

Міністерство освіти і науки України  
Сумський державний університет  
Навчально-науковий інститут бізнес-технологій «УАБС»  
Кафедра економічної кібернетики

## КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему «АВТОМАТИЗАЦІЯ ПРОЦЕСІВ УПРАВЛІННЯ РОЗПОДІЛЕНИМИ  
КОМАНДАМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ЧАТ-БОТІВ»

Виконав студент 2 курсу, групи ЕК.м-81а  
(номер курсу) (шифр групи)

Спеціальності 051 «Економіка»

(«Економічна кібернетика»)

Козирка В. Є.

(прізвище, ініціали студента)

Керівник доцент, к.т.н. Яценко В.В.

(посада, науковий ступінь, прізвище, ініціали)

## ЗМІСТ

1 ДОСЛІДЖЕННЯ ПРОЦЕСІВ УПРАВЛІННЯ В КОМПАНІЇ І ФОРМУВАННЯ ВИМОГ ДО АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	5
1.1 Загальна характеристика компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab).....	5
1.2 Комунікаційні потоки в компанії .....	9
1.3 Аналіз стану управління персоналом в компанії і доступних для впровадження HRMS рішень для автоматизації управління персоналом ....	13
1.4 Формування вимог до інформаційної системи .....	18
2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	22
2.1 Моделі бізнес-процесів на підприємстві .....	22
2.2 Архітектура автоматизованої інформаційної системи та технології вирішення поставлених завдань .....	26
2.3 Функціональна структура завдання .....	28
2.4 Характеристика системного і прикладного програмного забезпечення та підсистем для забезпечення функціональної частини. ....	31
3 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	34
3.1 Реалізація, налаштування, загальний алгоритм роботи чат-бота .....	34
3.2 Контрольний приклад та інструкція щодо використання чат-бота.....	45
3.3 Оцінювання очікуваного ефекту від впровадження розробленого автоматизованого рішення.....	49
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	54
ДОДАТКИ.....	62
ДОДАТОК А.....	63
ДОДАТОК Б.....	64
ДОДАТОК Г .....	67

## ВСТУП

Актуальність теми дослідження. Робота в команді і безпосередня тісна взаємодія між усіма учасниками команди є важливою і невід'ємною частиною процесу роботи у ІТ-компаніях. Актуальним явищем сучасного світу і наслідком глобалізації є поява так званих «розподілених команд» — термін, зазвичай відноситься до групи спеціалістів, які працюють разом з різних географічних місць і покладаються на комунікаційні технології, такі як електронна пошта (Gmail, Outlook), послуги відео або голосового конференц зв'язку для спільної роботи (Skype, Zoom, Hangouts), месенджери для швидкого обміну інформацією (Telegram, Slack, Teams, Skype) [61, 62, 63, 69]. Термін також може відноситися до команд чи окремих ІТ-професіоналів — іноземців, на посадах, пов'язаних зі створенням та запровадженням на підприємстві результатів комп'ютерного програмування або високо захищених засобів збереження і захищення інформації в компаніях, які працюють разом асинхронно або на всіх рівнях організації [11]. Пауелл, Пікколи і Айвс визначають віртуальні команди як «групи географічно, організаційно та/або розкиданих по часу працівників, об'єднаних інформаційними і телекомунікаційними технологіями для виконання організаційних задач» [16]. Зважаючи на поширення розподілених команд у сучасних компаніях є достатньо актуальною задача по впровадженні інструментів для налагодження, покращення і пришвидшення організаційних процесів у командах. Одним із інструментів для покращення таких процесів можуть бути чат-боти, які дозволяють виконати чи повідомити про певні зміни в структурі команди чи робочих планах [58, 59, 43].

Мета роботи. Дослідження процесів управління розподіленими командами та програмна реалізація прототипу чат-боту підтримки взаємодії у команді.

Об'єкт дослідження. Процеси управління розподіленими командами в ІТ-компанії з тестування програмного забезпечення.

Предмет дослідження. Інформаційна система управління розподіленими командами на основі чат-боту.

В рамках дослідження ставляться такі завдання:

- дослідити загальну характеристика компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab);
- дослідити комунікаційні потоки в компанії;
- провести аналіз стану управління персоналом в компанії і доступних для впровадження HRMS-рішень для автоматизації управління персоналом;
- сформулювати вимоги до інформаційної системи;
- дослідити моделі бізнес-процесів на підприємстві;
- розробити архітектуру автоматизованої інформаційної системи та підібрати технології вирішення поставлених завдань;
- дослідити функціональну структура завдання;
- підібрати характеристики системного і прикладного програмного забезпечення та підсистем для забезпечення функціональної частини;
- розробити і провести налаштування чат-ботів і навести загальний алгоритм їх роботи;
- надати контрольний приклад та інструкція щодо використання чат-бота;
- провести оцінювання очікуваного ефекту від впровадження розробленого автоматизованого рішення.

Методи дослідження. Теоретичною базою дослідження є сучасні теорії і наукові концепції, монографії та періодичні публікації українських і зарубіжних учених з питань інформаційних технологій, менеджменту, формування механізмів управління, формування конкурентних стратегій. У роботі використано законодавчі і нормативні акти України [1], інструктивні відомчі матеріали.

Інформаційну базу дослідження склали: економічна література і періодичні видання, аналітичні дані з різних напрямків діяльності підприємства, програми розвитку підприємства, рішення органів державної влади та місцевого самоврядування, а також результати досліджень, виконаних автором. У роботі використані статистичні та евристичні методи, порівняльний та техніко-економічний аналіз.

# 1 ДОСЛІДЖЕННЯ ПРОЦЕСІВ УПРАВЛІННЯ В КОМПАНІЇ І ФОРМУВАННЯ ВИМОГ ДО АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

## 1.1 Загальна характеристика компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab)

ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab) є однією з найвідоміших аутстафінгових/аутсорсингових українських організацій орієнтованих на тестування програмного забезпечення, що забезпечує весь спектр таких послуг. Компанія була створена у 2005 році і з того часу продовжує свій стрімкий ріст.

У 2019 році представництва компанії розміщуються вже в 7 регіонах держави. Компанія за 14 років своєї професійної діяльності реалізувала більш ніж 3000 проектів різного рівня складності для провідних світових ІТ-компаній. До портфоліо компанії входять такі компанії як: Bosch, G8, Yandex technology, Wire, Breadmery, Schell та багато інших.

Багатоманітність, складність і варіативність завдань і проектів, які входять до сфери відповідальності компанії, а також кількість виконаних замовлень, десятки тисяч задокументованих дефектів і помилок і багато іншого — все це дозволяє компанії стверджувати що її співробітники є визнаними майстрами в тестуванні, а також йти в ногу з прогресом.

Впродовж декількох останніх років QATestLab стабільно входить до рейтингу «Top 20 Software Application Testing Firms», які публікує рейтингова агенція CLUTCH. Також компанія QATestLab є афілійованими учасниками «Ukrainian Hi-Tech Initiative», «Central and Eastern European Outsourcing Association», а також фундації «Brain Basket».

Станом на 4 квартал 2019 року у компанії працює понад 200 тестувальників програмного забезпечення, і компанія працює в 5 напрямках:

- web digital;
- mobile;
- game;
- eCommerce;
- test automation.

Компанія пропонує повний спектр послуг із забезпечення якості, включаючи написання планів тестування, створення і виконання контрольних прикладів або контрольних списків, виконання як ручного, так і автоматичного тестування, а також підготовку докладних звітів про тестування.

Структура організації ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab) представлена гендиректором, виконавчим директором, директором з розвитку, бухгалтерського відділу, менеджерів програм, програмістів, тестувальників, адміністраторів систем. Детальна структура управлінської гілки відображена на рисунку 1.1.

До сфери відповідальності генерального директора входять:

- розробка технічних аспектів стратегії компанії для забезпечення відповідності її бізнес-цілям;
- досліджує і впроваджує нові технології, які дають конкурентну перевагу;
- допомагає відділам зважено і правильно використовувати технології;
- контролює інфраструктуру системи для забезпечення функціональності та ефективності;
- займається побудовою процесів забезпечення якості та захисту даних;
- займається моніторингом КРІ і ІТ-бюджетів для оцінки технологічних показників;
- використовує відгуки зацікавлених сторін для інформування про необхідні поліпшення і коректування технології;
- повідомляє технологічну стратегію партнерам і інвесторам.

Виконавчий директор здійснює свою діяльність за наступними напрямками:

- аналізує поточні і минулі фінансові дані, такі як звіти про продажі, і пропонує стратегії для скорочення витрат і збільшення доходів;

- проводить дослідження ринку, щоб визначити нові можливості і поспілкуватися з керівниками, щоб розробити стратегії для реалізації цих нових можливостей;
- створює та покращує пропозиції для існуючих і нових клієнтів.
- забезпечує навчання та наставництво для учасників команди розвитку бізнесу.

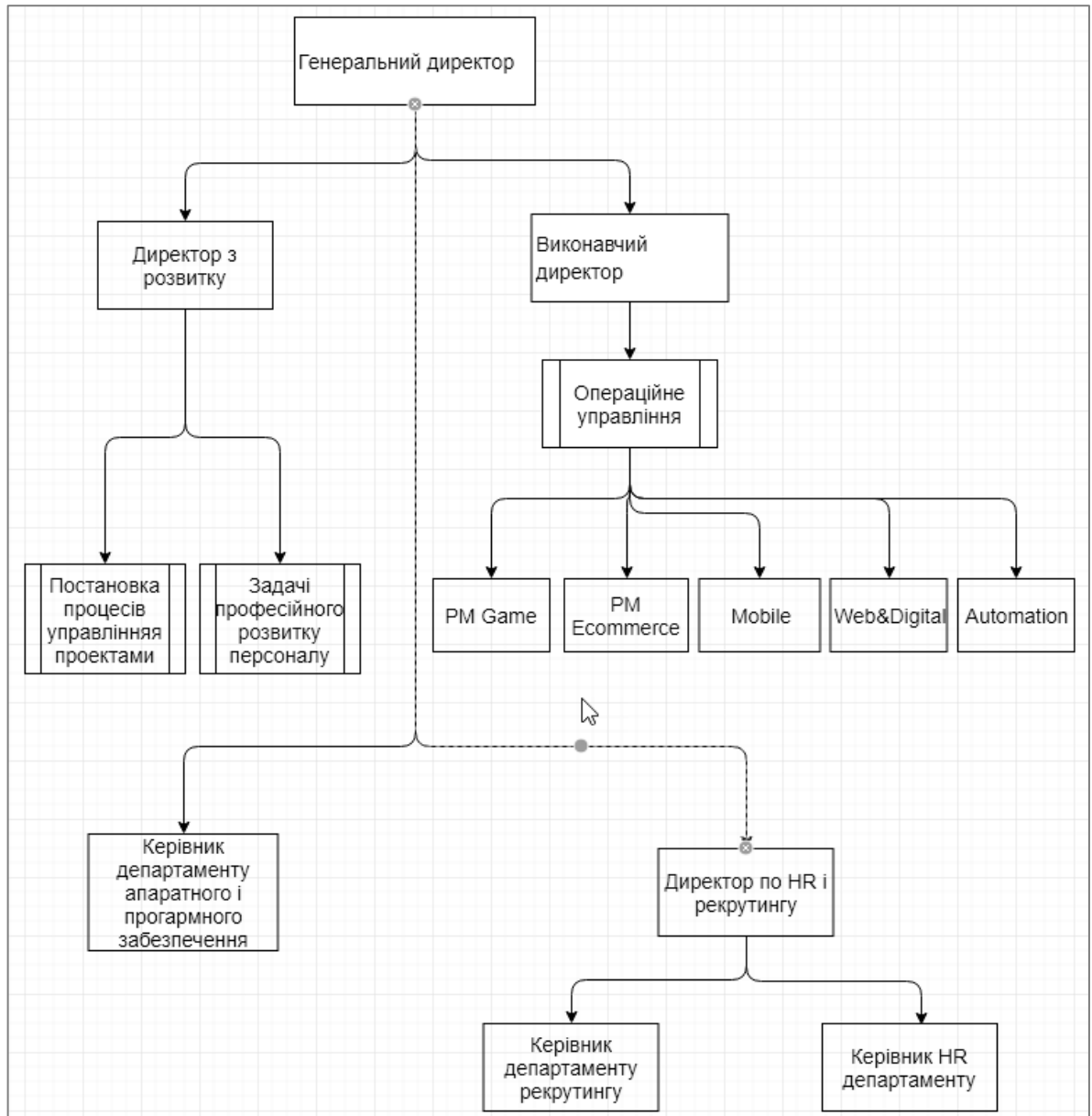


Рисунок 1.1 — Організаційна структура ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab)

Сферою відповідальності працівника на посаді бухгалтера є такі обов'язки:

- облік фінансових операцій шляхом створення рахунків; проводки транзакцій; забезпечує відповідність вимогам законодавства;

- розробляє систему обліку фінансових операцій шляхом складання плану рахунків; визначення бухгалтерських політик і процедур.

- ведення допоміжних рахунків шляхом перевірки, розподілу і проводки транзакцій.

- баланс допоміжних рахунків шляхом звірки записів.

- ведення головної книги шляхом перенесення допоміжних рахунків.

- баланси бухгалтерської книги шляхом підготовки пробного балансу; узгодження записів.

- ведення історичних записів шляхом подачі документів.

- готує фінансові звіти, збираючи, аналізуючи та узагальнюючи інформацію про діяльність.

- займається поданням звітів; консультуванням керівництва щодо необхідних дій.

Адміністратор систем в своїй операційній діяльності займається такими напрямками як:

- Установка і налаштування програмного забезпечення, обладнання і мереж;

- Моніторинг продуктивності системи і усунення неполадок;

- Забезпечення безпеки та ефективності ІТ-інфраструктури.

Програмісти у своїй діяльності виконують такі задачі:

- дослідження, проектування, впровадження та управління програмними засобами;

- визначення областей для модифікації в існуючих програмах та подальшої розробки цих модифікацій;

- написання та реалізація ефективного коду;

- визначення експлуатаційної практичності;

- розробка процедур забезпечення якості;



- розгортання програмних засобів, процесів та показників;
- підтримання та модернізація існуючих систем;
- навчання користувачів.

Тісно співпрацюючи з іншими розробниками, UX-дизайнерами, бізнес-системними аналітиками відділ тестування у своїй діяльності виконує такі задачі:

- забезпечення якості продукту чи рішення, що випускаються у відповідності до заданих вимог та ресурсів;
- автоматизація тестування продукту;
- комунікації з клієнтом;
- забезпечення контролю розробки продукту;
- пошук та аналіз необхідного контенту, організації контенту, а також підтримка актуальності інформації на двох мовах в корпоративній базі знань;
- допомога в складанні документів, презентацій, статей, а також перевірка помилок.

## 1.2 Комунікаційні потоки в компанії

Комунікація в рамках бізнесу може включати в себе різні типи комунікацій співробітників і різні функціональні взаємозалежності між частинами організації, різними рівнями керування і різними офісами корпорацій. Моделі такого спілкуванні і взаємодії найчастіше називаються потоками, і вони зазвичай класифікуються відповідно до напрямку взаємодії: низхідні(спадні), висхідні, горизонтальні, діагональні, зовнішні.

Спадний потік комунікації. Комунікація, яка тече від більш високого рівня в організації до більш низького рівня, є низхідною комунікацією. Іншими словами, спілкування начальників з підлеглими в командному ланцюжку є низхідним спілкуванням. Цей комунікаційний потік використовується менеджерами для передачі пов'язаної з роботою інформації співробітникам на більш низьких рівнях.

Співробітникам потрібна ця інформація для виконання своєї роботи і для задоволення очікувань своїх менеджерів. Низхідний зв'язок використовується менеджерами в наступних цілях:

- надання зворотного зв'язку про роботу співробітників;
- надання робочих інструкцій;
- забезпечення повного розуміння роботи співробітників, а також інформування їх про те, як їх робота пов'язана з іншими роботами в організації;
- передача місії і бачення організації співробітникам;
- «підсвічування» важливих областей.

Висхідний потік спілкування. Комунікації, які переходять на більш високий рівень в організації, називають висхідним спілкуванням. Такий тип потоку забезпечує зворотний зв'язок про те, наскільки добре організація працює. Підлеглі використовують висхідне спілкування, щоб донести свої проблеми і потреби до керівництва. Підлеглі також використовують висхідний тип спілкування, щоб сказати, наскільки добре вони зрозуміли директиви отримані через спадний тип спілкування. Співробітники також можуть використовувати його для обміну думками та ідеями, а також для участі в процесі прийняття рішень.

