

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

Система технічної підтримки фінансових акаунтів клієнтів банку

Студентки 4 курсу, 7 групи,
напряму підготовки (6. 050103
«Програмна інженерія»)
спеціалізації Інженерія
програмного забезпечення

Смирнової Анни
Михайлівни

підпис студента

Науковий керівник
кандидат технічних наук,
професор

Пашорін Валерій
Іванович

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

підпис гаранта

Київ 2019

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Система технічної підтримки фінансових акаунтів клієнтів банку	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В.				Зміст	1	25
Керівник		Пашорін В.І.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Смирнова А.М.						
					Зміст			

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ІНФОРМАЦІЯ ЩОДО ВИКОРИСТАНИХ ІНСТРУМЕНТІВ	5
1.1 Задіяні мови програмування	5
1.2 Тестування систем	6
1.3 Формування технічного завдання	7
1.4 Висновок до розділу 1	9
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ.....	10
2.1 Моделювання додатку	10
2.2 Організація файлів проекту	14
2.3 Розробка користувацького інтерфейсу	15
2.4 Висновок до розділу 2	18
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ	19
3.1 Створення макету та паркування сайту	19
3.2 Побудова інтерфейсу	20
3.3 Створення логіки додатку	22
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	24
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	25

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.		Криворучко О.В			Система технічної підтримки фінансових акаунтів клієнтів банку	Стадія	Аркуш	Акрушів
Керівник		Пашорін В.І.				Зміст	2	25
Гарант		Цензура М.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив		Смирнова А.М.						
					Зміст			

ВСТУП

На створення будь-якого програмного забезпечення може наштовхнути ціла група мотиваторів, серед яких прибуток, цікавість, попит, необхідність написати наукову роботу або пошук нових способів працювати ефективніше.

Для цієї роботи вибір теми був скоріше результатом наявності ідеї та досвіду роботи у предметній сфері. В процесі написання це рішення виправдало себе бо відпала необхідність проводити додаткові дослідження для побудови моделей опрацювання запитів або сортувати та тестувати конкурентне програмне забезпечення що дозволило використати звільнений час для роботи на практичну частину роботи.

Метою написання роботи є винайдення найоптимальнішого алгоритму обробки задач менеджером та машиною у тандемі.

Завданням є створення системи для організації роботи команди технічної підтримки банку.

Об'єктом дослідження є діяльність банку

Предметом дослідження є процес створення системи автоматизації процесу звітування у департаменті технічної підтримки банку.

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Система технічної підтримки фінансових акаунтів клієнтів банку	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В.				В	3	25
Керівник		Пашорін В.І.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Смирнова А.М.						
					Вступ			

Веб додаток створювався як ідея для програмного забезпечення яке б допомогло менеджеру команди пришвидшити свою роботу та автоматизувати та покращити всі аспекти які доцільно за допомогою машин. Корисно імплементувати такі надбання при напруженій роботі з клієнтами особливо у такій важливій та складній галузі як банківська справа. Необхідність людського опрацювання унікальних запитів які викликані діями клієнтів або а примхами законодавства досить багато тому доцільно для створення найефективнішою команди для опрацювання цих елементів забезпечити тренуваних спеціалістів найкращим програмним забезпеченням для роботи. У сучасній системі багато ланок що можуть потенційно сповільнити роботу на значний період та при цьому не являються підконтрольними менеджерам. Необхідно аби елементи що все ж таки підконтрольні стороні компанії були настільки швидкій у переробці що можливі відхилення від еталонного часу по провині клієнта опрацювання запиту були ними компенсовані.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		4

РОЗДІЛ 1

ІНФОРМАЦІЯ ЩОДО ВИКОРИСТАНИХ ІНСТРУМЕНТІВ

1.1 Задіяні мови програмування

Щоб розробити веб-додаток, потрібно обрати сервер, базу даних, мову програмування, фреймворк та засоби інтерфейсу які ви збираєтеся використовувати. Ці технології веб-розробки розгортаються одна на одну і тому ще називаються стеком (Рис 1.1).

