

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

Розробка web-порталу медіа-сервісу

Студента 4 курсу, 10 групи,
спеціальності (6.050103
«Програмна інженерія»)
спеціалізації («Інженерія
програмного забезпечення»)

підпис студента

Гамалія Богдана
Сергійовича

Науковий керівник
кандидат педагогічних наук,
старший викладач

підпис керівника

Жирова Тетяна
Олександрівна

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

ЗМІСТ

Вступ	3
Розділ 1. Теоретичні основи розробки баз даних	5
1.1. Базы даних та системи управління базами даних	5
1.2. Flask microframework	6
1.3. Jinja 2	8
1.4. Шаблонізатори	9
Висновки до розділу 1	12
Розділ 2. Проектування UML діаграм.....	13
2.1. Use Case diagram.....	13
2.2. Sequence diagram	16
Висновки до розділу 2	19
Розділ 3. Використання мов програмування у проєкті	20
3.1. Використання мови Python	20
3.2. Створення головної сторінки хостингу	22
Висновки та пропозиції	25
Список використаної літератури.....	26

					<i>КНТЕУ 6.050103 10-3.БР</i>			
Змн	Арк.	№ докум.	Підпис	Дата	Розробка web-порталу медіа-сервісу	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				3	2	26
Керівник		Жирова Т. О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
Гарант		Цензура М.О.						
Розробив		Гамалій Б.С.						

Вступ

Основні ідеї сучасної інформаційної технології базуються на концепції, згідно якої дані повинні бути організовані у бази даних з метою їх адекватного відображення реального світу, що змінюється, й задоволення інформаційних потреб користувачів.

Збільшення обсягу й структурної складності даних, що зберігаються, розширення кола користувачів інформаційних систем призвели до широкого розповсюдження найбільш зручних й порівняно простих для розуміння реляційних (табличних) СУБД. Для забезпечення одночасного доступу багатьох користувачів до баз даних, часто розміщених досить далеко один від одного і від місця збереження баз даних, створені мережеві мультикористувацькі версії баз даних, основані на реляційній структурі. В них тим чи іншим шляхом вирішуються специфічні проблеми паралельних процесів, цілісності (правильності) й безпеки даних, а також санкціонування доступу.

У сучасному світі відео-хостинги займають дуже важливе місце у житті молоді. Активна позиція кожної людини має на увазі використання нових технологій, а також створення даних медіа-формату (відео, фото). Було вирішено об'єднати інформаційні технології, їх обробку та зберігання з потребами молоді у зберіганні спогадів з подальшою можливістю повернутися до них.

					<i>КНТЕУ 6.050103 10-3.БР</i>			
					<i>Розробка web-порталу медіа-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
Змн	Арк.	№ докум.	Підпис	Дата		В	3	26
Зав. Каф.	Криворучко О.В					<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>		
Керівник	Жирова Т. О.							
Гарант	Цензура М.О.							
Розробив	Гамалій Б.С.							

Об'єкт дослідження: робота з відеоінформацією в середовищі Інтернет

Предмет дослідження: процеси медіа-сервісу

Мета роботи: дослідити теоретичний аспект створення медіа-сервісу, а також обрати оптимальну технологію для його створення і впровадження.

					КНТЕУ 6.050103 10-4.БР	Лист
						4
Изм	Лист	№ Документа	Подпись	Дата		

Розділ 1. Теоретичні основи розробки баз даних

1.1. Базы даних та системи управління базами даних

База даних - це організований збір структурованої інформації або даних, які зазвичай зберігаються в електронній формі в комп'ютерній системі. База даних зазвичай контролюється системою управління базами даних (СУБД). Дані в рамках найбільш поширених типів баз даних, що діють сьогодні, зазвичай моделюються рядками і стовпцями таблиць, щоб зробити обробку та запит даних ефективними. Дані можуть бути легко доступні, керовані, змінені, оновлені, керовані та організовані. Більшість баз даних використовують структуровану мову запитів (SQL) для запису і запиту даних.

З поняттям бази даних тісно зв'язано поняття системи управління базами даних (СУБД). Це комплекс програмних засобів, призначених для створення структури нової бази, наповнення її даними, зміни даних і візуалізації інформації. Під візуалізацією інформації бази розуміється відбір даних, що відображаються, у відповідності до заданих критеріїв, їх сортування, оформлення й наступна видача на пристрої виводу чи передачі по каналам зв'язку.

У світі існує багато систем управління базами даних. Не дивлячись на те, що вони можуть по-різному працювати з різними об'єктами й надають користувачу різні функції засоби, більшість СУБД опираються на єдиний комплекс основних понять. Це дає можливість розглянути одну систему й узагальнити її поняття, прийоми й методи на весь клас СУБД.

