

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інформаційна технологія формування запитів техпідтримки на підставі системних та користувальницьких даних»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-наукова програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.м.н-71 Бабак Богдан Олександрович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«__» травня 2019 р.

Науковий керівник

(підпис)

к.т.н., доц., Марченко А. В.

Голова комісії

(підпис)

Шифрін Д. М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2019

РЕФЕРАТ

Дипломна робота містить 91 сторінку, 5 таблиць, 38 рисунків, список використаної літератури з 32 найменувань, 2 додатки.

Актуальність даної роботи полягає в тому, що рішення проблеми нерациональної витрати робочого часу спеціалістів і задовільної якості обслуговування дозволяє знизити навантаження на робочий персонал відділу технічної підтримки, скоротити час очікування, підвищити довіру співробітників до відділу.

Об'єктом дослідження – процес підтримки користувачів.

Предмет дослідження – процес формування заявки на техпідтримку з урахуванням отримання системних та користувацьких даних.

Метою даного проекту є розробка інформаційної технології формування запитів техпідтримки з метою оптимізації діяльності відділу технічної підтримки для підвищення якості його роботи і оптимізації витрат робочого часу співробітників.

Ключові слова: бізнес-процеси, змістовна база даних, інтуїтивно-зрозумілий інтерфейс, інформаційна технологія, service desk, helpdesk, техпідтримка, запит, диспетчер, навантаження, користувальницькі дані, системні дані.

ЗМІСТ

| | |
|--|----|
| Вступ..... | 6 |
| 1 Аналіз предметної області..... | 9 |
| 1.1 Огляд системи HelpDesk..... | 9 |
| 1.2 Огляд літератури | 11 |
| 1.3 Огляд можливих альтернатив | 12 |
| 2 Постановка задачі та методи дослідження | 19 |
| 2.1 Мета та задачі дослідження..... | 19 |
| 2.2 Методи дослідження | 19 |
| 3 Моделювання інформаційної технології формування запитів техпідтримки на підставі системних та користувальницьких даних | 25 |
| 3.1 Концептуальне моделювання..... | 25 |
| 3.2 Логічне моделювання..... | 28 |
| 3.3 Фізичне моделювання | 29 |
| 3.4 Розробка діаграми варіантів використання | 30 |
| 3.5 Розробка діаграм нотацій моделювання IDEF0 та IDEF3..... | 31 |
| 3.6 Розробка схеми інформаційної технології..... | 36 |
| 4 Реалізація інформаційної технології формування запитів техпідтримки на підставі системних та користувальницьких даних | 39 |
| 4.1 Розробка бази даних | 39 |
| 4.2 Розробка програмного продукту..... | 40 |
| 4.3 Розробка інсталяційного пакету | 49 |
| Висновок..... | 52 |
| Список використаної літератури | 54 |
| ДОДАТОК А Планування робіт | 57 |
| ДОДАТОК Б Дамп бази даних..... | 68 |
| ДОДАТОК В Код програмного продукту..... | 75 |

ВСТУП

Автоматизація відділів технічної підтримки користувачів (ОТПП) – задача на сьогоднішній день відома і поширена. Можна говорити про величезну кількість вдало впроваджених проектів в різних галузях, в різних за масштабом компаніях, із застосуванням різних засобів автоматизації. Коли ми говоримо про автоматизацію ОТПП, ми завжди маємо на увазі впровадження автоматизованого засобу з групи програмного забезпечення Help Desk. Фактично поняття HelpDesk стало аналогом поняття ОТПП, що говорить про те, що в даний час більшість ІТ-служб в тій чи іншій мірі переглянули свій підхід до структури і процесів управління ІТ.

Служба підтримки відіграє важливу роль у допомозі користувачам. Сучасна, всеосяжна консультативна служба є фронт-офісом для всієї ІТ-організації і може обробляти більшість потреб і запитів користувачів без допомоги професіоналів. Підтримка користувачів є єдиною точкою контакту з ІТ-організацією, яка забезпечує своєчасне вирішення їх проблем.

Іншими словами, якщо сервісна служба допомоги користувачам не витратить час на нескінченний пошук фахівців, які можуть вирішити свої проблеми. Часто служба допомоги не тільки обробляє зовнішні запити від користувачів, але ці скарги були ініційовані в рамках ІТ-організації, наприклад, інциденти, вирішені автоматично або вручну ІТ-персоналом, виявили або отримали сервісні запити від інших ІТ-організацій відділів.

Наукова проблема полягає у тому, що служби Help Desk, які є аналогами програмного продукту магістерської роботи, являються лише способом передачі загальної інформації від користувача про заявку до спеціаліста. Жодний програмний продукт-аналог не має функціоналу передачі системної інформації, яка надала б змогу ідентифікувати проблему без необхідності присутності спеціаліста біля комп'ютера користувача. Наявність подібного функціоналу надала б змогу у

середньому пришвидшити швидкість обробки заявок та зекономити час як спеціаліста, так і користувача.

Об'єктом дослідження виступає процес підтримки користувачів. Предмет дослідження представляє собою процес формування заявки на техпідтримку з урахуванням отримання системних та користувацьких даних.

Актуальність даної роботи полягає в тому, що рішення проблеми нерациональної витрати робочого часу спеціалістів і задовільної якості обслуговування дозволяє знизити навантаження на робочий персонал відділу технічної підтримки, скоротити час очікування, підвищити довіру співробітників до відділу.

Після визначення актуальності має місце визначення мети дипломного проекту: розробка інформаційної технології формування запитів техпідтримки з метою оптимізації діяльності відділу технічної підтримки для підвищення якості його роботи і оптимізації витрат робочого часу співробітників. Для досягнення мети проекту були визначені наступні задачі:

- Аналіз даних та бізнес-процесів предметної області
- Розроблення бази даних
- Розроблення інтерфейсу системи
- Тестування розробленої системи

Під час дослідження були застосовані такі методи дослідження, як спостереження та порівняння.

Наукова новизна є критерієм наукового дослідження, що визначає ступінь перетворення, доповнення та конкретизації наукових даних. Наукова новизна - це означає, що інші не дослідили, які результати були досягнуті вперше. Визначення наукової новизни відноситься до всього дослідження в цілому.

Є 3 рівні наукової новизни:

- 1) перетворення відомих даних, радикальні зміни;
- 2) розширення і додавання відомих даних без зміни їх поняття;
- 3) уточнення відомих даних, поширення відомих результатів на новий клас об'єктів або систем [1].

Наукова новизна магістерської роботи відноситься до 2 рівня, так як клас об'єктів та систем залишився незмінним. Програмний продукт, розроблений в рамках виконання кваліфікаційної роботи магістра матиме змогу не просто отримувати заявку до техпідтримки від користувача, а й автоматично зчитувати інформацію з його персонального комп'ютера (список процесів, завантаженість центрального процесору, поточні служби і т.п.).

Наукова новизна магістерської роботи має другу форму (Введення нової ознаки [1]), так як функціонал зчитування інформації з користувальницького комп'ютера є саме новою ознакою до сформованого роками процесу обробки заявок сервісами HelpDesk.

Узагальнюючи, наукова новизна кваліфікаційної магістерської роботи полягає у розробленні інформаційної технології врахування системної інформації при автоматизованому формуванні заявки на техпідтримку, що дозволить оптимізувати ідентифікацію причини проблеми в великій частці випадків і, як результат, знизить час на обробку заявки. Окрім цього, для користувача програма матиме максимально спрощений інтерфейс, що допоможе швидше розібратися із створенням заявки користувачам не залежно від рівня комп'ютерної освіченості.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд системи HelpDesk

HelpsDesk, служба підтримки, інколи сервісна служба - система інформаційної підтримки, вирішення проблем для користувачів з комп'ютерами, апаратним забезпеченням та програмним забезпеченням. Це важливий компонент ІТІЛ (бібліотека інфраструктури інформаційних технологій). Він допомагає визначити проблемні області ІТ-інфраструктури та оцінити ефективність ІТ-відділу.

Великі організації мають у своєму розпорядженні на сьогодні складною ІТ-інфраструктурою, 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 дозволять ІТ-службам забезпечити якісне виконання цієї функції.

Системи HelpDesk (іноді звані Service Desk) забезпечують:

- єдину точку звернення в службу 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 співробітника служби ІТ, навіть якщо той не зобов'язаний займатися підтримкою).
- стандартний метод реєстрації та віднесення завдань до фахівців;
- контроль за порядком роботи, витраченим часом та ресурсами;
- визначення пріоритетів в залежності від характеру запиту, конкретного користувача або інших обставин;
- ескалація запитів та інцидентів, повідомлення відповідних адміністраторів

- зберігання бази знань за попередніми запитами дозволяє фахівцям швидко вирішувати подібні проблеми, які вже зустрічаються;

- звітування про час і гроші на виконання запитів. Звітність по наданню послуг може бути використана для формалізації відносини між користувачами інформаційних систем підприємства та службою ІТ. Очікуваний рівень підтримки може бути підтверджено і приведений у відповідність фінансування і чисельності відділу ІС.

За допомогою звітності системи HelpDesk можуть також виявляти закономірності в потоці запитів, що надходять, виділяючи таким чином "вузькі місця" в інфраструктурі підприємства.

Серед запитів, що обслуговуються системами HelpDesk, виділяються:

- сервісні запити (стандартні запити на підтримку роботи системи);
- запит на інцидент (інцидент визначається як відхилення, яке перевищує допустимі межі, такі як серйозне порушення системи або незапланована тривалість, і істотно ускладнює функціонування організації);
- запит на зміни стану системи (встановлення нового обладнання та програмного забезпечення).

Service desk складається з наступних логічних компонентів:

- модуль реєстрації заявок про інциденти
- база даних заявок
- система відстеження статусу заявки та оповіщення
- база знань
- панель адміністрування
- модуль звітності

Системи Help Desk також можуть бути інтегровані в апаратні засоби білінгу. Таким чином, може існувати спільний контроль за кількістю і типом пристроїв, і завжди є інформація про те, чи існує організація пристроїв, що відповідає певним вимогам [2].

1.2 Огляд літератури

Під час роботи над проектом була розглянута робота «A Guide to Help Desk Technology, Tools & Techniques» [3], автором якої є Dione McBride. Книга розкриває питання про технології, інструменти та методи, які використовуються у роботі ефективної служби допомоги. Посібник з техніки довідкової служби, інструментів і технологій розглядає багато опцій програмного забезпечення, доступних для відстеження та управління масовим припливом даних, які надходять у службу підтримки щоденно. Книга також проводить аналіз отриманих користувацької даних та пропозиції причин, що могли призвести до проблеми користувача, який буде використаний у даній роботі.

Також була розглянута книга автора Donna Knapp «A Guide to Help Desk Concepts» [4], яка знайомить читачів з концепціями послуг, наборами навичок, кар'єрними шляхами та діяльністю галузі допомоги. Це одна з перших книг, яка представляє концепції HelpDesk з точки зору освіти та надає огляд HelpDesk для осіб, зацікавлених у проведенні кар'єри у сфері підтримки клієнтів. Досвід автора дає сильні приклади комп'ютерної підтримки, тематичні дослідження та вправи в усій книзі, які допомогли визначити основні техніки комунікації між клієнтом та диспетчером, які будуть використані у даній роботі.

Вказані вище роботи допомогли проаналізувати предметну область дослідження, отримати вичерпне представлення щодо сучасних можливостей систем типу HelpDesk та сформулювати функціональні вимоги. Програмний продукт дипломного проекту повинен виконувати наступні функції:

- Реєстрація заявок на виконання робіт;
- Зчитування системної інформації, зведення отриманих даних у зручний формат для перегляду спеціалістом;
- Архівація завершених заявок.

1.3 Огляд можливих альтернатив

В поточному підрозділі представлені результати огляду сучасних аналогів. У список аналогів, які слід розглянути, таким чином можна включити такі інформаційні системи:

1. Jitbit HelpDesk;
2. Spiceworks;
3. GLPI;
4. Freshdesk

Розглянемо кожну систему детально.

Коротка характеристика аналогів

Jitbit HelpDesk використовують такі компанії, як VMware, Xerox, HP, Philips, Siemens та ін. В першу чергу Jitbit, це красивий і зручний інтерфейс. Працювати з заявками приємно. Якщо говорити про вартість, то молодші тарифні плани не витримують цінової конкуренції з іншими системами HelpDesk. Зате навіть найдешевша версія On-Premise не обмежує вас в кількості агентів і стоїть при цьому досить демократично. Jitbit все-таки більше підходить компаніям з кількістю фахівців служби підтримки хоча б більше 10-ти.

Spiceworks - безкоштовний HelpDesk і система обліку обладнання, призначені в основному для служб підтримки IT і працює локально на Windows-сервері з авторизацією на community.spiceworks.com. Spiceworks - унікальний в своєму роді продукт. З одного боку повністю безкоштовний, а з іншого - зроблений для людей, які не бажають витратити час на вивчення і установку, чого не можна сказати про Request Tracker. Якщо ви шукаєте HelpDesk для роботи IT підтримки і не хочете витратити багато часу на впровадження, Spiceworks відмінно підійде вам, встановлення та налаштування займає 10 хвилин. Але при цьому звичайно доведеться пожертвувати тим, що є в аналогічних системах.

GLPI - програмний засіб ITSM, який допомагає легко планувати та керувати змінами в IT, ефективно вирішувати проблеми, коли вони з'являються, і дозволити вам отримати законний контроль над IT-бюджетом компанії та витратами.

Freshdesk - сервіс HelpDesk від індійської компанії Freshdesk займає одні з перших рядків в рейтингах. Користуючись даним програмним забезпеченням, здається, що розробники перед тим, як приступити до створення Freshdesk, довго вивчали пропозиції конкурентів і врахували більшість їх недоліків [5].

Короткий огляд показників якості, що характеризують альтернативи

Проаналізувавши літературу про служби HelpDesk були виокремлені наступні показники якості:

- 1) Вартість – визначає, скільки коштує програмне забезпечення (ознака з найбільшою вагою);
- 2) Прийом запитів – визначає, яким шляхом буде відбуватися прийом запитів (ознака з найменшою вагою);
- 3) Політика SLA – визначає, яким чином реалізовується тайм менеджмент запитів (ознака з третьою по важливості вагою);
- 4) Автоматизація роботи з запитами – визначає, чи реалізована автоматична відповідь на найпоширеніші запитання користувачів (ознака з другою по важливості вагою).

Обґрунтування методу вибору доцільного варіанту

Для евалюації альтернатив та вибору оптимальної альтернативи може бути використана система підтримки прийняття рішень (СППР).

Система підтримки прийняття рішень (СППР) "Вибір" заснована на процесі аналітичної ієрархії (MAI), є простим і зручним інструментом, який допомагає структурувати проблему, будує набір варіантів, вибирає характеристику їх факторів, їх важливість, які ставлять альтернативи для оцінки для кожного з факторів виявлених неточностей і протиріч у міркуванні людини (ОПР) / експерти ранжирують альтернативні рішення, вирішують проаналізувати і обґрунтувати результати. Система заснована на математичному методі аналізу ієрархій Томаса Сааті.

СППР заснований на процесі аналітичної ієрархії, є простим і зручним інструментом для структурування проблеми, побудови набору варіантів, вибору для

характеристики їх факторів, оцінки важливості цих факторів для альтернатив для кожного з факторів неточностей і протиріч Рішення ДМ / експерта знайдено Поворот альтернатив, аналіз рішення і обґрунтування результатів.

За допомогою СППР можна вирішити наступні типові завдання:

- оцінка якості організаційних, проектних і проектних рішень;
- визначення інвестиційної політики в різних сферах;
- завдання розміщення (розташування небезпечних і небезпечних виробництв, пунктів обслуговування);
- розподіл ресурсів;
- аналіз проблеми за методом економічної ефективності;
- стратегічне планування;
- проектування та підбір обладнання, товарів;
- вибір професії, місця роботи, вибір персоналу.

МАІ вирішує слабо структуровані і неструктуровані проблеми. Метод вирішення таких завдань базується на системному підході, в якому проблема розглядається як результат взаємодії і, крім того, як взаємна залежність множини неоднорідних об'єктів, а не тільки як ізольований і автономний набір [6].

Реалізація процедури оцінювання альтернатив і вибору оптимальної альтернативи

Для найточнішої ідентифікації вагів матриці парних порівнянь необхідно побудувати таблицю, де зібрана інформація по усім критеріям кожної альтернативи (табл. 1.1):

Таблиця 1.1 – Інформація про альтернативи за критеріями

| Альтернатива | Вартість | Приєм запитів | Політика SLA | Автоматизація роботи з запитам |
|------------------------|-----------------|----------------------|---------------------|---------------------------------------|
| <i>Jitbit HelpDesk</i> | \$24,92 - \$249 | E-mail | Термін виконання | Відсутня |
| <i>Spiceworks</i> | Бескоштовно | E-mail | Вручну | Відсутня |

| | | | | |
|------------------|--|---|--------------------------------|----------|
| GLPI | €10 - €1000 | Програмне середовище | Вручну | Присутня |
| Freshdesk | Безкоштовно для 3 користувачів, далі – \$16 - \$70 | Програмне середовище телефон, соц. мережі, e-mail | Термін реагування та виконання | Відсутня |
| Дипломний проект | Безкоштовно | Програмне середовище | Термін реагування та виконання | Присутня |

Опираючись на таблицю, а також на літературу про служби HelpDesk, побудуємо матрицю парних порівнянь всіх критеріїв (рис. 1.1).

Получение матрицы парных сравнений

Относительно фактора **Ціль. Вибір оптимальної служби Help** необхідно провести парное сравнение следующих факторов уровня **Рівень критеріїв**

| № | Фактор | Вес |
|---|---------------------|-------|
| 1 | Вартість | 0,572 |
| 2 | Приём запитів | 0,047 |
| 3 | Політика SLA | 0,233 |
| 4 | Автоматизація ро... | 0,148 |

Матрица парных сравнений:

| | 1 | 2 | 3 | 4 |
|---|-----|---|-----|-----|
| 1 | 1 | 7 | 5 | 3 |
| 2 | 1/7 | 1 | 1/5 | 1/5 |
| 3 | 1/5 | 5 | 1 | 3 |
| 4 | 1/3 | 5 | 1/3 | 1 |

Какой из факторов предпочтительнее ?

Вартість
 Вартість
 Одинаково важны
 Не могу сказать

Степень предпочтения:

Абсолютно превосходит
 Промежуточное значение
 Значительно превосходит
 Промежуточное значение
 Существенно превосходит
 Промежуточное значение
 Умеренно превосходит
 Промежуточное значение
 Одинаково важны

Просмотр проекта $\lambda = 4,389$ ИС = 0,130 ОС = 0,144

OK Отмена

Рисунок 1.1 – Матриця парних порівнянь всіх критеріїв

Далі відбувається побудова матриці парних порівнянь для рівня всіх критеріїв: вартість (рис. 1.2), прийом запитів (рис. 1.3), політика SLA (рис. 1.4) та автоматизація роботи з запитами (рис. 1.5).

Получение матрицы парных сравнений

Относительно фактора **Рівень критеріів.Вартість** необходимо провести парное сравнение следующих факторов уровня **Рівень альтернатив**

| № | Фактор | Вес |
|---|------------------|-------|
| 1 | Jitbit HelpDesk | 0,038 |
| 2 | Spiceworks | 0,383 |
| 3 | GLPI | 0,038 |
| 4 | Freshdesk | 0,158 |
| 5 | Дипломний проект | 0,383 |

Матрица парных сравнений:

| | 1 | 2 | 3 | 4 | 5 |
|---|---|-----|---|-----|-----|
| 1 | 1 | 1/9 | 1 | 1/5 | 1/9 |
| 2 | 9 | 1 | 9 | 3 | 1 |
| 3 | 1 | 1/9 | 1 | 1/5 | 1/9 |
| 4 | 5 | 1/3 | 5 | 1 | 1/3 |
| 5 | 9 | 1 | 9 | 3 | 1 |

Какой из факторов предпочтительнее ?

Степень предпочтения:

- Абсолютно превосходит
- Промежуточное значение
- Значительно превосходит
- Промежуточное значение
- Существенно превосходит
- Промежуточное значение
- Умеренно превосходит
- Промежуточное значение
- Одинаково важны

Jitbit HelpDesk
 Jitbit HelpDesk
 Одинаково важны
 Не могу сказать

$\lambda = 5,040$
 $ИС = 0,010$
 $ОС = 0,009$

Рисунок 1.2 – Матрица парных порівнянь за критерієм «Вартість»

Получение матрицы парных сравнений

Относительно фактора **Рівень критеріів.Приём запитів** необходимо провести парное сравнение следующих факторов уровня **Рівень альтернатив**

| № | Фактор | Вес |
|---|------------------|-------|
| 1 | Jitbit HelpDesk | 0,059 |
| 2 | Spiceworks | 0,059 |
| 3 | GLPI | 0,153 |
| 4 | Freshdesk | 0,576 |
| 5 | Дипломний проект | 0,153 |

Матрица парных сравнений:

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|-----|-----|-----|
| 1 | 1 | 1 | 1/3 | 1/7 | 1/3 |
| 2 | 1 | 1 | 1/3 | 1/7 | 1/3 |
| 3 | 3 | 3 | 1 | 1/5 | 1 |
| 4 | 7 | 7 | 5 | 1 | 5 |
| 5 | 3 | 3 | 1 | 1/5 | 1 |

Какой из факторов предпочтительнее ?

Степень предпочтения:

- Абсолютно превосходит
- Промежуточное значение
- Значительно превосходит
- Промежуточное значение
- Существенно превосходит
- Промежуточное значение
- Умеренно превосходит
- Промежуточное значение
- Одинаково важны

Jitbit HelpDesk
 Jitbit HelpDesk
 Одинаково важны
 Не могу сказать

$\lambda = 5,095$
 $ИС = 0,024$
 $ОС = 0,021$

Рисунок 1.3 – Матрица парных порівнянь за критерієм «Приём запитів»

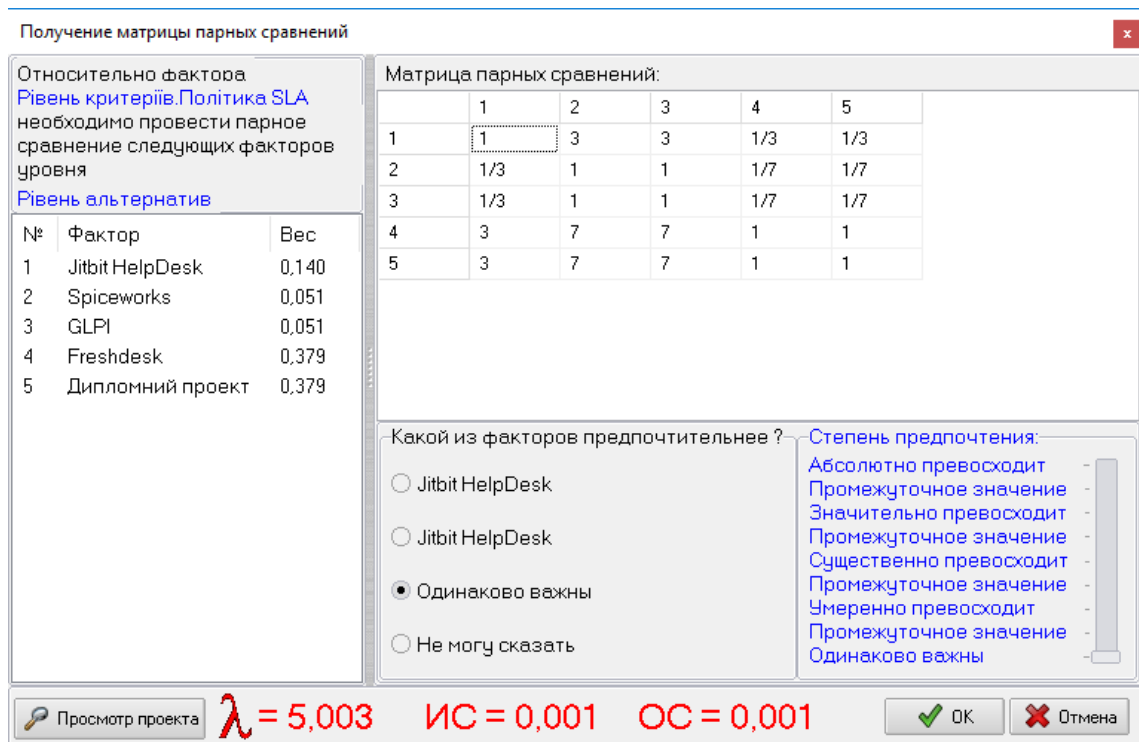


Рисунок 1.4 – Матрица парних порівнянь за критерієм «Політика SLA»

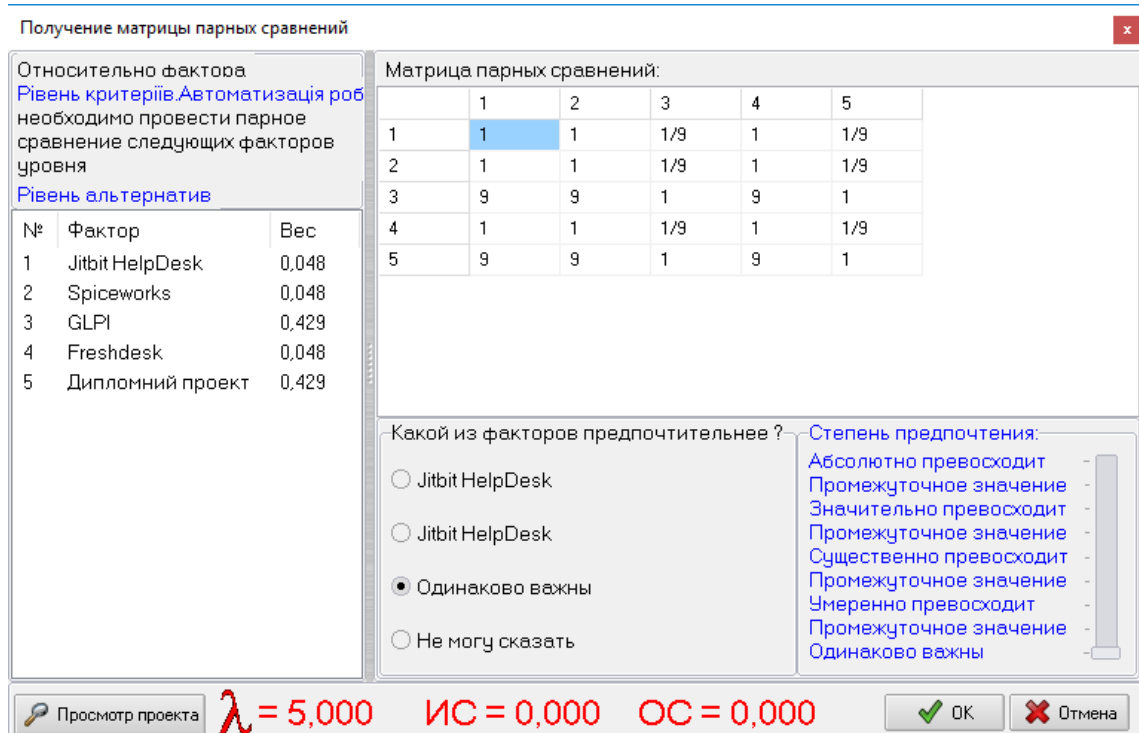


Рисунок 1.5 – Матрица парних порівнянь за критерієм «Автоматизація роботи з запитами»

У результаті отримаємо діаграму з результатами обчислень (рис. 1.6). Як бачимо, програмний продукт дипломного проектування, що реалізує розроблену інформаційну технологію, який є результатом роботи дипломного проекту є найбільш оптимальним серед аналогів за даними критеріями.

