

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ**

Факультет Магістерської підготовки

Кафедра Інформаційних технологій

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Розробка хелпдеск для моніторингу результатів аудиту ІС

Виконав студент 2 курсу групи МІС-19  
спеціальності 122 Комп'ютерні науки

Кульчицький Ярослав Олександрович

Керівник д.т.н., професор  
Казакова Надія Феліксівна

Рецензент регіональний координатор  
Програми EGAP  
Копиченко Іван Юрійович

Одеса 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки  
Кафедра Інформаційних технологій  
Рівень вищої освіти магістр  
Спеціальність 122 Комп'ютерні науки  
(шифр і назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_

“ 26 ” жовтня 2020 р.

**З А В Д А Н Н Я**  
**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Кульчицького Ярослава Олександровича

(прізвище, ім'я, по батькові)

1. Тема роботи « Розробка хелпдеск для моніторингу результатів аудиту ІС »  
керівник роботи Казакова Надія Феліксівна, д.т.н., професор,  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 16 ” жовтня № 194 «с»

2. Строк подання студентом роботи 7 грудня 2020р.

3. Вихідні дані до роботи 1. Аудит системи моніторингу інформаційних систем.

2. Аналіз предметної області.

3. Порівняння існуючих Open Source HelpDesk/ServiceDesk систем.

4. Проектування інформаційної системи HelpDesk відділу технічної підтримки UALinux.

5. Реалізація інформаційної системи HelpDesk відділу технічної підтримки UALinux.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

## 6. Консультанти розділів роботи

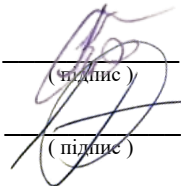
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 26 » жовтня 2020 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1.	Аудит системи моніторингу інформаційних систем	<b>26.10.2020</b>	80	<b>відм</b>
2.	Аналіз предметної області	31.10.2020	75	<b>відм</b>
3.	Порівняння існуючих Open Source HelpDesk/ServiceDesk систем	2.11.2020	80	<b>відм</b>
	Рубіжна атестація	<b>19.11.2020</b>	80	<b>відм</b>
4.	Проектування інформаційної системи HelpDesk відділу технічної підтримки UALinux	21.11.2020	80	<b>відм</b>
5.	Реалізація інформаційної системи HelpDesk відділу технічної підтримки UALinux	30.11.2020	80	<b>відм</b>
	Подання роботи на кафедру	<b>07.12.2020</b>	75	<b>викон</b>
	Перевірка на плагіат	<b>08.12.2020</b>		<b>викон</b>
	Рецензування	<b>16.12.2020</b>		<b>викон</b>
	<b>Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)</b>		75	<b>добре</b>

Студент

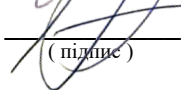


(підпис)

**Кульчицький Я.О.**

(прізвище та ініціали)

Керівник роботи



(підпис)

**Казакова Н.Ф.**

(прізвище та ініціали)

## АНОТАЦІЯ

на магістерську кваліфікаційну роботу  
«Розробка хелпдеск для моніторингу результатів аудиту ІС»,  
студента Кульчицького Ярослава Олександровича

Актуальність теми магістерської кваліфікаційної роботи обумовлена тим, що великі організації мають у своєму розпорядженні складну ІТ-інфраструктуру, безперервне функціонування всіх елементів якої на належному рівні є обов'язковою умовою для виконання організацією своїх основних функцій. Підтримка цієї інфраструктури в робочому стані є однією з основних функцій ІТ-служби підприємства. Системи HelpDesk дозволяють ІТ-службам забезпечити якісне виконання цієї функції.

Метою роботи є проектування системи HelpDesk для компанії «UALinux», яка займається створенням комплексних технологічних рішень і підготовок висококваліфікованих фахівців, здатних ефективно застосовувати і обслуговувати сучасні телекомунікаційні та інформаційні технології в найширших областях діяльності.

Об'єкт дослідження – є проектування системи HelpDesk.

Предмет дослідження – є теоритичні, методологічні, програмні та прикладні аспекти проектування HelpDesk-системи.

В ході даної роботи була дослідженна інформаційна система технічної підтримки, обрано та перероблено один з аналогів HelpDesk-систем, який найбільш відповідає умовам ТЗ для компанії UALinux. Для виконання поставленої задачі, було обрано скриптовану мову програмування PHP, серверну операційну систему Ubuntu Server, вже існуюча opensource HelpDesk система HESK, та система керування базами даних MariaDB.

Магістерська кваліфікаційна робота містить 78 сторінок, 43 рисунка, 1 таблицю та 17 джерела.

Ключові слова: АНАЛІЗ, HELPDESK, SERVICEDESK, АУДИТ, ІТІЛ, ІНФОРМАЦІЙНА СИСТЕМА, ТЕХНІЧНА ПІДТРИМКА, ЗАПИТИ, УРАЗЛИВІСТЬ.

## **SUMMARY**

for a master's degree

"Development of Helpdesk for Monitoring the Results of the Information System Audit", students of Kulchitskyi Yaroslav

The relevance of the topic of master's qualification work is due to the fact that large organizations have a complex IT infrastructure, the continuous operation of all elements of which at the appropriate level is a prerequisite for the organization to perform its core functions. Maintaining this infrastructure in working order is one of the main functions of the company's IT service. HelpDesk systems will allow IT services to ensure quality performance of this function.

The aim of the work is to design a HelpDesk system for UALinux, which creates complex technological solutions and training of highly qualified specialists capable of effectively applying and servicing modern telecommunication and information technologies in the widest areas of activity.

The object of research is the design of the HelpDesk system.

The subject of research – is the theoretical, methodological, software and applied aspects of the design of HelpDesk-system.

In the course of this work, the technical support information system was researched, one of the analogues of HelpDesk-systems was selected and redesigned, which best meets the terms of the TR for UALinux. To accomplish this task, a scripted PHP programming language, the Ubuntu Server operating system, the existing opensource HelpDesk system HESK, and the MariaDB database management system were selected.

The master's thesis contains 78 pages, 43 figures, 1 table and 17 sources.

**Keywords: ANALYSIS, HELPDESK, SERVICEDESK, AUDIT, ITIL, INFORMATION SYSTEM, TECHNICAL SUPPORT, REQUESTS.**

## ЗМІСТ

Вступ.....	10
1 Аудит системи моніторингу інформаційних систем.....	11
1.1 Основні відомості про аудит.....	11
1.2 Основні етапи проведення аудита.....	12
1.3 Об'єкт та межі аудиту.....	14
2 Аналіз предметної області.....	16
3 Порівняння існуючих Open Source HelpDesk/ServiceDesk систем .....	22
3.1 OTRS.....	23
3.2 osTicket.....	24
3.3 Request Tracket.....	25
3.4 GLPI.....	26
3.5 FreeScout.....	27
3.6 HESK .....	28
4 Проектування інформаційної системи HelpDesk відділу технічної підтримки UALinux.....	30
4.1 Вимоги до проєктованої інформаційної системи відділу технічної підтримки компанії UALinux.....	32
4.1.1 Вимоги до структури системи .....	32
4.1.2 Вимоги до функцій, виконуваних системою .....	32
4.2 Вибір архітектури інформаційної системи.....	33
4.3 Архітектура інформаційної системи.....	39
4.4 Вибір операційної системи .....	40
4.4.1 Windows Server 2019.....	41
4.4.2 FreeBSD.....	42
4.4.3 Ubuntu Server .....	44
4.4.4 Solaris.....	45
4.5 Вибір системи управління базами даних.....	46
4.5.1 Microsoft SQL Server.....	46
4.5.2 MySQL .....	48

4.5.3 MariaDB .....	49
5 Реалізація інформаційної системи HelpDesk відділу технічної підтримки UALinux 51	
Висновки .....	76
Перелік джерел посилань .....	78



## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

БД – база даних.

ПЗ – програмне забезпечення.

ІС – інформаційна система.

ІТ – Information Technology.

ІТІЛ – Infrastructure Library.

ІТSM – IT Service Management.

UNIX – це сімейство багатозадачних і багатокористувацьких операційних систем, які засновані на ідеях оригінального проекту AT&T.

PHP – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.

СУБД – система управління базами даних.

SQL – Structured Query Language.

.NET Framework – це технологія, яка підтримує створення і виконання веб-служб і додатків Windows.

ТЗ – технічне завдання.

PS – Presentation Services.

PL – Presentation Logic.

BL – Business or Application Logic.

DL – Data Logic.

DS – Data Services.

FS – File Services.

ОС – операційна система.

SLA – зовнішній документ, існуючий між замовником і виконавцем, що описує параметри послуги, що надається.

## ВСТУП

Інформаційні технології (ІТ), що стрімко розвиваються, вносять помітні зміни в наше життя. Інформація стала товаром, який можна придбати, продати, обміняти. При цьому вартість інформації часто у багато разів перевищує вартість комп'ютерної системи, в якій вона зберігається. Від ступеня безпеки ІТ в даний час залежить добробут, а часом і життя багатьох людей. Для отримання об'єктивних якісних і кількісних оцінок про поточний стан інформаційної системи компанії, а також рекомендацій щодо інформаційних ризиків, проводиться системний процес аудиту ІС.

Актуальність теми роботи обумовлена тим, що великі організації мають у своєму розпорядженні складну ІТ-інфраструктуру, безперервне функціонування всіх елементів якої на належному рівні є обов'язковою умовою для виконання організацією своїх основних функцій. Системи HelpDesk дозволяють ІТ-службам забезпечити якісне виконання цієї функції.

Метою роботи є проектування системи HelpDesk для компанії «UALinux», яка займається створенням комплексних технологічних рішень і підготовок висококваліфікованих фахівців, здатних ефективно застосовувати і обслуговувати сучасні телекомунікаційні та інформаційні технології в найширших областях діяльності. [1]<sup>1)</sup>

Для досягнення поставленої мети треба вирішити наступні задачі:

- розкрити теоретичні основи систем HelpDesk та поняття аудиту інформаційної системи;
- виконати порівняльний огляд існуючих аналогів;
- обґрунтування проектних рішень з програмного, технічного та інформаційного забезпечення;
- виконати аналіз ефективності розробленого HelpDesk системи.

---

<sup>1)</sup> [1] Про компанію UALinux. URL: <https://ualinux.com/ru/about> (дата звернення: 12.08.2020)

# 1 АУДИТ СИСТЕМИ МОНІТОРИНГУ ІНФОРМАЦІЙНИХ СИСТЕМ

## 1.1 Основні відомості про аудит

Аудит – систематичний, незалежний і документований процес отримання даних аудиту та їх об'єктивне оцінювання з метою встановлення відповідності критеріям аудиту.

Внутрішні аудити, які іноді називають «аудити першої сторони», проводяться самою організацією або від її імені з метою аналізу з боку керівництва та для інших внутрішніх цілей (наприклад, для підтвердження результативності системи менеджменту або для отримання інформації про підвищення ефективності системи менеджменту).

Зовнішні аудити – це аудити, які проводяться другою і третьою сторонами. Аудити другої сторони проводяться сторонами, що мають інтерес до організації (наприклад, споживачами), або іншими особами від їх імені. Аудити третьої сторони проводяться незалежними аудиторськими організаціями (наглядовими органами або організаціями, що здійснюють сертифікацію).

Аудит дає керівнику розуміння проблем і шляхи їх вирішення, відповідність ІТ планам розвитку організації, регламентацію процесів та оптимізацію бюджету і збереження інвестицій в ІТ.

Співробітникам він надає план розвитку ІТ, регламентацію діяльності, розуміння напрямку розвитку процесів, цілей впровадження ІТ та стабільну роботу.

Професія ІТ-аудитора вимагає спеціального досвіду у сфері інформаційних технологій та з питань забезпечення організації управлінською інформацією. ІТ-аудитори повинні бути ознайомлені із загальноприйнятими методами і прийомами внутрішнього аудиту, в т.ч. з тими, які використовуються під час ІТ-аудиту, тестуванням та оцінкою ризиків, пов'язаних з ІТ. Вони також повинні володіти інформацією про вартість ІТ та

спеціальними навичками у деяких сферах прикладного програмного забезпечення. Збільшення актуальності ІТ посилює вимоги до якості, яка очікується від ІТ-аудиторів при виконанні завдань.

## **1.2 Основні етапи проведення аудита**

При здійсненні аудиту моніторингу інформаційної безпеки в інформаційних системах, повинна забезпечуватися можливість отримання інформації про зареєстровані події безпеки та інших даних від різних джерел, таких як: засоби захисту інформації, общесистемное і прикладне програмне забезпечення, спеціальне програмне забезпечення і програми-сервіси, інші джерела даних .

Компетентні висновки щодо стану справ в компанії з інформаційною безпекою можуть бути зроблені аудитором тільки за умови наявності всіх необхідних вихідних даних для аналізу. Аудиторський обстеження починається з отримання інформації про організаційну структуру користувачів ІС і обслуговуючих підрозділів. Призначення і принципи функціонування ІС багато в чому визначають існуючі ризики і вимоги безпеки, що пред'являються до системи. Далі, аудитору потрібно більш детальна інформація про структуру ІС. Це дозволить усвідомити, яким чином здійснюється розподіл механізмів безпеки за структурними елементами і рівнями функціонування ІС.

Етапи проведення аудиту системи моніторингу інформаційних систем:

1. Інтерв'ю з відповідальним співробітником за моніторинг інформаційних систем і аналіз наявної документації. На даному етапі, як правило, уточнюються вимоги до результатів, узгоджується перелік ризиків, які необхідно розглянути і оцінити. Також визначаються межі аудиту, тобто список бізнес-процесів і підрозділів, які будуть досліджуватися.

2. Формування робочої групи. До складу робочої групи повинні входити фахівці компанії виконавця (компанії проводить аудит) та співробітники компанії замовника. Співробітники замовника забезпечують уявлення всієї необхідної інформації, контролюють процеси проведення обстеження, а також беруть участь в узгодженні його результатів (проміжних і кінцевих). Фахівці виконавця відповідають за кваліфіковане проведення робіт з обстеження предметних областей відповідно до визначених цілями і завданнями проекту, узгодять процеси і результати проведення обстеження.
3. Збір первинної інформації та її аналіз (накази, акти контролю, журнали інцидентів і т.д.) і наданих даних за рівнями моніторингу інформаційної безпеки:
  - рівень джерел даних (засоби захисту інформації; програмне забезпечення; програмно-технічні засоби, інформаційні сервіси і додатки; операційні системи, СУБД);
  - рівень збору даних (збір вихідних даних (логи) в обсязі, необхідному для проведення аналізу і різного роду оцінок стану інформаційної безпеки;
  - рівень зберігання і обробки даних (які функції і заходи реалізовані для зберігання, обробки даних моніторингу; терміни і формат зберігання даних про події інформаційної інфраструктури аудиту.
  - рівень представлення інформації даних моніторингу (результати подій в ІС: дані від різних джерел в ІС; кореляція даних (логів); контроль, облік і статистичний аналіз зібраних даних, виявлення та пошук інцидентів інформаційної безпеки; отримання даних про нові загрози. (Аналіз інцидентів).
4. Проведення тестування системи моніторингу ІС.
5. Виявлення необхідності внесення змін до системи моніторингу інформаційної системи.

6. Проведення підсумкової оцінки відповідності нормативним і законодавчим актам відповідно до отриманих і обробленими свідоцтвами (Крім стандартних рекомендацій щодо усунення виявлених в ході аудиту невідповідностей (зауважень), результати аудиту можуть бути використані для розробки ключових рекомендацій з тактики і стратегії вдосконалення системи ІТ-управління в цілому і окремих ІТ-процесів, а також для формування методичної основи систем внутрішнього контролю і аудиту за інформаційними технологіями організації). [2]<sup>1)</sup>

### 1.3 Об'єкт та межі аудиту

Проведення аудиту ІТ-середовища організації, а відповідно визначення правильного об'єкта ІТ-аудиту та критеріїв аудиту, слід базувати на ризиках. ІТ-аудитори повинні вирішувати, які сфери підлягають аудиту на основі оцінки ризиків шляхом визначення критичнихб активів ІТ та врахування потенційних ризиків і важливих заходів контролю. Активи – це цінні інформаційні системи, процедури і правила. Ризики – це потенційно «погані події», які можуть відбутися з активами. Заходи контролю – це пом'якшуючі фактори для захисту активів від потенційних ризиків.

