

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему: Дослідження методів та розробка інформаційної системи
управління проектами компанії

Виконав: студент групи 6141 мз

_____ ***Касап Ю. С.***
(підпис)

Керівник роботи:

доцент кафедри ІУСТ, PhD,
(посада, науковий ступінь вчене звання)

_____ ***Ворона М. В.***
(підпис)

м. Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Ажищев В.Ф.

" ____ " _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Касапу Юрію Семеновичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів та розробка інформаційної системи управління проектами компанії _____

Керівник роботи Ворона Михайло Владиславович _____

Затверджені наказом ректора № 1098-уч. від "18" вересня 2023 року.

2. Термін подання роботи: 15 грудня 2023 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

• Титульний аркуш, завдання на дипломну роботу, анотація (українською, російською, англійською), зміст, перелік умовних означень, символів, одиниць та термінів (при необхідності) • Вступ • Аналіз предметної області і постановка задачі • Проведення досліджень за темою магістерської роботи • Реалізація проекту інформаційної системи (технічна документація по проекту) • Результати розробки • Організаційно-економічний розділ • Розділ з охорони праці • Висновки • Список використаної літератури • Додатки
Технічне завдання, текст програми, інструкція користувачу

5. Перелік презентаційних матеріалів _____

5.1 Об'єкт роботи, предмет і мета 5.2 Діаграма варіантів використання 5.3
Модель інформаційної роботи 5.4 Логічна структура бази даних 5.5 Фізична

модель бази даних 5.7 Технічне забезпечення 5.8 Алгоритм розрахунку
індексів якісної оцінки 5.9 Вихідні документи 5.10 Вихідні документи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ворона М. В.		
2	Ворона М. В.		
3	Ворона М. В.		
4	Ворона М. В.		
5	Ворона М. В.		
6	Ворона М. В.		

7. Дата видачі завдання 05 вересня 2022 р.

К

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області	07.09 - 08.10	
2	Розробка концепції системи	09.10. - 20.10	
3	Розділ з розробки проектних рішень по системі (та її частинах)	21.10 - 10.11	
4	Реалізація проекту та розробка робочої документації	11.11 - 03.12.	
5	Вирішення питань охорони праці, охорони навколишнього середовища	03.12 - 05.12	
6	Виконання графічних документів	06.12 - 07.12	
7	Оформлення пояснювальної записки	06.12 - 08.12	
8	Подання дипломної роботи (ДР) на кафедральний захист	09.12	
9	Подання ДР рецензенту	11.12	
10	Подання ДР на державний захист		

Студент

_____ (підпис)

Касап Ю. С.

(ПІБ)

Керівник роботи

_____ (підпис)

Ворона М. В.

(ПІБ)

АНОТАЦІЯ

Дана дипломна робота присвячена розробці інформаційної системи управління проектами. Дипломна робота містить короткий опис структури та проблематики управління проектами в компанії, проектну, технічну та експлуатаційну документацію, а також розділи з економіки, охорони праці та охорони навколишнього середовища.

Дипломна робота викладена на сторінках машинописного тексту й містить: рисунки, таблиці та список використаних джерел з найменувань.

Ключові слова: системний аналіз, інформаційна система, компоненти системи

ABSTRACT

This graduate work is devoted to the development of an integrated project management system for enterprise-level companies. Thesis contains a brief description of the structure and problems of project management in the company, the development of technical specifications, project of informational system, also section on economics, civil defense, measures for labour protection and protection of natural environment.

The degree work is committed on pages of typescript and contains: pictures, tables, appendixes and literature list of titles.

Keywords: system analysis, service system, information system, system components

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ	6
ВСТУП	7
1 ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ДЛЯ КОМПАНІЇ	8
1.1 Опис діяльності компанії "GeeksForLess"	8
1.2 Огляд існуючих рішень	14
1.3 Недоліки існуючих рішень	19
1.4 Розробка концепцій інтегрованої системи управління проектами рівня підприємства для компанії "GeeksForLess"	22
1.5 Функціонально-вартісний аналіз розроблених варіантів концепцій та вибір оптимального варіанту концепції	22
1.6. Постановка задачі	28
2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ДЛЯ КОМПАНІЇ "GeeksForLess"	29
2.1 Загальносистемні рішення	29
2.2 Рішення з інформаційного забезпечення	33
2.3 Рішення з математичного забезпечення	36
2.4 Рішення з програмного забезпечення	52
2.5 Рішення з організаційного забезпечення	64
2.6 Рішення з технічного забезпечення	65
3 РЕЗУЛЬТАТИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ДЛЯ КОМПАНІЇ "GeeksForLess"	69
3.1 Реалізація проекту розробки інформаційної системи	69
3.2 Тестування інформаційної системи	69
ВИСНОВКИ	72
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	73

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

АІУС – Автоматизована Інформаційна Управляюча Система.

АСУ – Автоматизована Система Управління.

БД – База Даних.

СУБД – Система Управління Базою Даних.

AMD – Advanced Micro Devices.

HTML – HyperText Markup Language.

LAN – Local Area Network.

SQL – Structured Query Language.

UML – Unified Modeling Language.

USB – Universal Serial Bus.

ПЗ – програмне забезпечення.

HTTP - HyperText Transfer Protocol (протокол передачі гіпертекста) .

SSL – Secure Sockets Layer (протокол захищених сокетів).

IP – Internet Protocol (Інтернет протокол)

ВСТУП

В наш час інформаційних технологій дуже розвинуті компанії які розробляють інформаційні системи. Якщо компанія займається розробкою багатьох проектів для багатьох замовників то дуже важко керівництву компанії контролювати розробку всіх проектів, та забезпечувати своєчасно виконання проектів. В цій роботі керівництву компанії може допомогти інформаційна система управління проектами.

Інформаційна система управління проектами підприємства (ІСУПП) – це набір інтегрованих програм, які комплексно, в єдиному інформаційному просторі підтримують всі основні аспекти управлінської діяльності підприємства - планування ресурсів (фінансових, людських, матеріальних) для виробництва товарів (послуг), оперативне управління виконанням планів (включаючи постачання, збут, ведення договорів), всі види обліку, ведення господарської діяльності.

Таким чином, якщо організація має багаторівневу систему управління і включає в себе кілька територіально розподілених підрозділів або філій, то управляти, а головне швидко і вчасно приймати правильні управлінські рішення, досить складно, це займає багато часу і сил. Тому, підприємству необхідна система, яка допомагає у вирішенні подібних завдань.

Метою даної дипломної роботи є розробка інформаційної системи автоматизованого управління проектами для компанії, яка має можливість створення звітів по проектам, ведення обліку робочого часу працівників та надання статистики роботи як окремих працівників так і відділів.

У задачі роботи входять опис структур, механізмів і можливостей типових систем управління проектами, дослідження й аналіз досвіду використання систем управління проектами, огляд ринку програмних продуктів з автоматизації управління проектами й самостійна розробка інформаційної системи управління проектами для компанії.

1 ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ДЛЯ КОМПАНІЇ

1.1 Опис діяльності компанії "GeeksForLess"

Компанія "GeeksForLess" була заснована в Миколаєві в 2004 році і за короткий час стала відома як надійний аутсорсинг-партнер в індустрії інформаційних технологій.

Аутс́орсинг (від англ. *outsourcing: (outer-source-using)* використання зовнішнього джерела / ресурсу) – передача організацією на підставі договору певних бізнес-процесів або виробничих функцій на обслуговування іншій компанії, що спеціалізується у відповідній області. На відміну від послуг сервісу і підтримки, що мають разовий, епізодичний, випадковий характер і обмежених початком і кінцем, на аутсорсинг передаються зазвичай функції по професійній підтримці безперебійної працездатності окремих систем і інфраструктури на основі тривалого контракту (не менше 1 року). Наявність бізнес-процесу є відмінною рисою аутсорсинга від різних інших форм надання послуг та абонентського обслуговування.

ІТ-аутсорсинг (ІТО) - передбачає делегування зовнішньої спеціалізованої компанії вирішення питань, пов'язаних з розробкою, впровадженням і супроводом інформаційних систем як цілком на рівні інфраструктури підприємства (супровід обладнання або ПЗ), так і обсягів робіт, пов'язаних з розвитком і / або підтримкою функціонування окремих ділянок системи (програмування, хостинг, тестування і т. д.)

На даний момент у компанії "GeeksForLess" існує більше десяти різних відділів. У кожному відділі може працювати до 50 співробітників. Крім цього, відділ може бути розподілений на кілька команд, які працюють як разом, так і незалежно один від одного.

В кожному відділі є свій менеджер. Якщо відділ займається розробкою більш одного проекту, то за виконанням та розробкою проекту стежить і управляє проектний менеджер або менеджер проектів (ПМ).

Менеджер проектів - це фахівець в області управління проектами, який несе відповідальність за планування, підготовку та виконання конкретного

проекту. Зазвичай така посада і діяльність має місце в галузях, схильних до проектних робіт, наприклад: ІТ-області, будівництві та ін

При цьому, до основних навичок (крім професійних навичок у певній галузі) менеджера проектів в першу чергу відносяться навички управління людьми, планування робіт, а також управління ризиками.

Виконувати проект може як одна, так і декілька команд. У кожній команді, якщо вона займається розробкою, є старший розробник (тім-лідер). Схема відділу зображена на рисунку 1.1

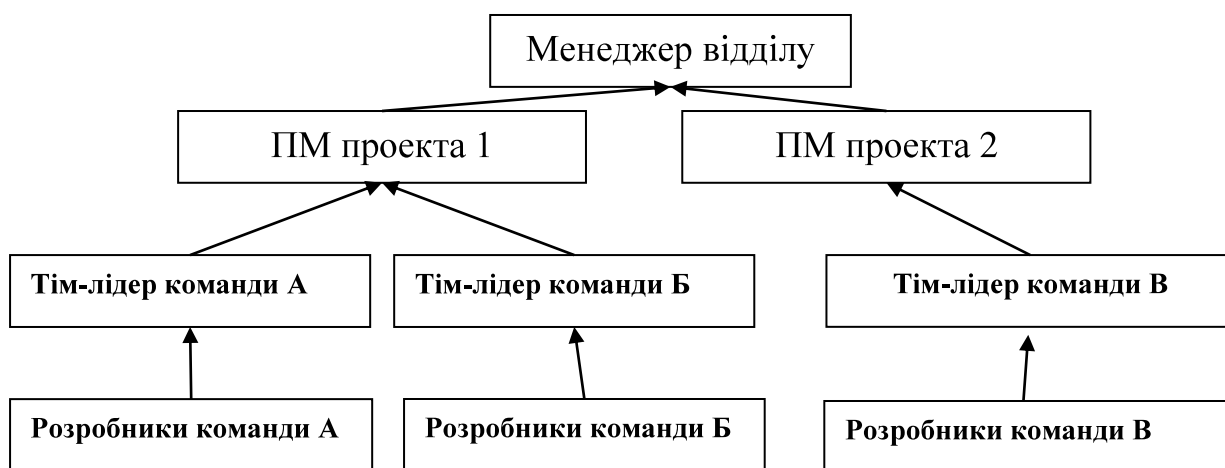


Рисунок 1.1 – Структура відділу розробників

Якщо відділ не займається розробкою, а працює, наприклад, над тестуванням додатків, то він розбивається, у випадку, якщо він великий, на кілька менших за кількістю співробітників підвідділів. У такій ситуації у відділі є головний менеджер, а підвідділами керують помічники менеджерів. Теж саме стосується й інших відділів, які не займаються розробкою. Наприклад, відділів технічної підтримки та відділу системних адміністраторів. Схема такого відділу зображена на рисунку 1.2.

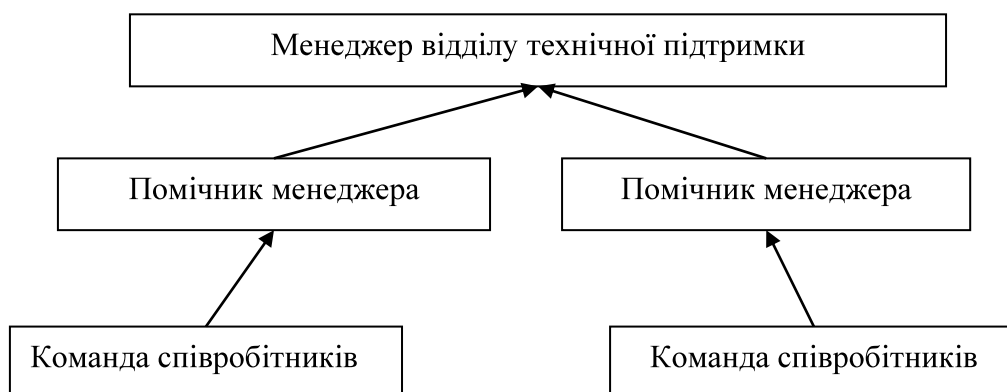


Рисунок 1.2. – Структура відділу технічної підтримки.

Декількома відділами, об'єднаними по замовнику, або іншою ознакою керують віце-президенти компанії (ВП), які в свою чергу підпорядковуються і звітують президенту компанії.

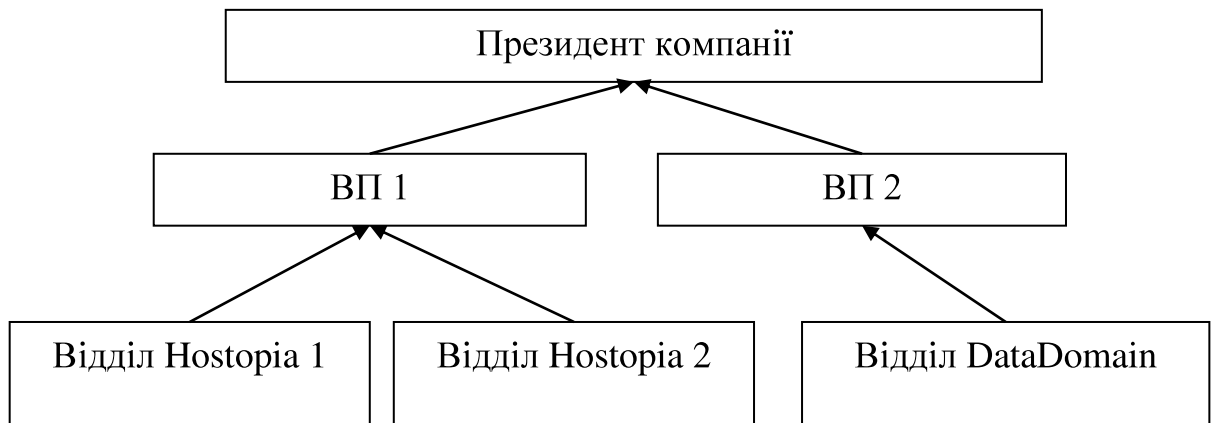


Рисунок 1.3 – Структура управління компанії.

Крім цього, можливі ситуації, коли групою розробників або інших співробітників керує безпосередньо тим-лідер або проектний менеджер замовника. Він регулює завантаженість, роздає завдання та контролює їх виконання.

Також можливі ситуації, коли команда розробників або співробітників, керовані проектним менеджером, працюють спільно із ще однією командою співробітників замовника. Якою, у свою чергу, керує інший проектний менеджер, а координує дії – менеджер відділу з боку замовника.

У компанії є відділ кадрів, який веде облік співробітників, їх рівень кваліфікації, освіти і технологій, якими вони володіють.

Отже, на даний час, старше керівництво отримує письмові або усні, на зборах менеджерів відділів, звіти про стан справ і проектів у відділах. Так само на зборах приймають рішення про нові проекти, якому відділу зайнятися його розробкою. Визначають, які фахівці необхідні, скільки і в який відділ.

Управління проектами (англ. *project management*) — область діяльності, в ході якої визначаються та досягаються чіткі цілі при балансуванні між обсягом робіт, ресурсами (такими як гроші, праця, матеріали, енергія, простір і ін), часом, якістю та ризиками в рамках деяких проектів, спрямованих на досягнення певного результату при зазначених обмеженнях. Головним фактором успіху проектного управління є наявність

чіткого, заздалегідь визначеного, плану мінімізації ризиків і відхилень від нього (на відміну від процесного, функціонального управління, управління рівнем послуг).

Планування і управління проектами в кожному відділі відбувається по-різному. У відділах, де розробка проекту ведеться «з нуля» або за проектом, проводиться великий обсяг заздалегідь поставлених завдань. Менеджер проекту розподіляє роботу, навантаження тощо між учасниками команди.

Для спрощення автоматизації і полегшення своєї роботи і роботи команди він може використовувати програмне забезпечення для управління проектом, таке як "MS Project" або будь-яке інше програмне забезпечення для управління проектом. Крім цього, при роботі над проектом для опису перегляду статусу (виправлена \ немає) знайдених помилок, використовують системи відстеження помилок (багтрекер).

Система відстеження помилок (англ. *bug tracking system*) — прикладна програма, розроблена з метою допомогти розробникам програмного забезпечення (програмістам, тестувальникам та ін) враховувати і контролювати помилки (баги), знайдені в програмах, побажання користувачів, а також стежити за процесом усунення цих помилок і виконання або невиконання побажань.

А також відбувається написання різного виду документів і документації. Інформацію, про використовувані системи відстеження помилок, програмного забезпечення для управління проектами та написання документації, класифікованої по відділах, наочно видно в таблиці 1.

Таблиця 1 – Технології, що використовуються по відділам.