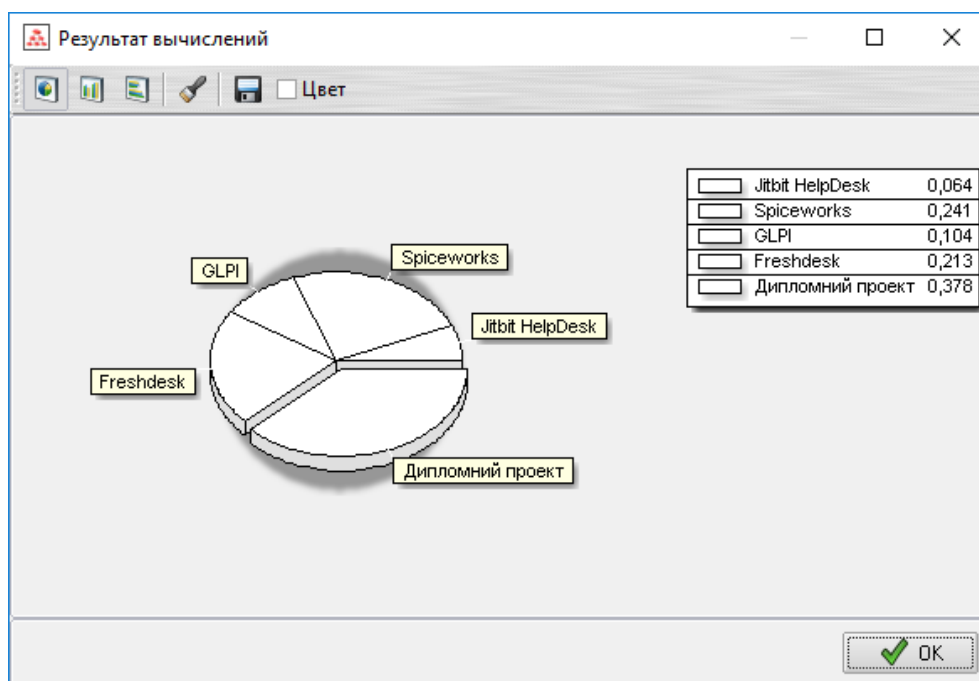


Рисунок 1.6 – Матриця парних порівнянь за критерієм «Прийом запитів»

У даному розділі було розглянуто систему HelpDesk, як інформаційну систему технічної підтримки, вирішення проблем користувачів з комп'ютерами, апаратним та програмним забезпеченням. Був проведений огляд тематичної літератури, який допоміг сформулювати функціональні вимоги до системи, яка буде розроблюватися. Огляд можливих альтернатив надав змогу оцінити сучасні програмні продукти та визначити необхідний функціонал, щоб створити конкурентне програмне забезпечення.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою даного проекту є розробка інформаційної технології формування запитів техпідтримки з метою оптимізації діяльності відділу технічної підтримки для підвищення якості його роботи і оптимізації витрат робочого часу співробітників. Програмний продукт, який реалізує розроблену ІТ, дозволить підвищити якість обслуговування заявок, що надходять до технічного відділу будь-якої компанії, знизити витрати вільного часу спеціалістів і зробити їх роботу більш ефективною за рахунок отримання бази знань, яка характеризує системний стан комп'ютера користувача, що дозволить оптимізувати ідентифікацію причини проблеми в великій частці випадків.

Для досягнення мети роботи були визначені задачі, обумовлені необхідністю проведення аналізу даних та бізнес-процесів предметної області, моделюванням та розроблення бази даних, яка буде використовуватися у проекті, розробленням інтуїтивно-зрозумілого користувацького інтерфейсу системи та тестування розробленої системи.

2.2 Методи дослідження

Вибір методів наукового дослідження

Метод наукового дослідження – це спосіб пізнання об'єктивної дійсності. Спосіб є певну послідовність дій, прийомів, операцій.

Спостереження – це вид пізнання об'єктивного світу, що ґрунтується на доцільному сприйнятті явищ і предметі органами почуттів, не враховуючи у процесі дослідника.

Порівняння – це знаходження відмінності між об'єктами або знаходження в них спільного, що здійснюється як за допомогою спеціальних пристроїв, так і за допомогою органів почуттів [7].

Математична модель – наближений опис будь-якого класу явищ зовнішнього світу, виражений за допомогою математичної символіки. Аналіз математичної моделі дозволяє проникнути в сутність досліджуваних явищ. Математична модель – потужний метод пізнання зовнішнього світу, а також прогнозування та управління [8].

Математична модель буде визначатися за допомогою систем масового обслуговування. Система масового обслуговування (СМО) – система, яка виробляє обслуговування надходять в неї вимог. Обслуговування вимог у СМО проводиться обслуговуючими приладами [9].

У процесі реєстрації запитів до техпідтримки є один обслуговуючий елемент – диспетчер, потік заявок ніяк не обмежений за часом і кількістю. Тому скористаємося одноканальною СМО з необмеженою довжиною черги.

Проведення дослідження

Розглянемо модель до впровадження інформаційної технології у підприємство для розуміння необхідності автоматизації процесів. За даними спостережень, проведених протягом року, працюючи в ПАТ «ВНІДАЕН» адміністратором Office додатків, середня частота надходження звернень $\lambda = 5$ заявок / год, при цьому диспетчер комфортно справляється з великим числом заявок $\mu = 6$ заявок / год, середній час обслуговування одного співробітника, який подав заявку $x = 1/5 = 0,167$ години. Диспетчер працює один, так що число каналів $m = 1$.

Знайдемо навантаження на СМО: $\rho = \lambda / (m\mu) = 0,83$.

Знайдемо ймовірність простою за формулою $P_0 = 1 - \rho$: $P_0 = 0,17$.

Знайдемо середню довжину черги за формулою:

$$\bar{q} = \frac{\rho^2}{2(1-\rho)}$$

Отримаємо: $q = 4,17$ заявки.

Відділ технічної підтримки приймає на обслуговування всі вступники заявки (відмов в обслуговуванні немає). Тому $P_{отк} = 0$, $P_{обсл} = 1$.

Коефіцієнт завантаження:

$$U = \rho(1 - P_{отк}), U = 0,83;$$

Середнє число заявок в обслуговуванні:

$$\bar{S} = mU, S = 0,83 \text{ заявок};$$

Середнє число заявок в СМО:

$$\bar{k} = \bar{q} + \bar{S}, K = 5 \text{ заявок};$$

Пропускна здатність СМО:

$$\gamma = \lambda(1 - P_{отк}), \Gamma = 5 \text{ заявок / год};$$

Середній час перебування заявки в черзі:

$$\bar{w} = \frac{\bar{q}}{\gamma}, W = 0,417 \text{ години};$$

Середній час перебування заявки в СМО:

$$\bar{t} = \frac{\bar{k}}{\gamma}, T = 0,5 \text{ години [10]}.$$

Обробка результатів дослідження

Проаналізуємо отримані характеристики СМО.

Диспетчер завантажений на 83%, тобто зайнятий обслуговуванням співробітників, які звернулися до техпідтримки протягом 83% всього часу своєї роботи. Протягом 17% часу диспетчер простоє через відсутність заявок. Таким чином, завантаження диспетчера досить висока. Таку завантаження можна вважати нормальною. Однак подальше збільшення завантаження небажано.

Формулювання висновків

В середньому в черзі перебуває 4,17 заявки, а в черзі і на обслуговуванні – 5 заявок. Диспетчер обслуговує в середньому 5 заявок на годину, тобто всі вхідні заявки. Час від інформування диспетчера про заявку, що надійшла, до початку її обслуговування (тобто час перебування заявки в черзі) становить в середньому 0,417 години. Час від надходження заявки до закінчення її обслуговування (час перебування заявки в очікуванні рішення про виконання) становить в середньому 0,5 години. Якщо порівняти цей час з тривалістю робочого дня і врахувати той факт, що до моменту появи на місці фахівця може пройти ще більше часу, то даний показник можна вважати незадовільним.

На основі отриманих даних на етапі тестування відбудеться порівняння результатів до впровадження системи і після її впровадження.

Вибір методів реалізації

Інформаційне забезпечення реалізовно у вигляді БД. Для вибору СКБД, було проаналізовано декілька робіт цієї тематики. Перша з них - Корнієнко М. М. «Інформатика. Бази даних. Системи управління базами даних» [11]. Видання містить основні теоретичні відомості про бази даних, системи управління базами даних та інформаційно-пошукові системи, етапи проектування й створення баз даних і принципи роботи з ними на прикладі СКБД Microsoft Access.

Окрім того, було розглянуто «MySQL. Справочник по языку» [12]. Ця книга, написана фахівцями компанії MySQL AB, є довідником по мові SQL, який використовується для організації запитів до баз даних, а також, особливості реалізації стандарту SQL в сервері MySQL. У книзі розглянуто весь спектр питань, що стосуються мовної структури, допустимих типів стовпців, операторів, операцій і функцій, а також існуючих розширень MySQL: крім того, представлена інформація, призначена для досвідчених програмістів і адміністраторів. Як відомо, MySQL займає лідируючі позиції серед безлічі систем управління базами даних з відкритим вихідним кодом. Завдяки високій продуктивності і простоті налаштування, багатому вибору API-інтерфейсів, а також функціональним засобам роботи з мережами, сервер MySQL

став одним з найбільш вдалих варіантів для розробки Web- додатків, взаємодіючих з базами даних. Книга розрахована на розробників Web -додатків і адміністраторів будь-якої кваліфікації, а також на студентів і викладачів відповідних дисциплін.

Для вирішення поставленої задачі було обрано СКБД MySQL – вільної системи керування базами даних з відкритим кодом, яка була створена як альтернатива комерційним системам.

Серед альтернатив написання програмного коду інтерфейсу системи було виділено крос-платформену мову програмування Java [13], мова програмування загального призначення C++ [14] та об'єктно-орієнтована мова програмування C#.

Вибір мови програмування також було складною задачею. Прийняти остаточне рішення допоміг аналіз відповідної літератури щодо трьох варіантів: C++, Java, C#.

«Язык программирования C++. Специальное издание.» [15]. Книга написана Бьорном Страуструпом, автором мови програмування C ++, і є доцільним викладом можливостей цієї мови. На додаток до докладного опису власне мови на сторінках книги можна знайти підходи до вирішення різних за складністю ефективних завдань програмування і проектування. Численні приклади мають доцільний стиль програмування на ядрі C ++ C і сучасний об'єктно-орієнтований підхід до розробки ПЗ. Третє видання бестселера було перероблено автором. Результатом цієї роботи стала більша доступність книги для починаючих програмістів. У той же час текст був збагачений методами програмування і інформацією, які можуть бути корисними навіть для досвідчених C ++ фахівців. Це спеціальне видання відрізняється від третього двома новими додатками, що присвячені локалізації та безпечній обробці виключень за допомогою стандартної бібліотеки), багатьма уточненнями в іншому тексті, а також виправленням великої кількості помилок. Книга адресована програмістам, які використовують мову програмування C ++ в своїй повсякденній роботі. Він також буде корисний студентам, вчителям і всім, хто хоче ознайомитися з описом мови.

Кей С. Хорстманн, автор і знавець Java, у своїй книзі «Java SE 8. Вводный курс.» [13] висвітлює найбільш цінні з нових мовних засобів у версії Java 8, а також розглядає ті засоби, які були впроваджені в версії Java 7, але не удостоїлися належної

уваги програмістів. Також автором надані практичні рекомендації та приклади коду, що демонструють нововведення у версії Java 8.

Книга Албахарі Джозефа «С# 3.0. Справочник» [15], в свою чергу, являє собою докладний довідник програмування на мові С # 3.0. реалізованому в Microsoft Visual Studio 2008. Перші глави цілком присвячені мові С#, починаючи з основ синтаксису, типів і змінних і закінчуючи більш складними темами, такими як небезпечний код або передпроцесорні директиви.

Коли методи розробки інформаційної системи обрані, настає час для формування проекту як такого. «The Project Management Life Cycle» [16] дозволяє освоїти існуючі методології управління проектами. У книзі описано 4 етапи життєвого циклу проекту: ініціювання, планування, виконання та закриття. Автор книги пише чітко, професійно та зрозуміло, що дозволяє уникнути складної термінології, яка існує в даній галузі.

Після розробки проекту важливо також протестувати продукт перед тим, як надавати його у користування замовнику. Книга «Pragmatic Unit Testing in C# with NUnit, 2nd Edition» [17] описує зрозумілі та перевірені способи впровадження модульного тестування у відповідний процес, що призводить до вищої якості та меншої кількості помилок. Зазвичай, програмні команди використовують тестування пристроїв як для перевірки їх коду, так і для того, щоб допомогти їм вдосконалювати той код. Ця книга унікальна, так як вона охоплює два аспекти: демонстрація розробникам як тестувати та допомагати їм визначати те, що потрібно протестувати.

Після аналізу вищезгаданих мов програмування було вирішено використовувати мову програмування С# через те, що вона заснована на строгій компонентній архітектурі і реалізує передові механізми забезпечення безпеки коду.

Майже всі описані вище методи були включені в учбову програму і тому використання цих методів засноване на стійкому розумінні переваг та недоліків даних методів і, як наслідок, після проведення аналізу альтернатив – їх вибір для реалізації інтерфейсу системи.

3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ ЗАПИТІВ ТЕХПІДТРИМКИ НА ПІДСТАВІ СИСТЕМНИХ ТА КОРИСТУВАЛЬНИЦЬКИХ ДАНИХ

Трирівнева схема даних визначає задачі і етапи проектування БД:

1. Етап концептуального моделювання;
2. Етап логічного моделювання;
3. Етап фізичного моделювання [18].

3.1 Концептуальне моделювання

Характеристика об'єктів та атрибутів

Концептуальне або інфографічне моделювання - це процес створення моделі, яка використовується в інформації про компанію і не залежить від фізичних аспектів її презентації. Результатом інфологічного моделювання є отримання семантичних моделей, що відображають інформаційний зміст певної теми, тобто концептуальної моделі даних для аналізованої частини підприємства. У табл. 3.1 та табл. 3.2 наведено характеристику об'єктів бази даних та їх атрибутів відповідно [18].

Таблиця 3.1 – Характеристика об'єктів бази даних

| Назва об'єкту | Характеристика об'єкта |
|---------------------|---|
| Автор (author) | Зберігає у собі інформацію про ПІБ, місце роботи та ім'я комп'ютеру авторів запитів |
| Запит (issue) | Зберігає у собі всю інформацію, введену автором у форму та додатково системні дані |
| Процеси (processes) | Зберігає у собі інформацію про усі процеси, що були активні у автора під час створення запиту |
| Тип (type) | Зберігає у собі інформацію про типи запитів |

Таблиця 3.2 – Опис атрибутів об'єктів

| Назва атрибута | Характеристика атрибута |
|------------------|--|
| author_id | Атрибут, який виступає первинним ключем об'єкту author і позначає його порядковий номер. Зв'язує таблиці author та issue |
| computername | Ім'я комп'ютера автора |
| name | Ім'я автора |
| surname | Прізвище автора |
| thirdname | Ім'я по-батькові автора |
| organization | Місце роботи автора |
| department | Відділ автора |
| headline | Заголовок заявки |
| description | Опис заявки |
| screenshot | Скріншот, прикріплений до заявки |
| creation_date | Дата створення заявки |
| end_date | Дата закриття заявки |
| status | Статус заявки (Открыто/Закрито) |
| type_id | Атрибут, який виступає первинним ключем об'єкту type і позначає його порядковий номер. Зв'язує таблиці type та issue |
| cpu | Завантаження CPU комп'ютера автора на момент створення заявки |
| ram | Завантаження RAM комп'ютера автора на момент створення заявки |
| devmgmt | Скріншот вікна «Диспетчер пристроїв» |
| minidump | Мінідамп файл |
| windowsupd | Лог файл оновлень віндос |
| process_id | Атрибут, який виступає первинним ключем об'єкту process і позначає його порядковий номер. Зв'язує таблиці process та issue |
| process_name | Назва процесу |
| PID | PID процесу |
| status | Статус процесу (Responding/Not responding) |
| username | Користувач, що запустив процес |
| memory | Кількість пам'яті, що використовує процес |
| description | Опис процесу |
| type_description | Тип заявки (Обычный/Срочный) |

Зв'язки між об'єктами бази даних:

- Автор (author) створює Заявку (issue)
- Заявка (issue) включає Процеси (processes)
- Заявка (issue) має Тип (type)

Отже, результат розроблення концептуальної моделі бази даних продемонстрований на рис. 3.1.

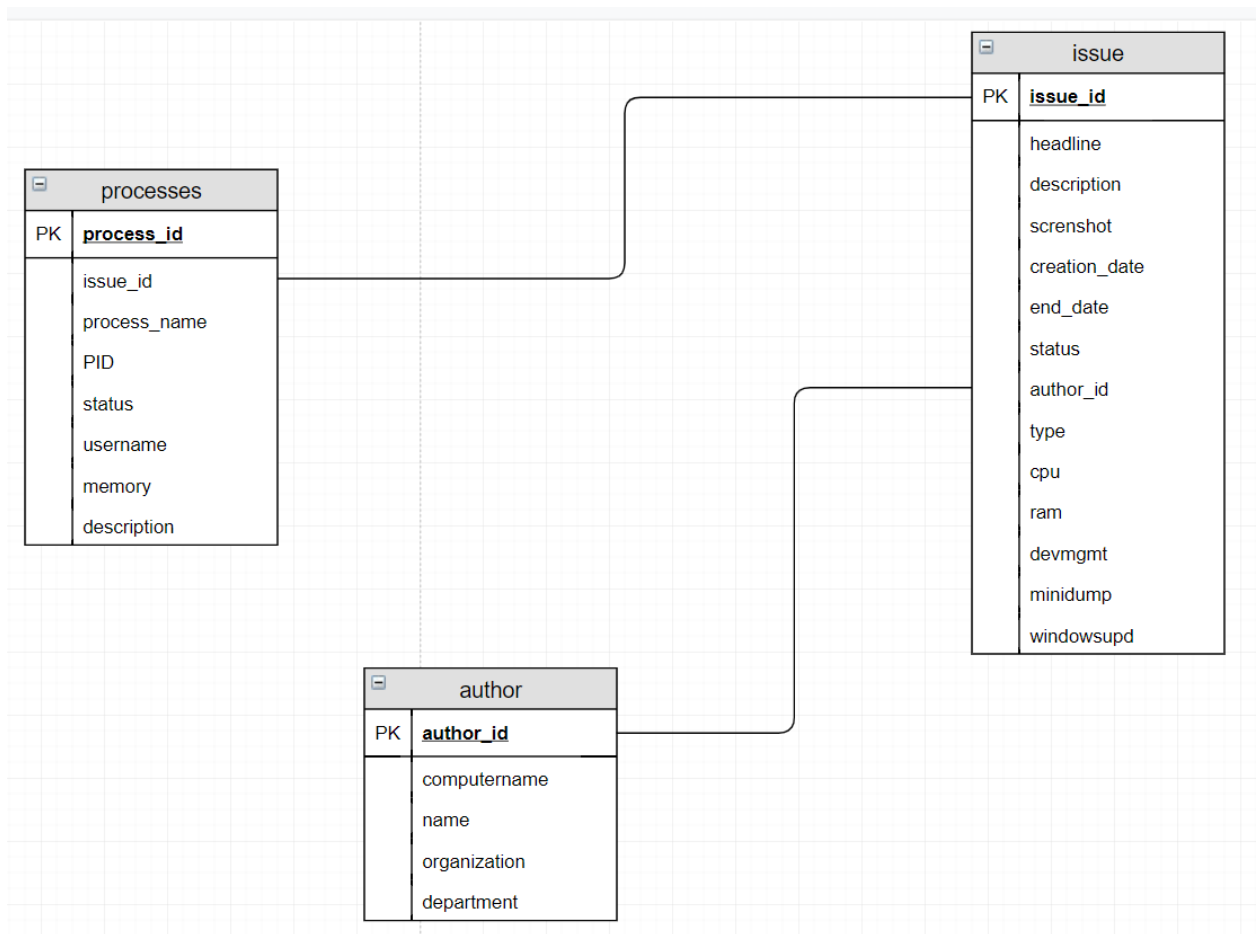


Рисунок 3.1 – Концептуальна модель бази даних

3.2 Логічне моделювання

Логічне моделювання - процес створення моделі, що використовується в бізнес-інформації, з урахуванням зворотної моделі організації даних, але незалежно від типу цільової СКБД та інших фізичних аспектів реалізації. На цій фазі створюється логічна модель даних на рівні моделі даних для частини досліджуваного підприємства шляхом уточнення та трансформації концептуальної моделі з урахуванням характеристик вибраної організації даних у цільовій СКБД [18]. Результат розробки логічної моделі бази даних продемонстрований на рис. 3.2.

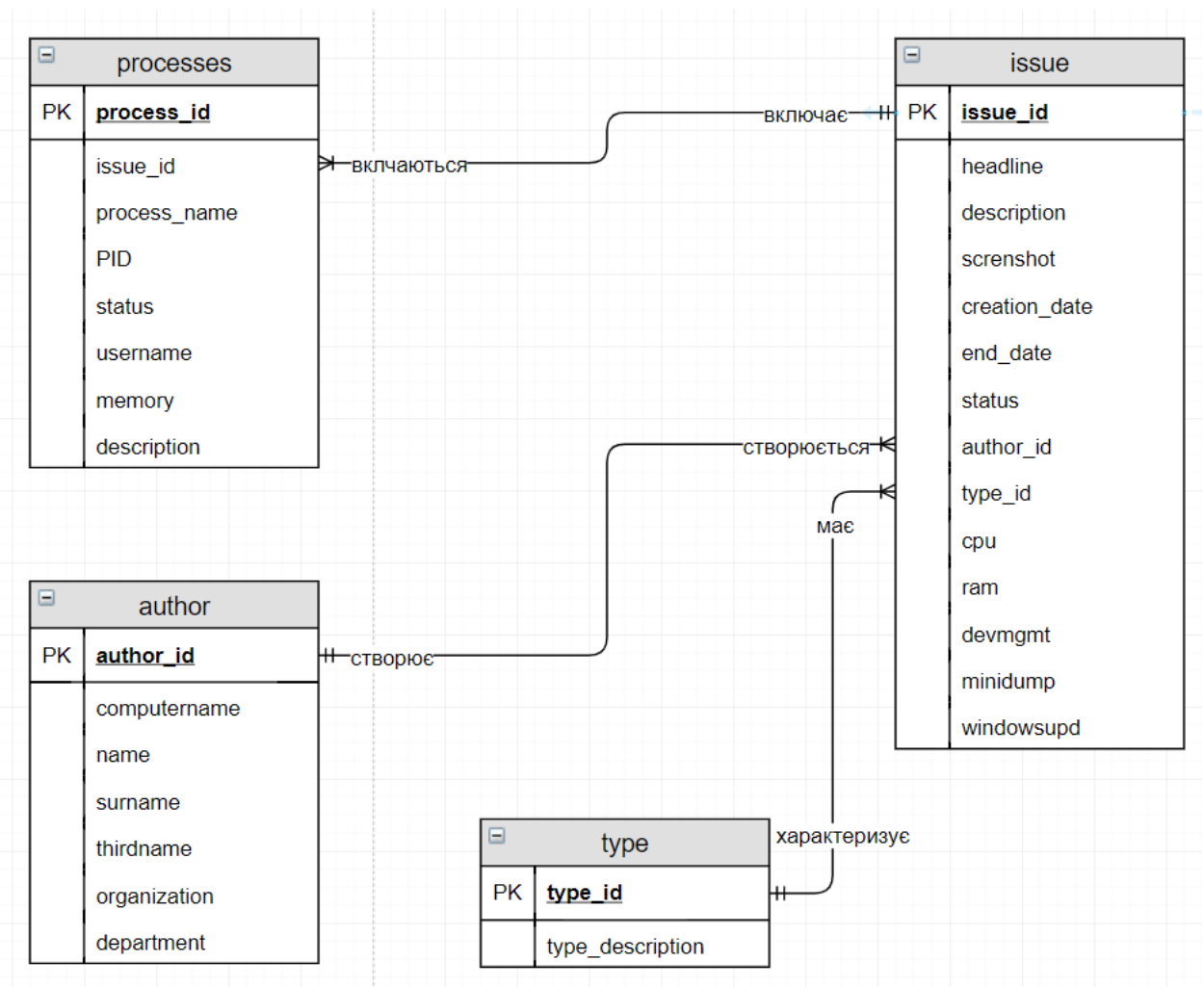


Рисунок 3.2 – Логічна модель бази даних

Визначення первинних та зовнішніх ключів:

Заявка (issue): issue_id – первинний, author_id, type_id – зовнішні;

Процеси (processes): process_id – первинний, issue_id – зовнішній;

Автор (author): author_id – первинний;

Тип (type): type_id – первинний.