Деякі типові фактори ризику для інформаційних технологій можуть включати:

- характер інформаційної системи, напр., з використанням Інтернету чи ні. Або чи це критична фінансова система чи базова реєстраційна система;

---

<sup>1)</sup> [2] Аудит систем мониторинга информационных систем. URL: <https://rtmtech.ru/services/audit-sistemy-monitoringa-is/#:~:text=Вопрос-ответ-,Аудит%20системы%20мониторинга%20информационных%20систем,и%20уязвимостей%20в%20информационных%20системах> (дата звернення: 26.10.2020)

- складність інформаційної системи, напр., чи це проста незалежна система чи ланка ланцюжка взаємозалежних систем;
- зрілість інформаційної системи, напр., чи це нова система з багатьма питаннями, які можуть виникнути вперше, чи це зріла система, яку вже неодноразово перевіряли аудитом;
- вартість інформаційної системи часто пов'язана із складністю інформаційної системи. Актуальна у тому контексті, що дуже дорогі системи привертають більше уваги вищого керівництва;
- вразливість і захист персональних даних, напр., інформація у медичній системі вразливіша за інформацію у фінансовій системі;
- аспекти безперервності, напр., деякі інформаційні системи повинні постійно функціонувати, тоді як інші – тільки у робочий час.

## 2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

HelpDesk – інформаційна система технічної підтримки, вирішення проблем користувачів з комп'ютерами, апаратним та програмним забезпеченням. Це важлива складова ІТІЛ – дозволяє виявити проблемні ділянки інфраструктури ІТ, оцінити ефективність роботи відділу ІТ.

Система HelpDesk є кращою на сьогоднішній день базою даних для зберігання всієї інформації, пов'язаної зі зверненнями клієнтів, видачею завдань фахівцям. Така система має простий і зрозумілий інтерфейс, допомагає стежити за тим, наскільки послідовно виконуються роботи, вести облік інцидентів, щоб оповіщати про них адміністраторів і стежити за тим, скільки часу і грошей було витрачено на той чи інший запит, розстановка пріоритетів в клієнтських заявках і т.д. Одним словом, HelpDesk – це багатофункціональне програмне забезпечення, здатне багато в чому полегшити і прискорити роботу компанії, а також підвищити її рентабельність і престиж.

Часто споживачі плутають терміни HelpDesk і ServiceDesk. У цих систем однакові завдання: вони надають технічну підтримку клієнтам, надають їм висококласний сервіс і допомогу у вирішенні складних ситуацій, тому для споживача немає абсолютно ніякої різниці між ServiceDesk і HelpDesk. Для клієнта важливо лише якість надання послуги та швидкість вирішення проблеми.

Термін HelpDesk з'явився в 80-і роки. У ті часи це була система, незв'язана з ІТ-технологіями, але покликана вирішувати внутрішні проблеми бізнесу, які виникали вкрай рідко. Зараз початкової потребою системи HD вважається обслуговування споживачів і надання їм технічної підтримки. Завдяки цій роботі з'явилася точка комунікації для споживачів і персоналу ІТ-відділів, які в єдиному вікні можуть переглянути інформацію про клієнтів і надати необхідну їм допомогу.



Поняття ServiceDesk було введено в ужиток на початку 2000-х, дана система з'явилася в результаті розвитку HD, а сам термін був узятий з бібліотеки ITIL, яка описує ідею управління IT як послугою. Відповідно до бібліотекою ITIL ServiceDesk являє собою єдину точку для взаємодії провайдерів послуг і споживачів. Основне завдання цієї служби управляти інцидентами і заявками на надання сервісної допомоги, а також надавати зворотний зв'язок користувачеві.

Існує 6 основних відмінностей між ServiceDesk і HelpDesk(далі SD та HD):

- HD створювалася як IT-служба для вирішення внутрішніх складнощів, а SD є службою, розвиваючу ідею надання IT як послуги;
- HD допомагає клієнтам, SD надає їм обслуговування, тобто переслідує більш широкі цілі, які між тим тісно пов'язані з обслуговуванням споживачів;
- HD фокусується на вирішенні інцидентів, а SD крім цього допомагає в управлінні заявками на обслуговування і надання інформації;
- HD носить технічний характер і використовується для досягнення однієї конкретної мети, в той час як SD – це частина стратегічної концепції, на якій ґрунтується робота IT-інфраструктури;
- SD – це доповнена HD, що увібрала від свого «попередника» основні принципи надання технічної підтримки, попередньо їх упорядкувавши. У новій службі наведено лад в реєстрації заявок, їх категоризації і встановленні пріоритетності виконання, також в цій системі можна відстежувати всі етапи виконання запитів.[3]<sup>1)</sup>

Таким чином, термін SD більш зрозумілий споживачам, який розуміє і сприймає поради бібліотеки ITIL, в той час як HD є простим інструментом надання технічної підтримки, а не окремим сервісом.

---

<sup>1)</sup> [3] Helpdesk или ServiceDesk: 6 основных отличий. URL: <https://it-guild.com/info/blog/6-razlichiy-mezhdu-helpdesk-i-servicedesk> (дата звернення: 01.11.2020)

Основним компонентом будь-якого HelpDesk-рішення є система управління запитами (або інцидентами, тікетами, багами). В англійській мові такі системи називають "issue tracker".

Системи HelpDesk забезпечують:

- єдину точку звернення до служби підтримки;
- зручний і зрозумілий для користувачів механізм дозволить направляти запити до служби підтримки, минаючи менш ефективні способи вирішення проблем (спроби вирішити самостійно або за допомогою колег, звернення до першого ліпшого співробітнику служби ІТ, навіть якщо той не зобов'язаний займатися підтримкою).
- Стандартний спосіб реєстрації і видачі завдань фахівцям;
- контроль за послідовністю виконання робіт, витраченим часом і ресурсами;
- призначення пріоритетів запитам залежно від типу запиту, конкретного користувача або інших обставин;
- ескалація запитів і інцидентів, сповіщення відповідних адміністраторів;
- зберігання бази знань по минулим запитам, що дозволяє фахівцям швидко вирішувати проблеми, схожі з уже виникали;
- звітність за витратами часу і коштів на виконання запитів.

Під час отримання запиту від клієнта (по телефону, по email, через веб-сайт), в системі створюється "тікет". Більшість сучасних helpdesk вміють створювати тікети з email і web-запитів автоматично. Цей тікет, в залежності від його змісту і важливості, ставиться в чергу одному зі співробітників служби підтримки і той вже працює з клієнтом над вирішенням проблеми. В ході цього процесу статус тікета оновлюється і керівник служби підтримки може контролювати, як співробітники саппорта справляються з навантаженням.

Говорячи про спільну роботу, у HD систем широкі можливості по розподілу прав, наприклад, можуть бути співробітники які мають доступ до

певного типу заявок – інші їм не доступні. Або поділ може відбуватися по департаментам, наприклад службі технічної підтримки зовсім не обов'язково бачити заявки які відносяться до обговорення договорів.

Крім підтримки клієнтів, HD отримали широке застосування в ІТ-відділах великих компаній для підтримки внутрішніх клієнтів (співробітників) і для управління ІТ інфраструктурою.

У терміні helpdesk відображена спрямованість служби технічної підтримки на допомогу користувачам. У компанії можуть бути ІТ-фахівці, що відповідають за різні аспекти підтримки ІТ-ландшафту. Однак користувачеві не потрібно знати зону відповідальності кожного співробітника. Він звертається до фахівця служби helpdesk за загальними для всіх каналів. Співробітник приймає заявку, деталізує опис проблеми, визначає її пріоритет і можливі шляхи вирішення. Якщо проблема проста, фахівець helpdesk може розв'язати цю проблему сам. У складних випадках співробітник підтримки може перепризначити заявку своїм колегам, які будуть працювати з нею далі. Перенаправлення заявок в helpdesk відбувається непомітно для користувача. Ефективно організований інтерфейс і система нагадувань в таких helpdesk системах дозволяють мінімізувати час роботи за кожною заявкою.

Основне завдання Helpdesk – технічна підтримка: обробка запитів на обслуговування, управління інцидентами і рішення проблем. Helpdesk-система – «єдине вікно», через який користувачі отримують необхідну допомогу.

Впровадження Helpdesk допомагає впорядкувати і прискорити роботу зі зверненнями користувачів, правильно розставляти пріоритети і категорії заявок, оптимально розподіляти завдання між виконавцями і підвищувати задоволеність клієнтів.

Helpdesk організовує єдину точку звернення в ІТ-службу. Користувачеві не потрібно розбиратися в суті проблеми, щоб передати її конкретного фахівця або вирішити самотійно. Підтримка користувачів стає впорядкованою, швидкою і ефективною.

Helpdesk-система спрощує і прискорює роботу співробітників техпідтримки. У заявці фахівець бачить всю необхідну інформацію про запит або інцидент, а також про користувача. Служба Helpdesk позбавляється від більшої частини операцій, які зазвичай виконуються вручну.

Впровадження Helpdesk на базі багатофункціональної платформи автоматизації ІТ-процесів може стати першим кроком до переходу на модель надання ІТ як послуги.

Головне правило при впровадженні SD системи – послідовність. Потрібно розробити чіткий план, що враховує особливості ІТ-процесів в компанії: їх поточний рівень зрілості, больові точки, організаційно-функціональну структуру та стратегічні плани розвитку. Оскільки підрозділ технічної підтримки в кожній компанії має свої особливості, кожен напрямок має курируватися відповідальними за напрямок особами. Це дозволить уникнути виникнення непередбачених збоїв в роботі програмного комплексу, в тому числі допоможе уникнути багатьох проблем на ранніх стадіях впровадження автоматизованої системи обслуговування користувачів в організації [4]<sup>1)</sup>.

Внаслідок впровадження сучасних систем автоматизації обслуговування користувачів виходить досягти значних результатів. За рахунок них ІТ-департамент виходить на новий рівень розвитку, підвищується його значимість і імідж в компанії. Забезпечується це за рахунок наступного:

- Відбувається зменшення операційних витрат. З'являється можливість значного скорочення витрат на персонал. При цьому підвищується швидкість, якість обробки заявок, виключається ймовірність появи збоїв через людський фактор.

---

<sup>1)</sup> [4] Зачем внедряют HelpDesk в компании. URL: <https://vc.ru/services/45344-zachem-vnedryat-helpdesk-v-kompanii> (дата звернення: 01.11.2020)

- Забезпечується дотримання встановлених норм з обслуговування, в тому числі по SLA. Це обумовлює підвищення якості і швидкості обслуговування.
- Відкриваються нові перспективи зростання, розвитку за рахунок підвищення ефективності та розробці заходів щодо оптимізації діяльності ІТ.

### **3 ПОРІВНЯННЯ ІСНУЮЧИХ OPEN SOURCE HELPDESK/SERVICEDESK СИСТЕМ**

В наш час, існує безліч різноманітних open source систем HelpDesk та ServiceDesk. Завдяки тому, що дані системи з відкритим кодом – не має потреби створювати щось нове з самого початку, оскільки можна допрацювати вже існуючу систему.

Підтримка клієнтів є вирішальним аспектом для будь-якого бізнесу. Бізнес не може існувати без клієнтів, і обслуговування клієнтів повинно бути пріоритетом кожного бізнесу. Клієнтам потрібна допомога для прийняття економічно ефективних рішень та правильного використання продуктів. Вони вимагають детальної інформації та навчання щодо встановлення, обслуговування, модернізації та утилізації продуктів. Підприємства повинні швидко реагувати на запити, занепокоєння та скарги клієнтів для задоволення та утримання клієнтів. Більшість компаній пропонують своїм клієнтам підтримку за допомогою різних методів, таких як миттєві SMS та обмін повідомленнями, безкоштовні номери та електронні листи. Вони також наймають представника довідкової служби, який пропонує персоналізовану допомогу клієнтам компанії.

Програмне забезпечення довідкової служби – це інтегрований набір, який допомагає автоматизувати обслуговування клієнтів і дозволяє компаніям реагувати на запитання та запити клієнтів у найкоротші терміни.

На сьогоднішній день, найкращим безкоштовним програмним забезпеченням довідкової служби вважається: Fresh Desk Free, HelpSpot, Teamwork Desk, Mojo Helpdesk, Solarwinds Helpdesk, C Desk, HelpDeskZ, Fave.

Серед найкращого програмного забезпечення довідкової служби з відкритим кодом, відокремлюють наступні системи: eTicket, OTRS, osTicket, Brimir, Request Tracker, GLPI, MantisBT, SimpleDesk, PHD Help Desk, Liberum, Bugzilla, Sinergia, Adefhelpdesk, UVdesk.

### 3.1 OTRS

OTRS (аббр. Від англ. Open-source Ticket Request System) – відкрита система обробки заявок.

Безкоштовна open-source система тикетів (на Perl). Відмінна інтеграція з Email. Підтримує різні платформи, БД, LDAP. Автоматизовані багато дрібниць. Гнучка система звітів, великі можливості пошуку, управління користувачами. Можлива інтеграція з уже наявними БД клієнтів та співробітників. Система легко розширяема через додаткові модулі: база знань / FAQ, календар, файловий менеджер, ITSM і ін. [5]<sup>1)</sup>

OTRS – найсуворіша, функціональна, лаконічна і ділова з перерахованих: мінімум «примочок», максимум користі. OTRS Community Edition – тикет система з відкритим вихідним кодом, заснована на системі OTRS. Це означає, що ви зможете допрацьовувати систему так, як вам потрібно (ну або замовляти доопрацювання). Перший реліз відбувся 18 років тому, система стабільна, перевірена часом, використовується багатьма великими і не дуже фірмами. Ще у OTRS є важлива перевага – розвинене міжнародне ком'юніті, готове допомогти, підказати, дати сотню посилань на GitHub і, в разі необхідності, попрацювати з конфігурацією за плату. Всі ми знаємо, наскільки важливо ком'юніті для такого ПО і наявність великої спільноти вже багато що говорить про саму систему.

У тикет-системі OTRS сть облік клієнтів, власне ведення тикетів, календар, звітність, дашборда, панель адміністратора. Тикети мають скромний інтерфейс, але в них є все, що необхідно: тіло тикета, пріоритет в черзі, статус, функції делегування і ескалації і т.д. Є шаблони, підказки, чат. [6]<sup>2)</sup>

---

<sup>1)</sup> [5] OTRS на русском – обзоры, отзывы. URL: <https://www.helpdeski.ru/tool/851/> (дата звернення: 02.11.2020)

<sup>2)</sup> [6] Тикет-системы: как бесплатная OTRS три платных уделала? Блог компании RUVDS.com. URL: <https://habr.com/ru/company/ruvds/blog/483418/> (дата звернення: 02.11.2020)

### 3.2 osTicket

osTicket – це широко використовувана система заявок на підтримку з відкритим вихідним кодом. Він легко інтегрує запити, створені за допомогою електронної пошти та веб-форм, в простий і зручний багатокористувацький веб-інтерфейс. osTicket створена, щоб спростити прийом заявок і поліпшити ефективність обслуговування клієнтів, надаючи всі необхідні засоби для швидкої та ефективної підтримки.

Можливості системи osTicket:

- користувацькі форми, теми та довідки;
- тема довідки, яка налаштовується;
- портал підтримки клієнтів;
- внутрішні замітки щодо квитків;
- правила маршрутизації вхідних квитків;
- агент запобігання зіткнень;
- форматований текст або електронна пошта HTML;
- угоди про рівень обслуговування;
- налаштування автоматичних відповідей;
- призначення і перенесення квитків. [7]<sup>1)</sup>

Особливості osTicket:

- підтримка через веб і по електронній пошті: Тікети можуть бути створені по електронній пошті, через онлайн-форми або по телефону. Гнучка настройка і відображення;
- автовідповідач: автоматична відповідь, яка відправляється при відкритті нового тикета або отриманні повідомлення;
- внутрішні замітки: додавання внутрішніх заміток до тикети для персоналу;

---

<sup>1)</sup> [7] OSTicket – обзор, плюсы и минусы программы, отзывы пользователей. URL: <https://8d9.ru/program/osticket> (дата звернення: 02.11.2020)



- розділи довідки: настроюються розділи довідки для веб-тікетів;
- сигнали та сповіщення. Персонал і клієнти постійно отримують повідомлення по електронній пошті. Конфігуровані і гнучкі настройки;
- рольовий доступ: контроль рівня доступу персоналу на основі груп і відділів.
- призначення і передача тікетів: призначення тікетів персоналу і / або відділу;
- історія підтримки: Всі запити в службу підтримки і відповіді архівуються.

