

Міністерство освіти і науки України  
Одеський національний політехнічний університет  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

## **КУРСОВА РОБОТА**

з дисципліни «Технології створення програмних продуктів»

за темою

«ICS – media – онлайн кінотеатр»

Частина Перша

Виконав:

студент 3-го курсу

групи АІ-185

Додон Д.І.

Перевірив:

Блажко О. А.

Одеса-2020

## Анотація

В курсовій роботі розглядається процес створення програмного продукту «ICS – media». В пояснювальній записці у розділах «Проектування» та «Конструювання» детально описано особливості конструювання:

- структур даних в системі керування базами даних MySQL;
- програмних модулів в інструментальному середовищі sublime з використанням бібліотеки redBeanPHP та мови програмування PHP.

Результати роботи розміщено на *github*-репозиторіях за адресою:

[Github-репозиторій](#)

## **Перелік скорочень**

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

ПП – програмний продукт

UML – уніфікована мова моделювання

## **Зміст**

<b>1 Вимоги до програмного продукту.....</b>	<b>7</b>
<b>1.1 Визначення потреб споживача.....</b>	<b>7</b>
<b>1.1.1 Ієрархія потреб споживача.....</b>	<b>7</b>
<b>1.1.2 Деталізація матеріальної потреби.....</b>	<b>7</b>
<b>1.2 Бізнес вимоги до ПП.....</b>	<b>8</b>
<b>1.2.1 Опис проблем користувача.....</b>	<b>8</b>
<b>1.2.1.1 Концептуальний опис проблеми споживача.....</b>	<b>8</b>
<b>1.2.1.2 Метричний опис проблеми користувача.....</b>	<b>9</b>
<b>1.2.2 Мета створення ПП.....</b>	<b>9</b>
<b>1.2.2.1 Проблемний аналіз існуючих ПП.....</b>	<b>9</b>
<b>1.2.2.2 Мета створення ПП.....</b>	<b>10</b>
<b>1.2.3 Назва ПП.....</b>	<b>10</b>
<b>1.2.3.1 Гасло ПП.....</b>	<b>10</b>
<b>1.2.3.2 Логотип ПП.....</b>	<b>11</b>
<b>1.3 Вимоги користувача до ПП.....</b>	<b>11</b>
<b>1.3.1 Історія користувача ПП.....</b>	<b>11</b>
<b>1.3.2 Діаграма прецедентів ПП.....</b>	<b>12</b>
<b>1.3.3 Сценарії використання прецедентів ПП.....</b>	<b>12</b>
<b>1.4 Функціональні вимоги до ПП.....</b>	<b>18</b>
<b>1.4.1 Багаторівнева класифікація функціональних вимог.....</b>	<b>18</b>
<b>1.4.2 Функціональний аналіз існуючих ПП.....</b>	<b>20</b>
<b>1.5 Нефункціональні вимоги до ПП.....</b>	<b>21</b>
<b>1.5.1 Опис зовнішніх інтерфейсів.....</b>	<b>21</b>

1.5.1.1	Опис інтерфейса користувача.....	21
1.5.1.1.1	Опис INPUT-інтерфейса користувача.....	21
1.5.1.1.2	Опис OUTPUT-потоків.....	23
1.5.1.2	Опис інтерфейсу з зовнішніми пристроями.....	24
1.5.1.3	Опис програмних інтерфейсів.....	25
1.5.1.4	Опис інтерфейсів передачі інформації.....	25
1.5.1.5	Опис атрибутів продуктивності.....	25
2	Планування процесу розробки програмного продукту.....	26
2.1	Планування ітерацій розробки програмного продукту.....	26
2.2	Концептуальний опис архітектури програмного продукту.....	27
2.3	План розробки ПП.....	27
2.3.1	Оцінка трудомісткості розробки ПП.....	27
2.3.2	Визначення дерева робіт з розробки ПП.....	31
2.3.3	Графік робіт з розробки ПП.....	34
2.3.3.1	Таблиця з графіком робіт.....	34
2.3.3.2	Діаграма Ганта.....	35
3	Проектування ПП.....	36
3.1	Концептуальне та логічне проектування структур даних ПП.....	36
3.1.1	Концептуальне проектування на основі UML-діаграми концептуальних класів.....	36
3.1.2	Логічне проектування структур даних.....	36
3.2	Проектування програмних класів.....	37
3.3	Проектування алгоритмів роботи методів програмних класів....	37
3.4	Проектування тестових наборів методів програмних класів.....	42
4	Конструювання ПП.....	45

4.1 Особливості конструювання структур даних.....	45
4.1.1 Особливості інсталяції та роботи з СУБД.....	45
4.1.2 Особливості створення структур даних.....	45
4.2 Особливості конструювання програмних модулів.....	46
4.2.1 Особливості роботи з інтегрованим середовищем розробки.....	46
4.2.2 Особливості створення програмної структури з урахуванням спеціалізованого фреймворку.....	47
4.2.3 Особливості створення програмних модулів.....	54
4.2.4 Особливості розробки алгоритмів методів програмних класів або процедур/функцій.....	55
4.3 Тестування програмних модулів.....	66
5 Розгортання та валідація ПП.....	70
5.1 Інструкція з встановлення ПП.....	70
5.2 Інструкція з використання ПП.....	71
5.3 Результати валідації ПП.....	76
Висновки.....	77
Джерела.....	77

## 1 Вимоги до програмного продукту

### 1.1 Визначення потреб споживача

#### 1.2.3 Ієрархія потреб споживача

Відомо, що в теорії маркетингу потреби людини можуть бути представлені у вигляді ієрархії потреб ідей американського психолога Абрахама Маслоу включають рівні:

- фізіологія (вода, їжа, житло, сон);
- безпека (особиста, здоров'я, стабільність),
- приналежність (спілкування, дружба, любов),
- визнання (повага оточуючих, самооцінка),
- самовираження (вдосконалення, персональний розвиток).

На рисунку 1.1 представлено одну ієрархію потреби споживача, яку хотілося б задовольнити, використовуючи майбутній програмний продукт.



Рис. 1 – Ієрархія потреб споживача наш

Даний ПП покриває потреби в приналежності та фізіологічні.

#### 1.2.4 Деталізація матеріальної потреби

Матеріальні потреби – це сукупність матеріальних благ, які мають матеріальну форму і виступають як головний спонукальний мотив трудової діяльності людини.

Споживач – це людина, яка споживає продукти виробництва і сфери послуг для відтворення робочої сили.

Тому була винайдена деталізація матеріальної потреби для нашого ПП.

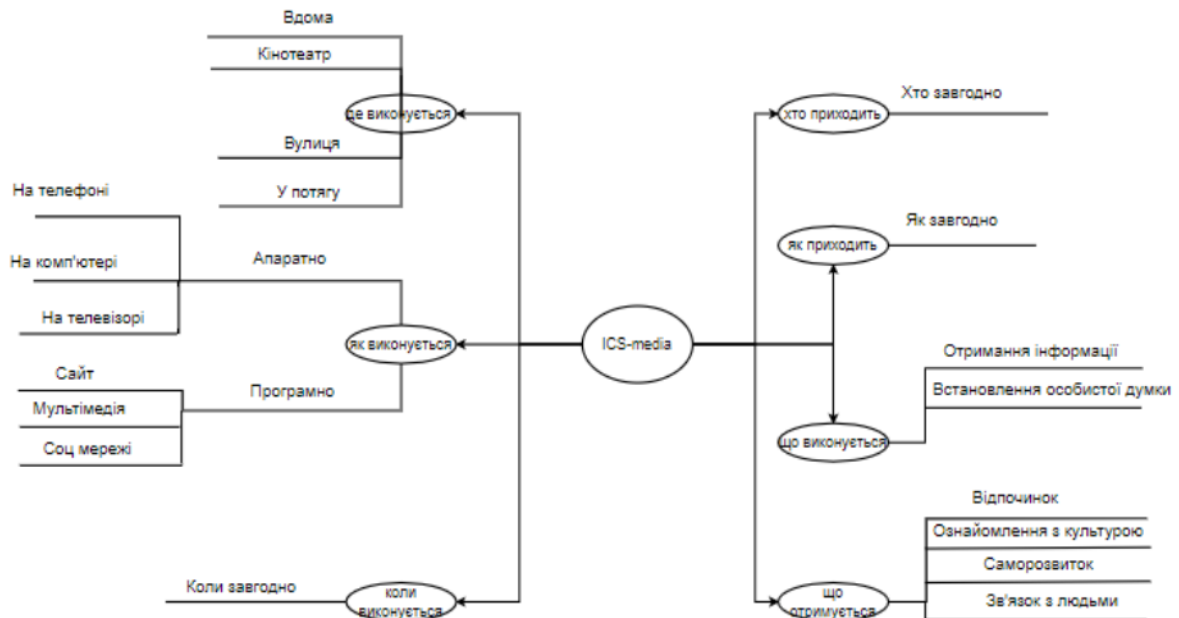


Рисунок 2 - Mindmap майбутнього ПП

### 1.3 Бізнес вимоги до ПП

#### 1.2.1 Опис проблем користувача

##### 1.2.1.1 Концептуальний опис проблеми споживача

Умова задоволення потреб переведення даних в інформацію:

- Доступність
- Представленість мовою споживача
- Інтерес
- Актуальність

Загальний опис проблеми: неможливо загрузити фільми особистого виробництва з можливістю підтримки режисера.



### 1.2.1.2 Метричний опис проблеми користувача

№	Параметр незадоволеності
1	Немає сервісів, де люди можуть показувати свої фільми людям та отримувати пожертвування на нові фільми.

Таблиця 1 – Параметри незадоволеності

Рівень користності для виробників CL можна визначити як

$$CL = NA / N,$$

NA – кількість просмотрів;

N – кількість пожертвувань

### 1.2.2 Мета створення ПП

#### 1.2.2.1 Проблемний аналіз існуючих ПП

№	Назва продукту	Вартість	Ступінь готовності	Примітка
1	ivi.ru	Безкоштовно/ платно	1	Немає знижок для студентів. Більшість фільмів платні.
2	film.ru	Безкоштовно	1	Працює не стабільно. Використовує сторонні ресурси.

3	gidonline.io	Безкоштовно	1	Багато реклами. Часто фільми у поганій якості.
4	okko.tv	Безкоштовно/ платно	1	Не доступно в Україні.
5	baskino.me	Безкоштовно	1	Багато реклами.

Таблиця 2 – Аналіз існуючих ПП

#### 1.2.2.2 Мета створення ПП

Метою створення даного ПП є: надати можливість загрузати свої фільми починаючим режисерам в яких є гарні ідеї для фільмів, але недостатньо фінансування.

#### 1.2.3 Назва ПП

ICS - media

##### 1.2.3.1 Гасло ПП

«Загружайте, дивіться!» - ICS - media.

### 1.2.3.2 Логотип ПП



Рисунок 3 – Логотип ПП

## 1.4 Вимоги користувача до ПП

### 1.3.1 Історія користувача ПП

- Як гість, я можу зареєструватися, щоб стати користувачем.
- Як гість, я можу подивитися будь-який фільм з каталогу.
- Як гість, я можу сортувати каталог за різними фільтрами.
- Як гість, я можу шукати за назвою.
- Як користувач, я можу коментувати.
- Як користувач, я можу віддячити творця.
- Як користувач, я можу завантажити фільм.
- Як адміністратор, я можу видаляти фільми.

### 1.3.2 Діаграма прецедентів ПП

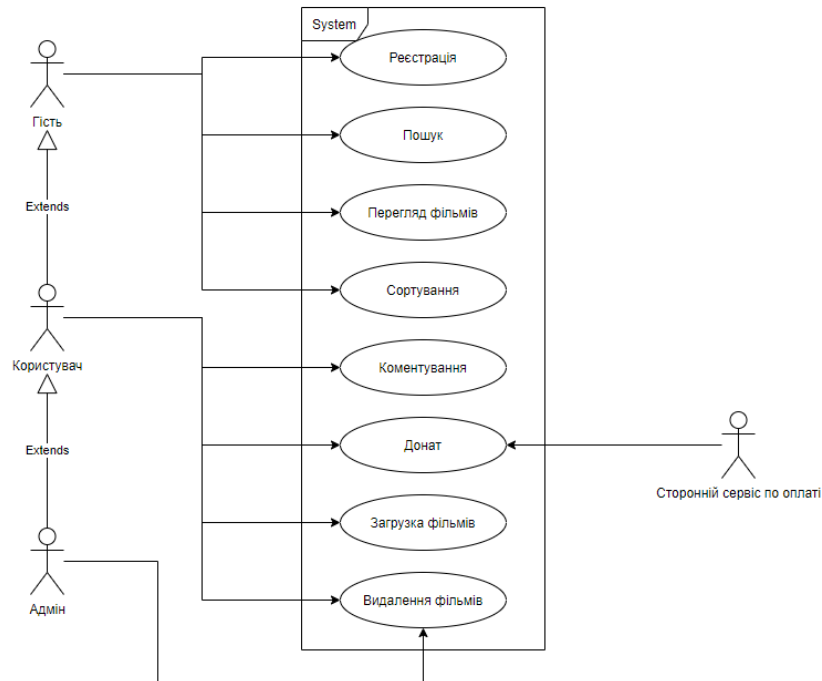


Рисунок 4 – Діаграма прецедентів ПП

### 1.3.3 Сценарії використання прецедентів ПП

- Реєстрація користувача.

