

Вступ

Шановна комісіє, шановні присутні! Вашій увазі пропонується рішення на тему— «Підсистема попередження та контролю виконання елементів розпорядку дня чергового військової частини»

Актуальність розробки підсистеми контролю розпорядку дня

Чітке дотримання розпорядку дня є фундаментальним для будь-якого військового підрозділу. Це не просто набір правил, а основа військової дисципліни, запорука високої бойової готовності, злагодженості дій особового складу та здатності оперативно реагувати на будь-які зміни в обстановці. Особлива відповідальність за це покладається на чергову зміну.

Однак, як показує аналіз, існуючі методи контролю за виконанням розпорядку дня часто спираються на застарілі підходи. Переважно це ручний контроль та ведення численних паперових журналів. Такий підхід неминуче пов'язаний з низкою проблем: високий ризик помилок через людський фактор, низька оперативність отримання узагальненої інформації про стан справ, а також значні витрати часу на ведення документації, що відволікає особовий склад від виконання інших важливих завдань.

Саме тому виникає гостра потреба в автоматизації цих процесів. Впровадження сучасної інформаційної підсистеми дозволить не тільки підвищити ефективність служби чергового, а й забезпечити більш точний та своєчасний контроль, зменшити адміністративне навантаження та надати керівництву дієві інструменти для аналізу та прийняття рішень.

Мета та завдання кваліфікаційної роботи

Метою даної кваліфікаційної роботи є підвищення ефективності несення служби черговим військової частини.

Ми прагнули посилити контроль за виконанням встановленого розпорядку дня, розробивши для цього спеціалізовану програмну підсистему.

Важливою складовою було використання сучасних веб-технологій, що забезпечують гнучкість та доступність рішення.

Для досягнення цієї мети було поставлено та вирішено наступні основні завдання:

По-перше, ми провели детальний аналіз предметної області, вивчили специфіку служби чергового та існуючі підходи до контролю розпорядку дня.

По-друге, на основі аналізу було обґрунтовано вибір технологічного стеку — це Java Spring Boot для серверної частини,

React для клієнтської та PostgreSQL для бази даних — а також спроектовано загальну архітектуру підсистеми та її модель даних.

Третє, і ключове, завдання полягало в реалізації основних програмних модулів.

Сюди входить управління розкладом дня, система автоматичних сповіщень для чергового, механізми фіксації виконання завдань,

а також інструменти для формування звітів та адміністрування системи.

Четверте – забезпечення безпеки даних шляхом впровадження надійних механізмів автентифікації та

розмежування доступу користувачів на основі їхніх ролей.

І на завершення, було проведено комплексне тестування розробленої підсистеми для перевірки її функціональності та оцінки потенційної ефективності.

Аналіз предметної області та існуючих рішень

Для ефективної розробки нашої підсистеми ми провели детальний аналіз предметної області.

Служба чергового військової частини відіграє ключову роль у повсякденній діяльності та забезпеченні боєготовності.

Вона охоплює широкий спектр завдань, від контролю за дотриманням особовим складом розпорядку дня до реагування на позаштатні ситуації.

Особливу увагу було приділено елементам розпорядку дня, які потребують чіткого контролю та своєчасного виконання.

Це включає як планові заходи – такі як шикування, проведення занять, прийом їжі, так і оперативні завдання, пов'язані з отриманням та передачею сигналів і команд, а також веденням відповідної службової документації.

Аналіз існуючих методів контролю показав, що вони переважно базуються на ручних операціях та паперовому документообігу.

Це, на жаль, призводить до низки проблем: по-перше, значний вплив людського фактора, що може спричиняти помилки та пропуски важливої інформації.

По-друге, низька оперативність отримання узагальнених даних про стан виконання розпорядку дня.

І по-третє, значна трудомісткість ведення паперової звітності.

Нами також було розглянуто існуючі програмні рішення, проте більшість з них або не повністю відповідають специфіці військової служби, або є занадто загальними та не враховують усіх потреб чергового військової частини.