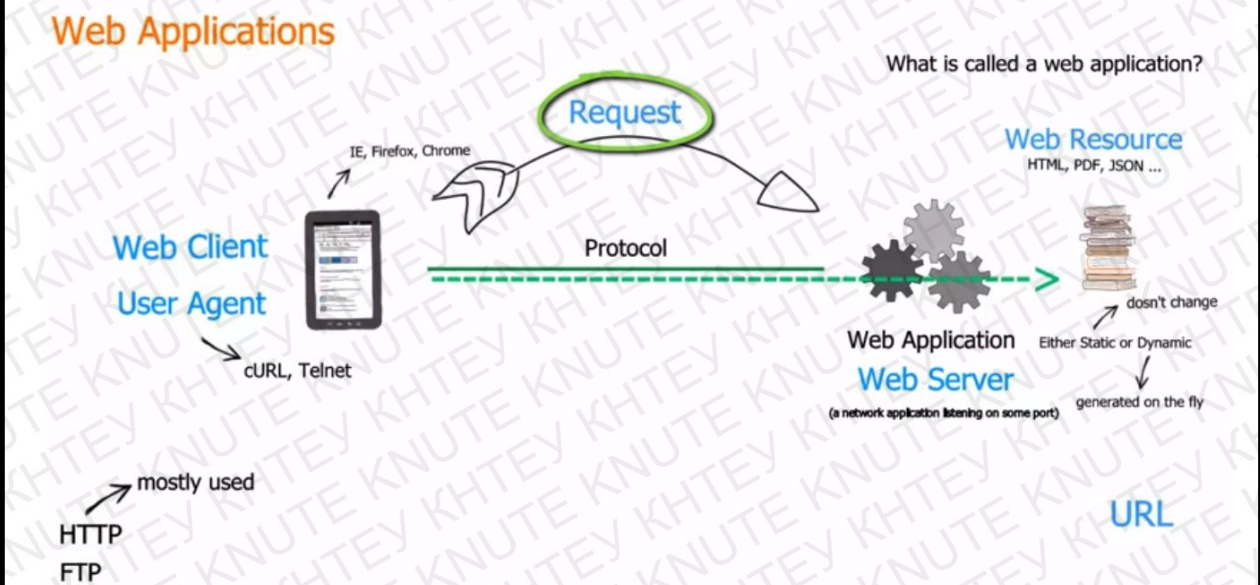


Рис. 1.1 Спрощена схема роботи з web-додатками

Існує дві сторони веб-розробки: клієнтська і серверна. Клієнтська сторона також називається фронтенд. Серверне програмування включає в себе додаток (і мова програмування, що керує ним), базу даних і сам сервер.

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Система технічної підтримки фінансових акаунтів клієнтів банку	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				РІ	5	25
Керівник		Пашорін В.І.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Смирнова А.М.			Розділ 1			

Веб-розробка зі сторони клієнта (тобто інтерфейс) включає в себе все, що користувачі бачать на своїх екранах, наприклад, html, css, JavaScript тощо.

Сторона сервера закрита для користувачів, там виконується створення логіки веб-сайтів і програм. Фреймворки для мов програмування пропонують безліч інструментів для більш простого і швидкого кодування, а деякі з них - python (django), java (spring), php (laravel) Node.js, рантайм JavaScript, також використовується для програмування на сервері.

1.2 Тестування систем

На усіх етапах роботи важливо перевіряти стан проекту та його відповідність наданим вимогам. Для різних ситуацій та методів роботи над програмними рішеннями існують різні типи тестування програмного забезпечення, такі як функціональне тестування, не функціональне тестування, автоматизоване тестування, Agile тестування та їх підтипи.

Кожен тип тестування має свої особливості, переваги та недоліки. Однак у цій роботі були використані мною обрані методи і доцільно провести їх огляд

- а) Приймальна перевірка (Acceptance Testing) Клієнтом проводиться приймальний тест і перевіряє, чи відповідає проект бізнес-вимогами і чи працює відповідно до потреб кінцевого користувача. Клієнт приймає програмне забезпечення тільки тоді, коли всі функції та функціональні можливості працюють як слід
- б) Тест доступності (Accessibility Testing). Метою тестування доступності є визначення чи доступне програмне забезпечення або система для людей з обмеженими можливостями.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		6

в) Бета тестування (Beta Testing). Бета-тестування проводиться для виявлення серйозних збоїв у програмному забезпеченні або продукті та задоволення бізнес-вимог з точки зору кінцевого користувача. Бета-тестування є успішним, коли клієнт приймає програмне забезпечення. Як правило, це тестування виконується кінцевими користувачами. Це остаточне тестування перед видачею програми з комерційною метою. Зазвичай бета-версія випущеного програмного забезпечення або продукту обмежена певною кількістю користувачів у певній області.

г) Back-end тестування (Back-end Testing). Кожен раз, коли вхідні дані або дані вводяться в інтерфейсному додатку він зберігає їх у базі даних і тестування такої бази даних називається тестуванням бази даних або тестуванням Backend. Існують різні бази даних, такі як SQL Server, MySQL і Oracle. Тестування баз даних включає тестування структури таблиці, схеми, процедур збереження, структури даних тощо.

Вищезгадані типи тестування програмного забезпечення є лише частиною тестування. Тим не менш, є ще список з більш ніж 100 + видів тестування, але всі типи тестування не використовуються у всіх типах проектів.

1.3 Розробка технічного завдання

Головним завданням проекту є створення систем для автоматизації менеджерських завдань з організації команд та їх проектів у банку. Другорядною метою є можливість використання додатку у звітних цілях. Система має відкриватися у браузері на десктопних пристроях. Мова інтерфейсу додатку англійська. Інформація про працівників, проекти та команди має зберігатися у базі даних на сервері. Дизайн користувацького

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		7

кабінету має бути інтуїтивним, ергономічним та легким в оволодінні. У системі передбачені наступні екрани:

- а) Екран авторизації
- б) Екран створення команди (екран за замовчуванням)
- в) Екран перегляду команд
- г) Екран створення проекту
- д) Екран перегляду проектів
- е) Екран реєстрації персоналу
- ж) Екран перегляду персоналу

Екран створення команди має надавати можливість вписати у форму коротку та повну назви команд та зберегти у базі даних інформацію. Екран перегляду команд має показати всі створені команди та надати можливість відредагувати дані або видалити запис. Екрани роботи з проектами потребують аналогічного функціоналу

Екран реєстрації персоналу має мати наступні характеристики:

- а) Секція персональних даних
- б) Секція підв'язки до команди та/або персонального проекту
- в) Секція контактних даних
- г) Перевірка введених даних

					КНТЕУ 6.050103 07-20.БР	Арквш
Изм.	Дикли	№ докум	Підпис	Дат		8

1.4 Висновок до розділу 1

Технології для розробки веб-додатків різноманітні та цікаві для аналізу та порівняння. Обираючи для себе фреймворк чи мову для написання сайтів розробник повинен провести серйозне дослідження конкуренції та перспективи для кожного інструменту. Розглянувши навіть порівняно невеликий обсяг інформації стає зрозуміло що цей вибір вирішує все – чи ви викинете свій час на оволодіння технологією що вийде з використання через рік, чи знайдете інструмент для швидкісного розвитку у сфері. Правильно сформоване технічне завдання дозволить перейти до проектування системи та подальшої розробки проекту.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Допили	№ докум	Підпис	Дат		9

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Моделювання додатку

Програмні моделі - це способи вираження програмного забезпечення. Зазвичай для оформлення програмного забезпечення використовують якусь абстрактну мову або зображення. Для об'єктно-орієнтованого програмного забезпечення використовується мова об'єктного моделювання, наприклад UML, для розробки та вираження дизайну програмного забезпечення.

