

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук,
управління та адміністрування

Кафедра Інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Створення системи управління командною розробкою
ІТ-проектів

Виконав студент 2 курсу групи МІС-20
спеціальності 122 Комп'ютерні науки
Клюшник Ігор Валентинович

Керівник к.т.н., доцент
Гнатовська Ганна Арнольдівна

Рецензент к.геогр.н., доцент
Кузніченко Світлана Дмитрівна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук, управління та адміністрування

Кафедра Інформаційних технологій

Рівень вищої освіти магістр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

“ 28 ” жовтня 2021 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Клюшнику Ігорю Валентиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи Створення системи управління командною розробкою ІТ-проектів

керівник роботи Гнатовська Ганна Арнольдівна, к.т.н., доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 18 ” жовтня 2021р. №216 «С»

2. Строк подання студентом роботи 9 грудня 2021р.

3. Вихідні дані до роботи: ресурси мережі Інтернет; технологія ASP.Net; мова програмування C#; Angular, HTML5, CSS3, СУБД MSSQL.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): Дослідження та аналіз завдань систем управління ІТ-проектами. Архітектура та програмні засоби реалізації системи управління проектами. Проектування системи управління ІТ-проектами. Програмна реалізація системи управління. Висновки.

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання ви- дав	завдання прийняв

7. Дата видачі завдання « 28 » жовтня 2021 р..

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1.	Дослідження та аналіз дослідження та аналіз завдань систем управління	28.10.21 – 02.11.21	96%	відмінно
2.	Дослідження особливостей ІТ-проектів. Класифікація ІТ-проектів	02.11.21 – 06.11.21	94%	відмінно
3.	Огляд та аналіз функціональних можливостей існуючих систем. Постановка завдання	06.11.21 – 10.11.21	92%	відмінно
4.	Вибір архітектури та програмних засобів реалізації системи	10.11.21 – 12.11.21	96%	відмінно
5.	Проектування систем управління командною розробкою ІТ-проектів	12.11.21 – 16.11.21	96%	відмінно
6.	Програмна реалізація систем управління командною розробкою ІТ-проектів	16.11.21-04.12.21	94%	відмінно
7.	Рубіжна атестація	22.11.20	96%	відмінно
8.	Оформлення пояснювальної записки та презентації	04.12.21-08.12.21	100%	відмінно
9.	Подання роботи на кафедру	09.12.21		
10.	Перевірка на плагіат	10.12.21		
11.	Рецензування	16.12.21		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)		96%	відмінно

Студент _____ Ключник І.В.
(підпис) (прізвище та ініціали)Керівник роботи _____ Гнатовська Г.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Тема магістерської роботи «Створення системи управління командною розробкою ІТ-проектів».

Актуальність магістерської роботи полягає в розробці та застосуванні спеціальних програм, які призначені для створення єдиного інформаційного простору для управління проектами, що забезпечує автоматизацію управління ІТ-проектами та суттєво підвищує ефективність діяльності розробників.

Об'єкт дослідження – процеси проектування та розробки системи управління командною розробкою ІТ-проектів, що забезпечить автоматизацію процесів управління, спрямованих на підвищення ефективності діяльності команди розробників.

Предмет дослідження – засоби та завдання здійснення управління розробкою ІТ-проектів для створення системи автоматизації процесів управління при командній розробці проектів.

Метою створення системи управління командною розробкою ІТ-проектів є забезпечення розробників ІТ-проектів різної складності автоматизованими засобами та інструментами, що надають можливість якісної розробки програмних продуктів та спрямовані на отримання переваг при здійсненні командної роботи над проектами.

В роботі було проведено дослідження та аналіз завдань систем управління, враховуючи особливості розробки ІТ-проектів, проведено вибір архітектури та програмних засобів реалізації системи, здійснена розробка системи управління. Практична цінність роботи полягає в тому, що створена система забезпечує розробників програмних продуктів зручними засобами автоматизованого управління розробкою ІТ-проектами.

Ключові слова: СИСТЕМА УПРАВЛІННЯ, РОЗРОБКА ІТ-ПРОЕКТІВ, ПРОЕКТУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ..

Магістерська робота містить 76 сторінок, 12 таблиці, 37 рисунків, 19 посилань.

ANNOTATION

The theme of master's work is "Elaboration of a Management System for Team Development of IT Projects".

The relevance of the master's thesis is the development and application of special programs that are designed to create a single information space for project management, which automates the management of IT projects and significantly increases the efficiency of developers.

The object of research is the processes of designing and developing a management system for team development of IT projects, which will automate management processes aimed at improving the efficiency of the development team.

The subject of the research is the means and tasks of managing the development of IT projects to create a system for automating management processes in team project development.

The purpose of creating a management system for team development of IT projects is to provide developers of IT projects of varying complexity with automated tools and tools that provide quality software development and aimed at gaining benefits in teamwork on projects.

The study and analysis of management systems, taking into account the peculiarities of the development of IT projects, the choice of architecture and software for the implementation of the system, the development of the management system.

The practical value of the work is that the created system provides software developers with convenient means of automated management of IT project development.

Key words: MANAGEMENT SYSTEM, IT PROJECT DEVELOPMENT, DESIGN, SOFTWARE .

The master's thesis contains 76 pages, 12 tables, 37 figures, 19 references.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧОК І ТЕРМІНІВ.....	10
ВСТУП	11
1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЗАВДАНЬ СИСТЕМ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ.....	14
1.1 Основні засади автоматизації процесів управління проектами	14
1.2 Цілі та завдання автоматизації управління командною розробкою ІТ-проектів	16
1.3 Дослідження особливостей ІТ-проектів	19
1.4 Класифікація ІТ-проектів	22
1.5 Визначення основних вимог до систем управління проектами	24
1.6 Огляд та аналіз функціональних можливостей існуючих систем	27
1.7 Постановка завдання.....	38
2 АРХІТЕКТУРА ТА ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ.....	40
2.1 Вибір архітектурного рішення для створення системи	40
2.2 Програмні засоби розробки серверного додатку системи.....	43
2.3 Програмні засоби та технології створення web-клієнтів	47
2.4 Система управління базою даних.....	49
2.5 Вибір засобів для створення Unit-тестів.....	50
3 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ	53
3.1 Загальні вимоги до системи управління проектами.....	53
3.2 Функціональні можливості користувачів web-системи.....	54
3.3. Проектування бази даних системи	55
3.4 Схема функціонування системи управління ІТ-проектами.....	65
4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ.....	67
4.1 Реалізація загальних елементів інтерфейсу користувачів	67
4.2 Робота з додатком користувача системи управління	68
4.3 Реалізація функцій користувача-адміністратора системи	75
ВИСНОВКИ.....	77
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	78

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧОК І ТЕРМІНІВ

CSS – Cascading Style Sheets – каскадні таблиці стилів

HTML – HyperText Markup Language – мова гіпертекстової розмітки

JavaScript – об'єктно-орієнтована скриптова мова програмування.

Веб-інтерфейс – це сукупність засобів, за допомогою яких користувач взаємодіє з веб-сайтом або будь-яким іншим додатком через браузер.

Веб-клієнт – клієнтська частина, програма, яка запитує послуги, ресурси, дані або обробку в іншій програмі через мережу Інтернет.

C# – об'єктно-орієнтована мова програмування. Розроблена в 1998-2001 роках групою інженерів компанії Microsoft.

API (Application Programming Interface) – це набір готових класів, процедур, функцій, структур и констант, що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах.

TypeScript – мова програмування, позиціонується як засіб розробки веб-застосунків, що розширює можливості JavaScript.

ASP.NET (Active Server Pages для .NET) – платформа розробки веб-додатків від компанії Майкрософт, до складу якої входить: веб-сервіси, програмна інфраструктура, модель програмування.

.NET Framework – програмна платформа від компанії Майкрософт.

CLR (Common Language Runtime) – це загальнономовне виконуюче середовище, компонент пакету Microsoft .NET Framework.

Unit testing – це метод тестування програмного забезпечення, який полягає в створенні окремих модулів для тестування основної програми.

ВСТУП

Сьогодні ефективне управління проектами немислимо без використання сучасних програмних засобів – адже розміри проектів зростають, зростає обсяг інформації, збільшується кількість співробітників, які працюють над одним і тим самим завданням. Щодня з'являються все більше команд, компаній та самостійних розробників які виконують задачі найрізноманітнішої складності. Управління розробкою програмних систем (software management) – це діяльність, спрямована на забезпечення необхідних умов для роботи колективу розробників програмного забезпечення (ПЗ), на планування та контроль діяльності цього колективу з метою забезпечення необхідної якості ПЗ, виконання термінів та бюджету розробки ПЗ.

Сьогодні вже існують багато практик, які накопичили в собі великий опит та запропонували структури для забезпечення ефективного та якісного процесу, який дозволяє в певний проміжок часу та в умовах обмежених ресурсів виконати поставлені задачі що до розробки програмних систем. Отже для кожної команди розробників програмного забезпечення дуже важливо обрати в самій якій спосіб буде формуватися їх робота. Організація процесу розробки програмного забезпечення – складний процес, управління яким не завжди підходять системи відстеження запитів чи управління проектами. Як правило, існує значний обсяг робіт, що вимагає обробки за допомогою бізнес-процесів, що гнучко конфігуруються, що дозволяють налаштовувати переходи між завданнями, затверджувати і контролювати їх, взаємодіяти в єдиній системі. Серед них такі процеси, як: відстеження помилок; управління запитами на зміну; управління інцидентами та проблемами [1]¹⁾.

На ринку програмного забезпечення представлено велика кількість програмних систем, які дозволяють частково або повністю автоматизувати

¹⁾ [1] Фесенко Т. Г. Управління проектами: теорія та практика виконання проектних дій: навч. посібник. Харк. нац. акад. міськ. госп-ва. – Х. : ХНАМГ, 2012. – 181 с.

деякі процеси командної розробки програмного забезпечення. Програмне забезпечення автоматизує деякі стадії виконання та планування, відстежує та планує етапи виконання проекту, а також може надавати статистичну інформацію, яка допомагає вчасно ввести необхідні зміни в процеси реалізації проектів. Крім того, працівники, які задіяні у проекті, можуть фізично не розміщуватися в одному приміщенні та спільно використовувати інструментарій для управління вимогами та процесом розробки [2]¹⁾.

Типові системи управління процесами розробки можуть підійти далеко не кожному підприємству, тому що кожна компанія прагне впровадити максимально ефективні технології для підтримки своєї роботи, що чимало важливо, прийнятні з питань фінансування. Для здійснення проектної діяльності сучасні компанії практикують використання спеціальних програм, які призначені для створення єдиного інформаційного простору для управління проектами. Отже розробка таких систем управління командною розробкою проектів є актуальним завданням, що забезпечує автоматизацію управління ІТ-проектами та суттєво полегшить роботу працівників, підвищить ефективність діяльності.

Об'єкт дослідження – процеси проектування та розробки системи управління командною розробкою ІТ-проектів, що забезпечить автоматизацію процесів управління, спрямованих на підвищення ефективності діяльності команди розробників.

Предмет дослідження – засоби та завдання здійснення управління розробкою ІТ-проектів для створення системи автоматизації процесів управління при командній розробці проектів.

Метою створення системи управління командною розробкою ІТ-проектів є забезпечення розробників ІТ-проектів різної складності автоматизованими засобами та інструментами, що надають можливість якісної розро-

¹⁾ [2] Що таке управління проектами: як здійснюється і що в себе включає? URL: <https://blog.agrokebety.com/shcho-take-upravlinnya-proektamy> (дата звернення 10.09.2021).

бки програмних продуктів та спрямовані на отримання переваг при здійсненні командної роботи над проектами.

Система управління командною розробкою ІТ-проектів являє собою програмне забезпечення, яке забезпечує етапи планування та реалізації проекту. Система надасть можливість учасникам проекту комунікації з командою на кожному етапі; можливість розподіляти привілеї кожному користувачу, та надійно нотувати та зберігати усі дії та рішення.

На етапі реалізації обраного командного ІТ-проекту система повинна надавати можливість, чітко і зрозуміло передавати затверджені завдання до частини команди які займаються втіленням їх у реальність. Для них може бути забезпечено виконання наступних функцій: перегляд змісту постановки задачі; стеження за строком виконання завдань; ведення обліку питань, які б зберігалися потім для звітності; надання звіту про виконання завдань; відстеження на якому етапі знаходиться виконання того чи іншого завдання.

Створення системи управління командною розробкою ІТ-проектів не ставить за мету виконання проекту замість розробників, а надає лише набір методик та інструментарію, який спрямований на значне підвищення ймовірності не провалити проект та отримати значні переваги при створенні.

На сучасному етапі розвитку проектної діяльності, коли далеко не кожен задум має успішний фінал, автоматизація управління проектами – справа дуже затратна та вимагає постійної роботи над покращенням загального процесу, розробляються оригінальні рішення, тому у майбутньому такі системи вимагатимуть можливості здійснення віртуального управління.

В наш час буде дуже складно знайти команду яка б не використовувала який небудь сервіс у своїй роботі. В майбутньому такі сервіси будуть все гнучкіше і масштабніше для надання можливості як найшвидше та якісніше спланувати та завершити вдалий проект.

1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЗАВДАНЬ СИСТЕМ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ

1.1 Основні засади автоматизації процесів управління проектами

Створення якісного ІТ-продукту – це, передусім створення складних систем взаємодії між елементами продукту на всіх етапах життєвого циклу проекту. Завдання проектного підходу у такому разі – забезпечення такої взаємодії на якісно високому рівні із максимальним ефектом. Цей процес значно ускладнюється широким спектром напрямів ІТ-проектів за своєю основною цільовою орієнтацією. Така особливість подібних проектів наочно демонструє необхідність докладного аналізу та виділення класифікаційних особливостей. Згідно з цільовою спрямованістю, ІТ-проект – це проект, до якого входять роботи, пов'язані з інформаційними технологіями, які у свою чергу спрямовані на створення, розвиток та підтримку інформаційних систем.

Грунтуючись на тому, що під інформаційними системами розуміють комплекс, що включає обчислювальне та комунікаційне обладнання, програмне забезпечення, лінгвістичні засоби та інформаційні ресурси, а також системний персонал та що забезпечує підтримку динамічної інформаційної моделі деякої частини реального світу для задоволення інформаційних потреб користувачів, стає очевидним, що ІТ-проекти є комплексними, відрізняються такими характеристиками як складність, масштабність та різноманітність [3]¹⁾.

Система управління командною розробкою ІТ-проектів передбачає застосування ряду методик та інструментів технологічного та організаційного характеру, що сприяють управлінню проектною діяльністю ІТ-компанії та націлених на підвищення ефективності реалізації проектів.

Такі системи управління проектами, розглядаються як універсальні ін-

¹⁾ [3] Ершов С.В. Управление проектами и программами. Конспект лекций. Архангельск: САФУ, 2015. 226 с.

струменти, які надають можливість спілкування учасникам проекту. Від однозначного розуміння мови проектного менеджменту залежить результат реалізації командного проекту з огляду на обрані критерії, такі як час, вартість, якість. Для здійснення проектної діяльності сучасні ІТ-компанії практикують використання спеціального програмного забезпечення, до якого входять такі, які призначені для створення єдиного інформаційного простору для управління проектами. Система управління командною розробкою ІТ-проектів надає зручний набір інструментів, методів, методологій, ресурсів і процедур, які використовуються командою розробників та дозволяє менеджеру проекту ефективно завершити проект. Тому можливо стверджувати, що система управління командною розробкою ІТ-проектів – це комплекс програмно-технічних рішень для реалізації ІТ-політики організації. Крім цього, необхідно врахувати, що для користувачів системи величезне значення має зовнішній вигляд та сутнісний аспект, що полягають у зручному комунікаційному та сервісному використанні, зрозумілому інтерфейсі взаємодії та логічності. Тому дуже важливо створювати систему управління командною розробкою ІТ-проектів, яка найбільше підійде для реалізації саме обраного проекту [4]¹⁾.

Система управління проектами – це ряд процесів і пов'язаних з ними функцій контролю, об'єднаних в єдину цілеспрямовану структуру, яка будується на основі плану управління проектом, який описує те, як буде використовуватися система. Зміст системи управління проектом змінюється в залежності від галузі застосування, особливостей організації, складності проекту і доступності необхідних ресурсів. Система будується так, щоб максимально відповідати стратегічним цілям і виробничим ресурсам клієнтської організації. Програмне забезпечення для систем управління командною розробкою ІТ-проектів – це комплекс програмного забезпечення, що включає в себе програми для планування завдань, складання розкладу, контролю ціни і

¹⁾ [4] Онопко А.С., Жигалкевич Ж.М. Застосування інформаційних технологій в управлінні підприємством. Актуальні проблеми економіки та управління. – 2017. № 11. URL: <http://ape.fmm.kpi.ua/article/viewFile/102782/97865> (дата звернення: 22.10.2021)

управління бюджетом, розподілу ресурсів, спільної роботи, спілкування, швидкого управління, документування та адміністрування системи, яке використовуються спільно для управління великими командними проектами [5]¹⁾.

Однією з найбільш поширених можливостей систем управління командною розробкою ІТ-проектів є можливість планування подій і управління задачами. Вимоги можуть відрізнятись в залежності від того, як використовується цей інструмент для реалізації того чи іншого проекту, серед яких можливо виділити найбільш значні: планування різних подій, що залежать одна від одної; планування розкладу роботи співробітників і управління ресурсами; розрахунок часу, необхідного на рішення кожної з задач; сортування завдань залежно від термінів їх завершення.

Сьогодні ефективне управління ІТ-проектами неможливе без використання сучасних програмних систем, що зумовлено зростанням розмірів проектів, обсягів інформації, кількості співробітників, які працюють над одним і тим самим завданням на значній відстані.

1.2 Цілі та завдання автоматизації управління командною розробкою ІТ-проектів

Насамперед, необхідно дати визначення терміну «ІТ-проект». ІТ-проект – це тимчасове підприємство, спрямоване на розробку унікального продукту, що має чітко визначений термін виконання, обмеження за ресурсами, свої критерії якості та поняття про успішне завершення.

Автоматизація управління проектами може бути спрямована на досягнення різних цілей, які залежать від масштабності та діяльності ІТ-компанії. Рівень складності завдань, пов'язаних із впровадженням систем управління

¹⁾ [5] Система управління проектами. URL: [https://uk.wikipedia.org/wiki/ Система_управління_проектами](https://uk.wikipedia.org/wiki/Система_управління_проектами) (дата звернення 10.09.2021).

проектами, залежить від багатьох факторів: розміру компанії, структури управління, типу та складності проектів. Але навіть у простих на перший погляд ситуаціях результат автоматизації безпосередньо залежить від планування. Головне завдання проектного підходу – залучення потенційних користувачів до командної роботи, що гарантує підтримку кожного учасника. Без цього автоматизація управління проектами навряд чи дасть бажаний результат.

Система управління командною розробкою ІТ-проектів передбачає облік та надання користувачам великої кількості необхідної інформації, серед якої можливо зазначити наступну:

- список завдань для команди розробників та інформацію про розподіл ресурсів;
- актуальну інформацію щодо термінів виконання завдань;
- своєчасні попередження про можливі ризики, що пов'язані з проектом;
- аналітичну інформацію щодо навантаження учасників проекту;
- інформацію щодо хід виконання стадій проекту, статистичні показники та їх прогнозування.

Основними цілями в управлінні ІТ-проектами є: ефективне використання персоналу, зайнятого у сфері ІТ; ефективність роботи ІТ-департаментів. Проект завжди націлений на результат, на досягнення певних цілей, на певну предметну область. Реалізація проекту здійснюється керівництвом проекту, менеджером проекту й командою проекту, яка працює з керівництвом, іншими учасниками проекту, які виконують специфічні види діяльності, бізнес-процеси проекту. У роботах над проектом можуть брати участь представники лінійних і функціональних підрозділів компаній, відповідальних за виконання покладених на них завдань, видів діяльності, функцій, включаючи планування, керівництво, контроль, організацію, адміністрування й інші загальносистемні функції.

Застосування системи управління проектами дозволяє визначити цілі

проекту, обґрунтувати його й оцінити життєздатність, виявити структурні елементи проекту: підцілі, завдання, задачі, роботи, які необхідно виконати; визначити необхідні обсяги та джерела фінансування; визначити терміни реалізації проекту; скласти графік виконання робіт, розрахувати необхідні ресурси, кошторис; скласти бюджет проекту, враховувати ризики, забезпечити контроль за реалізацією проекту. Успіху будь-якого ІТ-проекту залежить від вироблення вимог, визначення вимог та управління вимогами. Витрати часу ІТ-компанії на розробку програмного забезпечення значно збільшують перш за все внаслідок неефективного підходу до управління вимогами. В ІТ-компаніях, які не мають достатніх можливостей бізнес-аналізу, ІТ-проекти значно частіше закінчуються невдачею, ніж успіхом. За умови правильного визначення вимог та управління ними перевитрати за проектом можна знизити завдяки скороченню кількості неточних, неповних та втрачених вимог[6]¹⁾.