Бічна/горизонтальна комунікація. Комунікація, яка відбувається на тих же рівнях ієрархії в організації, називається бічною комунікацією, тобто комунікацією між колегами, між менеджерами на тих же рівнях або між будь-яким горизонтально-еквівалентним членом організації. Переваги горизонтального спілкування полягають в наступному:

- економія часу;
- полегшення координації;
- полегшення співпраці між учасниками команди;
- забезпечення емоційної і соціальної допомоги членам організації;
- допомога у вирішенні різних організаційних проблем;
- обмін інформацією.

Такий спосіб комунікації також може бути використаний для вирішення конфліктів відділу з іншими відділами або конфліктів усередині відділу.

Діагональне спілкування. Спілкування між менеджером і співробітниками інших робочих груп називається діагональним спілкуванням. Зазвичай подібні взаємодії не відображаються в організаційній структурі компанії і є неформальними. Структура цього типу комунікації відображена на рисунку 1.2

Зовнішня комунікація. Комунікація, яка відбувається між менеджером і зовнішніми групами, такими як постачальники, банки, фінансові інститути, сторонні фірми і підприємства, тощо. Наприклад, для залучення капіталу керуючий директор взаємодіє з менеджером банку.

Комунікація є ключовим елементом у створенні цілісної системи взаємозв'язків між усіма відділами і всередині всіх робочих груп. Відмінне спілкування може гарантувати, що всі на одній хвилі і мають достатньо можливостей для своїх запитань і вирішення проблем, а також створення безпечного простору для відкритого діалогу. З іншого боку, погана якість і складність спілкування може привести до проблем і труднощів через неправильне спілкування, не кажучи вже про низький моральний дух колективу чи навіть можливість виснаження і морального вигорання в такому колективі [75, 52].

Нижче розглянемо детальніше схему комунікацій в середині компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab). Для компанії є характерним строго вибудований ієрархічний спадний потік інформації з певними вкрапленнями висхідного, діагонального і горизонтального спілкування, а також зовнішнє спілкування з клієнтами і їх командами. В компанії всі інформаційні потоки можна охарактеризувати як потік згори вниз. Для прикладу доцільно розглянути як буде відбуватися рух будь-якої інформації від генерального директора до підлеглих найнижчої ланки, а саме тестувальників програмного забезпечення:

1. Генеральний директор віддає наказ;
2. Наказ надходить до виконавчого директора або директора з розвитку;
3. Директор доводить інформацію до відома програм менеджерів усіх (чи тільки якогось одного) напрямків роботи;

4. Програм менеджери поширюють інформацію до відомих своїх підлеглих груп-координаторів;

5. Груп-координатори, в свою чергу, поширюють інформацію між своїми підлеглими тестувальниками;

6. Тестувальники використовують отриману інформацію.

Рисунок 1.3 відображає інформаційні потоки в компанії ТОВ «Центр мережних технологій ВЕБ100» (QATestLab).

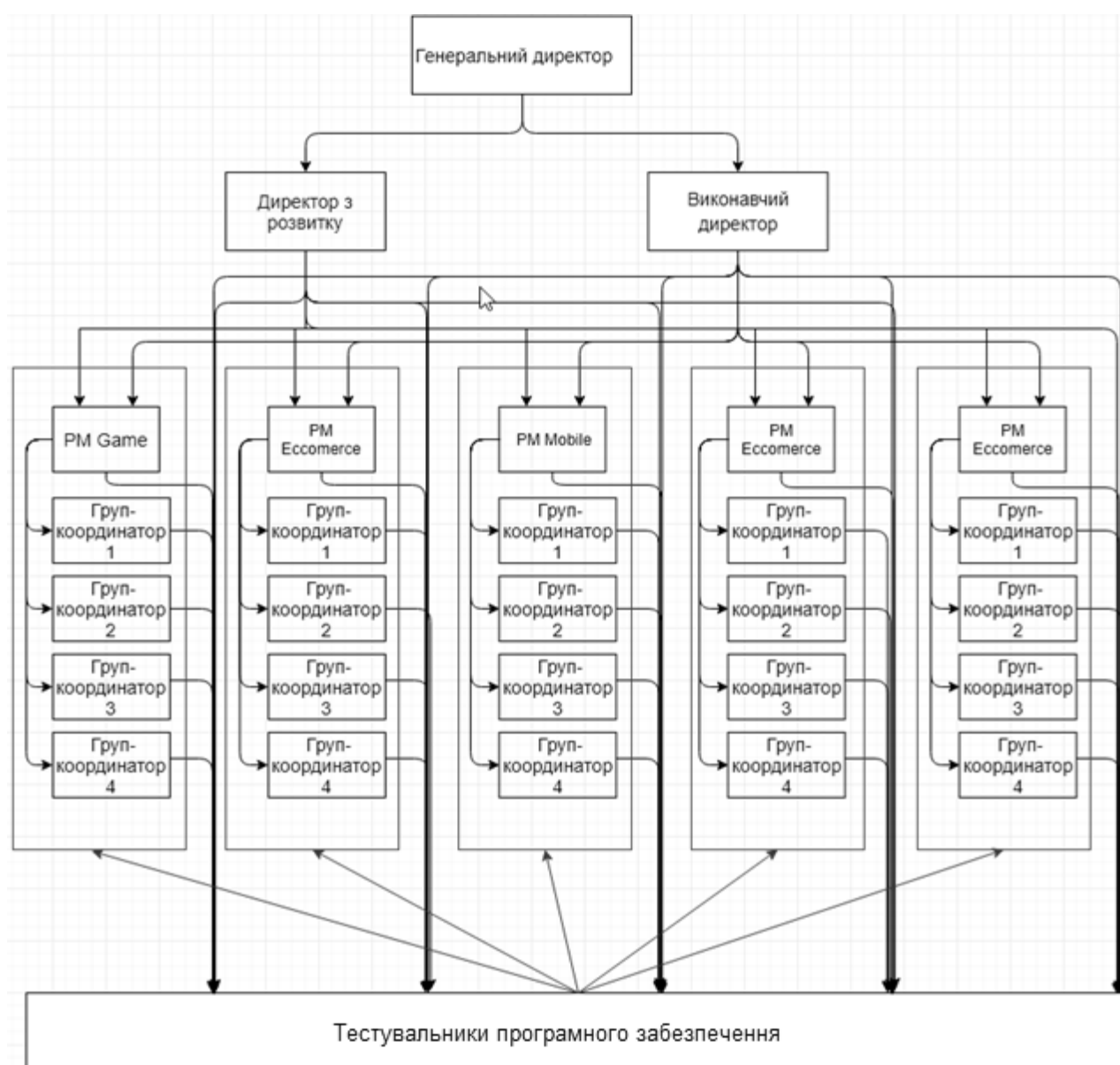


Рисунок 1.3 — Візуалізація інформаційних потоків в компанії ТОВ «Центр мережних технологій ВЕБ100» (QATestLab)

Як уже помітно з візуалізації і опису руху інформації, в компанії є певні обмеження в швидкості поширення інформації, що в свою чергу може призвести до певних проблем у поширенні критично важливих повідомлень і навіть призвести до великих економічних збитків у випадку несвоєчасного отримання найсвіжішої інформації, або ж у випадку викривлення якихось суттєвих деталей під час її транспортування (поширення).

### 1.3 Аналіз стану управління персоналом в компанії і доступних для впровадження HRMS рішень для автоматизації управління персоналом

Доцільно детально розглянути основні особливості функціонування територіально розподіленого підприємства, як об'єкта управління і наявні в компанії системи для управління розподіленими командами.

Під підприємством будемо розуміти самостійно діючий господарюючий суб'єкт, який має організаційну структуру, потребу в ресурсному забезпеченні, один або кілька бізнес-процесів, потребу в збуті готової продукції (наданні послуг). Територіально розподілене підприємство, як об'єкт управління можна розглядати як складну систему, що володіє децентралізованою структурою і об'єднану централізованим стратегічним рівнем управління.

Централізований рівень управління досягається за рахунок загального керуючого і координуючого органу (вище керівництво і центральний підрозділ підприємства), що визначає загальну стратегію функціонування і розвитку підприємства. Укрупнена організаційна структура територіально розподіленого підприємства може бути представлена у вигляді ієрархії підрозділів. Треба відзначити, що управління на рівні підрозділу може бути як частковим (коли частина рішень приймається централізовано, а частина самостійно), так і повністю децентралізованим. Таким чином, організаційна структура являє собою ієрархію підрозділів з різним ступенем децентралізації управління. В рамках

процесного підходу будь-яке підприємство розглядається як бізнес-система, яка являє собою пов'язане безліч бізнес процесів, кінцевою метою яких являється випуск продукції або послуг. М. Хаммер і Дж. Чампі визначають бізнес-процес «як сукупність різних видів діяльності, в рамках якої на «вході» використовуються один або більше видів ресурсів, і в результаті цієї діяльності на «виході» створюється продукт, що представляє цінність для споживача» [11]. В ході своєї діяльності підприємство споживає певні ресурси (організаційні, матеріальні, кадрові та ін.) і виробляє продукцію (або послуги).

В рамках компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab), станом на момент дослідження, не впроваджено єдиного комплексного рішення з автоматизації управління персоналом і розподіленими командами, що призводить до складнощів у керуванні і організації чистих і прозорих процесів управління. У компанії впроваджено декілька автоматизованих інструментів для управління персоналом на різних рівнях менеджерських експертиз. Для зручності дослідження розділимо автоматизовані рішення за рівнем їх використання відносно працівників в компанії.

Навчальний рівень — характеризується низьким рівнем безпосередньої взаємодії з наявними працівниками компанії і є першим роком до працевлаштування. На цьому рівні відбувається навчання курсантів на навчальних курсах компанії з можливістю подальшого працевлаштування. Для навчального рівня характерний тільки один засіб керування з боку компанії — платформа <https://clients.QATestLab.com>, яка є особистим кабінетом курсанта і де він може відслідковувати свої успіхи в навчанні. На рисунку 1.4 відображено карту лекцій на платформі.

Trainee рівень — характеризується початком безпосередньої тісної взаємодії з куратором і груп-координатором на початку входу до компанії. Платформи, які використовуються в цей час — Skype, Gmail, Google Docs, Telegram.

Junior рівень — характеризується безпосередньою тісною взаємодією з куратором, груп-координатором і колегами в рамках проектів. До використовуваних платформ можна віднести Skype, Gmail, Google Docs, Telegram.

Модуль компетенцій (внутрішня розробка компанії) з атестаціями для досягнення необхідного рівня знань і умінь, і корпоративний Confluence.



Рисунок 1.4 — Карта виконання практичних завдань в рамках навчання

Junior рівень — характеризується безпосередньою тісною взаємодією з куратором, груп-координатором і колегами в рамках проектів. До використовуваних платформ можна віднести Skype, Gmail, Google Docs, Telegram. Модуль компетенцій (внутрішня розробка компанії) з атестаціями для досягнення необхідного рівня знань і умінь, і корпоративний Confluence.

Medium рівень — характеризується безпосередньою тісною взаємодією з груп-координатором і колегами в рамках проектів, а також самостійністю у процесі спілкуванні з замовниками і вирішенні не складних проблем. Інструменти — Skype, Gmail, Google Docs, Confluence, Slack(або його аналоги), Telegram.

QA Lead (Груп-координаційний) рівень — характеризується тісною взаємодією з іншими лідерами проектів і груп-координаторами, а також вмінням вирішувати проблеми менш досвідчених колег. Інструментарій управління більшою мірою складається з Skype, Telegram Google Calendar, Gmail і відеовикликів для обговорення важливих рішень і дискусій.

Рівень топ-менеджменту — операційне керування компанією і розподіл задач підлеглим. Найчастіше використовуваними інструментами для управлінських рішень є Skype, Gmail, Trello, Telegram Google Calendar для

відслідковування руху задач в рамках запланованих активностей, особисті зустрічі для всебічного обговорення і затвердження управлінських рішень [60, 70, 71, 72, 73, 74].

Зважаючи на відсутність єдиної автоматизованої системи для менеджменту людських ресурсів і стрімкий розвиток компанії, що виражається у активно розширенні мережі офісів у різних містах, а також на активне зростання кількості персоналу у компанії розглядають в майбутньому перехід на уніфіковану HRMS (Human Resource Management System) систему для оптимального управління розподіленими командами на всіх рівнях моделі керування.

Система управління персоналом (HRMS) або інформаційна система управління персоналом (HRIS) є формою програмного забезпечення управління персоналом (HR), яка об'єднує ряд систем і процесів для забезпечення легкого управління людськими ресурсами, бізнес-процесами та даними. Програмне забезпечення для управління персоналом використовується підприємствами для об'єднання ряду необхідних функцій управління персоналом, таких як зберігання даних про співробітників, управління розрахунками заробітної плати, процесами найму, адміністрування пільг та відстеження записів відвідуваності.

Система управління персоналом гарантує, що повсякденні процеси управління персоналом є керованими і легкодоступними. Вона об'єднує людські ресурси як дисципліну і, зокрема, основні HR-процеси і процеси з областю інформаційних технологій, в стандартизовані процедури і пакети програмного забезпечення для планування ресурсів підприємства (ERP). В цілому, ці системи ERP розглядаються в рамках програмного забезпечення, яке об'єднує інформацію з різних додатків в одну універсальну базу даних.

Інформаційні системи управління персоналом надають можливості для збору, зберігання, аналізу та поширення інформації серед різних зацікавлених сторін — HRMS дозволяють поліпшити традиційні процеси і поліпшити прийняття стратегічних рішень [38, 45]. Хвиля технічного прогресу зробила революцію в усіх сферах життя сьогодні, включаючи HR. Ранні системи були вузькими за обсягом, як правило, орієнтовані на одну задачу, наприклад, на



поліпшення процесу розрахунку заробітної плати або відстеження робочого часу співробітників. Сучасні системи охоплюють великий спектр завдань, пов'язаних з відділами кадрів, включаючи відстеження та підвищення ефективності процесів, управління організаційної ієрархією, відстеження відсутності і щорічної відпустки, спрощення фінансових транзакцій і надання звітів про дані про працівників.

Розглянемо декілька найбільш популярних HRMS систем:

BambooHR — система управління персоналом для малого і середнього бізнесу. Хмарна система пропонує інтегроване відстеження кандидатів (ATS), вбудовані інструменти, електронні підписи, відстеження вихідних даних і управління продуктивністю, з простими звітами і зручним мобільним додатком для співробітників [1].

Conyer PSA надає комплексне рішення для консалтингових, кадрових, рекрутингових і професійних сервісних організацій. Conyer покращує всі аспекти бізнесу: робочий процес, відстеження і контроль. Це просте в реалізації рішення, кероване через Інтернет, підвищує організаційну ефективність, максимізує використання ресурсів, полегшує групове співробітництво, підвищує індивідуальну відповідальність і продуктивність [5].

BizMerlinHR пропонує можливість формувати записи про співробітників, вбудовані інструменти, електронний підпис, відстеження відпусток, управління ефективністю, зворотний зв'язок, ATS і звітність [21].

Система iCIMS допомагає залучати, спілкуватися і відстежувати кандидатів в одному місці, щоб спростити процес набору і найняти кваліфікованих співробітників [19].

Провівши ґрунтовне дослідження в наявних на ринку пропозицій автоматизованих систем керування персоналом і виділеними командами можна виділити декілька основних рис і можливостей, які характеризують усі подібні системи, таких як:

- утримання і розвиток персоналу;
- найм нових працівників;

- вступне навчання нових працівників;
- адміністрування;
- управління заробітною платою;
- кадрове планування;
- управління ефективністю і оцінювання якості роботи;
- звітність та аналітика;
- розгляд скарг.

В результаті дослідження можна стверджувати, що наведені системи є достатньо функціональними управлінськими засобами в стратегічному плані, але з одним недоліком, який потребує доопрацювання, а саме — у всіх систем відсутній засіб, який можна використовувати для швидкого інформування робітників щодо важливої невідкладної інформації, тобто в них відсутні можливості до швидкого поширення термінових даних про зміни. Відсутність подібних модулів може призвести до негативних економічних чи моральних наслідків. Наведені дані свідчать про необхідність впровадження в компанії системи швидких сповіщень працівників.