3.3 Фізичне моделювання

Фізичне проектування створює опис реалізації бази даних на вторинних пристроях зберігання, включаючи структури зберігання і методи доступу для організації ефективної обробки даних. Фізична фаза конструкції дозволяє вибрати раціональну структуру зберігання і методи доступу на основі різноманітних методів і засобів, наданих розробникові конкретної СКБД [18]. Результати фізичного моделювання наведені у дампі бази даних у додатку Б.

Створення унікального індексу первинного ключа:

```
CREATE UNIQUE INDEX NDXCLASS ON ISSUE (issue_id);
```

```
CREATE UNIQUE INDEX NDXCLASSROOM ON PROCESSES (processes_id);
```

```
CREATE UNIQUE INDEX NDXHOURS ON AUTHOR (author_id);
```

```
CREATE UNIQUE INDEX NDXSCHEDULE ON TYPE (type_id);
```

3.4 Розробка діаграми варіантів використання

Діаграми варіантів використання (Use case diagram) описують взаємини і залежності між групами варіантів використання і дійових осіб, які беруть участь в процесі.

Важливо розуміти, що діаграми варіантів використання не призначені для представлення проекту і не можуть описувати внутрішню систему системи. Діаграми використання призначені для спрощення взаємодії з майбутніми користувачами системи та з клієнтами, зокрема, для визначення необхідних функцій системи. Іншими словами, діаграми використання вказують на те, що система повинна робити без вказівки найбільш часто використовуваних методів.

З точки зору поточної особи, випадок використання описує групу дій в системі, які призводять до конкретного результату.

Варіанти використання - це опис типових взаємодій між користувачами системи і самої системи, які відображають зовнішній інтерфейс системи і вказують форму, яку повинна виконувати система.

Дійова особа або актор є зовнішнім джерелом, який взаємодіє з системою через варіант використання. Дійові особи можуть бути як реальними людьми, так і іншими комп'ютерними системами або зовнішніми подіями.

Дійові особи представляють не фізичних осіб або системи, а їх ролі. Це означає, що коли людина взаємодіє з системою різними способами, вона виступає декількома діючими особами. Наприклад, людина, що працює в службі підтримки і приймає від клієнтів замовлення, буде відображатися в системі як «учасник відділу підтримки» і «учасник відділу продажів». [19]. Розроблена діаграма варіантів використання наведена на рис. 3.3.

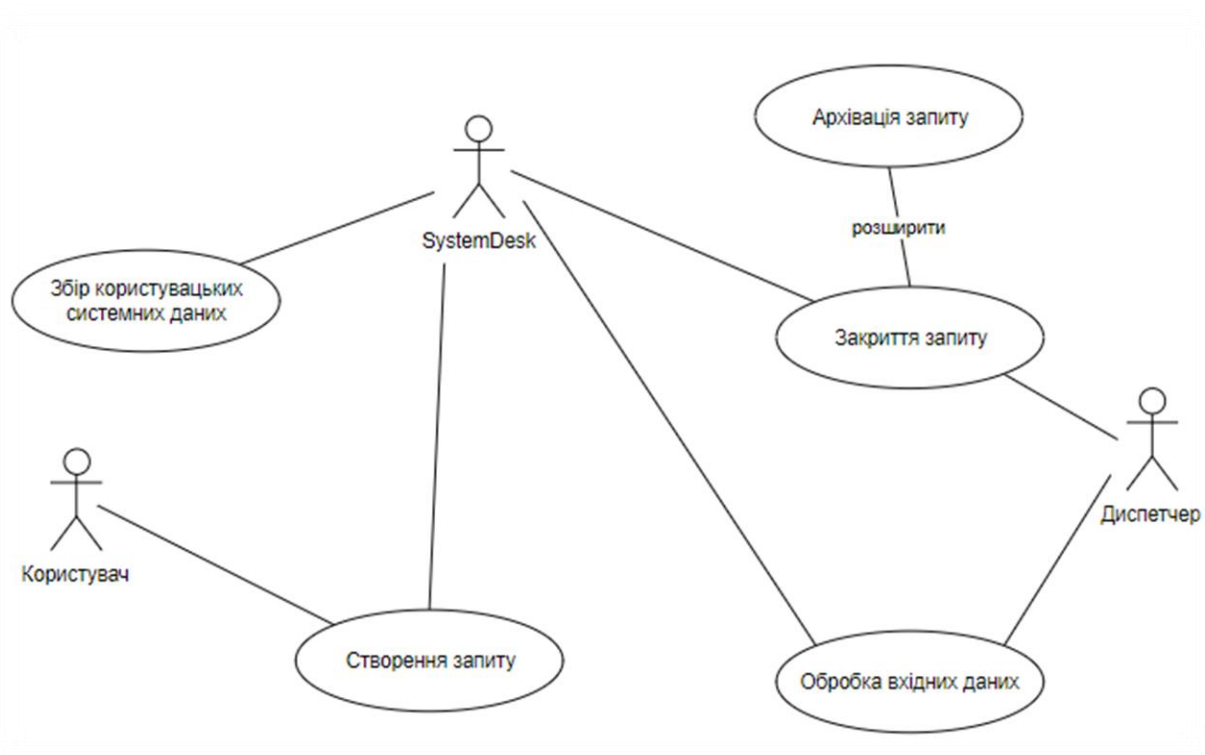


Рисунок 3.3 – Діаграма варіантів використання

3.5 Розробка діаграм нотацій моделювання IDEF0 та IDEF3

IDEF0 (Integration Definition for Function Modeling) – визначена нотація опису бізнес-процесів, що засновується на методології SADT [20].

SADT [21] (Structured Analysis and Design Technique, технологія структурного аналізу і проектування) - графічні позначення і комплексний підхід до опису систем. Розробка SADT має корінь в 1969 році і була випробувана практично в компаніях різних галузей.

IDEF0 є результатом програмної комп'ютеризації ВВП США. Автоматизація корпоративної діяльності вимагала відповідних методів і інструментів. Перед розробкою програмного забезпечення було необхідно чітко і зрозуміло описати бізнес-процеси. Інструменти, призначені для програмування завдань, також можуть бути корисні для адміністративних завдань. Примітка може бути використана для моделювання різноманітних автоматизованих і неавтоматизованих систем.

Ідея IDEF0 полягає в тому, що відображення бізнес-процесів представлене у вигляді прямокутника з вхідними та вихідними стрілками.

Для IDEF0 сторінка процесу та пов'язана стрілка є важливими:

- зліва - стрілка - вхід бізнес-процесу - інформація (документ), яка буде конвертована під час процесу;
- виходить стрілка - вихід бізнес-процесу - перетворена інформація (документ);
- Угорі розташована стрілка - Управління бізнес-процесами - інформація або документ, який визначає, яким чином буде виконуватися бізнес-процес, як вводити вивід;
- кінець стрілки - це механізм бізнес-процесу - який перетворює вхідні дані у висновок: співробітник або техніка. Передбачається, що механізм не змінюється протягом одного циклу процесу.

Перевага IDEF0 в тому, що вона показує взаємодію процесів взагалі без зайвих деталей. Недолік IDEF0 полягає у неможливості бачити алгоритм ведення бізнес-процесів. Потрібен певний тренінг для розробки та читання позначень.

Основними споживачами IDEF0 є керівники, які повинні бачити і розуміти зв'язок процесів, не ігноруючи дрібниці [20]. Контекстна діаграма та перший рівень декомпозиції у нотації IDEF0 наведені на рис. 3.4 та 3.5 відповідно.

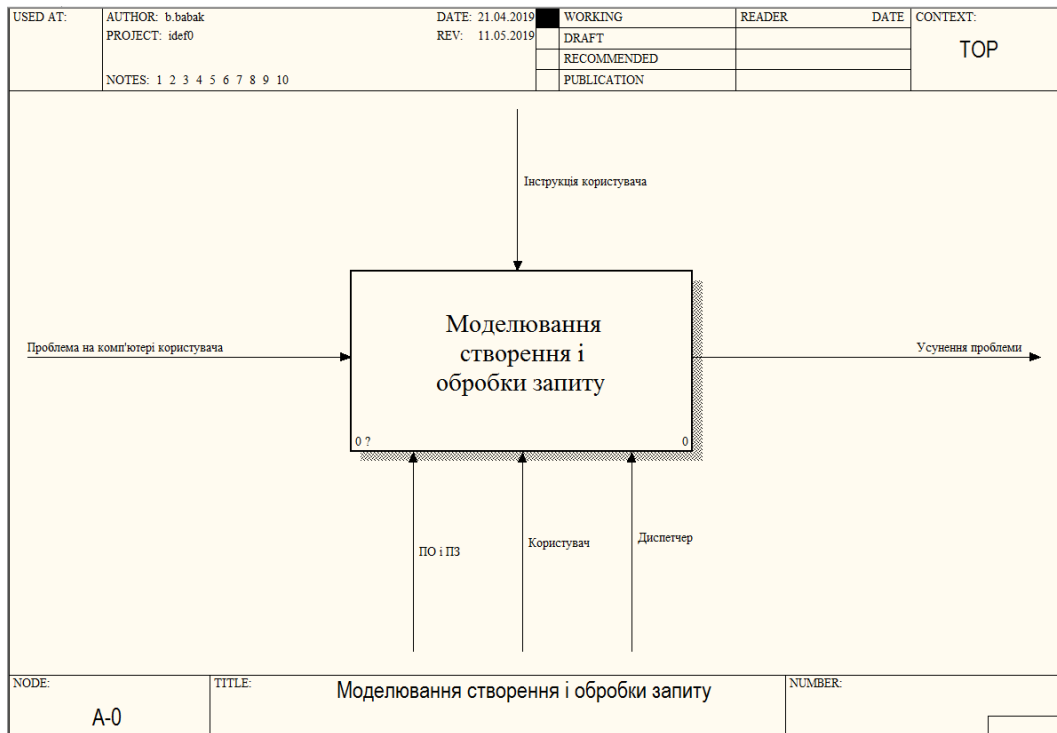


Рисунок 3.4 – Контекстна діаграма нотації IDEF0

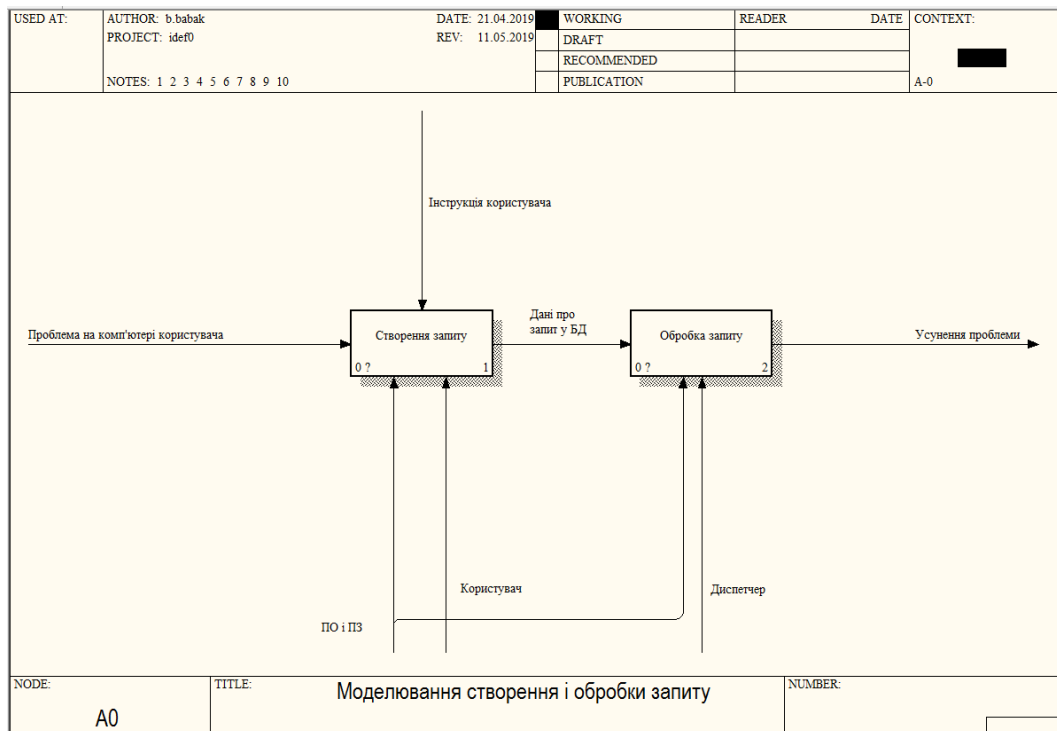


Рисунок 3.5 – Перший рівень декомпозиції нотації IDEF0

Нотація IDEF3 повинна описувати робочі процеси. IDEF3 часто використовується для створення моделей бізнес-моделей на найнижчому рівні - описує роботу, виконану у відділах та робочих місцях.

Основними графічними об'єктами моделі, використовуваної в IDEF3, є квадрати і стрілки. Перші використовуються для опису функцій (роботи, процесів), другі - для представлення тимчасової послідовності функцій або послідовності виконання функцій за рахунок потоку матеріальних ресурсів у моделі.

Щоб уникнути неоднозначності при описі робочих процесів, позначення IDEF3 включає додаткові об'єкти, які служать для представлення можливих розгалужень робочих процесів і об'єднання варіантів за певних умов. Ці об'єкти є логічними символами трьох типів:

- оператор «І»;
- оператор «АБО»;
- оператор – виключає «АБО».

На відміну від позначення IDEF0 в нотації IDEF3, чотиристороння сторінка, що представляє функцію (робота, процес), не використовується для зв'язування входів різних типів. Крім того, тільки одна стрілка може входити і виходити з площі. Інакше будуть порушені правила створення діаграм в IDEF3.

Доцільно використовувати позначення IDEF3 у відносно простих процесах на нижньому рівні розкладання, тобто процесах на рівні робочої станції. У цьому випадку схема процесу може служити основою для створення документів, що регулюють роботу виконавців. При цьому нотації досить складно створити комбіновані моделі, які поєднують опис робочих процесів і процесів управління цією роботою [20]. Декомпозиція процесів другого рівня у нотації IDEF3 зображена на рис. 3.6– 3.7.

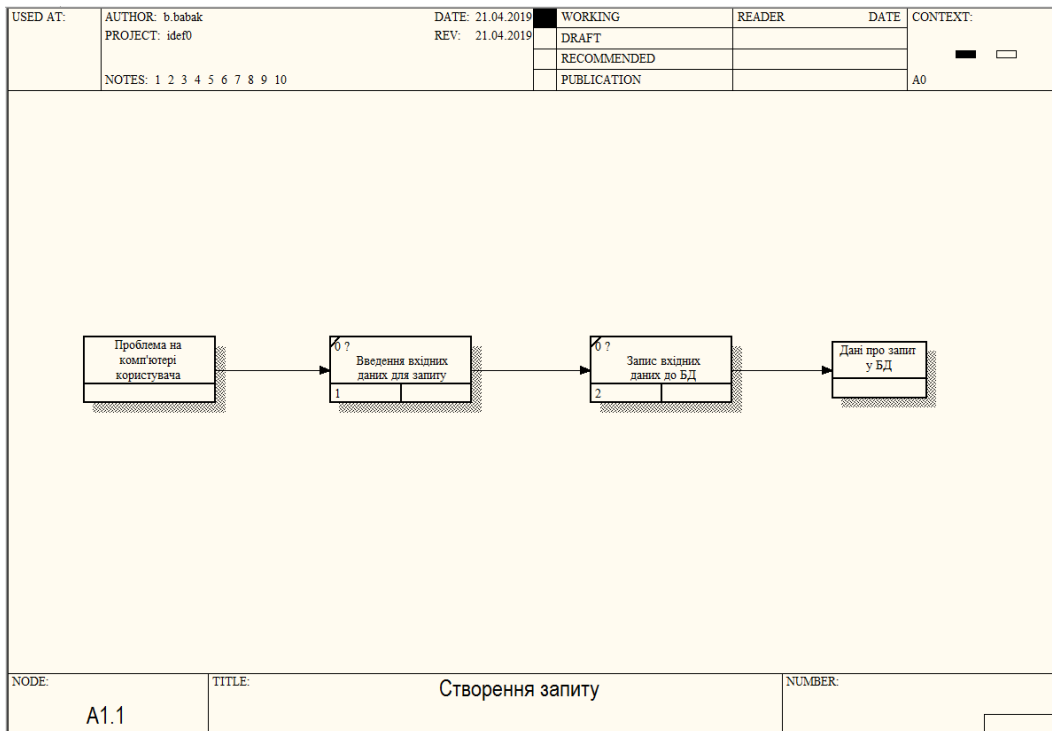


Рисунок 3.6 – Декомпозиція процесу «Створення запиту» нотації IDEF3

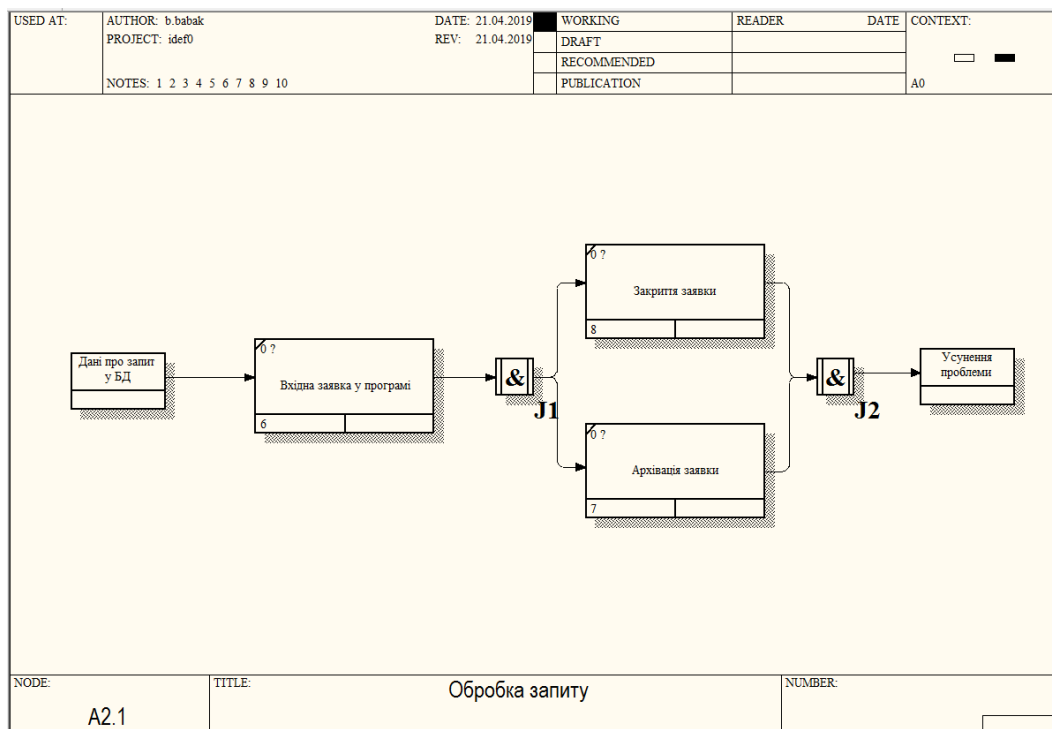


Рисунок 3.7 – Декомпозиція процесу «Обробка запиту» нотації IDEF3

3.6 Розробка схеми інформаційної технології

Інформаційна технологія дипломного проектування полягає у врахуванні системної інформації при автоматизованому формуванні заявки на техпідтримку, що дозволить оптимізувати ідентифікацію причини проблеми в великій частці випадків. Схема вирішення проблеми у техпідтримці (рис. 3.8) містить вхідні дані (опис проблеми користувачем), обробку даних диспетчером, отримання шаблонного рішення проблеми або отримання додаткових даних (системні дані комп'ютера користувача) для формування рішення проблеми. Інформаційна технологія дипломного проектування дозволяє автоматизувати етап отримання додаткових даних у великій частці випадків, у середньому пришвидшуючи процес вирішення проблеми заявки.

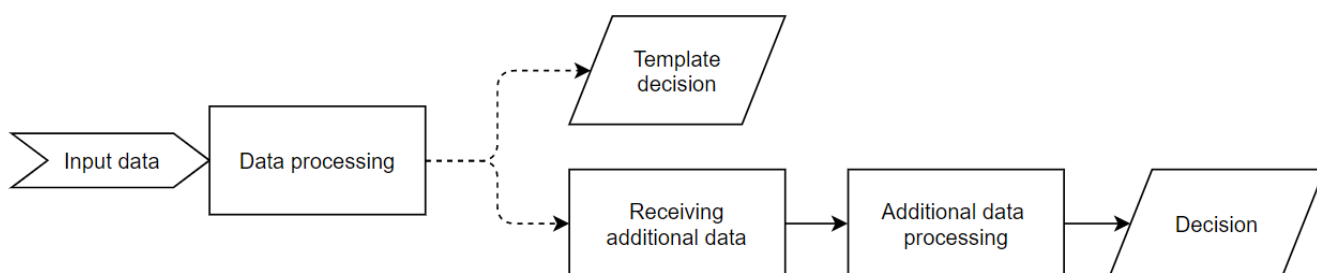


Рисунок 3.8 – Схема інформаційної технології

Архітектура програмного продукту, який реалізує розроблену ІТ (рис. 3.9), складається з користувацького інтерфейсу SystemDesk, а також адмінського додатку SystemDeskAdmin, що мають зв'язок з однією базою даних. Користувачі при виникненні проблем створюють заявку через інтерфейс, який в свою чергу потрапляє як запис у базі даних і, нарешті, відображається диспетчеру у вхідних заявках.

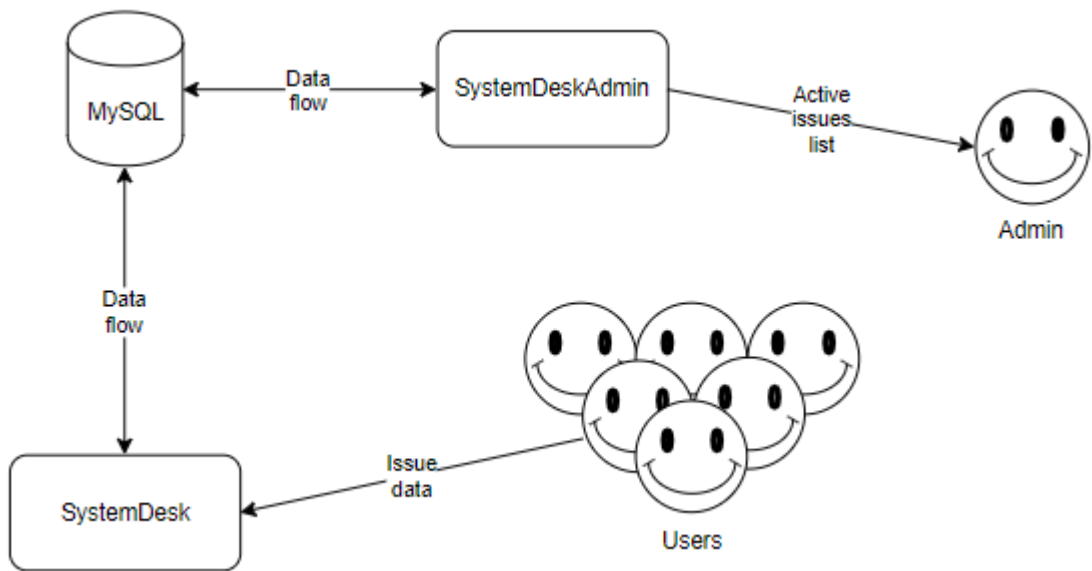


Рисунок 3.9 – Архітектура програмного продукту

В третьому розділі описані етапи концептуального, логічного та фізичного моделювання даних, що вміщує інформацію, необхідну для розкладу. Результати представлені у вигляді концептуальної та логічної моделей даних. Протягом моделювання бази даних були виділені основні об'єкти бази даних та їх атрибути, виділені зв'язки у базі даних, ідентифіковані первинні та зовнішні ключі. Був проведений аналіз даних та бізнес-процесів у нотаціях IDEF0, IDEF3 та BPMN, що дозволить визначити процеси, які будуть автоматизовані при виконанні проекту.

