

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційна система підготовки та публікації наукових текстів

Освітньо-кваліфікаційний рівень “бакалавр”
Напрямок підготовки 6.050102 – “комп’ютерна інженерія”

Керівник проекту:

(підпис)

Барбарук В.М.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Деркач А.М.

(ініціали, прізвище)

Група:

КІ-14з

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки 6.050102 – “комп'ютерна інженерія”
(шифр і назва)
Спеціальність _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри КНІ
_____ І.С. Скарга-Бандурова
« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Деркач Анастасії Миколаївни
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система підготовки та публікації наукових текстів

керівник проекту (роботи) Барбарук Віктор Миколайович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " _____ " _____ 201_ р. № _____

2. Термін подання студентом роботи 16.06.2018

3. Вихідні дані до роботи тестові зображення форматів, методи аналізу текстурних зображень, теоретичні відомості про методи застосувань сучасних HTML технологій , середа розробки PHP Storm

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної галузі та постановка задачі, аналіз та

, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2018

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання на дипломне проектування	05.05.2018 - 13.05.2018	
2	Аналіз завдання, робота з літературою.	14.05.2018 - 22.05.2018	
3	Розробка програмної системи	22.05.2018 - 02.06.2018	
4	Тестування програмної системи	02.06 .2018- 11.06.2018	
5	Розробка розділу «Охорона праці»	11.06.2018 - 13.06.2018	
6	Оформлення пояснювальної записки електронних плакатів	13.06.2018 - 16.06.2018	

Здобувач вищої освіти

_____ (підпис)

Деркач А.М.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Барбарук В.М.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка дипломної роботи бакалавра:
76 с., 13 рис., 5 табл., 18 джерел посилань, 2 додатки.

Дана робота присвячена розробці Web портала з застосуванням сучасних HTML технологій. В якості технологій використовуються AngularJS, MVC, JQUERY, SPA. Був побудований алгоритм пошуку та розроблена мова для публікацій наукових статей.

Проведено тестування роботи порталу. Можлива авторизація різних користувачів. Пошук та публікація з потрібною категорією. Додання тегів та своя мова розмітки markdown, яка дозволяє використовувати html теги та вставляти формули. Також використовується веб додаток Single Page Application, який виконується на стороні клієнта у веб браузері.

Була розроблена формула для оцінювання актуальності публікації, воно допоможе враховувати думки авторів о публікації, популярність.

Ключові слова: web портал, ANGULARJS, MVC, MVVM, MARKDOWN, публікації наукових статей, SINGLE PAGE APPLICATIONS,

Умови одержання дипломного проекту: СНУ ім. В. Даля,
пр. Центральний 59-А, м. Сєверодонецьк, 93400.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	10
1.1 Напрямки розвитку сучасних web-технологій.....	10
1.1.1 Поява DHTML	10
1.1.2 Технології, що застосовуються Web-клієнтами	11
1.1.3 Технології створення серверних Web-додатків.....	16
1.2 Застосування технологій Single Page Application (SPA)	19
1.2.1 Загальна архітектура технологій SPA.....	19
1.2.2 Порівняння патернів проектування	23
1.3 Постановка задачі	27
2 ЗАСТОСУВАННЯ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ	
ІНТЕРАКТИВНИХ HTML ЗАСТОСУВАНЬ НА ПРИКЛАДІ WEB-	
ПОРТАЛУ ДЛЯ ПІДГОТОВКИ ТА ПУБЛІКАЦІЇ НАУКОВИХ ТЕКСТІВ..	29
2.1 Визначення загальних вимог до Web-порталу.....	29
2.2 Визначення функціоналу Web-порталу	30
2.3 Ролі та права користувачів	31
2.4 Архітектура Web-порталу	32
2.5 Побудова макету інтерфейсу користувача	35
2.6 Розробка бізнес-логіки клієнтського ПЗ Web-порталу	38
2.7 Проектування програмного забезпечення клієнтської частини.....	43
2.8 Рейтинг статей на сайті	46
3 ОХОРОНА ПРАЦІ	48

3.1 Аналіз потенційно небезпечних і шкідливих виробничих факторів , що впливають на персонал.....	48
3.2 Заходи щодо техніки безпеки	52
3.3 Заходи, що забезпечують виробничу санітарію і гігієну праці.....	55
3.4 Рекомендації по пожежній профілактиці	59
ВИСНОВКИ.....	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	65
ДОДАТОК А. Лістинг коду	67
ДОДАТОК Б. Електронні плакати	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

JS – Java Script

API – Application Programming Interface

AJAX – Asynchronous JavaScript And XML

HTML – Hyper Text Markup Language

CSS – Cascading Style Sheets

MVC – Model View Controller

ВСТУП

На сьогоднішній день комп'ютерні та мережеві технології відіграють все більшу роль в різних областях людської діяльності. Процеси впровадження нових інформаційних технологій впливають на всі її сфери. Швидкий розвиток комунікаційних технологій трансформує багато процесів в сучасному суспільстві.

Інтернет, як найбільш доступна і зручна система глобального обміну інформацією між користувачами не тільки довела свою життєздатність, але і починає витісняти інші способи і канали комунікацій, що відбувається на основі більш низької вартості послуг, високої швидкості передачі даних, більш широкого спектру представленої і переданої інформації.

Сучасні інформаційні системи і системи міжнародного зв'язку дають можливість відійти від традиційної паперової документації як головного носія інформації. Організація електронного інформаційного обміну між користувачами мережі Інтернет має багато переваг, зокрема, можливість швидкого доступу до потрібної інформації в будь-якому місці і в будь-який час, маючи лише вихід до Інтернету.

На сьогоднішній день мільйони людей у світі використовують сучасні технології, які дозволяють їм мати доступ до Інтернету та використовувати безмежні можливості світового інформаційного простору. Саме тому на зміну паперовій документації приходить електронна, з року в рік зростає кількість електронних книг, з'являються електронні бібліотеки.

Багато людей має чим поділитися зі світом, тому переважна більшість сучасних Інтернет-ресурсів повинні мати функціонал, який не лише забезпечує для користувачів можливість перегляду його вмісту, але також і можливість розміщення та редагування власної інформації.

Інтернет-технології на сьогоднішній день, також як і промислові та фінансові технології, визначають засоби і форму, в яких реалізується спільна

діяльність людей заради досягнення певних цілей. Тобто Інтернет-технології це все, що пов'язано з Глобальною мережею, а зокрема і в основному це Web-сторінки в безлічі різних варіантів: форуми, чати, пошта, Інтернет-магазини, новини, фотобанки і т.д. Вміст цих сторінок побудований і працює у відповідності з певними правилами, за певними методам на базі певних програм і технічних засобів (серверів, мереж та ін.).

Перед розробниками сучасних Інтернет-технологій, зокрема, Web-ресурсів, часто постає непросте питання відповідності певним вимогам, а також продуктивності ресурсу, що розробляється. Адже паралельно з впровадженням нових та вдосконаленням існуючих Web-технологій безупинно зростають вимоги до сучасних Web-ресурсів.

Якщо ще 10 років тому переважна більшість Web-сайтів представляла собою набір статичних HTML-сторінок. Сьогодні подібні сайти зустрічаються вкрай рідко, зазвичай так виконані невеликі персональні Web-сайти, а також сайти невеликих компаній, які не претендують ні на що, крім розміщення відносно невеликого обсягу інформації, яка рідко змінюється.

В процесі перетворення Інтернету з набору інформаційних ресурсів на інструмент ведення бізнесу технології створення сайтів істотно змінилися, і на сьогоднішній день більшість Web-сайтів представляють собою набір додатків, яким властиві інтерактивність, засоби персоналізації, засоби взаємодії з клієнтами (наприклад, прийом замовлень і платежів) та партнерами, а нерідко - і засоби інтеграції з «внутрішніми» корпоративними додатками власника сайту, наприклад, деякої компанії.

Із зростанням складності та об'ємів інформації, яка передається та обробляється в ході взаємодії користувача з Web-ресурсом, враховуючи вимоги до функціоналу, який повинен мати ресурс, актуальним залишається питання продуктивності сучасних Web-ресурсів.

Не зважаючи на те, що на сьогоднішній день більшість користувачів мають доступ до швидкісного Інтернету, враховуючи складність сучасних Web-ресурсів, швидкість завантаження сайтів та роботи з ними залишається

актуальним питанням.

Тому актуальним є і питання створення таких Web-ресурсів, які б поєднували у собі зручний інтерфейс користувача з усіма необхідними функціями, можливість одночасної роботи багатьох користувачів, зберігання досить великих об'ємів інформації, і в той же час забезпечували швидкість та надійність роботи із ними.

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Напрямки розвитку сучасних web-технологій

1.1.1 Поява DHTML

На зміну статичним HTML-сторінкам прийшли нові технології, які забезпечили широкі можливості для Web-сайтів [1]. І на сьогоднішній день HTML-сторінки використовуються не лише для відображення деякої статичної інформації, сьогодні – це цілий простір для дій. На сторінках сайтів робляться замовлення, здійснюються розрахунки, демонструється відео, відображається інтерактивна реклама, запускаються онлайн-ігри, тощо.

Важливим кроком у розвитку web-технологій стала поява такої технології як DHTML (Dynamic HTML - Динамічний HTML). Назва DHTML - це комерційний термін, введений Netscape і Microsoft для опису цілого ряду технологій, які були введені в четвертому поколінні Web-браузерів і призначалися для розширення їх динамічних можливостей.

Ці технології були створені для того, щоб обійти ті обмеження, які, як вважалося, існували при розробці Web-сторінок за допомогою HTML. Інтернет досить непогано працював як з текстом, так і з графікою, але люди, які звикли до мультимедіа, чекали чогось більшого. І поява нової технології перевернула уявлення про можливості Web-сторінок.

Створена на основі DHTML сторінка могла змінюватися, не звертаючись до сервера за додатковими даними. Програмісти називають такий код клієнтським, або кодом на стороні клієнта (client-side code).

Користувач, що сьогодні має справу з Web-додатками (а останнім часом - і з Web-сервісами), спілкується з ними за допомогою Інтернет-клієнтів (найчастіше браузерів, але не тільки їх - існують і інші типи клієнтських додатків, наприклад чат-клієнти).

Однією із беззаперечних переваг використання технологій динамічної зміни HTML сторінок без звернення до сервера є швидкість завантаження,

дані технології зручні тим, що дозволяють приховувати, відображати і міняти елементи сторінок, не завантажуючи нові. Це значно збільшує швидкість роботи Web-сайту [2].

Одним з напрямків розвитку сучасних Web-додатків стало розміщення деякої частини логіки додатка (такий як, наприклад, перевірка коректності даних, що вводяться) в самому Web-клієнті, наприклад, в Web-браузері. Зокрема, сучасні Web-браузери здатні інтерпретувати код на скриптових мовах, виконувати Java-аплети і елементи управління ActiveX, використовувати інші спеціалізовані додатки, такі як Macromedia Flash Player.

В даний час більшість браузерів дозволяють працювати з наступними технологіями:

1) Каскадні таблиці стилів (CSS). За допомогою CSS визначаються властивості елементів на сторінці.

2) Фрагменти коду на JavaScript та інших скрипкових мовах. За допомогою них здійснюється управління функціонуванням об'єктів Web-сторінки.

3) Об'єктна модель документа (DOM).

Можна також відмітити, що при розробці Web-додатків часто постає питання відокремлення завдання Web-дизайну від завдань, пов'язаних з реалізацією функціональності додатків. Адже зазвичай при виконанні певних дій на сторінці, що призводять до зміни її динамічного вмісту, загальний дизайн сторін повинен залишатись без змін.

DHTML додатки, які цілком автономні в браузері і можуть функціонувати без серверної підтримки, такої як база даних, часто змушені звертатися до технологій Single Page Applications, або SPA.

1.1.2 Технології, що застосовуються Web-клієнтами

Розглянемо технології, які використовуються Інтернет-клієнтами, зокрема, сучасними Web-браузерами [3].

Одна з основних таких технологій – це скриптові мови. Більшість сучасних Web-браузерів здатні інтерпретувати код, написаний на скриптових мовах, таких як VBScript, JavaScript [2], тощо. Код на цих мовах вбудовується в Web-сторінку і інтерпретується браузером. Типовий приклад застосування скриптових мов - перевірка коректності даних, що вводяться користувачем у відповідні поля HTML-форми, безпосередньо в процесі введення або після нього, без звернення до Web-серверу. Такі приклади застосування скриптових мов можна побачити при заповненні деяких анкет і отриманні повідомлень про те, що не заповнені обов'язкові поля (варто відзначити, що далеко не всі анкети реалізовані подібним чином).

Однак є й інші приклади застосування скриптових мов, що реалізують як чисто дизайнерські ідеї, наприклад кнопки, що змінюють свій вигляд при наведенні на них курсора, «біжучі рядки», так і іншу функціональність, наприклад вбудовані у Web-сторінки засоби звернення до пошукових систем, відображення діалогових панелей, управління іншими об'єктами, вбудованими в Web-сторінку (наприклад, Java-апплетами або елементами управління ActiveX).

Варто зазначити, що код, створений за допомогою скриптових мов, не може працювати самостійно - він виконується в адресному просторі браузера. Крім того, скриптові мови містять обмежений набір засобів (наприклад, вони не володіють засобами доступу до файлової системи).

Наступною технологією, яка використовується з боку клієнта є Java-апплети, які здатні відображати і виконувати майже всі сучасні браузери. Java-апплети – це спеціальні Java-додатки, які користувач отримує у складі Web-сторінки. Ці додатки нерідко включаються до складу Web-сторінок з метою додавання функціональності, яку складно або неможливо реалізувати за

допомогою скриптових мов. Аплети можуть виконуватися на всіх платформах, для яких доступна віртуальна Java-машина.

Аплети зазвичай створюються відповідно до правил, що обумовлюють період їхнього існування та способи взаємодії зі своїм оточенням. Найчастіше ці способи дуже обмежені (наприклад, такі операції, як зчитування і запис файлів, за замовчуванням для аплетів заборонені; якщо ж подібні операції необхідні, дозволу на їх виконання для конкретних аплетів і конкретних файлів описуються на клієнтському комп'ютері; мережевий доступ з аплету можливий тільки до того комп'ютера, з якого він був завантажений; запуск інших програм на комп'ютері користувача з аплетів неможливий). Однак аplet здатний зчитувати значення параметрів (наприклад, кольору, шрифтів, файлів з графічними зображеннями, які використовуються при виконанні аплету) з його Web-сторінки, на якій він міститься, і відповідно до цих параметрів змінювати свою поведінку. Крім того, параметри аплету можна змінювати динамічно за допомогою коду на скриптових мовах, що міститься в складі тієї ж сторінки.

Оскільки аплети реалізують виконання коду на комп'ютері клієнта, вони певною мірою є потенційно небезпечним вмістом. Саме тому всі сучасні браузері мають доступні для користувача засоби обмеження можливостей виконання аплетів.

Деякі з сучасних браузерів (зокрема, Microsoft Internet Explorer) можуть служити контейнерами для елементів управління ActiveX - спеціальних СОМ-серверів, що виконуються в адресному просторі браузера і також містяться у складі Web-сторінки. Це ще одна важлива можливість браузера як Інтернет-клієнта.

За допомогою елементів управління ActiveX, як і за допомогою Java-аплетів, можна реалізувати будь-яку функціональність, в тому числі і несприятливу для комп'ютера користувача, при цьому, на відміну від Java-аплетів, при виконанні елементів управління ActiveX в загальному випадку немає ніяких обмежень на доступ до файлів і інших ресурсів операційної

системи та мережі, а код, що міститься в них, виконується від імені користувача, який їх завантажив. Як і Java-аплети, елементи управління ActiveX можуть зчитувати свої властивості зі сторінки, на якій вони містяться; крім того, властивості елемента управління ActiveX можна міняти динамічно з коду на скриптовій мові, що містяться в складі тієї ж сторінки; в тому ж коді можна обробляти події, що виникають в таких елементах управління.

