

ЗМІСТ

Вступ.....	4
1 Постановка задачі дослідження.....	5
1.1 Аналіз предметної області.....	5
1.2 Огляд існуючих програмних продуктів..	5
1.2.1 Додаток Diary-anek.....	5
1.2.2 Додаток МойДневник	7
1.2.3 Додаток DnevniX	9
1.2.4 Додаток Фрагментар	10
1.3 Постановка задачі	12
2 Специфікація вимог до веб-додатку	14
2.1 Загальна інформація щодо веб-застосунку.....	14
2.1.1 Огляд продукту.....	14
2.1.2 Мета	14
2.1.3 Межі.....	16
2.1.4 Маштаб проєту	16
2.1.5 Обмеження реалізації.....	15
2.1.6 Перспективи продукту.....	15
2.1.7 Діаграми потоків даних	15
2.2 Вимоги до веб-додатку	15
2.2.1 Функціональні вимоги	16
2.2.2 Нефункціональні вимоги	18
2.2.2.1 Надійність	18
2.2.2.2 Доступність	18
2.2.2.3 Безпека.....	18
2.2.2.4 Супровід	18
2.2.2.5 Кросплатформність	18
2.2.2.6 Продуктивність.....	18
2.3 Опис обраних технологій для розробки веб-додатку.....	19

2.3.1 HTML5.....	20
2.3.2 CSS3.....	20
2.3.3 ADO.NET.....	22
2.3.4 Microsoft Visual Studio	24
2.3.5 JavaScript	25
2.4 Опис архітектури веб-додатку	26
2.4.1 Діаграма розміщення	28
2.4.2 Діаграма компонентів	29
2.5 Модель бази даних	30
3 Опис веб-додатку	31
3.1 Тестування веб-додатку.....	31
3.1.1 Верифікація та валідація веб-додатку	31
3.2.1 Ручне тестування	32
3.2 Керівництво користувача	33
3.2.1 Загальний опис можливостей	34
3.2.2 Об'єкти	35
3.2.3 Демонстрація записів у щоденнику	34
3.2.4 Фільтрація по даті	35
Висновки	36
Список джерел інформації	37

ВСТУП

Інформаційні технології увійшли досить міцно в наше життя, вони застосовуються у всіх сферах життя людства, виконуючи особливо значущу двоїсту роль. Інформаційні технології представляють весь накопичений досвід людства в форматизованому вигляді, придатному для прикладного використання. І в ньому сконцентровані наукові знання і матеріалістичний досвід для здійснення суспільних процесів, при цьому економляться витрати праці, часу, енергії, речових засобів. І з кожним днем з непомірною силою роль ця збільшується. Це явище пояснюється властивостями, якими володіють інформаційні технології.

Об'єкт дослідження – онлайн щоденник для ведення записів. Предметом дослідження є створення веб-додатку «Особистий щоденник» на основі технологій .NET і JavaScript. Предметна область – веб-додаток щоденник.

Ведення особистого щоденника – це ефективний спосіб записувати інформацію у вигляді тексту або медіафайлів для тих, хто бажає завжди мати спосіб закріплення якоїсь інформації для подальшого її використання.

Веб-додатки істотно допомагають людині зручно вести важливі записи з доступом у будь-який час.

Метою курсової роботи є розробка веб-додатку особистого щоденника, який допоможе людині швидко та з будь якого пристрою в будь який час записувати важливу для себе інформацію. Під час її виконання планується дослідити дану предметну область і вивчити теоретичні питання розробки даного веб-додатку і на підставі вивченого матеріалу розробити веб-застосунок з використанням сучасних технологій розробки.

1 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1. Аналіз предметної області

Ведення особистого щоденника – це ефективний спосіб записувати інформацію у вигляді тексту або медіафайлів для тих, хто бажає завжди мати спосіб закріплення якоїсь інформації для подальшого її використання.

Додаток буде корисний тим людям, яким часто потрібно записувати важливу для себе інформацію.

Актуальність роботи полягає в тому, що онлайн доступ до своїх записів має велике значення на сьогоднішній день. В час, коли суспільство поглинає велику кількість інформаційного шуму, буде дуже корисно вести свій власний щоденник з дійсно важливою для себе інформацією.

1.2. Огляд існуючих програмних продуктів

Огляд існуючих веб-додатків, спрямованих на записування, редагування та відображення даних записаних людиною. Існуючі додатки орієнтовані на людей, які бажають зручно вести запис своїх даних.

Критерії оцінки кожного веб-додатку: дизайн, функціонал, простота і зрозумілість додатку, можливість відображати записи у щоденнику за певною датою, зберігання даних, можливість відображення дати та часу відповідного запису.

1.2.1. Додаток *Diary-anek*.

Додаток розроблений для користувачів комп'ютерів та мобільних пристроїв, що надає можливість його використання у зручний час [1].

Інтерфейс додатку зображений на рисунках 1.1 та 1.2.



Рисунок 1.1 – Интерфейс dodatku Diary-anek

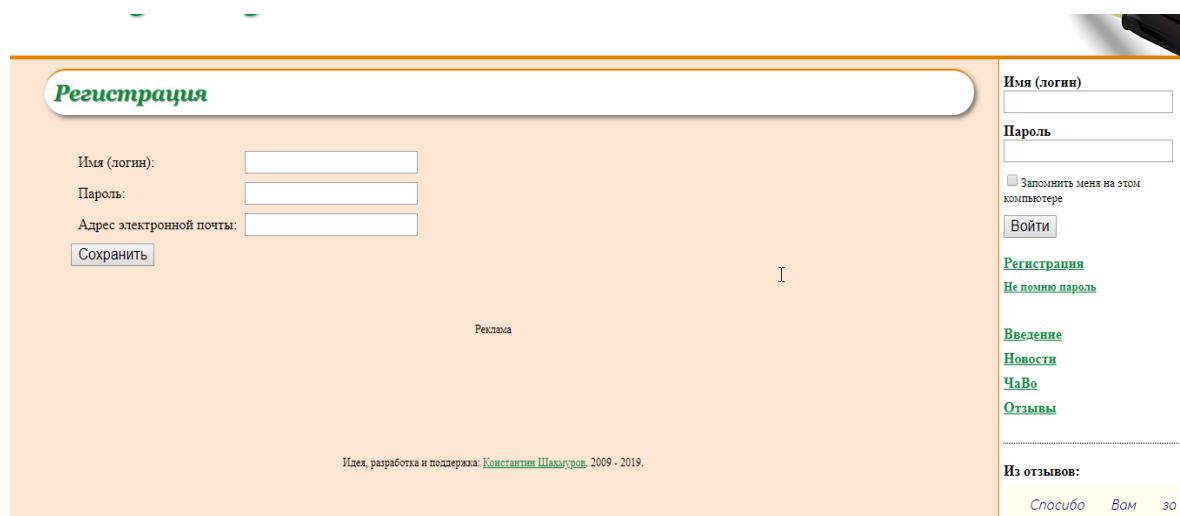


Рисунок 1.2 – Интерфейс dodatku Diary-anek

Даний щоденник не має широкого функціоналу але спрямований на швидку та зручну можливість введення онлайн записів.

Додаток має наступні функції:

- реєстрація та авторизація користувача;
- можливість залишати відгуки;
- автозбереження записів;

– доступність на багатьох платформах;

До недоліків даного сайту можна віднести не гнучкі налаштування свого профілю та не сучасний дизайн. Також на сайті немає переключення між мовами і є можливість використовувати лише російську.

1.2.2. Додаток МойДневник

Додаток розроблений для користувачів комп'ютерів та мобільних пристроїв, що надає можливість його використання у зручний час [2].

Інтерфейс додатку зображений на рисунках 1.3 та 1.4.

Можливістю додатку є можливість створення багатьох окремих щоденників в одного користувача.