4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ ЗАПИТІВ ТЕХПІДТРИМКИ НА ПІДСТАВІ СИСТЕМНИХ ТА КОРИСТУВАЛЬНИЦЬКИХ ДАНИХ

4.1 Розробка бази даних

При створенні таблиць використовувалися стандартні конструкції мови SQL, які наведені в пункті Фізичне моделювання попереднього розділу.

Цілісність даних - це надійність і точність інформації, що зберігається в базі даних. Обмеження цілісності є правилами, які обмежують всі можливі стани бази даних і перехід з одного стану в інший. Таким чином, обмеження на цілісність визначає безліч "допустимих" станів і переходів між ними. База даних знаходиться в цілісному стані, якщо вона відповідає всім вимогам до її цілісності [22].

У дипломній роботі було використано цілісність даних як для об'єктів, так і для атрибутів та зв'язків.

Для реалізації обмежень цілісності відносин між відносинами використовувалися методи обмеження зовнішнього ключа (NOT NULL вказує, що з'єднання є обов'язковим) і цілісність зв'язків.

Для цілісності атрибутів визначено значення типів даних та їх розмірів, визначення необхідності значень та встановлення умов для значень атрибутів. Цілісність первинного ключа була застосована до цілісності відносин (атрибут первинного ключа не є NULL і не може бути дубльований у відносинах).

Схема реалізованої бази даних у веб-додатку phpMyAdmin 3.5.1 з відкритим кодом на мові PHP для адміністрування СКБД MySQL наведена на рис. 4.1. Дамп бази даних наведений у додатку В.

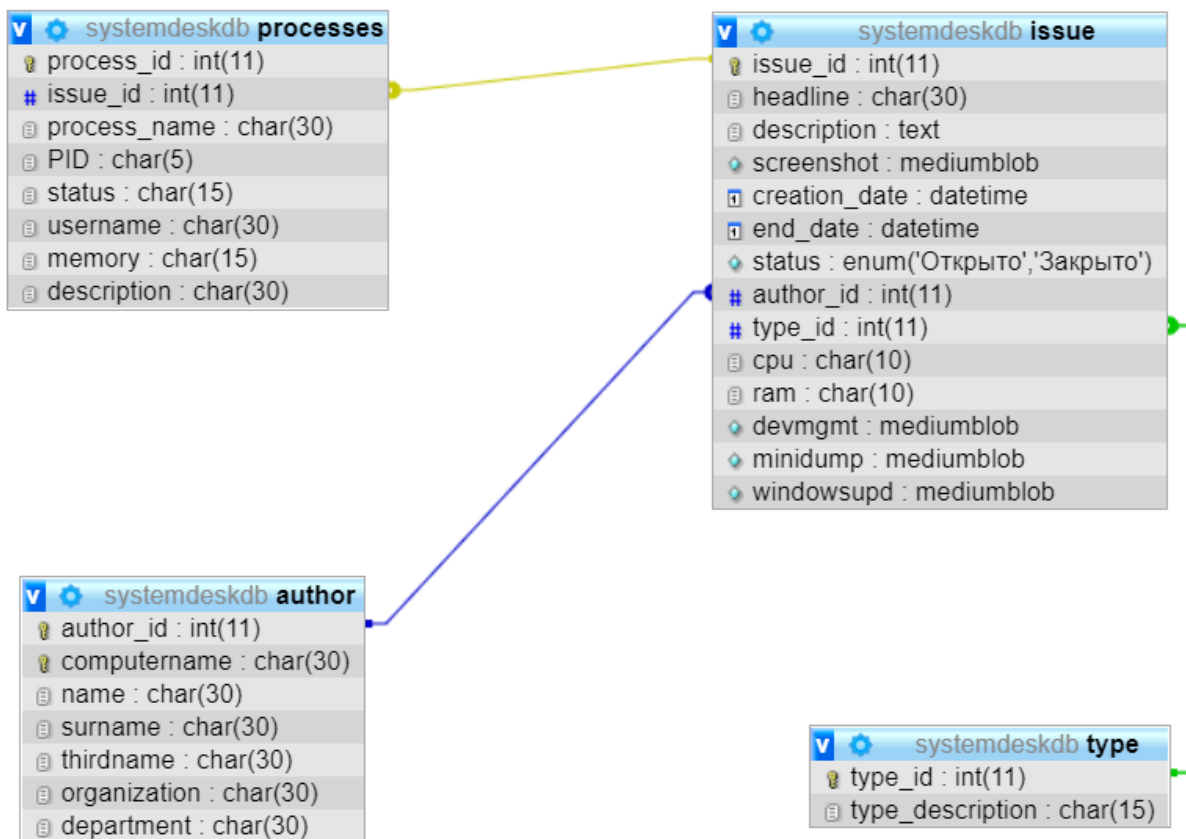


Рисунок 4.1 – Схема розробленої бази даних

4.2 Розробка програмного продукту

Наступним кроком після створення бази даних є створення користувацького інтерфейсу – програмного додатку Windows Forms мовою програмування C# на платформі Microsoft Visual Studio 2010.

Технологія Windows Forms є технологією, яка використовується в Visual C # для створення інтелектуальних клієнтських програм на основі Windows, які виконуються в середовищі .NET Framework. Windows Forms надає для проекту такі компоненти, як діалогові вікна, меню, кнопки і багато інших елементів управління, які є частиною стандартного інтерфейсу (UI) Windows. По суті, ці елементи управління є просто класами з бібліотеки .NET Framework. Для створення призначеного для користувача інтерфейсу використовується конструктор Windows

Forms, і користувач отримує доступ до інших можливостей часу розробки і часу виконання, в число яких входять такі:

- Розгортання ClickOnce.
- Велика підтримка баз даних, завдяки елементу управління DataGridView (Windows Forms).
- Панелі інструментів та інші елементи, призначені для користувача інтерфейсу, які можуть мати зовнішній вигляд і поведінку Microsoft Windows XP, Microsoft Office або Microsoft Internet Explorer [23].

Було розроблено 2 додатки – SystemDesk (користувацький) та SystemDeskAdmin (адмінський).

Користувацький додаток працює у фоновому режимі (рис. 4.2) і при виникненні необхідності користувач може розвернути вікно і ввести необхідні дані для створення заявки (рис. 4.3). У налаштуваннях програми відображується детальна інформація користувача, яка зберігається в реєстрі користувача (рис. 4.4).

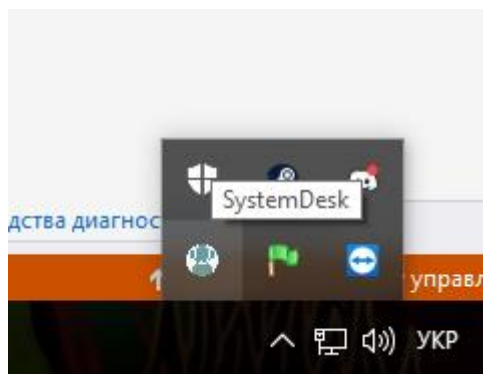


Рисунок 4.2 – Відображення програми у треї

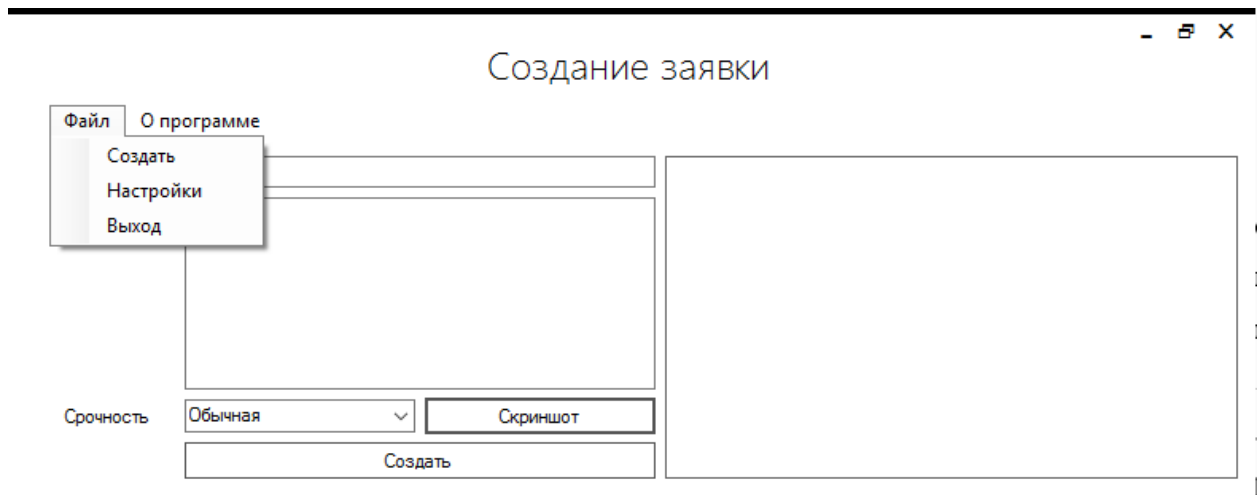


Рисунок 4.3 – Форма створення заявки

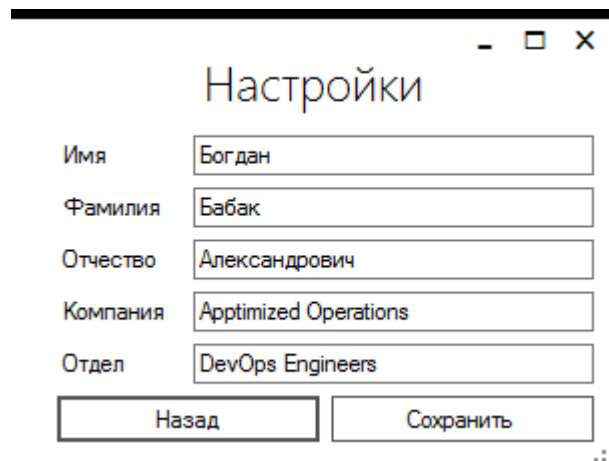


Рисунок 4.4 – Вікно налаштувань

Необхідні поля для заповнення форми «Створення заявки» (рис.4.5) – це заголовок, опис проблеми, терміновість заявки (звичайна та термінова) та можливість зробити скріншот за допомогою функціоналу, подібному до встановленої тули «Ножиці» (рис. 4.6). Після натиснення на кнопку «Створити» введена та системна інформація додається як запис у базі даних (рис. 4.7).

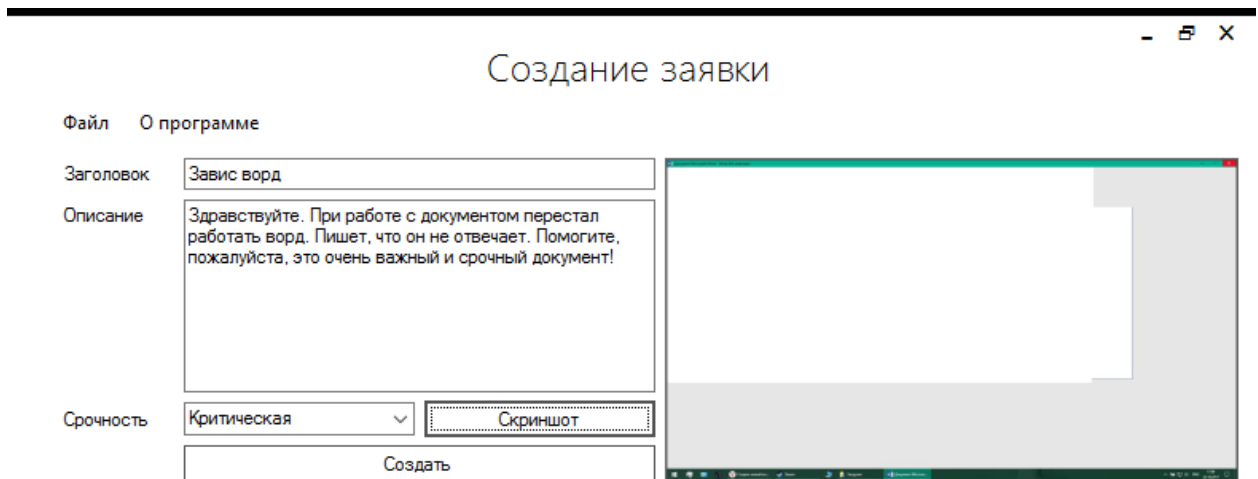


Рисунок 4.5 – Форма створення заявки із введеними даними

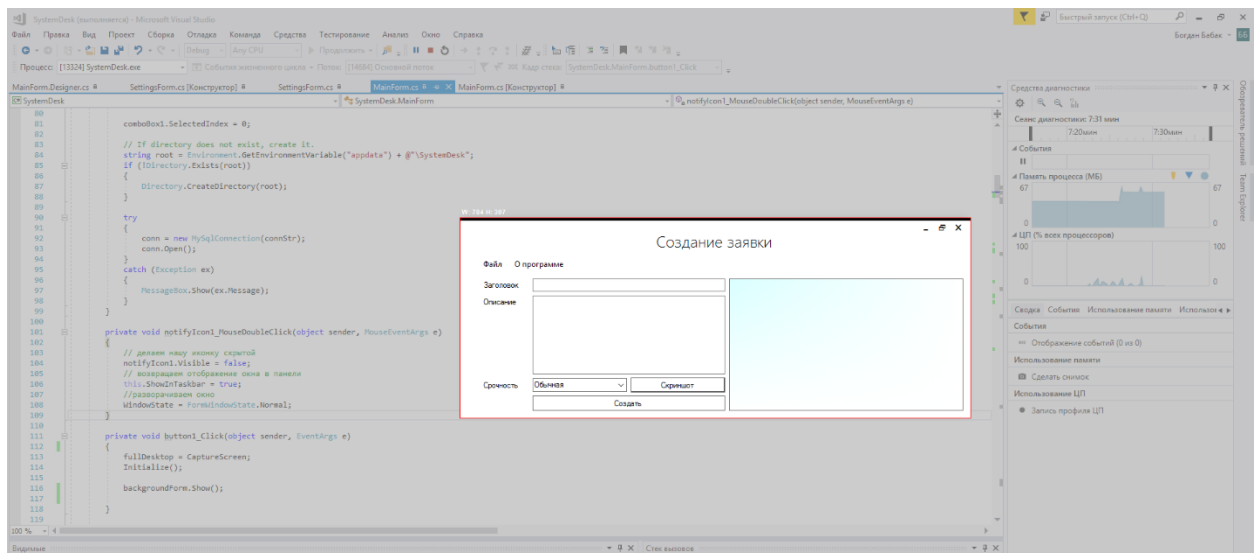


Рисунок 4.6 – Приклад роботи функціоналу створення скріншоту

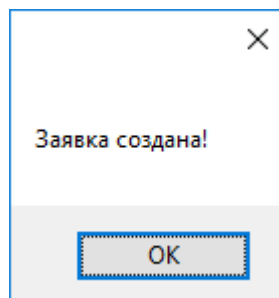
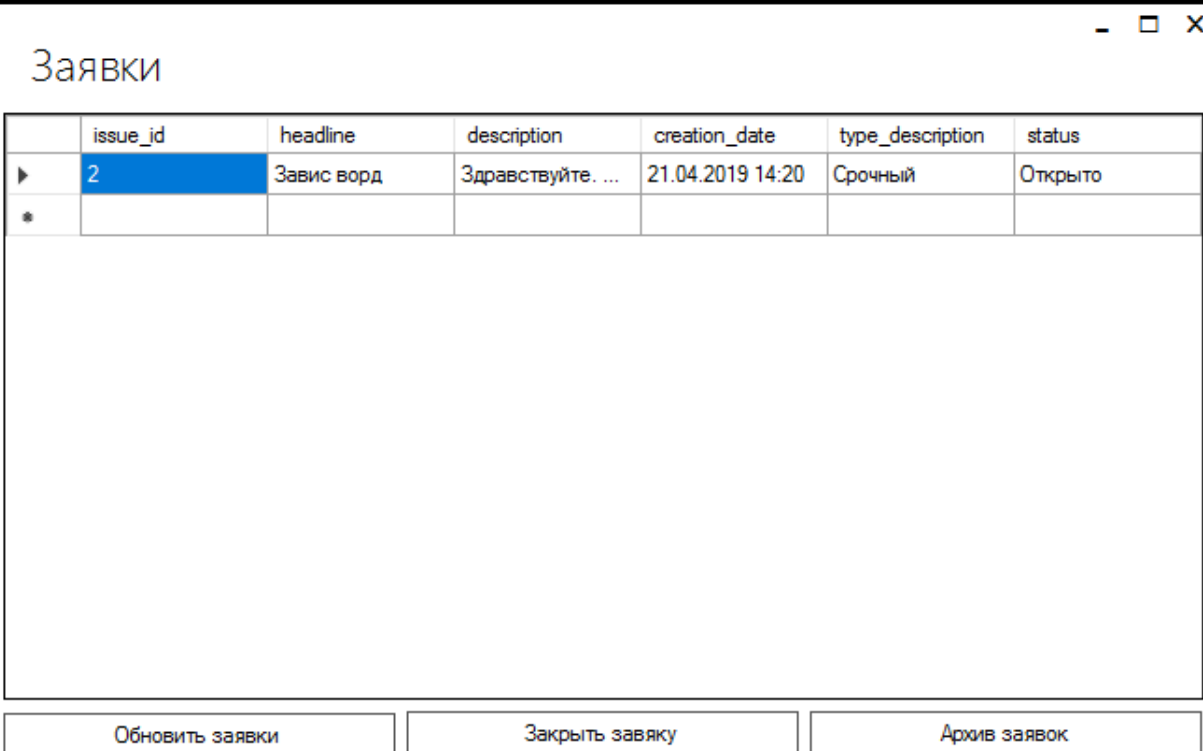


Рисунок 4.7 – Повідомлення про створення заявки

Адмінський додаток SystemDeskAdmin відображує вхідні відкриті заявки. Подвійним кліком диспетчер має змогу відкрити детальну інформацію про заявку (рис. 4.8).



| | issue_id | headline | description | creation_date | type_description | status |
|---|----------|------------|-------------------|------------------|------------------|---------|
| ▶ | 2 | Завис ворд | Здравствуйте. ... | 21.04.2019 14:20 | Срочный | Открыто |
| * | | | | | | |

Обновить заявки Закрыть заявку Архив заявок

Рисунок 4.8 – Форма відображення вхідних заявок

Форма «Деталі заявки» (рис. 4.9) включає у себе основну інформацію про заявку, що ввів користувач, скріншот, зроблений користувачем та системну інформацію, яка автоматизовано була збережена до бази даних. Системна інформація містить завантаження процесора та оперативної пам'яті та наступні файли: скріншот вікна «Диспетчер устроїв», необхідний для перевірки правильності підключення пристроїв до комп'ютеру, WindowsUpdateLog файл, що містить інформацію про коректність оновлення операційної системи, minidump файл, що містить інформацію про критичні завершення роботи комп'ютера, так звані «blue screen».

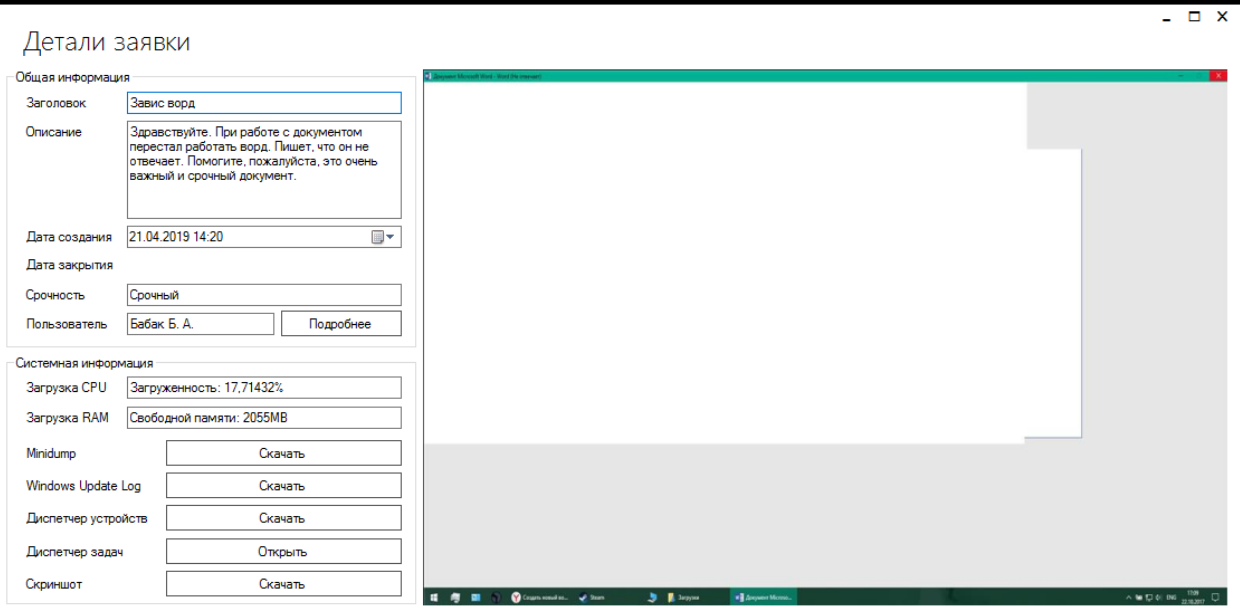


Рисунок 4.9 – Форма відображення деталей вибраної заявки

Усі файли разом із скріншотом можуть бути завантажені локально на комп'ютер диспетчера для детальної перевірки (рис. 4.10 – 4.12).

