

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 Друга

Виконав:  
студент 3-го курсу  
групи ЕАІ-185  
Панібратцев О.Ю.  
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](#)

## ЗМІСТ

1	Вимоги до програмного продукту .....	6
1.1	Визначення потреб споживача .....	6
1.1.1	Ієрархія потреб споживача .....	6
1.1.2	Деталізація матеріальної потреби .....	7
1.2	Бізнес-вимоги до програмного продукту.....	7
1.2.1	Опис проблеми споживача .....	7
1.2.1.1	Концептуальний опис проблеми споживача.....	7
1.2.1.2	Метричний опис проблеми споживача .....	8
1.2.2	Мета створення програмного продукту .....	8
1.2.2.1	Проблемний аналіз існуючих програмних продуктів .....	8
1.2.2.2	Мета створення програмного продукту .....	8
1.2.3	Назва програмного продукту .....	9
1.2.3.1	Гасло програмного продукту .....	9
1.2.3.2	Логотип програмного продукту.....	10
1.3	Вимоги користувача до програмного продукту .....	10
1.3.1	Історія користувача програмного продукту .....	10
1.3.2	Діаграма прецедентів програмного продукту .....	11
1.3.3	Сценаріїв використання прецедентів програмного продукту.....	11
1.4	Функціональні вимоги до програмного продукту.....	16
1.4.1.	Багаторівнева класифікація функціональних вимог .....	16
1.4.2	Функціональний аналіз існуючих програмних продуктів .....	19
1.5	Нефункціональні вимоги до програмного продукту.....	19

					ІС КР 122 ЕАІ185 ПЗ			
Змі	Арк.	№ докум.	Підпис	Дата				
Виконав	Панібратцев О.Ю.				«ICS – media – онлайн кінотеатр» Частина Друга	Літ.	Лист	Листів
Перев.	Блажко О.А.						3	68
Реценз.						ОНПУ, каф. ІС, г. ЕАІ-185		
Н. Контр.								
Утверд.								

1.5.1	Опис зовнішніх інтерфейсів.....	19
1.5.1.1	Опис інтерфейса користувача .....	19
1.5.1.1.1	Опис INPUT-інтерфейса користувача .....	19
1.5.1.1.2	Опис OUTPUT-інтерфейса користувача .....	21
1.5.1.2	Опис інтерфейсу із зовнішніми пристроями.....	23
1.5.1.3	Опис програмних інтерфейсів .....	24
1.5.1.4	Опис інтерфейсів передачі інформації .....	24
1.5.1.5	Опис атрибутів продуктивності.....	24
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 програмного продукту .....	27
2.3.1	Оцінка трудомісткості розробки програмного продукту .....	27
2.3.2	Визначення дерева робіт з розробки програмного продукту .....	30
2.3.3	Графік робіт з розробки програмного продукту .....	33
3	Проектування програмного продукту .....	34
3.1	Концептуальне та логічне проектування структур даних програмного продукту. ....	34
3.1.1	Концептуальне проектування на основі UML-діаграми концептуальних класів. ....	34
3.1.2	Логічне проектування структур даних. ....	34
3.2	Проектування програмних класів .....	35
3.3	Проектування алгоритмів роботи методів програмних класів .....	35
3.4	Проектування тестових наборів методів програмних класів .....	40
4	Конструювання програмного продукту .....	42

4.1 Особливості конструювання структур даних .....	42
4.1.1 Особливості інсталяції та роботи з СУБД .....	42
4.1.2 Особливості створення структур даних .....	43
4.2 Особливості конструювання програмних модулів .....	44
4.2.1 Особливості роботи з інтегрованим середовищем розробки .....	44
4.2.2 Особливості створення програмної структури з урахуванням спеціалізованого Фреймворку .....	44
4.2.3 Особливості створення програмних модулів .....	47
4.2.4 Особливості розробки алгоритмів методів програмних класів або процедур/функцій .....	48
5 Розгортання та валідація програмного продукту .....	62
5.1 Інструкція з встановлення програмного продукту .....	62
5.2 Інструкція з використання програмного продукту .....	63
5.3 Результати валідації програмного продукту .....	66
Висновок .....	68

# 1 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

## 1.1 Визначення потреб споживача

### 1.1.1 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. 1.1 – Ієрархія потреб споживача нашого ПП.

### 1.1.2 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.

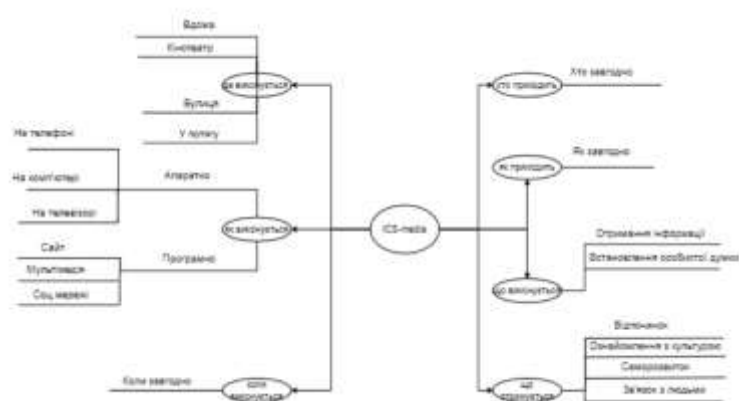


Рис. 1.2 – Деталізація матеріальних потреб.

## 1.2 Бізнес-вимоги до програмного продукту

### 1.2.1 Опис проблеми споживача

#### 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 загрузати фільми особистого виробництва з можливістю виплати виробнику.

#### 1.2.1.2 Метричний опис проблеми споживача

Метричні показники незадоволеності споживача: виробники не можуть загрузати свої творіння на окремий сервіс та отримувати гроші з можливості користувачів платити за фільм творця.

Рівень користності для виробників CL можна визначити як

$$CL = NA / N,$$

NA – кількість просмотрів;

N – кількість донатів.

#### 1.2.2 Мета створення програмного продукту

##### 1.2.2.1 Проблемний аналіз існуючих програмних продуктів

На таблиці 1.1 вказано аналіз існуючих програмних продуктів.

##### 1.2.2.2 Мета створення програмного продукту

Надати можливість загрузати свої фільми людям які бажають отримувати винагороду.

					ІС КР 122 ЕАІ185 ПЗ	АР
						8
Зм.	Арк.	№ докум.	Підп.	ДА		



Таблиця 1.1 – Аналіз продуктів.

№	Назва продукту	Вартість	Ступінь готовності	Примітка
1	ivi.ru	Безкоштовно/ платно	4	Немає знижок для студентів. Більшість фільмів платні.
2	film.ru	Безкоштовно	2	Працює не стабільно. Використовує сторонні ресурси.
3	gidonline.io	Безкоштовно	3	Багато реклами. Часто фільми у поганому якості.
4	okko.tv	Безкоштовно/ платно	4	Не доступно в Україні.
5	baskino.me	Безкоштовно	2	Багато реклами.

## 1.2.3 Назва програмного продукту

## 1.2.3.1 Гасло програмного продукту

«Загружайте, дивіться!» - ICS - media.

### 1.2.3.2 Логотип програмного продукту



Рис. 1.3 – Логотип ICS – media.

## 1.3 Вимоги користувача до програмного продукту

### 1.3.1 Історія користувача програмного продукту

- Як гість, я можу зареєструватися.
- Як гість, я можу включити будь-який фільм з каталогу.
- Як гість, я можу сортувати каталог.
- 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.

					ІС КР 122 ЕАІ185 ПЗ	АР
						10
Зм.	Арк.	№ докум.	Підп.	ДА		

### 1.3.2 Діаграма прецедентів програмного продукту

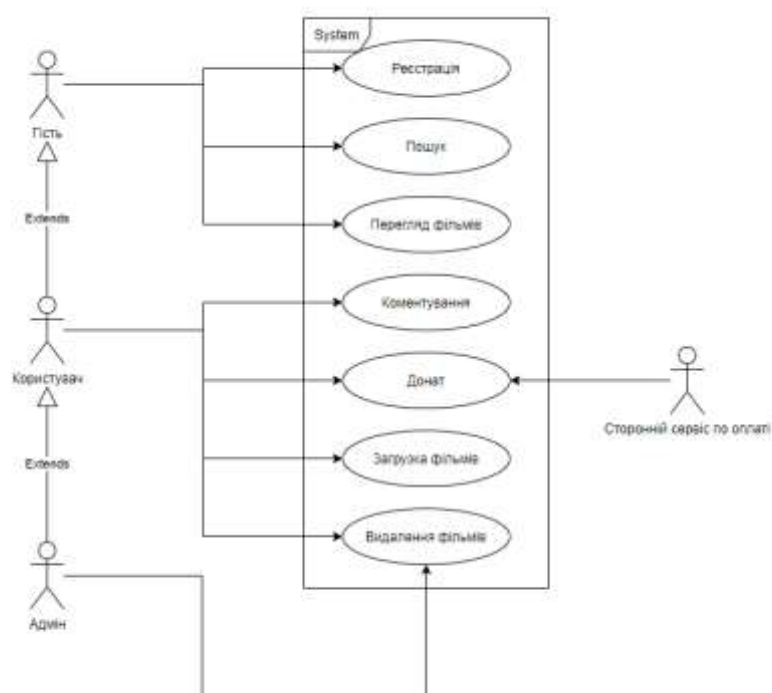


Рис. 1.3 – Діаграма прецедентів.