					<i>КНТЕУ 6.050103 10-3.БР</i>			
					<i>Розробка web-порталу медіа-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
Змн	Арк.	№ докум.	Підпис	Дата		Р1	5	26
Зав. Каф.	Криворучко О.В					<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>		
Керівник	Жирова Т. О.							
Гарант	Цензура М.О.							
Розробив	Гамалій Б.С.							

1.2. Flask microframework

Станом на грудень 2016 року стабільна версія Flask має номер 0.12. Flask використовується для розробки таких проектів як Pinterest, LinkedIn, а також сторінка спільноти Flask.

Flask – це microframework. Він достатньо малий для того, щоб коли Ви познайомились з ним, Ви змогли читати та розуміти його вихідний код. Проте той факт, що він малий, не означає, що він менш корисний за інші фреймворки. Flask – був розроблений як фреймворк з багатьма розширеннями; він забезпечує тверде ядро з основними функціями, а розширення забезпечують інше. Через можливість обирати лише необхідні пакети розширень, у кінцевому результаті Ви маєте «легкий» стек без наворотів, який робить саме те, що потрібно. Отримані підсистеми маршрутизації, налагодження та інтерфейсу шлюзу веб-сервера (WSGI). Підсистеми маршрутизації, налагодження та Web Server Gateway Interface (WSGI) надходять з Werkzeug, а підтримка шаблонів забезпечується Jinja2. Werkzeug і Jinja2 є продуктами основного розробника Flask. У Flask не існує вбудованої підтримки для доступу до баз даних, перевірки веб-форм, аутентифікації користувачів або інших завдань високого рівня. Ці та багато інших ключових сервісів, що потрібні для більшості веб-додатків, доступні через розширення, які інтегруються з базовими пакетами. Як розробник, ви маєте право обирати розширення, які найкраще підходять для вашого проекту, або навіть писати власні розширення, якщо маєте достатній рівень навичок. Це відрізняє Flask від великих фреймворків, де більшість виборів було зроблено за вас, і їх важко або іноді неможливо змінити.

Найбільш зручним способом встановлення Flask є використання віртуального середовища. Віртуальне середовище - це приватна копія інтерпретатора Python, на якій Ви можете встановлювати пакунки

					КНТЕУ 6.050103 10-4.БР	Лист
						6
Изм	Лист	№ Документа	Подпись	Дата		

приватно, не впливаючи на глобальний інтерпретатор Python, встановлений у системі. Віртуальні середовища дуже корисні, оскільки вони запобігають конфліктам версій пакетів і версій в інтерпретаторі Python. Створення віртуального середовища для кожного додатка гарантує, що програми мають доступ лише до тих пакунків, які вони використовують, тоді як глобальний інтерпретатор залишається акуратним і чистим і служить лише як джерело, з якого можна створити більше віртуальних середовищ. В якості додаткової переваги віртуальним середовищам не потрібні права адміністратора. Віртуальні середовища створюються за допомогою сторонньої утиліти virtualenv.

Ось як описує свою роботу Армін Грінберг: "Все починалось як першоквітневий жарт, проте заживши великої слави, проект став по праву серйозним додатком." (ориг. It came out of an April Fool's joke but proved popular enough to make into a serious application in its own right)

Flask базується на засобі Werkzeug WSGI а також рушієві шаблонів Jinja2, що їх було створено як проекти Росоо у 2007 та 2008 роках відповідно, коли Ронакер та Георг Брандл створювали систему дошки оголошень на Python.

Незважаючи на відсутність головного релізу, Flask став надзвичайно популярним серед шанувальників Python. Станом на середину 2016 року, він був найбільш популярним веб-фреймворком Python на GitHub.

Властивості Flask:

1. Містить сервер для розробки та відлагоджувач
2. Вбудована підтримка юніт-тестів
3. Управління запитами RESTful
4. Використовує шаблони Jinja2
5. Має підтримку безпечних куків (сесії на стороні клієнта)

					КНТЕУ 6.050103 10-4.БР	Лист
						7
Изм	Лист	№ Документа	Подпись	Дата		

6. 100% відповідність WSGI 1.0
7. Підтримка Unicode
8. Докладна документація
9. Сумісність з Google App Engine
10. Наявність розширень для забезпечення бажаної поведінки

1.3. Jinja 2

Jinja — шаблонізатор для мови програмування Python створений Арміном Ронакером з ліцензією BSD. На відміну від схожого рушія шаблонів у Django, використовує вирази у стилі мови Python та використовує пісочницю для шаблонів. Завдяки тому, що шаблони Jinja засновані на текстовому форматі, тому створення розмітки документу стає подібним до написання сирцевого коду.

Шаблони рушія Jinja надають можливості налаштування тегів, фільтрів, тестів та глобальних параметрів. Також, на відміну від рушія Django, Jinja дозволяє розробнику шаблонів викликати функції з об'єктами у якості аргументів. Jinja є основним рушієм шаблонів у Flask.