Використання системи управління командною розробкою ІТ-проектів передбачає, що можливо уявити складний проект у вигляді взаємопов'язаного паралельно-послідовного комплексу робіт та операцій. Представлений комплекс робіт та операцій можна ув'язати як у вигляді стрічкового графіка Ганта, так і мережевого графіка, і надалі, після введення вихідних даних, оперативно отримувати параметри мережного та стрічкових графіків у вигляді проекту робіт. Системи управління розробкою ІТ-проектів дозволяє, як інструмент оперативної роботи менеджера, проаналізувати управлінські рішення менеджера одночасно у різних системах координат: часової, ресурсної, організаційної та економічної. Кожен ІТ-проект незалежно від складності й обсягу робіт, необхідних для його виконання, проходить певні етапи розвитку: від початкового до заключного. Для інвесторів початок проекту пов'язаний з початком його реалізації й, відповідно, початком вкладення коштів. Закінченням існування проекту можуть бути: введення в дію об'єктів, поча-

¹⁾ [6] Рішення для управління проектами. URL: <https://blog.hubstaff.com/project-management-software-solutions/> (дата звернення 18.10.2021).

ток їх експлуатації й використання результатів виконання проекту; переведення персоналу проекту на іншу роботу; досягнення проектом поставлених цілей; припинення фінансування проекту; початок робіт із внесення у проект серйозних змін, не передбачених початковим задумом (модернізація); висновок щодо експлуатації об'єктів, передбачених проектом (рис. 1.1).

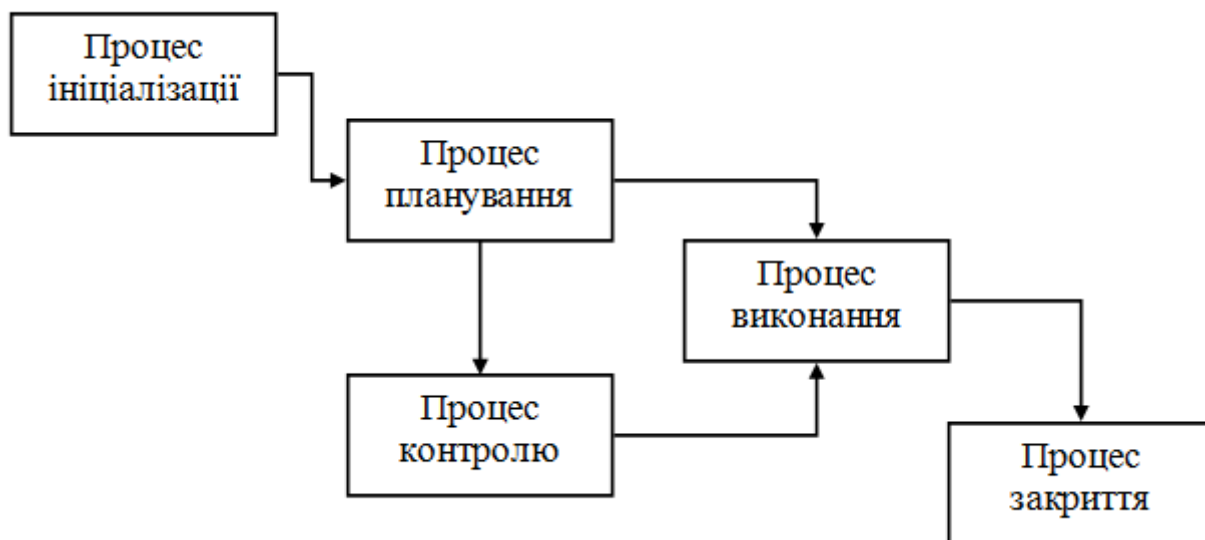


Рисунок 1.1 – Схема процесу управління ІТ-проектом

Застосування методологій управління ІТ-проектами дає можливість визначити кількісні, часові, вартісні та якісні параметри проекту, виділити та оцінити ризики.

Системи управління командною розробкою ІТ-проектів надають можливість використання різноманітних інструментів, що полегшують роботу над проектом.

1.3 Дослідження особливостей ІТ-проектів

Проекти у сфері інформаційно-комунікаційних технологій відрізняються складністю та належать до розряду високотехнологічних проектів. Виді-

ляють такі типи ІТ-проектів:

- проекти розробки та розвитку програмного забезпечення та ІС;
- проекти впровадження інформаційних систем;
- інфраструктурні та організаційні проекти зміни існуючих рішень (міграція на нову версію ПЗ, заміна апаратних частин, реструктуризація ІТ-відділу тощо).

Розробка програмного забезпечення здійснюється в рамках методологій, методів та підходів програмної інженерії. ІТ-проекти розробки та розвитку здійснюються згідно з керівними документами, що регламентують питання проектування та реалізації ІС, а також відповідно до практик програмної інженерії та методології розробки ПЗ. Методологія розробки – це єдина система принципів, понять, методів і засобів, що визначають стиль організації процесів створення ефективного програмного продукту найбільш рентабельним способом. ІТ-проекти повинні відповідати великій кількості вимог, які можна назвати різноплановими за своїм змістом. Основними з таких вимог є необхідність реалізації проектів у рамках мінімально можливих термінів при досягненні високого рівня ефективності, що є важкодосяжним завданням. Це продиктовано тим, що практично неможливо точно визначити терміни, необхідні для реалізації таких проектів. Найчастіше це викликано необхідністю внесення незапланованих змін, що є невід'ємною частиною життєвого циклу ІТ-проектів, метою яких є необхідність доопрацювання різних елементів продукту проекту. Враховуючи описані особливості, розробки в ІТ-сфері передбачають проектну форму діяльності. Також використання проектного підходу пояснюється орієнтацією на іноземний досвід та використання передових технологій [7]¹⁾.

Можливо виділити ряд особливостей ІТ-проектів, що впливають на формування ефективної системи управління:

¹⁾ [7] Богославец А.А. Классификация IT-проектов. Харьк. нац. унив. городского хозяйства им. А.Н. Бекетова. «Комунальне господарство міст», 2014, випуск №118. ISSN 0869-1231. URL: <https://core.ac.uk/download/pdf/33757322.pdf> (дата звернення 21.10.2021).

- нестандартний життєвий цикл, який може включати також тестовий, гарантійний та післягарантійний етапи розробки;
- необхідність чіткого визначення, вже на етапі ініціації, вимог до ІТ-проектів незважаючи на рухливість та неоднозначність деяких напрямів у ІТ-сфері;
- необхідність оперативного внесення змін на етапі тестування, що створює складнощі, з якими стикаються практично усі керівники ІТ-проектів, внаслідок чого відбувається відставання запланованих термінів.

За статистичними даними 70% ІТ-проектів не укладаються у відведені терміни, що призводить до перевищення бюджету проекту, і, як наслідок, невиконання основних заявлених вимог до проекту. З практичної точки зору, проект можна розглядати як поєднання очікуваних результатів та робіт з їхнього досягнення. Результати мають бути отримані з урахуванням наявних обмежень, чи то за ресурсами, якістю, строками тощо. Часто, виходячи з обмежень, можуть бути сформульовані критерії успішності проекту. Як правило, такими критеріями успішності проекту є задоволеність замовника, якість, термін, вартість, час. Відповідно проект вважається успішним, якщо він завершено: у повному обсязі; у рамках бюджету; у встановлені терміни; з заданим рівнем якості; при задоволенні замовника. Самі роботи в будь-якому проекті поділяються на дві частини: роботи предметної області (наприклад, для ІТ-проектів це створення коду, протяжка кабелів, встановлення серверів і т.п.) та роботи з управління проектом (створення планів, написання документів, проведення зустрічей і нарад тощо) [8]¹⁾.

Основою управління проектом є складання плану робіт та відстеження ходу робіт по ньому. При цьому постійно йде пошук компромісу між обсягом, термінами, бюджетом, якістю та задоволеністю замовника з урахуван-

¹⁾ [8] Молоткова Н.В., Сахаров И.С. Инфраструктура управления качеством ИТ-проектов. URL: http://vernadsky.tstu.ru/pdf/2008/03g_22g_13.pdf (дата звернення 24.10.2021).

ням наявних ризиків. Зміна одного з параметрів проекту автоматично спричиняє зміну одного або кількох інших. Основних параметрів найчастіше вважають три: терміни, бюджет, якість. В останнє «за умовчанням» включають обсяг та задоволеність замовника. Особливо це проявляється у ІТ-проектах: одна невелика зміна, наприклад, обсягу шляхом додавання кількох нових вимог може викликати непропорційне збільшення бюджету та/або термінів виконання проекту (рис. 2.1.).



Рисунок 1.2 – Розподіл робіт у проекті

На даний час управління проектами є професійною сферою, який являє собою набір спостережень, найкращих практик, застосування яких дає позитивний ефект. Особливо важливу роль відіграють саме стандарти – при їх розробці збирається, аналізується та зводяться в єдиний документ усі досягнення спільноти керівників проектів.

1.4 Класифікація ІТ-проектів

Системи управління командною розробкою ІТ-проектів використовують компанії, працівники яких не можуть фізично розміщуватися в одному приміщенні та спільно використовувати інструментарій для управління ви-

могами та процесом розробки процесом розробки. Для зручності аналізу та синтезу проектів, а також систем управління ними безліч різноманітних ІТ-проектів класифікується за різних підстав. У науковій літературі зустрічаються різні підходи до класифікації проектів [9]¹⁾. Найчастіше застосовується класифікація, наведена на рисунку 1.3.

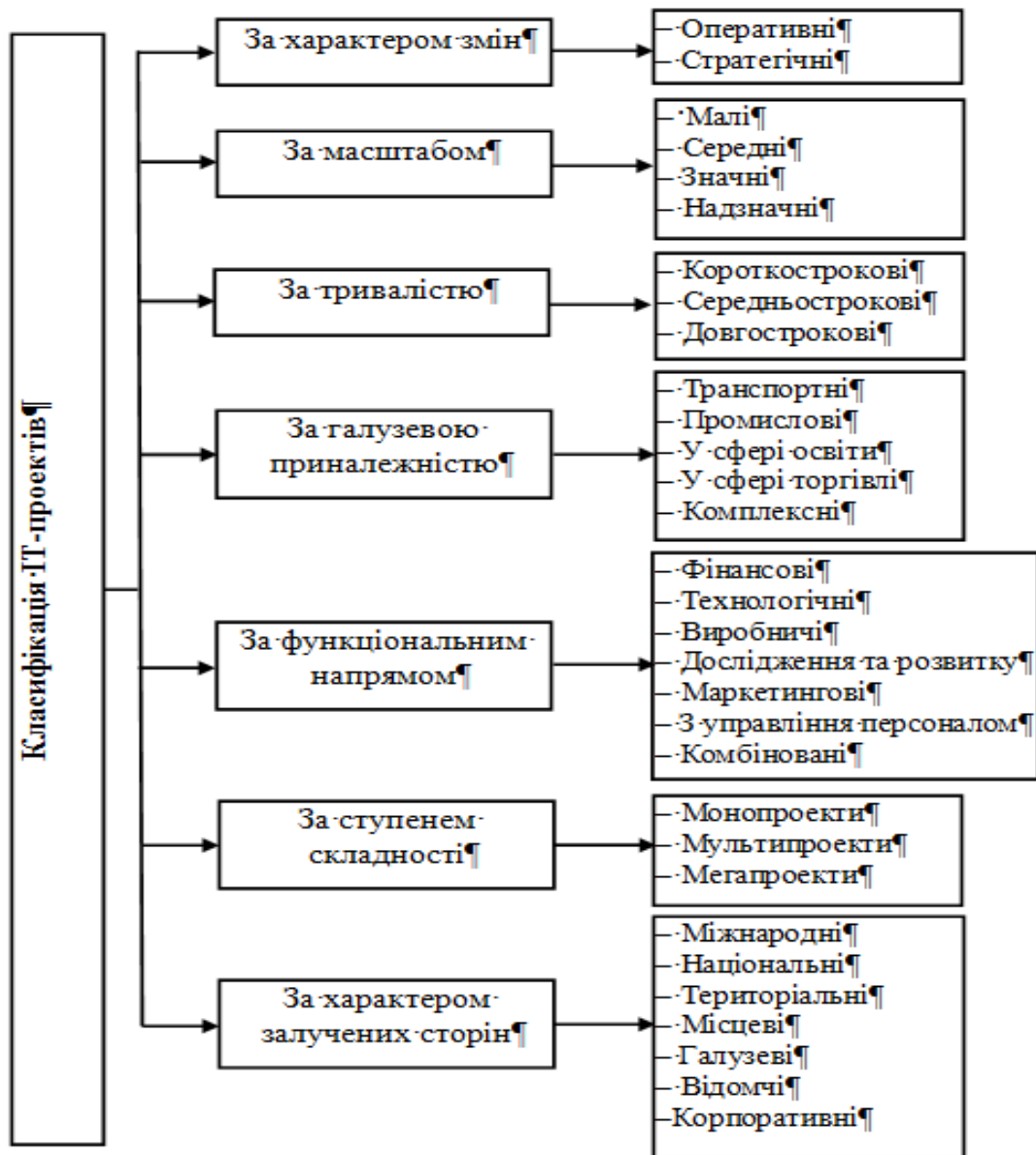


Рисунок 1.3 – Загальна класифікація ІТ- проектів

¹⁾ [9] Коньшунова А.Ю. К вопросу о классификации проектов в проектном управлении. URL: <http://sibac.info/11760> (дата звернення 24.10.2021).

Ефективність використання таких систем ІТ-компаніями забезпечує підтримку наступного переліку зручного інструментарію:

- спілкування замовника (користувачів) та розробників;
- групової (командної) роботи розробників над проектом;
- підготовки вимог;
- управління проектом (планування термінів та складу релізів, відстеження доробок та помилок);
- підготовки довідкової документації;
- обміну артефактами (моделями, кодом та ін. документами) проекту;
- трасування між вимогами, завданнями та довідковою документацією.

Зазначена класифікація є загальною і при викладанні знань або досвіду з управління проектами не використовується. На практиці, найчастіше застосовується класифікація як описове визначення «опорних» проектів, а саме поняття «середній» і «складний» проект. «Середній» – це проект згідно з розробкою програмного забезпечення тривалістю близько року, за участю 7–10 осіб та досить простий у виконанні. Поняття «складного» проекту застосовується з метою оцінки кандидатів для сертифікації спеціалістів з управління проектами. На практиці при класифікації проектів, застосовуються поняття «колір» або «масштаб проекту» і «твердість» або «відповідальність проекту». Корисність обумовлюється присутністю методології з організації виконання проектів кожної проекції (колір і твердість). На жаль, сфера дії цієї класифікації обмежена лише розробкою програмних систем. Резюмуючи все вищезгадане, можна зробити висновок, що для розробки програмних систем найбільш цікавою є класифікація, для якої застосовуються поняття «колір» (масштаб проекту) та «твердість» (відповідальність проекту).

1.5 Визначення основних вимог до систем управління проектами

Визначимо основні вимоги до розробки автоматизованих систем управління командною розробкою ІТ-проектів.

Комунікація команди проекту. Системи управління проектами повинна забезпечити розробників інструментами, які уможливають спілкування в режимі реального часу, що дозволяє всі питання вирішуватися у міру їх виникнення.

Ведення документації. Розробка та реалізація проектів супроводжується великою кількістю різноманітних документацій: звіти, аналізи, прогнози. Вбудований функціонал документообігу дозволяє проводити своєчасне оновлення статусів документів, забезпечуючи наочність та комунікацію.

Управління витратами. Автоматизація управління проектами, як правило, має на увазі наявність функціонала, що дозволяє контролювати витрати на реалізацію проекту, що є важливою перевагою системи.

Управління прогнозами, ризиками та бюджетами. За допомогою системи можна своєчасно виявляти ризики та контролювати бюджет проекту, прогнозувати результати роботи.

Складання звітів. Система повинна передбачати гнучкий формат звітів та надавати можливість оперативного доступу до даних, що підвищить можливість своєчасного вирішення завдань.

Просте впровадження. Робота із системою практично вимагає спеціального навчання, оскільки програмне забезпечення побудовано з допомогою панелей управління. Завдяки простоті встановлення та експлуатації системи, автоматизація управління проектами досить швидко підвищує ефективність роботи [10]¹⁾.

Незважаючи на досить вагомі переваги, які дає автоматизація управління проектами, засобами використання системи управління командною розробкою ІТ-проектів, варто визначити й її недоліки. Висока вартість програмного забезпечення не гарантує повної відповідності вимогам ІТ-компанії. Як правило, функціонали системи досить великі, але вимагають індивідуаль-

¹⁾ [10] Грекул В. И., Коровкина Н. Л., Куприянов Ю. В. Методические основы управления ИТ-проектами. Учебник для вузов. М.: БИНОМ. Лаб. знаний ИНТУИТ, 2010. 392 с.

ного доопрацювання або придбання програм, що супроводжують, що ускладнює використання та збільшує витрати.

Автоматизація управління проектами може зробити проект складнішим. Існує чимало компаній, які використовують подібні системи, однак у них немає жодної необхідності, якщо справа стосується простих проектів. Неналежне використання системи серйозно ускладнить реалізацію. Реальні дії гальмуються через розміщення статусів. Статус – це нагадування про поточне завдання. Іноді завдання можна вирішити набагато швидше, ніж здійснюється розміщення цих статусів. Безумовно, застосування систем управління командною розробкою ІТ-проектів, звісно, має свої мінуси, але використання таких систем значно підвищує ефективність роботи команди, тому вони можуть стати добрим помічником керівників та полегшити роботу співробітникам [11]¹⁾.

Процеси планування, моніторингу та контролю виконання проекту є найбільш постійними та трудовими у зв'язку з обробкою великої кількості даних. Вони реалізуються з використанням спеціальних інструментів. Прикладним засобом реалізації методів проектного управління є відповідне програмне забезпечення, інформаційна система управління проектами. Система управління проектами – це програмний засіб для планування робіт по проектах і моніторингу їх виконання з метою прогнозування варіантів розвитку подій і прийняття управлінських рішень.

Основним призначенням систем управління проектами є:

- структурування поточної діяльності по взаємозалежним проектам;
- візуалізація порядку управління роботою;
- розподіл відповідальності між учасниками проектів;
- оперативний моніторинг стану проектів;
- аналіз проектних ризиків;

¹⁾ [11] Заговора О.В., Концевич В.Г. Учёт особенностей ИТ-проектов при определении их жизненного цикла. URL: <http://cyberleninka.ru/article/n/uchet-osobennostey-itproektov-pri-opredelenii-ih-zhiznennogo-tsikla> (дата звернення 26.10.2021).

- скрізний контроль витрат та інвестицій між проектами та всередині них;
- автоматизація процесу збору даних про стан проектів;
- єдиний інформаційний простір управління (планування та моніторинг роботи, зберігання файлів, обговорення, наради, облік робочого часу і т. п.).

Проекти у сфері інформаційно-комунікаційних технологій відрізняються складністю та належать до розряду високотехнологічних проектів.

1.6 Огляд та аналіз функціональних можливостей існуючих систем

Ключові вимоги до системи управління IT-проектами – користувачі повинні отримувати оперативний доступ до бази знань та іншої актуальної інформації, мати можливість використовувати свіжу версію одного і того ж файлу та відстежувати зміни реального часу. Функціонал системи управління проектами обов'язково повинен включати можливість створювати задачі, призначати відповідальних завдань, ставити дедлайни і контролювати їх виконання. Таким чином система управління проектами повинна забезпечувати кожному співробітнику власний фронт робіт, а керівнику – прозорі обсяги та терміни виконання проекту, що є значною перевагою для командної роботи. Важливу роль відіграють загальні дискусії у рамках команди, а для роботи віддалених команд – групові чати і командні месенджери.

Визначимо основні критерії вибору систем управління проектами з типових завдань проектного менеджменту:

- наявність наступних функціональних можливостей: стратегічне та оперативне планування, управління завданнями, проектами та портфелем проектів, план/фактний аналіз виконання робіт, бюджетування, облік витрат та надходжень;
- єдиний інформаційний простір зберігання даних та взаємодії учасників (підтримка проектного документообігу);

- спільна робота кількох учасників у режимі реального часу;
- простота інтерфейсу та надійність;
- веб-доступ, зручність та швидкість розгортання;
- наявність методичної бази (довідники тощо) за продуктом;
- супровід та підтримка, невисока ціна придбання та використання;

Вимоги до функціональних можливостей систем управління проектами залежать від нюансів управління і визначених проектів в кожній конкретній компанії. Проведемо стислий огляд найбільш популярних систем, які надають різні можливості і функціонал для визначення більш корисних функцій, які повинна виконувати система управління командною розробкою ІТ-проектів.

Teamwork – це ПЗ для управління проектами, яке надає компаніям інструменти, серед яких: списки завдань, моніторинг часу, надсилання файлів та обмін повідомленнями, діаграми Ганта, додаток таймера робочого столу і т.п. Teamwork допомагає групам в управлінні груповими цілями, спілкуванні та встановленні корпоративних процедур (рис. 1.4).