### 1.3.3 Сценаріїв використання прецедентів програмного продукту

- Реєстрація користувача.

Передумова: Користувач 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.

- 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 запусився, 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.

					ІС КР 122 ЕАІ185 ПЗ	АР
						12
Зм.	Арк.	№ докум.	Підп.	ДА		

- 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.

					ІС КР 122 ЕАІ185 ПЗ	АР
						13
Зм.	Арк.	№ докум.	Підп.	ДА		

- 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.

					ІС КР 122 ЕАІ185 ПЗ	АР
						14
Зм.	Арк.	№ докум.	Підп.	ДА		

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

					ІС КР 122 ЕАІ185 ПЗ	АР
						15
Зм.	Арк.	№ докум.	Підп.	ДА		

- 1.3 ПП дає можливість видалити фільм;
- 1.4 Адміністратор клікає на кнопку видалення фільму;
- 1.5 Повідомлення що фільм видалено.

- 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 видає 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.

#### 1.4 Функціональні вимоги до програмного продукту

##### 1.4.1. Багаторівнева класифікація функціональних вимог

					ІС КР 122 ЕАІ185 ПЗ	АР
						16
Зм.	Арк.	№ докум.	Підп.	ДА		



Таблиця 1.2 – Класифікація функцій.

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 коректно 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.

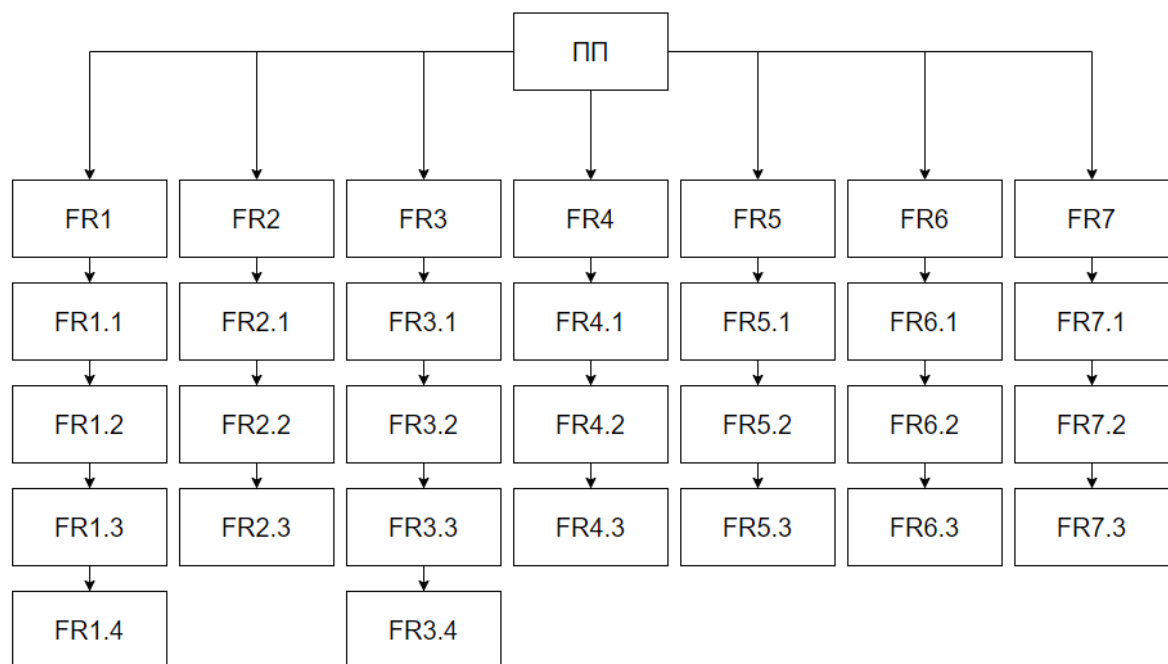


Рис 1.4 – Дерево ієрархій функцій.

#### 1.4.2 Функціональний аналіз існуючих програмних продуктів

Таблиця 1.3 – Таблиця аналізу існуючих програмних продуктів.

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

#### 1.5 Нефункціональні вимоги до програмного продукту

##### 1.5.1 Опис зовнішніх інтерфейсів

##### 1.5.1.1 Опис інтерфейса користувача

##### 1.5.1.1.1 Опис INPUT-інтерфейса користувача

Таблиця 1.4 – Опис INPUT.

Ідентифікатор функції	Засіб INPUT-потoku	Особливості використання
FR1.1	Комп'ютерна	Використання кнопок

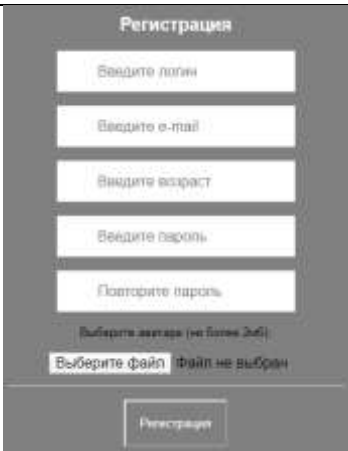
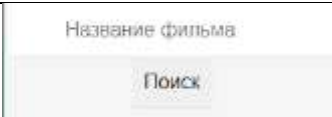
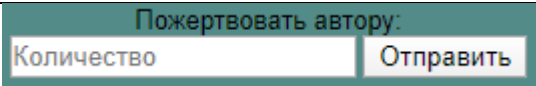
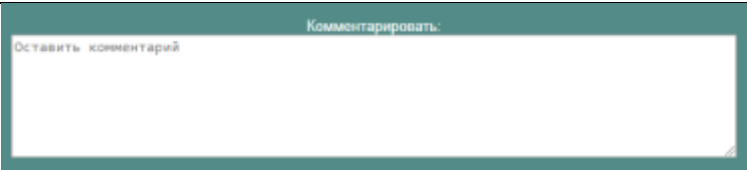

FR1.2	клавіатура; маніпулятор типу «миша»; сенсорний екран;	миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення даних про користувача.
FR1.3		
FR1.4		
FR2.1		
FR2.2		Використання кнопок миші для навігації по сторінці та клікання по кнопках та віртуальної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


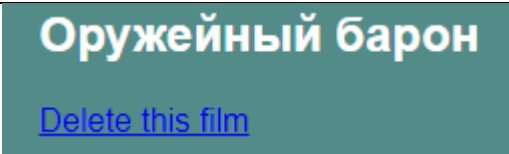
		коментаря
FR5.1		Використання кнопок миші для запуску фільма.
FR5.2		
FR5.3		
FR6.1		Використання кнопок миші для навігації по сторінці та клікання по кнопках та віртуальної чи фізичної клавіатури для введення даних про фільм.
FR6.2		
FR6.3		
FR7.1		Використання кнопок миші для навігації по сторінці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

#### 1.5.1.1.2 Опис OUTPUT-інтерфейса користувача

OUTPUT – потік у нашому ПП це вікно браузера.

Таблиця 1.5 – Макети.

Ідентифікатор функції	OUTPUT
FR1	 <p>Registration form mockup with fields: Введіть логін, Введіть e-mail, Введіть вік, Введіть пароль, Повторіть пароль, and a file upload section with buttons: Виберіть файл, Файл не вибран, and a Register button.</p>
FR2	 <p>Movie search form mockup with a text input field: Название фильма, and a Search button: Поиск.</p>
FR3	 <p>Donate form mockup with a header: Пожертвовать автору:, a text input field: Количество, and a button: Отправить.</p>
FR4	 <p>Comment form mockup with a header: Комментировать:, and a text area: Оставить комментарий.</p>
FR5	 <p>Video player mockup showing a video frame with a volume control bar at 100%: Громкость: 100%.</p>

FR6	
FR7	

### 1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

Таблица 1.6 – Опис зовнішніх інтерфейсів.

Ідентифікатор функції	Зовнішній пристрій
FR1	Смартфон, планшет, Desktop-персональний комп'ютер, laptop;
FR2	
FR3	
FR4	
FR5	
FR6	
FR7	

### 1.5.1.3 Опис програмних інтерфейсів

Таблиця 1.7 – Версії програмних інтерфейсів.

