

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Автоматизована система управління проєктами для малих
комерційних підприємств»

Виконав (-ла):

студент (-ка) IV курсу, групи ІК-92

Пилипчук Марія Вячеславівна

Керівник:

Доцент кафедри ІСТ, к.т.н., доцент

Мамедова Катерина Юріївна

Рецензент:

Директор КБІС, к.т.г., доцент

Фіногенов Олексій Дмитрович

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ
на дипломний проєкт студенту
Пилипчук Марії Вячеславівні

1. Тема проєкту «Автоматизована система управління проєктами для малих комерційних підприємств», керівник проєкту Мамедова Катерина Юріївна, к.т.н., доцент, затверджені наказом по університету від «31» травня 2023 р. № 2101-с

2. Термін подання студентом проєкту: 12 червня 2023 р.

3. Вихідні дані до проєкту:

Мова програмування C#, додатково WindowsForms, фреймворк ASP.NET, середовище програмування – Visual Studio.

4. Зміст пояснювальної записки:

- 1) Формулювання проблеми, дослідження предметної області та актуальності;
- 2) Огляд існуючих рішень для вирішення проблеми.
- 3) Аналіз сучасних засобів та методів розробки та їх вибір;
- 4) Опис процесу проектування та розробки програмного рішення;

- 7) Опис процесу тестування розробленого програмного рішення;
- 8) Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- 1) Діаграма послідовності;
- 2) Дерево проблем проекту;
- 3) Дерево результатів проекту;
- 4) Діаграма класів;
- 5) ER – діаграма;
- 6) Структура БД;
- 7) Презентація роботи.

6. Дата видачі завдання 1 березня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	21.04.2023	
2	Аналіз аналогів	25.04.2023	
3	Вибір стеку технологій	30.04.2023	
4	Проектування системи	03.05.2023	
5	Розробка бази даних	08.05.2023	
6	Розробка програмної частини	28.05.2023	
7	Тестування системи	30.05.2023	
8	Створення графічних матеріалів	03.06.2023	
9	Оформлення пояснювальної записки	05.06.2023	

Студент

Марія ПИЛИПЧУК

Керівник

Катерина МАМЕДОВА

АНОТАЦІЯ

Пилипчук М.В. Автоматизована система управління проектами для малих комерційних підприємств. КПП ім. Ігоря Сікорського, Київ, 2023.

Проект містить 63 с. тексту, 21 рисунок, 7 таблиць, посилання на 17 літературних джерел, додатки та 6 конструкторських документа.

Ключові слова: управління проектами, проект, команда, малі підприємства, автоматизована система, серіалізація даних.

Об'єктом розробки є система управління проектами.

Мета розробки – метою даного дипломного проекту є покращення ефективності планування для малого бізнесу не технічної галузі, в умовах обмеженості ресурсів.

У дипломному проекті було проведено дослідницьку роботу щодо ролі управління проектів на невеликих підприємствах. Проаналізовано причини відмови від методів проектного управління та методи їх вирішення. Було протестовано існуюче програмне забезпечення, виділено його переваги та недоліки. Розроблено та описано процес розробки програмного забезпечення для досягнення поставлених цілей.

Для розробки програмного забезпечення було обрано мову C# за її універсальність, та фреймворк ASP.Net через можливості та інструменти, які він надає. Для роботи та збереження даних використовується мова SQL та методи серіалізації та десеріалізації даних.

Отримані результати можуть бути використані під час як планування проекту, так і для планування повсякденної роботи. Розроблене програмне забезпечення цілком підходить для малих підприємств з обмеженими ресурсами, бюджетом та не потребує додаткових знань від персоналу.

SUMMARY

Pilipchuk M. V. automated project management system for small commercial enterprises. Igor Sikorsky KPI, Kiev, 2023.

The project contains 63 pages of text, 21 Figures, 7 tables, links to 17 literature sources, annexes and 6 design documents.

Keywords: Project Management, Project, Team, small enterprises, automated system.

The object of development is a project management system.

The goal of this diploma project is to improve the efficiency of planning for small businesses in the non-technical industry, in conditions of limited resources.

In the diploma project, research work was carried out on the role of Project Management in small enterprises. The reasons for abandoning project management methods and methods of solving them are analyzed. The existing software was tested and its advantages and disadvantages were highlighted. The software development process for achieving your goals has been developed and described.

For software development, the C# language was chosen for its versatility, and the framework ASP.Net because of the features and tools it provides. The SQL language and data serialization and deserialization methods are used to work and save data.

The results obtained can be used during both project planning and day-to-day work planning. The developed software is quite suitable for small businesses with limited resources, budget and does not require additional knowledge from staff.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	ІК92.280БАК.005 ПЗ	Пояснювальна записка	63		
6	A3	ІК92.280БАК.005 Д1	Автоматизована система управління	1		
7			проектами для малих комерційних			
8			підприємств. Діаграма послідовності.			
9	A3	ІК92.280БАК.005 Д2	Автоматизована система управління	1		
10			проектами для малих комерційних			
11			підприємств. Дерево проблем.			
12	A3	ІК92.280БАК.005 Д3	Автоматизована система управління	1		
13			проектами для малих комерційних			
14			підприємств. Дерево результатів.			
15	A3	ІК92.280БАК.005 Д4	Автоматизована система управління	1		
16			проектами для малих комерційних			
17			підприємств. Діаграма класів.			
18	A3	ІК92.280БАК.005 Д5	Автоматизована система управління	1		
19			проектами для малих комерційних			
20			підприємств. ER-діаграма.			
21	A3	ІК92.280БАК.005 Д6	Автоматизована система управління	1		
22			проектами для малих комерційних			
23			підприємств. Структура бази даних.			
24						
25						
26						
27						
28						
Зм.	Аркуш	№ докум.	Підпис	Дата	ІК92.280БАК.005 ТП Автоматизована система управління проектами для малих комерційних підприємств. Відомість проекту Літ. Аркуш Аркушів Т 1 1 КП ім. Ігоря Сікорського Група ІК-92	
Розроб.		Пилипчук М.В				
Керівн.		Мамедова К.Ю				
Затв.						

Пояснювальна записка
до дипломного проєкту
на тему: «Автоматизована система управління
проєктами для малих комерційних підприємств»

Київ – 2023 року

ЗМІСТ

СПИСОК СКОРОЧЕНЬ.....	5
ВСТУП	6
1 Загальна характеристика предметної області і вирішуваних завдань.....	8
1.1 Опис основного бізнес-процесу, який підлягає автоматизації	8
1.2 Склад і ієрархія завдань і підзадач, які вирішуються в програмі	9
1.3 Схеми вхідних і вихідних інформаційних потоків для кожного завдання і підзадачі.....	10
1.4 Послідовність вирішення завдань і підзадач.....	13
1.5 Опис програмних модулів і їх функцій.....	13
1.6 Опис структури даних та алгоритмів	14
Висновки за розділом	15
2 Аналіз вимог до програмного забезпечення	16
2.1 Загальні положення-формулювання проблеми, дерево завдань, дерево проблем, дерево результатів	16
2.2 Опис предметної області.....	19
2.3 Постановка задачі	20
Висновки за розділом	21
3 Інструменти та алгоритми для управління проєктами	22
3.1 Kendo Manager.....	22
3.2 Intasker	23

					ІК92.280БАК.005 ПЗ			
Зм.	Лист	№ докум.	Підпис		Автоматизована система управління проєктами для малих комерційних підприємств. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Пилипчук М.В					Т	2	69
Перевірив	Мамедова К.Ю					КПІ ім. Ігоря Сікорського		
						Група ІК-92		
Затв.								

3.3	Trello	25
3.4	Asana	26
3.5	Microsoft Project	27
	Висновки за розділом	29
4	Вибір технологій та методів розробки.....	30
4.1	.NET Framework	30
4.2	Structured Query Language	30
4.3	Microsoft SQL Server Management Studio	31
4.4	Microsoft Visual Studio.....	33
4.5	Серіалізація і десеріалізація даних JSON в C#.....	34
	Висновки за розділом	35
5	Розробка програмного забезпечення	37
5.1	Структура системи	37
5.2	Модель зберігання даних	40
	Висновки за розділом	44
6	Тестування системи	45
6.1	Тестування елемента керування Chart.....	45
6.1.1	Модульне тестування.....	45
6.1.2	Інтеграційне тестування	46
6.1.3	Візуальне тестування	46
6.1.4	Сценарне тестування.....	47
6.1.5	Регресійне тестування.....	47
6.2	Тестування ProjectManager	48
6.3	Тестування класу ' ProjectManager`	51

6.4 Використання елемента керування Ganntt Chart.....	53
Висновки за розділом	58
ВИСНОВОК.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61

					ІК92.280БАК.005 ПЗ	Арк.
						4
Зм.	Лист	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

СУП - Система управління проєктами;

МП - малі підприємства;

БП - бізнес-процес;

ІТ - Інформаційні технології;

ПО - програмне забезпечення;

ІС - Інформаційна система;

СУБД - Система управління базами даними;

БД – База даних;

SSMS - Microsoft SQL Server Management Studio;

JSON – JavaScript Object Notation;

PK – Primary Key;

FK – Foreign Key.

					ІК92.280БАК.005 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

В даний час малі підприємства відіграють важливу роль в економіці кожної країни, включаючи Україну. МП є основою функціонування регіональної економіки, забезпечуючи значну кількість робочих місць, зайнятість населення та зниження рівня безробіття по всій країні. Вони є джерелом податкових надходжень, формуються державні бюджети, бо саме мале підприємництво відрізняється гнучкістю та, мають високий рівень економічної ініціативи. Однак управління проектами на таких підприємствах часто виявляється складним через обмежені ресурси та необхідність швидкого реагування на зміни зовнішнього середовища. Необхідність ініціювання проектного підходу та досягнення бажаних результатів малому бізнесу настільки ж необхідно, як і великим компаніям. При проведенні адаптації, з урахуванням особливостей та можливостей малого бізнесу, методів і інструментів використання проектного менеджменту більш ніж реально.

Перш за все, необхідно виділити, що існують певні проблеми та бар'єри, з якими стикаються МП при переході на методи управління проектами, серед них:

1) інформаційний бар'єр. він поєднує в собі, як і відсутність у бізнес-суспільстві культури проектного підходу, так і відсутність або недостатня кількість знань методів та інструментів проектного управління;

2) обмежені можливості. зазвичай мп не має великих бюджетів для навчання персоналу, що провокує недостатність знань та відсутність необхідних кваліфікацій;

3) бажання. дуже часто в колективі є наявний дефіцит професійної мотивації, який провокується недостатньою увагою керівництва у робочий процес, відсутністю заохочень за ініціативу. також наявно, коли персонал не має бажання підвищувати свої кваліфікації;

4) Дефіцит ресурсів.

Ефективне вирішення цих проблем можливо за рахунок автоматизації управління проектами та впровадження СУП, що зумовило вибір теми даного дипломного проекту.

					ІК92.280БАК.005 ПЗ	Арк.
						6
Зм.	Лист	№ докум.	Підпис	Дата		

Об'єкт дослідження: автоматизовані системи управління проєктами, їх застосування на малих підприємствах.

Предмет дослідження: основні характеристики систем управління проєктами – обсяг використаної пам'яті, складність інтерфейсу, вартість, тощо.

Мета і завдання: метою даного дипломного проєкту є покращення ефективності планування для малого бізнесу не технічної галузі, в умовах обмеженості ресурсів:

- 1) проаналізувати існуючі методи і технології управління проєктами;
- 2) вивчити специфіку та особливості малих підприємств;
- 3) розробити вимоги до автоматизованої системи управління проєктами для малих підприємств;
- 4) вибрати оптимальні технології для розробки системи;
- 5) розробити програмне забезпечення системи;
- 6) провести тестування системи.

Практичне значення отриманих результатів: результати даної роботи можуть бути використані малими підприємствами для автоматизації управління своїми проєктами, що дозволить поліпшити ефективність їх роботи, скоротити витрати і прискорити прийняття рішень.

Положення, які винесені на захист: розроблена система відрізняється від існуючих рішень урахуванням специфіки роботи малих підприємств, простотою впровадження і використання, а також гнучкістю і можливістю швидкої адаптації до мінливих умов зовнішнього середовища.

Структура роботи: дипломний проєкт складається з наступних розділів: введення, загальна характеристика предметної області і розв'язуваних завдань, аналіз вимог до програмного забезпечення, аналіз відомих програмних рішень, вибір технологій для розробки, Розробка програмного забезпечення, тестування системи, Висновки, список використаних джерел. Графічна частина включає схеми і діаграми, що ілюструють основні етапи розробки і функціонування системи.

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ І ВИРІШУВАНИХ ЗАВДАНЬ

Автоматизована система управління проєктами для малого бізнесу - це сфера, яка займається спрощенням та структуруванням процесів, пов'язаних з управлінням проєктами. Під "проєктами" в даному контексті ми маємо на увазі як виробничі завдання, так і внутрішні процеси підприємства.

1.1 Опис основного бізнес-процесу, який підлягає автоматизації

У цьому розділі детально опишемо бізнес-процес, який система буде автоматизувати. Основний БП, який підлягає автоматизації, пов'язаний з управлінням проєктами на малих підприємствах. Цей процес включає всі етапи, починаючи від створення проєкту до його завершення [6-11].

1) етап ініціації проєкту:

- визначення цілей і завдань проєкту;
- визначення обмежень та ризиків проєкту;
- складання бюджету та плану проєкту;
- затвердження проєкту керівництвом.

2) етап планування проєкту:

- розробка детального плану проєкту з визначенням завдань, термінів і ресурсів;
- розподіл завдань між учасниками проєкту;
- визначення залежностей між завданнями і створення графіка проєкту;
- оцінка ризиків та розробка стратегії управління ризиками.

3) етап виконання проєкту:

- призначення завдань виконавцям і контроль їх виконання;
- відстеження прогресу проєкту та внесення необхідних коригувань;
- управління комунікацією між учасниками проєкту;
- забезпечення своєчасного завершення завдань і контроль дотримання термінів.