Звісно, Microsoft Internet Explorer володіє засобами обмеження можливостей виконання елементів управління ActiveX, в тому числі можна обмежити управління ними з коду на скриптових мовах. Проте для контролю безпеки їх виконання є ще один засіб - так званий електронний цифровий підпис. Цифровий підпис поміщається всередину елемента управління ActiveX, для чого потрібна наявність відповідного електронного сертифіката. Електронний підпис, крім відомостей про фірму-виробника, містить і іншу корисну інформацію. Так, наприклад, якщо файл з елементом управління ActiveX після додавання електронного підпису був змінений, то про це буде негайно повідомлено перед запуском такого елемента управління - при додаванні підпису до елемента управління ActiveX відбувається обчислення контрольної суми відповідного файлу. Проте навіть наявність електронного сертифікату не гарантує відсутності потенційно небезпечного вмісту, а лише дозволяє клієнту встановити його джерело.

Слід пам'ятати про те, що при роботі з елементами управління ActiveX і Java-аплетами користувач може не покладатися на антивірусне програмне забезпечення (неважливо, клієнтське воно чи серверне): ознак, характерних для вірусів (таких як здатність впроваджуватися всередину виконуваних файлів і документів), подібні додатки, як правило, не містять. Щоб запобігти шкідливому впливу подібних елементів можна лише заборонити завантаження або виконання відповідного коду або на рівні налаштувань браузера, або на рівні корпоративних або персональних брандмауерів.

І ще однією поширеною технологією такого роду є додатки Macromedia

Flash, котрі на сьогодні являють собою найбільш популярне розширення функціональності Web-браузерів - з їх допомогою багато Web-дизайнерів надають своїм сайтам інтерактивності та оригінальності.

Модель безпеки додатків Flash заснована на тому, що Macromedia Flash Player, як і віртуальна Java-машина, виконує програми в обмеженому адресному просторі, при цьому виконувані програми не мають доступу до файлової системи (крім одного конкретного каталогу, використовуваного Macromedia Flash Player для службових цілей) та інших ресурсів комп'ютера користувача; виключення робиться для мікрофонів і відеокамер, проте користувач повинен дати дозвіл на передачу даних, отриманих з цих пристроїв. Доступ до мережевих ресурсів обмежується доменом, з якого було отримано додаток. Важливою особливістю Flash додатків є те, що вони також можуть управлятися за допомогою коду JavaScript, присутнього на тій же сторінці, де вони розміщені. Проте сам Macromedia Flash Player для Microsoft Internet Explorer є елементом управління ActiveX і використовує можливості елементів управління ActiveX для доступу до властивостей додатків Flash з скриптових мов.

Крім перерахованих вище найбільш популярних засобів розширення функціональності браузерів є і ряд інших засобів, реалізованих зазвичай у вигляді так званих модулів розширення (plug-in). Оскільки модулі розширення також являють собою виконуваний код, сучасні браузери також мають засоби обмеження можливостей, пов'язаних з їх завантаженням та виконанням.

Усі перераховані засоби розширення функціональності HTML-сторінок можуть бути використані в динамічних сторінках, що генеруються серверними Web-додатками. Так, останнім часом широкого поширення набули засоби створення Web-додатків, що виконуються під управлінням Web-серверів і генеруючих динамічні HTML-сторінки з вбудованим у них кодом на скриптових мовах, призначеним для інтерпретації браузером.

1.1.3 Технології створення серверних Web-додатків

З описаного вище можна зробити висновки, що можливості, пов'язані з виконанням коду в Web-клієнтах, можуть бути істотно обмежені як технологічно, так і за допомогою адміністрування та налаштувань. Це в цілому відповідає повністю виправданим вимогам безпеки. Саме тому, поряд з розвитком засобів розширення функціональності браузерів, розвивалися і технології, пов'язані з виконанням коду додатків не в браузерах, а на самих Web-серверах. Розглянемо найбільш поширені з цих технологій.

Common Gateway Interface (CGI) - це стандартний інтерфейс, що дозволяє виконувати серверні додатки, що викликаються через URL [4]. Вхідною інформацією для таких додатків служить вміст HTTP-заголовка або тіло запиту, залежно від застосовуваного протоколу. CGI-додатки генерують HTML-код, який повертається браузеру. Свого часу широко використовувався термін «CGI-скрипт», походження якого пояснюється тим, що подібні додатки писалися на скриптових мовах типу Perl, що виконуються, тим не менш, не в браузері, а на сервері. CGI-програми можна створювати за допомогою практично будь-якого засобу розробки, генеруючого консольні додатки для операційної системи, під управлінням якої функціонує Web-сервер.

Основна проблема всіх CGI-додатків полягає в тому, що при кожному клієнтському запиті сервер завантажує цей додаток в окремий адресний простір, а потім ініціює його виконання і вивантаження. Ця особливість обмежує продуктивність додатків і можливість одночасної обробки великої кількості клієнтських запитів.

Проблему обмеженої продуктивності Web-додатків, які виконуються в окремому адресному просторі, можна вирішити, створивши додаток у вигляді бібліотеки, що завантажується в адресний простір Web-сервера і при

необхідності залишається там для обробки наступних запитів від інших клієнтів; звісно, в цьому випадку Web-сервер повинен підтримувати завантаження таких бібліотек. Подібні додатки для Microsoft Internet Information Service носять назву ISAPI (Internet Server Application Program Interface), а для вельми популярного Web-сервера Apache такі бібліотеки називаються Apache DSO (Dynamic Shared Objects).

Недоліком такого підходу є те, що при створенні як CGI-, так і ISAPI-додатків досить складно відокремити завдання Web-дизайну від завдань, пов'язаних з реалізацією функціональності і логіки додатків, - подібні додатки генерують Web-сторінки повністю, тому всі дані, пов'язані з дизайном цих сторінок, повинні в загальному випадку міститися всередині виконуваного файлу.

Черговим кроком у розвитку технологій створення Інтернет-додатків стала поява засобів, що дозволяють відокремити завдання Web-дизайну від завдань, пов'язаних з реалізацією функціональності додатків. Першою з таких технологій стала Active Server Pages (ASP) [5], побудована на основі ISAPI-фільтра. Основна ідея ASP полягає у створенні Web-сторінок з вбудованими в них фрагментами коду на скриптових мовах. Однак, на відміну від розглянутих вище засобів застосування скриптових мов для розширення функціональності браузерів, зазначені фрагменти коду інтерпретуються не браузером, а сервером (точніше, спеціальною призначеною для цього ISAPI-бібліотекою), і результат виконання цих фрагментів коду виконує заміну самого фрагменту коду в тій версії сторінки, яка передається в користувальницький браузер.

Незабаром після ASP з'явилися й інші технології, що реалізують ідею розміщення всередині Web-сторінки коду, виконуваного Web-сервером. Найбільш відомою з них сьогодні є технологія JSP (Java Server Pages), основна ідея якої - одноразова компіляція Java-коду (сервлета) при першому зверненні до нього, виконання методів цього сервлета і приміщення результатів виконання цих методів в набір даних, що відправляються в

браузер. Також ще однією популярною технологією подібного типу є PHP (Personal Home Pages), яка використовує CGI-додатки, що інтерпретують впроваджений в HTML-сторінку код на скриптовій мові.

Новітньою версією технології Active Server Pages є ASP .NET, ключова в архітектурі Microsoft .NET Framework. Основна відмінність цієї технології від ASP з точки зору архітектури додатків полягає в тому, що код, присутній на Web-сторінці, не інтерпретується, а компілюється і кешується, а це сприяє підвищенню продуктивності додатків. За допомогою ASP.NET можна створювати Web-додатки і Web-сервіси, які не тільки дозволяють реалізувати динамічну генерацію HTML-сторінок, але і інтегруються з серверними компонентами і можуть використовуватися для вирішення широкого кола бізнес-завдань, що виникають перед розробниками сучасних Web-додатків.

У загальному випадку клієнтом Web-сервера може бути не тільки персональний комп'ютер, оснащений звичайними Web-клієнтами (наприклад, Web-браузером), але і мобільні пристрої, що відрізняються обмеженим розміром екрану, малим обсягом пам'яті, а нерідко і неможливістю відображення графіки. Для цих пристроїв існують свої протоколи передачі даних (Wireless Access Protocol, WAP) і відповідні мови розмітки (WML, Wireless MarkupLanguage, XHTML, Compact HTML і т.п.). При цьому необхідно передавати дані на мобільний пристрій у відповідному форматі, для чого нерідко створюються спеціальні сайти (наприклад, з підтримкою WAP та WML). Більш зручним є створення додатків, які здатні генерувати той чи інший код залежно від типу клієнта. Саме такий підхід і реалізований в Microsoft ASP.NET.

З ростом обсягу використовуваних даних і числа відвідувачів Web-сайтів зростають вимоги до надійності, продуктивності і масштабованості Web-додатків. Для задоволення цих вимог бізнес-логіка, що реалізується в Web-додатку, а також сервіси обробки даних і реалізації транзакцій, відокремлюються від інтерфейсу додатків і переносяться на сервер додатків у вигляді бізнес-об'єктів. Сервери додатків і відповідні бізнес-об'єкти можуть

бути різного типу (найбільш поширеними з них сьогодні є сервери, що підтримують специфікацію Java2 Enterprise Edition, і сервери, що базуються на технологіях COM і Microsoft.NET). Бізнес-об'єкти часто надають доступ до даних корпоративних інформаційних систем або реалізують яку-небудь частину їх функціональності, здійснюючи функції інтеграції Web-додатків з іншими додатками, що використовуються на підприємстві

1.2 Застосування технологій Single Page Application (SPA)

1.2.1 Загальна архітектура технологій SPA

Модель SPA орієнтована на створення інтерактивної Web-сторінки, що не перезавантажується під час сеансу роботи додатка. Всі взаємодії користувача та зміни стану додатка обробляються в контексті одного Web-документа. Архітектура SPA-застосунку є модульною [6]. Модульна архітектура складається з наступних компонентів:

- базова бібліотека - це основа, на якій будується додаток;
- модуль - незалежна одиниця функціональності на сторінці, яка складається з бізнес-логіки й певних елементів інтерфейсу;
- шаблони найбільш частих дій, які може виконувати модуль (взаємодія з іншими модулями, створення запитів Ajax, приєднання і від'єднання обробників подій);
- ядро - керує життєвим циклом модуля (запуск і зупинка), взаємодією між модулями та обробкою помилок.

Головними складовими модуля є бізнес-логіка та елементи інтерфейсу. На сьогоднішній день в модулях SPA-застосунків часто використовується бібліотека jQuery, яка фокусується на взаємодії JavaScript та HTML. Бібліотека jQuery допомагає легко отримувати доступ, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними, а також надає

зручний API по роботі з AJAX.

Модульна архітектура має наступні переваги:

- пакети визначаються одним спільним публічним інтерфейсом;
- імена є локальними всередині одного пакета;
- деталі реалізації недоступні поза межами пакету;
- порядок завантаження не має значення;
- залежності задаються явно;
- кожен файл представляє один модуль;
- запуск з командної строки без браузера.

В моделі SPA сервер виконує наступні функції:

- віддає статичну частину клієнту;
- віддає дані (XML, JSON і т.п.).

Клієнт в свою чергу:

- агрегує дані;
- будує об'єкти;
- шаблонізує;
- створює DOM;
- контролює View.

Таким чином, вся обробка даних відбувається саме на стороні клієнта.

Варто відмітити, що іноді при реалізації SPA архітектури використовується ще хмарний сервіс [7]. В даному випадку сервер виконує функцію передачі та зберігання статичних даних, а також пере направлення запитів. А зберігання даних, їх обробка та обчислення виконується на хмарі.

З усього вищезазначеного можна зробити висновок, що додаток, побудований з використанням технологій SPA - це web-додаток, розміщений на одній сторінці, яка для забезпечення роботи завантажує всі javascript-файли (модулі, віджити, контроли і т.д.), а також файли CSS разом із завантаженням самої сторінки [8, 9].

Архітектура SPA ефективна і для досить складних додатків, які містять багатий функціонал, наприклад, система електронного документообігу, і в

яких кількість файлів зі скриптами може досягати декількох сотень, а іноді навіть тисяч. Досягається це за рахунок наявності інтерфейсу AMD. При завантаженні сайту всі сотні й тисячі файлів зі скриптами не завантажуються одразу, AMD реалізує можливість завантаження скриптів на вимогу. Наприклад, якщо для «головної сторінки» Web-сайту, написаного за допомогою SPA знадобилося 3 скрипта, вони будуть завантажені одразу перед стартом програми. А якщо користувач перейшов на іншу сторінку Web-сайту, наприклад, «Контакти», то принцип AMD завантажить модуль (скрипт + розмітка) тільки перед переходом на цю сторінку.

Головні причини, чому зручно застосовувати технології SPA при розробці Web-ресурсів, таких як Web-портал, який повинен бути розроблений в ході роботи, перераховані нижче.

1) Додатки написані з використанням технологій SPA відмінно працюють на пристроях як стаціонарних, так і мобільних. Персональні комп'ютери, планшети, смартфони, і навіть прості телефони (деякі) можуть безперешкодно працювати з сайтами побудованими за принципом SPA. Можливість роботи на великій кількості пристроїв дозволяє значно розширити аудиторію користувачів у порівнянні з використанням інших підходів.

2) Можливість розробки багатого користувальницького інтерфейсу. Оскільки web-сторінка одна, побудувати багатий, насичений користувальницький інтерфейс набагато простіше. Простіше зберігати інформацію про сеанс, управляти станами представлень (views), керувати анімацією (в деяких випадках), тощо.

3) Технології SPA дозволяють істотно (у рази) скоротити повторні завантаження одного і того ж контенту. Якщо сайт (або портал) використовує шаблон, то разом з основним змістом якої-небудь сторінки відвідувач сайту обов'язково завантажує розмітку шаблону. Таким чином досягається скорочення витрат часу і ресурсів на кешування у порівнянні з іншими підходами.

4) Існує велика кількість базових бібліотек (фреймворків - від англійського слова *framework* - "основа, структура, каркас"), які реалізують принцип *Single Page Application*. Їх призначення: забезпечення базових принципів для SPA розробки, мінімізуючи трудовитрати на рішення універсальних задач; використання досвіду створення сайтів багатьох програмістів.

5) Принципи будь-якого фреймворку, який реалізує парадигму SPA повинні дотримуватися наступних понять і визначень:

6) SPA підтримує клієнтську навігацію. Всі переходи користувача між модулям-сторінкам однозначно фіксуються в історії навігації, причому навігація є «глибокою», тобто, якщо користувач скопіює і відкриє посилання на внутрішню модуль-сторінку в іншому браузері або вікні, він потрапить на відповідну сторінку.

7) SPA розміщується на одній web-сторінці, отже все необхідне для роботи сайту (порталу): скрипти і стилі повинні бути визначені в одному місці проекту - на єдиній web-сторінці.

8) SPA зберігає постійний стан (важливі змінні) роботи клієнта (клієнтського скрипта) в кеші браузера або в *Web Storage*.

9) SPA завантажує всі скрипти, що вимагаються для старту додатку при ініціалізації web-сторінки.

10)

РА поступово підвантажує модулі за вимогою.

Фреймворки, реалізують SPA, використовують різні патерни проектування. Для того, щоб визначитись, який з них краще підійде для вирішення поставленої задачі, варто більш детально розглянути кожен з них.

Сучасні SPA-додатки зазвичай структуровані наступним чином.

