальних ключів на таблиці перекладів значень параметрів застосовується наступний принцип: ключ формується за структурою: "послідовність зовнішніх ключів на основні таблиці. первинний ідентифікатор таблиці. назва поля". Посилальний ключ генерується на рівні програми без участі користувача програми.

Вставка даних відбувається у три етапи.

1 етап – вставка нового рядка в таблицю msgid із поверненням нового ідентифікатора під умовним ім'ям returning msgid. Приклад запиту на вставку нового рядка:.

```
insert в msgid (msgid_id, ms_title, ms_abb, blocks_amount, standart_version)
values (default, "", "", 0, default)
returning msgid_id;
```

2 етап – оновлення створеного рядка на додавання значення ключів посилань на таблицю перекладу і кількості блоків в оброблюваному шаблоні. Ключі формуємо з повернутого первинного ідентифікатора рядка та назви параметра за прикладом 121.title та 121.abb. Приклад запиту на оновлення рядка:

```
update msgid
set (ms_title, ms_abb, blocks_amount) = ('returning msgid.title', 'returning msgid.abb', 7)
where msgid_id=returning msgid;
```

3 етап – вставка нового рядка в таблицю перекладу trnsl\_msgid з ключами посилань з таблиці msgid і значенням перекладу на відповідну мову.

Приклад запиту:

```
insert into trnslt_msgid (id, key, ua)
values (default, 'returning msgid.title', 'barrier report'),
(default, 'returning msgid.abb', 'barrep');
```

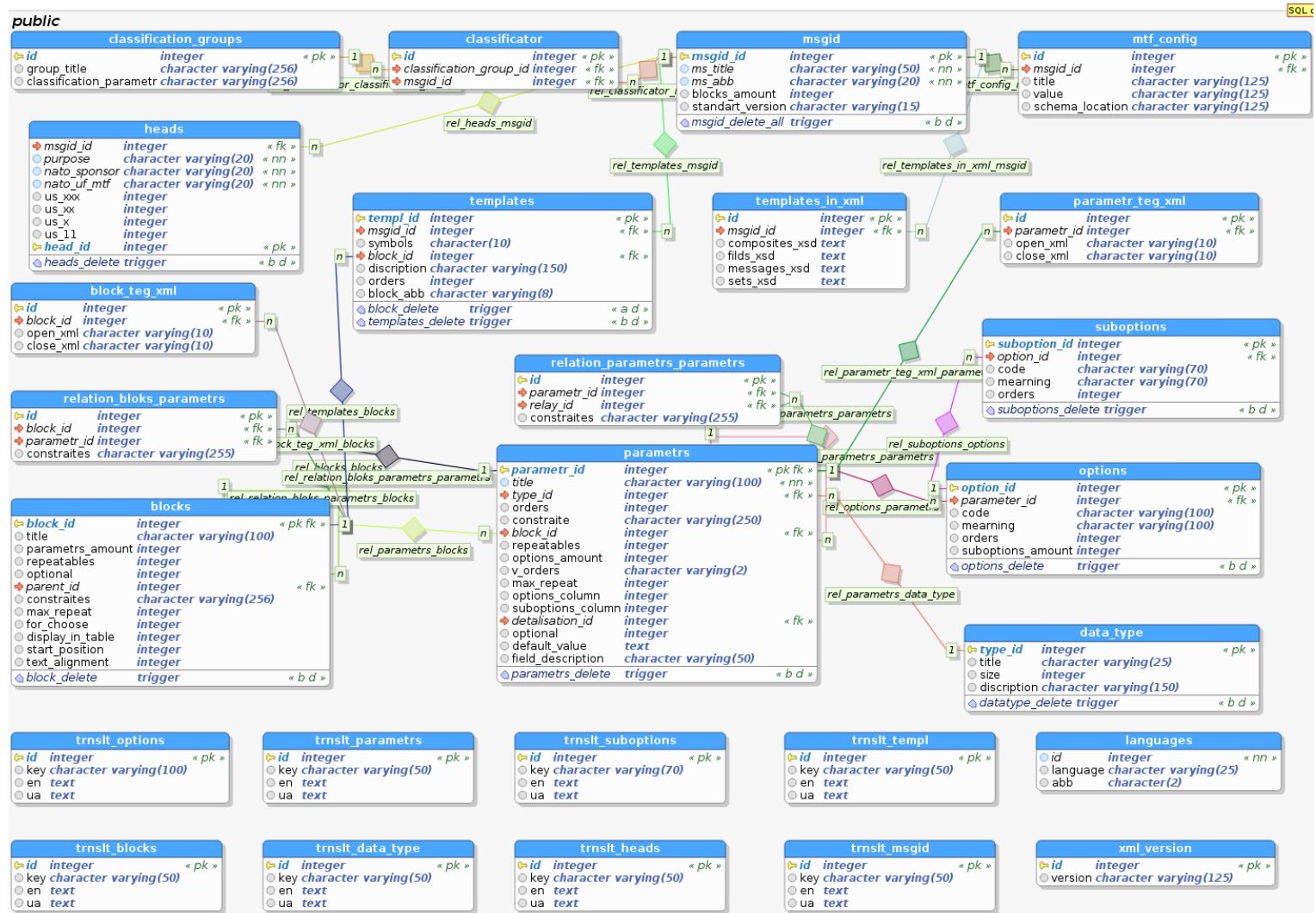


Рисунок 4.1 – Схема бази даних dbTemplates

Джерело: розроблено автором

#### 4.1.2. Опис взаємозв'язків між таблицями бази даних dbTemplates.

Оскільки таблиці перекладів є динамічними, то зв'язки між відповідними основними таблицями та таблицями перекладів були реалізовані тригерними функціями, які перелічені в таблиці 4.1.

Таблиця 4.1 – Опис тригерів та тригерних функцій для реалізації зв'язків між основними таблицями та таблицями перекладу

Назва тригеру	Назва тригерної функції	Основна таблиця	Таблиця перекладу, пов'язана із основною
msgid_delete_all	trigger_msgid_delete_translate_all	msgid	trnslt_msgid
heads_delete	trigger_head_delete_translate	heads	trnslt_heads
block_delete	trigger_block_delete_translate	blocks	trnslt_blocks
templates_delete	trigger_templates_delete_translate	templates	trnslt_templates
parametrs_delete	trigger_parametrs_delete_translate	parametrs	trhslt_parametrs
options_delete	trigger_options_delete_translate	options	trnslt_options
suboptions_delete	trigger_suboptions_delete_translate	suboptions	trnslt_suboptions
datatype_delete	trigger_datatype_delete_translate	data_type	trnslt_data_type

*Джерело: побудовано автором*

Проілюструємо синтаксис однієї тригерної функції trigger\_msgid\_delete\_translate\_all для реалізації зв'язку між таблицями msgid та trnslt\_msgid:

```
CREATE OR REPLACE FUNCTION trigger_msgid_delete_translate_all()
  RETURNS trigger AS
  $BODY$
begin
if (select count(*) from trnslt_msgid tms where trim(tms.key)=trim(OLD.ms_title))>0
then delete from trnslt_msgid where trim(trnslt_msgid.key)=trim(OLD.ms_title);
end if;
if (select count(*) from trnslt_msgid tms where trim(tms.key)=trim(OLD.ms_abb))>0
then delete from trnslt_msgid where trim(trnslt_msgid.key)=trim(OLD.ms_abb);
end if;
return OLD;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION trigger_msgid_delete_translate_all()
  OWNER TO postgres;
```

Тригерна функція виконує порівняння ключів записів основної таблиці і зовнішніх ключів підлеглої таблиці для видалення всіх пов'язаних записів перекладів на всі доступні мови локалізації. Тригер спрацьовує перед видаленням запису з таблиці msgid, викликаючи відповідну функцію. Повний синтаксис всіх тригерних функцій приведений у додатку Д пояснювальної записки.

#### 4.1.3. Опис значень за замовченням для таблиць бази даних dbTemplates.

У базі даних dbTemplates передбачені стандартні значення для полів, описаних у таблиці 4.2, відповідно до вимог шаблонів формалізованих повідомлень.

Таблиця 4.2 – Стандартні значення для таблиць бази даних dbTemplates

№	Назва таблиці	Назва поля	Значення за замовченням	Опис значення за замовченням
1	blocks	optional	1	обов'язкове
2	blocks	max_repeat	0	без повторів
3	blocks	for_choose	0	без вибору
4	blocks	display_in_table	0	відображення текстом
5	blocks	start_position	0	без відступу
6	blocks	text_alignment	0	вирівнювання за лівим краєм
7	parametrs	max_repeat	0	без повторів
8	parametrs	options_column	2	відображення значень для вибору в дві колонки
9	parametrs	suboptions_column	2	відображення підзначень для вибору в дві колонки
10	parametrs	optional	1	обов'язкове
11	parametrs	field_description	null	не має опису поля
12	msgid	standart_version	APP-11(D)	поточна версія стандарту
13	mtf_config	title	GeneralInformationMessage	стандартний заголовок xml-документу
14	mtf_config	value	urn:nato:mtf:app-11(d):1:geninfomsg	поточне значення mtf-конфігурації для xml-документу

*Джерело: побудовано автором*

4.1.4. Опис обмежень на допустимі значення для полів таблиць бази даних dbTemplates.

Під час заповнення бази даних передбачено обмеження на значення полів таблиць, описані в таблицях. Обмеження обумовлені стандартом шаблонів формалізованих повідомлень та приведені в таблицях 4.3 – 4.5.

Таблиця 4.3 – Обмеження на значення полів у таблиці heads

№	Назва поля	Допустимі значення	Опис допустимих значень
1	us_xxx	0	не може використовуватись
		1	може використовуватись
2	us_xx	0	не може використовуватись
		1	може використовуватись
3	us_x	0	не може використовуватись
		1	може використовуватись
4	us_11	0	не може використовуватись
		1	може використовуватись

*Джерело: побудовано автором*

Таблиця 4.4 – Обмеження на значення полів у таблиці blocks

№	Назва поля	Допустимі значення	Опис допустимих значень
1	optional	0	не обов'язкове
		1	обов'язкове
2	for_choose	0	без вибору
		1	з вибором
3	display_in_table	0	відображення текстом
		1	відображення у вигляді таблиці
4	repeatables	0	без повторів
		1	може повторюватись
5	text_alignment	0	вирівнювати за лівим краєм
		1	вирівнювати за правим краєм

*Джерело: побудовано автором*



Таблиця 4.5 – Обмеження на значення полів у таблиці `parameters`

№	Назва поля	Допустимі значення	Опис допустимих значень
1	repeatables	0	без повторів
		1	може повторюватись
2	options_column	2	відображення значень в дві колонки
		4	відображення значень в чотирі колонки
3	suboptions_column	2	відображення підзначень в дві колонки
		4	відображення підзначень в чотирі колонки
4	optional	0	не обов'язкове
		1	обов'язкове

*Джерело: побудовано автором*

## 4.2 Реалізація бази даних сформованих повідомлень `dbMessage`

База даних `dbMessage` призначена для зберігання інформації про вхідні та вихідні повідомлення, структура яких відповідає шаблонам повідомлень, затверджених стандартом версії APP-11 NATO Message Catalogue Edition Version 1. November 2015.

Дані до бази даних заносяться автоматично в програмному режимі після перевірки відповідності стандартним шаблонам. Після успішної перевірки повідомлення структурується по блоках згідно з шаблоном та записується в таблиці бази даних. Одночасно відбувається запис результатів перевірки повідомлення.

Схема бази даних представлена на рисунку 4.2.

### 4.2.1. Порядок заповнення бази даних `dbMessages`.

База даних повинна заповнюватись у певній послідовності від головної таблиці до підлеглих згідно зі схемою, наведеною на рис.4.2, використовуючи ідентифікатори для формування зв'язків.

При заповненні значень первинних ідентифікаторів використовується стандартне значення (`default`), яке автоматично генерується створеною послідовністю (`sequences`). Послідовності є унікальними для кожного первинного ідентифікатора таблиці.

При заповненні значень посилальних ключів на таблиці перекладів значень параметрів застосовується наступний принцип: ключ формується за структурою: "послідовність зовнішніх ключів на основні таблиці. первинний ідентифікатор таблиці.назва поля". Посилальний ключ генерується на рівні програми без участі користувача програми.

Проілюструємо порядок заповнення таблиць даними на прикладі заповнення таблиці messages та підлеглої таблиці service\_heads.

Вставка даних відбувається в один етап: вставлення нового рядка в таблицю messages з поверненням нового ідентифікатора під умовним ім'ям returning\_messages\_id, який потім буде доданий до підлеглої таблиці як значення зовнішнього ключа. Приклад запиту на вставку нового рядка:

```
insert in messages (message_id, date_in, contents, validation, error_amount, error_description,
status, msgid_id_in_tmpl, msgid_abb_in_tmpl)
values (default, default, 'test content', default, "", "", 1, msgid_id, 'msgid_abb')
returning message_id;
```

Вставка даних в таблицю service\_heads відбувається в один етап: вставлення нового рядка в таблицю service\_heads з поверненням нового ідентифікатора під умовним ім'ям returning\_shead. Приклад запиту на вставку нового рядка:

```
insert в service_heads (service_heads, message_id, transfer_instruction, priority, sender,
receiver, receiver_info, exception, groups_amount, channel_name, channel_number,
message_priority, rout_station_id, depature_station_name, depature_station_name
values (default, returning message_id, 'instruction', 'priority', 'sender', 'receiver', default,
'exception', default, 'channel_name', number, 'priority', 'rout_station_id', 'depature_station
depature_station_code', 'submission_date_for_sending')
returning heads_id;
```

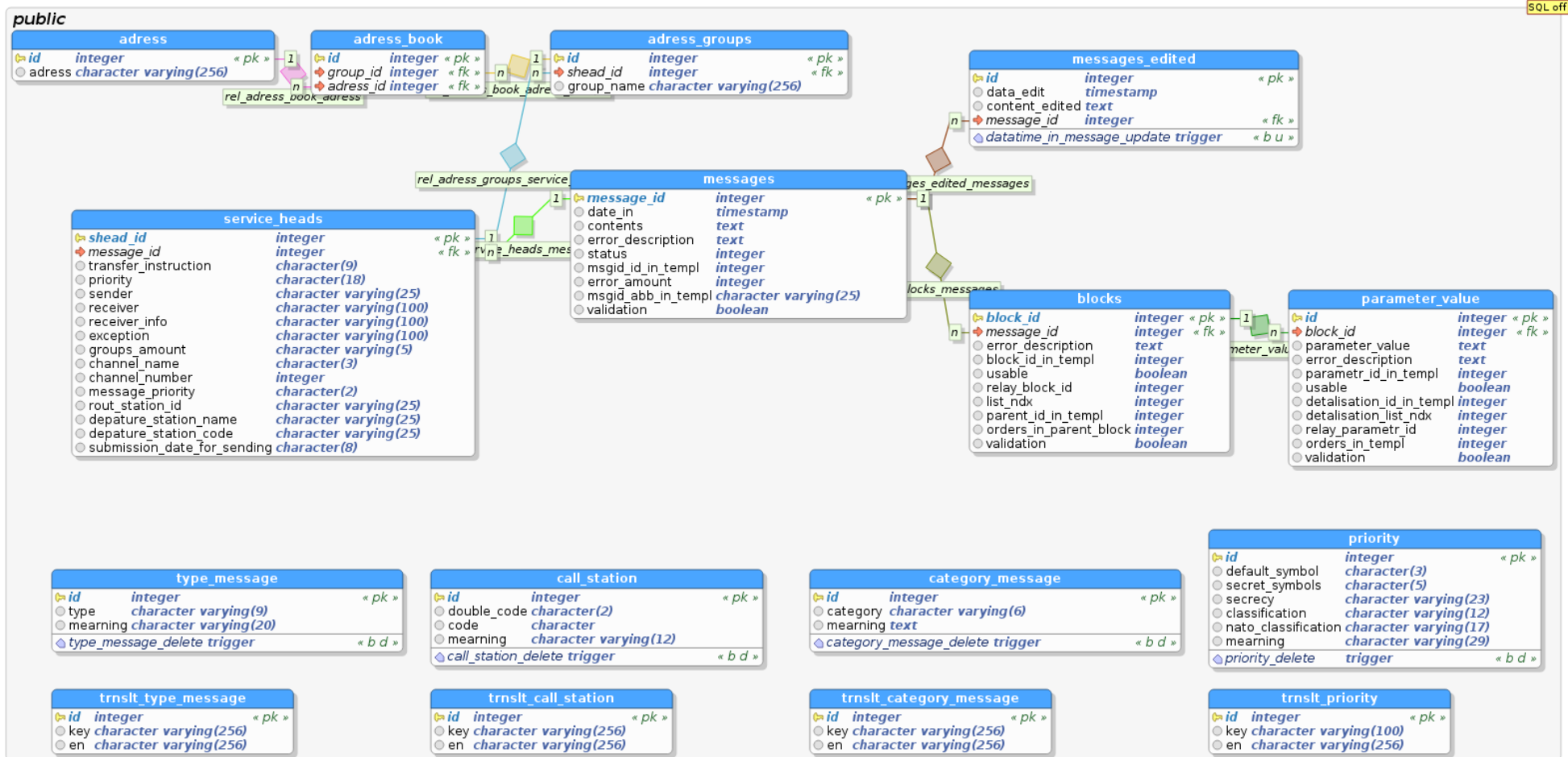


Рисунок 4.2 – Схема бази даних dbMessages

Джерело: розроблено автором

#### 4.2.2. Опис значень за замовчення для таблиць бази даних dbMessages.

У базі даних dbMessages передбачені стандартні значення для полів, описаних у таблиці 4.6.