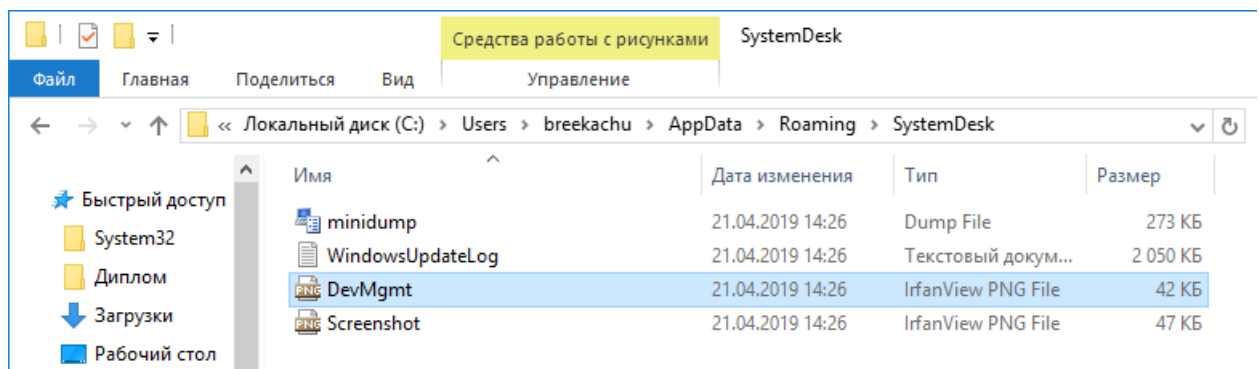


Рисунок 4.10 – Завантажені локально файли із бази даних

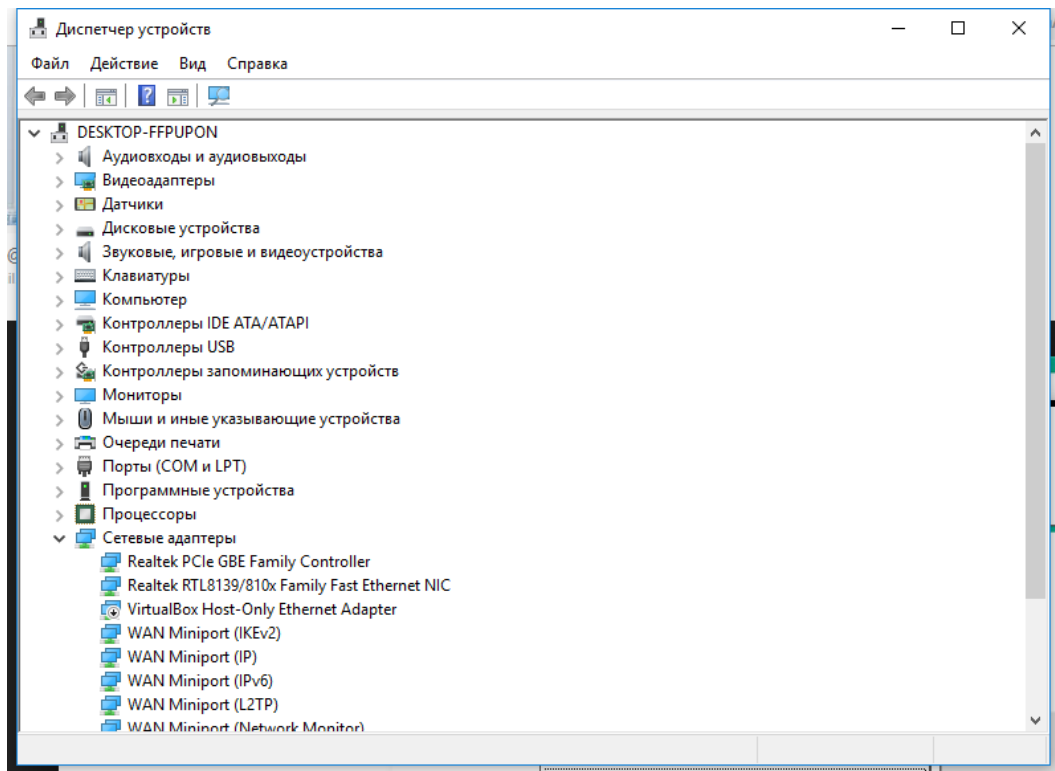


Рисунок 4.11 – Скріншот вікна «Диспетчер пристроїв»

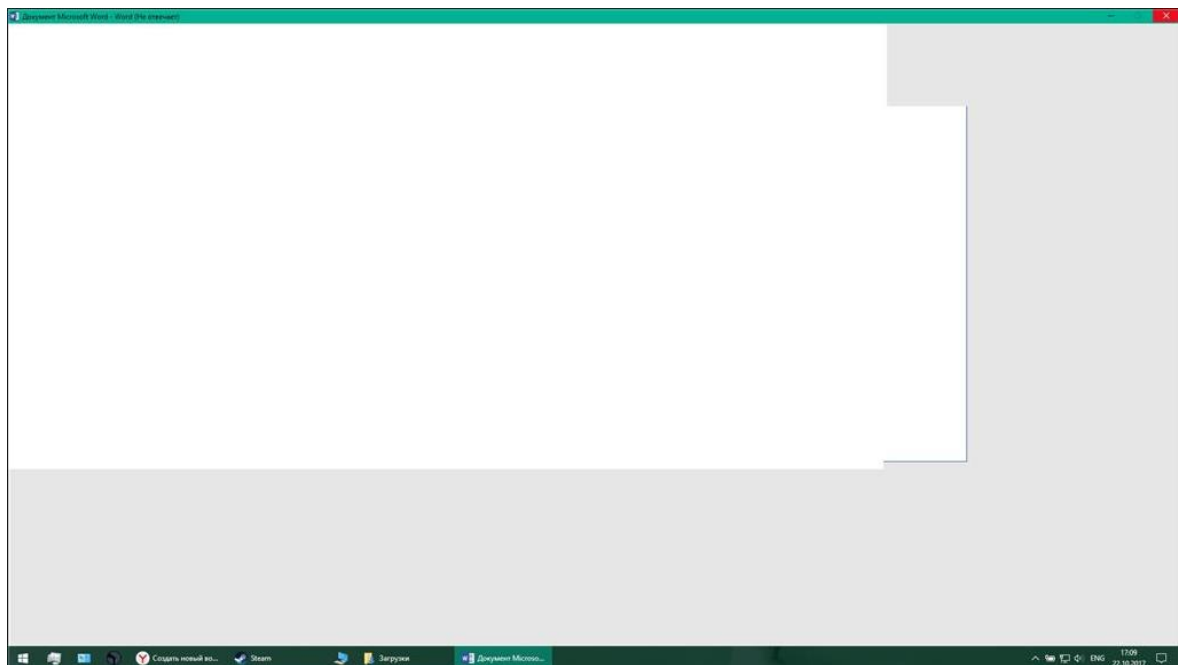
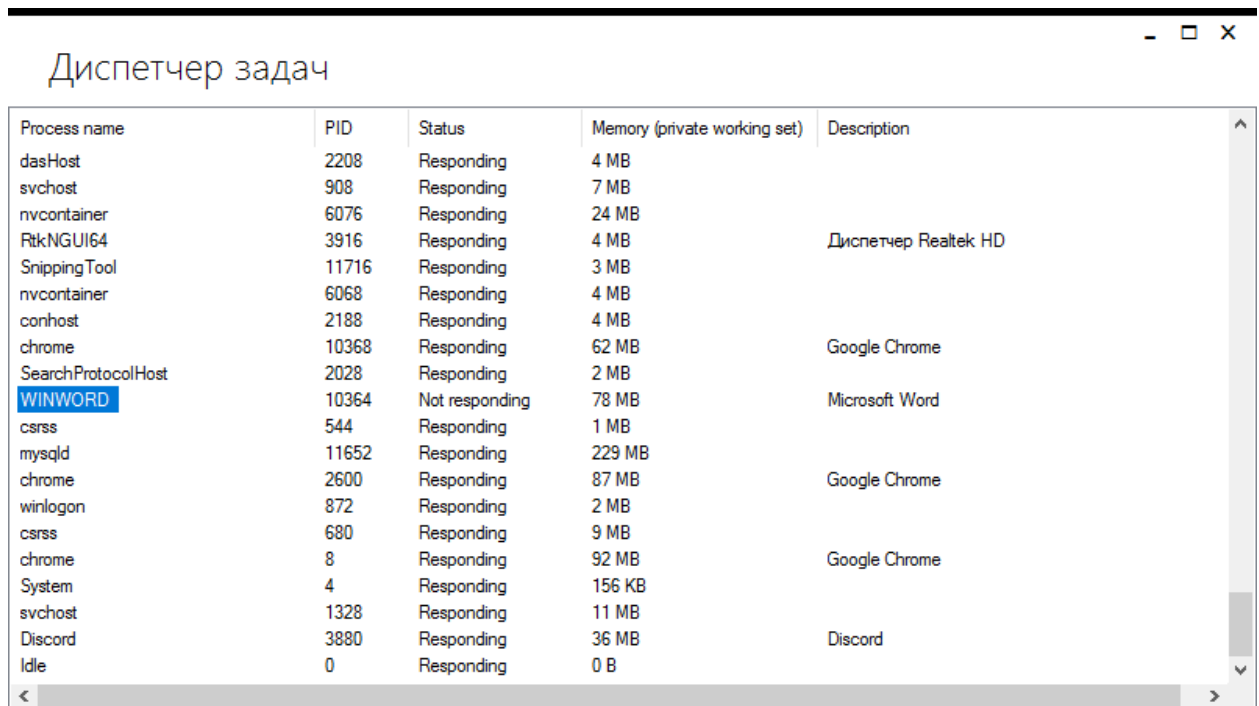


Рисунок 4.12 – Скріншот, створений у користувацькому додатку

Також диспетчер має змогу переглянути усі процеси, відкриті у користувача на момент створення заявки (рис. 4.13).



Диспетчер задач

| Process name | PID | Status | Memory (private working set) | Description |
|--------------------|-------|----------------|------------------------------|----------------------|
| dasHost | 2208 | Responding | 4 MB | |
| svchost | 908 | Responding | 7 MB | |
| nvcontainer | 6076 | Responding | 24 MB | |
| RtkNGUI64 | 3916 | Responding | 4 MB | Диспетчер Realtek HD |
| SnippingTool | 11716 | Responding | 3 MB | |
| nvcontainer | 6068 | Responding | 4 MB | |
| conhost | 2188 | Responding | 4 MB | |
| chrome | 10368 | Responding | 62 MB | Google Chrome |
| SearchProtocolHost | 2028 | Responding | 2 MB | |
| WINWORD | 10364 | Not responding | 78 MB | Microsoft Word |
| csrss | 544 | Responding | 1 MB | |
| mysqld | 11652 | Responding | 229 MB | |
| chrome | 2600 | Responding | 87 MB | Google Chrome |
| winlogon | 872 | Responding | 2 MB | |
| csrss | 680 | Responding | 9 MB | |
| chrome | 8 | Responding | 92 MB | Google Chrome |
| System | 4 | Responding | 156 KB | |
| svchost | 1328 | Responding | 11 MB | |
| Discord | 3880 | Responding | 36 MB | Discord |
| Idle | 0 | Responding | 0 B | |

Рисунок 4.13 – Форма відображення диспетчера задач користувача

Після завершення роботи над заявкою, диспетчер відповідною кнопкою помічає, що заявка закрита (рис. 4.14), а заявка в свою чергу переміщається з «Вхідних заявок» до «Архіву заявок» (рис. 4.15).

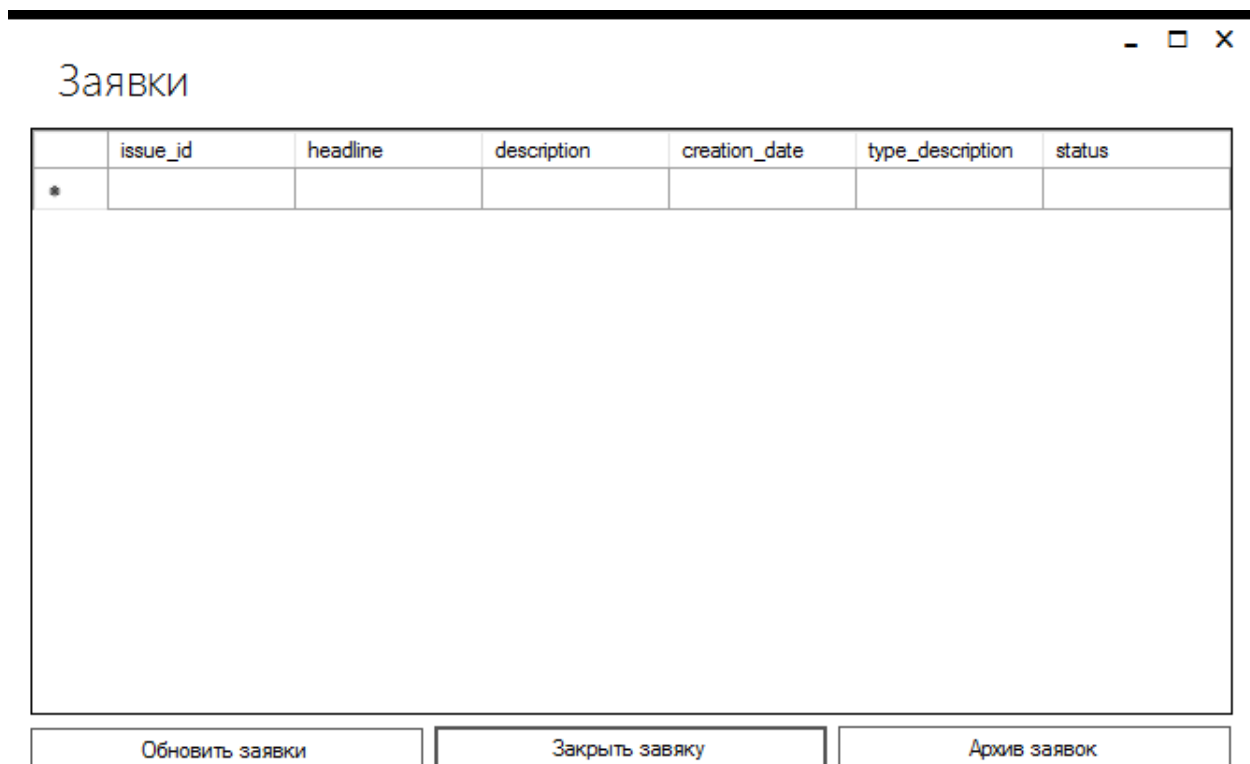


Рисунок 4.14 – Форма вхідних заявок після закриття єдиної заявки

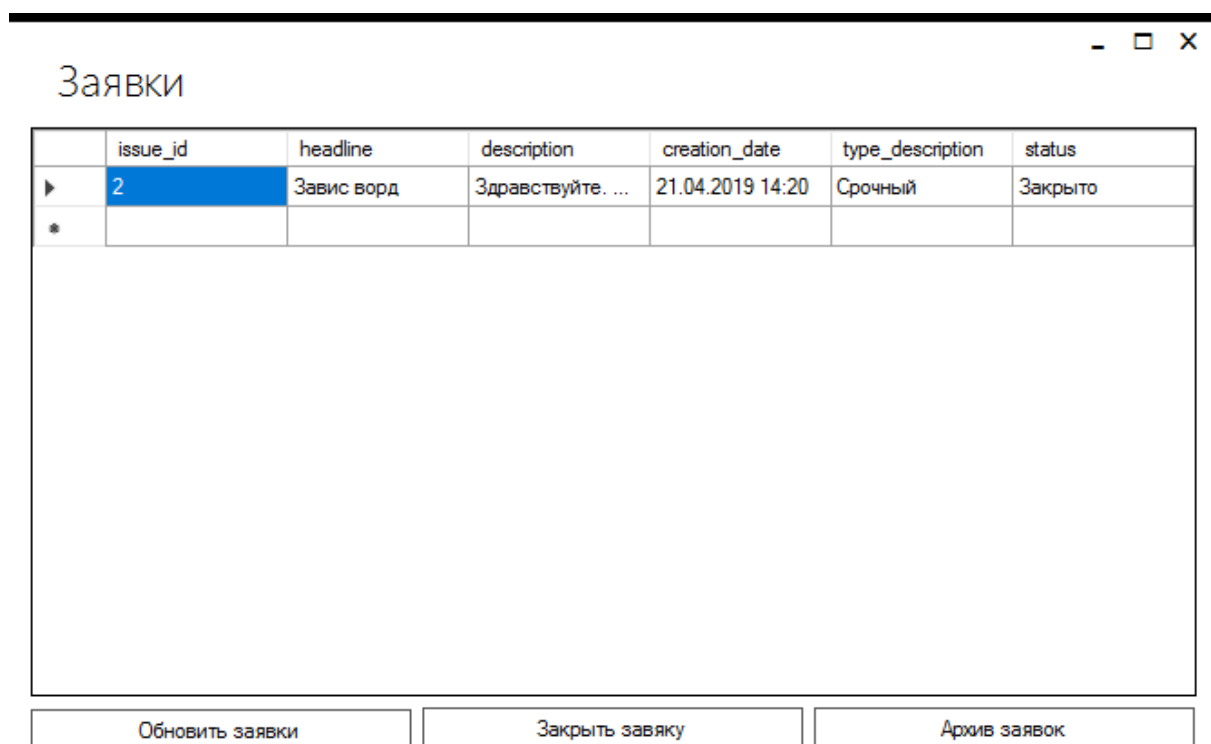


Рисунок 4.15 – Архів закритих заявок

4.3 Розробка інсталяційного пакету

Для використання інтерфейсу системи замовником, був сформований інсталяційний пакет засобами MS Visual Studio. На рис. 4.16 – 4.19 показаний процес інсталяції програмного додатку на тестовий комп'ютер.

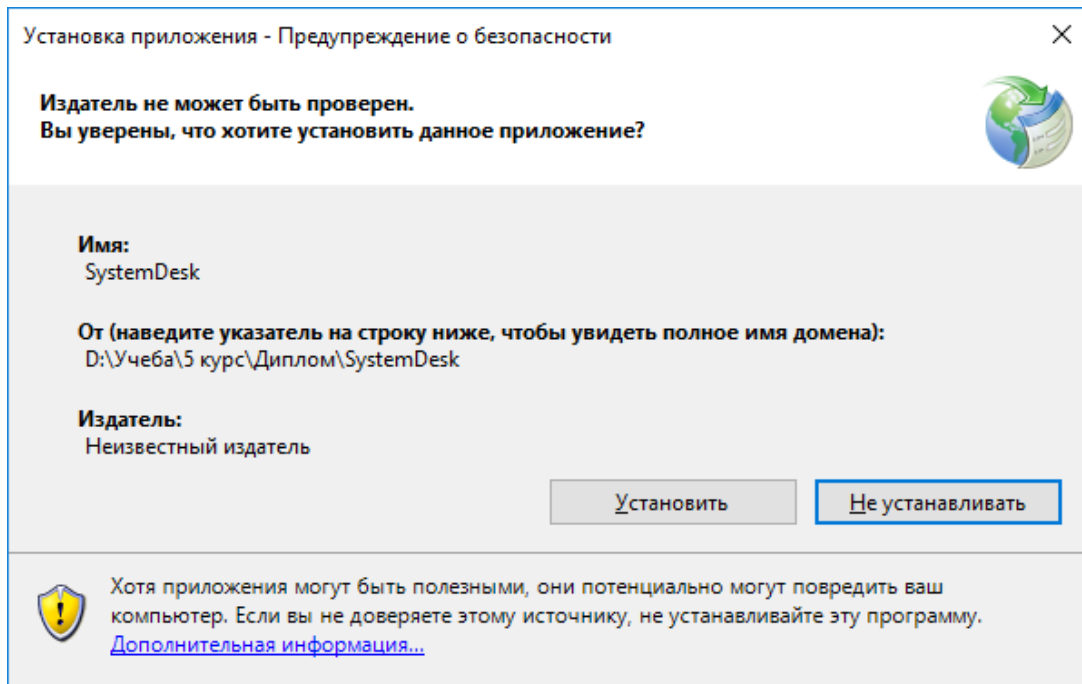


Рисунок 4.16 – Вікно інсталятора SystemDesk

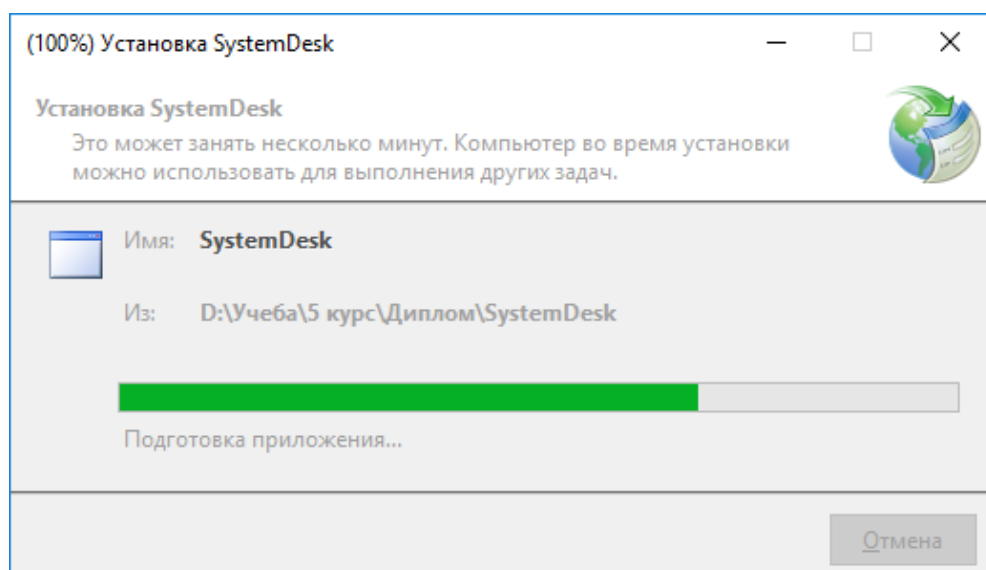


Рисунок 4.17 – Процесс установки SystemDesk

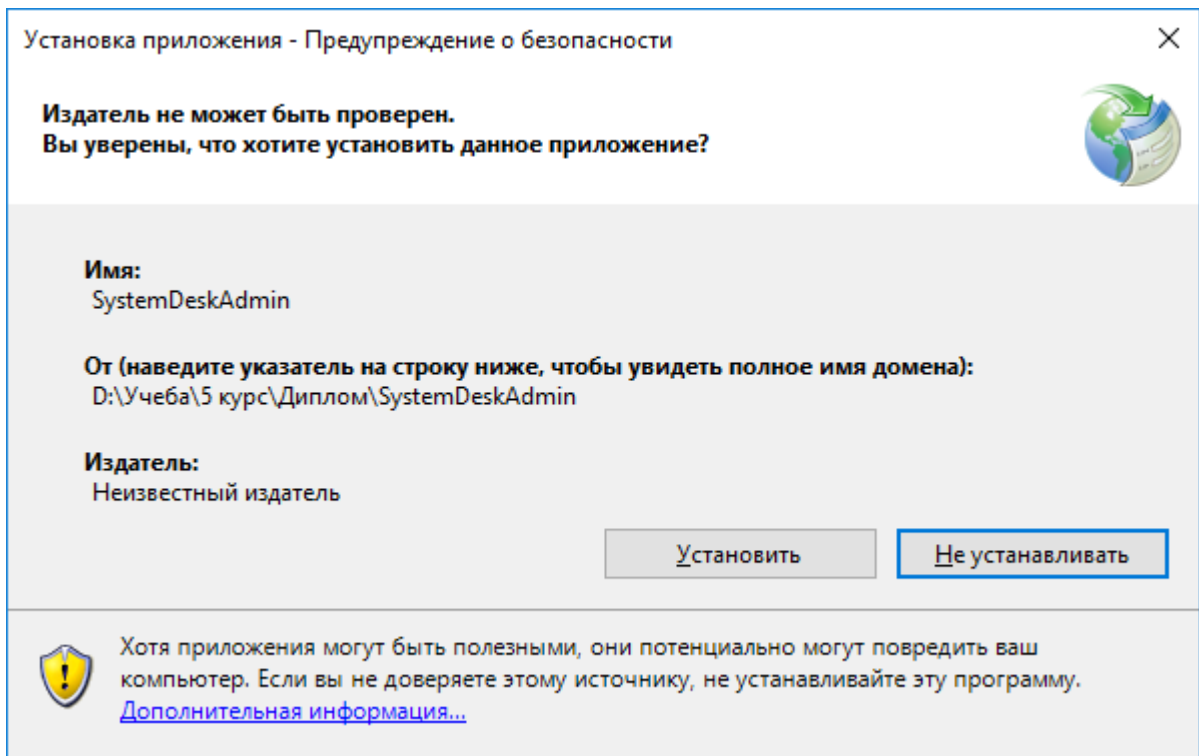


Рисунок 4.18 – Вікно інстальатора SystemDesk

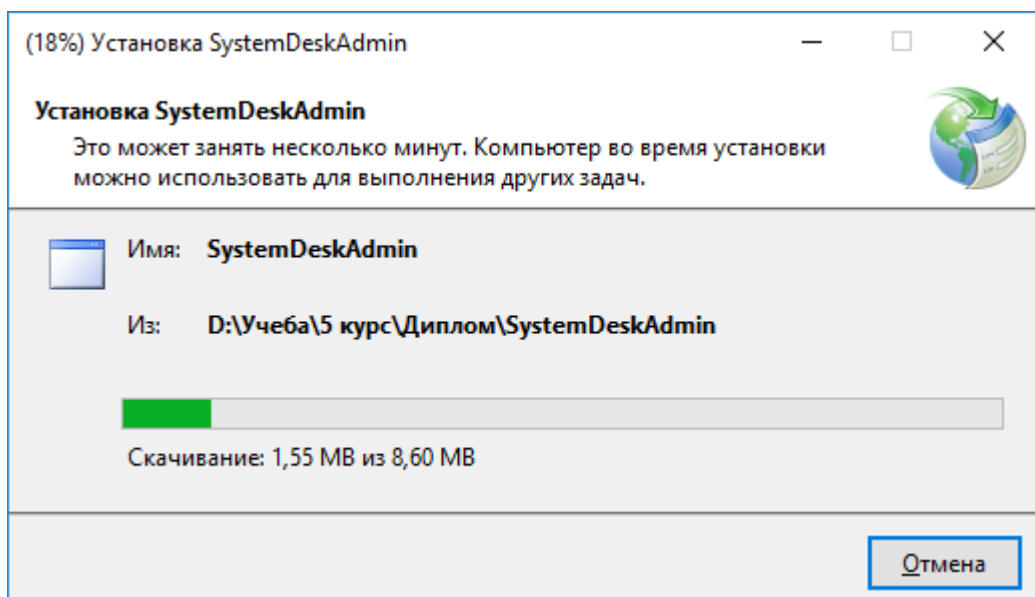


Рисунок 4.19 – Процес установки SystemDesk

В четвертому розляді проілюстровані результати реалізації продукту дипломного проектування, що реалізує розроблену інфомаційну технологію, база даних для зберігання інформації про об'єкти системи та користувацький інтерфейс, що дає змогу користувачу всеціло контролювати роботу додатку за допомогою зручної навігації між основними формами програми. Програмний код реалізованого інтерфейсу системи наведений у додаткових файлах дипломного проекту на диску.

ВИСНОВОК

Мета проекту полягає у розробці інформаційної технології формування запитів техпідтримки з метою оптимізації діяльності відділу технічної підтримки для підвищення якості його роботи і оптимізації витрат робочого часу співробітників. Інформаційна технологія дипломного проектування полягає у врахуванні системної інформації при автоматизованому формуванні заявки на техпідтримку, що дозволить автоматизувати етап отримання додаткових даних у великій частці випадків, у середньому пришвидшуючи процес вирішення проблеми заявки.

Для обґрунтування актуальності розробки ІТ було проаналізовано існуючі аналоги систем/сервісів за чотирма основними критеріями для сервісів HelpDesk: вартість, прийом запитів, політика SLA та автоматизація роботи з запитами.

Для розуміння необхідності автоматизації процесів було визначено основні методи наукового дослідження: спостереження та порівняння, результати якого показали, що показник завантаженості диспетчера і час обробки заявок є незадовільним з чого випливає, що процес потребує автоматизації.