1) Жодний стан / дані не зчитуються з DOM. Додатки повертають HTML та операції над елементами, але нічого не зчитують з DOM. При зберіганні стану в DOM складно дуже швидко отримати доступ до елемента: набагато зручніше мати одне місце де зберігаються дані і надавати інтерфейс

доступу до даних, особливо коли ті ж самі дані повинні бути представлені в декількох місцях в інтерфейсі.

2) Замість того щоб зберігати дані в DOM або у деяких об'єктах, є велика кількість моделей в пам'яті, які представляють всі стани / дані у додатку.

3) Подання відображають зміст моделей. Коли кілька подань залежать від однієї моделі, не потрібно вручну відстежувати кожне залежне від моделі подання. Замість цього є система подій змін, через які подання отримують повідомлення про зміни від моделей і змінюються відповідно до змін моделі.

4) Замість того, щоб робити речі глобальними, створюються невеликі підсистеми, що не залежать одна від одної.

5) Для забезпечення багатоплатформності мінімізується та ізолюється DOM-залежний код.

1.2.2 Порівняння патернів проектування

Model-View-Controller (MVC) - це фундаментальний патерн, який знайшов застосування в багатьох технологіях, дав розвиток нових технологій і кожен день полегшує життя програмістам [10]. MVC складається з трьох компонент: подання (View, користувальницький інтерфейс), модель (Model, бізнес-логіка) і контролер (Controller, містить логіку на зміну моделі при певних діях користувача). Основна ідея цього патерну полягає в тому, що і контролер і подання залежать від моделі, але модель ніяк не залежить від цих двох компонент. Саме це дозволяє розробляти і тестувати модель, нічого не знаючи про подання і контролер. В ідеалі контролер так само нічого не повинен знати про подання (хоча на практиці це не завжди так), і в ідеалі для одного подання можна перемикаати контролери, а так само один і той же контролер можна використовувати для різних подань (так, наприклад,

контролер може залежати від користувача, який увійшов в систему). Схема патерну MVC наведена на рис. 1.1.

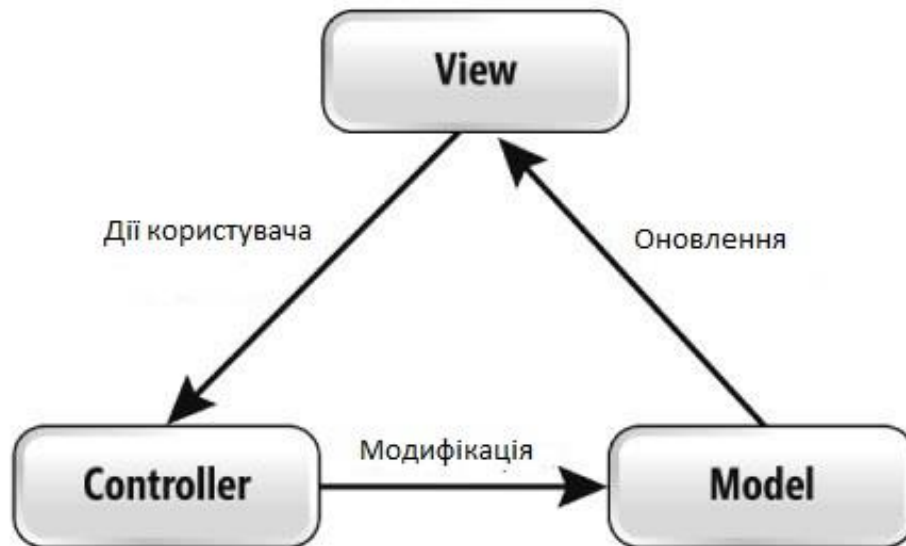


Рисунок 1.1 – Схема патерну проектування MVC

Користувач бачить уявлення, на ньому ж виконує якісь дії, ці дії подання перенаправляє контролеру і підписується на зміну даних моделі, контролер в свою чергу виконує певні дії над моделлю даних, подання отримує останній стан моделі і відображає її користувачеві.

Ознаки контролера:

- контролер визначає, яке подання має бути відображено в даний момент;
- події подання можуть вплинути тільки на контролер, контролер може вплинути на модель і визначити інше подання.

Контролер перехоплює подію ззовні (викликану діями користувача) і відповідно до закладеної в нього логіки реагує на цю подію, змінюючи модель, за допомогою виклику відповідного методу. Після зміни модель використовує подію щоб повідомити про зміни, і всі підписані на цю подію подання, отримавши повідомлення, звертаються до моделі за оновленими даними, після чого їх і відображають.

Даний патерн використовується, наприклад, в MVC ASP.NET, Angular JS.

Model-View-Presenter (MVP) патерн також складається з трьох компонент [11]. Але в даному випадку поданням немає потреби підписуватися на зміни моделі, тепер контролер, названий в даному підході Presenter (презентер) дає знати поданням про зміни.

Даний підхід дозволяє створювати абстракцію подання. Реалізувати даний патерн можна за допомогою винесення інтерфейсів подання. У кожного подання буде інтерфейс з певним набором методів і властивостей, потрібних презентеру, який в свою чергу ініціалізується даним інтерфейсом, підписується на події подання і за необхідності віддає дані. Даний підхід дозволяє розробляти програми з використанням методології TDD (Test-driven development).

Ознаки презентера:

- двостороння комунікація з поданням;
- представлення взаємодіє безпосередньо з презентером, шляхом виклику відповідних функцій або подій презентера;
- презентер взаємодіє з поданням шляхом використання спеціального інтерфейсу, реалізованого поданням;
- один примірник презентера пов'язаний з одним поданням (на відміну від MVC).

Кожне подання має реалізовувати відповідний інтерфейс. Інтерфейс подання визначає набір функцій і подій, необхідних для взаємодії з користувачем. Презентер повинен мати посилання на реалізацію відповідного інтерфейсу, яке зазвичай передають в конструкторі.

Логіка подання повинна мати посилання на екземпляр презентера. Всі події подання передаються для обробки в презентер і практично ніколи не обробляються логікою подання (в т.ч. створення інших подань).

Приклад використання цього патерну Windows Forms.

Патерн Model-View-ViewModel (MVVM) використовується в WPF і

Silverlight [11]. Тут також присутні три компоненти: модель (як і в MVC представляє дані предметної області), подання (відображає модель подання і посилає дії користувача моделі подання) і третій компонент - додаткова модель під назвою ViewModel (модель подання, представляє абстрактне відображення подання). Схема патерну MVVM наведена на рис. 1.2.

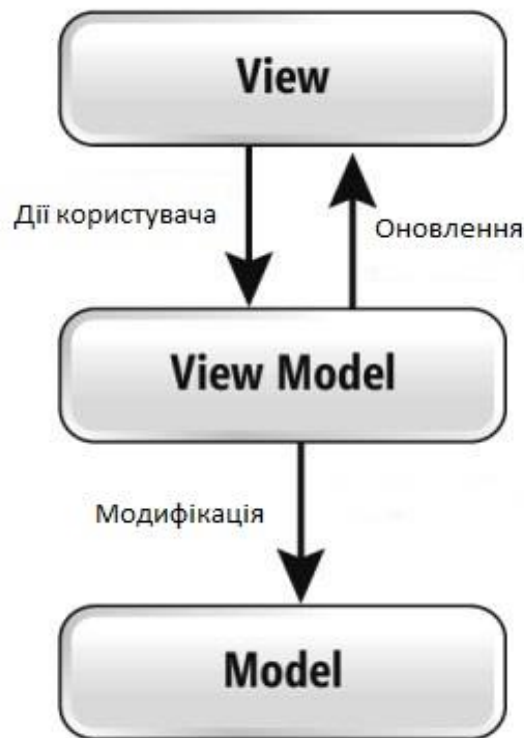


Рисунок 1.2 – Схема патерну проектування MVVM

Даний патерн підходить до таких технологій, де присутня двостороння синхронізація елементів управління на модель, як в WPF. Відмінність від MVP та MVC патернів полягає в тому, що властивості подання повинні знаходитись не в поданні, а в моделі подання (ViewModel), і вони повинні синхронізуватися з необхідними полями в поданні. В цьому і є основна ідея WPF. Модель подання - це деякий суперконвертор, який перетворює дані моделі в подання, в ньому описуються основні властивості подання, а також логіка взаємодії з моделлю.

Ознаки моделі подання:

- двостороння комунікація з поданням;
- модель подання - це абстракція подання, зазвичай це означає, що властивості подання збігаються з властивостями моделі подання/моделі;
- модель подання не має посилання на інтерфейс подання, зміна стану моделі подання автоматично змінює подання і навпаки, оскільки використовується механізм скріплення даних;
- один екземпляр моделі подання пов'язаний з одним відображенням.

При використанні цього патерну, подання не реалізує відповідний інтерфейс. Подання має мати посилання на джерело даних, яким у даному випадку є модель подання. Елементи подання пов'язані з відповідними властивостями і подіями моделі подання. У свою чергу, модель подання реалізує спеціальний інтерфейс, який використовується для автоматичного оновлення елементів подання.

В JavaScript-інфраструктурі MVVM поданням є розмітка, а моделлю подання - код.

1.3 Постановка задачі

Метою роботи є дослідження сучасних технологій розробки інтерактивних HTML застосувань та вибір оптимальних методів та засобів розробки Web-порталу для підготовки та публікації наукових текстів.

Для досягнення мети в рамках ВКР необхідно вирішити такі завдання:

- вивчити і проаналізувати предметну область;
- зробити порівняльний аналіз існуючих рішень та обрати оптимальний варіант;
- визначити основні функції Web-порталу;
- спроектувати архітектуру Web-порталу;
- описати інтерфейс користувача, створити макет інтерфейсу;

– розробити алгоритми роботи логіки системи, описати запити, що використовуватимуться в системі та підсистемах.

Також в рамках ВКР потрібно проаналізувати умови роботи на робочому місці користувача ПК, на якому проектуватиметься та розроблятиметься Web-портал, перевірити відповідність цих умов нормативним вимогам.

2 ЗАСТОСУВАННЯ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ІНТЕРАКТИВНИХ HTML ЗАСТОСУВАНЬ НА ПРИКЛАДІ WEB-ПОРТАЛУ ДЛЯ ПІДГОТОВКИ ТА ПУБЛІКАЦІЇ НАУКОВИХ ТЕКСТІВ

2.1 Визначення загальних вимог до Web-порталу

Для того, щоб будь-який портал працював ефективно, надійно і гнучко, а також мав досить тривалий життєвий цикл, він повинен задовольняти наступним загальним вимогам:

- інтегрованість, наявність можливості інтеграції з іншими прикладними системами та базами даних;
- багатоплатформеність (різні операційні системи та апаратні засоби);
- масштабованість за кількістю користувачів, обсягом збережених даних, інтенсивності обміну даними, швидкості обробки запитів і даних, способам забезпечення доступу і т.п. ;
- надійність (постійна працездатність, мінімізація часу простою і часу відновлення, забезпечення засобами збереження і відновлення даних, резервування);
- безпека - захист даних, додатків і транзакцій, включаючи внутрішню і зовнішню захист для запобігання несанкціонованого доступу до мережі; унікальність реєстрації при санкціонуванні доступу до захищених і персоналізованих порталних додатків на рольовій основі;
- забезпечення швидкої реакції порталу на запити користувачів.

Враховуючи переваги технологій SPA, задля задоволення таких вимог як багатоплатформеність, масштабованість та забезпечення швидкої реакції порталу на запити користувачів, при розробці доцільно використовувати саме ці технології.

Для забезпечення виконання вимог до безпеки порталу, повинна бути створена система авторизації користувачів, можливість підтримки різних

ролей користувачів.

Оскільки для розробки використовуватиметься одна із сучасних платформ, які зазвичай містять засоби інтеграції з іншими продуктами, наприклад, з БД, на цьому питанні можна детально не зупинятись.

Для забезпечення надійності роботи з ресурсом з програмних засобів можна відмітити наявність перевірки помилок, чітко визначити режими доступу до ресурсів (щоб уникнути такої ситуації, коли, наприклад, два користувача одночасно вносять зміни до однієї статті), а також можливість періодичного авто збереження статті, що редагується.

2.2 Визначення функціоналу Web-порталу

Після того як визначено загальні вимоги, які повинен задовольняти будь-який Web-портал, можна перейти до визначення функціональних вимог до порталу, що розробляється.

Портал призначений для підготовки та публікації (під публікацією мається на увазі публікація їх в мережі Інтернет, тобто, представлення на загальний огляд) наукових текстів, тому він повинен виконувати наступні функції:

- зберігання великих об'ємів інформації, з часом додаватимуться нові статті, відповідно об'єми інформації, розміщеної на порталі, зростатимуть;
- підтримка пошуку – забезпечує користувачам та гостям доступ до широкого спектру пошуку всередині порталу;
- застосування засобів авторизації доступу до інформації для різних груп користувачів;
- організація інтерактивних діалогів користувачів порталу (чати, дискусії, голосування, коментарі і т.д.);
- персоніфікація доступної інформації у відповідності з правами та

привілегами користувачів, ідентифікація користувача відбувається при аутентифікації;

- перегляд інформації (яка інформація доступна до перегляду, визначається групою користувача);
- розміщення, видалення та редагування елементів (статей);
- інтерактивне редагування інформації, в режимі редагування статті користувачу повинні бути доступні перевірка тексту на помилки, додавання формул, спец символів, графіків.

2.3 Ролі та права користувачів

Засоби персоналізації дозволяють вибрати інформаційне наповнення сторінок для різних користувачів на основі профайлів цих користувачів або на основі бізнес-логіки.

Web-портал призначений для публікації наукових статей, тому важливим є питання захисту інформації, яку користувачі розміщуватимуть на порталі. Авторизація користувачів та чітке визначення ролей і відповідного набору привілежій для кожної ролі - важлива складова комплексу засобів безпеки.

Нижче наведено список ролей користувачів порталу зі списком доступних для них прав:

1) Гість (незарєєстрований користувач):

- перегляд опублікованих статей;
- пошук інформації (серед опублікованих статей).

2) Зареєстрований користувач:

- перегляд опублікованих статей;
- додавання коментаріїв до опублікованих статей;
- пошук інформації (серед опублікованих статей та своїх статей);

- участь в дискусіях та чатах на порталі;
- доступ до особистого кабінету;
- розміщення статей;
- видалення своїх статей;
- редагування своїх статей;
- передача статті на публікацію;
- аналіз популярності статі.

3) Редактор статей:

- перегляд опублікованих статей;
- перегляд статей, переданих до публікації;
- додавання коментаріїв до опублікованих та переданих на публікацію статей;
- пошук інформації (серед опублікованих переданих на публікацію статей);
- редагування статей, переданих на публікацію;
- участь в дискусіях та чатах на порталі;
- доступ до особистого кабінету.

підтвердження публікації статті (в результаті чого стаття публікується) та відхилення публікації (в даному випадку стаття не публікується і повертається автору, автор повинен отримати відповідне повідомлення з коментарями редактора).

2.4 Архітектура Web-порталу

Оскільки портал повинен зберігати велику кількість інформації, для зберігання статей та іншої інформації доцільно використовувати базу даних.

Оскільки портал розробляється з використанням технологій SPA, основна частина функцій повинна бути реалізована на клієнтській стороні.

Загальна схема архітектури ПЗ Web-порталу представлена на рис. 2.1.

Як видно з рис. 2.1, частина програмного забезпечення все ж таки знаходиться на серверній стороні. В основному це модулі, які обробляють запити, пов'язані з базою даних. Також на стороні сервера обробляються і запити на оновлення чатів, адже ці дані повинні синхронно оновлюватись у всіх (або кількох) користувачів.

Все що пов'язано з переглядом та редагуванням статей виконується на стороні клієнта аж поки не буде запит на збереження внесених змін.