Таблиця 4.6 – Стандартні значення для таблиць бази даних dbMessages

№	Назва таблиці	Назва поля	Значення за умовчанням	Опис значення за промовчанням
1	messages	date_in	now	поточна дата
2	messages	validation	false	помилка перевірки
3	messages_edited	data_edit	now	поточна дата
4	service_heads	receiver_info	NULL	відсутня на даний момент
5	service_heads	groups_amount	NULL	відсутня на даний момент
6	blocks	validation	false	помилка перевірки
7	blocks	relay_block_id	-1	немає зв'язаних блоків
№	Назва таблиці	Назва поля	Значення за умовчанням	Опис значення за промовчанням
8	parameter_value	detalisation_id_in_tmpl	-1	немає параметрів уточнення
9	parameter_value	detalisation_list_ndx	-1	немає уточнюючих параметрів
10	parameter_value	relay_parametr_id	-1	пов'язаних параметрів немає
11	parameter_value	validation	false	помилка перевірки

*Джерело: побудовано автором*

#### 4.2.3. Опис обмежень на допустимі значення для полів таблиць бази даних dbMessages.

При заповненні бази даних передбачені обмеження значення полів таблиць, описані в таблицях 4.7 – 4.11.

Таблиця 4.7 – Обмеження на значення полів у таблиці messages

№	Назва поля	Допустимі значення	Опис допустимих значень
1	status	0	редагується
		1	отримано
		2	надіслано
2	validation	0	помилка перевірки
		1	успішна перевірка

*Джерело: побудовано автором*

Таблиця 4.8 – Обмеження на значення полів у таблиці blocks

№	Назва поля	Допустимі значення	Опис допустимих значень
1	validation	false	помилка перевірки
		true	успішна перевірка
2	usable	false	блок не використовується
		true	блок використовується

*Джерело: побудовано автором*

Таблиця 4.9 – Обмеження значення полів у таблиці parameter\_value

№	Назва поля	Допустимі значення	Опис допустимих значень
1	validation	false	помилка перевірки
1	validation	true	успішна перевірка
2	usable	false	параметр не використовується
2	usable	true	параметр використовується

*Джерело: побудовано автором*

Для таблиці service\_heads перелік допустимих значень зберігається у базі даних як окремі таблиці згідно з описом у таблицях 4.10 – 4.11 далі.

Таблиця 4.10 – Обмеження значення поля message\_priority таблиці service\_heads

№	Назва таблиці, що містить допустимі значення	Назва поля, що містить допустимі значення	Допустимі значення	Опис допустимих значень
1	call_station	double_code	ZZ	Блискавка
			OO	Термінове
			PP	Пріоритет
			RR	Регламентне

*Джерело: побудовано автором*

Таблиця 4.11 – Обмеження на значення поля transfer\_instruction таблиці service\_heads

№	Назва таблиці, що містить допустимі значення	Назва поля, що містить допустимі значення	Допустимі значення	Опис допустимих значень
1	priority	secret_symbols	UUUUU	Не секретне
			RRRRR	Для обмеженого користування
			CCCCC	Конфіденційне
			SSSSS	Секретне
			TTTTT	Цілком таємно

*Джерело: побудовано автором*

#### 4.4 Реалізація модуля редактора шаблонів повідомлень

Програма TemplateConstructor призначена для створення та редагування шаблонів військових повідомлень.

Програма дозволяє створювати, редагувати, видаляти та зберігати шаблони військових повідомлень у базі даних. Підтримується багатомовний графічний інтерфейс користувача: англійська, українська, російська. Програма дозволяє створювати та зберігати шаблони військових повідомлень різними мовами. Кількість мов визначається користувачем.

#### 4.4.1. Модель даних програми.

Модель даних, що використовується у програмі, повторює модель даних шаблонів, прийняту в 4.1. Для опису структури шаблону застосовано підхід, який використовує принцип вкладених «контейнерів». Кожен блок (набір) шаблону розглядається як клас, одним із полів якого є вектор, що містить об'єкти, вкладені в цей блок. Вектор може зберігати як блоки такого самого типу, так і об'єкти типу «поле». Використання вкладених один одного контейнерів дозволяє повністю повторити схему побудови шаблону військового повідомлення.

Для опису структури типу блок використовується клас `atommess` з полями, описаними в таблиці 4.12.

Таблиця 4.12 – Перелік полів класу `atommess`

Тип даних	Назва поля	Значення
int	id_db_am	Значення ключа в БД (table blocks)
Tassorty	obj_arr	Вектор, що містить вкладені в блок об'єкти. Об'єкти, що містять вектор типу Tassorty, мають тип QObject
QString	Letter	Елемент, що містить номер блока в шаблоні
QString	abbr	Абревіатура
QString	Title	Назва блоку
bool	repeatable	Признак повторюваності блоку
int	max_repeat	Максимальне число повторів блоку. Якщо значення дорівнює 0, максимальна кількість повторів не обмежено (as necessary)
bool	relayed_	Признак залежності заповнення блоку від значення іншого поля
Ref_field	relay_b	Поле, від начення якого залежить заповнюваність блоку. Ref_field - тип даних, що описує поле, що реферується (пов'язане). Містить ім'я пов'язаного поля, його «шлях» в ієрархічній структурі шаблону та індекс у БД
QString	cond_rel_	Умова для заповнюваності поля
bool	optional	Признак того, що блок не бов'язковий для заповнення
bool	for_choose	true - у повідомленні з дочірніх об'єктів вибирається один для відображення та заповнення
bool	tbl	true - відображати вміст блоку таблицею; false - неприємний випадок
int	start_pos	Початкова позиція у кодограмі
bool	text_alig	true – вирівнювання ліворуч; false - вирівнювання праворуч

*Джерело: побудовано автором*

Для опису абстракції типу "поле" використовується клас `field_str`, що представляє конструкцію з полями, описаними в таблиці 4.13.

Таблиця 4.13 – Перелік полів класу `field_str`

Тип даних	Ім'я поля	Значення
Int	<code>id_db_fs</code>	значення ключа у БД ( <code>table params</code> )
QString	<code>let</code>	Вказівник поля (літери латинського алфавіту) При введенні вказівника поля цифра, що стоїть перед літерою, опускається
QString	<code>nm</code>	Назва поля (ім'я поля)
Bool	<code>repeat_</code>	Ознака повторюваності поля
Int	<code>num_repeat</code>	Максимальна кількість повторень поля. Якщо значення дорівнює 0, то максимальна кількість повторів не обмежена ( <i>as necessary</i> )
Int	<code>dt</code>	Індекс типу даних
Bool	<code>sel</code>	Ознака того, що значення поля вибирається зі списку
Tselect_info	<code>svi</code>	Список з якого проводиться вибір значень поля Tselect_info — тип даних, що є структурою, що містить значення кількості колонок, що відображаються користувачеві, і вектор значень з якого проводиться вибір значень поля
Bool	<code>reference</code>	Ознака того, що поле є «уточнюючим» для іншого поля шаблону
Ref_field	<code>rf</code>	Поле, значення якого уточнюється Ref_field - Тип даних, що описує поле, що реферується (пов'язане). Містить ім'я пов'язаного поля, його «шлях» в ієрархічній структурі шаблону та індекс у БД
Bool	<code>relayed</code>	Ознака залежності заповнення поля від значення іншого поля
Ref_field	<code>relay_f</code>	Поле, від значення якого залежить заповнюваність даного поля Ref_field - Тип даних, що описує поле, що реферується (пов'язане). Містить ім'я пов'язаного поля, його «шлях» в ієрархічній структурі шаблону та індекс у БД
QString	<code>cond_rel</code>	Умова заповнення поля
Bool	<code>option</code>	Ознака того, що поле не є обов'язковим для заповнення
QString	<code>defValue</code>	Значення поля за промовчанням
QString	<code>fieldDescr</code>	Дескриптор поля Зазвичай у шаблоні повідомлення для альтернативної альтернативи не призначається дескриптор поля.

*Джерело: побудовано автором*



Для опису властивостей заголовка шаблону воєнного повідомлення використовується тип даних `Tdata_type_head`. Він є структурою, поля якої наведені нижче в таблиці 4.14.

Таблиця 4.14 – Перелік полів типу даних `Tdata_type_head`

Тип даних	Назва поля	Значення
QString	<code>_f_tit</code>	Повна назва шаблону військового повідомлення
QString	<code>_short</code>	Скорочена назва шаблону військового повідомлення
QString	<code>_purp</code>	Ціль
bool	<code>_XXX</code>	
bool	<code>_XX</code>	
bool	<code>_X</code>	
QString	<code>_ii_spons</code>	Ініціатор повідомлення
QString	<code>_nato_uf</code>	

*Джерело: побудовано автором*

Шаблон повідомлення розміщується в екземплярі об'єкта типу `atom_constr_tmpl`, який в якості полів містить змінну `head_data` типу `Tdata_type_head` та покажчик `a_mes` на об'єкт типу `atommess`. Відповідно, в змінній `head_data` знаходиться заголовок шаблону, в об'єкті, на який вказує змінна `a_mes` знаходиться тіло шаблону військового повідомлення.

#### 4.4.2. Використані методи.

Під час створення програми використовувалися відомі методи, що у основі об'єктно-орієнтованого програмування [26, 27, 28] – успадкування і поліморфізм.

Для збереження шаблону повідомлення у вигляді вкладених один в одного блоків і полів були створені контейнери, що представляють вектори, що зберігають елементи типу `QObject` [29, 30].

Для гнучкості обробки екземплярів об'єктів було створено єдиний «батько» – базовий клас типу `MyObj`. Він є спільним предком для об'єктів, які описують блоки, та об'єктів, що описують поля шаблонів. Такий підхід

дозволяє в одному контейнері (об'єкт типу `vector`) зберігати, залежно від положення блоку в ієрархічній структурі шаблону, або вкладені блоки, або значення полів.

Під час створення графічного інтерфейсу програми використовувалися об'єкти, батьківським класом яких був `QWidget` [31].

Інтерфейсна частина програми повторює ієрархічну структуру шаблону, що зберігається в оперативній пам'яті.

Для відображення заголовка шаблону сконструйовано клас `constr_tit`, який містить елементи керування та компоненти вводу/виводу, що відображають інформацію заголовка повідомлення.

Для гнучкості управління та відображення елементів тіла шаблону був створений клас `tu_w`, що є спільним предком для класів, що відображають блоки шаблону та його поля.

Тіло шаблону конструюється за допомогою класу `constr_templ`, який має контейнер, що зберігає покажчики на об'єкти типу `tu_w`. Тому в контейнері можна розміщувати як вкладені блоки, так і вкладені поля, залежно від структури військового повідомлення. Клас `constr_templ` має як предка клас `tu_w`.

Клас `constr_templ` у своєму складі має елементи керування та компоненти, які дозволяють вводити/відображати інформацію про блок, створювати вкладені блоки в собі, створювати вкладені поля в собі (розміщувати їх у вбудованому контейнері).

Клас `tu_constr_w` призначений для відображення та введення інформації про зміст поля повідомлення. Його предком є клас `tu_w`. Примірник об'єкта класу `tu_constr_w` містить елементи керування, які дозволяють вводити/відображати значення для шаблону поля, видаляти це поле та вводити додаткову інформацію про поведінку поля в тексті військового повідомлення.

## 4.5 Реалізація модуля редактора повідомлень

Програма призначена для підготовки, прийому та перегляду військових повідомлень відповідно до стандартів НАТО. Вона забезпечує формування вихідних повідомлень, обробку вхідних повідомлень (ідентифікацію та парсинг), ідентифікацію стандарту вхідного повідомлення та перетворення військових повідомлень. Також програма забезпечує доступ до бази даних шаблонів військових повідомлень і до бази даних повідомлень.

Ім'я файлу - MessageComputer.

Підтримується багатомовний графічний інтерфейс користувача: англійська, українська, російська.

### 4.5.1. Модель даних

Для опису структури самого повідомлення застосовано аналогічний підхід, що і для опису структури шаблону (описаного в п.4.4), і ґрунтується на принципі вкладених контейнерів. Кожен блок (набір) повідомлення розглядається як клас, одним із полів якого є вектор, що містить об'єкти, вкладені в цей блок. Вектор може зберігати як блоки такого самого типу, що і він сам, так і об'єкти типу «поле».

Використання вкладених один в одного контейнерів дозволяє повністю повторити схему побудови військового повідомлення. Крім того, кожен блок повідомлення містить набір правил, які визначають його призначення та поведінку. Правила формуються з урахуванням даних, які містить шаблон військового повідомлення.

Для опису структури типу «блок» використовується клас `mess_block` з полями, наведеними в таблиці 4.15.

Таблиця 4.15 – Перелік полів класу `mess_block`

Тип даних	Назва поля	Значення
int	id_bl_db_mes	значення ключа в БД (table blocks)
Tassorty	obj_cont	вектор, що містить вкладений в блок об'єкти Об'єкти, які зберігають вектор типу Tassorty мають тип QObject
TBlockRules	rules_b	Набір правил для відображення блоку
bool	usable_b	Признак включення блоку в текст повідомлення
Ref_field	reley_b	Поле, від значення якого залежить заповнюваність блоку
int	list_el_idx	Для контейнера : індекс обранного елемента

*Джерело: побудовано автором*

Для опису абстракції типу «поле» повідомлення використовується клас `mess_field`, який представляє конструкцію з полями, описаними в таблиці 4.16.

Таблиця 4.16 – Перелік полів класу `mess_field`

Тип дани	Назва поля	Значення
int	int id_fl_db_mes	значення ключа в БД (table parametrs_value)
TFieldRules	rules_f	Набір правил для відображення блоку
QString	f_str	Зміст поля повідомлення
bool	f_usable	Признак включення поля в текст повідомлення
Ref_field	rf	Поле, значення якого уточнюється
int	pos_ref	Індекс деталізованого значення зі списку деталізації
Ref_field	reley_f	Поле, від значення якого залежить заповнюваність даного поля

*Джерело: побудовано автором*

Для опису властивостей заголовка шаблону військового повідомлення створено тип даних `THead_`. Він є структурою, поля якої описані в таблиці 4.17.

Таблиця 4.19 – Перелік полів типу даних THead\_

Тип даних	Назва поля	Значення
Int	id_db_head	Значення ключа у БД (table blocks БД templates)
QString	_ns	Назва станції або каналу
QString	_nk	Номер каналу
QString	_pr2	Пріоритет повідомлення
QString	_is	Індикатор станції маршрутизації
	_nsp	Назва станції підготовки
QString	_ksp	Код станції підготовки
QString	_dps	Дата подачі повідомлення для відправки
QString	_dtg_pr	Пріоритет повідомлення (DTG)
QString	_gs	Гриф секретності
QString	_receiver	Одержувач
QString	_inf_receiver	Інформація про одержувача
QString	_otpr	Відправник

*Джерело: побудовано автором*

Повідомлення розміщується в екземплярі об'єкта типу mess\_full, який як поля містить змінну head\_ типу THead\_ і покажчик body на об'єкт типу mess\_block. Відповідно, у змінній head\_ знаходиться заголовок повідомлення, в об'єкті, на який вказує змінна body, знаходиться тіло військового повідомлення.

#### 4.5.2. Використані методи

Для збереження повідомлення у вигляді вкладених один в одного блоків і полів були створені контейнери, що представляють вектори, що зберігають елементи типу QObject.

Для гнучкості обробки екземплярів об'єктів було створено єдиний «батько» - базовий клас типу MyObj. Він є спільним предком для об'єктів, які описують блоки, та об'єктів, що описують поля повідомлень. Такий підхід дозволяє в одному контейнері (об'єкт типу vector) зберігати, залежно від положення блоку в ієрархічній структурі повідомлення, або вкладені блоки, або значення полів.

Під час створення графічного інтерфейсу програми використовувалися об'єкти, батьківським класом яких був QWidget.

Інтерфейсна частина програми повторює ієрархічну структуру повідомлення, що зберігається в оперативній пам'яті.

Для відображення заголовка повідомлення сконструйовано клас `header_mes`, який містить елементи керування та компоненти вводу/виводу, що відображають інформацію заголовка повідомлення.

Для гнучкості управління та відображення елементів тіла повідомлення було створено клас `tu_w`, що є спільним предком для класів, що відображають блоки повідомлення та його поля.

Тіло повідомлення конструюється за допомогою класу `v_mess_block`, який має контейнер, що зберігає покажчики об'єкти типу `tu_w`. Тому в контейнері можна розміщувати як вкладені блоки, так і вкладені поля, залежно від структури військового повідомлення. Клас `v_mess_block` має як предка клас `tu_w`.

