

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

(повне найменування факультету)

Кафедра обчислювальної техніки

(повна назва кафедри)

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: «Програмний засіб для локалізації веб та мобільних додатків.

Частина 2. «Клієнтська частина»

Виконала: студентка 2 курсу, групи 1КІ-19м
напряму підготовки:

123 «Комп'ютерна інженерія»

(шифр і назва напряму підготовки)

Мельник Ж. А.

(прізвище та ініціали)

Керівник:

Черняк О. І.

(прізвище та ініціали)

м. Вінниця – 2020 року

АНОТАЦІЯ

У даній роботі розглянуто актуальність глобалізації та її частин: локалізації та інтернаціоналізації. Досліджено існуючі способи, методи та шляхи її реалізації на фронтенді. Обрано найбільш актуальні технології та інструменти для розробки клієнтської частини додатку. Наведено оптимальний спосіб вирішення мовної проблеми у вигляді запропонованого універсального програмного засобу для локалізації веб та мобільних додатків. Розроблено клієнтську частину додатку на основі потужного фронтенд фреймворку і таким реалізовано дружній до користувача та адаптивний інтерфейс.

ANNOTATION

This paper considers the relevance of globalization and its parts: localization and internationalization. Investigated existing ways, methods and ways of its realization on the frontend. Selected the most relevant technologies and tools for developing the client part of the application. The optimal way of solving the language problem is presented in the form of the proposed universal platform for localization of the web and mobile applications. And also, the client part of the application has been developed on the basis of a powerful frontend framework and thus a user-friendly and adaptive interface has been implemented.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ІСНУЮЧИХ СПОСОБІВ ТА ЗАСОБІВ ЛОКАЛІЗАЦІЇ	9
1.1 Глобалізація та роль локалізації при глобалізації	9
1.1.1 Поняття глобалізації	9
1.1.2 Інтернаціоналізація	9
1.1.3 Локалізація як фінальна частина глобалізації.....	10
1.2 Важливість локалізації в сучасний час	10
1.3 Типи локалізації	12
1.4 Принципи успішної локалізації	14
1.5 Етапи реалізації локалізації	16
1.6 Мовні стандарти та коди	18
1.6.1 Функція мовних кодів.....	18
1.6.2 Правила написання мовних тегів та стандарти ISO	19
1.6.3 Дерево фолбеків	20
1.7 Машинний переклад та його недоліки.....	21
1.8 Дослідження сучасних способів реалізації клієнтської частини додатку.....	23
1.8.1 Визначення мови користувача.....	23
1.8.3 Глобалізація додатків в Angular	26
2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	28
2.1 Вибір інструментів для розробки клієнтської частини додатку	28
2.1.1 Вибір мови програмування для фронтенду.....	28
2.1.2 Забезпечення авторизації користувачів	29
2.1.3 Реалізація роботи в реальному часі.....	29
2.1.4 Вибір стилю дизайну для веб-сайту.....	30
2.2 Розробка веб-інтерфейсу	30

					<i>08-23.МКР.021.00.000 ПЗ</i>			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Мельник Ж. А.</i>			<i>ПРОГРАМНИЙ ЗАСІБ ДЛЯ ЛОКАЛІЗАЦІЇ ВЕБ ТА МОБІЛЬНИХ ДОДАТКІВ. ЧАСТИНА 2. «КЛІЄНТСЬКА ЧАСТИНА» Пояснювальна записка</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>		<i>Черняк О. І.</i>					6	115
<i>Н. контр.</i>		<i>Швец С. І.</i>				<i>ВНТУ, гр. ІКІ-19м</i>		
<i>Затвердж.</i>		<i>Азаров О. Д.</i>						

2.2.1 Налаштування маршрутизації до адатку.....	30
2.2.2 Авторизація користувачів	33
2.2.3 Створення користувацького інтерфейсу	34
2.2.5 Налаштування сповіщень.....	40
3 РОБОТА ПРОГРАМИ.....	42
3.1 Керування для менеджера	43
3.2 Керування для перекладача.....	47
4 ЕКОНОМІЧНА ЧАСТИНА	50
4.1 Оцінювання комерційного потенціалу розробки	50
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторськ-технологічної роботи.....	52
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки....	56
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності.....	58
ВИСНОВКИ	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	64
ДОДАТОК А Технічне завдання	Ошибка! Закладка не определена.
ДОДАТОК Б Реалізація навігації всередині веб-додатку	Ошибка! Закладка не определена.
ДОДАТОК В Лістинг реалізації вкладок	Ошибка! Закладка не определена.
ДОДАТОК Г Лістинг реалізації вікна для входу в систему .	Ошибка! Закладка не определена.
ДОДАТОК Д Лістинг компонента «Projects»	Ошибка! Закладка не определена.
ДОДАТОК Е Приклад файлу локалізації	Ошибка! Закладка не определена.
ДОДАТОК Ж Алгоритм приєднання до команди	Ошибка! Закладка не определена.
ДОДАТОК И Схема бази даних.....	Ошибка! Закладка не определена.
ДОДАТОК К Схема роботи системи	Ошибка! Закладка не определена.

ВСТУП

Сьогодні ми живемо у глобальному світі, де кожна людина, яка прагне до пошуків інформації, не обмежена географічними рамками. Майже все, що потрібно можна знайти в мережі Інтернет у вільному доступі.

Проте, актуальність та корисність інформації не завжди відповідатиме очікуванням. Окрім наведених вище причин, бар'єром для сприйняття інформації стає мова, якою вона опублікована. На привеликий жаль, переважна меншість людей може розуміти більше ніж 2-3 мови, решті ж доводиться покидати веб-джерела один за одним, поки не знайдуть підходящу для себе мовну версію. Причиною даної ситуації є те, що користувачі всесвітньої мережі Інтернет не бажають витратити власний час на ретельний переклад того, що їм пропонується, адже вони очікують на доступ до бажаних ресурсів та інформації знайомою мовою.

Більшість компаній розуміють важливість того, щоб зробити їхні веб-сайти та додатки доступними для людей у різних країнах і тому додають різні варіанти мови на свої сайти. Проте, насправді перекладу зовсім недостатньо.

Сухий переклад контенту додатку на іншу мову, може показати розробників не в найкращому світлі. Адже можна випадково створити повідомлення, яке буде заплутувати користувачів або стане комічним для них. Часом, речення, які мають позитивний характер в одній мові, можуть стати негативними при перекладі на іншу. Через це замість сухого перекладу тексту необхідно, щоб контент був повністю локалізованим. У загальному, це означає, що кожна частина додатку буде пристосована до культури та мови цільової аудиторії.

Коли представники веб-додатків починають думати про локалізацію їх продукту, вони припускаються однієї найчастішої помилки. Першою із мов, яку обирають для даного процесу є англійська, адже вона вважається міжнародною мовою. Проте приблизно 70% користувачів мережі Інтернет не є англомовами і віддають перевагу сайтам де є їх рідна мова [1].

Основами для забезпечення вдалої локалізації є визначення цільової аудиторії та пошук гарних перекладачів. Останні мають допомогти із сухим перекладом тексту на обрані мови та подальшою його адаптацією. Очевидно, що

цей процес є достатньо дорогим та довготривалим процесом. Для економії часу та коштів, можна використати платформи, на яких уже зібрані спеціалісти-перекладачі.

Наразі існує невелика кількість платформ, які займаються локалізацію. Зокрема, такі сервіси, як Nitro, Lilt, Rev, Level Up Translation та інші. Проте, кожен з них має ряд відштовхуючих проблем, такі як: занадто висока ціна послуг, абстрагованість замовника від роботи, яка проводиться над його проектом, відсутність всіх необхідних інструментів та інше.

Мінуси, що наведені вище, роблять розробку дійсно професійного додатку, який надасть можливість замовнику спостерігати та контролювати роботу перекладачів, а також забезпечуватиме зручний та інтуїтивно зрозумілий користувацький інтерфейс, **актуальною задачею**.

Мета дослідження магістерської роботи це полегшення процесу розширення цільової аудиторії для власників веб-додатків та інших застосунків, за рахунок надання зручної платформи для локалізації їхнього інтерфейсу та представленої інформації, де тисячі перекладачів зможуть надавати свої послуги. А також забезпечення доцільного розподілу трудових ресурсів, поєднання живого та машинного перекладу, надання можливості відслідковувати роботу перекладачів та інструментів для проведення аналізу їх прогресу і розробка гнучкого та швидкого веб-додатку з інтуїтивно зрозумілим інтерфейсом для спрощення роботи та комунікації перекладачів та замовників.

Задачами дослідження магістерської роботи є:

- дослідження сучасних методів та варіантів втілення локалізації різних додатків на клієнтській частині;
- розробка user-friendly інтерфейс;
- забезпечення модульності додатку;
- реалізація авторизації користувачів;
- забезпечити сповіщення користувачів;
- налаштування маршрутизації додатку між серверною та клієнтською частинами.

Об'єкт дослідження це явище локалізації веб-додатків та інших застосунків, яке забезпечує примітивний переклад необхідного тексту на цільову мову та подальшу його адаптацію згідно з обраною країною, регіоном чи мовною групою. А також варіанти реалізації даного явища із клієнтської частини додатку.

Предметом дослідження є процес розробки клієнтської частини додатку, яка надасть зручний та адаптивний інтерфейс для забезпечення гарного користувацького досвіду, а також можливість взаємодії замовників та перекладачів у вигляді відслідковування процесу, коментування та спілкування на пряму для того, щоб індивідуально покращувати процес локалізації.

Наукова новизна отриманих результатів полягає в покращенні методу локалізації при розробці будь-яких веб та мобільних додатків за рахунок полегшеного методу авторизації для користувачів, підвищеної якості інтерфейсу, який є не складним, тому кожен клієнт може з легкістю зрозуміти принцип роботи з ним, а також за допомогою надання коректних сповіщень про зміни у системи, які стосуються користувача. Використання програмного засобу, поліпшеного даним способом, дозволяє полегшити процес локалізації.

Практичне значення одержаних результатів магістерської роботи:

- спрощено процес локалізації додатків для розробників;
- надано можливість користувачам-замовникам розширювати регіони використання їх застосунків;
- забезпечено можливість сповіщення користувачів системи;
- реалізовано клієнтську частину платформи для локалізації додатків.

Апробацією результатів магістерської роботи є те, що після проведення ряду досліджень, було написано статтю, яка висвітлює проблему локалізації веб та мобільних додатків. Дана стаття була розглянута на XLIX Науково-технічній конференції підрозділів Вінницького національного технічного університету (2020) [2].

1 АНАЛІЗ ІСНУЮЧИХ СПОСОБІВ ТА ЗАСОБІВ ЛОКАЛІЗАЦІЇ

1.1 Глобалізація та роль локалізації при глобалізації

1.1.1 Поняття глобалізації

Глобалізація означає таке проектування і реалізацію систем, при яких один екземпляр програмного забезпечення, може надавати культурно вірну інформацію і приймати багатомовні дані для обробки. Для того, щоб вважатися культурно вірним, даний додаток чи веб-сайт повинен не тільки надавати інформацію на відповідній мові, а й підтримувати особливості та риси різних країн та національностей. Наприклад, сортування списків повинне відповідати стандартному для даної місцевості порядку, дати мають бути відформатовані відповідно до національних стандартів, а числа — розділятися загальноприйнятими знаками.

Інакше кажучи, глобалізація є об'єднання технічної та перекладацької частин. Таким чином, її поділяють на дві складові: локалізацію та інтернаціоналізацію (див. рис. 1.1).



Рисунок 1.1 — Процес глобалізації

1.1.2 Інтернаціоналізація

Інтернаціоналізація — це процес проектування та розвитку такого додатку, який повністю абстрагований від особливостей мови, культури чи регіону ринків, на які він націлений. Часто дане поняття пишеться як $i18n$, де 18 — це кількість

букв між «i» та «n» в англійському варіанті слова.

Інтернаціоналізація звичайно включає:

- створення та розробку додатку таким, який не мав би перешкод при використанні на міжнародному ринку, а саме уникання залежності коду від значень елементів інтерфейсу чи контенту для користувача;
- додавання до розмітки та стилів основ для підтримки двостороннього тексту, вертикального тексту або нелатинських друкарських властивостей;
- забезпечення можливості модульно додавати нові мови за потреби.

1.1.3 Локалізація як фінальна частина глобалізації

Локалізація — це процес адаптації вмісту додатку, програми або веб-сайту до мовних, культурних та інших вимог певного цільового ринку. Як і інтернаціоналізація, локалізації має скорочений варіант написання, а саме l10n, де 10 це кількість букв між літерами «l» та «n».