#### 1.4 Формування вимог до інформаційної системи

У сучасному світі значного розвитку набувають технології швидкісного обміну інформацією і повідомленнями між користувачами за допомогою мережі інтернет. Ця тенденція актуальна не тільки для особистого користування, а і активно вкоріняється в корпоративні процеси компаній, що відкриває нові можливості для тісної взаємодії між співробітниками, які територіально знаходяться дуже далеко один від одного [17, 20]. Зрушення в бік розвитку і покращення процесів і комунікацій у розподілених командах є характерним і для компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab), що в свою

чергу спонукає до розробки проекту з автоматизації найпріоритетніших процесів управління в таких, розподілених, командах [18].

Доцільність і важливість розроблюваного рішення полягає в наступних аспектах:

— зручний, інтуїтивно зрозумілий інтерфейс. легкий у використанні і вивченні графічний інтерфейс є одним із надважливих пунктів для проекту, адже він дозволить користуватися чат-ботом всім учасникам розподілених команд, незалежно від рівня їх технічних компетенцій;

— незалежність від платформи. Чат-бот може бути запущений на будь-якій платформі з мінімальними характеристиками, незалежно від використовуваної операційної системи;

— необмежена кількість клієнтів. платформа Telegram дозволяє створювати сутності, до яких може приєднатися до 200 тис. незалежних користувачів, що в реальних умовах, зважаючи на розміри компанії, є фактично безлімітною кількістю юзерів [41, 64, 65, 66, 67, 68].

В розроблюваній системі заплановано перелічені нижче функціональні можливості, які будуть делегуватися на два рівні доступу користувачів — Admin для доступу до адміністративних опцій і Usual — для доступу до інформаційної системи компанії всіх її діючих працівників.

Ґрунтуючись на наданому Usual рівні доступу система повинна забезпечувати доступність таких опцій для працівників компанії:

- відправка запиту на лікарняний;
- відправка запиту на роботу з дому;
- відправка запиту на покупку нових пристроїв необхідних для виконання обов'язків на проекті;
- відправка сповіщення про проблеми відповідальним особам;
- відправка запиту на відпустку до відповідальних осіб;
- відправка запиту на роботу в вихідні;
- отриманні важливих сповіщень від менеджменту компанії;

- отримання нагадувань про обов'язкову звітність в очікуваний день її відправки;

- отримання сповіщень про збої в роботі важливих інтернет сервісів і постачальників послуг;

- отримання автоматичних сповіщень про появу нових доступних відеокурсів/семінарів від компанії.

Ґрунтуючись на наданому Admin рівні доступу система повинна забезпечувати доступність таких додаткових опцій, окрім перерахованих вище:

- відправка повідомлень про важливі оновлення в компанії;
- налаштування нагадувань про регулярні повторювані події;
- додавання і видалення працівників, а також налаштування їх прав доступу;

- робота з інформацією про працівників на лікарняному;

- робота з інформацією про працівників, які працюють з дому;

- робота з інформацією про запити на відпустку;

- робота з інформацією про запити на роботу в вихідні;

- робота з запитами на нові пристрої;

- опрацювання повідомлень про проблеми.

Для успішного використання розроблюваного рішення також необхідна наявність пристроїв, які будуть відповідати мінімальним вимогам для створюваного чат-бота. Нижче наведено перелік мінімальних вимог для встановлення програми Telegram, як основи, на базі якої і розроблюється чат-бот:

Персональний комп'ютер на базі операційної системи (ОС) Windows:

- ОС — Windows 10 версії 14316.0 або новішої;

- архітектура — x64;

- клавіатура — інтегрована клавіатура;

- миша — інтегрована миша;

- пам'ять — 300 мегабайт;

- обов'язковий доступ до інтернету.

Персональний комп'ютер на базі операційної системи MacOS:

- ОС — OS X версії 10.11 або новішої;
- архітектура — x64;
- клавіатура — інтегрована клавіатура;
- миша — інтегрована миша;
- пам'ять — 300 мегабайт;
- обов'язковий доступ до інтернету.

Мобільний телефон/планшет на базі операційної системи Android:

- ОС — Android версії 4.4 або новішої;
- архітектура — x64 /x86;
- клавіатура — стандартна клавіатура телефону;
- пам'ять — 50-100 мегабайт (залежить від моделі телефону);
- обов'язковий доступ до інтернету.

Мобільний телефон/планшет на базі операційної системи iOS:

- ОС — iOS 9.0 або новіша;
- архітектура — x64 /x86;
- клавіатура — стандартна клавіатура телефону;
- пам'ять — 100-200 мегабайт (залежить від моделі телефону);
- обов'язковий доступ до інтернету.

Мобільний телефон на базі операційної системи Windows Mobile:

- ОС — Windows 8 Mobile;
- архітектура — x86, x64, arm, arm64;
- клавіатура — стандартна клавіатура телефону;
- пам'ять — 100-200 мегабайт (залежить від моделі телефону);
- обов'язковий доступ до інтернету [12, 15, 16].

## 2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Моделі бізнес-процесів на підприємстві

BPM (business process model) — це група принципів, методів і інструментів для проектування, аналізу, виконання та відстеження бізнес-процесів з метою отримання більшої віддачі від них. Це може бути внутрішнє значення, тобто найкраща ефективність, більш висока якість продукції, більш швидкі терміни доставки — або це може бути цінність для клієнта (краще обслуговування, більш швидкий час відповіді і т.п.).

Офіційне визначення BPM, згідно bpm.com, таке — «Управління бізнес-процесами (BPM)» — це концепція, що включає будь-яку комбінацію моделювання, автоматизації, виконання, контролю, вимірювання та оптимізації потоків ділової активності, для підтримки цілей підприємства, що охоплюють системи, співробітників, клієнтів і партнерів в межах і за межами підприємства.»

Простіше кажучи, управління бізнес-процесами — це кроки, які вживає організація для поліпшення своїх бізнес-процесів. Шляхи, якими це досягається:

- аналіз того, що є в даний момент (перегляд поточних процесів; перегляд їх роботи, облік будь-яких проблем);
- ідеї мозкового штурму, щоб зробити процеси краще;
- зміна процесів з метою їх поліпшення;
- моніторинг результатів і внесення додаткових змін при необхідності;

Іншими словами: управління бізнес-процесами — це мистецтво і наука, що дозволяє переконатися, що все в компанії працює так добре, як це можливо, і приносить якомога більшу цінність.

Зазвичай виділяють три типи бізнес-процесів:

Операційні процеси. Це процеси, в яких люди «домагаються мети». На виробництві це може бути процес складання продукту. У логістиці це може бути процес доставки. В управлінні складом це може бути процес одержання товару.

Процеси управління. Це процеси, які контролюють операції і забезпечують правильну роботу. Такі речі, як забезпечення якості, бюджетний нагляд і так далі.

Процеси підтримки або управління. Це процеси, які забезпечують відповідність бізнесу нормам; такі речі, як бухгалтерський облік, технічна підтримка, навчання персоналу.

Як виглядають хороші бізнес-процеси? За словами експерта BPM Марлона Дюма, мета всіх зусиль BPM — максимізувати цінність [16]. Зокрема, він визначив 7 типів значень, які BPM може допомогти забезпечити:

- прозорість;
- якість;
- адаптивність;
- інтеграційність;
- ефективність;
- дотримання стандартів;
- обмін інформацією.

Перша і найважливіша з цих цінностей — прозорість. Прозорість в бізнес-процесах допомагає:

- отримати чітке уявлення про те, як насправді виконуються завдання;
- отримати уявлення про те, як насправді працюють всі аспекти бізнесу;
- прибрати обмеження між відділами та сприяти співробітництву;
- краще виміряти вплив змін на процеси;
- вносити активні (а не реактивні) зміни в процеси;
- оптимізувати організаційні процеси;
- усунути витрати (часу, грошей або ресурсів).

Прозорість в бізнес-процесах допомагає якісно і зважено приймати рішення, вимірювати зміни і більш ефективно управляти ризиками.

Отже, перш за все, хороший бізнес-процес — прозорий. Це означає, що легко побачити, як саме він працює, як він впливає на інші процеси і чи працює він правильно.

Для глибшого розуміння моделі бізнес процесів в компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab) доцільно побудувати ER (Entity Relationship) діаграму, яка відобразить сутності в компанії, а також зв'язки між цими сутностями [3, 4].

Модель відносин між сутностями (модель ER) описує структуру даних за допомогою діаграми, яка відома як діаграма відносин між особами(сутностями) (ER Diagram). Модель ER — це дизайн або план бази даних, який згодом може бути реалізований як база даних. Основними компонентами моделі ER є сукупність сутностей та набір відносин між ними. Діаграма ER показує взаємозв'язок між сукупностями сутностей. Набір суб'єктів — це група подібних сутностей, і ці сутності можуть мати атрибути. З точки зору СУБД, сутність — це таблиця або атрибут таблиці в базі даних, тому, показуючи взаємозв'язок між таблицями та їх атрибутами, діаграма ER показує повну логічну структуру бази даних [4, 23]. На рисунку 2.1 відображено приклад ER діаграми взаємодії сутностей з задачами в рамках компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab). Пояснення деяких умовних позначень на діаграмі:

- прямокутники — сутності (посада чи роль) в компанії;
- ромби — поставлені задачі чи проблеми над якими ведеться робота;
- овали — атрибути (характеристики) окремих сутностей;
- трикутники — проблема чи задача з додатковим обговоренням;
- лінії — взаємозв'язки наведених сутностей.



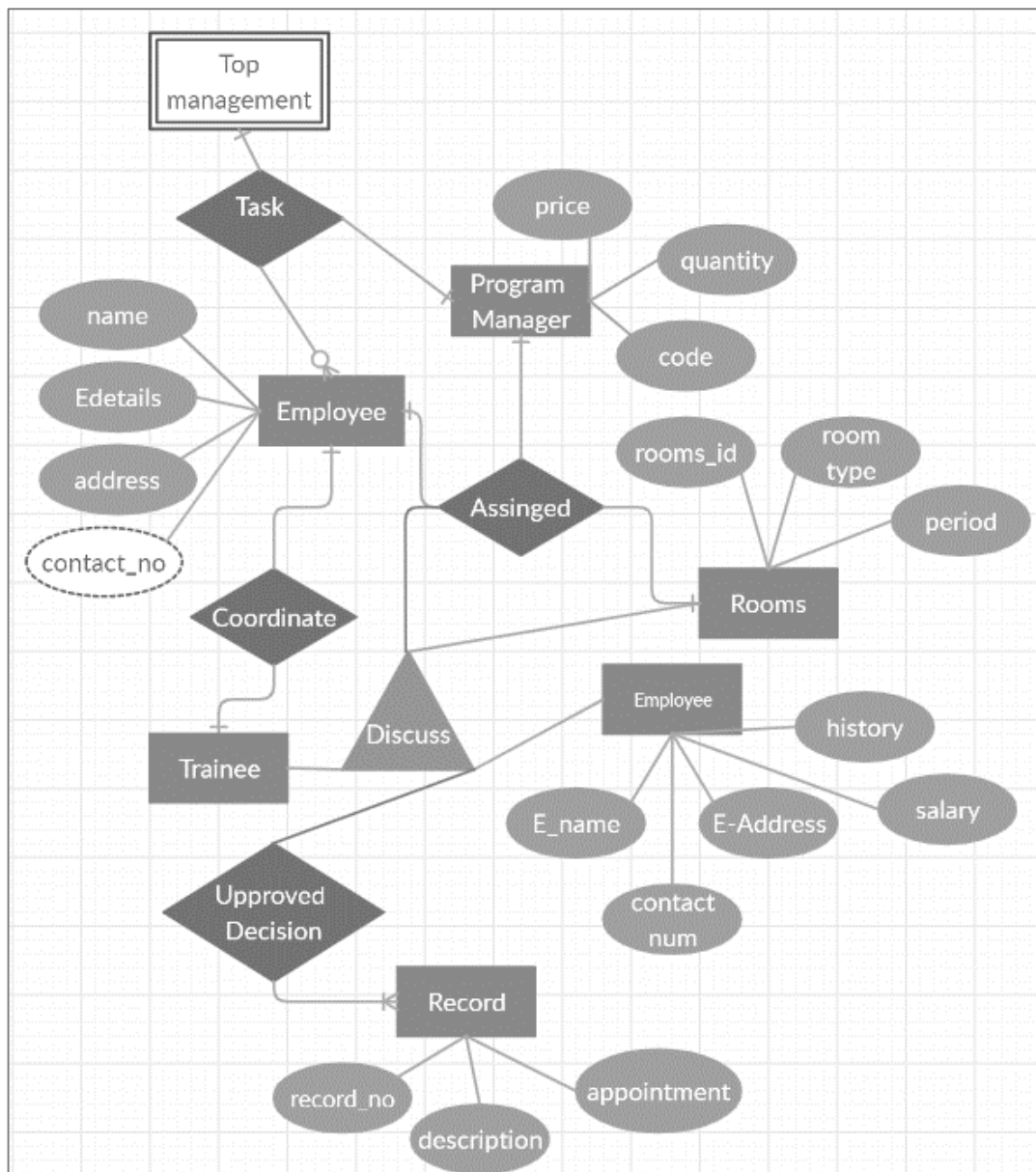


Рисунок 2.1 — ER діаграма сутностей ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab) для деяких задач

При детальному ознайомленні з наведеною вище спрощеною ER моделлю можна помітити її певні особливості, такі як:

- велика кількість атрибутів у сутностей;
- велика кількість залежностей у відносинах між сутностями;
- складна організаційна структура в компанії;
- повільне протікання процесів внаслідок структурних залежностей [6];

## 2.2 Архітектура автоматизованої інформаційної системи та технології вирішення поставлених завдань

Правильно обрана і втілена архітектура є серцем розвитку чат-ботів. Виходячи із зручності використання та контексту ділових операцій, архітектура, що обрана для побудови чат-боту, адаптується до конкретних ділових потреб компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab). Отже, виходячи з вимог компанії, як клієнта, потрібно створювати і змінювати елементи; але без змін основного потоку зв'язків [8, 9];

Нижче наведені ключові компоненти архітектури чат-боту який розробляється:

- оточення — тут відбувається основне тлумачення контексту;
- система запитань і відповідей — система запитань інтерпретує питання та відповідає запрограмованими відповідями з бази знань;
- модулі / компоненти — користувач посилається на повідомлення, яке він розмістив раніше, бот повинен правильно інтерпретувати це та вносити зміни до системи. Плагін управління діалогом дозволяє це зробити;
- вузловий сервер / сервер трафіку — сервер, який обробляє запити про трафік від користувачів і спрямовує їх на відповідні компоненти. Сервер трафіку також спрямовує відповідь від внутрішніх компонентів назад до інтерфейсу;
- інтерфейс користувача — функціональна частина з якою безпосередньо взаємодіє користувач в процесі роботи з чат-ботом [56, 57];

Фронтальними системами можуть бути будь-які клієнтські платформи. Вони можуть бути реальними інтерфейсами чат-ботів, які знаходяться на різних платформах, таких як:

- Facebook;
- Slack;
- Skype for Business;
- Microsoft Teams;

— Telegram [76];

Рисунок 2.2 відображає втілювану архітектуру розроблюваного чат-бота

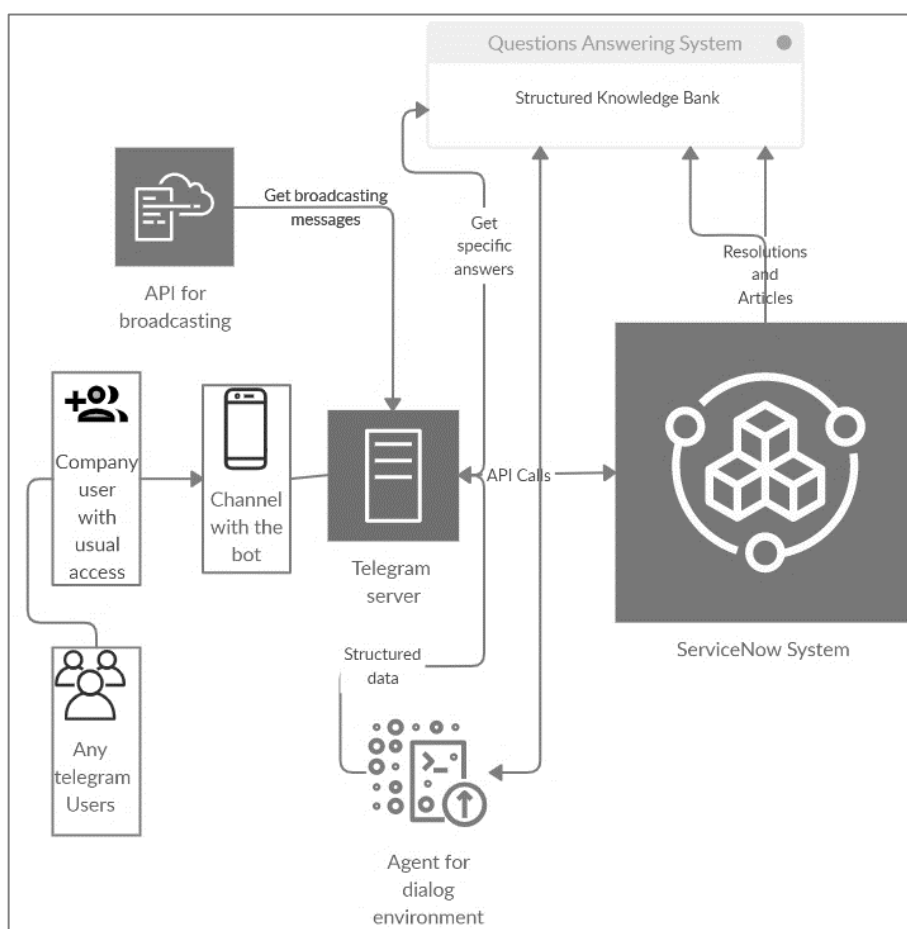


Рисунок 2.2 — Архітектура розроблюваного чат-бота.