Діаграма UML - це діаграма, заснована на UML (Unified Modeling Language) з метою візуального представлення системи разом з її основними учасниками, ролями, діями, артефактами або класами, щоб краще розуміти, змінювати, підтримувати або документувати інформацію про систему. Існує кілька типів діаграм UML, і кожна з них має іншу мету, незалежно від того, чи вона розробляється до реалізації або після (як частина документації).

Дві найпоширеніші категорії, які охоплюють всі інші типи, - це поведінкова діаграма UML і структурна UML-діаграма. Як впливає з назви, деякі діаграми UML намагаються проаналізувати та зобразити структуру системи або процесу, тоді як інші описують поведінку системи, її суб'єктів та її будівельних компонентів. У процесі розробки цього проекту були побудовано 2 поведінкові та 1 структурна діаграми.

Основною частиною системи є функціональні вимоги, які виконуються системою. Діаграми Use Case використовуються для

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Система технічної підтримки фінансових акаунтів клієнтів банку Розділ 2	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В.				Р2	10	25
Керівник		Пашорін В.І.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Смирнова А.М.						

аналізу вимог високого рівня. Ці вимоги виражаються через різні Use Case діаграми (Рис 2.1).

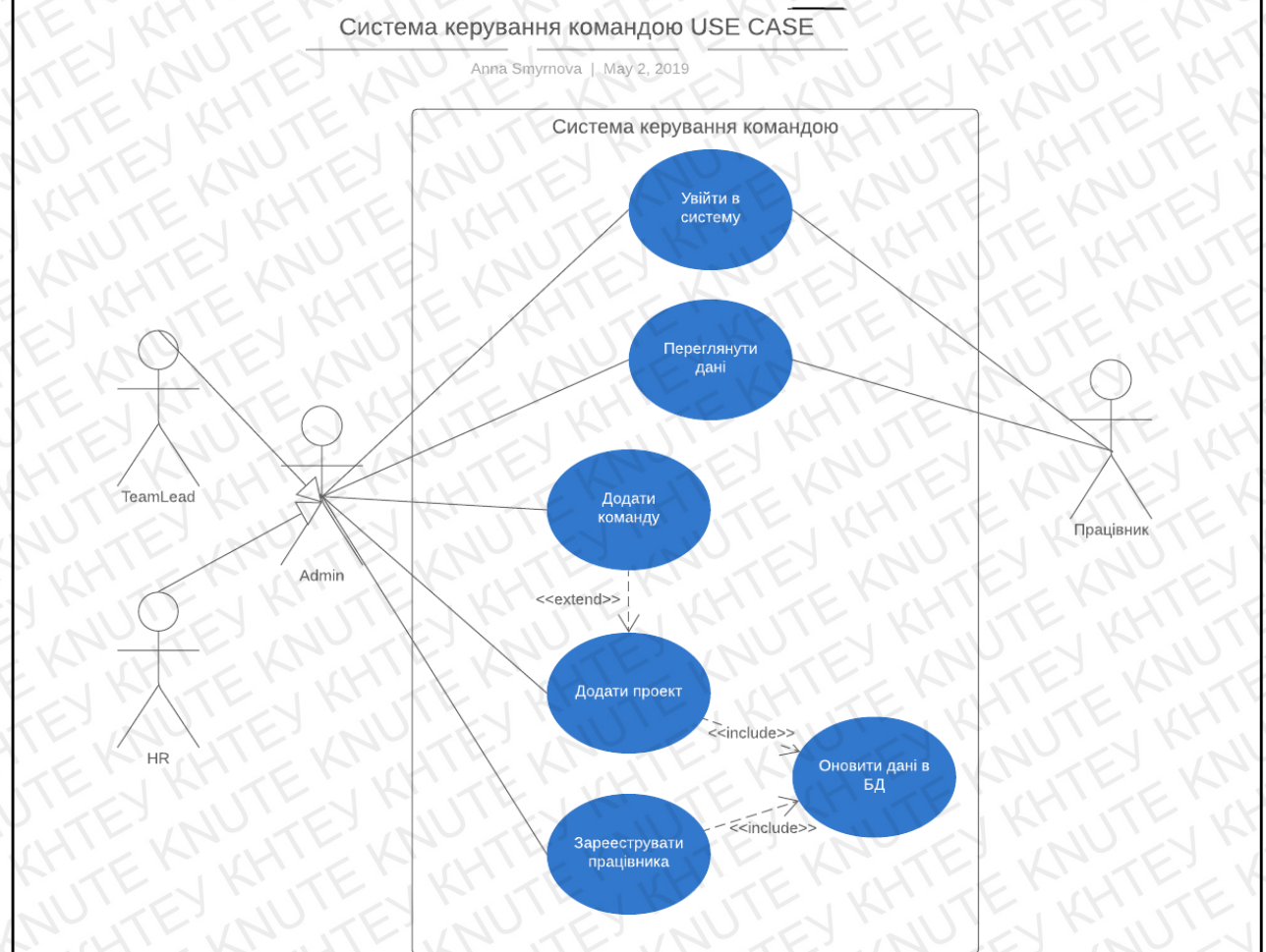


Рис. 2.1. Use Case діаграма проекту

Наступною розробленою в процесі роботи діаграмою була Sequence diagram або діаграма послідовностей. Діаграми послідовностей є найважливішими UML-діаграмами не тільки в комп'ютерній науці, але й як моделі моделей для розробки бізнес-додатків. Як випливає з назви, діаграми послідовностей описують послідовність повідомлень і взаємодій, які відбуваються між акторами і об'єктами. Актори або об'єкти можуть бути активними тільки тоді, коли це необхідно або коли інший об'єкт хоче спілкуватися з ними. Всі комунікації представлені в хронологічному порядку (Рис 2.2)