Проаналізована література обумовила використання методів реалізації, засноване на стійкому розумінні переваг та недоліків даних методів. В якості СКБД була обрана «MySQL», так як вона характеризується великою швидкістю, стійкістю і простотою використання. В якості мови програмування була обрана мова «C#», через те, що вона заснована на строгій компонентній архітектурі і реалізує передові механізми забезпечення безпеки коду.

Для обґрунтування доцільності створення магістерської роботи була визначена її цінність, яка полягає у реалізації одного з можливих рішень проблеми нерациональної витрати робочого часу спеціалістів і задовільної якості обслуговування, що дозволяє знизити навантаження на робочий персонал відділу технічної підтримки та скоротити час обробки запитів до технічного відділу.

Для визначення ступеню перетворення, доповнення, конкретизації наукових даних була визначена наукова новизна кваліфікаційної магістерської роботи, яка полягає у розробленні інформаційної технології врахування системної інформації при автоматизованому формуванні заявки на техпідтримку, що дозволить оптимізувати ідентифікацію причини проблеми в великій частці випадків і, як результат, знизить час на обробку заявки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Понятие научной новизны.— Режим доступу:
<https://studfiles.net/preview/5331365/page:4/>
2. Системы Helpdesk (Service Desk).— Режим доступу:
<http://www.itsmonline.ru/helpdesk/>
3. Dione McBride. A Guide to Help Desk Technology, Tools & Techniques. — MI: Course Technology, 2000. — 358 с.
4. Donna Knapp. A Guide to Help Desk Concepts. — PA: Course Technology, 2003. — 351 с.
5. Сравнение систем HelpDesk.— Режим доступу:
<https://ergonotes.ru/sravneniy-sistem-helpdesk/>
6. Система поддержки принятия решений (сппр) "Выбор".— Режим доступу:
<https://studfiles.net/preview/994349/page:6/>
7. Научные методы исследования.— Режим доступу:
<http://idschool225.narod.ru/metod.htm>
8. Общие понятия моделирования.— Режим доступу: <https://studopedia.org/8-45254.html>
9. Система массового обслуживания.— Режим доступу: <https://lektsii.com/3-23174.html>
10. СМО с ожиданием (очередью).— Режим доступу:
https://math.semestr.ru/смо/смо_length.php
11. Корнієнко М. М. Інформатика. Бази даних. Системи управління базами даних./ М. М. Корнієнко, І.Д. Іванова. — Х.: Видавництво «Ранок», 2009. — 48 с.
12. Компания MySQL A.B. MySQL. Справочник по языку: Пер. з англ. — М : Видавничий дім «Вільямс» 2005. — 432 с.
13. Кей С. Хорстманн. Java SE 8. Вводный курс. — М.: «Вильямс», 2014. — 208 с.

14. Бьёрн Страуструп. Язык программирования C++. Специальное издание. — М.: Бином-Пресс, 2007. — 1104 с.
15. Албахарі, Дж. С# 3.0. Справочник: Пер. з англ./Дж. Албахарі, Б. Албахарі. — 3-є вид. — СПб.: БХВ-Петербург, 2009. — 944 с.: іл.
16. Westland, Jason. Project management life cycle: a complete step-by-step methodology for initiating, planning, executing and closing the project successfully / Jason Westland. — GL.: Bell & Bain, 2006. — 229 p.
17. Hunt A., Thomas A., Hargett M. Pragmatic Unit Testing in C# with NUnit. — 2 edition: Pragmatic Bookshelf, 2007 — 227 с.
18. Дунаев В.В. Базы данных. Язык SQL. — СПб.: БХВ-Петербург, 2006. — 288 с.: ил.
19. Орлов С.А. Программная инженерия. Учебник для вузов. — СПб.: Пітер, 2016. — 640 с.
20. Репін В.В., Єліферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов/ Володимир Репін, Віталій Єліферов. — М. Манн, Іванов і Фербер, 2013 — 544 с.
21. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. — М.: Финансы и статистика, 1998. — 176 с.
22. Карпова Т.С. Базы данных: модели, разработка, реализация / Т.С. Карпова. — СПб.: Питер, 2002. — 304с.: ил.
23. Нейгел К., Ивѐн Б., Глинн Дж., Уотсон К. С# и платформа .NET 4 для профессионалов.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2011. — 1440 с.: ил.
24. Хацін С.М. Управління інноваційними проектами: учбовий посібник/ Хацін С.М., Зозуля Д.М., Сафонов А.Є. — Ростов н/Д: Видавничий центр ДГТУ, 2013. — 226 с.
25. «Управління проектами: практичні аспекти реалізації стратегій регіонального розвитку», Рач В. А., Россошанська О. В., Медведева О. М. — Видавництво «К.І.С.», Умов. друк. арк. 22,25.

26. Мазур И.И., Шапиро В.Д., Ольдерогге Н.Г. М12 Управление проектами: Учебное пособие / И.И. Мазур. — 2-е изд. — М.: Омега-Л, 2004. — с. 664.
27. Рогачева Е. В. Интеллектуализация базы знаний систем Service Desk [Текст] / Е. В. Рогачева, В. В. Ломакин // Молодой ученый. — 2012. — №2. — С. 63-66.
28. Кирьянов В. Комплексный интерфейс Service Desk на базе технологии Web-сервисов // Byte, №6, 2007, стр. 72-76
29. Иванов Д.Б., Юрочкин А.Г. Использование систем класса Service Desk для повышения качества поддержки услуг // Интеллектуализация управления в социальных и экономических системах: Труды Всерос. конф. Воронеж, 2008
30. Программирование на языке высокого уровня. Часть 1. Среда Visual Studio: методические указания к лабораторным работам / Сост. В. И. Кручинин; Волгоград. гос. техн. ун-т. – Волгоград, 2009. – 30 с.
31. Программирование на языке высокого уровня. Часть 2. Язык С#: методические указания к лабораторным работам / Сост. В. И. Кручинин; Волгоград. гос. техн. ун-т. – Волгоград, 2009. – 31 с.
32. Пугачев С., Шериев А., Кичинский К. Разработка приложений для Windows на языке С#; БХВ-Петербург, 2013. - 416 с.

ДОДАТОК А ПЛАНУВАННЯ РОБІТ

Деталізація мети проекту методом SMART.

Продуктом дипломного проекту є створення додаткового функціоналу отримання основних системних та користувальницьких даних кожної заявки для системи HelpDesk. Результати деталізації методом SMART розміщені у табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

| | |
|----------------------------------|---|
| Specific (конкретна) | Створити додатковий функціонал отримання основних системних та користувальницьких даних кожної заявки для зниження завантаження диспетчера |
| Measurable (вимірювана) | Результатом роботи проекту є оцінка замовника |
| Achievable (досяжна) | Реалізації системи здійснюється за допомогою середовища розробки Microsoft Visual Studio на мові програмування С# |
| Relevant (реалістична) | У наявності є всі необхідні технічні та програмні засоби. Розробники достатньо кваліфіковані для виконання поставлених задач. |
| Time-framed (обмежена у часі) | Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником проекту. Проект повинен бути виконаний згідно з календарним планом. |

Планування змісту структури робіт

WBS – це графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з продуктом проекту. На верхньому першому рівні WBS фіксується продукт проекту. Він повинен відповідати продукту проекту.

Наступний II рівень відповідає діям або основним заходам для досягнення продукту проекту.

Потім триває розбивка цих дій доти, поки не відбувається виконання дій елементарних робіт. Елементарні роботи – це роботи, які мають один чіткий результат, який використовується при прийнятті цієї роботи; на які призначений один конкретний відповідальний; на неї можна обчислити витрати праці і тривалість виконання. Зазвичай декомпозиція завершується тоді, коли для розкриття змісту потрібні вузькі фахівці, що знають технологічні особливості їх виконання.

В одному проекті кількість рівнів за різними гілками WBS може бути різною. На верхньому рівні для позначення елементів зазвичай використовують іменники (система, підготовлена група персоналу, модель тощо) але внизу переходять до віддієслівних іменників (написання n сторінок керівництва користувача, вивчення предметної галузі тощо). Важливо пам'ятати, що при побудові WBS варто ставити одну єдину ціль – визначити всі роботи, які необхідні для виконання проекту. У WBS часовий фактор не враховується.

Потрібно розуміти, що роботи нижнього рівня WBS названі елементарними з погляду того, що для цілей управління проектами вини діла не деталізуються. Насправді ці роботи також можуть бути розділені ще на більш дрібні і доведені до операцій. Але для цього потрібні специфічні технологічні знання. Це можна зробити на робочому місці їх виконання [24].

WBS-структура проекту наведена на рис. А.1.

Планування структури організації, для впровадження готового проекту (OBS)

Після побудови WBS розробляють організаційну структуру виконавців.

OBS структура проекту – організаційна структура виконавців (організацій) проекту. Визначається за переліком пакетів робіт нижнього рівня кожної гілки WBS-структури. Представляється відповідальними (відповідальні – це не обов'язково керівники організацій(відділів), а ті люди які безпосередньо організують виконання робіт) за виконання пакетів робіт.

Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту. На верхньому рівні OBS розташована команда проекту. На наступному рівні фіксуються виконавці: організації, відділи тощо. Потім, рівнем нижче, для кожного виконавця вказують прізвища конкретних осіб, які будуть відповідати за виконання елементарних робіт WBS. Потрібно пам'ятати, що відповідальні – це не обов'язково керівники, а ті співробітники, які безпосередньо організують і відповідають у виконавця за виконання елементарної роботи, зазначеної у WBS. Для них ця елементарна робота також є проектом (у порівнянні з загальним проектом). Для себе вони також можуть побудувати WBS-структуру й застосовувати інші інструменти планування. Саме на цьому рівні закладається певна якість майбутнього програмного продукту [24]. OBS-структура проекту наведена на рис. А.2.

Команда IT-проекту:

1. Виконавець проекту
2. Замовник проекту
3. Керівник проекту
4. Спеціаліст з аналізу бізнес-процесів
5. Спеціаліст з програмування

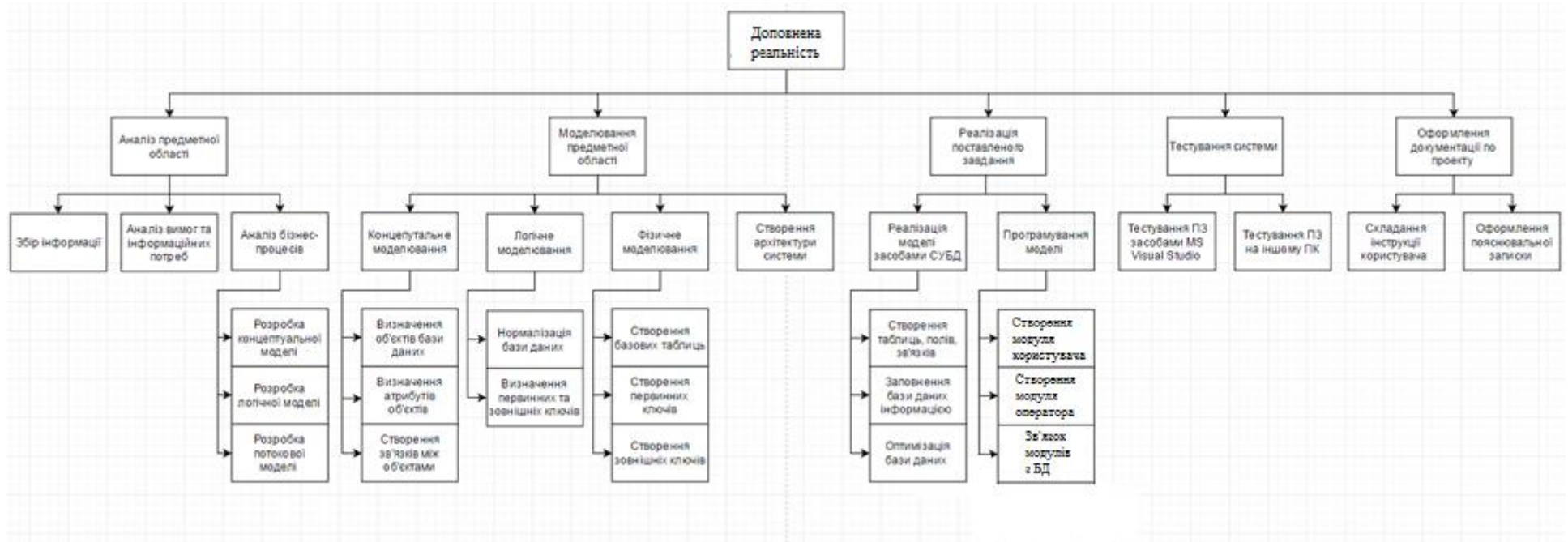


Рисунок А.1 – Структура роботи IT-проекту (WBS)

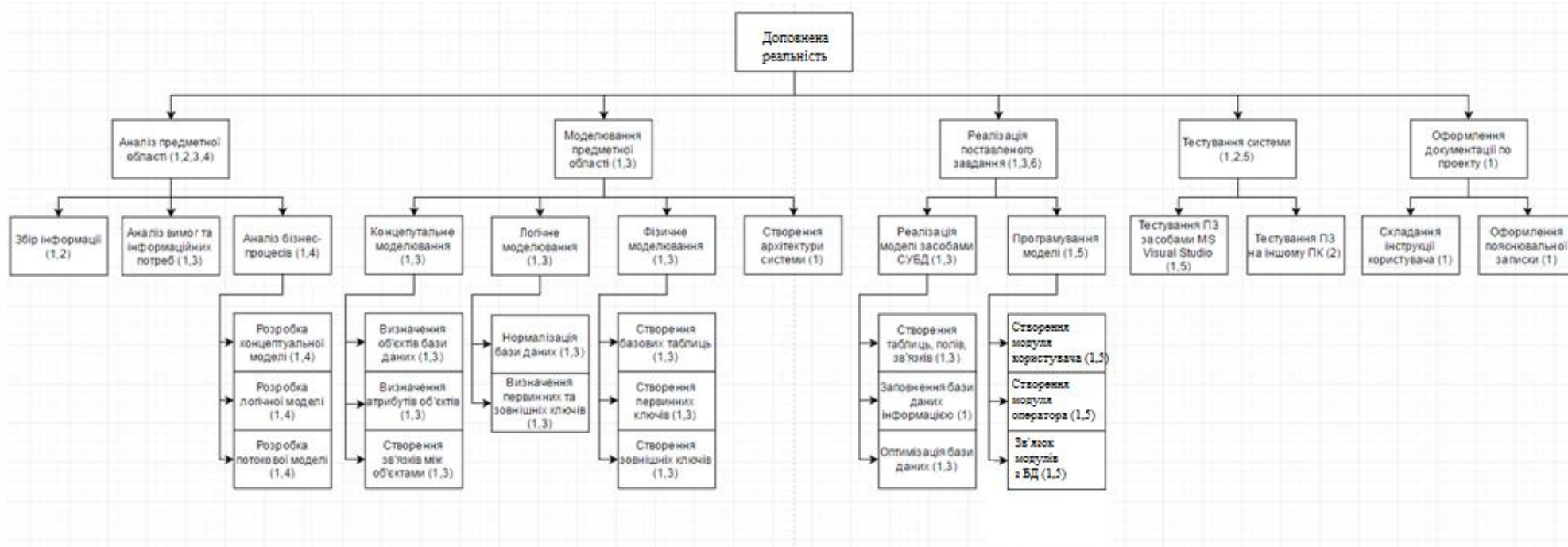


Рисунок А.2 – Структура роботи ІТ-проекту з урахуванням відповідальних (OBS)

Побудова матриці відповідальності

На підставі OBS та WBS структур будують матрицю відповідальності проекту. Матриця відповідальності закріплює за кожною елементарною роботою виконавця.

Тобто матриця відповідальності дозволяє відповісти на запитання: «Хто відповідає за виконання кожної елементарної роботи?». Потрібно пам'ятати, що за одну елементарну роботу може бути тільки один відповідальний. Часто в лінійному графіку відповідальності, крім виконавців, додають інших учасників проекту. При цьому по кожній елементарній роботі їм доручається або функція контролю, або OBS консультування тощо.

Потім для кожної елементарної роботи з урахуванням технічних і технологічних критеріїв до продукту проекту і його складових частин розроблюються робочі завдання. Їх призначення – забезпечення максимальної ймовірності того, що виконавці представлять саме той продукт (або послугу), що був закладений при складанні WBS. Тобто робоче завдання використовується як основа взаєморозуміння між відповідальним та командою проекту. Робоче завдання включає детальний опис елементарної роботи і докладний технічний опис результату роботи, що далі буде використовуватись при проведенні процедури приймання робіт. Воно повинно бути настільки чітким, повним і коротким, наскільки це можливо.

На підставі робочого завдання виконавці визначають: чи здатні вони виконати цю роботу й надати необхідний продукт (послугу) потрібної якості; скільки на це потрібно часу; як багато це буде коштувати. Для відповіді на ці питання виконавці будують для елементарної роботи приватні сітьові моделі й детальний календарний план [24]. Матриця відповідальності зображена на рис. А.3.

| | Бабак Б.О. | Замовник | Марченко А.В. | Концевич В.Г. | Ващенко С.М. |
|--|------------|----------|---------------|---------------|--------------|
| Збір інформації | | | | | |
| Аналіз вимог та інформаційних потреб | | | | | |
| Розробка концептуальної моделі | | | | | |
| Розробка логічної моделі | | | | | |
| Розробка потокової моделі | | | | | |
| Визначення об'єктів бази даних | | | | | |
| Визначення атрибутів об'єктів | | | | | |
| Створення зв'язків між об'єктами | | | | | |
| Нормалізація бази даних | | | | | |
| Визначення первинних та зовнішніх ключів | | | | | |
| Створення базових таблиць | | | | | |
| Створення первинних первинних ключів | | | | | |
| Створення зовнішніх ключів | | | | | |
| Створення архітектури системи | | | | | |
| Створення таблиць, полів, зв'язків | | | | | |
| Заповнення бази даних інформацією | | | | | |
| Оптимізація бази даних | | | | | |
| Створення модуля користувача | | | | | |
| Створення модуля оператора | | | | | |
| Зв'язок модулів з БД | | | | | |
| Тестування ПЗ засобами MS Visual Studio | | | | | |
| Тестування ПЗ на іншому ПК | | | | | |
| Оформлення пояснювальної записки | | | | | |
| Складання інструкції користувача | | | | | |

Рисунок А.3 – Матриця відповідальності

Побудова календарного графіку виконання ІТ-проекту

Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі, а також, проекту в цілому з урахуванням вихідних та святкових днів, будують календарний графік робіт. Він є реальним розподілом робіт з пакету за календарними датами, тобто своєрідним розкладом виконання робіт.

Графік Ганта є достатньо зручним для користування. Його будують так. На горизонталі фіксують календар у тих одиницях часу, які обрані для проекту (години, дні). Ліворуч на вертикалі розташовують найменування всіх робіт. На полі, що утворилось, поставляють у вигляді прямокутників роботи, довжина яких по горизонталі відповідає їхній тривалості. Між роботами лініями вказують логічні зв'язки [25]. Розроблена діаграма Ганта наведена на рис. А.4.

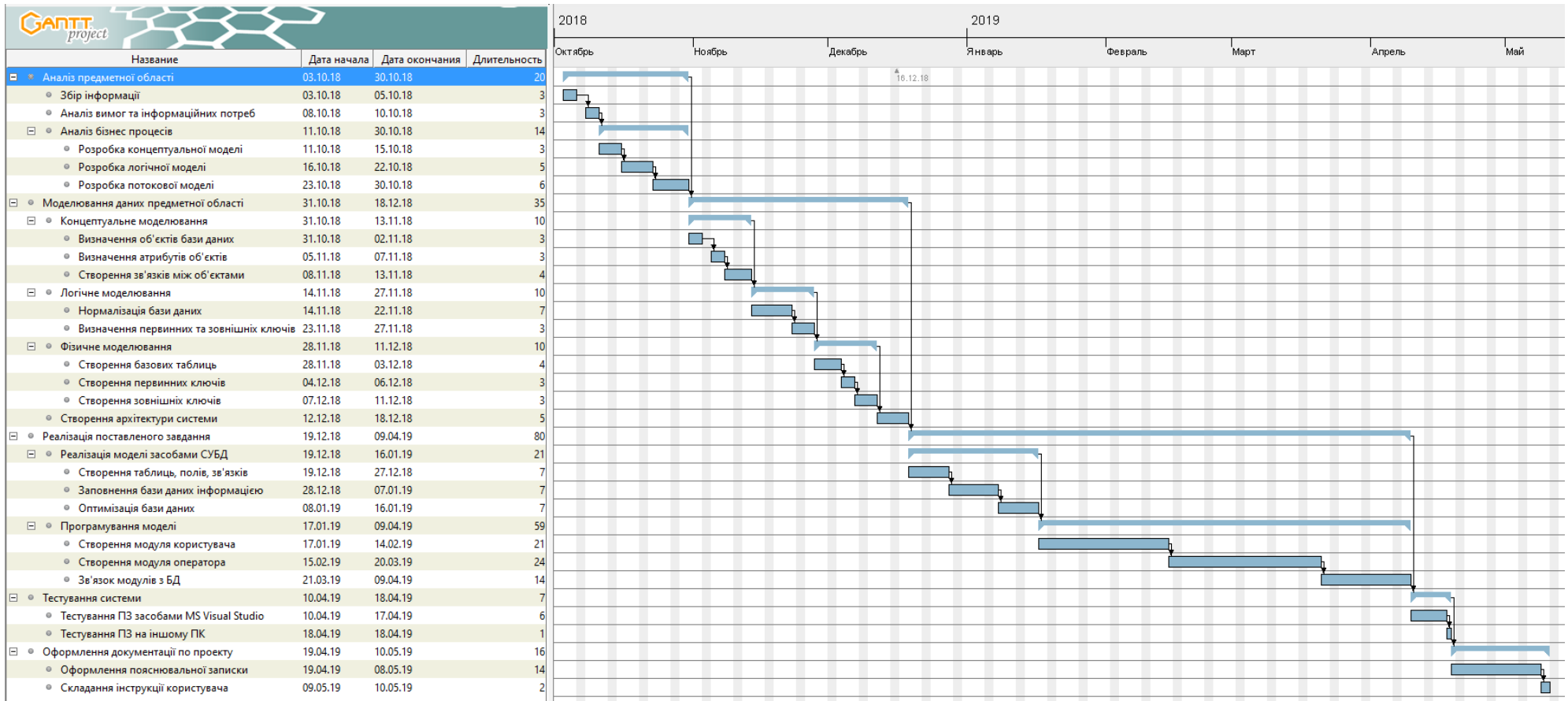


Рисунок А.4 – Список завдань ІТ-проекту та діаграма Ганта

Дослідження ризиків

При здійсненні процесу управління ризиками проекту необхідно дотримуватися певної послідовності дій. Розглянемо більш детально етапи процесу управління ризиками при прийнятті рішення на прикладі інвестування проекту будівництва офісного центру.

Етап I. Необхідно провести фіксацію ризиків, тобто обмежити кількість існуючих ризиків за принципом «розумної достатності». Для цього використовується анкетування та інтерв'ювання фахівців, а також досвід впровадження аналогічних проектів. У нашому випадку ми розглядаємо ризик з точки зору управлінської діяльності: наявність невизначеності, необхідність вибору альтернативи і можливість якісної і кількісної оцінки ймовірності здійснення того чи іншого варіанту. Елементами, наявність яких визначає ситуацію ризику, є: можливість відхилення від передбаченої мети, невпевненість в досягненні бажаного результату, ймовірність настання негативних подій при реалізації тих чи інших дій в умовах невизначеності, матеріальні або інші витрати, очікування небезпеки при реалізації обраної альтернативи.

Ефективним інструментом, який допоможе визначити всі можливі ризики, може стати карта процесів. У ній визначаються завдання проекту, цілі та процеси (заходи), які необхідно здійснити для їх досягнення. Побудова такої карти дозволяє охопити процеси в їх взаємозв'язку, а потім визначити найбільш істотні ризики. Далі за допомогою якісного аналізу об'єктивних і суб'єктивних факторів, що впливають на збільшення ступеня ризику, визначають процеси, які найбільшою мірою схильні до ризиків і перешкоджають досягненню цілей [19].

Етап II. На цьому етапі проводиться кількісна оцінка виявлених ризиків, яка може бути виражена відносним або абсолютним рівнем витрат і вимірюється ймовірністю виникнення ризику і ступенем впливу ризику при його виникненні. Для визначення цих показників використовують такі градації: високий, середній і низький. Однак на практиці важливо визначити ступінь впливу кожного ризику в кількісному вираженні, для чого рекомендуємо використовувати шкалу від 1 до 5 (рис. А.5) [26].