Основні властивості Jinja:

1. Виконання у пісочниці.
2. Потужна автоматизація перевірки HTML для запобігання атак міжсайтового скриптингу (XSS).
3. Спадковість шаблонів.
4. Виконується компіляція в оптимальний python код на момент використання.
5. Опціонально доступна дострокова компіляція шаблонів, легке налагодження (номери рядків винятків безпосередньо вказують на правильний рядок шаблону).
6. Синтаксис можна налаштувати.

					КНТЕУ 6.050103 10-4.БР	Лист
						8
Изм	Лист	№ Документа	Подпись	Дата		

Jinja, подібно Smarty, має легку у використанні систему фільтрів подібно до конвеєру у Unix.

Також, крім веб-фреймворків, Jinja 2 використовується, наприклад, у системі управління конфігураціями SaltStack и Ansible.

Шаблон Jinja – це просто текстовий файл. Jinja може генерувати будь-який текстовий формат (HTML, XML, CSV, LaTeX і т.д.). Цей шаблон не потребує спеціального розширення: .html, .xml або будь-якого іншого розширення.

Шаблон містить змінні та/або вирази, які замінюються значеннями, коли шаблон відображається; і теги, які контролюють логіку шаблону. Синтаксис шаблону дуже схожий на синтаксис Django і Python.

1.4. Шаблонізатори

Шаблонізатор (template engine) – всі численні способи генерування текстових файлів (будь-яких: конфігураційних файлів, HTML-сторінок, звітів та інших) можна розділити на три великі категорії:

1. Інтерполяція тексту.
2. Embedded-мови.
3. Використання шаблонізатору.

Інтерполяція текст знайома всім програмістам на shell, на python, perl та іншими мовами програмування. У початковому тексті присутні певні послідовності, що позначають імена змінних, які при обробці замінюються їх значеннями.

Інтерполяція, як правило, дуже безпорадна. Максимум, що вона може, це форматування значень певного типу (наприклад, вивести дійсне число з такою-то точністю після коми) або найпростіші команди (наприклад, командна підстановка вигляду "` cmd `", в shell або perl). Ні циклів, ні умов, нічого більш складного в інтерполяції, зазвичай, не передбачено.

					КНТЕУ 6.050103 10-4.БР	Лист
						9
Изм	Лист	№ Документа	Подпись	Дата		

На іншому полюсі знаходяться embedded-мови, яскравим представником яких є PHP. Embedded, тобто впроваджений, означає що мова безпосередньо використовується в тексті, і для конструювання результату може використовуватися вся міць мови: умови, цикли, функції, бібліотеки, об'єктна орієнтація і так далі.

Мінус цього підходу в іншому - шаблон з початкового тексту із змінними значеннями, ніж він спочатку був, перетворюється в повноцінну програму, в якій вже і вихідного тексту не видно, а редагування шаблону перетворюється в написання програми, з усіма наслідками, що випливають звідси.

Шаблонізатори беруть найкраще з двох світів: з одного боку вони залишають шаблон максимально схожим на шаблон, не перетворюючи його в програму, з іншого боку - вони дозволяють автоматизувати найбільш очевидні рутинні завдання по генерації частин шаблону (але ні в якому разі не забираючи на себе частину, відповідальну за логіку)

Можливо, в майбутньому, інтерполяції рядків в скриптових мовах взагалі не буде, а вона буде замінена на шаблонізатор. У будь-якому випадку інтерполяція розвивається в цю сторону і різниця між ними все більше розмивається.

Найважливішою властивістю шаблонізатора є розв'язка даних, логіки і шаблону. Через те, що в шаблоні тільки логіка, що стосується генерації тексту, це дозволяє використовувати шаблонізатор в абсолютно різних системах, починаючи від web-фреймворків (для яких вони спочатку і розроблялися) і закінчуючи системами управління конфігурацією: Jinja інтегрована і в Ansible, і в SaltStack, і полегшує написання playbook'ів неймовірно.

Ви можете використовувати шаблонізатори всюди, де є шаблони та динамічно змінювані дані:

					КНТЕУ 6.050103 10-4.БР	Лист
						10
Изм	Лист	№ Документа	Подпись	Дата		

- Веб;
- Конфігураційні файли;
- Рахунки;
- Листи;
- Лабораторні та курсові;
- Заготовки для документів;
- Звіти.

Зараз найчастіше шаблонизатори пишуться на Python і для Python і серед них можна назвати найбільш популярні:

1. Django (частина Django - шаблонизатор) - з нього і почалася особлива популярність;
2. Jinja2 - як в Django, але набагато могутніше + є незалежним модулем;
3. Мако - інший синтаксис, більше свободи (може бути мінусом для шаблонизатор), швидкий (важливою для навантажених сайтів);
4. Genshi - дуже цікаве рішення для XML-шаблонів (але дуже повільне, якщо когось хвилює швидкість).