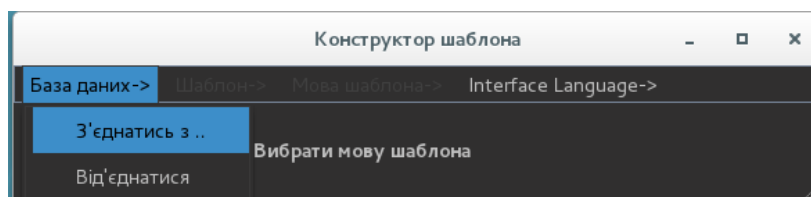
Клас `v_mess_block` у своєму складі має елементи керування та компоненти, які дозволяють відображати інформацію про блок, створювати копію блоку в залежності відправила формування повідомлення, видаляти поточний екземпляр блоку за командою оператора, приховувати/відображати свій вміст.

Клас `v_mess_field` призначений для відображення та введення інформації про зміст поля повідомлення. Його предком є клас `tu_w`. Примірник об'єкта класу `v_mess_field` містить елементи керування, які дозволяють вводити/відображати значення, створювати копію поля, видаляти це поле, приховувати/відображати вміст поля військового повідомлення

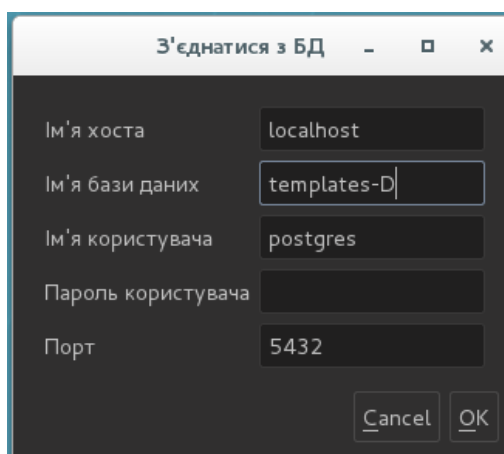
## 4.6 Ілюстрація впровадження інформаційної технології

### 4.6.1. Ілюстрація роботи модуля редактора шаблонів повідомлення.

Для початку роботи модуля редактора шаблонів військових повідомлень необхідно запустити файл TemplateConstructor і з'єднати програму з базою даних шаблонів повідомлення dbTemplates, попередньо обравши мову інтерфейсу та шаблону (рис. 4.3).



а) перехід до команди з'єднання з базою даних



б) форма з'єднання із базою даних шаблонів повідомлень

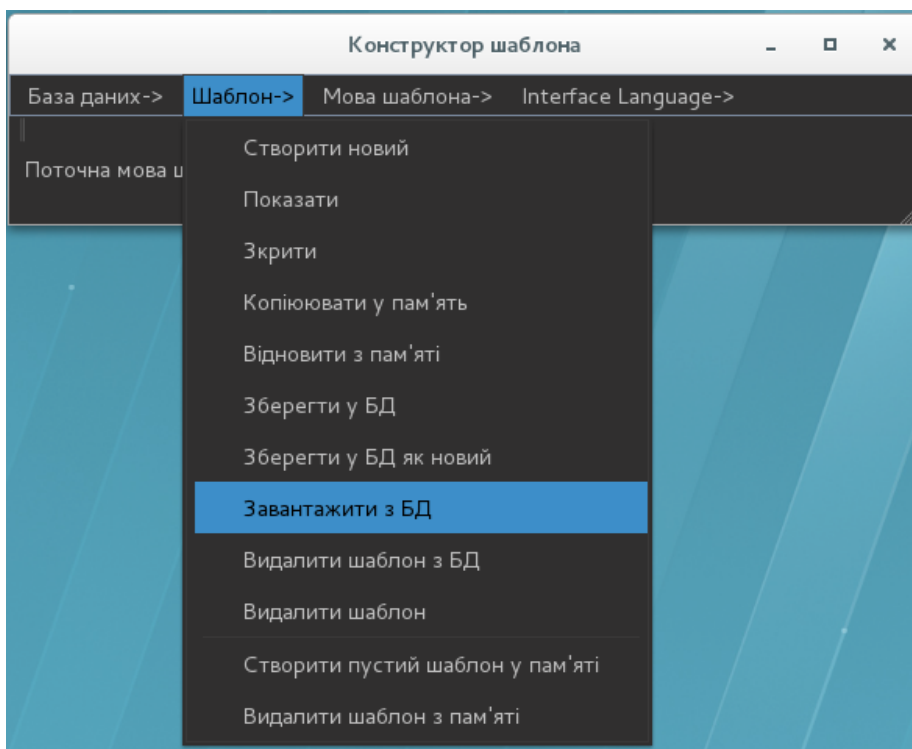
Рисунок 4.3 – Ілюстрація з'єднання з базою даних шаблонів повідомлень

*Джерело: розроблено автором*

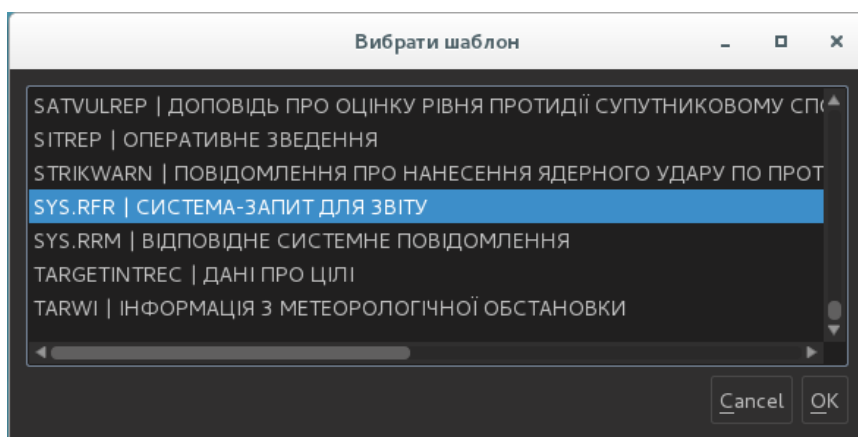
При створенні нового шаблону програма не потребує вхідних даних.

Під час редагування шаблону програма використовує дані, які зберігаються у базі даних шаблонів військових повідомлень.

Для відкриття перезавантаженого шаблону повідомлення необхідно скористатись командою «Завантажити з БД» (рис. 4.4). Відкритий шаблон повідомлення, доступний для відображення, зображений на рис. 4.5.



а) ілюстрація доступу до команди завантаження збереженого шаблону



б) форма вибору збереженого шаблону для відображення

Рисунок 4.4 – Ілюстрація процесу відкриття збереженого шаблону повідомлення з бази даних

*Джерело: розроблено автором*



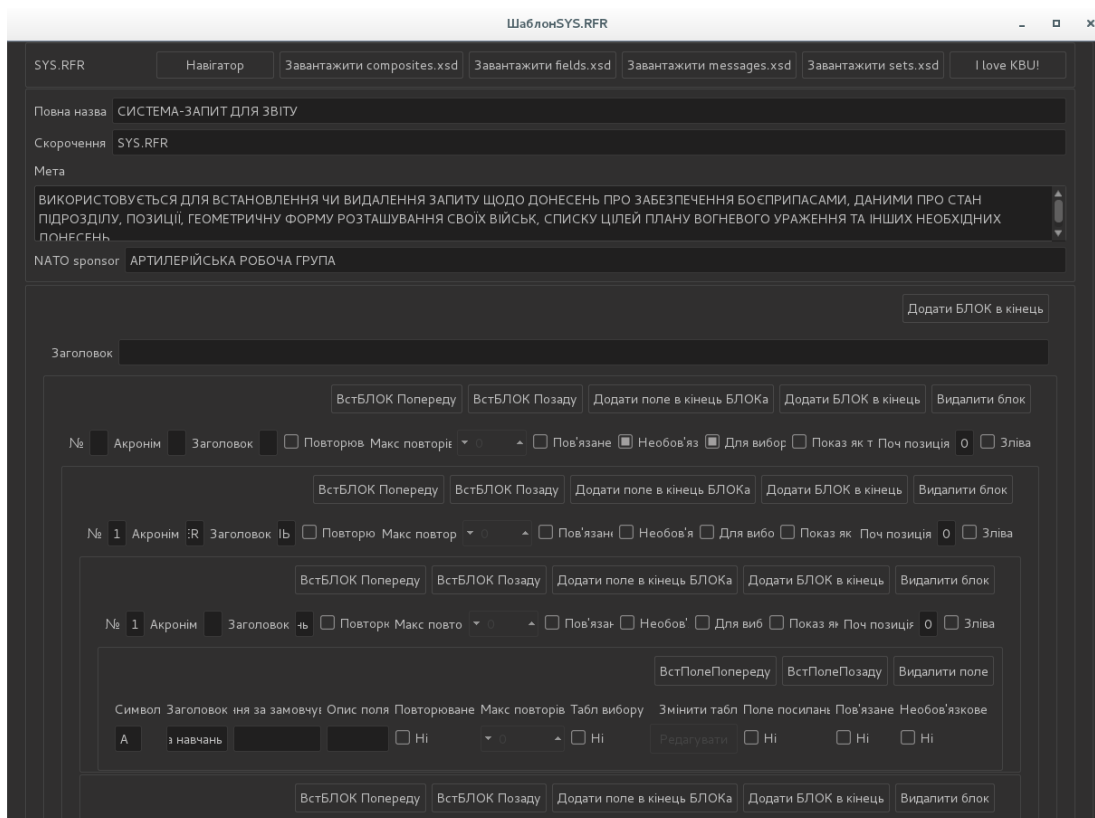


Рисунок 4.5 – Ілюстрація відображеного шаблону повідомлення

*Джерело: розроблено автором*

При записі шаблону військового повідомлення в базу даних програма зберігає дані, формат і значення яких описані в перших розділах пояснювальної записки.

#### 4.6.2. Ілюстрація роботи модуля редактора повідомлень.

Для початку виконання програми редактора військових повідомлень необхідно запустити файл MessageComputer, що здійсниться, і здійснити з'єднання програми з базами даних шаблонів повідомлень dbTemplates та повідомлень dbMessages (рис. 4.6).

Під час створення нового повідомлення програма потребує наявності вхідних даних у вигляді шаблону повідомлення, який має знаходитись у базі даних військових повідомлень.

Під час редагування військового повідомлення програма використовує дані, які зберігаються в базі даних шаблонів військових повідомлень та базі даних повідомлень.

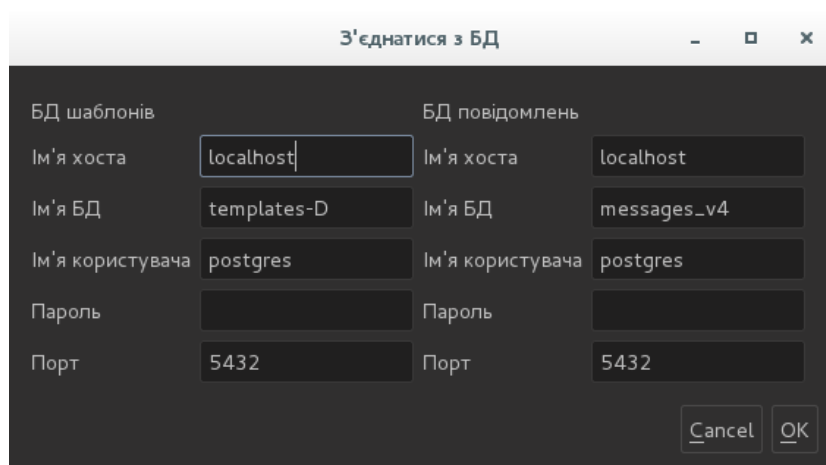


Рисунок 4.5 – Ілюстрація з'єднання з базою даних повідомлень та шаблонів повідомлень в модулі редактора повідомлень

*Джерело: розроблено автором*

Заповнення полів шаблону повідомлення регулюється правилами, закладеними на рівні таблиць бази даних. Заповнене повідомлення (рис. 4.6) можна зберегти в базі даних повідомлень для подальшого редагування з використанням відповідної команди.

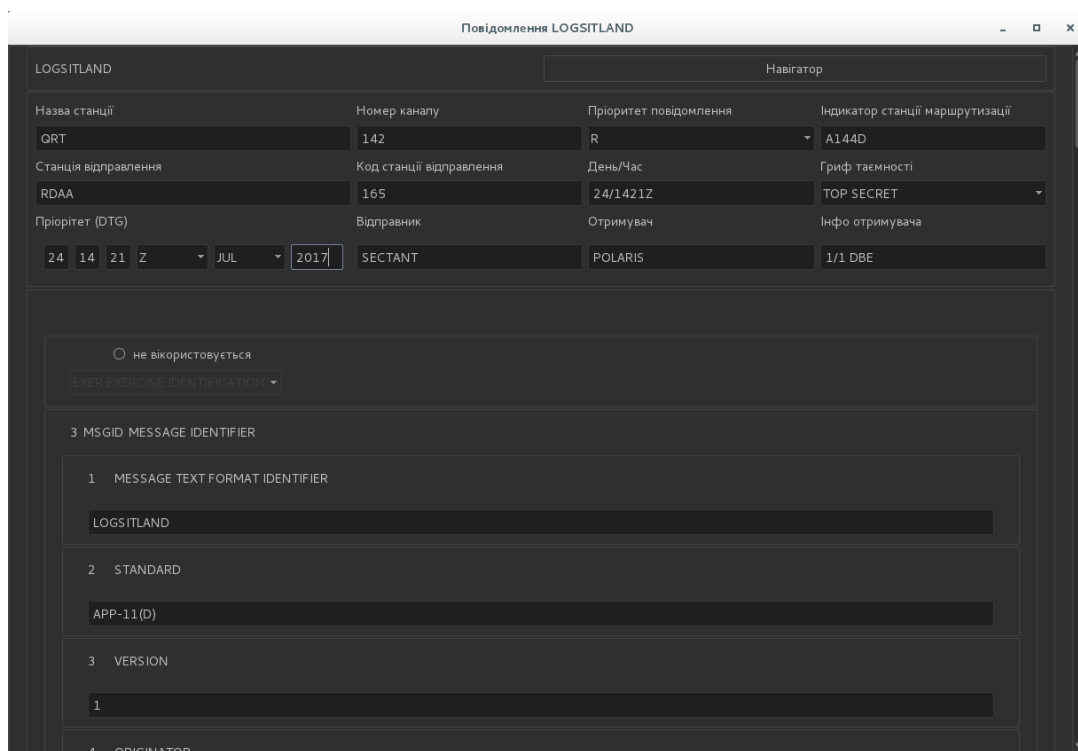
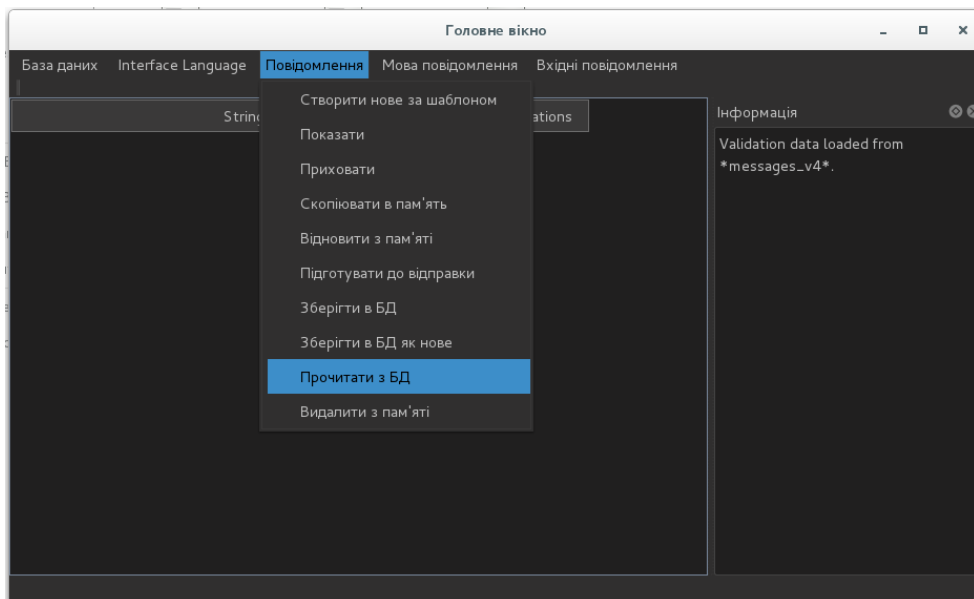


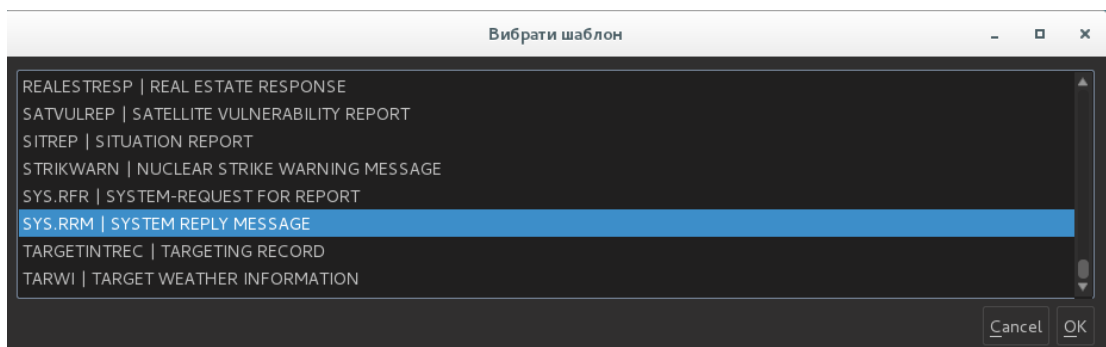
Рисунок 4.6 – Ілюстрація відображення сформованого повідомлення

*Джерело: розроблено автором*

Збережене повідомлення можна відкрити і, за потреби, редагувати значення полів або надіслати адресату з використанням команди «Прочитати з БД» меню «Повідомлення» редактору повідомлень (рис. 4.7).