					ІК92.280БАК.005 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

4) етап контролю та оцінки проєкту:

- моніторинг і контроль виконання завдань і досягнення цілей проєкту;
- оцінка якості роботи і задоволеності замовника;
- внесення необхідних коригувань для досягнення успіху проєкту;
- підготовка звітності про проєкт і його результативності.

5) етап завершення проєкту:

- оцінка та аналіз результатів проєкту;
- уроки, витягнуті з проєкту і документування досвіду;
- завершення роботи по проєкту і передача його замовнику;
- оцінка участі та результатів кожного учасника проєкту.

1.2 Склад і ієрархія завдань і підзадач, які вирішуються в програмі

Склад і ієрархії завдань і підзадач, які можуть бути включені в програмному забезпеченні, що розроблюється:

1) управління проєктами:

- створення нового проєкту;
- редагування інформації про проєкт;
- видалення проєкту;
- перегляд списку активних проєктів;
- перегляд списку завершених проєктів.

2) планування проєкту:

- визначення цілей і завдань проєкту;
- визначення термінів і пріоритетів завдань;
- розподіл ресурсів на завдання;
- створення графіка виконання проєкту;
- управління залежностями між завданнями.

3) управління завданнями:

- створення нового завдання;

					ІК92.280БАК.005 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

- редагування інформації про завдання;
 - призначення відповідальних виконавців;
 - відстеження прогресу виконання завдання;
 - управління пріоритетами та термінами завдань.
- 4) управління ресурсами:
- облік доступних ресурсів (людських, матеріальних і фінансових);
 - розподіл ресурсів на завдання і проєкти;
 - моніторинг використання ресурсів та їх доступності.
- 5) комунікація та спільна робота:
- обмін повідомленнями та коментарями щодо проєкту та завдань;
 - спільне редагування документів і файлів;
 - повідомлення про зміни та події в проєкті;
 - можливість проведення онлайн зустрічей і нарад.
- 6) моніторинг та звітність:
- відстеження прогресу проєкту і виконання завдань;
 - створення звітів про статус проєкту;
 - аналіз витрат часу і ресурсів на завдання;
 - візуалізація даних про проєкт у вигляді графіків і діаграм.
- 7) управління ризиками та змінами:
- ідентифікація та оцінка ризиків проєкту;
 - планування та реалізація заходів з управління ризиками;
 - облік і управління змінами в проєкті;
 - аналіз та оцінка впливу змін на проєкт.

1.3 Схеми вхідних і вихідних інформаційних потоків для кожного завдання і підзадачі

Кожна задача і підзадача має свої вхідні і вихідні потоки інформації. Наприклад, для завдання "створення нового проєкту" вхідний потік інформації – це

					ІК92.280БАК.005 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

вимоги до проекту (цілі, терміни, ресурси), а вихідний потік – це створений проект із заданими параметрами.

1) Ініціація – створення, проекту (рисунок 1.1):

а. вхідний інформаційний потік:

- назва;
- дата початку;
- дата кінця;

б. вихідний інформаційний потік:

- результат створення проекту.

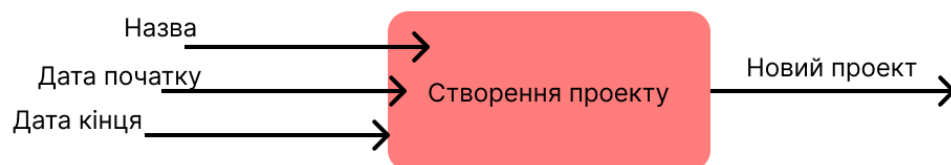


Рисунок 1.1 — Схема потоків для створення проекту.

2) Створення задачі (рисунок 1.2):

а. вхідний інформаційний потік:

- завдання;
- інформація про причетність до проекту;
- терміни виконання;
- виконавець;
- «батьківська» задача, якщо наявна.

б. вихідний інформаційний потік:

- результат створення задачі у проекті.

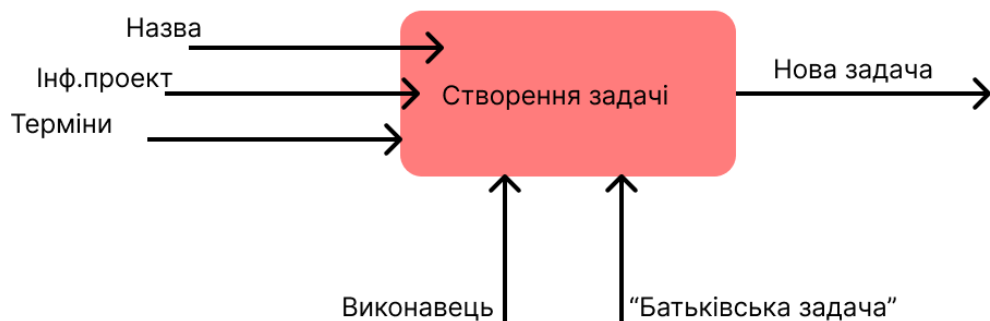


Рисунок 1.2 — Схема потоків для створення задачі.

3) Редагування задачі (рисунок 1.3):

а. вхідний інформаційний потік:

- зміни, що вносяться в початкові дані;
- зміна проценту виконання задачі;

б. вихідний інформаційний потік:

- результат внесення змін у задачу.

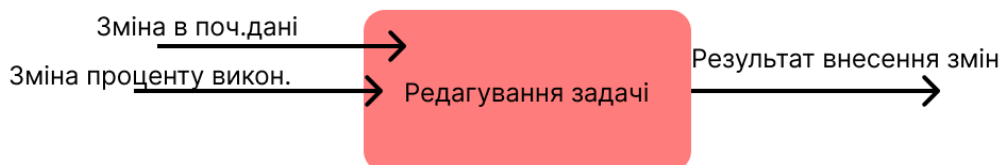


Рисунок 1.3 — Схема потоків для редагування задачі.

4) Відстеження плану та прогресу (рисунок 1.4):

а. вхідний інформаційний потік:

- назва задачі;
- терміни задачі;
- процент виконання задачі;

б. вихідний інформаційний потік:

- створена аналітична діаграма.

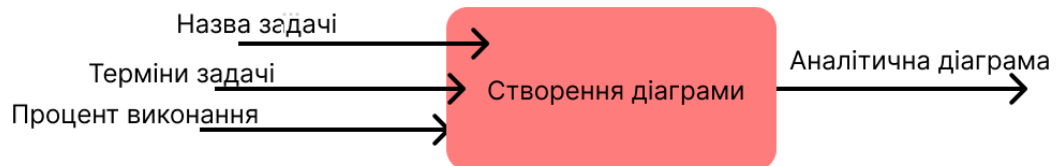


Рисунок 1.4 — Схема потоків для створення діаграми.

1.4 Послідовність вирішення завдань і підзадач

Завдання і підзадачі вирішуються в певній послідовності. Наприклад, перед створенням завдань в проєкті необхідно створити сам проєкт, визначити вимоги до нього, призначити учасників проєкту і задати початкові параметри. Після цього можна переходити до створення завдань всередині проєкту. Більш детально побачити послідовність виконання завдань можна на кресленнику ІК92.280БАК.005 Д1.

1.5 Опис програмних модулів і їх функцій

Програма може складатися з декількох модулів, кожен з яких виконує певні функції:

- модуль діаграм: призначений для візуалізації плану проєкту у вигляді діаграми.
- модуль списку завдань: дозволяє створювати, редагувати та відстежувати завдання всередині проєкту.
- модуль відстеження проблем: використовується для відстеження та управління проблемами та помилками, які виникають у процесі виконання проєкту.
- модуль управління часом: допомагає контролювати час, витрачений на виконання завдань і проєктів, а також оптимізувати робочий час учасників проєкту.
- модуль управління змінами: дозволяє контролювати зміни в проєкті і управляти ними.

- модуль управління ризиками: призначений для ідентифікації, аналізу та управління ризиками, пов'язаними з проектом.
- модуль управління ресурсами: дозволяє ефективно розподіляти та контролювати використання ресурсів у проекті.
- модуль звітів: надає можливість створення звітів про процес виконання проекту.
- модуль управління бюджетом та витратами: використовується для планування та контролю бюджету проекту та пов'язаних з ним витрат.
- центр сповіщень: надає сповіщення про важливі події та зміни в проекті.
- модуль комунікації: забезпечує обмін інформацією між учасниками проекту.

1.6 Опис структури даних та алгоритмів

Для забезпечення роботи програми використовуються різні структури даних, включаючи списки завдань, списки проектів, списки учасників проекту і т.д. кожен проект, завдання або учасник проекту являють собою об'єкти з певними атрибутами і методами.

Алгоритми програми забезпечують виконання ключових функцій, таких як створення нових проектів і завдань, призначення учасників проекту, відстеження статусу завдань, управління ресурсами і звітність [1].

В процесі управління проектами в програмі активно використовуються алгоритми, які допомагають вирішити різні завдання. Наприклад:

Алгоритми вирішення завдань планування: використовуються для визначення оптимального графіка виконання завдань проекту. Вони дозволяють згенерувати майже оптимальний набір графіків для кожного співробітника і необхідних ресурсів для успішного завершення проекту. Такі завдання можуть бути розділені на проблеми призначення графіка, проблеми тільки планування і спільні завдання призначення або планування.

					ІК92.280БАК.005 ПЗ	Арк.
						14
Зм.	Лист	№ докум.	Підпис	Дата		

Алгоритми призначення ресурсів: використовуються для ефективного розподілу ресурсів за завданнями проєкту. Це може включати проблеми з обмеженими ресурсами, призначення декількох ресурсів на завдання проєкту, часткове та багаторазове призначення тощо.

Евристичні та оптимальні алгоритми: використовуються для пошуку найбільш оптимального рішення для задачі проєкту. Евристика може допомогти скоротити час пошуку рішення, пропонуючи приблизне рішення. Серед них можна виділити алгоритми конструкції, пошукові алгоритми і гібридні алгоритми.

Висновки за розділом

У цьому розділі було представлено загальний опис предметної області і вирішуваних завдань.

В рамках дослідження було розгорнуто описано основний бізнес процес, що підлягає автоматизації. Описано програмні модулі, процес обміну даними. Визначено алгоритми, які дозволяють ефективно вирішувати завдання, пов'язані з управлінням проєктами, і забезпечує гнучкість в адаптації під специфіку конкретного проєкту або команди.

Загалом, глава 1 підкреслила важливість інтегрованих систем управління проєктами та їх роль у сучасному бізнесі. Було показано, що вибір відповідного програмного забезпечення для управління проєктами може мати значний вплив на ефективність роботи над проєктом та його успішне завершення.

2 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Загальні положення-формулювання проблеми, дерево завдань, дерево проблем, дерево результатів

Проблема, яку наш проєкт прагне вирішити, полягає в тому, що багато команд мають труднощі з координацією роботи, обміном інформацією та контролем над ходом виконання завдань під час управління проєктами. Ця проблема посилюється зі збільшенням розміру команди та складності проєкту [1-5]. Для більш наочного представлення проблеми, завдань і результатів можна побудувати дерева завдань, проблем і результатів.

Дерево завдань (рисунок 2.1) включає такі основні елементи: управління завданнями, управління ресурсами, управління комунікаціями та управління ризиками. Кожне з цих завдань включає підзадачі, які необхідно вирішити для досягнення спільної мети. Дерево завдань для нашого проєкту:

- 1) управління проєктами:
 - створення нового проєкту;
 - редагування інформації про проєкт;
 - видалення проєкту;
 - перегляд списку активних проєктів;
 - перегляд списку завершених проєктів.
- 2) планування проєкту:
 - визначення цілей і завдань проєкту;
 - розподіл ресурсів на завдання;
 - створення графіка виконання проєкту;
 - управління залежностями між завданнями.
- 3) управління завданнями:
 - створення нового завдання;
 - редагування інформації про завдання;
 - призначення відповідальних виконавців;
 - відстеження прогресу виконання завдання;;

- управління пріоритетами та термінами завдань.
- 4) управління ресурсами:
 - облік доступних ресурсів (людських, матеріальних і фінансових);
 - розподіл ресурсів на завдання та проєкти;
 - моніторинг використання ресурсів та їх доступності.
- 5) комунікація та спільна робота:
 - обмін повідомленнями та коментарями щодо проєкту та завдань;
 - спільне редагування документів і файлів;
 - повідомлення про зміни та події в проєкті;
 - можливість проведення онлайн-зустрічей і нарад;
 - моніторинг та звітність;
 - відстеження прогресу проєкту та виконання завдань;
 - створення звітів про статус проєкту;
 - аналіз витрат часу і ресурсів на завдання;
 - візуалізація даних про проєкт у вигляді графіків і діаграм.
- 6) управління ризиками та змінами;
 - ідентифікація та оцінка ризиків проєкту;
 - планування та реалізація заходів з управління ризиками;
 - облік та управління змінами в проєкті.
- 7) аналіз та оцінка впливу змін на проєкт.



Рисунок 2.1 — Дерево завдань.

Дерево проблем відображає проблеми, які виникають при управлінні проектами, включаючи відсутність координації між членами команди, труднощі з відстеженням прогресу завдань, труднощі з управлінням ресурсами та ризиками.

Дерево проблем (див. креслення ІК92.280БАК.005 Д2) проекту:

- 1) відсутність централізованої системи управління проектами;
- 2) недостатня прозорість і контроль завдань;
- 3) недостатня комунікація та співпраця між учасниками проекту;
- 4) низька ефективність планування та розподілу ресурсів;
- 5) відсутність систематичного моніторингу та звітності про проекти;
- 6) неефективне управління ризиками та змінами в проектах.

Дерево результатів показує очікувані результати від реалізації проекту, включаючи поліпшене управління завданнями, ефективне управління ресурсами, поліпшену комунікацію в команді і зниження ризиків. Дерево результатів (див. креслення ІК92.280БАК.005 Д3) проекту:

- 1) покращена ефективність управління проектами на малих підприємствах;
- 2) підвищення прозорості та контролю завдань;
- 3) покращена комунікація та співпраця між учасниками проекту;
- 4) оптимізація планування та розподілу ресурсів;
- 5) покращений моніторинг та звітність про проекти;
- 6) ефективне управління ризиками та змінами в проектах.