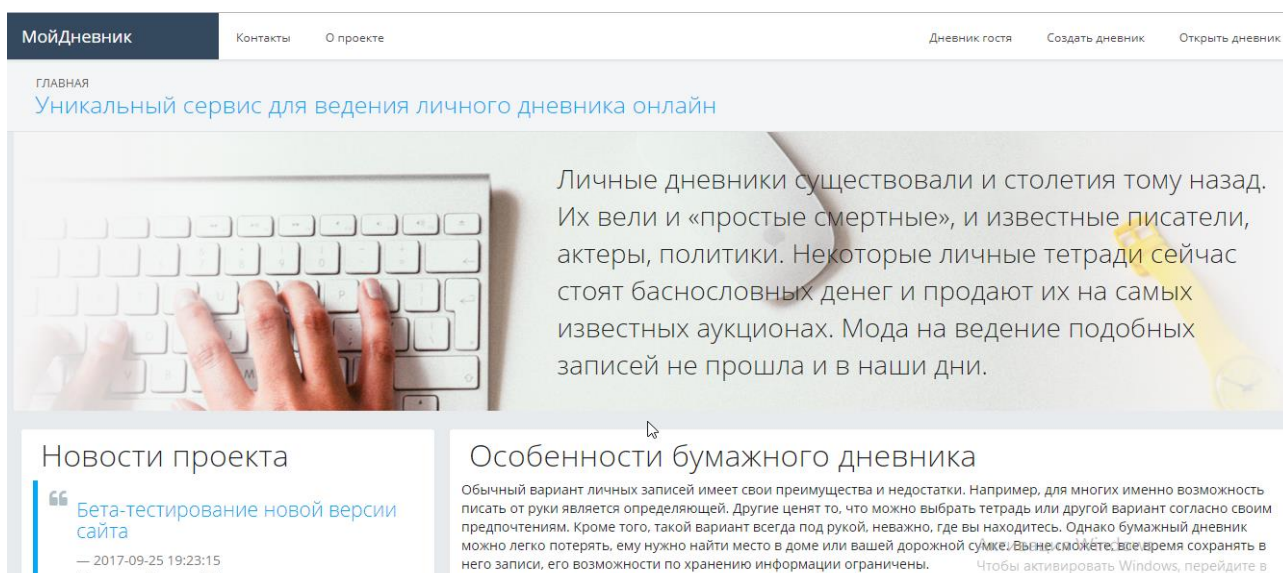


Рисунок 1.3 – Интерфейс додатку МойДневник

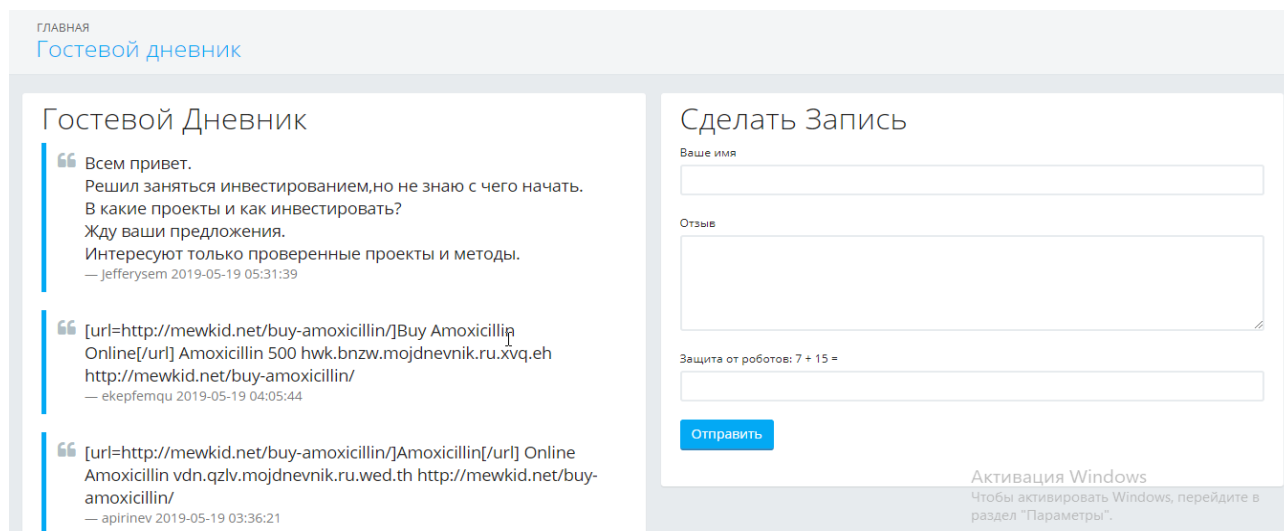


Рисунок 1.4 – Інтерфейс додатку МойДневник

Додаток має наступні функції:

- реєстрація та авторизація користувача;
- автозбереження записів;
- можливість створення багатьох окремих щоденників в одного користувача;
- можливість зайти як гість або авторизований користувач;

Одним з найважливіших недоліків цього сайту є не дуже зрозумілий для недосвідченого користувача інтерфейс та інтуїтивно не зрозумілий функціонал. Також сайт має певну кількість програмних помилок, таких як: неповне відображення дати запису та сайт не поділяє записи на сторінки, а показує їх на одній сторінці, що робить перегляд незручним.

1.2.3. Додаток DnevniX

Онлайн щоденник, що забезпечить надійність та зручність користування. Також сайт має гнучкі налаштування, що зробить його максимально зручним для будь-якого користувача. Щоб використовувати цей особистий щоденник, все що вам потрібно, це будь-який пристрій з доступом в Інтернет. Будинки в комп'ютері або на ходу з телефону- якщо у вас є раптове бажання писати, ви можете отримати доступ до даних в будь-який час і з будь-якої точки світу.

Интерфейс dodatku zobrazenij na risunkax 1.5 ta 1.6.

