

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
*Кафедра автоматизованих систем обробки інформації та управління*

УДК 004.9

**«До захисту допущено»**  
**В.О. завідувача кафедри**  
\_\_\_\_\_ О.А.Павлов  
(підпис) (ініціали, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Дипломний проект**  
**на здобуття ступеня бакалавра**

з напрямку підготовки \_\_\_\_\_ 6.050101 «Комп'ютерні науки» \_\_\_\_\_

на тему: « Система підтримки процесу управління створенням медіа  
матеріалу » \_\_\_\_\_

**Виконав :**

Студент 4 курсу, групи ІС-52

Акімов Дмитро Дмитрович  
(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Керівник**

доц. Жураковська О.С.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

**Консультант з  
графічної  
документації**

ст.викл. Халус О.А.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

**Рецензент**

\_\_\_\_\_ (посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

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

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) \_\_\_\_\_ *інформатики та обчислювальної техніки* \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ *автоматизованих систем обробки інформації та управління* \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) \_\_\_\_\_ *6.050101* \_\_\_\_\_

«Комп'ютерні науки» («Інформаційні управляючі системи та технології»)

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ *О.А.Павлов* \_\_\_\_\_  
(підпис) (ініціали, прізвище)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

*Акімову Дмитру Дмитровичу*  
\_\_\_\_\_ (прізвище, ім'я, по батькові)

**1. Тема проекту** « *Система підтримки процесу управління створенням  
медіа матеріалу* » \_\_\_\_\_

керівник проекту \_\_\_\_\_ *Жураковська Оксана Сергіївна, к.т.н., доцент* \_\_\_\_\_

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

затверджені наказом по університету від “23” *квітня* 2019 р. №1181-с

**2. Термін подання студентом проекту** “03” *червня* 2019 року

**3. Вихідні дані до проекту**

*Технічне завдання*

**4. Зміст пояснювальної записки**

*1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі*

*2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних*

*3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання*

*4. Програмне та технічне забезпечення: засоби розробки, вимоги до*

*технічного забезпечення, архітектура програмного забезпечення, побудова звітів*

*5. Технологічний розділ: керівництво користувача, методика випробувань*

програмного продукту

## 5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема структурна діяльності

3. Схема бази даних

4. Схема структурна діаграми класів

## 6. Консультанти розділів проекту

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

7. Дата видачі завдання «15» лютого 2019 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	15.02.2019	
2.	Аналіз існуючих методів розв'язання задачі	22.02.2019	
3.	Постановка та формалізація задачі	01.03.2019	
4.	Розробка інформаційного забезпечення	05.03.2019	
5.	Алгоритмізація задачі	08.04.2019	
6.	Обґрунтування використовуваних технічних засобів	10.05.2019	
7.	Розробка програмного забезпечення	15.05.2019	
8.	Налагодження програми	18.05.2019	
9.	Виконання графічних документів	22.05.2019	
10.	Оформлення пояснювальної записки	26.05.2019	
11.	Подання ДП на попередній захист	28.05.2019	
12.	Подання ДП на основний захист	01.06.2019	
13.	Подання ДП рецензенту	05.06.2019	

Студент

\_\_\_\_\_ Д.Д. Акімов  
(підпис)

Керівник проекту

\_\_\_\_\_ О.С. Жураковська  
(підпис)

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
			<b><u>Документи текстові</u></b>		
1	A4	ДП ІС-5201.1181-с.ПЗ	Пояснювальна записка	60	
2	A4	ДП ІС-5201.1181-с.ТЗ	Технічне завдання	9	
3	A4	ДП ІС-5201.1181-с.ПМВ	Програма та методика випробувань	12	
			<b><u>Матеріали графічні</u></b>		
4	A3	ДП ІС-5201.1181-с.ССВ	Схема структурна варіантів використань	1	
5	A3	ДП ІС-5201.1181-с.ССД	Схема структурна діяльності	3	
6	A3	ДП ІС-5201.1181-с.СБД	Схема бази даних	1	
7	A3	ДП ІС-5201.1181-с.ССК	Схема структурна класів програмного забезпечення	1	

					<b>ДП ІС-5201.1181-с.ВДП</b>		
<b>Зм.</b>	<b>Арк.</b>	<b>ПІП</b>	<b>Підп.</b>	<b>Дата</b>	<b>Відомість дипломного проекту</b>		
<i>Розробн.</i>		Акімов Д.Д.					
<i>Керівн.</i>		Жураковська О.С.			<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>
<i>Консульт.</i>						1	1
<i>Н/контр.</i>		Халус О.А.			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52		
<i>Зав. каф.</i>		Павлов О.А.					

## **Пояснювальна записка до дипломного проекту**

на тему: «Система підтримки процесу управління створенням медіа  
матеріалу»

---

---

Київ – 2019 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з шести розділів, містить 24 рисунка, 12 таблиць, 1 додаток.

Метою розробки дипломного проекту є підвищення ефективності процесу управління створенням та організацією медіа проектів.

У розділі загальні положення було описано предметне середовище, що включало в себе опис процесу діяльності, і функціональної моделі, розглянуті існуючі аналоги створюваної системи, визначена точна постановка задачі, цілі та мета розробки.

У розділі з інформаційного забезпечення описані вхідні та вихідні дані, описана та проілюстрована структура бази даних.

У розділі з математичного забезпечення було описано змістовну і математичну постановку задачі, надано інформацію про математичну складову дипломного проекту, обґрунтовано метод вирішення, який був реалізований у застосунку.

У розділі з програмного забезпечення наведено вимоги до системи, архітектура програмного забезпечення, та описані засоби розробки.

У технологічному розділі наведена інструкція користувача системою, та показані результати випробувань програмного продукту.

### СТВОРЕННЯ МЕДІА МАТЕРІАЛУ, СИСТЕМА ПІДТРИМКИ

					<b>ДП ІС-5201.1181-с.ПЗ</b>			
		<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>	<i>Система підтримки процесу управління створенням медіа матеріалу</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>	<i>Акімов Д.Д.</i>							
<i>Перевірив.</i>	<i>Жураковська О.С.</i>						2	60
<i>Н. кон.</i>	<i>Халус О.А.</i>					<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>		
<i>Зав.</i>	<i>Павлов О.А.</i>							

## ABSTRACT

**Structure and scope of work.** The explanatory note of the diploma project consists of six sections, containing 23 figures, 12 tables, 1 application.

The purpose of developing the diploma project is to increase the effectiveness of the management process for the creation and organization of media projects.

In the general provisions section, a subject environment was described that included a description of the process of activity and a functional model, considered the existing analogs of the system being created, the exact definition of the problem, the goals and purposes of the development were determined.

The section on information provision describes the input and output data, describes and illustrates the structure of the database.

In the section on mathematical support, the content and mathematical formulation of the task was described, the information about the mathematical component of the diploma project was provided, the solution method that was implemented in the application was substantiated.

The software section explains the system requirements, software architecture, and development tools.

The technology section demonstrates how to use the system, and shows the results of software testing.

CREATING MEDIA MATERIAL, SUPPORT SYSTEM

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

## ЗМІСТ

ВСТУП .....	5
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>7</b>
<b>1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА .....</b>	<b>7</b>
<i>1.1.1 Опис процесу діяльності.....</i>	<i>9</i>
<i>1.1.2 Опис функціональної моделі.....</i>	<i>10</i>
<b>1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ .....</b>	<b>11</b>
<b>1.3 ПОСТАНОВКА ЗАДАЧІ.....</b>	<b>15</b>
<i>1.3.1 Призначення розробки.....</i>	<i>15</i>
<b>Висновок до розділу .....</b>	<b>16</b>
<b>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>17</b>
<b>2.1 ВХІДНІ ТА ВИХІДНІ ДАНІ .....</b>	<b>17</b>
<b>2.2 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....</b>	<b>21</b>
<b>Висновок до розділу .....</b>	<b>21</b>
<b>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>22</b>
<b>3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>22</b>
<b>3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>23</b>
<b>3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ.....</b>	<b>24</b>
<b>3.4 ПОРІВНЯННЯ МЕТОДІВ МОДЕЛЮВАННЯ .....</b>	<b>25</b>
<b>Висновок до розділу .....</b>	<b>27</b>
<b>4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>28</b>
<b>4.1 ЗАСОБИ РОЗРОБКИ .....</b>	<b>28</b>
<b>4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>29</b>
<i>4.2.1 Загальні вимоги.....</i>	<i>29</i>
<b>4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>32</b>
<i>4.3.1 Діаграма класів .....</i>	<i>Ошибка! Закладка не определена.</i>
<i>4.3.2 Діаграма послідовності.....</i>	<i>32</i>
<i>4.3.3 Діаграма компонентів.....</i>	<i>34</i>
<b>Висновок до розділу .....</b>	<b>34</b>
<b>5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>35</b>



<b>5.1</b>	<b>КЕРІВНИЦТВО КОРИСТУВАЧА .....</b>	<b>35</b>
<b>5.2</b>	<b>ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>42</b>
5.2.1	<i>Мета випробувань .....</i>	42
5.2.2	<i>Загальні положення .....</i>	42
5.2.3	<i>Результати випробувань .....</i>	43
	<b>Висновок до розділу .....</b>	<b>47</b>
	<b>ЗАГАЛЬНІ ВИСНОВКИ .....</b>	<b>48</b>
	<b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>	<b>49</b>
	<b>ДОДАТОК А .....</b>	<b>50</b>

					<b>ДП ІС-5201.1181-с.ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

## ВСТУП

В теперішній час набуло дуже широкого розповсюдження використання різних форм медіа матеріалу. Важко уявити собі день, в якому людина не подивиться декілька відео, або свій улюблений серіал через Інтернет, або жодного разу не ввімкне телевізор.

Однак процес створення медіа матеріалу є дуже трудомістким, вимагає залучення багатьох типів ресурсів, а використовуючи певний медіа матеріал для навчання, або відпочинку, більшість користувачів навіть не замислювались, скільки часу та грошей було витрачено на середнє відео тривалістю 15-20 хвилин. Думки про підбір акторів, побудова плану зйомок, костюми, монтаж, знаходяться в голові продюсера, чи режисера, і очевидно, що знайти оптимальний розподіл задач у таких умовах стає занадто складною справою. Це призводить до того, що план робіт зазвичай далекий від ідеального, а деякі задачі, що були заплановані на початку робіт над проектом просто забуваються, і їх частину доводиться доопрацьовувати, або навіть перероблювати. У зв'язку із цим дуже актуальною проблемою є ефективне управління процесом створення медіаматеріалів, а також вирішення проблеми розподілу задач, що виникають в цьому процесі.

Саме тому мною було прийняте рішення розробити систему, що допоможе керівникам проектів головувати процесом створення медіа продукту, шляхом створення звіту, в якому вказані точний перелік задач, що необхідні для завершення проекту, з оптимальним розподілом виконавців на задані роботи по критеріям найшвидшого, або найбільш економічно ефективного призначення робітників. Завдяки системі продюсери та режисери зможуть точно знати, які задачі вже виконані, а які ще ні, послідовність їх виконання, та час, за який робота над проектом має бути завершена. Виконавці, у свою чергу, будуть знати свої завдання, та час на їх виконання.

					ДП ІС-5201.1181-с.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Як результат застосування даної системи, медіа продукція стане дешевшою у виробництві, що допоможе кіно команді за ті ж кошти і час, що вони витрачали раніше, зробити набагато якіснішу продукцію, що зможе привернути до себе увагу значно ширшої аудиторії, кіно критиків, зацікавить зарубіжних інвесторів, і просто примножить прибуток кінокомпанії. Масове використання системи буде значним поштовхом для українського кінематографу, і здобуття статусу світового лідера у даній галузі.

**Практичне значення одержаних результатів.** Система підтримки процесу управління створенням медіа матеріалу може бути успішно впроваджена у галузь кіноіндустрії, покращивши часові, і матеріальні показники під час роботи над проектами.

					ДП ІС-5201.1181-с.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Опис предметного середовища

У всьому світі є державні, або часні компанії, що професійно займаються кіновиробництвом. Мільйони людей виготовляють медіа матеріал для свого задоволення, використовують сучасні можливості і умови для створення якісного продукту без значних затрат на обладнання, не збираючи велику команду, часто справляючись навіть наодинці.

Сучасні відео і аудіо редактори, програми для анімації, малювання мультфільмів, не потребують професійної, технічної освіти, для їх опанування достатньо лише подивитись в інтернеті навчальне відео, або прочитати статтю про певну програму. Техніка, що використовувалась ще двадцять років тому була незручною, не була цифровою, для її використання потребувались полімерні, або паперові носії інформації. Для її опанування люди втрачали багато часу. Сучасний, якісний кінопродукт можна зняти навіть на камеру мобільного телефону. Це дозволяє перетворити колись трудомістку сферу, на хоббі будь-якої людини.

Сфера кіномистецтва активно розвивається, через надлишок різного медіа матеріалу, глядачі стали більш примхливі, і тому кінопродюсерам і режисерам доводиться значно піднімати рівень своїх робіт, застосовуючи все більше і більше різних прийомів для залучення аудиторії, що потребують значних витрат. Очевидно, чим складніше становиться будь-яка система, тим більше в ній помилок, а про деякі речі, що були заплановані на початку проекту, керівництво може і зовсім забути.

Саме тому, стає очевидною необхідність розробити систему, що допоможе працівникам телебачення, командам кіновиробництва, створювати медіа матеріал, найбільш ефективним чином використовуючи наявні ресурси. Ефективний розподіл завдань та ресурсів дозволить зменшити витрати на зв'язок між керівником проекту, та кіно командою, допоможе раціонально

					ДП ІС-5201.1181-с.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

розділити обов'язки між людьми, та вчасно завершити всю заплановану роботу.

Користувачами системи є керівники проекту, серед яких режисер і продюсер, і виконавці: оператори монтажери, аніматори.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

### 1.1.1 Опис процесу діяльності

Процес створення медіаматеріалу (розглянемо на прикладі фільму) складається з декількох етапів. На діаграмі активності керівника, виконавця, та діаграмі активності усіх членів системи, можна побачити поетапний перелік завдань, що необхідно виконати під час створення фільму.

Схема структурна активності знаходиться в графічному матеріалі.

Робота режисера і продюсера полягає у створенні нових завдань, перевірці виконаних завдань, та знаходження працівників для виконання потрібної роботи.

Робітник у системі має отримувати завдання, та звітувати про їх виконання, або не виконання.

Спочатку керівник обирає завдання з переліку, якщо завдання у списку немає, керівник через форму повинен додати його у базу даних, після чого використовувати. У формі він повинен заповнити наступні дані:

- назва роботи;
- середній час на її виконання;
- типова посада виконавця;
- можливість виконання роботи не спеціалістом.

Якщо було створено нове завдання, у базі даних робітників автоматично всім виконавцям ставиться мінімальний рівень навичку виконання роботи. Керівнику необхідно виправити це через форму змін даних робітників, задля коректного результату отриманого після генерації звіту. Далі керівник перевіряє наявність усіх необхідних робітників у системі і генерує звіт. У разі, якщо керівника не влаштовує план робіт, він може змінити порядок їх виконання за власним розсудом. Після виконання робітником певного завдання, він відмічає його у системі, що буде відображено для керівника проекту.

Виконавець, після отримання нового завдання виконує його, і відмічає у системі успішність роботи.

					ДП ІС-5201.1181-с.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.1.2 Опис функціональної моделі

Акторами системи є: керівник проекту, виконавець.

Функціональну модель системи зобразимо за допомогою діаграми варіантів використання. Схема структурна варіантів використання знаходиться в графічному матеріалі. На діаграмі зображено можливості акторів у системі.

Перелік того, що повинні робити актори в системі:

Керівник проекту - організує виробництво певного художнього витвору, здійснює художній контроль за ходом виробництва, здійснює організацію творчо-виробничого процесу створення фільму. На основі літературного сценарію розробляє художню концепцію фільму.

Обов'язки керівника проекту:

- створення проекту;
- пошук працівників;
- видача завдань;
- перегляд звітів.

Виконавець - всі, хто беруть участь у реалізації проекту(оператор, фотограф, монтажер...)

Обов'язки виконавця:

- отримати завдання;
- відмітити виконані завдання.

Таблиця 1.1 – Типи залежностей між варіантами використання

<i>Актор</i>	Варіант використання	Опис дії варіанта використання
Виконавець	Відмітити виконані завдання	Актор ставить мітку "Виконано" біля завершеного завдання.