2.2 Опис предметної області

Предметна область проєкту - це управління проєктами. Управління проєктами включає планування, організацію, управління та контроль над ресурсами для досягнення конкретних цілей у рамках проєкту. Це включає такі аспекти, як управління завданнями, ресурсами, комунікаціями та ризиками [13]. У даній області важливо забезпечити ефективне планування, організацію та виконання проєктів, щоб досягти поставлених цілей в задані терміни і з обмеженими ресурсами. Автоматизація управління проєктами на малих підприємствах допомагає оптимізувати процеси, підвищити прозорість і контроль, скоротити час і ресурси, а також поліпшити комунікацію і співпрацю між учасниками проєктів [11]. Основні особливості предметної області включають:

1) обмеженість ресурсів: малим підприємствам часто не вистачає ресурсів (людей, фінансів, часу), тому ефективне управління ресурсами та їх оптимальний розподіл стають важливими завданнями.

2) гнучка структура: малі підприємства, як правило, не мають складної ієрархічної структури, що вимагає гнучкого підходу до управління проєктами та прийняття рішень.

3) комунікація та співпраця: учасники проєктів часто знаходяться на різних місцях або навіть працюють віддалено. тому важливо забезпечити ефективне спілкування та співпрацю між ними, щоб забезпечити гармонійну взаємодію та обмін інформацією.

4) гнучкість і адаптивність: малим підприємствам часто потрібно швидко реагувати на зміни і адаптуватися до нових умов. тому система управління проєктами повинна бути гнучкою і дозволяти швидко вносити зміни в плани і завдання.

Детальний опис предметної області допоможе читачам зрозуміти контекст і специфіку вашої роботи, а також зрозуміти, чому автоматизована система управління проєктами має важливість для малих підприємств. Зверніть увагу на особливості та виклики, з якими стикаються малі підприємства, і поясніть,

як ваша система допоможе вирішити ці проблеми та покращити ефективність управління проектами.

2.3 Постановка задачі

Завдання нашого проекту - це розробка програмного забезпечення для управління проектами, яке буде допомагати командам більш ефективно планувати свою роботу та відстежувати наскільки проєкт завершено. Наше програмне забезпечення має бути гнучким та адаптивним, щоб відповідати різним типам проєктів та ресурсам.

Завдання роботи:

1) аналіз існуючих проблем і недоліків в управлінні проектами на малих підприємствах, включаючи обмеження ресурсів, недостатню прозорість і контроль, неефективну комунікацію і співпрацю.

2) вивчення та аналіз існуючих програмних рішень для управління проектами та їх застосовності до малого бізнесу.

3) визначення вимог до автоматизованої системи управління проектами, враховуючи особливості та потреби малих підприємств.

4) вибір відповідних технологій та інструментів для розробки системи, що забезпечують гнучкість, зручність використання та масштабованість.

5) розробка структури та функціональності системи, включаючи модулі для планування, управління завданнями, обліку ресурсів, комунікації та моніторингу проєктів.

6) реалізація програмного забезпечення на основі розробленої структури і функціональності.

7) проведення тестування системи для перевірки її функціональності, надійності та відповідності вимогам.

8) оцінка ефективності та переваг системи, порівняння її результатів з існуючими підходами до управління проектами на малих підприємствах.

					ІК92.280БАК.005 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

Висновки за розділом

В результаті аналізу вимог до програмного забезпечення була сформульована основна проблема, яку ми прагнемо вирішити — це складності, пов'язані з управлінням проєктами, особливо у невеликих командах і складних проєктах. Визначено основні завдання, які має вирішувати наше програмне забезпечення: управління завданнями, ресурсами, комунікаціями та ризиками.

Було надано опис предметної області, який включає в себе різні аспекти управління проєктами. Завдання розробки програмного забезпечення було чітко сформульовано, і воно полягає у створенні інструменту, який допоможе командам ефективніше керувати своїми проєктами, забезпечуючи гнучкість і адаптивність під різні типи проєктів і команд.

В цілому, цей розділ підкреслює важливість глибокого розуміння вимог до програмного забезпечення перед його розробкою, щоб забезпечити його ефективність і адекватність для вирішення поставлених завдань.

					ІК92.280БАК.005 ПЗ	Арк.
						21
Зм.	Лист	№ докум.	Підпис	Дата		

3 ІНСТРУМЕНТИ ТА АЛГОРИТМИ ДЛЯ УПРАВЛІННЯ ПРОЄКТАМИ

Розглянемо кілька прикладів програмного забезпечення для управління проєктами, що використовуються різними типами підприємств або команд.

3.1 Kendo Manager

Kendo Manager - це програмне забезпечення для управління проєктами на рівні підприємства, яке підходить для менеджерів проєктів, учасників проєкту, організації проєктів, компаній та клієнтів для спілкування та відстеження статусу проєкту в будь-який момент з будь-якого місця.

Він надає необхідну допомогу для більш ефективного розподілу та організації людських ресурсів у проєкті. Ключовим фактором успішного проєкту є ефективна організація людських ресурсів та спілкування в проєкті. Зовнішній вигляд представлено на рисунку 3.1.

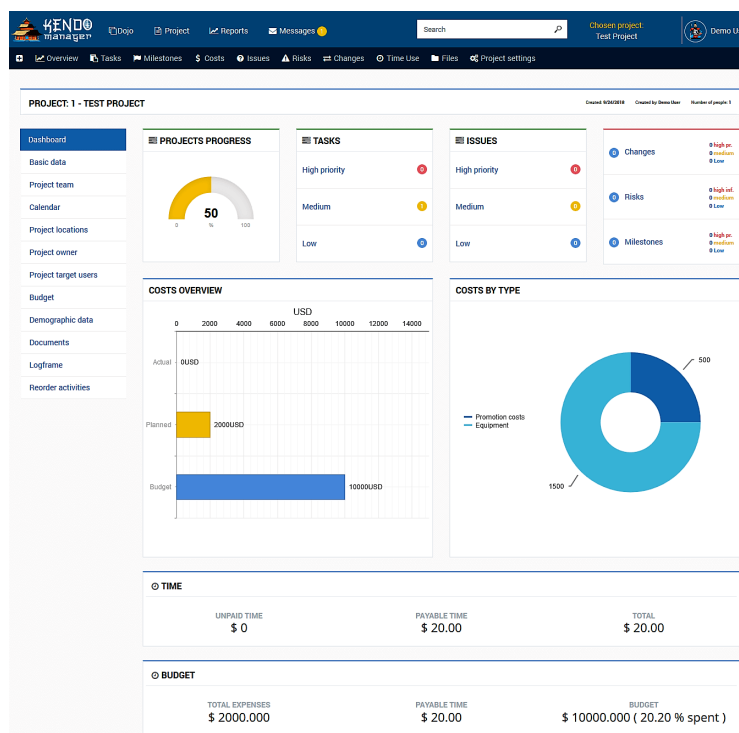


Рисунок 3.1 — UI додатку Kendo Manager.

Kendo Manager гнучке програмне забезпечення для управління проєктами, яке дозволяє управляти різними типами проєктів. Воно складається з багатьох модулів, які можна знайти в будь-якому програмному забезпеченні для управління

проектами на рівні підприємства. Він включає діаграму Ганта, список завдань, відстеження проблем, управління часом, управління змінами, управління ризиками, Управління ресурсами, модуль звітності, модуль бюджету та витрат, центр сповіщень та модуль спілкування.

Переваги:

- інтуїтивний користувальницький інтерфейс, що полегшує використання і навігацію по системі;
- гнучка система планування проєктів з можливістю створення завдань, визначення залежностей, призначення ресурсів і установки термінів;
- можливість візуального відображення графіків виконання проєкту і діаграм Ганта;
- інтеграція з іншими інструментами та сервісами, такими як електронна пошта та хмарні сховища;
- вбудована система управління комунікацією, що дозволяє обмінюватися повідомленнями і файлами всередині проєкту.

Недоліки:

- обмежена безкоштовна версія зі зниженим функціоналом, що може зажадати придбання платної ліцензії для отримання повного набору можливостей;
- відсутність деяких просунутих функцій, які можуть бути корисні для складних проєктів з великим числом учасників і завдань.

3.2 Intasker

Intasker пропонує два тарифних плани: базовий і Про. Базовий тарифний план безкоштовний і включає в себе можливість створення до трьох проєктів, запрошення до п'яти осіб в команду і надає 500 МБ для зберігання даних. З іншого боку, план коштує 10 євро на місяць і пропонує необмежену кількість проєктів, команд та 25 ГБ для зберігання даних. Зовнішній вигляд представлено на рисунку 3.2.

					ІК92.280БАК.005 ПЗ	Арк.
						23
Зм.	Лист	№ докум.	Підпис	Дата		

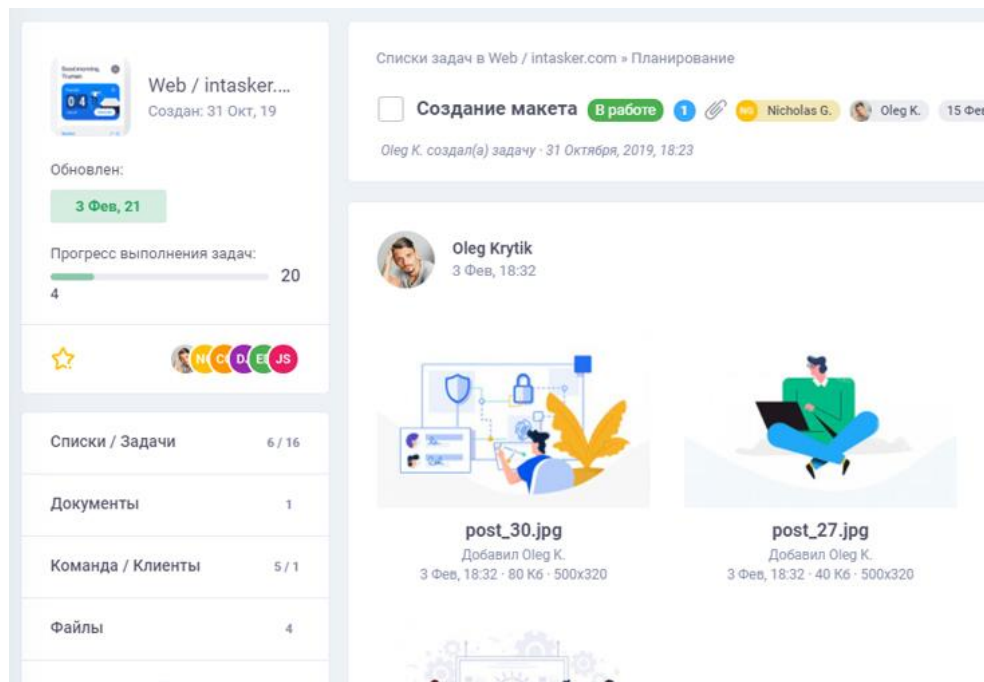


Рисунок 3.2 — UI додатку Intasker.

Пошукові алгоритми, які частково завершені і поєднуються з новими рішеннями в надії знайти ще краще оптимальне рішення. Приклади цього можуть включати штучні алгоритми колонії бджіл, алгоритми сходження на пагорб та диференціальну еволюцію. Гібридні алгоритми, які включають як пошукові, так і конструктивні підходи.

Переваги:

- простий і інтуїтивний інтерфейс, що спрощує впровадження і використання системи для користувача.
- можливість створення завдань, встановлення термінів, управління пріоритетами та призначення відповідальних осіб.
- інтеграція з електронною поштою, що дозволяє отримувати повідомлення та оновлення по електронній пошті.
- можливість створення звітів і діаграм, що відображають прогрес проєкту і розподіл ресурсів.
- доступність як веб-версії, так і мобільних додатків для зручності використання в будь-якому середовищі.

Недоліки:

- відсутність деяких вдосконалених функцій, таких як автоматизація завдань та інтеграція з іншими інструментами та послугами.
- обмежені можливості налаштування та гнучкості, особливо для складних проєктів з унікальними вимогами.

3.3 Trello

Trello - це онлайн-інструмент управління проєктами, який використовує методологію "дошки, списки та картки" для організації та відстеження завдань та проєктів. Зовнішній вигляд представлено на рисунку 3.3.

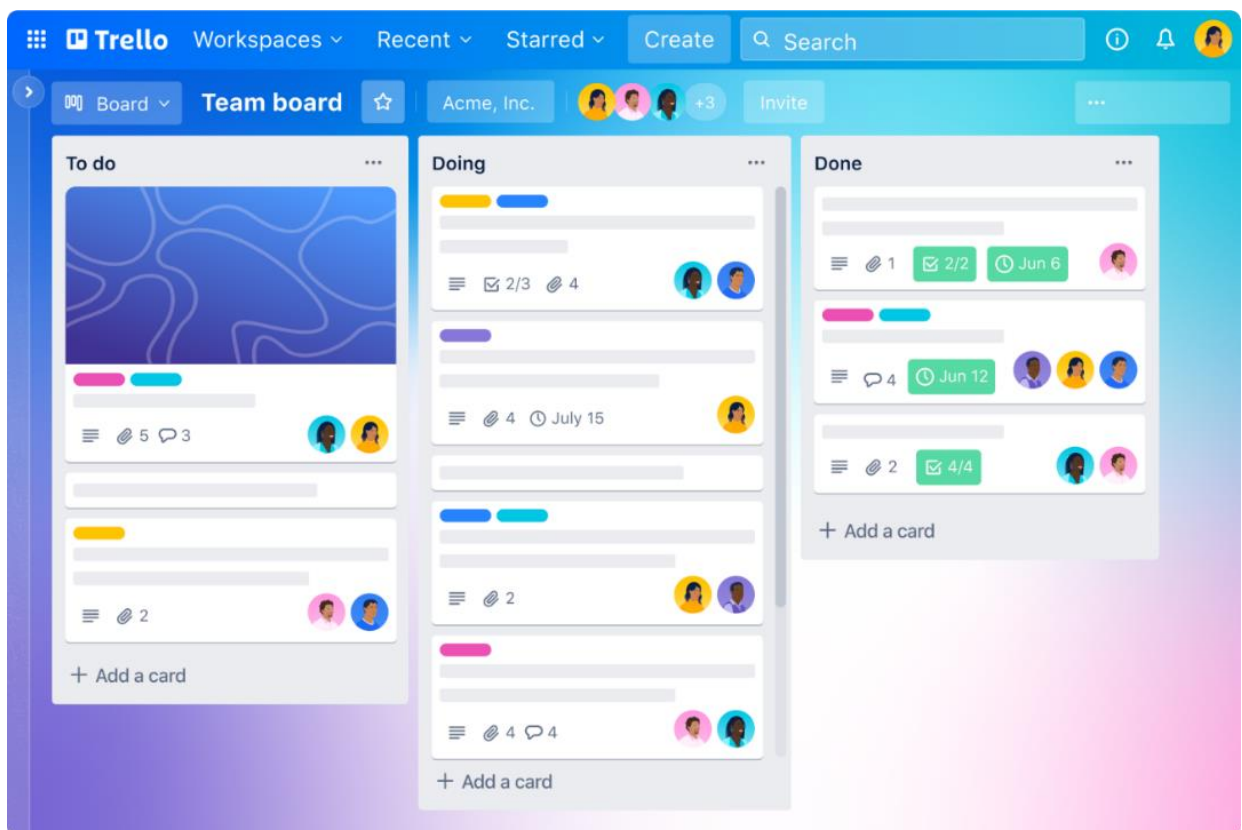


Рисунок 3.3 — UI додатку Trello.