Для уникнення втрати даних у випадку непередбачуваних обставин, варто передбачити ще один спеціальний модуль автоматичного збереження даних. Цей модуль також знаходитиметься на стороні клієнта, і якщо стаття знаходиться в режимі редагування, користувач вносить зміни, то періодично (наприклад, раз на 10 хвилин) відправлятиме тимчасову копію статті на сервер, де вона тимчасово зберігатиметься без звернення до бази даних.

В системі передбачені різні ролі користувачів, для яких на одних і тих самих сторінках повинні відображатись різні елементи. Наприклад, і користувачі і редактори мають доступ до сторінки «Особистий кабінет», проте користувач в особистому кабінеті повинен бачити список своїх статей, а редактор – список статей, надісланих авторами для публікації. Відрізняються для цих ролей і елементи управління статтями, так у користувача повинні бути інструменти для створення нової статті або видалення існуючої, а у редактора – елементи на підтвердження та відхилення публікації статті, тощо.

Враховуючи особливість патерну MVC, що дозволяє використовувати різні контролери для одного й того самого подання, на відміну від інших патернів, доцільно використовувати саме цей патерн в розробці Web-порталу. Тоді при проходженні авторизації користувачем, на сервері буде обрано контролер, що відповідає ролі даного користувача, і передано клієнтській частині саме цей контролер.

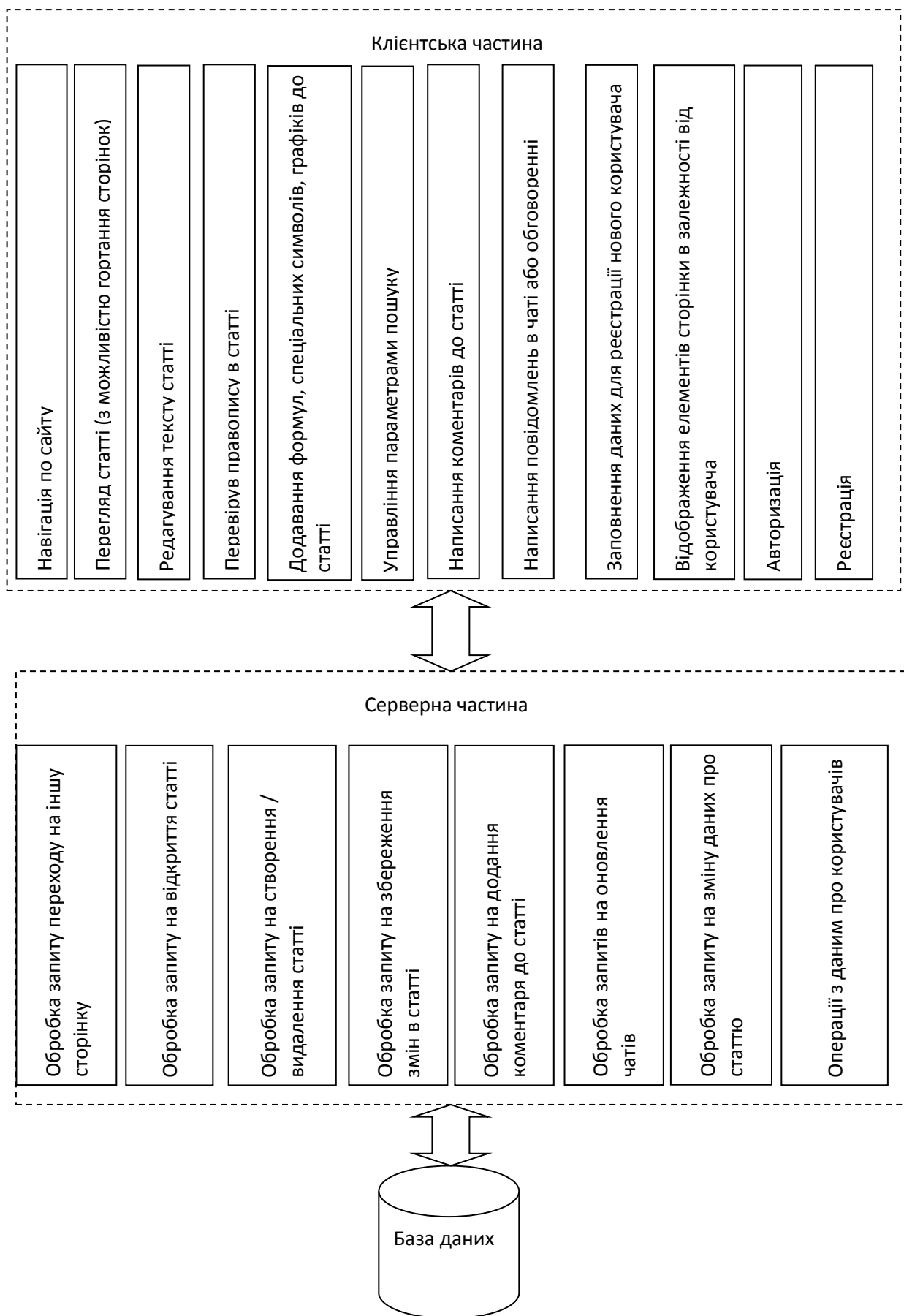


Рисунок 2.1 – Загальна схема архітектури ПЗ Web-порталу

2.5 Побудова макету інтерфейсу користувача

Загальна схема інтерфейсу Web-порталу для підготовки та публікації наукових текстів представлена на рис. 2.2.

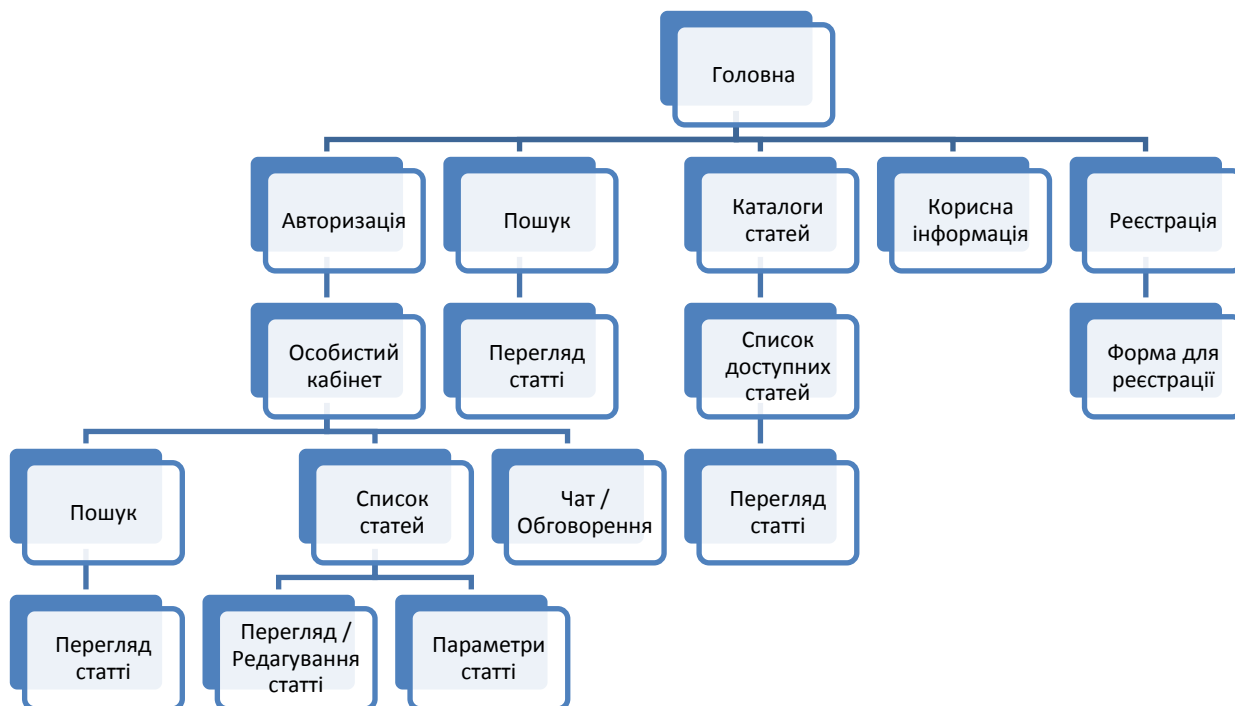


Рисунок 2.2 – Схема інтерфейсу Web-порталу для підготовки та публікації наукових текстів

Для більш зручної роботи, на кожній сторінці повинна бути панель навігації з посиланнями на всі доступні сторінки, за допомогою якої користувач може перейти на потрібну йому сторінку.

Також меню пошуку та авторизації теж зручно зробити на кожній сторінці.

На рис. 2.3 – 2.4 зображені макети сторінок «Головна» та «Список доступних сайтів».

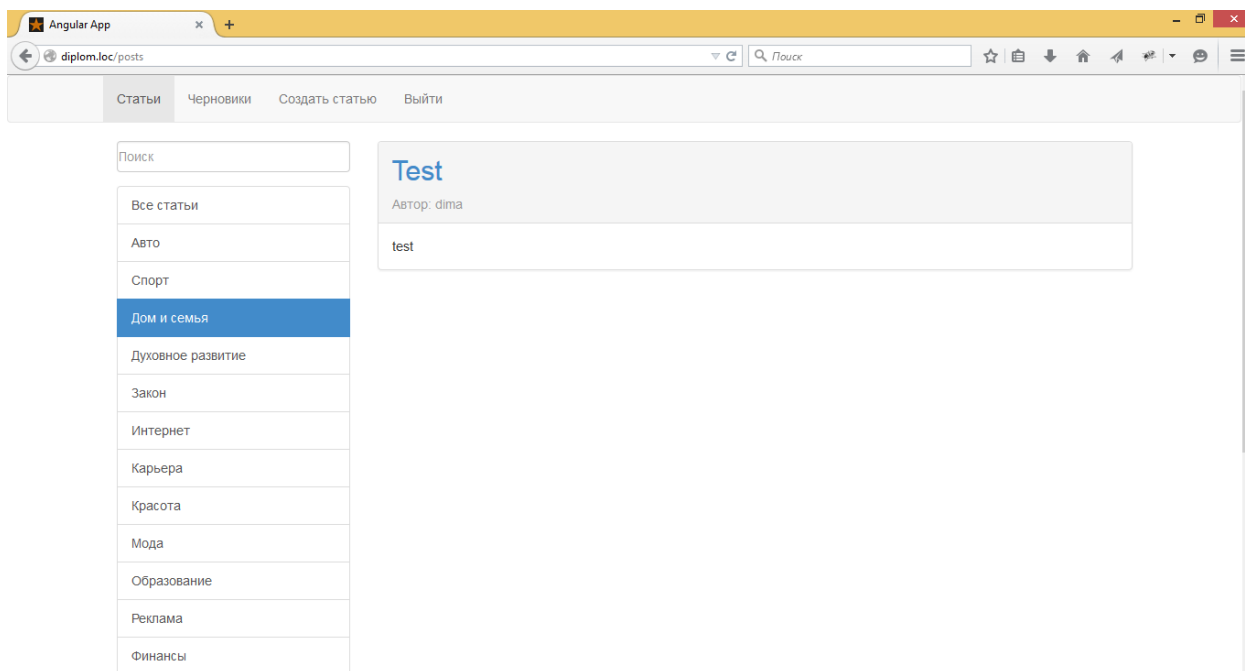


Рисунок 2.3 – Макет сторінки «Головна»

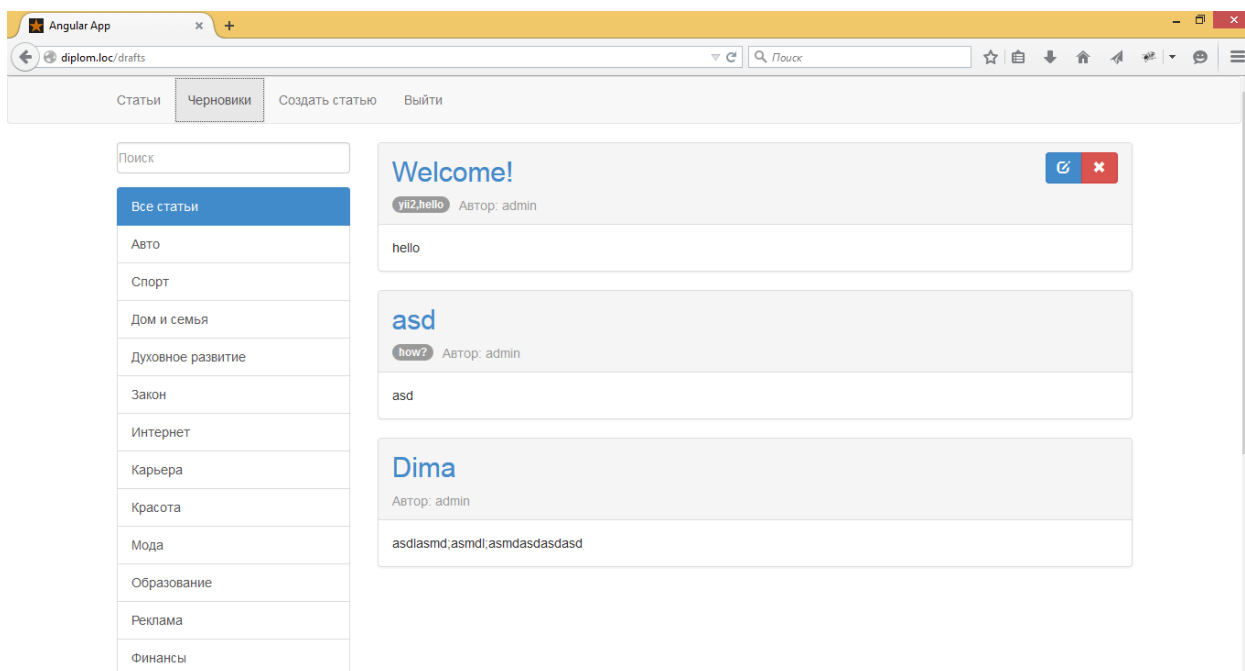


Рисунок 2.4 – Макет сторінки «Список черновиків»

Як видно з рисунків 2.3 – 2.4, сторінки досить схожі між собою. Сторінка «Список доступних сайтів» відрізняється лише тим, що на ній відображається список статей, які можна переглянути.

Хоча на рис. 2.2 перегляд та редагування статті представлені однією

сторінкою, насправді, вигляд сторінки відрізняється для цих випадків. На рисунках 2.5 – 2.6 представлено макети цих двох сторінок.