а) ілюстрація доступу до команди завантаження збереженого повідомлення



б) форма вибору збереженого шаблону для відображення

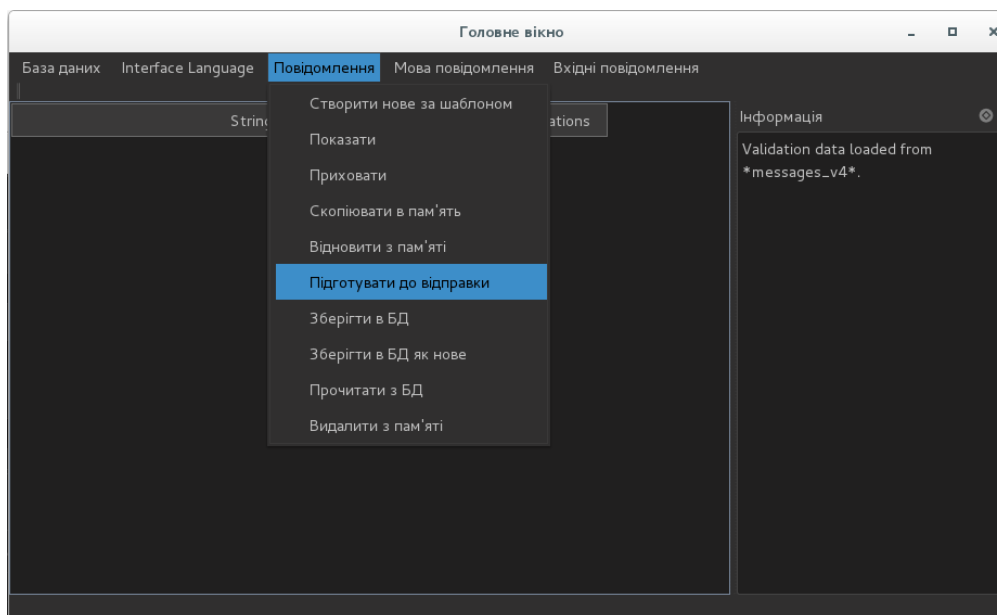
Рисунок 4.7 – Ілюстрація процесу відкриття збереженого повідомлення для редагування/відправлення

*Джерело: розроблено автором*

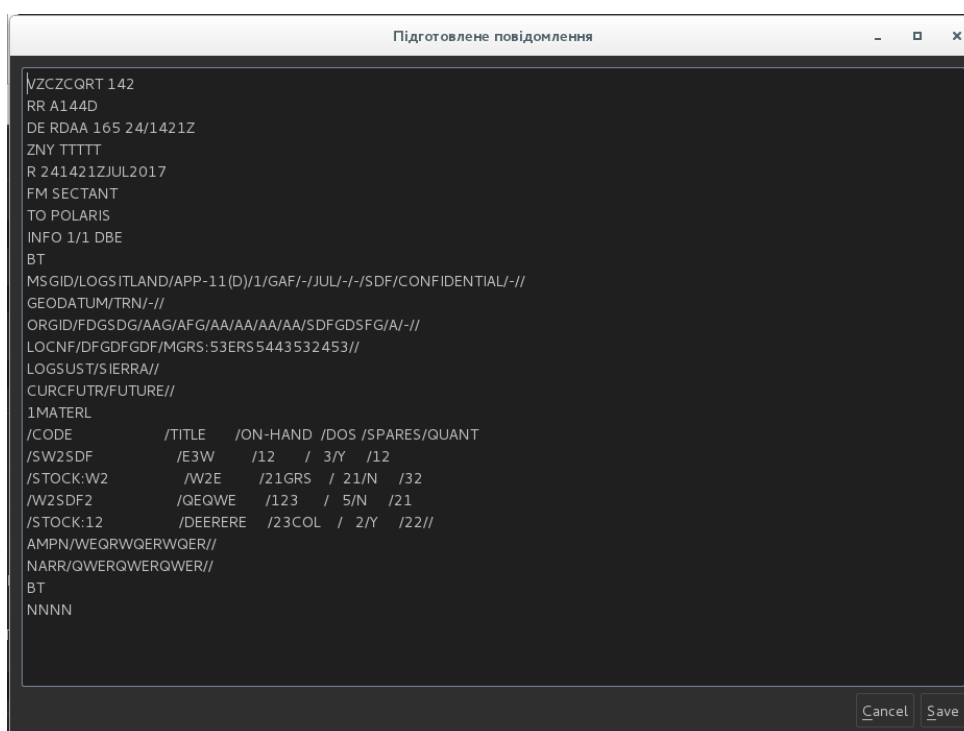
При записі військового повідомлення в БД повідомлень програма зберігає дані, формат і значення.

Після завершення формування військового повідомлення оператор може сформулювати текстове подання повідомлення, яке буде готове для відправки, як

проілюстровано на рис. 4.8. При виконання команди «Підготувати до відправки» потрібне повідомлення має бути відкрите в модулі редактора.



а) ілюстрація доступу до команди підготовки повідомлення до відправки



б) ілюстрація повідомлення, готового до відправки

Рисунок 4.8 – Ілюстрація процесу підготовки військового повідомлення до відправки

*Джерело: розроблено автором*

## ВИСНОВКИ

В результаті кваліфікаційної роботи магістра здобувачем була досягнута мета проекту, а саме, розроблена інформаційна технологія формування повідомлення на основі шаблонів формалізованих повідомлень стандарту НАТО. Інформаційна технологія базується на поєднанні технологій реляційних моделей даних та метамоделі EAV для реалізації блочно-модульної структури шаблону формалізованого повідомлення та логіки формування повідомлення на основі збережених шаблонів.

Проведений детальний аналіз специфіки обміну формалізованими повідомленнями при веденні бойових дій ракетних підрозділів Збройних Сил України та військових підрозділів альянсу НАТО. Сучасний етап становлення і розвитку Збройних Сил України обґрунтовує актуальну потребу розроблення технологій інтеграції процесів підготовки і ведення бойових дій ракетних підрозділів ЗСУ до системи, що впроваджена в військових підрозділах НАТО.

За результатами аналізу структури та алгоритму формування формалізованого повідомлення відповідно стандарту APP-11 визначено функціональні та нефункціональні вимоги до інформаційної технології. Результати проектування інформаційної технології представлені діаграмами бізнес-процесів контекстного рівня та першого рівня декомпозиції в нотації IDEF0 та діаграми варіантів використання в нотації UML. Система баз даних містить дві бази даних відповідно для шаблонів формалізованих повідомлень та самих сформованих повідомлень на основі шаблонів.

Для впровадження інформаційної технології використані такі інструменти та технології: модулі редакторів написані мовою програмування C++ з використанням середовища розробки QT 4.8, gcc та інтегрованого середовища розробки Qt Creator. Моделі баз даних реалізовані засобами СУБД PostgreSQL.

Впроваджена інформаційна технологія відповідає в повному обсязі заявленим функціональним та нефункціональним вимогам до неї та реалізує весь необхідний функціонал при формуванні військових формалізованих повідомлень на основі стандарту НАТО APP-11 NATO Message Catalogue Edition D Version 1 November 2015.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. APP-11 NATO Message Catalogue Edition D Version 1. November 2015. [Електронний ресурс]. Доступ з екрану: <https://nso.nato.int/nso/nsdd/main/standards/ap-details/1503/EN>
2. ADatP-3 NATO Message Text Formatting System (Formets) – Concept of Formets (Conformets). [Електронний ресурс]. Доступ з екрану: <https://standards.globalspec.com/std/14396946/ADATP-03>
3. ACP 127 Communications Instructions – Tape Relay Procedures. [Електронний ресурс]. Доступ з екрану: <https://archives.nato.int/acp-127-nato-suppl-1-communication-instructions-on-line-cryptographic-tape-relay-procedures>
4. STANAG 4406 Edition 2 – 2006 Military Message Handling System. [Електронний ресурс]. Доступ з екрану: [https://infostore.saiglobal.com/en-us/standards/stanag-4406-2006-736163\\_saig\\_nato\\_nato\\_1788065/](https://infostore.saiglobal.com/en-us/standards/stanag-4406-2006-736163_saig_nato_nato_1788065/)
5. Why use PostgreSQL [Електронний ресурс]. Доступ з екрану: <https://www.postgresql.org/about/>
6. PostgreSQL Tutorial [Електронний ресурс]. Доступ з екрану: <https://www.postgresql.org/docs/online-resources/>
7. Системи баз даних та знань [Текст] : підручник. Кн.1 : Організація баз даних та знань / А. Ю. Берко, О. М. Верес, В. В. Пасічник; за заг. ред. В. В. Пасічника. – Львів : Магнолія-2006, 2023. – 440 с
8. Системи баз даних та знань [Текст] : підручник. Кн.2 : Системи управління базами даних та знань / А. Ю. Берко, О. М. Верес, В. В. Пасічник; за заг. ред. В. В. Пасічника. – Львів : Магнолія-2006, 2023. – 584 с
9. Meier, Andreas, and Kaufmann, Michael. SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management// Springer Fachmedien Wiesbaden, 2019. – 229 p.

10. Kaufmann, Michael, and Meier, Andreas. SQL and NoSQL Databases: Modeling, Languages, Security and Architectures for Big Data Management// Springer Nature Switzerland, 2023. – 254 p.
11. Batra, Shivani & Sachdeva, Shelly & Bhalla, Subhash. Entity Attribute Value Style Modeling Approach for Archetype Based Data// Information 2018,9,2; doi:10.3390/info9010002
12. Saumyadipta Pyne, B.L.S. Prakasa Rao, S.B. Rao Big Data Analytics. Methods and Application//Springer India 2016, – 276 p. ISBN 978-81-322-3626-3
13. Advances in Model and Data Engineering in the Digitalization Era /MEDI 2021 International Workshops: DETECT, SIAS, CSMML, BIOC, HEDA, Tallinn, Estonia, June 21–23, 2021, Proceedings// Switzerland, Springer International Publishing, 2021. – 326 p.
14. Exploring Intelligent Decision Support Systems: Current State and New Trends// Germany, Springer International Publishing, 2018. – 238 p.
15. Pai, Praseed, and Abraham, Peter. C++ Reactive Programming: Design Concurrent and Asynchronous Applications Using the RxCpp Library and Modern C++17// Great Britain, Packt Publishing, 2018. – 348 p.
16. Rischpater, Ray. Application Development with Qt Creator: Design and Build Dazzling Cross-platform Applications Using Qt and Qt Quick//India, Packt Publishing, 2014. – 264 p.
17. Qt Creator Manual [Электронный ресурс]. Доступ з екрану: <https://doc.qt.io/qtcreator/>
18. Gregoire, Marc. Professional C++// USA, Wiley, 2021. – 1312p.
19. Lospinoso, Josh. C++ Crash Course: A Fast-Paced Introduction//USA, No Starch Press, 2019. – 792 p.
20. C++ Language Reference. [Электронный ресурс]ю Доступ з екрану: <https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170>
21. C++ Language [Электронный ресурс]. Доступ з екрану: <https://cplusplus.com/doc/tutorial/>



22. International Organization for Standardization. (2012, Sept. 15). ISO/IEC/IEEE 31320-1:2012. Information technology. Modeling Languages. Part 1: Syntax and Semantics for IDEF0. Geneva, 2012, 120 p.
23. Documents Associated With Unified Modeling Language™ (UML®) Version 2.5. [Електронний ресурс]. Доступ з екрану: <http://www.omg.org/spec/UML/2.5/>
24. Product Documentation for Red Hat Enterprise Linux 7 [Електронний ресурс]. Доступ з екрану: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7)
25. PostgreSQL 9.5.25 Documentation. libpq - C Library [Електронний ресурс]. Доступ з екрану: <https://www.postgresql.org/docs/9.5/libpq.html>
26. А. І. Коваль. Порівняння об'єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення/ А. І. Коваль, О. М. Яшина, Г. І. Радельчук, Ю. В. Форкун// Вісник Хмельницького національного університету. №3, 2021 (297). – 34-38 с.
27. Kealan Parr. The Four Pillars of Object-Oriented Programming [Електронний ресурс]. Доступ з екрану: <https://www.freecodecamp.org/news/four-pillars-of-object-oriented-programming/>
28. Алхімова С. М. Об'єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення /С. М. Алхімова. – Київ: КГП ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 192 с
29. QObject Class. QTDocumentation. [Електронний ресурс]. Доступ з екрану: <https://doc.qt.io/qt-6/qobject.html>.
30. Understanding QObject in Qt [Електронний ресурс]. Доступ з екрану: <https://bcairns.medium.com/understanding-qobject-in-qt-97de374ca0cd/>
31. Layout Management [Електронний ресурс]. Доступ з екрану: <https://doc.qt.io/qt-6/layout.html>

## ДОДАТОК А

**Перелік ФС відповідно „APP-11 NATO Message Catalogue Edition D  
Version 1 November 2015” для використання в ланках керування ВЗУ-  
рдн-сбатр при підготовці та веденні бойвих дій РП**

Таблиця А.1 – Перелік ФС

№ п/п	ФС відповідно APP-11 NATO Message Catalogue Edition D Version 1 November 2015		Скорочена назва ФС
	Найменування ФС	Найменування ФС українською	
1	AMMUNITION FIRE UNIT- AMMUNITION STATUS	Артилерійський вогневий підрозділ – забезпеченість боєприпасами	AFU.AMS
2	AMMUNITION FIRE UNIT- DEPLOYMENT COMMAND	Команда щодо розміщення позиційного району вогневого підрозділу	AFU.DCMD
3	AIR ATTACK WARNING	Повітряна тривога	AIRATTACKWARN
4	AIR RAID WARNING	Повітряна тривога	AIRRAIDWNG
5	BOMB, SHELL, ROCKET OR MORTAR REPORT	Звіт щодо бомбардування/донесення про ведення артилерійського та мінометного вогню/донесення про мінометний обстріл	BOMBREP
6	CASUALTY EVACUATION REQUEST	Запит на евакуацію поранених та хворих	CASEVACREQ
7	CBRN 1 BIOLOGICAL REPORT	РХБЗ 1 Доповідь про біологічне зараження	CBRN 1 BIO
8	CBRN 1 CHEMICAL REPORT	РХБЗ 1 Доповідь про хімічне зараження	CBRN 1 CHEM
9	CBRN 1 NUCLEAR REPORT	РХБЗ 1 Доповідь про радіаційне зараження	CBRN 1 NUC
10	CBRN 1 RADIOLOGICAL REPORT	РХБЗ 1 Доповідь про радіологічне зараження	CBRN 1 RAD
11	CBRN 2 BIOLOGICAL REPORT	РХБЗ 2 Доповідь про біологічне зараження	CBRN 2 BIO
12	CBRN 2 CHEMICAL REPORT	РХБЗ 2 Доповідь про хімічне зараження	CBRN 2 CHEM
13	CBRN 2 NUCLEAR REPORT	РХБЗ 2 Доповідь про радіаційне зараження	CBRN 2 NUC
14	CBRN 2 RADIOLOGICAL REPORT	РХБЗ 2 Доповідь про радіологічне зараження	CBRN 2 RAD
15	CBRN 3 BIOLOGICAL REPORT	РХБЗ 3 Доповідь про біологічне зараження	CBRN 3 BIO
16	CBRN 3 CHEMICAL REPORT	РХБЗ 3 Доповідь про хімічне зараження	CBRN 3 CHEM
17	CBRN 3 NUCLEAR REPORT	РХБЗ 3 Доповідь про радіаційне зараження	CBRN 3 NUC
18	CBRN 3 RADIOLOGICAL REPORT	РХБЗ 3 Доповідь про радіологічне зараження	CBRN 3 RAD
19	CBRN 4 BIOLOGICAL REPORT	РХБЗ 4 Доповідь про біологічне зараження	CBRN 4 BIO
20	CBRN 4 CHEMICAL REPORT	РХБЗ 4 Доповідь про хімічне зараження	CBRN 4 CHEM
21	CBRN 4 NUCLEAR REPORT	РХБЗ 4 Доповідь про радіаційне зараження	CBRN 4 NUC