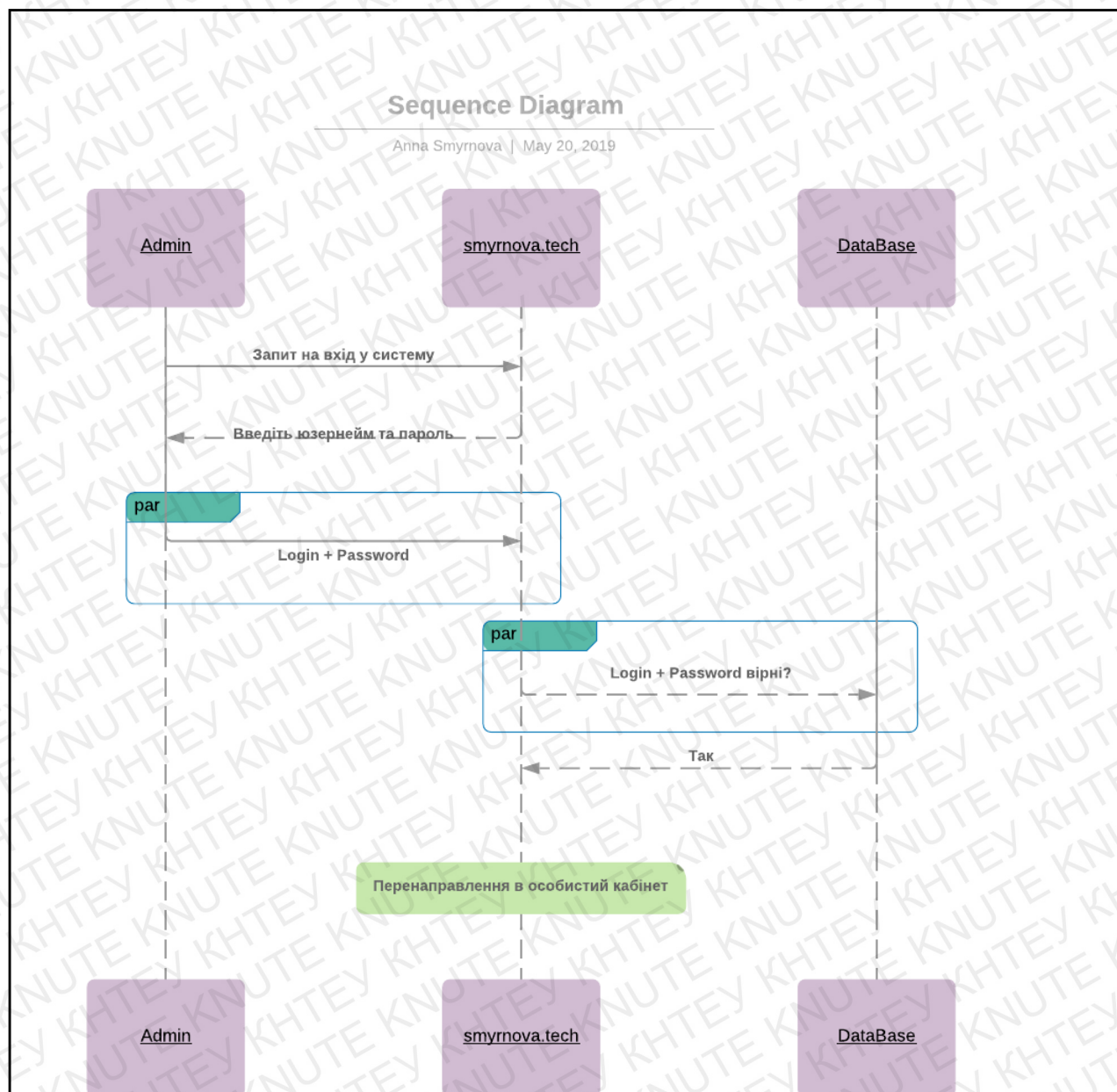


Рис. 2.2. Sequence діаграма проекту

Для створення структурної діаграми був обраний підтип Class diagram. Діаграма класу є найпоширенішим типом діаграми для програмної документації. Оскільки більшість програмних засобів, створених сьогодні, все ще базується на парадигмі об'єктно-орієнтованого програмування, використання діаграм класів для документування програмного забезпечення виявляється логічним. Це

відбувається тому, що ООП базується на класах і відносинах між ними.