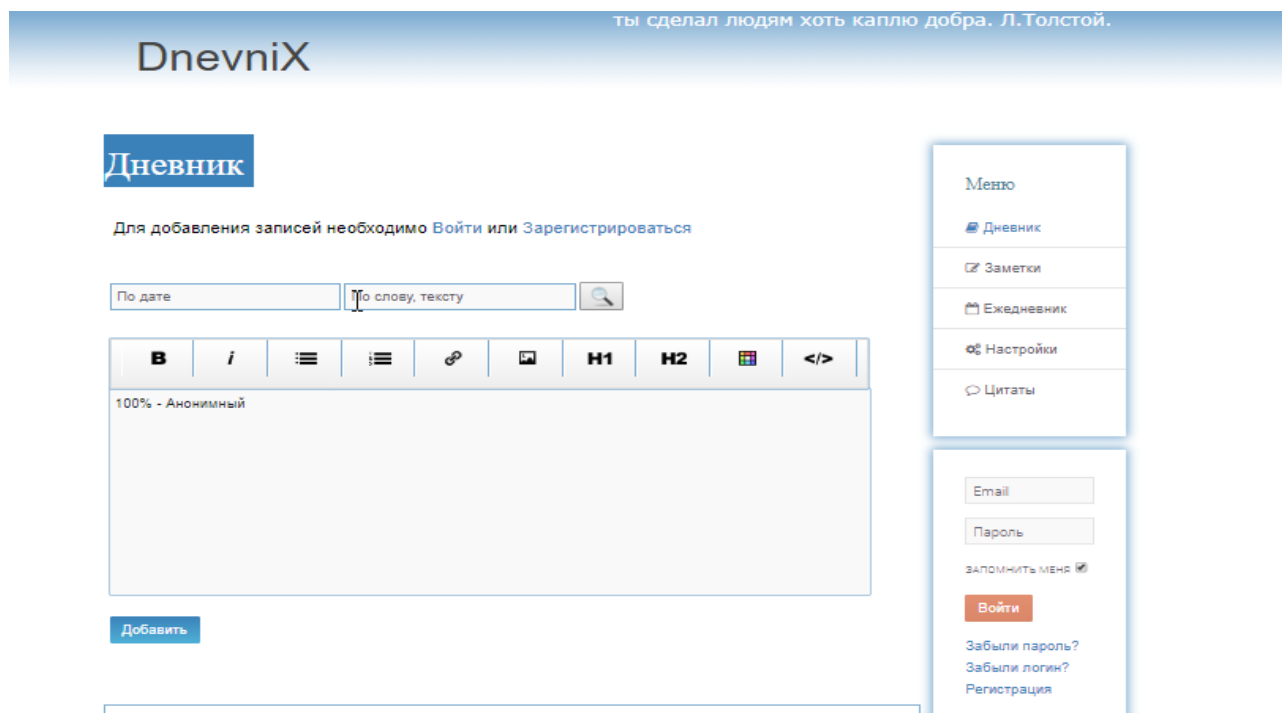


Рисунок 1.5 – Интерфейс dodatku DnevniX



Рисунок 1.6 – Интерфейс додатку DnevniX

Онлайн-щоденник DnevniX використовує стандарти безпеки для передачі і зберігання даних. Ваші дані зберігається тільки в зашифрованому вигляді на захищених серверах. Це означає, що навіть системні адміністратори DnevniX не можуть читати ваші дані. Ето також означає, що немає можливості передати дані державним органам. Онлайн-щоденник DnevniX пропонує безпечне місце для ваших самих особистих тем.

Додаток має наступні функції:

- надійність та зручність користування;
- сайт має гнучкі налаштування;
- використовує стандарти безпеки для передачі і зберігання даних;

Незважаючи на свої переваги, сайт має деякі проблеми, такі як незручне та неправильне розташування компонентів на ресурсі. Також до недоліків можна віднести платну версію, хоча через це на сайті відсутня реклама.

1.2.4. Додаток Фрагментар

Онлайн щоденник Фрагментар є розробкою українських спеціалістів. Який допоможе користувачу записувати всю важливу інформацію в свій аккаунт. [4].

Інтерфейс додатку зображений на рисунках 1.7 та 1.8.

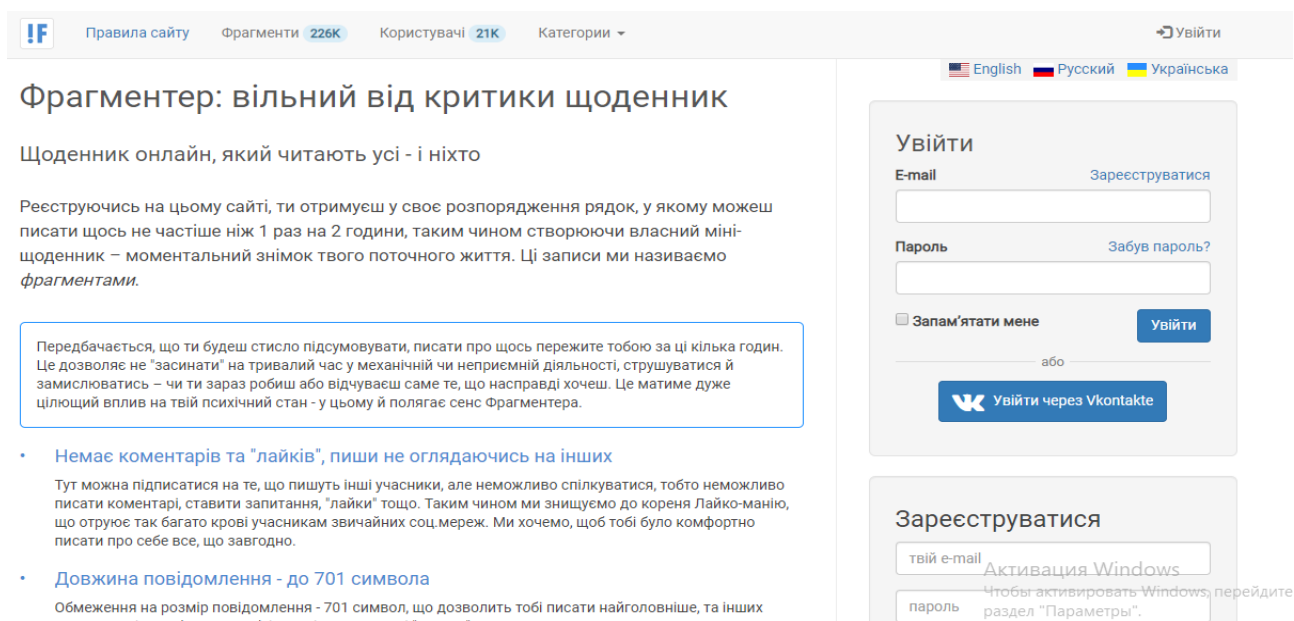


Рисунок 1.7 – Інтерфейс додатку Фрагментар

Що ми маємо

У нашому он-лайн щоденнику можна розміщувати не частіше одного разу на дві години повідомлення (не більше як 700 символів), яке потрапляє до загальної стрічки. Роблячи це, ти зможеш виринути з суєти і турбот, озирнутися на прожитий час і написати те, що захочеться за підсумками пережитого, того що сталося чи не сталося. Дивним чином таке "виринання" помітно збільшує тверезість твого стану й насиченість життя, призводить до появи нових бажань, ідей, ясноти. Фрагментар допомагає навіть у тому випадку, якщо ти не пишеш черговий фрагмент через нехтіть чи відсутність чого писати. У будь-якому випадку ти за звичкою подумки повертаєшся до пережитого проміжку життя, даєш йому хоча б поверхневу уявну оцінку, що в будь-якому випадку привносить у твоє життя більше тверезості, і внаслідок цього – бодай невелике збільшення насиченості.

Ненормативна лексика дозволена, але велике прохання не зловживати нею й не "лятисся матом", а "розмовляти матом", коли це тобі здається доречним. Мало осмисленні повідомлення з великою кількістю мату будуть просто видалені. І пам'ятай - чим більше матюків та нісенітниць у твоїх повідомленнях, тим вище ймовірність, що багато хто занесе тебе в особистий бан і надалі взагалі не бачитиме твоїх повідомлень.

Є можливість робити приватні записи, але пам'ятай - в мережі Інтернет цілковитої анонімності не буває.

У своєму "профілі" ти зможеш написати про себе, дати посилання на свої інші блоги та сайти, вибрати собі аватарку (функція аватарки буде зроблена найближчим часом).

У міру додавання нових функцій, тут будуть додаватися їх описи.

Чого ми не маємо і не матимемо

У нас немає функції видалення записів.

У нас немає можливості ставити лайки. Таким чином ми знищуємо до кореня Лайко-манію, яка отрує так багато крові учасникам звичайних соц. мереж. У нас можна спокійно робити свої щоденникові записи й не перейматися тим - скільки тобі наставлять лайків чи антилайків.

Видаляються повідомлення вочевидь сміттєві, спамові або такі, що містять особисті дані своїх чи третіх осіб.

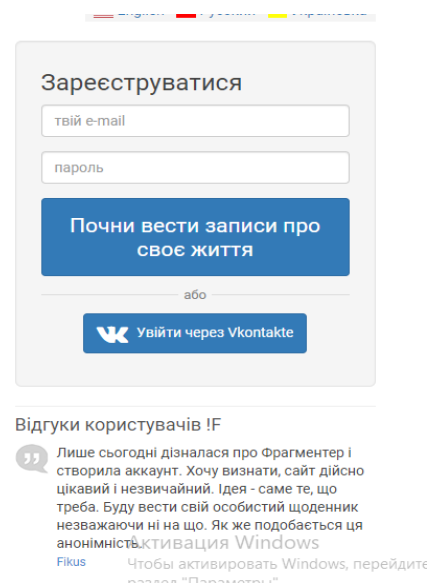


Рисунок 1.8 – Інтерфейс додатку Фрагментар

Перевагою сайту є те, що користувачу можна авторизуватися через інші соціальні мережі й не витратити час на реєстрацію.