## Продовження таблиці А1

№ п/п	ФС відповідно APP-11 NATO Message Catalogue Edition D Version 1 November 2015		Скорочена назва ФС
	Найменування ФС	Найменування ФС українською	
22	CBRN 4RADIOLOGICAL REPORT	РХБЗ 4 Доповідь про радіологічне зараження	CBRN 4RAD
23	CBRN 5BIOLOGICAL REPORT	РХБЗ 5 Доповідь про біологічне зараження	CBRN 5BIO
24	CBRN 5CHEMICAL REPORT	РХБЗ 5 Доповідь про хімічне зараження	CBRN 5CHEM
25	CBRN 5NUCLEAR REPORT	РХБЗ 5 Доповідь про радіаційне зараження	CBRN 5NUC
26	CBRN 5RADIOLOGICAL REPORT	РХБЗ 5 Доповідь про радіологічне зараження	CBRN 5RAD
27	CBRN 6BIOLOGICAL REPORT	РХБЗ 6 Доповідь про біологічне зараження	CBRN 6BIO
28	CBRN 6CHEMICAL REPORT	РХБЗ 6 Доповідь про хімічне зараження	CBRN 6CHEM
29	CBRN 6NUCLEAR REPORT	РХБЗ 6 Доповідь про радіаційне зараження	CBRN 6NUC
30	CBRN 6RADIOLOGICAL REPORT	РХБЗ 6 Доповідь про радіологічне зараження	CBRN 6RAD
31	CBRN BASIC WIND REPORT	РХБЗ Доповідь про основний напрямок вітру	CBRN BWR
32	CBRN CHEMICAL DOWNWIND REPORT	РХБЗ Доповідь про хімічне зараження за напрямком вітру	CBRN CDR
33	CBRN EFFECTIVE DOWNWIND REPORT	РХБЗ Доповідь про напрям та силу дійсного вітру	CBRN EDR
34	CBRN HAZARDOUS WARNING REPORT	РХБЗ Попередження про небезпеку	CBRN HAZWARN
35	CBRN SITUATION REPORT	РХБЗ Доповідь про ситуацію	CBRN SITREP
36	CCIS STATUS REPORT	Доповідь про готовність командно-інформаційної системи	CCISSTATREP
37	COMMUNICATIONS SPOT REPORT	Доповідь про стан засобів зв'язку	COMSPOT
38	DEPLOYMENT END REPORT	Доповідь про завершення розгортання	DEPENDREP
39	DEPLOYMENT REPORT	Доповідь про розгортання	DEPLOYREP
40	DEPLOYMENT START REPORT	Доповідь про початок розгортання	DEPSTARTREP
41	ENGINEER RECCE ORDER	Наказ на інженерну розвідку	ENGRECCEORD
42	ENGINEER RECCE REPORT	Доповідь про інженерну розвідку	ENGRECCREP
43	EVACUATION REQUEST	Запит на евакуацію	EVACREQ
44	FIRE MISSION - CALL FOR FIRE	Вогневе завдання – виклик вогню	FM.CFF

## Продовження таблиці А1

№ п/п	ФС відповідно APP-11 NATO Message Catalogue Edition D Version 1 November 2015		Скорочена назва ФС
	Найменування ФС	Найменування ФС українською	
45	FIRE MISSION - FIRE MISSION COMMAND	Вогневе завдання – команда на виконання вогневого завдання	FM.FMC
46	FIRE MISSION - SUBSEQUENT ADJUSTMENT	Вогневе завдання – подальше коригування	FM.SUB
47	FRAGMENTARY ORDER	Окреме бойове розпорядження	FRAGO
48	INCIDENT REPORT	Доповідь про інцидент	INCREP
49	INCIDENT SPOT REPORT	Термінова доповідь про інцидент	INCSPOTREP
50	INTELLIGENCE REPORT	Доповідь про розвідку	INTREP
51	LOGISTIC ASSISTANCE REQUEST	Запит на матеріально-технічне забезпечення	LOGASREQ
52	LOGISTIC ASSISTANCE RESPONSE	Відповідь на запит на матеріально-технічне забезпечення	LOGASRESP
53	LOGISTIC SITUATION REPORT LAND FORCES	Доповідь щодо матеріально-технічного забезпечення в підрозділах Сухопутних військ	LOGSITLAND
54	LOGISTIC UPDATE REPORT	Уточнена доповідь по матеріально-технічному забезпеченню	LOGUPDATE
55	MAP REQUEST	Запит на карти	MAPREQ
56	MOVEMENT COMPLETION REPORT	Доповідь про завершення переміщення	MCR
57	METEOROLOGICAL-COMPUTER	Метеорологічні дані (комп'ютерний варіант)	MET.CM
58	METEOROLOGICAL-REQUEST FOR MET	Метеорологічний запит	MET.RFM
59	MEACONING, INTRUSION, JAMMING, INTERFERENCE WARNING REPORT	Оповіщення про роботу засобів радіоелектронної боротьби	MIJWARNREP
60	MESSAGE CORRECTION OR CANCELLATION	Повідомлення про відміну попереднього повідомлення	MSGCORRCANX
61	NON-NUCLEAR FIRE PLANNING-COMPUTE FIRE PLAN	План-розрахунок неядерного порядку вогневого ураження	NNFP.COMFP
62	NON-NUCLEAR FIRE PLANNING - FIRE PLAN	Планування вогню неядерними боєприпасами – план вогню	NNFP.FP
63	ORDER OF BATTLE – LAND	Бойовий склад і дислокація частин та підрозділів – сухопутні війська	ORBATLAND
64	OWN LAND FORCES SITUATION REPORT	Доповідь про стан підрозділу	OWNSITREP

## Продовження таблиці А1

№ п/п	ФС відповідно APP-11 NATO Message Catalogue Edition D Version 1 November 2015		Скорочена назва ФС
	Найменування ФС	Найменування ФС українською	
5	PERSONNEL REPORT	Доповідь про особовий склад	PERSREP
66	PRESENCE REPORT	Доповідь про дислокацію	PRESENCE
67	REAL ESTATE REQUEST	Запит на розташування	REALESTREQ
68	REAL ESTATE RESPONSE	Відповідь на запит на розташування	REALESTRESP
69	SATELLITE VULNERABILITY REPORT	Доповідь про оцінку рівня протидії супутниковому спостереженню	SATVULREP
70	SITUATION REPORT	Оперативне зведення	SITREP
71	NUCLEAR STRIKE WARNING MESSAGE	Повідомлення про нанесення ядерного удару по противнику	STRIKWARN
72	SYSTEM-REQUEST FOR REPORT	Система-запит для звіту	SYS.RFR
73	SYSTEM REPLY MESSAGE	Відповідне системне повідомлення	SYS.RRM
74	TARGETING RECORD	Дані про цілі	TARGETINTREC
75	TARGET WEATHER INFORMATION	Інформація з метеорологічної обстановки	TARWI

*Джерело: розроблено автором*

## ДОДАТОК Б

### ШАБЛОНИ ПОВІДОМЛЕНЬ

Таблиця Б.1 – Шаблон заголовка текста повідомлення

Вибір з альтернатив	Обов'язковість блоку	Обов'язковість значень в блоці	Повтор блока (поля)	ІН Поля	№ блокц	Назва поля		Розмірність
						Англійська	Українська	
Заголовок повідомлення								
1#				EXER	1	EXERCISE IDENTIFICATION	НАЗВА НАВЧАНЬ	
		+				Exercise nickname	Назва навчань	(56 Char) A-Z, 0-9, SPC
						Exercise identifier (1)	Ідентифікація навчань	(16 Char)
1#				OPER	2	OPERATION CODEWORD	НАЗВА ОПЕРАЦІЇ	
		+				Operation codeword	Назва операції	(32 Char) A-Z, 0-9, SPC
						Plan originator and number	План операції та номер	(36 Char) A-Z, 0-9, SPC
						Option nickname	Основні опції	(23 Char) A-Z, 0-9, SPC
						Secondary option nickname	Додаткові опції	(23 Char) A-Z, 0-9, SPC
	+			MSGID	3	MESSAGE IDENTIFIER	ІДЕНТИФІКАЦІЯ ПОВІДОМЛЕННЯ	
		+				Message text format identifier	Ідентифікатор шаблону повідомлення	(19 Char) A-Z, 0-9, SPC
		+				Standard	Стандарт	(20 Char) A-Z, 0-9, SPC
		+				Version	Версія	(20 Char) A-Z, 0-9, SPC
		+				Originator	Відправник	(20 Char) A-Z, 0-9, SPC
						Message serial number	Серійний номер повідомлення	(20 Char) A-Z, 0-9, SPC
		+				Reference time of publication (2)	Дата формування повідомлення	(16 Char)
						Qualifier (3)	Обмеження	(3 Char)

## Продовження таблиці Б.1

Вибір з альтернатив Обов'язковість блоку	Обов'язковість значень в блоці	Повтор блока (поля)	ІН Поля	№ блоку	Назва поля		Розмірність
					Англійська	Українська	
					Serial number of qualifier	Серійний номер обмеження	(3 Figs) 1-999
	+				Message security policy	Політика безпеки повідомлення	(50 Char) A-Z, 0-9, SPC
	+				Message classification (4)	Класифікація повідомлення	(50 Char)
					Message security category	Категорія безпеки повідомлень	(50 Char) A-Z, 0-9, SPC
		Повтор за необхідності	REF	4	REFERENCE	ДОВІДКА	
	+				Serial letter	Серійний номер	(1 Char) A-Z
	+				Communications type (5)	Тип зв'язку або ІД формат текстового повідомлення	(30 Char)
					Title of document	Назва документа	(54 Char) A-Z, 0-9, SPC
	+				Originator	Відправник	(30 Char) A-Z, 0-9, SPC
	+				Date and or time of reference (6)	Дата та (або) час	(16 Char)
					Reference serial number	Серійний номер	(30 Char) A-Z, 0-9, SPC
					Special notation (7)	Спеціальні примітки	(5 Chars)
		Повтор за необхідності			SIC or file number (8)	Номер SIC або номер файлу	(10 Char) A-Z, 0-9, SPC

*Джерело: розроблено автором*

Таблиця Б.2 – Кодограма заголовка повідомлення

№ поля	Кодограма
1	EXER/Text1/Text2 <sup>1</sup> //
2	OPER/ Text1/Text2/Text3/Text4//
<b>Використовується блок EXER або блок OPER</b>	
3	MSGID/Text1/Text2/Text3/Text4/Text5/Text6 <sup>2</sup> /Text7 <sup>3</sup> /Text8/Text9/Text10 <sup>4</sup> /Text11//
4	REF/Text1/Text2 <sup>5</sup> /Text3/Text4/Text5 <sup>6</sup> /Text6/Text7 <sup>7</sup> /Text8 <sup>8</sup> //  <b>Блок REF может повторюватись. Поле Text8 в блоці REF може повторюватись.</b>
<b>Примітка:</b>	
1	– Вибір з:
a)	Дискриптор з Додатка № 4      таблиця 40      + Додаток № 4      таблиця 113;
b)	Дискриптор з Додатка № 4      таблиця 40      + 16      символів (A-Z, 0-9, SPC);
2	– Вибір з:
a)	Дискриптор з Додатка № 4      таблиця 41      + Додаток № 3      таблиця 115;
b)	Дискриптор з Додатка № 4      таблиця 41      + Додаток № 5      таблиця 4;
3	– Додаток № 3      таблиця 117;
4	– Вибір з:
a)	Дискриптор з Додатка № 4      таблиця 42      + Додаток № 3      таблиця 118;
b)	Дискриптор з Додатка № 4      таблиця 42      + 50      символів (A-Z, 0-9, SPC);
5	– Вибір з:
a)	Дискриптор з Додатка № 4      таблиця 1      + 30      символів (A-Z, 0-9, SPC);
b)	Дискриптор з Додатка № 4      таблиця 1      + Додаток № 3      таблиця 1;
6	– Вибір з:
a)	Дискриптор з Додатка № 4      таблиця 1      + Додаток № 5      таблиця 1;
b)	Дискриптор з Додатка № 4      таблиця 1      + Додаток № 5      таблиця 2;
c)	Дискриптор з Додатка № 4      таблиця 1      + Додаток № 5      таблиця 3;
d)	Дискриптор з Додатка № 4      таблиця 1      + Додаток № 5      таблиця 4;
7	– Додаток № 3      таблиця 2;
8	– Вибір з:
a)	Дискриптор з Додатка № 4      таблиця 2      + 3      символів (A-Z, 0-9);
b)	Дискриптор з Додатка № 4      таблиця 2      + 10      символів (A-Z, 0-9, SPC).

*Джерело: розроблено автором*



Таблиця Б.3 – Шаблон додаткових блоків

Вибір з альтернатив	Обов'язковість блоку	Обов'язковість значень в блоці	Повтор блока (поля)	ІН Поля	№ блоку	Назва поля		Розмірність
						Англійська	Українська	
				AMPN		AMPLIFICATION	ПІДСИЛЕННЯ	
						FREE TEXT	Вільний текст	(Необмежений) A-Z, a-z,0-9, SPC
				NARR		NARRATIVE	УТОЧНЕННЯ	
						FREE TEXT	Вільний текст	(Необмежений) A-Z, a-z,0-9, SPC
				RMKS		REMARKS	ПРИМІТКИ	
						FREE TEXT	Вільний текст	(Необмежений) A-Z, a-z,0-9, SPC

*Джерело: розроблено автором*

Таблиця Б.4 – Кодограма додаткових блоків

№ поля	Кодограма
	AMPN/FREE TEXT//
	NARR/FREE TEXT//
	RMKS/ FREE TEXT//

*Джерело: розроблено автором*

## ДОДАТОК В

### Приклад шаблону формалізованого повідомлення на основі APP-11

Таблиця В.1 - AFU.AMS

Message identifier (Name):	AFU.AMS (AMMUNITION FIRE UNIT-AMMUNITION STATUS)	АРТИЛЕРІЙСЬКИЙ ВОГНЕВИЙ ПІДРОЗДІЛ – ЗАБЕЗПЕЧЕНІСТЬ БОЄПРИПАСАМИ
Related Documents:	ASCA-012 02	СТІДР STANAG 2432
Purpose:	THE AFU.AMS IS USED TO ESTABLISH OR DELETE AN AMMUNITION STATUS FOR A FIRE UNIT IN CURRENT OR PLANNED OPERATIONS.	ПРИЗНАЧЕНА ДЛЯ ВИЗНАЧЕННЯ СТАНУ ЗАБЕЗПЕЧЕНOSTІ БОЄПРИПАСАМИ В ПОТОЧНИХ ЧИ ЗАПЛАНОВАНИХ ОПЕРАЦІЯХ.
Sponsor:	NSA (ARMY) ARTILLERY WORKING GROUP (ARTYWG)	АРТИЛЕРІЙСЬКА РОБОЧА ГРУПА
Notes:	none	
Status:	Published	

*Джерело: розроблено автором*

Вибір з альтернатив	Обов'язковість блоку	Обов'язковість значень і полі	Повтор блока (поля)	ІН Поля	№ блока	Назва поля		Тип даних та розмірність
						Англійська	Українська	
Текст повідомлення								
	+			OPT	5	OPTION	ОПЦІЇ	
		+				Primary option (1)	Основні опції	(3 Char)
						Secondary option (2)	Другорядні опції	(3 Char)
FIELD 2 IN SET 5 (OPT) IS PROHIBITED IF FIELD 1 IN SET 3 (MSGID) EQUALS "AFU.".						ЗНАЧЕННЯ 2 В БЛОЦІ 5 (OPT) НЕ ВИКОРИСТОВУЄТЬСЯ ЯКЩО Поле 1БЛОК 3 (MSGID) МАЄ ЗНАЧЕННЯ "AFU.".		
	+			UNT	6	FIRE UNIT	ВОГНЕВИЙ ПІДРОЗДІЛ	
						Section	Відділення	(1 Char) A-Z, 0-9
						Platoon/Troop	Взвод/Загін	(1 Char) A-Z, 0-9
		+				Battery/Company	Батарея/Рота	(1 Char) A-Z, 0-9
		+				Battalion/Regiment	Батальйон/Полк	(3 Char) A-Z, 0-9
		+				Regiment/Brigade/Division	Полк/Бригада/Дивізія	(3 Char) A-Z, 0-9
	+			DTG	7	DATE-TIME GROUP (3)	ДАТА	(14 Chars)
				FPN	8	FIRE PLAN NAME	НАЙМЕНУВАННЯ ПЛАНУ	(6 Chars) A-Z, 0-9

Вибір з альтернатив	Обов'язковість блоку	Обов'язковість значень і полі	Повтор блоку (поля)	ІН Поля	№ блока	Назва поля		Тип даних та розмірність
						Англійська	Українська	
							ВОГНЕВОГО УРАЖЕННЯ	
			Повторяється не більше 7 раз	PRJ	9	PROJECTILES	ЗАБЕЗПЕЧЕНІСТЬ СНАРЯДАМИ (РАКЕТАМИ)	
		+				Projectile Type (4)	Тип снаряду (ракети)	(3 Chars)
		+				Quantity	Кількість	(4 Figs) 1-9999
SET 9 (PRJ) IS REQUIRED IF SET 10 (PRP) DOES NOT OCCUR AND (SET 11 (FZE) DOES NOT OCCUR AND FIELD 1 IN SET 5 (OPT) EQUALS "XMT").						БЛОК 9 (PRJ) Є ОБОВ'ЯЗКОВИМ ЯКЩО НЕ ВИКОРИСТОВУЄТЬСЯ БЛОК 10, 11 (PRP), (FZE) ТА Поле 1 БЛОКУ 5 (OPT) МАЄ ЗНАЧЕННЯ "XMT".		
			Повторяється не більше 7 раз	PRP	10	PROPELLANTS	БОЄЗАРЯД	
		+				Propellant Type (5)	Тип боєзаряду	(5 Chars)
		+				Quantity	Кількість	(4 Figs) 1-9999
SET 10 (PRP) IS REQUIRED IF SET 9 (PRJ) DOES NOT OCCUR AND (SET 11 (FZE) DOES NOT OCCUR AND FIELD 1 IN SET 5 (OPT) EQUALS "XMT").						БЛОК 10 (PRP) Є ОБОВ'ЯЗКОВИМ ЯКЩО НЕ ВИКОРИСТОВУЄТЬСЯ БЛОК 9, 11 (PRJ), (FZE) ТА Поле 1 БЛОКУ 5 (OPT) МАЄ ЗНАЧЕННЯ "XMT".		