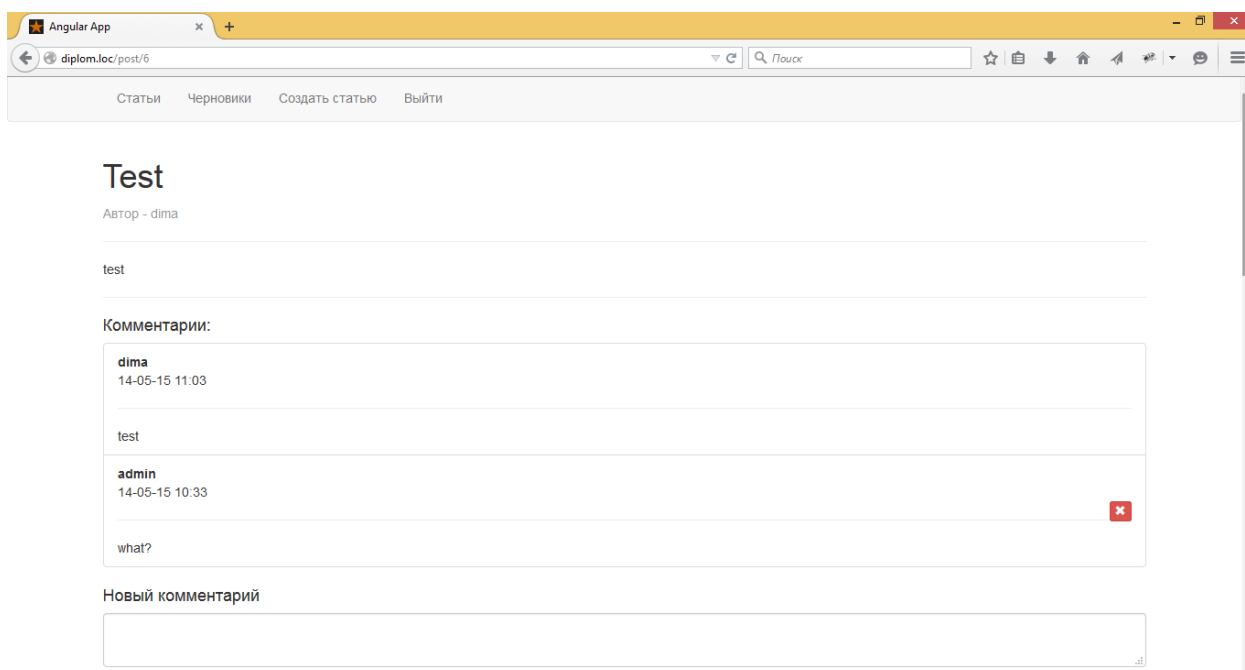


Рисунок 2.5 – Макет сторінки «Перегляд статті»

Варто зазначити, що форма авторизації повинна бути доступна лише якщо відвідувач порталу не авторизувався. Якщо користувач пройшов аутентифікацію, форма авторизації відображатись не повинна. На рис. 2.6 зображено редагування статі.

Меню авторизації в режимі редагування не відображається, тому що редагувати статті можуть лише авторизовані користувачі.

Меню редагування статті – це інтерактивне меню, в якому можна обрати такі дії як додати формулу, графік і т.п.

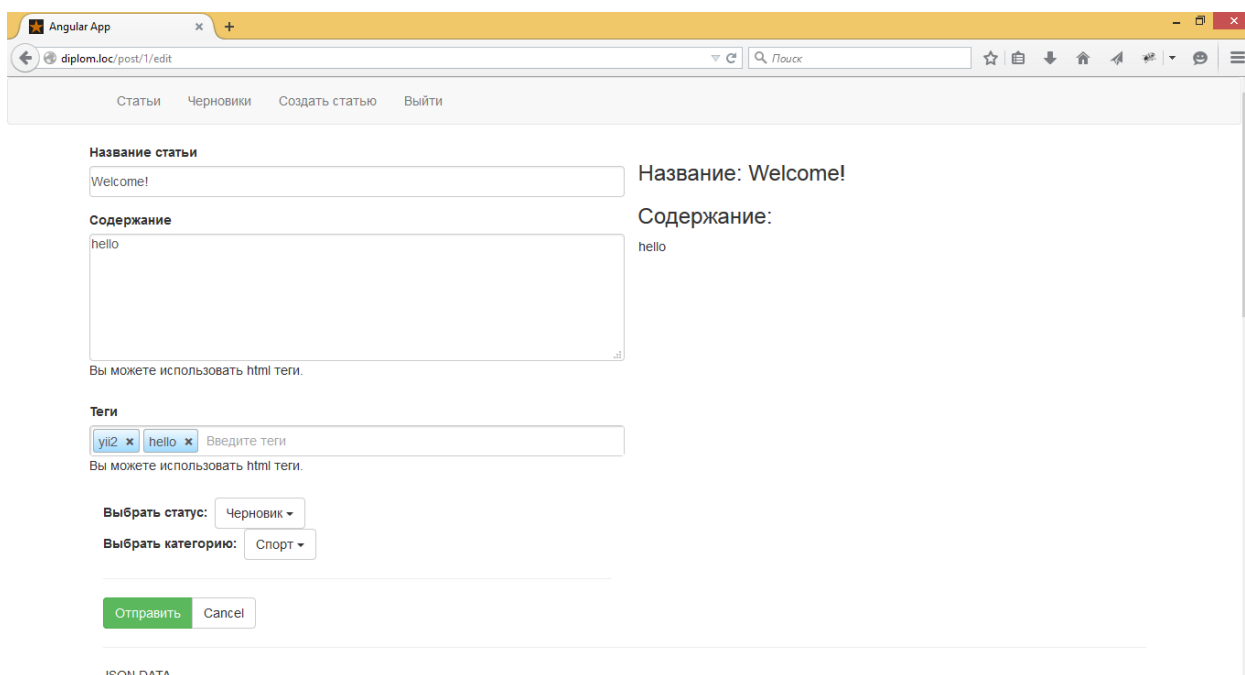


Рисунок 2.6 – Макет сторінки «Редагування статті»

2.6 Розробка бізнес-логіки клієнтського ПЗ Web-порталу

Для розробки бізнес-логіки роботи Web-порталу потрібно чітко визначити дії користувачів, які вони виконуватимуть на порталі, а також реакцію програмного забезпечення на це.

Оскільки для різних ролей користувачів порталу будуть розроблені різні контролери, описуючи діяльність, варто враховувати різних користувачів.

На рис. 2.8 зображена діаграма діяльності для авторизації та реєстрації.

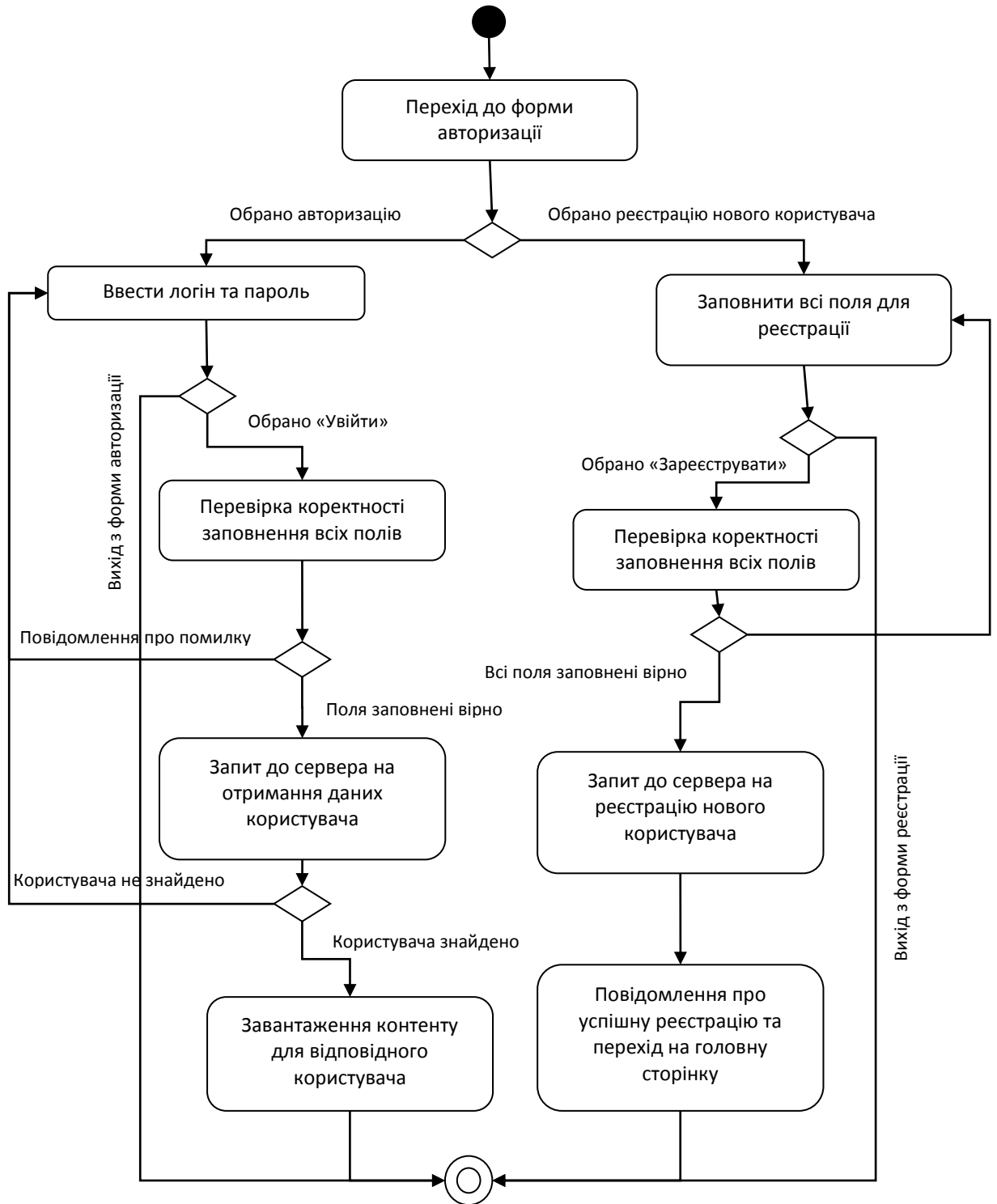


Рисунок 2.8 – Діаграма діяльності для переходу користувача до форми авторизації

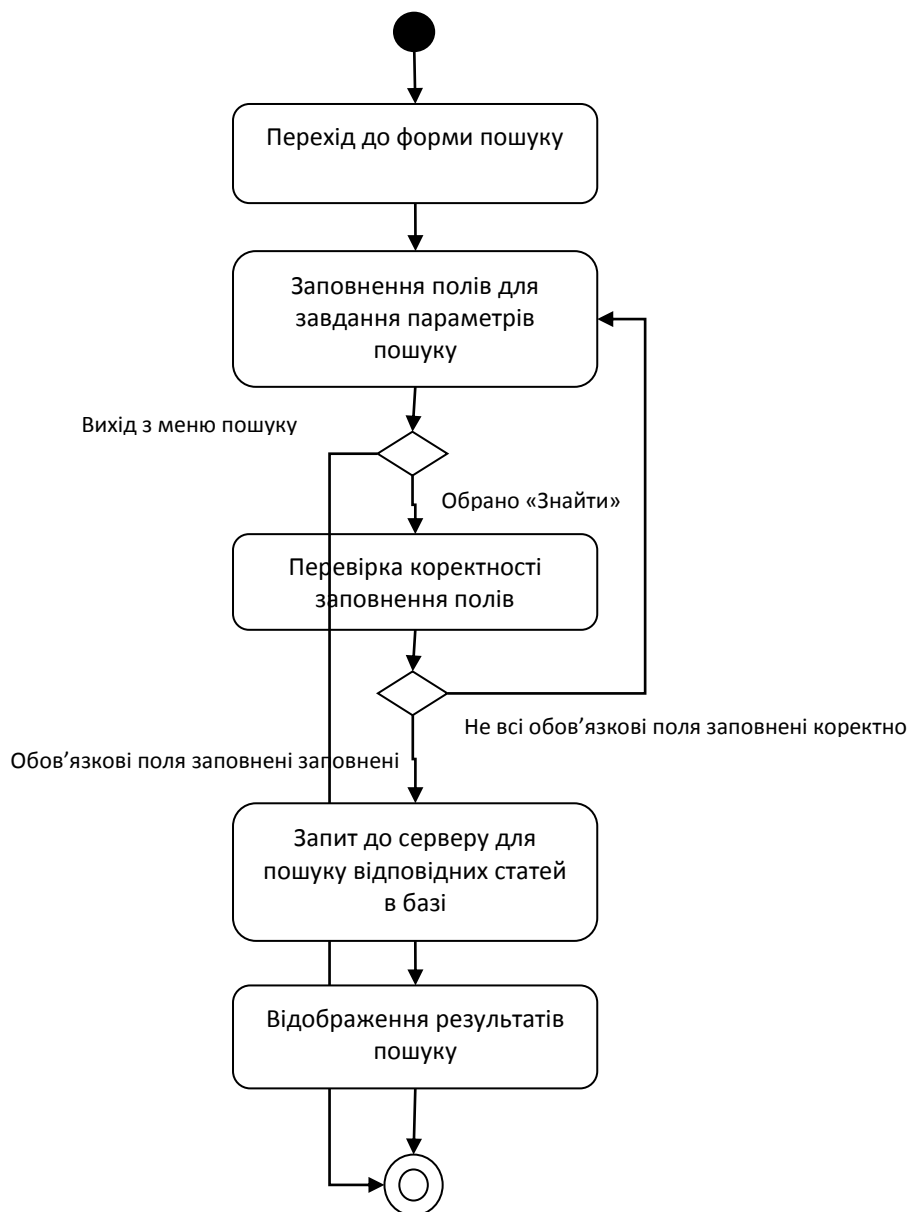


Рисунок 2.9 - Діаграма активності для пошуку статей на порталі

На рис. 2.10 зображено діаграму активності для перегляду статті.

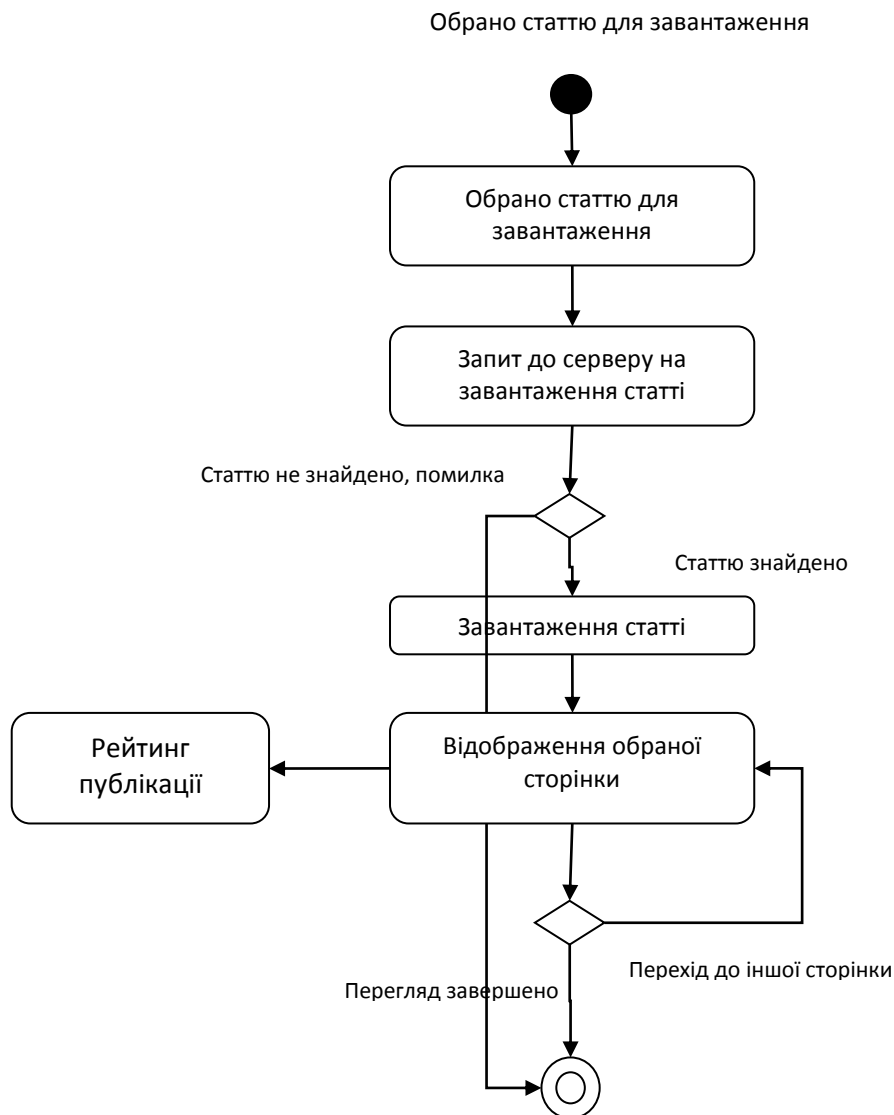


Рисунок 2.10 – Діаграма активності для перегляду статті

Процес редагування статті є одним з найбільш важливих процесів, більшість дій якого повинні виконуватись на стороні клієнта, на рис. 2.11 представлено діаграму активності для цього процесу.