## Продовження таблиці 1.1

Актор	Варіант використання	Опис дії варіанта використання
Виконавець	Авторизація	Виконавець вводить свій логін і пароль, після чого входить в систему.
Керівник проекту	Створити проект	Створити новий проект, вказавши його тип, та назву.
	Створити список завдань	Обрати завдання, які необхідно виконати для вдалого завершення проекту.
	Генерувати звіт	Натиснувши відповідну кнопку отримати звіт, і за потребою змінити його.
	Додати нове завдання	Створити у системі нове завдання, додавши його у відповідну базу даних.
	Додати нового виконавця	Створити у системі нового виконавця додавши його у відповідну базу даних.
	Роздати завдання виконавцям	Після погодження керівника проекту зі звітом, роздати завдання виконавцям.

**1.2 Огляд наявних аналогів**

Система, що розроблюється повинна забезпечувати ефективну організацію робочого процесу та надавати можливість найбільш ефективним чином розподіляти наявні ресурси. Від аналогів дана система відрізняється тим, що в ній залишилися лише ті пункти, що необхідні для створення медіа матеріалу, усі інші були видалені. Програма також відрізняється тим, що

					ДП ІС-5202 1180-с.ПЗ	Арк.
						1
Змн.	Арк.	№ докум.	Підпис	Дата		



людині не потрібно буде вводити в програму десятки числових значень, програма буде сама вирішувати завдання, і пропонувати оптимальний варіант його вирішення.

Переглянемо існуючі аналоги розробленої системи:

- органайзер **THINGS** [1];

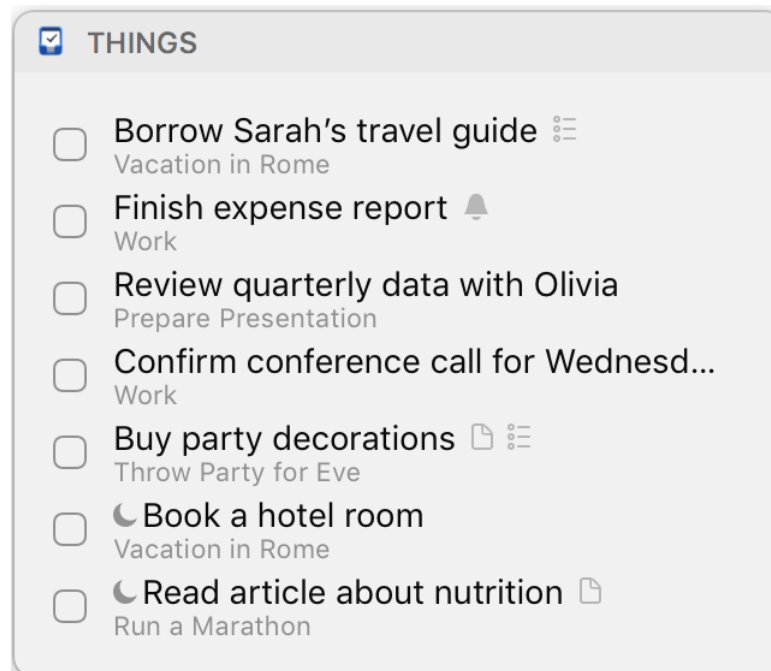


Рисунок 1.1 – Демонстрація роботи органайзера THINGS

Потужний засіб зі зручним, простим дизайном. Недоліком даного органайзера є те, що він допоможе організувати роботу, лише однієї людини, команда не зможе користуватись things.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

– програма для організації спільної роботи **ЛидерТаск** [2];

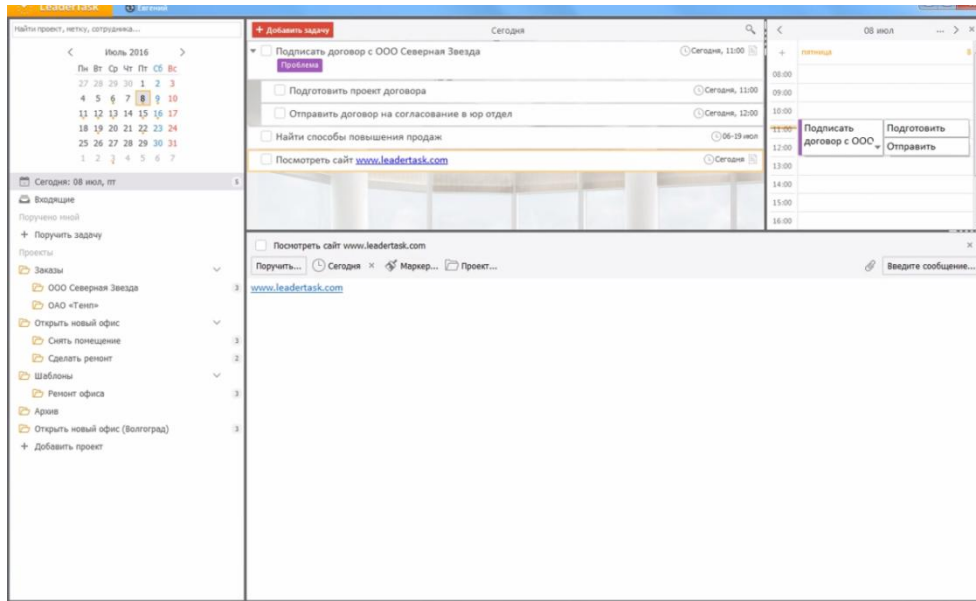


Рисунок 1.2 – Демонстрація роботи програми для організації спільної роботи ЛидерТаск

Один з найбільш потужних конкурентів розроблюваної системи. Зручний дизайн, багато потрібних функцій, швидка організація командної роботи. Недоліком системи є відсутність математики, а тому користувачу-керівнику буде складніше обрати оптимальний план робіт.

– веб сервіс **Todoist** [3].

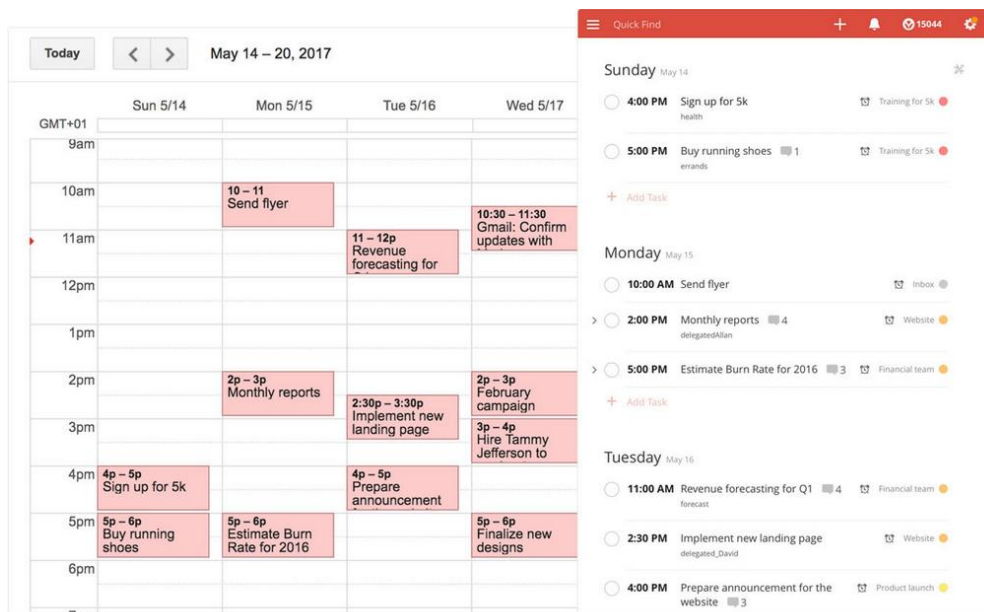


Рисунок 1.3 – Демонстрація роботи веб сервісу Todoist

Змн.	Арк.	№ докум.	Підпис	Дата

Ще один близький по функціоналу засіб для організації роботи. Один з найпопулярніших у світі веб-сервіс для організації роботи.

Таблиця 1.2 – Порівняння даного продукту і Things

Критерії порівняння	Порівнюваний продукт	
	Даний продукт	Things
Мова інтерфейсу	Українська	Англійська
Можливість командної роботи	Наявна	Відсутня
Обмеження у кількості проектів	Відсутня	Усі задачі в одному проекті
Обмеження у кількості учасників	Відсутня	Один
Допомога у реалізації планів	Математичні задачі для оптимізації роботи	Відсутня
Підтримка платформ	Windows	Ios, MacOS
Розширення для браузерів	Відсутня	Відсутня

Таблиця 1.3 – Порівняння ЛидерТаск і Todoist

Критерії порівняння	Порівнюваний продукт	
	ЛидерТаск	Todoist
Мова інтерфейсу	Російська	Багатоязичний
Можливість командної роботи	Наявна	Наявна
Обмеження у кількості проєктів	Відсутня	200 проєктів
Обмеження у кількості учасників	Відсутня	50 учасників
Допомога у реалізації планів	Вбудований месенджер	Відсутня
Підтримка платформ	Windows	Win,Android,Ios,MacOs ,Linux, WinPhone
Розширення для браузерів	Відсутня	Наявна

### 1.3 Постановка задачі

#### 1.3.1 Призначення розробки

Призначенням системи є підтримка процесу створення медіа контенту.

#### 1.3.2 Цілі та задачі розробки

Ціль системи є спрощення процесу створення медіа контенту, шляхом ефективної організації роботи і розподілу задач.

Для досягнення поставленої цілі мають бути вирішені такі задачі:

- ведення проєктів, завдань, виконавців;
- створення інтерфейсу системи;

- реалізація алгоритмів вирішення задачі, що буде знаходити оптимальний план робіт;
- формування звітів.

### Висновок до розділу

В даному розділі було обґрунтовано доцільність створення системи підтримки процесу управління створення медіа матеріалу. Були висунуті основні вимоги до системи, виявлені основні групи користувачів, для яких може бути корисною дана система. Були наведені основні аналоги – організатори для різних платформ, чий функціонал максимально наближений до розроблюваної програми. Був проведений аналіз існуючих аналогів, на основі якого зроблено висновок про доцільність створення системи.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні та вихідні дані

Вхідні дані для системи підтримки процесу управління створення медіа матеріалу надходять від користувача системи у вигляді представленому у таблиці 2.1.

Таблиця 2.1 – Вхідні дані

<i>Дані</i>	<i>Опис</i>
Дані для бази даних виконавців	Дані користувачів, які заповнюються у формах авторизації, та використовуються системою для розрахунку оптимального плану робіт.
Дані для бази даних робіт	Дані, що використовуються системою для розрахунку оптимального плану робіт.
Перелік робіт і виконавців, що працюють над проектом	Інформація, що надає можливість системі розробити план, використовуючи необхідних виконавців.

Вихідні дані представлені в таблиці 2.2.

Таблиця 2.2 – вихідні дані

<i>Дані</i>	<i>Опис</i>
Звіт з оптимальним планом по часовому параметру	Звіт, що отримає керівник проекту, де вказаний план є таким, що витрата часу на завершення проекту буде мінімальним.
Звіт з оптимальним планом по параметру грошових витрат	Звіт, де план є таким, що витрата грошей на завершення проекту буде мінімальною.

На рисунку 2.1 зображений звіт з планом по часовому параметру, що отримає керівник.

На рисунку 2.2 зображений звіт з планом по параметру грошових витрат, що отримає керівник.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

**Звіт**

Оптимальний розподіл робіт (по параметру мінімального часу) між працівниками у проєкті "Новий проєкт"

Перелік задач, та їх виконавці:

Найменування роботи	Назва роботи	Виконавець	Посада	Дата закінчення роботи
Фільмування	Сцена 1	Антищенко В.В.	Оператор	24.12.19(12:30)
Фільмування	Сцена 4	Кань В.Д.	Оператор	20.12.19(22:55)
Монтаж	Сцена 1	Бедов В.К.	Монтажер	26.12.19(14:00)
Монтаж	Сцена 4	Бедов В.К.	Монтажер	22.12.19(14:00)
Монтаж	Загальний	Бедов В.К.	Монтажер	29.12.19(14:00)
Звукозапис	Редакція звуку, додавання музики, звукові ефекти	Паганець И.П.	Інженер звукозапису	10.01.20(13:55)
Фотографування	На обкладинку	Сева К.Р.	Фотограф	14.12.19(10:00)

Рекомендована кількість робітників - 5 з яких:

- Оператор 2
- Монтажер 1
- Інженер звукозапису 1
- Фотограф 1

Приблизна дата завершення проєкту 10.01.20.

Витрати 25431грн.

5 Грудня 2019р.(22:30)

Рисунок 2.1 – Приклад звіту з планом, що мінімізує витрати часу



## Звіт

Оптимальний розподіл робіт (по параметру мінімальних витрат грошей) між працівниками у проекті "Новий проект"

Перелік задач, та їх виконавці:

Найменування роботи	Назва роботи	Виконавець	Посада	Дата закінчення роботи
Фільмування	Сцена 1	Копотенко Д.С.	Оператор	27.12.19(15:30)
Фільмування	Сцена 4	Жане В.Д.	Оператор	25.12.19(22:55)
Монтаж	Сцена 1	Бедов В.К.	Монтажер	3.01.20(19:00)
Монтаж	Сцена 4	Бедов В.К.	Монтажер	29.12.19(14:00)
Монтаж	Загальний	Бедов В.К.	Монтажер	04.02.20(19:00)
Звукозапис	Редакція звуку, додавання музики, звукові ефекти	Пинько І.П.	Інженер звукозапису	18.02.20(13:55)
Фотографування	На обкладинку	Сева К.Р.	Фотограф	14.12.19(10:00)

Рекомендована кількість робітників - 5 з яких:

- Оператор 2
- Монтажер 1
- Інженер звукозапису 1
- Фотограф 1

Приблизна дата завершення проекту 19.02.20.

Витрати 15335грн.

5 Грудня 2019р.(22:30)

Рисунок 2.2 – Приклад звіту з планом, що мінімізує витрати грошей

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

## 2.2 Опис структури бази даних

Використовуючи утиліту Microsoft SQL Server Management Studio сформовано дві таблиці: «users» і «tasks».

Опишемо їх призначення:

- users - таблиця, в якій розміщені дані користувачів для авторизації і обробки інформації за допомогою системи;
- tasks - таблиця, в якій присутні дані про завдання, що будуть використовувати виконавці.

Схема структурна бази даних наведена в графічному матеріалі.

### Висновок до розділу

У даному розділі були описані вхідні та вихідні дані для системи підтримки процесу управління створення медіа матеріалу. Розроблена, та впроваджена база даних, описана її структура. Були наведені приклад звіту, що пропонується користувачу після вводу вхідних даних.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Задача про призначення є задачею комбінаторної оптимізації в галузі оптимізації, або дослідження операцій в математиці. До її розв'язання може бути застосований угорський метод, або метод Мака, що може достатньо точно визначити, яких робітників слід направити на роботу, для максимально швидкої реалізації проекту [4].

Призначенням цієї задачі є знаходження оптимального рішення про призначення робітників на роботу за критерієм.

У найкращому варіанті кількість робітників і робіт співпадають, але у системі підтримки процесу управління створення медіа матеріалу, у більшості випадків такий розвиток подій може бути лише випадковим, зазвичай кількість робіт буде значно перевищувати кількість робітників [5].

Уперше задача про призначення була розглянута в геометричній формі Гаспаром Монжем у 1784 році. Проте на початку ХХ століття була встановлена некоректність розв'язку Монжа. Наступні кроки в розв'язанні задачі про призначення зроблені Кенігом і Егерварі в першій третині ХХ століття. Кеніг і Егерварі розглядали цю задачу як задачу пошуку досконалого паросполучення мінімальної ваги у зваженому дводольному графі. Їх роботи стали основою для угорського методу, розробленого Куном у 50-х роках минулого століття [6]. У 1947 році Данциг запропонував симплекс-метод для розв'язання загальної задачі лінійного програмування, до якої легко зводиться задача про призначення. Задача про призначення, поставлена Данцигом і Фалкерсоном, може також розглядатися як задача про максимальний потік мінімальної вартості. У 1961 році Басакер і Гоуен опублікували алгоритм для її розв'язання. Цей алгоритм для загальної задачі, як і алгоритм симплекс-методу, має експоненціальну складність, а для задачі про призначення - поліноміальну. Поліноміальний алгоритм для задачі про

					ДП ІС-5201.1181-с.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

максимальний потік мінімальної вартості, заснований на алгоритмі Клейна, що був опублікований у 1967 році, вперше запропонований у 1987 році Гольбергом і Гар'яном.

Теоретичний аналіз складності алгоритмів показує, що алгоритми Куна й Басакера-Гоуена мають однакову теоретичну складність, меншу, ніж алгоритм Гольдберга-Гар'яна. Проте відповідь на питання про визначення найкращого алгоритму вимагає емпіричних досліджень [7]. Аналізуючи отримані результати, можна зробити висновок: найбільш швидким алгоритмом для розв'язання задачі про призначення є угорський метод [8].