Часом поняття локалізації вживається як синонім перекладу, але даний термін є значно ширшим. Воно може включати адаптацію наступних аспектів:

- формати чисел, дати;
- грошові одиниці;
- використання клавіатури;
- символи, знаки та кольори;
- графічних зображень;
- різні правові чи релігійні вимоги;
- та багато інших.

Тобто, інтернаціоналізація готує фундамент, який значно полегшує локалізацію. А вона, у свою чергу, є головним та фінальним етапом усього процесу глобалізації. [3]

1.2 Важливість локалізації в сучасний час

Важливість процесу локалізації додатків можна визначити оглянувши п'ять найвагоміших причин.

Першою є можливість захоплення ширшої аудиторії. Локалізація полегшує вихід застосунку на нові ринки та підвищує їхню впізнаваність за кордонами країни, на яку спершу був націлений додаток. Також вона дозволяє надати такий рівень довіри, який допомагає перетворити потенційних клієнтів на постійних.

Другою причиною можна назвати бажання користувачів отримувати персоналізований контент. Він дуже важливий для кожного відвідувача, і саме цим займається локалізація — робить контент найбільш близьким до користувача. Таким чином, використання локалізації збільшує швидкість зростання числа аудиторії та відвідувачів.

Третя причина — це відсутність ризику. У різних країнах та культурах тварини, рослини та кольори можуть мати різний сенс та викликати різні асоціації, а це безпосередньо впливає на сприйняття веб-сайту або інтерфейсу програмного забезпечення. Також важливу роль відіграють релігійні та політичні моменти в текстах, зображеннях або символах, які можуть стати причиною заборони використання застосунку на території тієї чи іншої країни. Тому уникнення подібних нюансів має стати для замовника та розробника першим пріоритетом, а саме локалізація допоможе добитись відсутності подібного ризику.

Четвертою причиною є збільшення кількості користувачів за рахунок покращення користувацького досвіду. Локалізуючи документацію, дописи в блозі, поширені запитання, електронні книги, путівники, звіти та підтримку клієнтів, додаток зможе гарантувати, що задовольнятиме конкретні потреби кожного користувача. А позитивно налаштовані клієнти будуть залишати відгуки та поширюватимуть інформацію про даний застосунок. Вони будуть тими, хто створює та підтримує певну спільноту та тими, хто забезпечуватиме поширюваність додатку, адже найкращий маркетинг — це наявність реальних користувачів, які скажуть про нього в Інтернеті та на будь-яких інших ринках.

І останньою, п'ятою причиною є те, що локалізація необхідна для глобального розвитку. Компанії, які вийшли на міжнародний ринок, можуть із впевненістю сказати, що локалізація є ключем до міжнародного успіху. Проте варто вкотре зазначити, що локалізація не є бездоганним варіантом для охоплення

ширшої аудиторії, адже потрібні час, гроші та дослідження, щоб налаштувати свій бізнес на глобальне розширення. Але якщо почати правильно планувати локалізацію заздалегідь, включити етап інтернаціоналізації в розробку додатку, то в майбутньому етап локалізації буде в рази простішим [4-5].

1.3 Типи локалізації

Враховуючи те, що процес локалізації є дорогим та достатньо складним, розробники додатків та веб-сайтів реалізують його не в повній мірі. Отже, виділяють декілька видів локалізації за рівнем поглибленості: локалізація інтерфейсу, текстова, графічна або звукова локалізація. Проте, необхідно зазначити, що простіша локалізація означає менша розповсюдженість додатку.

Гарним прикладом для висвітлення даних видів локалізації можуть бути комп'ютерні ігри. Найбільш поверхневим є перший зазначений тип локалізації, а саме локалізація інтерфейсу. Із назви даного типу зрозуміло, що користувацький інтерфейс, із яким взаємодіє відвідувач веб-сайту, буде лише перекладено (кнопки, вказівки, поля для введення і подібні елементи).

Зазвичай відеоігри мають сюжет, який часто представлений у текстовому форматі. Якщо вони є адаптованими, то доречно стверджувати, що при процесі глобалізації застосовували текстову локалізацію (яка містить в собі локалізацію інтерфейсу). Переклад тексту на картинках називається графічною локалізацією, яка вміщає також два попередні види.

Незалежно від локалізації інтерфейсу, текстової та графічної локалізації існує звукова. Так як відеоігри практично не можуть існувати без розповідей, монологів чи діалогів, часом розробники вдаються до локалізації звукового супроводу.

Найбільш глибоким видом локалізації є культурна адаптація. Це приведення всього матеріалу до відповідності з традиціями, ментальністю, моральними нормами і засадами певної місцевості, країни або регіону з метою підвищення рівня його сприйнятливості кінцевими користувачами. Перекладач повинен досконало володіти цільовою мовою, а також темою, конкретними технічними термінами, ринком, цільовою аудиторією тощо. Локалізація такого типу використовується

достатньо рідко, адже часто однакові сторінки сайту, відображені різними мовами, дуже сильно відрізняються, що додає зайвих проблем. Але саме такі продукти є найбільш затребуваними.

При культурній адаптації враховується безліч факторів. Наприклад, в різних країнах та культурах тварини, рослини та кольори можуть мати різний сенс та викликати різні асоціації, що безпосередньо впливає на сприйняття веб-сайту або інтерфейсу програмного забезпечення. Важливу роль відіграє уникання релігійних та політичних моментів у текстах, зображеннях або символах, адже це може стати причиною заборони використання застосунку на території тієї чи іншої країни. Проте, найбільш складною ланкою у культурній адаптації є будь-яке зображення людини, так як виникають проблеми з інтерпретацією жестів, етносом. Саме тому розробники інтерфейсів намагаються уникати подібних зображень і єдиним прийнятним варіантом є умовна фігура без одягу, пальців на руках та волосся [6].

Культурна адаптація відеоігор може викликати підстави для повної обробки персонажів та сюжету, так як, наприклад, гра по китайській міфології буде не ясною для користувачів із Європи чи Америки. Через це потрібно детально продумувати кожного персонажа у відповідності до країни, у якій дана відеогра буде поширюватись. Проте, звертати увагу варто не лише на головних героїв, а й на другорядних, щоб не втратити увесь ефект від проведеної культурної адаптації. Наприклад, на рисунку 1.2 зображені різні офіціанти відповідно для західної та східної культури.



Рисунок 1.2 — Приклад локалізації героїв

1.4 Принципи успішної локалізації

Для будь-якої справи, ціль якої бути популярною по всьому світу, визначним аспектом є глобальна локалізація контенту. Існує шість настанов для створення вдалої локалізації, які можна визначити переглянувши статті із блогу перекладів та локалізації («Translation & Localization Blog»).

Першою настановою є заклик прохання подумати про "різноманіття" кожної культури. Розробляючи свій план для певної місцевості, потрібно впевнитись, що враховуються особливості, наприклад, як користувачі обирають і купують товари, а також як вони розуміють колір або цифри.

До прикладу виробник м'ячів для гри у гольф вирішив скласти їх в коробки по чотири штуки в Японії. У східній культурі (як у японській так і в китайській) люди вірять в особливе значення цифр навколо себе. Певні числа прийнято вважати негативними тому, що їх вимова схожа з вимовою певних страшних слів, тому в побуті їх не використовують. Однією із найбільш поганих цифр вважається «4», що означає чотири м'ячів в одному наборі — погана ідея.

Друга порада — дотримуватись культурної чутливості.

Слід тісно співпрацювати з тими, хто має глибокі знання культури, в якій перебуває цільова аудиторія. Культурні нюанси мають великий вплив на те, як варто зацікавлювати певну аудиторію. Потрібно переконатись, що не використовуються зображення, словниковий запас або кольори, які можуть образити людей. Наприклад, у Великобританії білий колір символізує такі речі, як елегантність та чистота, а у Китаї та інших країнах Азії білий колір насправді означає нещастя.

Третя порада зазначає, що глобальна локалізація контенту — це розуміння особливостей місцевих споживачів. Найбільш популярною думкою є така, що всі запити споживачів ідентичні у всьому світі. Але це зовсім не так, наприклад, у Англії користувачі надають перевагу оплаті своїх замовлень через Інтернет за допомогою банківської картки, а у Японії багато користувачів досі вважають, що найкращий спосіб оплати — готівка.

Саме тому потрібно приділити достатньо часу дослідженню того, якими є

потреби та очікування цільової аудиторії, і не покладатись на жодні припущення. Таким чином мережа однотипних універсальних магазинів Walmart помилилась припустивши, що найвдалішим місцем розташування для їх точок, будуть місця поблизу індустріальних парків у Китаї, не врахувавши те, що споживачі там воліють робити покупки біля свого будинку, а не біля роботи.

Четвертою порадою є створення всесвітніх торгових марок. Якщо обрати таку, яку не доведеться перекладати, то можна створити репутацію міжнародного бренду. А якщо все-таки вибрати назву, яке не потребуватиме перекладу, можна витратити багато часу та ресурсів, оскільки потрібно буде вигадати нові торгові марки для поширення по всьому світу. Таким чином, при виборі назви слід зважити всі позитивні та негативні наслідки, які можуть виникнути у результаті його використання.

Створення послідовного, якісного контенту є п'ятою порадою.

Створюючи вміст, намагайтеся не використовувати занадто багато сленгу чи ідіом. Якщо їх уникнути, контент стане легшим для перекладу, і буде менше шансів зіткнутися з проблемами згодом. Створіть деякі редакційні вказівки та переконайтеся, що автори вмісту дотримуються їх у будь-який час. Також варто інвестувати в хороших перекладачів, щоб можна було бути повністю впевненим, що кожному слову у тексті контенту приділяється необхідна увага фахівця [7].

Останньою, шостою, порадою є заклик не змушувати користувача задумуватись. Для кожної людини найкраще сприймається матеріал написаний його рідною мовою. Якщо користувач знаходить потрібну йому інформацію, але вона написана іншою мовою, у більшості випадків простіше ще трохи зайнятись пошуками, аніж читати іноземний текст, тому що це змушує клієнта задумуватись і витратити на опрацювання матеріалу більше часу ніж йому хотілося б. Тому гарна локалізація це та, яка стане найбільш близькою для будь-якого користувача і та, яка дозволить йому не займатись подальшими пошуками або витратити час на самостійний переклад окремих слів, розбір сталих виразів, які можуть не вдало перекладатись на різні мови [8].

Глобальна локалізація вмісту, яка виділяється, є важливою для будь-якого

бізнесу, який прагне підкорити міжнародну аудиторію в Інтернеті. Створення та локалізація послідовного, якісного контенту, який враховує відмінності між культурами, поставить додаток, веб-сайт, відеогру чи будь-який інший продукт на шлях до успішного міжнародного ринку.

1.5 Етапи реалізації локалізації

Доцільно виокремити наступні п'ять етапів для успішної імплементації локалізації додатку, застосунку чи веб-сайту:

Першим етапом можна назвати вибір цільової аудиторії, тобто визначення, у яких географічних регіонах буде використовуватись додаток. Влучним способом для реалізації подібного вибору є огляд та розбір аналітики схожих між собою веб-сайтів чи застосунків. Необхідно виділити всі великі групи користувачів або відвідувачів, які проживають у певній місцевості, для того щоб в подальшому застосовувати локалізацію для неї.

Потім варто розпочати дослідження про даний ринок. Необхідно пам'ятати те, що якщо застосунок успішно працює для однієї аудиторії, то це не обов'язково означає, що він буде таким для іншої. Тому доцільно витратити певний час на дослідження. Але варто відмітити, що додатку, можливо, потрібен буде час на пристосування для встановлення зв'язку з новою аудиторією.

Наступним етапом є впровадження технічної підготовки додатку. Необхідно дослідити всі технічні особливості між базовою мовою застосунку та цільовим ринком — слід приділити достатньо уваги правилам плюралізації, цифрами, даті, часу та валюті. Варто також звертати увагу на нюанси цифрових сервісів, наприклад, платіжні системи.

Прийнято думати, що весь світ читає текст або цифри так само, як і ми. Насправді світ — це дуже різноманітне місце. Саме тому варто працювати із спеціалістом з локалізації, які повинні консультувати щодо будь-яких технічних відмінностей, які слід враховувати. Даний етап доцільно поєднати з інтернаціоналізацією додатку.

Найцікавішим етапом є третій, а саме переклад контенту з мови оригіналу на

мову для кожної обраної країни. Але даний етап не настільки простий, як може здатись спочатку.