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><li>- Сервіс по оплаті – Intercassa.</li><li>- Бібліотека RedBean для роботи з БД.</li></ul>
--	--

### 1.5.1.4 Опис інтерфейсів передачі інформації

Таблиця 1.8 – Версії програмних інтерфейсів.

Опис інтерфейсів передачі інформації, які знадобляться при реалізації більшості функцій ПП.	<ul style="list-style-type: none"><li>- Провідні інтерфейси: Ethernet;</li><li>- Безпроводні інтерфейси: Wi-Fi;</li></ul>
---	---

### 1.5.1.5 Опис атрибутів продуктивності

Таблиця 1.11 – Опис атрибутів продуктивності.

Ідентифікатор функції	Максимальний час реакції ПП на дії користувачів (секунди)
FR1	15



FR2	15
FR3	30
FR4	10
FR5	50
FR6	20
FR7	10

## 2 ПЛАНУВАННЯ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

### 2.1 Планування ітерацій розробки програмного продукту

Таблиця 2.1 – Ітерації розробки.

Ідентифікатор функції	Назва функції	Функціональні залежності	Вплив на досягнення мети, %	Пріоритет функції
FR1	Реєстрація користувача.	-	10%	C
FR2	Здійснити пошук фільму.	-	5%	M
FR3	Пожегтвування на фільм.	FR1	25%	M
FR4	Коментування фільмів.	FR1	5%	S
FR5	Перегляд фільмів.	FR6	25%	M
FR6	Завантаження фільмів.	FR1	25%	S
FR7	Видалення фільмів.	FR1	5%	S

## 2.2 Концептуальний опис архітектури програмного продукту

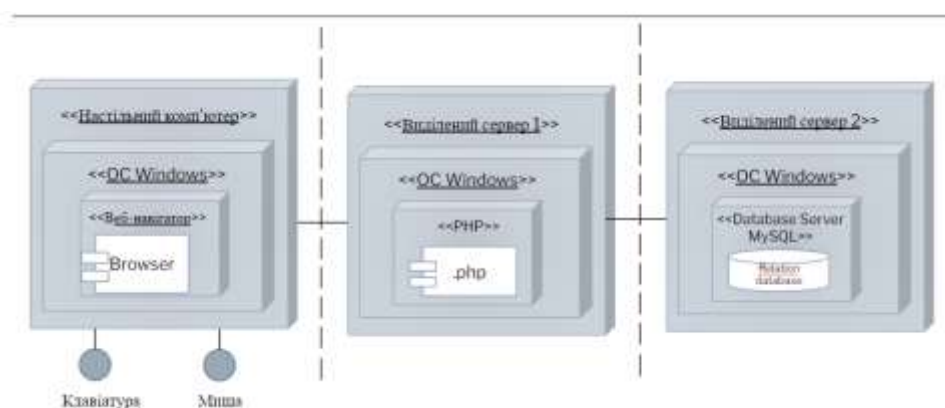


Рис. 2.1 – Діаграма архітектури.

## 2.3 План розробки програмного продукту

### 2.3.1 Оцінка трудомісткості розробки програмного продукту

Таблиця 2.1 – Вагові коефіцієнти акторів.

Актор	Ваговий коефіцієнт
Гість	1
Користувач	2
Адміністратор	3

Таблиця 2.2 – Вагові коефіцієнти прецедента.

Ідентифікатор функції	Тип прецедента	Кількість кроків сценарію	Ваговий коефіцієнт
FR1	Середній	4-7	15
FR2	Простий	<=3	10

FR3	Середній	4-7	5
FR4	Простий	$\leq 3$	5
FR5	Простий	$\leq 3$	15
FR6	Простий	$\leq 3$	10
FR7	Простий	$\leq 3$	5

Визначення UUCP :

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

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

$$UUCP = 6 + UC = 71$$

Таблиця 2.3 – Визначення технічної складності.

Показник	Опис показника	Вага
T1	Розподілена система	1
T2	Висока продуктивність (пропускна здатність)	2
T3	Робота кінцевих користувачів в режимі онлайн	1
T4	Складна обробка даних	0,5
T5	Повторне використання коду	0,5
T6	Простота установки	0,5
T7	Простота використання	1

T8	Переносимість	1
T9	Простота внесення змін	2
T10	Паралелізм	0
T11	Спеціальні вимоги до безпеки	1
T12	Безпосередній доступ до системи з боку зовнішніх користувачів	3
T13	Спеціальні вимоги до навчання користувачів	1

Визначення TCF :

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

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

Таблиця 2.4 – Рівень класифікації.

Показник	Опис показника	Вага
F1	Знайомство з технологією	2
F2	Досвід розробки додатків	1
F3	Досвід використання об'єктно-орієнтованого підходу	1
F4	Наявність провідного аналітика	0
F5	Мотивація	3

F6	Стабільність вимог	4
F7	Часткова зайнятість	1
F8	Складні мови програмування	1

Визначення EF :

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

$$UUCP = 71$$

$$TCF = 0,745$$

$$EF = 1$$

$$UCP = UUCP * TCF * EF = 52,895$$

$$\text{Трудомісткість} = 52,895 * 20 = 1057,9 \text{ люд./год.}$$

### 2.3.2 Визначення дерева робіт з розробки програмного продукту

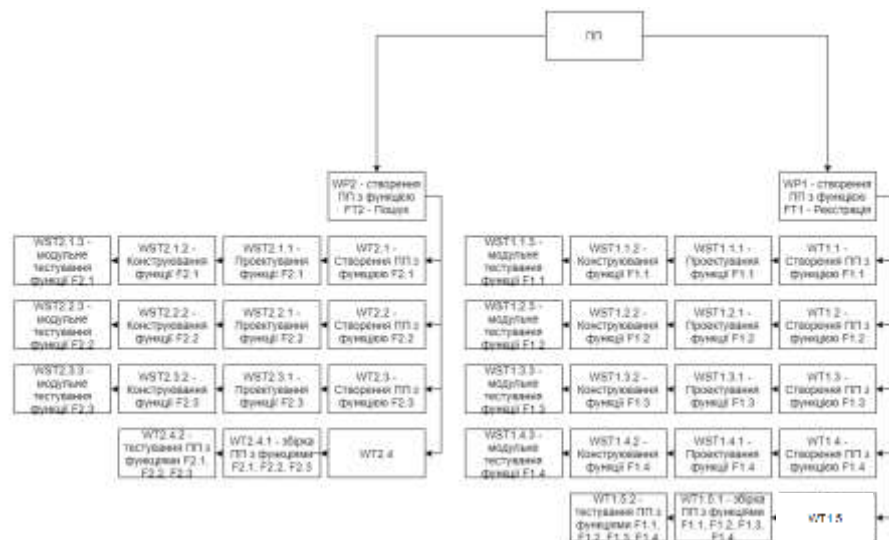


Рис. 2.2.1 – Дерево робіт.

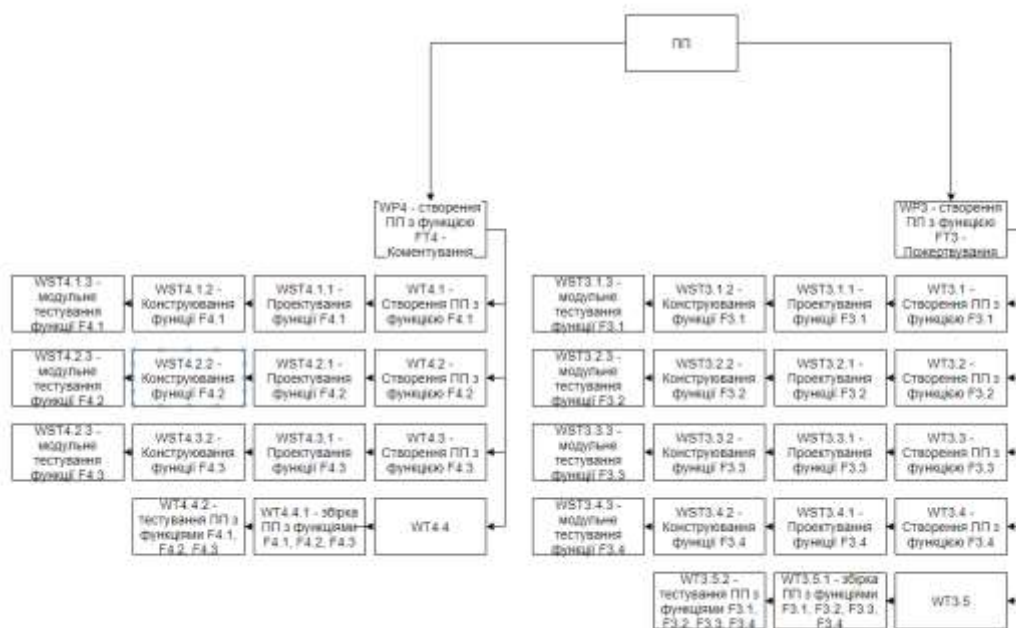


Рис. 2.2.2 – Дерево робіт.