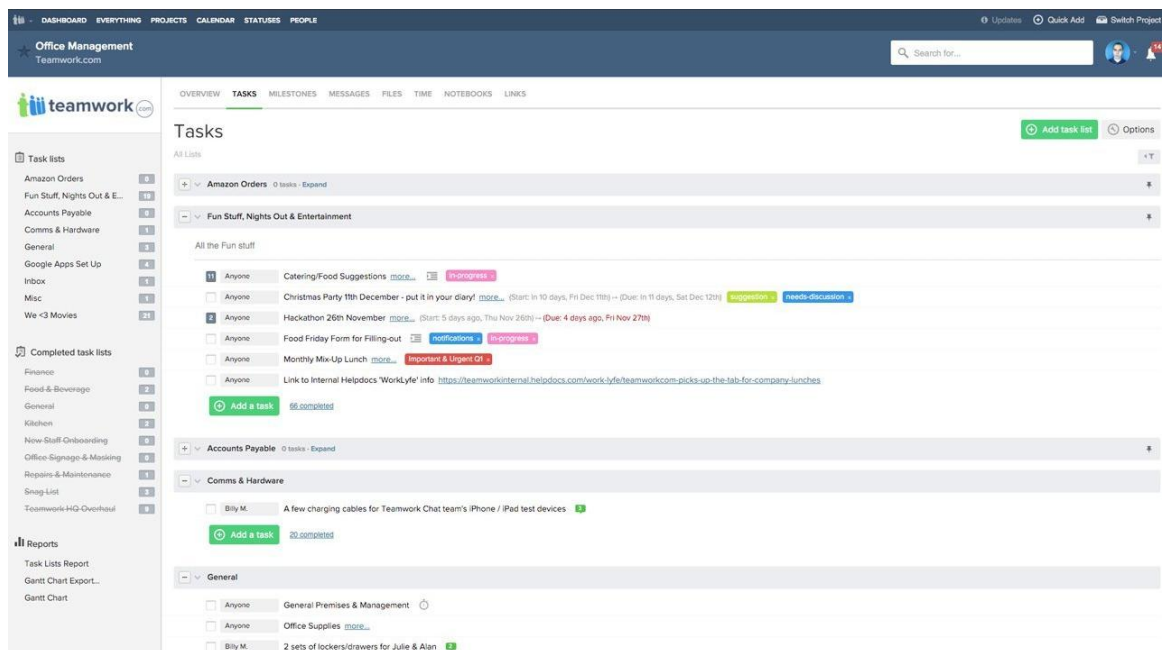


Рисунок 1.4 – Вікно спільної роботи команди у Teamwork

До переваг застосування цієї системи управління можливо віднести наступні: наявність центральної системи зберігання файлів; зручний спосіб для прикріплення файлів до задач; зручний інструмент для перегляду файлів не завантажуючи їх на комп'ютер; зручне керування за допомогою електронної пошти. Але система має і деякі недоліки: керування завданнями може бути неефективним і повільним; для розширення можливостей платформи потрібно вибрати один із платних тарифів; відсутність зручного механізму до редагування завдань (необхідно індивідуально натиснути на завдання); відсутня інтеграція з CRM.

Наступною було розглянуто систему управління проектами Wrike (рис. 1.5).

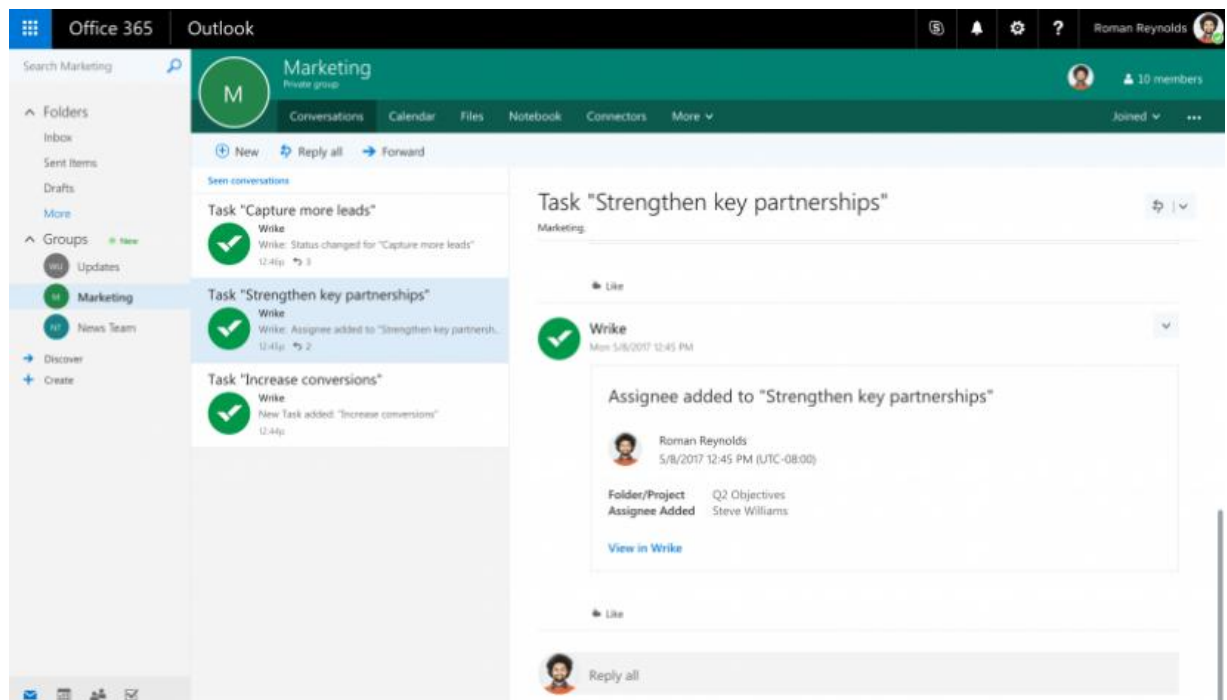


Рисунок 1.5 – Вікно повідомлень у додатку системи Wrike

Ця система допомагає командам керувати безмежними проектами, під-проектами, завданнями та даними. Wrike має всі важливі функції: дошки, діаграми Ганта, автоматична звітність, поновлення в режимі реального часу, розмежування прав доступу, поля, що настроюються, розміщення пріорите-

тів, облік робочого часу, спільне редагування документів та інших файлів, керування завантаженістю команди. Це допоможе координувати команди з будь-якого місця, оскільки ця система має мобільні версії. Недоліками використання системи Wrike є запропонований безкоштовний пробний період тільки 14 днів, який дозволяє створювати команду не більш ніж з 5 учасниками. Функції формування будь-яких звітів, створення графіків та діаграм Ганта доступні лише у платній версії, що не дозволяє оцінити переваги системи.

Ще одною з найпоширеніших систем управління проектами є система Asana, яка дозволяє створювати проекти, ставити завдання, призначати їм терміни, пріоритет-теги та відповідальних, відстежувати прогрес. Asana надає і тарифи з індивідуальними налаштуваннями під специфіку проекту та з брендуванням (рис. 1.6).

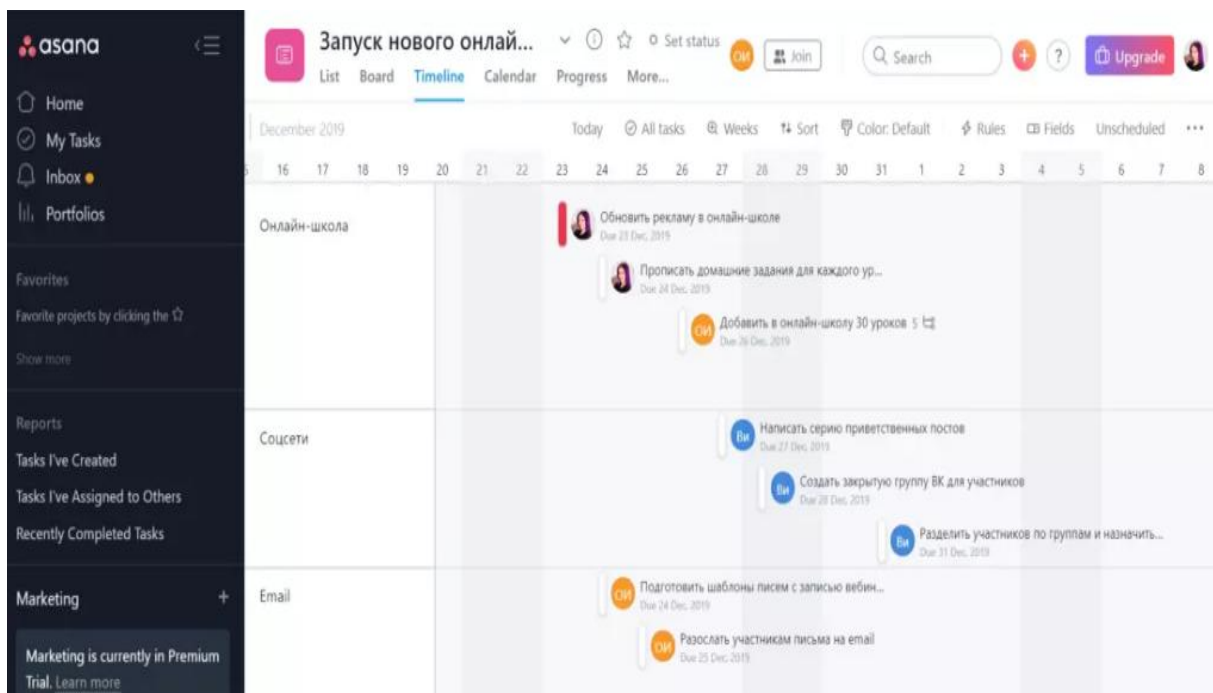


Рисунок 1.6 – Вікно сервісу управління в системі Asana

До переваг застосування системи управління проектами Asana можливо віднести наступні: можливість поділити складний проект на сфери діяльності; доступний високий ступінь деталізації проектів; інтуїтивно зрозумілий ін-

терфейс; інтеграція з Dropbox, Google Drive, Evernote та ін. Безкоштовна версія не надає для тестування весь функціонал, що не дозволяє протестувати систему. Безкоштовна версія не дозволяє: переглядати звіти; контролювати терміни, виконання завдань; здійснювати перегляд незавершених завдань; відсутня можливість стежити за оновленнями у завданнях; проводити роботи із формами; відсутня візуалізація робочих процесів; немає діаграми Ганта, подання проекту у часі.

Система управління проектами Basecamp – є ефективним і популярним інструментом для управління роботою компанії, забезпечує простий користувацький інтерфейс і легкі організаційні структури. Функціонує веб-версія та мобільні додатки. Включає списки справ із фірмовими звітами Basecamp, дошки, календар, хмарне зберігання файлів, груповий чат та клієнтський доступ. В системі формуються звіти щодо проектів у вигляді інформативних графіків. (рис. 1.7).

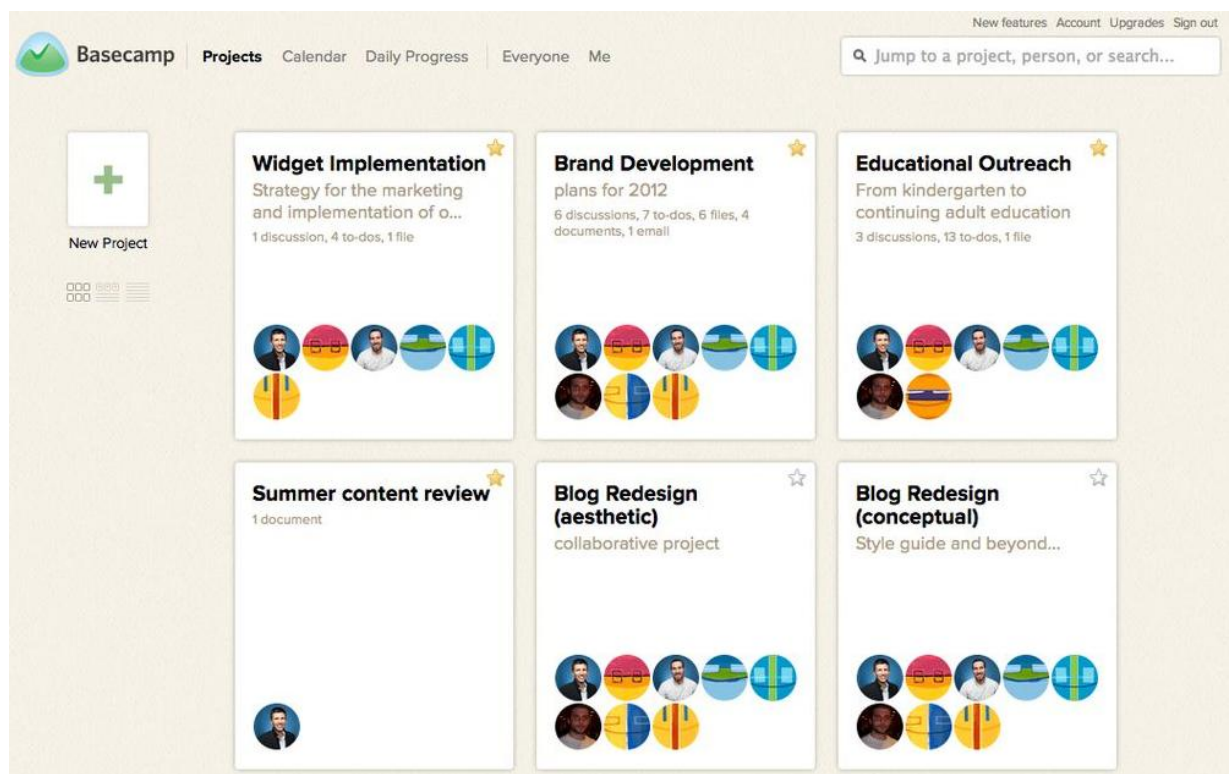


Рисунок 1.7 – Вікно управління проектом системи Basecamp

Перевагами системи Basecamp є наявність зручного функціоналу для зберігання файлів та швидка та зручна пошукова система для задач. До недоліків системи можливо віднести: відсутність функціоналу для створення взаємодносин між завданнями; неможливо увявити проект повністю; немає функціоналу для встановлення часу на виконання завдань, для створення звітності про виконані завдання. Система стає незручною із збільшенням кількості задач та не надає інструментів стратегічного планування.

Слід розглянути популярну систему управління проектами YouGile (рис. 1.8).

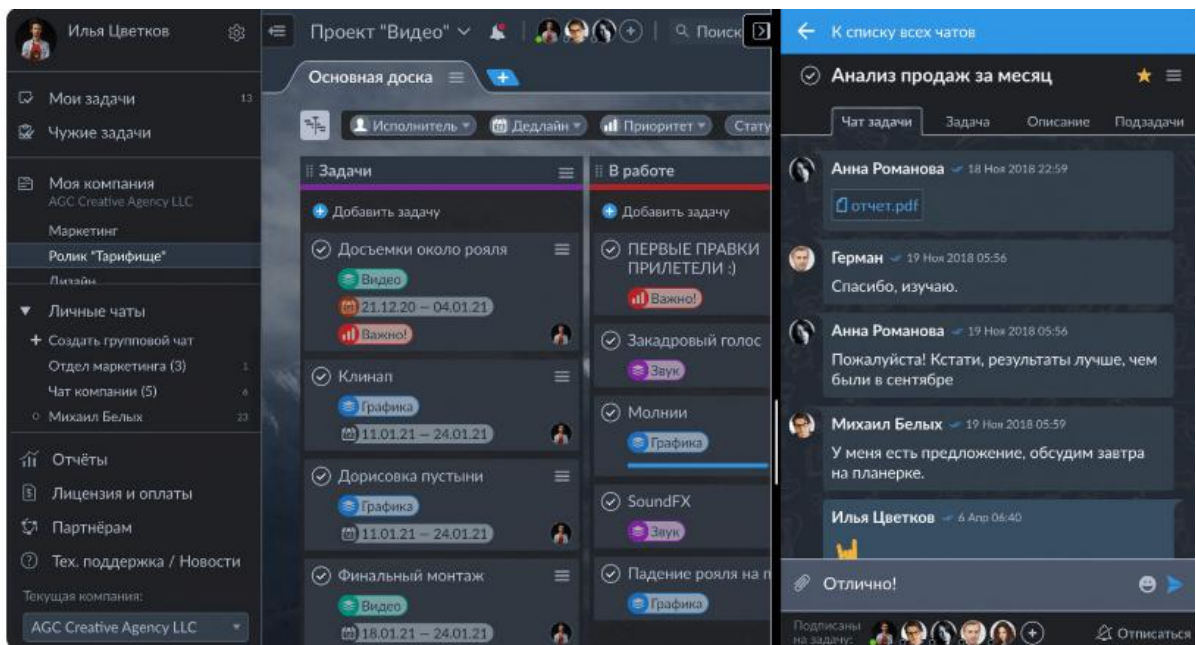


Рисунок 1.8 – Вікно системи управління проектом в YouGile

Це простий у використанні інструмент для великих команд, який дозволяє виконувати щоденну роботу із завданнями, використовуючи зручні інтерфейси і неформальні комунікації по завданням. Система підтримує гнучку систему звітів для створення прозорості та дозволяє здійснювати детальне налаштування прав доступу. Для команд менш ніж 10 співробітників

повністю безкоштовна. Доступне спілкування в чатах, обмін файлами без обмежень за вагою, створення будь-якої кількості проектів, дощок, завдань; налаштування прав, шарінг дощок та особистий планувальник завдань.

Недоліками системи є відсутність тегів та інтеграції з хмарними сховищами та сторонніми сервісами, наприклад OneDrive та OneNote.

Система Smartsheet дуже проста і зрозуміла система управління проектами і задачами з інтерфейсом типа Excel. Створює індивідуальні веб-форми, які роблять збір даних більш керованим (рис. 1.9).

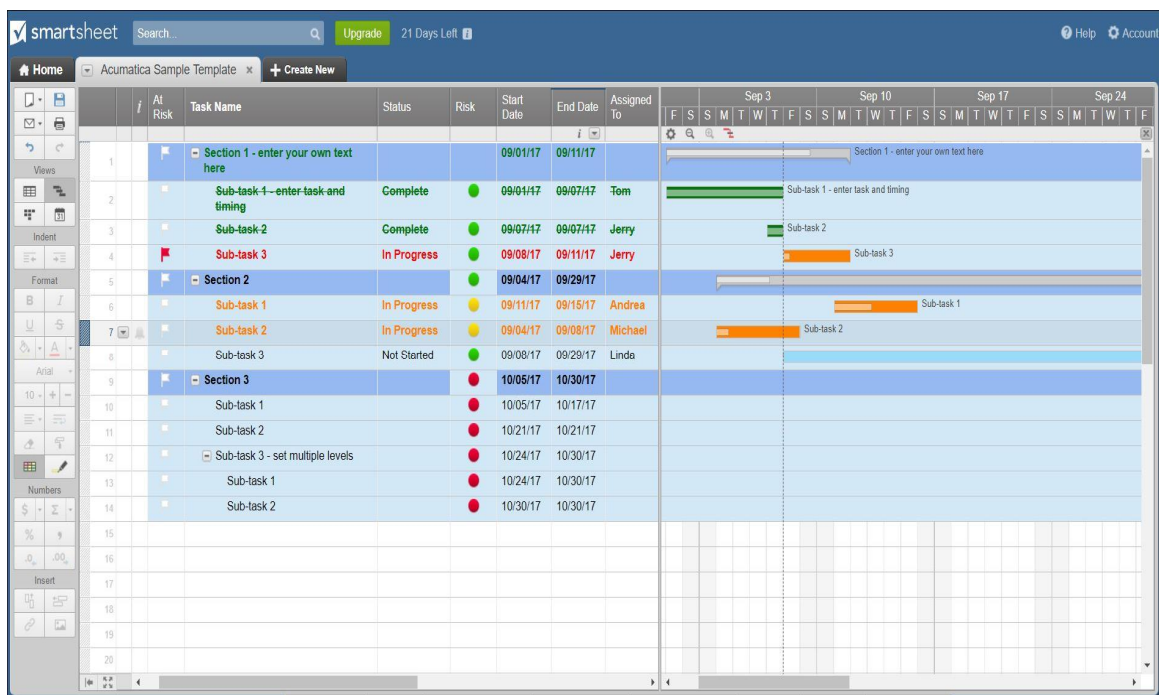


Рисунок 1.9 – Вікно системи управління проектом в Smartsheet

В систему можна імпортувати будь-який шаблон із Excel або таблицю з Google, перетворювати її на діаграми Гантта, легко автоматизувати багато процесів: збирання та обробку даних від віддалених співробітників, управління ресурсами та планування бюджету. Відмінно підходить для малих та середніх компаній. Серед слабких сторін можливо зазначити наступні: складне форматування таблиць займає дуже багато часу, оскільки необхідно задавати безліч умов, часто по одній – на кожен стовпець і кожен дію; незручна

фільтрація; не можна відкрити відразу кілька вкладок. Після переходу в систему з Excel працювати незручно та незвично, немає звичайних базових формул. Система також немає зручного розгалуження доступів до проектів для груп користувачів; немає зручної файлової системи. Всі можливості системи доступні тільки у платній версії.

Система Redbooth проста у використанні та гнучка, підходить для спільної роботи та комунікацій, можна влаштовувати відеоконференції. Використовують команди, які запрошують до робочого простору замовників (рис. 1.10).