Передумова: Гість має знаходитись на сайті.

Актори які можуть виконувати: Гість.

Актор-основний ініціатор: Гість.

Успішний результат: Гість успішно зареєструвався.

Успішний сценарій:

1. ПП дає можливість реєстрації;
2. Гість переходить до сторінки реєстрації;

3. ПП дає можливість ввести такі дані:

- Логін;

- Пароль.

4. Гість вводить свої дані до полей Логін, Пароль;

5. ПП видає повідомлення «Успішне створення користувача»;

Альтернативний сценарій:

4.1.1 Гісь залишає невірні поля (пусті);

4.1.1 Повідомлення, про те що поля пусті.

4.2.1. Гість вводить логін який вже існує в базі;

4.2.2. Повідомлення про те що користувач вже існує за даним логіном.

- Використання пошуку

Передумова: Користувач має знаходитись на головній сторінці ПП.

Актори які можуть виконувати: Гість, Адміністратор, Користувач.

Актор-основний ініціатор: Користувач, Гість, Адміністратор.

Успішний результат: Пошук видасть коректний результат введений Користувачем.

Успішний сценарій:

1. ПП видає строку пошуку;

2. Користувач вводить запит у строку пошуку;

3. Пошук видає потрібний запит який ввів користувач.

Альтернативний сценарій:

2.1. Користувач вводить невірний (пустий) запит, або запит даних яких не існує;

2.2. Повідомлення, про те що немає даних за введеним запитом.

- Перегляд фільму.

Передумова: Користувач повинен перейти на сторінку фільму.

Актори які можуть виконувати: Гість, Адміністратор, Користувач.

Актор-основний ініціатор: Користувач, Гість, Адміністратор.

Успішний результат: Фільм який вибраний Користувачем успішно запустився, та працює без багів.

Успішний сценарій:

1. ПП видає список фільмів на головній сторінці;
2. Користувач переходить на сторінку будь-якого фільму;
3. Користувач натискає на кнопку старта відео;
4. Запускається фільм.

Альтернативний сценарій:

4.1. Повідомлення, що до фільму немає доступу в даний момент.

- Додавання фільмів.

Передумова: Користувач має бути на головній сторінці після «Реєстрації».

Актори які можуть виконувати: Адміністратор, Користувач.

Актор-основний ініціатор: Користувач, Адміністратор.

Успішний результат: Фільм успішно завантажено до сайту, його можна знайти у пошуку.

Успішний сценарій:

1. ПП дає можливість перейти до особистого кабінету;
2. Користувач переходить до особистого кабінету;
3. ПП дає можливість перейти до сторінки загрузки фільму;
4. Користувач переходить до сторінки загрузки;
5. ПП дає можливість загрузити свій фільм з такими полями:

- Назва;
- Жанр;
- Країна;
- Посилання на постер;
- Рік;
- Опис;
- URL;

6. Користувач вводить дані, та клікає «Завантажити»;
7. ПП видає повідомлення що фільм загрузено.

Альтернативний сценарій:

- 4.1 Користувач залишає поля пустими;
- 4.2 ПП видає повідомлення що поля не можуть бути пустими.

- Пожертвування творцям фільмів.

Передумова: Користувач має бути на головній сторінці після «Реєстрації».

Актори які можуть виконувати: Адміністратор, Користувач.

Актор-основний ініціатор: Сторонній сервіс по переводу коштів.

Успішний результат: Успішно відправлено гроші до творця.

1. ПП видає список фільмів на головній сторінці;
2. Користувач переходить до сторінки будь-якого фільму;
3. ПП дає можливість пожертвування;
4. Користувач клікає на кнопку «Пожертвувати»;
5. Відкривається сторонній сервіс для переводу коштів, та користувач проводить дії на зовнішньому сервісі;
6. Сервіс перенаправляє до ПП;
7. ПП видає повідомлення про успішний перевід коштів.

Альтернативний сценарій:

3.1 ПП не видає можливості пожертвування, бо користувач який загрузив фільм не ввів свої банківські дані.

7.1 ПП видає повідомлення про не успішний перевід коштів.

- Коментування фільмів.

Передумова: Користувач має бути на сторінці фільму як у «Перегляд фільму».

Актори які можуть виконувати: Адміністратор, Користувач.

Актор-основний ініціатор: Користувач.



Успішний результат: Успішно прикріплено коментар до фільму.

1. ПП дає можливість написати коментар;
2. Користувач пише свій коментар та нажимає кнопку відправити;
3. ПП видає повідомлення що успішно прикріплено коментар до фільму.

Альтернативний сценарій:

2.1 Користувач залишає поле коментаря пустим та нажимає кнопку відправити;

2.2. Повідомлення, що коментар не має бути пустим.

- Видалення фільмів.

Передумова: Користувач має бути на головній сторінці після реєстрації «Реєстрації» та має бути хоча б один фільм завантажено.

Актори які можуть виконувати: Адміністратор, Користувач.

Актор-основний ініціатор: Користувач, Адміністратор.

Успішний результат: Успішно видалено фільм.

1. ПП дає можливість перейти до особистого кабінету;
2. Користувач переходить до особистого кабінету;
3. ПП дає можливість переглянути свої завантажені фільмів;
4. Користувач переходить до сторінки свого фільму;
5. ПП дає можливість видалити фільм;
6. Користувач клікає на кнопку видалення фільму;
7. Повідомлення що фільм видалено.

Альтернативний сценарій:

1.1 Адміністратор переходить до головної сторінки сайту.

1.2 Адміністратор переходить до сторінки будь-якого фільму з головної сторінки

1.3 ПП дає можливість видалити фільм;

1.4 Адміністратор клікає на кнопку видалення фільму;

1.5 Повідомлення що фільм видалено.

## **1.5 Функціональні вимоги до ПП**

### **1.4.1 Багаторівнева класифікація функціональних вимог**

Ідентифікатор функції	Назва функції
FR1	Реєстрація користувача.
FR1.1	ПП дає можливість ввести данні .
FR1.2	Користувач передає свої данні.
FR1.3	ПП перевіряє введені дані.
FR1.4	ПП надає доступ користувача до сайту.
FR2	Здійснити пошук фільму.
FR2.1	ПП надає можливість ввести дані за якими буде знаходитися фільм.
FR2.2	Користувач вводить данні фільму.
FR2.3	ПП видає користувачеві бажаний фільм.
FR3	Пожертвування на фільм.
FR3.1	ПП надає можливість пожертвувати творцям фільмів.
FR3.2	ПП перенаправляє користувача на сторонній сервіс по

	оплаті.
FR3.3	Користувач здійснює оплату через сторонній сервіс.
FR3.4	ПП висилає користувачу мели з підтвердження пожертвування.
FR4	Коментування фільмів.
FR4.1	ПП надає можливість залишити коментар.
FR4.2	Користувач залишає коментар.
FR4.3	ПП залишає коментар на сторінці фільму.
FR5	Перегляд фільмів.
FR5.1	ПП надає можливість дивитися фільм.
FR5.2	Користувач переходить до сторінки фільму.
FR5.3	ПП коректо запускає фільм.
FR6	Завантаження фільмів.
FR6.1	ПП надає можливість завантажити фільм.
FR6.2	Користувач завантажує фільм.
FR6.3	ПП видає фільм в каталозі.
FR7	Видалення фільмів.
FR7.1	ПП надає можливість видалити фільм.
FR7.2	Користувач видаляє фільм.
FR7.3	ПП більше не має фільм в каталозі.

Таблиця 3 – Функціональні вимоги до ПП

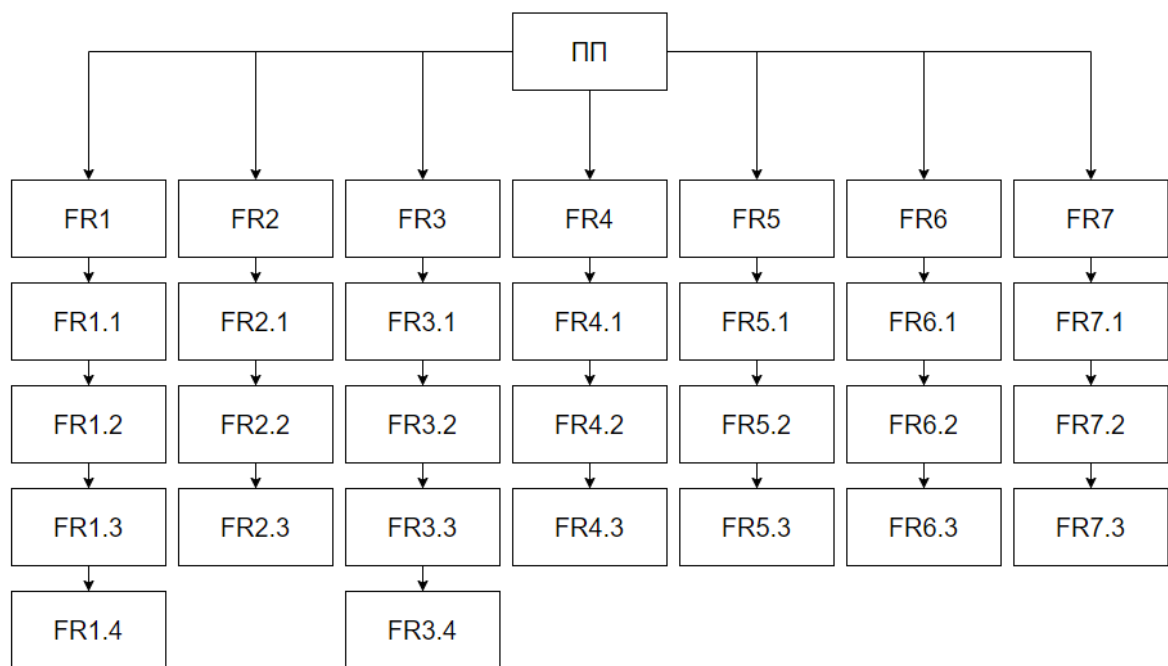


Рисунок 5 – WBS структура вимог

#### 1.4.2 Функціональний аналіз існуючих ПП

Ідентифікатор	Гид онлайн	Ivi	Baskino	Okko	Megogo
FR1	+	+	+	+	+
FR2	+	+	+	+	+
FR3	-	-	-	-	-
FR4	+	+	+	+	+
FR5	+	+	+	+	+
FR6	-	-	-	-	-
FR7	+	+	+	+	+