### 3.2 Математична постановка задачі

Для знаходження оптимального рішення нам знадобляться наступні дані:

- множина  $A$ (agents, виконавці);
- множина  $T$ (tasks, задачі);
- вагова функція  $C$ (costs, час виконання певної роботи(або витрати));

Задача полягає у тому, що нам потрібно знайти бієкцію  $A \rightarrow T$ , таку, щоб функція загальних витрат  $\sum_{a \in A} C(a, f(a))$  мала б мінімальне значення.

Зазвичай вагова функція  $C(a, t)$  задається як множина чисел  $C_{ij}$  для усіх пар  $(i, j)$ , де  $i \in A, j \in T$ .

Задача записується, як ЗЛП:

$$\sum_{i \in A} \sum_{j \in T} C_{ij} x_{ij} \quad (3.1)$$

з обмеженнями:

$$\sum_{j \in T} x_{ij} = 1 \text{ для усіх } i \in A \rightarrow \min \quad (3.2)$$

$$\sum_{i \in A} x_{ij} = 1 \text{ для усіх } j \in T \quad (3.3)$$

$$x_{ij} \geq 0 \text{ для усіх } (i, j) \in A \times T \quad (3.4)$$

					ДП ІС-5201.1181-с.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.3 Обґрунтування методу розв'язання

Для вирішення поставленої задачі було вирішено обрати Угорський метод. Особливості проблем призначення на роботи породили ефективний метод їх вирішення. Основною ідеєю угорського методу є перехід від початкової матриці витрат  $C$  до еквівалентної матриці  $C_e$  з невід'ємними елементами і системою з  $n$  незалежних нулів, з яких два не належать до одного рядка або ж стовпця. Для заданого  $n$  існує  $n!$  прийнятних рішень. Якщо в матриці присвоєння  $X$  є  $n$  одиниць таким чином, що кожен рядок і стовпець містять тільки одну одиницю, розташовану відповідно до  $n$  незалежних нулів еквівалентної вартості матриці  $C_e$ , то отримуємо дійсні рішення задачі призначення.

Для того, щоб показати реалізацію методу в застосунку, наведемо псевдокод даного методу, в якому масив `way` містить інформацію про місця досягнення мінімумів, що згодом допоможе відновити послідовність дій. Алгоритм являє собою зовнішній цикл по строкам матриці, всередині якого відбувається додавання на розгляд  $i$ -ої строки матриці. Внутрішня частина являє собою цикл `do-while` який працює, поки не буде знайдений вільний стовпець  $j$ . Кожна ітерація циклу позначає відвіданих новий стовпець з номером  $j_0$ , що порахували минулої ітерації, що спочатку був рівним нулю. Через появу нових відвіданих рядків  $i_0$  потрібно відповідним чином перерахувати масив `minv`, заодно знайшовши величину `della`, що являється мінімумом даного масиву. `INF` - константа, що являється числом, що  $\rightarrow \infty$

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

```

vector<int> u (n+1), v (m+1), p (m+1), way (m+1);
for (int i=1; i<=n; ++i) {
    p[0] = i;
    int j0 = 0;
    vector<int> minv (m+1, INF);
    vector<char> used (m+1, false);
    do {
        used[j0] = true;
        int i0 = p[j0], delta = INF, j1;
        for (int j=1; j<=m; ++j)
            if (!used[j]) {
                int cur = a[i0][j]-u[i0]-v[j];
                if (cur < minv[j])
                    minv[j] = cur, way[j] = j0;
                if (minv[j] < delta)
                    delta = minv[j], j1 = j;
            }
        for (int j=0; j<=m; ++j)
            if (used[j])
                u[p[j]] += delta, v[j] -= delta;
            else
                minv[j] -= delta;
        j0 = j1;
    } while (p[j0] != 0);
    do {
        int j1 = way[j0];
        p[j0] = p[j1];
        j0 = j1;
    } while (j0);
}

```

Рисунок 3.1 – Псевдокод Угорського методу вирішення задачі

Слід мати на увазі, що для будь-якого неприйняттого призначення відповідне значення умовно приймається рівним достатньо великому числу  $M$  в мінімальних завданнях. Якщо вихідна матриця не є квадратною, то потрібно ввести додаткову необхідну кількість рядків або стовпців, а їхні елементи можуть бути приписані значенням, визначеним умовами проблеми, можливо, після редукації, а домінуючі альтернативи, що коштують дорого, або дешево виключити.

### 3.4 Порівняння методів моделювання

Порівняємо Угорський метод вирішення задачі про призначення і ще один класичний метод Мака. Для цього зробимо окрему програму, що буде вирішувати матриці розмірності від 100 на 100, до 1000 на 1000.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

На рисунку 3.1 графічно зображено час, що необхідний для розв'язання квадратної матриці двома запропонованими методами. Лістинг програми знаходиться у розділі Додаток А.

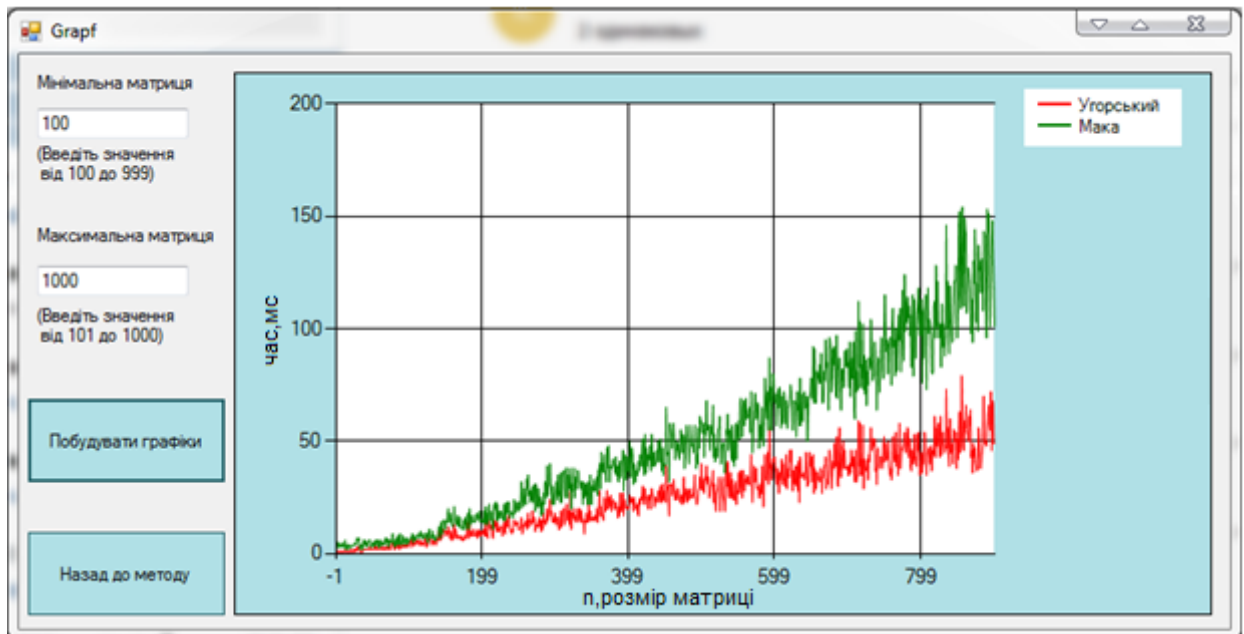


Рисунок 3.2 – Графік порівняльний швидкості роботи методів розв'язання задачі про призначення

Як можна побачити на графіку, що відображає час, необхідний програмі для обчислення матриці розмірністю до 1000 на 1000 Угорський алгоритм стає ефективнішим на 250-300%, отже його використовувати значно краще. Дійсно, для генерації звіту, задачу необхідно буде вирішити лише 2 рази, а розмірність більшості матриць буде меншою за 50 елементів, отже звіт буде генерований лише на 1мс. швидше. Але, в деяких медіа продуктах можуть бути задіяні сотні виконавців, а також не менша кількість робіт. Отже звіт при таких об'ємах даних може бути згенеровано більш ніж на 20 секунд швидше. Якщо керівник проекту не зможе одразу побачити звіту, він може подумати, що система працює некоректно, тим самим зіпсувати рейтинг продукту.

Змн.	Арк.	№ докум.	Підпис	Дата

**Висновок до розділу**

В даному розділі була сформульована змістовна та математична постановки задачі оптимального розподілу виконавців та їх завдань. Наведено результат застосування алгоритму вирішення даної задачі. Показано ефективність методу у порівнянні з іншими. Обґрунтовано необхідність використання саме Угорського методу.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27



## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Програмне забезпечення, що використовується при розробці:

- платформа: OS Windows – Операційна система для комп'ютерів, розроблена компанією Microsoft;
- середовище розробки: Microsoft Visual Studio – Інтегроване середовище розробки програмного забезпечення;
- мова програмування: С# – Об'єктно орієнтована мова програмування;
- РСУБД(реляційна система управління базою даних): MS SQL SMS – Утиліта для керуванням і адмініструванням базами даних;
- мова програмування: SQL[ – Декларативна мова програмування, створена для керування даними у РСУБД;
- пакет застосувань: Microsoft Office – Офісний пакет для роботи з різними типами документів;
- інструмент моделювання: Sparcs Systems Enterprise Arcitect – Інструмент візуального проектування і моделювання засновану на UML;
- мова моделювання: UML – Уніфікована мова моделювання, що використовується під час розробки програмного забезпечення;
- веб сервер: Apache – http сервер, що необхідний для розгортання бази даних;
- мова програмування: PHP – Мова призначена для розробки веб застосувань;
- графічний редактор: Abobe Photoshop – Редактор растрових зображень;

					ДП ІС-5201.1181-с.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

- графічний редактор: Adobe Illustrator – Редактор векторних зображень;
- відео редактор: Edius – Відеоредактор для ОС Windows;
- застосунок: Bandicam – Програма для запису екрана.

Основним засобом розробки стане інтегрована середа розробки програмного забезпечення Microsoft Visual Studio.

Дана середа дозволяє розробляти як консольні, так і графічні програми на мовах програмування C,C++,C#. Для програмування у Visual Studio була обрана мова C#.

Одною з причин вибору Visual Studio стала можливість застосувати Microsoft SQL Server, системою керування реляційних баз даних, які будуть застосовуватись у роботі середовища.

Для створення якісних баз даних, керуванням і адмініструванням всіх елементів Microsoft SQL Server, буде використана утиліта Microsoft SQL Server Managment Studio (Microsoft SQL SMS).

## **4.2 Вимоги до технічного забезпечення**

### **4.2.1 Загальні вимоги**

Система підтримки процесу управління створення медіа матеріалу не буде використовувати значної кількості ресурсів комп'ютера, адже найскладніше обчислення, що буде виконуватись - це обчислення матриці, один раз за цілий проект. Якщо комп'ютер користувача не буде дуже потужним, процес обрахування матриці може зайняти декілька хвилин.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Для правильної роботи даної програми до складу технічних засобів повинні входити:

- процесор з тактовою частотою не нижче 2.0 ГГц;
- оперативна пам'ять об'ємом не менше 1 ГБ;
- мережева карта;
- жорсткий диск чи SSD з об'ємом пам'яті не менше 5Гб;
- операційна система Windows 7, або новіші;
- Microsoft Word 2007, або новіші;
- монітор;
- мишка;
- клавіатура.

Наведемо функціональні і не функціональні вимоги до системи:

Обов'язкові вимоги:

Функціональні вимоги:

- користувач має можливість авторизуватись в системі;
- продюсер має можливість створювати, редагувати, та видаляти проекти безліч разів;
- усі користувачі-керівники можуть відправити завдання будь-якому користувачу-виконавцю;
- кожен створений проект повинен мати продюсера. Інших акторів може бути безліч, або не бути взагалі;
- усі користувачі-виконавці мають можливість відправити звіт користувачу-керівнику, що дав завдання;
- продюсер та режисер можуть додавати у проект скільки завгодно нових користувачів;
- усі завдання повинні бути записані у базу даних.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Не функціональні вимоги:

- режим роботи системи має бути цілодобовим;
- нормальна кількість збоїв в роботі системи не більше 1-2 за квартал;
- середній час відновлення роботи після збою, не більше 2 годин;
- розробка має вестись на платформі Visual Studio версії 2019 року;
- бази даних у системі повинні бути виконані у Microsoft SQL SMS версії 2010-2018 років;
- швидкість відклику системи на сигнали має бути не більше 0.5 секунд;
- під час розробки системи, особиста зустріч між представником розробника та замовника має відбуватись не менше ніж 1 раз на місяць;
- будь-які розходи на розробку системи мають обговорюватись між розробником та замовником особисто;
- система має відкриватись на комп'ютерах з системними характеристиками: Операційна система Windows 7 обсяг оперативної пам'яті 1Гб ;
- користувачі системи не повинні мати доступ до панелі адміністратора, або до даних інших користувачів.

Дуже бажані вимоги:

Функціональні вимоги:

- усі користувачі повинні бути записані у базі даних;
- система повинна запропонувати режисеру чи продюсеру список типових завдань до користувачів-виконавців;
- у списку типових завдань, система повинна показувати в першу чергу ті завдання, що підходять під спеціальність користувача-виконавця.

					ДП ІС-5201.1181-с.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Не функціональні вимоги

- обсяг пам'яті виділений на систему - не більше 5Гб.;
- інтерфейс програми має не бути різнокольоровим. Максимально допустима кількість кольорів на сторінці(не включаючи фотографії, або картинки) - 7 кольорів;
- шрифт у програмі повинен не бути менше 10п'єків, але і не більше 20.

Бажані вимоги:

Функціональні вимоги:

- якщо актор приймав участь у роботі над проектом, у базі даних повинна відобразитися така інформація, навіть якщо проект вже давно не існує;
- якщо користувач - виконавець отримав нове завдання, потрібно повідомити це йому, за допомогою електронної пошти;
- якщо користувач - виконавець отримав нове завдання, та впродовж половини строку його виконання, або 48 годин, не заходив у систему, потрібно повідомити керівника, за допомогою електронної пошти.

Не функціональні вимоги:

- при авторизації користувача, біометричні дані повинні не використовуватись;
- шрифт у програмі повинен бути Times New Roman, Arial або Calibri.

## 4.3 Архітектура програмного забезпечення

### 4.3.1 Діаграма класів

Діаграма класів — діаграма в UML, що демонструє загальну структуру класів у системі, та їх взаємозв'язок. Діаграма класів, є одною з найбільш важливих демонстрацій майбутньої системи.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

В даній діаграмі представлена взаємодія між класами, що використовуються в головному застосунку.

Схема структурна класів програмного забезпечення наведена у графічному матеріалі.

### 4.3.2 Діаграма послідовності

Діаграма послідовності — ще один вид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень. На рисунку 4.1 проілюстрована діаграма послідовності, де зображені основні об'єкти і актори, що взаємодіють між собою на певному проміжку часу.

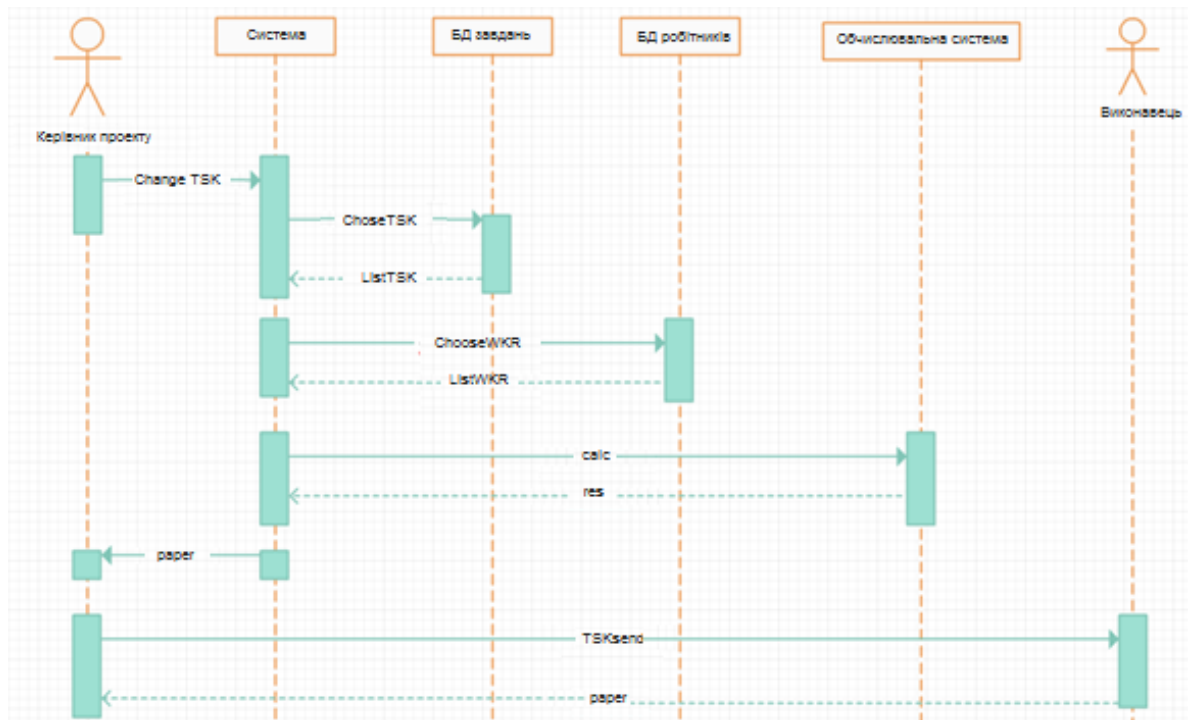


Рисунок4.1 – Схема структурна послідовності

Голова проекту взаємодіє із системою, для надання заповнення списку завдань. Система у свою чергу взаємодіє з базами даних, рахує дані, що вони запропонували за допомогою обчислювальної системи.