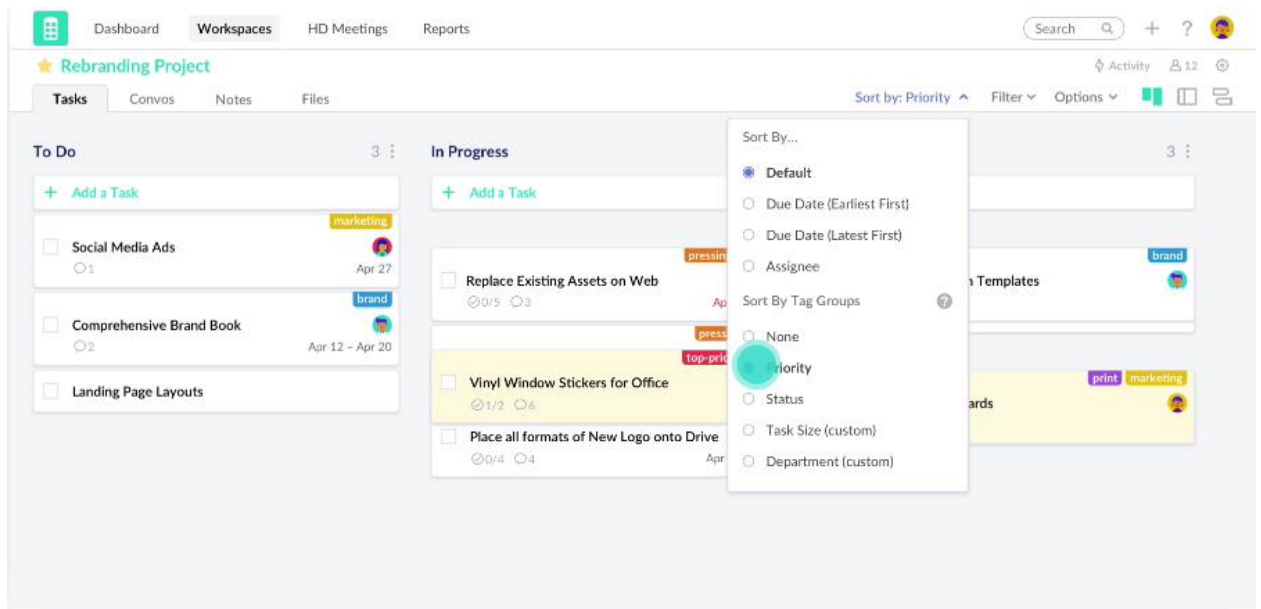


Рисунок 1.10 – Вікно системи управління Redbooth

Redbooth є прекрасним інструментом управління роботою для підприємств, регульованих законом США Sarbanes-Oxley, а також постачальниками медичних послуг та виробниками, які дотримуються правил HIPAA. Це комплексне рішення для співробітництва, яке дозволяє компаніям отримувати доступ, обмінюватися та керувати його вмістом за допомогою банківського захисту. Серед переваг системи можливо зазначити: текстові нотатки, особисті та групові чати, HD-відеоконференції, діаграми Гантта, загальний доступ

до файлів, відстеження часу, статистика, оцінка та візуалізація прогресу за проектами. До слабких сторін системи можливо віднести: дуже велику кількість повідомлень; не корисна для одночасного ведення кількох проектів; часті щодо проектів легко губляться.

Простота, універсальність та багатофункціональність – це переваги системи управління проектами Trello, яка надає інструменти для спільної роботи, який організовує проекти в дошки (рис. 1.12).

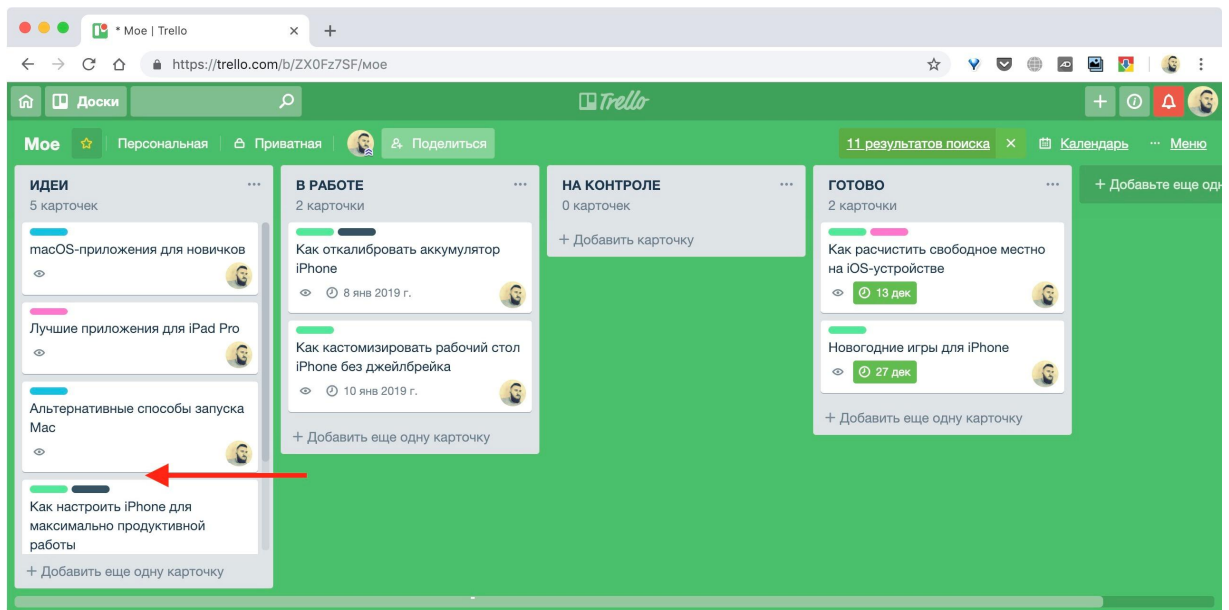


Рисунок 1.12 –Вікно системи Trello у форматі ToDo List

Це зручна система управління надає простий та універсальний онлайн-інструмент у вигляді веб-версії trello.com, мобільних програм, а також розширення для браузера Google Chrome. Також доступне підключення до Trello сторонніх сервісів та надбудов: календар, Gmail, Google.Drive, Dropbox та інші. Систему легко впровадити, інтерфейси візуально прості та барвисті, зрозумілі. Проте, вона не підходить для масштабних проектів. Сильними сторонами системи є: багато інтеграцій, основні операції із завданнями дуже швидко і легко проводити, інформативні дошки, гнучке настроювання карток. Ідеально підходить для простих лінійних завдань. Легко ділитися карт-

ками, запрошувати сторонніх людей на дошки, додавати швидкі нотатки та показувати їх колегам. У безкоштовній версії немає обмежень щодо кількості користувачів та персональних дощок, але до кожної дошки можна підключити тільки один сервіс.

До недоліків системи можливо віднести відсутність за замовчуванням діаграми Ганта, неможливо оцінити прогрес у часі, хоча це вирішується підключенням плагіна Planuway Team Planner. Система незручна для великих проектів: у великій кількості карток та списків легко заплутатися. Чим більше проектів ведеться, тим складніше в них орієнтуватися, інтерфейс швидко захаращується, довго шукати потрібну дошку. Немає врахування часу за завданнями у базовому функціоналі, мало можливостей для класифікації карток. Вести роботу у великих командах важко, у завданнях безладдя. Якщо ведеш серйозні проекти – стандартних звітів недостатньо.

Jira – сервіс управління проектами, належить компанії Atlassian, створений спеціально для розробників ІТ-проектів. Однак він застосовується не тільки в ІТ-компаніях. Інформаційна панель JIRA складається з багатьох корисних функцій, які дозволяють легко вирішувати проблеми (рис. 1.13).

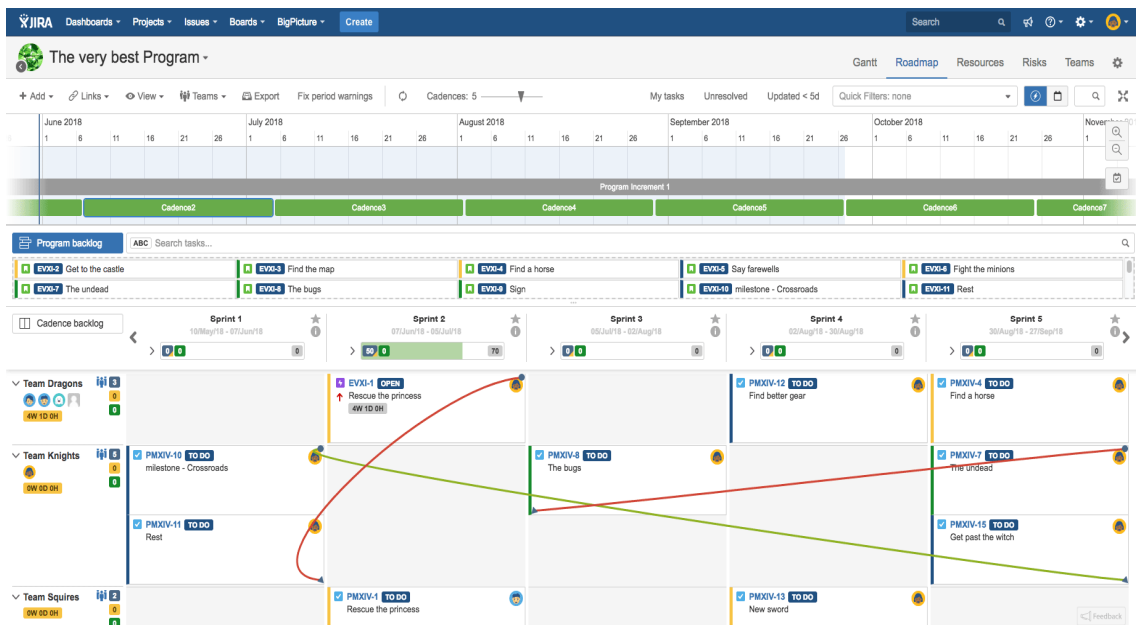


Рисунок 1.13 – Вікно календарного плану системи JIRA

Платформа адаптована для командної роботи за методологіями Agile, Kanban, Scrum. Дуже зручний інструмент для відстеження помилок (багів), тайм-трекінгом та звітністю зручніше займатися саме в Jira.

Серед переваг системи Jira можливо зазначити наступні: простий та зрозумілий інтерфейс; канбан або scrum-дошки; можливість створення звітів; можливість підключення сторонніх сервісів, більш ніж 1500 розширень: календар, діаграми, Excel-таблиці, Dropbox, Trello; інтуїтивна зручність для професійних розробників. Jira має велику спільноту яка створює нові інструменти, має гнучку систему для створення груп користувачів, дозволяє створювати взаємопов'язані завдання та календарний план. До слабких сторін системи можливо віднести наступні: із збільшенням задач використання системи дуже ускладнюються, потребує перед її використанням попереднього навчання. Спочатку ця система управління проектами розрахована на спеціалізовані команди розробників, включає можливості прив'язки до коду які не завжди необхідні для інших проектів опції.

У результаті проведення дослідження та аналізу функціональних можливостей існуючих систем були виявлені позитивні практики, які доцільно було б застосувати при розробці системи управління командною розробкою IT-проектів, серед них сформулюємо наступні:

- розподілення прав доступу до функціоналу системи;
- можливість створювати задачі та закріплювати їх за конкретним виконавцем в команді;
- можливість комунікації із членами команди;
- можливість створювати календарні плани;
- можливість змінювати статус готовності завдань;
- можливість представити задачі у формі ToDo List;
- наявність діаграм зі статистикою проекту;
- отримання нотифікацій про зміни у проекті.

Проведений аналіз систем управління проектами довів, що велика кількість систем із додатковим та поглибленим у різні етапи функціоналом не

безкоштовні та потребують попередньої підготовки перед їх застосуванням, а безкоштовні версії надають обмежений функціонал, іноді навіть не достатній для тестування системи. Також усі функціональні можливості не містяться в одній системі управління проектами, тому часто трапляється що команди покупають ліцензії для кількох програмних забезпечень. Крім того, визначено, що системи управління проектами, метою яких є управління саме командою розробників програмних продуктів, мають деякі специфічні, властиві тільки для ІТ-проектів функції.

1.7 Постановка завдання

Метою створення системи управління командною розробкою ІТ-проектів є забезпечення розробників ІТ-проектів різної складності автоматизованими засобами та інструментами, що надають можливість якісної розробки програмних продуктів та спрямовані на отримання переваг при здійсненні командної роботи над проектами.

Розроблена система управління командною розробкою ІТ-проектів дозволити користувачу, або групі користувачів планувати, стежити, та виконувати поставлені задачі для успішного завершення проекту.

Для створення програмної системи управління командною розробкою ІТ-проектів необхідно реалізувати у системі наступні функціональні можливості.

Загальні можливості системи:

- можливість реєстрації головного користувача-адміністратора;
- можливість створення команди з користувачів.

Функціонал управління проектом:

- користувачі, які мають певні привілеї повинні мати доступ до функціоналу управління проектами та їх завдань;
- на сторінці системи управління проектами користувачі повинні мати змогу регулювати план виконання цих завдань (складати календар-

ний план);

- користувачі повинні мати змогу виділити бюджет для кожної задачі;
- користувачі повинні мати змогу стежити за етапом виконання задач;
- після завершення задачі користувач повинен мати змогу сформувавши звіт та вивантажити його у формі документа.

Функціонал звітності:

- система повинна нотувати та стежити за усіма змінами в проєкті;
- система повинна формувати та надавати підсумкову документацію про виконаному проєкту.

Функціональність зв'язку з системою управління версій Git:

- система повинна надавати змогу користувачу зберігати посилання на зміни у проєкті які були внесені до системи контролю версій Git;
- головний користувач або члени команди з наданими привілеями повинні регулювати чи є обов'язковим збереження посилання до системи управління версій Git для завершення завдання розробником.

Функціонал для користувачів, які мають привілеї тестування завдань:

- система повинна мати функціонал для етапу після завершення внесення змін розробником для перевірки виконаної роботи іншими користувачами, які мають певні привілеї;
- користувач, який створює тестування повинен мати змогу перевести завдання у завершений статус;
- система повинна мати правила для переведення завдання до завершеного статусу, наприклад очікувати обов'язковий звітний матеріал від користувача, який провів тестування для формування звіту.

Система управління командною розробкою ІТ-проєктів являє собою ПЗ, яке забезпечує етапи планування та реалізації проєкту. Система надасть можливість учасникам проєкту комунікації з командою на кожному етапі; можливість розподіляти привілеї користувачам, та надійно нотувати та зберігати усі дії та рішення.

2 АРХІТЕКТУРА ТА ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ

2.1 Вибір архітектурного рішення для створення системи

Застосування архітектури «клієнт-сервер» передбачає поділ процесів надання послуг та надсилання запитів на них на різних комп'ютерах у мережі, кожен з яких виконує свої завдання незалежно від інших.

В архітектурі «клієнт-сервер» кілька віддалених систем комп'ютерів-клієнтів надсилають запити та отримують послуги від централізованого службового комп'ютера – сервера, який також може називатися хост-системою. Для полегшення взаємодії клієнтів з сервером, клієнти надають користувачам зручний інтерфейс.

До дворівневої архітектури клієнт-сервер слід відносити таку, в якій прикладні програми зосереджені на сервері додатків (Application Server), наприклад, сервері CRM, а на робочих станціях знаходяться програми-клієнти, які надають для користувачів інтерфейс для роботи з додатками на загальному сервері.

У трирівневій архітектурі сервер баз даних, файловий сервер та інші є окремим рівнем, результати роботи якого використовує сервер додатків. Логіка даних та бізнес-логіка знаходяться на сервері додатків. Всі звернення клієнтів до бази даних відбуваються через проміжне програмне забезпечення (middleware), яке знаходиться на сервері програм, що підвищує гнучкість роботи та продуктивність системи, у якій застосовується таке архітектурне рішення. В багаторівневій архітектурі кілька серверів використовують результати роботи один одного, а також дані від різних серверів баз даних, файлових серверів та інших видів серверів [12]¹⁾.

Перевагою використання багаторівневої архітектури є гнучкість надан-

¹⁾ [12] Эспозито. Разработка современных веб-приложений. Анализ предметных областей и технологий. М.: Диалектика-Вильямс, 2017. 464 с.

ня послуг, які можуть бути комбінацією роботи різних додатків серверів різних рівнів та елементів цих додатків. Недоліком використання саме такого архітектурного рішення є складність, багатокomпонентність такої архітектури.

В результаті проведеного дослідження та аналізу предметної області було з'ясовано що усі існуючі системи управління проектами реалізовані за допомогою web-архітектури клієнт-сервер. Схема функціонування архітектурного рішення за технологією клієнт-сервер, яка обрана для реалізації системи управління командною розробкою ІТ-проектів наведено на рисунку 2.1.



Рисунок 2.1 – Схема архітектурного рішення за технологією клієнт-сервер

Клієнт-серверна архітектура набула своєї популярності завдяки динамічному розвитку мережі Інтернет та зосередження значної частини інформації в базах даних на серверах. Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів. Така архітектура визначає такі типи компонентів:

- набір серверів, які надають інформацію або інші послуги програмам,

які звертаються до них;

- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Правила взаємодії між клієнтом і сервером називаються протоколом обміну. Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій [12]¹⁾:

- представлення даних, яка по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачу і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку додатку і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Для реалізації системи управління командною розробкою ІТ-проектів була обрана клієнт-серверна архітектура, де в якості сервера виступає АРІ.

Сьогодні переважна кількість додатків реалізована за допомогою Web-технологій, що зумовлено наявністю зручних інструментів, які дозволяють у швидкий спосіб створити Web-додаток. Проаналізувавши існуючий ринок програмного забезпечення були обрані наступні технології.

Технології для створення Web-серверу:

- Asp.Net який входить до платформи .Net Framework з мовою програмування C#;
- Dapper – CRM яка дозволяє здійснювати роботу з базою даних;
- MSSQL Server, як система для створення та роботи с базами даних.

Технології для створення Web-клієнту:

- Angular – Javascript фреймворк для створення Web-клієнтів за допо-

¹⁾ [12] Д. Эспозито. Разработка современных веб-приложений. Анализ предметных областей и технологий. М.: Диалектика-Вильямс, 2017. 464 с.

могою компонентів з використанням мови програмування TypeScript;

- HTML 5 – стандартна мова для створення web-сторінок;
- CSS3 – каскадна таблиця стилів, яка використовується для задання стилістики для блоків створених на мові HTML;
- Bootstrap – бібліотека стилів, яка містить у собі готові стилі для швидкого оформлення Web-сторінок;
- NodeJs – сервер для роботи із API-сервером який використовується для запуску Angular-клієнту.

Для реалізації додатка-клієнта системи управління командною розробкою IT-проектів були обрані такі технології, як Angular, HTML5, CSS3, що дозволяє швидко та якісно створити клієнтську частину додатку.

2.2 Програмні засоби розробки серверного додатку системи

Для реалізації системи управління командною розробкою IT-проектів був обраний .NET Framework, який має усі необхідні інструменти для створення Web-серверу.

.NET Framework являє собою платформу для розробки додатків для веб, Windows, Windows Phone, Windows Server і Microsoft Azure. Він складається з спільної мови (CLR) і бібліотеки класів .NET Framework, яка включає широкий спектр функціональних можливостей і підтримку багатьох галузевих стандартів. .NET Framework надає багато послуг, включаючи керування пам'яттю, безпеку типів і пам'яті, мережу і розгортання додатків. Крім того використання цього фреймворку надає прості у використанні структури даних і API, які абстрактно використовують операційну систему Windows нижнього рівня. Забезпечується можливість використовувати різні мови програмування з платформи .NET Framework, включаючи C #, F # і Visual Basic. Всі програми, написані для платформи .NET, незалежно від мови програмування, виконуються CLR. Всі версії включають CLR. CLR транслює

початковий код в байт-код на мові IL, реалізація компіляції якого компанією Microsoft називається MSIL [13]¹⁾.

.NET Framework також надає MSIL – програмам (а отже, і програмам, написаним на мовах високого рівня, що підтримують .NET Framework) доступ до бібліотеки класів .NET Framework, або так званою .NET FCL (Framework Class Library) платформи, які входять до складу .NET Framework наведено на рисунку 2.2.

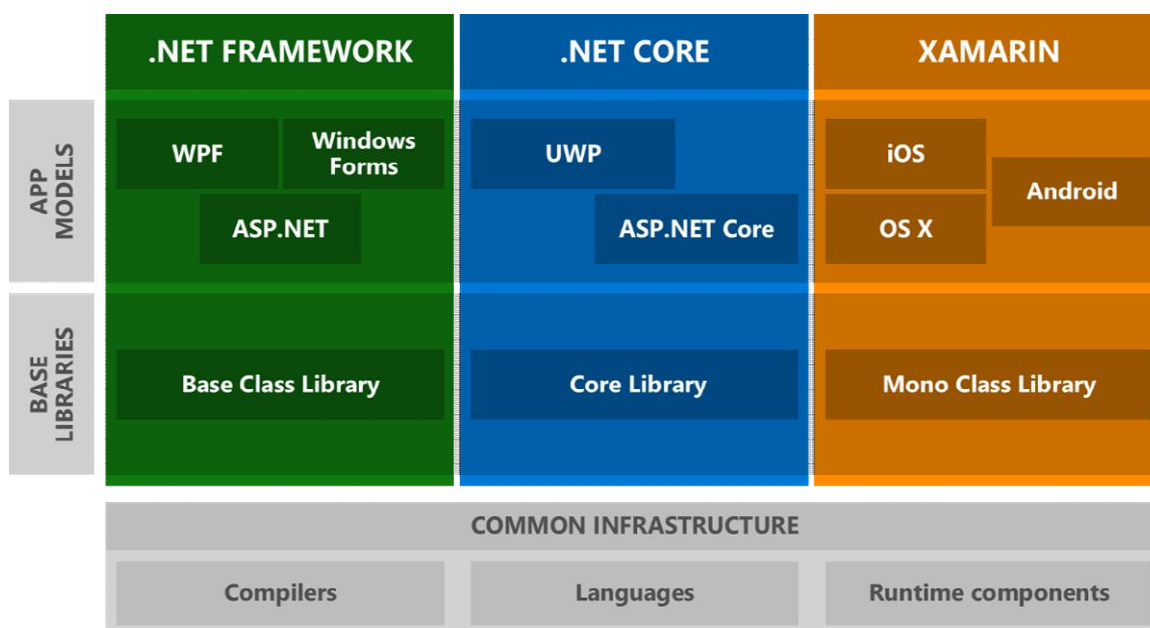


Рисунок 2.2 – Перелік платформ, що надає .NET Framework

Common Language Runtime (CLR), компонент віртуальної машини Microsoft .NET Framework, керує виконанням програм .NET. Компіляція вчасно перетворює керований код в машинні інструкції, які потім виконуються на процесорі комп'ютера. CLR надає додаткові послуги, включаючи управління пам'яттю, безпеку типу, обробку виключень, збір сміття, безпеку та управління потоками.