Таблиця 4 – Функціональний аналіз існуючих ПП

## 1.6 Нефункціональні вимоги до ПП

### 1.5.1 Опис зовнішніх інтерфейсів

#### 1.5.1.1 Опис інтерфейса користувача

##### 1.5.1.1.1 Опис INPUT-інтерфейса користувача

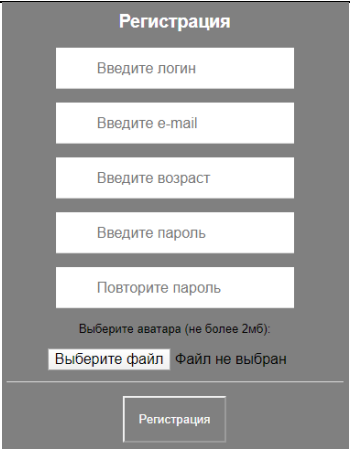
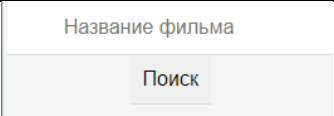
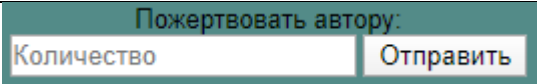
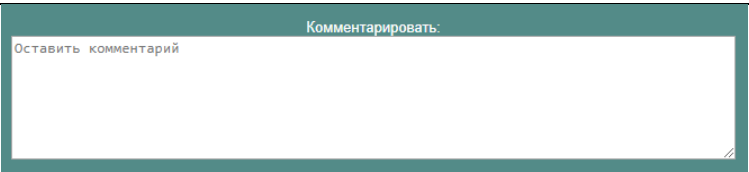
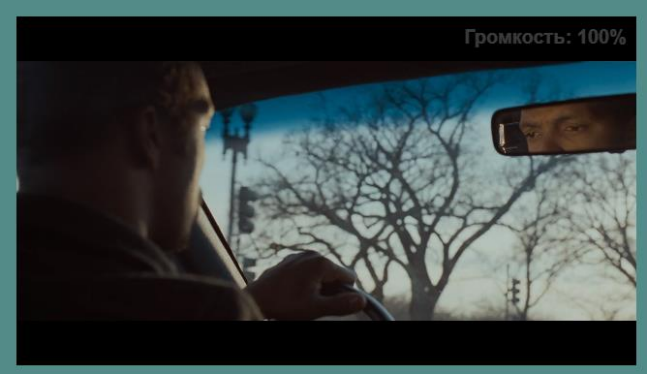
Ідентифікатор	Засіб INPUT	Особливості
FR1.1	Комп'ютерна клавіатура; маніпулятор типу «миша»; сенсорний екран;	Використання кнопок миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення даних про користувача.
FR1.2		
FR1.3		
FR1.4		
FR2.1		Використання кнопок миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення даних фільм (назва).
FR2.2		
FR2.3		
FR3.1		Використання кнопок миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення даних для донату.
FR3.2		
FR3.3		
FR3.4		
FR4.1		Використання кнопок


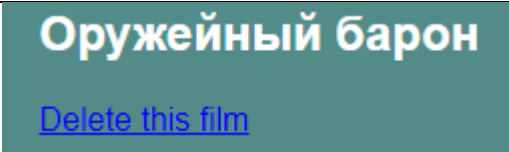
FR4.2		миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення тексту коментаря
FR4.3		
FR5.1		Використання кнопок миші для запуску фільма.
FR5.2		
FR5.3		
FR6.1		Використання кнопок миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення даних про фільм.
FR6.2		
FR6.3		
FR7.1		Використання кнопок миші для навігації по сторінці для видалення фільму.
FR7.2		
FR7.3		

Таблиця 5 – INPUT-інтерфейси

OUTPUT – потік у нашому ПП це вікно браузера.

### 1.5.1.1.2 Опис OUTPUT-потоків

Ідентифікатор функції	OUTPUT
FR1	
FR2	
FR3	
FR4	
FR5	

FR6	
FR7	

Таблиця 6 – Засоби OUTPUT-потоків

#### 1.5.1.2 Опис інтерфейсу з зовнішніми пристроями

Ідентифікатор функції	Зовнішній пристрій
FR 1	Смартфон, планшет, Desktop-персональний комп'ютер, Notebook;
FR 2	
FR 3	
FR 4	
FR 5	
FR 6	
FR 7	

Таблиця 7 – Опис інтерфейсу з зовнішніми пристроями



### 1.5.1.3 Опис програмних інтерфейсів

Версії операційних систем та програмних бібліотек, які знадобляться при реалізації більшості функцій ПП.

Версії операційних систем та програмних бібліотек, які знадобляться при реалізації більшості функцій ПП.	Для користування: <ul style="list-style-type: none"><li>- Windows 95 – 10, Linux.</li><li>- Будь-який браузер</li></ul> Для розробки: <ul style="list-style-type: none"><li>- PHP 7</li></ul>
----------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Таблиця 8 – опис програмних інтерфейсів

### 1.5.1.4 Опис інтерфейсів передачі інформації

Опис інтерфейсів передачі інформації, які знадобляться при реалізації більшості функцій ПП.

- Провідні інтерфейси:
  - Ethernet
- Безпроводні інтерфейси:
  - Wi-Fi;

### 1.5.1.5 Опис атрибутів продуктивності

Ідентифікатор	Максимальний час реакції, с
FR1	15
FR2	15
FR3	30
FR4	10
FR5	50
FR6	20
FR7	10

Таблиця 9 – Опис атрибутів продуктивності

## 2 Планування процесу розробки програмного продукту

### 2.1 Планування ітерацій розробки програмного продукту

Ідентифікатор функції	Назва функції	Функціональні залежності	Вплив на досягнення мети, %
FR1	Реєстрація користувача.	-	
FR2	Здійснити пошук фільму.	-	
FR3	Пожегтвування на фільм.	FR1	
FR4	Коментування фільмів.	FR1	
FR5	Перегляд фільмів.	FR6	
FR6	Завантаження фільмів.	FR1	
FR7	Видалення фільмів.	FR1	
FR2	Здійснити пошук фільму.	-	

Таблиця 10 – Планування ітерацій розробки

## 2.2 Концептуальний опис архітектури програмного продукту

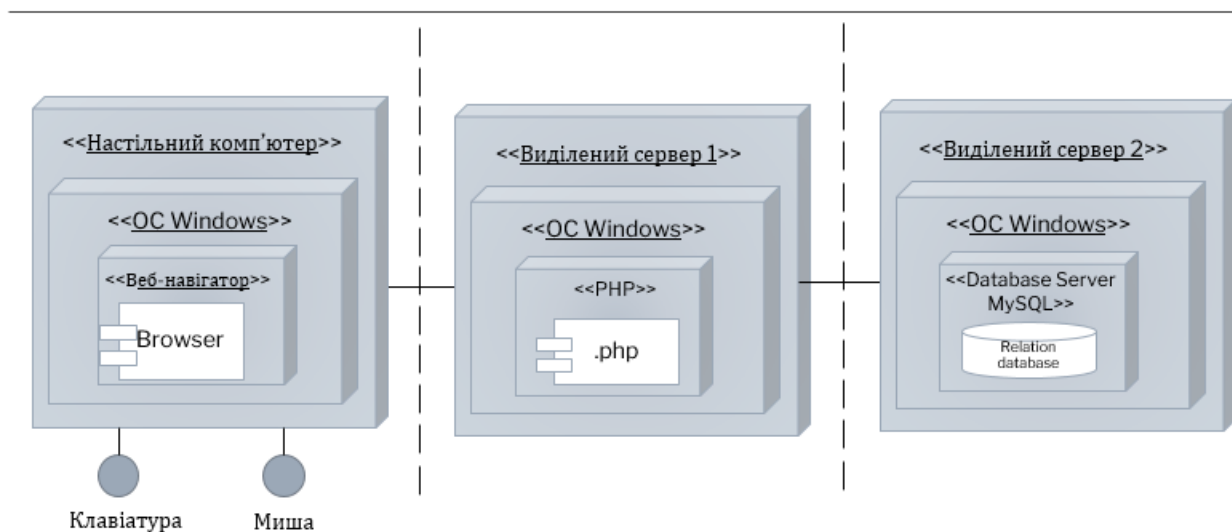


Рисунок 6 – Концептуальний опис архітектури ПП

## 2.3 План розробки ПП

### 2.3.1 Оцінка трудомісткості розробки ПП

#### 1. Визначення вагових показників акторів

Актор	Ваговий коефіцієнт
Гість	1
Користувач	2
Адміністратор	3

Таблиця 11 – Вагові коефіцієнти акторів

$$A = 1 + 2 + 3 = 6$$

#### 2. Визначення вагових показників прецедентів UC

Назва прецедента	Тип прецедента	Кількість кроків сценарію	Ваговий коефіцієнт
FR1	Середній	4	15
FR2	Простий	3	10
FR3	Середній	4	5
FR4	Простий	3	5
FR5	Простий	3	15
FR6	Простий	3	10
FR7	Простий	3	5

Таблиця 12 – Вагові показники прецедентів

$$UC = 15 + 10 + 5 + 5 + 15 + 10 + 5 = 65$$

### 3. Визначення UUCP

$$UUCP = A + UC = 65 + 6 = 71$$

### 4. Визначення технічної складності проекту

Показники	Опис	Вага
T1	Распределенная система	1
T2	Высокая производительность (пропускная способность)	2
T3	Работа конечных пользователей в режиме онлайн	1
T4	Сложная обработка данных	0,5

T5	Повторное использование кода	0,5
T6	Простота установки	0,5
T7	Простота использования	1
T8	Переносимость	1
T9	Простота внесения изменений	2
T10	Параллелизм	0
T11	Специальные требования к безопасности	1
T12	Непосредственный доступ к системе со стороны внешних пользователей	3
T13	Специальные требования к обучению пользователей	1

Таблица 13 – Визначення технічної складності проекту

$$TCF = 0,6 + (0,01 * (ST_i * Вага_i))$$

$$TCF = 0,6 + (0,01 * (14,5)) = 0,745$$

#### 5. Визначення рівня кваліфікації розробників

Показник	Опис показника	Вага
F1	Знакомство с технологией	2
F2	Опыт разработки приложений	1
F3	Опыт	1

	использования объектно- ориентированного подхода	
F4	Наличие ведущего аналитика	0
F5	Мотивация	3
F6	Стабильность требований	4
F7	Частичная занятость	1
F8	Сложные языки программирования	1