Після знаходження оптимального варіанту, продюсер отримує звіт, у якому затверджує план робіт і призначає виконавців.

### 4.3.3 Діаграма компонентів

Діаграма компонентів в UML - діаграма на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

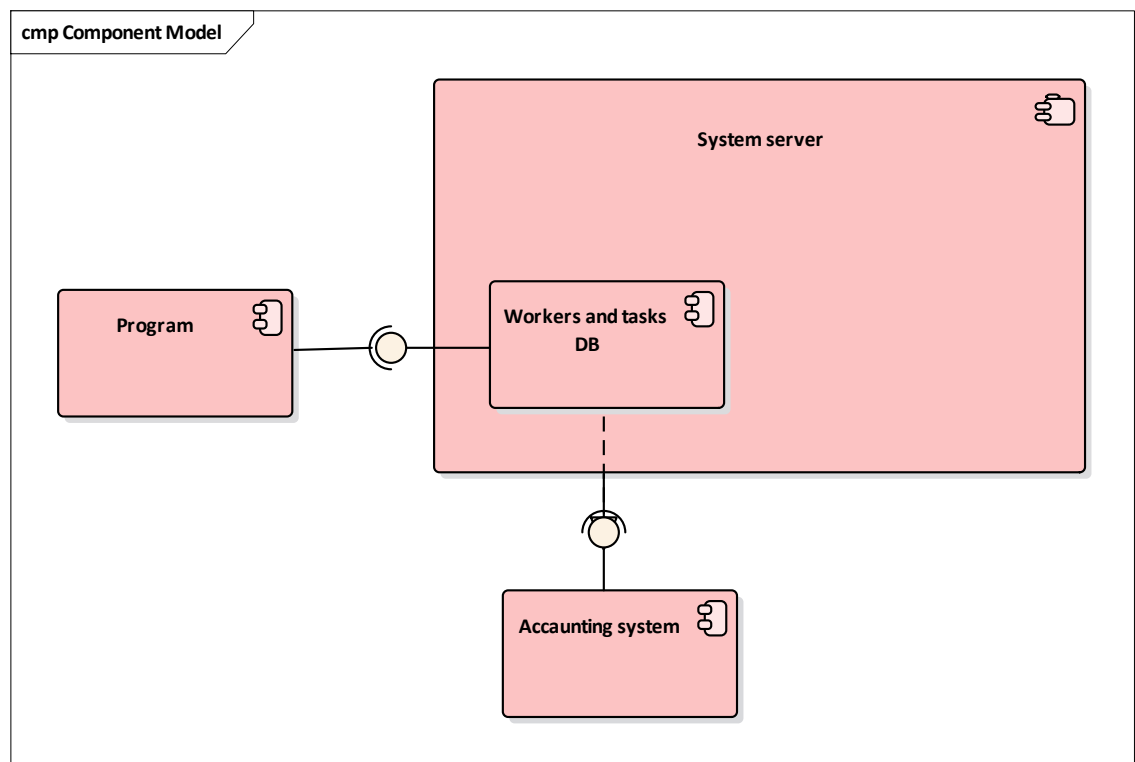


Рисунок 4.2 – Схема структурна компонентів

### Висновок до розділу

В даному розділі було обґрунтовано вибір інструментів, за допомогою яких було реалізовано програмний продукт. Були продемонстровані діаграми, що необхідні для розуміння структури системи.

Змн.	Арк.	№ докум.	Підпис	Дата

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

На початковій формі знаходиться форма авторизації (рисунок 5.1).

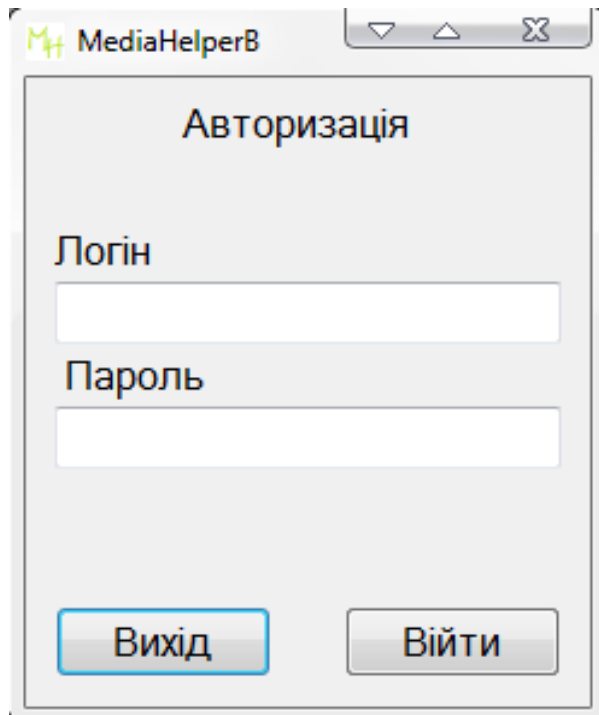


Рисунок 5.1 – Форма авторизації користувача

На даному полі необхідно заповнити поля з логіном та паролем, після чого користувач переходить на свою головну сторінку. Головна сторінка користувача продемонстрована на (рисунок 5.2), і (рисунок 5.3) відповідно сторінка виконавця, та керівника проекту.

Змн.	Арк.	№ докум.	Підпис	Дата



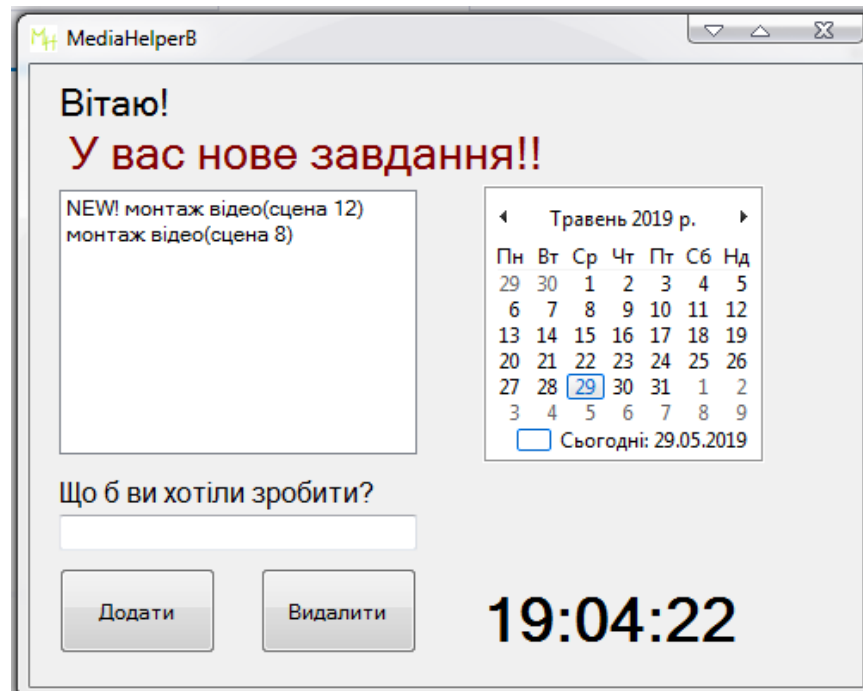


Рисунок 5.2 – Головна сторінка виконавця

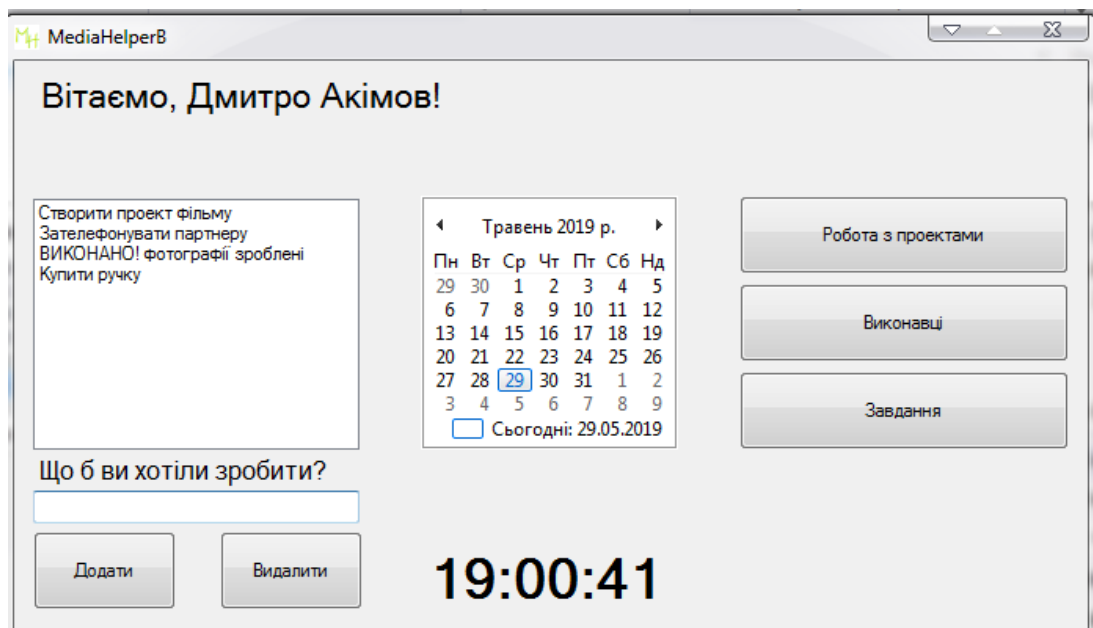


Рисунок 5.3 – Головна сторінка керівника проекту

Щоб переглянути виконавців і завдання, необхідно натиснути відповідну кнопку. На рисунку 5.4 зображена форма, на якій відображається перелік виконавців. Керівник проекту має можливість додати нового користувача, змінити його дані, або видалити з системи.

Змн.	Арк.	№ докум.	Підпис	Дата

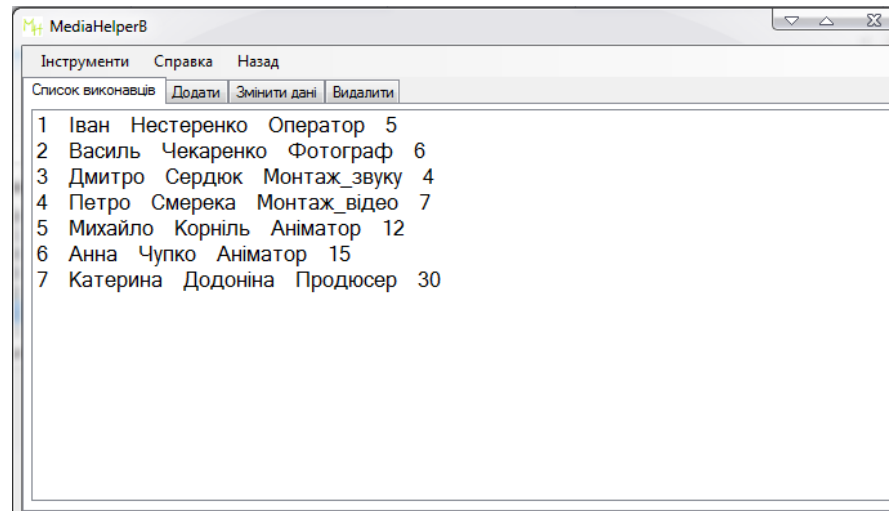


Рисунок 5.4 – Перегляд бази даних виконавців

Для того, щоб зареєструвати нового користувача, його дані необхідно додати до бази даних робітників. Таке право надано керівникам проекту, що мають доступ до баз даних. Для того, щоб додати нового виконавця необхідно заповнити поля на формі, що зображені на (рисунок 5.5.) У полях вводу необхідно заповнити наступні дані: Ім'я, прізвище, головна спеціальність, навички роботи по спеціальностям, доступ до баз даних. Натиснувши кнопку змінити дані, та видалити, можна виконати відповідні дії з користувачем (рисунок 5.6),( рисунок 5.7.)

Ім'я	<input type="text"/>	Монтаж звук	<input type="text"/>
Прізвище	<input type="text"/>	Монтаж відео	<input type="text"/>
Спеціалізація	<input type="text"/>	Анімація	<input type="text"/>
Доступ до БД	<input type="text"/>	Графіка	<input type="text"/>
Фото	<input type="text"/>	Оплата(год)	<input type="text"/>
Відео	<input type="text"/>	<input type="button" value="Додати"/>	

Рисунок 5.5 – Форма додавання виконавця в базу даних

MediaHelper8

Інструменти Справка Назад

Список виконавців Додати Змінити дані Видалити

Ім'я  Монтаж звук

Прізвище  Монтаж відео

Спеціалізація  Анімація

Доступ до БД  Графіка

Фото  Оплата(год)

Відео   Id

Рисунок 5.6 – Зміна даних користувача

MediaHelper8

Інструменти Справка Назад

Список виконавців Додати Змінити дані Видалити

Id

Рисунок 5.7 – Видалення користувача з системи

Після перевірки бази даних користувачів, керівник проекту має повторити процес з базою даних завдань. У цій базі також можна додати нове завдання(Рисунок 5.8), змінити існуюче(Рисунок 5.9) і видалити непотрібне (Рисунок 5.10).