Наприклад, якщо цільовою аудиторією є Мексика, то можна припустити, що вся іспанська мова не має різниці, але це зовсім не так, оскільки це дуже різноманітна мова з багатьма регіональними діалектами. Тобто мексиканець, звісно, може зрозуміти іспанську мову з Іспанії, оскільки базові речі у більшості випадків ідентичні та не завдадуть труднощів. Але подібний зміст стане викликом для користувача, оскільки йому необхідно буде самостійно перекладати певні вирази чи словосполучення. Найліпшим виходом із такої ситуації є найм перекладача родом із Мексики, який би переклав контент на потрібний діалект.

Четвертим етапом є перевірка контенту на місцевому ринку. Адже зважаючи на те, що перекладачем може бути місцева людина, навіть вона може не помітити всіх нюансів, які інші люди можуть випадково не так зрозуміти. Наприклад в 1987 році авіакомпанія Braniff Airlines потрапила в біду, коли хотіла поширити свої нові шкіряні сидіння на південь з тією ж кампанією, яку застосовували в США: «Літати в шкірі» («Fly in leather»). Хоча переклад іспанською мовою «Vuela en cuero» був доречним у більшій частині Латинської Америки, він мав різні відтінки в Мексиці, де цей вислів також означає «літати голим». Подібне гасло стало зовсім не тим, яке очікувала поширити авіакомпанія [9].

На щастя, сьогодні власники різноманітних застосунків мають можливість поширити локалізовані версії їхніх додатків обраній частині користувачів, для того, щоб в подальшому отримати від них відгуки та зможти проаналізувати чи вартий такий спосіб локалізації право на життя для даної аудиторії.

Останнім, п'ятим етапом є звернення до служби підтримки клієнтів. Неминучим наслідком локалізації є те, що ви почнете отримувати електронні листи та запити підтримки від клієнтів, які не говорять вашою мовою. Тому, звичайно, якщо застосунок хоче досягти успіху в підтримці клієнтів, потрібна стратегія підтримки, яка працює як для місцевих, так і для міжнародних клієнтів.

Одним із варіантів є часткове введення користувачів на даний ринок, для

допомоги із обробкою звернень та запитань. У випадку непрактичності або недоступності такого варіанту, можна розробити сторінку спеціального вмісту мовами відвідувачів, яка буде містити відповіді поширені запитання.

Варто часом переглядати статистику своїх клієнтів, дізнатись до чого вони намагаються отримати доступ, про що вони зазвичай запитують та пишуть. Потім потрібно дати детальні відповіді на будь-які поширені запити в документі підтримки, а також надати професійний переклад [10].

1.6 Мовні стандарти та коди

1.6.1 Функція мовних кодів

Велика кількість різних мов постійно використовувалась людьми у будь-який час. Це є основною причиною, чому необхідно ідентифікувати мову при надсиланні чи запиті інформації [11].

Мова інформаційного елемента або мовні налаштування користувача часто потрібно визначати, щоб у подальшому застосувати відповідну обробку. Наприклад, мовні налаштування користувача у веб-браузері можуть бути використані для відповідного вибору веб-сторінок. Інформацію про мову також можна використовувати для вибору серед інструментів (наприклад, словників), які допомагають обробляти або розуміти вміст різними мовами. Знання про конкретну мову, що використовується деяким інформаційним вмістом, може бути корисним або навіть потрібним для деяких типів обробки, наприклад, перевірки орфографії, синтезованого комп'ютером мовлення, транскрипції Брайля або якісного друку.

На сьогоднішній день, способом позначення мови є відмічення тексту ідентифікатором («тегом»). Їх також доцільно використовувати для пояснення уподобань клієнта при виборі того, який контент показувати, або для позначення опціональних атрибутів тексту. Як приклад можна навести, маркування визначеної інформації про діалект, варіант написання або орфографію, яка вживається на веб-сайті або ресурсі та має можливість дозволити відвідувачу отримати контент мовою, яка є зрозумілою для нього.

Узагальнивши, можна сказати що мовні теги використовуються, щоб

допомогти ідентифікувати мови (розмовні, письмові). Сюди входять різні всі існуючі мови, включаючи штучні, але не враховуючи ті, які не призначені для безпосереднього спілкування людей, а саме мови програмування.

1.6.2 Правила написання мовних тегів та стандарти ISO

Мовний тег містить в собі один або послідовності кількох умовних "підтегів", ціль яких полягає в уточненні або звуженні варіанту діапазону мови, який окреслений загальним тегом. Підтеги — це набір символів (літер або цифр), які розділяються за допомогою знаку «дефіс».

Існують різні типи підтегів, кожен з яких відрізняється довжиною, положенням у тезі та вмістом: тип кожного підтегу можна розпізнати виключно за цими ознаками. Також існують правила щодо використання великих літер у деяких підтегах, проте вони не повинні сприйматись як такі, що мають значення.

Тобто, мовний теги «mn-Cyrl-MN» та «MN-cyRL-mn» не мають різниці по сенсу, так само як і будь-яка інша варіація (наприклад, «mN-cYrL-Mn»), а також кожна з цих комбінацій має однакове значення, а саме: монгольська написана кирилицею, що використовується в Монголії.

Хоча відмінності між регістрами не мають значення в мовних тегах, їх послідовне форматування та подання допомагають у кращому розумінні. Таким чином Міжнародна організація зі стандартизації (англ. International Organization for Standardization, ISO), завданням діяльності якої є сприяння розвитку стандартизації і суміжних видів діяльності у світі з метою забезпечення міжнародного обміну товарами і послугами, а також розвиток співробітництва в інтелектуальній, науково-технічній і економічній галузях, розробила кілька наборів стандартів:

— ISO 639 — пов'язаний зі стандартизацією назв мов і мовних груп та складається з шести частин;

— ISO 15924 — стандарт для позначення назв письменностей. Визначає два набори кодів для ряду писемностей. Кожній писемності присвоюється два коди - числовий і літерний (містить чотири букви);

— ISO 3166-1 — частина стандарту ISO 3166, що містить коди назв країн і

підлеглих територій. Визначає три різних коди для кожної країни: двохлітерний, трьохлітерний та числовий. Для України це коди UA, UKR та 804 відповідно;

— та інші.

Дані стандарти включають багато різноманітних рекомендацій для написання мовних тегів або підтегів, наприклад:

— ISO639-1 рекомендує писати коди мов малими літерами («uk» — українська, «mn» — монгольська);

— ISO15924 рекомендує, щоб коди писемностей були написані малими літерами, що починаються з великої літери, тобто «Cyr1» — кирилиця, «Latn» — латиниця, «Phag» — монгольське квадратне письмо;

— ISO3166-1 рекомендує, щоб коди країн були написані великими літерами («UA» — Україна, «MN» — Монголія).

Таким чином можна сказати, що мовний ідентифікатор «mn-Cyr1-MN» є вірним за вказаними стандартами та сприйматиметься найкраще.

1.6.3 Дерево фолбеків

Частим місцем для зустрічі мовних ідентифікаторів є URL адреси веб-сайтів, як приклад: «uk.worldwidedictionary.org», «www.microsoft.com/uk-ua». Тут вони показують, якою мовою написаний контент веб-сторінки. Це дає змогу ще при пошуку потрібної інформації обирати найбільш зручну для розуміння статтю, звертаючи увагу на мовні ідентифікатори запропонованих варіантів.

Проте розробники чи власники веб-сторінок, не завжди можуть задовольнити повну потребу користувача стосовно зручності йому мови, тому будуть пропонувати варіанти свого контенту найбільш близькою для запрошеної мови. Для того щоб знайти найбільш такі мови застосовуються дерева фолбеків (див. рис. 1.3).

У них мови впорядковуються від найбільш сукупного варіанту застосування мови, до найвужчих її видів. Зважаючи на це, якщо власник веб-сайту немає варіантів тексту, який локалізований на китайську мову лише із спрощеною писемністю, що використовується виключно на території Сінгапуру, то він буде пропонувати відвідувачу споріднену, наприклад, спрощену писемність китайської

мови без розподілу на території.

1.7 Машинний переклад та його недоліки

Перед тим як локалізувати додаток або веб-сайт, перед розробником постає вибір яким чином отримати локалізований контент: за допомогою людей, які надають послуги перекладачів, або за допомогою машинного перекладу.

Недоліком першого варіанту є його вартість. Адже один спеціаліст не зможе надати вдалий та коректний переклад на всі запрошені розробником мови. Таким чином доведеться користуватись послугами декількох або навіть і декількома десятками перекладачів, все залежить від того на яку кількість країн та мов має поширюватись додаток. Саме це робить даний варіант дорогавартісним.

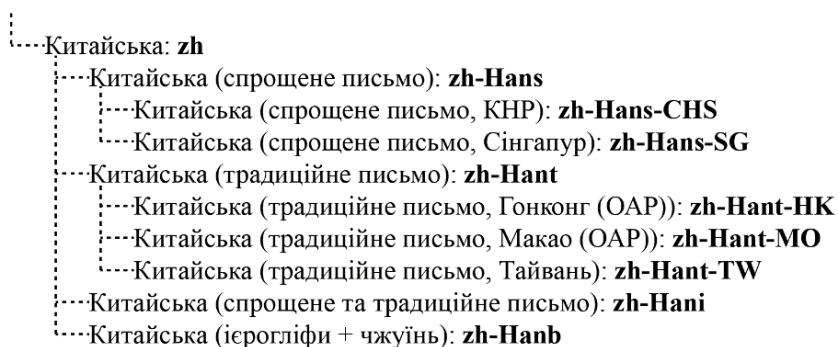


Рисунок 1.3 — Приклад дерева фолбеків

Другий варіант — машинний переклад (Machine translation, MT) — це процес перекладу текстів з однієї мови на іншу за допомогою спеціальної комп'ютерної програми. Зазвичай дані програми поділяють на:

— загальні (до них, зазвичай, відносяться такі платформи, як Google Translate, Bing та Naver, які надають MT для будь-яких можливих перекладів мільйонам людей);

— з можливістю налаштування;

— адаптивні, які поєднують в собі частини загальних платформ та здатність до навчання, як у програм з можливістю налаштування [12].

Програми з можливістю налаштування — це платформи, які мають базові набори словників та правил, і які можна навчити покращувати точність

термінології у вибраному сегменті (медична, юридична, технічна або власна вподобана термінологія). Наприклад, спеціалізований машинний переклад розроблений для Всесвітньої організації інтелектуальної власності (ВОІВ) перекладає патенти точніше, ніж загальні програми МТ, а рішення для eBay (американська інтернет-компанія) може зрозуміти та передати на інші мови сотні скорочень, що використовуються в електронній комерції;

Існує три основних підходи машинного перекладу:

- на основі правил;
- статистичний;
- нейронний.

Підхід на основі правил заснований на незліченних вбудованих лінгвістичних правилах та мільйонах двомовних словників для кожної мовної пари. Таке програмне забезпечення аналізує текст і створює перехідне подання, з якого генерується текст цільовою мовою. Для цього процесу потрібні великі словники з морфологічною, синтаксичною та семантичною інформацією та великі набори правил. Програмне забезпечення використовує ці складні правила, а потім трансформує граматичну структуру вихідної мови на цільову мову;

Статистичний використовує моделі статистичного перекладу, параметри яких впливають із аналізу одномовних та двомовних корпусів. Мовний корпус [13] — це великий, уніфікований, структурований, розмічений та філологічно компетентний масив мовних даних, який представлений в електронному вигляді.

Побудова статистичних моделей перекладу є швидким процесом, але технологія значною мірою спирається на існуючі багатомовні корпуси. Потрібно мінімум 2 мільйони слів для певного сегменту і навіть більше для загальної мови. Теоретично можна досягти певного варіанту гарної якості, але більшість компаній не має такої великої кількості існуючих багатомовних корпусів для побудови необхідних моделей перекладу. Крім того, статистичний машинний переклад є достатньо енергозатратним, тому що вимагає великої кількості процесорів і розширеної конфігурації обладнання для запуску моделей перекладу навіть для середнього рівня продуктивності.

А нейронні МТ (NMT) використовують технологію машинного навчання, щоб навчити програмне забезпечення досягти найкращого результату. Проте очевидним мінусом, як і у попередньому варіанті є енергозатратність, адже цей процес споживає велику обчислювальну потужність.

На сьогоднішній день найбільшою популярністю користуються технології машинного перекладу на основі правил та статистичні, кожна з яких має свої переваги та недоліки (див. табл. 1.1) [14].

Таблиця 1.1 — Порівняльна характеристика підходів машинного перекладу.