Таблица 14 – Визначення рівня кваліфікації розробників

$$EF = 1,4 + (-0,03 * (SF_i * \text{Вага}_i)) = 1$$

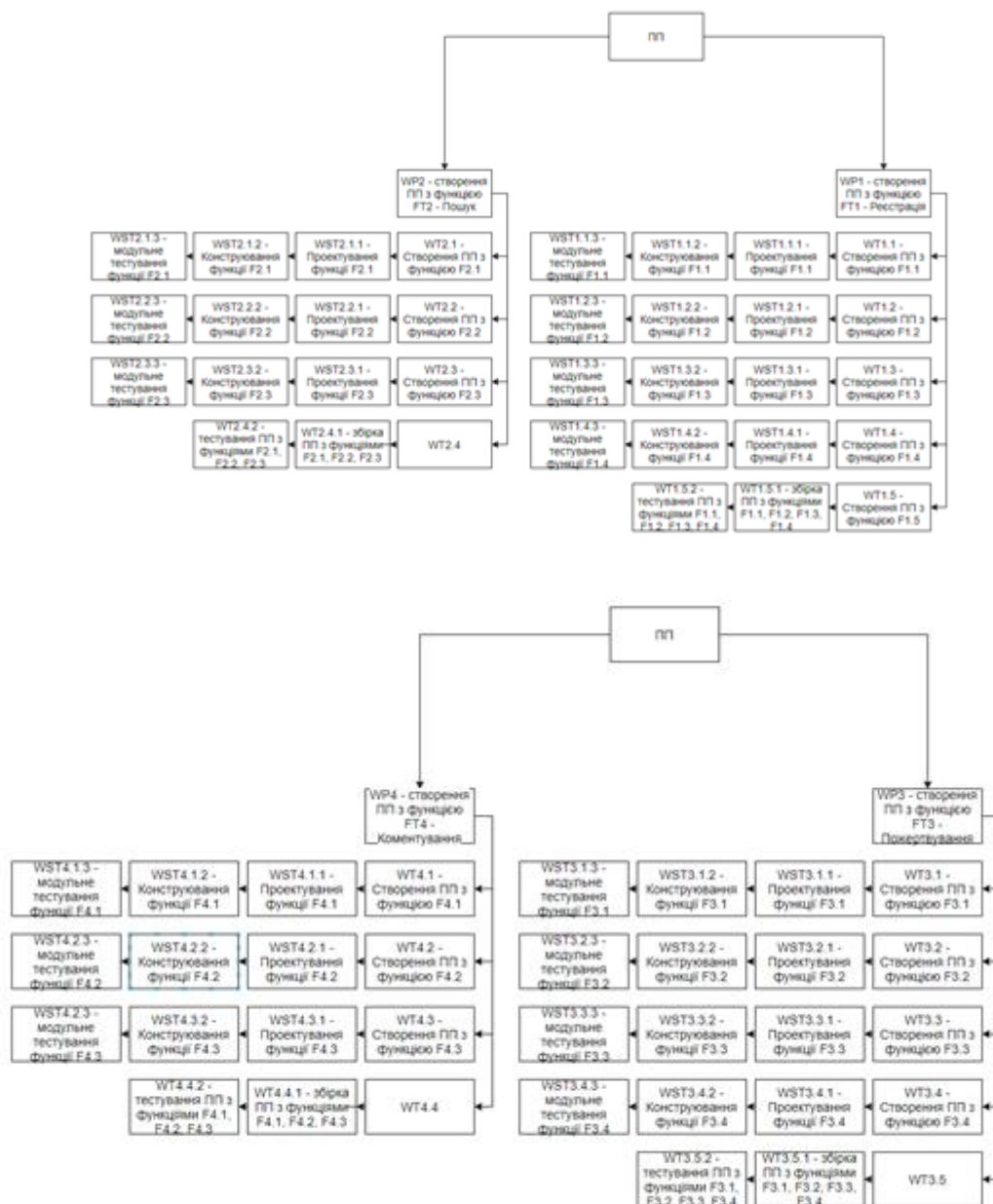
#### 6. Визначення UCP

$$UCP = UUCP * TCF * EF = 71 * 0,745 * 1 = 52,895$$

#### 7. Оцінка трудомісткості проекту

$$\text{Трудомісткість} = UCP * 20 = 1057,9 \text{ люд\год}$$

## 2.3.2 Визначення дерева робіт з розробки ПП



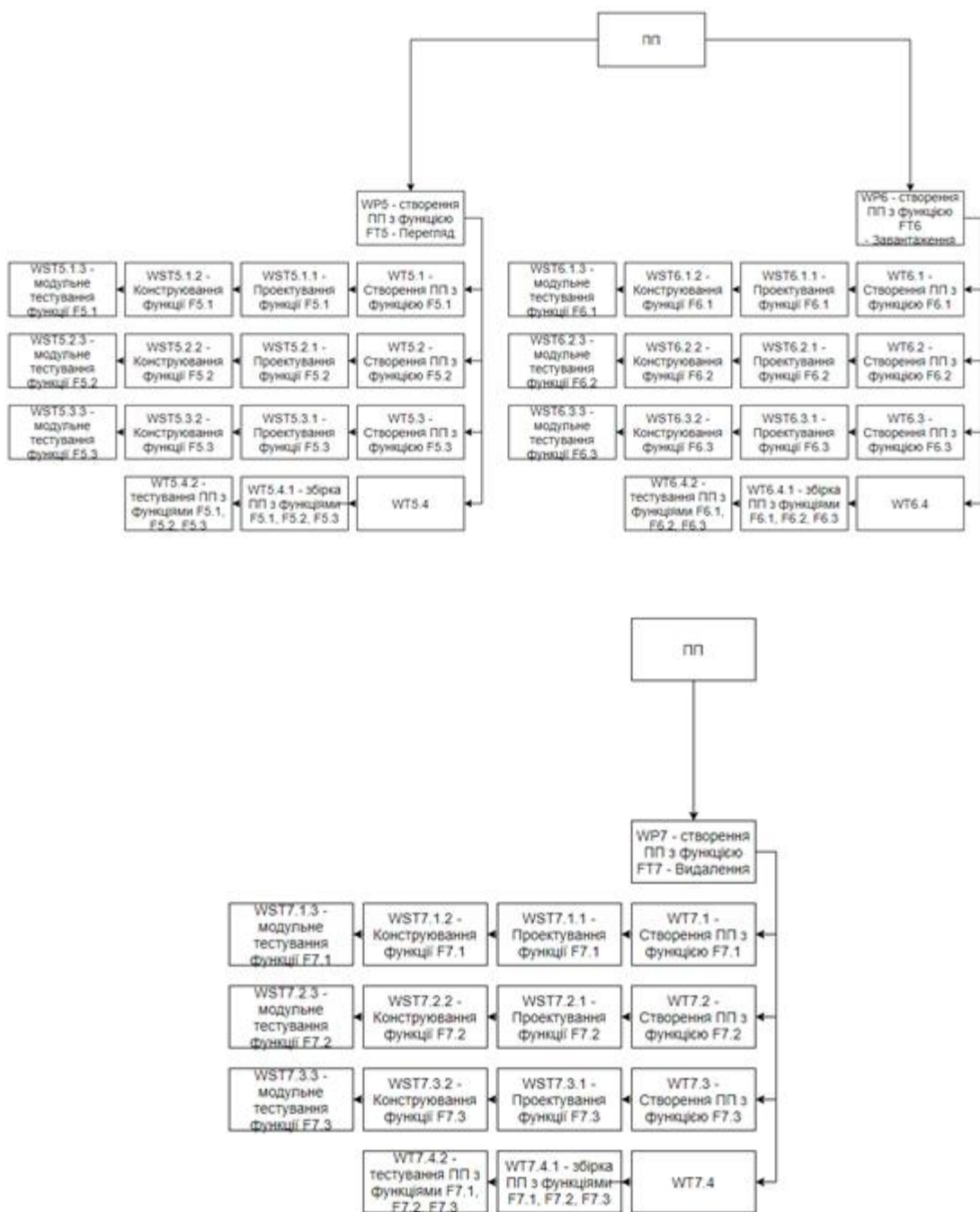


Рисунок 7 – WBS дерево робіт



Ідентифікатор	Хто виконував
1.1.*	Додон Д.І.
1.2.*	Панібратцев О.Ю.
1.3.*	Додон Д.І.
1.4.*	Панібратцев О.Ю.
1.5.*	Додон Д.І.
2.1.*	Панібратцев О.Ю.
2.2.*	Додон Д.І.
2.3.*	Панібратцев О.Ю.
3.1.*	Додон Д.І.
3.2.*	Панібратцев О.Ю.
3.3.*	Додон Д.І.
3.4.*	Панібратцев О.Ю.
3.5.*	Додон Д.І.
4.1.*	Панібратцев О.Ю.
4.2.*	Додон Д.І.
4.3.*	Панібратцев О.Ю.
4.4.*	Додон Д.І.
5.1.*	Панібратцев О.Ю.
5.2.*	Додон Д.І.
5.3.*	Панібратцев О.Ю.
5.4.*	Додон Д.І.
6.1.*	Панібратцев О.Ю.
6.2.*	Додон Д.І.
6.3.*	Панібратцев О.Ю.
6.4.*	Додон Д.І.
7.1.*	Панібратцев О.Ю.
7.2.*	Додон Д.І.
7.3.*	Панібратцев О.Ю.
7.4.*	Додон Д.І.

Рисунок 8 – Опис підзадач із закріпленням виконавця

### 2.3.3 Графік робіт з розробки ПП

#### 2.3.3.1 Таблиця з графіком робіт

Ідентифікатор	Дата початку виконання	Кількість днів витрачених	Дата кінця виконання	Хто виконував
1.1.*	01.10.2020	1	01.10.2020	Додон Д.І.
1.2.*	01.10.2020	1	01.10.2020	Панібратцев О.Ю.
1.3.*	01.10.2020	1	01.10.2020	Додон Д.І.
1.4.*	13.10.2020	1	13.10.2020	Панібратцев О.Ю.
1.5.*	13.10.2020	1	13.10.2020	Додон Д.І.
2.1.*	13.10.2020	1	13.10.2020	Панібратцев О.Ю.
2.2.*	25.10.2020	1	25.10.2020	Додон Д.І.
2.3.*	25.10.2020	1	25.10.2020	Панібратцев О.Ю.
3.1.*	25.10.2020	1	25.10.2020	Додон Д.І.
3.2.*	06.11.2020	1	06.11.2020	Панібратцев О.Ю.
3.3.*	06.11.2020	1	06.11.2020	Додон Д.І.
3.4.*	06.11.2020	1	06.11.2020	Панібратцев О.Ю.
3.5.*	10.11.2020	1	10.11.2020	Додон Д.І.
4.1.*	10.11.2020	1	10.11.2020	Панібратцев О.Ю.
4.2.*	10.11.2020	1	10.11.2020	Додон Д.І.
4.3.*	14.11.2020	1	14.11.2020	Панібратцев О.Ю.
4.4.*	14.11.2020	1	14.11.2020	Додон Д.І.
5.1.*	14.11.2020	1	14.11.2020	Панібратцев О.Ю.
5.2.*	22.11.2020	1	22.11.2020	Додон Д.І.
5.3.*	22.11.2020	1	22.11.2020	Панібратцев О.Ю.
5.4.*	22.11.2020	1	22.11.2020	Додон Д.І.
6.1.*	02.12.2020	1	02.12.2020	Панібратцев О.Ю.
6.2.*	02.12.2020	1	02.12.2020	Додон Д.І.
6.3.*	02.12.2020	1	02.12.2020	Панібратцев О.Ю.
6.4.*	06.12.2020	1	06.12.2020	Додон Д.І.
7.1.*	06.12.2020	1	13.12.2020	Панібратцев О.Ю.
7.2.*	07.12.2020	1	13.12.2020	Додон Д.І.
7.3.*	07.12.2020	1	13.12.2020	Панібратцев О.Ю.
7.4.*	07.12.2020	1	13.12.2020	Додон Д.І.

Рисунок 9 – Графік робіт

### 2.3.3.2 Діаграма Ганта

Іденти фікато р	Дата початку виконання	Кількість днів витрачен их	Дата кінця виконання	Хто виконував	01. 10. 20	13. 10. 20	25. 10. 20	06. 11. 20	10. 11. 20	14. 11. 20	22. 11. 20	02. 12. 20	06. 12. 20	07. 12. 20
1.1.*	01.10.2020	1	01.10.2020	Додон Д.І.										
1.2.*	01.10.2020	1	01.10.2020	Панібратцев О.Ю.										
1.3.*	01.10.2020	1	01.10.2020	Додон Д.І.										
1.4.*	13.10.2020	1	13.10.2020	Панібратцев О.Ю.										
1.5.*	13.10.2020	1	13.10.2020	Додон Д.І.										
2.1.*	13.10.2020	1	13.10.2020	Панібратцев О.Ю.										
2.2.*	25.10.2020	1	25.10.2020	Додон Д.І.										
2.3.*	25.10.2020	1	25.10.2020	Панібратцев О.Ю.										
3.1.*	25.10.2020	1	25.10.2020	Додон Д.І.										
3.2.*	06.11.2020	1	06.11.2020	Панібратцев О.Ю.										
3.3.*	06.11.2020	1	06.11.2020	Додон Д.І.										
3.4.*	06.11.2020	1	06.11.2020	Панібратцев О.Ю.										
3.5.*	10.11.2020	1	10.11.2020	Додон Д.І.										
4.1.*	10.11.2020	1	10.11.2020	Панібратцев О.Ю.										
4.2.*	10.11.2020	1	10.11.2020	Додон Д.І.										
4.3.*	14.11.2020	1	14.11.2020	Панібратцев О.Ю.										
4.4.*	14.11.2020	1	14.11.2020	Додон Д.І.										
5.1.*	14.11.2020	1	14.11.2020	Панібратцев О.Ю.										
5.2.*	22.11.2020	1	22.11.2020	Додон Д.І.										
5.3.*	22.11.2020	1	22.11.2020	Панібратцев О.Ю.										
5.4.*	22.11.2020	1	22.11.2020	Додон Д.І.										
6.1.*	02.12.2020	1	02.12.2020	Панібратцев О.Ю.										
6.2.*	02.12.2020	1	02.12.2020	Додон Д.І.										
6.3.*	02.12.2020	1	02.12.2020	Панібратцев О.Ю.										
6.4.*	06.12.2020	1	06.12.2020	Додон Д.І.										
7.1.*	06.12.2020	1	13.12.2020	Панібратцев О.Ю.										
7.2.*	07.12.2020	1	13.12.2020	Додон Д.І.										
7.3.*	07.12.2020	1	13.12.2020	Панібратцев О.Ю.										
7.4.*	07.12.2020	1	13.12.2020	Додон Д.І.										