					КНТЕУ 6.050103 10-4.БР	Лист
						11
Изм	Лист	№ Документа	Подпись	Дата		

Висновки до розділу 1

Було розглянуто переваги мови Python і шаблонізаторів в цілому.

Насамперед це:

- Зменшення кількості помилок.
- Економія часу.
- Консистенція і лаконічність.
- Економія грошей.
- Покращення задоволеності клієнтів

Шаблони мають багато переваг для компаній, які обирають їх використання, включаючи економію коштів і часу, збільшення послідовності та зменшення помилок. Це дозволяє підвищити ефективність роботи компанії в цілому.

					КНТЕУ 6.050103 10-4.БР	Лист
						12
Изм	Лист	№ Документа	Подпись	Дата		

Розділ 2. Проектування UML діаграм

2.1. Use Case diagram

Призначення діаграми прецедентів в UML - демонстрація різних способів, якими користувач може взаємодіяти з системою.

У Unified Modeling Language (UML), use-case diagram може підсумовувати інформацію про користувачів системи (також відомі як актори) і їх взаємодії з системою. Для побудови, Ви будете використовувати набір спеціалізованих символів і роз'ємів. Ефективна схема використання може допомогти менеджерам проектів обговорити і представити:

- Сценарії, в яких система або додаток взаємодіє з людьми, організації або зовнішніми системами
- Цілі, які ваша система або додаток допомагає ці об'єкти (відомі як актори) досягнення
- Обсяг вашої системи

Діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі.

Діаграма прецедентів не вдається в багато деталей, наприклад, не слід очікувати, що вона може змодельовати порядок, в якому виконуються кроки. Замість цього правильна схема зображує загальний огляд відносин між варіантами використання, акторами і системами. Експерти рекомендують використовувати тематичні схеми як додаток до більш описових текстовальних документів.

UML — є мовою моделювання інструментарію, яку Ви можете використовувати для створення Ваших діаграм. Прецеденти представлені

					<i>КНТЕУ 6.050103 10-3.БР</i>			
Змн	Арк.	№ докум.	Підпис	Дата	<i>Розробка web-порталу медіа-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
Зав. Каф.		Криворучко О.В				Р2	13	26
Керівник		Жирова Т. О.				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>		
Гарант		Цензура М.О.						
Розробив		Гамалій Б.С.						

з міченої овальною формою. Дотримуйтеся фігури представляють учасник процесу, а також участь актора в системі моделюються з лінією між актором і прецедентом. Для того, щоб зобразити кордону системи, намалюйте рамку навколо використання самої справи.

UML діаграми прецедентів ідеально підходять для:

- Подання цілей системи взаємодії з користувачем
- Визначення та організація функціональних вимог в системі
- Визначення контексту і вимог системи
- Моделювання основного потоку подій в разі використання

Для того, щоб відповісти на питання: «Що таке діаграма прецедентів?» Ви повинні спочатку зрозуміти його будівельні блоки. Загальні компоненти включають в себе:

- Актори: Користувачі, які взаємодіють з системою. Актор може бути особа, організація або зовнішня система, яка взаємодіє з додатком або системою. Вони повинні бути зовнішніми об'єктами, які виробляють або споживають дані.
- Система: Конкретна послідовність дій і взаємодій між учасниками і системою. Система також може згадуватися як сценарій.
- Цілі: Кінцевий результат більшості випадків використання. Успішна схема повинна описувати діяльність і варіанти, які використовуються для досягнення мети.

Use Case 1. Запуск відео

ХАРАКТЕРИСТИЧНА ІНФОРМАЦІЯ

Мета в контексті: користувач надсилає запит на сайт і очікує, що відеофайл запуститься.

Сфера застосування: браузер.

Рівень: запит

					КНТЕУ 6.050103 10-4.БР	Лист
						14
Изм	Лист	№ Документа	Подпись	Дата		

Передумови: відеофайл знаходиться у БД на сервері, наявність посилання на сайті.

Успішна стадія завершення: запит надійшов на сервер, запит оброблено, сервер повертає відеофайл, користувач проглядає відеофайл.

Невдале завершення стану: сервер не обробив запит, комп'ютер користувача не відправив запит, сервер не отримав запит, сервер не відправив запит, комп'ютер користувача не отримав запит на завантаження, завантаження не почалося.

Основний актор: Користувачі різного віку, статі, раси.

Вторинний актор: комп'ютер, що задовольняє потреби користувача

Тригер: бажання користувача задовольнити свою потребу у перегляді відеофайлів, натиснення кнопки запуску відеофайлу.

ГЕНЕРАЛЬНИЙ СЦЕНАРІЙ УСПІХУ

1. Користувач надсилає запит на завантаження і запуск відеофайлу.

2. Комп'ютер успішно надсилає запит на сервер

3. Сервер успішно знаходить потрібний файл у БД

4. Сервер успішно надсилає запит на завантаження на комп'ютер користувача

5. Комп'ютер користувача успішно оброблює запит на завантаження файлу.