Змн.	Арк.	№ докум.	Підпис	Дата

TaskDB

Інструменти Справка Назад

Список завдань Додати Змінити Видалити

Назва

Час

Опис

Додати

Рисунок 5.8 – Форма додавання завдання в базу даних

TaskDB

Інструменти Справка Назад

Список завдань Додати Змінити Видалити

Назва

Час

Опис

Змінити

Рисунок 5.9 – Зміна завдання

Змн.	Арк.	№ докум.	Підпис	Дата

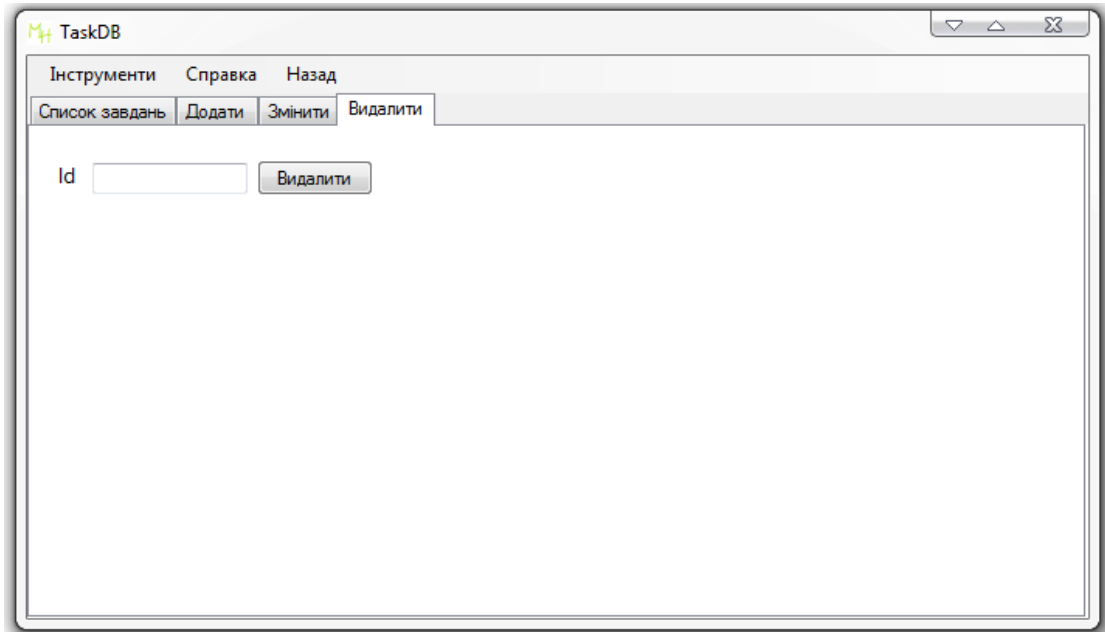


Рисунок 5.10 – Видалення завдання з системи

Після того, як керівник проекту додав у бази даних усіх необхідних робітників, і задачі, які вони будуть виконувати, користувач може створювати новий проект. Даний процес зображений на (рисунку 5.11) Після створення нового проекту необхідно обрати завдання, що мають бути виконані для успішного завершення проекту. Дана процедура зображена на (рисунку 5.12)

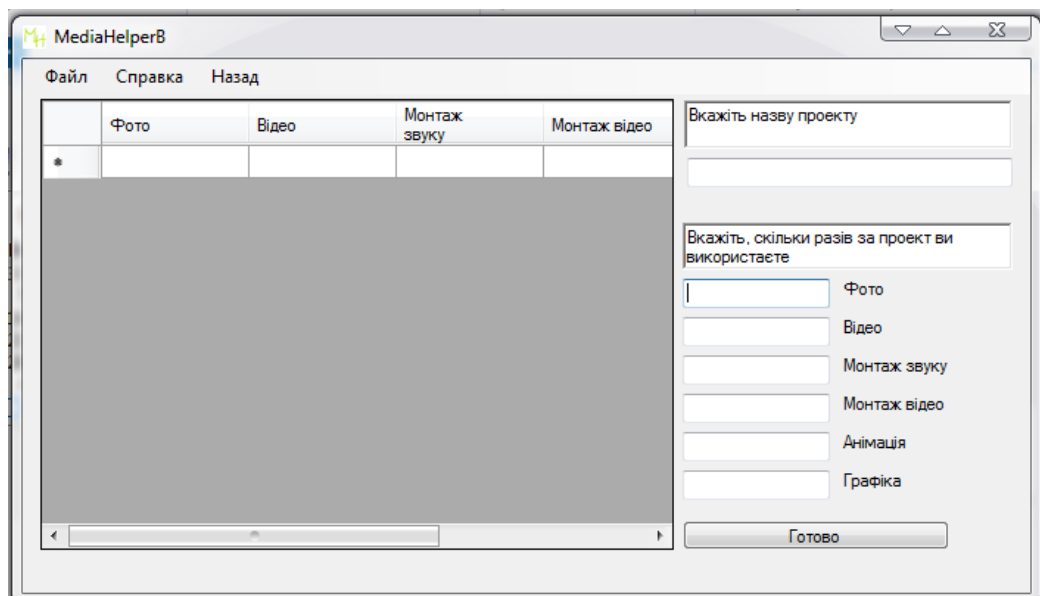


Рисунок 5.11 – Початок створення нового проекту

Змн.	Арк.	№ докум.	Підпис	Дата

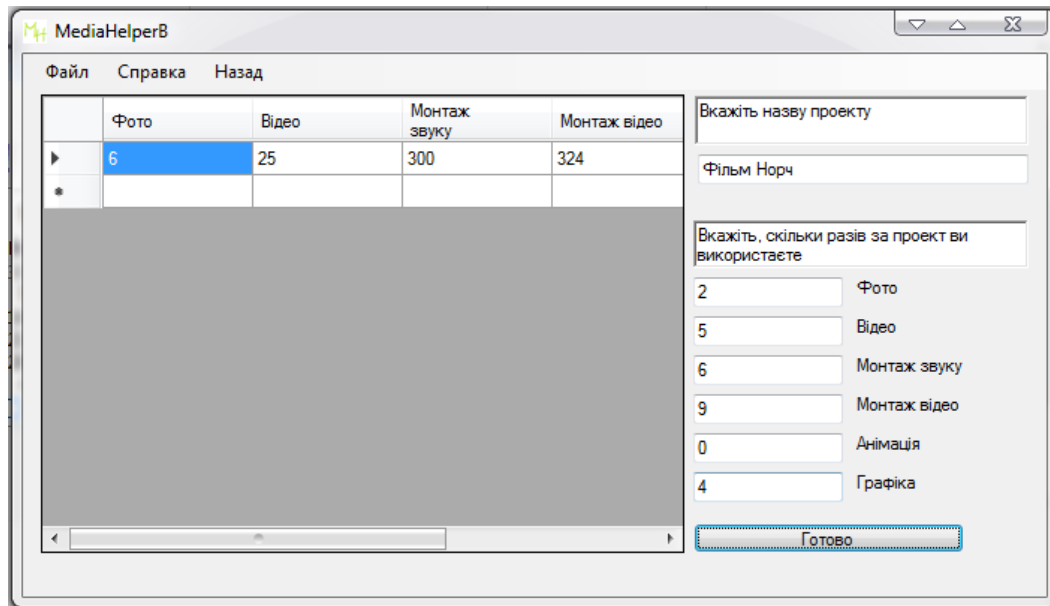


Рисунок 5.12 – Заповнення полей для створення проекту

Після того, як усі необхідні завдання були додані, керівник генерує звіт, що були зображені на (рисунку 2.1) і (рисунок 2.2) у розділі 2.1.

Після підтвердження звіту, виконавцям, що будуть задіяні у проекті, прийде повідомлення про нову роботу, дана ситуація зображена на (Рисунку 5.13), і після її завершення, виконавець відмічає даний факт, як зображено на (Рисунок 5.14)

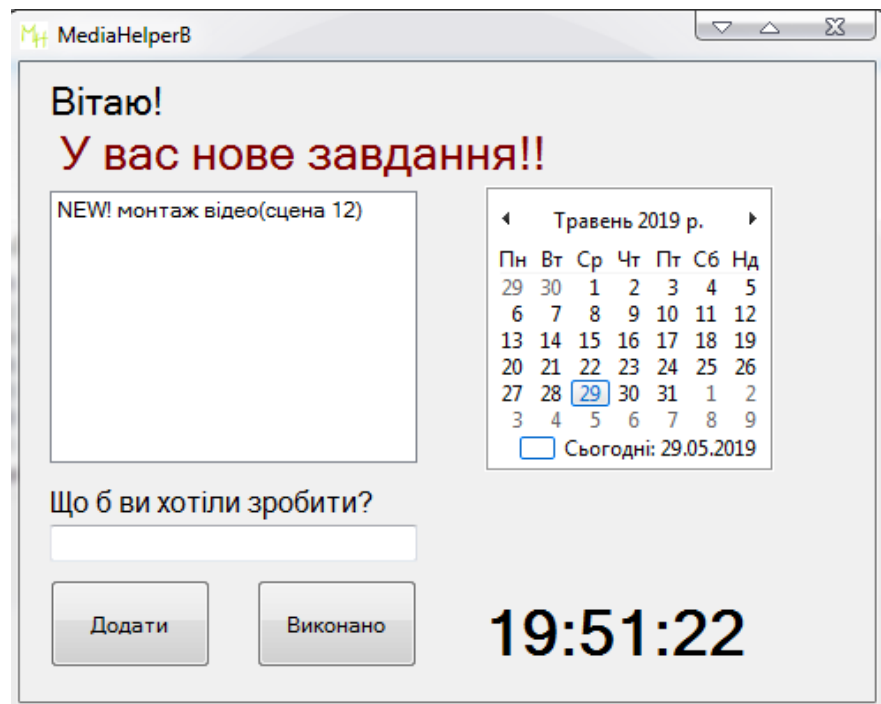


Рисунок 5.13 – Нове завдання було надіслано робітнику

Змн.	Арк.	№ докум.	Підпис	Дата

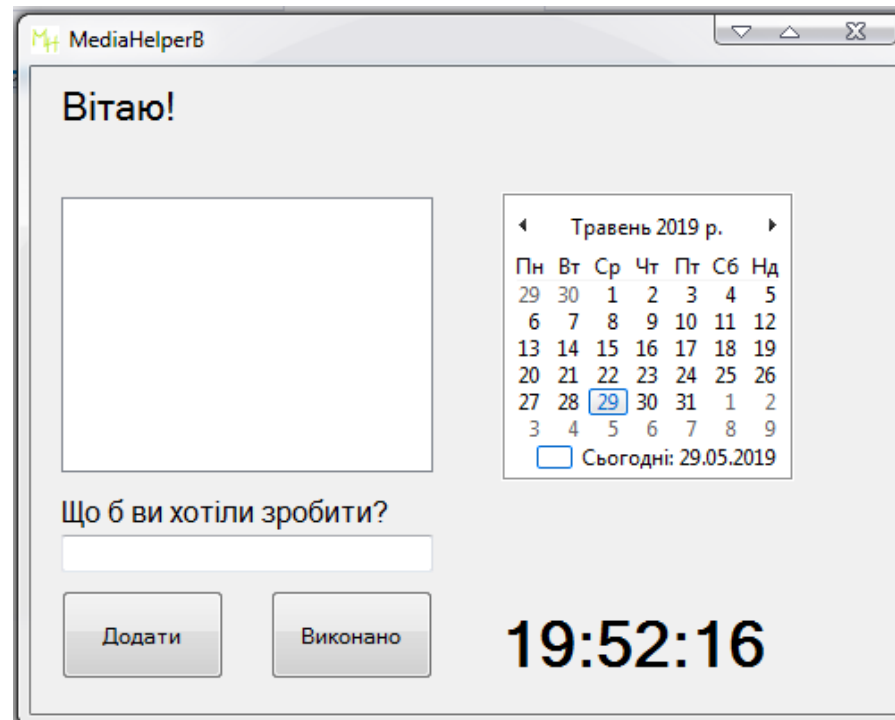


Рисунок 5.14 – Виконавець відмічає виконане завдання

## 5.2 Випробування програмного продукту

### 5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач Система підтримки процесу управління створення медіа матеріалу вимогам технічного завдання.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів;
- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

### 5.2.3 Результати випробувань

В результаті тестування була перевірена система підтримки процесу управління створення медіа матеріалу на відповідність функціональним вимогам. У наступних таблицях наведений перелік випробувань основних функціональних можливостей.

Таблиця 5.1 – Авторизація користувача у системі

Мета тесту	Перевірка функції «авторизація»
Початковий стан моделі	На екрані відкрита форма авторизації
Вхідні дані:	Логін, пароль
Схема проведення тесту:	Ввести вхідні дані користувача виконавця, і керівника проекту
Очікуваний результат:	Перехід на відповідну сторінку
Стан моделі після випробування:	Після вводу вхідних даних виконавця, відкрилась відповідна сторінка, так само опинилось і з даими керівника проекту.

Змн.	Арк.	№ докум.	Підпис	Дата



Таблиця 5.2 – Можливість керівника проекту редагувати, створювати, та видаляти проекти

<b>Мета тесту</b>	<b>Перевірка функції «робота з проектами»</b>
Початковий стан моделі	Відкрита головна сторінка керівника проекту
Вхідні дані:	Назва проекту
Схема проведення тесту:	Створити новий проект, відредагувати дані у ньому, і видалити
Очікуваний результат:	Створиться новий проект, зміняться дані, проект буде видалено
Стан моделі після проведення випробувань:	Після співпадіння з очікуваними результатами, система повернулася до початкового стану

Таблиця 5.3 – Перевірка можливості керівника проекту, роздавати завдання виконавцям за власним розсудом

<b>Мета тесту</b>	<b>Перевірка можливості розподілу завдань за власним розсудом</b>
Початковий стан моделі	Відкритий звіт, що згенерувала система
Вхідні дані:	Змінений необхідним чином звіт
Схема проведення тесту:	Змінити розподіл завдань у звіті і роздати його виконавцям
Очікуваний результат:	Виконавці отримали ті роботи, які обрав їм керівник проекту
Стан моделі після проведення випробувань:	Зберігся змінений звіт, виконавці отримали відповідні завдання

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 5.4 – Перевірка необхідної кількості робітників

Мета тесту	Перевірка можливості роботи системи без виконавців, та без керівників
Початковий стан моделі	Відкрита форма авторизації, у базі даних виконавців немає нікого
Вхідні дані:	Дані для генерації звіту
Схема проведення тесту:	Створити звіт без виконавців
Очікуваний результат:	Оскільки у базі немає жодного робітника, авторизація у систему неможлива
Стан моделі після проведення випробувань:	Вхід у систему неможливий

Таблиця 5.5 – Другий тест перевірки необхідної кількості робітників

Мета тесту	Перевірка можливості роботи системи без виконавців
Початковий стан моделі	Відкрита форма авторизації, у базі даних виконавців є лише керівник проекту
Вхідні дані:	Дані для генерації звіту
Схема проведення тесту:	Створити звіт без виконавців
Очікуваний результат:	Оскільки у базі присутня лише одна людина, то і вся робота має бути записана керівнику проекту
Стан моделі після проведення випробувань:	Згенеровано звіт, в якому всю роботу має виконати одна людина

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 5.6 – Перевірка можливості додавання в базу даних будь-якої кількості робітників

Мета тесту	Перевірка бази даних виконавців
Початковий стан моделі	У базі даних 100 робітників
Вхідні дані:	Дані ще 9999 робітників
Схема проведення тесту:	Додати 9999 робітників у бд
Очікуваний результат:	У базі даних додано 9999 робітників, для генерації звіту необхідно багато часу
Стан моделі після проведення випробувань:	У бд виконавців присутні 10099 виконавців, звіт генерується за 9 годин, при 5 роботах

Таблиця 5.7 – Перевірка можливості додавання в базу даних будь-якої кількості завдань

Мета тесту	Перевірка бази даних завдань
Початковий стан моделі	У базі даних 100 завдань
Вхідні дані:	Дані ще 9999 завдань
Схема проведення тесту:	Додати 9999 завдань у бд
Очікуваний результат:	У базі даних додано 9999 завдань
Стан моделі після проведення випробувань:	У бд виконавців присутні 10099 завдань, звіт генерується як завжди, адже більшість з них не використовуються

Змн.	Арк.	№ докум.	Підпис	Дата

**Висновок до розділу**

В даному розділі було описано функціональні можливості та наведено результати роботи системи. Були проведені групи тестів, що дозволяють користувачу переконатись у коректності роботи програмного продукту, з детальним поясненням до них. Було сформовано інструкцію користувача з приводу використання системи підтримки процесу управління створення медіа матеріалу, було проведено повний перелік функцій, які присутні в даній системі.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

## ЗАГАЛЬНІ ВИСНОВКИ

У даній роботі була проведена робота над проектом з темою: «Система підтримки процесу управління створенням медіа матеріалу».

У розділі "Загальні положення" описані ціль, мета, і задачі розробки, усі з яких були успішно реалізовані. За допомогою необхідних діаграм наглядно продемонстровано процес діяльності кожного користувача системи. Також були продемонстровані наявні аналоги системи, відмінності були показані у вигляді таблиць.

Вхідні та вихідні дані були описані у розділі з інформаційного забезпечення, продемонстровані звіти, що буде отримувати користувач під час роботи над проектом, спроектовані бази даних.

У розділі з математичного забезпечення описано змістовну і математичну постановку задачі, був здійснений порівняльний аналіз методів її вирішення, обґрунтовано вибір методу вирішення, наведено аналоги вирішення задачі, та результати.

Четвертий розділ містить опис вибору засобів, що були використані при розробці системи.

В останньому розділі представлена інструкція користувача і тести, що використовувались при тестуванні системи.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

## ПЕРЕЛІК ПОСИЛАНЬ

1. Things[Електронний ресурс] : [culturedcode.com/things/main](http://culturedcode.com/things/main)
2. ЛидерТаск[Електронний ресурс] : [leadertask.com/leadertask/main](http://leadertask.com/leadertask/main)
3. Todoist[Електронний ресурс] : [todoist.com/main/info](http://todoist.com/main/info)
4. Гуляницький Л.Ф. Методи комбінаторної оптимізації: теоретичні відомості./ Гуляницький Л.Ф., Мулеса О.Ю. – Ужгород – 2015. – с.25.
5. Недобачій С.І. Гвоздик Д.М. Новий метод розв'язання задачі про призначення. – 2009. – с.5.
6. Вагнер Г. Основи дослідження операцій./ Г.Вагнер – Ленінград – 1973. – с.504.
7. Косоруков О.А. Дослідження операцій./ О.А. Косоруков – Чернігів – 2003. – с.448.
8. Зайченко Ю.П. Дослідження операцій/ Ю.П. Зайченко – Київ – 2003. – с.688.