Class Diagram with UML Notation

Anna Smyrnova | May 20, 2019

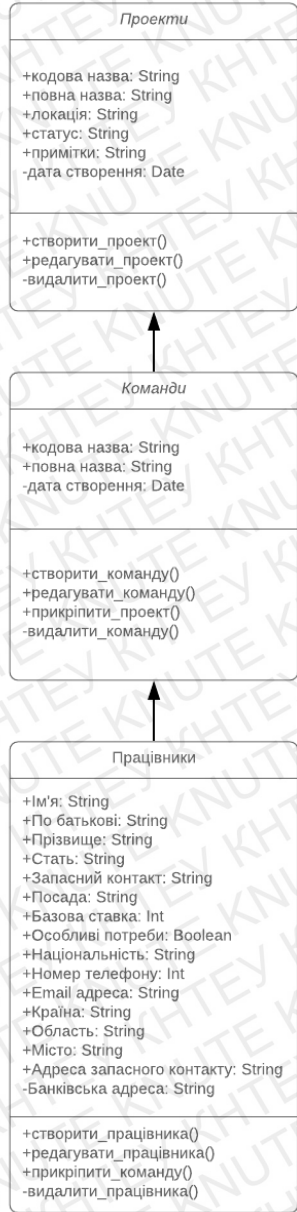


Рис. 2.3. Діаграма класів проекту

Розроблені діаграми дозволять провести розгорнуту специфікацію проекту та візуально представити напрацювання. На основі створених діаграм можливо продовжити розробку додатку після ухвалення замовником.

2.2 Систематизація файлів проекту

Внутрішня будова проекту - це роботи, які побачать лише інші технологи, але користувачі отримають користь від ретельного планування та продуманості. Обговорення структури сайту та сторінки зазвичай орієнтовані на найбільш помітні аспекти навігації, інтерфейсу користувача та організації вмісту, але увага до структури файлів і каталогів, а також те, як ви називаються об'єкти на сайті, можуть призвести до спрощення редагування у майбутньому:

- а) Розуміння: поточна команда, майбутні супроводжуючі та більшість користувачів сайту отримають користь від ретельної та послідовної номенклатури простого сайту
- б) Гнучкість: якщо кожен об'єкт на сайті має назву, все можна знайти, створити та програмно обробити набагато легше
- в) Доступність: іменовані об'єкти є більш доступними для керування програмами та таблицями стилів для всіх типів медіа та надають гнучкість у впровадженні універсальних функцій доступності на сайті
- г) Оптимізація пошуку: Ретельна номенклатура сайту значно полегшує оптимізацію сайту для видимості пошукової системи.
- д) Масштабованість: не можна збільшувати або автоматизувати зміни у випадково називаній, безсистемній збірці файлів і каталогів; Несистемою не можна систематично управляти або структурно розширювати

При розробці додатку було створене дерево папок стандартне для веб проектів що забезпечило збереження приведених характеристик проекту (Рис 2.4).

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		14

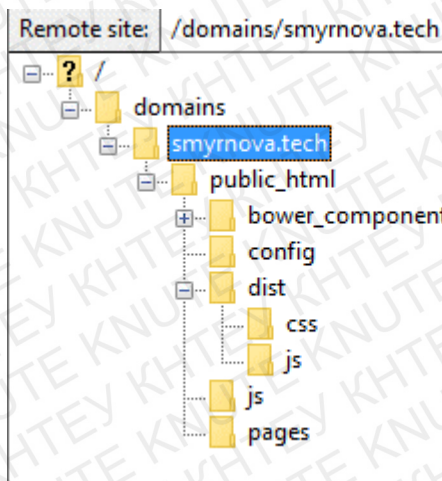


Рис. 2.4. Дерево файлів на сервері Hostinger

2.3 Розробка користувацького інтерфейсу

Дизайн інтерфейсу користувача сайту відіграє важливу роль у доведенні до нього великого обсягу веб-трафіку. Якісний користувацький інтерфейс не тільки збільшує зручність використання сайту, але й забезпечує плавне завершення будь-якої задачі, роблячи таким чином враження від використання приємним і та відповідним до вимог користувачів. Для створення гарних користувацьких інтерфейсів у команді розробників обов'язково повинен бути залучений UI дизайнер. Їхня робота полягає в тому, щоб переконатися, що кожна сторінка та кожен крок, який користувач переживатиме у взаємодії з готовим продуктом, відповідатиме загальному баченню, створеному клієнтом. В пошуках основи для створення інтерфейсу додатку було вирішено використати принципи Material design. Матеріальний дизайн - це візуальна мова, яка синтезує класичні принципи гарного дизайну з інноваційними технологіями і наукою. Надзвичайно важливо, щоб Material Design відповідав очікуванням користувачів щодо поведінки компонентів. Наприклад, екранні об'єкти виглядають більш переконливими, якщо вони

Изм.	Дикли	№ докум	Підпис	Дат

КНТЕУ 6.050103 07-20.БР

Архив
15

дотримуються законів гравітації. Застосування Material Design ефективно вимагає розуміння цих принципів:

- а) Метафора матеріалу. Дизайн показує користувачам атрибути які їм вже знайомі. Знайомі елементи реального світу, такі як кнопки, повинні миттєво дозволити користувачам побачити, що робити. Поверхні та шви / краї повинні доповнювати їх, пропонуючи підказки. Використовуйте реалістичне освітлення / затемнення, щоб розділити простір дизайну і розкрити рухомі частини.
- б) Сміливі, графічні та добре підібрані елементи. Зробити конструкції виразними, щоб користувачі миттєво помітили ієрархію і могли зосередитися і отримати відповідні дані. Відмінне використання кольорів та їх відсутності, сильні зображення та велика друкарська форма допомагають користувачам зануритися в досвід. Дизайн спонукає користувачів діяти і показувати функціональність, підкреслюючи значки і поверхні як підказки.
- в) Передача інформації рухом. Керування користувачами, показуючи зміни, що трапляються безпосередньо з їхніх дій. Зробити анімацію в одному середовищі/сцені, в безперервних поточних рухах. Відповіді є чіткими, але тонкими: відсутні паузи та непередбачувані рухи.