6. Комп'ютер користувача починає завантаження відеофайлу

7. Відеофайл починає відтворюватись на сайті.

8. Користувач переглядає відео

9. Користувач задовольнив свою потребу

SUB-варіації

Користувач може використовувати телефон.

Користувач може використовувати WEB-форму.

					КНТЕУ 6.050103 10-4.БР	Лист
						15
Изм	Лист	№ Документа	Подпись	Дата		

Пріоритет: TOP

Цільова ефективність: 1-3 секунд на запит до сервера, швидкість завантаження відеофайлу вища за швидкість відтворення.

Частота: мінімум 60 разів на день.

Служба надпорядкованого використання:

Автоматичне відтворення на інших сайтах

Канал первинного актора: БД основного сайту з відеофайлами.

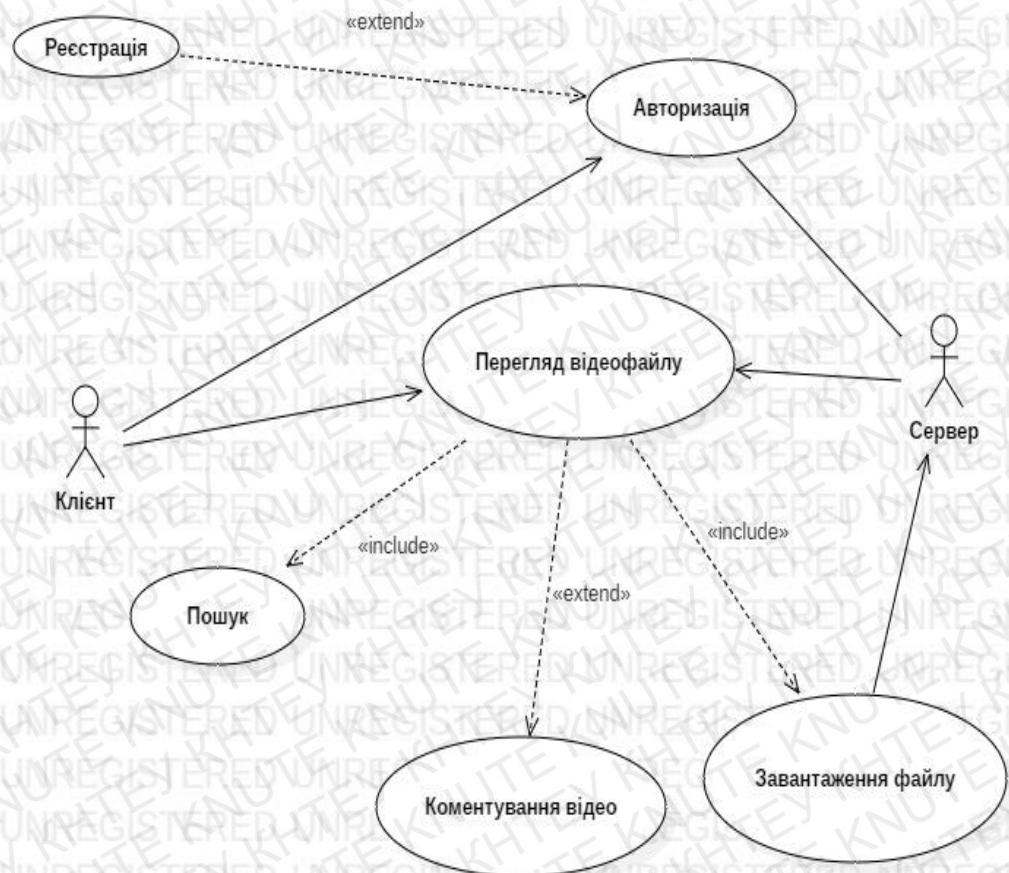


Рис. 2.1. Приклад Use case diagram на основі мого проекту

2.2. Sequence diagram

Sequence diagram – діаграми послідовностей є популярним рішенням динамічного моделювання в UML, тому що вони конкретно зосереджені на процесах і об'єктах, які існують одночасно, і

					КНТЕУ 6.050103 10-4.БР	Лист
						16
Изм	Лист	№ Документа	Подпись	Дата		

повідомленнях, якими вони обмінюються між собою для виконання функції до закінчення. Діаграми послідовностей моделюють взаємодії між об'єктами в одному випадку їх використання. Вони ілюструють, як різні частини системи взаємодіють один з одним для виконання функції, і порядок, в якому відбуваються взаємодії, коли виконується конкретний випадок використання.

Для того, щоб зрозуміти, що таке схема послідовності, важливо знати роль Unified Modeling Language, більш відома як UML. UML є моделюванням інструментарію, яка веде створення і позначення багатьох типів діаграм, в тому числі поведінкових схем, діаграм взаємодії та діаграм структури.