Вибір з альтернатив	Обов'язковість блоку	Обов'язковість значень і полі	Повтор блока (поля)	ІН Поля	№ блока	Назва поля		Тип даних та розмірність
						Англійська	Українська	
SET 10 (PRP) IS PROHIBITED IF SIGNIFICANT FIELD 1 IN EVERY OCCURRENCE OF FIELD GROUP IN SET 9 (PRJ) EQUALS «JED, JEE, JEH, JEL, JEM, JMT, JTA, JTB, JTC, BUS, BLA».						БЛОК 10 (PRP) НЕ ВИКОРИСТОВУЄТЬСЯ ЯКЩО В БЛОКУ 9 (PRJ) ВИКОРИСТОВУЄТЬСЯ ТИП «JED, JEE, JEH, JEL, JEM, JMT, JTA, JTB, JTC, BUS, BLA».		
			Повторяється не більше 7 раз	FZE	11	FUZES	ПІДРИВНИК	
		+				Fuze Type (6)	Тип підривника	(5 Chars)
		+				Quantity	Кількість	(4 Figs) 1-9999
SET 11 (FZE) IS REQUIRED IF SET 9 (PRJ) DOES NOT OCCUR AND (SET 10 (PRP) DOES NOT OCCUR AND FIELD 1 IN SET 5 (OPT) EQUALS "XMT").						БЛОК 11 (FZE) Є ОБОВ'ЯЗКОВИМ ЯКЩО НЕ ВИКОРИСТОВУЄТЬСЯ БЛОК 9, 10 (PRJ), (PRP) ТА Поле 1 БЛОКУ 5 (OPT) МАЄ ЗНАЧЕННЯ "XMT".		
SET 11 (FZE) IS PROHIBITED IF SIGNIFICANT FIELD 1 IN EVERY OCCURRENCE OF FIELD GROUP IN SET 9 (PRJ) EQUALS «JED, JEE, JEH, JEL, JEM, JMT, JTA, JTB, JTC, BUS, BLA».						БЛОК 11 НЕ ВИКОРИСТОВУЄТЬСЯ ЯКЩО В БЛОКУ 9 ВИКОРИСТОВУЄТЬСЯ ТИП «JED, JEE, JEH, JEL, JEM, JMT, JTA, JTB, JTC, BUS, BLA».		

Таблиця В.2 - Кодограма тексту повідомлення

№ поля	Кодограма
5	OPT/Text1 <sup>1</sup> /Text2 <sup>2</sup> //
6	UNT/Text1/Text2/Text3/Text4/Text5//
7	DTG/TextDTG <sup>3</sup> //
8	FPN/Text//
9	PRJ/Text1 <sup>4</sup> /Text2//  Блок може повторюватись.
10	PRP/Text1 <sup>5</sup> /Text2//  Блок може повторюватись
11	FZE/Text1 <sup>6</sup> /Text2//  Блок може повторюватись
Примітка:	
1	– Додаток № 3      таблиця 3;
2	– Додаток № 3      таблиця 4;
3	– Додаток № 5      таблиця 1;
4	– Додаток № 3      таблиця 5;
5	– Додаток № 3      таблиця 6;
6	– Додаток № 3      таблиця 7.

## ДОДАТОК Г

### ПЛАНУВАННЯ РОБІТ

**Деталізація мети проекту методом SMART.** Кінцевим продуктом дослідження є програмний продукт, що реалізує інформаційну технологію формування формалізованих повідомлень за шаблонами стандарту НАТО APP-11. Результати деталізації методом SMART розміщені у табл. Г.1.

Таблиця Г.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити інформаційну технологію формування формалізованого повідомлення за шаблонами формалізованих повідомлень, структура яких визначені стандартом НАТО APP-11, та розробити програмний додаток для демонстрації роботи представленої технології.
Measurable (вимірювана)	Результатом роботи є розроблений програмний додаток, що дозволяє створювати формалізоване повідомлення та переглядати отримане формалізоване повідомлення.
Achievable (досяжна)	Для виконання проекту наявні необхідні знання технологій реляційних баз даних, метамоделювання (EAV), мови SQL, C++ , практичні навички роботи в середовищі QT Creator, навичок написання документації. Враховуючи доступні ресурсні можливості та обмеження мета є такою, яку можливо досягти.

Продовження табл. Г.1

Relevant (реалістична)	Розроблена технологія дозволить автоматизувати процес формування структури та вмісту формалізованого повідомлення щодо дій ракетних підрозділів ЗСУ, валідувати та зберегти отримане повідомлення в базі даних та позитивно вплине на оперативність діяльності військових підрозділів ракетних військ ЗСУ та ефективність інтеграції з військовими підрозділами НАТО при виконанні бойових задач.
Time-framed (обмежена у часі)	Розробити інформаційну технологію формування повідомлення за шаблонами стандарту НАТО на основі сформованого календарного плану проекту

*Джерело: розроблено автором*

**Планування змісту структури робіт IT-проекту (WBS).** Структура декомпозиції робіт (WBS) у проектному менеджменті є орієнтованою на доконане виконання проекту декомпозицією проекту на менші частки. Структура декомпозиції робіт є ключовою часткою робіт по проекту, яка організовує командну роботу по проекту у керовані частини.

WBS є ієрархічною декомпозицією проекту у фази, кінцеві результати та пакети робіт. Вона є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль; наприклад, програма, проект чи договір. У проекті чи договорі, розробка WBS відбувається, починаючи з кінцевих цілей та успішного розподілу її у керовані частини, що можуть бути оцінені за критеріями розміру, тривалості та відповідальностей (наприклад, системи, підсистеми, компоненти, задачі, підзадачі та пакети робіт) та включають усі необхідні для досягнення мети проекту кроки.

Система декомпозиції робіт надає загальний каркас для природнього розвитку загального планування та контролю договору і є базисом для



розподілу роботи на частини, що можуть бути визначеними, та з яких може бути зроблене.

WBS організується навколо ключових запланованих результатів. Так як заплановані результати є бажаним завершенням проекту, вони формують відносно стабільний набір категорій. WBS діаграма проекту зображена на рисунку Г.1.



Рисунок Г.1 – WBS структура проекту

*Джерело: розроблено автором*

**Організаційна структура проекту (OBS).** Організаційна структура проекту (OBS) є інструментом управління проектами, який показує ієрархічну структуру організації, в якій реалізується проект. OBS поділяє її на окремі рівні та групує відповідні функціональні області зі схожими цілями та завданнями. OBS структура проекту наведена на рисунку Г.2. Використання OBS забезпечує ефективну комунікацію та співпрацю та сприяє успішному виконанню проекту та може допомогти зменшити витрати проекту.

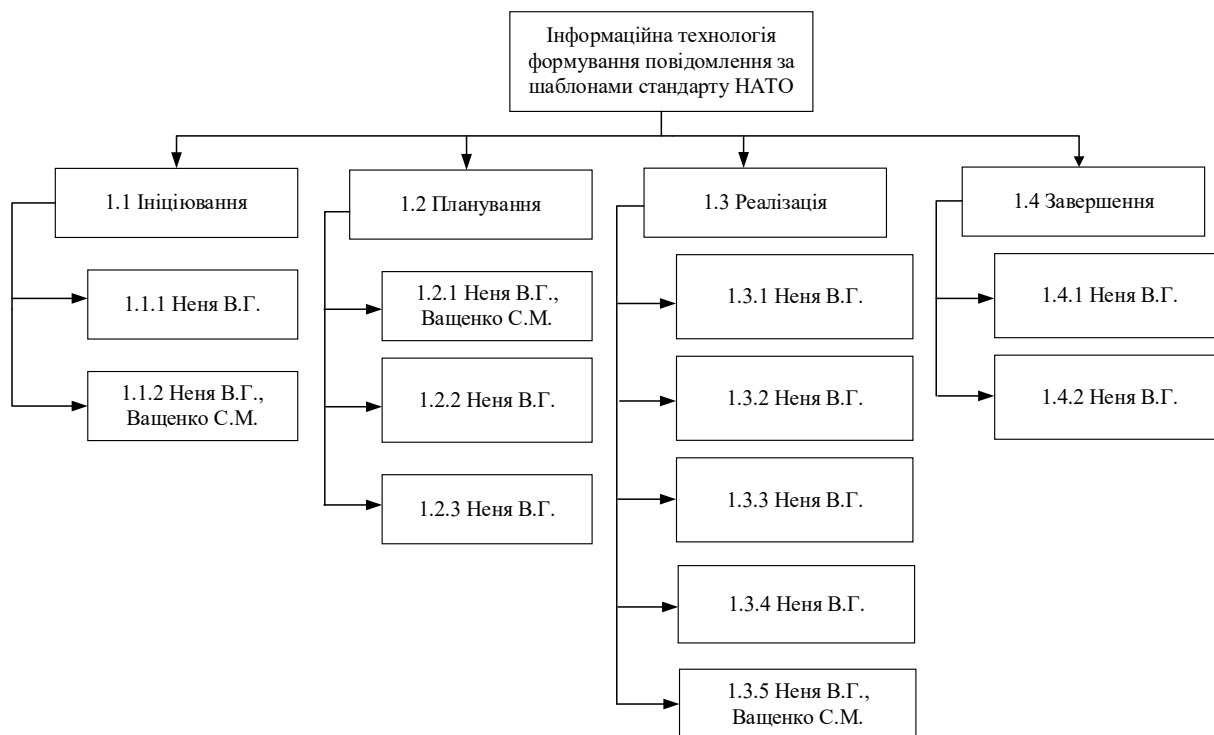


Рисунок Г.2. – OBS структура проекту

*Джерело: розроблено автором*

**Календарний графік виконання ІТ-проекту.** Календарний графік робіт призначений для відображення тривалості виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткових мережних моделей, а також, проекту в цілому з урахуванням вихідних і святкових днів.

Календарний графік робіт є реальним розподілом проектних робіт за календарними датами, тобто своєрідним розкладом виконання робіт. Діаграма Ганта представляє собою відрізки, які розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, як складова плану, розміщуються ліворуч у вертикальному напрямку. Початок, кінець і довжина відрізків на шкалі часу відповідають початку, кінця і тривалості завдання.

Діаграма Ганта, побудована для проекту дослідження, наведена на рисунках Г.3 та Г4.

## Інформаційна технологія формування повідомлення за шаблонами стандарту НАТО

Сумський державний університет, Кафедра інформаційних технологій

Неня В.Г.

Початок проекту: 10.09.2023

Scrolling increment: 0

Опис етапу проекту	Виконавець (ці)	Прогрес	Початок	Тривалість в Днях
<b>Ініціювання</b>				
Аналіз предметної області	Неня В.Г.	100%	10.09.2023	10
Постановка задачі	Неня В.Г., Ващенко С.М.	100%	20.09.2023	5
<b>Планування</b>				
Визначення вимог	Неня В.Г.	100%	25.09.2023	5
Визначення технологій та інструментів	Неня В.Г.	100%	30.09.2023	5
Виконання планування проекту	Неня В.Г.	100%	05.10.2023	6
<b>Реалізація</b>				
Проектування інформаційної технології	Неня В.Г.	100%	11.10.2023	14
Проектування та реалізація бази даних шаблонів та повідомлень	Неня В.Г.	100%	25.10.2023	10
Проектування та реалізація додаткових об'єктів бази даних	Неня В.Г.	100%	04.11.2023	5
Реалізація програмного додатку	Неня В.Г.	100%	09.11.2023	10
Тестування	Неня В.Г., Ващенко С.М.	100%	19.11.2023	3
<b>Завершення</b>				
Підготовка та оформлення документації	Неня В.Г.	100%	22.11.2023	3
Підготовка та оформлення інструкції з користування	Неня В.Г.	100%	25.11.2023	5

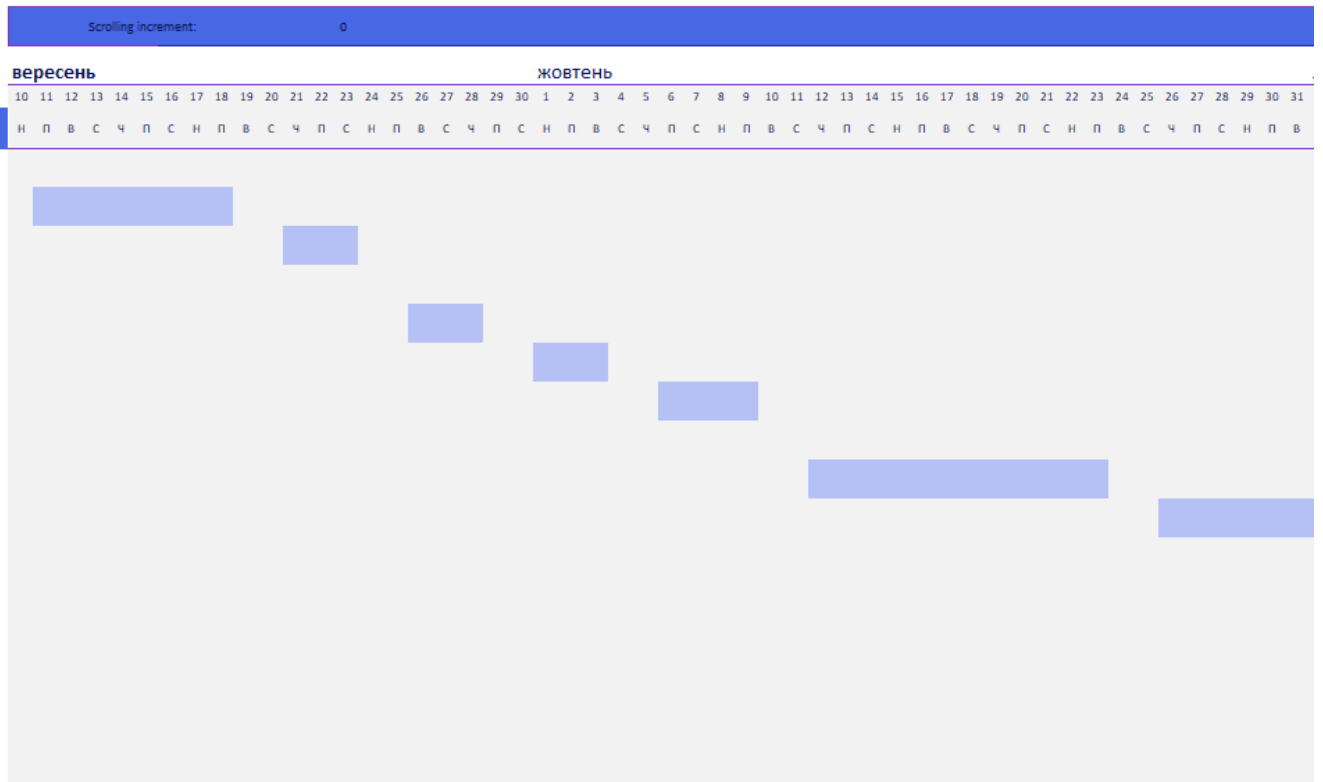


Рисунок Г.3 – Діаграма Ганта. Частина 1.

*Джерело: розроблено автором*

## Інформаційна технологія формування повідомлення за шаблонами стандарту НАТО

Сумський державний університет, Кафедра інформаційних технологій

Неня В.Г.

Початок проекту: 10.09.2023

Опис етапу проекту	Виконавець (ці)	Прогрес	Початок	Тривалість в днях
<b>Ініціювання</b>				
Аналіз предметної області	Неня В.Г.	100%	10.09.2023	10
Постановка задачі	Неня В.Г., Ващенко С.М.	100%	20.09.2023	5
<b>Планування</b>				
Визначення вимог	Неня В.Г.	100%	25.09.2023	5
Визначення технологій та інструментів	Неня В.Г.	100%	30.09.2023	5
Виконання планування проекту	Неня В.Г.	100%	05.10.2023	6
<b>Реалізація</b>				
Проектування інформаційної технології	Неня В.Г.	100%	11.10.2023	14
Проектування та реалізація бази даних шаблонів та повідомлень	Неня В.Г.	100%	25.10.2023	10
Проектування та реалізація додаткових об'єктів бази даних	Неня В.Г.	100%	04.11.2023	5
Реалізація програмного додатку	Неня В.Г.	100%	09.11.2023	10
Тестування	Неня В.Г., Ващенко С.М.	100%	19.11.2023	3
<b>Завершення</b>				
Підготовка та оформлення документації	Неня В.Г.	100%	22.11.2023	3
Підготовка та оформлення інструкції з користування	Неня В.Г.	100%	25.11.2023	5

листопад

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
с ч п с н п в с ч п с н п в с ч п с н п в с ч п с



Рисунок Г.4 – Діаграма Ганта. Частина 2.