Для розробки інтерфейсу на етапах узгодження з замовником було використано інструмент Fluid UI (Рис 2.5) що дозволив продемонструвати структуру та подання інформації. Доцільно зауважити що сама платформа Fluid UI побудована на основі та дотримується канонів Material Design що стало одним з критеріїв які розглядалися у пошуках інструменту для створення макету.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		16

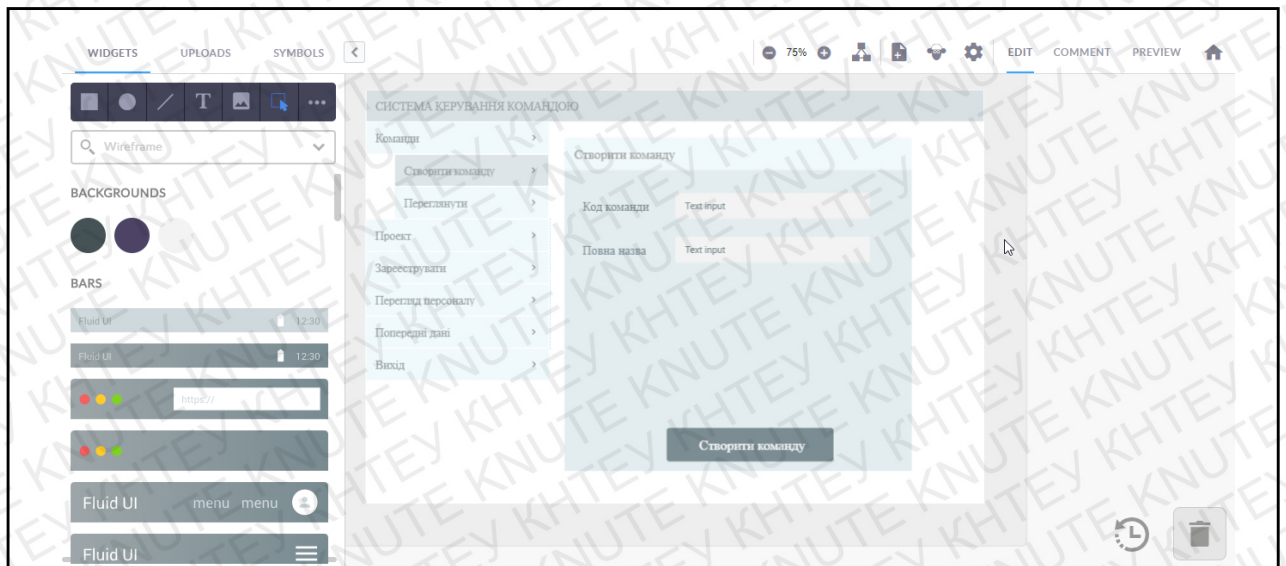


Рис. 2.5. Дизайн Fluid UI для розробки інтерфейсів

Після внесення правок та доробок дизайн було узгоджено і почалася робота над перенесенням макету в HTML та CSS.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Допилки	№ докум	Підпис	Дат		17

2.4 Висновок до розділу 2

Використовуючи наявну інформацію та рекомендації щодо процесу розробки разом з результатами дослідження зібраних у першому розділі був створений план для подальших дій та старту розробки додатку. На основі напрацювань що включають у себе три діаграми та макет дизайну буде доцільно приступити до розробки додатку та міграції системи у хмарне сховище. Було проведене ознайомлення з обраними інструментами для роботи та аналіз їх потенціалу для подальшої роботи. Відгуки про поточне виконання роботи від клієнта та незалежних експертів допоможе ввести корекції у процесі розробки та написання коду.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		18

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ

3.1 Створення макету та придбання домену

Роботу над реалізацією системи було розпочато зі створення скелету веб-сторінки та розробки таблиць та їх структури на локальному сервері. Після встановлення та конфігурації Apache, PHP, MySQL Server та PhpMyAdmin була підключена база даних до Index сторінки(Рис 3.1).

```
<?php
class Database {
    private $_connection;
    private static $_instance;
    private $_host = "localhost";
    private $_username = "u505761087_anna";
    private $_password = "diplanna";
    private $_database = "u505761087_dipl";

    public static function getInstance() {
        if(!self::$_instance) {
            self::$_instance = new self();
        }
        return self::$_instance;
    }

    public function __construct() {
        $this->_connection = new mysqli($this->_host, $this->_username, $this->
            _password, $this->_database);

        if(mysqli_connect_error()) {
            trigger_error("Failed to connect to MySQL: " . mysqli_connect_error(),
                E_USER_ERROR);
        }
    }

    // Magic method clone is empty to prevent duplication of connection
    private function __clone() { }

    public function getConnection() {
        return $this->_connection;
    }
}

?>
```

Рис. 3.1. Фрагмент коду Коннектора до бази даних

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.		Криворучко О.В.			Система технічної підтримки фінансових акаунтів клієнтів банку	Стадія	Аркуш	Акрушів
Керівник		Пашорін В.І.				РЗ	19	25
Гарант		Цензура М.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив		Смирнова А.М.						
					Розділ 3			

Під час тестування нового коду паралельно з розробкою було прийняте рішення перемістити додаток з локального сервера на хмарне сховище для отримання навичок роботи з хостингами та можливості презентації на машині без встановлених спеціальних програм. Для роботи був обраний хостинг сервіс Hostinger через вигідні умови та можливість придбати сервіс на відносно малий період часу. Був знайдений вільний домен smugnova.tech та відправлений запит на направлення цього домену на придбаний хостинг.