З недоліків можна виділити:

- незручний інтерфейс;
- відсутня можливість створити свої шаблони листів через форму;
- відсутність вбудованого захисту від спаму.

### **3.3 Request Tracket**

Request Tracker – система обліку і відстеження заявок рівня підприємства з відкритим вихідним кодом, що дозволяє управляти завданнями, проблемами, і зовнішніми запитами від користувачів. Система написана на об'єктно-орієнтованій мові Perl.

Система почала розроблятися з 1996, і використовується системними адміністраторами, співробітниками служб техпідтримки, IT-менеджерами, розробниками і маркетинговими відділами.

RT управляє ключовими завданнями, такими як ідентифікація, встановлення пріоритетів, призначення, рішення та повідомлення, необхідне важливими додатками підприємства, включаючи керівництво проектом,

сервісну службу, NOC ticketing, CRM і розробку програмного забезпечення. [8]<sup>1)</sup>

Request Tracker це відкрита система управління заявками, розроблена компанією Best Practical.

Завдяки цій системі нашої компанії вдається реалізовувати налагоджені бізнес-процеси обробки запитів клієнтів, а також ефективно управляти внутрішніми проектами. Система RT побудована на знайомих технологіях: Perl, MySQL, mod\_perl, Apache.

RT володіє розвиненими можливостями інтеграції з іншими корпоративними системами і дозволяє адаптувати функціональність під будь-які вимоги. [9]<sup>2)</sup>

### 3.4 GLPI

GLPI – аббревіатура для Gestionnaire libre de parc informatique (Вільний менеджер IT-інфраструктури), є системою роботи з заявками та інцидентами, а також для інвентаризації комп'ютерного обладнання (комп'ютери, програмне забезпечення, принтери і т.д.). Має розширені функції, такі як робота системи відстеження пошти з повідомленням і методи, для створення бази даних з основною інформацією про топологію мережі.

GLPI – це повноцінна open-source ITSM система для управління процесами, інвентаризації обладнання та софту, а також для управління заявками і інцидентами. GLPI це великий комбайн, в якому закладена велика кількість функціоналу з управління IT на підприємстві. Деякі з можливостей GLPI:

---

<sup>1)</sup> [8] Request Tracker – это... Что такое Request Tracker? URL: <https://dic.academic.ru/dic.nsf/ruwiki/1357331> (дата звернення: 02.11.2020)

<sup>2)</sup> [9] Технологии которые используют для разработки в Метод Лаб: Request Tracker. URL: <https://www.methodlab.ru/technology/rt.shtml> (дата звернення 02.11.2020)

- інвентаризація комп'ютерів, серверів, оргтехніки (принтери, монітори, МФУ, картриджі та інше);
- інвентаризація встановлених на комп'ютерах програм і ліцензій;
- управління заявками і інцидентами (реєстрація звернень користувачів, HelpDesk);
- база знань;
- ведення проектів;
- управління бюджетом. [10]<sup>1)</sup>

### 3.5 FreeScout

Додаток створено з упором на юзабіліті, простоту і зручність використання. FreeScout використовується і буде використовуватися приватними особами і компаніями, яким потрібна безкоштовна професійна система підтримки клієнтів по електронній пошті, але які не можуть дозволити собі Zendesk або Help Scout. Також FreeScout ідеально підходить тим, хто не хоче відмовлятися від приватності або потрапляти в залежність від сервісу, який вони не контролюють.

FreeScout супер «легкий» і швидкий, він не вимагає Node.js, Redis, memcached, composer або чого-небудь ще для роботи, тільки веб-сервер з PHP 7.0+ і MySQL (Maria DB) 5.0. Завдяки чому він може бути встановлений практично на будь-якому віртуальному хостингу.

Найбільш важливою частиною будь-якого хелпдеска є вікно написання відповіді, це серце всієї системи. З безлічі WYSIWYG-редакторів з відкритим вихідним кодом Summernote був обраний як найбільш підходящий і гнучкий. Summernote був створений ще в 2013 році і має величезне співтовариство на

---

<sup>1)</sup> [10] Установка и настройка GLPI и FusionInventory, инвентеризация ИТ инфраструктуры | Windows для системного администратора. URL: <https://winitpro.ru/index.php/2020/04/14/itsm-glpi-inventory/> (дата звернення: 03.11.2020)

GitHub (більше 1600 ФОРКОМ). Він підтримує з коробки всі функції, які були необхідні для хелпдеска. [11]<sup>1)</sup>

### 3.6 HESK

HESK – це безкоштовна HelpDesk система для підтримки користувачів написана на PHP. Як і багато php-програм, HESK Help Desk Software від PHPJunkyard безкоштовна.

Він легко налаштовується так, щоб він виглядав і відчував себе так само, як існуючий сайт, редагуючи стилі. Це те, що багато хто з платних програм не дозволяють або не передбачають.

Програма пропонує все, що потрібно всередині служби підтримки без будь-яких складнощів.

Головна сторінка пропонує два входи з різними рівнями безпеки: персонал і адміністратор. Використовуючи адміністративну обліковий запис, є можливість розробити основні настройки для всієї програми.

Служба підтримки Hesk пропонує необмежену кількість персоналізованих облікових записів для співробітників, необмежені категорії для категоризації квитків допомоги для бази знань, готових відповідей, вкладених файлів, повнофункціональну базу знань до статей, категоріями, функцією пошуку і багатьом іншим, що настраюються поля і захист від спаму. функції для квитків довідки, повідомлення по електронній пошті, система оцінки квитків і відповідей, функція автоматичного закриття для залишилися без відповіді квитків і мовної переклад.

Це програмне забезпечення поставляється з автоматично встановлюваним сценарієм, використовує базу даних MySQL, і було

---

<sup>1)</sup> [11] FreeScout: обзор, отзывы, аналоги, интеграция, сайт | BizzApps. URL: [https://bizzapps.ru/p/freescout/#:~:text=FreeScout%20—%20это%20бесплатная%20справочная%20служба,PHP7%20\(платформа%20Laravel%205.5](https://bizzapps.ru/p/freescout/#:~:text=FreeScout%20—%20это%20бесплатная%20справочная%20служба,PHP7%20(платформа%20Laravel%205.5) (дата звернення: 03.11.2020)

протестовано на серверах Linux, Windows і FreeBSD. Єдиними вимогами, необхідними для запуску цього скрипта на сервері, є база даних MySQL з php 4.3.2 або вище.

Можливості HelpDesk системи Hesk:

- запити від клієнта можуть надходити як з web інтерфейсу, електронної пошти або по телефону;
- є трекер заявок, можливо виставляти пріоритети;
- гнучка система категорій звернень;
- вбудована база знань для користувачів;
- підрахунок витраченого часу на квитки;
- гнучка система повідомлень;
- захист від спаму;
- звітність;
- є версія SaaS.

## **4 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ HELPDESK ВІДДІЛУ ТЕХНІЧНОЇ ПІДТРИМКИ UALINUX**

ІС будь-якого виду характеризуються життєвим циклом – від моменту виникнення ідеї розробки ІС до моменту відмови від використання ІС. Стадії життєвого циклу програм визначають склад і зміст робіт зі створення інформаційних систем.

При традиційній розробці програм розрізняють наступні етапи створення програмного продукту:

- аналіз вимог до проекту;
- проектування;
- реалізація;
- тестування продукту;
- впровадження та підтримка.

На етапі аналізу вимог до проекту формулюються цілі та завдання проекту, виділяються базові сутності та взаємозв'язку між ними. Тобто, створюється основа для подальшого проектування системи.

В рамках даного етапу не тільки фіксуються вимоги замовника, а й проводиться їх формування – клієнтам підбирається оптимальне вирішення їхніх проблем, визначається необхідний ступінь автоматизації, виявляються найбільш актуальні для автоматизації бізнес-процеси.

При аналізі вимог визначаються терміни і вартість розробки ПО, формується і підписується ТЗ на розробку програмного забезпечення.

На основі попереднього етапу проводиться проектування системи. Ця методологія проектування поєднує в собі об'єктну декомпозицію, прийоми уявлення фізичної, логічної, а також динамічної та статичної моделей системи.

Під час проектування розробляються проектні рішення щодо вибору платформи, де буде функціонувати система мови або мов реалізації, призначаються вимоги до призначеного для користувача інтерфейсу,

визначається найбільш підходяща СУБД. Розробляється функціональна специфікація ПЗ: вибирається архітектура системи, обумовлюються вимоги до апаратного забезпечення, визначається набір орг. заходів, які необхідні для впровадження ПО, а також перелік документів, що регламентують його використання.

Даний етап розробки ПЗ організований відповідно до моделей еволюційного типу життєвого циклу ПО. При розробці застосовуються експериментування і аналіз, будуються прототипи, як цілої системи, так і її частин. Прототипи дають можливість глибше вникнути в проблему і взяти всіх необхідних проектні рішення ще на ранніх етапах проектування. Такі рішення можуть зачіпати різні частини системи: внутрішню організацію, призначений для користувача інтерфейс, розмежування доступу і т.д. У результаті етапу реалізації з'являється робоча версія продукту.

Тестування тісно пов'язане з такими етапами розробки програмного забезпечення як проектування і реалізація. У систему вбудовуються спеціальні механізми, які дають можливість виробляти тестування системи на відповідність вимог до неї, перевірку оформлення і наявність необхідного пакету документації.

Результатом тестування є усунення всіх недоліків системи і висновок про її якість.

Впровадження системи зазвичай передбачає наступні кроки:

- установка системи,
- навчання користувачів,
- експлуатація.

До будь-якої розробки додається повний пакет документації, який включає в себе опис системи, керівництва користувачів і алгоритми роботи.

Підтримка функціонування ПО повинна здійснюватися групою технічної підтримки розробника.

## **4.1 Вимоги до проектованої інформаційної системи відділу технічної підтримки компанії UALinux**

### **4.1.1 Вимоги до структури системи**

Інформаційна система повинна мати в своєму складі наступні модулі:

- управління заявками. Облік заявок, відстеження і спільна робота. Відображення робіт в особистому кабінеті;
- управління завданнями. Оперативне управління роботою співробітників, відстеження, колективна взаємодія. Підтримка підзадач, регулярних завдань і тегів;
- облік часу. Листи обліку часу на підставі списаних годин в задачах і заявках;
- аналітичні звіти. Динаміка діяльності, задоволеність клієнтів, якість роботи служби підтримки та ін. В розрізі часу;
- мобільний доступ. При знаходженні поза офісом робота співробітників над заявками повинна бути доступна в повній мірі з підтримкою повідомлень про нові події.

### **4.1.2 Вимоги до функцій, виконуваних системою**

Інформаційна система обліку і контролю успішності повинна виконувати наступні функції:

- реєстрація заявок на виконання робіт, можливість створення заявки анонімно і без перевірки достовірності email-адреси;
- систематизація приходять заявок, в тому числі для призначення виконавців, категорій та пріоритетів;
- відстеження поточного статусу заявки;
- протоколювання робіт, виконуваних за заявкою, а також всіх внесених до неї змін;



- побудова звітів за обраними параметрами.

## 4.2 Вибір архітектури інформаційної системи

За способом організації групові та корпоративні інформаційні системи підрозділяються на наступні класи:

- системи на основі архітектури файл-сервер;
- системи на основі архітектури клієнт-сервер;
- системи на основі багаторівневої архітектури;
- системи на основі технології інтернет / інтранет.

В будь-якій інформаційній системі можна виділити необхідні функціональні компоненти, які допомагають зрозуміти обмеження різних архітектур інформаційних систем (таблиця 3.1).

Таблиця 3.1 – Типові функціональні компоненти інформаційної системи.

Найменування	Характеристика
Presentation Services (засоби представлення)	Забезпечуються пристроями, які приймають введення від користувача і відображають результати обробки.
Presentation Logic(логіка представлення)	Управляє взаємодією між користувачем і ЕОМ. Обробляє дії користувача при виборі команди в меню, натисканні кнопки або виборі елемента зі списку.
Business or Application Logic (прикладна логіка)	Набір правил для прийняття рішень, обчислень і операцій, які має виконати додаток.

Продовження таблиці 3.1

Data Logic (логіка управління даними)	Операції з базою даних (SQL-оператори), які потрібно виконати для реалізації прикладної логіки управління даними.
Data Services (операції з базою даних)	Дії СУБД, що викликаються для виконання логіки управління даними, такі як: маніпулювання даними, визначення даних, фіксація або відкат транзакцій і т. П. СУБД зазвичай компілює SQL-пропозиції.
File Services (файлові операції)	Дискові операції читання і запису даних для СУБД (файлові операції) і інших компонентів. Зазвичай є функціями операційної системи (ОС).

Розглянемо більш докладно особливості варіантів побудови інформаційних додатків. Поділ інформаційних систем за класами здійснюється на основі розташування функціональних компонент. Можна виділити необхідні функціональні компоненти, які допомагають зрозуміти обмеження різних архітектур інформаційних систем. Розглянемо більш докладно особливості варіантів побудови інформаційних додатків.

За способом організації інформаційні системи поділяються наступним чином:

- системи на основі архітектури файл-сервер;
- системи на основі архітектури клієнт-сервер;
- системи на основі багаторівневої архітектури.

Архітектура "файл-сервер" передбачає концентрацію обробки на робочих станціях. Основною перевагою цього варіанту є простота і відносна дешевизна. Подібне рішення прийнятно, поки число користувачів, що одночасно працюють з базою даних, не перевищує 5-10 чоловік. При

збільшенні кількості користувачів система може "захлинутися" через перевантаженість ЛВС великими потоками необробленої інформації.

Сервер, як правило, – найпотужніший і самий надійний комп'ютер. Він обов'язково підключається через джерело безперебійного живлення, в ньому передбачаються системи подвійного або навіть потрійного дублювання. В особливо відповідальних випадках можна підключити разом кілька серверів так, що при виході з ладу одного з них в роботу автоматично включиться "дублер". Таким чином, при концентрації обробки даних на сервері надійність системи в цілому обмежується лише матеріальними засобами, які замовники готові вкласти в технічне оснащення.

Рішення по автоматизації обліку і управління в корпоративних структурах передбачає розподілену обробку даних, організацію паралельних обчислень, глибоке розмежування рівнів доступу, можливість вибору різних операційних систем і серверних платформ. Якщо бізнес не великий, подібне рішення оптимально.

В архітектурі «клієнт-сервер» беруть участь дві сторони: клієнт і сервер. Тут все як у житті: клієнт – це замовник тієї чи іншої послуги, а сервер – постачальник послуг. Клієнт і сервер фізично представляють собою програми, наприклад, типовим клієнтом є браузер. Як сервер можна навести такі приклади: всі HTTP сервера (зокрема Apache), MySQL сервер, локальний веб-сервер AMPPS або готова збірка Denwer (останніх два приклади – це не проста сервера, а цілий набір серверів).

Клієнт і сервер взаємодію один з одним в мережі Інтернет або в будь-якій іншій комп'ютерної мережі за допомогою різних мережевих протоколів, наприклад, IP протокол, HTTP протокол, FTP та інші. Протоколів насправді дуже багато і кожен протокол дозволяє надавати ту чи іншу послугу. Наприклад, за допомогою HTTP протоколу браузер відправляє спеціальне HTTP повідомлення, в якому зазначено яку інформацію і в якому вигляді він хоче отримати від сервера, сервер, отримавши таке повідомлення, відсилає браузеру у відповідь схоже по структурі повідомлення (або кілька

повідомлень), в якому міститься потрібна інформація, як правило це HTML документ.

Повідомлення, які посилають клієнти отримали назви HTTP запити. Запити мають спеціальні методи, які говорять сервера про те, як обробляти повідомлення. А повідомлення, які посилає сервер отримали назву HTTP відповіді, вони містять крім корисної інформації ще й спеціальні коди стану, які дозволяють браузеру дізнатися те, як сервер зрозумів його запит.