(API) — це набір процедур, протоколів та інструментів для створення програмних додатків. По суті, API визначається, як спосіб взаємодії компонентів програми. Крім того, API використовуються при програмуванні компонентів графічного інтерфейсу користувача (GUI).

Server — центральний комп'ютер, з якого інші комп'ютери отримують інформацію.

Broadcasting — діяльність або функція по розсилці програм чи інформації на автоматизованими засобами.

### 2.3 Функціональна структура завдання

Функціональна структура є однією з найбільш поширених організаційних структур. В рамках цієї структури організація групує співробітників відповідно до спеціалізованих наборів ролей або завдань. У той час як функціональні структури добре працюють в стабільному середовищі, де бізнес-стратегії менш схильні до змін або динамічності, рівень бюрократії заважає організаціям швидко реагувати на зміни на ринку [13, 22].

Коли компанії використовують функціональну організаційну структуру, люди зі схожими знаннями та навичками групуються разом. Це дозволяє співробітникам стати фахівцями в своїй галузі. Загалом, це звичайно, вимагає системи управління ефективністю, яка дозволяє просувати, розвивати і бачити індивідуальні навички в межах їх функціональної області. Спеціалізація, яку відточують функціональні структури, допомагає забезпечити глибокі знання і розвиток навичок серед співробітників, що може допомогти в досягненні функціональних цілей вашої компанії.

Спеціалізація веде до підвищення ефективності і підвищенню рівня продуктивності. Завдяки своєму досвіду працівники зі спеціалізованими навичками можуть виконувати завдання швидко, ефективно і з більшою впевненістю, що знижує ймовірність виникнення помилок, пов'язаних з роботою. Крім того, чіткий характер шляху кар'єри в функціональному підрозділі дозволяє співробітникам мати високу мотивацію для просування по кар'єрних сходах в міру просування вгору по ієрархії.

Основною метою функціональних структур в організаціях є об'єднання всіх людських та інформаційних ресурсів для досягнення цілей організації. Це може бути ідеальною складовою малих підприємств, які зосереджені на одному продукті чи послугі, тому що можна максимізувати продуктивність, заохочуючи

співпрацю між різними підрозділами на різних рівнях управління за допомогою нагляду та координації [14, 22, 24];

Комунікації в організаціях з функціональними організаційними структурами можуть бути жорсткими через стандартизацію способів роботи і високого ступеня формалізації. Це може додатково зробити процес прийняття рішень повільним і негнучким [26]; Оскільки вони більш бюрократичні, функціональні підрозділи часто не підзвітні один одному, і може виникнути погана горизонтальна координація всередині відділів. Відсутність інновацій та обмежений погляд на цілі організації, а також надмірна концентрація можуть вплинути на мотивацію співробітників. У міру зростання компанії і створення більш функціональних підрозділів рівень автономії всередині підрозділів може зрости, що ускладнить ефективну координацію всіх підрозділів.

Хоча функціональні блоки часто працюють з високим рівнем ефективності, їх рівень взаємодії один з одним іноді ставиться під загрозу. Такі групи можуть зазнавати труднощів у роботі один з одним, оскільки вони можуть бути територіально розподіленими і не бажати співпрацювати [25]; Виникнення конфліктів між підрозділами може призвести до затримок, зниження зобов'язань через конкуренцію інтересів і втрати часу, через що проекти відстають від графіка. В кінцевому підсумку це може призвести до зниження рівня виробництва.

На рисунку 2.3 відображено загальну функціональну структуру розподілу наявних функцій відносно наданих ролей в чат-боті.

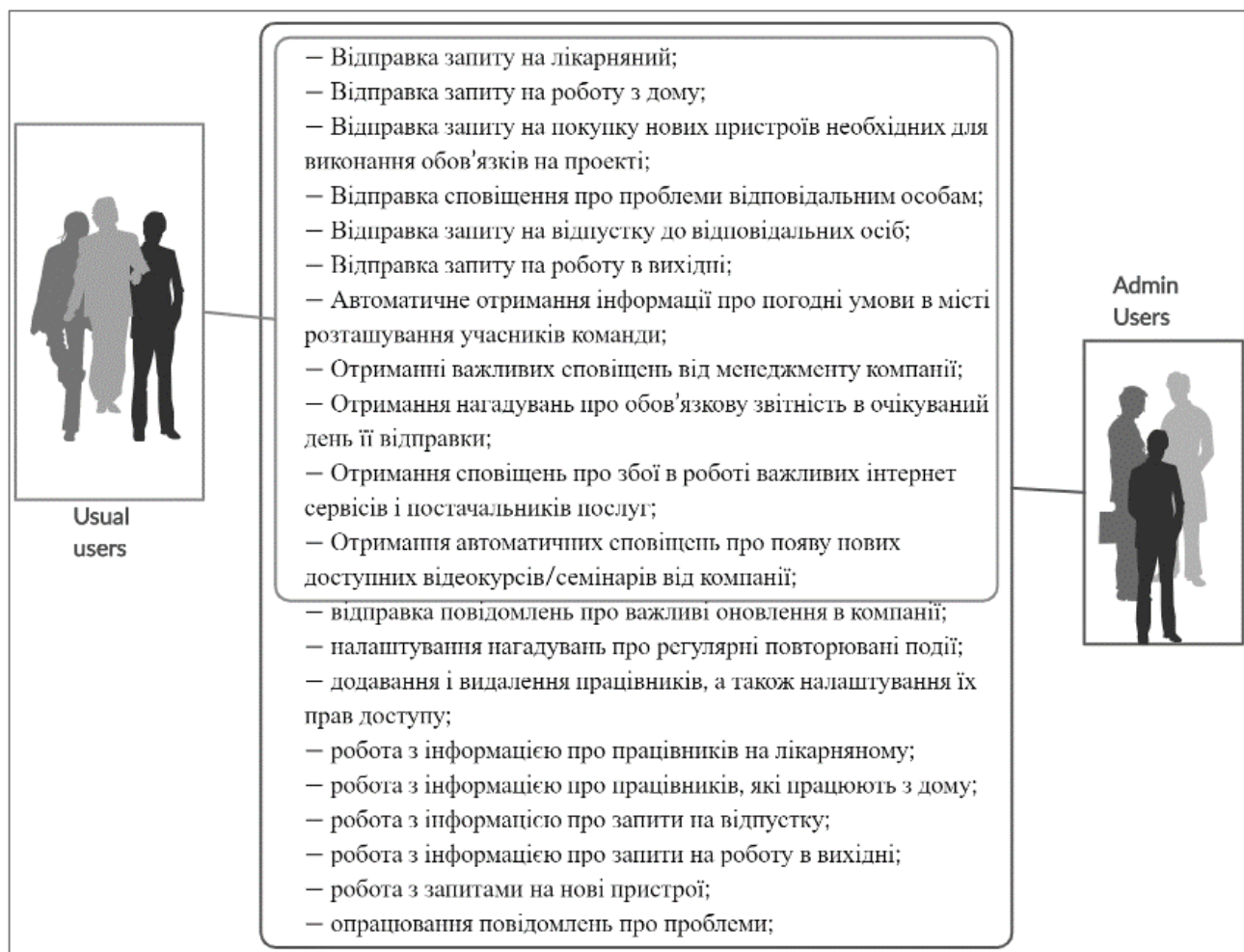


Рисунок 2.3 — Функціональна структура чат-бота відносно наданих ролей.

Функціональна організаційна структура допомагає організаціям вести свій бізнес і отримувати прибуток, групуючи співробітників на основі їх навичок та досвіду. Така структура забезпечує чітку ієрархію і зводить до мінімуму плутанину серед співробітників, які можуть бути не впевнені, кому вони повинні звітувати [22, 26]; Функціональна організаційна структура є достатньо ефективним способом ведення бізнесу, але і в такої структури можуть бути недоліки, оскільки вона може призвести до відсутності зв'язку між відділами і співробітниками, які не мають достатнього ентузіазму у виконанні своїх посадових обов'язків.

## 2.4 Характеристика системного і прикладного програмного забезпечення та підсистем для забезпечення функціональної частини.

Для забезпечення ефективного та детального робочого процесу ТОВ «Центр мережових технологій ВЕБ100» (QATestLab) використовує широкий діапазон різних інструментів і технологій, які підвищують продуктивність команди QA, спрощують управління діями та допомагають виявляти системні проблеми. Виходячи з специфіки особливостей продукту, вимог замовника та цілей проекту, обираються належні інструменти для оптимізації процесів та забезпечення безперебійного робочого процесу.

Використовувані на підприємстві інструменти:

### 1. Баг-трекери:

Jira, Bugzilla, Mantis, Trac, Redmine, PR-Tracker, Gemini, JTrac, Eventum;

### 2. Інструменти проектів:

Testlink, Confluence, Microsoft Project, HP / Mercury Quality, Center (Test Director);

Інструменти для різних типів тестування:

### 1. Автоматизоване тестування:

QTP, Selenium, Appium, Robotium, Serenity, Robot, Ranorex Studio, TestComplete, Watir;

### 2. Тестування баз даних:

Navicat, SQL Server;

### 3. Тестування API:

SoapUI, PostMan in GChrome;

### 4. Тестування сумісності:

Spoonium;

### 5. Тестування з'єднання:

F-Secure SSH Client, Google Voice;

### 6. Тестування дизайну:

Snagit, Photoshop;

7. Тестування посилань:

Xenu's Link Sleuth, A1 Website Analyzer, Qualidator;

8. Для мобільного тестування:

TestFlight, iFunBox, Xcode;

9. Тестування захищеності:

Tamper Data, HackBar, Add N Edit Cookies, Modify Headers, ShowIP;

10. Додаткові програмні рішення:

VPN Client, Proxifier, TeamViewer 8, PuTTY, Fiddler Web Debugger;

11. Віртуальні інструменти:

VMware Workstation, VirtualBox;

12. Тестування навантаження:

LoadRunner, JMeter, LoadUI, SoapUI, Wapt;

13. Засоби комунікацій:

Telegram, Skype, Slack, Teams;

Операційні системи, які використовуються при тестуванні:

1. Десктоп рішення:

Linux Family, Mac Family, Windows Family, BSD Family OS, Palm OS, Multi-platform;

2. Мобільні рішення:

iOS Family, Android, Windows Mobile;

Для якісного і ефективного виконання тестувальниками своїх обов'язків на підприємстві впроваджена типова мінімальна конфігурація комп'ютера, яка має наступні характеристики:

— материнська плата Asus H61M-K — материнська плата, яка підтримує процесори Intel Core i7, i5 та i3, що включають в себе інтегровані контролери пам'яті та шини PCI Express;

— процесор Intel Core i5-7400 який забезпечує високу продуктивність настільних персональних комп'ютерів, низьке енергоспоживання та широкі



можливості багатозадачності при повсякденній роботі. Intel Core i5-7400 має технології Turbo Boost і Hyper Threading;

— жорсткий диск Western Digital Blue 2 Tb 7200rpm 64MB WD10EZEX 3.5 SATA

— відеокарта NVIDIA GeForce GTX 1060 — графічний процесор GeForce GTX 1060 являється одним із самих потужних в класі середнього рівня на сьогодні, нова мікроархітектура Pascal надає йому неперевершену швидкість, а підтримка сучасних технологій максимально розширює його функціональність;

— блок живлення Xilence Performance A+ 530W — має форм-фактор ATX, також має систему корекції потужності, систему охолодження і складається з 120 мм вентилятора і кількох рівнів захисту;

— оперативна пам'ять має дві ланки Kingston DDR4 2666 8192MB PC4-17000 нового покоління, володіє чудовими характеристиками, високим ступенем продуктивності і малим ступенем енергоспоживання, цей модуль пам'яті був створений спеціально для персонального комп'ютера;

— корпус Zalman Z3 Plus Black — ATX корпус від Zalman, який використовує дизайн із застосуванням рішень для кращого охолодження;

— монітор 27" Samsung S27F358F — монітори Samsung з матрицею, виготовленою за технологією IPS, відтворюють по-справжньому живе зображення без спотворень, гарантуючи комфортний для очей перегляд навіть протягом тривалого часу;

— клавіатура бездротова Logitech K230 — великі клавіші забезпечують відмінну зворотну віддачу і прекрасні тактильні відчуття під час роботи;

— миша Logitech Corded M500 — забезпечує роботу на нерівних і рельєфних поверхнях.

## 3 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Реалізація, налаштування, загальний алгоритм роботи чат-бота

Дослідивши і проаналізувавши всі ключові причини створення бота і його очікуваний функціонал, є доцільним продумати організацію процесу розробки, а також, те які інструменти потрібно буде використати в процесі створення автоматизованої інформаційної системи. Далі розглянемо ключові моменти реалізації чат-бота. Інструментами впровадження будуть:

— pyTelegramBotAPI — бібліотека для створення ботів на мові програмування Python для Telegram;

— Manybot — бот для швидкого створення ботів в Telegram;

— IFTTT — бот для автоматизації отримання деяких автоматичних сповіщень від сервісів;

— «QuickInformer» — створений канал в телеграм, простір для організації отримання важливих сповіщень від чат-бота;

— «MakeMyDayBot» — розроблюваний чат-бот, для організації управління і взаємодії з командою.

Доцільність використання саме такого набору інструментів полягає в тому, що за короткий проміжок часу ми зможемо отримати MVP продукт (англ. Minimum viable product — MVP) в стислі терміни, що дозволить протестувати наскільки затребуваними в реальних умовах є розроблювані функції і які з них потребують основної уваги при розробці (тобто які з них є найпопулярнішими у використанні) [31]. MVP — «продукт з мінімальним функціоналом», який можна дати користувачам для використання. Використовується для тестування ідей у розробці програм з мінімальними затратами ресурсів. Зазвичай реалізується шляхом вдалого планування релізів, коли у перший реліз входять лише основні

функції (або одна найважливіша), що дає можливість користувачам уже почати працювати, не очікуючи остаточної версії.» [33].

Незалежно від того який інструмент буде використовуватися для розробки і налаштування чат-ботів, необхідно провести мінімальні налаштування системи для початку розробки. Потрібно встановити програму Telegram для ПК або ж відкрити сторінку веб-клієнта Telegram в браузері.

Наступний кроком після відкриття програми чи веб-сайту є логін вже існуючого чи створення нового користувача, для продовження роботи з основними функціями програми і для здійснення всіх необхідних подальших налаштувань для роботи з чат-ботами.

Після виконання початкових налаштувань необхідно створити бота, який і буде основним в роботі. Для створення необхідно виконати наступні кроки:

- відкрити додаток Telegram;
- знайти «@botfather» і запустити чат;
- надіслати «/newbot» команду і дотримуватись інструкцій.

Після виконання початкових кроків, отримуємо:

- власний токен;
- URL-адресу телеграм арі — «api.telegram.org/ <токен>»;
- посилання на документацію.