Діаграма послідовності являє собою тип діаграми взаємодії, так як вона описує, як і в якому порядку група об'єктів працює разом. Ці схеми використовуються розробниками програмного забезпечення та бізнес-професіоналами, щоб зрозуміти вимоги до нової системи або для документування існуючого процесу. Діаграми послідовності іноді відомі як діаграми подій або сценарії подій.

Існує два типи діаграм послідовності: UML діаграма і діаграма на основі коду. Остання надходить з програмного коду.

Діаграми послідовностей можуть бути корисними посиланнями для підприємств та інших організацій. Вони:

- Представляють деталі використання UML випадку.
- Модельну логіку складної процедури, функції або операції.
- Показують, як об'єкти і компоненти взаємодіють один з одним, щоб завершити процес.
- Допомагають планувати і зрозуміти детальну функціональність існуючого або майбутнього сценарію.

					КНТЕУ 6.050103 10-4.БР	Лист
						17
Изм	Лист	№ Документа	Подпись	Дата		

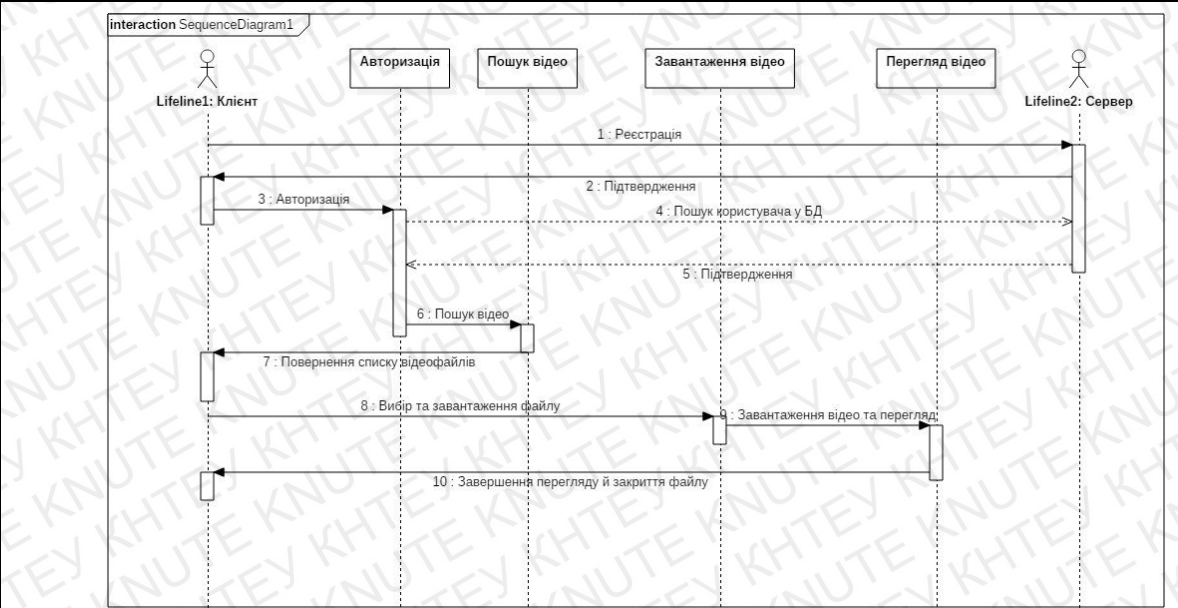


Рис. 2.2. Приклад Sequence diagram

Висновки до розділу 2

В основному, UML використовувалася як мова моделювання загального призначення в області програмної інженерії. Проте, вона вже знайшла свій шлях у документації декількох бізнес-процесів або робочих процесів. Наприклад, діаграми активності - тип UML-діаграм, які можуть використовуватися як заміна блок-схемам. Вони забезпечують більш стандартизований спосіб моделювання робочих процесів, а також більш широкий спектр функцій для поліпшення зручності читання та ефективності.

Сам UML знаходить різні способи використання в розробці програмного забезпечення та документації бізнес-процесів.

Були розроблені різноманітні діаграми для даного проекту. Це допомогло представити статичну структуру моделі, а також відношення між акторами й прецедентами системи.

					КНТЕУ 6.050103 10-4.БР	Лист
						19
Изм	Лист	№ Документа	Подпись	Дата		

Розділ 3. Використання мов програмування у проекті

3.1. Використання мови Python

Для створення відео-хостингу крім стандартної мови розмітки HTML у поєднанні з CSS я користувався мовою Python.

Python - це мова загального призначення. Це означає, що його можна використовувати для побудови будь-якої справи, яка буде спрощена за допомогою потрібних інструментів або бібліотек.

Професійно, Python відмінно підходить для веб-розробки, аналізу даних, штучного інтелекту та наукових обчислень. Багато розробників також використовували Python для створення продуктивності, ігор, і десктопних додатків, тому існує багато ресурсів, які допоможуть Вам навчитися робити це.