Рис. 2.2.3 – Дерево робіт.

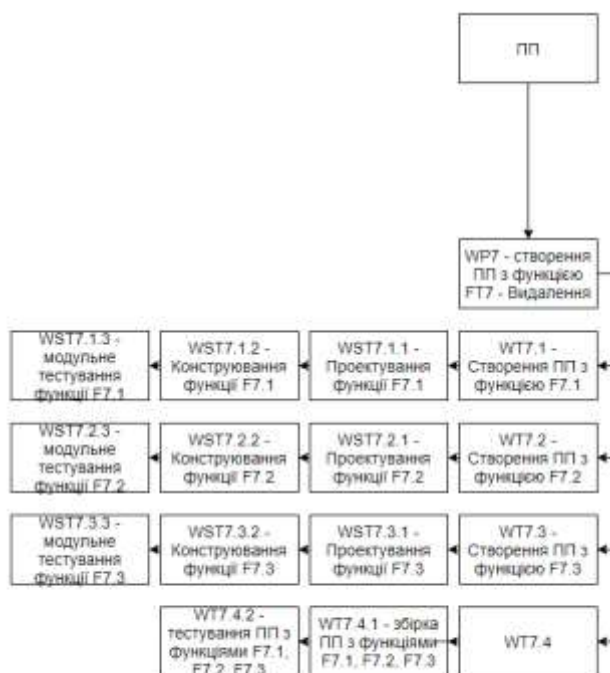


Рис. 2.2.4 – Дерево робіт.

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.

Рис. 2.3 – Таблиця з описом підзадач із закріпленням виконавців.



### 2.3.3 Графік робіт з розробки програмного продукту

Ідентифікатор	Дата початку виконання	Кількість днів витрачено	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.

Рис. 2.4 – Графік робіт.

Ідентифікатор	Дата початку виконання	Кількість днів витрачено	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.										

Рис. 2.5 – Діаграма Ганта.

### 3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Концептуальне та логічне проектування структур даних програмного продукту.

3.1.1 Концептуальне проектування на основі UML-діаграми концептуальних класів.

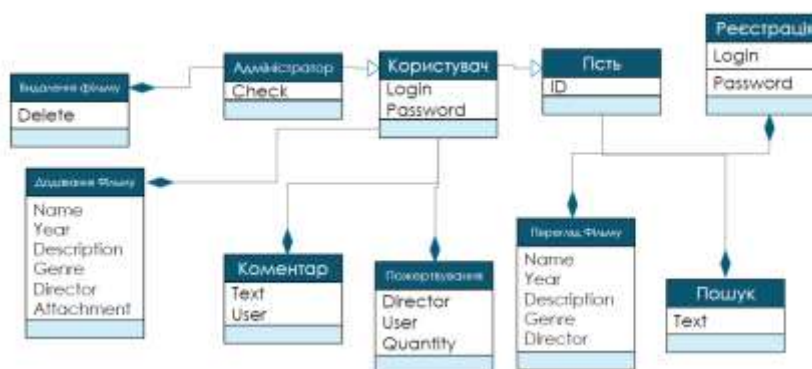


Рис. 3.1 – Діаграма концептуальних класів.

3.1.2 Логічне проектування структур даних.

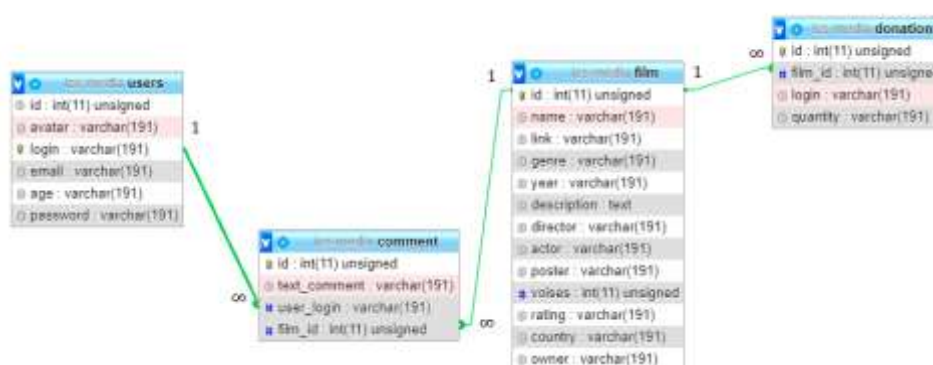


Рис. 3.2 – Проектування бази даних.

### 3.2 Проектування програмних класів

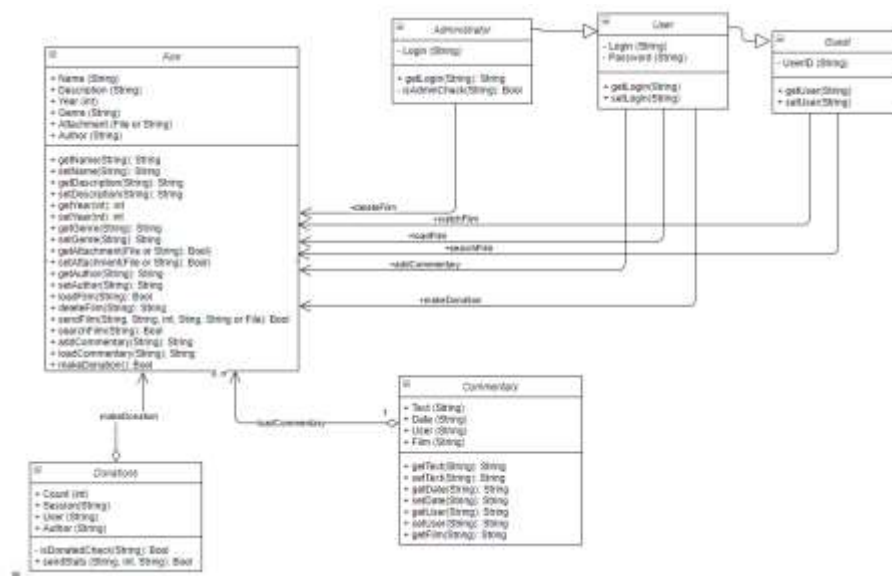


Рис. 3.2 – Діаграма програмних класів.

Назва	Хто виконував
Administrator	Панібратцев О.Ю.
User	Панібратцев О.Ю.
Guest	Панібратцев О.Ю.
Commentary	Додон Д.І.
Donations	Додон Д.І.
Film	Додон Д.І.

Рис. 3.3 – Задачі для учасників.

### 3.3 Проектування алгоритмів роботи методів програмних класів

Код UML – діаграми для реєстрації:

@startuml

title Registration()

```

start
:Виведення на екран полей для вводу для реєстрації;
note right
Mockup полей можна знайти у пункті 1.5.1.1.2 (FR1)
end note
if (Введені дані в усі обов'язкові поля?) then (yes)
:Введення в базу даних нового облікового запису користувача;
note left
INSERT INTO users (id, login, password) VALUES (1, 'user', 'password');
end note
else (no)
:Виведення помилки реєстрації;
endif
stop
@enduml

```



Рис. 3.4 – UML – алгоритм реєстрації.

Код UML – діаграми для пошуку:

```

@startuml
title Search()
start
:Виведення на екран поля для пошуку;

```

note right

Mockup полей можна знайти у пункті 1.5.1.1.2 (FR2)

end note

:Вводиться запит до строки пошуку;

:Відправляється запит до бази;

note left

SELECT \* FROM film WHERE film.name = "filmName"

end note

if (База має такий фільм) then (yes)

:Введення на екран користувача сторінку фільму

else (no)

:Виведення про відсутність фільма;

endif

stop

@enduml



Рис. 3.5 - UML – діаграма для пошуку.

Код UML – діаграми для донату:

```

@startuml
title Donate()
start
:Виведення на екран поля для донату;
note right
    Mockup полей можна знайти у пункті 1.5.1.1.2 (FR3)
end note
:Вводиться кількість грошей до строки донату;
:Перенаправлення до зовнішнього сервісу;
if (Оплата успішна) then (yes)
:Введення на екран успішного повідомлення та відправка даних до
бази;
note left
    INSERT INTO donations (id, film_id, login, quantity) VALUES (id, f_id,
login, 999)
end note
else (no)
:Виведення про не успішний перевід коштів;
endif
stop
@enduml

```



Рис. 3.6 – UML – діаграма для донату.