Середовище CLR є реалізацією специфікації CLI (Common Language

¹⁾ [13] А. Фримен. ASP.NET MVC 5 с прикладами. М.: Вільямс, 2015. 736 с.

Infrastructure), специфікації загальномовної інфраструктури, компанією Microsoft [14]¹⁾.

CLR реалізує віртуальну систему виконання (VES), як визначено в стандарті спільної мовної інфраструктури (CLI), спочатку розробленому самою компанією Microsoft. Державний стандарт визначає специфікацію спільної мовної інфраструктури. Віртуальна машина CLR дозволяє програмістам забути про багато деталей конкретного процесора, де виконуватиметься програма. Схема обробки та виконання CLR сумісних мов програмування наведено на рисунку 2.3.

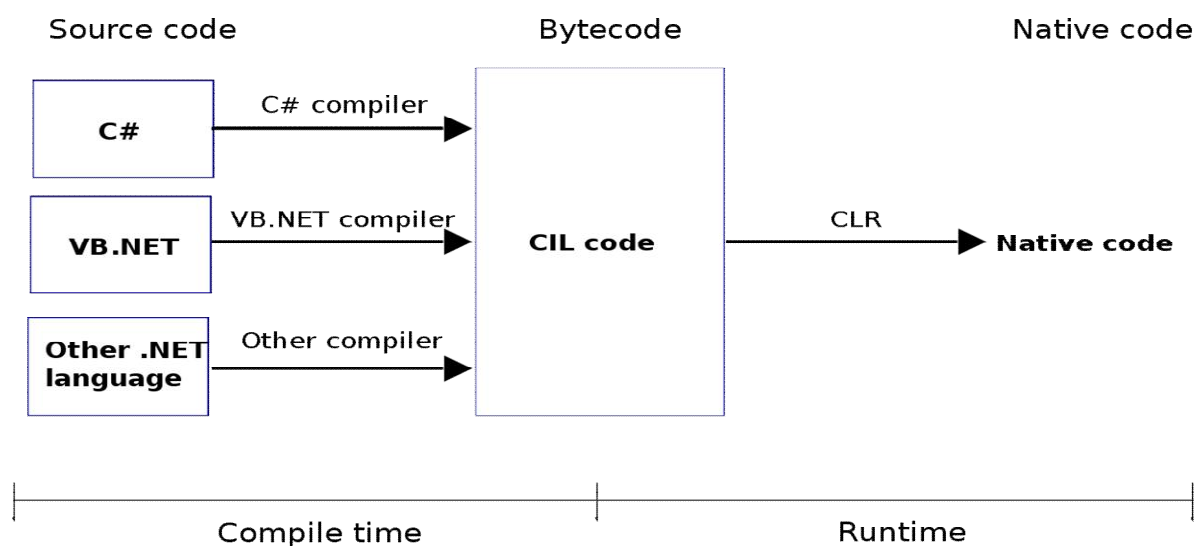


Рисунок 2.3 – Послідовність виконання CLR-сумісних мов програмування

Для веб-розробки та створення динамічних веб-сторінок використовувалась платформа веб-додатків з відкритим вихідним кодом ASP.NET. Вона була розроблена корпорацією Майкрософт, щоб дозволити програмістам створювати динамічні веб-сайти, веб-програми та веб-служби. Наступником ASP.NET є ASP.NET Core. Це повторна реалізація ASP.NET як модульної веб-структури, разом з іншими структурами, такими як Entity Framework.

¹⁾ [14] Д. Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. М.: Питер, 2017. 896 с.

Новий фреймворк використовує нову платформу для відкритих .NET Compiler Platform і є крос-платформним. ASP.NET MVC, веб-API ASP.NET і веб-сторінки ASP.NET об'єдналися в уніфікований MVC 6/

C# – це мова програмування, створена Microsoft для використання на платформі .NET. Це мова об'єктно-орієнтованого програмування і статично типізована. Ім'я цієї мови походить від C, з якого вона успадковує аналогічний синтаксис. C# був створений Microsoft і стандартизований ISO і ECMA. Він був офіційно випущений в 2002 році. З тих пір на цій мові з'явилося безліч поліпшень. Розробники програмного забезпечення C# часто називають розробниками .NET, оскільки мова C# майже виключно використовується з .NET Framework. Це досить популярна мова, який зазвичай входить в п'ятірку кращих на різних графіках популярності. Вона найбільш часто використовується в розробці корпоративного програмного забезпечення, але також має процвітаючу екосистему з відкритим вихідним кодом. Мова C# була створена для роботи в CLI (Common Language Infrastructure) і використовує .NET Framework. Вона включає декілька парадигм програмування, таких як об'єктно-орієнтована і функціональний. Мова скомпільована і статично типізована, що означає, що тип всіх змінних перевіряється компілятором при компіляції програми. Сьогодні C# є мовою з відкритим вихідним кодом і працює на крос-платформенном .NET Core. Специфікація C# визначає мінімальний набір бібліотек типів і класів, на який має розраховувати компілятор. На практиці, C# найчастіше використовується з якоюсь реалізацією Common Language Infrastructure (CLI), яка стандартизована як ECMA-335 CommonLanguageInfrastructure (CLI) [14]¹⁾.

Для реалізації системи управління командною розробкою ІТ-проектів була обрана платформа .Net Framework и мова програмування C#, що дозволило створити серверний додаток який буде відповідати усім сучасним стандартам.

¹⁾ [14] А. Фримен. ASP.NET MVC 5 с примерами. М.: Вильямс, 2015. 736 с.

2.3 Програмні засоби та технології створення web-клієнтів

Сьогодні Web-клієнти нічим не поступаються у можливостях іншим платформам, їх перевагою є тим що в загалом вони не потребляють великих ресурсів, для їх використання вам потрібен лише браузер, а усі складні операції виконує сервер. Раніше Web-клієнти були дуже простими та склалися лише із простих сторінок. Сьогодні завдяки мові сценаріїв JavaScript та великій кількості прибічників клієнт-серверних рішень великі компанії створюють та розвивають можливості веб-технологій, створюючи потужні інструменти для прискорення розробок та підвищення якості додатків які б відповідали сучасним реаліям. Один із таких інструментів є Angular, яка підтримується компанією Google.

Angular – це платформа, яка дозволяє легко створювати програми в інтернеті. Angular поєднує в собі декларативні шаблони, ін'єкції залежностей, набір найкращих готових рішень для типових задач різного рівня складності. Angular надає розробникам можливість створювати програми, які працюють у мережі, на мобільних пристроях або на робочому столі [15]¹⁾.

Крос-платформним JavaScript середовищем виконання, яке виконує код JavaScript поза браузером є Node.js з відкритим вихідним кодом. Node.js дозволяє розробникам використовувати JavaScript для написання інструментів командного рядка і для сценаріїв на стороні сервера – запуск скриптів на стороні сервера для створення динамічного вмісту веб-сторінки до того, як сторінка буде відправлена до веб-браузера користувача. Отже, Node.js представляє парадигму «скрізь JavaScript», що об'єднує розробку веб-додатків навколо однієї мови програмування, а не на різних мовах для серверних і клієнтських скриптів [16]²⁾.

JavaScript – це мова сценаріїв або програмування, яка дозволяє реалізо-

¹⁾ [15] А. Фримен. Angular для професіоналов. М.: Пітер, 2016. 800 с.

²⁾ [16] Про Node.js. URL: <https://nodejs.org/uk/about> (дата звернення 26.10.2021).

увати складні речі на веб-сторінках – кожен раз, коли веб-сторінка повинна відобразити більше, ніж просто статичну інформацію для перегляду – наприклад відображення своєчасного оновлення вмісту, інтерактивних карт, анімована 2D / 3D-графіка, і т.д.

TypeScript – це той самий же JavaScript, це означає, що перш за все він містить всі функціональні можливості JavaScript, а потім вже усі нові функції. Таким чином, будь-яка програма, написана на JavaScript, також запускатиметься, як і очікувалося, у TypeScript. Насправді, TypeScript просто компілюється у JavaScript. TypeScript пропонує нам більше контролю над нашим кодом через анотації типу, інтерфейси та класи. TypeScript був створений Microsoft і був випущений у 2012 році після двох років розробки. Він була створений, щоб дозволити додаткову статичну перевірку типу, яка була б особливо корисною при розробці великомасштабних програм. Цікаво, що однією з причин, чому Microsoft розробила TypeScript, було те, що їхні внутрішні команди мали проблеми зі створенням масштабу JavaScript для власних проєктів корпорації Майкрософт, зокрема, команди, яка працювала над Bing Maps [17]¹⁾.

HTML – мова розмітки гіпертекстових документів, стандартна мова розмітки для створення веб-сторінок і веб-додатків. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовано сторінку. Замість стандартної мови сценаріїв була обрана мова TypeScript як чудово інтегрується у Angular, яка представляє на фінальній стадії той самий JavaScript код, але дозволяє мати строгу типізацію та перевірку компілятором ще на етапі написання коду.

¹⁾ [17] TypeScript. URL: <https://ru.wikipedia.org/wiki/TypeScript> (дата звернення 02.11.2021).

2.4 Система управління базою даних

Для роботи із даними системи управління проектами був обраний продукт компанії Microsoft MSSQL Server, що легко інтегрується у .Net Framework та на сьогодні являються одним із най потужних додатків для створення та роботи над базами даних.

Microsoft SQL Server – це система управління реляційними базами даних, або RDBMS, яка підтримує широкий спектр обробки транзакцій, бізнес-аналітики та аналітичних додатків у корпоративних IT-середовищах. Це одна з трьох провідних технологій баз даних, разом з Oracle Database та IBM DB2.

Програмне забезпечення RDBMS, Microsoft SQL Server побудований на основі стандартизованої мови програмування SQL, адміністратори баз даних та інші IT-спеціалісти використовують для управління базами даних і запиту даних, які вони містять. SQL Server прив'язаний до Transact-SQL (T-SQL), реалізації SQL від Microsoft, що додає набір розширень власного програмування до стандартної мови. Оригінальний код SQL Server був розроблений у 1980-х роках колишньою компанією Sybase Inc., яка тепер належить SAP. Sybase спочатку побудував програмне забезпечення для роботи на системах Unix і міні комп'ютерних платформах. Компанія Microsoft і корпорація Ashton-Tate, яка тоді була провідним виробником ПК, створила першу версію Microsoft SQL Server, розроблену для операційної системи OS / 2 і випущену в 1989 році [18]¹⁾.

T-SQL (Transact-SQL) – це набір розширень програмування від Sybase і Microsoft, які додають до структурованого запиту мови (SQL) кілька функцій, включаючи керування транзакціями, виключення і обробку помилок, обробку рядків та оголошення змінних. Ідентифікатори T-SQL використовуються у всіх базах даних, серверах і об'єктах баз даних в SQL Server. До них відно-

¹⁾ [18] І. Бен-Ган. Microsoft SQL Server 2012. Основы T-SQL. М.: ЕКСМО, 2015. 400с.

сяться таблиці, обмеження, збережені процедури, види, стовпці і типи даних. Кожні ідентифікатори T-SQL мають унікальне ім'я, присвоюються коли створюється об'єкт і використовуються для ідентифікації об'єкта.

Для реалізації системи управління командних IT-проектів була обрана база даних MSSQL, яка чудово інтегрується у продукти компанії Microsoft, та є одним із лідерів посеред схожих продуктів.

2.5 Вибір засобів для створення Unit-тестів

Unit тестування – це підхід до тестування програмного забезпечення, який виконується під час розробки для тестування малих компонентів будь-якого програмного забезпечення. Це означає, що замість тестування великого модуля за один раз ви перевіряєте невелику частину цього модуля. Невеликий компонент в тому сенсі, що він може бути будь-якою функцією, будь-якою властивістю або будь-яким класом, який обробляє конкретну функціональність.

Unit-тести допомагають нам перевірити і з'ясувати, чи виконує вона індивідуальне завдання добре чи ні. В основному це гарантує нам, що система розроблена відповідно до вимог і допомагає нам з'ясувати помилки перед тим, як покласти фактичний код у середовище QA. Коли ми говоримо про Unit-тестування у розробці програмного забезпечення, у нас є багато переваг, якщо ми пишемо їх одночасно з фактичним кодом наступним чином:

- Unit-тести виявляють, якщо щось не так з дизайном програмного забезпечення, функціональністю програмного забезпечення, і якщо він не працює, як очікувалося, відповідно до вимог;
- завдяки Unit-тестам, ми можемо отримати або вловити помилки або проблеми на початку розгортання коду в іншому середовищі;
- додавання Unit-тестів є частиною циклу розробки програмного забезпечення, що робить процес більш гнучким. Якщо щось змінюється зі старим кодом іншими розробниками та трапляються помилки

або проблеми, то Unit-тести показують, ваш код порушує функціональність чи ні;

- Unit-тести допомагають нам завчасно зменшити кількість помилок. Якщо розробник зрозумів вимоги чітко і, якщо він пише код для реалізації вимог разом з Unit-тестами, то, напевно, він зможе зловити і виправити більшість помилок під час розробки;
- Unit-тестування допомагає іншим розробникам також зрозуміти фактичну функціональність, компонентів, які доступні в системі.

Отже написання Unit-тестів допомагає налагодити розробку програмного забезпечення, спланувати, як буде реалізовуватися система, знаходити помилки на ранніх етапах розробки і багато іншого. Але слід визначити й деякі недоліки: [19]¹⁾

- це збільшує тривалість завершення проекту, оскільки написання Unit тестування разом з реалізацією фактичних потреб потребує додаткового часу;
- це збільшує вартість проекту, оскільки відомо, що офактурювання в індустрії ПЗ базується на тривалості часу; якщо час збільшить, вартість також збільшиться. Розробка модульного тестування займає набагато більше часу, тому це збільшує тривалість і вартість проекту;
- іноді потрібно написати Unit-тести, де вимоги незрозумілі або змінюються занадто швидко, або треба чекати, поки не будуть встановлені остаточні вимоги або писати і змінювати тестові випадки кожного разу, якщо зміниться вимоги;
- покриття коду не на 100%, тому що, в будь-якому сценарії, розробник не можемо написати блок-тест для функціональності. Наприклад, якщо є алгоритм, який змінюється на основі даних;
- написання Unit-тестів і підтримання його навантажує розробників

¹⁾[19] А. Фримен. ASP.NET Core MVC с примерами на C # для профессионалов. М.: Диалектика-Вильямс, 2016. 992 с.

додатковою, монотонною роботою;

- блок тестування не може виявити всі типи помилок. Може бути багато помилок, що залишаться в системі після Unit модулів, які можна знайти за допомогою інтеграційних тестів або тестуванні End to End.

Для реалізації системи управління командною розробкою IT-проектів була обрана бібліотека X-Unit, яка написана на мові програмування C#, та легко інтегрується у проект.

3 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ

3.1 Загальні вимоги до системи управління проектами

При здійсненні проектування системи управління командною розробкою ІТ-проектів були визначені вимоги, яким повинна задовольняти розроблювальна система. Система має надати можливість створювати команди, створювати проекти та планувати задачі. У системі повинні бути присутні графіки з інформацією про стан виконання проекту, та представлення задач у вигляді календарного плану.

Визначимо ряд загальних вимог до реалізації програмного забезпечення системи управління командними ІТ-проектами:

- введення даних з клавіатури;
- застосування браузера для доступу до проекту;
- передачу даних по мережі;
- обробка запитів користувача за допомогою сервера, який працює за допомогою протоколу http;
- захист доступу до інформації за допомогою можливості створювати команди та розгалужувати ролі доступу;
- адаптивних інтерфейс для можливості отримати доступу з телефону.

Система управління ІТ-проектами повинна мати головного користувача, який би мав усі права на дії в проекті, також система повинна мати стандартні ролі для користувачів, які можна призначити для того щоб прискорити процес створення та управління командами.

Для того щоб розробити базу даних для системи управління командними ІТ-проектами перш за все треба продумати, як користувачі будуть взаємодіяти із системою, та які саме їх функціональними можливості повинні впливати на систему.

В результаті розробки система управління ІТ-проектами буде створений Web-сервер, який буде обробляти запити користувача за допомогою API архітектури, буде розроблений клієнтський додаток, що реалізує інтерфейс

для доступу користувача до системи. Користувач який перший реєструється в системі отримує усі права на проєкти, на далі система буде пропонувати стандартні ролі такі як «Розробник», «Тестувальник», «Програмний менеджер», а також можливість регулювати ролі самостійно.

3.2 Функціональні можливості користувачів web-системи

Головний користувач системи управління командною розробкою ІТ-проєктів має усі права у системі, та може регулювати їх. До його функціональних можливостей входять такі пункти:

- створювати команди;
- створювати проєкти;
- запрошувати користувачів до проєкту;
- регулювати права користувачів на проєкті;
- створювати задачі;
- регулювати статус готовності задачі;
- мати доступ до панелі управління проєктом;
- додавати та коментувати задачі;
- назначати задачі користувачам.

Розглянемо ролі членів команди у проєкті та їх функціональні можливості. «Програмний менеджер» – це стандартна роль яку пропонує система управління проєктами, це роль яка дозволяє управляти проєктом, що дозволяє отримувати наступні функціональні можливості:

- створення задач;
- додавати коментарі та матеріали до задачі;
- регулювати статус готовності задачі;
- завершити задачу;
- створювати та переглядати звіти задач;
- назначати задачі користувачам;
- мати доступ до панелі управління проєктом.

«Розробник» – стандартна роль яку пропонує система управління проектами, це роль яка задається користувачам, які повинні мати доступ до задач та регулювати їх статус, але не мають прав для управління проектом загалом. Такі користувачі можуть: переглядати задачі; залишати коментарі та додавати матеріали до задач; регулювати статус виконання задачі.

«Тестувальник» – це стандартна роль яку пропонує система управління проектами, що дозволяє виконувати необхідні дії для того щоб перевірити виконану роботу, та остаточно завершити виконання задачі. Користувачі даної ролі мають такі привілеї: переглядати задачі; залишати коментарі та додавати матеріали до задач; завершити виконання задачі.

Роль для користувача можна назначити у будь-який момент, головний користувач може як змінювати існуючі ролі на проекті так і створювати нові.

3.3. Проектування бази даних системи

База даних для створення системи управління командною розробкою ІТ-проектів використовуються в клієнт-серверній архітектурі системи та знаходиться на стороні серверу. Користувач-клієнт буде слати запити до сервера, а вони будуть оброблятися використовуючи сервер баз даних.

Для створення системи управління командною розробкою ІТ-проектів була обрана реляційна СУБД MSSQL.

Реляційна база даних – це база даних, в якій усі дані, доступні користувачу, організовані у вигляді таблиць, а всі операції над даними зводяться до операцій над таблицями. Для представлення реляційних баз даних розроблена формальна теорія баз даних, теоретичну основу якої складає алгебра та математична логіка. Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив. Реляційна модель бази даних для зберігання і організації інформації використовує атрибути (стовпці) і записи (рядки). Реляційна модель бази даних в даний час є найбільш широко використовуваної моделлю. Ієра-

рхічна модель бази даних об'єднує пов'язані записи і поля в логічну дерево-подібну структуру. Ця структура і взаємозв'язки між елементами даних, відрізняються від тих, які використовуються в реляційній базі даних. Мови реляційного числення базуються на класичному численні предикатів. На основі запиту система керування базами даних автоматично, шляхом формування нових відношень, формує бажаний результат. Мови реляційного числення є не процедурними. Загальним для різних видів серверів баз даних є використання реляційної мови SQL для запитів до даних.

Для системи управління розробкою ІТ-проектів інформація щодо користувачів зберігається у трьох таблицях, що дозволить реалізувати механізм аунтефікації за допомогою токенів, та профайл із даними (рис. 3.1).

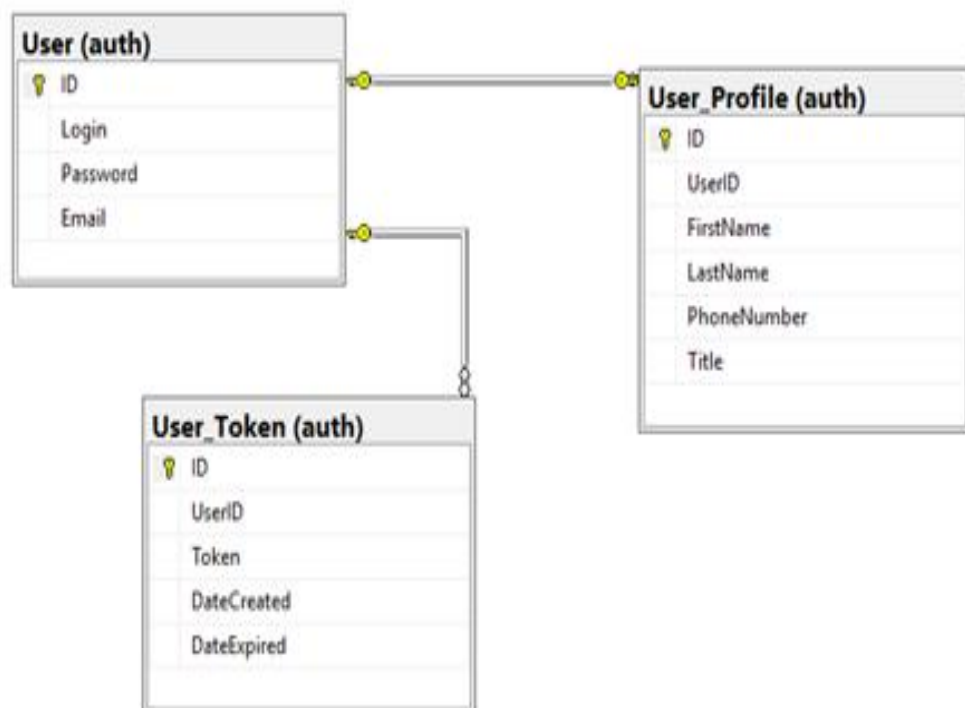


Рисунок 3.1 – Схема «сутність – зв'язок» для таблиць користувачів

Розглянемо сутності, наведеної схеми. Визначимо структуру таблиць із зазначенням типів даних і обмежень цілісності. Сутність «Аккаунт користувача» має атрибути, наведені у таблиці 3.1

Таблиця 3.1 – Сутність «User (auth)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор користувача
2	Login	varchar	Логін користувача для авторизації
3	Password	varchar	Пароль користувача
4	Email	varchar	Електронна пошта користувача

Сутність «User_Profile» містить інформацією про користувача та має атрибути наведені в таблиці 3.2.