Рисунок 10 – Діаграма Ганта

### 3 Проектування ПП

#### 3.1 Концептуальне та логічне проектування структур даних ПП

##### 3.1.1 Концептуальне проектування на основі UML-діаграми концептуальних класів

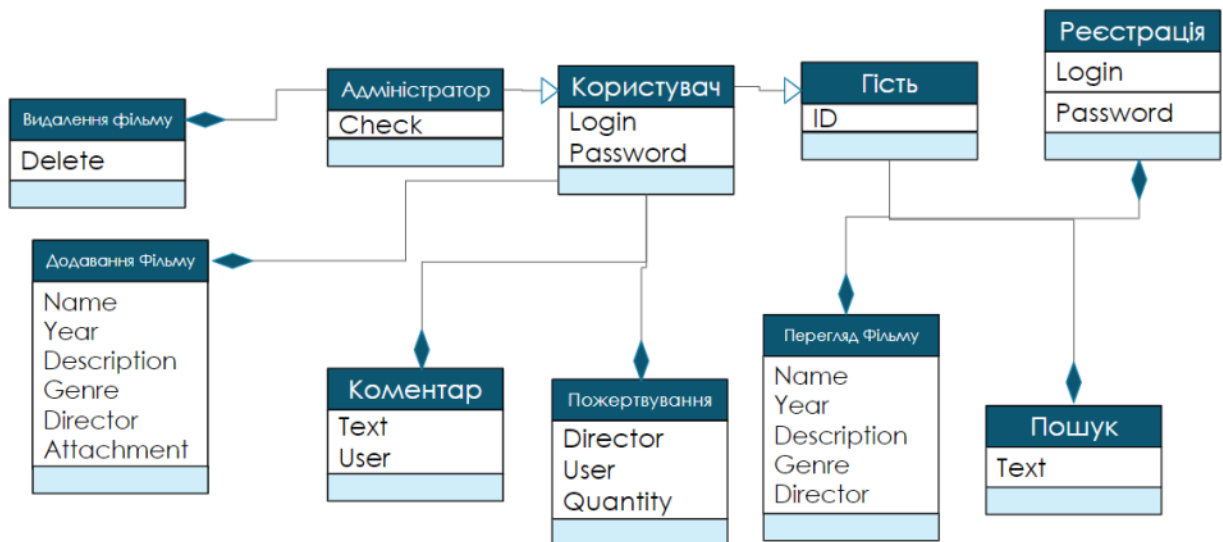


Рисунок 11 – UML-діаграма концептуальних класів

##### 1.1.2 Логічне проектування структур даних

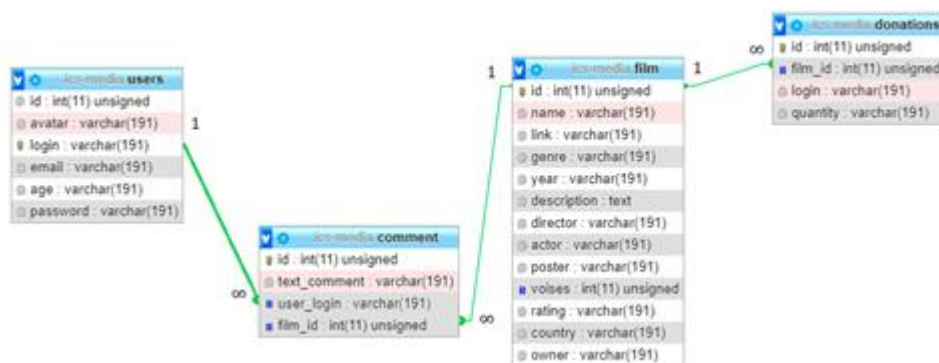


Рисунок 12 – Структура БД

## 1.2 Проектування програмних класів

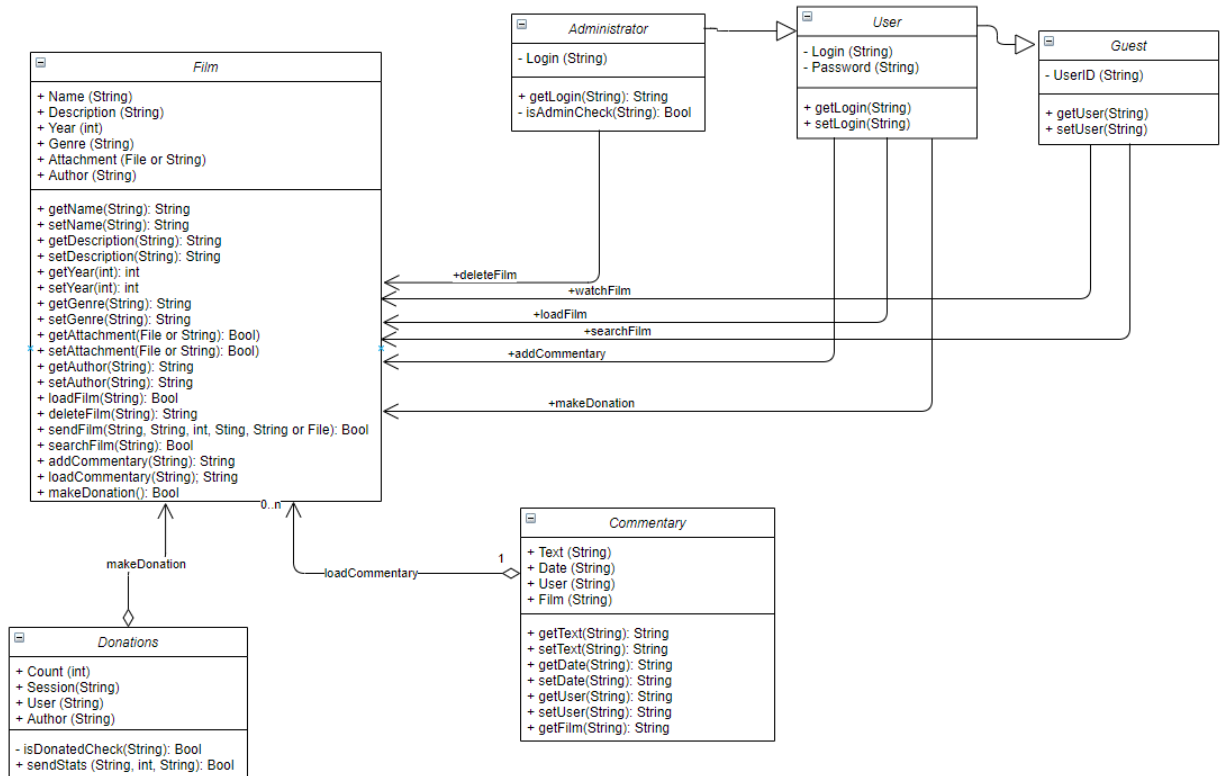


Рисунок 13 – Проектування програмних класів

## 1.3 Проектування алгоритмів роботи методів програмних класів

@startuml

title viewFilm()

start

if(Є підключення до бази даних?) then (да)

:Отримується з файлу, який здійснює пошук фільмів за критеріями id

фільму;

:Виконується запит до бази даних з отриманим id;

note right

```
SELECT * FROM film WHERE id = 16
```

end note

:Заповнення веб сторінки інформацією отриманною з бази даних;

:Відправлення веб сторінки до клієнта;

stop

else(нет);

:Виведення помилки з повідомлення, що не має можливості  
підключитися до бази даних;

stop

@enduml

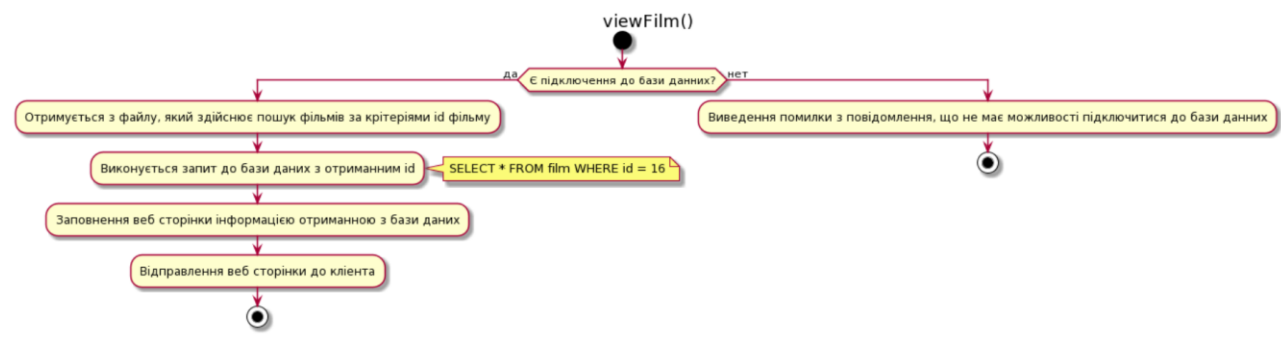


Рисунок 14 – алгоритм роботи viewFilm

@startuml

title addFilm()

start

if(Є підключення до бази даних?) then (да)

```

:Виведення на екран користувача форми для додання нового фільму;

:Отримання даних від користувача;

if(Форма для додавання фільму були пустими?) then (да)

    :Виведення сповіщення про помилку та запрошення перезавантажити
форму;

stop

else (нет)

    :Пошук фільму з таким самим ім'ям та роком виходу;

    note right

    SELECT name FROM film WHERE name = 'Мулан' and year = '2020'

    end note

    endif

    if(Фільм знайдено) then (да)

        :Виведення сповіщення про помилку та запрошення перезавантажити
форму;

        stop

        else (нет)

            :Збереження фільму до бази даних;

            note right

            INSERT INTO film VALUES(id, name, link, genre, year, description,
director, actor, poster)

            end note

            :Повідомлення про вдале додання фільму;

```

stop

endif

else (нет)

:Виведення помилки, з'єднання з базою даних;

stop

@enduml

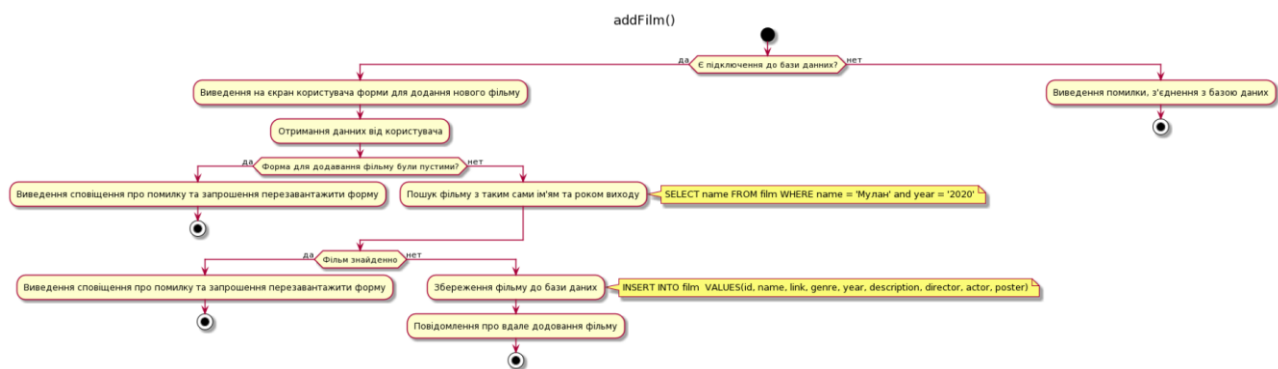


Рисунок 15 – алгоритм роботи addFilm

@startuml

title deleteFilm()

start

if(Є підключення до бази даних?) then (да)

:Взяття переданного з фронтенду значення id фільму;

:Пошук фільму в базі даних за айди;

note right

SELECT \* FROM film WHERE id = 12

end note

if(Фільм знайдено?) then (да)

:Виделення фільму з бази даних;



note right

```
DELETE FROM film WHERE id = '12'
```

end note

:Повідомлення про вдале видалення фільмів;

stop

else (нет)

:Повідомлення про те, що фільм не може бути видалений;

stop

endif

else (нет)

:Виведення помилки, з'єднання з базою даних;

stop

@enduml

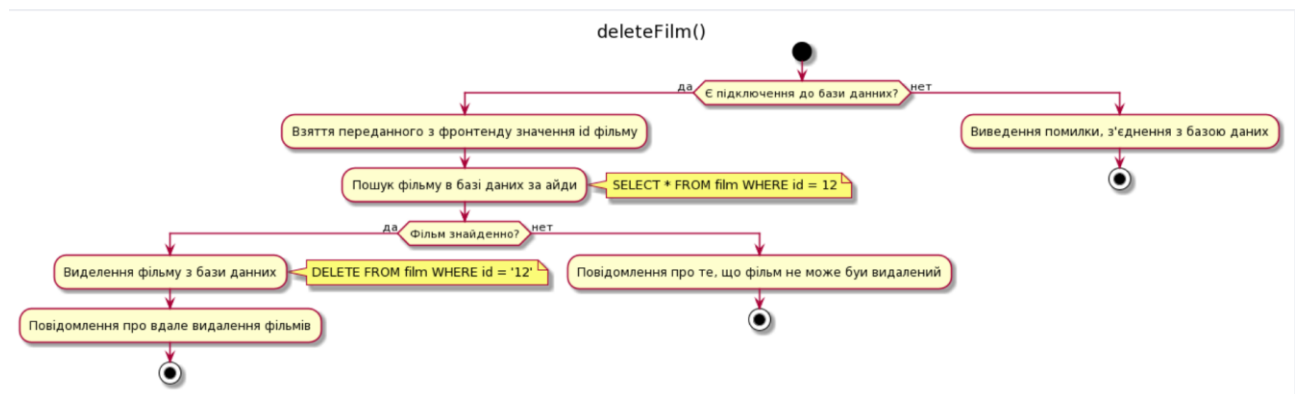


Рисунок 16 – алгоритм роботи deleteFilm



Назва	Хто виконував
Реєстрація	<u>Панібратцев О.Ю.</u>
Пошук	<u>Панібратцев О.Ю.</u>
Пожертвування	<u>Панібратцев О.Ю.</u>
Коментування	<u>Панібратцев О.Ю.</u>
Перегляд фільмів	<u>Додон Д.І.</u>
Завантаження	<u>Додон Д.І.</u>
Видалення	<u>Додон Д.І.</u>

Таблиця -15 Таблиця учасників

#### 1.4 Проектування тестових наборів методів програмних класів

##### 1. Коментування.

№ тесту	Скрипти	Опис очікуваних значень результату
1	1. text_comment = коментарий  2. Натиснути відправити	Коментар з'явився в списку під фільмом.
2	1. Поле текст пусте  2. Натиснути відправити	Сторінка перезавантажується .
3	1. Гість переходить до сторінки фільму.  2. Гість не може залишити коментар.	Повідомлення про реєстрацію.