Також варто зауважити, що в основі взаємодії клієнт-сервер лежить принцип того, що така взаємодія починає клієнт, сервер лише відповідає клієнту і повідомляє про те чи може він надати послугу клієнтові і якщо може, то на яких умовах. Клієнтське програмне забезпечення та серверне програмне забезпечення зазвичай встановлено на різних машинах, але також вони можуть працювати і на одному комп'ютері.

Дана концепція взаємодії була розроблена в першу чергу для того, щоб розділити навантаження між учасниками процесу обміну інформацією, а також для того, щоб розділити програмний код постачальника і замовника. Спрощена схема взаємодії клієнт-сервер можна побачити на рис. 3.1.

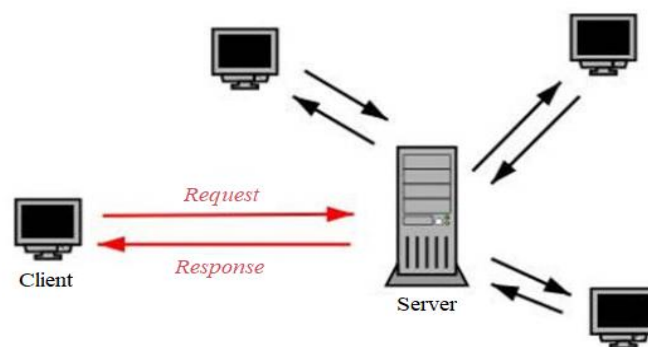


Рисунок 3.1 – Спрощена схема взаємодії клієнт-сервер

Багаторівнева архітектура одна з найпоширеніших архітектур. На її основі побудовано безліч великих фреймворків – Java EE, Drupal, Express. Мабуть, найвідоміший приклад цієї архітектури – це мережева модель OSI.

Система ділиться на рівні, кожен з яких взаємодіє лише з двома сусідніми. Тому запити до БД, яка зазвичай розташовується в самому кінці ланцюжка взаємодії, проходять послідовно крізь кожен «шар».

Архітектура не має на увазі якусь обов'язкову кількість рівнів – їх може бути три, чотири, п'ять і більше. Найчастіше використовують триланкову системи: з рівнем уявлення (клієнтом), рівнем логіки і рівнем даних.

Багаторівнева архітектура стала розвитком архітектури клієнт-сервер і в класичній формі складається з трьох рівнів:

- нижній рівень являє собою додатки клієнтів, виділені для виконання функцій і логіки уявлень PS і PL і мають програмний інтерфейс для виклику програми на середньому рівні;
- середній рівень являє собою сервер додатків, на якому виконується прикладна логіка BL і з якого логіка обробки даних DL викликає операції з базою даних DS;
- верхній рівень являє собою віддалений спеціалізований сервер бази даних, виділений для послуг обробки даних DS і файлових операцій FS (без використання збережених процедур). [12]<sup>1)</sup>

Спрощена схема рівнів багаторівневої архітектури можна побачити на рис. 3.2.

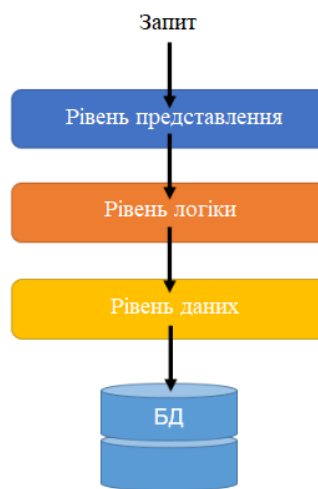


Рисунок 3.2 – Спрощена схема рівнів багаторівневої архітектури

---

<sup>1)</sup> [12] Зіставлення інформаційних систем з традиційними програмними продуктами – Лекції – Проектування інформаційних систем. URL: <https://uadoc.zavantag.com/text/2813/index-1.html?page=2> (дата звернення: 22.11.2020)

Трирівнева архітектура дозволяє ще більше збалансувати навантаження на різні вузли і мережу, а також сприяє спеціалізації інструментів для розробки додатків і усуває недоліки дворівневої моделі клієнт-сервер.

Централізація логіки додатка спрощує адміністрування і супровід. Чітко поділяються платформи та інструменти для реалізації інтерфейсу і прикладної логіки, що дозволяє з максимальною віддачею реалізовувати їх фахівцями вузького профілю. Нарешті, зміни прикладної логіки не зачіпають інтерфейс, і навпаки. Але оскільки межі між компонентами PL, BL і DL розмиті, прикладна логіка може з'явитися на всіх трьох рівнях. Сервер додатків за допомогою монітора транзакцій забезпечує інтерфейс з клієнтами та іншими серверами, може управляти транзакціями і гарантувати цілісність розподіленої бази даних. Засоби віддаленого виклику процедур найбільш відповідають ідеї розподілених обчислень: вони забезпечують з будь-якого вузла мережі виклик прикладної процедури, розташованої на іншому вузлі, передачу параметрів, віддалену обробку і повернення результатів. З ростом систем клієнт-сервер необхідність трьох рівнів стає все більш очевидною.

Таким чином, багаторівнева архітектура розподілених додатків дозволяє підвищити ефективність роботи корпоративної інформаційної системи і оптимізувати розподіл її програмно-апаратних ресурсів. Але поки на російському ринку як і раніше домінує архітектура клієнт-сервер.

Інтернет/Інтранет-технології. У розвитку технології інтернет-інтранет основний акцент поки що робиться на розробці інструментальних програмних засобів. У той же час спостерігається відсутність розвинених засобів розробки додатків, що працюють з базами даних. Компромісним рішенням для створення зручних і простих у використанні і супроводі інформаційних систем, ефективно працюють з базами даних, стало об'єднання Інтернет / Інтранет-технології з багаторівневою архітектурою. При цьому структура інформаційного додатка набуває такого вигляду: браузер – сервер додатків – сервер баз даних – сервер динамічних сторінок – web-сервер.

Завдяки інтеграції Інтернет / Інтранет-технології та архітектури клієнт-сервер процес впровадження і супроводу корпоративної інформаційної системи істотно спрощується при збереженні досить високої ефективності і простоти спільного використання інформації.

В якості постійного сховища даних використовується реляційна база даних. У базу даних поміщаються тільки дані у вигляді набору реляційних сутностей (пов'язаних таблиць) без елементів програмованої логіки (тригерів, збережених процедур, уявлень і т.п.). Система може використовувати сервер, який задовольняє певним вимогам до функціональності.

Сервер програми реалізується у вигляді WEB-додатки, яке може виконуватися в контекстесреди виконання, працюючого наприклад, наJava і реалізує стандарти Java-сервлетов. Для зниження навантаження на сервер додатків за рахунок кешування переданого контенту, а також реалізації додаткових функцій, таких як Virtual Hosts, SSL перед сервером додатків може проводитися установка одного з популярних Web-серверів: Apache, nginx, MS IIS.

### **4.3 Архітектура інформаційної системи**

Почнемо проектування інформаційної системи з опису ролей користувачів, які будуть створені при реалізації, і варіанти використання системи для цих ролей.

В системі передбачені наступні ролі:

**Адміністратор.** Відповідає за налаштування системи, реєстрацією нових користувачів, видачу їм прав і ролі, наповненням довідників. Має доступ до всіх об'єктів системи.

**Ініціатор.** Створює нову заявку, має доступ на перегляд своїх відкритих заявок, додає коментарі до своїх відкритим заявками, а також змінює статус. Після створення заявки її статус стає «Відкрита». Ініціатор може закрити свою заявку.

Виконавець. Виконавець може змінювати статус заявки на «Виконується» і «На перевірці», вносити коментарі. Отримавши або відредагувавши заявку, виконавець змінює її статус на «Виконується». Виконавши заявку, виконавець змінює її статус на «На перевірці».

Керівник. Якщо у виконавця є прапор керівника, то він може також змінювати пріоритет, крайній термін, а також призначати інших виконавців і класифікувати заявку.

#### **4.4 Вибір операційної системи**

Необхідно вибрати ОС, під управлінням якої буде працювати сервер. Серверна операційна система – призначена для управління програмним забезпеченням, яке в свою чергу обслуговує всіх користувачів мережі, як внутрішньої, так і зовнішньої. Серверна ОС надає спеціалізований функціонал управління, так би мовити більш зручний, ефективний, розроблений спеціально для конкретних цілей адміністрування та управління і такого функціоналу просто немає в призначених для користувача операційних системах, які, до речі, виступають клієнтами цих серверних операційних систем. До таких додатків ставляться сучасні системи управління базами даних, засоби управління мережами і аналізу подій в мережі, служби каталогів, засоби обміну повідомленнями і групової роботи, Web-сервери, поштові сервери, корпоративні брандмауери, сервери додатків найрізноманітнішого призначення, серверні частини бізнес-додатків. Вимоги до продуктивності і надійності вказаних операційних систем дуже високі; нерідко сюди входять і підтримка кластерів (набору ряду однотипних комп'ютерів, що виконують одну і ту ж задачу і ділять між собою навантаження), і можливості дублювання і резервування, і переконфігурації програмного і апаратного забезпечення без перезавантаження операційної системи.

Серверна операційна система, на відміну від користувача, надає більш широкий функціонал для управління ІТ-інфраструктурою, програмним



забезпеченням, встановленим не тільки на сервері, а й на комп'ютерах користувачів мережі, причому як внутрішньої, так і зовнішньої. Основна візуальна відмінна риса будь-серверної операційної системи – це можливість працювати без графічної оболонки (звичного більшості користувачів ПК інтерфейсу ОС). Наявність графічної оболонки операційної системи вимагає більшої кількості апаратних ресурсів. Отже, серверні ОС, розраховані на експлуатацію фахівцями або, як мінімум, «просунутими» користувачами, допускають можливість працювати без графічної оболонки, тобто безпосередньо з файлового менеджера. Багато сервери безпосередньо на етапі виробництва «заточуються» під певні операційні системи. Так, наприклад, сервери HP мають сертифікацію Microsoft, а також відповідний набір інструментів для швидкого розгортання операційної системи і швидкої інтеграції серверного обладнання в IT-інфраструктуру підприємства.

Вибір серверної операційної системи і апаратної платформи для неї в першу чергу залежить від того, які програми під її управлінням повинні виконуватися (як мінімум, вибрані додатки повинні існувати в версії для даної платформи) і які вимоги пред'являються до її продуктивності, надійності і доступності. [13]<sup>1)</sup>

Розглянемо серверні операційні системи.

#### **4.4.1 Windows Server 2019**

Windows Server являє собою групу операційних систем, розроблених корпорацією Майкрософт, яка підтримує управління на рівні підприємства, зберігання даних, додатків і комунікацій. Попередні версії Windows Server зосередилися на стабільності, безпеки, мереж, а також різні поліпшення в файлової системі. Інші поліпшення також включали поліпшення розгортання

---

<sup>1)</sup> [13] Серверные операционные системы ведущих производителей | Компьютер Пресс. URL: <https://compress.ru/article.aspx?id=12137> (дата звернення: 23.11.2020)

технологій, а також розширення підтримки апаратних засобів. Microsoft також створила спеціалізовані SKUs з Windows Server, які зосереджені на внутрішньому і зовнішньому ринках малого бізнесу.

Говорячи про нововведення в Windows Server 2019, перш за все, варто відзначити новий інтерфейс адміністратора Windows Admin Center (раніше відомий як Project Honolulu), заснований на веб-технологіях і дозволяє безпосередньо з браузера управляти локальними і віддаленими серверами, в тому числі робочими станціями з Windows 10 на борту. За допомогою Windows Admin Center можна конфігурувати сервери і комп'ютери в мережі організації і за її межами, отримувати доступ до файлів, формувати термінальні RDP-сесії і підключатися до робочих столів, запускати PowerShell-скрипти, керувати сервісами, службами, віртуальними машинами, правами користувачів, сертифікатами і всілякими налаштуваннями обслуговуються систем.

Друга цікава особливість Windows Server 2019 – можливість запуску робочих столів, бінарних виконуваних файлів і bash-скриптів Linux без використання віртуалізованих середовищ. Реалізовано це за допомогою спеціальної прошарку Windows Subsystem for Linux (WSL) в ядрі ОС, яка трансліює системні виклики Linux в виклики Windows. Вперше підсистема WSL з'явилася в Windows 10 з випуском поновлення Fall Creators Update, тепер же працювати з оточенням Linux можна і в серверній версії операційної системи.

#### **4.4.2 FreeBSD**

Система розповсюдження програмного забезпечення Berkeley Software Distribution (BSD) була створена на початку 90-х років випускниками Університету Берклі (Каліфорнія). Розробники UNIX-подібної операційної системи 386BSD виклали вихідні у відкритий доступ, на їх основі були написані базові ОС:

- BSD / OS, комерційна версія;
- NetBSD, open-source;
- FreeBSD, open-source.

Комерційна версія більше не підтримується, а інші open-проекти успішно розвиваються. В даний час існують 4 проекти BSD з відкритими кодами. Кожен проект заснований на своєму власному ядрі; вони створені для різних цілей, але практично мало відрізняються один від одного.

Сімейство BSD:

- FreeBSD – проект націлений на створення простий в управлінні системи з високою продуктивністю.
- DragonFlyBSD – високопродуктивна масштабуемая ОС призначена для підтримки багатопроцесорної обробки даних (SMP), створена з проекту FreeBSD;
- NetBSD – підтримує максимальну переносимість коду для різних обчислювальних ресурсів; підтримує застаріле обладнання; цю ОС використовували в космічних проектах NASA.
- OpenBSD – надійна ОС з підвищеним рівнем безпеки створена на базі проекту NetBSD; її встановлюють в банках і державних установах США.

На FreeBSD додатки можна встановити двома способами:

- за допомогою диспетчера пакетів (починаючи з версії 9.1 за замовчуванням пропонується нова, більш гнучка реалізація пакетів, практично не відстає від версій в портах);
- з використанням колекції портів.

На FreeBSD працюють одні з найбільших і завантажених сайтів в Інтернеті (наприклад, Yahoo!) і велика частка всіх інших сайтів також працюють на FreeBSD.

Але FreeBSD можна використовувати не тільки як платформу для Інтернет-сервера, але і в якості звичайного сервера, який виконує всі ті завдання, які виконують інші серверні операційні системи і обмежувати

можливості FreeBSD не потрібно. FreeBSD відрізняється високою безпекою і продуктивністю.

### 4.4.3 Ubuntu Server

Ubuntu Server – це безкоштовна серверна операційна система на базі ядра Linux. Ubuntu Server можна використовувати в якості платформи для Web-серверів, серверів баз даних, DNS-серверів, файлових серверів та інших типів серверів. Основний упор в Ubuntu робився на зручність і простоту використання, щоб кожен бажаючий без проблем міг познайомитися з операційною системою на базі Linux і почати її використовувати в повсякденному житті. Кодова база Убунту базується на Debian і не може існувати без неї. Спочатку вона планувалася як тимчасове відгалуження Debian, яке потім перетворилося в окрему операційну систему.

Ubuntu Server якоюсь особливою простотою і зручністю, в порівнянні з іншими серверними збірками не володіє, так що особливо desktop версії, де упор йде на зручне графічне оточення, не застосовні до серверного варіанту. Серверна система володіє іншими відмітними ознаками, які я перерахую нижче окремо у відповідному розділі.

На сьогоднішній момент Ubuntu Server не просто відгалуження десктопної гілки системи, а повноцінний серверний дистрибутив рівня Enterprise. І підтверджують це такі ознаки:

- На базі ubuntu server будується і підтримується робота хмарної інфраструктури на основі технологій openstack.
- Компанія Canonical, розробник Убунту, розвиває і підтримує сумісність з сучасною системою оркестрації контейнерів Kubernetes, яку можна будувати на базі обговорюваного серверного дистрибутива. Так само приділяється пильна увага роботи контейнерів docker, нові версії якої в першу чергу оптимізуються і тестуються під ubuntu server.