Код UML – діаграми для коментування:

@startuml

title Commentaries()

start

:Виведення на екран поля для вводу коментарів;

note right

Можливо поля можна знайти у пункті 1.5.1.1.2 (FR4)

end note

:Вводиться текст коментарю та відправляється до бази;

note left

INSERT INTO comment (id, text\_comment, user\_login, film\_id) VALUES  
(id, 'Классный фильм.', 'admin', f\_id)

end note

@enduml



Рис. 3.7 – UML – діаграма для коментування.

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

Рис. 3.8 – Таблиця учасників.

### 3.4 Проектування тестових наборів методів програмних класів

Таблиця 3.1 – Тестовий набір для реєстрації.

№ тесту	Опис значень вхідних даних	Опис очікуваних значень результату
1	1. login = '' 2. password = '' 3. натиснути send	Повідомлення про невведені дані.
2	1. login = 'login' 2. password = '' 3. натиснути send	Повідомлення про невведені дані.
3	1. login = '' 2. password = 'password' 3. натиснути send	Повідомлення про невведені дані.
4	1. login = 'login' 2. password = 'password' 3. Натиснути send	Залоговано і перенесло до головної сторінки

Таблиця 3.2 – Тестовий набір для перегляду фільмів.

№ тесту	Опис значень вхідних даних	Опис очікуваних значень результату
1	film_id = "1"	Загрузка фрейму з фільмом.
2	film_id = "231242"	ПП не видає інформації.



Таблиця 3.3 – Тестовий набір для додавання фільмів.

№ тесту	Скрипти	Опис очікуваних значень результату
1	1. \$id = id \$film = film \$poster = poster \$name = name \$link = link \$description = description 2. Натиснути відправити.	Повідомлення що фільм додано, можна знайти в каталозі.
2	1. Будь-яке поле з першого кейсу пусте. 2. Натиснути відправити.	«Вы пропустили это поле»

Назва	Хто виконував
Ресстрація	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.

Рис. 3.9 – Таблиця учасників.

## 4 КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ

### 4.1 Особливості конструювання структур даних

#### 4.1.1 Особливості інсталяції та роботи з СУБД

В даному ПП використовувалася СКБД MySQL 5.0.4 з програмою ХАМРР.

Встановлення на Windows:

1. Переходимо на сайт розробника.
2. Встановлюємо собі на ПК ХАМРР.
3. Запускаємо ХАМРР і можемо перейти до бази MySQL клікнувши на кнопку Admin.

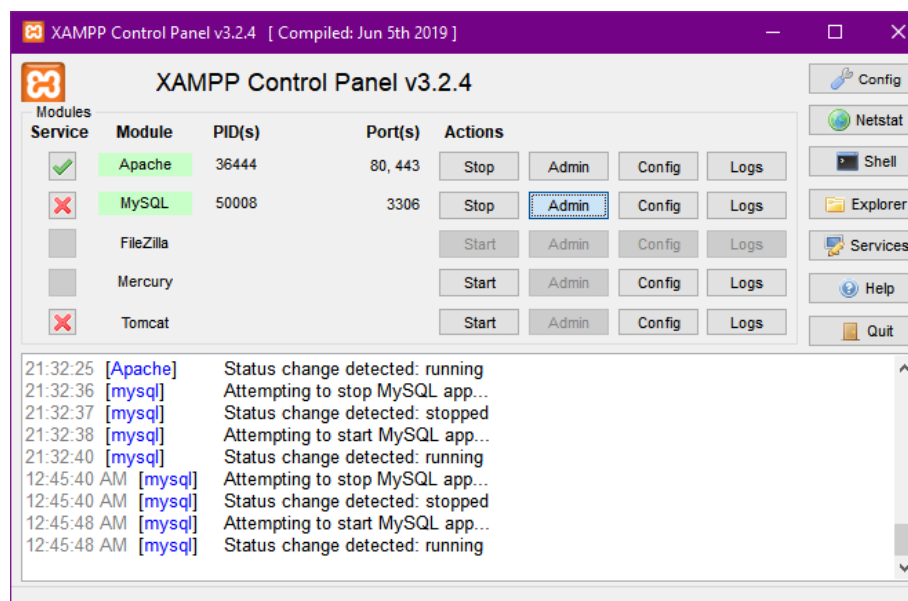


Рис. 4.1 – Вікно ХАМРР.

#### 4.1.2 Особливості створення структур даних

Створення таблиці фільмів:

Таблиця з фільмами містить інформацію про фільми, а саме – назва, автор, год, жанр, опис, коментарі за айді та рейтинг.

```
CREATE TABLE `film` (  
  `id` int(11) NOT NULL PRIMARY KEY,  
  `name` varchar(191),  
  `link` varchar(191),  
  `genre` varchar(191),  
  `year` int(11),  
  `description` text,  
  `director` varchar(191),  
  `actor` varchar(191),  
)
```

Таблиця користувачів містить унікальний логін та пароль.

```
CREATE TABLE `users` (  
  `login` varchar(191) NOT NULL PRIMARY KEY,  
  `password` varchar(191))
```

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

Рис. 4.2 – Таблиця учасників.

## 4.2 Особливості конструювання програмних модулів

### 4.2.1 Особливості роботи з інтегрованим середовищем розробки

Так як розробка проходила в середовищі Sublime, ми мали можливість застосовувати багато плагінів. Sublime дуже швидкий, безкоштовний, інтуїтивно зрозумілий.

Також, Sublime має швидкий доступ до github. З ним, ми набагато швидше робили push.

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

Особливість створення нашого коду в тому, що ми використовували бібліотеку яка набагато зручніше працювала з БД. Називається вона RedBean.

Особливості написання коду коментування:

```

<?php
require_once 'db.php';

// Указываем, что будем работать с таблицей comment
$comment = R::dispense('comment');

$error = "";

if(trim($_POST['text_comment']) != ""){//проверка на пустоту комментария
    $com = htmlspecialchars($_POST['text_comment']);
    @$comment['text_comment'] = $com;
}else{
    $error .= "comment is empty";
}

```

Рис. 4.3 – Приклад конекту до БД та перевірки на пустоту.

У цьому моменті ми бачимо як відбувається конект до БД та вибір таблиці.

```

<td align="center">
<?php //вивід коментарів
$filmId = $id;
$comments = R::find('comment', 'film_id = ?', [$filmId]);

$login = []; //создам массив в котором будем хранить логин пользователя

foreach ($comments as $key => $value) {
    $login[$key] = $comments[$key]['user_login'];
}

$img = [];

foreach ($login as $key => $login) { //беру логин пользователя из базы данных
    $users = R::findOne('users', 'login = ?', [$login]);
    $img[$key] = $users['avatar'];
}

if (empty($comments)) {
    echo "<table width = '100%' border = '1' class = 'comment'>";
    foreach ($comments as $key => $value) {
        if($comments[$key]['text_comment'] != ''){
            echo "<tr>";
            if(file_exists($img[$key])){
                echo "<td> " . "<img src = " . $img[$key] . " class = 'log_img'> " . "<td> " . $comments[$key]['user_login'] . " " . "<td> " . $comments[$key]['text_comment'] . " " . "<td>";
            }
            echo "</tr>";
        }
    }
    echo "</table>";
}else{
    unset($filmId);
}

```

Рис. 4.4 – Вивід коментарів до сторінки фільму.

Особливості написання коду перегляду фільмів:

					ІС КР 122 ЕАІ185 ПЗ	АР
						45
Зм.	Арк.	№ докум.	Підп.	ДА		

```
<?php
require_once 'db.php';
$id = $_GET['id'];
$film = R::findOne('film', 'id = ?', [$id]);
$poster = $film['poster'];
$name = $film['name'];
$link = $film['link'];
$description = $film['description'];
$rating = $film['rating'];
$voices = $film['voices'];
$owner = $film['owner'];
$director = $film['director'];
$year = $film['year'];
$country = $film['country'];
$genre = $film['genre'];
```

Рис. 4.5 – Приклад додавання даних с таблиці в масив.

```
<!-- Блок -->
<div class="filmcenter">
  <iframe src="http://echo "link": ?" align="center" width="640" height="360" frameborder="0" allowfullscreen"></iframe>
</div>
```

Рис. 4.6 – Додавання фільму до сторінки.

У цьому прикладі можна побачити як відбувається заповнення масиву даними про фільм, а потім, фільм відображається на сторінці в <iframe>.