Відділ	ПЗ для управління проектами	Система відстеження помилок	Документація
DataDomain	Google Docs	Bugzilla	Google Docs
UPA Hostopia Admins	Redmine	Bugzilla	Open Office
Template Rovers	MS Project	Bugzilla	MS Office
Brisk Mobile	Primavera	Bugzilla	MS Office
Buce	Redmine	Redmine	Google Docs
Hostopia QA	-	Bugzilla	Open Office
Deluxe	Redmine	Redmine	Google Docs

Hostopia Programmers	MS Project	На основе Bugzilla	MS Office
Designers	-	На основе Bugzilla	MS Office
Tech Support	-	На основе Bugzilla	Google Docs

Використання того чи іншого програмного забезпечення для управління проектами, системи відстеження помилок та написання документації обумовлено вимогами клієнтів. Організаційна схема управління проектами підприємства з використанням програмного забезпечення для управління проектами зображена на рисунку 1.4

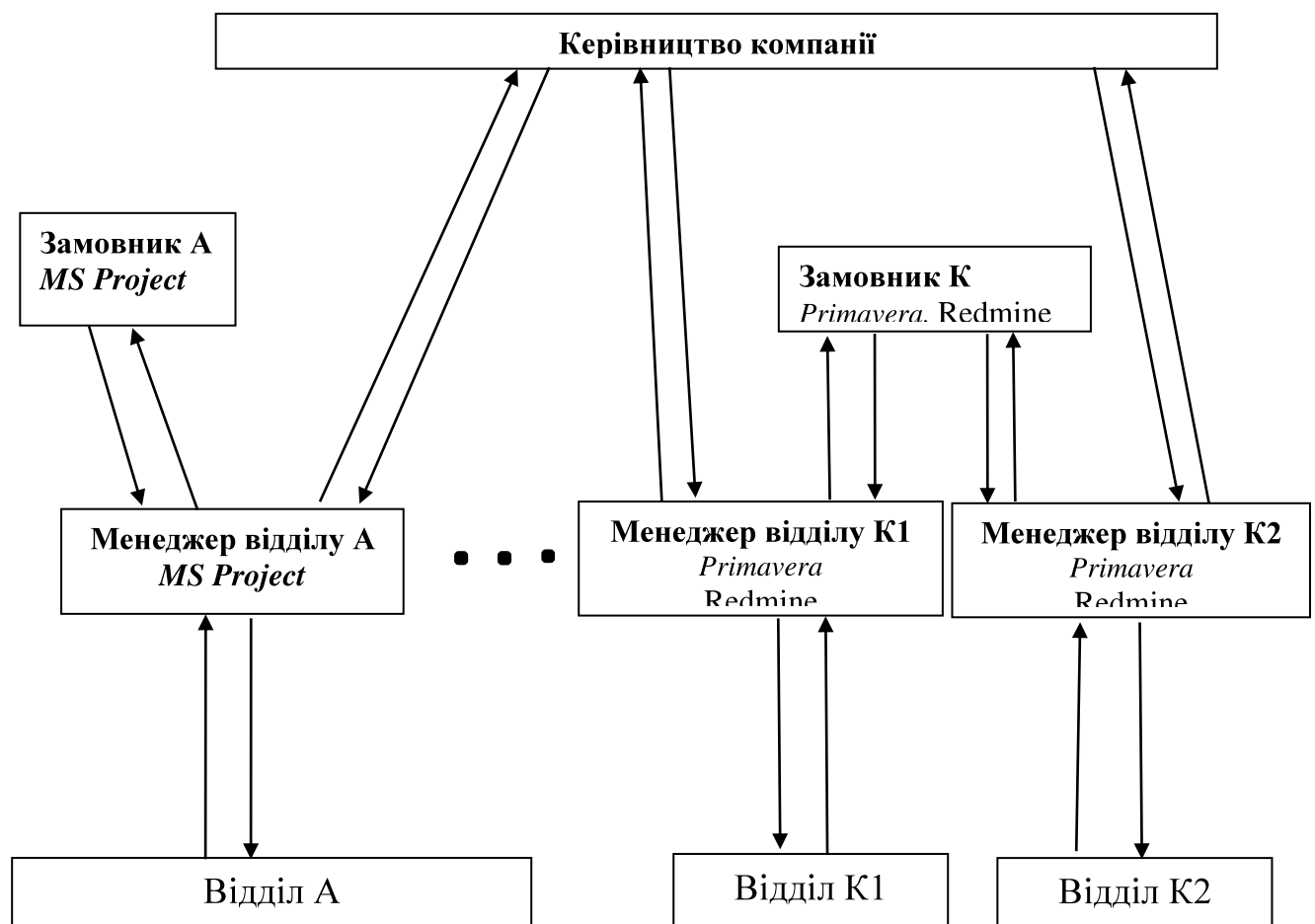
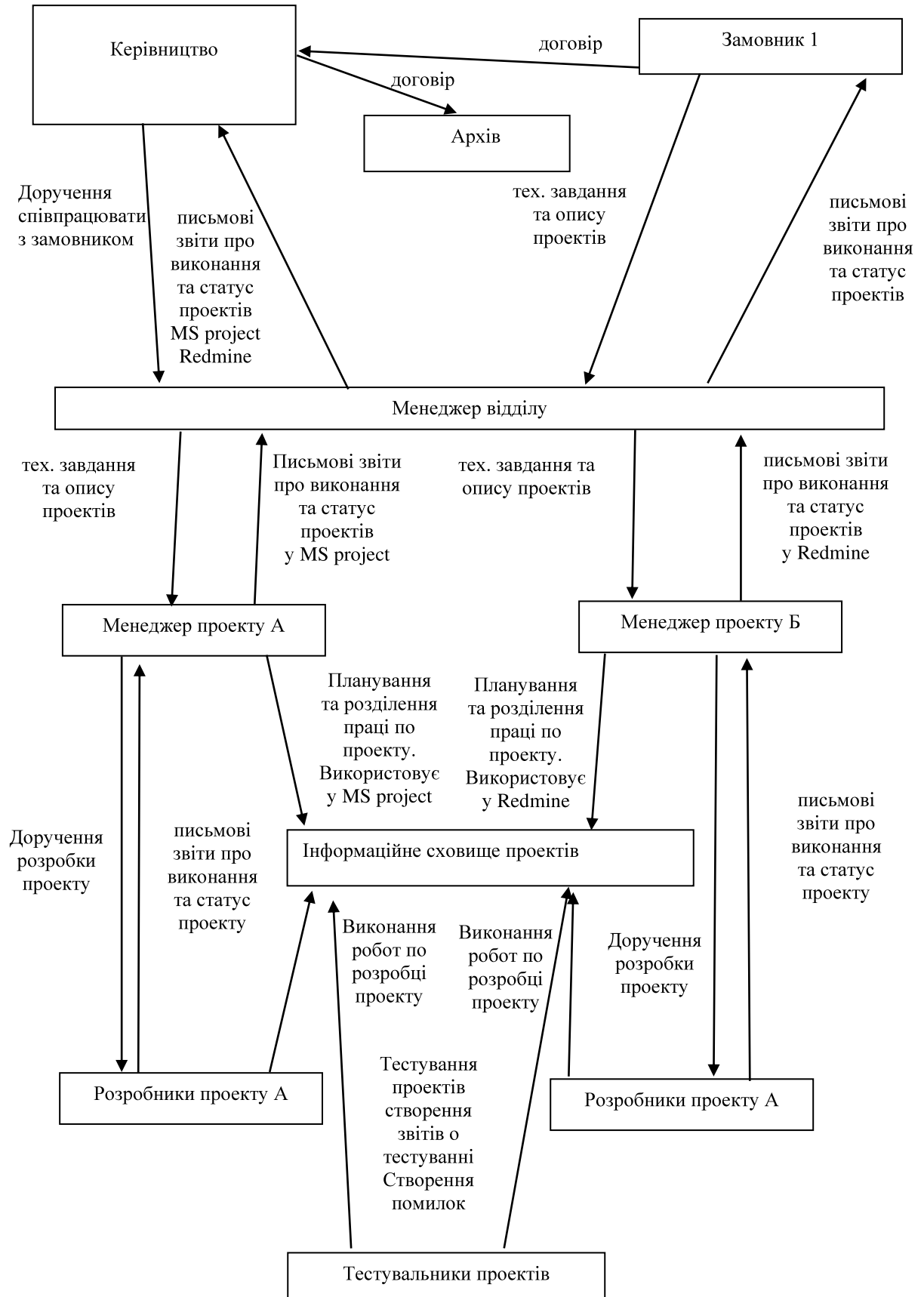


Рисунок 1.4 – Схема організації управління проектами підприємства

Для вивчення інформаційних потоків між відділами, менеджерами, розробниками, керівництвом та замовниками компанії "GeeksForLess", та відношень між ними розробимо модель організаційної структури підприємства. Для розробки моделі організаційної структури може бути використана нотація DF діаграм. Схема інформаційних потоків організаційної структури управління проектами компанії показана на рисунку 1.5.

Співробітники відділів у кінці робочого дня в листі або усно звітують проектному менеджеру про об'єм виконаної роботи. Менеджер проектів, виходячи з цієї інформації, вносить дані про просування проекту, звітує менеджеру відділу або замовнику. Таким чином і відбувається робота компанії.



1.2 Огляд існуючих рішень

Система управління проектами підприємства - це набір інтегрованих програм, які комплексно в єдиному інформаційному просторі підтримують всі основні аспекти управлінської діяльності підприємства - планування ресурсів (фінансових, людських, матеріальних) для виробництва товарів (послуг), оперативне управління виконанням планів (включаючи постачання, збут, ведення договорів), всі види обліку, ведення господарської діяльності. Також інтегрована система управління проектами підприємства - це централізація даних у єдиній базі, близький до реального часу режим роботи, збереження загальної моделі управління для підприємств будь-яких галузей, підтримка територіально розподілених структурних одиниць підприємства.

Нині існує безліч різноманітного програмного забезпечення для управління проектами. Найбільш поширені програми для вирішення завдань планування термінів і ресурсів. Розглянемо найбільш популярні з них.

MS Project

Розробник: Microsoft Corp./ США

Дозволяє планувати нескладні комплекси робіт з урахуванням їх взаємозв'язку і розподілу між виконавцями з вирівнюванням ресурсів.

- Автоматизована підготовка звітів та "знімки" поточного стану проекту
- Обмін інформацією між менеджером і виконавцем через MS Outlook
- Має WEB-орієнтований додаток MS Project Central, який надає інструменти для підтримки спільної роботи членів команди
- З основних особливостей, пов'язаних із плануванням і контролем за виконанням проекту, що з'явилися у новій версії, можна відзначити наступні:
 - Ієрархічні коди структур для завдань і ресурсів;
 - необхідні матеріали як вид ресурсів;
 - необхідні матеріали як вид ресурсів;
 - місяць як одиниця тривалості робіт;

- індивідуальні календарі для завдань;
 - графічні індикатори для наочного подання "проблемних" завдань; поля, що обчислюються формулами, обумовленими користувачами;
 - дві шкали часу (основна і допоміжна);
 - зазначення приблизної періоду виконання завдання (з подальшим уточненням);
 - створення шаблонів проектів.
- Поліпшені функції підтримки ієрархічної структури робіт (WBS), визначення пріоритетів для завдань, вирівнювання завантаження ресурсів, розрахунку критичного шляху в групі проектів, перегляду мережевий діаграми проекту.
 - Microsoft адресує свій продукт не тільки початківцям, але й професійним керівникам проектів. Для розширення функціональності системи виробником випускаються Add-in-модулі, доступні для безкоштовного завантаження через Internet. Крім того, існує велика кількість безкоштовних і комерційних продуктів сторонніх виробників.
 - Що стосується зручності роботи, то тепер пакет підтримує такі функції, як які самостійно меню, одновіконний інтерфейс, установку по запиту, переміщувані профілі користувачів, режим Windows Terminal Server тощо. Довідкова система дозволяє користувачам додавати в неї свою інформацію.
 - Особливу увагу приділено організації обміну інформацією в команді проекту. Новий продукт Microsoft Project Central дозволяє всім учасникам проекту працювати з актуальною проектною інформацією. Доступ здійснюється або через Web-браузер, або через спеціальний додаток MS Project Central Client. Для мобільних та віддалених користувачів передбачена робота в режимі off-line з подальшою синхронізацією інформації.
 - Працюючи із системою, можна не тільки переглядати поточну інформацію, а й оновлювати дані — вносити нові задачі, перерозподіляти задачі між виконавцями і т. д. Права доступу

визначаються адміністратором. Перегляд можливий як у режимі таблиць, так і у вигляді діаграми Ганнта.

- Підтримується двонаправлений обмін інформацією з Outlook. Керівник проекту може передати виконавцям дані про завдання (у вигляді списку), які необхідно виконати, а ті, у свою чергу, можуть інформувати його про всі зміни в робочому календарі. Крім того, користувачі MS Outlook 2000 мають можливість переглядати всю проектну інформацію з цього додатка.

Primavera Project Planner

Розробник: Primavera Systems Inc. / США

Центральний програмний продукт сімейства Primavera для координації великої кількості середніх і великих проектів в рамках обмежених ресурсів.

Дозволяє об'єднання фрагментів проектів, підготовлених у SureTrak.

Має кілька десятків стандартних шаблонів представлення проекту, а також можливість створювати і зберігати власні макети та звіти з публікацією на web-сайті компанії.

Розвинена функція глобальної заміни для внесення змін в дані проекту з використанням логічних, арифметичних і рядкових виразів.

Дозволяє експорт даних в програми Dbase і Lotus

Мережева версія дозволяє працювати паралельно, оновлюючи й аналізуючи інформацію, готуючи звіти по безлічі проектів одночасно.

Дозволяє працювати з необмеженим набором основних проектів і підпроектів в рамках прав доступу, що обмежують можливості користувача переглядом і / або зміною тієї або іншої інформації у відповідності з його повноваженнями.

Open Plan

Розробник: Welcom Software / США

- Повністю русифікована професійна програма планування і контролю великих проектів у масштабах підприємства

- Потужні засоби ресурсного і вартісного планування
- Ефективна організація мультипроектного і розрахованої на багато користувачів роботи, що дозволяє створити відкрите і масштабне рішення для всього підприємства
- Функція ДУП (директор управління проектами) для автоматизації повторюваних процесів при управлінні проектами
- Реалізовано планування при обмежених часі й ресурсах та тип матеріальних ресурсів з обмеженим терміном зберігання
- Розмежування рівнів доступу до проектних даних здійснюється за допомогою спеціальної утиліти SysAdm
- Вбудована функція створення архіву проекту і модуль WEB Publisher для публікації даних проекту на WEB-сервері
- Випускається у варіантах Enterprise, Professional і Desktop, кожен з яких відповідає різним потребам менеджерів і виконавців проекту
- Комбінація з Cobra дозволяє побудувати інтегровану систему управління календарним графіком та витратами проекту

WelcomHome

Розробник: Welcom Software / США

- Засіб організації групової роботи на основі WEB-технологій для ведення проектів з використанням єдиного WEB-інтерфейсу доступу до всієї проектної інформації в режимі реального часу через Internet
- Взаємодія учасників проектів забезпечується в рамках єдиного порталу через електронну пошту, дошки оголошень і дискусійні форуми, при цьому також регламентується введення прогресу робіт виконавцями
- Використовуються шаблони процедур управління проектами, що налаштовуються для забезпечення формалізованого і стандартизованого підходу до виконання робіт та уніфікації корпоративних управлінських процесів
- Можлива інтеграція з системами календарного планування MS Project, OpenPlan і системою бюджетування проектів Cobra для забезпечення

доступу членам проектної групи до необхідних календарним і вартісним даним за проектом

- Платформа: Windows NT/2000/XP
- Бази даних: Oracle, MySQL, MS Access
- WEB: сервер MS IIS, Apache, браузер Internet Explorer, система безпеки забезпечує розмежування доступу до проектних даних і функціонування системи за рахунок адміністрування користувачів, груп і ролей.

Redmine

Розробник: Jean-Philippe Lang

- ведення декількох проектів;
- гнучка система доступу, що заснована на ролях;
- система відстеження помилок;
- діаграми Ганта та календар;
- ведення новин проекту, документів та управління файлами;
- оповіщення про зміни за допомогою RSS-потоків та електронної пошти;
- вікі для кожного проекту;
- форуми для кожного проекту;
- облік часових витрат;
- настраюються довільні поля для інцидентів, тимчасових витрат, проектів та користувачів;
- легка інтеграція із системами управління версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs);
- створення записів про помилки на основі отриманих листів;
- підтримка множинної аутентифікації LDAP;
- можливість самостійної реєстрації нових користувачів;
- багатомовний інтерфейс (у тому числі російська);
- підтримка СУБД MySQL, PostgreSQL, SQLite, Oracle.

1.3. Недоліки існуючих рішень

Розглянемо недоліки існуючих рішень.

Порівняльну характеристику розглянутого програмного забезпечення для управління проектами можна подивитися у таблиці 2.

Як видно з таблиці 2 майже всі ПЗ працюють на платформі Windows. Крім цього можна відзначити такі недоліки:

- Управління файлами і документами в Redmine зводиться до їх додавання, видалення та редагування. Правами доступу ні до файлів, ні до окремих документів управляти не можна.
- Відсутні оповіщення про зміну документів.
- У Redmine не можна керувати правами доступу на рівні окремих полів завдання. Наприклад, на даний момент від клієнтів не можна приховати оцінки часу роботи над проектом або інформацію про витрачений час.
- У Redmine можна керувати правами доступу на рівні проектів, але не можна призначити права на якусь версію проекту або окреме завдання. Це означає, що якщо користувачу потрібен доступ усього до однієї задачі, то доведеться давати доступ до всього проекту.
- Якщо користувач Redmine отримав доступ до проекту, то зараз не можна обмежити його активність якимись окремими типами задач (трекерами). Наприклад, не можна дозволити переглядати тільки «свої» завдання (це часто буває потрібно для організації техпідтримки) або дозволити створювати завдання тільки якогось певного типу.
- У Redmine всі додаткові поля доступні всім користувачам, всі учасники проекту зможуть їх бачити і змінювати. Це обмеження може призвести до складностей при наявності неоднорідної команди, коли доступ до проекту мають і менеджери, і розробники, і клієнти.
- У Redmine немає прав на окремі типи конвертацій в workflow. Наприклад, зараз не можна вказати, що коли хтось закінчує виправляти помилку, він повинен вибрати відповідальним тестувальника і повинен

вказати номер білду. Також не можна приховати внутрішню переписку між програмістами від клієнта.

- MS Project розрахований на меншу кількість завдань, ніж вам потрібно. Середній документ, ймовірно, містить менше ста завдань.
- MS Project не передбачає що виконавці взаємозамінні.
- MS Project, Primavera, Open Plan, WelcomHome не мають системи відстеження помилок.