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Додаток А

**Тексти програмного коду**

Система підтримки процесу управління створенням медіа  
матеріалу

---

(Найменування програми (документа))

*DVD-R*

---

(Вид носія даних)

*10 аркушів, 12 мб*

---

(Обсяг програми (документа) , арк..)

Київ – 2019 року

					ДП ІС-5201.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

## 1. Угорський алгоритм

```

class Ugorsky
{
    public static int index = 0;
    public static int[] KuhnMunkres(int[,] a)
    {
        int N = a.GetLength(0);
        try
        {
            if (N == 0)
                return new int[0];

            int[] lx = new int[N], ly = new int[N];
            int[] mx = new int[N], my = new int[N];
            int[] px = new int[N], py = new int[N];
            int[] stack = new int[N];

            for (int i = 0; i < N; i++)
            {
                lx[i] = a[i, 0];
                for (int j = 0; j < N; j++)
                    if (a[i, j] > lx[i]) lx[i] = a[i, j];
                ly[i] = 0;

                mx[i] = my[i] = -1;
            }

            for (int size = 0; size < N;)
            {
                int s;
                for (s = 0; mx[s] != -1; s++) ;

                for (int i = 0; i < N; i++)
                    px[i] = py[i] = -1;
                px[s] = s;

                int t = -1;
                stack[0] = s;
                for (int top = 1; top > 0;)
                {
                    int u = stack[--top];
                    for (int v = 0; v < N; v++)
                    {
                        if (lx[u] + ly[v] == a[u, v] && py[v] == -1)
                        {
                            if (my[v] == -1)
                            {
                                t = v;
                                py[t] = u;
                                top = 0;
                                break;
                            }

                            py[v] = u;
                            px[my[v]] = v;
                            stack[top++] = my[v];
                        }
                    }
                }
            }
        }
    }
}

```



```

        if (t != -1)
        {
            while (true)
            {
                int u = py[t];
                mx[u] = t;
                my[t] = u;
                if (u == s) break;
                t = px[u];
            }
            ++size;
        }
        else
        {
            int delta = int.MaxValue;
            for (int u = 0; u < N; u++)
            {
                if (px[u] == -1) continue;
                for (int v = 0; v < N; v++)
                {
                    if (py[v] != -1) continue;
                    int z = lx[u] + ly[v] - a[u, v];
                    if (z < delta)
                        delta = z;
                }
            }

            for (int i = 0; i < N; i++)
            {
                if (px[i] != -1) lx[i] -= delta;
                if (py[i] != -1) ly[i] += delta;
            }
        }
    }

    bool correct = true;
    for (int u = 0; u < N; u++)
    {
        for (int v = 0; v < N; v++)
        {
            correct &= (lx[u] + ly[v] >= a[u, v]);
            if (mx[u] == v)
                correct &= (lx[u] + ly[v] == a[u, v]);

            if (!correct)
            {
                throw new Exception();
            }
        }
    }

    for (int i = 0; i < mx.Length; i++)
    {
        mx[i] += 1;
    }
    return mx;
}
catch (Exception eee)
{
    UgorskyMethod.SelfRef.error();
    return null;
}
}

```

}

## 2. Метод Мака

```

using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AkimovDmytro_IS_52_Diploma
{
    public partial class Mak : Form
    {
        int[,] a;
        string file;
        public Mak()
        {
            InitializeComponent();
            SelfRef = this;
        }
        public static Mak SelfRef
        {
            get;
            set;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            size AI = new size();
            AI.Show();
        }

        private void Mak_Load(object sender, EventArgs e)
        {
        }

        private void button3_Click(object sender, EventArgs e)
        {
            richTextBox1.Clear();
            if (openFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)//путь к файлу
            {
                file = openFileDialog1.FileName;
            }
            try
            {
                string[] lines =
File.ReadAllLines(file).Take(1000).ToArray();//считываем с файла
                string[] textMass;
                textMass = lines[0].Split(' ');
                MakMeth.index = textMass.Length;
                a = new int[MakMeth.index, MakMeth.index];
                for (int i = 0; i < MakMeth.index; i++)
                {
                    int[] row = lines[i].Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries).Select(Int32.Parse).ToArray();//записываем в массив
                    for (int j = 0; j < MakMeth.index; j++)
                    {
                        a[i, j] = row[j];
                    }
                }
            }
            for (int i = 0; i < MakMeth.index; i++)//выводим массив

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

        {
            for (int j = 0; j < MakMeth.index; j++)
            {
                richTextBox1.Text += a[i, j] + "\t";
            }
            richTextBox1.Text += "\n";
        }
    }
    catch (Exception eee)
    {
    }
}

private void button2_Click(object sender, EventArgs e)
{
    if (MakMeth.index != 0)
    {
        try
        {
            int sum = 0;
            int[] b = new int[MakMeth.index];
            Array.Copy(MakMeth.AkimovMakAlgo(a), b,
MakMeth.AkimovMakAlgo(a).Length);
            for (int i = 0; i < b.Length; i++)
            {
                richTextBox1.Text += ("\nВ " + (i + 1) + " рядку вибираємо " +
b[i] + " елемент.\n");
            }
            for (int i = 0; i < MakMeth.index; i++)
            {
                sum += (a[i, b[i] - 1]);
            }
            richTextBox1.Text += ("\nРезультат максимізації: " + sum + "\n");
        }
        catch (Exception eee)
        {
        }
    }
    else
    {
        MessageBox.Show("Введіть або згенеруйте матрицю");
    }
}

private void chart1_Click(object sender, EventArgs e)
{
    for (double x = 0; x < 100.0; x += 0.01)
    {
    }
}

private void button4_Click(object sender, EventArgs e)
{
    Graf AI = new Graf();
    AI.Show();
}

private void button5_Click(object sender, EventArgs e)
{
    UgorskyMethod AI = new UgorskyMethod();
    AI.Show();
    this.Hide();
}

```

```

    }

    public void random()
    {
        SelfRef = this;
        richTextBox1.Clear();
        Random rand = new Random();
        a = new int[MakMeth.index, MakMeth.index];
        for (int i = 0; i < MakMeth.index; i++)
        {
            for (int j = 0; j < MakMeth.index; j++)
            {
                a[i, j] = rand.Next(0, 1000);
            }
        }
        for (int i = 0; i < MakMeth.index; i++)//ВЫВОДИМ МАССИВ
        {
            for (int j = 0; j < MakMeth.index; j++)
            {
                richTextBox1.Text += a[i, j] + "\t";
            }
            richTextBox1.Text += "\n";
        }
    }

    private void Mak_FormClosing(object sender, FormClosingEventArgs e)
    {
        Application.Exit();
    }
}
}

```

### 3. Клас, що відповідає за роботу методу Мака

```

class MakMeth
{
    public static int index = 0;
    public static int[] AkimovMakAlgo(int[,] a)
    {
        int N = a.GetLength(0);
        if (N == 0)
            return new int[0];

        int[] lx = new int[N], ly = new int[N];
        int[] mx = new int[N], my = new int[N];
        int[] px = new int[N], py = new int[N];
        int[] stack = new int[N];

        for (int i = 0; i < N; i++)
        {
            lx[i] = a[i, 0];
            for (int j = 0; j < N; j++)
                if (a[i, j] > lx[i]) lx[i] = a[i, j];
            ly[i] = 0;

            mx[i] = my[i] = -1;
        }

        for (int size = 0; size < N; )
        {
            int s;

```

```

for (s = 0; mx[s] != -1; s++) ;

for (int i = 0; i < N; i++)
    px[i] = py[i] = -1;
px[s] = s;

int t = -1;
stack[0] = s;
for (int top = 1; top > 0; )
{
    int u = stack[--top];
    for (int v = 0; v < N; v++)
    {
        if (lx[u] + ly[v] == a[u, v] && py[v] == -1)
        {
            if (my[v] == -1)
            {
                t = v;
                py[t] = u;
                top = 0;
                break;
            }

            py[v] = u;
            px[my[v]] = v;
            stack[top++] = my[v];
        }
    }
}

```

#### 4. Побудова графіка

```

public partial class Grapf : Form
{
    public Grapf()
    {
        InitializeComponent();
        pane.Show();
    }
    int work = 0;
    int[,] a;
    int x_min;
    int x_max;

    private void button3_Click(object sender, EventArgs e)
    {

        pane.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        pane.Series[1].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;

        try
        {

            x_min = Int32.Parse(textBox1.Text);
            x_max = Int32.Parse(textBox2.Text);
            if (100 <= x_min && x_min < x_max && x_max <= 1000)
            {
                if (work == 1)
                {
                    pane.Series[0].Points.Clear();
                    pane.Series[1].Points.Clear();
                }
            }
        }
    }
}

```

```

    }

    int x = 0;
    for(int z= x_min; z<= x_max; z+=1)
    {
        Ugorsky.index = z;
        Random rand = new Random();
        a = new int[Ugorsky.index, Ugorsky.index];
        for (int i = 0; i < Ugorsky.index; i++)
        {
            for (int j = 0; j < Ugorsky.index; j++)
            {
                a[i, j] = rand.Next(0, 100);
            }
        }
        try
        {
            Stopwatch sw = new Stopwatch();
            sw.Start();
            Ugorsky.KuhnMunkres(a);
            sw.Stop();
            Random rand1 = new Random();
            pane.Series[0].Points.AddXY(x, sw.ElapsedMilliseconds);
            pane.Series[0].Color = Color.Red;
            if(x<150)

                x++;
        }
        catch (Exception eee)
        {
        }
    }
    work = 1;
}
else MessageBox.Show("Введіть правильні числа!");
}
catch (Exception eee)
{
    MessageBox.Show("Введіть правильні числа!");
}
}

private void Grapf_FormClosing(object sender, FormClosingEventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}

private void Grapf_Load(object sender, EventArgs e)
{
}
}
}

```

Змн.	Арк.	№ докум.	Підпис	Дата

## 5.Головна форма

```

public partial class LeaderMain : Form
{
    ProjectNew pjn = new ProjectNew();
    WorkerDB WDBs = new WorkerDB();
    public LeaderMain()
    {
        InitializeComponent();
    }

    private void Button3_Click(object sender, EventArgs e)
    {
        WDBs.Show();
        this.Hide();
    }

    private void Timer1_Tick(object sender, EventArgs e)
    {
        label1.Text = DateTime.Now.ToString("T");
    }

    private void LeaderMain_Load(object sender, EventArgs e)
    {
        timer1.Start();
    }

    private void MonthCalendar1_DateChanged(object sender, DateRangeEventArgs e)
    {
        monthCalendar1.AddBoldedDate(new DateTime(2019, 05, 22));
    }

    private void Button1_Click(object sender, EventArgs e)
    {
        pjn.Show();

        this.Hide();
    }

    private void Button2_Click(object sender, EventArgs e)
    {
        string add = textBox1.Text;
        listBox1.Items.Add(textBox1.Text);
    }

    private void Button5_Click(object sender, EventArgs e)
    {
        listBox1.Items.RemoveAt(listBox1.SelectedIndex);
    }

    private void Label1_Click(object sender, EventArgs e)
    {
    }

    private void TextBox1_TextChanged(object sender, EventArgs e)
    {
    }

    TaskDB TDBs = new TaskDB();
    private void Button4_Click(object sender, EventArgs e)
    {
        TDBs.Show();
        this.Hide();
    }
}

```

Змн.	Арк.	№ докум.	Підпис	Дата

## 6. Вивід інформації з бази даних на форму

```

private async void ProdusserMain_Load(object sender, EventArgs e)
{
    string connectionString = @"Data Source=MITYA-ПК\SQLEXPRESS;Initial
Catalog=IS5201_Diploma_DB;Integrated Security=True";
    sqlConnection = new SqlConnection(connectionString);
    await sqlConnection.OpenAsync();
    SqlDataReader sqlReader = null;
    SqlCommand command = new SqlCommand("SELECT * FROM [Workers]",
sqlConnection);

    try
    {
        sqlReader = await command.ExecuteReaderAsync();
        while (await sqlReader.ReadAsync())
        {
            listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " +
Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["Surname"]) + "
" + Convert.ToString(sqlReader["Specialisation"]) + " " +
Convert.ToString(sqlReader["Salary"]));
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(),
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (sqlReader != null)
            sqlReader.Close();
    }
}

private async void Button1_Click(object sender, EventArgs e)
{
    SqlCommand command = new SqlCommand("INSERT INTO [Workers] (Name, Surname,
Specialisation, Leader, skill_photo, skill_video, skill_soundmontage, skill_videomontage,
skill_animation, skill_graphics, salary)VALUES(@Name, @Surname, @Specialisation, @Leader,
@skill_photo, @skill_video, @skill_soundmontage, @skill_videomontage, @skill_animation,
@skill_graphics, @salary)", sqlConnection);
    command.Parameters.AddWithValue("Name", textBox1.Text);
    command.Parameters.AddWithValue("Surname", textBox2.Text);
    command.Parameters.AddWithValue("Specialisation", textBox3.Text);
    command.Parameters.AddWithValue("Leader", textBox4.Text);
    command.Parameters.AddWithValue("skill_photo", textBox5.Text);
    command.Parameters.AddWithValue("skill_video", textBox6.Text);
    command.Parameters.AddWithValue("skill_soundmontage", textBox7.Text);
    command.Parameters.AddWithValue("skill_videomontage", textBox8.Text);
    command.Parameters.AddWithValue("skill_animation", textBox9.Text);
    command.Parameters.AddWithValue("skill_graphics", textBox10.Text);
    command.Parameters.AddWithValue("salary", textBox11.Text);

    await command.ExecuteNonQueryAsync();
}

```





НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_ О.С.Жураковська  
(підпис) (ініціали, прізвище)

“13” квітня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.О. завідувача кафедри**

\_\_\_\_\_ О.А.Павлов  
(підпис) (ініціали, прізвище)

“14” квітня 2019 р.

«Система підтримки процесу управління створенням медіа матеріалу»

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр *ДП ІС-5201.1181-с.ТЗ*

На 9 сторінках

Київ – 2019 року

# ЗМІСТ

<b>1</b>	<b>ЗАГАЛЬНІ ПОЛОЖЕННЯ.....</b>	<b>3</b>
1.1	Повне найменування системи та її умовне позначення .....	3
1.2	Найменування організації-замовника та організацій-учасників робіт.....	3
1.3	Перелік документів, на підставі яких створюється система.....	3
1.4	Планові терміни початку і закінчення роботи зі створення системи ..	4
<b>2</b>	<b>ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ .....</b>	<b>4</b>
2.1	Призначення розробки.....	4
2.2	Цілі створення застосування .....	4
<b>3</b>	<b>ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....</b>	<b>5</b>
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>6</b>
4.1	Вимоги до функціональних характеристик.....	6
4.2	Вимоги до надійності .....	6
4.4	Вимоги до складу і параметрів технічних засобів .....	7
<b>5</b>	<b>СТАДІЇ І ЕТАПИ РОЗРОБКИ .....</b>	<b>8</b>
<b>6</b>	<b>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....</b>	<b>8</b>
6.1	Види випробувань .....	8

					<b>ДП ІС-5201.1181-с.ТЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>	<i>Система підтримки процесу управління створенням медіа матеріалу</i>	<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>
<i>Розроб.</i>		<i>Акімов Д.Д.</i>						
<i>Перевірив.</i>		<i>Жураковська О.С.</i>					2	9
<i>Н. кон.</i>		<i>Халус О.А.</i>				<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>		
<i>Затв.</i>		<i>Павлов О.А.</i>						

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Повне найменування системи та її умовне позначення

**Повна назва системи:** Система підтримки процесу управління створення медіа матеріалів.

**Коротке найменування системи:** «MediaHelper».

### 1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником є кафедра автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" (далі за текстом — Замовник). Адреса замовника: м. Київ, п. Перемоги 37, 18 корпус ФІОТ.

Розробник сервісу — студент групи ІС-52 кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" Акімов Дмитро Дмитрович.

### 1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації

Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи;

					ДП ІС-5201.1181-с.ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

#### **1.4 Планові терміни початку і закінчення роботи зі створення системи**

Плановий строк початку роботи по розробці «MediaHelperβ» — 15 квітня 2019 року.

Плановий строк закінчення роботи по розробці «MediaHelperβ» — 28 травня 2019 року.

## **2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ**

### **2.1 Призначення розробки**

Призначенням системи є підтримка процесу створення медіа контенту.

### **2.2 Цілі створення системи**

Цілями розробки системи є:

- структуризація завдань, що необхідні для створення медіа матеріалу;
- оптимізація оптимальної роботи кіно команди, за певним критерієм;
- підбір оптимальної команди для вирішення обраних задач.

Для досягнення поставлених цілей необхідно реалізувати наступні задачі.

Задачами розробки є:

- створення бази даних робітників;
- створення бази даних задач;
- розробка математичної програми для оптимізації;
- можливість виконувати звіти і ставити задачі.