Все це підтвердило необхідність розробки спеціалізованої автоматизованої підсистеми, яка б вирішувала виявлені проблеми.

Архітектура підсистеми

Для реалізації нашої підсистеми попередження та контролю була обрана класична трирівнева клієнт-серверна архітектура, яка забезпечує чітке розділення функціональних компонентів та гнучкість системи.

На слайді ви можете бачити основні компоненти цієї архітектури.

Клієнтська частина, або фронтенд, розроблена з використанням бібліотеки **React.js**.

Вона відповідає за весь користувацький інтерфейс, з яким взаємодіють чергові, командний склад та адміністратори.

Саме клієнтська частина забезпечує відображення розпорядку дня, сповіщень, форм для введення даних та звітів.

Вона спілкується з серверною частиною через RESTful API, обмінюючись даними у форматі JSON.

Серверна частина, або бекенд, є ядром нашої підсистеми і розроблена на платформі **Java Spring Boot**.

Вона взяла на себе всю основну бізнес-логіку: управління розкладом дня, генерацію та відправку попереджень,

обробку статусів виконання завдань, а також управління користувачами та їхніми правами доступу.

Для забезпечення безпеки використовується механізм JWT-токенів.

І нарешті, **рівень даних** представлений системою управління базами даних **PostgreSQL**.

Тут зберігається вся необхідна інформація: структура розпорядку дня, дані користувачів, історія виконання завдань, налаштування системи та інша службова інформація.

Така архітектура дозволяє незалежно розвивати та масштабувати клієнтську та серверну частини, а також забезпечує надійне зберігання даних.

Модель даних підсистеми (Структура Баз Даних)

Для зберігання всієї необхідної інформації нашої підсистеми було спроектовано реляційну базу даних під управлінням СУБД PostgreSQL.

На слайді представлена спрощена ER-діаграма, що ілюструє ключові сутності та зв'язки між ними. Якщо хочете побачити _ додаток

Реалізація ключових механізмів підсистеми

Перейдемо до ключових аспектів реалізації нашої підсистеми.

(Вказуючи на блок/схему серверної частини) Серверна частина була розроблена на платформі Java Spring Boot.

Ми реалізували набір RESTful API ендпоінтів, які забезпечують всю необхідну взаємодію з клієнтською частиною.

Основна бізнес-логіка, така як управління елементами розпорядку дня, генерація попереджень та контроль за виконанням, зосереджена у відповідних сервісних класах.

Для взаємодії з базою даних PostgreSQL використовується Spring Data JPA та Hibernate, що дозволяє працювати з даними на об'єктному рівні.

Важливим аспектом є безпека: для автентифікації користувачів та захисту API реалізовано механізм на основі JSON Web Tokens, інтегрований за допомогою Spring Security.

(Вказуючи на блок/схему клієнтської частини або скріншот коду) Клієнтська частина, створена на React, побудована за компонентним принципом.

Це дозволило нам створити гнучкий та легко підтримуваний інтерфейс. Для управління станом компонентів та загальним станом додатку ми використовували стандартні хуки React,

такі як `useState`, `useEffect` та `useContext`. Взаємодія з бекендом здійснюється через спеціально розроблений сервіс `apiService.js`, який обробляє всі HTTP-запити та роботу з JWT-токенами.

Навігація між сторінками реалізована за допомогою бібліотеки `React Router`.

Всі ці компоненти та технології були інтегровані для створення функціонального та надійного програмного продукту.

Функціонал чергового військової частини

Після успішної автентифікації користувач з роллю "Черговий військової частини" потрапляє на головну робочу сторінку, розроблену для максимальної зручності та оперативності.

(Показуючи на скріншот головної панелі чергового) Як ви бачите, центральне місце займає актуальний розпорядок дня на поточну добу.

Усі заплановані заходи відображені у хронологічному порядку із зазначенням часу, назви та короткого опису.

Система автоматично виділяє поточні завдання та ті, час виконання яких наближається, що дозволяє черговому ефективно планувати свої дії.

Важливою функцією є **система сповіщень**. Підсистема завчасно інформує чергового про наближення часу виконання ключових елементів розпорядку дня.