Додаток має наступні функції:

- можливість ведення як і особистого щоденника так і публічного блогу;
- редагування та видалення записів;
- додавання медіафайлів;
- безпечність через стандарти шифрування;

Хоч на ресурсі використовується підтримка трьох мов, але переклад не завжди являється коректним та правильним. Також на сайті присутньо багато реклами, що заважає зручному перегляду та веденню своїх записів.

Таблиця 1.1 – Порівняння веб-додатків

	Diary-anek	МойДневник	DnevniX	Фрагментар
Дизайн	Сучасний, яскравий	Хороший, простий, приємний для ока	Стильний, сучасний	Задовільний
Рівень функціональності	Високий	Середній	Високий	Високий
Історія записів у щоденнику	+	+	+	+
Підтримка в реальному часі	+	–	+	+
Платна версія	–	–	+	–

З вище перерахованих додатків онлайн-щоденника жодного змішаного додатку. Всі представлені додатки задовольняють мінімальним критеріям. У порівнянні з веб-додатками конкурентів, особистий щоденник, розроблений у

рамках цієї курсової роботи відповідає усім вимогам. Веб-додаток має простий, мінімалістичний та інтуїтивно зрозумілий інтерфейс. Проста авторизація для користувача дає змогу швидко приступити до роботи з аплікацією. У порівнянні з іншими онлайн щоденниками, додаток не має реклами. Великою перевагою особистого щоденника є можливість працювати у режимі оффлайн. Навіть якщо у користувача раптово зникло інтернет з'єднання, або сервер не відповідає на запити, можна користуватися додатком. Така функція відсутня у всіх представлених сервісах.

Аналізуючи статистику відвідування сайтів виробників додатків та скачування додатків, з ведення особистого щоденника, можна прийти до висновку, що особистий онлайн щоденник в нинішній час є досить актуальним і популярним. Дивлячись на розвиток розробки онлайн щоденників, ми бачимо, що розробка додатків даного типу втратить актуальність ближнім часом не зможе.

1.3. Постановка задачі

Всього двадцять років тому в моді ще були рукописні щоденники. Тепер їх місце зайняли публічні блоги. Але як бути тим, хто хоче вести щоденник в інтернеті, але виключно для себе?

В епоху блогів і соціальних медіа, особисті щоденники і раніше мають вірних послідовників. Тут ви можете бути повністю чесним, і зберегти ваші думки назавжди. Щоб стежити за своїм минулим і думати про своє майбутнє доступність – використання в будь-який час доби і в будь-якому місці;

Онлайн щоденник має мати такий функціонал:

- можливість отримувати звіти, які аналізують процес зміни записів, внесених користувачем;
- збереження введеної інформації і неможливість її втрати.

Завдання зберігання і обробки інформації є актуальним для сфери організації роботи веб-додатку з ведення онлайн щоденника.

Метою роботи є створення веб-додатку для декількох користувачів, переглядати та редагувати свої записи. Програмний продукт повинен бути доступним для користування людей різних вікових категорій, а саме бути простим і зрозумілим.

Відповідно до поставленої мети і вимогами були сформульовані наступні завдання:

- дослідження і опис предметної області – організація роботи додатку з записами особистого щоденника;
- розробка моделі даних;
- вибір СУБД і засобів розробки веб-додатку;
- розробка веб-застосунку;
- розробка простого і зручного інтерфейсу;
- створення можливості редагування, додавання, перегляду і видалення записів;

Для досягнення поставлених завдань виділимо наступні задачі:

- вивчення принципів побудови веб-додатків;
- дослідження структури додатку;
- аналіз програмних засобів побудови веб-додатків;
- тестування веб-додатку;
- обґрунтування ефективності даного проекту;

Предмет дослідження – технології .NET для створення системи контролю процесу розробки веб-додатку «Особистий щоденник».

2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-ДОДАТКУ

2.1. Загальна інформація щодо веб-застосунку

Веб-додаток «Notes (Your Personal Diary)» призначений для надання можливості перегляду, редагування та видалення записів у щоденнику. Введені дані зберігаються в інформаційній базі даних, що дозволяє отримувати звіти з динаміки змін всіх показників за вибраний інтервал часу.

2.1.1 Огляд продукту

«Notes (Your Personal Diary)» – це веб-додаток, який допоможе користувачу зручно вести записи у щоденнику. Контекстна діаграма (рис 2.1) показує зовнішній об'єкт і системні інтерфейси для версії 1.0.



Рисунок 2.1 – Контекстна діаграма процесу внесення записів у щоденнику

Передбачаються випуски кількох версій системи для отримання універсального веб-додатку особистого щоденника, який дозволяв би

користувачеві створювати декілька щоденників для різних цілей, а також ділитися певними записами з іншими користувачами системи.

2.1.2 Мета

Визначити користувачів веб-додатку для покращення інтерфейсу та розширення функціональності.

2.1.3 Межі

Використання веб-додатку поширюється на всі країни.

Веб-додаток застосовується для всіх користувачів.

2.1.4 Масштаб проекту

Веб-додаток дозволить користувачам додавати об'єкти через інтернет, які представляють із себе звичайні записи у щоденнику.

Функції, реалізація яких запланована в цьому випуску:

1. Створення, перегляд, зміна і видалення записів у особистому щоденнику.
2. Забезпечення доступу до веб-додатку через мережу інтернет.
3. Додавання нових користувачів до бази даних.
4. Редагування інформації про користувача.

2.1.5 Обмеження реалізації

Веб-додаток створений за допомогою HTML 5.0, CSS3, JavaScript мови C# та технології ADO.NET.

2.1.6 Перспективи продукту

Досягти збільшення середнього рейтингу з щоквартального опитування за відповідністю потребам роботи веб-додатку на 0,5 бали протягом 3 місяців після першого випуску системи і на 1,0 бали протягом 12 місяців після другого випуску системи.

2.1.7 Діаграми потоків даних

Для опису присутньої функціональності була побудована діаграма потоків даних (DFD – data flow diagram) (рис 2.2).

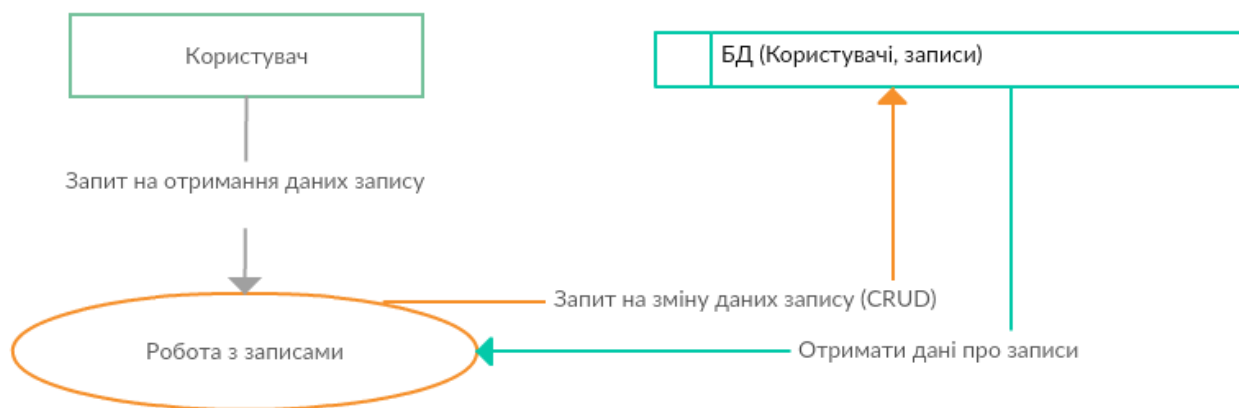


Рисунок 2.2 – Діаграма потоків даних (користувач)

Діаграми потоків даних описують зовнішні по відношенню до системи джерела і адресати даних, логічні функції, потоки даних та сховища даних, до яких здійснюється доступ.