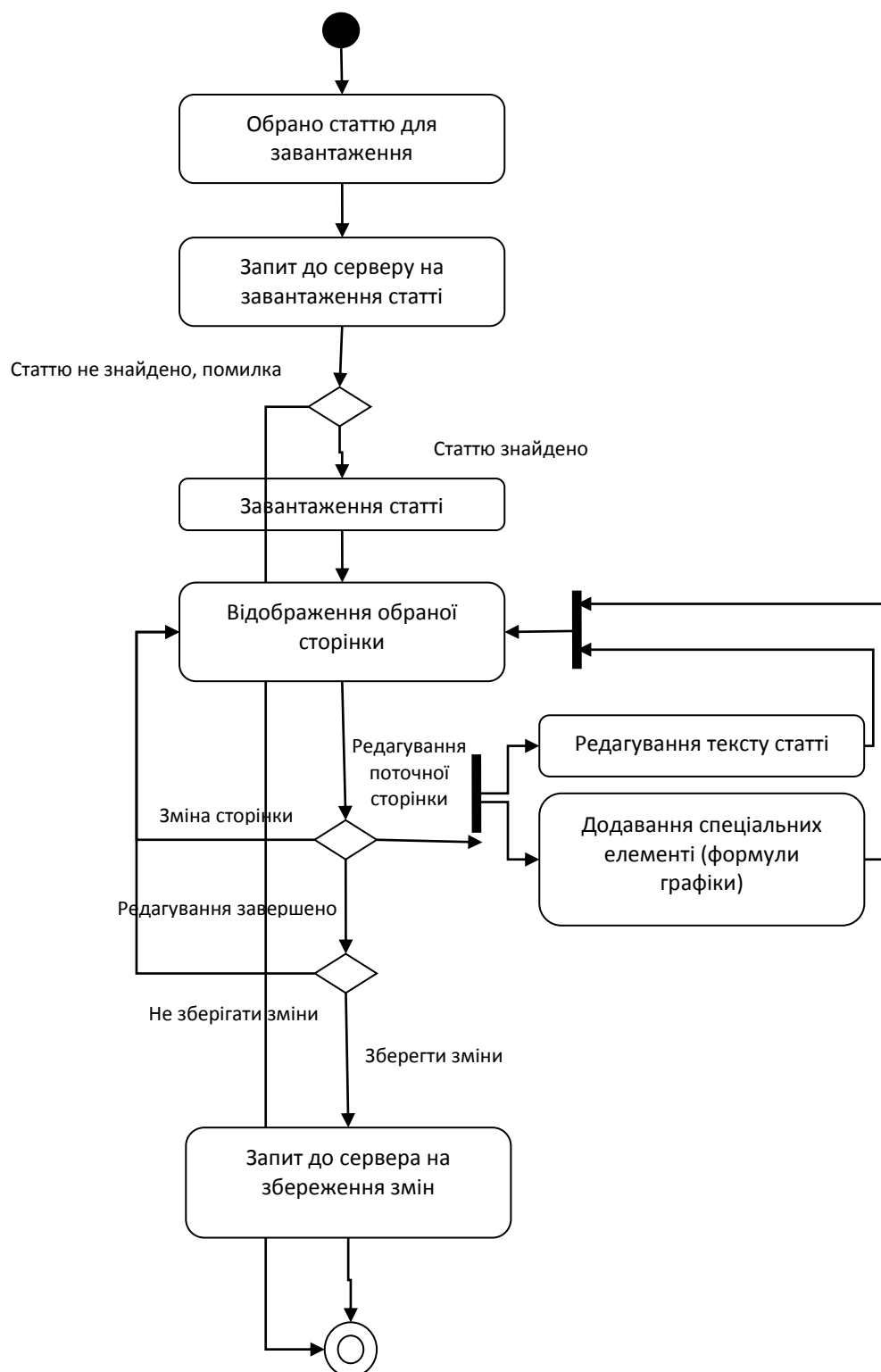


Рисунок 2.11 – Діаграма активності для редагування статті

2.7 Проектування програмного забезпечення клієнтської частини

Сторінки порталу, форми з відповідними полями (такі як форми реєстрації та авторизації, пошуку), а також статті в режимі перегляду будуть представлені відповідними моделями.

Для статей варто передбачити можливість поступового підвантаження. Оскільки стаття може складатись з багатьох сторінок, при завантаженні статті користувачем час завантаження всієї статті повністю може займати тривалий час. Тому при завантаженні статті важливо передбачити механізм завантаження таким чином, щоб користувач міг побачити перші кілька сторінок в той час, коли інші ще завантажуються. Також для дуже великих статей варто врахувати можливість завантаження клієнтською стороною лише фрагменту статті, оскільки повне завантаження надто великих статей може зменшити продуктивність роботи клієнтського ПЗ.

Для реакції на дії користувачів повинен бути написаний ряд контролерів, що реагують на відповідні дії та викликають ті чи інші методи моделей. Відповідні контролери повинні завантажуватись в залежності від ролі поточного користувача, а також відкритої сторінки порталу. Кожен контролер представлятиме собою програмний модуль. Повинні бути передбачені такі програмні модулі для перехоплення дій користувача:

- модуль, який завантажується при відкритті порталу, відповідає за дії з переміщення сайтом, цей самий контролер діятиме для гостей сайту (неавторизованих користувачів);
- модуль для авторизованих користувачів (авторів статей);
- модуль для редакторів (завантажуватиметься після авторизації користувачів з роллю редактора).

Оскільки для архітектури за шаблоном MVC, методи, що відповідають за зміни, прописуються в самій моделі, тому буде достатньо по одному контролеру (модулю з логікою реакції на відповідні дії) для кожного типу

користувачів.

Варто детальніше розглянути кожен з моделей, які повинні бути реалізовані.

Сторінки сайту, такі як «Головна», «Корисна інформація», «Список статей», «Особистий кабінет» і т. п. - різні представлення однією моделі. Для даної моделі повинні бути розроблені наступні методи:

- 1) відобразити / приховати форму пошуку;
- 2) відобразити / приховати форму авторизації;
- 3) відобразити / приховати текст (інформаційне наповнення сторінки);
- 4) запит на отримання списку статей (для відображення, якщо користувач перейшов до списку статей). Для даного методу можна реалізувати різні його варіанти, що приймають різні параметри (задані формою пошуку, або відповідним авторизованим користувачем). Відповідно, при завантаженні сторінки, може бути одразу завантажений лише той варіант методу, що відповідає поточному користувачу;

- 5) запити на завантаження / зміну / видалення / створення статті (для кожної дії повинен бути окремий метод);

- б) запити на підтвердження / відхилення публікації статті (методи, що викликатимуть дані запити, завантажуватимуться лише для редактора).

Форми для заповнення полів для авторизації та реєстрації доцільно реалізувати у вигляді окремих моделей, в яких будуть реалізовані методи, орієнтовані на коректність заповнення кожного поля.

Процес перегляду та редагування статей варто розглянути більш детально. Сама стаття повинна буде представлена моделлю, що має певні властивості та методи. Стаття має містити наступні методи:

- 1) Перехід на іншу сторінку. У властивостях статті повинен бути створений набір сторінок, до яких буде прив'язана відповідна інформація. Методом зміни сторінки повинен передати поточному поданню всю інформацію, що міститься на відповідній сторінці.

2) Зміна наповнення статті. Метод зміни наповнення статті повинен вносити відповідні зміни до моделі при змінах статті користувачем. Важливою особливістю цього методу є те, що при внесенні змін до статті, автоматично повинна бути виконана переіндексація всієї інформації з прив'язкою її до відповідних сторінок.

3) Метод додавання тексту. Додатковий метод, що повинен викликатись методом зміни наповнення статті та додавати (або видаляти) відповідні фрагменти тексту.

4) Метод додавання спеціального символу. Додатковий метод, що повинен викликатись методом зміни наповнення статті та додавати спеціальний символ, обраний користувачем. Окремо також має бути заданий набір спеціальних символів, при великій кількості символів може бути доречним передбачити їх не повне завантаження (наприклад, завантаження до клієнтської частини найбільш часто використовуваних символів, інші підвантажувати за необхідності).

5) Метод додавання формули. Додатковий метод, що повинен викликатись методом зміни наповнення статті та додавати формулу, задану користувачем. В даному методі може бути передбачений навіть автоматичний розрахунок даних за заданою формулою.

6) Метод додавання графіку. Додатковий метод, що повинен викликатись методом зміни наповнення статті та додавати графік, що автоматично будується. Часто в статтях повинні бути певні графіки або діаграми, що відображають стан досліджуваної області, порівняння результатів дослідження, тощо. Такі графіки можуть розраховуватись автоматично за заданим набором даних, подібно до того, як це робиться в Microsoft Excel. Для графіків варто передбачити заданий набір графіків з можливістю часткового підвантаження його елементів клієнтським ПЗ для зменшення його ресурсоемкості.

2.8 Рейтинг статей на сайті

Кожен науковець хоче знати актуальність, ефективність, інформативність своєї статі в науковому суспільстві. Для цього на сайті була створена мною система рейтингу. Розроблена емпірична формула. Потрібно дізнатись рейтинг автора. Це річ дуже суб'єктивна і важливо точно определити рейтинг. Наприклад, знаменитий автор статі може отримати поганій рейтинг за свою публікацію, через те, що публікація занадто інтерактивна і її не всі можуть зрозуміти, або через те, що вона дуже погана і не викликає ніякої уваги. Водночас високий рейтинг може отримати новий автор, навіть студент, якщо в нього цікава публікація. Наприклад, Білл Гейтс, засновник Майкрософт, не закінчив університет, але його публікації дуже добре оцінювали та читали інші науковці. Враховуючи все, що ми сказала, у якості рейтинга авторів були вибрані сто авторів за кількістю популярності. Усі вони упорядковуються за спаданням від 100 до 1.

Для того, щоб розрахувати рейтинг треба розуміти, заходили автори з великим рейтингом на сайт. Система у стані розробки, тому можуть бути легкі похибки. На сайт вводиться коефіцієнт появи автора. Оскільки існує ця рейтингова система, значить автори вже були на моєму порталі.

Формула також повинна враховувати людей, які не знаходяться на вершині рейтингу, але дивились потрібну публікацію. Коефіцієнт має бути приблизно 1.

Написана публікація може попасти в період року, коли науковці беруть відпустки, канікули. Наприклад, влітку рівень наукових робіт падає. Тому був створений параметр сезонності. Весь період року розділен на інтервал в 3 місяці. Період більше ніж будь-яка відпустка, тому він нам підходить, тому що людина, яка була на сайті, повинна на нього зайти за 3 місяці. У коефіцієнті ми беремо логарифм для того, щоб ми не отримували дуже великі, або дуже маленькі значення.

Необхідно також враховувати, які відгуки залишили користувачі про публікацію, яку вони прочитали. Для цього вводиться ще один параметр відгуків.

У результаті ми отримали формулу рейтинга публікації

$$Rating = \left(\sum_{i=1}^{100} R[i] * P[i] * Otz + \sum \frac{Ko}{O} \right) * \text{Log} \frac{1}{Sez}$$

де Sez – коефіцієнт сезонної активності авторів;

R[i] – рейтинг авторів (впорядковані спаданням затребуваності автори - 100;

P[i] – відвідуваність сайту рейтинговим автором за обраний період (3 місяці);

O – Загальне число користувачів, які відвідали сайт;

Ko – число користувачів відкривших вашу публікацію;

Otz – кількість позитивних відгуків про публікацію.

Сайт у стані розробки, тому формула може не повністю правильно робити, і це досить суб'єктивно, але результати показували досить точні показники.

3 ОХОРОНА ПРАЦІ

3.1 Аналіз потенційно небезпечних і шкідливих виробничих факторів , що впливають на персонал

Персональні ЕОМ типу IBM PC AT має наступні характеристики:

- споживана потужність 350 Вт;
- робоча напруга 220 В;
- напруга джерел живлення +12 В, -12 В, 5 В;
- робоча частота 50 Гц.

Виходячи з приведених характеристик, очевидно, що для користувача існує небезпека поразки електричним струмом у разі недбалого поводження з комп'ютером і порушення правил експлуатації (невиконання огляду відкритих частин ПЕВМ, що знаходяться під напругою або знятих для ремонту вузлів і т. д.).

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;

У відповідності з [12] до легкої фізичної роботи відносяться всі види діяльності, вироблювані сидячи і не вимагаючи фізичної напруги. Робота користувача розробленого пакету програм відноситься до категорії Ia.

Згідно з [18] приміщення для ПЕОМ по ступеню небезпеки поразки людини електричним струмом відноситься до приміщень без підвищеної небезпеки (немає струмопровідної половини, вогкості, підвищеної температури, можливості одночасного дотику до корпусів устаткування з "землею" і до струмонесучих частин).

У відповідності з [13] при обслуговуванні ПЕВМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищена або знижена рухливість повітря;
- підвищена або знижена вогкість повітря;
- відсутність або недолік природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

Щодо до впливу на довкілля, то програмний засіб, який було розроблено під час дипломного проекту на довкілля ніяк не впливає.

Діяльність за темою магістерської роботи в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду

забруднюючих речовин в повітря робочої зони.

В процесі діяльності виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- батарейки та акумулятори (малі) - III клас небезпеки;
- макулатура - IV клас небезпеки;
- матеріали пакувальні, що не вміщують целюлозу - IV клас небезпеки;
- матеріали пакувальні, що вміщують п/ет, п/пр - IV клас небезпеки;
- змінні носії інформації - IV клас небезпеки.

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог Державних санітарних правил і норм ДСанПіН 2.2.7.029.

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця тимчасового зберігання відходів відповідно до затвердженої схеми їх розміщення. Зазначені для зберігання відходів місця чи об'єкти повинні використовуватися лише для заявлених відходів.

Не допускається зберігання відходів у невстановлених схемою місцях, а також перевищення норм тимчасового зберігання відходів.

Способи тимчасового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- Відходи III класу небезпеки зберігаються в тарі, яка забезпечує локалізацію зберігання, дозволяє виконувати вантажно-розвантажувальні і транспортні роботи і виключає поширення в ОС шкідливих речовин;
- Відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автонавантажувачем перевантажують у самоскид і доставляють на місце утилізації або захоронення;
- В разі тимчасового зберігання відходів у стаціонарних складах або промислових приміщеннях повинні бути забезпечені санітарно-гігієнічними етичними вимогами до повітря робочої зони згідно з ГОСТ 12.1.005.

Не допускається змішування відходів різних видів і класів небезпеки з будівельними і побутовими відходами, відходами дерев'яної, металевої, синтетичної тари, відходами текстильних матеріалів (старий спецодяг, ганчірки) та інші.

Проведення заготовки, здачі, переробки та реалізації металобрухту встановлені окремо Законом України «Про металобрухт».

Всі відходи, що утворюються в процесі діяльності/роботи, підлягають обліку.

Вимоги безпеки при поводженні з відходами:

Під час роботи з відходами (прибирання виробничих приміщень, збір і сортування, навантаження, транспортування, розвантаження та ін.) працівники та обслуговуючий персонал підприємства повинні бути забезпечені засобами індивідуального захисту та дотримуватися вимог інструкцій з охорони праці, що діють на підприємстві.

Наведено перелік деяких відходів, які передаються на утилізацію організаціям, які мають ліцензію на поводження з відходами як вторинної сировини:

- лом і кускові відходи міді, бронзи, латуні, алюмінію, свинцю;
- брухт чорних металів;
- макулатура;
- склобій;
- матеріали текстильні вторинні;
- відходи деревини кускові;
- відпрацьовані фільтрувальні засоби індивідуального захисту;
- відпрацьовані вогнегасники;
- матеріали пакувальні вторинні.