					<b>ДП ІС-5201.1181-с.ТЗ</b>	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Для використання програмного забезпечення, керівник проекту повинен зареєструватись, та авторизуватись в системі, та потім зареєструвати робітників, що не знаходяться в базі даних.

Об'єктом автоматизації є процес призначення виконавців на відповідні роботи.

					ДП ІС-5201.1181-с.ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Застосунок має допомагати керівнику проекту оптимально розподіляти робітників на роботи, та допомагати у комунікації робітника і керівника.

Застосунок має виконувати наступні функції:

- реалізувати реєстрацію та авторизацію користувачів;
- створення списку робіт, необхідних для завершення проекту;
- формування звіту керівнику проекту;
- створення можливостей виконавцю відмітити виконані завдання
- надання можливостей комунікації виконавців і керівників проекту.

### 4.2 Вимоги до надійності

Система повинна функціонувати, навіть при виявлених не доопрацюваннях у функціоналі, або поломок, що виникли під час експлуатації системи. У випадку поломки програмного забезпечення, усі данні і звіти повинні зберегтись.

					ДП ІС-5201.1181-с.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4.4 Вимоги до складу і параметрів технічних засобів

Для коректної роботи даного продукту на стороні користувача мають бути такі технічні характеристики:

- процесор з тактовою частотою не нижче 2.0 ГГц;
- оперативна пам'ять об'ємом не менше 1 ГБ;
- мережева карта;
- жорсткий диск чи SSD з об'ємом пам'яті не менше 5Гб;
- операційна система Windows 7, або новіші;
- Microsoft Word 2007, або новіші;
- монітор;
- мишка;
- клавіатура.

					ДП ІС-5201.1181-с.ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки системи процесу управління створення медіа матеріалу.

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	16.04.19	
2.	Розробка сценарію роботи	17.04.19	
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	20.04.19	
4.	Розробка інформаційного забезпечення	22.04.19	
5.	Розробка програмного забезпечення	13.05.19	
6.	Налагодження програми	24.05.19	
7.	Тестування програми	26.05.19	
8.	Здача готового програмного продукту замовнику	28.05.19	

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

### 6.1 Види випробувань

Для контролю правильності роботи програмного забезпечення буде проведено функціональне тестування. В ході тестування буде проведено випробування основних функціональних характеристик програми.

					ДП ІС-5201.1181-с.ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Тестування реєстрації та авторизації полягає у введенні вірних або невірних даних.

Тестування додавання нових робітників і завдань полягає у перевірці їх наявності у застосунку.

Тестування комунікації у вигляді перевірки функціонування кнопки виконане завдання.

Тестування звіту полягає у ручній перевірці математичної складової системи, та співпадіння з наданим звітом.

					ДП ІС-5201.1181-с.ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_  
(підпис) О.С.Жураковська  
(ініціали, прізвище)

“16” травня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.О. Завідувача кафедри**

\_\_\_\_\_  
(підпис) О.А. Павлов  
(ініціали, прізвище)

“17” травня 2019 р.

«Система підтримки процесу управління створенням медіа  
матеріалу»

**ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ**

Шифр ДП ІС-5201.1181-с.ПМВ

На 12 сторінках

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>ОБ'ЄКТ ВИПРОБУВАННЯ.....</b>	<b>3</b>
1.1	Найменування програми .....	3
1.2	Область застосування .....	4
1.3	Умовне позначення програми.....	4
<b>2</b>	<b>МЕТА ВИПРОБУВАНЬ .....</b>	<b>5</b>
<b>3</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>6</b>
3.1	Вимоги до функціональних характеристик .....	6
3.1.1	<i>Вимоги до складу виконуваних функцій</i> .....	6
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....</b>	<b>7</b>
<b>5</b>	<b>СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ.....</b>	<b>8</b>
<b>6</b>	<b>МЕТОДИ ВИПРОБУВАНЬ .....</b>	<b>9</b>

					<b>ДП ІС-5201.1181-с.ПМВ</b>					
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>	<i>Система підтримки процесу управління створенням медіа матеріалу</i>					
<i>Розроб.</i>		<i>Акімов Д.Д.</i>						<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>
									2	12
<i>Перевірів.</i>		<i>Жураковська О.С.</i>						<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>		
<i>Н. кон.</i>		<i>Халус О.А.</i>								
<i>Затв.</i>		<i>Павлов О.А.</i>								

## 1 ОБ'ЄКТ ВИПРОБУВАННЯ

### 1.1 Найменування програми

Темою дипломного проекту є «Система підтримки процесу управління створенням медіа матеріалу» із назвою програмного продукту – «MediaHelper».

### 1.2 Область застосування

Областю застосування програми є кінокомпанії будь-якого масштабу, де для створення медіа матеріалу необхідно вирішити багато задач.

Система допомагає керівнику медіа проекту призначити виконавців на роботи, розподіливши завдання, що їм потрібно виконати, за певним критерієм.

### 1.3 Умовне позначення програми

Умовне позначення програми – система «MediaHelper».

					ДП ІС-5201.1181-с.ПМВ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

**2 МЕТА ВИПРОБУВАНЬ**

Мета проведення випробувань – перевірка відповідних характеристик розробленої системи функціональним і окремим іншим видам вимог, викладених в документі технічного завдання.

					ДП ІС-5201.1181-с.ПМВ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Вимоги до функціональних характеристик

Програмне застосування повинно:

- користувач має можливість авторизуватись в системі;
- продюсер має можливість створювати, редагувати, та видаляти проекти безліч разів;
- усі користувачі-керівники можуть відправити завдання будь-якому користувачу-виконавцю;
- кожен створений проект повинен мати продюсера. Інших акторів може бути безліч, або не бути взагалі;
- усі користувачі-виконавці мають можливість відправити звіт користувачу-керівнику, що дав завдання;
- продюсер та режисер можуть додавати у проект скільки завгодно нових користувачів;
- усі завдання повинні бути записані у базу даних;
- усі користувачі повинні бути записані у базі даних.

##### 3.1.1 Вимоги до складу не функціональних функцій

Програмне застосування повинно:

- обсяг пам'яті виділений на систему - не більше 5Гб.;
- інтерфейс програми має не бути різнокольоровим. Максимально допустима кількість кольорів на сторінці(не включаючи фотографії, або картинки) - 5 кольорів;
- шрифт у програмі повинен не бути менше 10п'єків, але і не більше 20.

					ДП ІС-5201.1181-с.ПМВ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмний продукти розробляється на основі Технічного Завдання.

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

					ДП ІС-5201.1181-с.ПМВ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		



## 5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

**Умови проведення випробувань:** програма з готовими модулями та зв'язками між ними.

**Умови початку та завершення окремих етапів тестування:** тестування кожних елементів програми має відбуватися з урахуванням всіх можливих виключних ситуацій в залежності від функціональних можливостей програмного продукту.

**Обмеження щодо умов проведення тестування:** тестування має проводитися в рамках функціонального апарату системи.

**Вимоги до технічного обслуговування системи:** система має бути застосунком на операційній системі Windows.

**Міри, забезпечуючі безпеку та безаварійність проведення тестування:** тестування системи не може визвати аварійних ситуацій.

**Порядок взаємодій організацій, які беруть участь у тестуванні:** тестування проводить один студент КІІ групи ІС-52 Акімов Дмитро Дмитрович.

					ДП ІС-5201.1181-с.ПМВ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

**6 МЕТОДИ ВИПРОБУВАНЬ**

У наступних таблицях наведений перелік випробувань основних функціональних можливостей.

Таблиця 6.1 – Авторизація користувача у системі

<b>Мета тесту</b>	<b>Перевірка функції «авторизація»</b>
Початковий стан моделі	На екрані відкрита форма авторизації
Вхідні дані:	Логін, пароль
Схема проведення тесту:	Ввести вхідні дані користувача виконавця, і керівника проекту
Очікуваний результат:	Перехід на відповідну сторінку
Стан моделі після випробування:	Після вводу вхідних даних виконавця, відкрилась відповідна сторінка, так само опинилось і з даими керівника проекту.

Таблиця 6.2 – Можливість керівника проекту редагувати, створювати, та видаляти проекти

Мета тесту	Перевірка функції «робота з проектами»
Початковий стан моделі	Відкрита головна сторінка керівника проекту
Вхідні дані:	Назва проекту
Схема проведення тесту:	Створити новий проект, відредагувати дані у ньому, і видалити
Очікуваний результат:	Створиться новий проект, зміняться дані, проект буде видалено
Стан моделі після проведення випробувань:	Після співпадіння з очікуваними результатами, система повернулась до початкового стану

Таблиця 6.3 – Перевірка можливості керівника проекту, роздавати завдання виконавцям за власним розсудом

<b>Мета тесту</b>	<b>Перевірка можливості розподілу завдань за власним розсудом</b>
Початковий стан моделі	Відкритий звіт, що згенерувала система
Вхідні дані:	Змінений необхідним чином звіт
Схема проведення тесту:	Змінити розподіл завдань у звіті і роздати його виконавцям
Очікуваний результат:	Виконавці отримали ті роботи, які обрав їм керівник проекту
Стан моделі після проведення випробувань:	Зберігся змінений звіт, виконавці отримали відповідні завдання

Таблиця 6.4 – Перевірка необхідної кількості робітників

<b>Мета тесту</b>	<b>Перевірка можливості роботи системи без виконавців, та без керівників</b>
Початковий стан моделі	Відкрита форма авторизації, у базі даних виконавців немає нікого
Вхідні дані:	Дані для генерації звіту
Схема проведення тесту:	Створити звіт без виконавців
Очікуваний результат:	Оскільки у базі немає жодного робітника, авторизація у систему неможлива
Стан моделі після проведення випробувань:	Вхід у систему неможливий

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 6.5 – Другий тест перевірки необхідної кількості робітників

<b>Мета тесту</b>	<b>Перевірка можливості роботи системи без виконавців</b>
Початковий стан моделі	Відкрита форма авторизації, у базі даних виконавців є лише керівник проекту
Вхідні дані:	Дані для генерації звіту
Схема проведення тесту:	Створити звіт без виконавців
Очікуваний результат:	Оскільки у базі присутня лише одна людина, то і вся робота має бути записана керівнику проекту
Стан моделі після проведення випробувань:	Згенеровано звіт, в якому всю роботу має виконати одна людина

Таблиця 6.6 – Перевірка можливості додавання в базу даних будь-якої кількості робітників

<b>Мета тесту</b>	<b>Перевірка бази даних виконавців</b>
Початковий стан моделі	У базі даних 100 робітників
Вхідні дані:	Дані ще 9999 робітників
Схема проведення тесту:	Додати 9999 робітників у бд
Очікуваний результат:	У базі даних додано 9999 робітників, для генерації звіту необхідно багато часу
Стан моделі після проведення випробувань:	У бд виконавців присутні 10099 виконавців, звіт генерується за 9 годин, при 5 роботах

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 6.7 – Перевірка можливості додавання в базу даних будь-якої кількості завдань

Мета тесту	Перевірка бази даних завдань
Початковий стан моделі	У базі даних 100 завдань
Вхідні дані:	Дані ще 9999 завдань
Схема проведення тесту:	Додати 9999 завдань у бд
Очікуваний результат:	У базі даних додано 9999 завдань
Стан моделі після проведення випробувань:	У бд виконавців присутні 10099 завдань, звіт генерується як завжди, адже більшість з них не використовуються

**Додаток А**

***Тексти програмного коду***

*Система підтримки процесу управління створення медіа матеріалів*

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*10 аркуша, 12 мб*

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019 року

Змн.	Арк.	№ докум.	Підпис	Дата

# **Графічний матеріал до дипломного проекту**

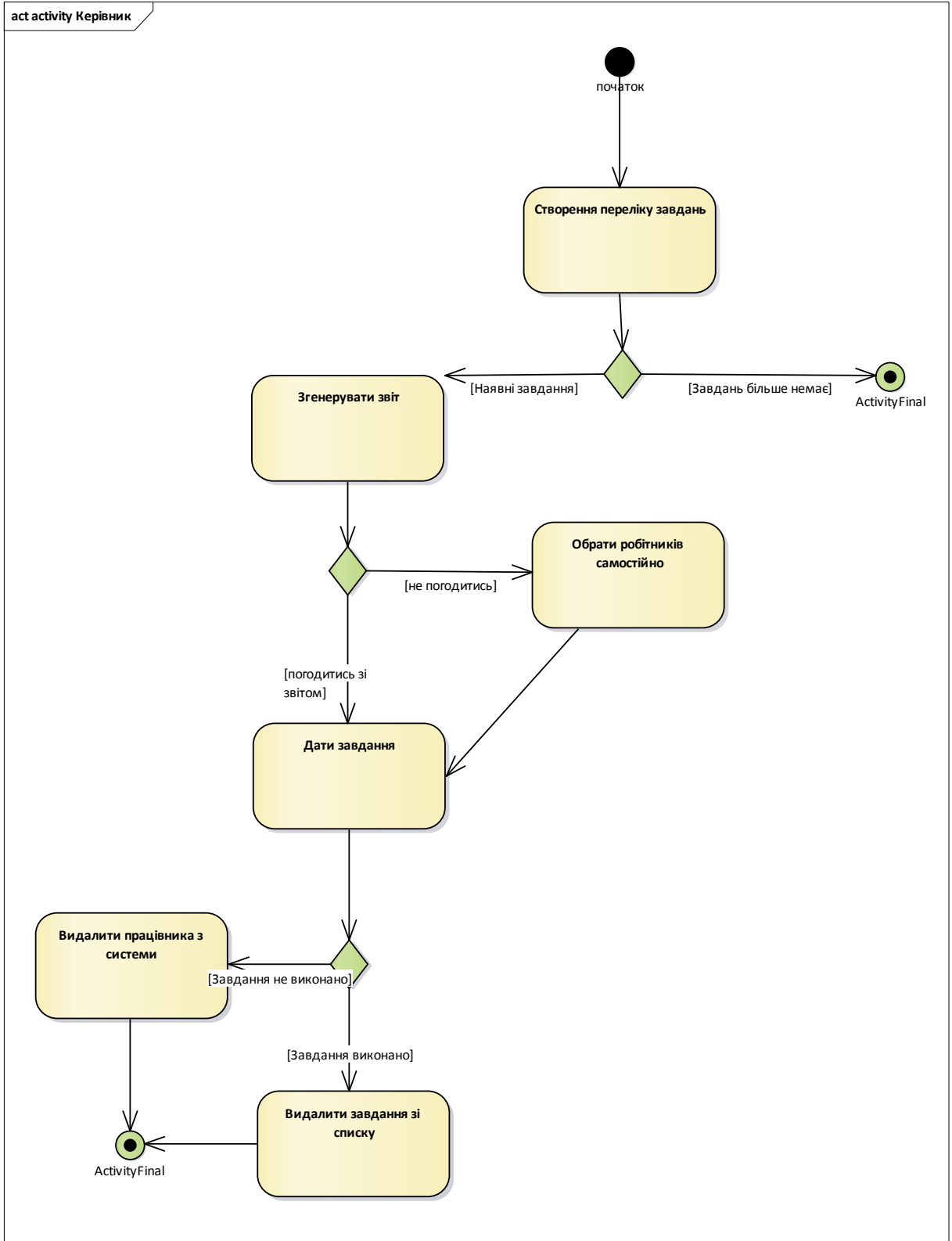
на тему: «Система підтримки процесу управління створенням медіа  
матеріалу»