Інформаційна система приймає ззовні потоки даних. Для позначення елементів середовища функціонування системи використовується поняття зовнішньої сутності. Всередині системи існують процеси перетворення інформації, які породжують нові потоки даних. Потоки даних можуть надходити на вхід до інших процесів, розміщуватися (і вилучатись) в накопичувачі даних, передаватися до зовнішніх сутностей.

2.2 Вимоги до веб-додатку

2.2.1 Функціональні вимоги

Функціональні вимоги наведені в таблиці 2.1, де під об'єктом мається на увазі «користувач». Так як додаток розрахований на декількох користувачів, тому є опція вибору «об'єкта».

Таблиця 2.1 – Опис функціональних вимог до веб-додатку

Вимога	Описання
Об'єкт. Вибір. Наявність	Для внесення даних або їх редагування, користувач має внести текст.
Об'єкт. Додання	Для додання нових записів, користувач має використати форму додання нового медіафайлу, заповнити поля та натиснути на кнопку «Create new note». Після додання, новий запис буде внесений до списку.
Об'єкт. Редагування	Для редагування записів, користувач має в таблиці вимірів натиснути на кнопку «Edit», заповнити поля та натиснути на кнопку «Edit note». Після редагування, запис буде змінено в списку.
Об'єкт. Видалення	Для видалення записів, користувач має в таблиці вимірів натиснути на кнопку «Delete», попередньо вибравши запис, який буде підлягати видаленню. Після проведених дій, запис буде видалений зі списку.
Об'єкт. Збереження	Для збереження запису, користувач має натиснути кнопку «Create new note» Після проведених дій, запис буде збережений.

2.2.2 Нефункціональні вимоги

2.2.2.1 Надійність

Якщо з'єднання між користувачем і системою розривається до того, як дані були додані, змінені або видалені, система веб-додатку повинна дозволити користувачеві відновити дані для перегляду.

2.2.2.2 Доступність

Веб-додаток «Notes (Your Personal Diary)» повинен бути доступним для всіх користувачів через мережу інтернет. Веб-додаток буде працювати 24 години на добу, тобто не має часових обмежень.

2.2.2.3 Безпека

Система повинна надавати користувачеві повний доступ до внесення змін в дані.

Читабельність коду можна розглядати як важливу складову супроводу. Код повинен бути зручним для аналізу, змін та тестування.

Модифікація, яка здійснюється для усунення дефектів, удосконалення або адаптації веб-додатку до змін в умовах функціонування, а також в складі і особливостях необхідних функцій є також одним із важливих факторів.

Додавання, видалення, зміна даних про об'єкти та їх показники повинні бути зручними для користувача веб-додатку.

2.2.2.4 Кросплатформність

Веб-додаток повинен мати властивість кросбраузерності.

Веб-додаток інсталяції не потребує, так як він буде доступний користувачеві за допомогою мережі інтернет через браузер.

2.2.2.5 Продуктивність

Веб-додаток повинен мати відмінну швидкість доступу до бази даних та завантаження сторінок, не навантажувати процесор та оперативну пам'ять.

2.3 Опис обраних технологій для розробки веб-додатку

2.3.1 HTML5

HTML (від англ. HyperText Markup Language - «мова гіпертекстової розмітки») – стандартизована мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузером; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою [5].

Мова HTML до 5-ї версії визначалася як додаток SGML (стандартної узагальненої мови розмітки за стандартом ISO 8879). Специфікації HTML5 формулюються в термінах DOM (об'єктній моделі документа).

Мова XHTML є більш суворим варіантом HTML, вона є наслідком синтаксису XML і є додатком мови XML в області розмітки гіпертексту.

У всесвітній павутині HTML-сторінки, як правило, передаються браузером від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування.

Переваги HTML5:

- легкість у створенні програмного коду;
- відсутність таблиць div;
- використання елементів article, footer, nav, header та ін.;
- відтворення Flash (анімації, відеоролики, музику) у вікні браузера і для цього не потрібні драйвера і утиліти;
- обмін даними з сервером став поліпшеним;
- отримання оповіщень про помилки при роботі сайту;

Недоліки HTML:

- трудомісткість;
- дорога вартість;
- низька безпека;

– не всі браузери мають можливість правильного відображення сайту на HTML5.

В цілому дана технологія є непоганою заміною застарілого HTML.

2.3.2 CSS3

CSS (англ. Cascading Style Sheets – каскадні таблиці стилів) – формальна мова опису зовнішнього виду документа, написаного з використанням мови розмітки [6].

Переважає використання як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватись до будь-яких XML-документів, наприклад, до SVG або XUL.

CSS використовується творцями веб-сторінок для задання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду веб-сторінок. Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (яка проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яка тепер проводиться за допомогою формальної мови CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

Переваги CSS:

1. Простота самої мови CSS разом з принципом відділення оформлення від змісту дає можливість скоротити час на розробку і підтримку сайту.
2. Є кілька варіантів дизайну сторінки для перегляду на різних пристроях. Наприклад, дизайн на екрані комп'ютера розрахований на одну ширину, і буде повністю виводитися на екран, а на мобільних пристроях він буде

підлаштовуватися до розмірів екрану і деякі елементи будуть виключені від показу, також і при друці, буде друкуватися потрібний текст, без зайвого (наприклад, без шапки меню).

3. Зменшується час завантаження сторінок web-сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. Завдяки цьому браузер завантажує тільки структуру документа, а також дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і може бути закешованим, – завдяки цьому зменшується трафік, час завантаження, а також навантаження на сервер.

4. Простота зміни дизайну. Один CSS управляє відображенням безлічі HTML-сторінок. Коли виникає необхідність змінити дизайн сайту, то нема чого правити кожен сторінку. Для подальшої зміни дизайну всього лише потрібно змінити CSS-файл, і як результат, зміна дизайну робиться швидше.

5. CSS надає додаткові можливості форматування.

6. Підвищення сумісності з різними платформами за рахунок використання web-стандартів.

Недоліки CSS:

1. Різне відображення верстки в різних браузерах. Якщо браузери застарілі, то можливо, що одні й ті ж дані CSS по-різному ними інтерпретуються.

2. Необхідність виправляти не тільки один CSS-файл, але і теги HTML. Часто зустрічається на практиці необхідність виправляти не тільки один CSS-файл, але і теги HTML, які пов'язані з селекторами CSS. Іноді це значно збільшує час редагування, а також і тестування.

Отже, CSS – це найсильніший інструмент і один з основних складових практично будь-якого web-сайту. Без CSS просто неможливо уявити сучасну діяльність web-розробників. Сьогодні CSS – це загальноприйнятий стандарт розробки, який приймається усіма без винятку компаніями-розробниками, що явно показує його значимість і необхідність.

2.3.3 ASP.NET

ASP.NET (Active Server Pages для .NET) – платформа розробки веб-додатків, до складу якої входить: веб-сервіси, програмна інфраструктура, модель програмування від компанії Майкрософт [7]. ASP.NET входить до складу платформи .NET Framework і є розвитком попередньої технології Microsoft ASP.

ASP.NET зовні багато в чому зберігає схожість з попередньою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі .NET і, отже, використовує всі нові можливості, що надаються цією платформою.

Оскільки ASP.NET ґрунтується на Common Language Runtime (CLR), яка є основою всіх додатків Microsoft .NET, розробники можуть писати код для ASP.NET, використовуючи мови програмування, що входять в комплект .NET Framework (C #, Visual Basic.NET, J# і JScript .NET).

Програмна модель ASP.NET ґрунтується на протоколі HTTP і використовує його правила взаємодії між сервером і браузером. При формуванні сторінки закладена абстрактна програмна модель Web Forms і на ній заснована основна частина реалізації програмного коду.