Відвантаження таких відходів здійснюється відповідно до договору (контракту).

Побутові та будівельні відходи вивозяться на полігон твердих побутових відходів міста, також відповідно до договору з комунальним

дорожньо-експлуатаційним управлінням.

Особи, винні в порушенні встановленого порядку поводження з відходами (порушення правил обліку відходів, самовільне складування і видалення відходів, передача відходів в інші підприємства/організації з порушенням встановлених правил), згідно законодавства несуть дисциплінарну, адміністративну або кримінальну відповідальність.

3.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка усугубляє тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм надає на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Ступінь ураження людини електричним струмом залежить від наступних факторів:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Даним проектом передбачаються наступні технічні способи і засоби, застережливі поразки людини електричним струмом:

- заземлення електроустановок;

- занулення;
- захисне відключення;
- електричне розділення сітей;
- використання малої напруги;
- ізоляція струмоведучих частин;
- огорожа електроустановок.

Проведемо розрахунок заземлюючого пристрою.

Початкові дані для розрахунку заземлюючого пристрою:

- напруга установки, що заземляється, - 220В;
- режим нейтралу мережі - з ізольованою нейтралюю;
- питомий опір ґрунту – 100 Ом·м(суглинок);
- гранично допустимий опір заземлюючого пристрою - 4 Ом;
- характеристика кліматичної зони (III):
 - а) середня багаторічна низька температура, оС - від -14 до -10;
 - б) тривалість замерзання вод, дні - 150;
 - в) коефіцієнт сезонності для вертикального електроду завдовжки 3м -1,5.

Визначимо розрахунковий опір ґрунту (Ом·м) по формулі (3.1).

$$\rho_{расч} = \psi \cdot \rho = 1,5 \cdot 100 = 150 \text{ Ом} \cdot \text{м} \quad (3.1)$$

де ρ - питомий опір ґрунту;

ψ – кліматичний коефіцієнт, що враховує стан ґрунту під час вимірювань (табл.

Розрахуємо опір розтіканню одиночного трубчастого заземлювача по формулі (4.2).

$$R_{з,1} = \left(\frac{\rho_{расч}}{2 \cdot \pi \cdot l} \right) \cdot \ln \left(4 \cdot \frac{l}{d} \right) \quad (3.2)$$

де l – довжина заземлювача ($l=5\text{м}$);

d – діаметр труби і стрижня ($d=0,05\text{м}$);

$$R_{s,1} = \left(\frac{\rho_{расч}}{2 \cdot \pi \cdot l} \right) \cdot \ln\left(4 \cdot \frac{l}{d}\right) = \left(\frac{150}{2 \cdot 3,14 \cdot 5} \right) \cdot \ln\left(4 \cdot \frac{5}{0,05}\right) = 28,6 \text{ Ом}$$

Розрахуємо кількість паралельно сполучених одиночних заземлювачей по формулі (3.3).

$$n = \frac{R_{s,1}}{R_{доп} \cdot \eta} = \frac{28,6}{4 \cdot 0,47} = 15,2 \quad (3.3)$$

де $R_{доп}=4$. – самий допустимий опір заземлюючого пристрою;

η - коефіцієнт використання ґрунтового заземлення (для шістки заземлювачей $\eta=0,47$);

Округлятимемо отримане значення у більшу сторону $n=[15,2]=16$.

Розрахуємо довжину горизонтальної сполучної смуги по формулі (3.4).

$$L = a \cdot (n - 1) = 3 \cdot (16 - 1) = 45\text{м} \quad (3.4)$$

де a – відстань між вертикальними заземлювачами ($a=3\text{м}$);

n – кількість вертикальних заземлювачей ($n=16$).

Розрахуємо опір сполучної смуги по формулі (5.5).

$$R_n = \frac{\rho_{расч}}{2 \cdot \pi \cdot l} \cdot \ln\left(\frac{L^2}{d \cdot h}\right) \quad (3.5)$$

де d – еквівалентний діаметр смуги шириною $l=5$ ($d=0,05\text{м}$);

h – глибина заставляння смуги ($h=0,8\text{м}$).

$$R_n = \frac{\rho_{расч}}{2 \cdot \pi \cdot l} \cdot \ln\left(\frac{L^2}{d \cdot h}\right) = \frac{150}{2 \cdot 3,14 \cdot 5} \cdot \ln\left(\frac{45^2}{0,05 \cdot 0,8}\right) = 51,7 \text{ Ом}$$

Розрахуємо результуючий опір заземлюючого електроду з урахуванням сполучної смуги по формулі (4.6).

$$R_{zp} = \frac{R_{з.1} \cdot R_n}{R_{з.1} \cdot \eta_n + R_n \cdot n \cdot \eta_з} \leq R_{дон} \quad (3.6)$$

де η_n – коефіцієнт використання сполучної смуги (для 6-ї заземлювачей $\eta_n=0,27$).

$$R_{zp} = \frac{R_{з.1} \cdot R_n}{R_{з.1} \cdot \eta_n + R_n \cdot n \cdot \eta_з} = \frac{26,6 \cdot 51,7}{26,6 \cdot 0,27 + 51,7 \cdot 16 \cdot 0,47} = 3,47 \text{ Ом}$$

$3,47 < 4 \Rightarrow$ умова забезпечення електробезпеки персоналу виконується.

Таким чином, остаточна кількість заземлювачей 15 шт.

3.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Підвищення працездатності людини і збереження його здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень – це поєднання температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і пітovidільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

В приміщенні для виконання робіт операторського типу, пов'язаних з

нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (див. табл. 3.1).

Таблиця 3.1 - Оптимальні параметри мікроклімату в робочій зоні виробничого приміщення для категорії робіт 1

Період року	Температура, оС	Відносна вологість %	Швидкість руху повітря, м/с
Холодний	22.24	40.60	0,1
Теплий	23.25	40.60	0,1

Оскільки в приміщенні немає джерел виділення шкідливих речовин, можна використовувати природну вентиляцію. Площа приміщення складає 32 м². Для забезпечення прийнятних параметрів мікроклімату в приміщенні з такою площею можна використовувати 1 кондиціонер типу БК-2000.

Спектр випромінювання монітора комп'ютера включає рентгенівську, ультрафіолетову, інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння нехтує мала, оскільки цей вид випромінювання поглинається речовиною екрану.

Для зниження дії електромагнітного випромінювання пропонується захист часом і відстанню. Захист часом передбачає обмеження часу перебування людини в зоні дії полів. Тривалість роботи на ПЕОМ повинна складати не більше 3.5–4.5 години.

Також необхідно забезпечити раціональне освітлення в робочому приміщенні. В проекті, що розробляється, передбачається використовувати суміщене освітлення. В світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи володіють високою

світловою віддачею до 75 Лам/Вт і більш, тривалим терміном служби до 10000 годин, спектральним складом випромінюваного світла, близьким до сонячного.

Зорова робота оператора ПЕВМ відповідно до [16] відноситься до розряду Va з світловим потоком $\Phi_{л}=3120$ кожна. Нормована освітленість на робочому місці (E_n) при загальному освітленні складає 200 лк.

Проведемо розрахунок кількості світильників в робочому приміщенні завдовжки $a=6$ м, шириною $b=3$ м, заввишки $c=4$ м. Формула розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку (3.7):

$$\Phi_{л} = \frac{E_n \cdot S \cdot Z \cdot K}{N \cdot U \cdot M} \quad (3.7)$$

де $\Phi_{л}$ – світловий потік, Лм;

E_n – нормована освітленість;

S – площа підлоги, кв.м;

$Z=1.1-1.3$ - поправочний коефіцієнт світильника (для стандартних світильників);

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації;

N – число світильників;

$U=0.55-0.6$ – коефіцієнт використання, залежний від типу світильника, по висоті;

M – число ламп в світильнику.

З формули (3.7) виразимо N і визначимо кількість світильників для даного приміщення:

$$N = \frac{200 \cdot 18 \cdot 1,2 \cdot 1,5}{3120 \cdot 0,6 \cdot 2} = 1,7$$

Виходячи з цього, рекомендується використовувати 2 світильники. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. (3.1).

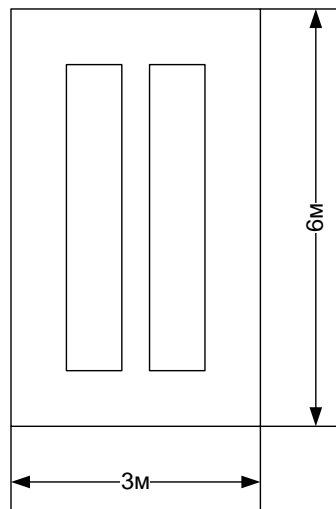


Рисунок 3.1 – Схема розташування світильників

Зниження шуму можна добитися раціонально розпланувавши приміщення, установкою устаткування на спеціальні амортизуючі прокладки. Згідно вимогам [15] рівні звуку не повинні перевищувати 50 дБ.

Для зниження стомлюваності обслуговуючого персоналу в приміщеннях, де розташовані обчислювальні засоби передбачаються використовувати спокійні колірні поєднання і покриття, що не дають відблисків. Від електромагнітного випромінювання, витікаючого від ПЕОМ, використовуються захисні екрани.

Для забезпечення чистоти повітря і відповідних мікрокліматичних умов пропонується застосувати приточування-витяжну вентиляцію. Для зменшення дії шкідливих речовин і загазованості для роботи з розплавленими матеріалами робоче місце забезпечується примусовою витяжною вентиляцією. Цей метод забезпечує притоку потрібної кількості свіжого повітря ($30 \text{ м}^3 / \text{ч}$ на одного працюючого).

Кількість повітря, яка необхідна подавати в приміщення для забезпечення необхідних параметрів повітряного середовища, визначається на підставі кількості тепла, вологи і шкідливих речовин, що поступають в приміщення, а також враховуючи видалення повітря місцевими

відсмоктуваннями від устаткування, загальнообмінною вентиляцією.

3.4 Рекомендації по пожежній профілактиці

Пожежі представляють небезпеку для життя людини і зв'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що спричиняє за собою порушення ходу технологічного процесу.

Горючими матеріалами в приміщенні, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми. Горюча речовина. Температура samozapalennya 420 оС, енергія запалення 2мДж;
- полівінілхлорид - ізоляційний матеріал. Горюча речовина. Температура samozaymannya 480 єС, енергія запалення 50мДж;
- склостоліт ДЦ - матеріал друкарської платні. Складногорючий матеріал;
- пластикат кабельний No.489 - матеріал ізоляції кабелю. Складногорючий матеріал. Температура samozaymannya 1500 єС;
- плита деревостружкова - будівельний і обробний матеріал, матеріал з якого виготовлені меблі. Складноzapalennyy матеріал. Показник горючості 1.8;
- папір – довідкова і робоча документація, література. Горючий матеріал. Показник горючості більше 2.1.

Відповідно до [17] приміщення відноситься до категорії В (пожежовибухонебезпечної).

Джерелами запалення можуть бути:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору;

- розряди статичної електрики.

Для того, щоб зупинити реакцію горіння, порушують умови її виникнення і підтримки. Звичайно для гасіння використовуються порушення двох основних умов сталого стану – пониження температури і режим руху газів. Пониження температури може бути досягнутий шляхом введення речовин, які поглинають багато тепла в результаті випаровування і дисоціації (наприклад, вода, порошки).

При повному тому, що згоряє органічних сполук утворюються С, SO, H₂O, CO₂, N₂, а при тому, що згоряє неорганічних з'єднань – оксиди. Залежно від температури плавлення і тривалості реакції можуть знаходитися або у вигляді розплавів (Al₂O₃, TiO₂), або підійматися в повітря у вигляді диму (P₂O₅, Na₂CO₃, MgO).

Склад продуктів неповного згоряє горючих речовин складений і різноманітний. Це можуть бути горючі речовини:

- H₂, C, CH₄;
- атомарний водень і кисень;
- різні радикали – OH, CH₃.

Продуктами неповного згоряє можуть бути також оксиди азоту, спирти, альдегіди, кетони і високотоксичні з'єднання, наприклад, синильна кислота.

Для захисту персоналу від дій небезпечних і шкідливих чинників пожежі проектом передбачено застосування промислового фільтруючого протигаза з коробкою марки В (жовтий).

До системи запобігання пожежі відносяться: запобігання утворення горючого середовища і освіти в горючому середовищі джерел запалення, забезпечення пожежебезпеки устаткування.

Щоб запобігти пожежі в обчислювальних центрах, проектом пропонується виконання наступних вимог:

– електроживлення ЕОМ має автоматичне блокування відключення електроенергії на випадок перегріву системи, що може бути результатом зупинки системи охолодження і кондиціонування;

– система вентиляції обчислювальних центрів обладнується блокуючими пристроями, що забезпечують її відключення на випадок пожежі. Система обладнується вогнеперегороджуючими клапанами;

– застосування устаткування, що задовольняє вимогам електростатичної іскробезпеки [13];

– після закінчення роботи, перед закриттям приміщення, всі електроустановки і персональні комп'ютери відключаються від сіті електроживлення;

– в приміщеннях обчислювальних центрів забороняється:

- 1) влаштовувати електророзетки на основах, що згорають;
- 2) використовувати синтетичні доріжки і килими;
- 3) користуватися побутовими електронагрівальними приладами;
- 4) захарашувати евакуаційні виходи і проходи;
- 5) влаштовувати на вікнах глухі ґрати;
- 6) залишати без нагляду включену в електричну мережу апаратуру, що використовується для вимірювань і нагляду.

Для протипожежного захисту проектом пропонується обладнати приміщення площею 18 м², яке відноситься до категорії В, автоматичною протипожежною сигналізацією із застосуванням датчиків сповіщення РІД-1 (оповіщувач димовий іонізаційний) в кількості 1 штуки і застосовується в первинних засобах пожежегасінні. Площа контролювана оповіщувачем 150 м².

Крім того, необхідно проводити навчання робочого персоналу правилам пожежної безпеки.

Розрахуємо вірогідність виникнення пожежі у виробничому приміщенні у разі запалювання транзистора:

$$Q = l \cdot T \cdot P_{\text{кз/отк}} \cdot Q_{\text{воспл}} \cdot P_{\text{защ}} \quad (3.8)$$

де l – інтенсивність відмов пожежеопасних ЕРІ;

T – час роботи пожежеопасного ЕРІ за оцінюваний інтервал часу;

$P_{\text{кз/отк}}$ - умовна вірогідність виходу ЕРІ в стан короткого замикання при його

$Q_{\text{воспл}}$ - вірогідність запалювання ЕРІ, що знаходиться в стані короткого зами

$P_{\text{защ}}$ – вірогідність відмови захисту пожежеопасного ЕРІ. Якщо захист відсутн

Вірогідність виникнення пожежі у разі запалювання транзистора:

$$Q = 1 \cdot 10^{-6} \cdot 1 \cdot 10^{-4} \cdot 0.1 \cdot 1 \cdot 10^{-4} = 1 \cdot 10^{-15}$$

Розрахована вірогідність виникнення пожежі значно менше допустимої, яка складає $1 \cdot 10^{-6}$.

В даному розділі були проаналізовані небезпечні і шкідливі виробничі чинники, що роблять вплив на персонал, розроблені заходи щодо техніки безпеки, заходу, забезпечуючи виробничу санітарію і гігієну праці, а також заходи щодо пожежної профілактики.

ВИСНОВКИ

В процесі виконання випускної кваліфікаційної роботи було проведено аналіз напрямків розвитку розробки сучасних Web-додатків, зокрема було приділено особливу увагу сучасним тенденціям до створення Web-ресурсів з інтерактивною поведінкою, наповнення сторінок котрих здатне динамічно змінюватись в залежності від дій користувача. Було розглянуто різні підходи до реалізації програмного забезпечення таких ресурсів, як клієнтської, так і серверної частини.