Переваги:

- простота використання: інтуїтивний інтерфейс робить Trello легким в освоєнні і використанні навіть для нових користувачів.

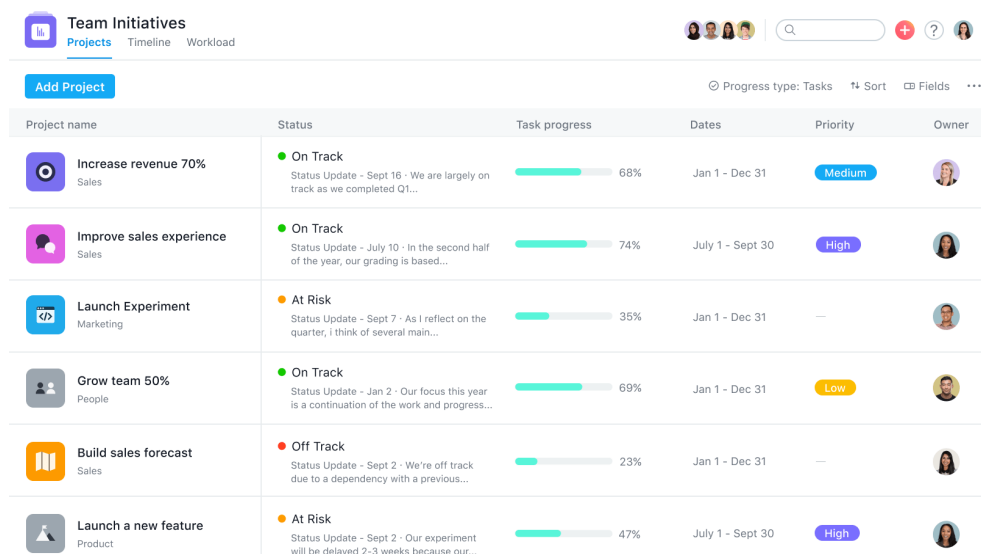
- гнучкість: створення дощок, списків і карток дозволяє гнучко організовувати завдання і проєкти.
- візуальна наочність: кольорове кодування, мітки і можливість перетягування карток дозволяють легко відстежувати і управляти завданнями.
- співпраця: Trello надає можливість запрошувати учасників, коментувати завдання та взаємодіяти з командою.
- інтеграція: можна інтегрувати Trello з іншими інструментами, такими як Slack, Google Drive та іншими.

Недоліки:

- обмежені функціональні можливості: Trello надає базовий набір функцій і може не відповідати вимогам більш складних проєктів.
- відсутність додаткових функцій: деякі функції, такі як облік ресурсів або Розширена звітність, відсутні в Trello.

3.4 Asana

Asana - це онлайн-платформа управління проєктами, яка пропонує широкий спектр функцій для планування, управління завданнями, комунікації та звітності. Зовнішній вигляд представлено на рисунку 3.4.



Project name	Status	Task progress	Dates	Priority	Owner
Increase revenue 70% Sales	● On Track Status Update - Sept 16 - We are largely on track as we completed Q1...	68%	Jan 1 - Dec 31	Medium	
Improve sales experience Sales	● On Track Status Update - July 10 - In the second half of the year, our grading is based...	74%	July 1 - Sept 30	High	
Launch Experiment Marketing	● At Risk Status Update - Sept 7 - As I reflect on the quarter, I think of several main...	35%	Jan 1 - Dec 31	—	
Grow team 50% People	● On Track Status Update - Jan 2 - Our focus this year is a continuation of the work and progress...	69%	Jan 1 - Dec 31	Low	
Build sales forecast Sales	● Off Track Status Update - Sept 2 - We're off track due to a dependency with a previous...	23%	Jan 1 - Dec 31	—	
Launch a new feature Product	● At Risk Status Update - Sept 2 - Our experiment will be delayed 2-3 weeks because our...	47%	July 1 - Sept 30	High	

Рисунок 3.4 — UI додатку Asana.

Переваги:

- багатий набір функцій: Asana пропонує безліч функцій, включаючи створення завдань, визначення залежностей, управління ресурсами, відстеження прогресу та звітність.
- гнучкість: користувачі можуть налаштувати Asana під свої конкретні потреби і процеси роботи.
- спілкування та співпраця: можливість обміну повідомленнями, коментування завдань та співпраці над проєктами.
- візуалізація даних: Asana пропонує графіки та діаграми Ганта для наочного відображення прогресу проєктів.
- інтеграція: інтеграція з іншими інструментами та сервісами, такими як Slack, Google Drive, Dropbox та багатьма іншими.

Недоліки:

- складне керування: інтерфейс Asana може зайняти певний час для освоєння, особливо для нових користувачів.
- обмежена безкоштовна версія: для доступу до розширених функцій та інтеграцій може знадобитися платна підписка.

3.5 Microsoft Project

Microsoft Project - це програмне забезпечення для управління проєктами, яке надає розширений набір функцій для планування, управління ресурсами та контролю прогресу проєктів. Зовнішній вигляд представлено на рисунку 3.5.

					ІК92.280БАК.005 ПЗ	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

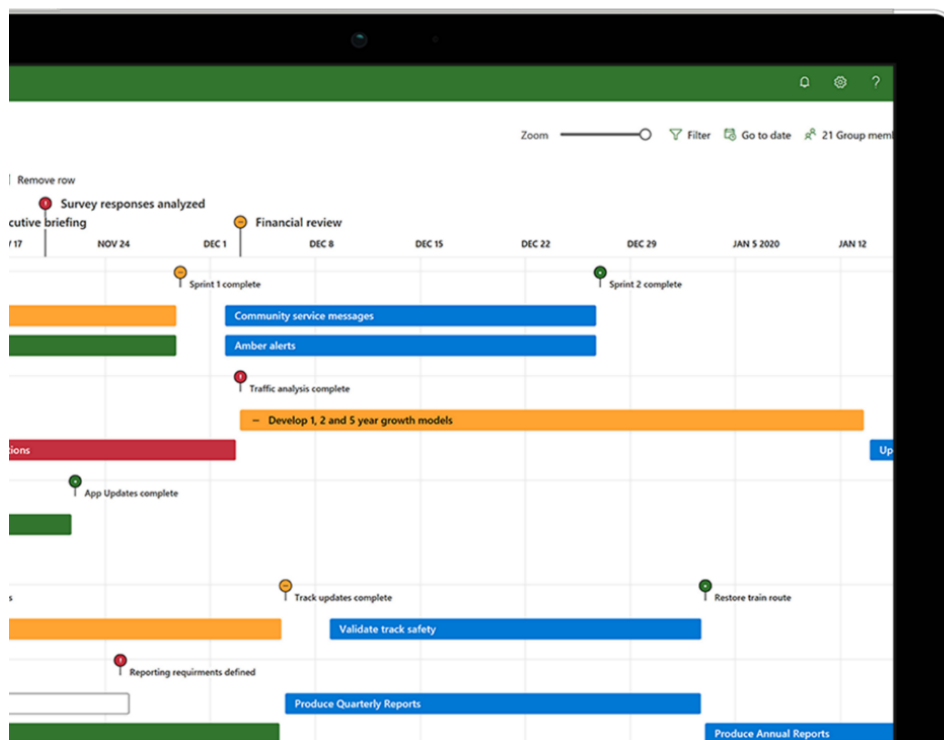


Рисунок 3.5 — UI додатку Microsoft Project.

Переваги:

- потужні функції планування: Microsoft Project пропонує інструменти для планування, визначення залежностей, аналізу ресурсів та бюджету проєктів.
- інтеграція з іншими програмами Microsoft: Програма може інтегруватися з Excel, SharePoint та іншими програмами Microsoft, що забезпечує співпрацю та обмін даними.
- гнучкість: Microsoft Project дозволяє налаштовувати і адаптувати систему під унікальні вимоги проєктів.
- контроль і звітність: Інструменти для моніторингу прогресу, створення звітів і візуалізації даних.

Недоліки:

- складний інтерфейс: інтерфейс Microsoft Project може бути складним для нових користувачів і вимагати часу для освоєння.
- орієнтація на великі проєкти: Microsoft Project більше підходить для складних та великих проєктів, що може бути зайвим для малого бізнесу.

– висока вартість і вимоги до ліцензування: Microsoft Project є платним програмним забезпеченням і вимагає відповідних ліцензій для використання.

Висновки за розділом

Після проведення аналізу відомих програмних рішень для управління проектами, включаючи Kendo Manager, Intasker та інші, можна зробити наступні висновки:

– кожне програмне рішення має свої переваги і недоліки. Наприклад, Kendo Manager пропонує широкий спектр функцій та зручний користувацький інтерфейс, що робить його привабливим для малого бізнесу. Однак, у нього також є обмеження в налаштуванні і гнучкості. З іншого боку, Intasker може бути більш гнучким та налаштованим, але може бути складним для нових користувачів завдяки своєму інтерфейсу.

– при виборі програмного рішення необхідно враховувати вимоги проекту, бюджет, потреби користувачів і переваги команди розробників. Важливо вибрати рішення, яке найкраще відповідає унікальним потребам та цілям вашого проекту.

– проведення додаткових досліджень та прототипування може допомогти у виборі найбільш підходящого програмного рішення. Це дозволить оцінити функціональність, зручність використання і відповідність вимогам проекту.

4 ВИБІР ТЕХНОЛОГІЙ ТА МЕТОДІВ РОЗРОБКИ

4.1 .NET Framework

.NET Framework - це платформа розробки програмного забезпечення, розроблена компанією Microsoft. Вона надає розробникам набір інструментів і бібліотек для створення і запуску додатків під управлінням операційних систем Windows [17]

Переваги .NET Framework:

- великі бібліотеки: .NET Framework володіє широким спектром бібліотек, що надають готові компоненти і функції для розробки додатків, включаючи роботу з базами даних, мережевими протоколами, веб-технологіями та іншими;
- підтримка мов програмування: .NET Framework підтримує різні мови програмування, такі як C#, VB.NET, F# та інші, що дає можливість вибрати найбільш зручний і знайомий мову розробки;
- потужне інтегроване середовище розробки: Microsoft Visual Studio надає розробникам потужне середовище розробки з інструментами для налагодження, профілювання та створення користувальницьких інтерфейсів;

Недоліки .NET Framework:

- обмеження операційних систем: .NET Framework працює тільки на операційних системах Windows, що обмежує переносимість додатків на інші платформи;
- залежність від встановленої версії: розробка під .NET Framework вимагає встановленої версії відповідної платформи на цільовій системі, що може викликати проблеми сумісності;
- велике навантаження на ресурси: використання .NET Framework може вимагати вищих системних ресурсів порівняно з деякими іншими технологіями;

4.2 Structured Query Language

SQL (Structured Query Language) — це мова програмування, яка

					ІК92.280БАК.005 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

використовується для роботи з реляційними базами даних. SQL дозволяє створювати, змінювати та керувати базами даних, а також виконувати запити для отримання та обробки даних.

Переваги SQL:

- широке поширення: SQL є стандартом для роботи з реляційними базами даних і підтримується безліччю СУБД, таких як Microsoft SQL Server, MySQL, Oracle та інші;
- потужна мова запитів: SQL надає потужні можливості для формування запитів до баз даних, включаючи вибірку, фільтрацію, сортування, агрегацію та з'єднання даних;
- цілісність даних;
- SQL забезпечує механізми для підтримки цілісності даних, включаючи обмеження, зв'язки та транзакції;
- управління даними: SQL дозволяє виконувати операції з додавання, оновлення та видалення даних у базі даних;

Недоліки SQL:

- обмеження на роботу з не реляційними даними: SQL оптимізований для роботи з реляційними базами даних і може бути менш ефективним при роботі з не реляційними або напів структурованими даними;
- складність деяких запитів: деякі складні запити можуть вимагати вдосконалених знань SQL та оптимізації для досягнення оптимальної продуктивності;
- залежність від конкретної СУБД: SQL мова стандартизований, але кожна СУБД може мати деякі відмінності і особливості в реалізації SQL запитів;

4.3 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio - це інтегроване середовище розробки та управління базами даних SQL Server від Microsoft. SSMS надає розробникам та адміністраторам баз даних набір інструментів для роботи з SQL Server, включаючи

					ІК92.280БАК.005 ПЗ	Арк.
						31
Зм.	Лист	№ докум.	Підпис	Дата		

створення, зміну та видалення баз даних, Виконання запитів, налаштування безпеки, моніторинг та оптимізацію продуктивності.