Переваги ASP.NET перед ASP:

- компільований код виконується швидше, більшість помилок усуваються ще на стадії розробки;
- значно поліпшена обробка помилок під час виконання запущеної готової програми, з використанням блоків try..catch;
- призначені для користувача елементи управління (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
- використання метафор, вже застосовуються в Windows-додатках, наприклад, таких як елементи управління і події;
- розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти додатки;

- ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на VB.NET, Delphi.NET, Visual C #, J # і т. д. ;
- можливість кешування всієї сторінки або її частини для збільшення продуктивності;
- можливість кешування даних, що використовуються на сторінці;
- можливість поділу візуальної частини та бізнес-логіки у різні файли («code behind»);
- розширювана модель обробки запитів;
- розширена подієва модель;
- розширювана модель серверних елементів управління;
- наявність Master-сторінок для завдання шаблонів оформлення сторінок;
- підтримка CRUD-операцій при роботі з таблицями через GridView;
- вбудована підтримка AJAX.

2.3.4 Microsoft Visual Studio

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ [8].

Переваги Microsoft Visual Studio:

1. Вбудований Web-сервер. Для обслуговування веб-додатків ASP.NET необхідний Web-сервер, який очікує веб-запити і обробляє відповідні сторінки. Наявність в Visual Studio інтегрованого Web-сервера дозволяє запускати веб-сайт прямо з середовища проектування, а також підвищує безпеку, виключаючи можливість отримання доступу до тестового Web-сайту з будь-якого зовнішнього комп'ютера, оскільки тестовий сервер може приймати з'єднання тільки з локального комп'ютера.

2. Підтримка безлічі мов при розробці. Visual Studio дозволяє писати код у своїй мові або будь-яких інших переважних мов, використовуючи весь час один і той же інтерфейс (IDE). Більше того, Visual Studio також дозволяє створювати веб-сторінки на різних мовах. Єдиним обмеженням є те, що в кожній Web-сторінці можна використовувати лише одну мову.

3. Менше коду для написання. Для створення більшості додатків потрібна достатня кількість стандартного стереотипного коду, а також Web-сторінки ASP. NET тому не виключення. Наприклад, додавання Web-елемента управління, Приєднання обробників подій і корегування форматування вимагає встановлення в розмітці сторінки ряду деталей. У Visual Studio такі деталі встановлюються автоматично.

4. Інтуїтивний стиль кодування. За замовчуванням Visual Studio форматує код за його введенням, автоматично вставляє необхідні відступи і застосовує колірне кодування для виділення елементів типу коментарів. Такі незначні відмінності роблять код більш зручним для читання та менш піддається помилці. Приймаються Visual Studio автоматично параметри форматування, які можна навіть налаштовувати, що дуже зручно.

5. Більш висока швидкість розробки. Багато з функціональних можливостей Visual Studio направлені на те, щоб допомогти розробнику зробити свою роботу швидше. Зручні функції, такі як функції IntelliSense (яка здатна відловлювати помилки та пропонувати правильні варіанти), функції пошуку та заміни (яка дозволяє шукати ключові слова як в одному файлі, так і в цілому проекті) та функції автоматичного додавання та видалення коментарів (які можуть тимчасово закрити блоки коду), дозволяють розробнику працювати швидко і ефективно.

6. Можливості налагодження. Параметри інструментів, що пропонуються в Visual Studio, є найкращим засобом для відстеження загадкових помилок та діагностики дивної поведінки.

Visual Studio також має й інші функції: можливість управління проектом; вбудована функція управління вихідним кодом; можливість рефакторизації

коду; потужна модель розширюваності. Більше того, у випадку використання Visual Studio 2008 Team System розробник отримує розширені можливості для модульного тестування, спільної роботи та управління версіями коду (що значно більше того, що пропонується в простих інструментах, як Visual SourceSafe).

До недоліків Microsoft Visual Studio можна відзначити неможливість відладчика (Microsoft Visual Studio Debugger) відстеження в коді режиму ядра. Налагодження в Windows в режимі ядра в загальному випадку виконується при використанні WinDbg, KD або SoftICE.

2.3.5 JavaScript

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу[9].

2.4 Опис архітектури веб-додатку

2.4.1 Діаграма розміщення

Діаграми розгортання (рис. 2.4) представляють фізичне розташування системи, показуючи, на якому фізичному обладнанні запускається та чи інша складова програмного забезпечення.

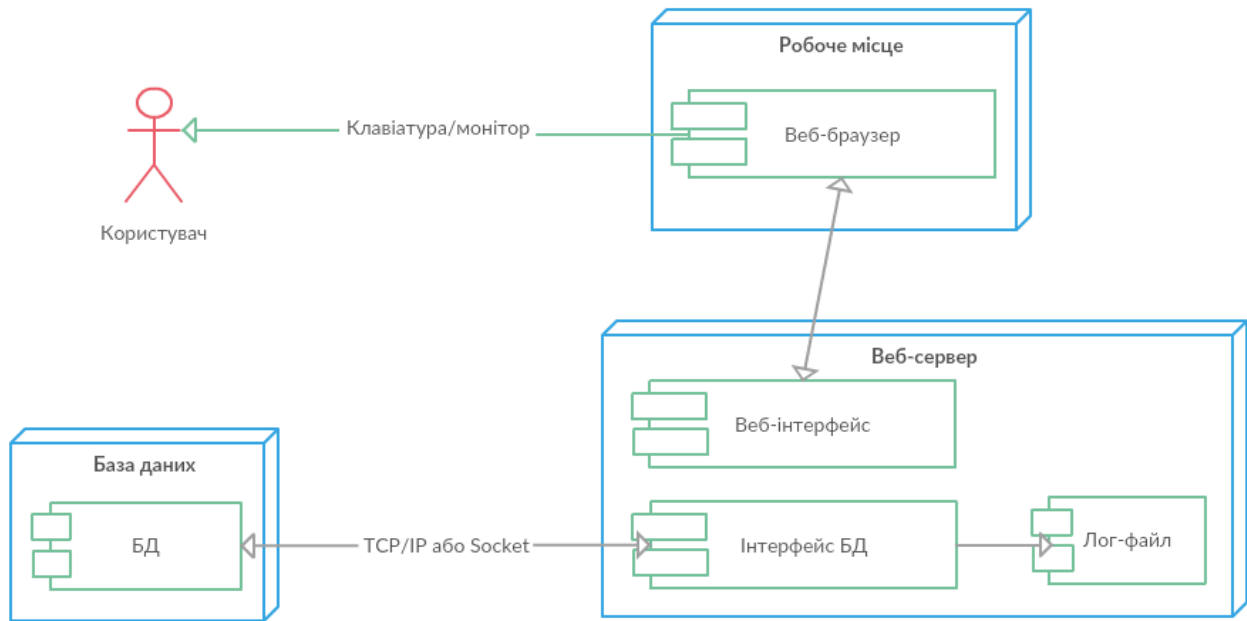


Рисунок 2.4 – Діаграма розміщення

Вузол (node) – це те, що може містити програмне забезпечення. Вузли бувають двох типів: пристрій і середовище виконання. Пристрій (device) – це фізичне обладнання: комп'ютер або пристрій, пов'язане з системою. Середовище виконання (execution environment) – це програмне забезпечення, яке може включати інше програмне забезпечення, наприклад операційну систему або процес-контейнер.

Вузли можуть містити артефакти (artifacts), які є фізичним уособленням програмного забезпечення; зазвичай це файли. Такими файлами можуть бути виконувані файли (такі як файли .exe, виконавчі файли, файли DLL, файли JAR, складання або сценарії) або файли даних, конфігураційні файли, HTML-документи і т. д. Перелік артефактів всередині вузла вказує на те, що на даному вузлі артефакт розгортається в систему, яка запускається.

Артефакти можна зображувати у вигляді прямокутників класів або перераховувати їх імена всередині вузла.

Артефакти часто є реалізацією компонентів. Це можна показати, задавши значення-мітки всередині прямокутників артефактів.