*Джерело: розроблено автором*

**Управління ризиками.** Під час реалізації проекту можуть виникнути небажані умови, ситуації та непередбачувані наслідки, які називаються ризиками. Управління ризиком полягає у відповіді на ці події під час виконання проекту шляхом моніторингу та контролю за ними.

Для врахування ймовірності появи ризикових подій, які можуть мати негативні наслідки, створимо таблицю класифікації ризиків (табл. Г.2) та матрицю ризиків (табл. Г.3).

Таблиця Г.2. – Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Некоректно сформовані вимоги	2	4
2	Порушення термінів виконання	3	3
3	Зміна цілей у ході реалізації проекту	3	5
4	Недостатнє тестування	2	3
5	Неправильно обрана стратегія розвитку технології	4	5
6	Помилки у роботі серверу	2	3

*Джерело: розроблено автором*

Таблиця Г.3 – Матриця впливу

Вірогідність виникнення	Матриця впливу				
5			3	5	
4		4			
3		6	2		
2				1	
1					
Ступінь впливу	1	2	3	4	5

*Джерело: розроблено автором*

## ДОДАТОК Д

### ОПИС ТАБЛИЦЬ БАЗИ ДАНИХ DBTEMPLATES

Таблиця Д.1 – Структура таблиці languages

№	Назва поля	Тип даних	Призначення поля
1	id	Integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	language	character varying(25)	Поле зберігає назву мови
3	abb	character(2)	Поле зберігає абрєвіатуру мови

*Джерело: побудовано автором*

Таблиця Д.2 – Структура таблиці msgid

№	Назва поля	Тип даних	Призначення поля
1	msgid_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	ms_title	character varying(50)	Поле зберігає посилання на переклад найменування шаблону повідомлення в таблиці trnslt_msgid
3	ms_abb	character(20)	Поле зберігає посилання на переклад абрєвіатури найменування шаблону повідомлення в таблиці trnslt_msgid
4	blocks_amount	integer	Поле зберігає інформацію про кількість блоків у шаблонах повідомлень
5	standart_version	character varying(15)	Поле зберігає інформацію про версію стандарту шаблонів повідомлень

*Джерело: побудовано автором*

Таблиця Д.3 – Структура таблиці heads

№	Назва поля	Тип даних	Призначення поля
1	head_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	msgid_id	integer	Поле зберігає ключ посилання на головну таблицю msgid; зовнішній ідентифікатор таблиці
3	purpose	character varying(20)	Поле зберігає посилання на переклад призначення шаблону повідомлення в таблиці trnslt_heads
4	nato_sponsor	character varying(20)	Поле зберігає посилання на переклад ідентифікатора NATO Sponsor шаблону повідомлення в таблиці trnslt_heads
5	nato_uf_mtf	character varying(20)	Поле зберігає посилальний ключ на переклад параметра формату користувача / текстового формату повідомлення NATO User Format (UF) / Message Text Format (MTF) шаблону повідомлення в таблиці trnslt_heads
6	us_xxx	integer	Поле зберігає інформацію про можливість застосування шаблону повідомлення рівня XXX і вище
7	us_xx	integer	Поле зберігає інформацію про можливість застосування шаблону повідомлення для рівня XX
8	us_x	integer	Поле зберігає інформацію про можливість застосування шаблону повідомлення для рівня X
9	us_11	integer	Поле зберігає інформацію про можливість застосування шаблону повідомлення для рівня II та нижче

*Джерело: побудовано автором*

Таблиця Д.4 – Структура таблиці templates

№	Назва поля	Тип даних	Призначення поля
1	templ_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	msgid_id	integer	Поле зберігає ключ посилання на головну таблицю msgid; зовнішній ідентифікатор таблиці
3	block_id	integer	Поле зберігає ключ посилання на головну таблицю blocks; зовнішній ідентифікатор таблиці
4	symbols	character(10)	Поле зберігає значення символічного позначення блоків у шаблонах повідомлень
5	discription	character varying(150)	Поле зберігає посилання на переклад додаткового опису блоків у шаблонах повідомлень у таблиці trnslt_templ
6	orders	integer	Поле зберігає інформацію про порядок проходження блоків у шаблонах повідомлень для сортування
7	block_abb	character varying(8)	Поле зберігає інформацію про аббревіатуру найменування блоків шаблонів повідомлень

*Джерело: побудовано автором*

Таблиця Д.5 – Структура таблиці blocks

№	Назва поля	Тип даних	Призначення поля
1	block_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	title	character varying(100)	Поле зберігає ключ посилання на переклад найменування блоків шаблонів повідомлень у таблиці trnslt_blocks
3	parametrs_amount	integer	Поле зберігає інформацію про кількість параметрів блоків шаблонів повідомлень
4	repeatables	integer	Поле зберігає інформацію про можливість повторення блоків у шаблонах повідомлень
5	max_repeat	integer	Поле зберігає інформацію про кількість можливих повторень блоків шаблонів повідомлень



## Продовження таблиці Д.5

№	Назва поля	Тип даних	Призначення поля
6	optional	integer	Поле зберігає інформацію про обов'язковість заповнення блоків шаблонів повідомлень
7	parent_id	integer	Поле зберігає інформацію про первинний ідентифікатор батьківського блоку, від наявності якого залежить наявність поточного блоку в шаблонах повідомлень
8	constraites	character varying(256)	Поле зберігає текстовий опис можливих обмежень на утримання блоків шаблонів повідомлень
9	for_choose	integer	Поле зберігає інформацію про відображення лише одного обраного (1) або всіх (0) вкладених блоків у цьому блоці
10	display_in_table	integer	Поле зберігає інформацію про відображення блоків шаблонів повідомлень у вигляді таблиці
11	start_position	integer	Поле зберігає інформацію про позицію початку введення тексту в блок шаблонів повідомлень
12	text_alignment	integer	Поле зберігає інформацію про параметр вирівнювання тексту в блоках шаблонів повідомлень

*Джерело: побудовано автором*

Таблиця Д.6 – Структура таблиці parametr

№	Назва поля	Тип даних	Призначення поля
1	parametr_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	block_id	integer	Поле зберігає ключ посилання на головну таблицю blocks; зовнішній ідентифікатор таблиці
3	type_id	integer	Поле зберігає ключ посилання на головну таблицю data_type; зовнішній ідентифікатор таблиці
4	title	character varying(100)	Поле зберігає ключ посилання на переклад найменування параметрів блоків шаблонів повідомлень у таблиці trnslt_blocks

## Продовження таблиці Д.6

№	Назва поля	Тип даних	Призначення поля
5	orders	integer	Поле зберігає інформацію про порядок проходження параметрів у блоках шаблонів повідомлень
6	constraite	character varying(250)	Поле зберігає текстовий опис можливих обмежень на значення параметрів блоків шаблонів повідомлень
7	repeatables	integer	Поле зберігає інформацію про можливість повторення параметрів у блоках шаблонів повідомлень
8	max_repeat	integer	Поле зберігає інформацію про кількість можливих повторень параметрів у блоках шаблонів повідомлень
9	options_amount	integer	Поле зберігає інформацію про кількість варіантів допустимих значень параметрів блоків шаблонів повідомлень
10	v_orders	character varying(2)	Поле зберігає інформацію про порядковий номер параметра в батьківському блоці
11	options_column	integer	Поле зберігає інформацію про кількість колонок для відображення допустимих значень параметрів блоків шаблонів повідомлень
12	suboptions_column	integer	Поле зберігає інформацію про кількість колонок для відображення допустимих внутрішніх значень для значень параметрів блоків шаблонів повідомлень
13	detalisation_id	integer	Поле зберігає інформацію про первинний ідентифікатор параметрів, які деталізують поточний параметр блоку шаблонів повідомлень
14	optional	integer	Поле зберігає інформацію про обов'язковість заповнення блоків повідомлень
15	default_value	text	Поле зберігає інформацію про можливі значення за замовчуванням для параметрів блоків шаблонів повідомлень
16	field_description	character varying(50)	Поле зберігає додатковий опис параметрів блоків шаблонів повідомлень

*Джерело: побудовано автором*

Таблиця Д.7 – Структура таблиці options

№	Назва поля	Тип даних	Призначення поля
1	option_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	parameter_id	integer	Поле зберігає ключ посилання на головну таблицю parametrs; зовнішній ідентифікатор таблиці
3	code	character varying(100)	Поле зберігає посилання на переклад кодування допустимого значення параметрів блоків шаблонів повідомлень у таблиці trnslt_options
4	mearning	character varying(100)	Поле зберігає посилальний ключ для перекладу розшифровки кодування допустимого значення параметрів блоків шаблонів повідомлень у таблиці trnslt_options
5	orders	integer	Поле зберігає інформацію про порядок дотримання допустимих значень параметрів блоків шаблонів повідомлень
6	suboptions_amount	integer	Поле зберігає інформацію про кількість допустимих внутрішніх значень для допустимих значень параметрів блоків шаблонів повідомлень

*Джерело: побудовано автором*

Таблиця Д.8 – Структура таблиці suboptions

№	Назва поля	Тип даних	Призначення поля
1	suboption_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	option_id	integer	Поле зберігає ключ посилання на головну таблицю options; зовнішній ідентифікатор таблиці
3	code	character varying(70)	Поле зберігає посилання на переклад кодування допустимого внутрішнього значення параметрів блоків шаблонів повідомлень у таблиці trnslt_suboptions

## Продовження таблиці Д.8

№	Назва поля	Тип даних	Призначення поля
4	mearning	character varying(70)	Поле зберігає посилання на переклад розшифровки кодування допустимого внутрішнього значення параметрів блоків шаблонів повідомлень у таблиці trnslt_suboptions
5	orders	integer	Поле зберігає інформацію про порядок проходження допустимих внутрішніх значень параметрів блоків шаблонів повідомлень

*Джерело: побудовано автором*

## Таблиця Д.9 – Структура таблиці data\_type

№	Назва поля	Тип даних	Призначення поля
1	type_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	title	character varying(25)	Поле зберігає посилання на переклад кодування допустимого внутрішнього значення параметрів блоків шаблонів повідомлень у таблиці trnslt_data_type
3	size	integer	Поле зберігає інформацію про розмір типів даних
4	discription	character varying(150)	Поле зберігає додатковий опис типів даних

*Джерело: побудовано автором*

## Таблиця Д.10 - Структура таблиці relation\_bloks\_params

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	block_id	integer	Поле зберігає ключ посилання на головну таблицю blocks; зовнішній ідентифікатор таблиці; вказує який блок визначає наявність пов'язаного параметра у блоках шаблону

## Продовження таблиці Д.10

№	Назва поля	Тип даних	Призначення поля
3	parametr_id	integer	Поле зберігає ключ посилання на головну таблицю params; зовнішній ідентифікатор таблиці; вказує який параметр виступає пов'язаним із визначальним блоком шаблону
4	constraites	character varying(255)	Поле зберігає опис умови, за якої зазначені блок і параметр пов'язані між собою

*Джерело: побудовано автором*

## Таблиця Д.11 – Структура таблиці relation\_params\_params

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	parametr_id	integer	Поле зберігає ключ посилання на головну таблицю params; зовнішній ідентифікатор таблиці; вказує який параметр визначає наявність у шаблоні пов'язаного параметра
3	relay_id	integer	Поле зберігає ключ посилання на головну таблицю params; зовнішній ідентифікатор таблиці; вказує, який параметр виступає пов'язаним з визначальним параметром шаблону
4	constraites	character varying(255)	Поле зберігає опис умови, за якої зазначені параметри пов'язані між собою

*Джерело: побудовано автором*

Таблиця Д.12 – Структура таблиці trnslt\_msgid

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(50)	Поле зберігає ключ посилання на поля ms_title, ms_abb таблиці msgid
3	en	text	Поле зберігає переклад значень ms_title, ms_abb таблиці msgid англійською мовою
4	ua	text	Поле зберігає переклад значень ms_title, ms_abb таблиці msgid українською мовою

*Джерело: побудовано автором*

Таблиця Д.13 – Структура таблиці trnslt\_heads

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(50)	Поле зберігає ключ посилання на поля purpose, nato_sponsor, nato_utf таблиці heads
3	en	text	Поле зберігає переклад значень purpose, nato_sponsor, nato_utf таблиці heads англійською мовою
4	ua	text	Поле зберігає переклад значень purpose, nato_sponsor, nato_utf таблиці heads українською мовою

*Джерело: побудовано автором*

Таблиця Д.14 – Структура таблиці trnslt\_tmpl

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(50)	Поле зберігає ключ посилання на поле discription таблиці templates
3	en	text	Поле зберігає переклад значень discription таблиці templates англійською мовою
4	ru	text	Поле зберігає переклад значень discription таблиці templates українською мовою

*Джерело: побудовано автором*

Таблиця Д.15 – Структура таблиці trnslt\_blocks

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(50)	Поле зберігає посилальний ключ на полі.
3	en	text	Поле зберігає переклад значень title таблиці blocks на англійську мову
4	ua	text	Поле зберігає переклад значень title таблиці blocks українською

*Джерело: побудовано автором*

Таблиця Д.16 – Структура таблиці trnslt\_paramtrs

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(50)	Поле зберігає посилальний ключ на полі.
3	en	text	Поле зберігає переклад значень title таблиці paramtrs англійською мовою
4	ru	text	Поле зберігає переклад значень title таблиці paramtrs українською мовою

*Джерело: побудовано автором*

Таблиця Д.17 – Структура таблиці trnslt\_options

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(100)	Поле зберігає ключ посилання на поля code, meaning таблиці options
3	en	text	Поле зберігає переклад значень code, meaning таблиці options англійською мовою
4	ua	text	Поле зберігає переклад значень code, meaning таблиці options українською мовою

*Джерело: побудовано автором*

Таблиця Д.18 – Структура таблиці trnslt\_suboptions

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(100)	Поле зберігає ключ посилання на поля code, meaning таблиці suboptions
3	en	text	Поле зберігає переклад значень code, meaning таблиці suboptions англійською мовою
4	ua	text	Поле зберігає переклад значень code, meaning таблиці suboptions українською мовою

*Джерело: побудовано автором*

Таблиця Д.19 – Структура таблиці trnslt\_data\_type

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(100)	Поле зберігає ключ посилання на поле title таблиці data_type
3	en	text	Поле зберігає переклад значень title таблиці data_type англійською мовою
4	ua	text	Поле зберігає переклад значень title таблиці data_type українською мовою

*Джерело: побудовано автором*



Таблиця Д.20 – Структура таблиці templates\_in\_xml

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	msgid_id	integer	Поле зберігає ключ посилання на головну таблицю msgid; зовнішній ідентифікатор таблиці
3	composites_xsd	text	Поле зберігає xsd схему композиту у форматі xml
4	fields_xsd	text	Поле зберігає xsd схему полів у форматі xml
5	messages_xsd	text	Поле зберігає xsd схему повідомлення у форматі xml
6	sets_xsd	text	Поле зберігає xsd схему налаштувань у форматі xml

*Джерело: побудовано автором*

Таблиця Д.21 – Структура таблиці block\_teg\_xml

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	block_id	integer	Поле зберігає ключ посилання на головну таблицю blocks; зовнішній ідентифікатор таблиці
3	open_xml	character varying(10)	Поле зберігає інформацію про тег блоку, що відкривається, для xml-документа
4	close_xml	character varying(10)	Поле зберігає інформацію про тег блоку, що закривається, для xml-документа

*Джерело: побудовано автором*

Таблиця Д.22 – Структура таблиці parametr\_teg\_xml

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	parametr_id	integer	Поле зберігає ключ посилання на головну таблицю params; зовнішній ідентифікатор таблиці
3	open_xml	character varying(10)	Поле зберігає інформацію про тег параметра для xml-документа, що відкривається.
4	close_xml	character varying(10)	Поле зберігає інформацію про тег параметра, що закривається, для xml-документа

*Джерело: побудовано автором*

Таблиця Д.23 – Структура таблиці xml\_version

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	version	character varying(125)	Поле інформацію про версію xml-документа

*Джерело: побудовано автором*

Таблиця Д.24 – Структура таблиці mtf\_config

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	msgid_id	integer	Поле зберігає ключ посилання на головну таблицю msgid; зовнішній ідентифікатор таблиці
3	title	character varying(125)	Поле зберігає заголовок конфігурації xml-документа
4	value	character varying(125)	Поле зберігає значення mtf-конфігурації xml-документа
5	schema_location	character varying(125)	Поле зберігає інформацію про схему розташування xml-документа

*Джерело: побудовано автором*

Таблиця Д.25 – Структура таблиці classification\_groups

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	group_title	character varying(256)	Поле зберігає назву групи класифікацій шаблонів повідомлень
3	classification_parametr	character varying(256)	Поле зберігає назву параметра, за яким класифікуються шаблони у поточній групі класифікацій

*Джерело: побудовано автором*

Таблиця Д.26 – Структура таблиці classifier

№	Назва поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	msgid_id	integer	Поле зберігає ключ посилання на головну таблицю msgid; зовнішній ідентифікатор таблиці
3	classification_group_id	integer	Поле зберігає ключ посилання на головну таблицю classification_groups; зовнішній ідентифікатор таблиці