Таблиця 3.2 – Сутність «User_Profile (auth)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	UserId	int	Посилання на ідентифікатор із таблиці користувачів
3	FirstName	varchar	Ім'я користувача
4	LastName	varchar	Прізвище користувача
5	PhoneNumber	varchar	Мобільний телефон
6	Title	varchar	Статус користувача

Сутність «User_Token» містить інформацією для реалізації процесу аутентифікації користувача в системі та має атрибути наведені в таблиці 3.3.

Таблиця 3.3 – Сутність «User_Token (auth)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	UserId	int	Посилання на ідентифікатор із таблиці користувачів
3	Token	varchar	Токен користувача
4	DateCreate	date	Дата створення токена
5	DateExpire	date	Дата придатності токена

Для того щоб розробити функціонал для команд та проектів було створено декілька суміжних таблиці для створення відносин «один до кількох».

Ці таблиці використовуються для того, щоб мати змогу додати користувачів до команди, з метою подальшої фільтрації даних які б відносилися лише до конкретної команди (рис. 3.2).

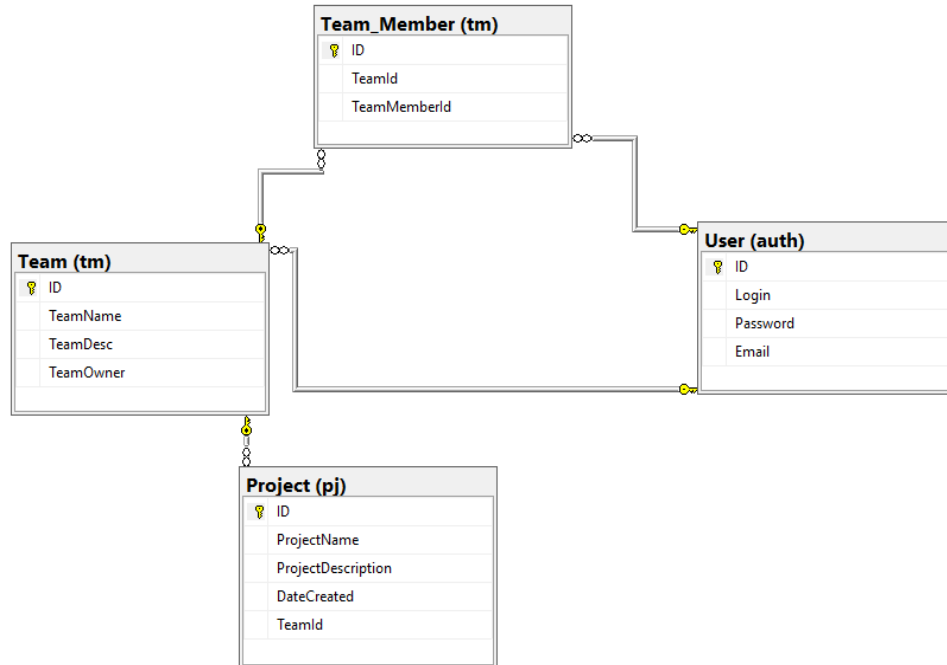


Рисунок 3.2 – Схема «сутність – зв'язок» для таблиць команд та проектів

Розглянемо сутності, наведеної схеми та визначимо структуру таблиць із зазначенням типів даних і обмежень цілісності, що забезпечують функціонування команд під кожен проект у системі. Сутність «TeamMember» та сутність «Team» має атрибути, які наведено у таблиці 3.4 та таблиці 3.5.

Таблиця 3.4 – Сутність «TeamMember»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	TeamId	int	Посилання на ідентифікатор проекту
3	TeamMemberId	int	Посилання на ідентифікатор користувача

Сутність «TeamMember» містить інформацією про членів проекту із

команди.

Сутність «Team» надає інформацію про команди, які зібрані для кожного з проєктів. Дана сутність містить усю необхідну інформацію про власника команди (головного користувача), час створення, ідентифікатор який відображає ким була створена команда (табл. 3.5).

Таблиця 3.5 – Сутність «Team»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор команди
2	TeamName	int	Назва команди
3	TeamDesc	varchar	Опис команди
4	TeamOwner	varchar	Ідентифікатор власника команди (головного користувача)
5	CreatedTime	datetime	Відображає час коли була створена команда
6	CreatedBy	int	Ідентифікатор який відображає ким була створена команда

Для бази даних системи управління командною розробкою ІТ-проєктів для реалізації доступу користувачів до проєкту були створені наступні сутності: Project_Member , що надає інформацію про членів проєкту із команди та має зв'язок с сутністю Project, яка надає дані щодо проєкти в системі. Надалі визначимо атрибути для кожної сутності, а також поставимо для всіх сутностей первинні ключі. Далі розставимо зв'язки між сутностями, що забезпечує зв'язок між членами команди та самими проєктами.

Сутність Project_Member творена для реалізації доступу користувача до проєкту, яка містить у собі усі необхідні зв'язки, які пов'язують користувача із усіма іншими частинами проєкту, дозволяють надати доступи, відобразити персональні дані та збирати статистику, яка буде відобразитися для інших користувачів, які мають певні привілеї у проєкті. Користувачі мають зв'язки у командах, що являються головним зв'язком для користувача, який дозволяє почати повноцінно використовувати систему управління командною розроб-

кою ІТ-проектів. Схема, яка відображає зв'язок сутності Project_Member з іншими таблицями бази даних системи наведено на рисунку 3.3.

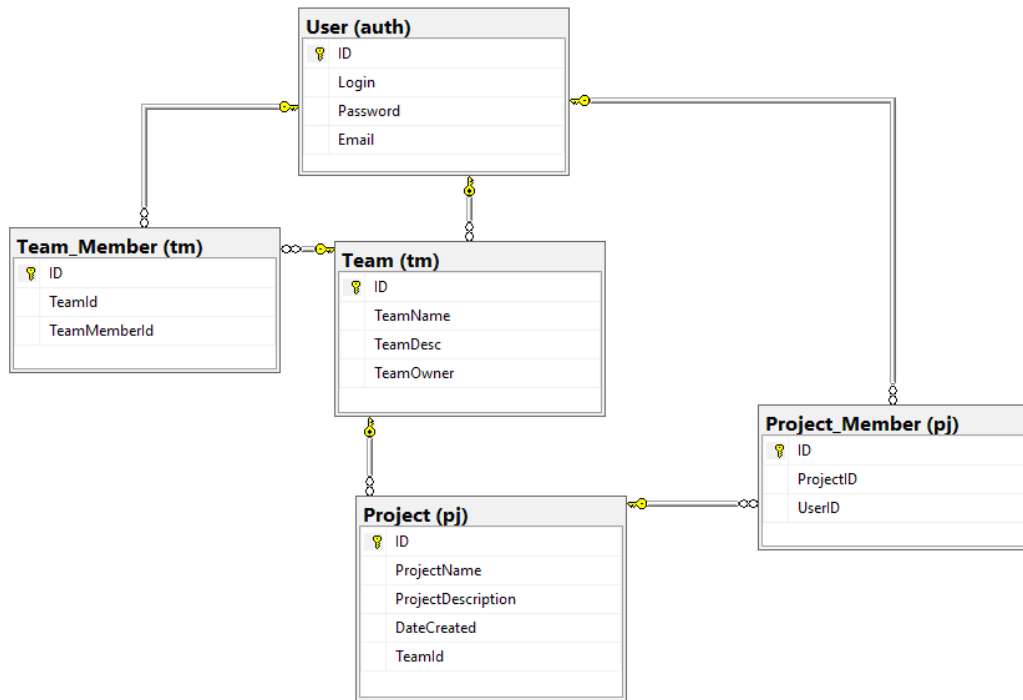


Рисунок 3.3 – Схема «сутність – зв'язок» між сутностями членів команди та проектами

Сутність «Project_Member» надає інформацію щодо користувачів, яким доступний проект (табл. 3.6).

Таблиця 3.6 – Сутність «Project_Member»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	ProjectId	int	Посилання на ідентифікатор проекту
3	UserId	int	Посилання на ідентифікатор користувача

Для функціонування системи управління командною розробкою ІТ-проектів були створені сутності: Project, Project_Card, Project_Card_Task, які

необхідні для створення завдань проекту, внесення завдань у групи у вигляді карток, створення поміток. Схема «сутність-зв'язок» бази даних системи та сутностей, що відповідають за проект наведена на рисунку 3.4.

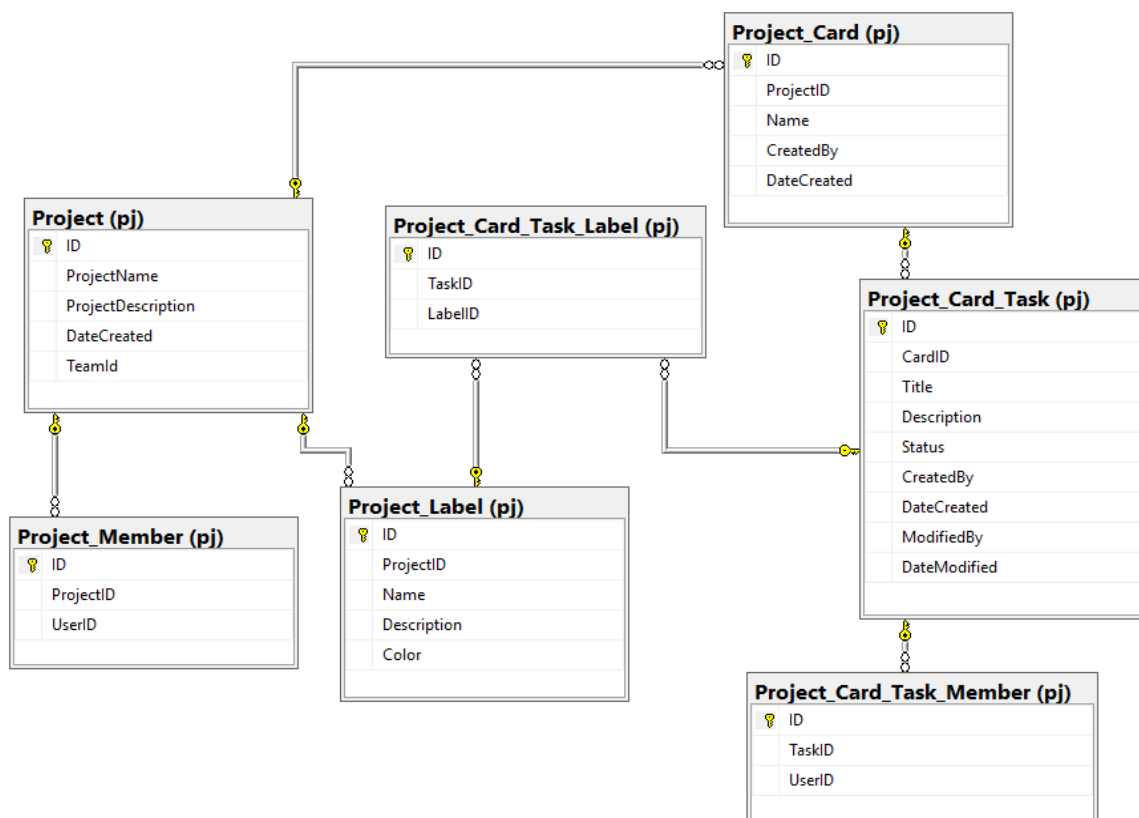


Рисунок 3.4 – Схема «сутність-зв'язок» для проектів в системі

Сутність «Project(pj)» містить інформацією про проекти в системі (табл. 3.7)

Таблиця 3.7 – Сутність «Project(pj)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор проекту
2	ProjectName	int	Назва проекту
3	ProjectDescription	varchar	Опис проекту
4	DateCreated	date	Дата початку проекту
5	TeamId	int	Посилання на ідентифікатор команди

Сутність «Project_Card(pj)» містить інформацією про картки, які будуть містити задачі проектів в системі (табл. 3.8).

Таблиця 3.8 – «Project_Card(pj)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	ProjectID	int	Посилання на ідентифікатор проекту
3	CreatedBy	int	Посилання на ідентифікатор користувача який створив картку
4	DateCreated	date	Дата створення картки

Сутність «Project_Card_Task(pj)» містить інформацією що до кожного з завдань у проекті: назву, опис завдання, дата створення, користувача, який створив та модифікував задачу (табл. 3.9).

Таблиця 3.9 – Сутність «Project_Card_Task(pj)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	CardID	int	Посилання на ідентифікатор картки
3	Title	varchar	Назва завдання
4	Description	varchar	Опис завдання
5	CreatedBy	date	Посилання на ідентифікатор користувача яких створив задачу
6	DateCreated	date	Дата створення задачі
7	ModifiedBy	date	Ідентифікатор користувача яких останній модифікував задачу

Для забезпечення функціональних можливостей для надання користувачу змоги виконувати завдання, а також для забезпечення привабливості задачам були створені сутності: користувачів, які повинні виконувати задачу, маркерів проекту, маркерів які використовуються для завдань, які забезпечують зв'язки між користувачами у командах та задачами. Також були ство-

рені т Сутності для збереження інформативних дощок (лейблів). Сутність, яка зберігає інформацію щодо користувачів, які повинні виконувати задачу, наведена у табл. 3.10.

Таблиця 3.10 – Сутність «Project_Task_Member(pj)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор таблиці
2	TaskId	int	Посилання на ідентифікатор задачі
3	UserId	int	Посилання на ідентифікатор користувача

Сутність «Project_Label(pj)» зберігає інформацію щодо маркерів проекту: назву, опис, кольори маркеру (табл. 3.11).

Таблиця 3.11 – Сутність «Project_Label(pj)»

№	Назва атрибуту	Тип даних	Призначення
1	ID	int	Ідентифікатор маркеру
2	ProjectId	int	Посилання на ідентифікатор проекту
3	Name	varchar	Назва маркеру
4	Description	varchar	Опис маркеру
5	Color	varchar(6)	Поле із Нех значенням яке використовуються для кольору маркеру

Сутність «Project_Card_Task_Label(pj)» зберігає інформацію щодо маркерів, які використовуються для завдань проекту (табл. 3.12).

Таблиця 3.12 – Сутність «Project_Card_Task_Label(pj)»

Назва атрибуту	Тип даних	Призначення
ID	int	Ідентифікатор таблиці
TaskId	int	Посилання на ідентифікатор завдання
LabelId	varchar	Посилання на ідентифікатор маркеру

Усі Сутності та зв'язки між ними, які формують базу даних системи управління командною розробкою ІТ-проектів та формують необхідний фун-

кціонал системи управління наведені на рисунку 3.5.

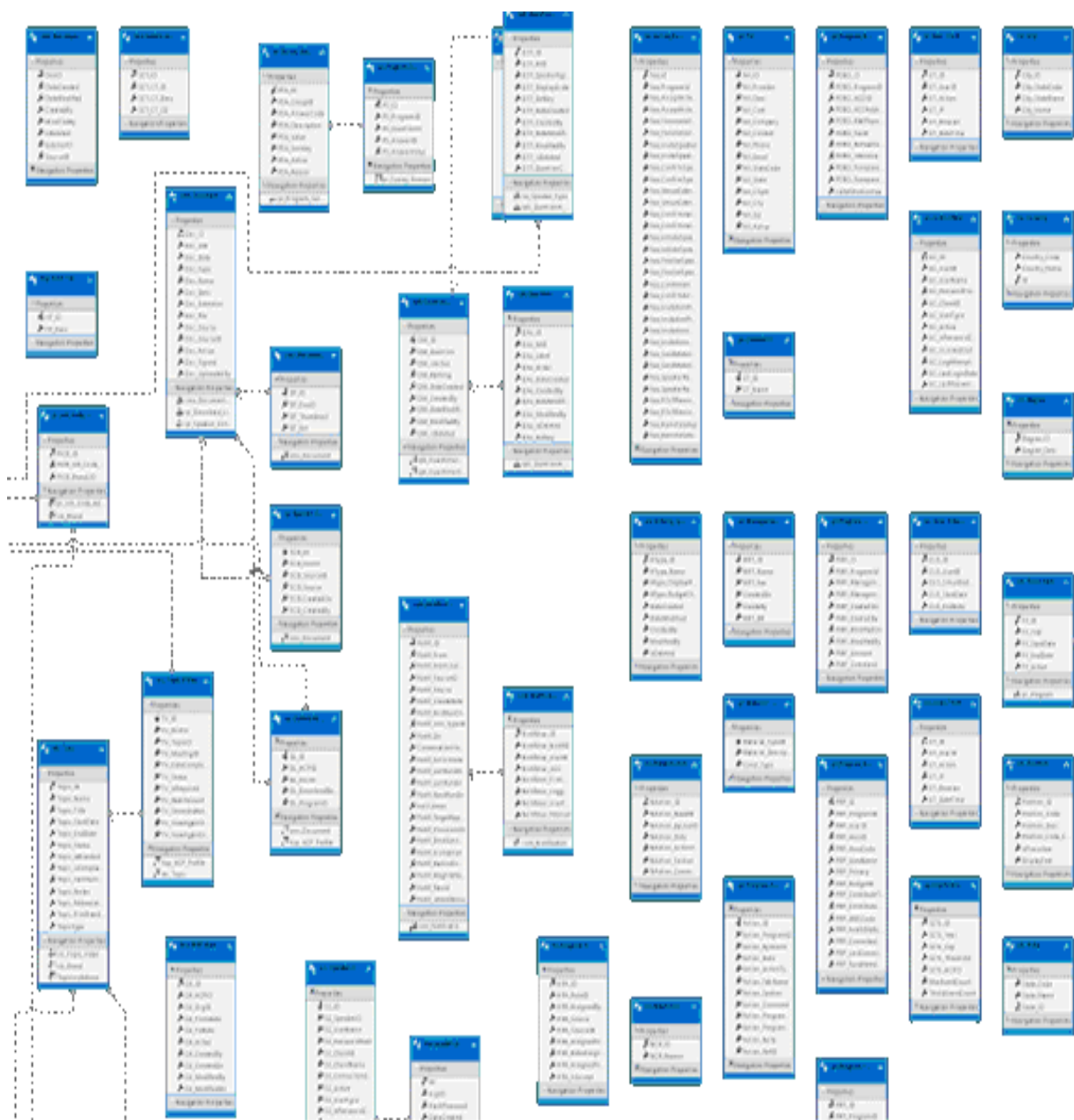


Рисунок 3.5 – Структура бази даних системи управління ІТ-проектами

В ході проведеного проектування системи управління командною розробкою ІТ-проектів, створена база даних системи, яка наведена у вигляді діаграм «сутність-зв'язок», що дозволяє реалізувати усі функціональні можливості системи та дозволяє здійснювати розширення системи у майбутньому додаючи необхідні сутності у існуючу базу даних.

3.4 Схема функціонування системи управління ІТ-проектами

Одною із головних цілей системи управління командною розробкою ІТ-проектів є надання користувачам простого інструменту та зрозумілого інтерфейсу, який би дозволяв легко та ефективно працювати із важкою логікою серверної частини, яка повністю захована від користувача. Створений графічний інтерфейс додатку системи управління командною розробкою ІТ-проектів дозволяє інтуїтивно та без попередньої підготовки використовувати усі можливості системи. (рис. 4.1).