Інформаційні шляхи між вузлами представляють обмін інформацією в системі. Можна супроводжувати ці шляхи інформацією про використання інформаційних протоколів.

2.4.2 Діаграма компонентів

На діаграмі компонентів (рис. 2.5) показаний вид «білого ящика» внутрішньої структури двох пов'язаних підсистем – «Записи» і «Об'єкти». В UML «підсистема» представляє собою стандартний компонентний стереотип для більших компонентів, які в свою чергу містять дрібніші компоненти.

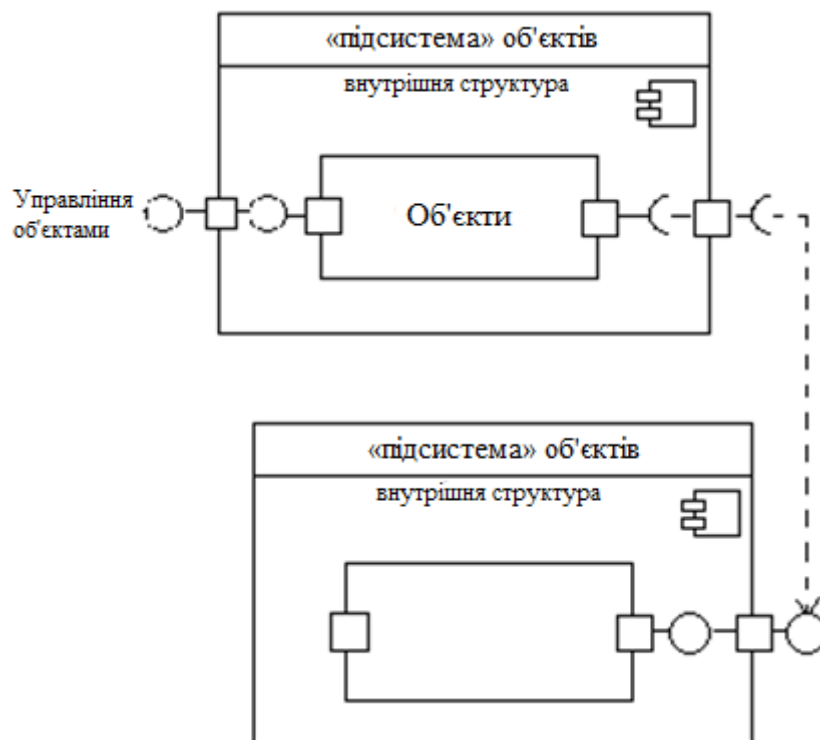


Рисунок 2.5 – Діаграма компонентів

Підсистема «Об'єкти» містить один компонент, пов'язаний з веб-додатком – «Об'єкти». Компонент «Об'єкти» дозволяє управляти елементами, використовуючи необхідний інтерфейс «Записи», що надається компонентом «Записи».

2.5 Модель бази даних

База даних, представлена у форматі MDF, складається з двох сутностей: User, Note. Модель бази даних представлена на рисунку 2.6.

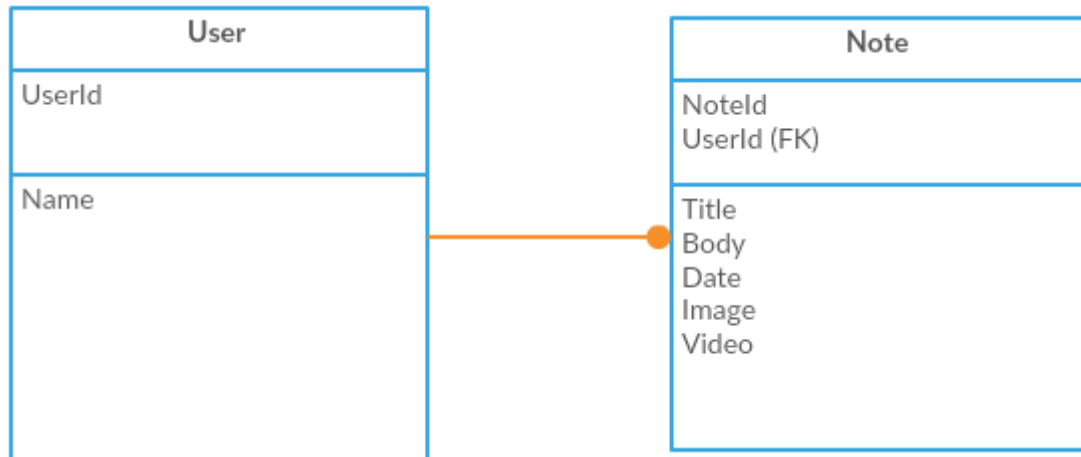


Рисунок 2.6 – Модель бази даних

Таблиця «User» призначена для зберігання інформації про об'єкти веб-додатку. Кожний запис складається з наступних полів: UserId, Name. Структура таблиці «User» наведена в таблиці 2.2.

Таблиця 2.2 – Опис структури таблиці «User»

Ключ	Ім'я поля	Тип даних	Опис
PK	UserId	Чисельний	ID користувача
	Name	Текстовий	Ім'я користувача

Таблиця «Note» призначена для зберігання інформації про кожен запису у щоденнику. Кожний запис складається з наступних полів: NoteId, UserId, Title, Body, Date, Image, Video. Структура таблиці «Note» наведена в таблиці 2.3.

Таблиця 2.3 – Опис структури таблиці «Note»

Ключ	Ім'я поля	Тип даних	Опис
PK	NoteId	Чисельний	Id нотатки
FK	UserId	Чисельний	Id користувача
	Title	Текстовий	Заголовок запису
	Body	Текстовий	Тіло запису
	Date	Дата	Дата запису
	Image	Blob	Прикріплене зображення
	Video	Blob	Прикріплене відео

В результаті було сформовано закінчений опис поведінки та структури веб-додатку який потрібно розробити для кінцевого користувача. Виконуючи специфікацію вимог, були визначені і описані функції системи, а також основні вимоги до зовнішнього інтерфейсу.

ОПИС ВЕБ-ДОДАТКУ

2.1 Тестування веб-додатку

2.1.1 Верифікація та валідація веб-додатку

При створенні нового об'єкта нерідко можуть виникати помилки. Але механізм валідації дозволяє інформувати користувача про помилки.

Розглянемо html-сторінку з формою додання користувачів до системи (рис. 3.1 – 3.3).

```
<form asp-controller="Note" asp-action="EntryProcess" method="get">
  <textarea type="text" asp-for="udModel.UserName" placeholder="" style="height:65px"></textarea>
  <textarea type="text" asp-for="udModel.Title" placeholder="" style="height:65px"></textarea>
  <textarea asp-for="udModel.strUpload" placeholder="What are you thinking right now?" required=""></textarea>

  <a title="" data-toggle="tooltip" data-placement="bottom" data-original-title="Picture"></a>
  <a title="" data-toggle="tooltip" data-placement="bottom" data-original-title="Video"></a>
  <button type="submit" class="btn btn-success red"><i class="fa fa-upload"></i>Create new note</button>
</form>
```

Рисунок 3.1 – Реалізація форми додання користувачів до системи

```
protected void FormView1_ItemInserted(object Sender, FormViewInsertedEventArgs e)
{
    if (e.Exception != null)
    {
        Response.Redirect("Oops: " + e.Exception.Message);
        e.ExceptionHandled = true;
    }
}
```

Рисунок 3.2 – Реалізація формування виключення

```

USE [WeightControlDatabase]
GO
/***** Object: Trigger [dbo].[DeleteTheSameUser]    Script Date: 30.03.2019 22:36:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[DeleteTheSameUser]
ON [dbo].[Users]
AFTER INSERT, UPDATE NOT FOR REPLICATION
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @new_user nchar(40)
    DECLARE @count int
    SELECT @new_user = Name FROM inserted
    SELECT @count=COUNT(Name) FROM Users WHERE Name = @new_user
    IF (@count>0)
    BEGIN
        ROLLBACK
    END
END

```

Рисунок 3.3 – Реалізація триггеру для запобігання додаванню користувачів з однаковими іменами

2.1.2 Ручне тестування

Тестові випадки наведені в таблицях 3.1 – 3.5.