Особливості написання коду про додавання фільму:

```
<?php
require_once 'db.php';
if (isset($_POST) && !empty($_POST)) {
    $film = R::findOne('film', 'name = ?', [$_POST['filmname']]);
    $login = $_SESSION['logged_user']->login;
    //перевірка на наявність фільму
    if (!empty($_POST['filmname']) && !empty($_POST['filmyear']) &&
        !isset($login) && $film->year != $_POST['year']) { //перевірка на єдиність фільму
        // Додавання фільму
        $film = R::dispense('film');
        $film->name = $_POST['filmname'];
        $film->country = $_POST['country'];
        $film->link = $_POST['linkurl'];
        $film->genre = $_POST['genre'];
        $film->year = $_POST['year'];
        $film->description = $_POST['description'];
        $film->director = $_POST['director'];
        $film->poster = $_POST['poster'];
        $film->rating = 0;
        $film->voices = 0;
        $film->owner = $login;
        echo "
        <script type='text/javascript'>
            alert('Фільм був доданий.');//
        </script>";
        R::store($film);
    } else {
        echo "Фільм з такою назвою вже є!"; //</script>";
    }
}
```

Рис. 4.7 – Додавання фільму.

					ІС КР 122 ЕАІ185 ПЗ	АР
Зм.	Арк.	№ докум.	Підп.	ДА		46

Можемо побачити перевірку на пустоту полей.

Знов заповняється масив даних \$film, а потім загружається до БД командою R::store.

Особливості написання коду про видалення фільму:

```
<?php
require_once 'db.php';
if(!empty($_GET)){
    $id = $_GET['id'];
    $film = R::load('film',$id);
    R::trash($film);
    echo "<script> alert('Фильм был удален');location='../index.php';</script>";
}
else{
    echo "<script> alert('Что то пошло не так');</script>";
}
```

Рис. 4.8 – Видалення фільмів.

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.

Рис. 4.9 – Таблиця учасників.

Бачимо найпростішу функцію яка просто видаляє поле. R::trash.

#### 4.2.3 Особливості створення програмних модулів

Всі наші програмні модулі зберігаються на власному ПК.

Атож, нам потрібно їх тримати на сервері. Як ми вже знаємо – сервер ХАМРР. Працює на Apache.

Тому всі програмні модулі тримаються у нас у файлах на локальному диску, та, у випадку необхідності, запускаються з відкритим портом до глобальної мережі інтернет.

Все програмні модулі зберігаються у папці cgi (мається на увазі «Сценарії користування», тобто – ядро).

filters	02.12.2020 23:38	Папка с файлами	
libs	02.12.2020 23:43	Папка с файлами	
addFilm.php	03.12.2020 1:21	Файл "PHP"	4 КБ
change_login_img.php	02.12.2020 14:55	Файл "PHP"	4 КБ
comment.php	02.12.2020 14:55	Файл "PHP"	2 КБ
db.php	08.12.2020 2:34	Файл "PHP"	1 КБ
delete_acc.php	02.12.2020 14:55	Файл "PHP"	3 КБ
delete_login_photo.php	02.12.2020 14:55	Файл "PHP"	3 КБ
deleteFilm.php	02.12.2020 14:55	Файл "PHP"	1 КБ
donation.php	07.12.2020 18:22	Файл "PHP"	2 КБ
error.php	02.12.2020 14:55	Файл "PHP"	1 КБ
index.php	02.12.2020 21:30	Файл "PHP"	5 КБ
login.php	02.12.2020 14:55	Файл "PHP"	3 КБ
logout.php	02.12.2020 14:55	Файл "PHP"	1 КБ
newPassword.php	02.12.2020 14:55	Файл "PHP"	6 КБ
randomFilms.php	06.12.2020 21:28	Файл "PHP"	1 КБ
search.php	02.12.2020 21:13	Файл "PHP"	2 КБ
show.php	06.12.2020 21:27	Файл "PHP"	6 КБ
signup.php	02.12.2020 14:55	Файл "PHP"	6 КБ
view.php	07.12.2020 18:22	Файл "PHP"	14 КБ

Рис. 4.10 – Структура програмних модулів.

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



## Особливості реєстрації:

```
<div class="signup">
<h2>Регистрация</h2>
<form action="/api/signup.php" method="POST" enctype="multipart/form-data">
<input type="text" name="login" placeholder="Введите логин" required="" value="{php echo $data['login']; }"><br><br>
<input type="email" name="email" placeholder="Введите e-mail" required="" value="{php echo $data['email']; }"><br><br>
<input type="text" name="age" placeholder="Введите возраст" required="" value="{php echo $data['age']; }"><br><br>
<input type="password" name="password" placeholder="Введите пароль" required="" value="{php echo $data['password']; }"><br><br>
<input type="password" name="password_2" placeholder="Повторите пароль" required="" value="{php echo $data['password_2']; }">
<label for="file">Выберите аватар (не более 2mb):</label>
<input type="file" name="login_img">
<br><br>
<button class="signupbtn" type="submit" name="do_signup">Регистрация</button>

<h2>{php echo $data['name']; }</h2>
</div>
```

Рис. 4.11 – Реєстрація.

Реєстрація – це проста html форма, в яку впроваджено php змінні.

## Особливості пошуку:

```
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) {
    $id .= "aid[]=".$value;
}

$url = $type.$id;

header("Location:show.php?url=".$url);//посылание гет запроса
}else{
    echo "<div class=\"validation\">Фильма с таким названием не найдено</div>";
}
}
```

Рис. 4.12 – Пошук.

Пошук працює з GET запитом із БД. Забирається масив фільмів, та зрештовується з запитом у пошуку.

					ІС КР 122 ЕАІ185 ПЗ	АР
						49
Зм.	Арк.	№ докум.	Підп.	ДА		

## Особливості донатування:

```
<form id="payment" name="payment" method="post" action="https://sch.interkassa.com/" enctype="utf-8">
<input type="hidden" name="ik_co_id" value="5f07f3aee39a9363d362a6c" />
<input type="hidden" name="ik_pa_no" value="<?php echo mt_rand(0,100000000000); ?>" />
<input type="hidden" name="ik_at" value="<?php echo $_GET['quantity']; ?>" />
<input type="hidden" name="ik_desc" value="Пожертвування" />
<input type="submit" value="Пожертвувати" />
</form>
```

Рис. 4.13 – Донат.

Донат використовує зовнішній сервіс intercassa. Модуль використовує рандомайзер для того щоб зробити випадковий номер запису.

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.

Рис. 4.14 – Таблиця учасників.

## 4.3 Тестування програмних модулів

### Особливості тестування:

BDD тести - це простий текст, на людській мові, написаний у формі історії (сценарію), яка описує деяку поведінку.

У Cucumber для написання тестів використовується Gherkin-нотація, яка визначає структуру тесту і набір ключових слів. Тест записується в файл з розширенням \*.feature і може містити як один, так і більш сценаріїв.

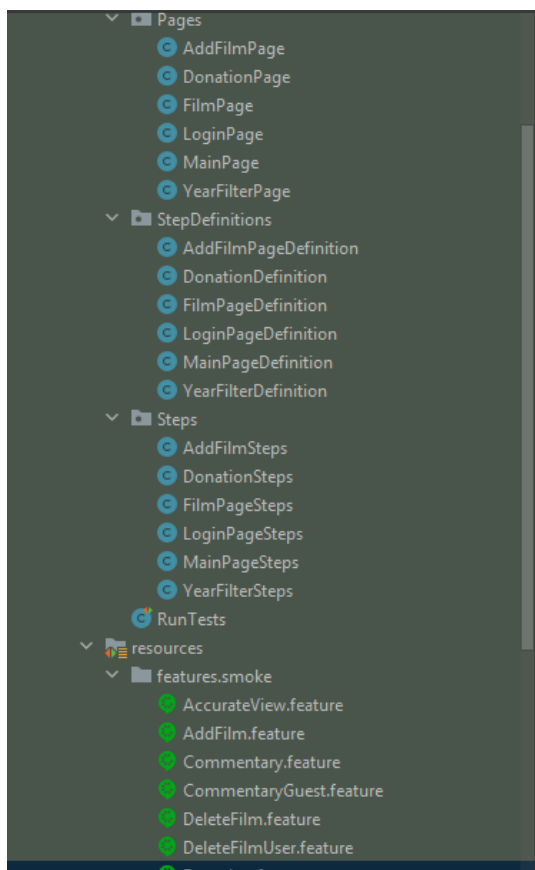


Рис. 4.15 – Файлова структура автоматичних тестів.

Serenity - это фреймворк с открытым исходным кодом для написания функциональных тестов, в который из коробки встроена система генерации отчетов о выполненных тестах.

Тести робились з чотирьохма видами написання сценаріїв.

					IC KP 122 EAI185 ПЗ	AP
						51
Зм.	Арк.	№ докум.	Підп.	ДА		

У цьому виді прописано сценарії використання програмою. Тестування робиться з зовнішнім інтерфейсом (метод BlackBox).