##### 2. Пошук

№ тесту	Скріпти	Опис очікуваних значень результату
1	1. Query = Железный человек  2. Натиснути пошук.	Сторінка з фільмом.
2	1. Назва фільму якого не існує, або пусте поле.  2. Натиснути пошук.	Фільм не знайдено.

### 3. Видалення фільму

№ тесту	Скріпти	Опис очікуваних значень результату
1	1. Адмін переходить до сторінки фільму  2. Клікає видалити фільм.  3. Фільм видалено.	Сторінка перезавантажується без повідомлень.
2	1. Користувач переходить до сторінки фільму.  2. Кнопки видалити фільм немає (тест кейс завалено).	Кнопки немає.

### 4. Донат користувачам

№ тесту	Скріпти	Опис очікуваних значень результату
---------	---------	------------------------------------

1	1. Користувач переходить до сторінки фільму. 2. Вводиє суму донату до поля. 3. Натискає відправити. 4. ПП переводить до зовнішнього сервісу.	Сторінка перезавантажується без повідомлень.
2	1. Гість переходить до сторінки фільму. 2. Не має поля для донату.	Кнопки немає.

Таблиця 16– Тестові набори

Назва	Хто виконував
Реєстрація	Панібратцев О.Ю.
Пошук	Панібратцев О.Ю.
Пожертвування	Додон Д.І.
Коментування	Додон Д.І.
Перегляд фільмів	Панібратцев О.Ю.
Завантаження	Додон Д.І.
Видалення	Додон Д.І.

Таблиця 17 - Таблиця учасників

## 4 Конструювання ПП

### 4.1 Особливості конструювання структур даних

#### 4.1.1 Особливості інсталяції та роботи з СУБД

В даному ПП використовувалася СУБД версії MySQL 5.0.4

Данна СУБД була розгорнута на веб сервері XAMPP, який був встановлений з офіційного сайту компанії. В встановленні даного веб сервера не виникали ніяких проблем.

Для встановлення веб серверу потрібно скачати виконуючий файл в форматі exe та запустити встановлення серверу. Прочитавши та погоджавшись з умовами користування вибрати каталог для встановлення програмного забезпечення.

Після вдалого встановлення запустити програму та включити Apache сервер та сервер MySQL.

#### **4.1.2 Особливості створення структур даних**

Таблиця коментарі:

```
CREATE TABLE comment (  
  
    id int(11) NOT NULL PRIMARY KEY,  
  
    text_comment varchar(191),  
  
    FOREIGN KEY user_login varchar(191) REFERENCES users (login),  
  
    film_id int(11)  
  
)
```

Таблиця пожертвувачів:

```
CREATE TABLE donations (  
  
    id int(11) NOT NULL PRIMARY KEY,  
  
    film_id int(11),
```

FOREIGN KEY`login` varchar(191) REFERENCES users (login),

quantity int(11)

)

Назва	Хто виконував
Користувачі	Панібратцев О.Ю.
Фільми	Панібратцев О.Ю.
Донати	Додон Д.І.
Коментування	Додон Д.І.

Таблиця 18 - Таблиця учасників

## **4.2 Особливості конструювання програмних модулів**

### **4.2.1 Особливості роботи з інтегрованим середовищем розробки**

У своїй роботі, програмісти користуються різними редакторами коду, такими як PHPStorm від JetBrains, Eclipse і іншими "важкими" IDE, але для розробки можна використовувати і більш "легкі" рішення, наприклад, Atom або Sublime Text 3. Наш програмний продукт був написаний у Sublime Text.

У чому переваги редактора Sublime Text 3:

Ми виділили кілька корисних якостей редактора коду Sublime Text 3:

1. Швидкість і низькі вимоги до ресурсів комп'ютері. Редактор дійсно швидко працює навіть на "слабкому" залозі.

2. Робота в популярних операційних системах. Ви можете запустити редактор коду Sublime Text 3 в операційних системах Windows, Linux і Mac OS.

3. Сторонні плагіни і доповнення. Якщо вас не влаштовують базові можливості Sublime Text, ви можете встановити різні доповнення, починаючи

від красивих тем оформлення, закінчуючи зручними інструментами, наприклад, Emmet.

#### 4.2.2 Особливості створення програмної структури з урахуванням спеціалізованого фреймворку

Singup.php

```
<?php
```

```
require 'db.php';//Подключения к базе данных
```

```
$data = $_POST;
```

```
$err = "";
```

```
if (isset($_FILES['login_img']['name']) &&  
!empty($_FILES['login_img']['name'])) {
```

```
    var_dump($_FILES);
```

```
    if ($_FILES['login_img']['size'] < 2000000) { //ограничение по размеру
```

```
        if (//ограничение на формат
```

```
            $_FILES['login_img']['type'] === 'image/jpeg'
```

```
            || $_FILES['login_img']['type'] === 'image/png'
```

```
            || $_FILES['login_img']['type'] === 'image/bmp'
```

```
            || $_FILES['login_img']['type'] === 'image/jpg'
```

```
        ) { //Файлы прошли проверки
```

```
            $path = '../user_login_img/'.$data['login'].'.jpg';
```

```
            move_uploaded_file($_FILES['login_img']['tmp_name'], $path); //сохранения  
            фотки на сервере
```

```
        }
```

```

else{

                                $err .= "Не правильный формат. ";

                                }

}

else{

                                $err .= "Ваш файл не правильного размера. ";

                                }

if (!empty($err)) {

                                echo "<script>alert(\"$err\");</script>";

                                }

}else{

                                $path = '../user_login_img/stock.png';

                                }

unset($_FILES);

//если кликнули на button

if ( isset($data['do_signup']) )

{

// проверка формы на пустоту полей

$errors = array();

if ( trim($data['login']) == " )

{

                                $errors[] = 'Введите логин';

```



```
}
```

```
if ( trim($data['email']) == " )
```

```
{
```

```
    $errors[] = 'Введите Email';
```

```
}
```

```
if ( $data['password'] == " )
```

```
{
```

```
    $errors[] = 'Введите пароль';
```

```
}
```

```
if ( $data['password_2'] != $data['password'] )
```

```
{
```

```
    $errors[] = 'Повторный пароль введен не верно!';
```

```
}
```

```
//проверка на существование одинакового логина
```

```
if ( R::count('users', "login = ?", array($data['login'])) > 0)
```

```
{
```

```
    $errors[] = 'Пользователь с таким логином уже существует!';
```

```
}
```

```

//проверка на существование одинакового email

if ( R::count('users', "email = ?", array($data['email'])) > 0)

{

    $errors[] = 'Пользователь с таким Email уже существует!';

}

if ( empty($errors) && empty($err) )

{

    //ошибок нет, теперь регистрируем

    $user = R::dispense('users');

    $user->avatar = $path;

    $user->login = $data['login'];

    $user->email = $data['email'];

    $user->age = $data['age'];

    $user->password = $data['password'];

    R::store($user); //загрузка записи в базу данных

    echo "<script>alert(\"Вы успешно зарегистрировались!\");
location=\"index.php\";</script>";

    $_SESSION['logged_user'] = $user;

    header("Location:../index.php");

}else

{

    echo '<div id="errors" style="color:red;">' . array_shift($errors).
'</div><hr>';

```

```

    }

}

?>

Search.php

<?php

    require_once 'db.php';//подключение к базе данных

    $filmName = $_GET['filmSearch'];

    $films = R::find('film', 'name = ?', [$filmName]);//выбираем все
фильмы по даному имени

    if(!empty($films)){//проверка на то что фильмы или фильм
возвращается и базы

        $newId = [];//массив который будет хранить в себе айдишки
фильмов

        foreach ($films as $key => $value) {//переписываем айдишки в
массив

            $newId[$key] = $value['id'];

        }

        $type = 'type=search';//задаем тип передаваемого на show.php

        $id = "";

        foreach ($newId as $value) {

            $id .= "&id[]=". $value;

        }

```

```

$url = $type.$id;

header("Location:show.php?$url");//посылание гет запроса

}else{

    echo "<div class=\"validation\">Фильма с таким названием не
    найдено</div>";

}

?>

```

Donation.php

```

require_once 'db.php';//подключение к базе данных

$donation = R::dispense('donations');

// Заполняем объект свойствами

$donation->filmId = $_GET['id'];

$donation->login = $_GET['login'];

$donation->quantity = $_GET['quantity'];

R::store($donation);//добавление записи в базу данных

<?php

    }else{ ?>

        <script>

            alert("Вы ввели не корректные данные. Попробуйте снова.");

            location = "view.php?id=<?php echo $_GET['id']; ?>";

        </script>;

```

<?php

} ?>

Назва	Хто виконував
Реєстрація	Додон Д.І.
Пошук	Додон Д.І.
Пожертвування	Додон Д.І.
Коментування	Панібратцев О.Ю.
Перегляд	Панібратцев О.Ю.
Завантаження	Панібратцев О.Ю.
Видалення	Панібратцев О.Ю.

Таблиця 19 - Таблиця учасників

#### **4.2.3 Особливості створення програмних модулів**

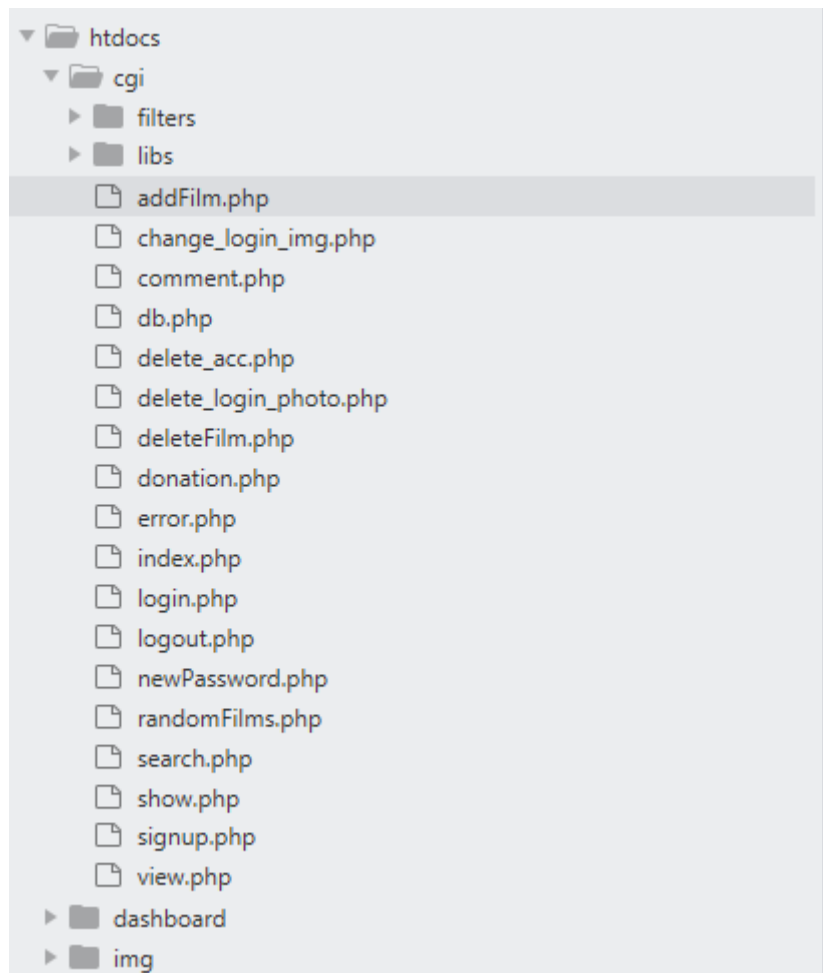


Рисунок 17 - Файлова структура

При розробці сайту було використано Apache HTTP-сервер для його відладки.