**Процесс оценки
идентифицированных
рисков в ходе
реализации проекта** 2



Рисунок А.5 – Процес оцінки ризиків у ході реалізації проекту

Етап III. Для визначення ступеня впливу розраховують величину втрат у відсотках від планової величини прибутку. Потім, множачи значення ймовірності виникнення і ступеня впливу, отримуємо індекс ризику – показник величини ймовірних втрат в балах, який визначається за допомогою матриці «Імовірність – Втрати» і дає можливість судити про ступінь впливу і рівні ризику [26]. Дані визначення ступеню впливу ризиків проекту зображені на рис. А.6.

| | Назва | Ймовірність | Втрати | Ступінь впливу | Характеристика |
|----|--|-------------|--------|----------------|----------------|
| 1 | DDos-атака/викрадення особистих даних | 2 | 5 | 10 | Помірний |
| 2 | Втручання розробників з боку замовника в код та архітектуру проекту без попередження | 2 | 4 | 8 | Незначний |
| 3 | Можливість тимчасової складності роботи користувачів з базою даних у разі відсутності доступу до інтернету | 2 | 2 | 4 | Ігнорований |
| 4 | Втрата комунікації між користувачами та розробниками | 2 | 2 | 4 | Ігнорований |
| 5 | Поява конкурентного програмного продукту з більшою функціональністю | 2 | 5 | 10 | Помірний |
| 6 | Некоректне технічне завдання | 4 | 2 | 8 | Незначний |
| 7 | Хвороба розробника | 1 | 1 | 1 | Ігнорований |
| 8 | Зміна вимог замовником на стадії завершення | 5 | 4 | 20 | Критичний |
| 9 | Несумісність програмного забезпечення з технічним забезпеченням замовника | 3 | 5 | 15 | Суттєвий |
| 10 | Втрата пріоритету виконання проекту у розробників. | 2 | 3 | 6 | Незначний |
| 11 | Некоректний графік розробки проекту | 4 | 2 | 8 | Незначний |
| 12 | Помилки в архітектурі через те, що компоненти ПЗ працюють не так, як повинні. | 4 | 5 | 20 | Критичний |
| | MAX | 5 | 5 | 25 | |

Рисунок А.6 – Визначення ступеню впливу ризиків проекту

Етап IV. Запропонуємо варіанти запобігання помірних, суттєвих та критичних ризиків та варіанти мінімізації наслідків їх виникнення (табл. А.2) [20].

Таблиця А.2 – Варіанти запобігання та мінімізація наслідків ризиків

| Назва | Варіант запобігання | Мінімізація наслідків |
|--|--|--|
| DDos-атака/викрадення особистих даних | Впровадження системи безпеки високого рівня спеціалістами з підтримки роботи програмного та апаратного забезпечення інформаційної системи, заздалегідь залученими до проекту | Блокування IP-адреси жертви сервіс-провайдером для мінімізації втрати залишкової інформації |
| Поява конкурентного програмного продукту з більшою функціональністю | Найм висококваліфікованих IT-спеціалістів для проекту, підвищення кваліфікації поточних робітників проекту, використання останніх технологій. | Реалізація додаткового функціоналу отримання основних системних та користувальницьких даних кожної заявки для системи HelpDesk виділяє проект серед конкурентних локальних проектів, що призведе до збереження бази клієнтів |
| Зміна вимог замовником на стадії завершення | Залучення замовника до роботи над проектом, щотижневий(щоденний) звіт про виконану роботу | Сповістити замовника, що відповідальність за будь-які затримки, спричинені змінами вимог близько до дедлайну повністю лежить на ньому |
| Несумісність програмного забезпечення з технічним забезпеченням замовника | Якомога раніше сформулювати вимоги до технічного забезпечення замовнику, потрібні для коректної роботи майбутнього програмного забезпечення | Надати замовнику потрібні пакети драйверів, файли конфігурації, додаткові модулі для виправлення несумісності |
| Помилки в архітектурі через те, що компоненти ПЗ працюють не так, як повинні | Завершити продукт проекту до робочої версії якомога раніше, для того, щоб більше часу залишилось на тестування та усунення несправностей | Реалізація в першу чергу тих компонентів, які відповідають протоколам спілкування між модулями і інтеграцією з зовнішніми компонентами |

ДОДАТОК Б ДАМП БАЗИ ДАНИХ

```
-- phpMyAdmin SQL Dump
-- version 4.8.3
-- https://www.phpmyadmin.net/
--
-- Хост: 127.0.0.1:3306
-- Время создания: Май 12 2019 г., 12:14
-- Версия сервера: 5.6.41
-- Версия PHP: 5.5.38

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База данных: `systemdesksdb`
--
-----

--
-- Структура таблицы `author`
--
CREATE TABLE `author` (
  `author_id` int(11) NOT NULL,
  `computername` char(30) NOT NULL,
  `name` char(30) NOT NULL,
  `surname` char(30) NOT NULL,
  `thirdname` char(30) NOT NULL,
  `organization` char(30) NOT NULL,
  `department` char(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `author`
--
INSERT INTO `author` (`author_id`, `computername`, `name`, `surname`, `thirdname`, `organization`, `department`) VALUES
(1, 'DESKTOP-FFPUPON', 'Богдан', 'Бабак', 'Александрович', 'Apptimized Operations', 'DevOps Engineers');
```

```

--
-- Структура таблицы `issue`
--

CREATE TABLE `issue` (
  `issue_id` int(11) NOT NULL,
  `headline` char(30) NOT NULL,
  `description` text NOT NULL,
  `screenshot` mediumblob,
  `creation_date` datetime NOT NULL,
  `end_date` datetime NOT NULL,
  `status` enum('Открыто','Закрото') NOT NULL DEFAULT 'Открыто',
  `author_id` int(11) NOT NULL,
  `type_id` int(11) NOT NULL,
  `cpu` char(10) NOT NULL,
  `ram` char(10) NOT NULL,
  `devmgmt` mediumblob NOT NULL,
  `minidump` mediumblob NOT NULL,
  `windowsupd` mediumblob NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `issue`
--

INSERT INTO `issue` (`issue_id`, `headline`, `description`, `screenshot`, `creation_date`, `end_date`, `status`, `author_id`,
`type_id`, `cpu`, `ram`, `devmgmt`, `minidump`, `windowsupd`) VALUES
(2, 'Завис ворд', 'Здравствуйте. При работе с документом перестал работать ворд. Пишет, что он не отвечает.
Помогите, пожалуйста, это очень важный и срочный документ.', 'BLOB' );
-----

--
-- Структура таблицы `processes`
--

CREATE TABLE `processes` (
  `process_id` int(11) NOT NULL,
  `issue_id` int(11) NOT NULL,
  `process_name` char(30) NOT NULL,
  `PID` char(5) NOT NULL,
  `status` char(15) NOT NULL,
  `username` char(30) NOT NULL,
  `memory` char(15) NOT NULL,
  `description` char(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `processes`
--

INSERT INTO `processes` (`process_id`, `issue_id`, `process_name`, `PID`, `status`, `username`, `memory`, `description`)
VALUES
(1, 2, 'chrome', '6460', 'Responding', 'breekachu', '237 MB', 'Google Chrome'),
(2, 2, 'chrome', '9044', 'Responding', 'breekachu', '336 MB', 'Google Chrome'),
(3, 2, 'SearchUI', '7748', 'Responding', 'breekachu', '69 MB', 'Search and Cortana application'),
(4, 2, 'chrome', '6452', 'Responding', 'breekachu', '19 MB', 'Google Chrome'),
(5, 2, 'wmpnetwk', '8604', 'Responding', 'Unknown', '6 MB', ''),
(6, 2, 'ServiceHub.DataWarehouseHost', '7740', 'Responding', 'breekachu', '36 MB', 'ServiceHub.Host.CLR.x64'),

```

(7, 2, 'explorer', '412', 'Responding', 'breekachu', '52 MB', 'Проводник'),
(8, 2, 'chrome', '6444', 'Responding', 'breekachu', '53 MB', 'Google Chrome'),
(9, 2, 'chrome', '6872', 'Responding', 'breekachu', '11 MB', 'Google Chrome'),
(10, 2, 'devenv', '4284', 'Responding', 'breekachu', '215 MB', 'Microsoft Visual Studio 2017'),
(11, 2, 'chrome', '6436', 'Responding', 'breekachu', '177 MB', 'Google Chrome'),
(12, 2, 'fontdrvhost', '10744', 'Responding', 'Unknown', '796 KB', ''),
(13, 2, 'chrome', '5140', 'Responding', 'breekachu', '31 MB', 'Google Chrome'),
(14, 2, 'chrome', '11172', 'Responding', 'breekachu', '300 MB', 'Google Chrome'),
(15, 2, 'conhost', '11600', 'Responding', 'breekachu', '4 MB', ''),
(16, 2, 'smss', '392', 'Responding', 'Unknown', '368 KB', ''),
(17, 2, 'chrome', '2544', 'Responding', 'breekachu', '29 MB', 'Google Chrome'),
(18, 2, 'ApplicationFrameHost', '10732', 'Responding', 'breekachu', '6 MB', ''),
(19, 2, 'NvTelemetryContainer', '2972', 'Responding', 'Unknown', '4 MB', ''),
(20, 2, 'svchost', '816', 'Responding', 'Unknown', '14 MB', ''),
(21, 2, 'SearchFilterHost', '7280', 'Responding', 'Unknown', '1 MB', ''),
(22, 2, 'chrome', '6412', 'Responding', 'breekachu', '83 MB', 'Google Chrome'),
(23, 2, 'nvcontainer', '2100', 'Responding', 'Unknown', '7 MB', ''),
(24, 2, 'svchost', '1236', 'Responding', 'Unknown', '20 MB', ''),
(25, 2, 'atiesrxx', '1664', 'Responding', 'Unknown', '1 MB', ''),
(26, 2, 'svchost', '2956', 'Responding', 'Unknown', '1 MB', ''),
(27, 2, 'chrome', '7264', 'Responding', 'breekachu', '221 MB', 'Google Chrome'),
(28, 2, 'VBCSCompiler', '7436', 'Responding', 'breekachu', '97 MB', ''),
(29, 2, 'PerfWatson2', '4244', 'Responding', 'breekachu', '38 MB', 'PerfWatson2.exe'),
(30, 2, 'spssvc', '1656', 'Responding', 'Unknown', '5 MB', ''),
(31, 2, 'WUDFHost', '1224', 'Responding', 'Unknown', '1 MB', ''),
(32, 2, 'chrome', '11136', 'Responding', 'breekachu', '31 MB', 'Google Chrome'),
(33, 2, 'chrome', '2512', 'Responding', 'breekachu', '84 MB', 'Google Chrome'),
(34, 2, 'chrome', '5096', 'Responding', 'breekachu', '60 MB', 'Google Chrome'),
(35, 2, 'svchost', '2940', 'Responding', 'Unknown', '98 MB', ''),
(36, 2, 'svchost', '784', 'Responding', 'Unknown', '11 MB', ''),
(37, 2, 'spoolsv', '2504', 'Responding', 'Unknown', '5 MB', ''),
(38, 2, 'WmiPrvSE', '3364', 'Responding', 'Unknown', '4 MB', ''),
(39, 2, 'ServiceHub.RoslynCodeAnalysisS', '8532', 'Responding', 'breekachu', '95 MB', 'ServiceHub.Host.CLR.x86'),
(40, 2, 'Open Server x64', '11548', 'Responding', 'breekachu', '10 MB', ''),
(41, 2, 'chrome', '8960', 'Responding', 'breekachu', '36 MB', 'Google Chrome'),
(42, 2, 'WmiApSrv', '2920', 'Responding', 'Unknown', '1 MB', ''),
(43, 2, 'HelpPane', '11108', 'Responding', 'breekachu', '3 MB', 'Справка и поддержка Microsoft'),
(44, 2, 'Discord', '5848', 'Responding', 'breekachu', '165 MB', 'Discord'),
(45, 2, 'chrome', '8948', 'Responding', 'breekachu', '52 MB', 'Google Chrome'),
(46, 2, 'amdow', '4552', 'Responding', 'breekachu', '2 MB', ''),
(47, 2, 'chrome', '1184', 'Responding', 'breekachu', '76 MB', 'Google Chrome'),
(48, 2, 'lsass', '752', 'Responding', 'Unknown', '6 MB', ''),
(49, 2, 'NVIDIA Web Helper', '8076', 'Responding', 'breekachu', '28 MB', 'NVIDIA Web Helper Service'),
(50, 2, 'LogWatNT', '6976', 'Responding', 'Unknown', '1016 KB', ''),
(51, 2, 'services', '744', 'Responding', 'Unknown', '3 MB', ''),
(52, 2, 'ServiceHub.Host.CLR.x86', '6344', 'Responding', 'breekachu', '52 MB', 'ServiceHub.Host.CLR.x86'),
(53, 2, 'chrome', '5476', 'Responding', 'breekachu', '347 MB', 'Google Chrome'),
(54, 2, 'ScriptedSandbox64', '14092', 'Responding', 'breekachu', '121 MB', 'ScriptedSandbox64.exe'),
(55, 2, 'Discord', '1040', 'Responding', 'breekachu', '24 MB', 'Discord'),
(56, 2, 'schtasks', '296', 'Responding', 'breekachu', '996 KB', ''),
(57, 2, 'Discord', '3312', 'Responding', 'breekachu', '7 MB', 'Discord'),
(58, 2, 'BPwin41', '7188', 'Responding', 'breekachu', '64 MB', 'BPwin Business Process Design'),
(59, 2, 'chrome', '7392', 'Responding', 'breekachu', '40 MB', 'Google Chrome'),
(60, 2, 'svchost', '2004', 'Responding', 'Unknown', '5 MB', ''),
(61, 2, 'chrome', '4588', 'Responding', 'breekachu', '66 MB', 'Google Chrome'),
(62, 2, 'sqlceip', '3292', 'Responding', 'Unknown', '29 MB', ''),
(63, 2, 'svchost', '2428', 'Responding', 'Unknown', '2 MB', ''),
(64, 2, 'conhost', '3780', 'Responding', 'breekachu', '4 MB', ''),

(65, 2, 'chrome', '10184', 'Responding', 'breekachu', '173 MB', 'Google Chrome'),
(66, 2, 'conhost', '6304', 'Responding', 'breekachu', '4 MB', ''),
(67, 2, 'sihost', '3468', 'Responding', 'breekachu', '5 MB', ''),
(68, 2, 'httpd', '7164', 'Responding', 'breekachu', '22 MB', 'Apache HTTP Server'),
(69, 2, 'chrome', '9748', 'Responding', 'breekachu', '52 MB', 'Google Chrome'),
(70, 2, 'conhost', '6296', 'Responding', 'breekachu', '4 MB', ''),
(71, 2, 'NVDisplay.Container', '1544', 'Responding', 'Unknown', '8 MB', ''),
(72, 2, 'conhost', '9300', 'Responding', 'breekachu', '5 MB', ''),
(73, 2, 'chrome', '11020', 'Responding', 'breekachu', '23 MB', 'Google Chrome'),
(74, 2, 'svchost', '2828', 'Responding', 'Unknown', '5 MB', ''),
(75, 2, 'chrome', '7136', 'Responding', 'breekachu', '29 MB', 'Google Chrome'),
(76, 2, 'RuntimeBroker', '4116', 'Responding', 'breekachu', '13 MB', ''),
(77, 2, 'chrome', '3684', 'Responding', 'breekachu', '63 MB', 'Google Chrome'),
(78, 2, 'chrome', '10148', 'Responding', 'breekachu', '36 MB', 'Google Chrome'),
(79, 2, 'Locator', '2820', 'Responding', 'Unknown', '480 KB', ''),
(80, 2, 'wininit', '664', 'Responding', 'Unknown', '1012 KB', ''),
(81, 2, 'svchost', '1956', 'Responding', 'Unknown', '4 MB', ''),
(82, 2, 'dllhost', '3248', 'Responding', 'breekachu', '5 MB', 'COM Surrogate'),
(83, 2, 'dllhost', '4540', 'Responding', 'breekachu', '2 MB', 'COM Surrogate'),
(84, 2, 'ServiceHub.SettingsHost', '5832', 'Responding', 'breekachu', '50 MB', 'ServiceHub.Host.CLR.x86'),
(85, 2, 'chrome', '10136', 'Responding', 'breekachu', '30 MB', 'Google Chrome'),
(86, 2, 'ServiceHub.IdentityHost', '1368', 'Responding', 'breekachu', '22 MB', 'ServiceHub.Host.CLR.x86'),
(87, 2, 'chrome', '1072', 'Responding', 'breekachu', '153 MB', 'Google Chrome'),
(88, 2, 'NVDisplay.Container', '1932', 'Responding', 'Unknown', '4 MB', ''),
(89, 2, 'OriginWebHelperService', '2784', 'Responding', 'Unknown', '5 MB', ''),
(90, 2, 'svchost', '2352', 'Responding', 'Unknown', '2 MB', ''),
(91, 2, 'chrome', '10828', 'Responding', 'breekachu', '69 MB', 'Google Chrome'),
(92, 2, 'MsMpEng', '2776', 'Responding', 'Unknown', '193 MB', ''),
(93, 2, 'chrome', '9660', 'Responding', 'breekachu', '41 MB', 'Google Chrome'),
(94, 2, 'chrome', '10088', 'Responding', 'breekachu', '156 MB', 'Google Chrome'),
(95, 2, 'svchost', '2760', 'Responding', 'Unknown', '7 MB', ''),
(96, 2, 'SearchIndexer', '4912', 'Responding', 'Unknown', '44 MB', ''),
(97, 2, 'SystemDesk', '12668', 'Responding', 'breekachu', '165 MB', 'SystemDesk'),
(98, 2, 'NisSrv', '6196', 'Responding', 'Unknown', '8 MB', ''),
(99, 2, 'MSASCuiL', '5764', 'Responding', 'breekachu', '4 MB', 'Windows Defender notification'),
(100, 2, 'PerfWatson2', '3608', 'Responding', 'breekachu', '34 MB', 'PerfWatson2.exe'),
(101, 2, 'AMDRSServ', '4468', 'Responding', 'breekachu', '115 MB', ''),
(102, 2, 'devenv', '4192', 'Responding', 'breekachu', '286 MB', 'Microsoft Visual Studio 2017'),
(103, 2, 'WmiPrvSE', '12164', 'Responding', 'Unknown', '2 MB', ''),
(104, 2, 'chrome', '4028', 'Responding', 'breekachu', '88 MB', 'Google Chrome'),
(105, 2, 'SystemSettings', '9124', 'Not responding', 'breekachu', '13 MB', 'Параметры'),
(106, 2, 'ShellExperienceHost', '12400', 'Not responding', 'breekachu', '22 MB', 'Windows Shell Experience Host'),
(107, 2, 'taskhostw', '4976', 'Responding', 'breekachu', '3 MB', ''),
(108, 2, 'jucheck', '5836', 'Responding', 'breekachu', '3 MB', 'Java Update Checker'),
(109, 2, 'chrome', '1000', 'Responding', 'breekachu', '163 MB', 'Google Chrome'),
(110, 2, 'MSBuild', '9184', 'Responding', 'breekachu', '26 MB', 'MSBuild.exe'),
(111, 2, 'RadeonSettings', '7028', 'Responding', 'breekachu', '206 MB', 'Radeon Settings: Host Applicat'),
(112, 2, 'conhost', '13060', 'Responding', 'breekachu', '5 MB', ''),
(113, 2, 'ServiceHub.Host.Node.x86', '7456', 'Responding', 'breekachu', '24 MB', 'Node.js: Server-side JavaScript'),
(114, 2, 'chrome', '556', 'Responding', 'breekachu', '2 MB', 'Google Chrome'),
(115, 2, 'sqlwriter', '2948', 'Responding', 'Unknown', '1 MB', ''),
(116, 2, 'Memory Compression', '3136', 'Responding', 'Unknown', '2 MB', ''),
(117, 2, 'svchost', '980', 'Responding', 'Unknown', '5 MB', ''),
(118, 2, 'chrome', '6148', 'Responding', 'breekachu', '30 MB', 'Google Chrome'),
(119, 2, 'chrome', '7560', 'Responding', 'breekachu', '405 MB', 'Google Chrome'),
(120, 2, 'chrome', '5712', 'Responding', 'breekachu', '1 MB', 'Google Chrome'),
(121, 2, 'chrome', '3556', 'Responding', 'breekachu', '29 MB', 'Google Chrome'),
(122, 2, 'dwm', '536', 'Responding', 'Unknown', '77 MB', ''),

```
(123, 2, 'svchost', '532', 'Responding', 'Unknown', '23 MB', ''),
(124, 2, 'POWERPNT', '10872', 'Responding', 'breekachu', '99 MB', 'Microsoft PowerPoint'),
(125, 2, 'atieclxx', '1820', 'Responding', 'Unknown', '2 MB', ''),
(126, 2, 'StandardCollector.Service', '11732', 'Responding', 'Unknown', '37 MB', ''),
(127, 2, 'svchost', '6128', 'Responding', 'breekachu', '6 MB', 'Хост-процесс для служб Windows'),
(128, 2, 'httpd', '6556', 'Responding', 'breekachu', '10 MB', 'Apache HTTP Server'),
(129, 2, 'ServiceHub.VSDetouredHost', '6980', 'Responding', 'breekachu', '97 MB', 'ServiceHub.Host.CLR.x86'),
(130, 2, 'chrome', '13444', 'Responding', 'breekachu', '13 MB', 'Google Chrome'),
(131, 2, 'Discord', '1804', 'Responding', 'breekachu', '64 MB', 'Discord'),
(132, 2, 'chrome', '6544', 'Responding', 'breekachu', '55 MB', 'Google Chrome'),
(133, 2, 'SppExtComObj', '7404', 'Responding', 'Unknown', '1 MB', ''),
(134, 2, 'jusched', '5248', 'Responding', 'breekachu', '1 MB', 'Java Update Scheduler'),
(135, 2, 'MpCmdRun', '6540', 'Responding', 'Unknown', '3 MB', ''),
(136, 2, 'svchost', '504', 'Responding', 'Unknown', '14 MB', ''),
(137, 2, 'ServiceHub.RoslynCodeAnalysisS', '12136', 'Responding', 'breekachu', '98 MB', 'ServiceHub.Host.CLR.x86'),
(138, 2, 'chrome', '10840', 'Responding', 'breekachu', '64 MB', 'Google Chrome'),
(139, 2, 'chrome', '5928', 'Responding', 'breekachu', '356 MB', 'Google Chrome'),
(140, 2, 'Discord', '6096', 'Responding', 'breekachu', '13 MB', 'Discord'),
(141, 2, 'conhost', '11696', 'Responding', 'breekachu', '5 MB', ''),
(142, 2, 'dasHost', '2208', 'Responding', 'Unknown', '4 MB', ''),
(143, 2, 'svchost', '908', 'Responding', 'Unknown', '7 MB', ''),
(144, 2, 'nvcontainer', '6076', 'Responding', 'breekachu', '24 MB', ''),
(145, 2, 'RtkNGUI64', '3916', 'Responding', 'breekachu', '4 MB', 'Диспетчер Realtek HD'),
(146, 2, 'SnippingTool', '11716', 'Responding', 'breekachu', '3 MB', ''),
(147, 2, 'nvcontainer', '6068', 'Responding', 'breekachu', '4 MB', ''),
(148, 2, 'conhost', '2188', 'Responding', 'breekachu', '4 MB', ''),
(149, 2, 'chrome', '10368', 'Responding', 'breekachu', '62 MB', 'Google Chrome'),
(150, 2, 'SearchProtocolHost', '2028', 'Responding', 'Unknown', '2 MB', ''),
(151, 2, 'WINWORD', '10364', 'Not responding', 'breekachu', '78 MB', 'Microsoft Word'),
(152, 2, 'csrss', '544', 'Responding', 'Unknown', '1 MB', ''),
(153, 2, 'mysqld', '11652', 'Responding', 'breekachu', '229 MB', ''),
(154, 2, 'chrome', '2600', 'Responding', 'breekachu', '87 MB', 'Google Chrome'),
(155, 2, 'winlogon', '872', 'Responding', 'Unknown', '2 MB', ''),
(156, 2, 'csrss', '680', 'Responding', 'Unknown', '9 MB', ''),
(157, 2, 'chrome', '8', 'Responding', 'breekachu', '92 MB', 'Google Chrome'),
(158, 2, 'System', '4', 'Responding', 'Unknown', '156 KB', ''),
(159, 2, 'svchost', '1328', 'Responding', 'Unknown', '11 MB', ''),
(160, 2, 'Discord', '3880', 'Responding', 'breekachu', '36 MB', 'Discord'),
(161, 2, 'Idle', '0', 'Responding', 'Unknown', '0 B', ''),
(501, 4, 'Idle', '0', 'Responding', 'Unknown', '0 B', '');
```

```
-----
```

```
--
```

```
-- Структура таблицы `type`
```

```
--
```

```
CREATE TABLE `type` (
  `type_id` int(11) NOT NULL,
  `type_description` char(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
```

```
-- Дамп данных таблицы `type`
```

```
--
```

```
INSERT INTO `type` (`type_id`, `type_description`) VALUES
(1, 'Обычный');
```

```

(2, 'Срочный');

--
-- Индексы сохранённых таблиц
--

--
-- Индексы таблицы `author`
--
ALTER TABLE `author`
  ADD PRIMARY KEY (`author_id`),
  ADD UNIQUE KEY `computername` (`computername`);

--
-- Индексы таблицы `issue`
--
ALTER TABLE `issue`
  ADD PRIMARY KEY (`issue_id`),
  ADD KEY `author_id` (`author_id`),
  ADD KEY `type_id` (`type_id`);

--
-- Индексы таблицы `processes`
--
ALTER TABLE `processes`
  ADD PRIMARY KEY (`process_id`),
  ADD KEY `issue_id` (`issue_id`);

--
-- Индексы таблицы `type`
--
ALTER TABLE `type`
  ADD PRIMARY KEY (`type_id`);

--
-- AUTO_INCREMENT для сохранённых таблиц
--

--
-- AUTO_INCREMENT для таблицы `author`
--
ALTER TABLE `author`
  MODIFY `author_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT для таблицы `issue`
--
ALTER TABLE `issue`
  MODIFY `issue_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT для таблицы `processes`
--
ALTER TABLE `processes`
  MODIFY `process_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=502;

--
-- AUTO_INCREMENT для таблицы `type`