Переваги Microsoft SQL Server Management Studio:

- Інтуїтивний інтерфейс: SSMS має зручний і інтуїтивно зрозумілий інтерфейс, що спрощує роботу з базами даних SQL Server.
- Потужні інструменти розробки: SSMS надає широкий спектр інструментів для створення, редагування та налагодження сценаріїв SQL, збережених процедур, функцій та подань.
- Адміністрування баз даних: SSMS дозволяє керувати різними аспектами баз даних, включаючи управління користувачами, налаштування безпеки, створення планів виконання запитів та моніторинг продуктивності.
- Інтеграція з іншими інструментами: SSMS інтегрується з іншими інструментами Microsoft, такими як Visual Studio, що забезпечує єдиний робочий простір для розробки та управління базами даних.
- Підтримка розширень: SSMS підтримує можливість установки додаткових розширень і плагінів для розширення функціональності.

Недоліки Microsoft SQL Server Management Studio:

- Обмеження на роботу тільки з базами даних SQL Server: SSMS призначений спеціально для роботи з базами даних SQL Server і не підтримує інші реляційні або не реляційні бази даних.
- Залежність від версії SQL Server: SSMS може вимагати відповідну версію SQL Server для повної підтримки всіх функцій і можливостей.
- Ресурсоємність: при роботі з великими базами даних і складними запитами, SSMS може споживати значні ресурси комп'ютера.

Microsoft SQL Server Management Studio є потужним інструментом для розробки і управління базами даних SQL Server. При виборі SSMS важливо враховувати сумісність з версією SQL Server, потреби в розробці та управлінні базами даних, а також рівень знайомства команди з цим середовищем розробки.

					ІК92.280БАК.005 ПЗ	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

4.4 Microsoft Visual Studio

Microsoft Visual Studio-це інтегроване середовище розробки (IDE), розроблене компанією Microsoft. Вона надає розробникам широкий набір інструментів і функцій для створення різних типів додатків, включаючи веб-додатки, настільні додатки, мобільні додатки та інші.

Переваги Microsoft Visual Studio:

- Потужний редактор коду: Visual Studio надає потужний редактор коду з функціональністю автозаповнення, підсвічування синтаксису, Налаштування та іншими інструментами, які полегшують процес написання коду.
- Багатий вибір шаблонів проєктів: Visual Studio пропонує безліч шаблонів проєктів, що спрощує початок роботи з розробкою і надає базову структуру проєкту.
- Інтеграція з іншими інструментами та сервісами: Visual Studio інтегрується з іншими інструментами та сервісами Microsoft, такими як Azure, SQL Server, GitHub та багатьма іншими, що забезпечує ефективну розробку та розгортання додатків.
- Підтримка різних мов програмування: Visual Studio підтримує різні мови програмування, включаючи C#, VB.NET, F#, C++, JavaScript, Python та інші, що дозволяє розробникам вибрати найбільш підходящу мову для своїх проєктів.
- Широкі інструменти для налагодження та тестування: Visual Studio надає інструменти для налагодження коду, профілювання продуктивності, модульного та функціонального тестування додатків.

Недоліки Microsoft Visual Studio:

- Високе навантаження на системні ресурси: Visual Studio може споживати значні ресурси комп'ютера, особливо при роботі з великими проєктами.
- Складне навчання: деякі функції та інструменти Visual Studio можуть вимагати час для освоєння і вивчення.

– Обмеження на розробку під платформу Windows: Visual Studio є основним інструментом розробки для створення додатків під операційну систему Windows, що обмежує його застосовність для багато платформної розробки.

Microsoft Visual Studio надає потужні інструменти для розробки різних типів додатків. Вибираючи Visual Studio, важливо враховувати вимоги до розробки, цільову платформу, доступність інтеграції з іншими інструментами та рівень знайомства команди з певним середовищем розробки.

4.5 Серіалізація і десеріалізація даних JSON в C#

Існує багато методів та алгоритмів збереження даних в файл та перетворення їх у бінарний формат даних, але не завжди вони можуть впоратись із складними структурами та об'єктами, які тягнуть за собою велику кількість параметрів та атрибутів. Саме з цією проблемою повинен впоратись механізм серіалізації. Вона представляє собою метод перетворення певних даних – об'єкта чи структури, у байтовий потік, який вже можна буде записати у файл або зберегти його у пам'яті на певний час. Байтовий формат є зручним та ефективним методом збереження даних у пам'яті. Зазвичай серіалізація використовується для передачі об'єктів та структур по мережі, за межі додатку, або для збереження об'єктів всередині програми або бази даних. Серіалізація допомагає створювати ПО, яке має можливість «запам'ятовувати» свій стан.

В той час, десеріалізація – це метод, який обертає процес серіалізації. Він допомагає прочитати серіалізований об'єкт або структуру, та повернути його в первісний вигляд. Загальний алгоритм перетворень представлено на рисунку 4.1.

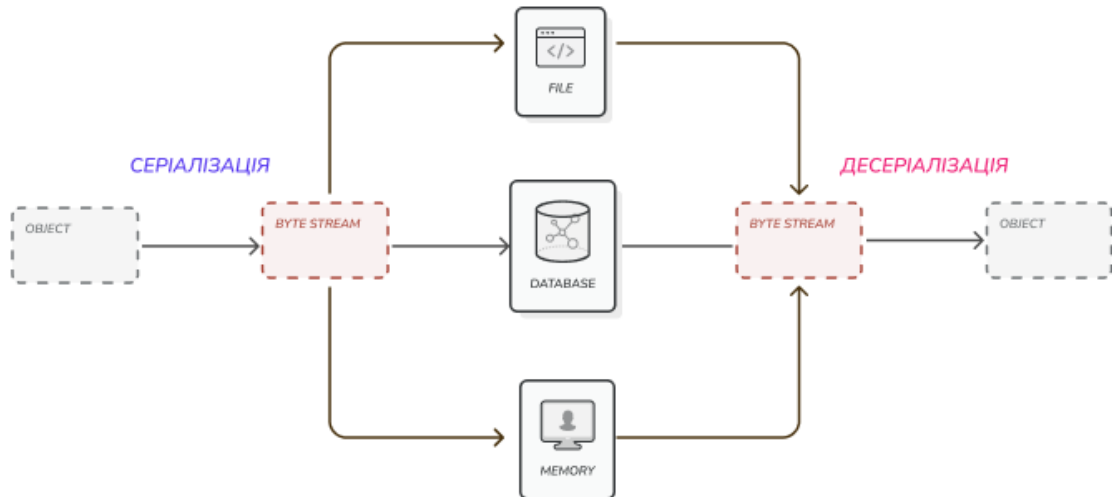


Рисунок 4.5 — Алгоритм роботи методів серіалізації та десеріалізації

Висновки за розділом

Після розгляду технологій. NET Framework і SQL, методів роботи з даним, для розробки автоматизованої системи управління проектами можна зробити наступні висновки:

1) . NET Framework пропонує широкий спектр інструментів та підтримку різних мов програмування, таких як C#, VB.NET і F#. Це забезпечує гнучкість і можливість вибору найбільш підходящої мови для розробки проекту;

2) SQL є стандартною мовою для роботи з реляційними базами даних і володіє потужними можливостями з управління даними. Використання SQL дозволяє ефективно працювати з базами даних і забезпечувати цілісність і безпеку даних;

3) При виборі технологій необхідно враховувати вимоги проекту, досвід і переваги команди розробників, а також доступність ресурсів та інфраструктури для розробки;

4) Ґрунтуючись на проведеному аналізі і з урахуванням вимог проекту, можна зробити обґрунтований вибір технологій.NET Framework і SQL, методів роботи з даними, такими як серіалізація і десеріалізація, для розробки автоматизованої системи управління проектами для малих підприємств.

В цілому, проведення аналізу відомих програмних рішень і вибору технологій дозволяє визначити найбільш підходящі варіанти для розробки системи управління проєктами, враховуючи вимоги проєкту, переваги команди і доступні ресурси.

					ІК92.280БАК.005 ПЗ	Арк.
						36
Зм.	Лист	№ докум.	Підпис	Дата		

5 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Структура системи

В даному розділі буде описана загальна структура системи управління проектами для малих підприємств. Будуть описані основні компоненти системи, їх взаємодія і функціональні можливості. Також буде представлена архітектура системи і вибрані патерни проектування [15].

Для більш докладного опису класів і створення зразкової діаграми класів (див.кресленник K92.280БАК.005 Д4), використаємо наступну структуру:

1) клас ProjectManager:

- основний клас системи, відповідальний за управління проектами та завданнями;
- містить список всіх проектів і методи для їх додавання, видалення і доступу до них;
- забезпечує функціональність для управління завданнями всередині проектів, включаючи створення, видалення, зміна параметрів і зв'язків між завданнями;
- методи для збереження і завантаження даних проектів і завдань.

2) клас Project:

- представляє окремий проект;
- містить інформацію про проект, таку як назва, опис та дата початку;
- включає в себе список завдань, що належать проекту, і методи для додавання, видалення і доступу до завдань.

3) абстрактний клас Task:

- визначає загальні атрибути і методи для всіх типів завдань;
- містить властивості, такі як ім'я завдання, тривалість, дати початку та закінчення, відсоток виконання та інші атрибути, які притаманні всім завданням;

- методи для зміни параметрів завдання, таких як установка дат початку і закінчення, установка відсотка виконання та інші операції із завданням.

4) клас GroupTask (успадковується від класу Task):

- представляє групу завдань, яка є контейнером для інших завдань;
- містить список вкладених завдань і методи для управління ними, включаючи додавання, видалення і доступ до завдань всередині групи;
- дозволяє встановлювати властивості групи завдань, такі як згорнутість/розгорнутість для зручного відображення і управління завданнями.

5) клас SplitTask. (успадковується від класу Task):

- Представляє розділену задачу, яка складається з декількох частин;
- Містить список частин завдання і методи для управління ними, включаючи додавання, видалення і доступ до частин;
- Дозволяє розділяти великі завдання на більш дрібні фрагменти для більш точного управління тимчасовими параметрами.

Встановлені між класами зв'язки більш докладно можна побачити на кресленнику ІК92.280БАК.005 Д5.

Збереження та завантаження даних реалізовано за допомогою серіалізації об'єктів у файли [16]. Наприклад, для збереження даних проєктів і завдань в файл можна використовувати JSON або XML формати. При збереженні проєкту, всі його атрибути і завдання будуть серіалізовані в файл. При завантаженні даних з файлу відбувається десеріалізація даних і відтворення об'єктів проєктів і завдань зі збереженими значеннями атрибутів.

Методи 'SaveData ()` та ` LoadData ()` у класі ` ProjectManager` можуть використовувати відповідні класи для серіалізації та десеріалізації даних. Наприклад, при використанні JSON-серіалізації, метод 'SaveData ()` буде серіалізувати список проєктів в JSON-рядок і зберігати її в файл. Метод 'LoadData ()` завантажить рядок JSON з файлу та десеріалізує його до списку проєктів та їх завдань. В системі також передбачений цілий список дій із завданнями:

					ІК92.280БАК.005 ПЗ	Арк.
						38
Зм.	Лист	№ докум.	Підпис	Дата		

1) лівий клік: при натисканні лівою кнопкою миші на завдання, вона вибирається, і її властивості відображаються в PropertyGrid для редагування;

2) перетягування лівою кнопкою миші: перетягування завдання з використанням лівої кнопки миші дозволяє змінити її початкову точку (Start). Ви можете переміщати завдання вздовж часової шкали проєкту;

3) перетягування правою кнопкою миші: перетягування завдання з використанням правої кнопки миші дозволяє змінити тривалість завдання (Duration). Переміщаючи праву межу завдання, ви можете змінити її тривалість;

4) перетягування середньою кнопкою миші: перетягування завдання з використанням середньої кнопки миші дозволяє змінити відсоток завершення завдання (Complete). Переміщаючи середню точку завдання вліво або вправо, ви можете змінити відсоток виконання завдання;

5) подвійне клацання лівою кнопкою миші: при подвійному клацанні лівою кнопкою миші на завданні відбувається перемикання згортання/розгортання групи завдань. якщо завдання є групою, воно може бути згорнуто (приховано) або розгорнуто (показано) для відображення вкладених завдань;

6) подвійне клацання правою кнопкою миші: при подвійному клацанні правою кнопкою миші на завданні, вона може бути розділена на частини. це корисно, коли завдання має різні етапи або підзадачі, які потребують окремого управління;

7) перетягування лівою кнопкою миші на інше завдання: перетягування завдання за допомогою лівої кнопки миші на інше завдання дозволяє групувати перетягуване завдання під цільовим завданням. таким чином, ви можете створити ієрархічну структуру завдань;

8) перетягування правою кнопкою миші на іншу частину завдання: перетягування частини завдання за допомогою правої кнопки миші дозволяє об'єднати її з іншими частинами завдання. це корисно, коли завдання має розділені фрагменти, які потрібно об'єднати в одне завдання;

9) SHIFT + перетягування лівою кнопкою миші на іншу задачу: при затиснутою клавіші SHIFT і перетягуванні завдання лівою кнопкою миші на іншу

					ІК92.280БАК.005 ПЗ	Арк.
						39
Зм.	Лист	№ докум.	Підпис	Дата		

задачу, можна зробити цільову задачу попередньої перетягувати задачі. Тобто, порядок виконання завдань буде змінено;

10) ALT + перетягування лівою кнопкою миші: при затиснутій клавіші ALT і перетягуванні завдання лівою кнопкою миші, можна розгрупувати перетягується завдання від цільової задачі або видалити цільову задачу зі списку попередніх завдань перетягується завдання. Це корисно, коли необхідно змінити зв'язки між завданнями;

11) SHIFT + перетягування лівою кнопкою миші: при затиснутою клавіші SHIFT і перетягуванні завдання лівою кнопкою миші, можна впорядкувати завдання. Це дозволяє легко перебудовувати порядок завдань, щоб оптимізувати розклад проєкту;

12) SHIFT + клік середньою кнопкою миші: при затиснутою клавіші SHIFT і кліці середньою кнопкою миші створюється нове завдання. Це зручний спосіб швидкого додавання нових завдань без необхідності вибору відповідного пункту меню;

13) ALT + клік середньою кнопкою миші: при затиснутою клавіші ALT і кліці середньою кнопкою миші можна видалити завдання. Це дозволяє швидко видаляти непотрібні завдання зі списку;

14) подвійне клацання лівою кнопкою миші: при подвійному клацанні лівою кнопкою миші на завданні відбувається перемикання згортання/розгортання групи завдань. якщо завдання є групою, воно може бути згорнуто (приховано) або розгорнуто (показано) для відображення вкладених завдань.