Apache HTTP-сервер — відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем.

Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation.

#### **4.2.4 Особливості розробки алгоритмів методів програмних класів або процедур/функцій**

Comment.php

<?php

```
require_once 'db.php';// подключаемся к базе данных
```

```
// Указываем, что будем работать с таблицей comment
```

```
$comment = R::dispense('comment');
```

```
$err = "";
```

```
if(trim($_POST['text_comment']) != ""){//проверка на пустоту  
комментария
```

```
    $com = htmlspecialchars($_POST['text_comment']);
```

```
    @$comment['text_comment'] = $com;
```

```
}else{
```

```
    $err .= "comment is empty";
```

```
}
```

```
$login = htmlspecialchars($_POST['user_login']);
```

```
$filmId = htmlspecialchars($_POST['film_id']);
```

```
if ($_POST['rating'] != 0) {//проверку на то отправлен ли рейтинг
```

```
    $rating = htmlspecialchars($_POST['rating']);
```

```
    @$coment['rating'] = $rating;
```

```

    }else{

        $err .= "rating is empty";

    }

    // Заполняем объект свойствами которые передаются
автоматически

    if ($err == "comment is empty" && $err != "rating is empty") {//если
пустой комментарий но не пустой рейтинг то грузим рейтинг

        $film = R::load('film' , $filmId);

        $film->rating = $film->rating+$rating;

        $film->voises++;

        R::store($film);//добавляем рейтинг в базу данных

    }

    if ($err != "comment is empty" && $err == "rating is empty"){//если
пустой рейтинг но не пустой комментарий

        @$comment['user_login'] = $login;

        @$comment['film_id'] = $filmId;

        R::store($comment);//добавляем комментарий в базу данных

    }

    header("Location: ".$_SERVER["HTTP_REFERER"]);// Делаем
перидект обратно

?>

View.php

<?php

    require_once 'db.php';//подключение к базе данных

```



```

$id = $_GET['id'];

$film = R::findOne('film', 'id = ?', [$id]); //поиск фильма по айди

$poster = $film['poster']; //объявление переменных и
присваивание им значений из ответа из базы данных

$name = $film['name'];

$link = $film['link'];

$description = $film['description'];

$rating = $film['rating'];

$voises = $film['voises'];

$owner = $film['owner'];

$director = $film['director'];

$year = $film['year'];

$country = $film['country'];

$genre = $film['genre'];

?>

<table class="filmTable" align="center">

    <!-- Постер -->

    <tr>

        <td align="center">

            " width="189"
height="255" alt="lorem">

            <p>Рейтинг: <?php echo round($rating/$voises, 2);
?></p>

```

```

        </td>

    </tr>

    <tr>

        <td>

            <!-- Имя фильма -->

            <h2><?php echo "$name"; ?></h2>

            <?php

                if (!empty($_SESSION['logged_user'])) {

                    $login = $_SESSION['logged_user']->login;

                    if ($login === 'admin') {

                        echo "<a href=\"deleteFilm.php?id=" .

$хid . "\">Delete this film</a>";

                    }

                }

            ?>

        </td>

    </tr>

    <tr>

        <td>

            <!-- описание -->

            <p><?php echo "$description"; ?></p>

```

```

        </td>

</tr>

<tr>

    <td><!-- режисер -->

        <?php

            echo "Режисер : $director";

        ?>

    </td>

</tr>

<tr><!-- год -->

    <td>

        <?php

            echo "Год : $year";

        ?>

    </td>

</tr>

<tr>

    <td>

</tr>

    <td><!-- страна -->

        <?php

            echo "Страна : $country";

        ?>

```

```

        </td>

    </tr>

    <tr><!-- жанр -->

        <td>

            <?php

                echo "Жанр : $genre";

            ?>

        </td>

    </tr>

    <tr>

        <td>

            <!-- Видео -->

            <div >

                <iframe src="<?php echo "$link"; ?>"

align="center" width="100%" height="100%" frameborder="0"

allowfullscreen></iframe>

            </div>

        </td>

    </tr>

    <tr>

        <td align="center">

            <?php //ВЫВОД КОММЕНТАРИЕВ

```

ЛОГИНЫ ЮЗЕРОВ

```
$filmId = $id;
```

```
$comments = R::find('comment', 'film_id = ?', [$filmId]);
```

```
$login = []; //создем массив в ктором будем хранить
```

```
foreach ($comments as $key => $value) {
```

```
    $login[$key] = $comments[$key]['user_login'];
```

```
}
```

```
$img = [];
```

базы данных

```
foreach ($login as $key => $login) { //берутся аватарки из
```

```
    $users = R::findOne('users', 'login = ?', [$login]);
```

```
    $img[$key] = $users['avatar'];
```

```
}
```

```
if (!empty($comments)) {
```

\<comment\>"

```
    echo "<table width = \"100%\" border = \"1\" class =
```

```
    foreach ($comments as $key => $value) {
```

```
        if($comments[$key]['text_comment'] != ' '){
```

```

        echo "<tr>";

        if(file_exists($img[$key])){

            echo "<td>" . "". "<br><br>" . $comments[$key]['user_login'] .
"</td>";

        }

```

```

        echo "<td>" .
$comments[$key]['text_comment'] . "</td>";

        echo "</tr>";

    }

```

```

    }

    echo "</table>";

} else

```

```

    unset($filmId);

?>

```

```

<?php

if (isset($_SESSION['logged_user'])) {

    $login = $_SESSION['logged_user']['login'];

```

```

?>

```

```

<!-- Комментарии -->

<form name="comment" action="comment.php" method="post">

<p>

<label>Комментарировать:</label>

<br />

<textarea name="text_comment" placeholder="Оставить
комментарий" id="comment" cols="40" rows="7"></textarea><!-- берем текст
комментария -->

<br><br>

</p>

<input type="hidden" name="user_login" value= <?php
echo "\"".$login."\""; ?>><!-- берем логи пользователя -->

<input type="hidden" name="film_id" value= <?php echo
"\"".$id."\""; ?> ><!-- берем айди фильма -->

<p><label>Выберите оценку:</label></p>

<select name="rating">

<option disabled>Выберите оценку</option>

<option value="0">0</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

```

```

        <option value="7">7</option>

        <option value="8">8</option>

        <option value="9">9</option>

        <option value="10">10</option>

    </select><br><br>

    <input type="submit" value="Отправить" >

</form>

<tr>

    <td>

        <label>Пожертвовать автору:</label>

        <form method="GET" action="donation.php">

            <input type="text" required=""

name="quantity" placeholder="Количество">

            <input type="hidden" name="id" value=?php
echo $id; ?>>

            <input type="hidden" name="login"

value=?php echo $login; ?>>

            <input type="submit" name="btn">

        </form>

    </td>

</tr>

<?php

```



```

    }else{

        echo "<div class=\"info\">Если вы хотите оставить
комментарий пожалуйста <a href=\"index.php\">зарегистрируйтесь</a></div>";

    }

?>

</td>

</tr>

</table>

```

| Назва         | Хто виконував    |
|---------------|------------------|
| Реєстрація    | Панібратцев О.Ю. |
| Пошук         | Панібратцев О.Ю. |
| Пожертвування | Панібратцев О.Ю. |
| Коментування  | Додон Д.І.       |
| Перегляд      | Додон Д.І.       |
| Завантаження  | Додон Д.І.       |
| Видалення     | Додон Д.І.       |

Таблиця 20 - Таблиця учасників

### 4.3 Тестування програмних модулів

Тестування модуля реєстрації:

Feature: Guest can register

Scenario:

Given User can access main page "http://lopik.zapto.org/"

Then User click at cabinet button

Then User click at register button

Then user enter login

Then user click register button

And Validation

And user enter email "lexa.pan2012@gmail.com"

And user enter age "21"

And user delete login

Then user click register button

And Validation

Then user enter login

And user enter passwordd

Then user click register button

And Validation

And user enter password\_2

Then user click register button

Then User click at cabinet button

Given user can see his username

Тестування модулю пошуку:

Feature: User can search

Scenario:

Given User can access main page "http://lopik.zapto.org/"

Then user click on the filter button

Then user enter search query

And user click search

Given searched results appeared

Тестування модулю пожертвувань:

Feature: User can donate

Scenario:

Given User can access main page "http://lopik.zapto.org/"

Then User click at cabinet button

And user click to authorization

Then user enter login "admin"

And user enter passwordr

Then click Log In button

Then user click at any page of film

And enter count of payment

Then click submit

Then click pay

Report generated 03-12-2020 02:48

User Can Donate

✓










| Steps                                                       | Screenshots                                                                       | Outcome | ⌚     |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------|---------|-------|
| ✓ Given User can access main page "http://lopik.zapto.org/" |  | SUCCESS | 2.69s |
| ✓ Then User click at cabinet button                         |  | SUCCESS | 0.46s |
| ✓ And user click to authorization                           |  | SUCCESS | 0.45s |
| ✓ Then user enter login "admin"                             |  | SUCCESS | 0.38s |
| ✓ And user enter passwordr                                  |  | SUCCESS | 0.37s |
| ✓ Then click Log In button                                  |  | SUCCESS | 0.76s |
| ✓ Then user click at any page of film                       |  | SUCCESS | 0.51s |
| ✓ And enter count of payment                                |  | SUCCESS | 0.39s |
| ✓ Then click button                                         |  | SUCCESS | 0.41s |

Рисунок 18 – Результат тестування модуля пожертвувань

User Can Search

✓








| Steps                                                       | Screenshots                                                                           | Outcome | ⌚     |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------|---------|-------|
| ✓ Given User can access main page "http://lopik.zapto.org/" |  | SUCCESS | 2.77s |
| ✓ Then user click on the filter button                      |  | SUCCESS | 0.65s |
| ✓ Then user enter search query                              |  | SUCCESS | 0.7s  |
| ✓ And user click search                                     |  | SUCCESS | 0.47s |
| ✓ Given searched results appeared                           |  | SUCCESS | 0.03s |
|                                                             |                                                                                       | SUCCESS | 4.64s |

Рисунок 19 – Результат тестування модуля пошуку



Guest Can Register





















| Steps                                                                                                                                                               | Screenshots                                                                         | Outcome | ⌚     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------|-------|
| <div>  <div>Given User can access main page "http://lopik.zapto.org/"</div> </div> |  | SUCCESS | 3.88s |
| <div>  <div>Then User click at cabinet button</div> </div>                         |  | SUCCESS | 0.44s |
| <div>  <div>Then User click at register button</div> </div>                        |  | SUCCESS | 0.48s |
| <div>  <div>Then user enter login</div> </div>                                     |  | SUCCESS | 0.38s |
| <div>  <div>Then user click register button</div> </div>                           |  | SUCCESS | 0.54s |
| <div>  <div>And Validation message appeared</div> </div>                           |  | SUCCESS | 5.29s |
| <div>  <div>And user enter email "lexa.pan2012@gmail.com"</div> </div>             |  | SUCCESS | 0.36s |
| <div>  <div>And user enter age "21"</div> </div>                                   |  | SUCCESS | 0.38s |
| <div>  <div>And user click login</div> </div>                                      |  | SUCCESS | 0.41s |

Рисунок 20 – Результат тестування модуля реєстрації

| Назва         | Хто виконував    |
|---------------|------------------|
| Реєстрація    | Додон Д.І.       |
| Пошук         | Додон Д.І.       |
| Пожертвування | Додон Д.І.       |
| Коментування  | Панібратцев О.Ю. |
| Перегляд      | Панібратцев О.Ю. |
| Завантаження  | Панібратцев О.Ю. |
| Видалення     | Панібратцев О.Ю. |

Таблиця 21 - Таблиця учасників

## 5 Розгортання та валідація ПП

### 5.1 Інструкція з встановлення ПП

Кроки встановлення ПП:

1) Клонувати репозиторій

```
git clone https://github.com/elxaz/kursa4AI/tree/master/project-template
```