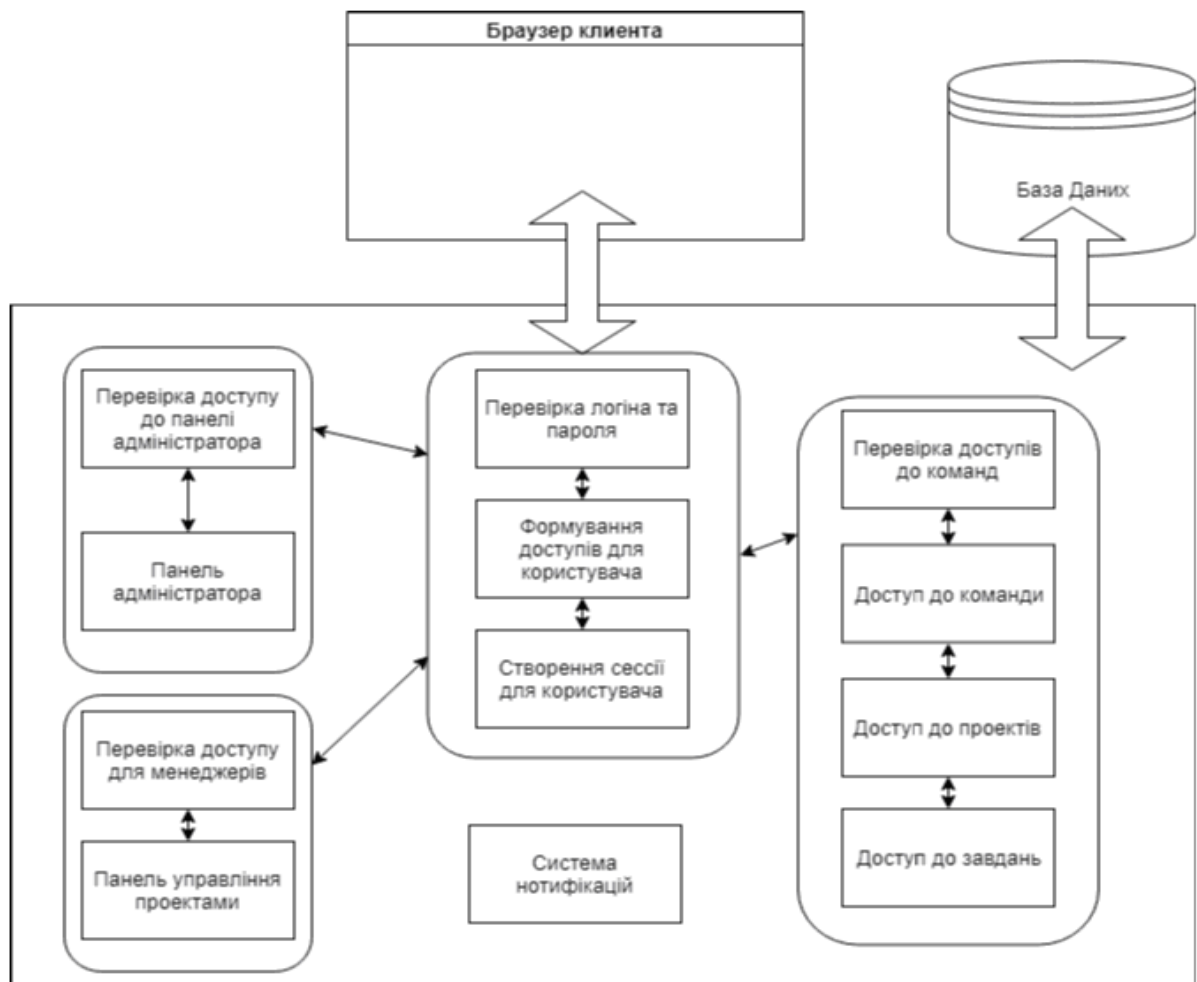


Рисунок 3.6 – Схема роботи системи управління ІТ-проектами

Приховування складних частин системи дозволяє звести к мінімуму мож-

ливість виникнення помилок. Створені інтуїтивні переходи, які надають користувачу інтуїтивні підказки, що йому слід зробити щоб опинитися у конкретній частині системи. Відсутність великої кількості функціональних можливостей безпосередньо на одному екрані забезпечує відсутність плутаниці із боку користувачів

Система управління командною розробкою ІТ-проектів надає наступний перелік можливостей:

- зручний спосіб доступу до системи за допомогою браузера на будь-якій платформі;
- можливість реєстрації в системі;
- можливість отримання доступу користувачам за допомогою посилань у вигляді запрошень;
- можливість налаштування розмежування доступу для користувачів;
- можливість створювати команди та запрошувати користувачів;
- можливість створювати проекти для окремих команд;
- можливість створювати задачі;
- спеціальна панель для управління задачами;
- система пріоритетів для завдань;
- можливість представляти завдання у вигляді календарного плану;
- можливість створювати взаємозв'язки між завданнями;
- можливість створювати повідомлення для користувачів;
- можливість створення повідомлень у задачах, які виконуються користувачами;
- можливість перегляду графіків, які відображують стан проекту.

Система управління ІТ-проектами задовольняє усім зазначеним вимогам, які потребують сучасні команди розробників ІТ-проектів, а також містить у собі усі необхідні інструменти для можливості створення, ведення та завершення вдалого проекту.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ

4.1 Реалізація загальних елементів інтерфейсу користувачів

Вікно авторизації у системі управління командною розробкою ІТ-проектів дозволяє користувачам отримати доступ до системи вже зареєстрованих користувачів та здійснити реєстрацію у системі для нових користувачів. Користувачі мають можливість створити новий аккаунт за допомогою внутрішніх механізмів системи, або за допомогою сторонніх сервісів таких як GitHub, LinkedIn, Google. Якщо користувач потрапляє до системи за допомогою сторонніх сервісів, система управління командною розробкою ІТ-проектів все одно збереже користувача із тими даними, які вона може отримати від сторонніх сервісів (рис. 4.1).

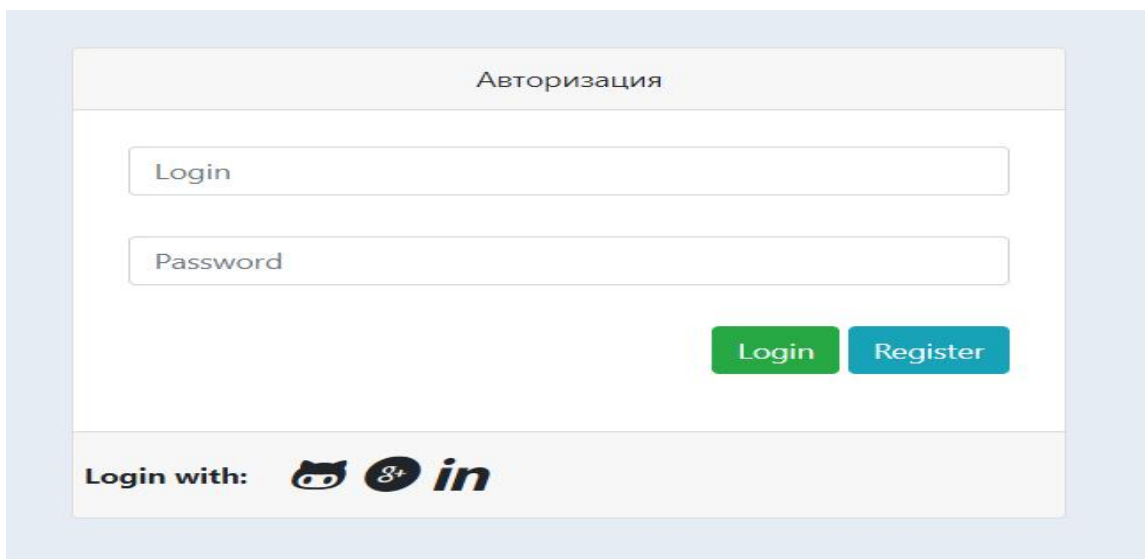


Рисунок 4.1 – Вікно для авторизації або реєстрації користувачів у системі

Після першої реєстрації користувач має змогу авторизуватися як за допомогою стороннього сервісу так і за допомогою внутрішніх механізмів системи, така інтеграція дозволяє користувачу використовувати один аккаунт будь де, якщо ця можливість існує (рис. 4.2).

The image shows a registration form titled "Реєстрація" (Registration). It contains the following fields and elements:

- A "Login" field.
- Two fields for "First Name" and "Last Name".
- An "Email" field.
- A "Password" field.
- A "Confirm password" field.
- A green "Register" button at the bottom right.

Рисунок 4.2 – Форма для реєстрації нових користувачів у системі

4.2 Робота з додатком користувача системи управління

Після авторизації у системі користувач потрапляє на сторінку, яка має інформацію щодо переліку команд у системі (рис. 4.3).

The image shows a dashboard titled "Team Traker" with a search bar "Find Project" and a user icon "ИК". The main content area is titled "Список команд" (List of teams) and includes two buttons: "Создать команду" (Create team) and "Управление командами" (Manage teams). Below the buttons is a grid of four team cards:

Smart World	LS Group	DGC Group	NB Group
Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisquam officis quo quod dicta.	Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisquam officis quo quod dicta.	Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisquam officis quo quod dicta.	Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisquam officis quo quod dicta.

On the right side, there is a "Информация" (Information) panel titled "Последняя информация" (Latest information) containing two notifications:

- 19.10.2021: Добро пожаловать! Теперь вам необходимо создать команду (Welcome! You now need to create a team).
- 19.10.2021: Тут будут появляться новости и оповещения когда что-либо происходит в проекте (Here you will see news and notifications when something happens in the project).

Рисунок 4.3 – Сторінка з переліком команд доступних користувачу

Головний користувач або користувачі, які мають права доступу до управління командами, мають кнопку за допомогою якої вони можуть потрапити до сторінки управління командами. На цій сторінці знаходиться інформація про команди, а саме: назву команди, логотип команди, та короткий опис який командний проект саме розробляється цією командою.

Будь-який користувач у системі має можливість створити нову команду. Для цього потрібно перейти на сторінку для створення команд (рис. 4.4).

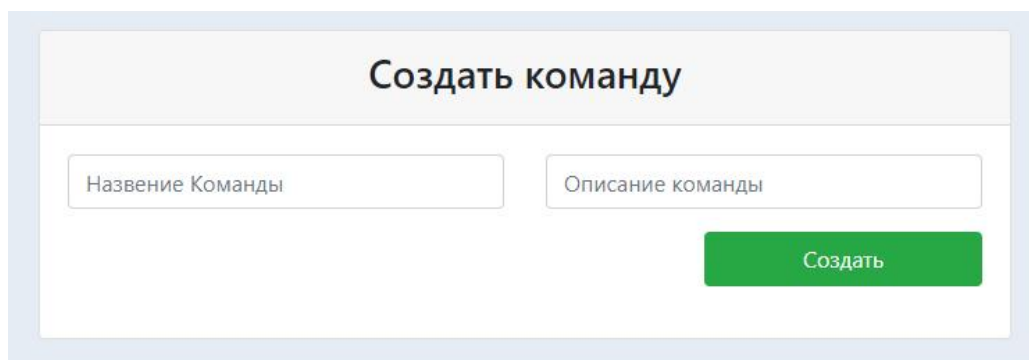


Рисунок 4.4 –Форма для створення нової команди

Користувачі з певними правами доступу мають змогу потрапити на сторінку управління існуючими командами (рис. 4.5).

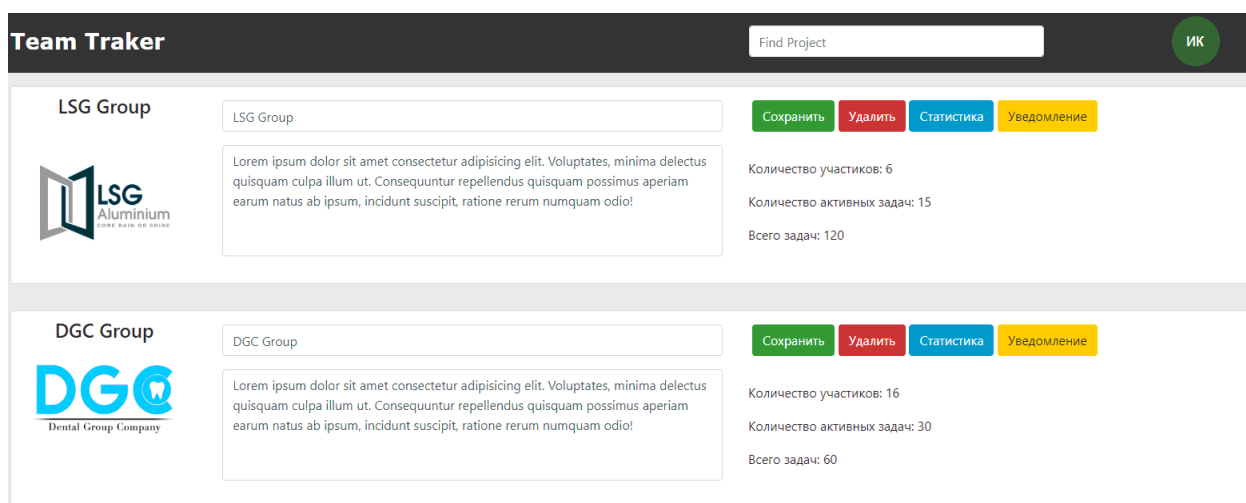


Рисунок 4.5 – Сторінка для управління командами проектів

На цій сторінці користувачі мають можливість змінити назву команди, надіслати повідомлення кожному з членів команди. Користувачу, який створив команду надається можливість видалити команду, а також здійснити перегляд статистичної інформації команди, натиснувши кнопку «Статистика».

На сторінці статистики команди користувачу надається наступна інформація: інформація про активні та завершені задачі, інформація про кількість завершених та активних проектів та графік активності користувачів, які входять до певної команди (рис. 4.6).

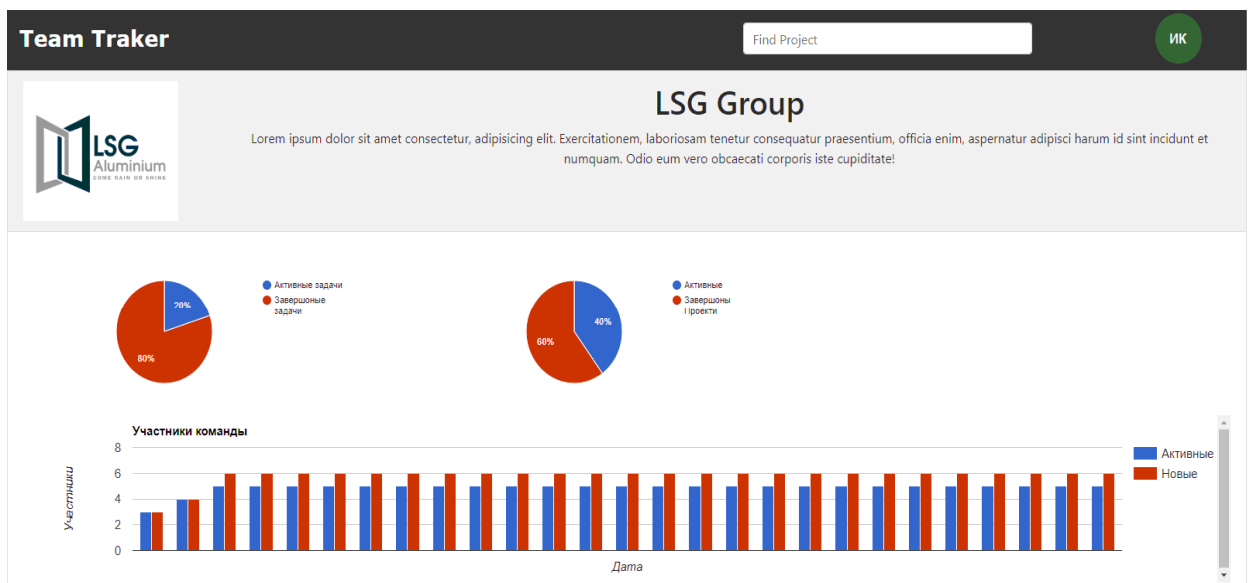


Рисунок 4.6 – Сторінка статистичної інформації про команди

Після обрання команди користувач потрапляє на сторінку із проектами, які доступні команді, на цій сторінці. Користувач має можливість побачити доступні йому проекти, користувачі із певними привілеями можуть перейти на сторінку управління проектами або створити новий проект для команди.

На сторінці задач проекту користувач може переглянути задачі, які йому доступні та при наявності відповідних привілеїв перейти на сторінку менеджера задач. Кожна з задач має наступну інформацію: задачі мають статус, назву та короткий опис. Кожна з задач може мати лейбли, які графічно візуалізують завдання та надають додаткову інформацію (рис. 4.7).

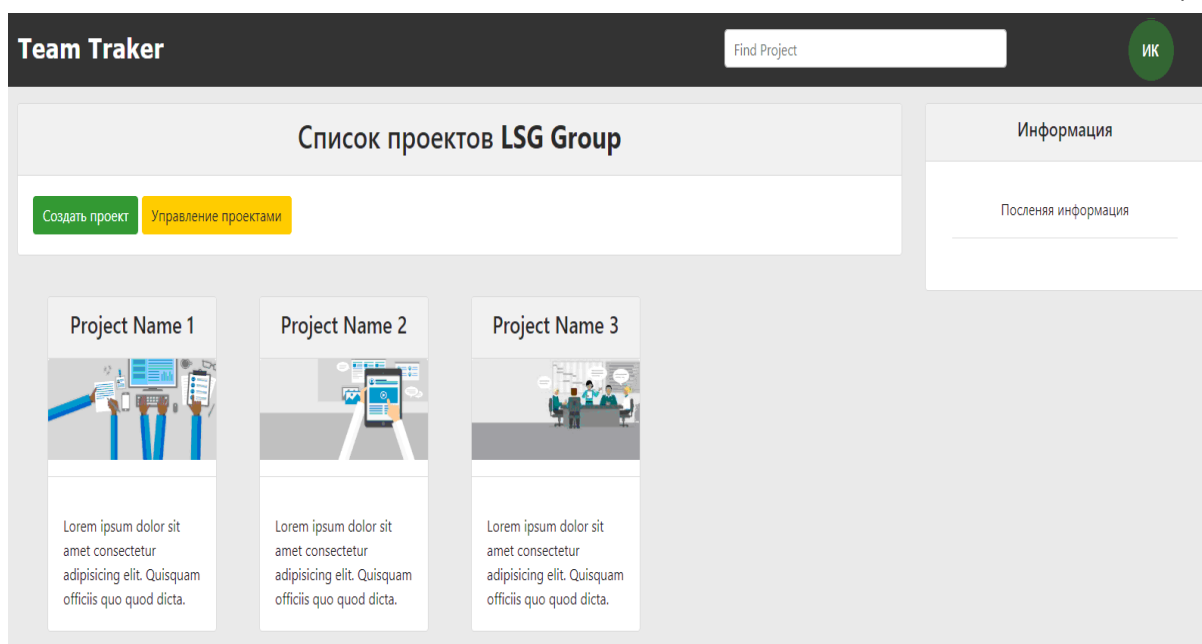


Рисунок 4.7 – Сторінка проектів, які доступні для обраної команди

Якщо користувач володіє певними правами у системі, йому доступна панель швидкого управління в якій користувач може виконувати наступний перелік дій:

- перейти на сторінку задачі;
- переглядати область дискусій щодо задачі;
- додати до задачі нову лейблу із додатковою інформацією;
- створити повідомлення для користувачів, які мають доступ до задачі;
- видалити задачу.

Кожна з задач може мати стандартні статуси для задач:

- Hold – задача знаходиться на утриманні, та не потребує подальшої розробки;
- Active – задача знаходиться у стані реалізації;
- Closed – задача зачинена або закінчена;
- Draft – нова задача над якою працює менеджер.

Статистичну інформацію про задачі проекту можна переглянути на сторінці управління задачами (рис. 4.8).

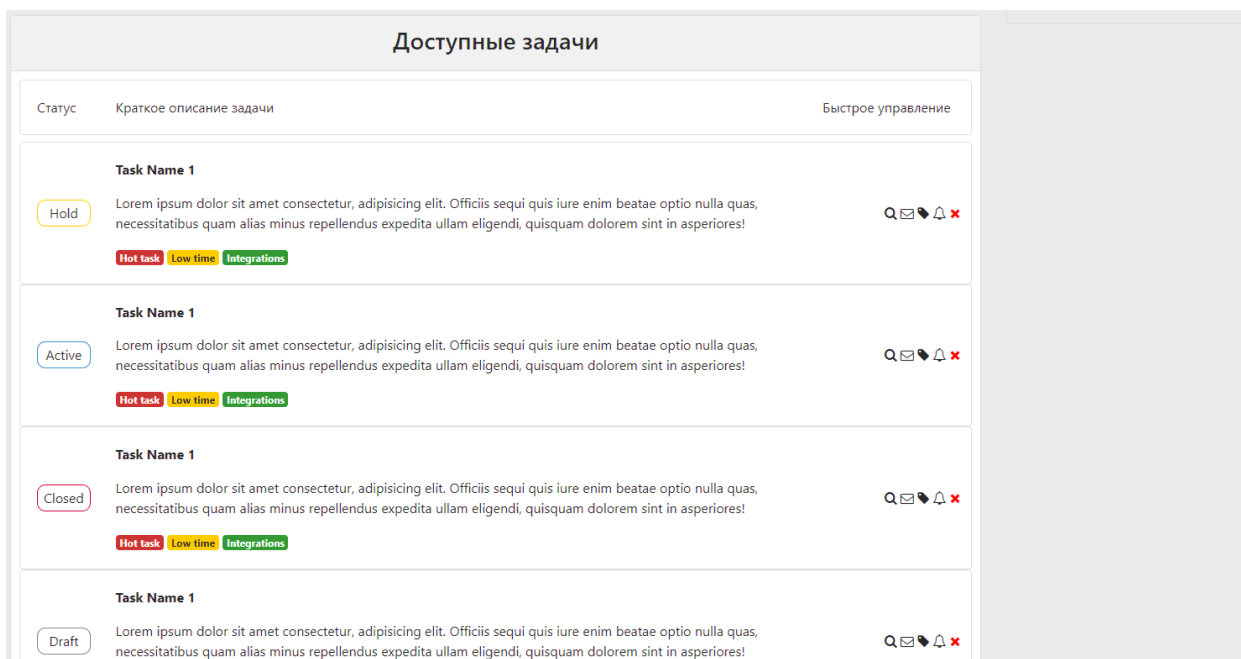


Рисунок 4.8 – Сторінка управління задачами проекту для користувача

При виборі задачі користувач потрапляє на сторінку, яка дозволяє працювати з обраною користувачем задачею. (рис. 4.9).