Таблиця 1.2 – Порівняльна характеристика програмного забезпечення для управління проектами

Параметр	MS Project	Primavera	Open Plan	Redmine	WelcomHome
<i>Розробник</i>	Microsoft	Primavera Systems	Welcom Software	Jean-Philippe Lang	Welcom Software
<i>Наявність версій різного ступеня складності</i>	ні	Велика кількість додаткових пакетів	Enterprise, Profesional, Desktop	ні	ні
<i>Імпорт/експорт</i>	MS Access, Excel, ODBC, FoxPro, CSV, Lotus, OLE	ODBC, OLE, DDE	SQL, Oracle, Sybase, xBase	формат CSV, Oracle, MySQL Server,	MS Access, SQL Server, Oracle
<i>Спільне використання з іншими додатками</i>	MS Exchange	SureTrak	MS Project, P3, SAP E/3, Baan	-	MS Project, Open Plan, Cobra
<i>Платформа</i>	Повна сумісність з MS Windows	Windows 9x / NT / 2000	Windows 9x / NT / ME / 2000 XP	Крос-платформений	Windows NT / 2000 / XP
<i>Безпека даних</i>	-	права доступу користувачів	утиліта SysAdm	контроль доступу користувачів	Адміністрування користувачів, їх груп та ролей
<i>Мережева версія</i>	MS Project Central	Primavera Post Office	так	так	-
<i>Інтеграція з Internet</i>	для мережевої версії інтеграція з MS Exchange	MS Mail, публікація звітів на WEB-сервері	WEB Publisher для публікації на WEB-сервері	так	WEB-інтерфейс, e-mail, дошки оголошень, форуми

Крім цього, специфіка роботи компанії не дозволяє нав'язати всім відділам використання тільки одного і ніякого іншого програмного забезпечення. Так як використання програмного забезпечення зумовлена вимогами замовника.

Жодна із систем управління проектами не дозволяє працювати зі всіма використовуваними компанією і замовниками системами управління. Тобто, немає можливості імпортувати всі дані в одну систему для використання вищим керівництвом компанії в цілях моніторингу працездатності та просувань різних відділів компанії.

В системах управління немає можливості оцінювати співробітника дивитися ступінь його завантаженості.

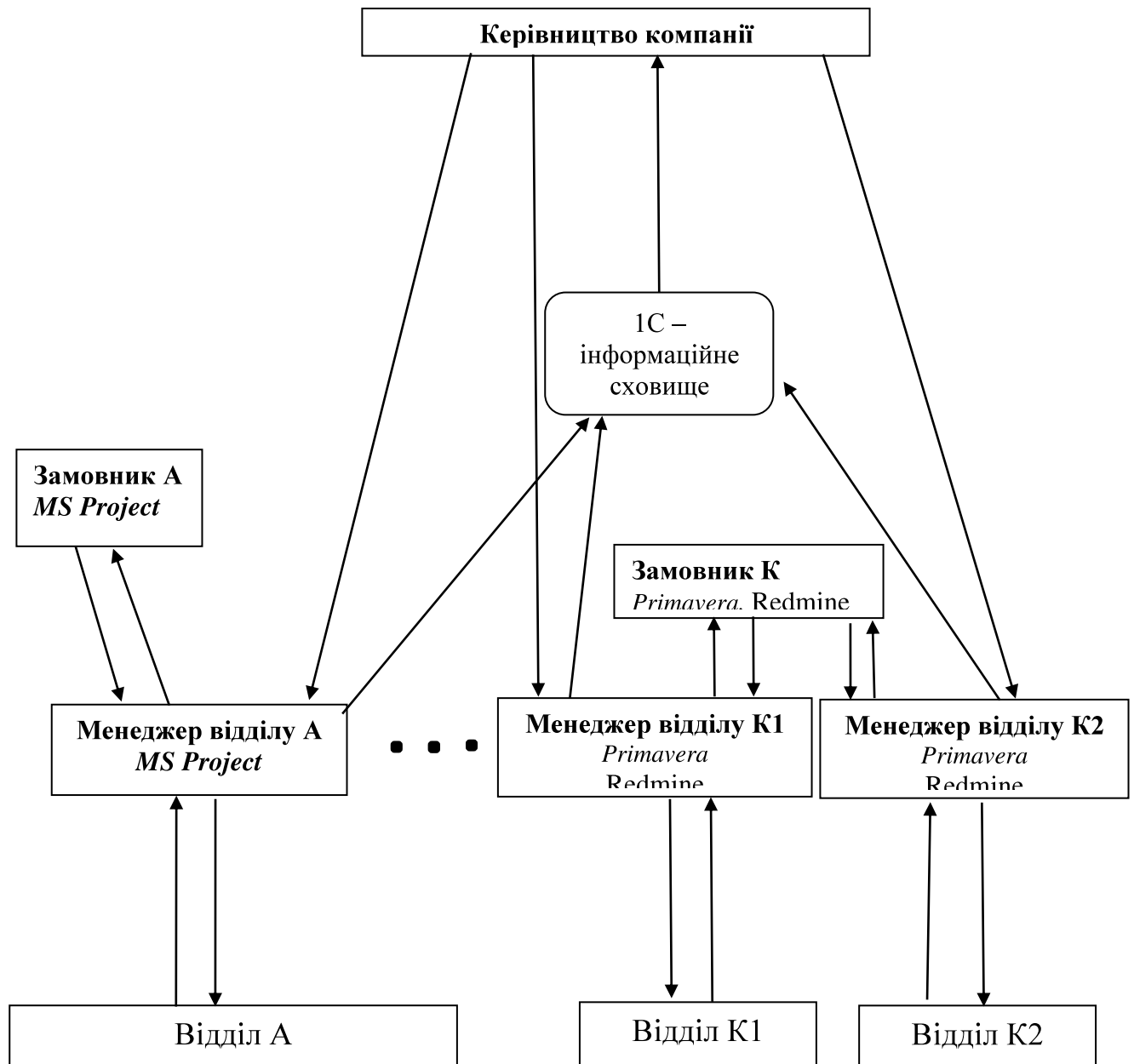


Рисунок 1.7 – Модель управління проектами на основі 1C – інформаційне сховище завдань та проектів

1.4 Розробка концепцій інтегрованої системи управління проектами рівня підприємства для компанії "GeeksForLess"

У ході аналізу діючої інформаційно-управляючої системи були виявлені істотні недоліки, які впливають на ефективність роботи фірми в цілому. Та у зв'язку з неможливістю використовувати ні одне з існуючих рішень для управління проектами. Так як для кожного відділу можливе використання своєї системи управління, обумовлене вимогами замовників. Тому були розроблені варіанти концепцій.

Перша концепція системи, на основі інформаційної системи – інформаційне сховище завдань та проектів. (рисунки 1.6) Система полягає впровадженні використання інформаційного сховища в якому буде зберігатися інформація про проекти та завдання по проектах. Яка буде надавати керівництву поточну інформацію про стан проекту. А керівництво на основі наданої інформації керує виконанням проекту та приймає різні рішення.

Друга концепція системи, на основі інформаційно-управляючої системи управління проектами компанії рівня підприємства. (рисунки 1.7) Система полягає впровадженні використання інформаційно - управляючої системи, яка буде імпортувати данні о проектах, завданнях помилках. Крім цього система буде аналізувати стан проекту кількість завдань та розподіляти їх між працівниками відділу. Надсилати інформацію керівникові та розробникам.

1.5 Функціонально-вартісний аналіз розроблених варіантів концепцій та вибір оптимального варіанту концепції

Для визначення оптимального варіанту концепції проведемо функціонально-вартісний аналіз. Маємо 2 варіанти концепцій:

- 1) ПЗ Нової системи управління проектами розміщена на окремому сервері на базі ОС Fedora 13 У користувачів встановлюється клієнтська програма з графічним інтерфейсом. Основна мова програмування серверної

частини Ruby on Rails з користувачами відбувається через WEB інтерфейс системи.

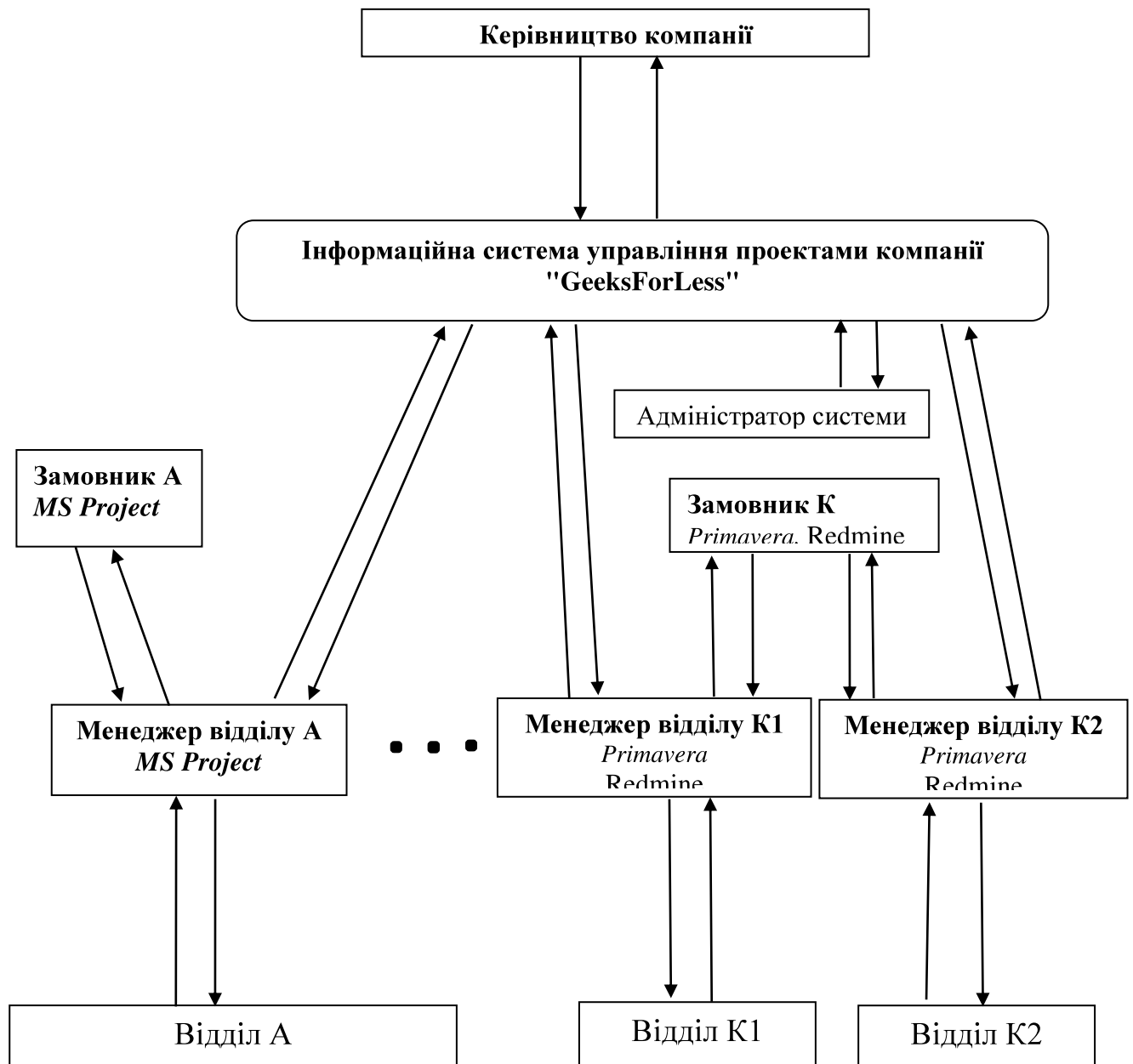


Рисунок 1.7. – Модель управління проектами на основі ІСУП.

- 2) ПЗ Нової системи управління проектами розміщена на окремому сервері на базі ОС Fedora 13 У користувачів встановлюється клієнтська програма з графічним інтерфейсом. Основна мова програмування серверної частини Ruby on Rails з користувачами відбувається через WEB інтерфейс системи.

Оцінимо кожну з концепцій по основних параметрах за допомогою оцінок експертів (таблиця 1.3), визначимо коефіцієнти значимості кожного з параметрів. Результат наведений у таблиці 14

Таблиця 1.3 - Оцінки експертів

Властивості системи	Експерт 1		Експерт 2		Експерт 3		Середнє	
	1	2	1	2	1	2	1	2
1. Зручність	8	10	9	9	7	9	8	9
2. Надійність системи	3	9	2	10	7	8	4.3	9
3. Функціональність системи	5	10	7	7	6	10	6	9

Таблиця 1.4 - Оцінка варіантів реалізації по параметрах і коефіцієнта значимості кожного параметра

Властивість	Значення параметра по варіантам концепцій, B_i		КЗ, φ_i
	1	2	
1.Зручність	8	9	0.2
2. Надійність системи	4	9	0.6
3. Функціональність системи	6	9	0,4

Після визначення вагомих коефіцієнтів необхідно визначити коефіцієнти технічного рівня для кожної з властивостей по кожному з варіантів по формулі 1.1:

$$K_{TV} = \sum_{i=1}^n \varphi_i B_i \quad (1.1)$$

Результати розрахунків коефіцієнтів технічного рівня приведені в таблиці 1.3.

За даними таблиці 1.5 розраховується коефіцієнт технічного рівня для шкірного з варіантів по формулі 1.2:

$$K_{TV}[F_i] = \sum_i K_{TV}[F_i] \quad (1.2)$$

Результати розрахунків наведені в таблиці 1.6.

Таблиця 1.5 - Розрахунок показників технічного рівня

Властивість F_i	Варіант	Оцінка властивостей у балах, B_i	Ваговий коефіцієнт, φ_i	Показник технічного рівня
1.Зручність адміністрування ОС	1	10	0,2	2,0
	2	10		2.0
2. Надійність системи	1	4	0,6	2,8
	2	9		5,4

3.Функціональність системи	1	7	0,2	1,4
	2	9		1,8

Таблиця 1.6 - Коефіцієнти технічного рівня

Варіант	Коефіцієнт технічного рівня, $K_{\text{ТУ}}$
1	7.1
2	8,4

Найкращим на етапі функціонального аналізу є варіант, що має максимальне значення коефіцієнта технічного рівня $K_{\text{ТУ}}$, тобто варіант 2.

На іншому етапі необхідно виконати вартісний аналіз. Для цього розглядаються зведені витрати по кожному j -му варіанту (формула 1.3):

$$Z_j = C_j + E_n K_j \quad (1.3)$$

де C_j - поточні витрати по j -му варіанту; K_j - капітальні вкладення по j -му варіанту; E_n - нормативний коефіцієнт економічної ефективності капітальних вкладень, що дорівнює 0,15.

Зведені витрати Z_j розраховуються в сфері виробництва (у розробника) або в сфері експлуатації (у користувача ІС або ПП).

Поточні витрати при розробці ІС або ПП C_j - це собівартість $C_{\text{ПП}}$ ІС

$$\text{або ПП: } C_{\text{ПП}} = C_3 + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}}, \quad (1.4)$$

де C_3 - заробітна плата розробників ІС або ПП; $C_{\text{ВІД}}$ - відрахування на соціальні заходи (47,5% від фонду оплати праці); $C_{\text{М}}$ - вартість машино-годин, необхідних для розробки і налагодження ІС або ПП; $C_{\text{Н}}$ - накладні витрати в розмірі 50-150% від витрат на оплату праці (охоплюють оплату праці управлінського персоналу з нарахуваннями; оплату службових відряджень; оплату консультаційно-інформаційних послуг; оплату ремонту і техобслуговування основних фондів, крім ПК; оплату оренди приміщення тощо).

$$\text{Витрати на оплату праці розробника ІС або ПП: } C_3 = C_{\text{Г}} T, \quad (1.5)$$

де C_{Γ} - погодинна оплата праці програміста; T - трудомісткість розробки ІС або ПП.

$$C_{31} = 350 \cdot 20 = 7000 \text{ грн.}$$

$$C_{32} = 350 \cdot 20 = 7000 \text{ грн.}$$

Вартість машинної години: $C_M = C_{M-\Gamma} t_M$, (1.6)

де t_M - тривалість машинної години, необхідного для розробки ІС або ПП, рік.; $C_{M-\Gamma}$ - собівартість однієї машино-години роботи ПК або приладу при використанні власного ПК і величина орендної плати за 1 час роботи ПК, якщо машина орендується.

$$\text{Собівартість 1 машино-години ПК: } C_{M-\Gamma} = \frac{\sum_i U_i}{T_{\text{ЕФ}}}, \quad (1.7)$$

де U_i - річні експлуатаційні витрати при обслуговуванні ПК, що охоплюють:

- заробітну плату спеціаліста, що обслуговує машину (з поправкою на коефіцієнт зайнятості робітника обслуговуванням ПК);
- нарахування на заробітну плату (37,5%);
- витрати на електроенергію, які визначаються як добуток вартості 1 квт за час електроенергії на обсяг потужності, що споживається ПК, і на ефективний фонд години ПК (за рік);
- накладні витрати-50-150% від витрат на оплату праці;

$T_{\text{ЕФ}}$ - ефективний фонд години роботи ПК за рік, що визначається виходячи з календарного річного фонду години за винятком вихідних та святкових днів.

Для усіх варіантів:

$$C_{M-\Gamma} = (200 + 200 \cdot 0,375 + 200 \cdot 0,5 + 328,86) / 1512 = 0,47 \text{ грн/рік}$$

Капітальні витрати - K_j включають витрати на придбання ПК, витрата на навчання персоналу, розширення і реконструкцію приміщення тощо, витрати, що мають одноразовий характер:

$$K_j = K_{\text{ПК}} \frac{t_M}{T_{\text{ЕФ}}}, \quad (1.8)$$

де $K_{\text{ПК}}$ - вартість придбання, транспортування та налагодження ПК або приладу.

Вартість необхідного програмного забезпечення на сьогоднішній день наведена в таблиці 1.6

Таблиця 1.6 - Вартість ПЗ

Варіант	Найменування ПЗ	Вартість, грн	Кількість, грн	Разом, грн
1	ОС Fedora 13, Rails, ruby	50	1	50
2	ОС Fedora 13, Rails, ruby	50	1	50

Капітальні вкладення по шкірному варіанті по кожній з концепції по потоковому стані ринку ПЗ приведені в таблиці 1.7

Таблиця 1.7 Капітальні вкладення по варіантах

Варіант	Вартість, грн
1	3547
2	3547

Визначається коефіцієнт техніко-економічного рівня по шкірному j-му варіанту: $K_{\text{ТЕР}_j} = \frac{K_{\text{ТР}_j}}{3_j}, \quad (1.9)$

Результати розрахунку приведені в таблиці 1.8

Таблиця 1.8 - Коефіцієнт техніко-економічного рівня по варіантах

Варіант	Зведені витрати
1	$2,228 \cdot 10^{-4}$
2	$2,711 \cdot 10^{-4}$

Найкращий варіант визначається за максимальним значенням коефіцієнта КТЕУ. Отже найкращий варіант АІУС за економічним аналізом є також другий варіант.