Для налаштування опції сповіщень про проблеми в роботі сайтів від Downtdetector скористаємося бібліотекою pyTelegramBotAPI, як фреймворком. На рисунку 3.1 відображено інтерфейс IDE для роботи з кодом.

```

main.py
129 credentials[0] + "\nYour notification status
is:\nE-mail: "
status = site_alert.execute_fetch_all("SELECT
mailnotification FROM Users WHERE mail = ?",
(credentials[0]))[0]
mymsg += status[0] + "\nTelegram: "
status = \
site_alert.execute_fetch_all("SELECT
telegramnotification FROM Users WHERE mail = ?",
(credentials[0]))[0]
mymsg += status[0] + "\nYou are registered to:"
for site in site_alert.execute_fetch_all("SELECT
name FROM Registered WHERE mail = ?", (credentials[0]
)):
mymsg = mymsg + "\n" + str(i) + " " + site[0]
i += 1
th.send_message(m_chat_id, mymsg)

```

```

Python 3.7.4 (default, Jul 9 2019, 00:06
:43)
[GCC 6.3.0 20170516] on linux
>
Repl.it: Updating package configuration
--> /usr/local/bin/python3 -m poetry init --no-interaction
This command will guide you through creating your pyproject.toml
config.
--> /usr/local/bin/python3 -m poetry add telebot
Using version ^0.0.3 for telebot
Updating dependencies
Resolving dependencies...
Writing lock file

```

Рисунок 3.1 — Інтерфейс IDE для роботи з кодом Python

Для отримання інформації про збої в роботі інтернет сервісів використаємо відкритий API від Twitter. Отримувати записи будемо від акаунту Downtdetector, сервісу, який збирає статистичні дані про всі важливі інтернет сервіси. Запит до Twitter API відображено на рисунку 3.2. Повний код запиту і JSON відповідь на запит відображені в додатку Г.

```

1 # Request Tweets from last 7 days
2
3 # Search query: from:Downtdetector OR #downtdetector
4
5 twurl "/1.1/search/tweets.json?
  q=from%3ADowntdetector%20OR%20%23downtdetector"

```

Рисунок 3.2 — Запит до Twitter для отримання інформації від Downtdetector

Для коректної роботи бота і бібліотеки pyTelegramBotAPI необхідно імпортувати додаткові бібліотеки і налаштувати кодування документу bot.py. Додаткові імпортовані бібліотеки відображено на рисунку 3.3.

```

# -*- coding: utf-8 -*-

import time
import eventlet
import requests
import logging
import telebot
from time import sleep

```

Рисунок 3.3 — Бібліотеки імпортовані для роботи з Python

Іноді, в результаті проблем зі з'єднанням чи з віддаленими серверами в центрах обробки даних, може виникати ситуація, коли розроблений бот не може отримати відповідь на запит з необхідною інформацією. Для того, щоб захиститися від ситуації, коли пропускаються важливі сповіщення необхідно

налаштувати таймаут, після якого бот зупиняється до наступного часу синхронізації. Налаштування таймаута відображено на рисунку 3.4.

```
#def get_data():
    timeout = eventlet.Timeout(10)
    try:
        feed = requests.get(URL_TW)
        return feed.json()
    except eventlet.timeout.Timeout:
        logging.warning('Got Timeout while retrieving TW JSON
data. Cancelling...')
```

Рисунок 3.4 — Налаштування таймауту для бота

Далі розглянемо покроковий алгоритм парсингу даних про збої роботи інтернет сервісів.

1. Відкрити файл, в якому зберігається останній відомий номер верхньої публікації.
2. Якщо метод `get_data()` повернув об'єкт із записами, починаємо проходити його з другого елемента, оскільки перший — це невідоме випадкове число.
3. Якщо номер менше або дорівнює останньому відомому, ми завершуємо сканування.
4. Перевіряємо наявність закріпленого запису. Якщо такий є, то ми передаємо у функцію передачі повідомлень усі записи, крім закріпленого. В іншому випадку ми просто передаємо все.
5. Вилучаємо ідентифікатор з кожного перевіреного запису, підставляємо його біля `BASE_POST_URL` і отримуємо повний ідентифікатор запису.
6. Відправляємо записи на канал.
7. Як тільки раунд завершився, беремо номер першого (другого, якщо перший закріплений) запису і записуємо у файл поверх старого значення.
8. Бот зупиняється до наступного запуску.

Після написання коду для роботи парсера необхідно також розробити логіку запуску всього процесу і ініціювати логер, який буде створювати лог-файл про всі зміни бота. Код запуску логіки парсера відображено на рисунку 3.5

```

if __name__ == '__main__':
    # Очищуємо технічну інформацію переслану бібліотекою requests
    logging.getLogger('requests').setLevel(logging.CRITICAL)
    # Налаштовуємо логгер
    logging.basicConfig(format='[%asctime)s] %(filename)s:%(lineno)d %
                        (levelname)s - %(message)s', level=logging.INFO,
                        filename='bot_log.log', datefmt='%d.%m.%Y
                        %H:%M:%S')
    check_new_posts_tw()
    # Пауза в 4 хвилини перед повторною перевіркою
    logging.info('[App] Script went to sleep.')
    time.sleep(60 * 4)
else:
    check_new_posts_tw()
    logging.info('[App] Script exited.\n')

```

Рисунок 3.5 — Код для запуску логера і парсера записів

На рисунку 3.6 відображається сповіщення від сервісу Downtdetector про проблеми в роботі сервісу Mixer, яке було надіслано до створеного і налаштованого каналу за допомогою розробленого чат-боту.

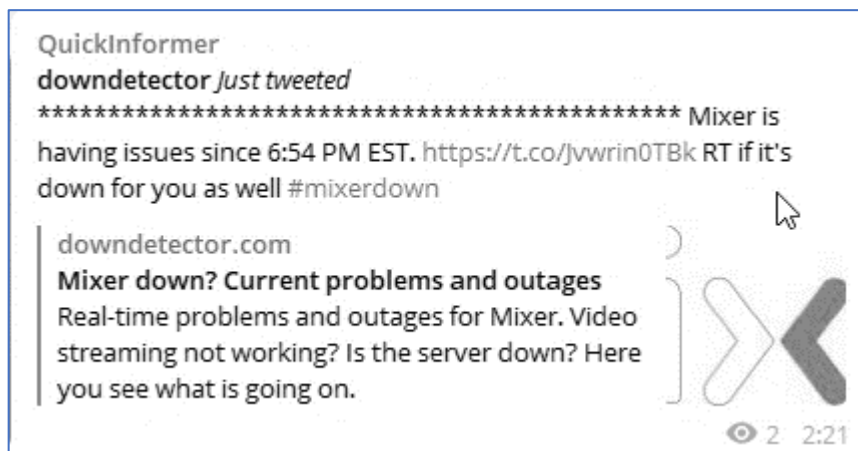


Рисунок 3.6 Сповідження в Telegram про проблеми в роботі одного з інтернет сервісів (Mixer)

Повний програмний код для опції сповіщень, а також код запиту і відповіді від Twitter API наведено в додатках Б і Г.

Для налаштування функції регулярних нагадувань про важливі події використаємо апплет «Automatically post a daily reminder to a Telegram chat» в інструменті IFTTT [32, 34, 35].

Крок 1. Відкрити сторінку <https://ifttt.com/> і створити аккаунт.

Крок 2. Ввести «Telegram» в поле пошуку на сторінці «Explore».

Крок 3. Знайти в списку апплет «Automatically post a daily reminder to a Telegram chat» і натиснути на нього.

Крок 4. Налаштування часу і днів відправки нагадувань в апплеті «Automatically post a daily reminder to a Telegram chat» відображено на рисунку 3.7.

The screenshot shows the configuration interface for the IFTTT applet "Automatically post a daily reminder to a Telegram chat". The interface is divided into several sections:

- Title and Platform:** The applet title is "Automatically post a daily reminder to a Telegram chat" with a Telegram icon.
- Description:** "Pick the days of the week and time to send encouraging messages like 'Lunch break, everyone!' or 'Now back to work, you maggots!' to your coworkers in a Telegram chat."
- Notifications:** A toggle switch for "Get notifications when this connection is active" is currently turned on.
- Frequency:** A dropdown menu is set to "Every day of the week at". Below it, the "Time of day" is set to "12 AM" and "00 Minutes".
- Days of the week:** A list of days from Monday to Sunday, each with an unchecked checkbox.
- Send message:** A checked toggle switch and the text "This action will send a text message to a Telegram chat."
- Target chat:** A dropdown menu set to "Private chat with @IFTTT". Below it, a note says "Use the @IFTTT bot on Telegram to connect new groups or channels."
- Message text:** An empty text input field with a "C" icon for copy. Below it, a note says "Some HTML tags are supported: b, i, a, br, pre, code."
- Save:** A large black button with the word "Save" in white.

Рисунок 3.7 — Налаштування часу і дати в апплеті «Automatically post a daily reminder to a Telegram chat»

Крок 5. Налаштувати канал до якого будуть надсилатися повідомлення.

Крок 6. Налаштувати текст повідомлення в апплеті «Automatically post a daily reminder to a Telegram chat». Налаштування тексту відображено на рисунку 3.8.

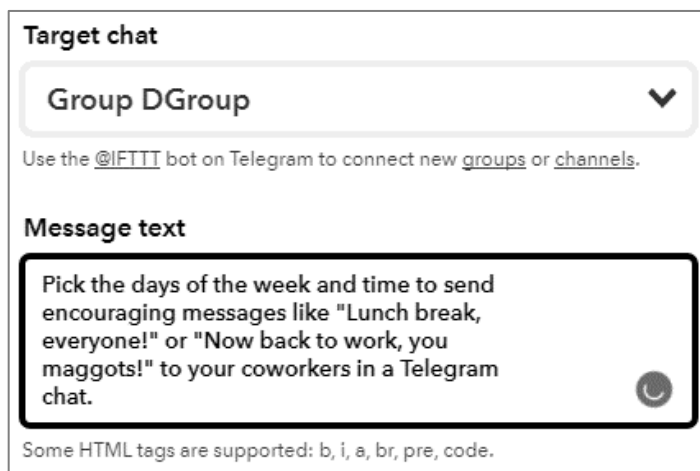


Рисунок 3.8 — Налаштування тексту в апплеті «Automatically post a daily reminder to a Telegram chat»

Крок 7. Зберегти виконані налаштування.

Результати роботи інструменту для розсилки нагадувань про важливі події чи про щоденні обов'язкові задачі відображено на рисунку 3.9.

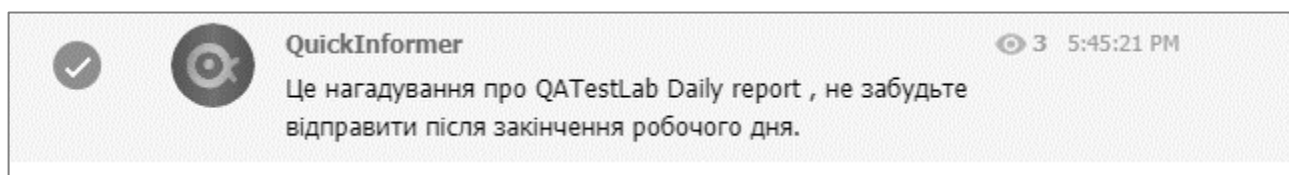


Рисунок 3.9 — Приклад налаштованого сповіщення

Розглянемо додавання налаштувань до розроблюваного бота за допомогою Manybot [36]. Для запуску роботи над чат-ботом необхідно відправити Manybot отриманий токен, що відкриє доступ до безпосереднього налаштування функціональної частини бота. Отримання доступу до всіх доступних налаштувань Manybot відображено на рисунку 3.10.



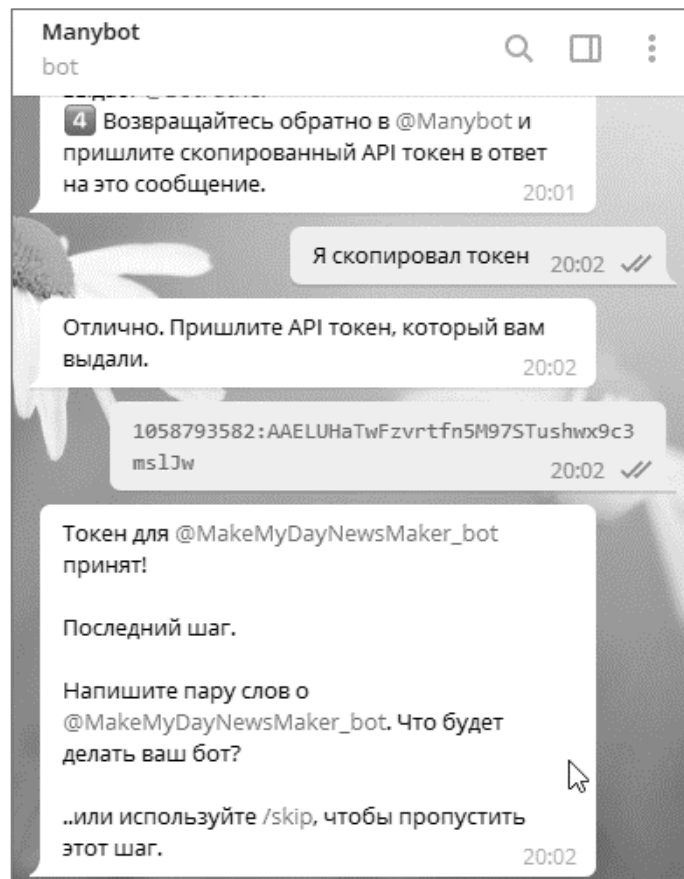


Рисунок 3.10 — Отримання доступу до налаштувань функціоналу в Manybot

Після надання всіх необхідних дозволів можна продовжувати налаштування функціональної частини бота, або ж спочатку надати всі необхідні привілеї іншим адміністраторам, які теж відповідають за створення і налаштування бота [37, 40]. Для того, щоб надати привілеї адміністратора необхідно виконати наступні кроки:

- виконати команду «/admins» в Manybot;
- обрати необхідного бота, для якого проводитимуться налаштування привілеїв;
- якщо в бота ще немає обраних адміністраторів, то відобразиться повідомлення про це;
- обрати опцію «додати нового адміністратора»;
- поширити отримане посилання працівникам, які будуть мати привілеї адміністраторів (посилання буде дійсне лише 24 години);
- користувач, який отримає посилання стане адміністратором бота, після переходу по ньому.

Наступним кроком після розподілу привілеїв є безпосереднє налаштування функціональних можливостей, тобто команд, які будуть доступні користувачам.

Налаштування користувацьких команд розглянемо на прикладі налаштування команди на запит на лікарняний.

Кроки, які необхідно виконати для налаштування команди на отримання лікарняного:

- перейти до створюваного чат-бота;
- викликати контекстне меню;
- обрати пункт користувацьких команд;
- обрати пункт створити нову;
- створити нову команду «`sick_leave`» (дозволена тільки латинка і символи нижнього підкреслювання);
- додати запит до користувача, який буде показано після виклику команди;
- додати варіанти відповідей (так чи ні, або ж будь-які інші);
- зберегти всі зміни.

Детальний приклад створення користувацької команди на отримання працівником лікарняного відображено на рисунку 3.11.