- Йде активне впровадження ubuntu в інтернет речей, для чого і була оголошена збільшена підтримка в 10 років для серверних версій. Це актуально насамперед для інтернету речей.
- Компанія Canonical забезпечує за оплату цілодобову технічну підтримку своєї системи і продуктів, заснованих на ній.[14]<sup>1)</sup>

#### 4.4.4 Solaris

ОС Solaris 10 відноситься до сімейства UNIX систем і позиціонується виробником як операційна система, оптимізована для роботи на основних серверних платформах. Вона побудована з урахуванням вимог, що пред'являються до критично важливих інформаційних середовищ високої готовності. Масштабованість Solaris охоплює широкий спектр системних конфігурацій – від ПК до суперкомп'ютерів, забезпечуючи гнучкість у виборі апаратної платформи з урахуванням конкретного завдання. Завдяки підтримці симетричного багатопроцесування і багатопоточності досягається прийнятний час відгуку додатків і висока пропускна здатність в багатокористувацьких операційних середовищах. Крім того, Solaris працює на найбільш популярних RISC- і CISC-платформах, включаючи SPARC, x86 і AMD64 / EM64T, тобто забезпечує незалежність від апаратної платформи і надає можливість вибору. Оскільки всі реалізації Solaris спираються на єдині вихідні тексти, то і користувачі, і адміністратори, і розробники мають право розраховувати на однакове поведінка ОС на всіх платформах.

Так як Solaris 10 є системою UNIX сімейства, вона володіє більшістю особливостей систем даного сімейства. Але, для того щоб зайняти лідируючі позиції на ринку серверних операційних систем компанія Sun Micro Systems

---

<sup>1)</sup> [14] Операційна система для сервера кращі. Серверні Операційні системи (ОС). Файловий сервер Linux. URL: <https://maylohack.ru/uk/android/operacionnaya-sistema-dlya-servera-luchshie-servernye.html> (дата звернення: 23.11.2020)

добре постаралася і забезпечила систему безліччю підтримуваних технологій і додаткових функцій.

Оперування зонами і контейнерами в ОС Solaris 10 допускає створення на одному сервері до 8192 «віртуальних» операційних систем, кожна з яких повністю автономна, зі своїми мережевими адресами, простором пам'яті, файлової системою, запущеними процесами і, зрозуміло, зі своєю базою користувачів. Цей набір незалежних віртуальних серверів Solaris 10, що працюють на одному фізичному сервері, – не що інше, як технологія віртуалізації в чистому вигляді, але реалізована повністю програмно.

У своєму дипломному проєкті я зупинився на Ubuntu Server, тому що він завоював репутацію завдяки ефективному використанню ресурсів, а також є партнером компанії UALinux.

## **4.5 Вибір системи управління базами даних**

Система управління базами даних (СКБД) – це комплекс програмно-мовних засобів, що дозволяють створити бази даних і управляти даними. Іншими словами, СУБД – це набір програм, що дозволяє організовувати, контролювати і адмініструвати бази даних. Більшість сайтів не можуть функціонувати без бази даних, тому СУБД використовується практично повсюдно.

### **4.5.1 Microsoft SQL Server**

Microsoft SQL Server – комерційна система керування бази даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для Запитів – Transact-SQL, Створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO відносно структурованої мови запитів SQL Із Розширення. Використовується як для невеликих и Середніх за розміром баз

Даних, так и для великих баз Даних масштабу підприємства. Багато років вдалині конкурує з іншими системами керування базами Даних [15]<sup>1)</sup>

SQL Server підтримує віддзеркалення і кластеризації баз даних. Кластер сервера SQL – це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Всі сервера мають одне віртуальне ім'я, і дані розподіляються по IP адресами машин кластеру протягом робочого циклу. Також в разі відмови або збою на одному з серверів кластера доступний автоматичний перенос навантаження на інший сервер.

SQL Server підтримує надлишкове дублювання даних за трьома сценаріями:

- Знімок: Проводиться «знімок» бази даних, який сервер відправляє одержувачам.
- Історія змін: Всі зміни бази даних безперервно передаються користувачам.
- Синхронізація з іншими серверами: Бази даних декількох серверів синхронізуються між собою. Зміни всіх баз даних відбуваються незалежно один від одного на кожному сервері, а при синхронізації відбувається звірка даних. Даний тип дублювання передбачає можливість вирішення протиріч між БД.

У SQL Server 2005 вбудована підтримка .NET Framework. Завдяки цьому, збережені процедури БД можуть бути написані на будь-якій мові платформи .NET, використовуючи повний набір бібліотек, доступних для .NET Framework, включаючи Common Type System (система поводження з типами даних в Microsoft .NET Framework). Однак, на відміну від інших процесів, .NET Framework, будучи базисною системою для SQL Server 2005, виділяє додаткову пам'ять і вибудовує засоби управління SQL Server замість

---

<sup>1)</sup>[15] Microsoft SQL Server – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server) (дата звернення 23.11.2020)

того, щоб використовувати вбудовані засоби Windows. Це підвищує продуктивність в порівнянні з загальними алгоритмами Windows, так як алгоритми розподілу ресурсів спеціально налаштовані для використання в структурах SQL Server.

#### 4.5.2 MySQL

MySQL є системою керування базами даних (СКБД). На сьогоднішній день це одна з найпопулярніших систем управління базами даних.

MySQL дозволяє зберігати цілочисельні значення зі знаком і беззнакові, довжиною в 1, 2, 3, 4 і 8 байтів, працює із строковими і текстовими даними фіксованої і змінної довжини, дозволяє здійснювати SQL-команди SELECT, DELETE, INSERT, REPLACE і UPDATE, забезпечує повну підтримку операторів і функцій в SELECT- і WHERE- частинах запитів, працює з GROUP BY і ORDER BY, підтримує групові функції COUNT (), AVG (), STD (), SUM (), MAX () і MIN (), дозволяє використовувати JOIN в запитах, в т.ч. LEFT OUTER JOIN і RIGHT OUTER JOIN, підтримує реплікацію, транзакції, роботу з зовнішніми ключами і каскадні зміни на їх основі, а також забезпечує багато інших функціональні можливості.[16]<sup>1)</sup>

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Є й інші типи таблиць, розроблені спільнотою.

---

<sup>1)</sup> [16] MySQL – система управления базами данных. URL: <https://web-creator.ru/articles/mysql> (дата звернення: 23.11.2020)



### 4.5.3 MariaDB

MariaDB – це популярний форк MySQL, створений оригінальними розробниками MySQL. Це вирросло з побоювань, пов'язаних з придбанням MySQL Oracle. Він пропонує підтримку як для невеликих завдань обробки даних, так і для корпоративних потреб. Він призначений для заміни MySQL, що вимагає простий деінсталяції MySQL і установки MariaDB. MariaDB пропонує ті ж функції MySQL і багато іншого.

MariaDB призначена для вільного доступу під ліцензією GNU GPL. Розробка та підтримка сайту MariaDB здійснює компанія MariaDB Corporation Ab і фонд MariaDB Foundation. Поштовхом до створення стала необхідність забезпечення вільного статусу СУБД, на противагу політиці ліцензування MySQL компанією Oracle. Засновниками проекту виступили початкові розробники MySQL. Система ліцензування MariaDB зобов'язує учасників, бажаючих додати свій код в основну гілку СУБД, обмінюватися своїми авторськими правами з MariaDB Foundation для охорони ліцензії і можливості створювати критичні виправлення для MySQL.

MariaDB намір підтримувати високу сумісність з MySQL, забезпечуючи точну відповідність з API і командами MySQL. До складу MariaDB включена підсистема зберігання даних XtraDB для можливості заміни InnoDB, як основний підсистема зберігання. Також включені підсистеми Aria, PBXT і FederateX.

У більшості випадків, MariaDB буде працювати ідентично MySQL: всі команди, інтерфейси взаємодії, бібліотеки і API, які вже існують для MySQL, реалізовані і в MariaDB. Немає необхідності конвертувати Ваші дані при перенесенні їх в MariaDB. MariaDB є повноцінною заміною MySQL!

Додатково MariaDB серйозно розширює доступний функціонал. Ви можете ознайомитися з нововведеннями і почати використовувати їх.

По-перше, це ряд нових двигунів (database engine) для зберігання даних. А саме: Aria – альтернатива таблиць MyISAM, швидша і стійка до збоїв.

Таблиці Aria використовуються в MariaDB для внутрішніх потреб, зокрема всі temporary tables працюють саме на движку Aria, за рахунок чого в ряді випадків вдалося досягти значно більшої продуктивності на складних запитах. По-друге, є функція, яка відрізняє його від своїх попередників, – це особливий акцент на безпеці. Коли виявлені критичні проблеми безпеки, наші інженери негайно готують і поширюють нову версію MariaDB якомога швидше. [17]<sup>1)</sup>

Крім того, в MariaDB відмінно масштабується і здатний обробляти десятки тисяч таблиць і мільярдів рядків даних. Він також може швидко і плавно керувати невеликими обсягами даних, що робить його зручним для невеликих підприємств або особистих проектів.

У своєму дипломному проекті в якості СУБД я зупинився на MariaDB, вона є кращим вибором для обраної мною HelpDesk системи.

---

<sup>1)</sup> [17] MariaDB – краткое руководство – CoderLessons. URL: <https://coderlessons.com/tutorials/bazy-dannykh/uznaite-mariadb/mariadb-kratkoe-rukovodstvo> (дата звернення: 24.11.2020)

## 5 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ HELPDESK ВІДДІЛУ ТЕХНІЧНОЇ ПІДТРИМКИ UALINUX

Для виконання поставленої задачі, було обрано скриптовану мову програмування PHP, серверну операційну систему Ubuntu Server, вже існуюча opensource HelpDesk система HESK, та система керування базами даних MariaDB.

Сформулюємо загальну структуру інформаційної системи Help Desk відділу технічної підтримки UALinux.

Головна сторінка технічної підтримки, яка включає в себе:

- сторінку створення запита;
- перевірка статусу існуючого запита;
- існуючу базу знань;
- авторизацію в систему.

Адмін панель, яка включає в себе наступні сторінки:

- сторінка з створеними запитами;
- сторінка з готовими шаблонами запитів;
- конфігурація бази знань;
- сторінка з категоризацією запитів;
- сторінка зі списком користувачів;
- сторінка зі звітом;
- сторінка з інструментами для адміністрування;
- налаштування системи.

Перше що побачить користувач, зайшовши на сайт технічної підтримки – це головну сторінку, яка зображена на рис. 5.1.

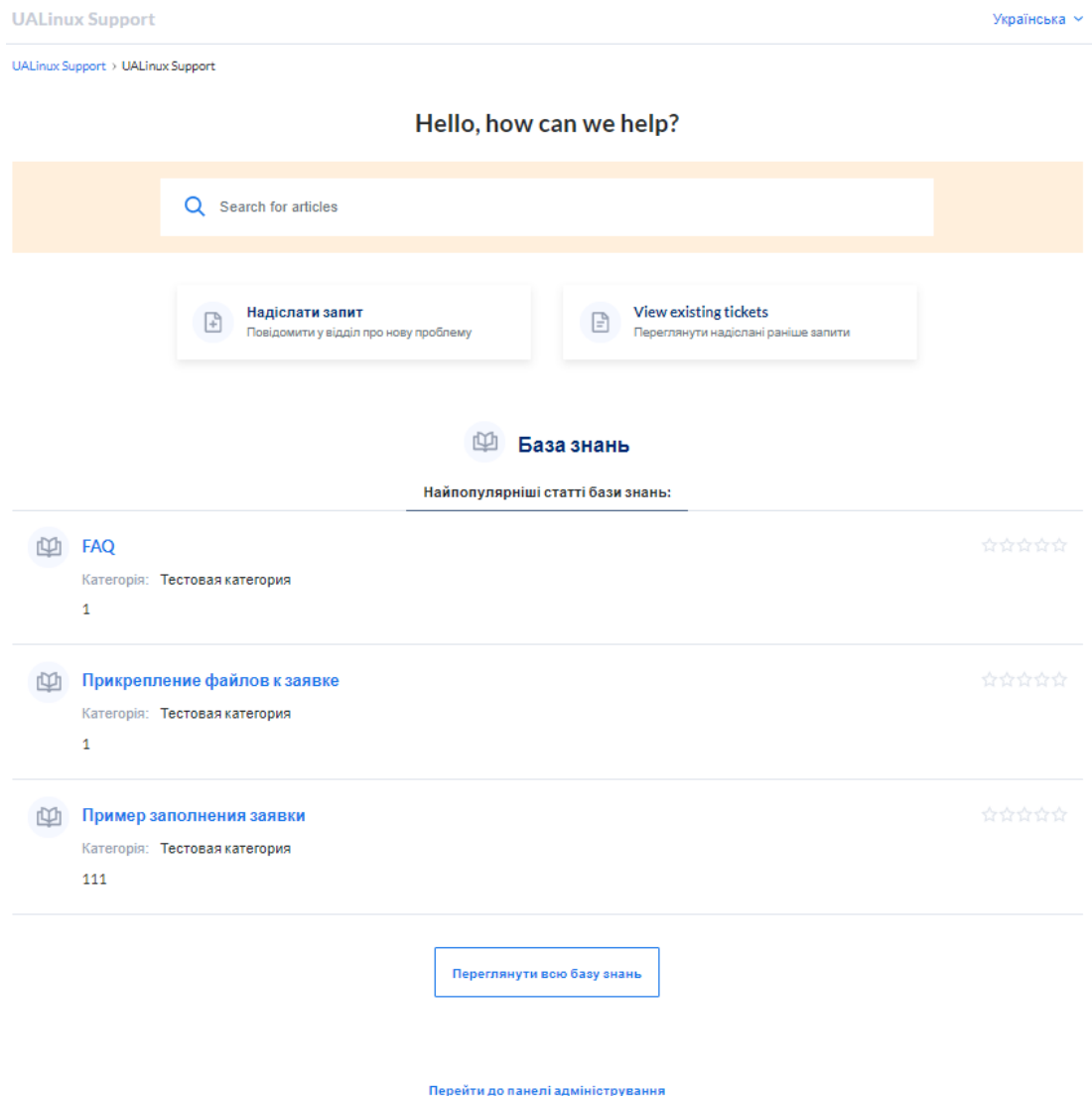



Рисунок 5.1 – Головна сторінка технічної підтримки

На ній можна побачити такі елементи як зміна мови, кнопка створення запиту, кнопка перегляду вже існуючих запитів, базу знань і посилання на сторінку авторизації в систему.

При натисканні на кнопку «Надіслати запит» користувача перенаправляє на сторінку створення запиту. Сторінка створення запита зображена на рис. 5.2.

 **Submit a Support Request**  
Обов'язкові поля позначено \*

Ел. адреса: \*

Пріоритет: \*  Низький ▾

---



Тема: \*

Повідомлення:

---

Вкладення:  Файл не вибран  
 Файл не вибран  
Maximum 2 attachments ⓘ

---

**SPAM Prevention**  
  
Введіть число на зображенні нижче. \* 

---

Рисунок 5.2 – Сторінка створення запиту

На цій сторінці є 4 поля для запису, 1, що випадає із зазначенням пріоритету заявки і можливість вкладення максимум 2-ох файлів.

Поля «Ел. адреса», «Тема» та поле з захистом від спаму повинні бути обов'язково заповнені.

Приклад заповненої форми створення запиту зображений на рис. 5.3.

Ел. адреса: \*  
dfor4w@gmail.com

Пріоритет: \* ■ Середній ▾

---

Тема: \*  
Test111222222

Повідомлення:  
TestTest

Рекомендовані статті бази знань:  
Не знайдено жодної відповідної статті.

---

Вкладення:  Tests.png  
 Файл не вибран  
Maximum 2 attachments ⓘ

---

**SPAM Prevention**  
8 6 5 7  
9 5 7  
Введіть число на зображенні нижче. \* ↻  
86957

---

Рисунок 5.3 – Приклад заповненої форми створення запиту

Якщо були заповнені обов'язкові поля і вірно введена захист від спаму, то після натискання на кнопку «Надіслати запит» користувача перенаправляє на сторінку, яка повідомляє про те, що запит був відправлений на розгляд і відобразить його ID. Відповідно до одного з вимог, email адреса не обов'язково повинен існувати. Сторінка успішного створення запиту зображена на рис. 5.4.