1.6. Постановка задачі

На основі опису структури компанії "GeeksForLess", потоків даних в ній, діючої моделі управління проектами рівня підприємства та розроблених концепцій з метою конкретизації задач проектування, сформуємо коло задач, які необхідно вирішити для досягнення цілей поставлених на початку роботи. Необхідно розробити систему управління проектами рівня підприємства, яка дозволить імпортувати дані з інших систем управління, що використовуються відділами, помітити усні та письмові звіти керівництву про стан справ, бачити актуальний стан справ по кожному проекту та відділу. Кількість задіяних і вільних співробітників, ступінь навантаження на співробітників

Детальна постановка задачі представлена в розробленому технічному завданні у додатку А.

2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ДЛЯ КОМПАНІЇ "GeeksForLess"

2.1 Загальносистемні рішення

2.1.1 Засоби проектування

У якості основного інструмента для проектування обрано UML. UML – уніфікована мова об'єктно-орієнтованого моделювання, використовується у парадигмі об'єктно-орієнтованого програмування [9]. Є невід'ємною частиною уніфікованого процесу розробки інформаційної системи.

Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

2.1.2 Схема функціональної структури

Для уточнення функціональності інформаційної системи, виявлення схеми функціональної структури, визначення модулів, які входять до її складу, і вивчення динаміки роботи системи в цілому побудуємо діаграму прецедентів, яка відображає системні прецеденти (use cases), системне оточення (діючих осіб або акторів - actors) та зв'язки між ними (діаграма прецедентів – use cases diagrams). Головна задача моделі прецедентів – представити собою єдиний засіб, який дає замовнику, кінцевому користувачу та розробнику разом обговорювати функціональність і поведінку системи.

Моделі прецедентів відображає інтерфейс системи з зовнішнім світом, а саме, інформаційні потоки між системою й зовнішніми сутностями, з якими вона повинна бути зв'язана.

Актор являє собою будь-яку зовнішню стосовно системі сутність, яка взаємодіє із системою й використовує її функціональні можливості для досягнення певної мети або вирішення завдань. При цьому актори служать для позначення погодженого безлічі ролей, які можуть відіграти користувачі в процесі взаємодії із проектованою системою. Кожний актор може розглядатися як якась окрема роль щодо конкретного варіанта використання.

Модель прецедентів представлена на Рисунку 2.1

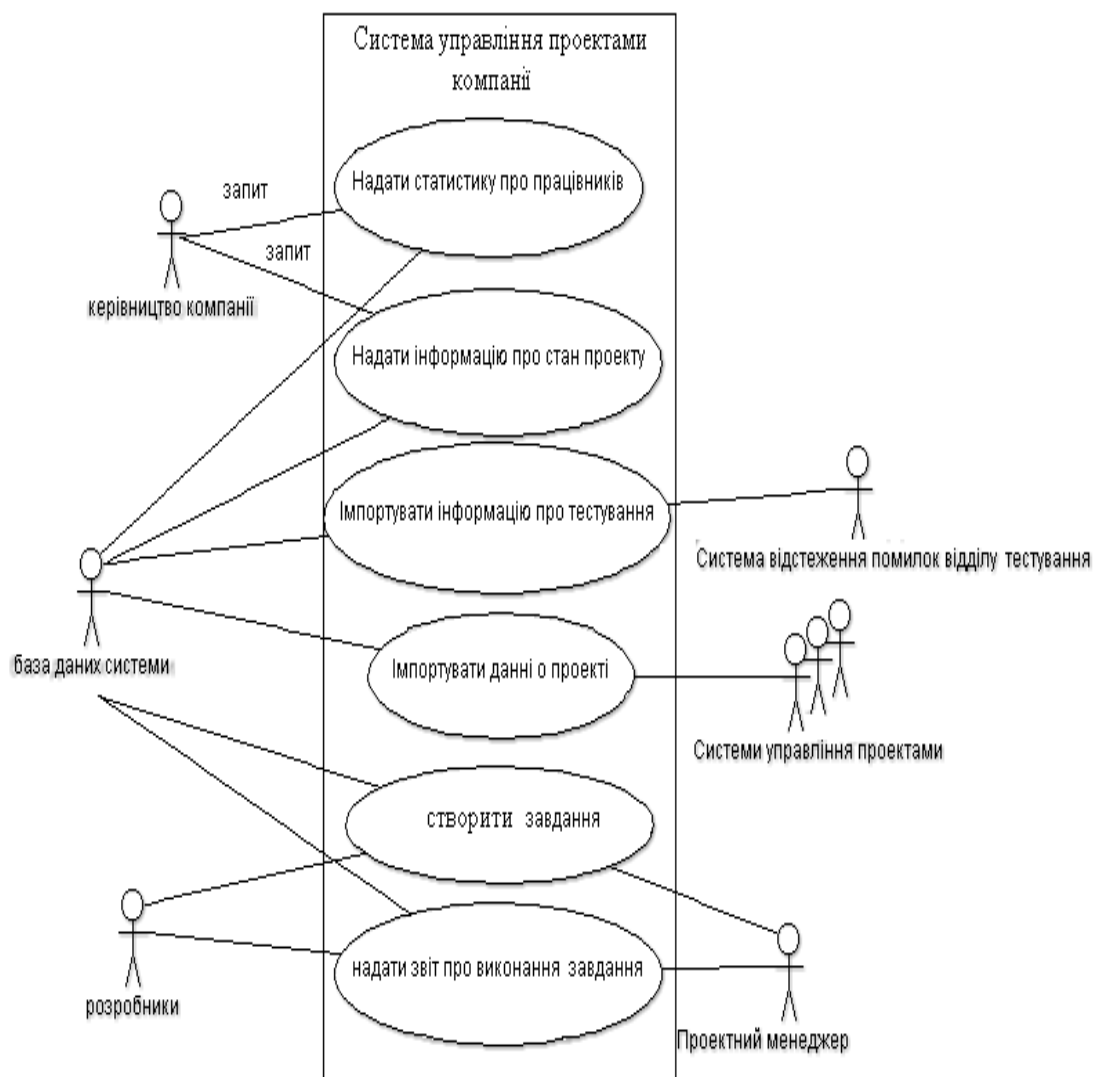


Рисунок 2.1 – Модель прецедентів функціональної структури системи

2.1.3 Опис функцій, що автоматизуються

Автоматизації підлягають функції, усі функції системи які зображені на моделі прецедентів функціональної структури системи:

- надання інформації про стан виконання кожного проекту
- надання статистики про об'єми та якості виконання робіт як виділом працівників так и по окремому з них.
- імпортування даних про проекти, з програмного забезпечення проектних менеджерів
- імпортування даних про тестування проекту з різних систем контролю якості та тестування.
- можливість створення завдання для групі так и для окремих працівників.
- відповідь, та відмітка виконання завдання.

2.1.4 Захист підсистеми від несанкціонованого доступу

Виходячи з того, що узгоджена концепція інформаційної системи передбачає доступ до неї через мережу Інтернет користувачів з різними правами доступу, першочергова увага повинна приділятися питанню захисту системи від несанкціонованого доступу.

Аутентифікація – це процес, у ході якого на підставі пароля, ключа чи якої-небудь іншої інформації користувач підтверджує, що є саме тим, за кого себе видає.

При вході користувача у систему першою справою відбувається його аутентифікація. Якщо введені логін і пароль збігаються з тими, що зберігаються в системі на сервері, то користувач успішно входить у систему, інакше йому відмовляється в доступі. Тут доречно контролювати кількість спроб, щоб уникнути підбора паролів.

Логін і пароль необхідно передавати винятково методом POST. При методі GET дані кешуються проксі-серверами і Web-оглядачами, тобто імовірність вкрасти секретний пароль значно вище.

Крім аутентифікації на HTTP-сервері, необхідно робити аутентифікацію на сервері баз даних. Аутентифікація на сервері баз даних реалізується засобами СУБД.

Авторизація – це процес, у ході якого з'ясовуються права доступу, привілеї, властивості і характеристики користувача на підставі його імені, логіна чи якої-небудь іншої інформації про нього.

Доступ до будь-яких даних у підсистемі відбувається вибіркою даних із джерел даних з використанням логіна. Таким чином, при відсутності логіна неможливо отримати необхідні дані.

Логін і пароль користувача повинні передаватися по захищеному з'єднанню (у зашифрованому виді), що створює HTTP-сервер. Інша інформація є складноінтерпретуємою, тому питання з її захистом не так актуально.

Однак у майбутньому в підсистему можуть бути додані нові дані (і функції), які потрібно захищати. Умовно канали зв'язку в підсистемі можна розділити на кілька видів:

- між користувачами і Web-сервером (захист, що рекомендується: захищене з'єднання з використанням SSL – Secure Sockets Layer – при використанні SSL сервер і Web-оглядач погодять між собою застосовуваний метод шифрування, обмінюючись сертифікатами і ключами і встановлюючи доступне тільки їм захищене з'єднання);
- між Web-сервером і сервером баз даних – захист, що рекомендується: засобами СУБД, яка використовується або SSL.

Необхідно розуміти, що захист каналів зв'язку пов'язаний з додатковими операціями по шифруванню даних, обміну ключами і сертифікатами, тому продуктивність роботи системи може погіршитися.

2.2 Рішення з інформаційного забезпечення.

2.2.1 Опис інформаційного забезпечення системи

Інформаційне забезпечення системи складається з бази даних, у якій зберігається вся інформація про проекти, працівників та інша необхідна інформація.

Крім цього необхідно зберігати наступну інформацію:

- інформація о замовниках
- інформація о завданнях
- інформація о відвідуванні
- інформація о помилках
- інформація о коментарях до помилок

Введення інформації в базу даних системи повинно здійснюватися за допомогою програмного забезпечення системи.

Наступна інформація повинна бути введена в систему до початку її експлуатації:

- інформація про всіх працівників, відділи в яких працюють працівники, за збір цієї інформації відповідає адміністратор системи, якому можуть допомагати працівники відділу кадрів;
- інформація про аутентифікацію; адміністратор повинен ввести інформацію про логіни та паролі працівників; за створення і збереження цієї інформації відповідає адміністратор системи.

Адміністратор системи, як правило, вводить інформацію локально (безпосередньо використовуючи Web-сервер або сервер баз даних), а працівники керівництва компанії – віддалено, використовуючи робочі станції і Web-браузер.

2.2.2 Логічна модель бази даних

Розробимо логічну модель бази даних, спираючись на діаграму класів UML. Так як в можливості цієї діаграми входить і встановлення відношень, то я не бачу необхідності розробляти концептуальну модель даних [9].

Оскільки на діаграмі немає наслідування, діаграма об’єктів системи буде повністю відповідати діаграмі класів.

У таблиці 2.1 приведемо сутності з атрибутами підсистеми, розроблені на основі інформаційного забезпечення підсистеми та з метою побудови логічної моделі.

Логічна модель бази даних системи представлена на рисунку 2.2, зі всіх можливих відношень між класами обрано відношення асоціації.

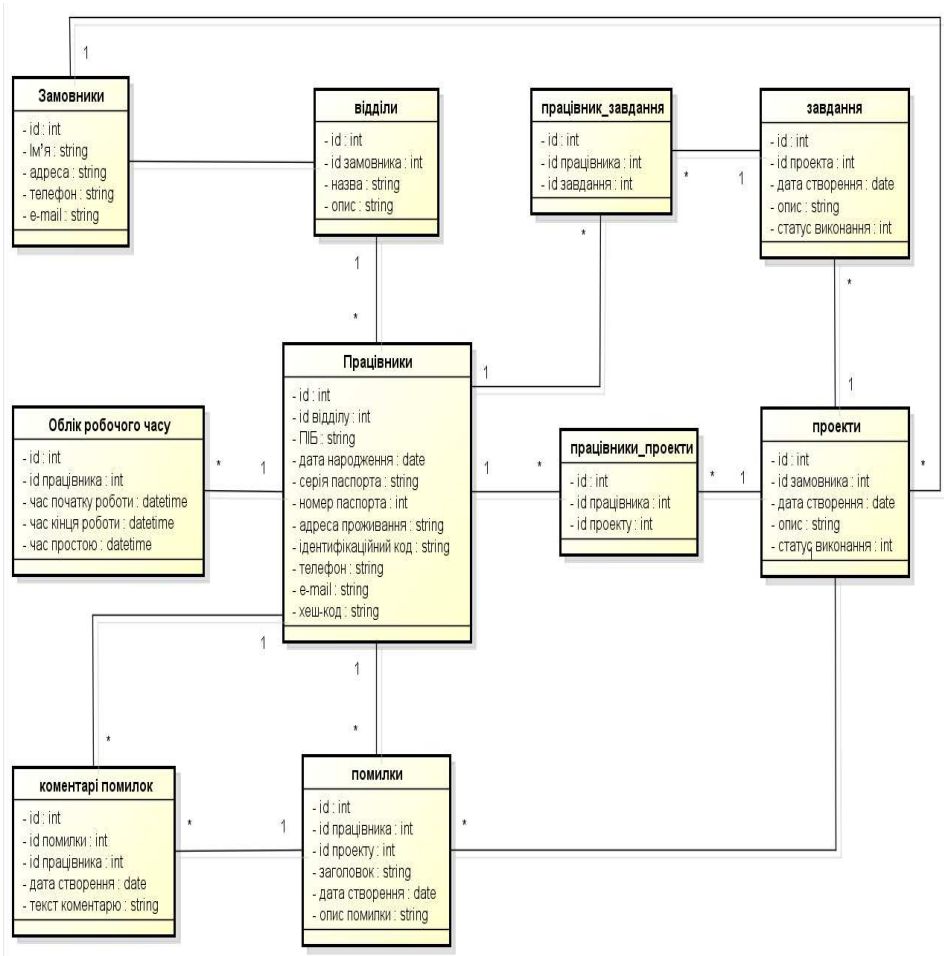


Рисунок 2.2 – Логічна модель структури бази даних системи

Таблиця 2.1 - Сутності та їх атрибути системи управління проектами компанії "GeeksForLess"

№	Сутність	Атрибути сутності
---	----------	-------------------

1	Працівник	— код — П.І.Б. — дата народження — серія паспорта — номер паспорта — адреса проживання — ідентифікаційний код — телефон — e-mail — пароль — хеш-код — роль
2	Замовник	— код — Ім'я . — адреса — телефон — e-mail
3	Відділ	— код — код замовника — назва — опис
4	Помилки	— код — код працівника — код проекту — заголовок — опис помилки — дата створення
5	Коментарі	— код — код помилки — код працівника — дата створення коментарю — текст коментарю
6	Працівник_Завдання	— код — код працівника — код завдання
7	Завдання	— код — код проекту — дата створення — опис — статус виконання — дата виконання
8	Працівник_Проекти	— код — код працівника — код проекту
9	Проекти	— код — код замовника — дата створення — опис — статус виконання
10	Облік робочого часу	— код — код працівника. — час початку роботи — час кінця роботи — час простою

2.3 Рішення з математичного забезпечення

2.3.1 Оцінка надійності системи

Для оцінки надійності підсистеми будемо використовувати принцип визначення показників надійності системи по характеристикам надійності комплектуючих елементів.

Найбільш важливим вузлом системи є, безумовно, сервер з встановленою інтегрованою системою управління проектами. Тому доцільно буде визначити надійність саме серверної станції. Можна виділити її наступні компоненти:

- блок живлення;
- джерело безперебійного живлення;
- комутатор мережі (світч).

Необхідно розрахувати ймовірність безвідмовної роботи кожного з перерахованих компонентів серверної станції, потім обчислити надійність системи в цілому.

Середній час роботи даних пристроїв візьмемо з їх технічної документації.

Середній час роботи блоку живлення: $T_{дж.ср} = 15000$ годин.

Середній час роботи джерела безперебійного живлення: $T_{дбж.ср} = 15000$ годин.

Середній час роботи комутатора: $T_{к.ср} = 18000$ годин.

Розрахунок проводимо для визначення ймовірності безвідмовної роботи кожного з перерахованих елементів протягом 1000 годин за формулою 2.1:

$$P(t) = \exp\left[-\int_0^t \lambda(t) dt\right] = \exp(-\lambda_0 t) \quad (2.1)$$

,де t – час, протягом якого розраховується надійність;

λ_0 – інтенсивність відмов елемента ІУС протягом середнього часу безвідмовної роботи системи.

Частота відмов елемента ІУС за середній час безвідмовної роботи $T_{ср}$ розраховується по формулі 2.2:

$$\lambda_0 = \frac{1}{T_{ср}} \quad (2.2)$$

Зробимо розрахунки:

Частота відмов для блоку живлення:

$$\lambda_0 = 1/15\ 000 = 0,000066$$

$$P(t)_{дж} = \exp(-0,000066 * 1000) = 0,936$$

Частота відмов для джерела безперебійного живлення:

$$\lambda_0 = 1/15\ 000 = 0,000066$$

$$P(t)_{дбж} = \exp(-0,000066 * 1000) = 0,943$$

Частота відмов для комутатора:

$$\lambda_0 = 1/18\ 000 = 0,000055$$

$$P(t)_к = \exp(-0,000055 * 1000) = 0,961$$

Введемо два допущення: випадки відмов елементів не залежать один від одного, і відмова елемента приводить до відмови ІУС. Дані допущення прийняті на основі практичного досвіду роботи серверу білінгової системи.

Прийняті допущення дозволяють застосувати для підрахунку ймовірності безвідмовної роботи ІУС теорему добутку ймовірностей, що представлена формулою 2.3:

$$P_{uc}(t) = \prod_{i=1}^n P_i(t) \quad (2.3)$$

де $P_i(t)$ – ймовірність безвідмовної роботи i -го елемента;

n - кількість елементів.

Надійність проекрованої підсистеми обліку і управління послугами ЦТБ протягом 1000 годин безперервної роботи визначається добутком ймовірностей безвідмовної роботи блоку живлення, джерела безперебійного живлення та концентратора.

$$P(t)_{ис} = P(t)_{дж} * P(t)_{дбж} * P(t)_к = 0,936 * 0,943 * 0,961 = 0,896$$

Виходячи з проведених вище розрахунків робимо висновок, що ймовірність безвідмовної роботи інтегрованої системи управління проектами протягом 1000 годин становить близько 90%.