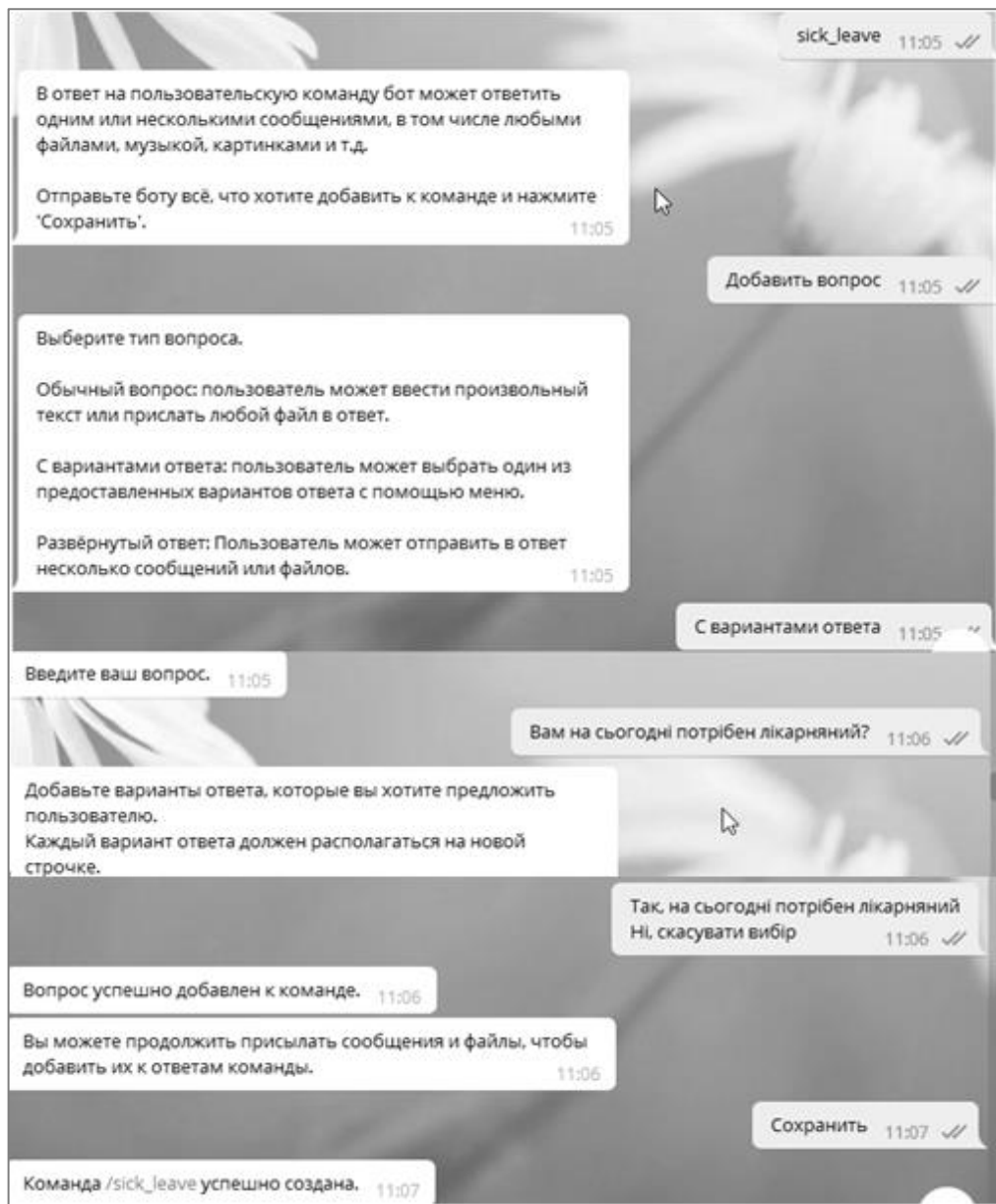


Рисунок 3.11 — Процес створення команди на отримання лікарняного

Додатково розглянемо додавання опції взяття лікарняного до головного меню розроблюваного бота:

- обрати пункт «Налаштувати головне меню»;
- обрати пункт «Додати пункт меню»;
- обрати потрібний пункт і нову назву для нього;
- перевірити, що новий пункт є в меню.

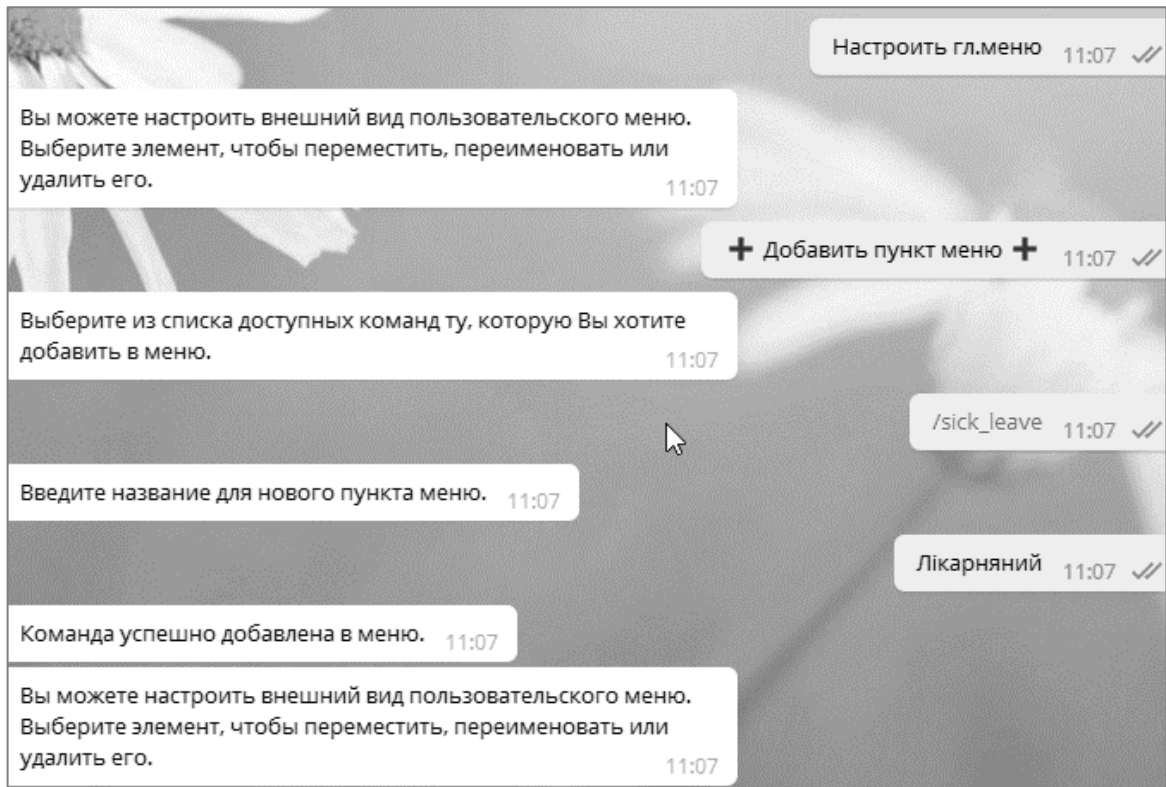


Рисунок 3.12 — Процесс наладки главного меню

Для наладки решетки пользовательских команд воспользуемся уже описанными выше шагами. Пользовательские команды необходимо создать для следующих опций:

- отправка запроса на больничный;
- отправка запроса на работу из дома;
- отправка запроса на покупку новых устройств необходимых для выполнения обязательств на проекте;
- отправка сообщения о проблемах ответственным лицам;
- отправка запроса на отпуск до ответственных лиц;
- отправка запроса на работу в выходные.

Главное меню бота после настройки всех необходимых опций отображено на рисунке 3.13. Изображение слева отображает меню доступное работнику с правами Usual, а изображению справа отображает меню доступное пользователям с правами Admin.

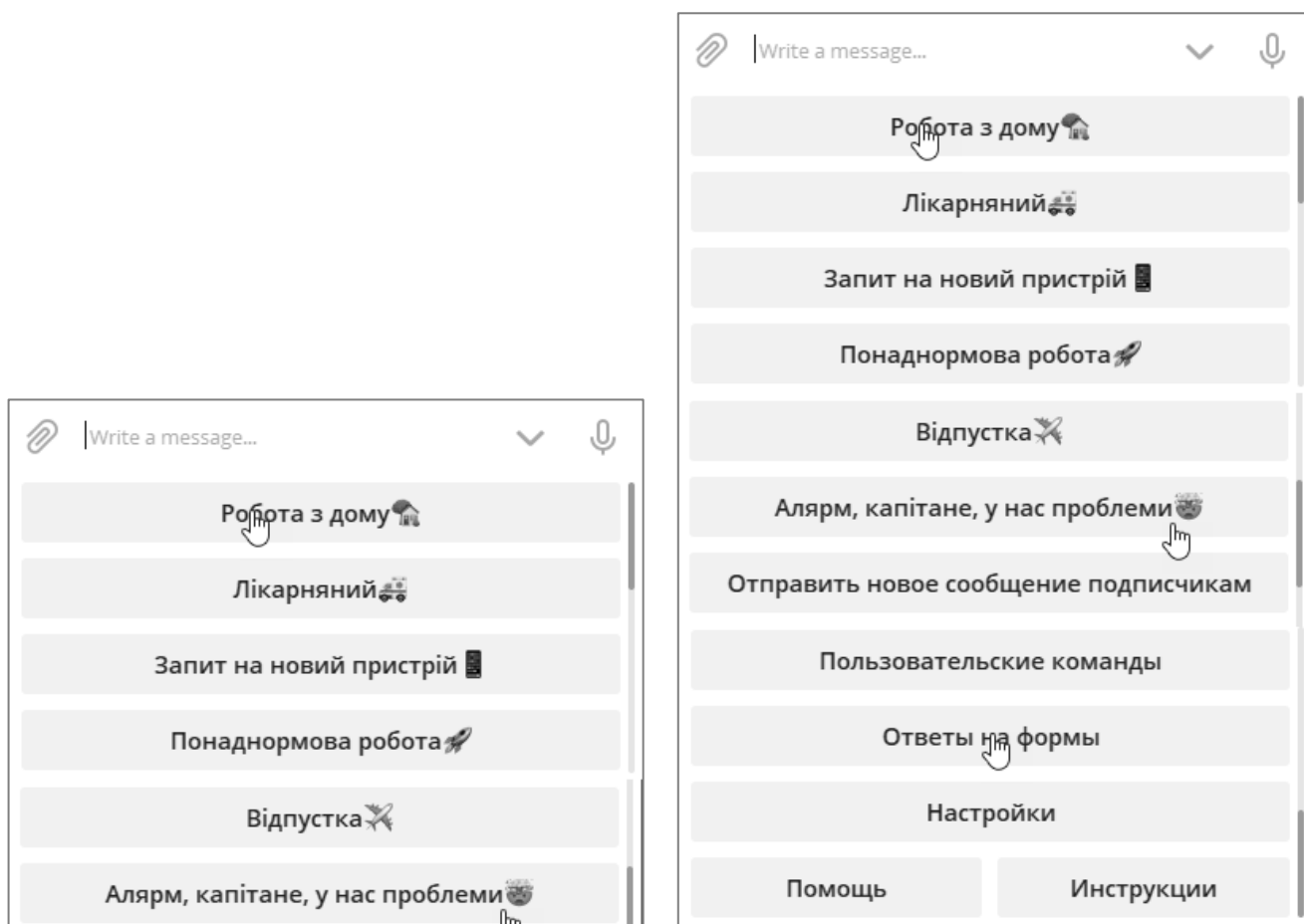


Рисунок 3.13 — Головне меню бота

Перші 6 пунктів будуть доступні для звичайних працівників, а доступ до всіх 12 пунктів надається виключно працівникам-адміністраторам бота або ж менеджменту компанії для опрацювання отриманих користувацьких відповідей.

### 3.2 Контрольний приклад та інструкція щодо використання чат-бота

Навчальні інструкції зазвичай використовуються як орієнтир у досягненні цілей для поставленого завдання. Хоча концепція інструкцій зазвичай стосується початківців, вони також можуть виявитися корисними для осіб, знайомих із завданнями, адже можуть детальніше розкрити певні аспекти задач, що дозволить

ефективніше їх опрацьовувати. Це дозволяє забезпечити рівномірність у розумінні і знаннях працівників і забезпечити максимально ефективну роботу.

Використання навчальних інструкцій не обмежується лише бізнесом, а є частим явищем у різноманітних установах, таких як: освітні заклади, військові заклади, релігійні організації тощо. Новий співробітник завжди потребує якихось базових правил і інструкцій для гарного початку своєї роботи в організації [39].

Розглянемо детально покрокову інструкцію з отримання доступу до бота і початку роботи над ним.

Крок 1 — необхідно встановити програму Telegram на комп'ютер чи телефон, для отримання доступу до її функцій (також можна скористатися веб версією телеграм, яка доступна за посиланням <https://web.telegram.org/#/login>, але варто зважати на те, що найновіші опції месенджера спочатку з'являються в додатках і тільки потім, в веб версії).

Крок 2 — виконати реєстрацію в програмі (виконується за номером телефону).

Крок 3 — потрібно дізнатися в свого адміністратора посилання на розробленого бота і канал в який будуть здійснюватися трансляції всіх важливих повідомлень і сповіщень і куди будуть надсилатися інші сповіщення чи нагадування.

Крок 4 — потрібно знайти необхідні канал і бота через пошук в програмі Telegram або ж перейшовши за посиланнями отриманими від адміністратора.

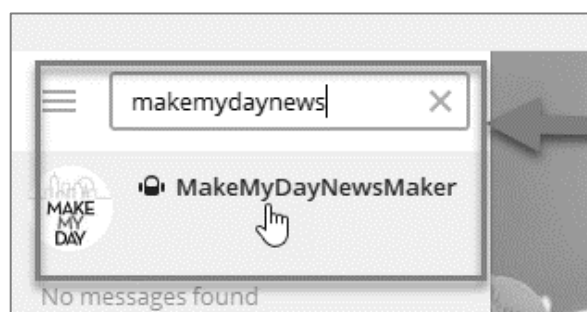


Рисунок 3.14 — Знаходження бота в телеграм через опцію пошуку

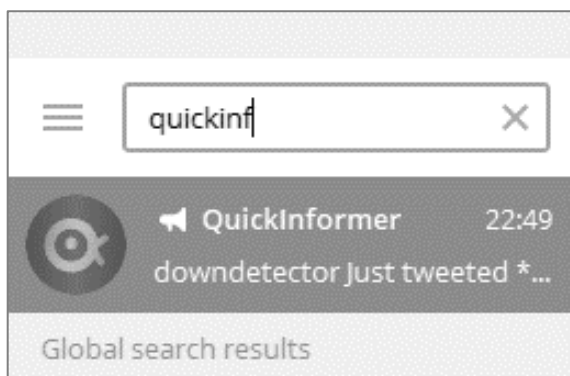


Рисунок 3.15 — Знаходження каналу в телеграм через опцію пошуку

Крок 5 — Після отримання доступу до бота почати його використання за призначенням.

Далі розглянемо використання чат-бота на основі контрольного прикладу з отримання працівником лікарняного.

Для отримання робітником лікарняного йому необхідно виконати налаштування чат-бота описані в кроках 1-5 вище. Для продовження процесу отримання лікарняного необхідно виконати наступні кроки:

Крок 1 — відкрити бота.

Крок 2 — виконати команду «/start».

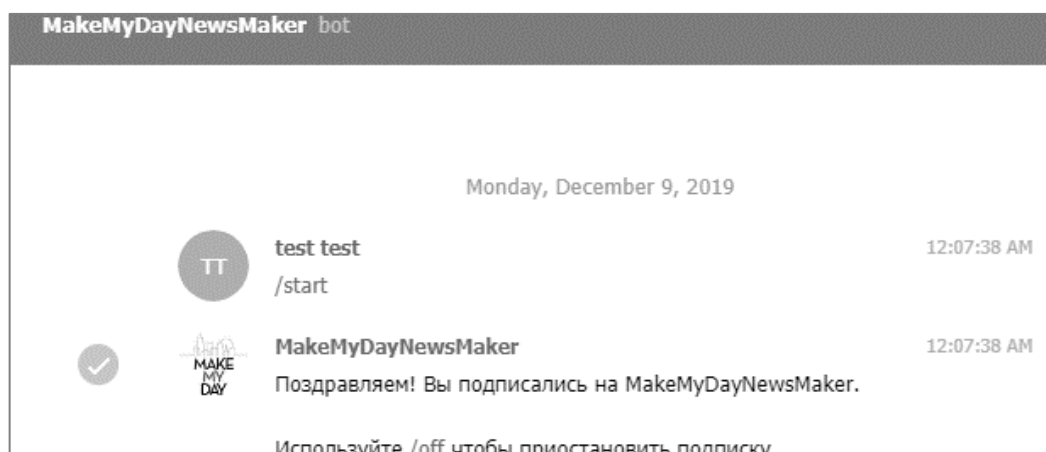


Рисунок 3.16 — Запуск роботи чат-бота.

Крок 3 — викликати головне меню бота.

Крок 4 — обрати підпункт «Лікарняний» з головного меню.

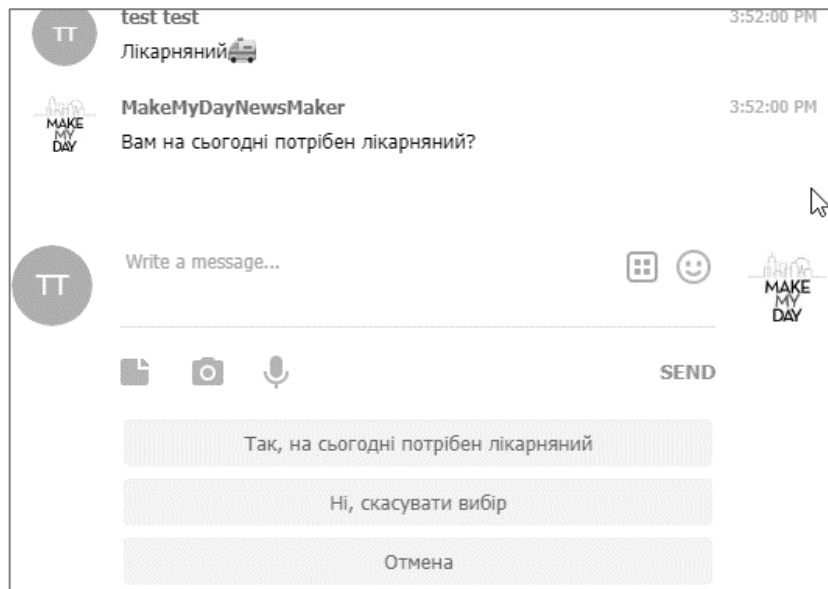


Рисунок 3.17 — Запуск процедури отримання лікарняного.