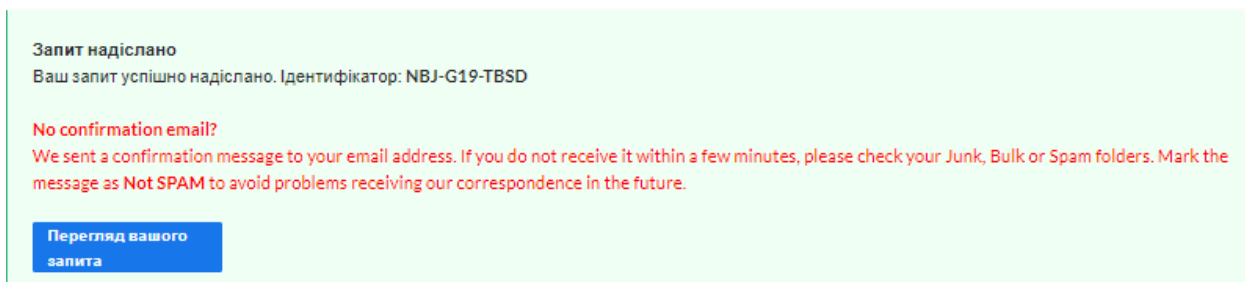


Рисунок 5.3 – Сторінка успішного створення запиту

Якщо на головній сторінці натиснути на кнопку «View existing tickets», то користувача перенаправляє на сторінку перегляду вже створених ним запитів. Сторінка перегляду створених запитів зображена на рис. 5.4.

[Переглянути існуючий запит](#)

Ідентифікатор запита \*

Ел. адреса \*

Пам'ятати мою електронну адресу

[Переглянути запит](#)

[Забули ідентифікатор відстеження?](#)

Рисунок 5.5 – Сторінка перегляду створених тікетів

Якщо користувач забув ID свого запиту, існує можливість його відновлення. Для цього необхідно натиснути на кнопку «Забули ідентифікатор відстеження?». Після натискання з'являється поле введення email адреси, на яку потрібно надіслати лист зі списком ID відкритих запитів користувача або

ж усіма запитами які створював користувач. Поле введення email адреси для відновлення ID запита зображено на рис. 5.6.

Рисунок 5.6 – Поле введення email адреси для відновлення ID запиту

Після натискання на кнопку «Надіслати мій ідентифікатор відстеження» на зазначену пошту буде відправлено листа з інструкціями.

Також на головній сторінці є база знань, в якій відображена коротка інформація найбільш важливих і популярних статей, серед яких різні керівництва, відповіді на актуальні питання та вирішення найбільш популярних запитів. При натисканні на якусь статтю, користувача перекидає на сторінку в якій стаття описано більш детально і є можливість оцінити її. Також нижче є кнопка «переглянути всю базу знань», при натисканні на яку відкривається сторінка з усією базою знань. База знань і статті зображені на рис. 5.7-5.9.

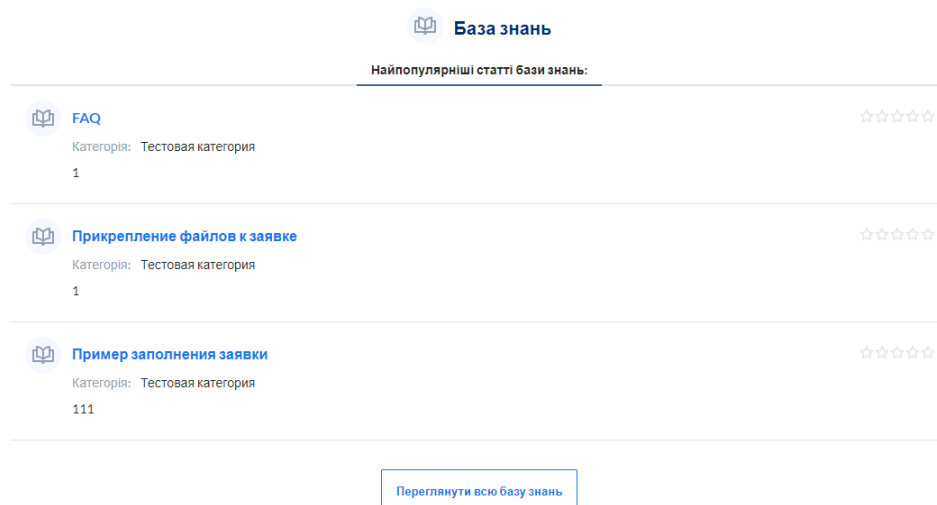


Рисунок 5.7 – База знань



## FAQ

1

Вам допомогла ця стаття? [Yes](#) | [No](#)

## Деталі статті


Ідентифікатор статті: 2


Категорія: [Тестовая категория](#)

Рейтинг: ☆☆☆☆☆

[← Повернутися](#)

Рисунок 5.8 – Стаття в базі знань

 **База знань**

 **Тестовая категория**

- [Пример заполнения заявки](#)
- [FAQ](#)
- [Прикрепление файлов к заявке](#)

**Найпопулярніші статті бази знань:**




	<b>FAQ</b>	☆☆☆☆☆
	Категорія: <a href="#">Тестовая категория</a>	
	1	
	<b>Пример заполнения заявки</b>	★★★★★
	Категорія: <a href="#">Тестовая категория</a>	
	111	
	<b>Прикрепление файлов к заявке</b>	☆☆☆☆☆
	Категорія: <a href="#">Тестовая категория</a>	
	1	

Рисунок 5.9 – Повна база знань

У самому низу на головній сторінці також є кнопка «Перейти до панелі адміністрування», при натисканні на яку користувача перенаправляє на сторінку авторизації, яка зображена на рис. 5.10.

Рисунок 5.10 – Авторизація в систему

На цій сторінці є 2 поля «Ім'я користувача» і «Пароль», а також позначку «Запам'ятати моє ім'я користувача».

При правильному вказівці комбінації імені користувача та пароля співробітника авторизує в систему. У разі якщо співробітник забув пароль передбачена можливість його відновлення, для цього необхідно потрібно натиснути на напис «Forgot your password?». Після цього необхідно вказати email адреса, пройти захист від спаму і натиснути на кнопку «Reset Password». Відновлення пароля зображено на рис. 5.11.

Рисунок 5.11 – Відновити пароль

При натисканні на кнопку «Reset Password» на зазначену пошту прийде лист з посиланням яке скине попередній пароль співробітника і попросить вказати новий. Створення нового пароля зображено на рис. 5.12.

**Пароль**

Новий пароль

Підтвердження пароля

Надійність пароля

Рисунок 5.12 – Створення нового пароля співробітника

При авторизації користувача в залежності від ролі, адміністратор або персонал, інтерфейс буде виглядати по-різному.

У разі якщо співробітник авторизувався з правами адміністратора, він побачить бічне меню з категоріями:

- запити;
- шаблони;
- база знань;
- категорії;
- команда;
- звіти;
- інструменти;
- настройки.

Бокове меню зображено на рис. 5.13.

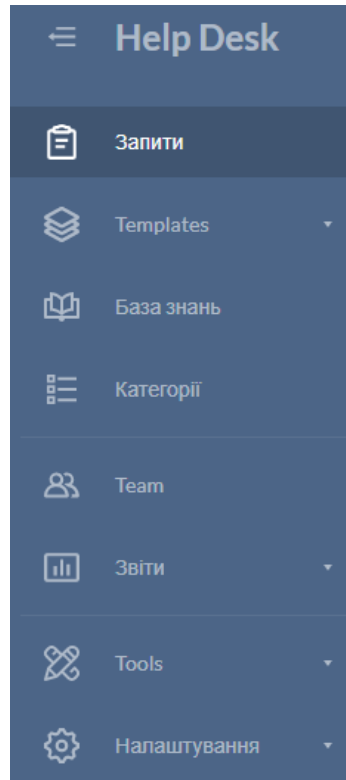


Рисунок 5.13 – Бокове меню адміністратора

Запити. Це сторінка в якій адміністратор може переглянути запити в залежності від налаштувань фільтра, вона зображена на рис. 5.14.

Запити (3) Auto reload page

ІДЕНТИФІКАТОР ВІДСТЕЖЕННЯ	КАТЕГОРІЯ	ТЕМА	СТАН	ВІСНИК	ВІДПОВІД	ВІДПОВІД (ПЕРСОНАЛ)	ОСТАННЯ ВІДПОВІДЬ ВІД	ЧАС РОБОТИ	ПРІОРИТЕТ
<a href="#">VBE-9NH-LYD5</a>	Загальне	<a href="#">Тест</a>	Новий	Дмитро Колачевський	0	0	Ініціатор	00:00:00	Середній
<a href="#">W95-BY4-B3A5</a>	Загальне	<a href="#">Test2</a>	Новий	Не призначено	0	0	Ініціатор	00:00:00	Середній
<a href="#">XMS-BHT-BP2V</a>	Загальне	<a href="#">Test</a>	Розглянуто	Не призначено	0	0	Анонім	00:00:00	Високий

Кількість запитів: 3 | Кількість сторінок: 1

Позначений запит  
 Призначено мені  
 Призначено іншим співробітникам

With selected: Set priority to: Низький Виконати

Assign selected to: -- Натисніть, щоб вибрати -- Призначити

Показати запити

Стан

Новий  
 Розглянуто  
 Очікування відповіді  
 Розглядається  
 Відповідь надіслано  
 На утриманні

Показати запити Більше параметрів

Рисунок 5.14 – Сторінка з існуючими запитами

За замовчуванням відображаються все не завершені запити. Адміністратор може вибрати кілька запитів відразу і присвоїти їм інший пріоритет або ж призначити виконавця всього за кілька кліків. Також він може змінити налаштування фільтра і, наприклад, відобразити тільки виконані запити як зображено на рис. 5.15.

Запити (1)  Auto reload page

<input type="checkbox"/>	ІДЕНТИФІКАТОР ВІДСТЕЖЕННЯ	КАТЕГОРІЯ	ТЕМА	СТАН	ВЛАСНИК	ВІДПОВІДЬ	ВІДПОВІДІ (ПЕРСОНАЛ)	ОСТАННЯ ВІДПОВІДЬ ВІД	ЧАС РОБОТИ	ПРІОРИТЕТ
<input type="checkbox"/>	<a href="#">ХМЗ-ВНТ-ВР9У</a>	Загальне	Test	Розглянуто	Не призначено	0	0	Аnonim	00:00:00	Високий

Кількість запитів: 1 | Кількість сторінок: 1

Позначений запит  
 Призначено мені  
 Призначено іншим співробітникам

With selected: Set priority to: Низький Виконати  
 Assign selected to: -- Натисніть, щоб вибрати -- Призначити

**Показати запити**

Стан

Новий  
 Розглянуто  
 Очікування відповіді  
 Розглядається  
 Відповідь надіслано  
 На утриманні

[Показати запити](#) [Більше параметрів](#)

Рисунок 5.15 – Відображення запитів зі станом «Розглянуто»

Також є більш детальний фільтр. Він включає в себе пошук запитів по:

- ID запиту;
- номеру запиту;
- імені ініціатора запиту;
- email ініціатора запиту;
- теми запиту;
- повідомленням;
- ключові слова;
- IP адресою ініціатора запиту;
- призначеному виконавцю.

Також щоб звузити обсяг пошуку можна вибрати в якій категорії шукати дану заявку і за яку дату. Даний фільтр зображений на рис. 5.16.

**Знайти запит**

Шукати

Шукати в

Категорія

Власник

Дата

Шукати в  Призначено мені  
 Призначено іншим  
 Непризначені запити  
 Тільки позначені запити

Показати  результатів на сторінку

[Знайти запит](#) [Менше параметрів](#)

Рисунок 5.16 – Детальний фільтр пошуку запитів

Також адміністратор може заповнювати базу знань новими статтями, видаляти старі, редагувати існуючі, систематизувати їх і т.д. Сторінка управління базою знань зображена на рис. 5.17.

**Керування базою знань** ⓘ

🔍 [Переглянути базу знань](#) ➕ [Нова стаття](#) ☰ [Нова категорія](#) ✎ Ідентифікатор статті:  [Редагувати](#)

Структура бази знань

- ➕ ☰ ⚙️ 📁 [База знань](#) (0, 0, 0)
- ➕ ☰ ⚙️ 📁 [Тестовая категория](#) (3, 0, 0)

➕ додати нову статтю до вибраної категорії.  
 ⚙️ керування вибраними категоріями (редагування, видалення, керування статтями).  
 (1, 2, 3) = кількість загальнодоступних, закритих статей і чернеток у категорії.

» Інструменти бази знань

- 🔍 [Перелік приватних статей](#)
- 🔍 [Перелік чернеток статей](#)
- ⚙️ [Перевірити й оновити кількість статей у категорії](#)
- 🔗 [Як написати хорошу статтю в базу знань?](#)

Рисунок 5.17 – Сторінка управління базою знань

Наступний пункт в меню – категорії. Адміністратор може створювати нові категорії для заявок, редагувати вже існуючі і видаляти непотрібні. Сторінка з категоріями зображена на рис. 5.18.

НАЗВА КАТЕГОРІЇ	ПРІОРИТЕТ	ЗАПИТІВ	ЗРОБИТИ ЦЮ КАТЕГОРІЮ ПРИВАТНОЮ (ТІЛЬКИ ПЕРСОНАЛ ЗМОЖЕ ВИБИРАТИ ІЇ)	
Загальне	Високий	3 (100%)	Це ЗАГАЛЬНОДОСТУПНА категорія (натисніть, щоб зробити її приватною)	Створити праме посилання

Рисунок 5.18 – Сторінка з категоріями заявок

Наступним пунктом адміністратор може переглянути весь персонал, зареєструвати нових користувачів, редагування і видалення вже зареєстрованих користувачів. Сторінка з переглядом всіх користувачів зображена на рис. 5.19.

ІМ'Я	ЕЛ. АДРЕСА	ІМ'Я КОРИСТУВАЧА	ROLE	
Admin	testoviytest559@gmail.com	admin	Адміністратор	✎
Александр Шевченко	alekshev@gmail.com	ShevA	Персонал	✎ 🗑
Володимир Воронін	voronv@gmail.com	VoronV	Персонал	✎ 🗑
Дмитро Копачевський	kopachdm@gmail.com	DmitroKop	Персонал	✎ 🗑
Ярослав Кульчицький	dtor4w@gmail.com	YarK	Адміністратор	✎ 🗑

Рисунок 5.19 – Сторінка перегляду існуючих користувачів системи

Згідно з ТЗ користувачі не можуть особисто зареєструватися в системі, цим повинен зайнятися адміністратор. Щоб додати нового співробітника в базу даних системи, необхідно натиснути на кнопку «New Team Member». При

натисканні на кнопку вилізе вікно як на рис. 5.20. На першому етапі адміністратор повинен ввести справжнє ім'я користувача, його email адреса та ім'я користувача. Також придумати надійний пароль.

← Додати нового користувача

1 2 3 4 5

Інформація про профіль Permissions Підпис Вподобання Сповіщення

Справжнє ім'я

Ел. адреса

Ім'я користувача

**Пароль**

Пароль

Підтвердження пароля

Надійність пароля

Next

Рисунок 5.20 – Вікно реєстрації нового користувача

На наступному етапі необхідно видати тип користувача: адміністратор або персонал. У разі вибору типу «персонал», необхідно також вказати права користувача. Другий етап реєстрації нового користувача і список всіх можливих прав користувача зображено на рис. 5.21.