2.3.2 Огляд алгоритму шифрування

Одним з важливих параметрів при розробці системи – є стійкість системи до несанкціонованого доступу до даних. Одним з важливих пунктів безпеки системи є шифрування пароллю користувача. На даний час у світі

розроблено багато алгоритмів шифрування, а для вибору алгоритму необхідного для дано розробки потрібно чітко розібратися в цих алгоритмах, і вибрати саме той, що буде найбільш ефективним та простим у використанні. Нижче представлено огляд основних алгоритмів шифрування.

2.3.2.1 Шифрування з закритим ключем

Шифрування з закритим ключем засноване на тому, що доступ до ключа має тільки авторизований персонал. Цей ключ повинен триматися в секреті. Як показано на рисунку 2.11 відправник і отримувач мають один і той ключ.

Найбільш широко використовується алгоритм з закритим ключем DES. Він розроблений компанією IBM в 70-х роках, прийнятий як американський стандарт для комерційних і державних несекретних комунікацій. Сучасні швидкості розрахунку на порядок перевищують швидкість розрахунку в 70-х роках, тому алгоритм DES вважається застарівши як мінімум з 1998 року.



Рисунок 2.3 - Процес шифрування за закритим ключем

Інші відомі алгоритми з закритим ключем – RC2, RC4, RC5Б потрійний DES та IDEA. Потрійний DES-алгоритм досить захищений. Цей алгоритм використовує той же метод шифрування, що і DES, але тричі, використовуючи при цьому до трьох різних ключів. Простий текст шифрується за допомогою першого ключа, дешифрується за допомогою другого, а потім шифрується за допомогою третього. Явний недолік алгоритмів з закритим ключем у тому, що для відправлення кому-небудь захищеного повідомлення необхідно мати захищений спосіб передачі цієї особі закритого ключа.

Прорив відбувся в 1976 році, коли Дифі Хелман опублікував перший алгоритм з відкритим ключем.

2.3.2.2 Шифрування з відкритим ключем

Шифрування з відкритим ключем базується на двох різних ключах – відкритому та закритому. На рисунку 2.12 показано як відкритий ключ використовується для шифрування повідомлення, а закритий для дешифровки.

Перевага цього алгоритму – відкритий ключ. Будь-яка людина, якій ви дали відкритий ключ може відправляти вам захищене повідомлення.

2.3.2.3 Цифрові підписи

Цифрові підписи відносяться до алгоритмів з відкритим ключем, але зі зміненими ролями відкритого та закритого ключа. Коли повідомлення відправлене, то отримувач може дешифрувати його за допомогою відкритого ключа відправника.

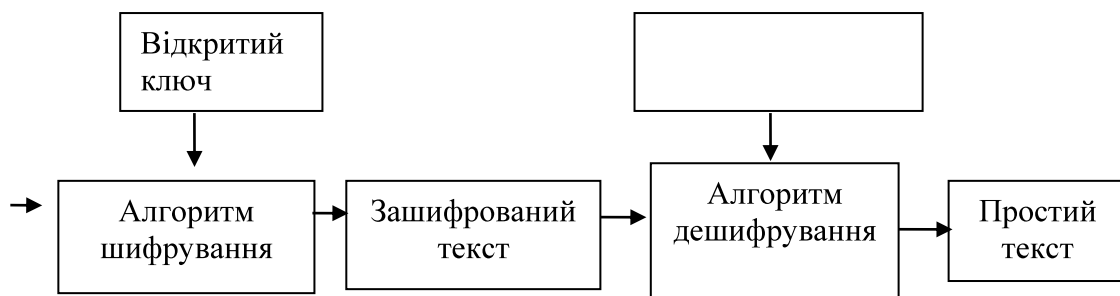


Рисунок 2.4 - Шифрування з відкритим ключем

Цифрові підписи можуть бути дуже корисними. Вони гарантують отримувачу, що повідомлення не було підроблено, а також не дозволять відправнику відмовитись від зобов'язань. Хоча повідомлення і шифрується. Але його можливо причитати тому, хто має відкритий ключ.

Оскільки алгоритми з відкритим ключем досить повільно працюють на великих повідомленнях, для підвищення працездатності використовують алгоритми іншого типу – хеш-функції.

Хеш-функція розраховує дайджест повідомлення, або хеш-значення, для кожного повідомлення. Зовсім не важливо, яке повідомлення генерує алгоритм. Важливо, що результат роботи хеш-функції дає детермінований

результат – результат буде один і той самий, якщо будуть одні й ті ж самі дані. Окрім того результат має невеликий розмір і алгоритм швидко працює.

Найбільш відомі цифрові підписи – MD5 та SHA.

Хеш-функція генерує дайджест, що відповідає відповідному повідомленню. Маючи повідомлення і його дайджест, можливо впевнитись, чи не підроблювалось повідомлення, але тільки в тому випадку, якщо дайджест не був підроблений разом з повідомленням.

І звичайний спосіб створення цифрового підпису – створення за допомогою швидкої хеш-функції дайджеста для свого повідомлення, а потім шифрування тільки короткого дайджесту з використанням повільного алгоритму з відкритим ключем.

2.3.2.4 Алгоритм шифрування SHA

Алгоритм SHA-1 розроблений Агентством національної безпеки США в 1995 році й включений Національним інститутом стандартів США в стандарт SHS. Хеш-функції використовуються в інформаційних технологіях, у системах електронного підпису, системах контролю цілісності програмного коду й даних, а також для побудови кодів аутентифікації.

SHA-1 використовується в переважній масі продуктів криптографічного захисту, вироблених і використовуваних у США, а також в інших країнах, які підтримують американські стандарти. Так, в операційних системах сімейства Windows вбудований ряд криптографічних служб, що використовують алгоритм SHA-1. Впровадження цього алгоритму в Windows настільки глибоко, що завантаження й використання операційних систем без звертання до нього зі штатних криптографічних сервісів неможливі. Стандарт SHS є алгоритмом SHA-1 використовується разом з іншими криптографічними стандартами США в інфраструктурі відкритих ключів, а також в інших розповсюджених криптографічних протоколах (SSL/TLS, IPSec, Kerberos і ін.).

Разом з тим SHA-1 не використовується в продуктах, що підтримують національні криптографічні стандарти. Зокрема, даний алгоритм не використовується за своїм прямим призначенням в російських

інформаційних системах, що оснащені сертифікованими засобами криптографічного захисту.

Побудова й використання колізій алгоритму хешування розкладається на два завдання:

- криптографічне завдання знаходження колізії;
- практичне завдання використання знайденої колізії для атаки на систему, що захищає алгоритмом хешування.

Тому, знайдені підходи до побудови колізій є значним досягненням в області криптографічного аналізу SHA-1. Питання про конкретне використання одержуваних колізій для реальних атак залишається відкритим. Необхідно враховувати, що для успішного проведення цілеспрямованої атаки колізії повинні бути семантично припустимі. Тому погроза легітимності підпису на базі хешування по алгоритму SHA-1 носить випадковий характер і вимагає оцінки ймовірності досягнення позитивного результату. Атаки в системах аутентифікації, засновані на побудові колізій алгоритму, неможливі через застосування ключового хешування.

Хеш-функція реалізує відображення безлічі хешуємих текстів на безліч значень хеш-функцій. Хешуємі тексті розбивається на класи еквівалентності за значеннями хеш-функції. Складність знаходження колізій хеш-функції залежить від аналітичної складності цього відображення.

Математичною моделлю завдання про оцінку складності числа колізій є завдання, відоме як «парадокс близнюків»: у множині з елементів, кожний з яких характеризується незалежно від інших одним з ознак при його випадковому й рівноймовірнісному виборі, середнє число елементів з ознакою, однакою для фіксованого елемента, дорівнює n/m , середнє число пар елементів з однаковою ознакою дорівнює $n/m-2$.

У застосуванні до завдання про колізії середнє число колізій у першому випадку дорівнює $|B|/|H|$ у другому $|B|/(|H|)-2$. За теоретичну оцінку складності знаходження колізій у криптографії приймається величина.

Хеш-функція в сучасних комп'ютерах реалізується ітерацією математичних операцій над проміжними значеннями обчислень. Для побудови криптографічних методів визначення колізій істотними є

аналітична складність значення хеш-функції від прообразу, а також спосіб організації поблочного перетворення хешуємого тексту, включаючи спосіб доповнення останнього неповного блоку до повного.

До можливості зниження криптографічної складності на три-чотири порядки привів ряд особливостей алгоритму SHA-1. Так, використовувані математичні операції (операція додавання, побітові логічні операції, операція зміщення) у своїй сукупності не дозволяють будувати перетворення з гарними перемішуючими властивостями. Крім того, застосовується послідовне хешування тексту блоками по 512 біт зі згортанням блоку до 160 біт. Використовуваний спосіб доповнення останнього неповного блоку хешуємого тексту до повного створює додаткові можливості для пошуку колізій.

Зазначені особливості характерні для алгоритмів MD4, MD5, SHA-256, SHA-512 і інших з тією же ідеологією побудови. Вони дозволяють застосовувати до алгоритмів цього класу методи диференціального аналізу для побудови колізій.

Для алгоритму SHA-1 теоретична стійкість дорівнює, однак на практиці стійкість знижується.

Алгоритм SHA-1 з його значенням споконвічно не мав криптографічного запасу й був розрахований на те, що оцінку знизити не вдасться.

На відміну від алгоритму SHA-1 алгоритм ДЕРЖСТАНДАРТ Р 34.11-94 характеризується наступними властивостями:

- перетворення блоку інформації при хешуванні здійснюється використанням перетворення, що шифрує, по алгоритму ДЕРЖСТАНДАРТ 28147-89;
- перетворення хешуємої інформації виробляється блоками по 256 біт з розміром перетвореного блоку також 256 біт; теоретична стійкість;
- в алгоритмі хешування використовуються спеціальні міри, що виключають можливість спрощення завдання пошуку колізій при неповному останньому блоці.

У перетворенні, що шифрує, по алгоритму ДЕРЖСТАНДАРТ 28147-89 на додаток до операцій, характерним для алгоритму SHA-1 і властивих йому, використається вузол замін, вибором яких вдається ускладнити перетворення, що переміщує, і істотно обмежити можливість застосування до алгоритму ДЕРЖСТАНДАРТ Р 34.11-94 методів диференціального аналізу для рішення завдання з пошуку колізій. Дотепер такі методи в процесі багаторічних криптографічних досліджень алгоритму ДЕРЖСТАНДАРТ Р 34.11-94 не знайдені. На нашу думку, методи побудови колізій, знайдені для алгоритму SHA-1, незастосовні до алгоритму ДЕРЖСТАНДАРТ Р 34.11-94.

Алгоритм SHA-1 (Secure Hash Algorithm) використовується в алгоритмах електронно-цифрових підписів (DSA - Digital Signature Algorithm). Для повідомлення (потoku даних) довжиною менше 2^{64} біти, даний алгоритм обчислює 160-бітний хеш. Отриманий хеш використовується для генерації "підпису" до повідомлення, а також для її перевірки. Будь-які зміни у вихідному повідомленні приведуть до зміни хешу з дуже великою ймовірністю.

SHA-1 розроблявся з метою мати наступні властивості: немає можливості з використанням комп'ютера знайти вихідне повідомлення, знаючи хеш, або знайти два різних повідомлення, що дають однаковий хеш.

У даному алгоритмі під "словом" розуміється кількість інформації в 32 біта, "байт" - 8 біт. Ціле число в проміжку від 0 до $2^{32} - 1$ включно може бути представлене як слово, у якому старший байт записаний першим, молодший - останнім. Ціле z ($0 \leq z < 2^{64}$) представляється у вигляді $z = (2^{32})x + y$, де x і y - цілі в проміжку від 0 до $2^{32} - 1$. Таким чином, z можна представити парою слів x і y . Блок з 512 біт може бути представлений послідовністю з 16 слів.

В алгоритмі використовуються наступні операції:

- $x \& y$
 - побітове "І"
- $x | y$
 - побітове "АБО"
- $x \wedge y$
 - побітове " що виключає АБО"

- $\sim x$

- побітове заперечення

$x \lll y$

- - циклічне зміщення x вліво на y біт

$x + y$

$= (x + y) \bmod 2^{32}$, тобто розглядаються молодші 32 біта результат

Вирівнювання повідомлення.

Довжина повідомлення приймається як кількість біт у вхідному потоці (можливо порожнє повідомлення довжиною 0 біт). Ціль вирівнювання: зробити довжину повідомлення кратної 512, тому що алгоритм SHA-1 обробляє вхідний потік блоками по 512 біт. Вирівнювання проходить у такий спосіб: додається один біт '1' (a), потім впливають біти '0' (b), після чого записується довжина повідомлення до вирівнювання (c). У результаті одержуємо довжину потоку рівну $n \cdot 512$.

Логічні функції

SHA-1 використовує послідовність логічних функцій $f(0), f(1), \dots, f(79)$, кожна з яких приймає 3 32-бітні слова й обчислює результат - слово. Функція $f(t; B, C, D)$ визначена наступним виглядом:

```
f(t; B, C, D) = (B & C) | (~B & D)
0 <= t <= 19
f(t; B, C, D) = B ^ C ^ D
20 <= t <= 39
f(t; B, C, D) = (B & C) | (B & D) | (C & D)
40 <= t <= 59
f(t; B, C, D) = B ^ C ^ D
60 <= t <= 79
```

Константи

У даному алгоритмі використовується послідовність константних слів $K(0)$,

$K(1), \dots, K(79)$:

$K(t) = 5A827999$

$0 \leq t \leq 19$

$K(t) = 6ED9EBA1$

$20 \leq t \leq 39$

$K(t) = 8F1BBCDC$

$40 \leq t \leq 59$

$K(t) = CA62C1D6$

$60 \leq t \leq 79$

Обчислення

Для обчислення SHA-1 хеш функції використовується буфер з 5 слів $\{H_0, H_1,$

H2, H3, H4}, що перед початком обчислень ініціалізується наступними значеннями:

H0 = 67452301
H1 = EFCDA89
H2 = 98BADCFE
H3 = 10325476
H4 = C3D2E1F0

2.3.2.5 Алгоритм шифрування MD5

Цей алгоритм був розроблений в 1991 році професором Рональдом Л. Рівестом. Алгоритм MD5 часто називають алгоритмом шифрування, але насправді це твердження помилково. Головною перевагою MD5 є те, що зашифровані дані відновити не можна. А що це за алгоритм шифрування, що шифрує дані так, що ніхто їх не зможе довідатися? MD5 - це хеш-функція.

Це і є проста хеш-функція. Якщо ми знаємо параметр функції, то однозначно можемо сказати, який буде результат. А якщо нам відомий результат, то чи можемо ми однозначно довідатися про параметр? Звичайно, немає. Для числа 234 параметр може бути 234,1234, 2234,3234... Тому однозначно відновити параметр не вийде.

Тепер подивимося, як саме працює MD5. Для обробки MD5 одержує деякий рядок. Цей рядок перетвориться в послідовність із нулів і одиниць. Як це робиться? У кожного символу є свій номер. Ці номери можна записати у двійковій системі числення. Виходить, кожний символ можна записати як послідовність нулів і одиниць. Якщо цим скористатися, одержимо рядок з послідовністю нулів і одиниць. Нехай q буде довжина послідовності, що вийшла (рівно 64 біта, можливо, з незначними нулями). До послідовності, що вийшла, приписується 1. У результаті довжина послідовності збільшується на 1. Потім до послідовності приписуються нулі, поки довжина не стане по модулю 512 дорівнює 448 ($\text{length mod } 512 = 448$). Далі до послідовності дописуються молодші 32 біта числа q , а потім - старші. Довжина послідовності стає кратною 512. Отриману послідовність назовемо S . Для підрахунку результату використовуються чотири подвійних слова (32 біта). Ці подвійні слова ініціалізуються наступними шістнадцятирічними значеннями, де першим треба самий молодший байт:

A: 01 23 45 67

B: 89 ab cd ef

C: fe dc ba 98

D: 76 54 32 10

Також для підрахунку результату використовуються наступні функції:

$F(X, Y, Z) = XY \vee \text{not}(X) Z$

$G(X, Y, Z) = XZ \vee Y \text{not}(Z)$

$H(X, Y, Z) = X \text{ xor } Y \text{ xor } Z$

$I(X, Y, Z) = Y \text{ xor } (X \vee \text{not}(Z))$

X, Y, Z - це подвійні слова. Результати функцій, також подвійні слова.

Для підрахунку використовується ще одна функція (назвемо її W). Вона хитро обробляє дані й повертає результат (докладно її описувати не буду, тому що вона виконує серію простих перетворень). Обробка даних відбувається з використанням функцій F, G, H, I.

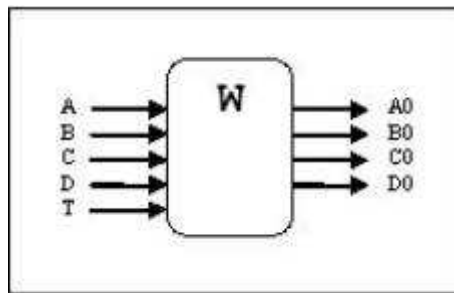


Рисунок 2.5 - Схематичне відображення функції W

На рисунку 2.5 схематично зображена функція. Ліворуч - вхідні дані, праворуч - вихідні.

Всі необхідні функції й позначення розглянуті. Тепер розглянемо, як відбувається прорахунок результату:

1. запам'ятовуємо перші 512 біт послідовності S;
2. видаляємо перші 512 біт послідовності S (можна обійтися й без видалення, але тоді на першому кроці треба брати не перші 512, а наступний 512 біт);
3. викликаємо функцію W. Параметри A, B, C, D - це поточні значення відповідних подвійних слів. Параметр T - це 512 біт, що були збережені;
4. додаємо до A A0;
5. $B = B + B0$;
6. $C = C + C0$;
7. $D = D + D0$;
8. якщо довжина послідовності 0, виходимо;

9. переходимо до кроку 1.

Тепер перейдемо до основного циклу алгоритму. Цей цикл продовжується, поки не вичерпаються 512-бітові блоки повідомлення.

Чотири змінні копіюються у інші змінні: A в a , B в b , C в c , D в d .

Головний цикл складається із чотирьох дуже схожих етапів (у MD4 було три). На кожному етапі 16 разів виконуються різні операції. Кожна операція являє собою нелінійну функцію над трьома з a , b , c та d . Потім вона додає цей результат до четвертої змінної, підблоку тексту та константи. Далі результат циклічно зсувається вправо на змінне число бітів та додає результат до однієї із змінних a , b , c та d . Нарешті результат замінює одну із змінних a , b , c та d . Існує чотири нелінійних функції, які використовуються по одній у кожній операції (для кожного етапу – інша функція).