Крок 5 — підтвердити необхідність отримання лікарняного натиснувши кнопку «Так, на сьогодні потрібен лікарняний».

Крок 6 — отримати інформаційне повідомлення від бота про те, що лікарняний отримано.

Далі розглянемо проходження процедури роботи з підтвердження лікарняного з точки зору адміністратора, тобто людини відповідальної за подібні процеси в компанії.

Крок 1. — необхідно ввійти до бота як адміністратор.

Крок 2 — перевірити наявність повідомлень про лікарняний.

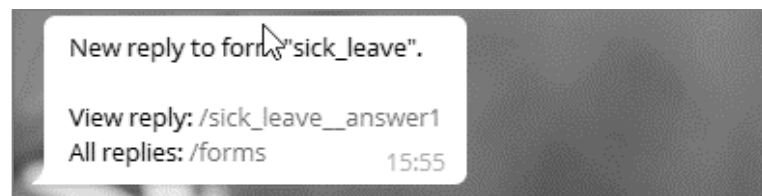


Рисунок 3.18 — Вигляд повідомлення про наявність запитів на лікарняний

Крок 3. — натиснути кнопку «View reply» для перегляду інформації про запит на лікарняний.



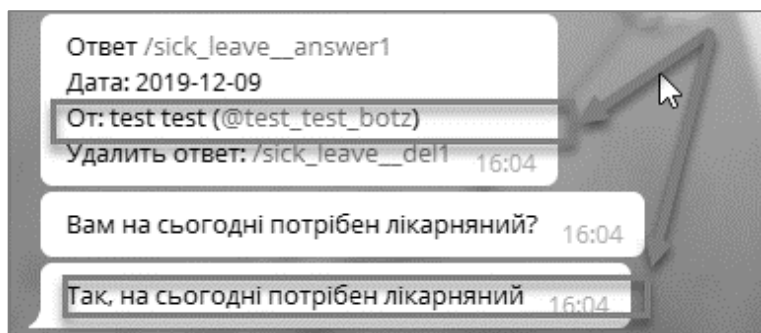


Рисунок 3.19 — Вигляд повідомлення з інформацією про запит на лікарняний.

Крок 4 — У випадку наявності активного запиту — перенести інформацію про отриманий запит (хто надіслав запит і яка дата отримання запиту) до системи роботи з лікарняними; якщо запит на отримання лікарняного має негативний статус, тобто лікарняний не потрібно, то жодних додаткових дій немає необхідності робити.

Крок 5 — повернутися до головного меню бота, для подальшої роботи з будь-якими іншими запитами.

### 3.3 Оцінювання очікуваного ефекту від впровадження розробленого автоматизованого рішення

Сучасна економіка країни диктує необхідність постійного підвищення ефективності функціонування суб'єктів господарської діяльності для забезпечення конкурентоспроможності продукції та конкурентоспроможних позицій підприємства [48, 50].

Інформація, особливо її автоматизована обробка, залишається важливим фактором підвищення ефективності діяльності будь-якої організації. Важливу роль у використанні інформації відіграють способи її реєстрації, обробки, нагромадження і передачі; систематизоване збереження інформації і її видача в потрібній формі; виробництво нової числової, графічної та іншої інформації [51, 52].

Економічний ефект від впровадження засобів автоматизації може бути лише непрямим, так як впроваджені засоби автоматизації не є прямим джерелом генерування доходу, а є або допоміжним засобом організації отримання прибутку, або ж якимось чином допомагають знизити фінансові витрати, або ж витрати часу. За результатами аналізу ефективності впровадження розроблюваного засобу автоматизації можна дійти висновку, що це вигідно, адже чат-бот дозволяє зменшити витрати часу на комунікації з приводу тих чи інших проблем, а також значно прискорити швидкість інформаційних потоків у всіх напрямках в рамках діяльності компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab).

Основний ефект від впровадження розробленого чат-боту полягає в:

- поліпшенні економічних показників роботи підприємства;
- поліпшенні господарських показників роботи підприємства;
- підвищенні оперативності управління;
- зниження трудовитрат на реалізацію процесу управління;
- зниженні фінансових витрат за рахунок наведених вище факторів.

Хоч вигода і непряма і не є обраховуваною, але вона буде помітна в середньо і довгостроковій перспективі зважаючи на велику кількість працівників у компанії і певний часовий лаг необхідний для повного впровадження і початку використання всіма учасниками процесу управління. Також впровадження цього засобу автоматизації може призвести до коригування і перегляду уже наявних бізнес-процесів роботи з персоналом, так як вже додані до чат-бота процедури будуть виконуватися швидше. Також інформація про використання впроваджених процедур може бути агрегована для подальшого аналізу їх використання і відповідно додавання інших важливих покращень чи навіть нових опцій.

Співробітники зможуть обробляти великі обсяги інформації за свій робочий час, що можна використовувати або для зменшення витрат на персонал або для швидкого розвитку бізнесу при незмінності кількості співробітників, зайнятих обробкою інформації.

Як показує практика автоматизація бізнес процесів, особливо таких як пришвидшення обміну інформацією в рамках управлінських процесів несе в собі великий потенціал для розвитку і матеріальну вигоду з плином часу.

Важливо відзначити, що в разі якщо одним програмним засобом автоматизуються різні підрозділи і співробітники, то зменшуються витрати на організацію документообігу між ними. Зменшуються як тимчасові так і матеріальні витрати.

## ВИСНОВКИ

Робота в команді і безпосередня тісна взаємодія між усіма учасниками команди є важливою і невід'ємною частиною процесу роботи у великих ІТ-компаніях.

З метою дослідження процесів управління розподіленими командами та реалізації прототипу чат-боту підтримки взаємодії у команді було виконано дослідження внутрішніх процесів у ІТ-компанії. Об'єктом виконаного дослідження виступили процеси управління розподіленими командами в ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab) — ІТ-компанії з тестування програмного забезпечення.

У кваліфікаційній роботі розв'язано такі задачі:

- досліджено загальну характеристика компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab);
- досліджено комунікаційні потоки в компанії;
- проведено аналіз стану управління персоналом в компанії і доступних для впровадження HRMS рішень для автоматизації управління персоналом;
- сформовані вимоги до інформаційної системи;
- досліджено моделі бізнес-процесів на підприємстві;
- розроблено архітектуру автоматизованої інформаційної системи та обрано технології вирішення поставлених завдань;
- досліджена функціональна структура завдання;
- підібрано характеристики системного і прикладного програмного забезпечення та підсистем для забезпечення функціональної частини;
- розроблено і проведено налаштування чат-ботів і наведено загальний алгоритм їх роботи;
- надано контрольний приклад та інструкція щодо використання чат-бота;
- проведено оцінювання очікуваного ефекту від впровадження розробленого автоматизованого рішення.

В розроблюваній системі реалізовано функціональні можливості, які делегуються на два рівні доступу користувачів — Admin для доступу до адміністративних опцій і Usual — для доступу до інформаційної системи компанії всіх її діючих працівників.

Для успішного використання розроблюваного рішення також необхідна наявність пристроїв, які будуть відповідати мінімальним вимогам для створюваного чат-бота.

Провівши аналіз стану автоматизації внутрішніх комунікацій всередині ТОВ «Центр мережових технологій ВЕБ100» (QATestLab) також можна намітити певні пропозиції для подальшої роботи, а саме – що доцільно продовжувати впровадження інструментів автоматизованого управління персоналом на підприємстві, а також необхідно детальніше дослідити і розробити систему управління та контролю за персоналом, впровадити і активно використовувати її.

Дослідне використання і тестування чат-боту підтвердило конструктивність підходу і дозволило намітити шляхи для продовження його розробки і розширення вже наявних функціональних можливостей, наприклад:

— автоматизувати додавання інформації про лікарняні до системи підприємства без участі співробітників;

— додати можливість використовувати календар в запитах про лікарняні чи відпустку;

— додати можливість проводити масові опитування співробітників для обирання найбільш затребуваних пристроїв;

— додати можливість обирати сервіси від яких співробітник хоче отримувати сповіщення про збої.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України про зайнятість населення URL: <https://zakon.rada.gov.ua/laws/show/5067-17> (дата звернення: 2.11.2019).
2. Можливості платформи Manybot. URL: <https://manybot.io/ru#features> (дата звернення: 2.11.2019).
3. Офіційний сайт BambooHR. URL: <https://www.bamboohr.com/> (дата звернення: 2.11.2019).
4. Офіційний сайт BizMerlinHR. URL: <https://www.bizmerlin.com/> (дата звернення: 2.11.2019).
5. Офіційний сайт Conrep PSA. URL: <https://dev.conrep.com/index.htm> (дата звернення: 2.11.2019).
6. Офіційний сайт iCims. URL: <https://www.icims.com/> (дата звернення: 2.11.2019).
7. Офіційний сайт Manybot. URL: <https://manybot.io/> (дата звернення: 2.11.2019).
8. Офіційний сайт Telegram. URL: <https://telegram.org/> (дата звернення: 27.11.2019).
9. Офіційний сайт компанії ТОВ Центр мережових технологій ВЕБ100 (QATestLab). URL: <http://QATestLab.com/> (дата звернення: 22.11.2019).
10. Сторінка початку створення бота на платформі Manybot. URL: <https://manybot.io/ru#create> (дата звернення: 27.11.2019).
11. "Application Architecture". Gartner IT Glossary) URL: <https://www.gartner.com/en/information-technology/glossary/application-architecture-aa> (дата звернення: 27.11.2019).
12. A framework for chatbot evaluation URL: <https://isquared.wordpress.com/2017/01/24/a-framework-for-chatbot-evaluation/> (дата звернення: 7.11.2019).
13. Abella A., Brown M. K., Buntschuh B. Development principles for dialog-based interfaces. Dialogue Processing in Spoken Language Systems Lecture Notes in

Computer Science. 2013. URL: [https://link.springer.com/chapter /10.1007%2F3-540-63175-5\\_43](https://link.springer.com/chapter/10.1007%2F3-540-63175-5_43) (дата звернення: 22.11.2019).

14. Ball G., Breese J., Emotion and personality in a conversational agent. In Embodied conversational agents. essay, Cambridge: MIT Press. 2009.

15. Betters E.. What is Google Assistant, how does it work, and which devices offer it?. 2017. URL: <https://www.pocket-lint.com/apps/news/google/137722-what-is-google-assistant-how-does-it-work-and-which-devices-offer-it> (дата звернення: 7.11.2019).

16. Carletta J., Isard S., Doherty-Sneddon G., Isard A., Kowtko J. C., Anderson A. H. The Reliability of a Dialogue Structure Coding Scheme . Association for Computational Linguistics/. 2007. URL: <http://www.aclweb.org/anthology/J97-1002> (дата звернення: 27.11.2019).

17. Che H., Tao J., Pan S., Letter-to-sound conversion using coupled Hidden Markov Models for lexicon compression. 2012 International Conference on Speech Database and Assessments. 2012. URL: <https://ieeexplore.ieee.org/document/6422464> (дата звернення: 22.11.2019).

18. Chung J., Gulcehre C., Cho K. H., Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. 2014 URL: <https://arxiv.org/pdf/1412.3555.pdf> (дата звернення: 27.11.2019).

19. Constance J., Perez S. Facebook Messenger now allows payments in its 30,000 chat bots. 2016. URL: [https://techcrunch.com /2016/09/12/messenger-bot-payments/](https://techcrunch.com/2016/09/12/messenger-bot-payments/) (дата звернення: 27.11.2019).

20. Dhillon R., Bhagat S., Carvey H., Shriberg E. Meeting Recorder Project: Dialog Act Labeling Guide. 2015. URL: <http://www1.icsi.berkeley.edu/ftp/pub/speech/papers/MRDA-manual.pdf> (дата звернення: 22.11.2019).

21. Entity Relationship Diagram – ER Diagram in DBMS. URL: <https://beginnersbook.com/2015/04/e-r-model-in-dbms/> (дата звернення: 27.11.2019).

22. Ferrucci D. Building Watson: An Overview of the DeepQA Project. Association for the Advancement of Artificial Intelligence. 2010. URL:

<http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303/2165/> (дата звернення: 2.11.2019).

23. Foxybot. URL: <https://www.messenger.com/t/foxybot> (дата звернення: 22.11.2019).

24. Functional organization: the advantages and the disadvantages. URL: <https://www.management-square.com/functional-organization/> (дата звернення: 27.11.2019).

25. Functional organizational structure introduction. URL: [www.orgcharting.com/functional-organizational-introduction/](http://www.orgcharting.com/functional-organizational-introduction/) (дата звернення: 7.11.2019).

26. Functional Organizational Structure: Everything You Need to Know. URL: <https://www.upcounsel.com/functional-organizational-structure> (дата звернення: 27.11.2019).

27. Functional structure of comparator's predicate in the compartment identification method. URL: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/39690> (дата звернення: 22.11.2019).

28. Functional Structure Organization Strength Weakness. URL: <https://smallbusiness.chron.com/functional-structure-organization-strength-weakness-60111.html> (дата звернення: 27.11.2019).

29. Galvão A. M., Barros F. A., Neves A. M. M., Ramalho G. L.. Adding Personality to Chatterbots Using the Persona-AIML Architecture. *Advances in Artificial Intelligence*. 2010. URL: [https://link.springer.com/chapter/10.1007%2F978-3-540-30498-2\\_96](https://link.springer.com/chapter/10.1007%2F978-3-540-30498-2_96) (дата звернення: 27.11.2019).

30. Ganguly D., Roy D., Mitra M., Jones G. J. Word Embedding based Generalized Language Model for Information Retrieval. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information*. 2015. URL: <https://dl.acm.org/citation.cfm?doid=2766462.2767780> (дата звернення: 22.11.2019).

31. Gruhn R. E., Minker W., Nakamura S. *Statistical Pronunciation Modeling for Non-Native Speech Processing*. 2014.

32. GymBot. URL: <https://gymbot.io/> (дата звернення: 2.11.2019).



33. Hiraoka T., Hummer G., Yoshino K., Toda T., Nakamura S. Active Learning for Example-Based Dialog Systems. Lecture Notes in Electrical Engineering Dialogues with Social Robots. 2016. URL: [https://link.springer.com/chapter/10.1007%2F978-981-10-2585-3\\_5](https://link.springer.com/chapter/10.1007%2F978-981-10-2585-3_5) (дата звернення: 22.11.2019).

34. How does IFTTT work? URL: <https://help.ifttt.com/hc/en-us/articles/115010158167-How-does-IFTTT-work-> (дата звернення: 7.11.2019).

35. How To Build Bots for Messenger. URL: <https://developers.facebook.com/blog/post/2016/04/12/bots-for-messenger/> (дата звернення: 27.11.2019).

36. Hung V., Gonzalez A., Demara R. Towards a Context-Based Dialog Management Layer for Expert Systems. 2009 International Conference on Information, Process, and Knowledge Management. 2009. URL: <https://ieeexplore.ieee.org/document/4782566/> (дата звернення: 27.11.2019).

37. Instalocate. URL: <https://www.messenger.com/t/instalocate/> (дата звернення: 27.11.2019).

38. Introduction of ER Model URL: <https://www.geeksforgeeks.org/introduction-of-er-model/> (дата звернення: 22.11.2019).

39. Koehn P. Phrase-Based Models. Statistical Machine Translation URL: <https://www.cambridge.org/core/books/statistical-machine-translation/phrasebased-models/0B9EC28212FB7D15563B8F9B85A2870B> (дата звернення: 27.11.2019).

40. Kral P., Cerisara C. Dialogue Act Recognition Approaches. Computing and Informatics/. 2010. URL: <https://ieeexplore.ieee.org/document/4685529> (дата звернення: 22.11.2019).

41. Kshirsagar S., Magnenat-Thalmann N. Virtual humans personified. Proceedings of the first international joint conference on Autonomous agents and multiagent systems. 2009. URL: <https://dl.acm.org/citation.cfm?doid=544741.544826> (дата звернення: 27.11.2019).