Сценарій тестування функції коментування:

**Feature:** User can add commentary

**Scenario:** User must be logged in

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

**Then** User click at cabinet button

**And** user click to authorization

**Then** user enter login2 "[user](#)"

**And** user enter passwordd

**Then** click Log In button

**Then** user click at any page of film

**And** user enter his commentary "[check](#)"

**Then** user click send commentary button

**Given** user can see his commentary

**Then** user click send commentary button

**And** page reloaded

Сценарій тестування функції додавання фільму:

**Feature:** User can add film to account

**Scenario:**

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

**Then** User click at cabinet button

**And** user click to authorization

					ІС КР 122 ЕАІ185 ПЗ	АР
						52
Зм.	Арк.	№ докум.	Підп.	ДА		

Then user enter login "admin"

And user enter passwordr

Then click Log In button

Then User click at cabinet button

Given user can add film by clicking button

Then user enter name of film "name"

And user enter url of film "check"

And user enter genre of film "genre"

And user enter year of film "2020"

And user enter director of film "director"

And user enter country of film "country"

And user delete url of film

Then click send button

And Validation message appeared

And user enter url of film "check"

And user enter link to poster "poster"

And user enter description "description"

Then click send button

Сценарій тестування функції видалення фільму:

**Feature:** Admin can delete film

**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"

					ІС КР 122 ЕАІ185 ПЗ	АР
						53
Зм.	Арк.	№ докум.	Підп.	ДА		

And user enter passwordr  
Then click Log In button  
Then user click at any page of film  
And click delete button

Сценарій тестування функції перегляду фільму.

Feature: user can watch films

Scenario:

Given User can access main page "<http://lopik.zapto.org/>"  
Then user click at any page of film  
Then user can see frame

Отримання елементів зі сторінки коментарів.

```
@FindBy(xpath = "/html/body/div[2]/form/input[3]")  
public WebElement sendCommentary;  
  
@FindBy(xpath = "//*[contains(text(), 'Check')]")  
public WebElement textCommentary;
```

Отримання елементів зі сторінки додавання фільмів.

```
@FindBy(name = "filmName")  
public WebElement name;  
  
@FindBy(name = "filmURL")
```

					ІС КР 122 ЕАІ185 ПЗ	АР
						54
Зм.	Арк.	№ докум.	Підп.	ДА		

```

public WebElement url;

@FindBy(name = "genre")
public WebElement genre;

@FindBy(name = "year")
public WebElement year;

@FindBy(name = "director")
public WebElement director;

@FindBy(name = "country")
public WebElement country;

@FindBy(name = "actor")
public WebElement actor;

@FindBy(name = "poster")
public WebElement poster;

@FindBy(className = "addBtn")
public WebElement addButton;

@FindBy(name = "description")
public WebElement description;

```

Отримання елементів зі сторінки видалення фільмів.

					IC KP 122 EAI185 ПЗ	AP
						55
Зм.	Арк.	№ докум.	Підп.	ДА		

```
@FindBy(xpath = "/html/body/div[2]/table/tbody/tr/td[5]/a")
public WebElement deleteButton;
```

Отримання елементів зі сторінки перегляду фільмів.

```
@FindBy(className = "filmCenter")
public WebElement cinema;
```

Опис шагів тестування коментування:

```
@Then("^user enter his commentary \"([^\"]*)\"$")
public void user_enter_his_commentary(String arg1) {
    filmPageSteps.enterCommentary();
}

@Then("^user click send commentary button$")
public void user_click_send_commentary_button() {
    filmPageSteps.loadCommentary();
}

@Given("^user can see his commentary$")
public void user_can_see_his_commentary() {
    filmPageSteps.isCommentary();
}
```

Опис шагів тестування додавання фільмів:

					ІС КР 122 ЕАІ185 ПЗ	АР
						56
Зм.	Арк.	№ докум.	Підп.	ДА		



```
@Then("^user enter name of film \"([^\"]*)\"$")
public void user_enter_name_of_film(String arg1) {
    addFilmSteps.enterName();
}
```

```
@Then("^user enter url of film \"([^\"]*)\"$")
public void user_enter_url_of_film(String arg1) {
    addFilmSteps.enterUrl();
}
```

```
@Then("^user enter genre of film \"([^\"]*)\"$")
public void user_enter_genre_of_film(String arg1) {
    addFilmSteps.enterGenre();
}
```

```
@Then("^user enter year of film \"([^\"]*)\"$")
public void user_enter_year_of_film(String arg1) {
    addFilmSteps.enterYear();
}
```

```
@Then("^user enter director of film \"([^\"]*)\"$")
public void user_enter_director_of_film(String arg1) {
    addFilmSteps.enterDirector();
}
```

```
@Then("^user enter country of film \"([^\"]*)\"$")
```

```

public void user_enter_country_of_film(String arg1) {
    addFilmSteps.enterCountry();
}

@Then("^user enter link to poster \"([^\"]*)\"$")
public void user_enter_link_to_poster(String arg1) {
    addFilmSteps.enterPoster();
}

@Then("^user enter description \"([^\"]*)\"$")
public void user_enter_description(String arg1) {
    addFilmSteps.enterDescription();
}

@Then("^click send button$")
public void click_send_button() {
    addFilmSteps.clickSubmit();
}

```

Опис шагів тестування видалення фільмів:

```

@Then("^click delete button$")
public void click_delete_film() {
    filmPageSteps.clickDelete();
}

```

Опис шагів тестування перегляду фільмів:

					ІС КР 122 ЕАІ185 ПЗ	АР
						58
Зм.	Арк.	№ докум.	Підп.	ДА		

```
@Then("^user can see frame$")
public void user_watch_film(){
    filmPageSteps.isWatch();
}
```

Результати тестів:

Результати тестування коментування:

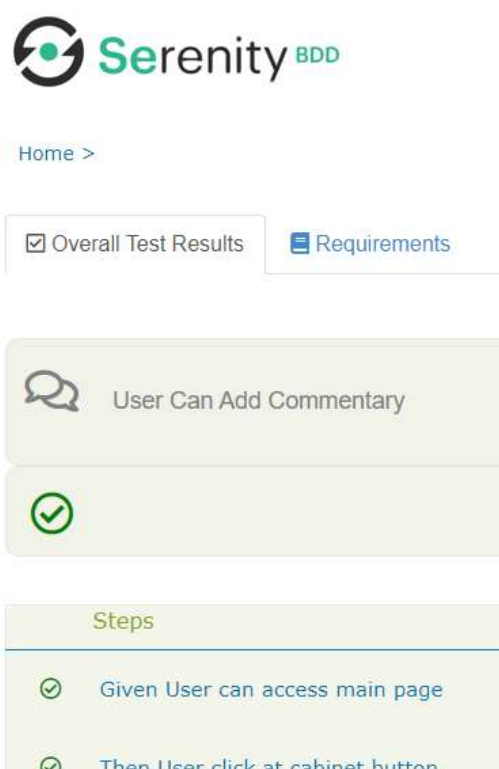


Рис. 4.16 – Результат тесту коментування.

					ІС КР 122 ЕАІ185 ПЗ	АР
						59
Зм.	Арк.	№ докум.	Підп.	ДА		

Результати тестування додавання фільмів:

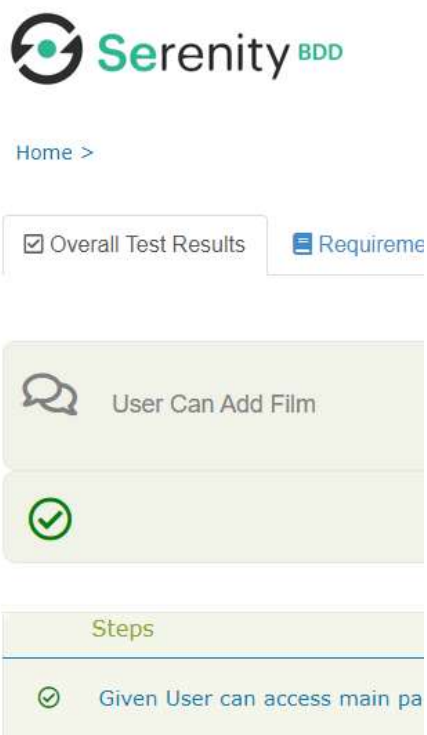


Рис. 4.17 – Результат тесту додавання фільму.

Результати тестування видалення фільмів:



Рис. 4.18 -

Результати тестування перегляду фільмів:

					ІС КР 122 ЕАІ185 ПЗ	АР
						60
Зм.	Арк.	№ докум.	Підп.	ДА		

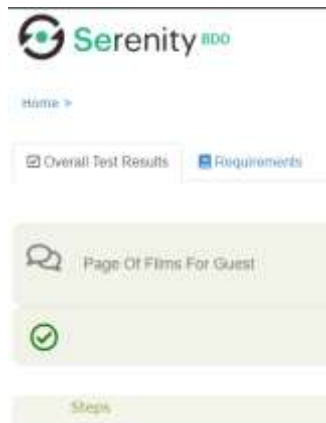


Рис. 4.19 – Результат тестування перегляду фільмів.