В ході дослідження особлива увага була приділена технологіям Single Page Application (SPA), які на сьогодні отримали широке розповсюдження. Важливими перевагами даного підходу в розробці Web-ресурсів є підвищення швидкості їх завантаження та реакції на дії користувача, зменшення споживання трафіку, а також підтримка різних платформ.

Також в ході роботи був проведений аналіз різних патернів проектування, які використовуються при розробці Web-ресурсів з застосуванням технологій SPA.

Послідовно була проведена робота з проектування Web-порталу для підготовки та публікації наукових текстів із застосуванням технологій SPA. На відповідних етапах проектування були побудовані діаграми активності для ресурсу, бізнес-логіка роботи порталу, визначені типи користувачів та їх права, на основі чого був проєктований інтерфейс користувача та визначені основні моделі та їм методи, а також контролери (модулі логіки ресурсу), що викликатимуть ті чи інші методи моделей.

Також в ході роботи було проведено аналіз робочого місця розробника Web-порталу з точки зору охорони праці. В ході аналізу робочого місця були виявлені деякі невідповідності державним нормам, що можуть негативно впливати на здоров'я розробника. Також було складено інструкцію з охорони праці для робочого місця розробника (програміста).

Створений в результаті роботи Web-ресурс володіє прекрасними можливостями, що дозволяють в режимі онлайн створювати, зберігати, редагувати та публікувати наукові тексти. В сучасному світі, де комп'ютерні технології посідають значне місце в житті людини, подібні ресурси є досить актуальними, адже користувачі ресурсу у будь-якому місці та в будь-який час зможуть отримати доступ до потрібної їм інформації (перегляд опублікованих статей), а також оперативно внести необхідні зміни до своїх власних статей, використовуючи при цьому навіть мобільні пристрої.

Розроблений портал можна вдосконалювати новими функціями, наприклад, додати пошук подібної інформації не лише в межах самого порталу, але й в Інтернеті, додати встроєну перевірку на анти плагіат безпосередньо в процесі редагування тексту статті, тощо.

Під час виконання розділу «Охорона праці» було проаналізовано робоче місце програміста на відповідність до встановлених норм. У ході аналізу умов праці було виявлено підвищений рівень шуму. У ході розрахунків було встановлено, що рівень шуму не перевищує допустимі значення. Додаткові заходи по покращенню умов праці не потрібні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Тиге Дж. К. DHTML и CSS [Текст]: пер. с англ. / Дж.К Тиге - М.: ДМК Пресс, 2003. – 506с.
- 2) Гудман.Д JavaScript и DHTML. Сборник рецептов для профессионалов [Текст]: пер. с англ./ Д.Гудман – Питер, 2004. – 523с.
- 3) Глушаков С.В. Программирование Web-страниц [Текст]/ С.В Глушаков. - Харьков: Фолио, 2005. – 390с.
- 4) Полянский А.А. Программирование на CGI [Текст]/ А.А Полянский – М.: Мойор, 2003. – 176с.
- 5) Джонс А. Active Server Pages 3: полное руководство [Текст]: пер. с англ./ А. Джонс – М.: Энтроп 2001. – 704 с.
- 6) Шаповалова С.І. Оптимізація моделі представлення Web-системи на основі SPA-архітектури[Текст]/ С.І. Шаповалова - Вісник НТУ «ХП», 2012, №68. – С. 71–75.
- 7) Особенности архитектуры Single Page Application [Электронный ресурс]. URL: <https://events.yandex.ru/lib/talks/2557/> - 12.03.2018 – Загл. з екрану.
- 8) Mikowski.M.S Single Page Web Applications[Текст]/ M.S.Mikowski, J.C. Powell – Manning, 2018. – 433р.
- 9) Single-page vs. Multi-page. Особенности автоматизации тестирования. [Электронный ресурс]. URL: <http://seleniumcamp.com/archive/selenium-camp-2013/materials/single-page/> - 14.03.2018 – Загл. з екрану.
- 10) Model-View-Controller [Электронный ресурс]. URL: [https://msdn.microsoft.com/ru-ru/library/ms978748\(en-us\).aspx](https://msdn.microsoft.com/ru-ru/library/ms978748(en-us).aspx) - 21.03.2018 - Загл. з екрану.
- 11) MVC vs. MVP vs. MVVM [Электронный ресурс]. URL: <https://nirajrules.wordpress.com/2009/07/18/mvc-vs-mvp-vs-mvvm/> - 26.03.2018

– Загл. з екрану.

12) ГОСТ 12.1.005–88. «ССБТ. Общие санитарно–гигиенические требования к воздуху рабочей зоны».

13) ГОСТ 12.0.003–74. «ССБТ. Опасные и вредные производственные факторы. Классификация.»

14) ГОСТ ГОСТ 12.1.030-81 «ССБТ. Электробезопасность. Защитное заземление. Зануление»

15) ГОСТ 12.1.003-83. «ССБТ. Шум. Общие требования безопасности»

16) ДБН В.2.5-28-2006. «Природне і штучне освітлення»

17) НАПБ Б.03.002-2007. «Нормы определения категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности»

18) ДСТУ Б А.3.2-13:2011 «Система стандартів безпеки праці. Будівництво. Електробезпе́чність. Загальні вимоги»

ДОДАТОК А.

Лістинг коду

```
<?php

namespace app\models;

use Yii;
use yii\behaviors\TimestampBehavior;
use yii\db\ActiveRecord;
use cebe\markdown\Markdown;

/**
 * This is the model class for table "tbl_post".
 *
 * @property integer $id
 * @property string $title
 * @property string $content
 * @property string $tags
 * @property integer $status
 * @property integer $created_at
 * @property integer $updated_at
 * @property integer $author_id
 * @property integer $category_id
 */
class Post extends ActiveRecord
{
    const STATUS_DRAFT = 1;
    const STATUS_PUBLISHED = 2;
    const STATUS_DELETED = 3;

    /**
     * @inheritdoc
     */
    public function delete()
    {
        $result = false;
        if ($this->beforeDelete()) {
            $result = $this->updateAttributes(['status' =>
self::STATUS_DELETED]);
            $this->afterDelete();
        };

        return $result;
    }

    /**
     * @inheritdoc
     */
    public static function find()
    {
        return parent::find()->where('status <>' . self::STATUS_DELETED);
    }

    /**
     * @inheritdoc
     */
    public static function tableName()

```

```

{
    return '{{%post}}';
}

/**
 * @inheritdoc
 */
public function behaviors()
{
    return [
        TimestampBehavior::className(),
    ];
}

/**
 * @inheritdoc
 */
public function rules()
{
    return [
        [
            ['title', 'content'],
            'filter',
            'filter' => function ($value) {
                return Yii::$app->formatter->asHtml($value);
            }
        ],
        [['title', 'content', 'status'], 'required'],
        [['category_id'], 'integer'],
        [['content', 'tags'], 'string'],
        [['status'], 'in', 'range' => [self::STATUS_DRAFT,
self::STATUS_PUBLISHED]],
        [['title'], 'string', 'max' => 128],
        [['tagsArray'], 'safe']
    ];
}

public function extraFields()
{
    return [
        'comments',
    ];
}

public function fields()
{
    return [
        'id',
        'title',
        'content',
        'tags',
        'status',
        'tagsArray',
        'author',
        'created_at' => function () {
            return date('d-m-y H:i', $this->created_at);
        },
        'updated_at' => function () {
            return date('d-m-y H:i', $this->updated_at);
        },
        'category_id',
        'comments',
    ];
}

```

```

        'category',

    ];
}

/**
 * @inheritdoc
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'title' => 'Title',
        'content' => 'Content',
        'tags' => 'Tags',
        'status' => 'Status',
        'created_at' => 'created',
        'updated_at' => 'updated',
        'author_id' => 'Author ID',
        'category_id' => 'Category ID',
    ];
}

public function setTagsArray($tagsArray)
{
    for ($i = 0; $i < count($tagsArray); $i++) {
        $tagsArray[$i] = trim($tagsArray[$i]['text']);
    }
    $this->tags = implode(",", $tagsArray);
}

public function getTagsArray()
{
    $tagsArray = explode(",", $this->tags);
    for ($i = 0; $i < count($tagsArray); $i++) {
        $tagsArray[$i] = ['text' => trim($tagsArray[$i])];
        if (empty($tagsArray[$i]['text']))
            {unset($tagsArray[$i]);}
    }
    return $tagsArray;
}

public function beforeSave($insert)
{
    if (!$this->author_id) {
        $this->author_id = Yii::$app->user->id;
    }
    return parent::beforeSave($insert);
}

public function getAuthor()
{
    return $this->hasOne(User::className(), ['id' => 'author_id']);
}

public function getComments()
{
    return $this->hasMany(Comment::className(), ['post_id' => 'id']);
}

public function getCategory()

```

```
{  
    return $this->hasOne(Category::className(), ['id' => 'category_id']);  
}
```

ДОДАТОК Б.
Електронні плакати

**ДИПЛОМНА РОБОТА
БАКАЛАВРА**

Інформаційна система підготовки
та публікації наукових текстів

Виконав:

Деркач А.М.

Керівник:

доц.
Барбарук В.М.

Актуальність

- * Актуальним є питання створення таких Web-ресурсів, які б поєднували у собі зручний інтерфейс користувача з усіма необхідними функціями, можливість одночасної роботи багатьох користувачів, зберігання досить великих об'ємів інформації, і в той же час забезпечували швидкість та надійність роботи із ними.

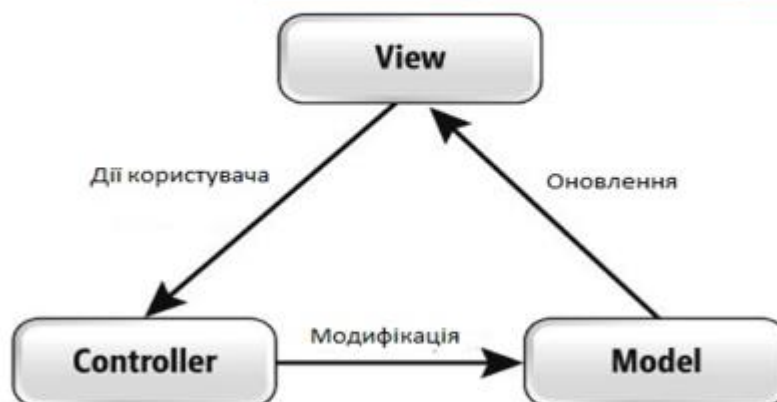
Постановка задачі

- * вивчити і проаналізувати предметну область;
- * зробити порівняльний аналіз існуючих рішень та обрати оптимальний варіант;
- * визначити основні функції Web-порталу;
- * спроектувати архітектуру Web-порталу;
- * описати інтерфейс користувача, створити макет інтерфейсу;
- * розробити алгоритми роботи логіки системи, описати запити, що використовуватимуться в системі та підсистемах.
- * Розробити формулу рейтинга публікації

Технології

1. HTML
2. CSS
3. PHP
4. MVC
5. MVVM
6. AngularJS
7. Javascript

Модель-Вид-Контроллер



Model-View-ViewModel



Архітектура Web порталу

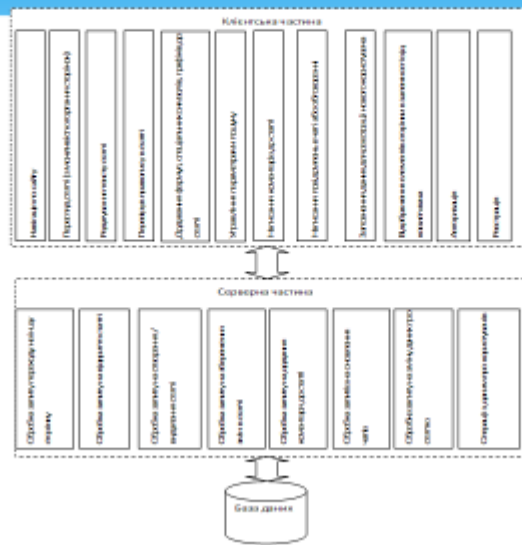
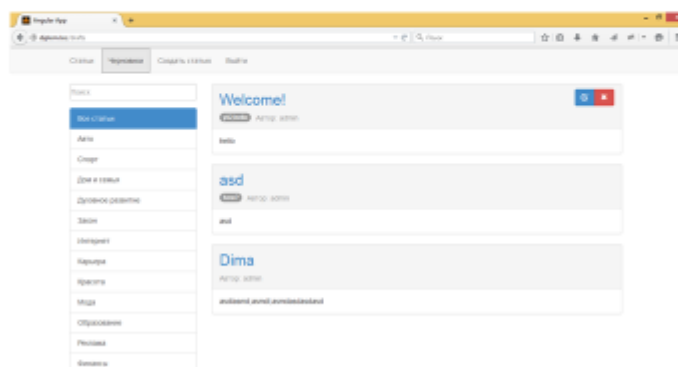


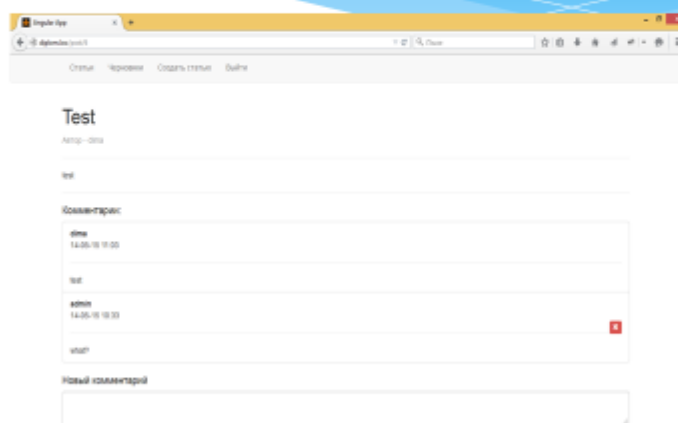
Схема інтерфейсу Web порталу



Головна сторінка



Перегляд сторінки



Формула рейтингу

$$Rating = \left(\sum_{i=1}^{100} R[i] + P[i] + O_{tz} + \sum \frac{K_o}{O} \right) \cdot \text{Log} \frac{1}{Sez}$$

Sez – коефіцієнт сезонної активності авторів;

R[i] – рейтинг авторів (впорядковані спаданням затребуваності автори -100;

P[i] – відвідуваність сайту рейтинговим автором за обраний період (3 місяці);

O – Загальне число користувачів, які відвідали сайт;

Ko – число користувачів відкривших вашу публікацію;

Otz – кількість позитивних відгуків про публікацію

Висновки

- * Створений в результаті роботи Web-ресурс володіє прекрасними можливостями, що дозволяють в режимі онлайн створювати, зберігати, редагувати та публікувати наукові тексти. В сучасному світі, де комп'ютерні технології посідають значне місце в житті людини, подібні ресурси є досить актуальними, адже користувачі ресурсу у будь-якому місці та в будь-який час зможуть отримати доступ до потрібної їм інформації (перегляд опублікованих статей), а також оперативно внести необхідні зміни до своїх власних статей, використовуючи при цьому навіть мобільні пристрої. Створений в результаті роботи Web-ресурс володіє прекрасними можливостями, що дозволяють в режимі онлайн створювати, зберігати, редагувати та публікувати наукові тексти. В сучасному світі, де комп'ютерні технології посідають значне місце в житті людини, подібні ресурси є досить актуальними, адже користувачі ресурсу у будь-якому місці та в будь-який час зможуть отримати доступ до потрібної їм інформації (перегляд опублікованих статей), а також оперативно внести необхідні зміни до своїх власних статей, використовуючи при цьому навіть мобільні пристрої