Критерії порівняння	МТ на основі правил	Статистична МТ
Якість перекладу неспеціалізованих текстів	Високоякісний	Низькоякісний
Якість перекладу текстів обраної сфери	Послідовна та передбачувана	Непередбачування
Продуктивність	Висока продуктивність та міцність	Високі вимоги до центрального процесора та дискового простору
Стабільність роботи	Відсутність стабільності	Хороша стабільність
Обробка граматичних правил	Присутня	Відсутня
Обробка винятків із правил	Погано справляється з обробкою винятків із правил	Добре відловлює винятки із правил

В підсумку можна сказати, що машинний переклад також не є оптимальним варіантом у виборі інструменту для отримання локалізованого контенту.

1.8 Дослідження сучасних способів реалізації клієнтської частини додатку

1.8.1 Визначення мови користувача

Для того, щоб локалізація додатку чи веб-сайту була доречною та корисною для користувача, перше, що потрібно зробити — це визначити, якій мові віддає

перевагу користувач. Іншими словами, потрібно дізнатись, який мовний ідентифікатор є найбільш зручним для нього.

Із клієнтської сторони додатку є кілька варіантів як можливо дізнатись дану інформацію:

- використання GeoIP чи GPS;
- збір інформації з веб-браузера;
- дослідження заголовку HTTP-запиту.

GeoIP та GPS — це форми геолокації, які визначають географічне місцезнаходження комп'ютера, телефону чи іншого подібного пристрою шляхом ідентифікації IP-адреси. GeoIP не настільки точний, як інші методи геолокації, зокрема GPS, але і GPS не забезпечує ідеальної точності. Крім того визначення геолокації не дасть коректної інформації про зручну для користувача мову, якщо він знаходиться в іншій країні. [15]

Зібравши інформацію з налаштувань веб-браузера можна виявити мовні переваги користувача, проте потрібно враховувати відмінності роботи різних браузерів, наприклад:

```
const userLanguage1 = navigator.browserLanguage; // для Internet Explorer
const userLanguage2 = navigator.language; // для більшості інших браузерів
```

Мінусом даного варіанту є те, що отримуємо інформацію лише про одну, основну мову інтерфейсу браузера. А наразі більшість браузерів підтримують обробку кількох мов, які налаштовуються як вподобані (див. рис. 1.4).

Останнім способом є дослідження інформації у заголовку HTTP-запиту. Поле `Accept-Language` (див. рис. 1.5), на відміну від попереднього варіанту, містить в собі інформацію про всі мови, які може розуміти користувач та які можуть бути достатньо зручним для нього [16].

1.8.2 Огляд мови програмування TypeScript та фреймворків

JavaScript — це мова скриптів, яка використовується для створення та управління динамічним вмістом веб-додатку, не вимагаючи перезавантаження веб-

сторінки вручну. TypeScript розширює JavaScript і пропонує більше контролю над кодом за допомогою типу, інтерфейсів та класів, а потім компілюється в JS.

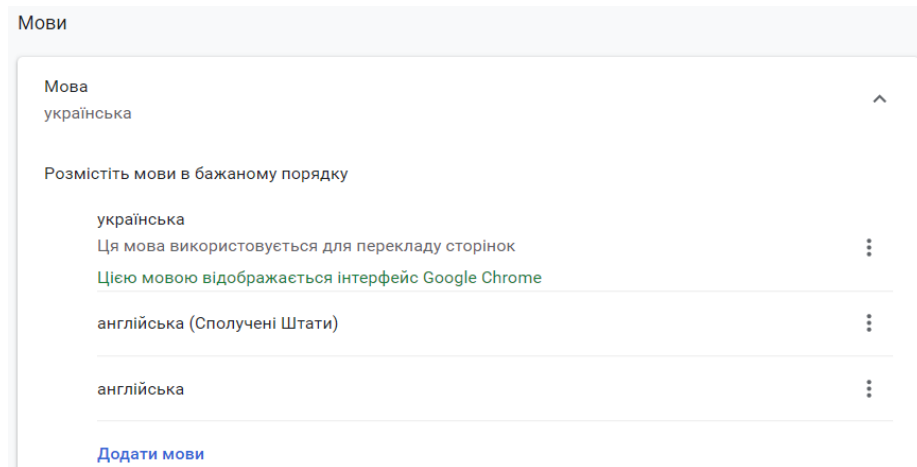


Рисунок 1.4 — Налаштування підтримуваних мов

```
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
```

Рисунок 1.5 — Приклад заголовку HTTP-запиту

Наразі майже неможливо створити веб-додаток не використовуючи JavaScript, завдяки ряду його переваг:

- швидкість, оскільки він часто запускається відразу в браузері клієнта (на сьогоднішній день, всі основні браузери підтримують компіляцію JIT, а це означає, що немає необхідності компілювати код перед його запуском);

- простота, так як синтаксис JS достатньо легкий, що дає можливість швидко зрозуміти його;

- сумісність (на відміну від PHP чи інших скриптових мов, JavaScript можна додати до будь-якої веб-сторінки);

- підтримка та оновлення, адже з моменту появи ECMAScript 5 (стандарт мов програмування, який використовується в якості основи для побудови

скриптових мов, такі як JS), компанія ECMA International займається оновленням JavaScript щорічно;

— розширення функціональності та інтерфейсів веб-застосунків [17].

Для того, щоб зробити використання JS простішим і більш плавним можна використовувати фреймворки. Уже кілька років поспіль три перші сходинки рейтингу найкращих фреймворків займають Angular, React та Vue.js, кожен з яких має свої переваги та недоліки.

Angular є одним із найпотужніших і найбільш ефективних фреймворків, розроблених компанією Google. Серед його особливостей можна назвати наступне:

— крос-платформеність, яка забезпечує можливість розробки не тільки веб-додатків, а і десктопних застосунків для Mac, Windows та Linux за допомогою таких же методів;

— універсальність, яка дозволяє створювати клієнтські частини для застосунків, серверна частина яких створена за допомогою Node.js, .NET, PHP;

— розбиття коду на компоненти, що створює можливість перевикористовувати розроблені модулі, покращує читабельність коду та дозволяє завантажувати лише ту компоненту, яка на даний момент потрібна для відтворення запиту;

— покращена взаємодія з TypeScript;

— високу ефективність роботи;

— доступність;

— а також Angular CLI (Command Line Interface — Інтерфейс командного рядка) [18].

Наведені переваги роблять вибір фреймворку Angular вдалим для розробки клієнтської частини додатку.

1.8.3 Глобалізація додатків в Angular

Так як Angular є сучасним фреймворком, який постійно оновлюється, він підтримує процес глобалізації. Для першого етапу (інтернаціоналізації) він пропонує:

— вбудовані методи для відображення дат, чисел, відсотків та валют у локальному форматі.

— позначення тексту у шаблонах компонентів для перекладу;

— позначення альтернативного тексту для перекладу.

Для форматування даних на основі обраної культури користувача, даний фреймворк надає вбудовані pipes («труби»), такі як:

— DatePipe, який форматує значення дати;

— CurrencyPipe, який перетворює число у валютний рядок;

— PercentPipe, який перетворює число у відсотки.

Для того щоб скористатись перевагами функцій локалізації (другого етапу глобалізації) в Angular спочатку потрібно додати пакет `@angular/localize` до проекту, використовуючи Angular CLI:

```
ng add @angular/localize
```

Наступним кроком є позначення полів у шаблонах компонентів для перекладу за допомогою атрибута `i18n`. Його потрібно помістити на кожен тег елемента з фіксованим текстом, який потрібно перекласти [19]:

```
<h1 i18n>Hello i18n!</h1>
```

2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір інструментів для розробки клієнтської частини додатку

2.1.1 Вибір мови програмування для фронтенду

Для розробки зовнішнього вигляду будь-якої клієнтської частини веб-додатку неможливо обійтись без HTML, CSS.

Hypertext Markup Language (HTML, мова розмітки гіпертексту) — це стандартна мова розмітки для документів, призначених для відображення у веб-браузері. Останньою, і поки найкращою її версією є HTML5, серед переваг якої можна виділити:

- покращена та чистіша розмітка, що дозволяє писати більш охайний та зрозумілий код;
- підтримка мультимедійних елементів, що означає вбудовану здатність відтворювати аудіо та відео;
- офлайн-кеш додатків, що дозволяє користувачу знову зайти на веб-сайт, який відвідував раніше, не зважаючи на те, що він не підключений до мережі Інтернет.

Cascading Style Sheets (CSS, каскадні таблиці стилів) використовується для визначення стилю додатку, адже він взаємодіє з HTML та привносить визначений стиль до кожного його тегу. Для розширення можливостей CSS були розроблені так звані препроцесори — це програми, які мають власний синтаксис та більший функціонал (що спрощує роботу розробника) і в результаті самостійно генерують код CSS.

На сьогоднішній день найпотужнішим препроцесором вважається Sass. Часом його називають продовженням CSS, через схожість розмітки і має наступний ряд переваг, таких як:

- повна сумісність з CSS, що вберігає від створення помилок при компіляції;
- підтримує змінні, вкладеності та змішування;
- надає безліч корисних функцій для роботи з кольорами та іншими значеннями;
- сумісний з усіма версіями CSS, що надає можливість використовувати

будь-які доступні бібліотеки CSS;

Також як було сказано в підрозділі 1.8.2, жоден веб-сайт практично не може існувати без використання JavaScript (або TypeScript), тому що саме вони додають йому інтерактивності. А використання фреймворку Angular, який описано у тому ж підрозділі, полегшить роботу над користувацькою частиною додатку та спростить взаємодію клієнтської та серверної частини за рахунок вбудованого поєднання з .NET.

2.1.2 Забезпечення авторизації користувачів

Так як платформою передбачено два види користувачів (перекладачі та менеджери) варто забезпечити розмежування їх прав. Для цього потрібно використовувати аутентифікацію для усіх користувачів, які при реєстрації в систему обирають собі відповідну роль.

На сьогоднішній день одну з провідних позицій на ринку BaaS (Backend-as-a-service) посідає продукт Google — Firebase. Це платформа для розробки програмного забезпечення, яка заснована як база даних у режимі реального часу і тепер має 18 служб та спеціальні API [20].

Однією із можливостей Firebase є користувацька аутентифікація. Він дбає про безпечну авторизацію та аутентифікацію, ідентифікує користувачів, які входять до системи та виходять з неї. Також Firebase надає різні методи для входу в систему за допомогою акаунтів Google, Facebook, Twitter, Microsoft або Yahoo та гарантує захищеність всіх даних [21].

2.1.3 Реалізація роботи в реальному часі

Так як запорукою створення успішної веб-платформи є забезпечення інтерактивності, то надання можливості роботи в реальному часі є необхідністю. Для даної задачі найкраще підходить бібліотека SignalR, яка обробляє підключення клієнтів та розсилає повідомлення для всіх таких користувачів (див. рис. 2.1). Для того щоб уникнути надсилання інформації клієнтам, які її не потребують, було вирішено уникнути роботи з постійними підключеннями, а перейти до роботи з

інструментом для реєстрації підключень лише у певному місці сайту — «хабом».

Не найменш важливим аспектом при виборі даної бібліотеки стало те, що її можна використовувати разом з ASP.NET Core.

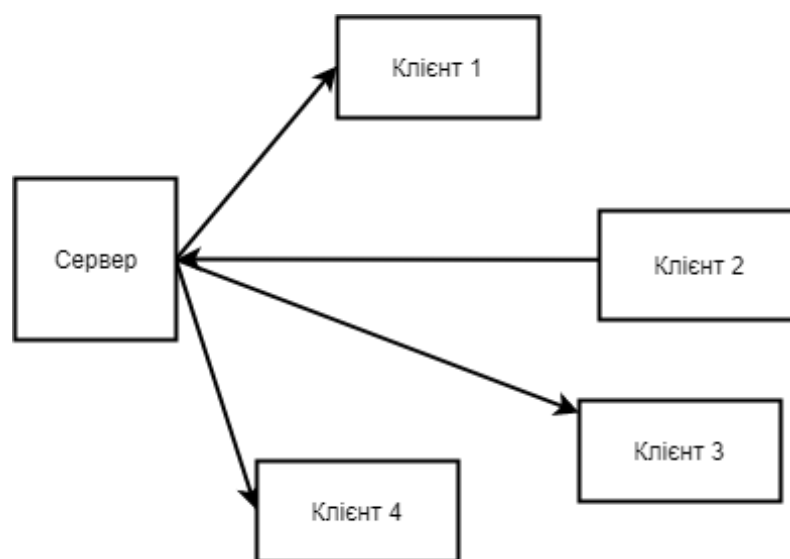


Рисунок 2.1 — Розсилка сповіщень

2.1.4 Вибір стилю дизайну для веб-сайту

Уже кілька років поспіль набуває популярності лаконічність та простота в дизайні. Серед провідних варіантів стилю є Material Design, який розроблений компанією Google. Він не є плоским, що робить його більш інтуїтивним для багатьох користувачів, та надає останнім зворотній зв'язок у вигляді невеликої анімації. Лаконічність даного дизайну зберігається за рахунок однотонного фону з додаванням яскравих кольорів на ключові елементи інтерфейсу.