Поки оброблявся запит на підв'язання доменного імені була реалізована функція авторизації у системі (Рис 3.2).

```
function login($loginid,$password){
    if(!ctype_alpha($loginid) || !ctype_alpha($password)){
        echo "<script>alert('Either LoginId or Password is Missing')</script>";
    }
    else{
        $db = Database::getInstance();
        $mysqli = $db->getConnection();
        $query = "SELECT loginid, password FROM tbl_login where loginid=? and password=? ";
        $stmt= $mysqli->prepare($query);
```

Рис. 3.2. Фрагмент коду перевірки юзернейму та паролю

Було вирішено використати відносно нескладний та надійний механізм що дозволяє редагувати юзерів та доступ через виділену таблицю у базі даних. При розміщенні системи на сервері компанії рекомендується надати прямий доступ до бази даних тільки IT спеціалісту що створить необхідний рівень захисту від непередбачуваних змін що буде достатнім для програми такого типу.

3.2 Міграція системи на сервер

Після отримання повідомлення про успішне під'єднання домену до серверу Hostinger стало можливо розпочати міграцію у хмарний

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		20

сервіс додатку. Спочатку було доцільно перенести створену раніше базу даних через інтерфейс PhpMyAdmin відкритий з панелі управління на хостингу (Рис 3.3).

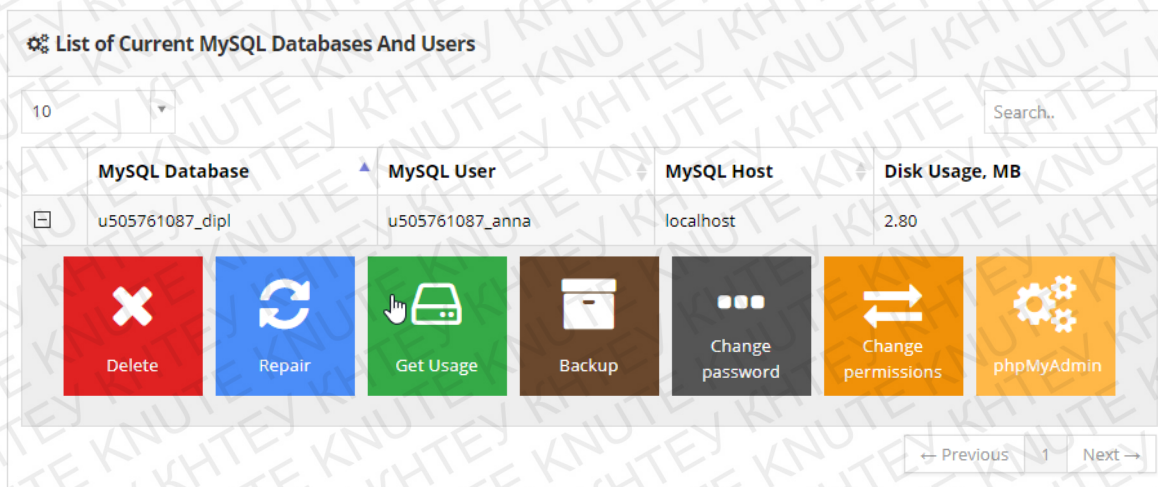
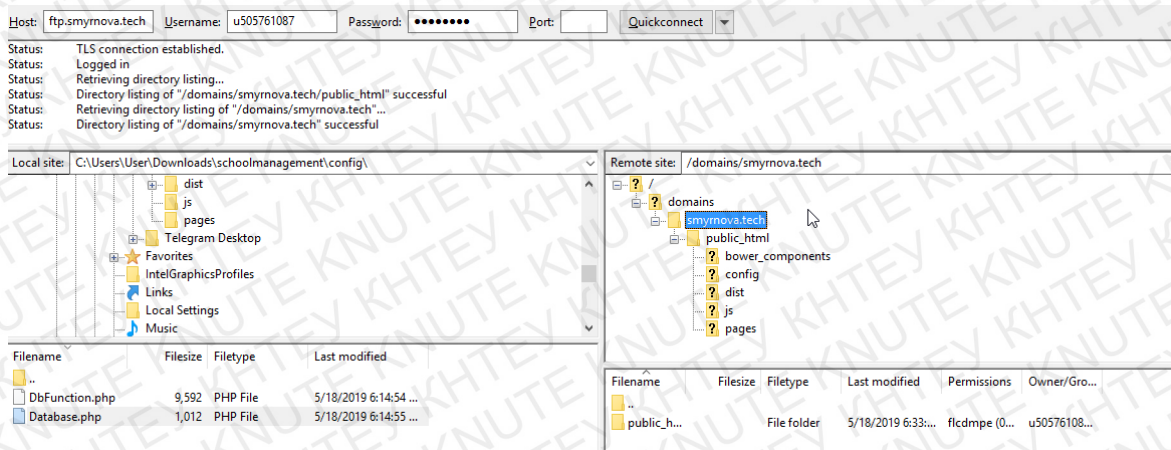


Рис. 3.3. Панель управління базою даних на сервері

Після імпорту таблиць та налаштування зв'язків у базі даних стало можливо перейти до перенесення файлів проекту на сервер. Hostinger дозволяє завантажувати файли на сервер через власний файловий менеджер що дозволить навіть вносити правки у код за потреби, але для значних обсягів редагування він не підходить та не підтримує завантаження папок та файлів у них групами. Необхідність заново створювати всі папки та завантажувати файли поодиночі спонукала перейти до окремого сервісу користування FTP зв'язку с сервером що допоміг би автоматизувати цей процес. Додаток FileZilla (рис 3.4) дозволив пришвидшити процес скопіювавши конфігурацію файлів у папці проекту на сервер без жодних змін.