```

```

--
ALTER TABLE `type`
  MODIFY `type_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- Ограничения внешнего ключа сохраненных таблиц
--

--
-- Ограничения внешнего ключа таблицы `issue`
--
ALTER TABLE `issue`
  ADD CONSTRAINT `c_type_id` FOREIGN KEY (`type_id`) REFERENCES `type` (`type_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `issue_ibfk_2` FOREIGN KEY (`type_id`) REFERENCES `type` (`type_id`),
  ADD CONSTRAINT `issue_ibfk_3` FOREIGN KEY (`author_id`) REFERENCES `author` (`author_id`);

--
-- Ограничения внешнего ключа таблицы `processes`
--
ALTER TABLE `processes`
  ADD CONSTRAINT `processes_ibfk_1` FOREIGN KEY (`issue_id`) REFERENCES `issue` (`issue_id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```


ДОДАТОК В

КОД ПРОГРАМНОГО ПРОДУКТУ

SystemDesk

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Diagnostics;
using System.Management;
using System.Dynamic;
using System.Management.Automation;
using System.Management.Automation.Runspaces;
using System.Runtime.InteropServices;
using MetroFramework;
using MetroFramework.Forms;
using MetroFramework.Drawing;

namespace SystemDesk
{
    public partial class MainForm : MetroForm
    {
        private static Image fullDesktop;
        private static Image partImage;
        private static Form backgroundForm;
        private static PictureBox pictureBox1;

        private static Point mouseDownAt;
        private static Point mouseIsAt;
        private static bool isMouseDown = false;

        private static Color backgroundColor = Color.DarkGray;
        private static int backgroundAlpha = 150;
        private static Color outlineColor = Color.Red;
        private static int outlineWidth = 1;

        MySqlConnection conn;
        string connStr = "server=localhost;database=systemdeskd;user=root;password=";

        [DllImportAttribute("User32.dll")]
        private static extern int FindWindow(String ClassName, String WindowName);

        [DllImport("user32.dll")]
```

```

static extern bool SetForegroundWindow(IntPtr hWnd);

[DllImport("user32.dll")]
static extern bool GetWindowRect(IntPtr hWnd, out RECT lpRect);

[DllImport("user32.dll")]
private static extern bool PrintWindow(IntPtr hwnd, IntPtr hdcBlt, uint nFlags);

[DllImport("user32.dll")]
private static extern IntPtr GetForegroundWindow();

[StructLayout(LayoutKind.Sequential)]
public struct RECT
{
    public int Left;    // x position of upper-left corner
    public int Top;    // y position of upper-left corner
    public int Right;   // x position of lower-right corner
    public int Bottom;  // y position of lower-right corner
}

public MainForm()
{
    InitializeComponent();
    notifyIcon1.Visible = true;
    WindowState = FormWindowState.Minimized;
    this.notifyIcon1.MouseDoubleClick += new MouseEventHandler(notifyIcon1_MouseDoubleClick);
    comboBox1.SelectedIndex = 0;
    string root = Environment.GetEnvironmentVariable("appdata") + @"\SystemDesk";
    if (!Directory.Exists(root))
    {
        Directory.CreateDirectory(root);
    }

    try
    {
        conn = new MySqlConnection(connStr);
        conn.Open();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void notifyIcon1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    notifyIcon1.Visible = false;
    this.ShowInTaskbar = true;
    WindowState = FormWindowState.Normal;
}

private void button1_Click(object sender, EventArgs e)
{
    fullDesktop = CaptureScreen;
    Initialize();

    backgroundForm.Show();
}

```

```

private static void Initialize()
{
    backgroundForm = new Form();
    backgroundForm.FormBorderStyle = FormBorderStyle.None;
    backgroundForm.WindowState = FormWindowState.Maximized;
    backgroundForm.Cursor = Cursors.Cross;

    pictureBox1 = new PictureBox();
    pictureBox1.Size = Screen.FromControl(backgroundForm).Bounds.Size;
    pictureBox1.Location = new Point(0, 0);
    pictureBox1.Image = fullDesktop;
    pictureBox1.Paint += new PaintEventHandler(OnPaint);
    pictureBox1.MouseDown += new MouseEventHandler(OnMouseDown);
    pictureBox1.MouseMove += new MouseEventHandler(OnMouseMove);
    pictureBox1.MouseUp += new MouseEventHandler(OnMouseUp);
    backgroundForm.Controls.Add(pictureBox1);
}

private static void OnPaint(object sender, PaintEventArgs e)
{
    SolidBrush opaqueWhiteBrush = new SolidBrush(Color.FromArgb(backgroundAlpha, backgroundColor.R,
backgroundAlpha, backgroundColor.G, backgroundAlpha, backgroundColor.B));
    e.Graphics.FillRectangle(opaqueWhiteBrush, 0, 0, pictureBox1.Width, pictureBox1.Height);

    if (isMouseDown)
    {
        Rectangle pos = getMouseMoveRect;
        e.Graphics.DrawRectangle(new Pen(outlineColor, outlineWidth), new Rectangle(pos.X - outlineWidth, pos.Y -
outlineWidth, pos.Width + outlineWidth, pos.Height + outlineWidth));
        e.Graphics.DrawImage(partImage, pos.Location);

        string displayText = "W: " + pos.Width + " H: " + pos.Height;
        Font font = new Font("Arial", 10, FontStyle.Bold, GraphicsUnit.Pixel);
        e.Graphics.DrawString(displayText, font, Brushes.White, pos.X, pos.Y - font.Size - 6, StringFormat.GenericDefault);
    }
}

private static void OnMouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Right && !isMouseDown)
    {
        Application.Exit();
    }
    else if (e.Button == MouseButtons.Left)
    {
        mouseDownAt = e.Location;
        Cursor.Position = new Point(e.Location.X + 1, e.Location.Y + 1);
        mouseIsAt = new Point(e.Location.X + 1, e.Location.Y + 1);
        partImage = ((Bitmap)fullDesktop).Clone(getMouseMoveRect, fullDesktop.PixelFormat);
        isMouseDown = true;
        pictureBox1.Refresh();
    }
}

private static void OnMouseMove(object sender, MouseEventArgs e)
{
    if (isMouseDown)

```

```

    {
        mouselsAt = e.Location;
        Rectangle rect = getMouseMoveRect;
        if (rect.Width != 0 && rect.Height != 0)
        {
            partImage = ((Bitmap)fullDesktop).Clone(rect, fullDesktop.PixelFormat);
            pictureBox1.Refresh();
        }
    }
}

```

```
private static void OnMouseUp(object sender, MouseEventArgs e)
```

```

{
    if (e.Button == MouseButtons.Left)
    {
        bool hasMoved = false;

        if (isMouseDown)
        {
            hasMoved = true;
            isMouseDown = false;
        }

        /*SaveFileDialog sfd = new SaveFileDialog();
        sfd.Filter = "PNG image (*.png)|*.png";
        if (sfd.ShowDialog() == DialogResult.OK)
        {
            partImage.Save(sfd.FileName, ImageFormat.Png);
        }
        else
        {
            hasMoved = false;
            pictureBox1.Refresh();
        }*/

        if (hasMoved)
        {
            backgroundForm.Close();
        }
    }
}

```

```
private static Rectangle getMouseMoveRect
```

```

{
    get
    {
        int x = 0;
        int y = 0;
        int width = 0;
        int height = 0;

        if (mouselsAt.X > mouseDownAt.X)
        {
            x = mouseDownAt.X;
            width = mouselsAt.X - mouseDownAt.X;
        }
        else
        {

```

```

        x = mouselsAt.X;
        width = mouseDownAt.X - mouselsAt.X;
    }

    if (mouselsAt.Y > mouseDownAt.Y)
    {
        y = mouseDownAt.Y;
        height = mouselsAt.Y - mouseDownAt.Y;
    }
    else
    {
        y = mouselsAt.Y;
        height = mouseDownAt.Y - mouselsAt.Y;
    }

    return new Rectangle(x, y, width, height);
}
}

private static Image CaptureScreen
{
    get
    {
        Bitmap image = new Bitmap(Screen.PrimaryScreen.Bounds.Width, Screen.PrimaryScreen.Bounds.Height);
        using (Graphics g = Graphics.FromImage(image))
        {
            g.CopyFromScreen(Screen.PrimaryScreen.Bounds.X,
                Screen.PrimaryScreen.Bounds.Y,
                0, 0,
                image.Size,
                CopyPixelOperation.SourceCopy);
        }

        return image;
    }
}

private void Form1_Activated(object sender, EventArgs e)
{
    if (partImage != null)
    {
        pictureBox1.Image = partImage;
        pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    using (SqlConnection con = new SqlConnection(connStr))
    {
        try
        {
            DirectoryInfo dirInfo = new DirectoryInfo(Environment.GetEnvironmentVariable("appdata") + @"\SystemDesk\");

            foreach (FileInfo f in dirInfo.GetFiles())
            {
                f.Delete();
            }
        }
    }
}

```

```

string minidumpname = "";

//get minidump file
if (IsDirectoryEmpty(Environment.GetEnvironmentVariable("windir") + @"\Minidump") == false)
{
    var directory = new DirectoryInfo(Environment.GetEnvironmentVariable("windir") + @"\Minidump");
    var myFile = directory.GetFiles()
        .OrderByDescending(f => f.LastWriteTime)
        .First();
    File.Copy(directory.ToString() + @"\" + myFile.ToString(), Environment.GetEnvironmentVariable("appdata") +
@"\SystemDesk\" + Path.GetFileName(myFile.ToString()));
    minidumpname = Path.GetFileName(myFile.ToString());
}

//get windows update log
RunspaceConfiguration psConfig = RunspaceConfiguration.Create();
var psRunspace = RunspaceFactory.CreateRunspace(psConfig);
psRunspace.Open();
using (Pipeline psPipeline = psRunspace.CreatePipeline())
{
    Command command = new Command("Get-WindowsUpdateLog");
    command.Parameters.Add("LogPath", Environment.GetEnvironmentVariable("appdata") +
@"\SystemDesk\WindowsUpdate.log");
    psPipeline.Commands.Add(command);
    var results = psPipeline.Invoke();
}

//get devmgmt.msc screenshot
ProcessStartInfo startInfo = new ProcessStartInfo();
startInfo.FileName = "mmc.exe";
startInfo.Arguments = "devmgmt.msc";
Process.Start(startInfo);

System.Threading.Thread.Sleep(3000);

System.Diagnostics.Process[] p = System.Diagnostics.Process.GetProcessesByName("mmc");
if (p.Length > 0)
{
    SetForegroundWindow(p[0].MainWindowHandle);
}

var foregroundWindowsHandle = GetForegroundWindow();
var rect = new RECT();

GetWindowRect(foregroundWindowsHandle, out rect);

Rectangle bounds = new Rectangle(rect.Left, rect.Top, rect.Right - rect.Left, rect.Bottom - rect.Top);

Bitmap bmp = new Bitmap(bounds.Width, bounds.Height);

using (Graphics g = Graphics.FromImage(bmp))
{
    g.CopyFromScreen(new Point(bounds.Left, bounds.Top), Point.Empty, bounds.Size);
}

bmp.Save(Environment.GetEnvironmentVariable("appdata") + @"\SystemDesk\devmgmt.png",
ImageFormat.Png);

```

```

p[0].Kill();

//get cpu and ram usage
PerformanceCounter cpuCounter;
PerformanceCounter ramCounter;

cpuCounter = new PerformanceCounter("Processor", "% Processor Time", "_Total");
ramCounter = new PerformanceCounter("Memory", "Available MBytes");

string ram = ramCounter.NextValue() + "MB";
string cpu = cpuCounter.NextValue() + "%";
System.Threading.Thread.Sleep(1000);
cpu = cpuCounter.NextValue() + "%";

Image image = pictureBox1.Image;
MemoryStream memoryStream1 = new MemoryStream();
image.Save(memoryStream1, ImageFormat.Png);
byte[] imageBt = memoryStream1.ToArray();

Image devmgmt = Image.FromFile(Environment.GetEnvironmentVariable("appdata") +
@"\SystemDesk\devmgmt.png");
MemoryStream memoryStream2 = new MemoryStream();
devmgmt.Save(memoryStream2, ImageFormat.Png);
byte[] devmgmtBt = memoryStream2.ToArray();

FileStream fs = new FileStream(Environment.GetEnvironmentVariable("appdata") +
@"\SystemDesk\WindowsUpdate.log", FileMode.Open, FileAccess.Read);
long fileSize = fs.Length;
byte[] windowsupd = new byte[fileSize];
fs.Read(windowsupd, 0, Convert.ToInt32(fileSize));
fs.Close();

fs = new FileStream(Environment.GetEnvironmentVariable("appdata") + @"SystemDesk\" + minidumpname,
FileMode.Open, FileAccess.Read);
fileSize = fs.Length;
byte[] minidump = new byte[fileSize];
fs.Read(minidump, 0, Convert.ToInt32(fileSize));
fs.Close();

string computername = System.Environment.MachineName;
string headline = textBox1.Text;
string description = textBox2.Text;
DateTime creation_date = DateTime.Today;
int type_id = 0;
string author_id = "";
string issue_id = "";
if (comboBox1.SelectedIndex==0)
{
    type_id = 1;
}
else
{
    type_id = 2;
}

string sql = "SELECT author_id FROM author WHERE computername='" + computername + "'";
MySqlCommand oCmd = new MySqlCommand(sql, con);
con.Open();

```

```

using (MySqlDataReader oReader = oCmd.ExecuteReader())
{
    while (oReader.Read())
    {
        author_id = oReader["author_id"].ToString();
    }
    con.Close();
}

sql = "INSERT INTO issue (issue_id, headline, description, screenshot, creation_date, author_id, type_id, cpu, ram,
devmgmt, minidump, windowsupd)" + " VALUES (null, '" + headline + "', '" + description + "', '" + @img + "', '" + NOW()" + "',
'" + author_id + "', '" + type_id + "', '" + cpu + "', '" + ram + "', '" + @devmgmt + "', '" + @minidump + "', '" + @windowsupd"
+ "));

MySqlCommand cmd = new MySqlCommand(sql, con);
cmd.Parameters.AddWithValue("@img", imageBt);
cmd.Parameters.AddWithValue("@devmgmt", devmgmtBt);
cmd.Parameters.AddWithValue("@minidump", minidump);
cmd.Parameters.AddWithValue("@windowsupd", windowsupd);
con.Open();
cmd.ExecuteNonQuery();
con.Close();
Process[] processList = Process.GetProcesses();
ImageList Imagelist = new ImageList();
foreach (Process process in processList)
{
    string status = (process.Responding == true ? "Responding" : "Not responding");
    dynamic extraProcessInfo = GetProcessExtraInformation(process.Id);
    string[] row = {
        process.ProcessName,
        process.Id.ToString(),
        status,
        extraProcessInfo.Username,
        BytesToReadableValue(process.PrivateMemorySize64),
        extraProcessInfo.Description
    };

    sql = "SELECT issue_id FROM issue WHERE description='" + description + "'";
    oCmd = new MySqlCommand(sql, con);
    con.Open();
    using (MySqlDataReader oReader = oCmd.ExecuteReader())
    {
        while (oReader.Read())
        {
            issue_id = oReader["issue_id"].ToString();
        }
    }

    sql = "INSERT INTO processes (process_id, issue_id, process_name, PID, status, username, memory,
description)" + " VALUES (null, '" + issue_id + "', '" + process.ProcessName + "', '" + process.Id.ToString() + "', '" + status + "', '"
+ extraProcessInfo.Username + "', '" + BytesToReadableValue(process.PrivateMemorySize64) + "', '" +
extraProcessInfo.Description + "));";
    cmd = new MySqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@img", imageBt);
    cmd.ExecuteNonQuery();
    con.Close();
}

MessageBox.Show("Заявка создана!");

```



```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

public void renderProcessesOnListView()
{
    Process[] processList = Process.GetProcesses();
    ImageList Imagelist = new ImageList();
    foreach (Process process in processList)
    {
        string status = (process.Responding == true ? "Responding" : "Not responding");
        dynamic extraProcessInfo = GetProcessExtraInformation(process.Id);
        string[] row = {
            process.ProcessName,
            process.Id.ToString(),
            status,
            extraProcessInfo.Username,
            BytesToReadableValue(process.PrivateMemorySize64),
            extraProcessInfo.Description
        };
        try
        {
            Imagelist.Images.Add(
                process.Id.ToString(),
                Icon.ExtractAssociatedIcon(process.MainModule.FileName).ToBitmap()
            );
        }
        catch { }
        ListViewItem item = new ListViewItem(row)
        {
            ImageIndex = Imagelist.Images.IndexOfKey(process.Id.ToString())
        };
    }
}

public string BytesToReadableValue(long number)
{
    List<string> suffixes = new List<string> { " B", " KB", " MB", " GB", " TB", " PB" };

    for (int i = 0; i < suffixes.Count; i++)
    {
        long temp = number / (int)Math.Pow(1024, i + 1);

        if (temp == 0)
        {
            return (number / (int)Math.Pow(1024, i)) + suffixes[i];
        }
    }

    return number.ToString();
}

public ExpandoObject GetProcessExtraInformation(int processId)
{

```

```

string query = "Select * From Win32_Process Where ProcessID = " + processId;
ManagementObjectSearcher searcher = new ManagementObjectSearcher(query);
ManagementObjectCollection processList = searcher.Get();
dynamic response = new ExpandoObject();
response.Description = "";
response.Username = "Unknown";

foreach (ManagementObject obj in processList)
{
    string[] argList = new string[] { string.Empty, string.Empty };
    int returnVal = Convert.ToInt32(obj.InvokeMethod("GetOwner", argList));
    if (returnVal == 0)
    {
        response.Username = argList[0];
    }
    if (obj["ExecutablePath"] != null)
    {
        try
        {
            FileVersionInfo info = FileVersionInfo.GetVersionInfo(obj["ExecutablePath"].ToString());
            response.Description = info.FileDescription;
        }
        catch {}
    }
}

return response;
}

private void настройкиToolStripMenuItem_Click(object sender, EventArgs e)
{
    SettingsForm settingsForm = new SettingsForm();
    settingsForm.Show();
}

private void оПрограммеToolStripMenuItem_Click(object sender, EventArgs e)
{
    AboutForm aboutForm = new AboutForm();
    aboutForm.Show();
}

public bool IsDirectoryEmpty(string path)
{
    return !Directory.EnumerateFileSystemEntries(path).Any();
}
}
}

```

SystemDeskAdmin

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Drawing.Imaging;
using MetroFramework;
using MetroFramework.Forms;

namespace SystemDeskAdmin
{
    public partial class IssueForm : MetroForm
    {

        MySqlConnection conn;
        string connStr = "server=localhost;database=systemdeskd;user=root;password=";
        string issue_id = "";
        string author_id = "";

        Image devmgmt = null;
        Image screenshot = null;

        public IssueForm(string selectedissue)
        {
            InitializeComponent();

            issue_id = selectedissue;

            try
            {
                conn = new MySqlConnection(connStr);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void IssueForm_Load(object sender, EventArgs e)
        {
            try
            {
                dateTimePicker1.Format = DateTimePickerFormat.Custom;
                dateTimePicker1.CustomFormat = "dd.MM.yyyy HH:mm";

                dateTimePicker2.Format = DateTimePickerFormat.Custom;
                dateTimePicker2.CustomFormat = "dd.MM.yyyy HH:mm";

                string sql = "SELECT * FROM issue WHERE issue_id='" + issue_id + "'";
                MySqlCommand oCmd = new MySqlCommand(sql, conn);
```

```

conn.Open();
using (MySqlDataReader oReader = oCmd.ExecuteReader())
{
    while (oReader.Read())
    {
        textBox1.Text = oReader.GetString("headline");
        textBox2.Text = oReader.GetString("description");
        dateTimePicker1.Text = oReader.GetMySqlDateTime("creation_date").ToString();
        if (oReader.GetMySqlDateTime("end_date").ToString() != "00.00.0000 0:00:00")
        {
            dateTimePicker2.Text = oReader.GetMySqlDateTime("end_date").ToString();
            dateTimePicker2.Visible = true;
        }
        string type = oReader.GetString("type_id");
        if (type == "1")
        {
            textBox3.Text = "Обычный";
        }
        else
        {
            textBox3.Text = "Срочный";
        }

        textBox5.Text = "Загруженность: " + oReader.GetString("cpu");
        textBox6.Text = "Свободной памяти: " + oReader.GetString("ram");

        author_id = oReader.GetString("author_id");

        byte[] picbyte = oReader["screenshot"] as byte[] ?? null;
        if (picbyte != null)
        {
            MemoryStream mstream = new MemoryStream(picbyte);
            pictureBox1.Image = System.Drawing.Image.FromStream(mstream);
            screenshot = System.Drawing.Image.FromStream(mstream);
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
        }

        picbyte = oReader["devmgmt"] as byte[] ?? null;
        if (picbyte != null)
        {
            MemoryStream mstream = new MemoryStream(picbyte);
            devmgmt = System.Drawing.Image.FromStream(mstream);
        }

    }

}

sql = "SELECT * FROM author WHERE author_id='" + author_id + "'";
oCmd = new MySqlCommand(sql, conn);
using (MySqlDataReader oReader = oCmd.ExecuteReader())
{
    while (oReader.Read())
    {
        string name = oReader.GetString("name");
        string surname = oReader.GetString("surname");
    }
}

```

```

        string thirdname = oReader.GetString("thirdname");

        textBox4.Text = surname + " " + name[0] + ". " + thirdname[0] + ".";
    }

}

conn.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

public bool ReadDBBlobToFile(MySqlDataReader parReader, string parFilePath, string parColumnName)
{
    bool retResult = false;
    if (parReader == null)
    {
        throw new NullReferenceException("MySqlCommand is null");
    }

    int id = parReader.GetOrdinal(parColumnName);
    if (!parReader.IsDBNull(id))
    {
        string dir = Path.GetDirectoryName(parFilePath);
        if (string.IsNullOrEmpty(dir))
        {
            dir = Path.GetDirectoryName(Path.GetFullPath(parFilePath));
        }
        Directory.CreateDirectory(dir);

        using (FileStream fs = new FileStream(parFilePath, FileMode.OpenOrCreate, FileAccess.ReadWrite))
        {
            using (BinaryWriter bw = new BinaryWriter(fs))
            {
                long currentIndex = 0;
                long len = 100;
                byte[] blob = new byte[len];

                long BytesReturned = parReader.GetBytes(id, currentIndex, blob, 0, (int)len);

                while (BytesReturned == len)
                {
                    bw.Write(blob);
                    bw.Flush();
                    currentIndex += len;
                    BytesReturned = parReader.GetBytes(id, currentIndex, blob, 0, (int)len);
                }
                if (BytesReturned > 0)
                {
                    bw.Write(blob, 0, (int)BytesReturned);
                }
                bw.Flush();
                bw.Close();
            }
        }
    }
}

```

```

        fs.Close();
    }
    retResult = true;
}
else
{
    retResult = false;
}
return retResult;
}

private void button3_Click(object sender, EventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.Filter = "PNG image (*.png)|*.png";
    if (sfd.ShowDialog() == DialogResult.OK)
    {
        devmgmt.Save(sfd.FileName, ImageFormat.Png);
    }
}

private void button4_Click(object sender, EventArgs e)
{
    string sql = "SELECT * FROM issue WHERE issue_id='" + issue_id + "'";
    MySqlCommand oCmd = new MySqlCommand(sql, conn);
    conn.Open();
    using (MySqlDataReader oReader = oCmd.ExecuteReader())
    {
        while (oReader.Read())
        {
            SaveFileDialog sfd = new SaveFileDialog();
            sfd.Filter = "LOG file (*.log)|*.log";
            if (sfd.ShowDialog() == DialogResult.OK)
            {
                string strFilename = sfd.FileName;
                ReadDBBlobToFile(oReader, strFilename, "windowsupd");
            }
        }
    }
    conn.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    string sql = "SELECT * FROM issue WHERE issue_id='" + issue_id + "'";
    MySqlCommand oCmd = new MySqlCommand(sql, conn);
    conn.Open();
    using (MySqlDataReader oReader = oCmd.ExecuteReader())
    {
        while (oReader.Read())
        {
            SaveFileDialog sfd = new SaveFileDialog();
            sfd.Filter = "DMP file (*.dmp)|*.dmp";
            if (sfd.ShowDialog() == DialogResult.OK)
            {
                string strFilename = sfd.FileName;
                ReadDBBlobToFile(oReader, strFilename, "minidump");
            }
        }
    }
}

```

```

    }
    }
    conn.Close();
}

private void button6_Click(object sender, EventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.Filter = "PNG image (*.png)|*.png";
    if (sfd.ShowDialog() == DialogResult.OK)
    {
        screenshot.Save(sfd.FileName, ImageFormat.Png);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    AuthorForm authorForm = new AuthorForm(author_id);
    authorForm.Show();
}

private void button7_Click(object sender, EventArgs e)
{
    TaskForm taskForm = new TaskForm(issue_id);
    taskForm.Show();
}
}
}
}

```