← Додати нового користувача

1 2 3 4 5  
 Інформація про профіль Permissions Підпис Вподобання Сповіщення

Account type

Адміністратор (доступ до всіх функцій і категорій)

Staff (ви можете обмежити кількість властивостей і категорій)

**Категорії**

Загальне

**Властивості**

Переглянути запити

Відповісти на запити

Видалити запити

Редагувати відповіді

Об'єднати запити

Can resolve tickets

Can submit tickets to any category

Видалити примітки до запита

Змінити категорію запита

Change ticket category (to allowed)

Керування базою знань

Керування користувачами

Керування категоріями

Керування шаблонами відповідей

Manage ticket templates

Керування налаштуваннями служби підтримки

Можна позначити запити

Можна призначити запити собі

Можна призначити запити іншим

Можна переглядати не призначені запити

Можна переглядати запити, призначені іншим

Can view tickets he/she assigned to others

Можна створювати звіти (власні)

Можна створювати звіти (усі)

Можна експортувати запити

Можна переглядати, хто з персоналу перебуває в системі

Can ban emails

Can unban emails (enables Can ban emails)

Can ban ips

Can unban ips (enables Can ban ips)

Can anonymize tickets

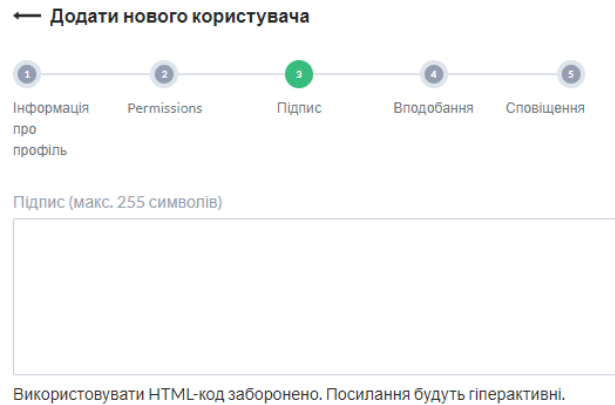
Edit service messages

Edit email templates

Back Next

Рисунок 5.21 – Етап присвоєння типу користувача, а також список всіляких прав які можна видати співробітнику

Далі необов'язковий етап із зазначенням додаткової інформації про користувача. Він зображений на рис. 5.22.



← Додати нового користувача

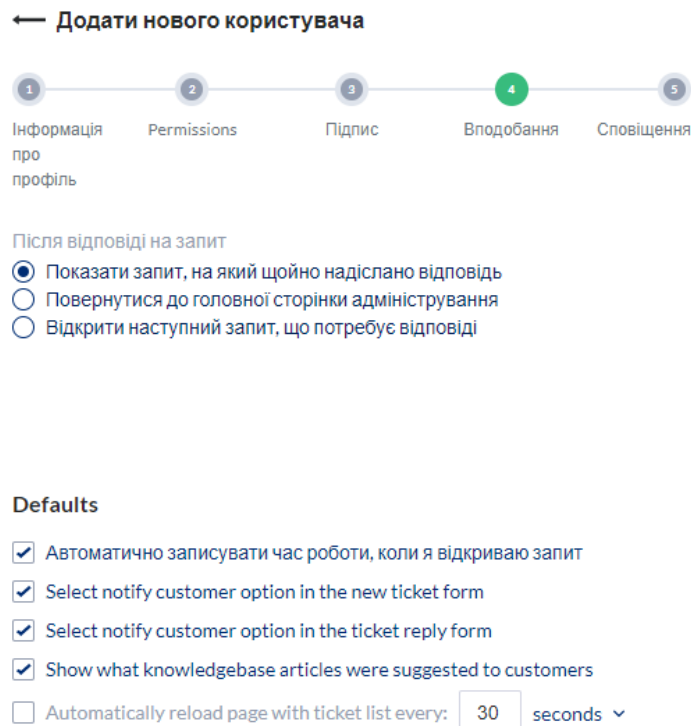
1 Інформація про профіль 2 Permissions 3 Підпис 4 Вподобання 5 Сповіщення

Підпис (макс. 255 символів)

Використовувати HTML-код заборонено. Посилання будуть гіперактивні.

Рисунок 5.22 – Вказування додаткової інформації про користувача

На передостанньому етапі необхідно вибрати різні настройки користувача які вказані на рис. 5.23.



← Додати нового користувача

1 Інформація про профіль 2 Permissions 3 Підпис 4 Вподобання 5 Сповіщення

Після відповіді на запит

- Показати запит, на який щойно надіслано відповідь
- Повернутися до головної сторінки адміністрування
- Відкрити наступний запит, що потребує відповіді

**Defaults**

- Автоматично записувати час роботи, коли я відкриваю запит
- Select notify customer option in the new ticket form
- Select notify customer option in the ticket reply form
- Show what knowledgebase articles were suggested to customers
- Automatically reload page with ticket list every: 30 seconds ▾

Рисунок 5.23 – Налаштування роботи користувача

І на останньому етапі настроїти повідомлення по електронній пошті. Він зображений на рис. 5.24.

← Додати нового користувача

1 2 3 4 5

Інформація про профіль Permissions Підпис Вподобання Сповіщення

Служба підтримки надсилатиме електронне сповіщення коли:

- Надіслано новий запит із зазначенням власника: Не призначено
- Надіслано новий запит із зазначенням власника: Призначено мені
- Користувач відповідає на запит із зазначенням власника: Не призначено
- Користувач відповідає на запит із зазначенням власника: Призначено мені
- Запит призначено мені
- Хтось додає примітку до запита, призначеного мені
- Мені надіслано внутрішнє повідомлення

Рисунок 5.24 – Налаштування повідомлення по електронній пошті

Також однією з вимог до HelpDesk системі була побудова звітів за обраними параметрами. Для цього є наступний пункт в меню – звіти. Існує можливість переглянути звіт онлайн за певний відрізок часу, кількість запитів на кожного користувача і на кожен категорію. Перегляд звітів онлайн зображені на рис. 5.25-5.27.

Створити звіти

Проміжок часу: Цей місяць: Грудень

Від: 11/20/2020 на: 12/20/2020

Тип звіту: Запитів за день

Показати звіт

Грудень 1, 2020 - Грудень 20, 2020

дата	нові запити	відкриті	розглянуто	час роботи
Всього		4	3	1
вівторок, Грудень 1, 2020	0	0	0	
середа, Грудень 2, 2020	0	0	0	
четвер, Грудень 3, 2020	0	0	0	
п'ятниця, Грудень 4, 2020	0	0	0	
субота, Грудень 5, 2020	0	0	0	
неделя, Грудень 6, 2020	0	0	0	

Рисунок 5.25 – Звіт за певний відрізок часу

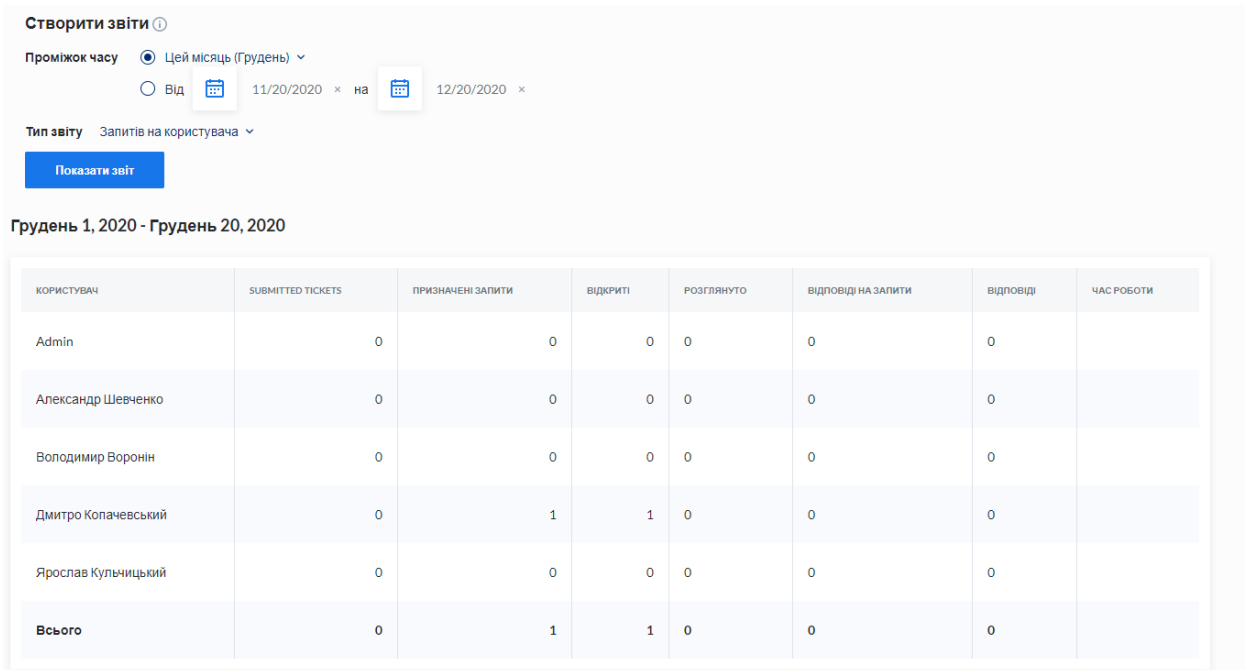


Рисунок 5.26 – Звіт запитів на кожного користувача

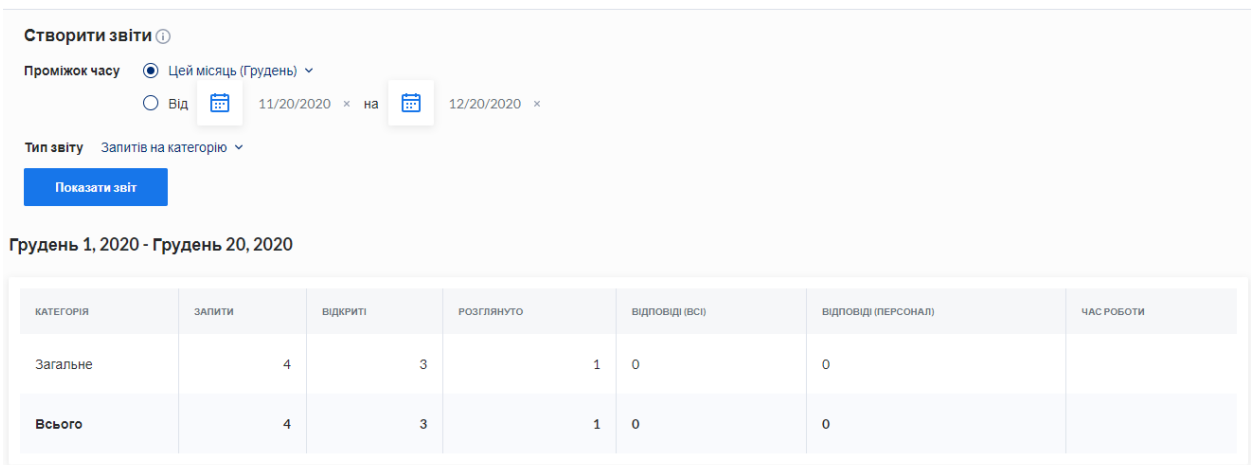


Рисунок 5.27 – Звіт запитів на кожну категорію

Передбачена можливість експорту запитів у вигляді excel файлу в архіві. Для цього необхідно вибрати необхідні параметри в фільтрі експорту звіту і натиснути на кнопку «Експортувати запити». Інтерфейс експорту запитів зображений на рис. 5.28, а приклад excel файлу із запитами зображений на рис. 5.29.

**Експортувати запити** ⓘ

Проміжок часу  Цей місяць (Грудень) ▾

Від  на

**Стан**

Новий

Очікування відповіді

Відповідь надіслано

Розглянуто

Розглядається

На утриманні

**Пріоритет**

\* Критично \*

Високий

Середній

Низький

**Assigned to**

Призначено мені

Непризначені запити

Призначено іншим

Тільки позначені запити

**Сортувати за**

Пріоритет

Оновлено

Ім'я

Тема

Стан

Послідовно

**Категорія**

Будь-яка категорія ▾

**Сортування**

за зростанням

за спаданням

**Експортувати запити**

Рисунок 5.28 – Сторінка експорту запитів

#	Ідентифікатор від	Дата	Оновлено	Ім'я	Ел. адреса	Категорія	Пріоритет	Стан	Тема	Повідомлення	Власник	Час роботи
1	1	XM3-BHT-BP9V	15.12.2020 15:37	19.12.2020 15:32	Анонім	dtor4w@gmail.com	Загальне	Високий	Розглянуто	Test		00:00:00
2	2	VBE-9NH-LYD5	18.12.2020 17:54	19.12.2020 15:33	Ініціатор	dtor4w@gmail.com	Загальне	Середній	Новий	Test	Дмитро Копачевський	00:00:00
3	3	W9S-BY4-B3A5	19.12.2020 15:44	19.12.2020 15:44	Ініціатор	testtest@gmail.com	Загальне	Середній	Новий	Test2		00:00:00
4	5	NVI-G19-TBSD	19.12.2020 17:23	19.12.2020 17:23	Ініціатор	dtor4w@gmail.com	Загальне	Середній	Новий	Test1112222	TestTest	00:00:00

Рисунок 5.29 – Приклад ексел файлу із запитам

У передостанньому пункті меню, який називається «інструменти» є можливість заблокувати email користувача або ж його IP адреса, а також переглядати вже заблокованих користувачів. Інтерфейс блокування і перегляд вже заблокованих email і IP адрес зображені на рис. 5.30-5.31.

**Ban Emails** ⓘ

Email address to ban Ban this email

Examples: john@example.com, @example.com

ЕЛ. АДРЕСА	BANNED BY	ДАТА	ПАРАМЕТРИ
No emails are being banned.			

Рисунок 5.30 – Інтерфейс блокування і перегляду вже заблокованих email адрес

**Ban IPs** ⓘ

IP address to ban Ban this IP

Examples:  
 123.0.0.0  
 123.0.0.1 - 123.0.0.53  
 123.0.0.0/24  
 123.0.\*\*

**Permanent IP bans:**

IP	IP RANGE	BANNED BY	ДАТА	ПАРАМЕТРИ
No IPs are being banned.				

Рисунок 5.31 – Інтерфейс блокування і перегляду вже заблокованих IP адрес

Якщо є необхідність повідомити всіх користувачів про щось передбачена можливість написати «Сервісне повідомлення» яке побачать всі

користувачі на головній сторінці. Приклад сервісного повідомлення зображений на рис. 5.32.

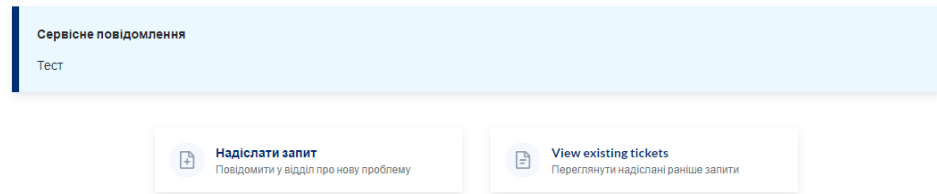


Рисунок 5.32 – Сервісне повідомлення на головній сторінці

В системі вже заздалегідь є створені шаблони email листів. Їх можна переглянути та редагувати в меню «Шаблони email листів». Приклад даної сторінки зображений на рис. 5.33, а також зміст деяких листів зображені на рис. 5.34-5.35.