Головні переваги Python:

Проста для новачків

Будучи дуже високим рівнем мови, Python читається, як англійська, тобто вона не вимагає багато синтаксичного навчання від кодерів-початківців. Python не має високої складності, тому він дуже зручний для початківців, оскільки дозволяє новачкам зосередитися на вивченні концепцій програмування, і не турбуватися про зайві деталі.

Дуже гнучка

Як динамічна мова, Python дійсно дуже гнучкий. Тобто не існує жорстких правил щодо того, як створювати функції, це означає більшу гнучкість у вирішенні проблем за допомогою різних методів. Крім того, Python також прощає більше помилок, тому Ви все ще можете компілювати і запускати свою програму, доки не потрапите в проблемну частину.

					<i>КНТЕУ 6.050103 10-3.БР</i>			
Змн	Арк.	№ докум.	Підпис	Дата	<i>Розробка web-порталу медіа-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
Зав. Каф.		Криворучко О.В				РЗ	20	26
Керівник		Жирова Т. О.				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>		
Гарант		Цензура М.О.						
Розробив		Гамалій Б.С.						

П'ята найбільша спільнота StackOverflow

StackOverflow - це сайт з програмування, який, без сумніву, стане близьким другом для початківців. Python має 85.9к підписників, з більш ніж 500k питаннями про Python. Питання Python також є третіми, на які, швидше за все, відповідатимуть порівняно з іншими популярними мовами програмування.

Третя найбільша спільнота Meetup

На зустрічах Ви можете спілкуватися і вчитися у колег-розробників. Зустрічі часто пропонують наставництво для тих, хто цього бажає. На сайті Meetup.com є 1300+ груп Python, загальна кількість членів яких становить 608 тисяч осіб. Таким чином, з точки зору мов програмування, Python є третьою найбільшою спільнотою.

Четверта найпоширеніша мова в GitHub

Чим кориснішими є проекти, тим більша імовірність, що хтось вже побудував потрібну Вам функцію, що значно прискорить процес розробки.

					КНТЕУ 6.050103 10-4.БР	Лист
						21
Изм	Лист	№ Документа	Подпись	Дата		

3.2. Створення головної сторінки хостингу

Головна сторінка відео-хостингу зустрічає користувача приємним зовнішнім виглядом у темних тонах і легким у розумінні інтерфейсом.

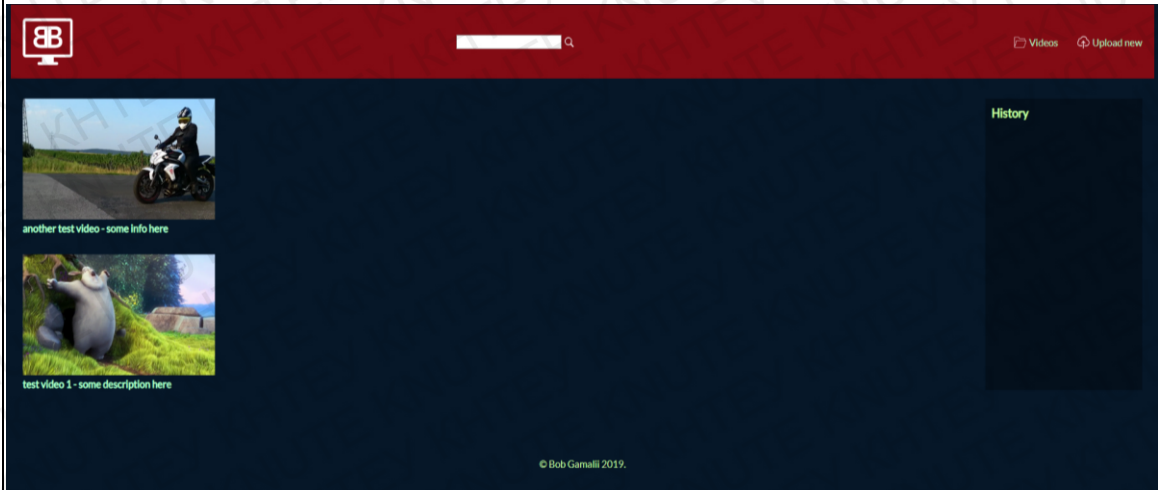


Рис. 3.1. Головна сторінка сайту

Отже розглянемо сам інтерфейс сайту:

Першим невеликим блоком на сайті можемо знайти так званий елемент меню «header». Це шапка у верхній частині сайту Header має інтерактивний логотип, при наведенні на який логотип трохи змінюється. Також це одна з навігаційних клавiш, за допомогою якої можна потрапити на головну сторiнку сайту. З правої сторони знаходиться блок з кнопками «Videos» та «Upload new», які переходять на відповідні сторінки – список відео та завантаження нового відео.