3.3 Наповнення даними

Основний функціонал системи у робочому режимі, вебсайт вимагає деяких косметичних правок що можна виконувати паралельно з останнім завданням – наповненням бази даних. Використовуючи онлайн генератор даних можливо наповнити базу даних тестовою інформацією з задовільним розкидом для демонстрації та тестування системи перед фінальною зустрічю з замовником та здачі проекту як завершеного.

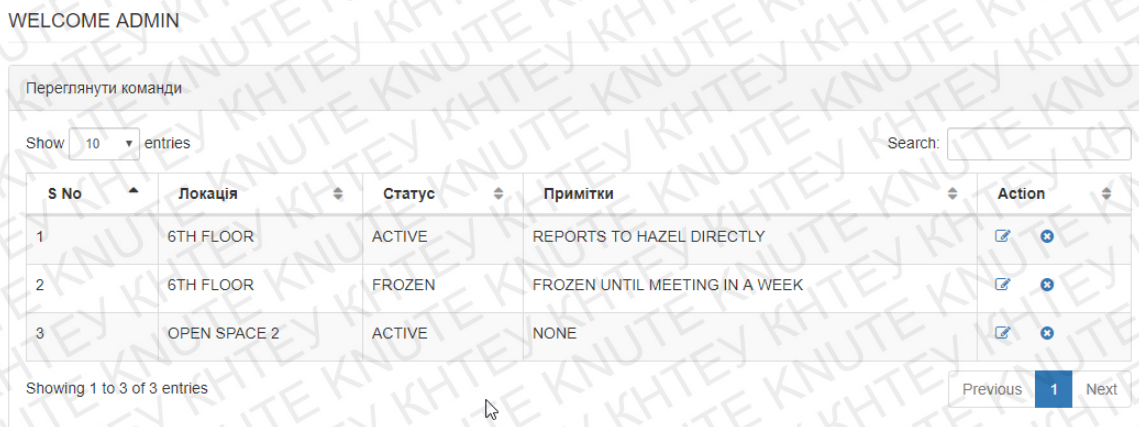


Рис. 3.5. Заповнена таблиця поточних проектів

					КНТЕУ 6.050103 07-20.БР	Аркшл
Изм.	Аркшл	№ докум	Підпис	Дат		22

3.4 Висновок до розділу 3

У процесі створення системи та розробки окремих модулів автором були вирішені багато питань що не виникали на етапі моделювання та планування проекту. Успішне створення конфігурації для локального сервера допомогло зрозуміти як вони будуються але отриманий досвід не перший раз доводить що для найоптимальнішого та найпривабливішого результату слід із самого початку використовувати сучасні здобутки техніки та використовувати хмарне сховище. Детальне технічне завдання та створені моделі допомогли притримуватися основних цілей у розробці.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		23

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Виконання робіт по проектуванню та реалізації продукту дозволили створити не тільки систему автоматизації робіт по менеджменту команд технічної підтримки а й розробити уявлення та концепцію для масштабного офісного застосунку з багатьма модулями та підтримкою інтеграції с сторонніми сервісами. Під час роботи над проектом були виконані завдання по створенню бази даних з необхідною структурою таблиць та веб-додаток для роботи з цією базою даних. Вимоги до мінімалістичного, сучасного дизайну для інтуїтивної роботи та технічні вимоги сформуvalи зовнішній вигляд додатку а внутрішня структура продиктована практиками роботи у департаменті технічної підтримки замовника. У перспективі розглядається можливість інтеграції додатку з іншими інструментами роботи в офісі , такі як програми для організації розкладу змін та оцінки роботи працівників.

					КНТЕУ 6.050103 07-20.БР	Архив
Изм.	Дикли	№ докум	Підпис	Дат		24

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергеев Александр Петрович. HTML і XML. Професійна робота = HTML и XML. Профессиональная работа. — М. : «Діалектика», 2004. — С. 880. — ISBN 5-8459-0676-8.
2. Роберт Тейбор. Реалізація XML Web-служб на платформі Microsoft .NET = Реализация XML Web-служб на платформе Microsoft .NET. — М. : «Вільямс», 2002. — С. 464. — ISBN 0-6723-2088-6.
3. Марко Беллиньясо Разработка Web-приложений в среде ASP.NET 2.0: задача — проект — решение = ASP.NET 2.0 Website Programming: Problem — Design — Solution. — М.: «Диалектика», 2007. — С. 640. — ISBN 0-7645-8464-2
4. Олищук Андрей Владимирович Разработка Web-приложений на PHP 5. Профессиональная работа. — М.: «Вильямс», 2006. — С. 352. — ISBN 5-8459-0944-9

Интернет ресурси:

1. Документація редактору коду Sublime Text [Електронний ресурс]. – [Режим доступу]: <https://www.sublimetext.com/support>
2. Basic Website Requirements for A High Converting Website In 2018 [Електронний ресурс]. – [Режим доступу]: <https://www.optimus01.co.za/basic-website-requirements/>
3. Lifecycle of a Website [Електронний ресурс]. – [Режим доступу]: <https://netlicensing.io/blog/2013/10/09/lifecycle-of-a-website/>

					КНТЕУ 6.050103 07-22.БР			
Змн.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.		Криворучко О.В			Система технічної підтримки фінансових акаунтів клієнтів банку	Стадія	Аркуш	Акрушів
Керівник		Пашорін В.І.				СД	25	25
Гарант		Цензура М.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив		Смирнова А.М.			Список використаних джерел			