42. Kuligowska K., Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents. 2015. URL: [doi:10.18483/rcbr.22](https://doi.org/10.18483/rcbr.22) (дата звернення: 7.11.2019).

43. Lauinger T., Pankakoski V., Balzarotti D., Kirda E. Honeybot Your Man in the Middle for Automated Social Engineering. LEET'10 Proceedings of the 3rd USENIX conference on Large-Scale exploits and emergent threats: botnets, spyware, worms, and more. 2010. URL: <https://publications.sba-research.org/publications/autosoc-leet2010.pdf> (дата звернення: 22.11.2019).

44. Lokman A.. One-Match and All-Match Categories for Keywords Matching in Chatbot. American Journal of Applied Sciences. 2010. URL: <https://www.thescipub.com/abstract/10.3844/ajassp.2010.1406.1411> (дата звернення: 27.11.2019).

45. Matney L. Kik users have exchanged over 1.8 billion messages with the platform's 20,000 chatbots. 2016. URL: <https://techcrunch.com/2016/08/03/kik-users-have-exchanged-over-1-8-billion-messages-with-the-platforms-20000-chatbots/> (дата звернення: 22.11.2019).

46. MeditateBot. URL: <https://botlist.co/bots/2021-meditatebot> (дата звернення: 27.11.2019).

47. Middleton C., Baeza-yates R. A Comparison of Open Source Search Engines. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.6955rep=rep1type=pdf> (дата звернення: 7.11.2019).

48. Narang S. Are You Talking to a Chatbot? How Spam Chatbots Will Get Smarter. 2014. URL: <https://www.linkedin.com/pulse/20140804184616-7077712-are-you-talking-to-a-chatbot-how-spam-chatbots-will-get-smarter/> (дата звернення: 27.11.2019).

49. Nass C., Brave S. Wired for speech: how voice activates and advances the human-computer relationship. Cambridge, MA: MIT Press.

50. O'Boyle B. Amazon Echo: What can Alexa do and what services are . 2016. URL: <https://www.pocket-lint.com/smart-home/news/amazon/138846-what-is-alexa-how-does-it-work-and-what-can-amazons-alexa-do> (дата звернення: 22.11.2019).

51. O'Boyle B. What is Siri? Apple's personal voice assistant explained. 2015. URL: <https://www.pocket-lint.com/apps/news/apple/112346-what-is-siri-apple-s-personal-voice-assistant-explained> (дата звернення: 27.11.2019).

52. Peaceful Habit. URL: <https://botlist.co/bots/1381-peaceful-habit> (дата звернення: 27.11.2019).

53. Pereira M., Coheur L.. Just. Chat - a platform for processing information to be used in chatbots. 2013. URL: <https://fenix.tecnico.ulisboa.pt/downloadFile/395145485809/ExtendedAbstract.pdf> (дата звернення: 22.11.2019).

54. Rashad M. Z., El-Bakry H. M., Isma'il I. R. An Overview of Text-To-Speech Synthesis Techniques. CIT'10 Proceedings of the 4th international conference on Communications and information technolog. 2010. URL: <https://pdfs.semanticscholar.org/5ee7/c71a362cc01441c27ced052a70ee6e0732dc.pdf> (дата звернення: 27.11.2019).

55. Reiser V., Lemon O., Champhi., Reinforcement Learning for Adaptive Systems. Edinburgh:Springer. 2011.

56. Sansonnet J.-P., Leray D., Martin J.-C. Architecture of a Framework for Generic Assisting Conversational Agents. Intelligent Virtual Agents Lecture Notes in Computer Science. 2009. URL: [http://doi.org/10.1007/11821830\\_12](http://doi.org/10.1007/11821830_12) (дата звернення: 27.11.2019).

57. Secure Messaging Scorecard. 2016. URL: <https://www.eff.org/node/101713/> (дата звернення: 22.11.2019).

58. Sejnowski T. J., Rosenberg C. R., NETtalk: a parallel network that learns to read aloud. In Neurocomputing: foundations of research. essay, Cambridge, MA: MITPress. 2008.

59. Shawar B. A., Atwell E. Machine learning from dialogue corpora to generate chatbots. School of Computing, University of Leeds (2010) URL: <https://pdfs.semanticscholar.org/8bf9/391940801b762de98991c0fb5a426777d104.pdf> (дата звернення: 27.11.2019).

60. Shrestha L., Mckeown K. Detection of question-answer pairs in email conversations. Proceedings of the 20th international conference on Computational Linguistics. 2009. URL: <https://dl.acm.org/citation.cfm?doid=1220355.1220483> (дата звернення: 27.11.2019).

61. Singh S., Keams M., Litman D., Walker M. Reinforcement Learning for Spoken Dialogue Systems. Advances in Neural Information Processing Systems. 2015. URL: <https://papers.nips.cc/paper/1775-reinforcement-learning-for-spoken-dialogue-systems.pdf> (дата звернення: 22.11.2019).

62. Steven Spewak; S. C. Hill. Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology. Boston, QED Pub. Group. 2016.

63. Stolcke A., Ries K., Coccaro N. Shriberg E., Bates R., Jurafsky D., ... Meteer M.. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics. 2015 URL: <https://www.mitpressjournals.org/doi/10.1162/089120100561737> (дата звернення: 22.11.2019).

64. Telegram Bot Platform. URL: <https://telegram.org/blog/bot-revolution> (дата звернення: 27.11.2019).

65. Telegram Inline Bot Platform. URL: <https://telegram.org/blog/inline-bots> (дата звернення: 22.11.2019).

66. Tur G., Mori R. D. Spoken language understanding: systems for extracting semantic information from speech. 2011. (дата звернення: 27.11.2019).

67. Vugt H. C., Bailenson J. N. Hoorn J. F., Konijn E. A. Effects of facial similarity on user responses to embodied agents. ACM Transactions on Computer-Human Interaction. 2010. URL: doi:10.1145/1746259.1746261

68. Webhooks service FAQ URL: <https://help.ifttt.com/hc/en-us/articles/115010230347-Webhooks-service-FAQ> (дата звернення: 27.11.2019).

69. What are Skype bots and how do I add them as contacts? URL: <https://support.skype.com/en/faq/FA34646/what-are-skype-bots-and-how-do-i-add-them-as-contacts> (дата звернення: 22.11.2019).

70. What is a Functional Organization Structure? URL: <https://pmstudycircle.com/2012/08/what-is-a-functional-organization-structure/> (дата звернення: 27.11.2019).

71. What is Application Architecture? URL: <https://www.orbussoftware.com/us/application-architecture/what-is-application-architecture/> (дата звернення: 27.11.2019).

72. What is IFTTT? URL: <https://help.ifttt.com/hc/en-us/articles/115010325748-What-is-IFTTT>- (дата звернення: 22.11.2019).

73. Why you shouldn't talk to your chatbot about everything URL: <https://venturebeat.com/2016/11/17/why-you-shouldnt-talk-to-your-chatbot-about-everything/> (дата звернення: 27.11.2019).

74. Wu Y., Wu Sequential Match Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots. 2016. URL: <https://arxiv.org/pdf/1612.01627.pdf> (дата звернення: 22.11.2019).

75. Yan Z., Duan N., Bao J., Chen P., Zhou M., Li Z., Zhou J. Doc Chat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. URL: <https://www.aclweb.org/anthology/P16-1049/> (дата звернення: 27.11.2019).

## ДОДАТКИ

ДОДАТОК А  
(Обов'язковий)

SUMMARY

Kozyrka V.E. Automation of the processes for managing distributed teams using chat-bot technology. — Masters-level Qualification Thesis. Sumy State University, Sumy, 2019.

The master's thesis conducted research on management processes in the company "Center Network Technology WEB100" (QATestLab) and the requirements for the automated information system were composed. The architecture of the automated information system has been designed. A prototype of an automated chatbot information system has been developed.

Keywords: automated information system, chatbot, distributed teams, automation of the processes for managing, prototype, communications, Telegram messenger.

АНОТАЦІЯ

Козирка В.Є. Автоматизація процесів управління розподіленими командами з використанням технології чат-ботів. — Кваліфікаційна магістерська робота. Сумський державний університет, Суми, 2019 р.

У роботі проведено дослідження процесів управління в компанії ТОВ «Центр мережевих технологій ВЕБ100» (QATestLab) і сформовано вимоги до автоматизованої інформаційної системи. Виконано проектування архітектури автоматизованої інформаційної системи. Створено прототип автоматизованої інформаційної системи на основі чат-бота.

Ключові слова: автоматизована інформаційна система, чат-бот, розподілені команди, автоматизація процесів управління, прототип, комунікації, месенджер Telegram.

## ДОДАТОК Б (Інформаційний)

### Код налаштування опцій сповіщень

```

# -*- coding: utf-8 -*-
from io import StringIO
from telebot import types
from SiteAlert import *
import time
import eventlet
import requests
import logging
import telebot
from time import sleep

# Отримуємо останні записи від акаунта Downtdetector
URL_TW= 'https://api.twitter.com/method/wall.get?domain=team&count=10&filter=owner&
access_token=1058793582:AAELUHaTwFzvrtfn5M97STushwx9c3mslJw '
FILENAME_TW = 'last_known_id.txt'
BASE_POST_URL = 'https://twitter.com/wall-22822305_'

BOT_TOKEN = '1058793582:AAELUHaTwFzvrtfn5M97STushwx9c3mslJw'
CHANNEL_NAME = '@QuickInformer'

bot = telebot.TeleBot(1058793582:AAELUHaTwFzvrtfn5M97STushwx9 c3mslJw)

#Налаштовуємо тайм-аут після якого бот зупиняється до наступного часу синхронізації
def get_data():
    timeout = eventlet.Timeout(10)
    try:
        feed = requests.get(URL_TW)
        return feed.json()
    except eventlet.timeout.Timeout:
        logging.warning('Got Timeout while retrieving TW JSON data. Cancelling...')
        return None
    finally:
        timeout.cancel()

```



```

def send_new_posts(items, last_id):
    for item in items:
        if item['id'] <= last_id:
            break
        link = '{!s}{!s}'.format(BASE_POST_URL, item['id'])
        bot.send_message(CHANNEL_NAME, link)
        # Затримка в 1 секунду, для захисту від помилок і проблем з відповіддю від сервера!)
        time.sleep(1)
    return

def check_new_posts_tw():
    # Поточний чат початку
    logging.info('[TW] Started scanning for new posts')
    with open(FILENAME_TW, 'rt') as file:
        last_id = int(file.read())
        if last_id is None:
            logging.error('Could not read from storage. Skipped iteration.')
            return
        logging.info('Last ID (TW) = {!s}'.format(last_id))
    try:
        feed = get_data()
        # Якщо перед цим спрацював таймаут, то пропускаємо 1 ітерацію. Якщо таймауту не було,
        то запускаємо парсинг постів.
        if feed is not None:
            entries = feed['response'][1:]
            try:
                # Пропускаємо закріплений пост
                tmp = entries[0]['is_pinned']
                # Запускаємо відправку нових записів
                send_new_posts(entries[1:], last_id)

```

```

# Оновлюємо last_id в файлі.

with open(FILENAME_TW, 'wt') as file:

    try:

        tmp = entries[0]['is_pinned']

        # Якщо перший пост закріплено, то зберігаємо ID другого
        file.write(str(entries[1]['id']))

        logging.info('New last_id (TW) is {}'.format((entries[1]['id'])))

    except KeyError:

        file.write(str(entries[0]['id']))

        logging.info('New last_id (TW) is {}'.format((entries[0]['id'])))

except Exception as ex:

    logging.error('Exception of type {} in check_new_post(): {}'.format(type(ex).__name__, str(ex)))

    pass

logging.info('[TW] Finished scanning')

return

if __name__ == '__main__':

    # Очищаємо технічну інформацію переслану бібліотекою requests
    logging.getLogger('requests').setLevel(logging.CRITICAL)

    # Налаштовуємо логгер
    logging.basicConfig(format='%(asctime)s      %(filename)s:%(lineno)d      %(levelname)s
%(message)s', level=logging.INFO,

                        filename='bot_log.log', datefmt='%d.%m.%Y %H:%M:%S')

    check_new_posts_tw()

    # Пауза в 4 хвилини перед повторною перевіркою
    logging.info('[App] Script went to sleep.')

    time.sleep(60 * 4)

else:

    check_new_posts_tw()

logging.info('[App] Script exited.\n')

```

ДОДАТОК Г  
(Інформаційний)

Запит і відповідь Twitter API від сервісу Downtdetector

```
# Request Tweets from last 7 days
# Search query: from:Downtdetector OR #downtdetector
twurl "/1.1/search/tweets.json?q=from%3ADowntdetector%20OR%20%23downtdetector"
# Response, an array of Tweet JSON:
{
  "statuses": [
    {
      "created_at": "Wed Nov 12 04:53:25 +0000 2017",
      "id": 852021818290352129,
      "id_str": "852021818290352129",
      "text": "Watch DOWNETECTOR's first - Engadget https:t.co\EfwAYeJpjF",
      "truncated": false,
      "entities": {
        "hashtags": [
          "downtdetector"
        ],
        "symbols": [
        ],
        "user_mentions": [
        ],
        "urls": [
          {
            "url": "https://t.co/EfwAYeJpjF",
            "expanded_url": "http://ift.tt/2orifBN",
            "display_url": "ift.tt/2orifBN",
            "indices": [
              54, 78
            ]
          }
        ]
      },
      "metadata": {
        "iso_language_code": "en",
        "result_type": "recent"
      },
      "source": "<a href='\"https://ifttt.com\"' rel='\"nofollow\">IFTTT</a>",
      "in_reply_to_status_id": null,
      "in_reply_to_status_id_str": null,
      "in_reply_to_user_id": null,
      "in_reply_to_user_id_str": null,
      "in_reply_to_screen_name": null,
      "user": {
        "id": 622857704,
```

```

"id_str": "622857704",
"name": "Crucial-Tech",
"screen_name": "crucial_tech",
"location": "Worldwide",
"description": "Technology News | Stories | Solutions | Workarounds | Gadgets",
"url": null,
"entities": {
  "description": {
    "urls": [
      ]
    }
  },
"protected": false,
"followers_count": 1917,
"friends_count": 841,
"listed_count": 513,
"created_at": "Sat Nov 30 14:28:07 +0000 2012",
"favourites_count": 33,
"utc_offset": -14400,
"time_zone": "Eastern Time (US & Canada)",
"geo_enabled": true,
"verified": false,
"statuses_count": 559097,
"lang": "en",
"contributors_enabled": false,
"is_translator": false,
"is_translation_enabled": false,
"profile_background_color": "131516",
"profile_background_image_url":
"http://pbs.twimg.com/profile_background_images/530442443057942528/vjgQgrriz.jpeg",
"profile_background_image_url_https":
"https://pbs.twimg.com/profile_background_images/530442443057942528/vjgQgrriz.jpeg",
"profile_background_tile": true,
"profile_image_url":
"http://pbs.twimg.com/profile_images/810537113288482816/AL7srBp3_normal.jpg",
"profile_image_url_https":
"profile_banner_url": "https://pbs.twimg.com/profile_banners/622857704/1415224702",
"profile_link_color": "3B94D9",
"profile_sidebar_border_color": "000000",
"profile_sidebar_fill_color": "000000",
"profile_text_color": "000000",
"profile_use_background_image": true,
"has_extended_profile": false,

```

```

    "has_extended_profile": false,
    "default_profile": false,
    "default_profile_image": false,
    "following": false,
    "follow_request_sent": false,
    "notifications": false,
    "translator_type": "none"
  },
  "geo": null,
  "coordinates": null,
  "place": null,
  "contributors": null,
  "is_quote_status": false,
  "retweet_count": 0,
  "favorite_count": 0,
  "possibly_sensitive": false,
  "lang": "en"
},
  "metadata": {
    "iso_language_code": "en",
    "result_type": "recent"
  },
  "source": "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>",
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": null,
  "in_reply_to_screen_name": null,
  "user": {
    "id": 18961853,
    "id_str": "18961853",
    "name": "Hongik Kim",
    "screen_name": "nottora2",
    "location": "Seoul, Korea",
    "description": "",
    "url": "http://t.co/WpOh7jvYit",
    "entities": {
      "url": {
        "urls": [
          {
            "url": "http://t.co/WpOh7jvYit",
            "expanded_url": "http://sttora2.net",

```

```
    "display_url": "sttora2.net",
    "indices": [
      "description": {
        "urls": [
          ]
        }
      }
    ],
  },
```