---

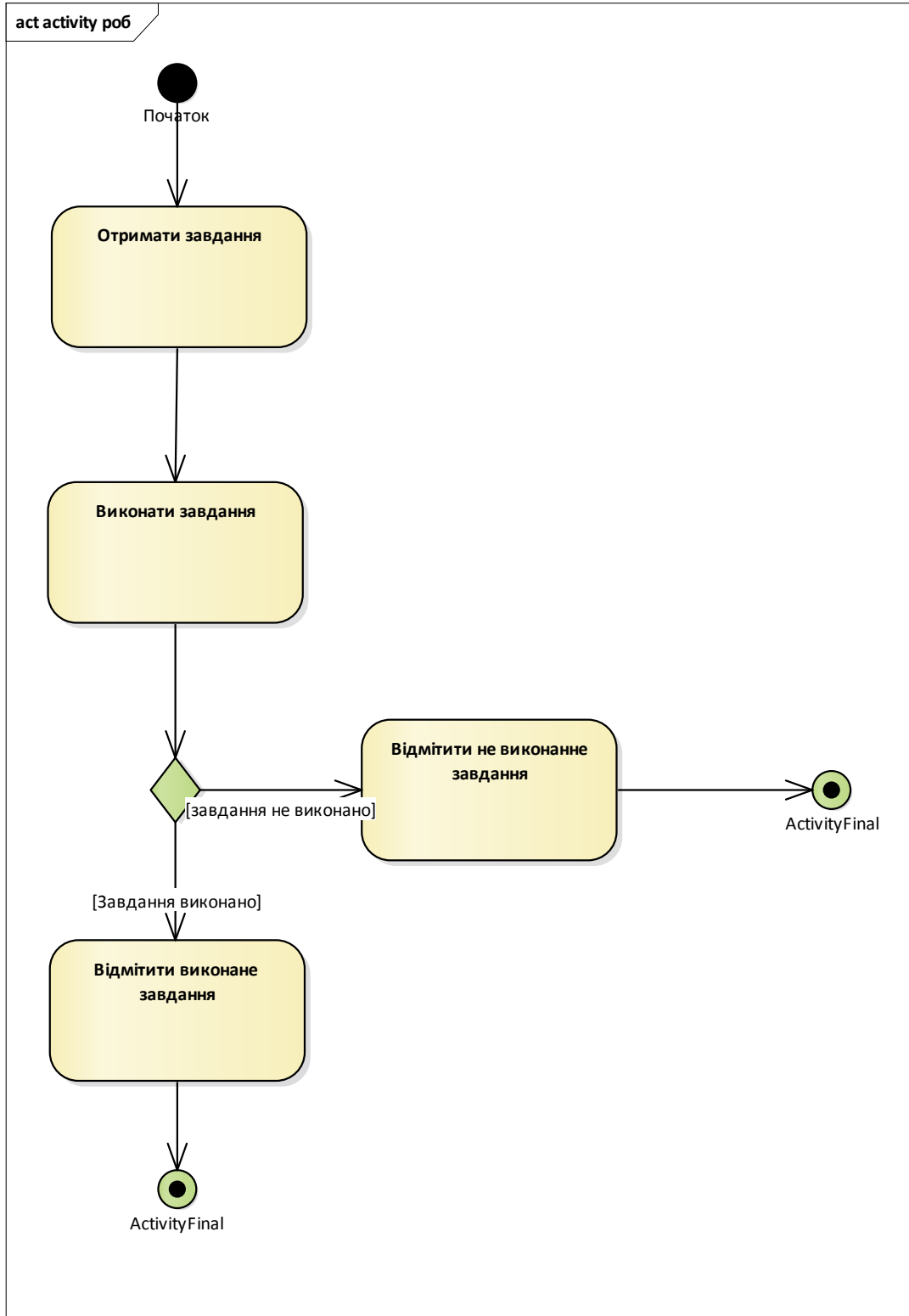
---

Київ – 2019 року

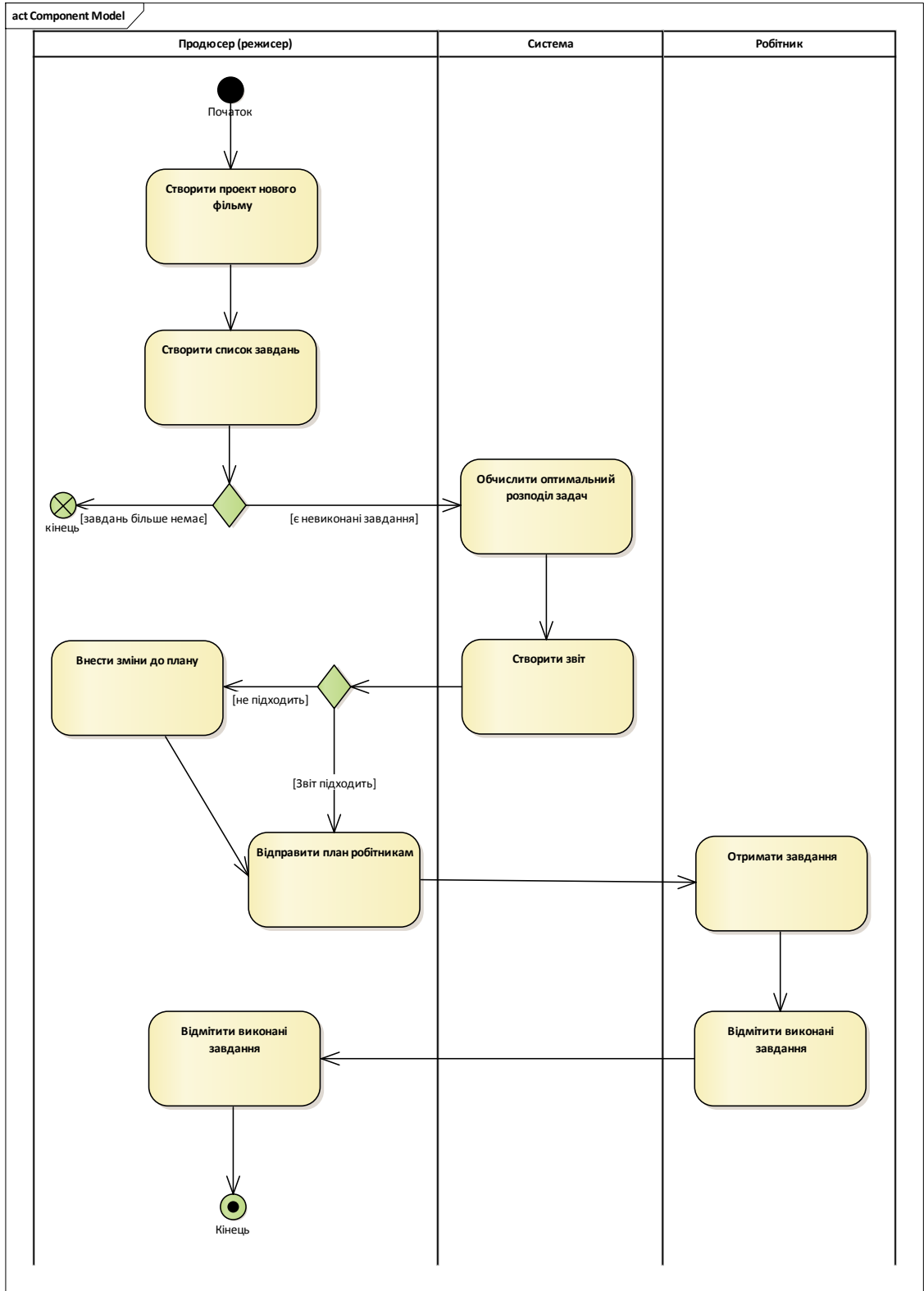




					<b>ДП ІС-5202.1181-с.ССД</b>			
						Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата	<i>Схема структурна діяльності</i>			
Розробив	Акімов Д.Д.							
Перевірив	Жураковська О.С.							
					Аркуш 1		Аркушів 6	
Т. кон.					<i>Система підтримки процесу управління створенням медіа матеріалу</i>			
Н. кон.	Халус О.А.							
Затвердив	Жураковська О.С.							
					<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>			



					<b>ДП ІС-5202.1181-с.ССД</b>			
						Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата	<i>Схема структурна діяльності</i>			
Розробив	Акімов Д.Д.							
Перевірив	Жураковська О.С.							
						Аркуш 2	Аркушів 6	
Т. кон.					<i>Система підтримки процесу управління створенням медіа матеріалу</i>	<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>		
Н. кон.	Халус О.А.							
Затвердив	Жураковська О.С.							



Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Акімов Д.Д.		
Перевірив		Жураковська О.С.		
Т. кон.				
Н. кон.		Халус О.А.		
Затвердив		Жураковська О.С.		

ДП ІС-5202.1181-с.ССД

Схема структурна діяльності

Система підтримки процесу управління створенням медіа матеріалу

Літера	Маса	Масштаб
Аркуш 3	Аркушів 6	

КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52

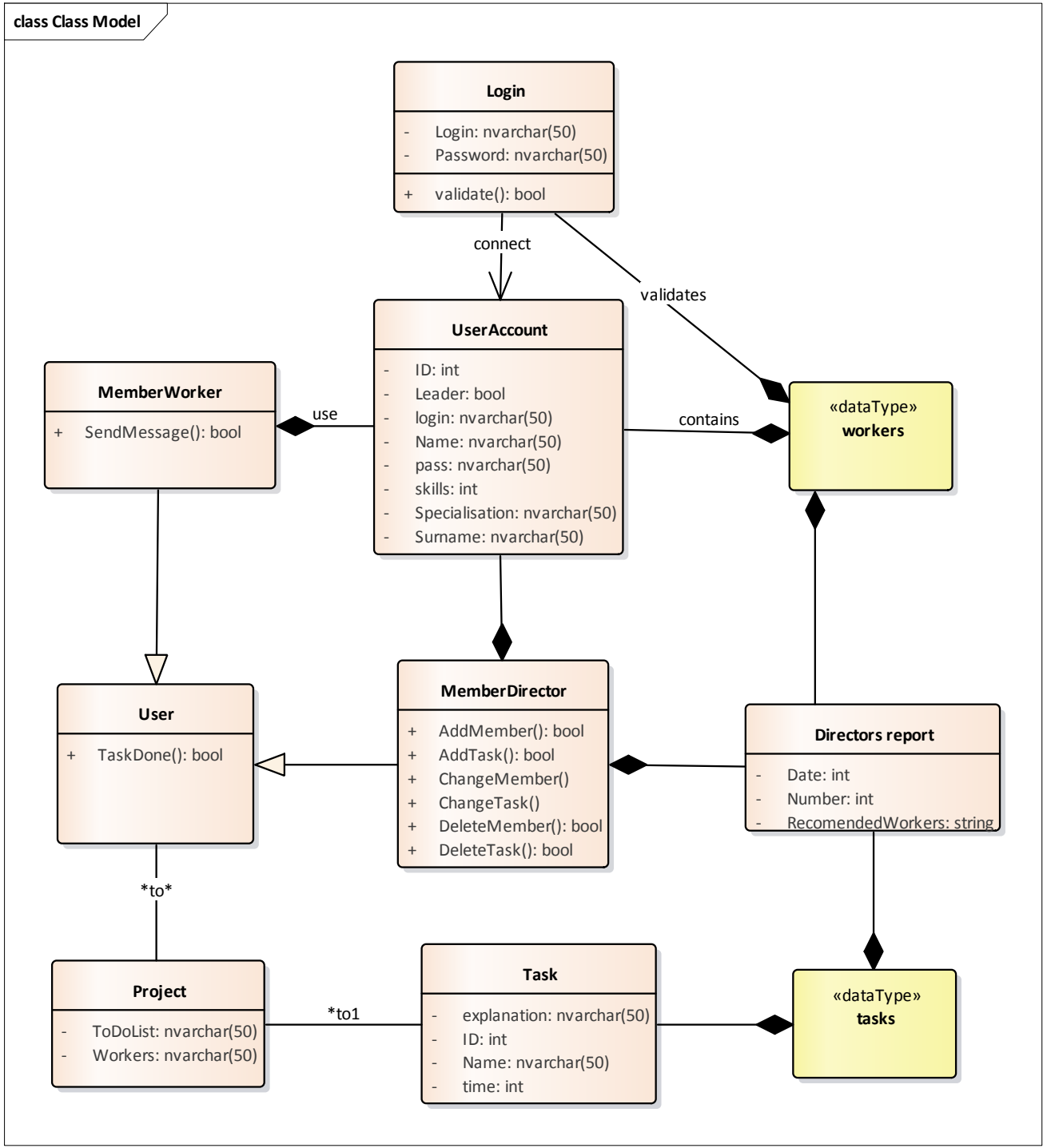
**Tasks**

🔑 Id
Name
Time
explanation

**Workers**

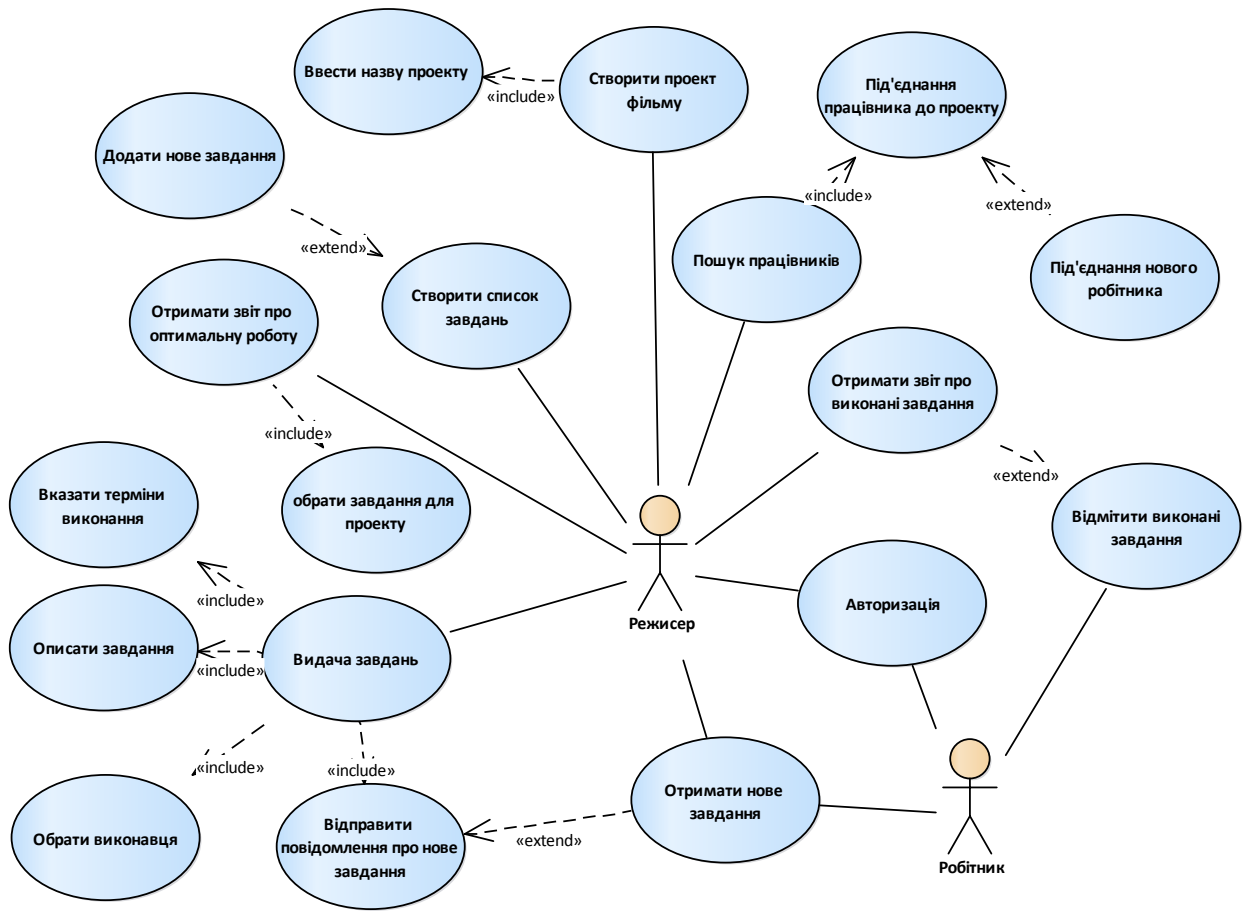
🔑 Id
Name
Surname
Specialisation
Leader
skill_photo
skill_video
skill_soundmontage
skill_videomontage
skill_animation
skill_graphics
salary
login
pass

					<b>ДП ІС-5202.1181-с.СБД</b>						
					<i>Схема бази даних</i>	Літера	Маса	Масштаб			
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Акімов Д.Д.									
Перевірив		Жураковська О.С.									
						Аркуш 1	Аркушів 6				
Т. кон.					<i>Система підтримки процесу управління створенням медіа матеріалу</i>	<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>					
Н. кон.		Халус О.А.									
Затвердив		Жураковська О.С.									



					ДП ІС-5202.1181-с.ССК		
Зм.	Арк.	№ документа	Підпис	Дата	Схеми структурна класів програмного забезпечення		
Розробив	Акімов Д.Д.						
Перевірив	Жураковська О.С.						
Т. кон.					Літера	Маса	Масштаб
Н. кон.	Халус О.А.						
Затвердив	Жураковська О.С.				Аркуш 1		Аркушів 6
					Система підтримки процесу управління створенням медіа матеріалу		
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52		

uc Use Case statya



					<b>ДП ІС-5202.1181-с.СВ</b>					
					<i>Схема структурна варіантів використання</i>			Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
					<i>Система підтримки процесу управління створенням медіа матеріалу</i>			Аркуш 1		Аркушів 6
Розробив		Акімов Д.Д.						<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>		
Перевірив		Жураковська О.С.								
Т. кон.										
Н. кон.		Халус О.А.								
Затвердив		Жураковська О.С.								