Багато фронтенд фреймворків роблять інтеграції з Material Design, в тому числі Angular.

2.2 Розробка веб-інтерфейсу

2.2.1 Налаштування маршрутизації додатку

Маршрути повідомляють маршрутизатору, що потрібно показати користувачу, якщо він натиснув на посилання. У фреймворку Angular є модуль `AppRoutingModule`, який забезпечує маршрутизацію додатку. Даний модуль має знаходитись всередині файлу `app-routing.module.ts`, згенерувати який можна за

допомогою наступної команди:

```
ng generate module app-routing --module=app
```

«--module=app», у даному випадку, показує CLI, що необхідно зареєструвати файл серед масиву усіх імпортованих файлів.

Одразу після генерації, файл має наступний вигляд:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
@NgModule({
  imports: [
    CommonModule
  ],
  declarations: []
})
export class AppRoutingModule { }
```

Маршрут у Angular складається із двох властивостей:

— *path* — URL-адреса в адресному рядку браузера;

— *component* — компонент, який потрібно відтворити під час навігації по заданому маршруту.

```
const routes: Routes = [
  { path: 'about-us', component: AboutUsComponent },
  { path: 'contact', component: ContactComponent },
  { path: 'profile', canActivate: [AuthGuard], component: UserProfileComponent },
  { path: 'newproject', canActivate: [AuthGuard, TranslatorGuardService],
component: NewProjectComponent },
  { path: 'newteam', canActivate: [AuthGuard, TranslatorGuardService],
component: NewTeamComponent },
  { path: 'profile/settings', canActivate: [AuthGuard], component:
UserSettingsComponent },
```

```

    { path: 'guideline', canActivate: [AuthGuard], component:
GuidelineComponent},
  ]

```

Детальна маршрутизація веб-сервісу наведена у додатку Б.

Властивість `canActivate` обмежує доступ користувачам, наприклад, значення `[AuthGuard]` вказує на те що дана сторінка доступна лише авторизованим користувачам:

```

export class AuthGuard implements CanActivate {
  constructor(private router: Router, private appState: AppStateService) { }
  canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot) {
    if (this.appState.LoginStatus) {
      if (!this.appState.currentDatabaseUser)
      {
        return this.appState.getDatabaseUser().pipe(
          skip(1),
          map((user: UserProfile) => {
            return true;
          })
        )
      } else {
        return true;
      }
    }
    this.router.navigate(['/dashboard']);
    return true;
  }
}

```

А значення `[TranslatorGuardService]` здійснює перевірку чи не є користувач перекладачем.

2.2.2 Авторизація користувачів

Якщо користувач вперше заходить до системи, то він має можливість зареєструватись у ній кількома способами:

- створивши новий логін та пароль;
- використавши сервіси Google або Facebook.

Після цього, дані в профілі користувача автоматично заповнюються.

Для того, щоб забезпечити можливість реєстрації за допомогою логіну та паролю, створена функція `signUpRegular`:

```
signUpRegular(email: string, password: string) {
    return from ( this._firebaseAuth.auth.createUserWithEmailAndPassword ( email,
password ));
}
```

Для того щоб надати можливість авторизації в системі для уже зареєстрованих користувачів, розроблену функцію `signInRegular` :

```
signInRegular(email: string, password: string) {
    return from ( this._firebaseAuth.auth.signInWithEmailAndPassword ( email,
password));
}
```

Для реалізації можливості реєструватись в додатку та в подальшому авторизуватись за допомогою сторонніх сервісів (Google або Facebook) написано наступні функції:

```
signInWithGoogle() {
    return from(this._firebaseAuth.auth.signInWithPopup(
        new firebase.auth.GoogleAuthProvider()
    ));
}
signInWithFacebook() {
    return from(this._firebaseAuth.auth.signInWithPopup(
```

```

    new firebase.auth.FacebookAuthProvider()
  ));
}

```

За замовчування, кожен користувач має обмежений час для перебування на веб-сайті після входу до неї. Потім система повертає його до початково стану та просить ввійти ще раз. Для того щоб забезпечити безперерійну роботу веб-додатку, потрібно встановити на об'єкт Auth «спостерігач»:

```

this._firebaseAuth.auth.onAuthStateChanged(async (user) =>
{
  if(user){
    await this.refreshToken();
    this.appState.currentFirebaseUser = user;
  }
});

```

Але, також користувач не може знаходитись в системі постійно. Саме тому варто реалізувати можливість виходу із неї. Для цього створено невелику функцію logout:

```

logout() {
  return from(this._firebaseAuth.auth.signOut()).pipe(
    tap(() => {
      this.appState.updateState(null, "", false, null);
      this.router.navigate(['/']);
    })
  );
}

```

2.2.3 Створення користувацького інтерфейсу

В односторінкових застосунках, при переході за посиланням всередині додатку, відбувається завантаження не повністю нової сторінки, а лише її певної

частини — компоненти, яка передбачена у властивості `component` маршруту.

Використання компонент дозволяє незалежно створювати частини інтерфейсу, які в подальшому можна дуже ефективно використовувати повторно. Такий підхід в разі підвищує продуктивність додатку та збільшує швидкість його розробки, адже розробнику, наприклад, не є необхідним створювати схожі кнопки на кожній із сторінок, йому варто лише раз написати гнучкий компонент для відображення кнопки та імпортувати його в потрібні місця. Імпорт відбувається наступним чином:

```
import { PlusBtnComponent } from '../components/PlusBtn/about-us.component';
```

Для завантаження модулів лише по мірі необхідності, Angular підтримує патерн дизайну, що називається «ліниве завантаження» (lazy loading). Прикладом цього можуть слугувати деталі проекту на веб-сайті (див. рис. 2.2).

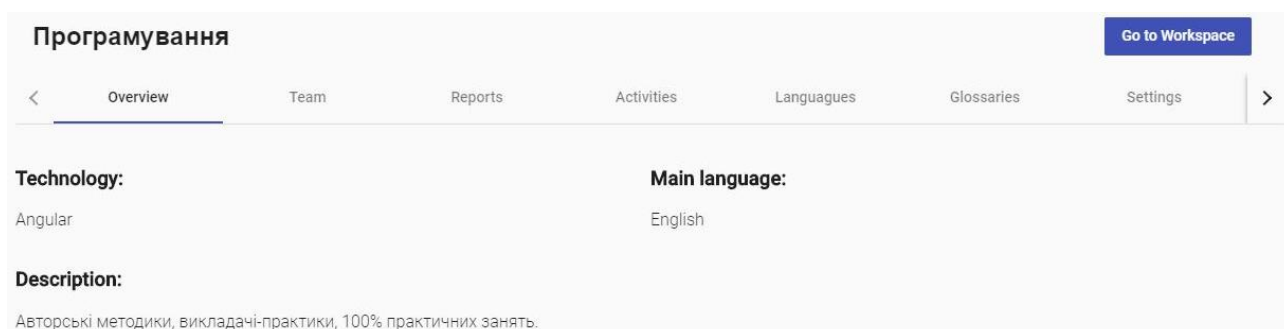


Рисунок 2.2 — Деталі проекту

На рисунку видно наявність різних вкладок («Overview», «Teams», «Activities», «Languages» та інші). Так як користувач не може одночасно переглядати їх всі, то і не потрібно попередньо завантажувати їх. Саме тому сюди чудово підходить застосування патерну лінивого завантаження. Для здійснення переходу між вкладками застосовується наступний код:

```
<mat-tab label="Team">
  <ng-template matTabContent>
    <app-project-team [IsCurrentUserManager]="checkIfUserManager" [projectId]
```

```

= "project.id"></app-project-team>
  </ng-template>
</mat-tab>
<mat-tab label="Reports">
  <ng-template matTabContent>
    <app-reports [projectId]="project.id"></app-reports>
  </ng-template>
</mat-tab>
<mat-tab label="Activities">
  <ng-template matTabContent>
    <app-project-activities [projectId]="project.id"></app-project-activities>
  </ng-template>
</mat-tab>

```

Повна реалізація вкладок наведена у додатку В.

Майже невід'ємною частиною кожного веб-сайта є використання діалогових вікон. При вході в систему чи реєстрації, при створенні нового проекту чи рядку доцільне їх використання. Зазвичай реалізація відбувається за рахунок модальних вікон, які містять в собі різні форми, які має заповнити користувач. Наприклад, реалізація вікна для входу в систему (детальніший опис наведено у додатку Г):

```

<form (submit)="onLoginFormSubmit(user, form)" class="d-flex flex-column px-3" action="javascript:void(0)" #form="ngForm">
  <h4 class="text-center">Login</h4>
  <mat-form-field class="my-1">
    <input id="email" name="email" type="email" [(ngModel)]="user.email"
required matInput placeholder="Your Email"
    #email="ngModel" autocomplete="on"
    pattern="[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~-
]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?"
    maxlength="50">

```

```

</mat-form-field>
<mat-form-field class="my-1">
  <input id="password" name="password" [type]="hide ? 'password' : 'text'"
  [(ngModel)]="user.password" required matInput placeholder="Your Password"
  #password="ngModel"
  maxlength="30">
  <mat-icon class="finger" matSuffix (click)="hide = !hide">{{hide ? 'visibility' :
  'visibility_off'}}</mat-icon>
</mat-form-field>
</form>

```

Для форм обов'язковим є валідація (перевірка на правильність введення). Часом користувач може ввести логін або пароль не вірно, додати зайвий пробіл там, де його не має бути. Задля того, щоб відловлювати подібні помилки, створено валідатор:

```

validate(control: AbstractControl): { [key: string]: any } {
  let isWhitespace = (control.value || "").trim().length === 0;
  let isValid = !isWhitespace;
  return isValid ? null : { 'WhiteSpace' : true };
}

```

Після того, як допущену помилку знайдено, варто повідомити користувача про неї і підказати що він зробив не так. Зазвичай помилки допущені в формах відображаються під відповідним полем для введення:

```

<mat-error *ngIf="password.errors && password.errors.required">
  Password is
  <strong>required</strong>
</mat-error>
<mat-error *ngIf="password.errors && password.errors.maxlength">
  Password <strong>cannot</strong> be more than 30 characters long

```

</mat-error>

2.2.4 Забезпечення адаптивності дизайну

Кожен користувач прагне аби веб-сайт «спілкувався з ним». Адаптивність дизайну є одним із способів подібного спілкування, адже, наприклад, при зміні ширини сторінки, додаток змінюватиме свій вигляд на більш підходящий.

Для досягнення найкращого варіанту адаптивності розроблено CSS Grid Layout, що є найпотужнішою на даний час системою для компоновання та розміщення елементів на сторінці, які доступна в CSS (та Sass відповідно). Її сенс полягає в побудові двовимірної сітки згідно з якою розташовуються різні рядки та стовпці.

На даному веб-сайті, користувачу доступні два можливих варіанти вигляду дошки з проектами. Все залежить від положення перемикача у лівому верхньому куті на вкладці «Projects». Першим варіантом є відображення проектів у вигляді списку (див. рис. 2.3), а другим — у вигляді карток (див. рис. 2.4).

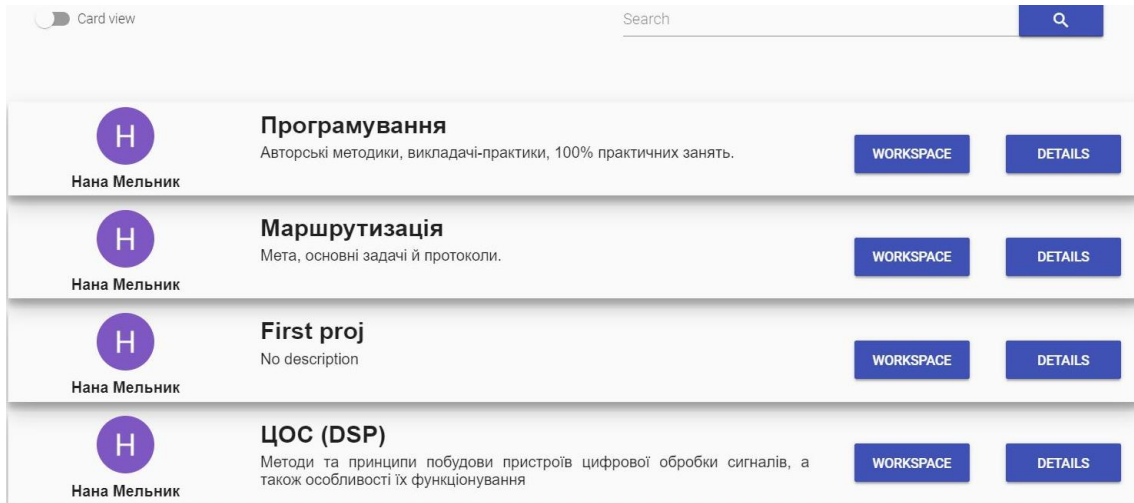


Рисунок 2.3. — Відображення проекту у вигляді рядків

Відображення проектів у вигляді рядків, відбувається за допомогою наступних налаштувань стилю:

```
.list-item
```

```
margin: 10px
```

padding: 5px

display: grid

grid-gap: 2%

box-sizing: border-box

grid-template-columns: 20% 78%

box-shadow: 0 10px 20px rgba(0,0,0,0.19), 0 6px 6px rgba(0,0,0,0.23)

max-height: 150px

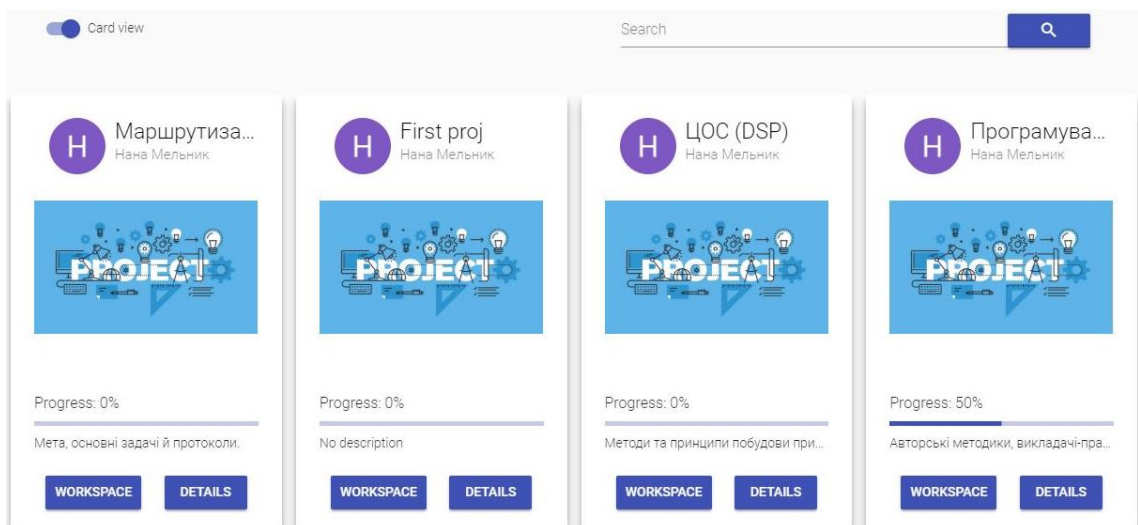


Рисунок 2.4 — Відображення проекту у вигляді карток

Та для відображення у вигляді сітки карток:

.container

margin-top: 30px

margin-bottom: 30px

display: grid

grid-template-columns: repeat(auto-fill, minmax(250px, 0.5fr))

grid-gap: 15px

justify-content: center

Ключовим є стиль відображення *grid*, який гарантує, що при додаванні нових проектів дошка продовжить зберігати гарний вигляд, за рахунок використання сітки.

Детальний лістинг компонентів наведено у додатку Д.

2.2.5 Налаштування сповіщень

Для покращення інтерфейсу користувача доцільно імплементувати у веб-додаток сповіщення. Таким чином, крім можливості працювати в режимі реального часу в одному місці сайту, будуть сповіщенні про певні зміни, які стосуються їх, у іншому місці. Створення зовнішнього виду сповіщень налаштовується за допомогою наступної розмітки:

```
<mde-popover #notifPopover="mdePopover" [mdePopoverOverlapTrigger]=
"false" mdePopoverArrowColor="#fff" [mdePopoverArrowOffsetX]="21">
  <div class=popover>
    <mat-card class="card" *ngFor="let notification of userNotifications">
      <mat-card-header>
        <mat-card-title class="card-main">
          <div>
            <div>From:
              <strong>{{ notification.sender.fullName }}</strong>
            </div>
            <p>
              <i>{{ notification.message }}</i>
            </p>
          </div>
        </mat-card-title>
      </mat-card-header>
      <mat-card-content class="card-navigation">
        <a *ngFor="let notificationOption of notification.options" mat-
button mat-raised-button color="primary" class="card-navigation-button"
(click)="buttonClick(notification, notificationOption.optionDefinition)">
          {{ getStringEnum(notificationOption.optionDefinition)}}
        </a>
      </mat-card-content>
```



```

        </mat-card>
    </div>
</mde-popover>

```

Для роботи зі сповіщеннями розроблено сервіс NotificationService, у якому реалізовані такі методи як надсилання сповіщення користувачу, їх отримання та видалення:

```

getCurrentUserNotifications() : Observable<Notification[]> {
    return this.dataService.sendRequest(RequestMethod.Get, this.api);
}
sendNotification(notification: Notification) : Observable<Notification> {
    return this.dataService.sendRequest(RequestMethod.Post, this.api,
undefined, notification);
}
removeNotification(notificationId:number) : Observable<Notification[]> {
    return this.dataService.sendRequest(RequestMethod.Delete, this.api,
notificationId);
}

```

Повний лістинг реалізації роботи зі сповіщеннями наведено у додатку .

3 РОБОТА ПРОГРАМИ

Перше, що можна побачити, відкривши веб-сайт Polyglot — це сторінку, яка пропонує зареєструватись в системі або увійти в неї, якщо користувач уже має обліковий запис (див. рис. 3.1).

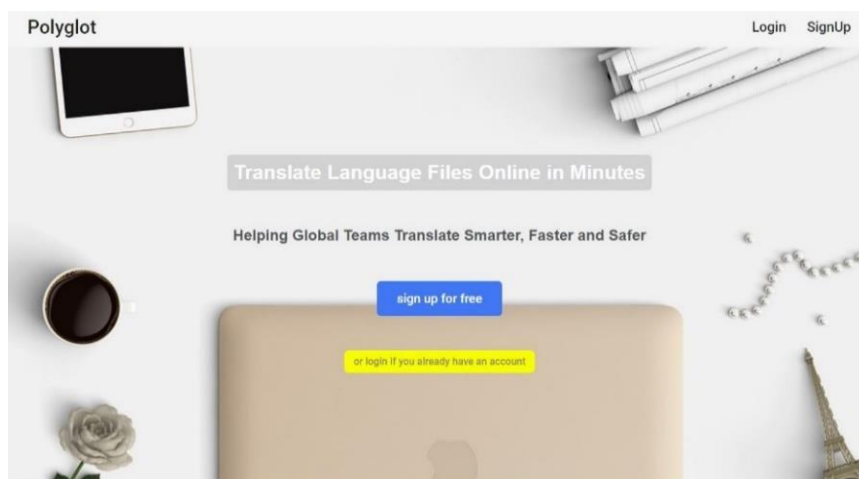


Рисунок 3.1 — Вигляд початкової сторінки веб-сайту

Легкі та зрозумілі форми для реєстрації та входу до системи (див. рис. 3.2) дозволяють власноруч ввести необхідні дані або скористатись можливістю авторизації за допомогою акаунта Google чи Facebook.

Рисунок 3.2 — Форми для реєстрації та входу до платформи Polyglot

Після натиснення на кнопки «Continue!» або «Log In», в залежності від ситуації, дані, які були введені, пересилаються до одного із шарів клієнта, а саме до шару сервісів. Таким чином починається логіка клієнтської частини додатку.

При першій реєстрації користувача, йому буде надана можливість обрати між двома ролями: менеджер або перекладач. Задача останнього полягає у локалізації контенту на визначену замовником мову.

Роль менеджера може виконувати як і власне замовник, так і спеціально підготована та обізнана в проекті людина, яка буде предстаником замовника на веб-сайті. До його обов'язків входить створення нових проектів, керування підбором найкращих та найкваліфікованіших для проекту фахівців. Для спрощення пошуку, особиста профіль кожного перекладача містить його резюме — коротка інформація про навички, володіння мовами, з якими він працює, та вірогідне перебування у складі уже існуючої команди.

Сторінка менеджера також має коротке резюме, тому, щоб перекладачу знати чи приймати пропозицію, можна відвідати особистий профіль останнього.

Повний список задач двох видів користувачів наведено на рисунку 3.3 у use-case діаграмі.

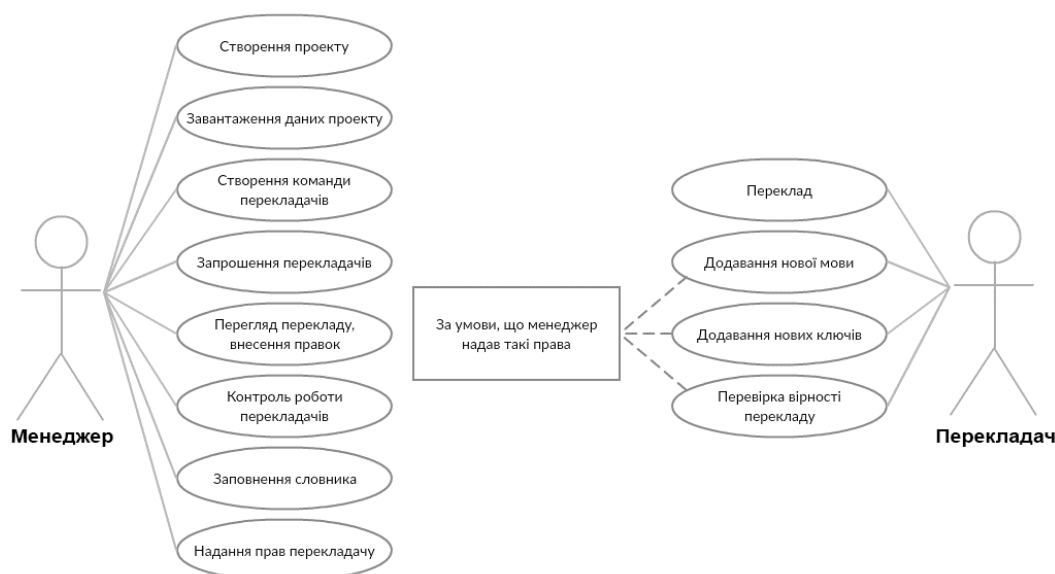


Рисунок 3.3 — Use case діаграма можливостей користувачів на веб-сайті відповідно до прав

3.1 Керування для менеджера

Як було згадано раніше, перше, що може здійснити менеджер, це створити новий проект, що відповідатиме додатку. Для цього необхідно клікнути кнопку «+»

у нижньому правому куті. В діалоговому вікні, що з'явилося (див. рис. 3.4), необхідно ввести назву, опис, вибрати основну мову для новоствореного проекту та сферу діяльності, а також є можливість додавання медіаресурсів.

Рисунок 3.4 — Форма для створення проекту

Після введення даних, необхідно натиснути кнопку «Create». Відтепер та надалі проект буде показаний на вкладці «Projects» (див. рис. 3.5).

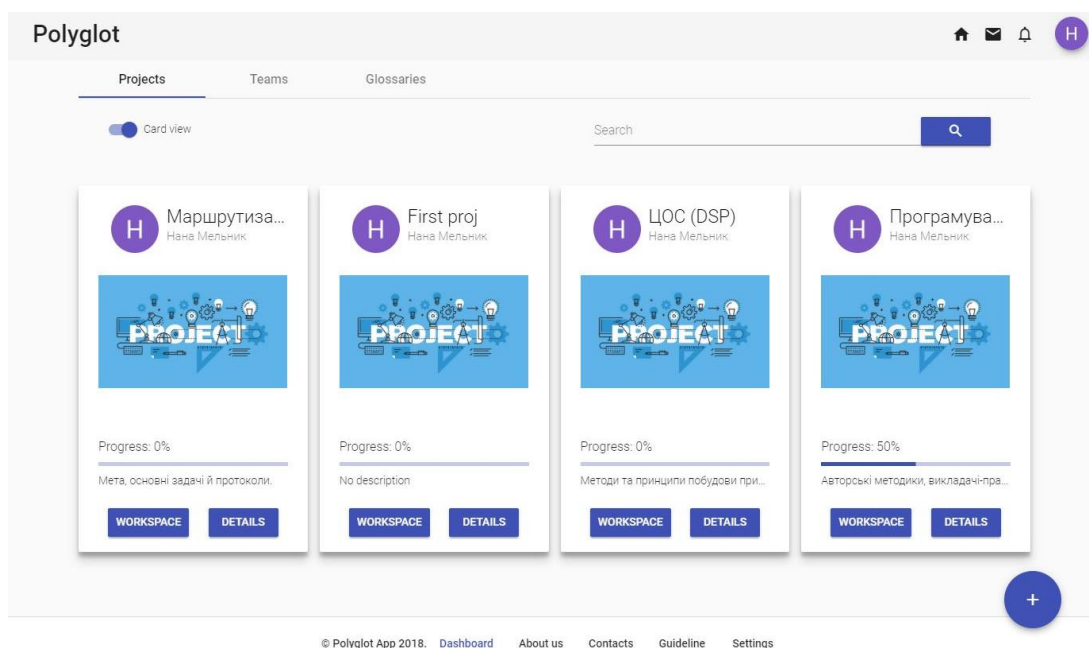


Рисунок 3.5 — Створені проекти на вкладці «Projects»

Після створення проекту, можна розпочати роботу над ним, перейшовши до сторінки «Workspace», клікнувши на однойменну кнопку (див. рис. 3.5). Саме тут менеджер має змогу додавати рядки, які у майбутньому будуть перекладені. Для цього необхідно натиснути кнопку «Add new string» і написати унікальний ключ та рядок оригінальною мовою.

Окрім того опціонально можна ввести опис (примітки) до обраного рядку та додати тег або медіа (див. рис. 3.6).

Рисунок 3.6 — Форма для додавання нового рядка

У випадку якщо менеджер має заготовлений для локалізації контент у файлах із розширенням `.resx` чи `.json`, то є можливість додати його до проекту (веб-сайт автоматично розіб'є його на рядки) (додаток Е). Здійснити це можна клікнувши на кнопку «Details» (див. рис. 3.5) та перейшовши до «Files» (див. рис. 3.7).

Тут же в кінці буде можливістю експортувати локалізований вміст у вигляді файлів із розширенням `.resx` чи `.json` (як було сказано раніше).

Аби почати роботу над проектом, менеджеру необхідно відібрати перекладачів та сформувати з них команду. Процес створення команди подібний до процесу створення проекту, тільки відбувається на вкладці «Teams». При натисненні на знак «+» у нижньому правому куті, з'явиться форма (див. рис. 3.8), де потрібно ввести назву команди та додати спеціалістів. У додатку Ж наведено алгоритм процесу приєднання перекладача до колективу.

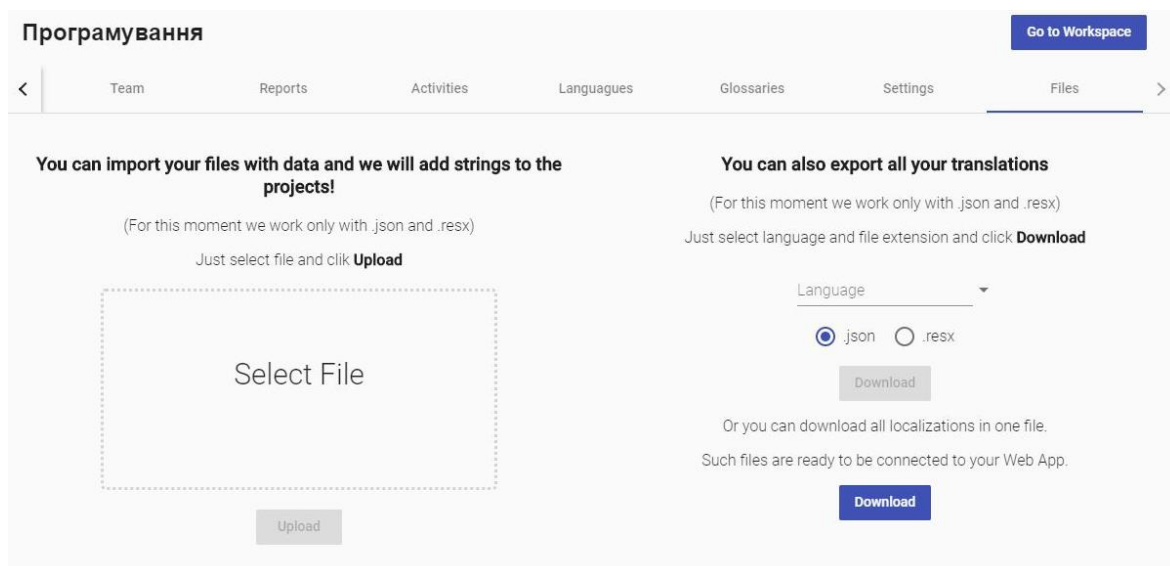


Рисунок 3.7 — Вкладка для імпорту та експорту файлів локалізації

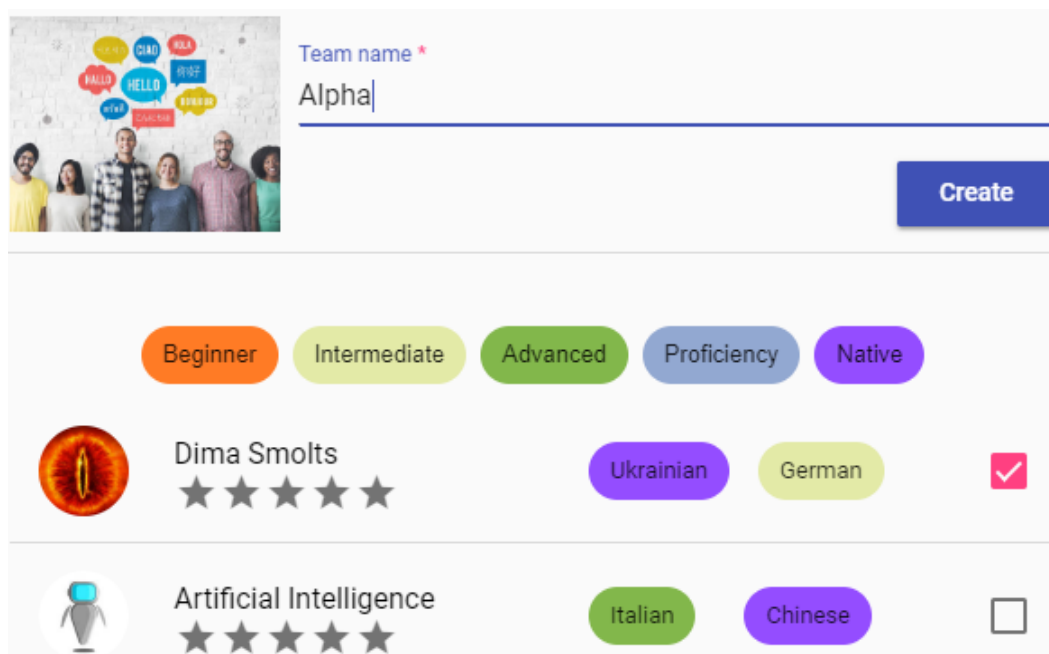


Рисунок 3.8 — Форма створення нової команди

Так як перекладачі можуть бути не достатньо обізнаними у спеціалізованому для проекту тексті, менеджер може полегшити їм роботу створивши своєрідний словник термінів та аббревіатур. Для цього потрібно перейти до вкладки «Glossaries», створити новий словник, натиснувши на знак «+», та заповнити його (див. рис. 3.9).

Для того, щоб приєднати створену команду до проекту, потрібно перейти до вкладки «Team» у його деталях. А для того, щоб приєднати словник необхідно

скористатись вкладкою «Glossaries».

Тепер менеджер може спостерігати за роботою спеціалістів та їх прогресом за допомогою вкладки «Report».

Programming			
Term	Explanation	Edit String	Delete String
OOP	Object-oriented programming	Edit	Delete
TDD	Test-driven development	Edit	Delete
KISS	Keep it simple, stupid	Edit	Delete
FIFO	First in first out	Edit	Delete
LIFO	Last in first out	Edit	Delete

Рисунок 3.9 — Приклад словника зі спеціалізованими термінами

3.2 Керування для перекладача

Користувач, який обрав роль перекладача, повинен або самостійно знайти проект, до якого бажає долучитись, або чекати поки менеджер не обере його як одного із спеціалістів для свого проекту. Останній названий випадок дає можливість погодитись на додавання до команди або ні.

Основне місце роботи для перекладача — це «Workspace» проекту, який розділений на три колонки (див. рис. 3.10).

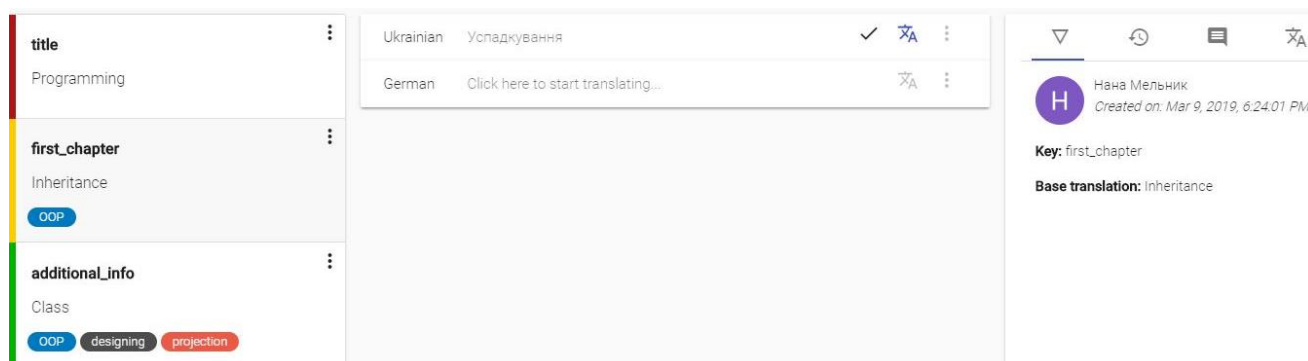


Рисунок 3.10 — Вигляд Workspace

Кожна колонка має власне призначення, наприклад:
— ліва, що містить список всіх рядків;

середня, яка є місцем виконання головної задачі — обробки обраних рядків та введення їх локалізованих варіантів;

— права, що містить кілька вкладок з додатковою інформацією про обрані рядки (див. рис. 3.11).

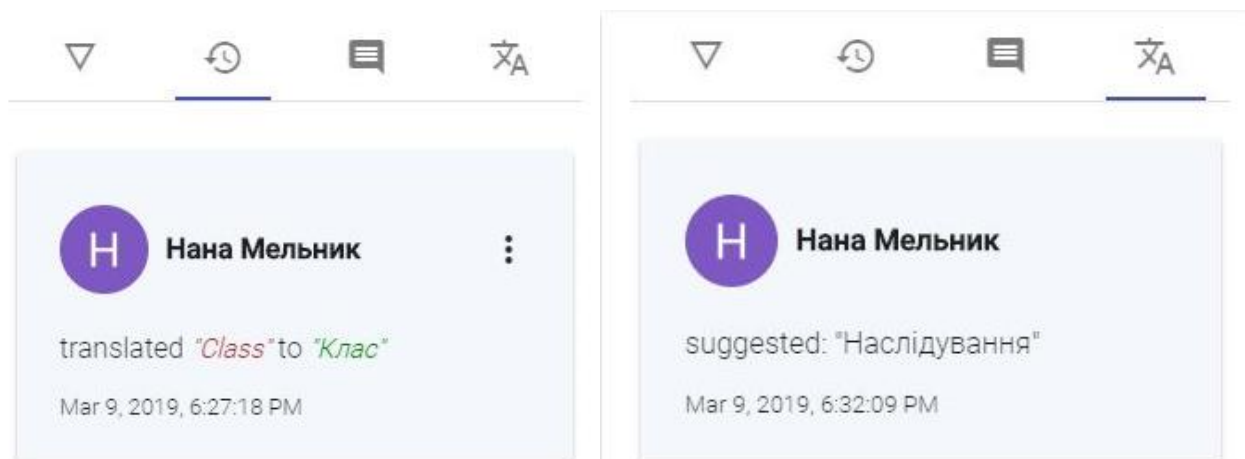


Рисунок 3.11 — Додаткова інформація про рядок

Якщо перекладач зустрічає невідомий символ чи аббревіатуру, він може скористатись словником, який додав до проекту менеджер. Для того, щоб дізнатись тлумачення, потрібно ввести дану аббревіатуру до середньої колонки Workspace і з'явиться коротке пояснення (див. рис. 3.12).

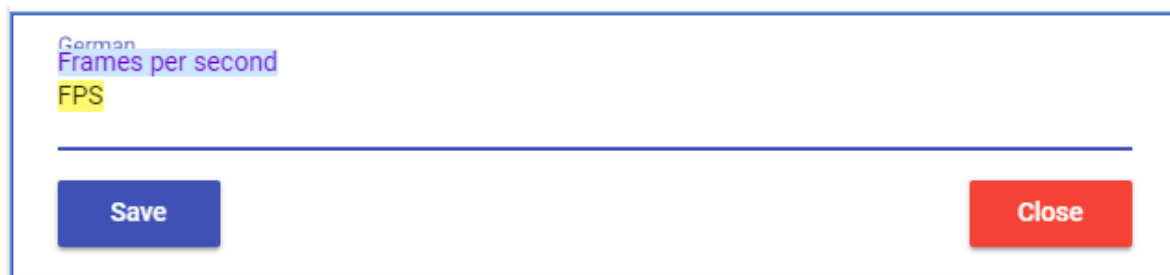


Рисунок 3.12 — Приклад застосування словника

Алгоритм життєвого циклу рядка під час його локалізації, наведено на рисунку 3.13.

Повна схема бази даних зображена у додатку И, а схема роботи система наведена у додатку К.



Рисунок 3.13 — Алгоритм життєвого циклу рядка під час його локалізації

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 3-х незалежних експертів. Такими експертами будуть Черняк О. І., Кадук О. В. та Захарченко С. М.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма рекомендованими критеріями (див. табл. 4.1) за 5-ти бальною шкалою.

Таблиця 4.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- те- Рій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Кінець таблиці 4.1

Бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.2.

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1. Черняк О. І.	2. Кадук О. В.	3. Захарченко С. М.
	Бали, виставлені експертами:		
1	4	4	4
2	3	3	3
3	4	4	3
4	4	4	4
5	4	4	4
6	3	3	4
7	3	3	3
8	4	4	4
9	3	4	4
10	4	4	4
11	4	3	3
12	4	4	4
Сума балів	СБ ₁ = 44	СБ ₂ = 44	СБ ₃ = 44
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = 44$		

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторськ-технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$З_0 = \frac{М}{T_p} \cdot t, \quad (4.1)$$

де М- місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 22$ дні;

t - число днів роботи розробника, t = 30 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.3.

Таблиця 4.3 — Розрахунки основної заробітної плати

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	8 500.00	404.76	6	2 428.57
Інженер- програміст 1	9 000.00	428.57	21	9 000.00
Інженер- програміст 2	9 000.00	428.57	21	9 000.00
Всього:				20 428.57

Додаткова заробітна плата Z_d всіх розробників розраховується як 10-12% від суми основної заробітної плати всіх розробників. Розрахуємо додаткову заробітну плату, взявши середнє відсоткове значення:

$$Z_{\text{дод}} = 0.11 \cdot 20\,428.57 = 2\,247.14 \text{ (грн.)}$$