2) Зберегти данні з репозиторію в кореневу папку веб сервера

3) Запустити веб сервер

4) На веб сервері створити базу даних з назвою ICS-media

5) Перейти на розташування веб серверу (домєне ім'я:порт)

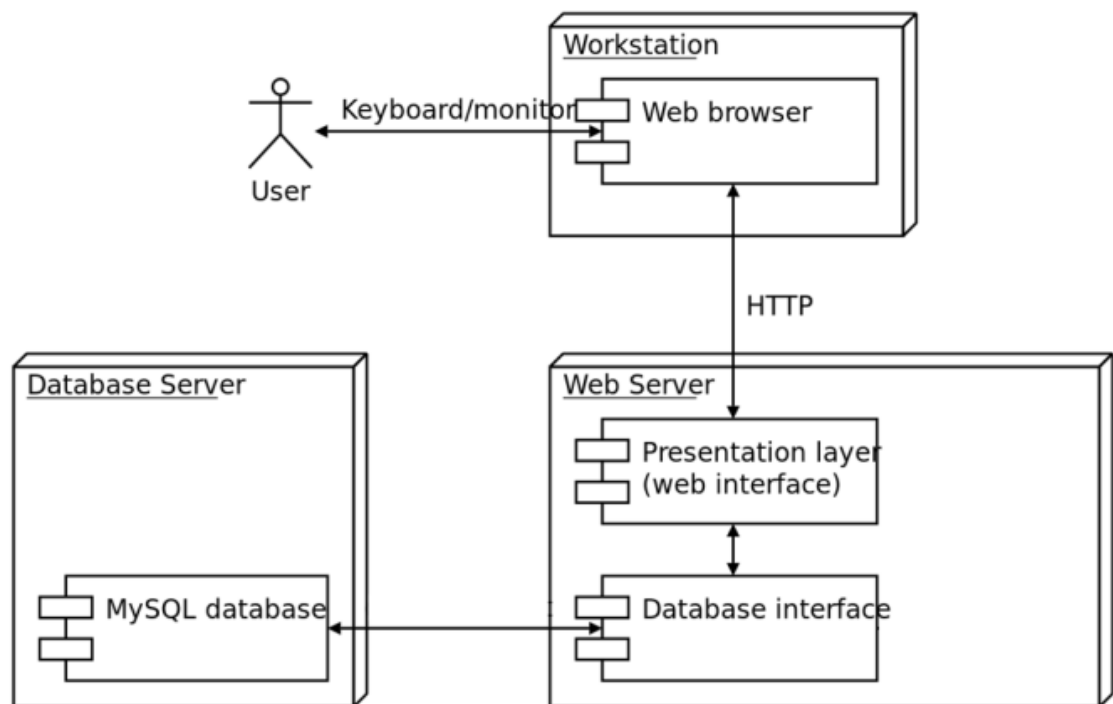
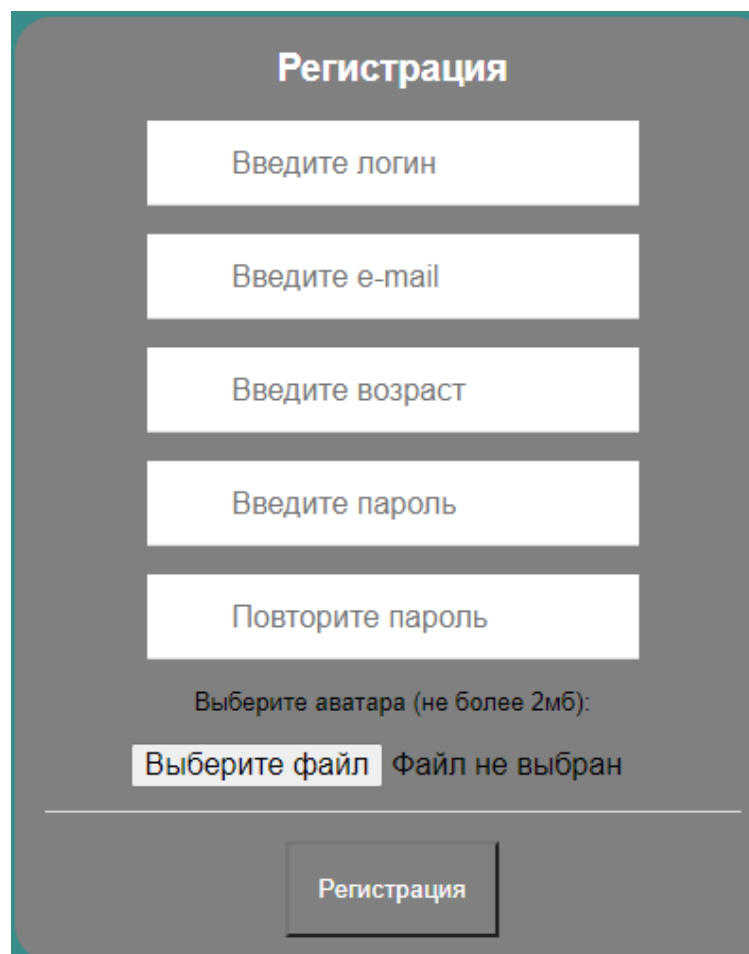


Рис. 21 – Діаграма розгортання

## 5.2 Інструкція з використання ПП



The image shows a registration form titled "Регистрация" (Registration) in Russian. The form is set against a dark gray background with teal-colored rounded corners. It contains several input fields for user information: login, email, age, password, and password confirmation. Below these is a section for selecting an avatar, with a file selection button and a status indicator. A "Регистрация" button is located at the bottom of the form.

**Регистрация**

Введите логин

Введите e-mail

Введите возраст

Введите пароль

Повторите пароль

Выберите аватара (не более 2мб):

Выберите файл    Файл не выбран

Регистрация

Рисунок 22 – Регестрація

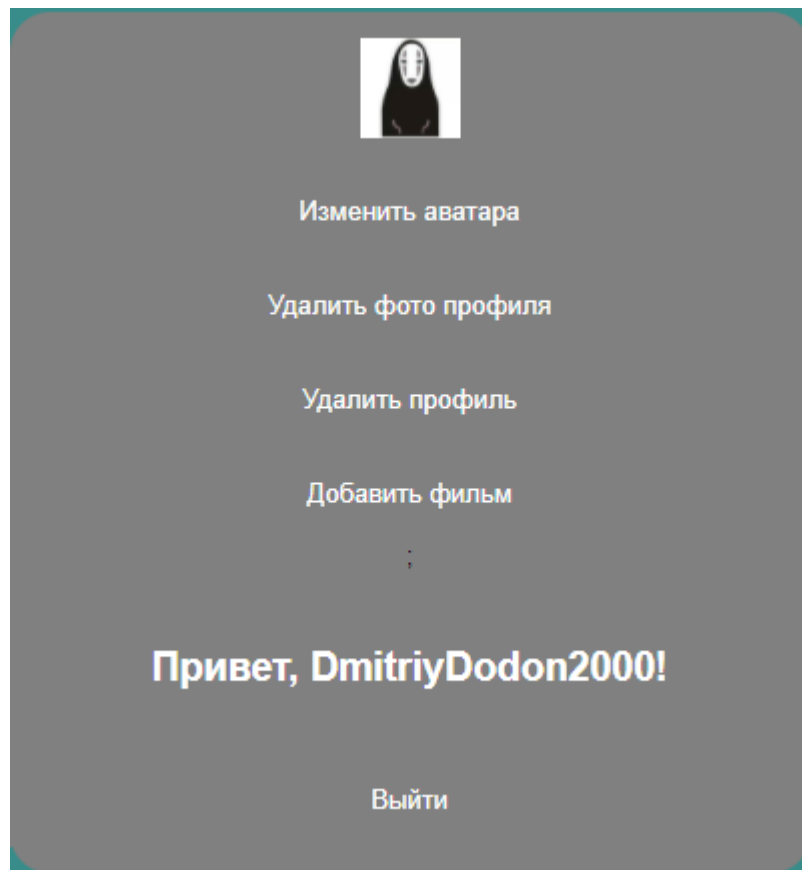





Рисунок 23 – Результат вдалої реєстрації





Фильтры






Рейтинг: 0


## Мулан

События мультика происходят много лет тому назад. В центре сюжета молодая девушка по имени Мулан, которая живет обычной жизнью в компании своих родственников. У нее также есть верный спутник - маленький огнедышащий дракон, который никогда не покидает Мулан в трудную минуту. Но однажды для всего народа наступили темные времена. Все дело в том, что многотысячные армии гуннов решили совершить неожиданное нападение, чтобы захватить столь желанную для себя территорию. По этой причине все представители мужского рода отправились на фронт, чтобы защищать народ от вражеских войск.

Режисер : Тони Бэнкрофт, Бэрри Кук  
 Год : 1998  
 Страна : США  
 Жанр : мультфильм



|                                                                                     |             |
|-------------------------------------------------------------------------------------|-------------|
|  | Мулан       |
| DmitriyDodon2000                                                                    |             |
|  | Комментарий |
| DmitriyDodon2000                                                                    |             |



Если вы хотите оставить комментарий, пожалуйста, [зарегистрируйтесь](#)



© ICS-media 2020
 


Рисунок 24 – Перегляд фільму в ролі гостя



Фильтры




Рейтинг: 0

## Мулан

[Delete this film](#)

События мультика происходят много лет тому назад. В центре сюжета молодая девушка по имени Мулан, которая живет обычной жизнью в компании своих родственников. У нее также есть верный спутник - маленький огнедышащий дракон, который никогда не покидает Мулан в трудную минуту. Но однажды для всего народа наступили темные времена. Все дело в том, что многотысячные армии гуннов решили совершить неожиданное нападение, чтобы захватить столь желанную для себя территорию. По этой причине все представители мужского рода отправились на фронт, чтобы защищать народ от вражеских войск.

Режисер : Тони Бэнкрофт, Бэрри Кук  
 Год : 1998  
 Страна : США  
 Жанр : мультфильм



|                                                                                                        |             |
|--------------------------------------------------------------------------------------------------------|-------------|
| <br>DmitryDodon2000 | Мулан       |
| <br>DmitryDodon2000 | Комментарий |

Комментировать:

Оставить комментарий

Выберите оценку:

0

Отправить

Пожертвовать автору:

Количество

© ICS-media 2020




Рисунок 25 – Перегляд фільму в ролі зареєстрованого користувача

### Добавления фильма

!

Заполните это поле.

Рисунок 26 – Неправильні дані в додаванні фільму

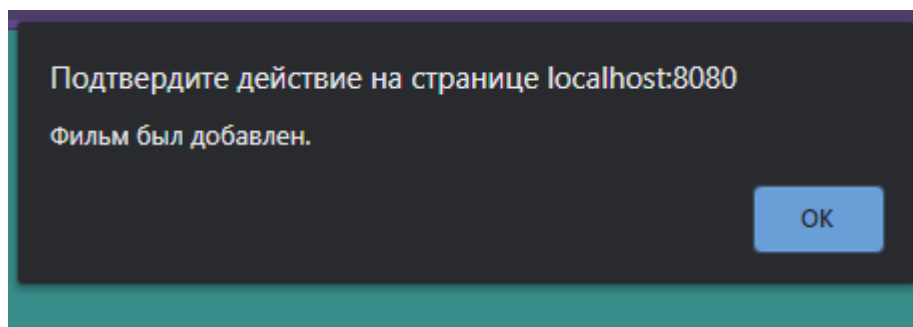


Рисунок 27 – Вдале додавання фільму

### 5.3 Результати валідації ПП

Метою створення ПП збільшити кількість матеріальної підтримки для молодих режисерів.

Рівень користності для режисерів CL можна визначити, як

$$CL = NA / N,$$

NA – кількість просмотрів;

N – кількість пожертвувань

До введення ПП цей показник складав ~4%. Після введення ПП цей показник становить ~15%. На основі цієї можна сказати, що ПП пройшов валідацію й виконує свою мету.

## **Висновки**

В результаті було створено програмний продукт, який допоможе молодим та амбіційним режисерам ділитися своїми фільмами зі всім світом, а звичайним глядачам знаходити нові, цікаві, авторські фільми.

В процесі створення програмного продукту виникли такі труднощі:

- 1) обмеженість в часі;
- 2) складність розробки ПП;
- 3) недостатні знання де-яких інструментів.

### **Джерела:**

1. <https://www.php.net/>
2. <http://al.tabfil.me/>
3. <https://prowebmastering.ru/redbeanphp-orm-dlya-php.html>
4. <http://htmlbook.ru/>
5. <http://htmlbook.ru/samcss>