*Джерело: побудовано автором*

## ДОДАТОК Е

### ОПИС ТАБЛИЦЬ БАЗИ ДАНИХ DBMESSAGES

Таблиця Е.1 – Структура таблиці messages

№	Назва Поля	Тип даних	Призначення поля
1	message_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	date_in	timestamp without time zone	Поле зберігає дату створення запису про повідомлення у базі даних
3	contents	text	Поле зберігає вихідне повідомлення у повному обсязі у текстовому форматі
4	error_description	text	Поле зберігає інформацію про помилку перевірки повідомлення на відповідність шаблону
5	status	integer	Поле зберігає інформацію про статус повідомлення
6	msgid_id_in_tmpl	integer	Поле зберігає інформацію про первинний ідентифікатор шаблону, за яким сформовано повідомлення
7	validation	boolean	Поле зберігає інформацію про результат перевірки повідомлення на відповідність шаблону
8	msgid_abb_in_tmpl	character varying(25)	Поле зберігає аббревіатуру шаблону, яким сформовано повідомлення
9	error_amount	integer	Поле зберігає інформацію про кількість помилок за результатами валідації повідомлення на відповідність шаблону

*Джерело: побудовано автором*

Таблиця Е.2 – Структура таблиці service\_heads

№	Назва Поля	Тип даних	Призначення поля
1	shead_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	message_id	integer	Поле зберігає ключ посилання на головну таблицю messages; зовнішній ідентифікатор таблиці
3	transfer_instruction	character(9)	Поле зберігає інформацію про гриф секретності повідомлення
4	priority	character(18)	Поле зберігає інформацію про пріоритет повідомлення (у форматі DTG)
5	sender	character varying(25)	Поле зберігає інформацію про відправника повідомлення
6	receiver	character varying(100)	Поле зберігає інформацію про ім'я одержувача або ім'я групи адрес
7	receiver_info	character varying(100)	Поле зберігає інформацію про одержувача повідомлення
8	exception	character varying(100)	Поле зберігає інформацію про виключення з групи адрес-адресатів повідомлення
9	groups_amount	character varying(5)	Поле зберігає інформацію про кількість груп закодованого повідомлення
10	channel_name	character(3)	Поле зберігає інформацію про назву каналу
11	channel_number	integer	Поле зберігає інформацію про номер каналу
12	message_priority	character(2)	Поле зберігає інформацію про рівень пріоритету повідомлення
13	rout_station_id	character varying(25)	Поле зберігає інформацію про індикатор станції маршрутизації
14	depature_station_name	character varying(25)	Поле зберігає інформацію про назву станції підготовки повідомлення
15	depature_station_code	character varying(25)	Поле зберігає код станції підготовки повідомлення
16	submission_date_for_sending	character(8)	Поле зберігає дату подачі повідомлення для відправки

*Джерело: побудовано автором*

Таблиця Е.3 – Структура таблиці address

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	adress	character varying(256)	Поле зберігає адреси можливих отримувачів повідомлень

*Джерело: побудовано автором*

Таблиця Е.4 – Структура таблиці address\_groups

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	shead_id	integer	Поле зберігає ключ посилання на головну таблицю service_heads; зовнішній ідентифікатор таблиці
3	group_name	character varying(256)	Поле зберігає назви груп адрес можливих одержувачів повідомлень

*Джерело: побудовано автором*

Таблиця Е.5 – Структура таблиці address\_book

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	group_id	integer	Поле зберігає ключ посилання на головну таблицю address_groups; зовнішній ідентифікатор таблиці
3	adress_id	integer	Поле зберігає ключ посилання на головну таблицю address; зовнішній ідентифікатор таблиці

*Джерело: побудовано автором*

Таблиця Е.6 – Структура таблиці messages\_edited

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	data_edit	integer	Поле зберігає дату редагування повідомлення
3	message_id	integer	Поле зберігає ключ посилання на головну таблицю messages; зовнішній ідентифікатор таблиці
4	content_edited	text	Поле зберігає відредаговане повідомлення в повному обсязі у текстовому форматі

*Джерело: побудовано автором*

Таблиця Е.7 – Структура таблиці blocks

№	Назва Поля	Тип даних	Призначення поля
1	block_id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	message_id	integer	Поле зберігає ключ посилання на головну таблицю messages; зовнішній ідентифікатор таблиці
3	error_description	text	Поле зберігає опис помилки, виявленої в результаті перевірки повідомлення на відповідність шаблону
4	block_id_in_tmpl	integer	Поле зберігає первинний ідентифікатор блоку шаблону в базі даних шаблонів повідомлень
5	usable	boolean	Поле зберігає інформацію про використання цього блоку у повідомленні
6	relay_block_id	integer	Поле зберігає первинний ідентифікатор зв'язаного блоку у повідомленні
7	list_ndx	integer	Поле зберігає порядковий номер виділеного блоку для перегляду
8	parent_id_in_tmpl	integer	Поле зберігає первинний ідентифікатор батьківського блоку в базі даних шаблонів повідомлень
9	orders_in_parent_block	integer	Поле зберігає порядок блоку в батьківському блоці повідомлення
10	validation	boolean	Поле зберігає результат перевірки блоку повідомлення на відповідність шаблону

*Джерело: побудовано автором*

Таблиця Е.8 – Структура таблиці parameter\_value

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	block_id	integer	Поле зберігає ключ посилення на головну таблицю blocks; зовнішній ідентифікатор таблиці
3	parameter_value	text	Поле зберігає значення параметра блоку повідомлення
4	error_description	text	Поле зберігає опис помилки, виявленої в результаті перевірки повідомлення на відповідність шаблону
5	parametr_id_in_tmpl	integer	Поле зберігає первинний ідентифікатор параметра бази даних шаблонів повідомлень
6	usable	boolean	Поле зберігає інформацію про використання цього параметра блоку у повідомленні
7	detalisation_id_in_tmpl	integer	Поле зберігає первинний ідентифікатор уточнюючого параметра у базі даних шаблонів
8	detalisation_list_ndx	integer	Поле зберігає порядковий номер уточнюючого блоку для перегляду
9	relay_parametr_id	integer	Поле зберігає первинний ідентифікатор пов'язаного параметра у повідомленні
10	orders_in_tmpl	integer	Поле зберігає порядковий номер параметра в блоці бази даних шаблонів
11	validation	boolean	Поле зберігає результат перевірки параметра блоку повідомлення на відповідність шаблону

*Джерело: побудовано автором*

Таблиця Е.9 – Структура таблиці call\_station

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	double_code	character(2)	Поле зберігає значення подвійного коду пріоритету повідомлення
3	code	character(1)	Поле зберігає значення коду пріоритету повідомлення
4	mearning	character varying(12)	Поле зберігає посилення на таблицю перекладу розшифровки рівнів пріоритету



*Джерело: побудовано автором*

Таблиця Е.10 – Структура таблиці category\_message

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	category	character varying(6)	Поле зберігає інформацію про категорії повідомлень
3	mearning	text	Поле зберігає посилання на таблицю перекладу категорій повідомлень на мови інтерфейсу trnslt_category_message

*Джерело: побудовано автором*

Таблиця Е.11 – Структура таблиці priority

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	default_symbol	character(3)	Поле зберігає ідентифікатор рядка рівня секретності повідомлення
3	secret_symbols	character(5)	Поле зберігає символічний код із першої літери назви (повторюється п'ять разів) рівня секретності повідомлення
4	secrecy	character varying(23)	Поле зберігає гриф секретності повідомлення
5	classification	character varying(12)	Поле зберігає класифікацію безпеки повідомлення
6	nato_classification	character varying(17)	Поле зберігає НАТО класифікацію повідомлення
7	mearning	character varying(29)	Поле зберігає посилання на таблицю перекладу рівнів секретності повідомлень на мови інтерфейсу trnslt_priority

*Джерело: побудовано автором*

Таблиця Е.12 – Структура таблиці type\_message

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	type	character varying(9)	Поле зберігає тип повідомлення
3	mearning	character varying(20)	Поле зберігає посилання на таблицю перекладу рівнів секретності повідомлень на мови інтерфейсу trnslt_type_message

*Джерело: побудовано автором*

Таблиця Е.13 – Структура таблиці trnslt\_call\_station

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(256)	Поле зберігає ключ посилання на полі mearning таблиці call_station
3	en	character varying(256)	Поле зберігає переклад значень mearning таблиці call_station англійською мовою

*Джерело: побудовано автором*

Таблиця Е.14 – Структура таблиці trnslt\_category\_message

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(256)	Поле зберігає ключ посилання на полі mearning таблиці category_message
3	en	character varying(256)	Поле зберігає переклад значень mearning таблиці category_message англійською мовою

*Джерело: побудовано автором*

Таблиця Е.15 – Структура таблиці trnslt\_priority

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(100)	Поле зберігає ключ посилання на полі meaning таблиці priority
3	en	character varying(256)	Поле зберігає переклад значень meaning таблиці priority англійською мовою

*Джерело: побудовано автором*

Таблиця Е.16 – Структура таблиці trnslt\_type\_message

№	Назва Поля	Тип даних	Призначення поля
1	id	integer	Поле зберігає первинний ідентифікатор рядка; індекс таблиці
2	key	character varying(256)	Поле зберігає ключ посилання на полі meaning таблиці type_message
3	en	character varying(256)	Поле зберігає переклад значень meaning таблиці type_message англійською мовою

*Джерело: побудовано автором*

## Синтаксис створення послідовностей для таблиць бази даних шаблонів повідомлень

Створення послідовності для таблиці msgid виконується згідно запиту:

```
CREATE SEQUENCE public.msgid_msgid_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 91  
CACHE 1;
```

Створення послідовності для таблиці heads виконується згідно запиту:

```
CREATE SEQUENCE public.heads_head_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 83  
CACHE 1;
```

Створення послідовності для таблиці templates виконується згідно запиту:

```
CREATE SEQUENCE public.templates_templ_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 297  
CACHE 1;
```

Створення послідовності для таблиці blocks виконується згідно запиту:

```
CREATE SEQUENCE public.blocks_block_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 292  
CACHE 1;
```

Створення послідовності для таблиці parameters виконується згідно запиту:

```
CREATE SEQUENCE public.parameters_parametrs_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 555
```

```
CACHE 1;
```

Створення послідовності для таблиці options виконується згідно запиту:

```
CREATE SEQUENCE public.options_option_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 22106  
CACHE 1;
```

Створення послідовності для таблиці suboptions виконується згідно запиту:

```
CREATE SEQUENCE public.suboptions_suboption_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 8986  
CACHE 1;
```

Створення послідовності для таблиці data\_type виконується згідно запиту:

```
CREATE SEQUENCE public.data_type_type_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці relation\_blocks\_parameters виконується згідно запиту:

```
CREATE SEQUENCE public.rel_block_param_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 30  
CACHE 1;
```

Створення послідовності для таблиці relation\_parameters\_parameters виконується згідно запиту:

```
CREATE SEQUENCE public.rel_param_param_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807
```

```
START 125  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_msgid` виконується згідно запити:

```
CREATE SEQUENCE public.trnslt_msgid_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 171  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_heads` виконується згідно запити:

```
CREATE SEQUENCE public.trnslt_heads_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 228  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_blocks` виконується згідно запити:

```
CREATE SEQUENCE public.trnslt_blocks_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 292  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_tmpl` виконується згідно запити:

```
CREATE SEQUENCE public.trnslt_tmpl_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 276  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_params` виконується згідно запити:

```
CREATE SEQUENCE public.trnslt_params_id_seg
```

```
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 545  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_options` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_options_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 44201  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_suboptions` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_suboptions_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 17972  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_data_type` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_data_type_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `templates_in_xml` виконується згідно запиту:

```
CREATE SEQUENCE public.templates_in_xml_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 4  
CACHE 1;
```

Створення послідовності для таблиці `block_teg_xml` виконується згідно

запиту:

```
CREATE SEQUENCE public.block_teg_xml_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `parametr_teg_xml` виконується

згідно запиту:

```
CREATE SEQUENCE public.parametr_teg_xml_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `xml_version` виконується згідно

запиту:

```
CREATE SEQUENCE public.xml_version_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `mtf_config` виконується згідно

запиту:

```
CREATE SEQUENCE public.mtf_config_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `classification_groups` виконується

згідно запиту:

```
CREATE SEQUENCE public.classification_groups_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807
```



```
START 1  
CACHE 1;
```

Створення послідовності для таблиці `classifier` виконується згідно запити:

```
CREATE SEQUENCE public.classifier_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

### **Синтаксис створення послідовностей для таблиць бази даних вхідних і вихідних повідомлень**

Створення послідовності для таблиці `messages` виконується згідно запити:

```
CREATE SEQUENCE public.income_message_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 8  
CACHE 1;
```

Створення послідовності для таблиці `messages_edited` виконується згідно запити:

```
CREATE SEQUENCE public.messages_edited_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `service_heads` виконується згідно запити:

```
CREATE SEQUENCE public.service_head_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 3  
CACHE 1;
```

Створення послідовності для таблиці templates виконується згідно запиту:

```
CREATE SEQUENCE public.templates_tmpl_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 297  
CACHE 1;
```

Створення послідовності для таблиці цы address виконується згідно запиту:

```
CREATE SEQUENCE public.address_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці address\_groups виконується згідно запиту:

```
CREATE SEQUENCE public.address_groups_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці address\_book виконується згідно запиту:

```
CREATE SEQUENCE public.address_book_id_seq  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці blocks виконується згідно запиту:

```
CREATE SEQUENCE public.block_id_seg  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1  
CACHE 1;
```

Створення послідовності для таблиці `parameter_value` виконується згідно запиту:

```
CREATE SEQUENCE public.parameter_id_seg
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 1
CACHE 1;
ALTER TABLE public.parameter_id_seg
OWNER TO postgres;
```

Створення послідовності для таблиці `call_station` виконується згідно запиту:

```
CREATE SEQUENCE public.call_station_id_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 4
CACHE 1;
```

Створення послідовності для таблиці `category_message` виконується згідно запиту:

```
CREATE SEQUENCE public.category_message_id
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 3
CACHE 1;
```

Створення послідовності для таблиці `priority` виконується згідно запиту:

```
CREATE SEQUENCE public.priority_id_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 5
CACHE 1;
```

Створення послідовності для таблиці `ицы type_message` виконується згідно запиту:

```
CREATE SEQUENCE public.type_message_id_seq
INCREMENT 1
```

```
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 4  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_call_station` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_call_station_id  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 4  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_category_message` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_category_message_id  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 3  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_priority` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_priority_id  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 5  
CACHE 1;
```

Створення послідовності для таблиці `trnslt_type_message` виконується згідно запиту:

```
CREATE SEQUENCE public.trnslt_type_message_id  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 8  
CACHE 1;
```

## ДОДАТОК Ж

### Синтаксис створення ригерних функцій та тригерів бази даних

#### dbMessages

Тригерна функція `trigger_category_message_delete_translate` для реалізації зв'язку між таблицями `category_message` та `trnslt_category_message`:

```
CREATE OR REPLACE FUNCTION public.trigger_category_message_delete_translate()
  RETURNS trigger AS
$BODY$
begin
delete from trnslt_category_message where
trim(trnslt_category_message.key)=trim(OLD.mearning);
return OLD;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.trigger_category_message_delete_translate()
  OWNER TO postgres;
```

Тригер `category_message_delete` спрацьовує перед видаленням запису з таблиці `category_message`, викликає функцію `trigger_category_message_delete_translate`. Код тригера:

```
CREATE TRIGGER category_message_delete
  BEFORE DELETE
  ON public.category_message
  FOR EACH ROW
  EXECUTE PROCEDURE public.trigger_category_message_delete_translate();
```

Тригерна функція `trigger_priority_delete_translate` для реалізації зв'язку між таблицями `priority` та `trnslt_priority`:

```
CREATE OR REPLACE FUNCTION public.trigger_priority_delete_translate()
  RETURNS trigger AS
$BODY$
begin
delete from trnslt_priority where trim(trnslt_priority.key)=trim(OLD.mearning);
return OLD;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.trigger_priority_delete_translate()
  OWNER TO postgres;
```

Тригер `priority_delete` спрацьовує перед видаленням запису таблиці `priority`, викликає функцію `trigger_priority_delete_translate`. Код тригера:

```
CREATE TRIGGER priority_delete
  BEFORE DELETE
  ON public.priority
  FOR EACH ROW
  EXECUTE PROCEDURE public.trigger_priority_delete_translate();
```

Тригерна функція `trigger_type_message_delete_translate` для реалізації зв'язку між таблицями `type_message` та `trnslt_type_message`:

```
CREATE OR REPLACE FUNCTION public.trigger_type_message_delete_translate()
  RETURNS trigger AS
  $BODY$
begin
delete from trnslt_type_message where trim(trnslt_type_message.key)=trim(OLD.mearning);
return OLD;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.trigger_type_message_delete_translate()
  OWNER TO postgres;
```

Тригер `type_message_delete` спрацьовує перед видаленням запису з таблиці `type_message`, викликає функцію `trigger_type_message_delete_translate`. Код тригера:

```
CREATE TRIGGER type_message_delete
  BEFORE DELETE
  ON public.type_message
  FOR EACH ROW
  EXECUTE PROCEDURE public.trigger_type_message_delete_translate();
```