5.2 Модель зберігання даних

У цьому підрозділі буде представлена модель бази даних, розроблена для системи управління проєктами. Будуть описані таблиці, зв'язки і поля, необхідні для зберігання і управління даними проєктів, завдань, користувачів та інших сутностей.

					ІК92.280БАК.005 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

Надалі, детальніше розглянемо процес збереження та завантаження даних за допомогою серіалізації та десеріалізації. Ми будемо використовувати JSON формат.

Опис процесу збереження даних:

1) клас «ProjectManager» містить метод збереження даних «SaveData». Цей метод реалізує процес серіалізації об'єкт. Програмний код методу представлено на рисунку 5.1;

```
public void SaveData(string filePath) {...}
{
    // Створимо список об'єктів TaskData для серіалізації задач
    project.TaskDataList = new List<TaskData>();

    foreach (var task in project.Tasks)
    {
        // Створимо об'єкт TaskData и заповнимо його атрибути із
        // задачі
        var taskData = new TaskData();
        taskData.Name = task.Name;
        taskData.Duration = task.Duration;
        taskData.StartDate = task.StartDate;
        taskData.EndDate = task.EndDate;
        taskData.Complete = task.Complete;

        // Додаємо TaskData в список
        project.TaskDataList.Add(taskData);
    }

    // Серіалізуємо об'єкт data в JSON-строку
    string jsonData = JsonConvert.SerializeObject(data);

    // Зберігаємо JSON-строку в файл
    File.WriteAllText(filePath, jsonData);
}
```

Рисунок 5.1 — Програмний код методу SaveData.

2) допоміжний клас «ProjectManagerData» зберігає дані для серіалізації у вигляду списку List <T>;

3) допоміжний клас «TaskData», який зберігатиме дані про завдання для серіалізації;

```
public class TaskData
{
    Ссылка: 0
    public string Name { get; set; }
    Ссылка: 0
    public int Duration { get; set; }
    Ссылка: 0
    public DateTime StartDate { get; set; }
    Ссылка: 0
    public DateTime EndDate { get; set; }
    Ссылка: 0
    public float Complete { get; set; }
}
```

Рисунок 5.2 — Програмний код методу TaskData.

Завантаження даних:

- 1) у класі «ProjectManager» створено метод «LoadData()». Він відповідає за десеріалізацію даних. Програмний код методу представлено на рисунку 5.3;

```
public void LoadData(string filePath)
{
    // Читаємо JSON-строку з файлу
    string jsonData = File.ReadAllText(filePath);

    // Десеріалізуємо JSON-строку в об'єкт ProjectManagerData
    var data =
        JsonConvert.DeserializeObject<ProjectManagerData>(jsonData);

    // Очищаємо поточний список проектів
    projects.Clear();

    // Проходимо по кожному проекту в data та створимо нові об'єкти
    // Project
    foreach (var projectData in data.Projects)
    {
        // Створимо новий об'єкт Project
        var project = new Project(projectData.Name,
            projectData.Description, projectData.StartDate);

        // Проходимо по списку TaskData та створимо нові об'єкти Task
        foreach (var taskData in projectData.TaskDataList)
        {
            // Створимо новий об'єкт Task і заповнимо його атрибути
            // з TaskData
            var task =
                new Task(taskData.Name, taskData.Duration, taskData.StartDate,
                    taskData.EndDate, taskData.Complete);

            // Додаємо задачу в проект
            project.AddTask(task);
        }

        // Додаємо проект в список проектів
        projects.Add(project);
    }
}
```

Рисунок 5.3 — Програмний код методу LoadData.

- 2) приклад JSON-об'єкту, який може бути збережено в файл, представлено на рисунку 5.4;

```
{
  "Projects": [
    {
      "Name": "Project 1",
      "Description": "Description of Project 1",
      "StartDate": "2023-01-01T00:00:00",
      "TaskDataList": [
        {
          "Name": "Task 1",
          "Duration": 5,
          "StartDate": "2023-01-01T00:00:00",
          "EndDate": "2023-01-05T00:00:00",
          "Complete": 0.8
        },
        {
          "Name": "Task 2",
          "Duration": 3,
          "StartDate": "2023-01-06T00:00:00",
          "EndDate": "2023-01-08T00:00:00",
          "Complete": 0.5
        }
      ]
    }
  ]
}
```

Рисунок 5.4 — Приклад JSON об'єкту.

Це лише приблизний формат JSON-об'єкта для збереження і завантаження даних. Фактичний формат може відрізнятися залежно від вимог та специфікації проекту.

Опис таблиць бази даних та зв'язків між ними описано в Таблиці 5.1 – Таблиці 5.5.

Таблиця 5.1 — Project.

Назва поля	Тип Поля	Опис Поля	Індекс
Id	Integer	Унікальний ідентифікатор	РК
ManagerId	Integer	Ідентифікатор менеджера	FK
Name	Varchar(255)	Назва проєкту	
Description	Text	Опис	
StartDate	DateTime	Дата початку	

Таблиця 5.2 — ProjectManager.

Назва поля	Тип Поля	Опис Поля	Індекс
Id	Integer	Унікальний ідентифікатор	РК

Таблиця 5.3 — Task.

Назва поля	Тип Поля	Опис Поля	Індекс
Id	Integer	Унікальний ідентифікатор	РК
ProjectId	Integer	Ідентифікатор проєкту	FK
Name	Varchar(255)	Назва задачі	
Duration	Integer	Тривалість задачі	
StartDate	DateTime	Дата початку	
EndDate	DateTime	Кінцева дата	
Complete	Float	Показник завершеності	

Таблиця 5.4 — GroupTask.

Назва поля	Тип Поля	Опис Поля	Індекс
Id	Integer	Унікальний ідентифікатор	РК
TaskId	Integer	Ідентифікатор задачі	FK

Таблиця 5.5 — SplitTask.

					ІК92.280БАК.005 ПЗ	Арк.
						43
Зм.	Лист	№ докум.	Підпис	Дата		

Назва поля	Тип Поля	Опис Поля	Індекс
Id	Integer	Унікальний ідентифікатор	PK
TaskId	Integer	Ідентицікатор задачі	FK

Структура бази даних представлена графічно на кресленнику ІК92.280БАК.005 Д6.

Висновки за розділом

Розділ "Розробка програмного забезпечення" являє собою важливу частину опису процесу розробки програмного забезпечення. У цьому розділі ми розглянули наступні ключові аспекти:

- Управління завданнями та ресурсами: у розробці програмного забезпечення ефективне управління завданнями та ресурсами відіграє вирішальну роль. Було реалізовано клас ProjectManager, який надає функціональність для управління завданнями та ресурсами в проєкті;
- визначення та властивості завдань: Завдання є основними будівельними блоками проєкту. Розглянули клас Task, який представляє завдання та має такі властивості, як початок, кінець, тривалість, завершення та відступ;
- розділення та об'єднання завдань: для більш гнучкого управління завданнями, ProjectManager надає методи для поділу завдання на кілька частин або об'єднання частин в одне завдання. Це дозволяє більш детально контролювати розподіл завдань і управляти тривалістю і завершеністю кожної частини;
- збереження і завантаження даних: для збереження і завантаження даних в проєкті, ми обговорили процес серіалізації і десеріалізації. Ви можете використовувати JSON або інші формати, щоб зберегти стан проєкту та відновити його, якщо це необхідно.

6 ТЕСТУВАННЯ СИСТЕМИ

6.1 Тестування елемента керування Chart

Елемент управління Chart являє собою важливий компонент в розробці програмного забезпечення, який відображає графік Гантта. Щоб переконатися в його коректній роботі, необхідно провести тестування функціональності і взаємодії з іншими елементами програми. У цьому розділі ми розглянемо основні аспекти тестування елемента керування Chart.

6.1.1 Модульне тестування

Модульне тестування дозволяє перевірити окремі модулі або компоненти елемента керування Chart на коректність і відповідність очікуваній поведінці. Прикладом таких тестів є додавання елемента керування на форму, ініціалізація, відображення та перемальовування.

У модульному тестуванні елемента керування Chart ми можемо використовувати модульний тестовий фреймворк, такий як Microsoft Visual Studio Test Framework або NUnit. Ми створюємо окремий клас 'ChartTest', в якому визначаємо тестові методи для кожного аспекту функціональності, яку ми хочемо перевірити.

Приклади модульних тестів можуть включати:

- перевірка додавання елемента управління chart на форму і його ініціалізації з використанням екземпляра класу 'projectmanager';
- тестування відображення елемента управління chart після його ініціалізації і при зміні даних;
- перевірка можливості перемальовування елемента управління chart без попередньої ініціалізації.

Модульне тестування дозволяє виявити потенційні проблеми і помилки всередині елемента управління Chart і забезпечити його правильне функціонування в ізолюваному середовищі.

					ІК92.280БАК.005 ПЗ	Арк.
						45
Зм.	Лист	№ докум.	Підпис	Дата		

6.1.2 Інтеграційне тестування

Інтеграційне тестування має на меті перевірити взаємодію елемента керування Chart з іншими компонентами програми. Це включає перевірку коректності передачі даних, взаємодію з базою даних, обробку подій та реакцію на введення користувачем.

Приклади інтеграційних тестів можуть включати:

- перевірка взаємодії елемента управління chart з базою даних, наприклад, збереження і завантаження даних проєкту;
- тестування обробки подій, таких як вибір завдання або зміна параметрів завдання;
- перевірка реакції елемента керування chart на введення користувачем, наприклад, зміна тривалості завдання за допомогою перетягування миші.

Інтеграційне тестування дозволяє переконатися в тому, що елемент управління Chart коректно взаємодіє з іншими компонентами програми і виконує свої функції відповідно до вимог.

6.1.3 Візуальне тестування

Візуальне тестування елемента управління Chart здійснюється шляхом перевірки його зовнішнього вигляду і відповідності заданим дизайнерським і стильовим вимогам. Це також включає перевірку зовнішнього вигляду графіків, міток осей, колірної схеми та інших візуальних атрибутів.

Приклади візуальних тестів можуть включати:

- перевірка правильного відображення графіків завдань з відповідними мітками і тимчасовими шкалами;
- перевірка відповідності колірної схеми і стильових налаштувань вимогам ui/ux дизайну.

Візуальне тестування допомагає переконатися в тому, що елемент управління Chart виглядає і функціонує так, як очікується, і відповідає вимогам до користувальницького інтерфейсу.

6.1.4 Сценарне тестування

Сценарне тестування елемента управління Chart призначене для перевірки його працездатності в реальних сценаріях використання. Це включає перевірку різних комбінацій функцій та операцій, а також їх послідовності.

Приклади сценарних тестів можуть включати:

- створення проєкту, додавання завдань, встановлення залежностей між завданнями та перегляд графіка ганта;
- зміна параметрів завдань, таких як тривалість і відсоток завершеності, і перевірка оновлення графіка;
- взаємодія з елементами управління, такими як кнопки і контекстні меню, і перевірка відповідних дій.

Сценарне тестування дозволяє переконатися в тому, що елемент управління Chart успішно виконує завдання і операції, необхідні для роботи з графіком Ганта в реальних ситуаціях.

6.1.5 Регресійне тестування

Регресійне тестування елемента керування Chart має на меті перевірити його стабільність та зберегти функціональність після внесення змін або виправлення помилок. Після кожної зміни в коді елемента керування необхідно запустити набір тестів, включаючи раніше перевірені сценарії, для виявлення потенційних регресійних помилок.

Приклади регресійних тестів можуть включати:

- Повторне виконання модульних, інтеграційних, візуальних та сценарних тестів після внесення змін до коду елемента керування;

					ІК92.280БАК.005 ПЗ	Арк.
						47
Зм.	Лист	№ докум.	Підпис	Дата		

– Перевірка коректності роботи елемента управління після виправлення помилок або додавання нових функцій;

Регресійне тестування допомагає виявити і усунути помилки, які можуть виникнути після внесення змін в елемент управління Chart і забезпечити його надійну роботу.

6.2 Тестування ProjectManager

ProjectManager - це компонент, який керує проєктом у контексті діаграми Ганта. Він надає методи та функціональність для створення, видалення, переміщення, групування та управління завданнями та групами завдань всередині проєкту. Проєктний менеджер виконує наступні основні завдання:

1) управління завданнями: ProjectManager надає методи створення та видалення завдань. це також дозволяє переміщувати завдання всередині проєкту, змінювати їх порядок та ієрархічну структуру. це дозволяє організувати завдання у вигляді дерева і визначити їх взаємозв'язку і залежності;

2) групування завдань: компонент дозволяє групувати завдання всередині інших завдань або груп завдань. це дозволяє створювати ієрархічну структуру проєкту, де завдання можуть бути організовані у вигляді вкладених груп. групування допомагає впорядкувати та структурувати завдання, що полегшує аналіз проєкту та його візуалізацію;

3) управління ресурсами: projectmanager надає можливість управління ресурсами, пов'язаними із завданнями. ресурси можуть бути призначені завданням і використані для відстеження та управління ресурсами, необхідними для виконання проєкту. менеджер проєкту надає методи додавання та видалення ресурсів, а також зв'язування ресурсів із завданнями;

4) надання інформації про проєкт: клас projectmanager дозволяє отримати інформацію про завдання та групи завдань у проєкті. це включає отримання списку дочірніх, попередніх, наступників та залежних завдань для конкретного завдання, а також отримання списку ресурсів, пов'язаних із завданням. ця інформація може

бути використана для аналізу та відображення проєкту, створення звітів та планування.

Використання ProjectManager спрощує управління проєктом в контексті діаграми Ганта, дозволяє організувати завдання і ресурси проєкту, а також забезпечує функціональність для аналізу і візуалізації проєкту.