Рис. 3.2. Header сайту

					КНТЕУ 6.050103 10-4.БР	Лист
						22
Изм	Лист	№ Документа	Подпись	Дата		

Сайт був розроблений таким чином що будь-якому користувачу буде зрозумілий інтерфейс сайту і він зможе легко пересуватись по ньому.

Нижче знаходяться два блоки:

Блок з усіма завантаженими відео. Кожне відео має власну назву та опис відео;

Блок з відео, які користувач вже переглянув.



Рис. 3.3. Список відео на сайті та історія переглянутих відео

Останнім блоком на сторінці є елемент footer це та частина сайту, яка не має жодного функціоналу. Там знаходиться лише мій підпис.

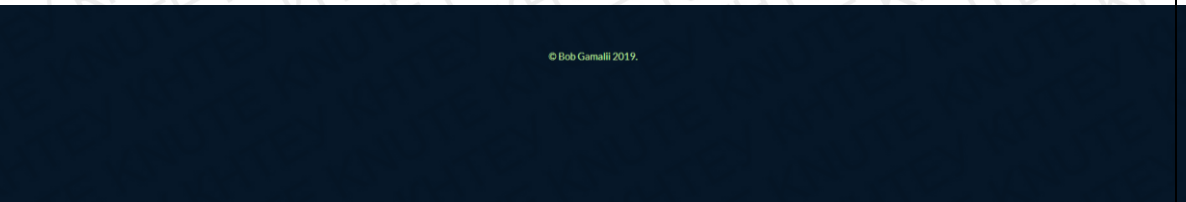


Рис.3.4. Footer сайту

Хостинг є дуже легким у супроводі і не потребує значних навичок або знань для модифікації. Кожен розробник, знайомий з мовою програмування Python, зможе модифікувати сайт.

					КНТЕУ 6.050103 10-4.БР	Лист
						23
Изм	Лист	№ Документа	Подпись	Дата		

Користувачем хостинг може використовуватись у якості хмарного сховища власних файлів. При необхідності користувач, який завантажив відео або написав коментар, може видалити його без втручання адміністратора. Інтерфейс хостингу є легким у розумінні і не потребує ніяких навичок для роботи з ним.

					КНТЕУ 6.050103 10-4.БР	Лист
						24
Изм	Лист	№ Документа	Подпись	Дата		

Висновки та пропозиції

Для розробки були застосовані наступні технології:

1. HTML 5
2. CSS 3
3. Мова програмування Python

Для створення проекту був використаний мікрофреймворк Flask.

Під час виконання проекту були визначені об'єкт, предмет, мета й завдання проекту.

В першому розділі були розглянуті теоретичні відомості щодо технологій, які були використані у процесі виконання проекту, а саме:

- Мікрофреймворк Flask;
- Двигун Jinja2;
- Шаблонізатори.

Другий розділ містить відомості про діаграми UML і їх користь у проектах в цілому. Також були розроблені власні діаграми для даного проекту.

У третьому розділі було продемонстровано візуальну складову сайту. Описано кожен блок та його функціонал. Інтерфейс був розроблений таким чином, щоб користувач відразу зміг отримати те, за чим прийшов на сайт. Були показані блоки сайту та їх функціонал. Також була використана технологія ORM, яка дозволила підвищити швидкість написання коду шляхом уникнення «копіпасту», а також підвищити гнучкість проекту.

					<i>КНТЕУ 6.050103 10-3.БР</i>			
					<i>Розробка web-порталу медіа-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
<i>Змн</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ВП</i>	<i>25</i>	<i>26</i>
Зав. Каф.		Криворучко О.В				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>		
Керівник		Жирова Т. О.						
Гарант		Цензура М.О.						
Розробив		Гамалій Б.С.						

Список використаної літератури

1. Вендров А.М. Практикум з проектування ПЗ економічних інформаційних систем: Учб. посібник / А.М. Вендров. – М.: Фінанси й статистика, 2004. – 192 с., – ISBN 5-279-02440-6.
2. Конгаловський М.Р. Енциклопедія технологій баз даних. – М.: Фінанси й статистика, 2002. – 800 с.: ISBN 5-279-02276-4.
3. Корнєєв В.В. Бази даних. Інтелектуальна обробка інформації / В.В. Корнєєв, А.Ф. Гарєєв, С.В. Васютін, В.В. Райх. – М.: Видавництво Молгачева С.В., Видавництво Нолидж, 2001, - 496 с.: ISBN 5-89251-100-6.
4. Armin Ronacher, Flask Documentation, випуск 0.9 English – 7 с.

					<i>КНТЕУ 6.050103 10-3.БР</i>			
Змн	Арк.	№ докум.	Підпис	Дата	<i>Розробка web-порталу медіа-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
Зав. Каф.	Криворучко О.В					СД	26	26
Керівник	Жирова Т. О.					<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>		
Гарант	Цензура М.О.							
Розробив	Гамалій Б.С.							