Шаблони email листів ⓘ		Українська ▾
TITLE	DESCRIPTION	
forgot_ticket_id.txt	(Customer) Forgot ticket tracking ID	✎
new_reply_by_staff.txt	(Customer) New staff reply	✎
new_ticket.txt	(Customer) Ticket received	✎
ticket_closed.txt	(Customer) Ticket closed/resolved	✎
category_moved.txt	(Staff) Ticket moved to a new category	✎
new_reply_by_customer.txt	(Staff) New customer reply	✎
new_ticket_staff.txt	(Staff) New ticket submitted	✎
ticket_assigned_to_you.txt	(Staff) A ticket was assigned to you	✎
new_pm.txt	(Staff) New private message	✎
new_note.txt	(Staff) New note on a ticket assigned to you	✎
reset_password.txt	(Staff) Reset your password	✎

Рисунок 5.33 – Готові шаблони email листів

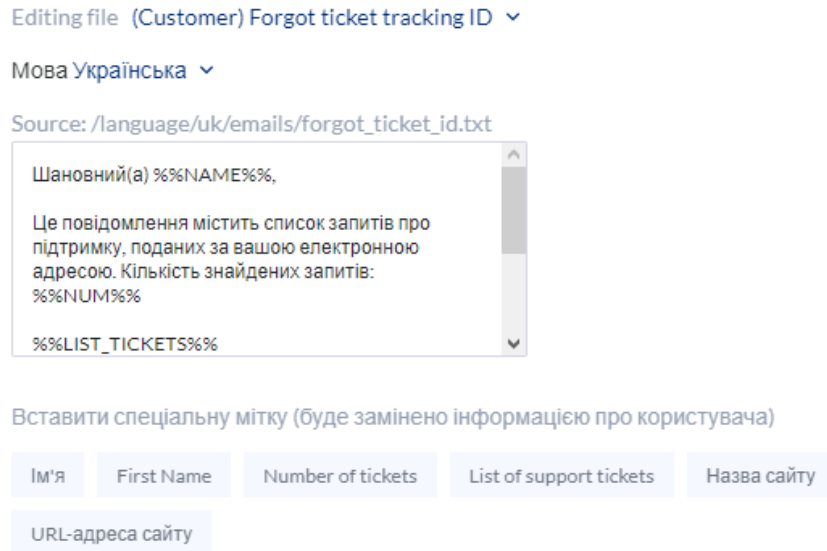


Рисунок 5.34 – Шаблон листа «Відновлення ID запита»

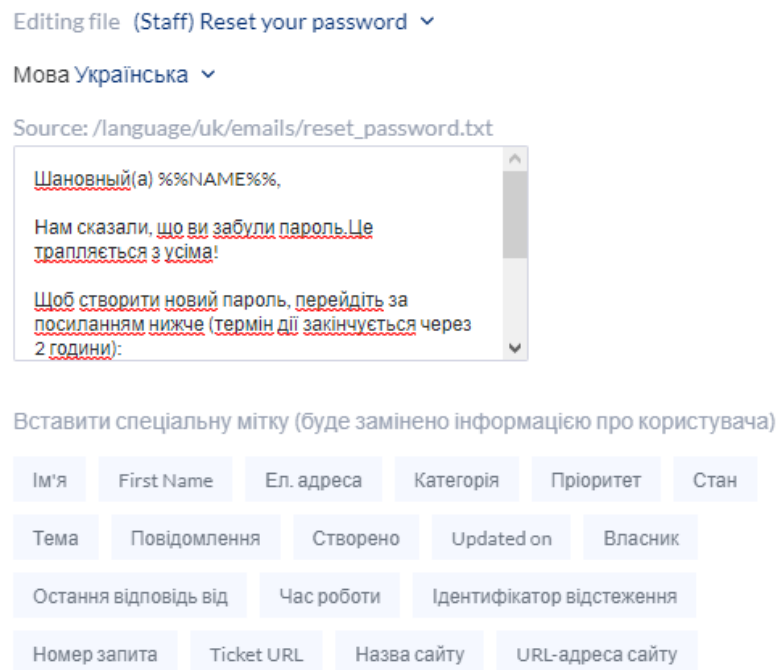


Рисунок 5.35 – Шаблон листа «Відновлення пароля користувача»

Приклад листа, який приходиться користувачеві на пошту зображено на рис. 5.36.



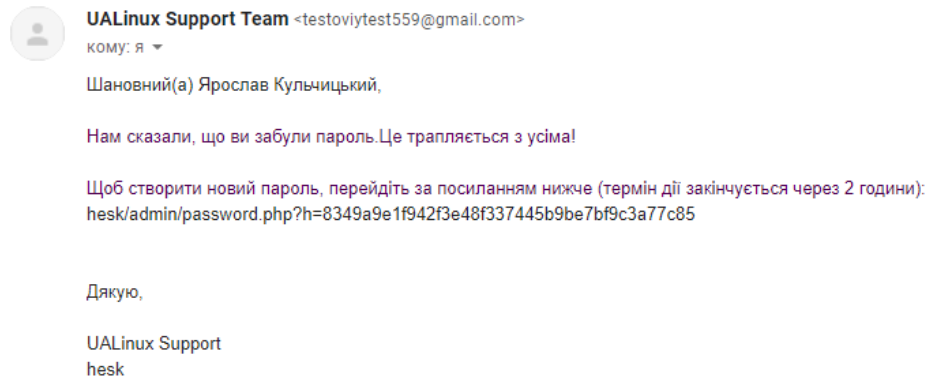


Рисунок 5.36 – Лист для відновлення пароля користувача

Для цього був створений тестовий email адрес testoviytest559@gmail.com і налаштований SMTP-сервер як зображено на рис. 5.37.

**Ел. адреса**

Надіслати електронні листи а допомогою *i*  PHP mail()  SMTP-сервера

Хост SMTP *i* smtp.gmail.com

Порт SMTP *i* 465

Час очікування SMTP *i* 20

Протокол SSL *i*

Протокол TLS *i*

Ім'я користувача SMTP *i* testoviytest559@gmail.com

Пароль SMTP *i* .....

[Перевірка SMTP-підключення](#)

**Успішно:** Підключення виконано успішно!

Рисунок 5.37 – Налаштування email для розсилки листів

Останній пункт в головному меню відведено для мінорних налаштувань системи. До цього пункту має доступ лише адміністратор. Наприклад настройка тієї ж email розсилки, налаштувати зовнішній вигляд інформації в таблиці запитів, обрати часовий пояс і формат часу і т.д. Приклади інтерфейсу налаштувань системи зображені на рис. 5.38-5.39.

Рисунок 5.38 – Налаштування відображення інформації в таблиці запитів

Рисунок 5.39 – Налаштування часового поясу, формату часу і т.д.

У разі якщо в систему авторизуватися звичайний співробітник, то у нього сторінка відрізнятиметься від сторінки адміністратора. Кількість відмінностей залежить від кількості прав які адміністратор видав співробітникам при реєстрації. Приклади головної сторінки звичайного виконавця і керівника зображені на рис. 5.40-5.41.

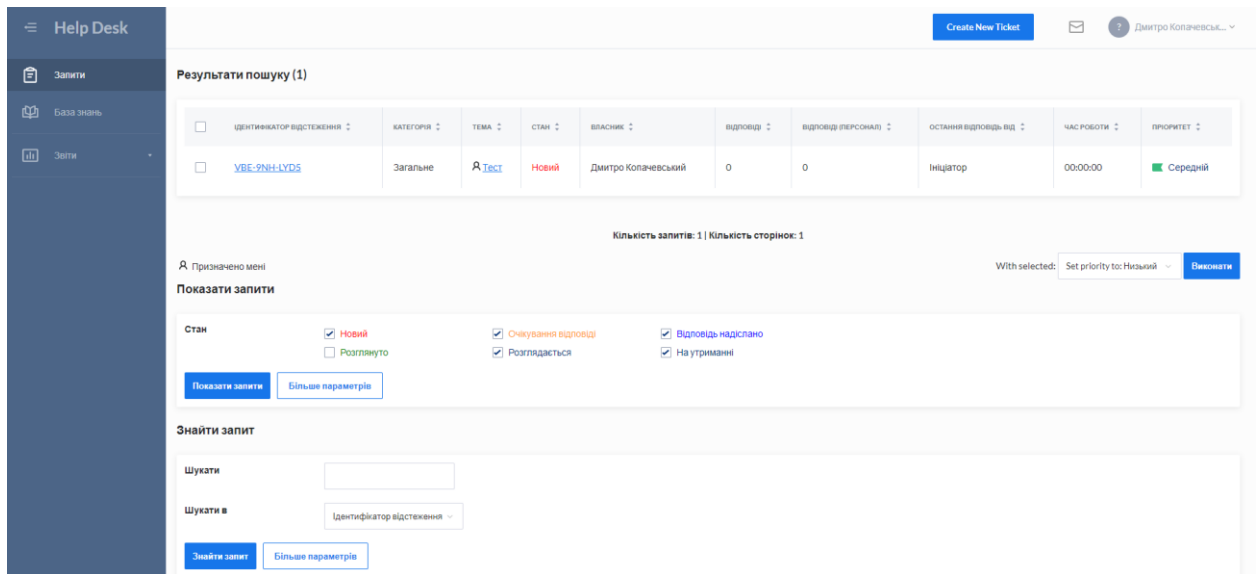


Рисунок 5.40 – Інтерфейс виконавця

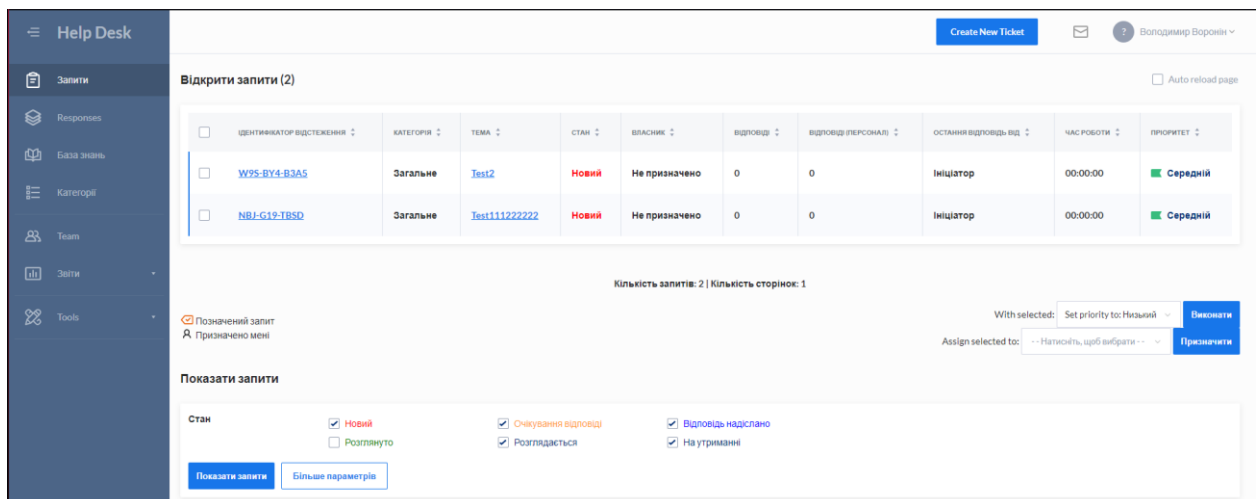


Рисунок 5.41 – Інтерфейс керівника

## ВИСНОВКИ

Впровадження HelpDesk вигідно для всіх – власників бізнесу, менеджерів компанії, IT підрозділів і безпосередніх споживачів.

Подібні сервіси усувають проблеми, які пов'язані з повільною реакцією IT служб на запити. Також підвищується якість послуг організацій, які мають кілька офісів і підрозділів в різних регіонах. Відповідно, поліпшується безпека IT інфраструктури компаній, зменшуються витрати і з'являється можливість їх прогнозувати.

Для користувачів впровадження Service Desk і Help Desk характеризується підвищенням рівня сервісу, що надається і зменшенням періодів зупинок через проблеми в IT інфраструктурі. А для самих IT служб впровадження подібної організації роботи стає рішенням проблеми, коли бізнес не може зрозуміти запитів айтішників. З'являється можливість просто обґрунтувати вложення в IT і планувати бюджет згідно з потребами в розвитку бізнес-процесів.

В ході даної магістерської кваліфікаційної роботи була проаналізована інформаційна система технічної підтримки – HelpDesk. В результаті даного аналізу був обраний один з аналогів HelpDesk-систем, який найбільш відповідає умовам ТЗ для компанії UALinux. При проектуванні ІС було застосовано наступне ПЗ, а саме – скриптовану мову програмування PHP, серверну операційну систему Ubuntu Server, вже існуюча open source HelpDesk система HESK, та система керування базами даних MariaDB. Для досягнення кінцевого результату були сформульовані і вирішені наступні завдання:

- розкрито теоретичні основи HelpDesk, ServiceDesk систем та поняття аудиту інформаційної системи;
- виконано порівняльний огляд існуючих аналогів;
- розглянуто проектні рішення з програмного, технічного та інформаційного забезпечення;
- виконано аналіз ефективності обраної HelpDesk-системи.

Послідовно вирішивши позначений на початку дослідження ряд завдань, нам вдалося досягти мети самого дослідження і проведення на його основі проектування, тобто розробити комплексний теоретично обґрунтований підхід до автоматизації технічної підтримки користувачів, а також спроектувати і реалізувати програмний продукт, що дозволяє вирішити ряд проблемних завдань, виділених в процесі дослідження.

Отримані в ході проведеного дослідження та проектування програмного продукту результати повинні мати практичну значимість, що виражається в суспільному і виробничому користь при використанні даних розробок.

У спроектованій системі можна автоматично реєструвати інциденти, присвоювати їм пріоритет і терміновість, призначати конкретного виконавця або групі, встановлювати термін ліквідації інциденту і налаштовувати схему ескалації, коли керівник оповіщається про найбільш важливих або не ліквідували в термін інциденти. Крім того, на базі підсистеми звітності можливо реалізувати аналіз звернень та інцидентів з метою виявлення повторюваних випадків і тим самим автоматизувати процес управління проблемами.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Про компанию UALinux. URL: <https://ualinux.com/ru/about> (дата звернення: 12.08.2020)
2. Аудит систем мониторинга информационных систем. URL: <https://rtmtech.ru/services/audit-sistemy-monitoringa-is/#:~:text=Вопрос-ответ-,Аудит%20системы%20мониторинга%20информационных%20систем,и%20уязвимостей%20в%20информационных%20системах> (дата звернення: 26.10.2020)
3. Helpdesk или ServiceDesk: 6 основных отличий. URL: <https://it-guild.com/info/blog/6-razlichiy-mezhdu-helpdesk-i-servicedesk> (дата звернення: 01.11.2020)
4. Зачем внедрять HelpDesk в компании. URL: <https://vc.ru/services/45344-zachem-vnedryat-helpdesk-v-kompanii> (дата звернення: 01.11.2020)
5. OTRS на русском – обзоры, отзывы. URL: <https://www.helpdeski.ru/tool/851/> (дата звернення: 02.11.2020)
6. Тикет-системы: как бесплатная OTRS три платных уделала? Блог компании RUVDS.com. URL: <https://habr.com/ru/company/ruvds/blog/483418/> (дата звернення: 02.11.2020)
7. OSTicket – обзор, плюсы и минусы программы, отзывы пользователей. URL: <https://8d9.ru/program/osticket> (дата звернення: 02.11.2020)
8. Request Tracker – это... Что такое Request Tracker? URL: <https://dic.academic.ru/dic.nsf/ruwiki/1357331> (дата звернення: 02.11.2020)
9. Технологии которые используют для разработки в Метод Лаб: Request Tracker. URL: <https://www.methodlab.ru/technology/rt.shtml> (дата звернення 02.11.2020)
10. Установка и настройка GLPI и FusionInventory, инвентеризация ИТ инфраструктуры | Windows для системного администратора. URL:

<https://winitpro.ru/index.php/2020/04/14/itsm-gli-inventory/> (дата звернення: 03.11.2020)

11. FreeScout: обзор, отзывы, аналоги, интеграция, сайт | BizzApps. URL: [https://bizzapps.ru/p/freescout/#:~:text=FreeScout%20—%20это%20бесплатная%20справочная%20служба,PHP7%20\(платформа%20Laravel%205.5](https://bizzapps.ru/p/freescout/#:~:text=FreeScout%20—%20это%20бесплатная%20справочная%20служба,PHP7%20(платформа%20Laravel%205.5) (дата звернення: 03.11.2020)

12. Зіставлення інформаційних систем з традиційними програмними продуктами – Лекції – Проектування інформаційних систем. URL: <https://uadoc.zavantag.com/text/2813/index-1.html?page=2> (дата звернення: 22.11.2020)

13. Серверные операционные системы ведущих производителей | Компьютер Пресс. URL: <https://compress.ru/article.aspx?id=12137> (дата звернення: 23.11.2020)

14. Операційна система для сервера кращі. Серверні Операційні системи (ОС). Файловий сервер Linux. URL: <https://maylohack.ru/uk/android/operacionnaya-sistema-dlya-servera-luchshie-servernye.html> (дата звернення: 23.11.2020)

15. Microsoft SQL Server – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server) (дата звернення 23.11.2020)

16. MySQL – система управления базами данных. URL: <https://webcreator.ru/articles/mysql> (дата звернення: 23.11.2020)

17. MariaDB – краткое руководство – CoderLessons. URL: <https://coderlessons.com/tutorials/bazy-dannykh/uznaite-mariadb/mariadb-kratkoe-rukovodstvo> (дата звернення: 24.11.2020)