Для перевірки функціональності класу ProjectManager, відповідального за управління завданнями і групами в діаграмі Ганта, був розроблений набір модульних тестів. Нижче наведено опис деяких з цих тестів:

1) тест "CreateAndRemoveTask":

- створення завдання і додавання її в менеджер проєкту;
- перевірка, що завдання успішно додана і кількість завдань дорівнює 1;
- перевірка, що завдання не має батьківського завдання;
- створення другого завдання, видалення першого завдання і перевірка, що друге завдання займає її місце.

2) тест "CreateAndRemoveGroup":

- створення групи і додавання її в менеджер проєкту;
- створення трьох завдань і додавання їх в менеджер проєкту;
- групування завдань всередині групи;
- перевірка, що кількість завдань в менеджері проєкту дорівнює 3, кількість дочірніх завдань у групи дорівнює 2 і у кожної дочірньої задачі встановлений батько – група;
- видалення групи і перевірка, що всі дочірні завдання звільняються від батьківського зв'язку.

3) тест "ProjectEmptyEnumerators":

- перевірка, що при ініціалізації менеджера проєкту списки завдань і ресурсів порожні.

4) тест "EnumerationShouldReturnEmptySet":

					ІК92.280БАК.005 ПЗ	Арк.
						49
Зм.	Лист	№ докум.	Підпис	Дата		

- перевірка, що при запиті дочірніх, попередніх, наступників, нащадків, залежних і ресурсів завдання, якої немає в менеджері проєкту, повертаються порожні безлічі.

5) тест "KnownTasksEnumeration":

- створення завдання і додавання її в менеджер проєкту;
- перевірка, що при запиті дочірніх, попередніх, наступників, нащадків, залежних і ресурсів цього завдання повертаються відповідні списки, які не є порожніми.

6) тести на переміщення завдань і груп всередині діаграми:

- MoveSingleTaskToCheckForOutOfBoundhandling: перевірка переміщення завдання при вказівці неприпустимого зміщення;
- MoveNonExistingTask: перевірка переміщення неіснуючої задачі;
- MoveTasksAroundSingleLevel: перевірка переміщення завдань в одному рівні і перевірка їх нового порядку;
- MoveGroupsAround: перевірка переміщення груп завдань і перевірка їх нової структури.

7) тести на розгрупування груп:

- UngroupUnknownTasksFromGroup: перевірка неможливості розгрупування невідомих завдань з групи;
- UngroupTaskFromUnknownGroups: перевірка неможливості розгрупування завдання з невідомої групи;
- UngroupUnknownGroup: перевірка неможливості розгрупування невідомої групи;
- UngroupAGroup: перевірка розгрупування групи та переміщення дочірніх завдань до батьківської групи.

Виконуючи ці тести, можна переконатися в правильності роботи класу ProjectManager і його відповідності очікуваної функціональності.

					ІК92.280БАК.005 ПЗ	Арк.
						50
Зм.	Лист	№ докум.	Підпис	Дата		

6.3 Тестування класу 'ProjectManager'

В даному розділі проводиться тестування функціональності 'ProjectManager' з використанням декількох тестових методів. Кожен метод перевіряє певний аспект роботи 'ProjectManager' і взаємодія з класом 'Task'.

Тестові методи:

1) 'SetTaskParameters': перевіряє встановлення параметрів завдання ('Task') у 'ProjectManager'. Створюється екземпляр 'ProjectManager' і завдання 'one'. Завдання додається в 'ProjectManager', а потім встановлюються і перевіряються значення різних параметрів завдання;

2) 'RelateBeforePrecedentEndsLate': перевіряє правильність установки зв'язку між двома завданнями і розрахунок значення властивості 'Slack'. Створюється екземпляр 'ProjectManager' і завдання 'one' і 'two'. Завдання

додаються в 'ProjectManager' і встановлюються їх параметри. Потім встановлюється залежність між завданнями і перевіряються значення властивостей завдань;

3) 'AdjustDependantScheduleOnRelate': перевіряє коректність зміни розкладу залежного завдання при зміні розкладу попереднього завдання. Створюється екземпляр 'ProjectManager' і завдання 'one' і 'two'. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім встановлюється залежність між завданнями, змінюється розклад завдання 'one' і перевіряються значення властивостей завдань;

4) 'PrecedentPushingDependantLater': перевіряє зміну розкладу залежного завдання при зміні розкладу попереднього завдання так, щоб залежне завдання починалося пізніше. Створюється екземпляр 'ProjectManager' і завдання 'one' і 'two'. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім встановлюється залежність між завданнями, змінюється розклад завдання 'one' і перевіряються значення властивостей завдань;

5) 'AdjustGroupScheduleOnMemberSchedulechange': перевіряє оновлення розкладу групи при зміні розкладу одного з її членів. Створюється екземпляр

					ІК92.280БАК.005 ПЗ	Арк.
						51
Зм.	Лист	№ докум.	Підпис	Дата		

'ProjectManager` і завдання` group`,` one` і`two'. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім створюється група, в яку входять завдання`one` і`two`. Змінюється розклад завдання`one` і перевіряються значення властивостей групи;

6) 'CannotSetCompleteOnGroup': перевіряє неможливість встановлення значення` Complete' для групи завдань. Створюється екземпляр 'ProjectManager` і завдання` group` і`member'. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім створюється група, в яку входять завдання`group` і`member`. Встановлюється значення`Complete' для групи і перевіряються значення властивостей завдань;

7) 'AdjustGroupCompleteOnMemberCompletechange': перевіряє оновлення значення` Complete` групи при зміні значення` Complete' одного з її членів. Створюється екземпляр 'ProjectManager` і завдання` group` і`member'. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім створюється група, в яку входять завдання`group` і`member`. Встановлюється значення`Complete` для завдання`member` і перевіряються значення властивостей завдань;

8) 'AdjustGroupScheduleOnGroup': перевіряє встановлення розкладу групи, узгодженого з розкладом одного з її членів. Створюється екземпляр 'ProjectManager` і завдання` group` і`one'. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім створюється група, в яку входять завдання`group` і`one`, і перевіряються значення властивостей групи;

9) 'DurationCannotLessThanOne': перевіряє обмеження на мінімальне значення` Duration' завдання. Створюється екземпляр 'ProjectManager` і завдання`one'. Завдання додається в 'ProjectManager' і встановлюються її параметри. Потім змінюються значення` Duration` завдання так, щоб воно було менше 1, і перевіряється, що` Duration` залишається рівним 1;

10) 'CompleteMustBetweenZeroAndOne': перевіряє обмеження на значення` Complete' завдання. Створюється екземпляр 'ProjectManager` і завдання`one'. Завдання додається в 'ProjectManager' і встановлюються її параметри. Потім

					ІК92.280БАК.005 ПЗ	Арк.
						52
Зм.	Лист	№ докум.	Підпис	Дата		

змінюються значення `Complete` завдання так, щоб воно було більше 1 або менше 0, і перевіряється, що `Complete` залишається обмеженим значеннями від 0 до 1;

11) 'DependantRescheduledStartEarlierthanprecedentend': перевіряє зміну розкладу залежного завдання, якщо новий розклад призводить до того, що воно починається до завершення попереднього завдання. Створюється екземпляр 'ProjectManager` і завдання` one` і`two'. Завдання і додаються в 'ProjectManager' і встановлюються їх параметри. Потім встановлюється залежність між завданнями, змінюється розклад завдання `two` і перевіряються значення властивостей завдань;

12) 'CriticalPath': перевіряє визначення критичного шляху в проєкті. Створюється екземпляр 'ProjectManager' і кілька завдань, включаючи групи завдань. Завдання додаються в 'ProjectManager' і встановлюються їх параметри. Потім встановлюється залежність між завданнями і перевіряється, що критичний шлях визначено коректно;

13) 'TaskStartLessThanEnd': перевіряє обмеження на значення `Start` та `End` завдання, гарантуючи, що `Start` повинен бути меншим за `End`. Створюється екземпляр 'ProjectManager` і завдання`task'. Завдання додається в 'ProjectManager' і встановлюються її параметри. Потім змінюються значення `Start` і `End` завдання так, щоб це обмеження порушувалося, і перевіряється, що `Start` стає менше `End`;

14) 'Setsplitschedule': перевіряє поділ завдання на кілька частин з установкою коректного розкладу для кожної частини. Створюється екземпляр 'ProjectManager` і завдання` split`,` part1`,` part2` і`part3'. Завдання 'split` додається в `ProjectManager' і встановлюються її параметри. Потім 'split` розділяється на частини `part1` і` part2`, а `part2` розділяється на `part2` і`part3'. Перевіряються значення властивостей кожного завдання.

6.4 Використання елемента керування Gantt Chart

У класі Example Full представлено використання елемента управління Gantt Chart для відображення і управління завданнями і ресурсами в проєкті. Елемент управління Gantt Chart призначений для візуалізації тимчасових даних у вигляді графіка Ганта, який дозволяє відстежувати прогрес виконання завдань і управляти

					ІК92.280БАК.005 ПЗ	Арк.
						53
Зм.	Лист	№ докум.	Підпис	Дата		

залежностями між ними. По суті невеликий додаток для управління завданнями користувачів. Його зовнішній вигляд представлений на рисунку 6.1.

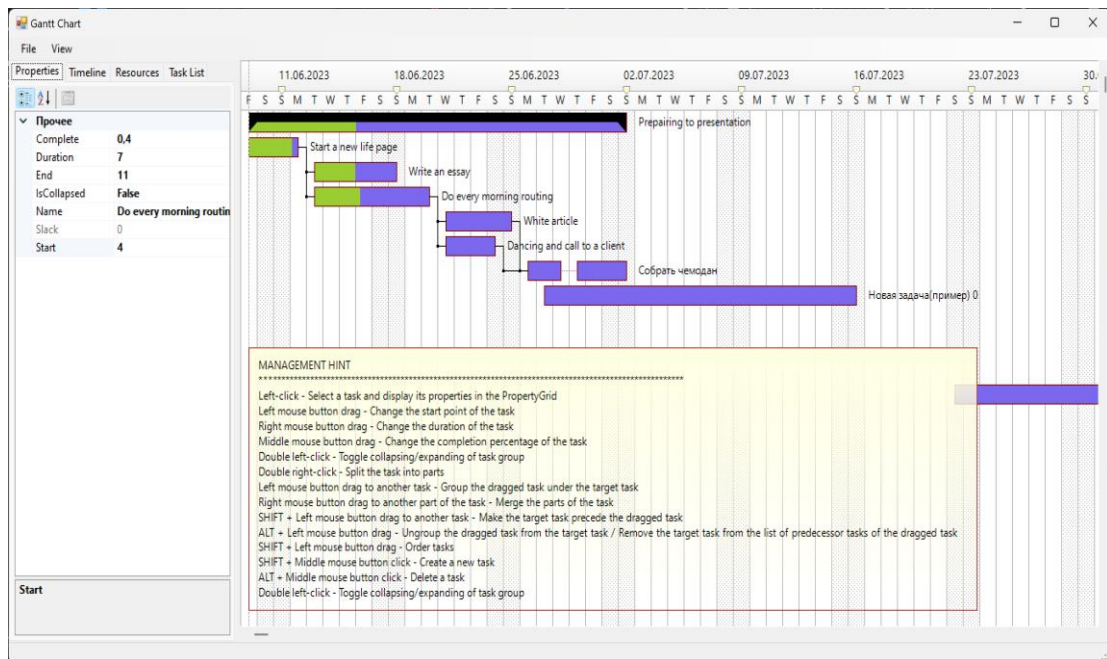


Рисунок 6.1 — Зовнішній вигляд додатку.

Загальний вигляд частини завдань на діаграмі Ганта представлений нижче на рисунку 6.2. Зеленим відображається стан виконання задачі.

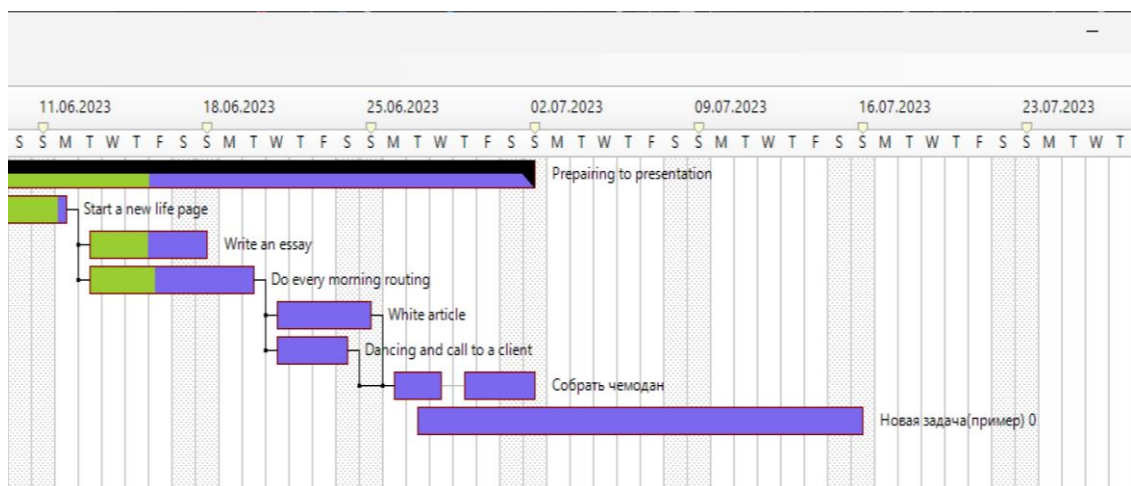


Рисунок 6.2 — Зовнішній вигляд задач проєкту.

Для користувача доступна графічна підказка про доступні йому дії в системі (рисунок 6.3).