Ось приклад такого сповіщення. Також передбачена індикація прострочених завдань.

Для **фіксації виконання завдань** черговий має простий та інтуїтивно зрозумілий інструмент.

(Показуючи на скріншот з можливістю відмітки) Навпроти кожного пункту розпорядку є можливість відмітити його як "Виконано". При цьому система автоматично фіксує час та дозволяє додати необхідні коментарі або деталі щодо виконання.

Ця інформація негайно стає доступною для перегляду командним складом.

Крім того, черговий має швидкий доступ до необхідної **довідкової інформації**, такої як завантажені інструкції, накази чи інші важливі документи.

Також з його робочого місця можливий доступ до карти повітряних тривог для оперативного реагування.

Функціонал коменданта

Для коменданта та осіб, відповідальних за контроль несення служби, наша підсистема надає низку важливих інструментів.

Після входу в систему, офіцер отримує доступ до узагальненої інформації про стан виконання розпорядку дня підпорядкованими черговими змінами.

Він може оперативно бачити, які завдання виконані, які перебувають у процесі виконання, а які – прострочені або не виконані.

Для забезпечення оперативного реагування, командний склад може отримувати **сповіщення про критичні порушення** або невиконання особливо важливих завдань черговою зміною.

Також, залежно від наданих прав, офіцер може мати можливість переглядати, затверджувати або вносити корективи до розкладу дня.

Таким чином, функціонал для командного складу спрямований на забезпечення ефективного контролю, надання актуальної інформації для прийняття рішень та підвищення загального рівня виконавчої дисципліни.

Адміністрування та додатковий функціонал

Для забезпечення стабільної роботи та гнучкого налаштування підсистеми передбачено окремий функціонал для адміністратора.

(Показуючи на скріншот(и) адмін-панелі) Адміністратор може керувати довідниками, наприклад, редагувати список типових елементів розпорядку дня,

які потім використовуються для формування розкладів, або налаштовувати параметри системи сповіщень.

Крім того, для всіх авторизованих користувачів або для певних ролей доступний додатковий функціонал, наприклад, інтегрована карта повітряних тривог (показати скріншот),

яка дозволяє оперативно відстежувати безпекову ситуацію в країні. Система також дозволяє централізовано керувати базою документів та інструкцій, завантажувати нові.

Таким чином, адміністративний функціонал забезпечує необхідні інструменти для підтримки та адаптації підсистеми до конкретних потреб військової частини.

ВИСНОВКИ

Підводячи підсумки виконаної кваліфікаційної роботи, можна стверджувати, що поставлена мета була досягнута.

Нами було успішно розроблено та реалізовано програмну підсистему попередження та контролю виконання елементів розпорядку дня чергового військової частини.

В процесі роботи було проведено всебічний аналіз предметної області, на основі якого обґрунтовано архітектурні рішення та обрано сучасний технологічний стек,

що включає Java Spring Boot, React та PostgreSQL.

Було спроектовано та реалізовано модель даних та всі ключові функціональні модулі, зокрема, управління розкладом, систему сповіщень,

механізми фіксації виконання завдань, інструменти для звітності та адміністрування, а також забезпечено безпеку на основі JWT-автентифікації та розмежування доступу за ролями.

Проведене тестування підтвердило працездатність розробленого програмного продукту.

Практична цінність розробленої підсистеми полягає у підвищенні ефективності контролю за несенням служби,

значному зменшенні впливу людського фактора та обсягів паперової роботи, а також у наданні командному складу зручних інструментів для оперативного отримання інформації та аналізу.

Це, в свою чергу, сприяє зміцненню військової дисципліни та оптимізації управлінських процесів.

Щодо можливих напрямків подальшого розвитку, ми бачимо потенціал у розширенні аналітичних можливостей та

функціоналу звітності, інтеграції з іншими інформаційними системами, що використовуються у військовій частині,

а також у розробці мобільного клієнтського додатку для ще більшої гнучкості та оперативності.

Фінал

Дякую за увагу, готовий дати відповіді на ваші запитання