На рисунку 2.7 зображено головний цикл MD5, а на рисунку 2.15 – одна операція MD5.

$$F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

($\oplus$ – це XOR, $\wedge$ – AND, $\vee$ – OR, $\neg$ – NOT).

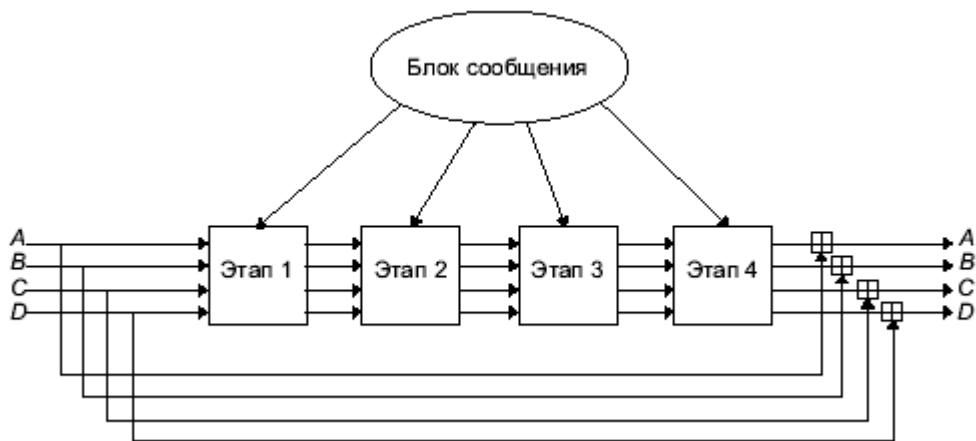


Рисунок 2.6 – Головний цикл алгоритму шифрування MD5

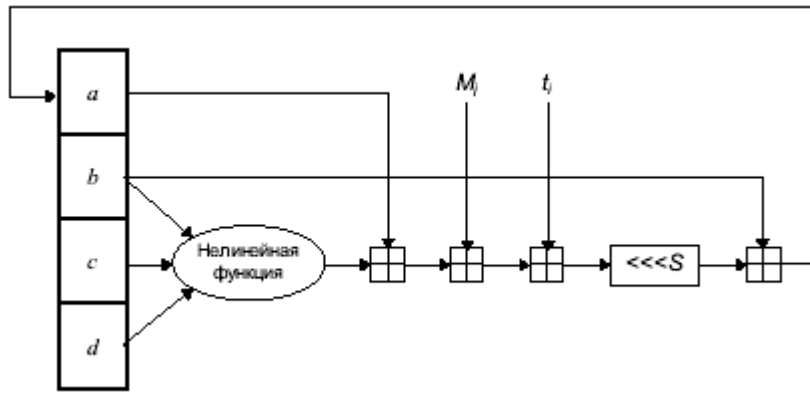


Рисунок 2.7 – Одна операція алгоритму шифрування MD5

Ці функції спроектовані так, щоб, якщо відповідні біти X , Y та Z незалежні та незміщені, кожен біт результату також був би незалежним та зміщеним. Функція F – це побітова умова: якщо X , то Y , інакше Z . Функція H – це побітова операція парності.

Якщо M_j має значення j -го підблоку повідомлення (від 0 до 15), а $\lll s$ позначає циклічний зсув на s бітів, то використовуються наступні чотири операції:

Після всього цього a , b , c та d додаються до A , B , C та D , відповідно, і алгоритм продовжується для наступного блоку даних. Кінцевим результатом буде об'єднання A , B , C та D (його довжина буде 128 біт). Часто можна бачити результат MD5 як послідовність із 32 символів 0..f. Це те ж саме, тільки результат записаний не у двійковій системі числення, а в шістнадцятиричній.

Злом MD5

В Інтернеті можна знайти багато програм, які обіцяють знайти рядок, для якої алгоритм MD5 видасть заданий результат. Ці програми дійсно працюють. Раніше відзначалося, що відновити параметр неможливо. Як же працюють ці програми? Вони перебирають всі можливі рядки, застосовують до них алгоритм MD5, а потім порівнюють із зразком. Якщо значення збіглися, це означає, що програма знайшла необхідний рядок. Але в цих програм є маленький недолік. Припустимо, відомо, що програмі потрібно перебрати всі слова довжиною в 8 символів, що складаються з маленьких і великих латинських букв. Скільки часу це займе? Скільки всього таких слів?

На першому місці може стояти кожний з $26 \cdot 2 = 52$ символів. На 2, 3, 4, 5, 6, 7 і 8 - теж 52. Виходить, усього таких слів буде:

$$52 \cdot 52 \cdot 52 \cdot 52 \cdot 52 \cdot 52 \cdot 52 \cdot 52 = 52^8 = 53 \cdot 1012.$$

А якщо використовуються не тільки латинські букви? То це ще більше. Перебір всіх варіантів на звичайному персональному комп'ютері займе дуже багато часу. В Інтернеті можна знайти сайти, які по уведеному хешу видають рядок, для якої буде точно такий хеш. Ці сайти використовують базу даних із заздалегідь переліченими хешами. Але в базах зберігаються не всі хеші, а тільки найбільш використовувані. Отже єдина рекомендація часто змінювати пароль притому число символів повинно бути не менше 8 (зі зміною регістра, мови та цифрами).

2.3.3 Вибір алгоритму

MD5 – це покращена версія MD4. Хоча вона складніше за MD4, їх схеми схожі, і результатом MD5 також є 128-бітове хеш-значення.

Після деякої первісної обробки MD5 обробляє вхідний текст 512-бітовими блоками, розбитими на 16 32-бітових підблоків. Виходом алгоритму є набір з чотирьох 32-бітових блоків які об'єднуються в єдине 128-бітове хеш-значення.

Особливістю алгоритму є те, що повідомлення доповнюється так, щоб його довжина була на 64 біта коротше числа, яке кратне 512. Головний цикл складається із чотирьох дуже схожих етапів (у MD4 було три).

Отже Рон Рівест здійснив наступні покращення MD5 у порівнянні з MD4:

1. Додав четвертий етап.
2. Тепер у кожній дії використовується унікальна константа, що додається.
3. Функція G на етапі 2 з $((X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z))$ була змінена на $(X \wedge Z) \vee (Y \wedge (\neg Z))$, щоб зробити G менш симетричною.
4. Тепер кожна дія одається до результату попереднього етапу. Це забезпечує більш швидкий лавинний ефект.

5. Змінився порядок, в якому використовувались підблоки повідомлення на етапах 2 та 3, щоб зробити шаблони менш подібними.

6. Значення циклічного зсуву ліворуч на кожному етапі були наближено оптимізовані для прискорення лавинного ефекту. Чотири зсуви, які використовувались на кожному етапі, відрізняються від значень, які використовувались на інших етапах.

Беручи до увагу вищевикладені факти та широку поширеність MD5 у практичних застосуваннях було обрано саме цей алгоритм для шифрування паролів користувачів.

В роботі подано алгоритм створення нового завдання для працівника. Після авторизації користувача в системі буде представлено список всіх працівників відділу. Обравши працівника або групу працівників випадального списку може обрати проект до якого створюються завдання. Крім цього можливо обрати пріоритет завдання та орієнтований час на виконання завдання. Заповнює поля з описом завдання та того що потрібно зробити.

Після занесення відповідної інформації натиснути *Сохранить*. Іде перевірка на відповідність заповнення в головні форми інформації. Якщо все вірно - внесена інформація потрапляє до бази даних, інакше – повідомлення про помилку. Після цього працівникам для яких створено завдання надсилається повідомлення. про те що для них створено нове завдання.

Виконавши завдання працівник робить свою помітку о виконанні, та зберігає змінення, а працівнику(менеджеру) буде надіслано повідомлення о виконанні завдання.

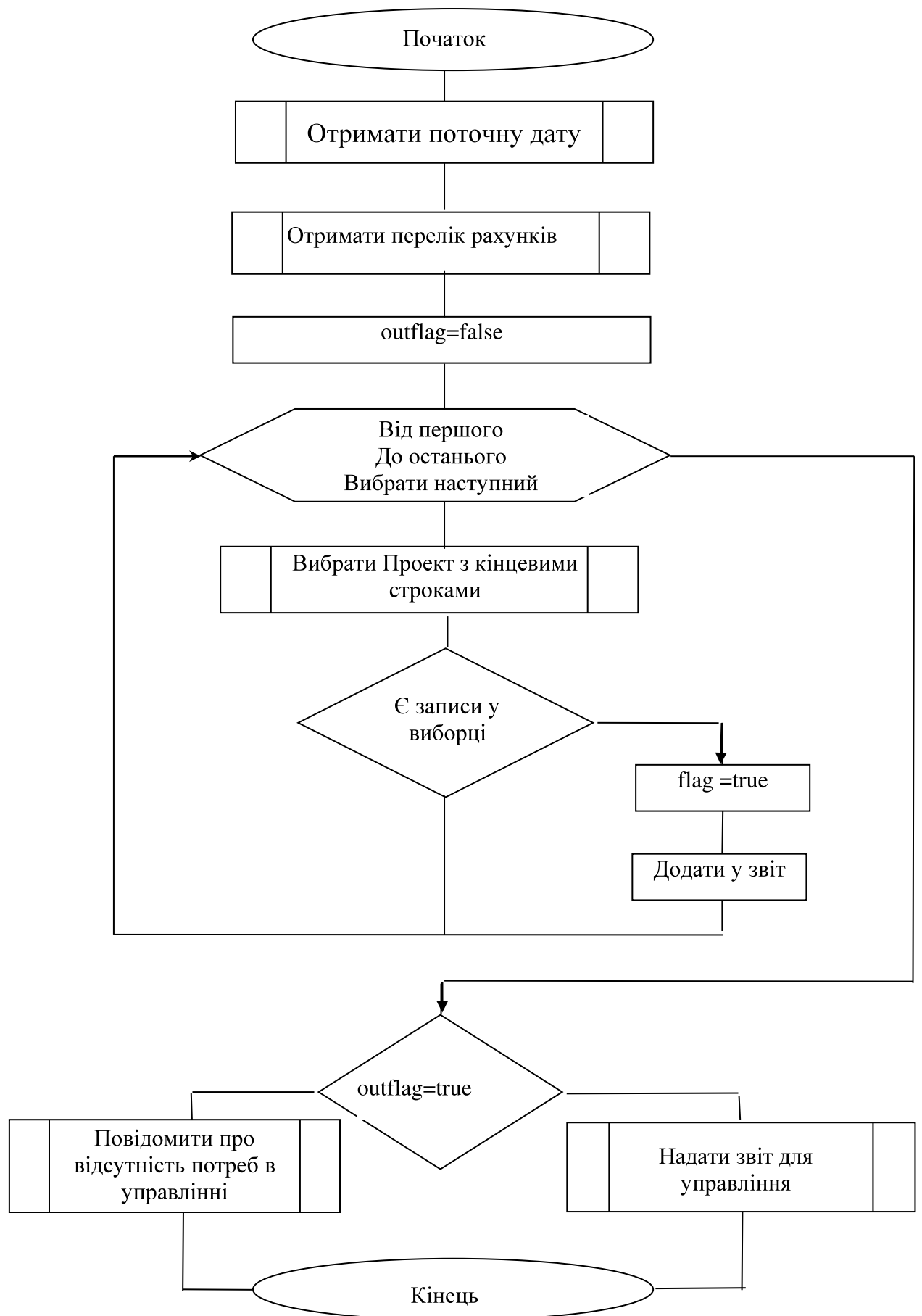


Рисунок 2.8 — Блок-схема управління проектами

Одним з керуючих алгоритмів є алгоритм управління проектами. Цей алгоритм повинен видавати інформацію для керування для уповноваженої

особи. Уповноваженою особою на даному підприємстві є менеджер відділу. Також до цієї функції має доступ безпосередньо керівник усього підприємства.

Суть даної функції у моніторингу діяльності у напрямку маркетингу, а саме звіт по строкам виконання проектів. В результаті своєї роботи повноважена особа отримує список невиконаних у строк задач, та тих задач, строк виконання яких близиться до кінця. За наявності даного звіту керівник може власним рішенням впливати на хід розвитку проекту.

2.4 Рішення з програмного забезпечення

2.4.1 Фізична модель даних

Фізична модель даних будується методом денормалізації та типізації логічної моделі даних. Розробимо фізичну структуру кожної таблиці, що входить у базу даних. Типи даних будемо вказувати для СУБД MySQL 5.4

При описі фізичних таблиць бази даних використані стандартні позначення ключів:

При описі фізичних таблиць бази даних використані стандартні позначення ключів:

- РК - первинний ключ;
- ALT - альтернативний ключ;
- FK - зовнішній ключ.

Якщо атрибут може приймати пусте значення, то він позначається як NULL. Таким чином, визначається детермінованість атрибутів.

Усі таблиці бази даних знаходяться в 1 нормальній формі (НФ), оскільки вони є двовимірними, не містять полів, що включають кілька значень, і порожніх ключових полів (РК).

Усі таблиці бази даних знаходяться в 2 НФ, тому що всі їхні поля, що не входять у первинний ключ, зв'язані повною функціональною залежністю з первинним ключем.

Усі таблиці бази даних знаходяться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Фізична структура таблиць, які входять до БД системи управління

проектами компанії «Хостінг Макс» представлена в таблицях 2.2-2.11.

Таблиця 2.2 – Структура таблиці «Працівники»/«workers»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код відділу	department_id	int(11)	FK
П.І.Б.	name	varchar(100)	INDEX
Дата народження	birthday	timestamp	
Серія паспорта	passport_serial	varchar(2)	
Номер паспорта	passport_id	int(4)	INDEX
Місце проживання	address	varchar(256)	
Ідентифікаційний код	icode	longint	INDEX
Е-mail	email	varchar(30)	INDEX
Телефон	phone	varhar(20)	INDEX
Роль	role	tinyint	

Таблиця 2.3 – Структура таблиці «Відділи»/«Departments»

Назва поля	Назва поля	Тип даних	Ключ
Код	id	int(11)	PK
Код замовника	customer_id	int(11)	FK
Назва	name	varchar(30)	
опис	descriptions	text	

Таблиця 2.4 – Структура таблиці «Замовники»/«Customers»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Адреса	address	varchar(256)	
Ім'я	name	varchar(256)	INDEX
E-mail	email	varchar(30)	INDEX
Телефон	phone	varhar(20)	INDEX

Таблиця 2.5 – Структура таблиці «Працівник_завдання»/«worker_tasks»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код працівника	worker_id	int(11)	FK
Код завдання	task_id	int(11)	FK

Таблиця 2.6 – Структура таблиці «Завдання»/ «tasks»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код проекту	project_id	int(11)	FK
Дата створення	start_date	datetime	
Дата виконання	end_date	datetime	
Опис	description	text	
Статус виконання	work_status	tinyint(1)	

Таблиця 2.7 – Структура таблиці «Працівник_проекти»/«worker_projects»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код працівника	worker_id	int(11)	FK
Код проекту	project_id	int(11)	FK

Таблиця 2.8 – Структура таблиці «Проекти»/ «projects»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код замовника	customer_id	int(11)	FK
Дата створення	start_date	datetime	
Дата виконання	end_date	datetime	
Опис	description	text	
Статус виконання	work_status	tinyint(1)	

Таблиця 2.9 – Структура таблиці «Помилки»/ «bugs»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код замовника	project_id	int(11)	FK
Код працівника	worker_id	int(11)	FK
Дата створення	end_date	datetime	
Опис	description	text	
Заголовок	title	varchar(255)	

Таблиця 2.10 – Структура таблиці «Облік робочого часу»/«Attendances»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код працівника	worker_id	int(11)	FK
Код проекту	project_id	int(11)	FK

Таблиця 2.11 – Структура таблиці «Коментарі»/ «comments»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	int(11)	PK
Код помилки	project_id	int(11)	FK
Код працівника	worker_id	int(11)	FK
Дата створення	end_date	datetime	
Текст коментарю	body	text	

2.4.2 Модель послідовностей

Діаграма послідовності відображає взаємодію об'єктів, упорядкованих за часом. На ній відображаються об'єкти, які використовуються в сценарії, та послідовність повідомлень, якими обмінюються об'єкти, для виконання сценарія. Діаграма послідовності дій відповідає реалізаціям прецедентів в логічному представленні системи.

Кожна з діаграм послідовностей відповідає варіанту використання системи. Побудуємо діаграми послідовностей (рисунки 2.9-1.10) для варіантів використання системи, представлених у моделі прецедентів (Рисунок 2.1).

2.4.3 Модель переходів станів

Головне призначення цієї діаграми - описати можливі послідовності станів і переходів, які в сукупності характеризують поведінку елемента моделі протягом його життєвого циклу. Діаграма станів представляє динамічну поведінку сутностей, на основі специфікації їх реакції на сприйняття деяких конкретних подій. Системи, які реагують на зовнішні дії від інших систем або від користувачів, іноді називають реактивними. Якщо такі дії ініціюються в довільні випадкові моменти часу, то говорять про асинхронному поведінці моделі.