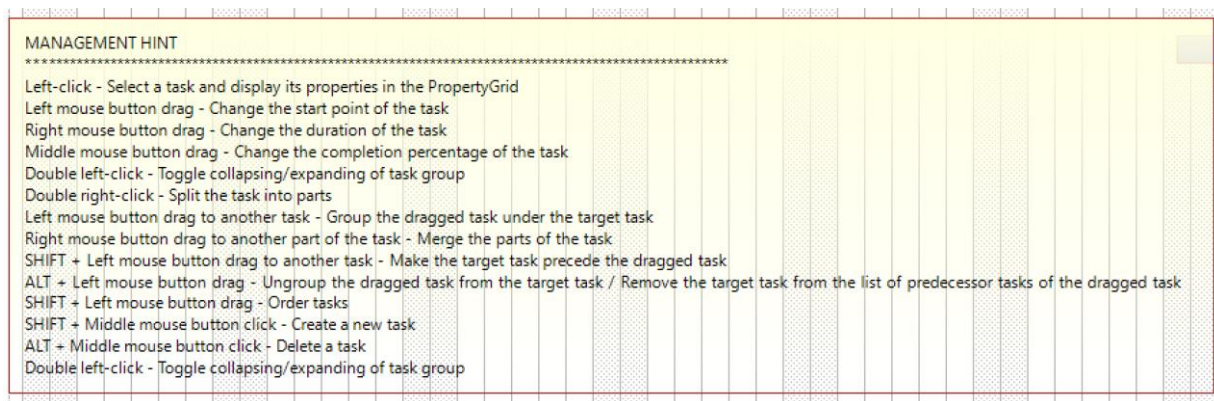


Рисунок 6.3 — Графічна підказка для користувачів.

Для кожного завдання користувач може редагувати необхідні властивості: назву, тривалість, власника, дату кінця/початку тощо. Властивості задачі відображено на рисунку 6.4.

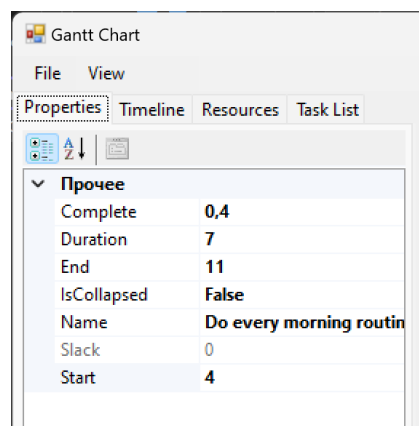


Рисунок 6.4 — Властивості задачі.

Ключові особливості і функціональність в даному прикладі:

1) створення проєкту та завдань:

- створення екземпляра класу 'ProjectManager', який управляє проєктом і завданнями;
- створення і додавання різних завдань ('MyTask') в проєкт.

2) визначення властивостей задач:

- установка тривалості, початкової точки і відсотка завершеності для кожного завдання.

3) групування завдань:

- групування завдань у різні категорії або ієрархічні структури.

4) встановлення залежностей між завданнями:

- установка попередніх завдань для кожного завдання, що визначають логічну послідовність виконання.

5) призначення ресурсів завданням:

- призначення ресурсів ('myresource') завданням проєкту.

6) візуалізація даних і взаємодія з елементом управління:

- відображення графіка ганта із завданнями та залежностями між ними.
- можливість наведення миші на завдання для отримання інформації про тимчасові параметри.
- можливість вибору завдання для відображення властивостей в бічній панелі.
- підтримка перетягування завдань для зміни тимчасових параметрів.
- різні дії, такі як групування завдань, розділення завдання, упорядкування завдань та видалення завдань.

7) друк і збереження даних:

- можливість друку графіка ганта з завданнями і залежностями.
- збереження проєкту у файл та завантаження раніше збереженого проєкту.

Елемент управління Gantt Chart дозволяє користувачам наочно представляти тимчасові дані і управляти проєктами і завданнями в зручному і ефективному форматі. Він може використовуватися в різних додатках, пов'язаних з управлінням проєктами, плануванням ресурсів, плануванням завдань та іншими сферами, де необхідне візуальне представлення тимчасових даних та їх управління. Порівняння Gantt Chart із існуючими рішеннями представлено у таблиці 6.1.

					ІК92.280БАК.005 ПЗ	Арк.
						56
Зм.	Лист	№ докум.	Підпис	Дата		

Таблиця 6.1 — Порівняння з існуючими рішеннями.

	Нативний інтерфейс	Наявність візуал. аналітики	Непотрібне дод.навчання	Відстеж. % вик-ня проєкту	Пасує будь- якій команді	Повністю безкоштовна
Kendo Manager						
Intasker						
Trello						
Asana						
MS Project						
Ganntt Chart						

Відповідно до таблиці, була побудована пелюсткова діаграма (див. рисунок 6.5) для більш наглядного представлення результатів.

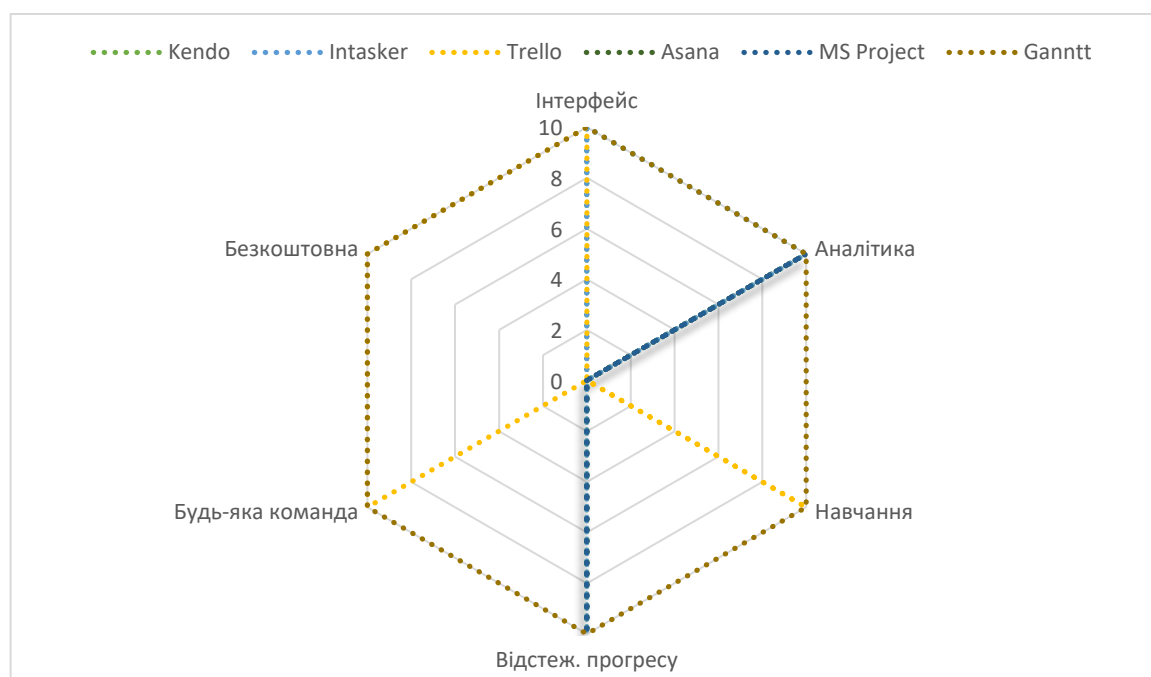


Рисунок 6.5 — Пелюсткова діаграма результатів.

Порівняння обсягу займаної пам'яті представлено в Таблиці 6.2.

Таблиці 6.2 — Порівняння розміру програм.

	Kendo Manager	Intasker	Trello	Asana	MS Project	Ganntt Chart
Обсяг займаємої пам'яті	Гілки веб- версія.	31,5МБ	71,3МБ	48,3 МБ	3,2ГБ	До 30МБ.

Висновки за розділом

У цьому розділі було представлено приклад використання елемента керування Gantt Chart для візуалізації та управління проєктами та завданнями. Цей елемент управління володіє широким набором функціональності, що дозволяє ефективно відстежувати прогрес виконання завдань і управляти залежностями між ними. Приклад демонструє створення проєкту та завдань, встановлення часових параметрів для завдань, групування завдань у категорії, визначення залежностей між завданнями та призначення ресурсів завданням. Це дозволяє структурувати проєкт і управляти його виконанням.

Елемент управління Ganntt Chart надає можливість візуального відображення графіка Ганта, на якому завдання представлені у вигляді смуг, а залежності між завданнями відображаються стрілками. Це забезпечує наочність і зрозумілість при аналізі та плануванні проєкту.

Інтерфейс користувача елемента керування дозволяє взаємодіяти із завданнями, наприклад, вибирати завдання для перегляду та редагування властивостей, перетягувати завдання для зміни параметрів часу та виконувати інші дії, такі як групування, розділення та видалення завдань.

Елемент управління Gantt Chart також підтримує функції друку і збереження проєкту, що забезпечує можливість зберегти дані і поділитися ними з іншими користувачами.

Приклад показує, як елемент керування Gantt Chart можна використовувати в різних додатках, пов'язаних з управлінням проєктами, плануванням ресурсів та

					ІК92.280БАК.005 ПЗ	Арк.
						58
Зм.	Лист	№ докум.	Підпис	Дата		

завданнями. Він забезпечує зручний та ефективний спосіб візуалізації та управління тимчасовими даними.

Загалом, розділ 6 демонструє можливості управління Gantt Chart і дозволяє розробникам створювати функціональні та інтуїтивно зрозумілі програми для управління проектами та завданнями.

					ІК92.280БАК.005 ПЗ	Арк.
						59
Зм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВОК

Автоматизована система управління проектами для малого бізнесу є важливим інструментом для ефективного планування, контролю та управління проектами. В рамках даної роботи був проведений аналіз предметної області, вимог до програмного забезпечення, відомих програмних рішень, а також вибір технологій для розробки. На основі цього аналізу була розроблена концепція та реалізація автоматизованої системи управління проектами.

В процесі аналізу предметної області був проведений детальний огляд основного бізнес-процесу, складу завдань і підзадач, а також інформаційних потоків, що дозволило визначити основні потреби і вимоги користувачів. Були виявлені ключові проблеми, які підлягають вирішенню за допомогою автоматизованої системи управління проектами. Кожне рішення має свої переваги та недоліки, і вибір конкретного інструменту повинен базуватися на вимогах проекту, бюджеті, потребах користувачів та уподобаннях команди розробників.

В результаті аналізу існуючих методів та інструментів розробки були обрані технології. NET Framework для розробки операційної структури системи і SQL для розробки моделі бази даних. . NET Framework надає широкий набір інструментів і розширеної функціональності для програмування, що забезпечує гнучкість і можливість вибору найкраще підручне мови для роботи. В той час SQL є стандартною мовою для роботи з релевантними базами даних і відстежує можливі можливості управління даними. Також було використано методи серіалізації та десеріалізації даних для пришвидшення роботи з даними, та метою зменшити обсяг використаної пам'яті.

У процесі розробки програмного забезпечення була визначена загальна структура системи, розроблена модель бази даних і реалізовані функціональні модулі, такі як модуль управління проектами, модуль управління завданнями, модуль діаграми та інші. Кожен модуль був інтегрований в загальну структуру системи, що забезпечує її функціональність і цілісність.

Тестування системи проводилося на різних етапах, включаючи модульне тестування, інтеграційне тестування, системне тестування та тестування

					ІК92.280БАК.005 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис	Дата		

користувачів. Це дозволило перевірити коректність роботи функціональних модулів, забезпечити стабільність і безпеку системи, а також переконатися у відповідності системи вимогам і очікуванням користувачів. Виявлені помилки були виправлені, а пропозиції щодо поліпшення були враховані в процесі розробки.

В результаті розробки автоматизованої системи управління проєктами була досягнута мета роботи-створення ефективного інструменту для планування, контролю та управління проєктами на малих підприємствах. Система володіє широким набором функціональних можливостей, таких як створення і управління проєктами, призначення і відстеження завдань, генерація звітів та інші. Однак, слід зазначити, що система не є вичерпним рішенням для всіх можливих сценаріїв управління проєктами. У подальшому розвитку системи можна розглянути додавання додаткових функцій, таких як інтеграція з іншими інструментами і сервісами, підтримка багатоплатформенності і розширення функціональності відповідно до потреб користувачів.

Загалом, автоматизована система управління проєктами для малого бізнесу є важливим інструментом для підвищення ефективності та успішного виконання проєктів. Розробка та впровадження такої системи вимагає ретельного аналізу вимог, вибору відповідних інструментів та технологій, а також тестування та постійного вдосконалення. Створена система може стати надійним партнером і інструментом для малих підприємств, допомагаючи їм досягати успіху в управлінні своїми проєктами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Project Management Institute - A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 2022.

					ІК92.280БАК.005 ПЗ	Арк.
						61
Зм.	Лист	№ докум.	Підпис	Дата		

2) Joseph Phillips - Project Management for Small Business: A Streamlined Approach from Planning to Completion. 2012.

3) Mark C. Layton, Steven J. Ostermiller - Agile Project Management for Dummies. 2017.

4) Harold Kerzner - Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 2017.

5) William P. Athayde - Project Management Essentials: A Quick and Easy Guide to the Most Important Concepts and Best Practices for Managing Your Projects Right. 2009.

6) Robert K. Wysocki - Effective Project Management: Traditional, Agile, Extreme. 2009.

7) Greg Horine - Project Management Absolute Beginner's Guide. 2017.

8) Eric Verzuh - The Fast Forward MBA in Project Management. 2017.

9) Paul Roberts - Project Management: A Practical Guide to Planning and Executing Successful Projects. 2013.

10) Charles G. Cobb - The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach. 2015.

11) Sid Kemp - Project Management for Small Business: How to Deliver Successful Projects on Time and Within Budget. 2016.

12) Juana Clark Craig - Project Management Lite: Just Enough to Get the Job Done...Nothing More. 2012.

13) Scott Berkun - The Art of Project Management. 2005.

14) Cynthia Snyder Dionisio - A Project Manager's Book of Forms: A Companion to the PMBOK Guide. 2022.

15.) М. Фаулер. - Архітектура корпоративних програмних додатків. Вільямс. 2007.

					ІК92.280БАК.005 ПЗ	Арк.
						62
Зм.	Лист	№ докум.	Підпис	Дата		

16) М. Мак-Дональд, М. Шпушта. - Microsoft 2.0 з прикладами на С# 2005 для професіоналів. Вільямс. 2007.

17) Стекер, М.А. - Розробка клієнтських Windows-додатків на платформі Microsoft. NET Framework. 2008.

					ІК92.280БАК.005 ПЗ	Арк.
						63
Зм.	Лист	№ докум.	Підпис	Дата		