Таблиця 3.1 – Тестовий випадок №1

Назва тесту	Тест на редагування запису.
Описання	Тестування редагування об'єкту в системі.
Тестові кроки	Введення інших даних до старого запису .
Очікуваний результат	Вдале редагування імені.
Фактичний результат	Вдале редагування імені.
Статус	Пройдений.

Таблиця 3.2– Тестовий випадок №2

Назва тесту	Тест на редагування запису без змін.
Описання	Тестування редагування запису в системі без дійсних змін.
Тестові кроки	Спроба редагування без змін
Очікуваний результат	Невдале редагування запису.
Фактичний результат	Невдале редагування запису.
Статус	Пройдений.

Таблиця 3.3 – Тестовий випадок №3

Назва тесту	Тест на видалення запису в системі.
Описання	Тестування видалення запису в системі.
Тестові кроки	Видалення обраного запису.
Очікуваний результат	Вдале видалення з системи.
Фактичний результат	Вдале видалення з системи.
Статус	Пройдений.

Таблиця 3.4 – Тестовий випадок №4

Назва тесту	Тест на додання нових записів.
Описання	Тестування додання записів.
Тестові кроки	Введення нових записів в щоденник.
Очікуваний результат	Вдале додання записів до системи.
Фактичний результат	Вдале додання записів до системи.
Статус	Пройдений.

Таблиця 3.5– Тестовий випадок №5

Назва тесту	Тест на додавання медіафайлів до системи
Описання	Тестування додавання медіафайлів до системи.
Тестові кроки	Введення нового медіафайлу до системи.
Очікуваний результат	Вдале додавання.
Фактичний результат	Вдале додавання.
Статус	Пройдений.

Перевіряючи працездатність всіх компонентів програми, ніяких дефектів в роботі програми не було виявлено.

2.2 Керівництво користувача

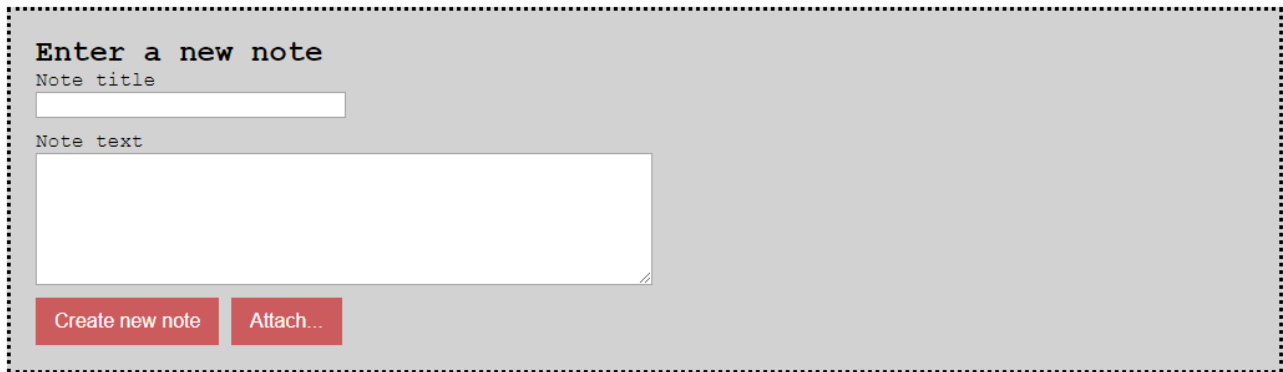
2.2.1 Загальний опис можливостей

Для адміністрування веб-додатку, користувач має наступні можливості:

- додавати, видаляти, змінювати інформацію про користувачів;
- додавати, видаляти та записи у щоденнику.

2.2.2 Об'єкти

Користувач має право додання нових об'єктів до системи, їх редагування та видалення. Для виконання даних дій користувач має використати форму, яка зображена на рисунку 3.4.



The image shows a web form titled "Enter a new note". It has a light gray background and is enclosed in a dashed border. The form contains two input fields: "Note title" (a single-line text box) and "Note text" (a multi-line text area). Below the text area are two red buttons: "Create new note" and "Attach...".

Рисунок 3.4 – Форма внесення змін до таблиці об'єктів

Після проведення певних дій, результати будуть відображені в таблиці.

2.2.3 Демонстрація записів в щоденнику

Користувач має право додання нових записів, їх редагування та видалення. Для виконання даних дій користувач має використати форму, яка зображена на рисунку 3.5.

The image shows two parts of a web application interface. The top part is a 'Your Notes' section with a date input field containing 'ДД.ММ.ГГГГ'. Below it is a list item 'Пример записи' with a timestamp '20/5/2019 18:50:01' and 'Delete' and 'Edit' buttons. The bottom part is an 'Edit selected note' modal with fields for 'New note title' (containing 'Пример записи') and 'New note text' (containing 'Пример записи'). It has 'Edit note', 'Attach...', and 'Close' buttons at the bottom.

Рисунок 3.5 – Форма внесення змін до таблиці записів об’єктів

Після проведення певних дій, результати будуть відображені в таблиці.

2.2.4 Фільтрування записів по даті.

Введені дані можна фільтрувати за датою для зручного перегляду записів в конкретну дату.

This screenshot shows the 'Your Notes' section with the date input field now containing '20.05.2019'. The list item 'Пример записи' and its timestamp '20/5/2019 18:50:01' are still visible, along with the 'Delete' and 'Edit' buttons.

Рисунок 3.6 – Приклад фільтрації

ВИСНОВКИ

В ході виконання курсової роботи було виконано наступні етапи:

1. Було проаналізовано предметну область індивідуального завдання курсової роботи.
2. Проведено первинну розробку серверної архітектури додатку.
3. Проведено проектування архітектури інформаційної системи з використанням технології «клієнт-сервер».
4. Проведено програмну реалізацію та налагодження розробленої інформаційної системи.

Веб-додаток щоденник призначений для внесення інформації користувачем. Областю застосування даного додатку виступає допомога для зручного зберігання, перегляду або редагування даних.

Отже, в ході виконання курсової роботи, послідовно застосовуючи принципи об'єктно-орієнтованого аналізу, документо-орієнтованого моделювання даних, і середу розробки MS Visual Studio для програмування веб-додатку, було розроблено і реалізовано онлайн ресурс особистий щоденник.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

- 1 Онлайн щоденник Diary-anek [Електронний ресурс] – Режим доступу: URL – <https://diary.anek.ws>, перегляд: 30.03.2019
- 2 Щоденник “МойДневник” [Електронний ресурс] – Режим доступу: URL – <http://mojdnevnik.ru>, перегляд: 30.03.2019
- 3 Додаток DnevniX[Електронний ресурс] – Режим доступу: URL – <https://dnevniX.ru>, 30.03.2019
- 4 Онлайн щоденник Фрагментар [Електронний ресурс] – Режим доступу: URL – fragmenter.net, перегляд: 30.03.2019
- 5 HTML5 [Електронний ресурс] – Режим доступу: URL – <https://ru.wikipedia.org/wiki/HTML5>, перегляд: 30.03.2019
- 6 CSS [Електронний ресурс] – Режим доступу: URL – <https://ru.wikipedia.org/wiki/CSS>, перегляд: 30.03.2019
- 7 ADO.NET [Електронний ресурс] – Режим доступу: URL – <https://ru.wikipedia.org/wiki/ADO.NET>, перегляд: 30.03.2019
- 8 Програма Visual Studio [Електронний ресурс] – Режим доступу: URL – <https://www.visualstudio.com/ru/>, перегляд: 30.03.2019
- 9 Флэнаган Д. JavaScript. Подробное руководство, 6-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2017. – 1080 с.