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.

Рис. 4.20 – Таблиця учасників.

## 5 РОЗГОРТАННЯ ТА ВАЛІДАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

### 5.1 Інструкція з встановлення програмного продукту

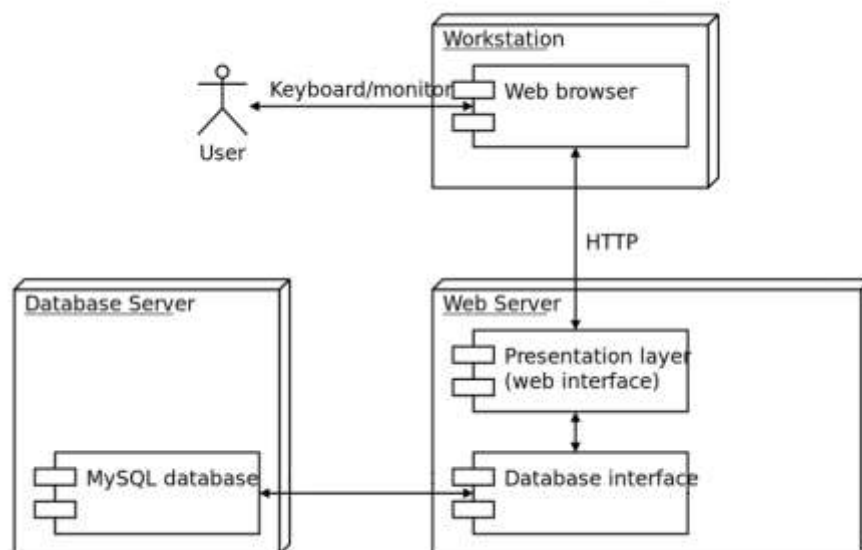


Рис. 5.1 – Діаграма розгортання.

Встановлення на Windows серверу:

1. Переходимо на сайт розробника ХАМРР та загрузаємо файл.
2. Встановлюємо собі на ПК ХАМРР.
3. Запускаємо ХАМРР і можемо перейти до бази MySQL та запустити сайт клікнувши на кнопку Admin.

Встановлення сайту для розробників:

1. Переходимо на GitHub репозиторій цього проекту.
2. Робимо клон собі на комп'ютер.

3. Переміщуємо папку template яка була зклонувана разом с проектом у папку htdocs програми ХАМРР.

## 5.2 Інструкція з використання програмного продукту

### Інструкція з реєстрації:

1. Ви повинні зайти до нашого сайту, та натиснути кнопку кабінету яка знаходиться біля лого нашого сайту.
2. Далі треба перейти до реєстрації, та заповнити поля які потребує ПП.




Рис. 5.2 – Реєстрація.

### Інструкція з перегляду фільмів:

1. Ви повинні зайти до нашого сайту та перейти до сторінки будь якого фільму з головної сторінки, або використавши пошук.
2. Натискаєте кнопку «Грати» в центрі вікна з фільмом.

					ІС КР 122 ЕАІ185 ПЗ	АР
						63
Зм.	Арк.	№ докум.	Підп.	ДА		



Рис. 5.3 – Перегляд фільму.

#### Інструкція з пошуку:

1. Ви повинні зайти до нашого сайту та натиснути на кнопку Фільтр яка знаходиться в хедері праворуч.
2. Вводите запит який має назву фільма, та натискаєте кнопку «Пошук».
3. ПП видає плити з фільмами які співпадають з запитом, ви можете вибрати фільм.

Название фильма

Поиск

Рис. 5.4 – Пошук.

#### Інструкція з коментування:

1. Ви повинні знаходитись на головній сторінці після реєстрації.
2. Переходимо на будь-яку сторінку фільму та шукаємо поле для коментарів.



3. Вводимо коментар, та клікаємо кнопку відправити.



Рис. 5.5 – Коментування.

Інструкція з додавання фільму:

1. Ви повинні знаходитись на сайті після реєстрації.
2. Переходимо до сторінки кабінету, та клікаємо на кнопку «Додати фільм»
3. Вводимо всі дані які необхідні ПП, та клікаємо кнопку «Відправити».

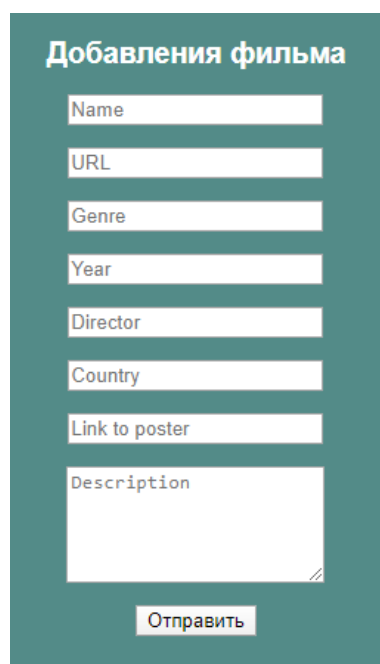


Рис. 5.6 – Додавання фільму.

Інструкція з пожертвування:

1. Ви повинні знаходитись на нашому сайті після реєстрації.
2. Переходимо до будь-якого фільму та шукаємо поле для донату.

Вводимо туди значення скільки бажаємо задонатити.

3. Проходимо шаги для відправлення коштів.

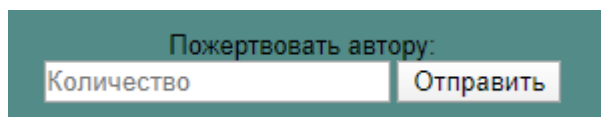


Рис. 5.7 – Пожертвування.

Інструкція для видалення фільмів для адмінів:

1. Ви повинні бути на нашому сайті після реєстрації.
2. Переходимо до будь якої сторінки фільму, та клікаємо кнопку

видалити.



Рис. 5.8 – Видалення фільму.

### 5.3 Результати валідації програмного продукту

					ІС КР 122 ЕАІ185 ПЗ	АР
						66
Зм.	Арк.	№ докум.	Підп.	ДА		

Шаги досягнення мети:

1. Користувач переходить до сторінки нашого сайту.
2. Користувач переходить до кабінету.
3. Користувач переходить до сторінки реєстрації.
4. Користувач проходить реєстрацію.
5. Користувач переходить до сторінки будь-якого фільму.
6. Користувач вводить кількість донату та клікає кнопку «Відправити».

6 шагів досягнення мети.

					ІС КР 122 ЕАІ185 ПЗ	АР
						67
Зм.	Арк.	№ докум.	Підп.	ДА		

## ВИСНОВОК

Наш ПП надає можливість отримувати кошти за свої режисерські творіння.

Заради підтвердження цього можна спробувати зробити це у тестовому режимі.

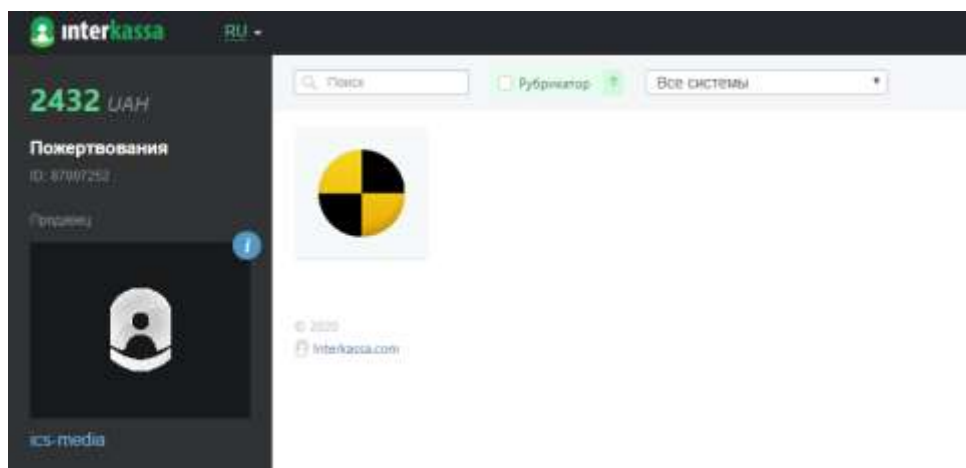


Рис. 6.1 – Тестовий режим донатів.

При проектуванні нашого ПП ми стикнулися з проблемами:

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

Все перелічені прецеденти були зроблені вчасно.

					ІС КР 122 ЕАІ185 ПЗ	АР
						68
Зм.	Арк.	№ докум.	Підп.	ДА		