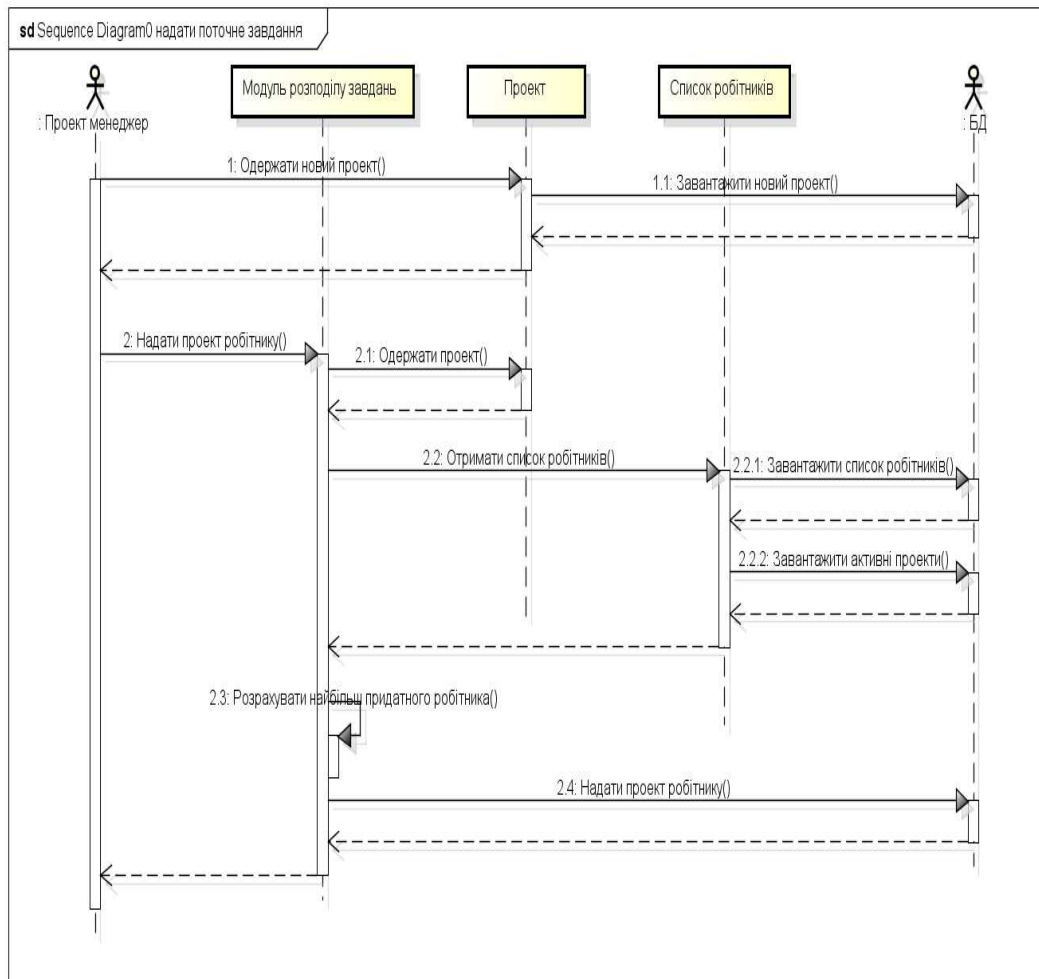


Рисунок 2.9– Діаграма послідовності «Створення завдання»

Хоча діаграми станів найчастіше використовуються для опису поведінки окремих екземплярів класів (об'єктів), але вони також можуть бути застосовані для специфікації функціональності інших компонентів моделей, таких як варіанти використання, актори, підсистеми, операції і методи.

Побудуємо діаграму переходів стану для модуля завдань, яка зображена на рисунку 2.11.

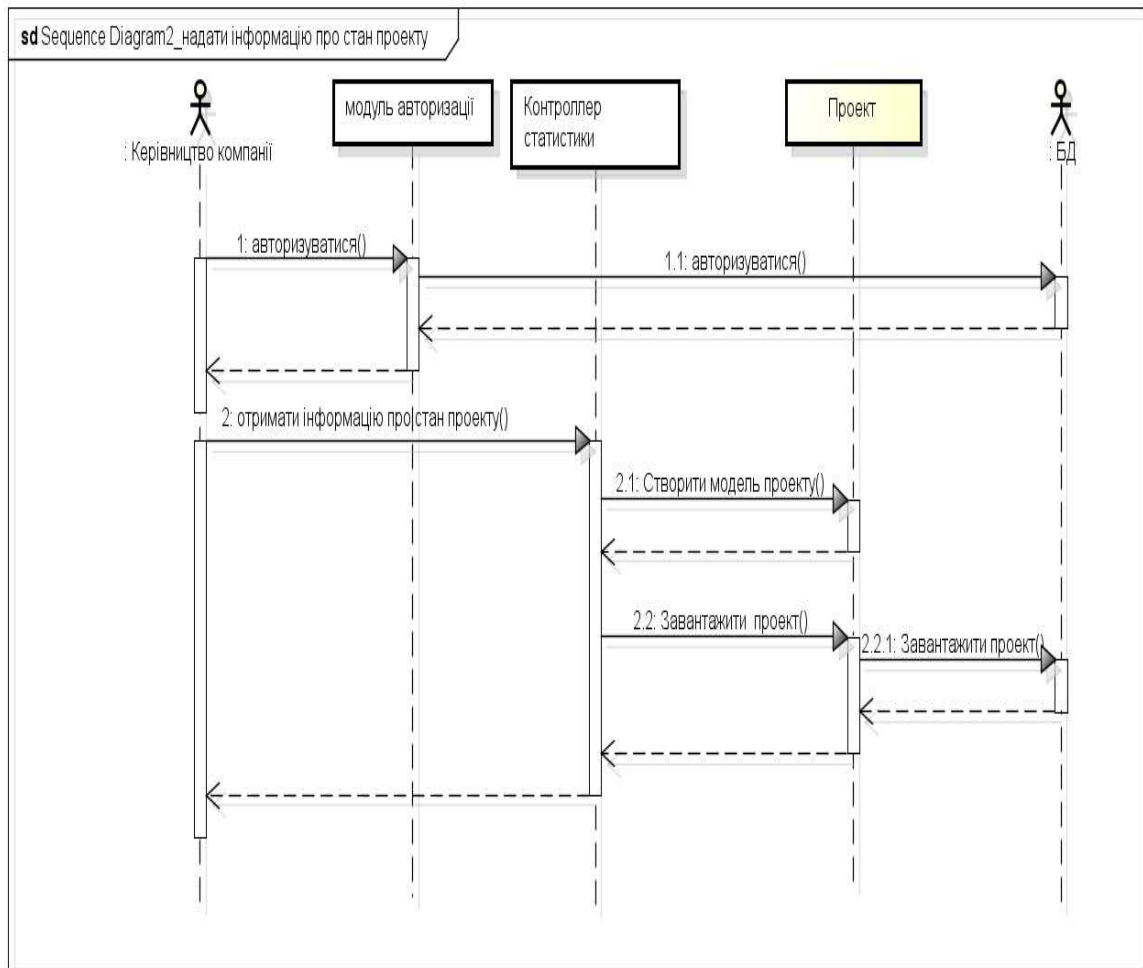


Рисунок 2.9— Діаграма послідовності «Створення завдання»

2.4.4 Структура програмного забезпечення

Виходячи з аналізу функціональної моделі інформаційної системи (див. пункт 2.1.3), можна розробити структуру програмного забезпечення. У даній роботі буде описана розробка програмного забезпечення для управління проектами. Оскільки програмне забезпечення для неї є розподіленим [16] (працює на декількох комп'ютерах), то воно повинно мати модульну структуру. Схема взаємодії модулів програмного забезпечення представлена на рисунку 2.12.

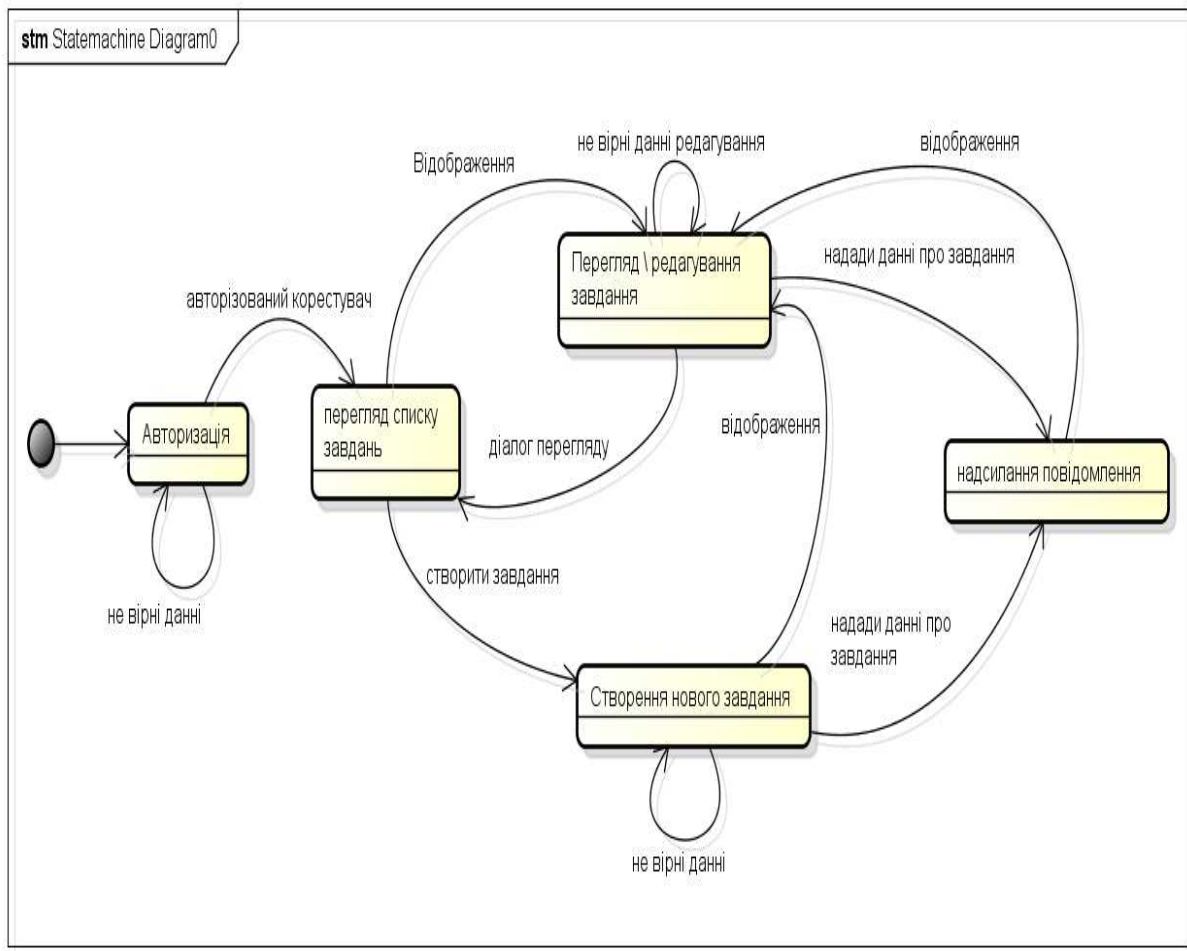


Рисунок 2.11 – Діаграма переходів стану модуля завдань

Перший модуль, що отримує управління, є модуль домашньої сторінки. Він виводить на екран головне меню, а також контент з інформацією про відділ або про компанію.

Наступний модуль, якому може передатися управління – це модуль аутентифікації. Користувач (менеджер відділу або керівник компанії) вводить свій логін і пароль, які звіряються з тими що створені адміністратором та записані в базу даних підсистеми. Далі виконується переадресація користувача знову на головну сторінку, де видно що він вже авторизований.

Головне вікно програми залишається активним. Воно містить меню, за допомогою якого можна викликати інші модулі програмного забезпечення. При цьому вся інформація й інтерфейс користувача, що надають різні модулі, будуть з'являтися в основній частині головного вікна. Таким чином,

користувач працює тільки з одним вікном, у якому є меню та інформація. Інформація змінюється в залежності від того, який пункт меню вибирається.

З модуля домашньої сторінки можна викликати (тимчасово передати управління) наступні модулі:

- модуль проектів;
- модуль завдань;
- модуль працівників;
- модуль замовників;
- модуль помилок;
- модуль обліку робочого часу;
- модуль аутентифікації.

Усі модулі напряму взаємодіють з базою даних. Будь-який з модулів може викликати інший, відповідно до прав доступу, визначених у підсистемі.

2.4.5 Специфікація модулів програмного забезпечення

Програмне забезпечення складається з 8 модулів. Нижче приведена специфікація всіх модулів.

1) Модуль домашньої сторінки.

Опис: модуль створює і відображає домашню сторінку, тобто сторінки з якої починається навігація сайтом. Сторінка складається з декількох частин: ліва, заголовок, контент (або тіло) та нижня. Модуль генерує код HTML для кожної частини. Якщо є залогінений користувач, то відповідно до прав доступу створюється меню навігації по розділам сайту, за допомогою якого можна викликати будь-який модуль.

Вхідні дані: Немає.

Вихідні дані: HTML-документ.

Функції модуля:

- створення сторінки;

– генерація елементів інтерфейсу за допомогою HTML.

2) Модуль аутентифікації.

Опис: модуль виконує аутентифікацію користувача при вході в систему.

Вхідні дані: логін і пароль.

Вихідні дані: немає.

Функції модуля: провести аутентифікацію користувача, визначити його права доступу, провести валідацію аутентифікації.

3) Модуль замовників.

Опис: до цього модуля має доступ лише адміністратор та керівництво компанії; модуль реалізує стандартний CRUD [16] (Create-Read-Update-Delete) для сутності «Працівник» у вигляді таблиці усіх замовників, з посиланнями на сторінки редагування, перегляду і видалення кожного замовника.

Вхідні дані: немає.

Вихідні дані: інформація про замовників з можливістю її перегляду, редагування та видалення.

Функції модуля: надання можливості керівникам компанії продивлятися інформацію про замовників створювати нових, редагувати

4) Модуль працівників.

Опис: інтерфейс модуля включає реєстраційну форму, яку працівники компанії заповнюють при зачислюванні в штат працівника, таблицю з усіма працівниками, які працюють в компанії, з можливістю перегляду, редагування, видалення та пошуку профілів працівників. Модуль обов'язково викликає модуль відділів.

Вхідні дані: П.І.Б., дата народження, серія паспорта, номер паспорта, місце проживання, ідентифікаційний код, телефон, e-mail працівника.

Вихідні дані: HTML-документ з даними про працівників.

Функції модуля:

- додавання нових працівників;
- управління профілями клієнтів (CRUD);
- проводити валідацію введених значень.

5) Модуль проектів.

Опис: додавання проектів, редагування, перегляд та видалення інформації про проекти, отримання інформації про виконані та не виконані проекти.

Вхідні дані: номер замовника, дата створення, опис, статус виконання

Вихідні дані: HTML-документ з даними про проект.

Функції модуля:

- управління проектами;
- додавання нових нових проектів;
- валідація, відмітка про виконання.

6) Модуль завдання.

Опис: управління завданнями по проектам. Модуль обов'язково викликає модулі працівників та проектів.

Вхідні дані: проект , опис, дата створення, дата виконання, статус виконання.

Вихідні дані: повідомлення про завдання, HTML-документ з даними про проект.

Функції модуля:

- управління завданнями;
- валідація введених значень;
- надсилання повідомлень про створення або виконання завдання.

7) Модуль помилок

Опис: управління помилками знайденими в результаті тестування проекту.

Вхідні дані: код проекту , код працівника, дата створення, заголовок опис.

Вихідні дані: HTML-документ з даними про помилки.

Функції модуля:

- управління помилками;
- створення коментарів до помилок;
- валідація введених значень.

8) Модуль відділів.

Опис: ведення обліку усіх відділів компанії.

Вхідні дані: назва відділу, опис.

Вихідні дані: немає.

Функції модуля:

- створення відділів;
- перегляд інформації про відділи;
- редагування інформації про відділи;
- видалення відділів.

2.4.6 Засоби розробки програмного забезпечення

На основі розробленого математичного забезпечення та специфікації модулів та з метою отримання програмного коду для створення завдань слід розглянути засоби розробки програмного забезпечення. Програмне забезпечення буде розроблятися відповідно до парадигми MVC (рисунок 2.13) [10]. При розробці програмного забезпечення використовується технологія швидкої розробки Web-додатків Ruby on Rails. Ruby on Rails надає архітектурний зразок Model-View-Controller для веб-додатків, а також забезпечує їх інтеграцію з веб-сервером та сервером бази даних. Ruby on Rails є відкритим програмним забезпеченням.

Безпосередньо автором буде розроблятися частина, яка стосується логіки створення завдань до проектів та логіки надіслання повідомлень учасникам проекту та завдань а також логіку авторизації в системі , а саме :

- контролер – Make Tasks Controller – tasks_controller.rb, rails 2.2 [110]
- контролер – MailerController – mailer_controller.rb, rails 2.2 [110]
- контролер – LoginController – login_controller.rb, rails 2.2 [110]

Програма та методика випробувань програмного забезпечення викладена в додатку В, інструкція адміністратора з установки програмного забезпечення на Web-сервер знаходиться у додатку Г, а інструкція користувача представлена у додатку Б, текст приведених вище модулів представлений у додатку Е.

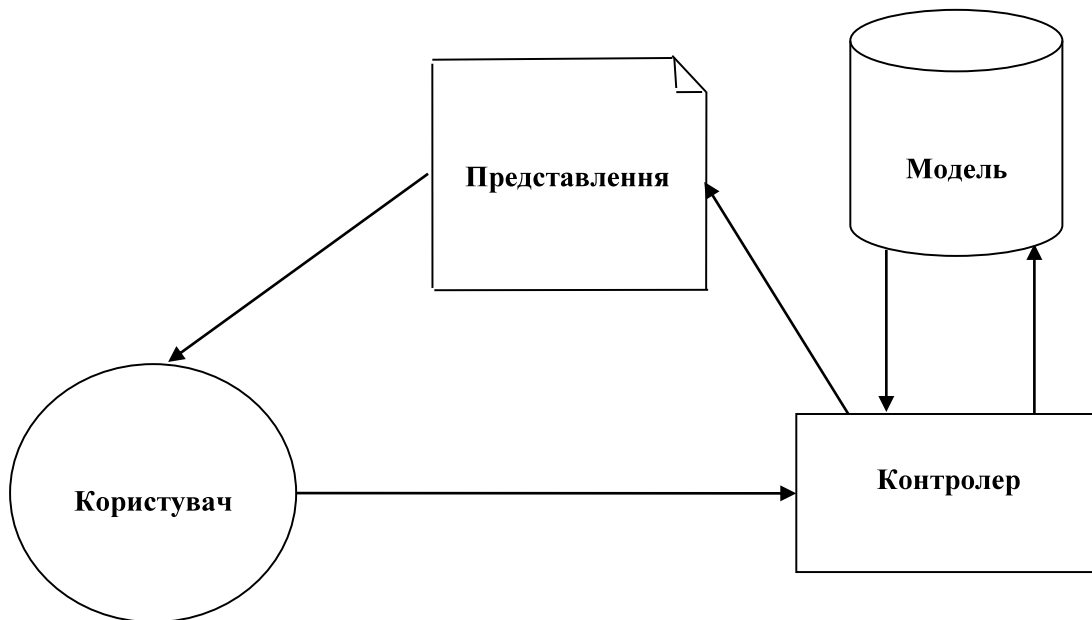


Рисунок 2.13 – Модель запиту та відповіді до програмного забезпечення розробленого в парадигмі MVC

2.5 Рішення з організаційного забезпечення

Розробляється інтегрована система управління проектами компанії "GeeksForLess" (далі система управління). Схема організаційної структури компанії "GeeksForLess" представлена на рисунках 1.1, 1.2, 1.3 та 1.4 . Виходячи з того, що впровадження системи управління потребує створення нової посади. – адміністратор підсистеми – організаційна структура компанії "GeeksForLess" зміниться.