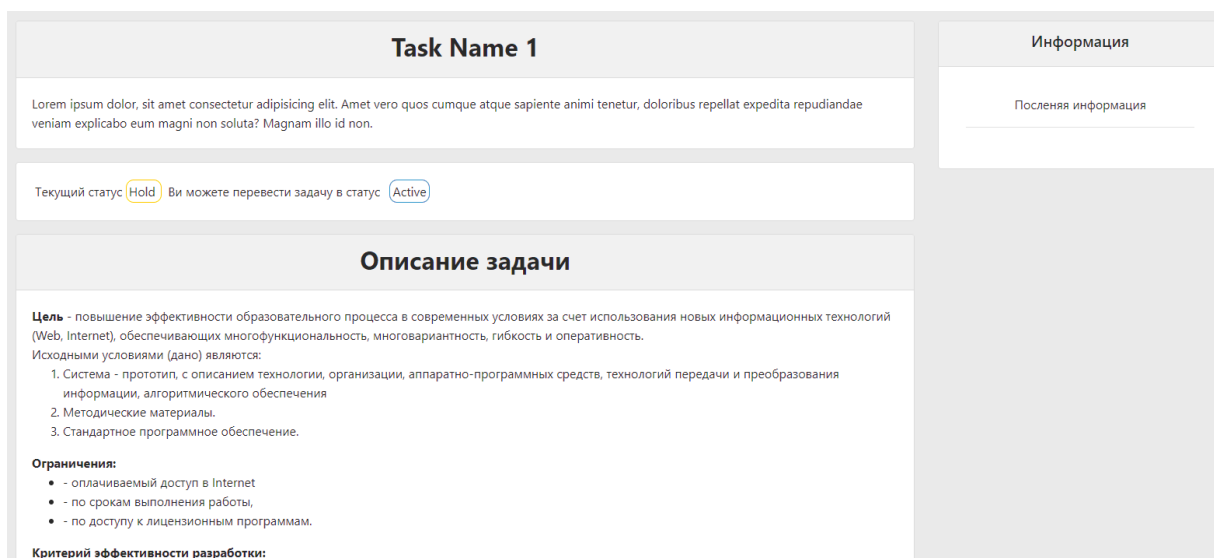


Рисунок 4.9 – Сторінка з інформацією про задачу

На цій сторінці користувач може переглянути назву задачі, її короткий

опис, та детальну інформацію. Користувачі з певними привілеями можуть змінити статус задачі. На сторінці із задачами у нижній секції користувач має змогу вести та переглядати дискусії, які проводяться в рамках обговорення саме цієї задачі (рис. 4.10).

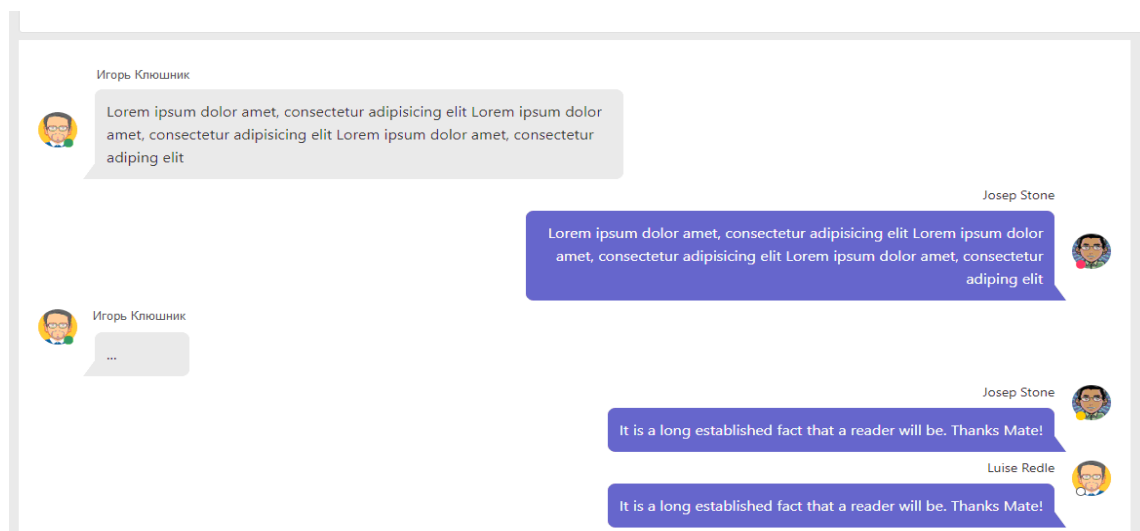


Рисунок 4.10 – Область дискусій для обраної задачі

Користувачу надається можливість перегляду задач у вигляді календарного плану при переході на відповідну сторінку (рис. 4.11).

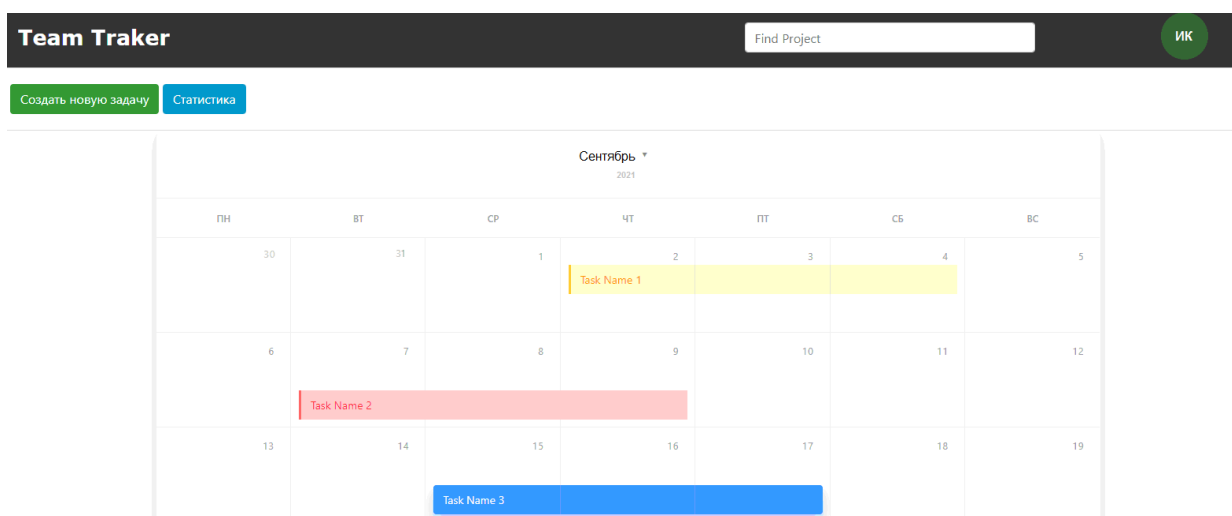


Рисунок 4.11 – Сторінка планування задач у вигляді календарного плану

Користувач має можливість створити нову задачу та за допомогою встановлення дат задача буде відображена на календарному плані. Календар містить три основних кольори:

- жовтий – для задач, які утримані, час для них не враховується;
- червоний – для задач, які були прострочені;
- блакитний – для активних задач, які не вийшли за межі плану.

На сторінці статистики користувач може переглянути статистичну інформацію у вигляді діаграм (рис. 4.12).

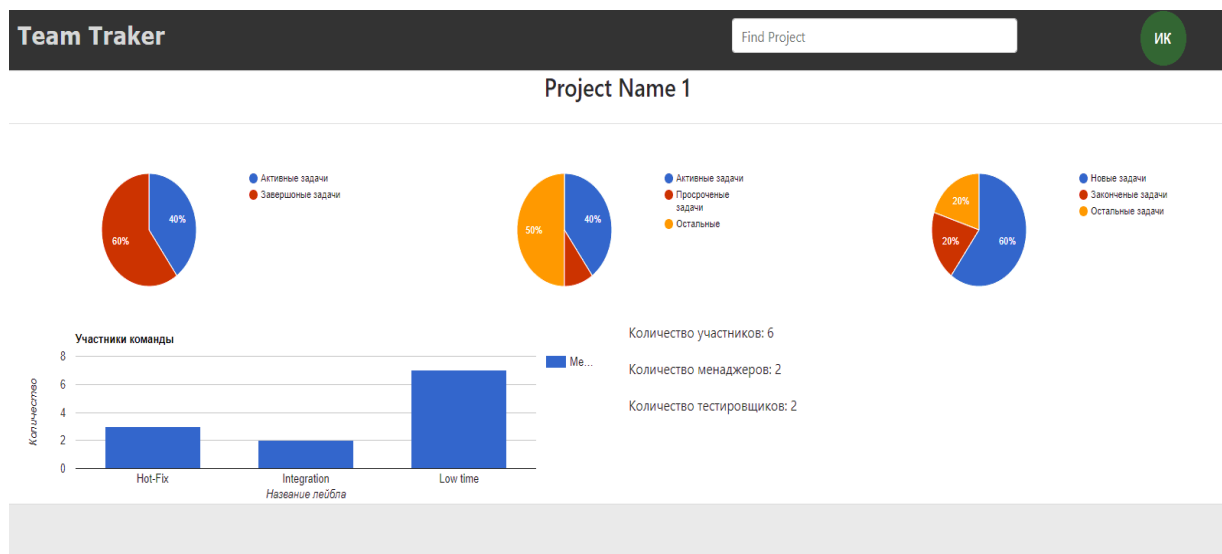


Рисунок 4.12 – Сторінка з статистичною інформацією у вигляді діаграм

Сторінка статистики надає користувачу наступну інформацію у вигляді діаграм:

- порівняльна статистика між активними та завершеними задачами;
- відображення задач на виконання яких вийшов запланований час;
- порівняльна статистика між новими та завершеними задачами;
- графік, який показує кількість інформаційних лейблів;
- інформація про загальну кількість користувачів на проекті;
- інформація про кількість менеджерів проекту;
- інформація про кількість тестувальників проекту.

4.3 Реалізація функцій користувача-адміністратора системи

Користувач-адміністратор системи управління командною розробкою IT-проектів, або користувач з певними привілеями за допомогою сторінки профілю можуть перейти на сторінку управління користувачами. Користувачу-адміністратору надається можливість призначити привілеї користувачам, формувати або змінювати склад команди для кожного з проектів. Також на цій сторінці користувачу-адміністратору надається змога розподіляти доступ користувачів до певних проектів, здійснювати пошук користувачів в системі (рис. 4.13).

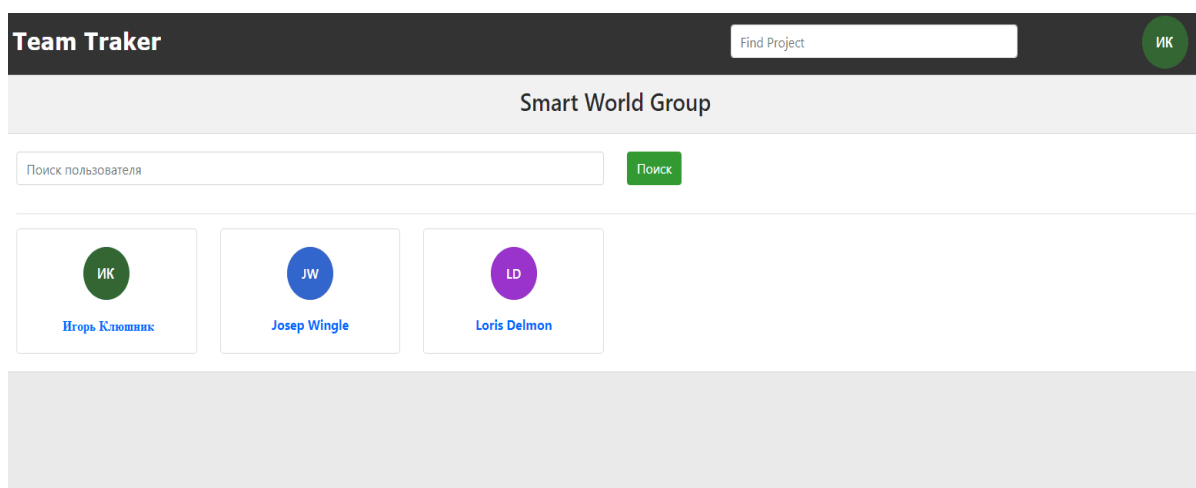


Рисунок 4.13 – Пошук користувачів на сторінці управління

Спочатку користувач-адміністратор повинен перейти до необхідної команди, та на спеціальній сторінці пошуку користувач може обрати користувача для подальшої конфігурації його профайлу. Після обрання користувача, адміністратор потрапляє на сторінку управління користувачем, на якій розташовані елементи згідно із привілеями які доступні для обраного користувача. Адміністратор має можливість видалити користувача із команди, оновити пароль, надіслати повідомлення, змінити інформацію про користувача, та встановити привілеї користувачу (рис. 4.14).

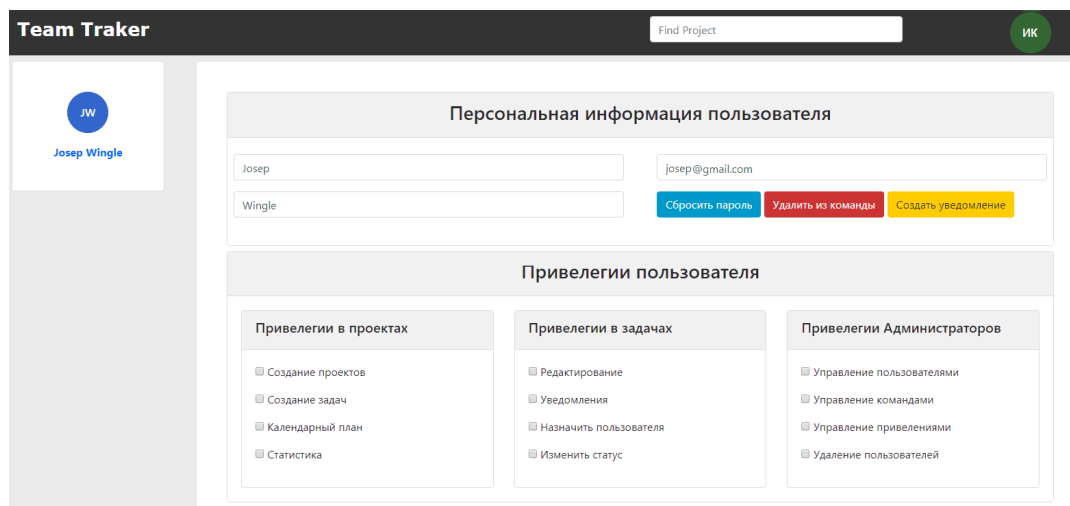


Рисунок 4.14 – Сторінка управління привілегіями користувачів

Користувач-адміністратор на цій сторінці має змогу переглянути до яких проектів має доступ користувач, та при необхідності надати доступ користувачу до іншого проекту, або видалити його з існуючого (рис. 4.15).

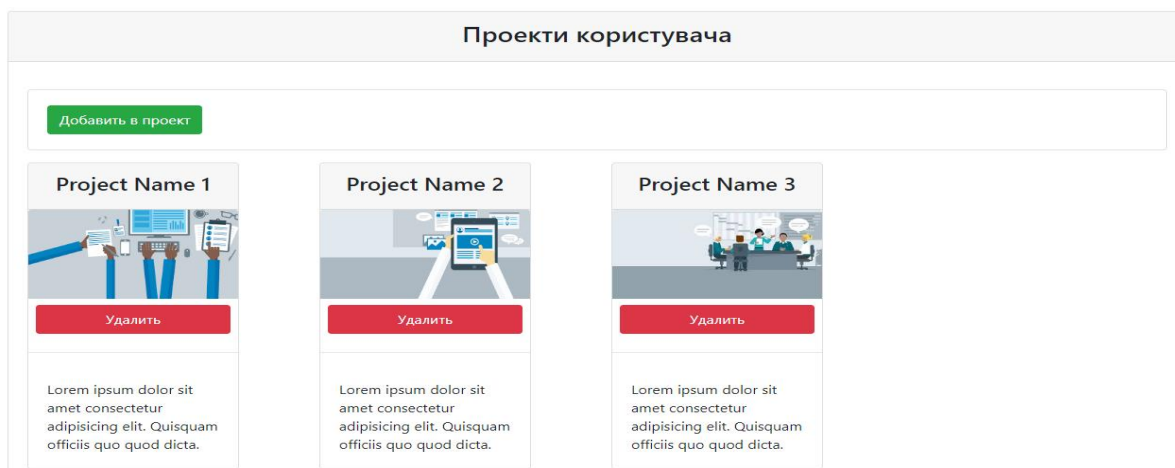


Рисунок 4.15 – Сторінка управління доступом до проектів

В результаті була створена програмна реалізація системи управління командною розробкою ІТ-проектів, що забезпечення розробників проектів автоматизованими засобами для розробки програмних продуктів.

ВИСНОВКИ

В результаті виконання магістерської роботи було проведено проектування та програмну реалізацію системи управління командною розробкою IT-проектів. Розробка такої системи надає можливість компаніям, задіяним у сфері IT-розробки, легко та зручно слідкувати за усіма етапами командної розробки проектів, контролювати хід виконання задач та стежити за ефективністю їх реалізації. При створенні системи проектування було проведено дослідження та аналіз завдань систем управління IT-проектами, визначені цілі, завдання, вимоги до систем управління з урахуванням особливостей систем для управління IT-проектами. Визначена архітектура системи, здійснено вибір та обґрунтування програмних засобів реалізації, проведено проектування системи, розроблена база даних системи, здійснена програмна реалізація.

Реалізована система управління командною розробкою IT-проектів має максимально зручний інтерфейс користувача і зрозумілі діалогові вікна, що забезпечує зручність роботи з системою. Програмна реалізація системи виконана з використанням сучасних технологій та програмних засобів розробки Web-застосувань: база даних MSSQL, сервер реалізовано за допомогою технології ASP.Net, мова програмування C#, технології для веб-клієнту – Angular, HTML5 та CSS3.

Система управління проектами дозволяє командам користувачів будь-якої кількості налагодити ефективний процес розробки програмних продуктів у зручний спосіб, що надає можливість якісного управління розробкою IT-проектів. Використання такої системи на сьогоднішній день є одним із обов'язкових атрибутів будь-якої IT-компанії, збереження та обробка даних дозволяє постійно підвищувати якість та спроможність компаній, які їх використовують.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Фесенко Т. Г. Управління проектами: теорія та практика виконання проектних дій: навч. посібник. Харк. нац. акад. міськ. госп-ва. – Х. : ХНАМГ, 2012. 181 с.
2. Що таке управління проектами: як здійснюється і що в себе включає? URL: <https://blog.agrokebety.com/shcho-take-upravlinnya-proektamy> (дата звернення 06.09.2021).
3. Ершов С.В. Управление проектами и программами. Конспект лекций. Архангельск: САФУ, 2015. 226 с.
4. Онопко А.С., Жигалкевич Ж.М. Застосування інформаційних технологій в управлінні підприємством. Актуальні проблеми економіки та управління. 2017. №11. URL: <http://ape.fmm.kpi.ua/article/viewFile/102782/97865> (дата звернення: 22.10.2021)
5. Система управління проектами URL: https://uk.wikipedia.org/wiki/Система_управління_проектами (дата звернення 10.09.2021).
6. Рішення для управління проектами. URL: <https://blog.hubstaff.com/project-management-software-solutions/> (дата звернення 18.10.2021).
7. Богославец А.А. Классификация IT-проектов. Харьк. нац. унив. городского хозяйства им. А.Н. Бекетова. «Комунальне господарство міст», 2014, випуск №118. ISSN 0869-1231. URL: <https://core.ac.uk/download/pdf/33757322.pdf> (дата звернення 21.10.2021).
8. Молоткова Н.В., Сахаров И.С. Инфраструктура управления качеством IT-проектов. URL: http://vernadsky.tstu.ru/pdf/2008/03g/22g_13.pdf (дата звернення 24.10.2021).
9. Коньшунова А.Ю. К вопросу о классификации проектов в проектном управлении. URL: <http://sibac.info/11760> (дата звернення 24.10.2021).

10. Грекул В. И., Коровкина Н. Л., Куприянов Ю. В. Методические основы управления ИТ-проектами. Учебник для вузов. М.: БИНОМ. Лаб. знаний ИНТУИТ, 2010. 392 с.
11. Заговора О.В., Концевич В.Г. Учёт особенностей ИТ-проектов при определении их жизненного цикла. URL: <http://cyberleninka.ru/article/n/uchet-osobennostey-itproektov-pri-opredelenii-ih-zhiznennogo-tsikla> (дата звернения 26.10.2021).
12. Д. Эспозито. Разработка современных веб-приложений. Анализ предметных областей и технологий. М.: Диалектика-Вильямс, 2017. 464 с.
13. А. Фримен. ASP.NET MVC 5 с примерами. М.: Вильямс, 2015. 736с.
14. Д. Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. М.: Питер, 2017. 896 с.
15. Д. А. Фримен. Angular для профессионалов. М.: Питер, 2016. 800 с.
16. Про Node.js. URL: <https://nodejs.org/uk/about> (дата звернения 26.10.2021).
17. TypeScript. URL: <https://ru.wikipedia.org/wiki/TypeScript> (дата звернения 02.11.2021).
18. И. Бен-Ган. Microsoft SQL Server 2012. Основы T-SQL. М.: ЕКСМО, 2015. 400 с.
19. А. Фримен. ASP.NET Core MVC с примерами на C # для профессионалов. М.: Диалектика-Вильямс, 2016. 992 с.