Розробляється підсистема резервування номерів автоматизованої інформаційної управляючої системи готельної мережі (далі підсистема). Схема організаційної структури сучасного готелю представлена на рисунку 1.1. Виходячи з того, що впровадження підсистеми резервування потребує створення нової посади – адміністратор підсистеми – організаційна структура готелю зміниться. Нова схема організаційної структури представлена на рисунку 2.14.

Користувачами підсистеми, що розробляється, будуть керівництво компанії та менеджери компанії та розробники проектів. Для

адміністрування підсистеми може бути найнятий адміністратор або це може бути один з LAN адміністраторів компанії.

Виходячи з розробленої схеми організаційної структури та з метою розробки документального підкріплення реорганізації структури компанії "GeeksForLess" слід розробити ряд документів, які повинні управляти створенням відділу адміністрування, призначенням на посаду адміністратора підсистеми та його посадовими правами та обов'язками [13].

Таким чином, зміни в організаційній структурі управління полягають у створенні робочого місця для адміністратора підсистеми. Відповідні документи, які становлять організаційне забезпечення представлені в додатку Д.

2.6 Рішення з технічного забезпечення

Програмне забезпечення системи управління проектами компанії "GeeksForLess" має розподілену архітектуру, тобто передбачає використання декількох серверів і клієнтських машин.

Для максимальної продуктивності і масштабування системи необхідно використовувати два сервери (хоча для коректної роботи системи досить і одного сервера, на який потрібно установити все програмне забезпечення):

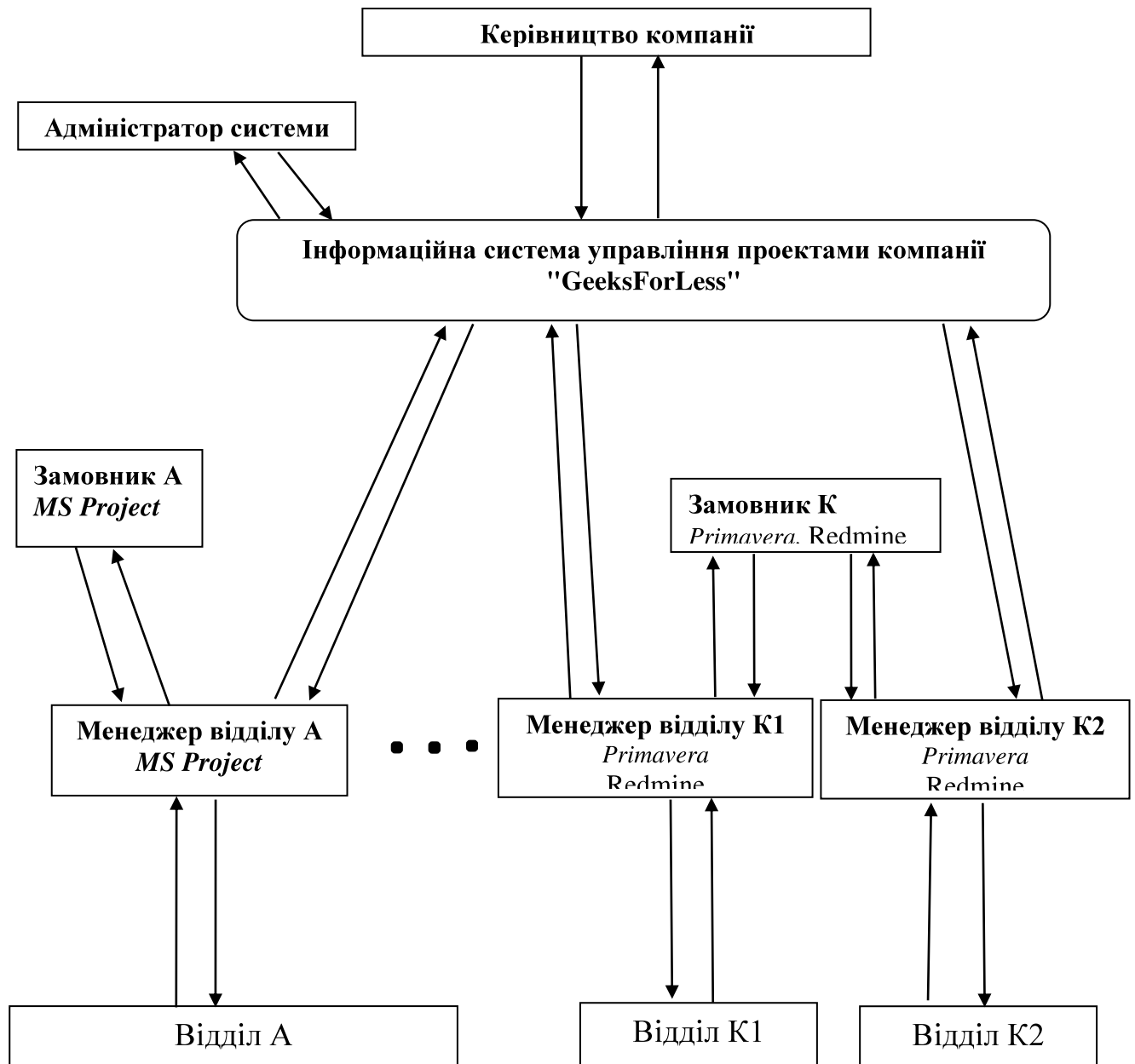


Рисунок 2.14 – Модель організаційної структури управління проектами компанії "GeeksForLess"

- web-сервер;
 - сервер баз даних (БД) MySQL 5.5
- Вимоги до технічного забезпечення Web-сервера:
- процесор класу AMD X2 64-bit 4200+ і вище;
 - оперативна пам'ять DDR II 800 4 Гб і більше;
 - жорсткий диск з вільним місцем 100 Гб і більше;

- мережева карта Ethernet, що підтримує швидкості передачі не менше 100 Мбіт/сек.

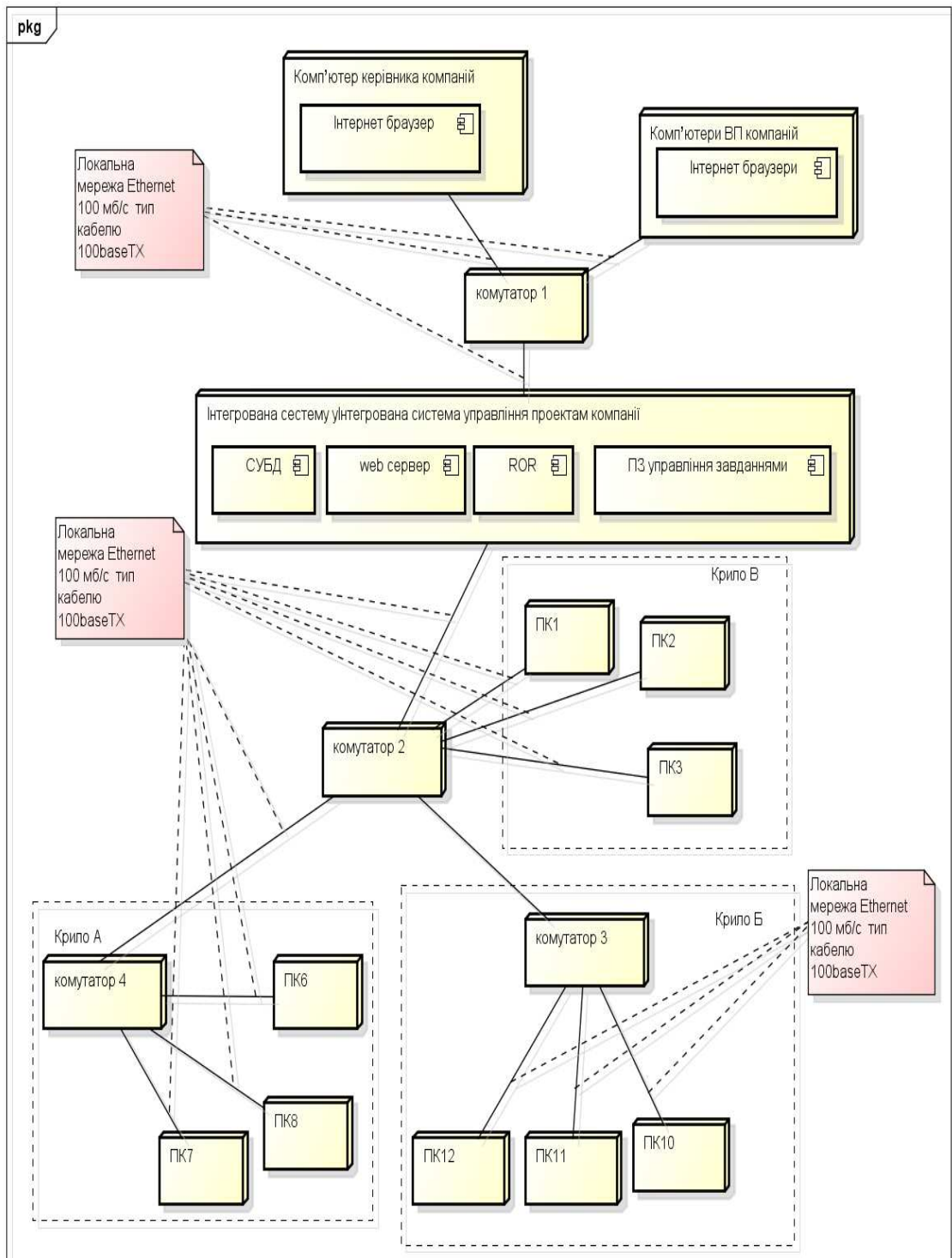
Оскільки робочі місця працівників компанії "GeeksForLess" вже встановлені та розташовані в одному будинку, обмін інформацією та доступ системи управління проектами відбувається через локальну мережу Fast Ethernet.

Усе необхідне обладнання, включаючи локальну мережу, проведену у відділі, свічі, підключення до мережі Інтернет вже встановлено.

Розташування кожного елемента системи проектувалося з урахуванням технічних вимог до установки й експлуатації, зазначених у супровідній документації до встаткування й техніки безпеки:

- проведення мережних кабелів у спеціальних кожухах, що запобігають механічним ушкодженням;
- заборона проведення силових кабелів поблизу мережного встаткування;
- кабелі не повинні бути передавлені або натягнуті;
- корпуси системних блоків повинні бути заземлені.

На рисунку 2.15 представлена схема розгортання системи



Рисунку 2.15 – Схема розгортання системи

4 РЕЗУЛЬТАТИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ДЛЯ КОМПАНІЇ "GeeksForLess"

4.1 Реалізація проекту розробки інформаційної системи

Розроблений проект інформаційної системи, вимоги до якого зазначені у технічному завданні (додаток А), призначений для покращення управління проектами компанії "GeeksForLess". Реалізація проекту відбулася та програмний код представлений у додатку В.

4.2 Тестування інформаційної системи

Тестування – це перевірка відповідності між реальною та очікуваною поведінкою інформаційної системи, що здійснюється на кінцевому наборі тестів, обраних певним чином.

Тестування інформаційної системи полягає в тестуванні реалізованих модулів, необхідних для функціонування розроблюваної ІС.

Первинною задачею є вибір стратегії тестування. Найбільшої популярності набули методи тестування «білого» та «чорного ящика».

Перший метод тестування «чорного ящика» являє собою стратегію тестування функціональної поведінки об'єкта (програми, системи), при якому не використовуються знання о внутрішній структурі об'єкта. Стратегія поведінкового тестування виходить з технічних вимог та їх специфікацій.

Другий метод «білого ящика» виконується з ціллю виявлення проблем у внутрішній структурі програми. Загальною задачею такого тестування є забезпечити перевірку кожного кроку по алгоритму роботи. Тестування інтегрованої системи в основному спрямовано на інтерфейс між блоками, що повинно гарантувати сумісність блоків та їх коректну спільну роботу.

Так як в нашому випадку тестування відбуватиметься через інтерфейси користувача, тому доцільніше обрати метод тестування «чорного ящика».

В цьому методі програма розглядається як «чорний ящик». Існує декілька прийомів тестування чорного ящика:

- Еквівалентне розбиття;
- Аналіз граничних значень;
- Аналіз причинно-наслідкових зв'язків;

- Припущення про помилку.

Скористаємося прийомом, що дозволяє провести досить повне тестування – еквівалентного розбиття. Основу методу складають два положення:

1. Вихідні дані необхідно розбити на кінцеве число класів еквівалентності. В одному класі еквівалентності містяться такі тести, що, якщо один тест з класу еквівалентності виявляє деяку помилку, то і будь-який інший тест з цього класу еквівалентності повинен виявляти цю ж помилку.

2. Кожен тест повинен включати, по можливості, максимальна кількість класів еквівалентності, щоб мінімізувати загальне число тестів.

Класи еквівалентності виділяються шляхом вибору кожної вхідної умови і розбиваються на дві і більше групи. Розроблені класи еквівалентності представлені в таблиці 3.1.

Таблиця 3.1 – Класи еквівалентності

№ з/п	Вхідна умова	Правильні класи	Неправильні класи
1	Вхід до системи	Запуск системи пройшов без помилок доступу	При запуску системи виникає помилка доступу до інформаційної бази
2	Ідентифікація користувача	Ідентифікація користувача виконана успішно	Не з'явилося вікно для ідентифікації користувача системи
3	Відображенні інтерфейсу	Усі складові інтерфейсу виведені у вікно роботи із системою	Не показаний будь-який елемент інтерфейсу, необхідний для роботи користувача
4	Відображення форм об'єктів	Форми для роботи з об'єктами системи відображаються успішно	Помилки у відображенні форм об'єктів
5	Заповнення полів вводу	Заповнення даних виконується без помилок	Заповнення чи збереження даних у полях вводу викликає помилки

Тести, що покривають правильні й неправильні класи еквівалентності представлені в таблиці 3.2.

Таблиця 3.2 – Тести класів еквівалентності

№ тесту	Класи	Вхідні дані	Вихідні дані
1	1	Система запущена без виникнення помилок	З'явилося вікно для ідентифікації користувача
2	1	Збій у запуску системи	Вікно з повідомлення про помилку доступу до інформаційної бази
3	2	Ідентифікація користувача виконана	З'являється головний інтерфейс користувача у вікні роботи з системою
4	2	Ідентифікація користувача не виконується	Вікно з повідомленням помилки ідентифікації
5	3	Інтерфейс користувача відображений у вікні роботи з системою	Відображені усі ярлики для роботи користувача
6	3	Інтерфейс користувача відображає не усі ярлики для роботи	Повідомлення про обмеження роботи користувача
7	3	Ярлики для роботи користувача не виконують відповідний перехід	Вікно з повідомленням помилки переходу до об'єкту роботи користувача
8	4	Усі форми об'єктів відображаються відразу по команді	Зображена форма відповідного об'єкта
9	4	При відображенні форми чи її елементів виникає помилка	Вікно з повідомлення про помилку відображення
10	5	Заповнюються поля вводу безпомилково	Повідомлення про успішне збереження введених даних
11	5	Дані, що необхідно ввести не заносяться	Вікно з повідомлення про помилку введення даних

Проведене тестування чорного ящика прийомом еквівалентних класів надає змогу зробити висновок про те, що система здійснює контроль введених даних та оперативно відповідає на помилкові дії користувача. Також при виникненні системних помилок виводиться детальна інформація про помилку, що дозволяє адміністратору в короткі строки вирішити проблему. Система відповідає вимогам висунутим в технічному завданні.

ВИСНОВКИ

Інформаційна система управління проектами підприємства (ІСУПП) – це набір інтегрованих програм, які комплексно, в єдиному інформаційному просторі підтримують всі основні аспекти управлінської діяльності підприємства - планування ресурсів (фінансових, людських, матеріальних) для виробництва товарів (послуг), оперативне управління виконанням планів (включаючи постачання, збут, ведення договорів), всі види обліку, ведення господарської діяльності.

В роботі була розроблена інформаційна система автоматизованого управління проектами для компанії, яка має можливість створення звітів по проектам, ведення обліку робочого часу працівників та надання статистики роботи як окремих працівників так і відділів.

У задачі роботи входять опис структур, механізмів і можливостей типових систем управління проектами, дослідження й аналіз досвіду використання систем управління проектами, огляд ринку програмних продуктів з автоматизації управління проектами й самостійна розробка інформаційної системи управління проектами для компанії.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Саварін, П.В. (2022). Комп'ютерні системи та мережі: Моделювання Систем масового обслуговування,
https://elib.lntu.edu.ua/sites/default/files/elib_upload,
2. TiketCRM офіційний сайт інформаційної системи
<https://promo.ticketcrm.com/ru>
3. Офіційний сайт програмного забезпечення. Автоматизація кінотеатрів, театрів, концерт-холів, музеїв, фестивалей
<https://expertsolution.com.ua/uk/avtomatizacija-kinoteatrov--teatrov--operi>
4. Event офіційний сайт інформаційної системи <https://event.net.ua/>
5. Elib. *Моделі даних. Реляційна модель даних*
https://elib.lntu.edu.ua/sites/default/files/elib_upload/BD_2016_3/page5.html
6. Гайна, Г.А. (17.02.2016) Основи проектування баз даних
<https://studfile.net/preview/5319758/page:3/>
7. Курсовая ИС (14.05.2016) Діаграми послідовності
<https://studfile.net/preview/6354085/page:4/>
8. Studwood. *Діаграми станів*
https://studwood.net/1884531/informatika/diagrama_staniv#588
9. Zosym, M.A. (03.19.2023) Моделювання станів (State Modelling)
<https://www.maxzosim.com/state-modelling/>
10. Wikipedia.UA. (12. 01. 200) Діаграма розгортання
<https://uk.wikipedia.org/wiki>
11. Паркер, А.Г. (09.26.2024) Діаграма розгортання: Підручник з UML із прикладом <https://www.guru99.com/uk/deployment-diagram-uml-example.html>
12. МПЗМ. Лекція 1. Приклади моделей ІВС Аналіз та комп моделювання
https://learn.ztu.edu.ua/pluginfile.php/331376/mod_resource/content/.pdf