Нарахування на заробітну плату $N_{\text{зп}}$ для працівників бюджетної сфери становить 22% від суми основної та додаткової заробітної плати:

$$N_{\text{зп}} = (Z_o + Z_d) \cdot 22\% = (20\,428.57 + 2\,247.14) \cdot 0,22 = 4\,988.66 \text{ грн.}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot N_a}{100} \cdot \frac{T}{12},$$

де $Ц$ — балансова вартість обладнання, грн;

N_a — річна норма амортизаційних відрахувань % (для програмного

забезпечення 25%);

T — Термін використання (у даному випадку 1 місяць).

Таблиця 4.4 — Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	12 000	25	1	249.00
Ноутбук	9600	25	1	199.20
Всього:				448.20

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_{1}^{n} N_i \cdot C_i \cdot K_i,$$

де n — кількість комплектуючих;

N_i — кількість комплектуючих і-го виду;

C_i — покупна ціна комплектуючих і-го виду, грн;

K_i — коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 4.5 — Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флеш-пам'ять	шт.	150	1	150
Блокнот	шт.	45	2	90
Ручка	шт.	15	2	30
Всього з урахуванням транспортних витрат:				297

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi},$$

де V — вартість 1кВт-години електроенергії ($V = 3$ грн/кВт);

Π — установлена потужність комп'ютера та ноутбука ($\Pi = 0,6$ кВт);

Φ — фактична кількість годин роботи комп'ютера та ноутбука ($\Phi = 336$ год.);

K_{Π} — коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,65$).

$$V_e = 3 \cdot 0.6 \cdot 336 \cdot 0.65 = 393.12 \text{ (грн.)}$$

Інші витрати $V_{ін}$ охоплюють: витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Їх можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p).$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 \cdot (20\,428.57 + 2\,247.14) = 22\,675.71 \text{ грн.}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$\begin{aligned} V &= Z_o + Z_d + H_{зп} + A + K + V_e + V_{ін} \\ V &= 20428.57 + 2247.14 + 4988.66 + 448.20 + 297 + 393.12 + 22675.71 = \\ &= 51\,478.40 \text{ грн.} \end{aligned}$$

Розрахуємо загальну вартість наукової роботи $V_{заг}$ формулою:

$$V_{заг} = \frac{V_{ін}}{\alpha}$$

де α — частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{\text{заг}} = \frac{51\,478.40}{1} = 51\,478.40$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta}$$

де β — коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{51\,478.40}{0.9} = 57\,198.22 \text{ (грн)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зрозуміло, що всі зроблені розрахунки будуть приблизними і не передбачають деталізації.

Виконання даної наукової роботи та впровадження її результатів займе один рік. Основні позитивні результати після впровадження результатів виконаної наукової розробки очікуються протягом 3-х років. А одним із основних позитивних результатів впровадження є зростання прибутку.

При впровадження результатів виконаної наукової розробки покращується якість програмного продукту, що дозволяє підвищити ціну реалізації на 1500 грн. Кількість одиниць реалізації програмного засобу також збільшиться: протягом першого року — на 40 шт., протягом другого року — ще на 30 шт., протягом третього року — ще на 20 шт.

Реалізація продукції до впровадження результатів наукової розробки орієнтовно складала 15 шт., а ціна — 4000 грн.

Не можливо прямо оцінити зростання чистого прибутку підприємства від провадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується

отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta C_{\text{я}} \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.2)$$

де $\Delta C_{\text{я}}$ — покращення основного якісного показника від впровадження результатів розробки у даному році;

N — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

C_o — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ — коефіцієнт, який враховує сплату податку на додану вартість (ставка 20%, коефіцієнт $\lambda = 0,8333$);

ρ — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток (18%).

Таким чином, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = [4000 \cdot 15 + (4000 + 1500) \cdot 40] \cdot 0,8333 \cdot 0,27 \cdot \left(1 - \frac{18}{100}\right) = 51\,520 \text{ грн}$$

Протягом другого року:

$$\begin{aligned} \Delta\Pi_2 &= [4000 \cdot 15 + (4000 + 1500) \cdot (40 + 30)] \cdot 0,8333 \cdot 0,27 \cdot \left(1 - \frac{18}{100}\right) \\ &= 81\,880 \text{ грн} \end{aligned}$$

Протягом третього року:

$$\begin{aligned} \Delta\Pi_3 &= [4000 \cdot 15 + (4000 + 1500) \cdot (40 + 30 + 20)] \cdot 0,8333 \cdot 0,27 \cdot \left(1 - \frac{18}{100}\right) \\ &= 102\,120 \text{ грн} \end{aligned}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Розрахований у підрозділі 4.3 комерційний ефект від можливого впровадження розробок не означає, що розробка буде реально впровадження. Основними показниками, які визначають доцільність фінансування наукової розробки інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

На першому кроці розраховується теперішня вартість інвестицій PV , що вкладаються в наукову розробку. Такою вартістю можемо вважати прогнозовану величину загальних витрат ZB на виконання та впровадження результатів НДДКР, розраховану за формулою, тобто будемо вважати, що $ZB = PV = 57\,198.22$ грн.

На другому кроці розраховується очікуване збільшення прибутку $\Delta\Pi$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами, за допомогою формули 4.2.

На третьому кроці будемо вісь часу, на яку наносимо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Загальні витрати ZB на виконання та впровадження результатів наукової роботи (або теперішня вартість інвестицій PV) дорівнює 57 198.22 грн. Результати вкладених у наукову розробку інвестицій почнуть виявлятися протягом трьох років. Вони виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 51 520 грн відносно базового року, у другому році — збільшення чистого прибутку на 81 880 грн (відносно базового року), у третьому році — збільшення чистого прибутку на 102 120 грн (відносно базового року).

Таким чином, рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рисунку 4.1.

На четвертому кроці розраховується абсолютна ефективність вкладених інвестицій $E_{\text{абс}}$.

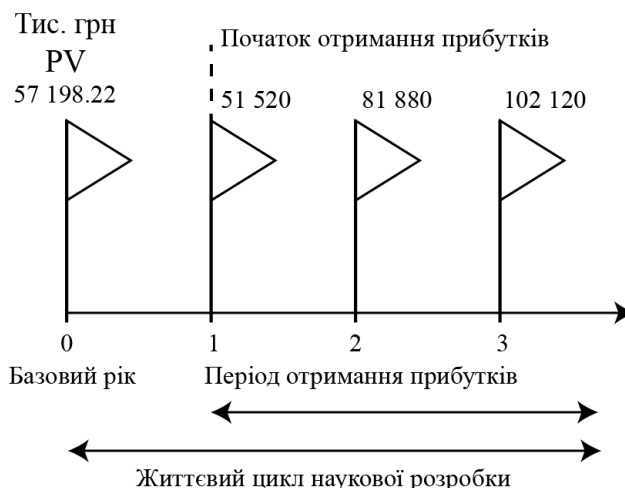


Рисунок 4.1 — Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів наукової роботи

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - \text{PV}),$$

де ПП — приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій $PV=ЗВ$, грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_1^m \frac{\Delta\Pi_i}{(1 + \tau)^t}$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

τ — період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t — період часу (в роках) від моменту отримання чистого прибутку до

точки «0».

Розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{51\,520}{(1 + 0,1)^1} + \frac{81\,880}{(1 + 0,1)^2} + \frac{102\,120}{(1 + 0,1)^3} = 191\,232,05 \text{ грн}$$

Тоді $E_{\text{абс}}$:

$$E_{\text{абс}} = 191\,232,05 - 57\,198,22 = 134\,033,83 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР є доцільним.

На п'ятому кроці розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ за формулою:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{\text{PV}}} - 1$$

де $E_{\text{абс}}$ — абсолютна ефективність вкладених інвестицій, грн;

PV — теперішня вартість інвестицій $\text{PV} = \text{ЗВ}$, грн;

$T_{\text{ж}}$ — життєвий цикл наукової розробки, роки.

Далі, розраховану величина $E_{\text{в}}$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = 0,2$;

f — показник, що характеризує ризикованість вкладень, середня величина $f = 0,08$.

$$\tau = 0,2 + 0,08 = 0,28$$

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{134\,033.83}{57\,198.22}} - 1 = 0.50 \text{ або } 50\%$$

Оскільки $E_B = 50\% > \tau_{\text{мін}} = 0,28 = 28\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

На шостому кроці розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}$$

Якщо $T_{\text{ок}}$ буде менше 5 років, то фінансування даної наукової розробки є доцільним.

Для даної розробки термін окупності вкладених у реалізацію проекту інвестицій $T_{\text{ок}}$ складе:

$$T_{\text{ок}} = \frac{1}{0.50} = 2 \text{ (роки)},$$

Так як $T_{\text{ок}} < 3...5$ -ти років, то фінансування даної наукової розробки є доцільним.

Отже, в даному розділі було здійснено оцінювання комерційного потенціалу даної наукової роботи.

Проведено технологічний аудит з залученням трьох незалежних експертів. Аналіз експертних даних показав, що наукова робота має високий рівень комерційного потенціалу.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальна вартість витрат на розробку і впровадження складає 57 198.22 грн.

Абсолютна ефективність вкладених інвестицій, яка дорівнює 134 033.83, є більшою 0 і показує, що інвестори будуть зацікавлені у нашій розробці.

Відносна ефективність розробки становить 50%, що є досить гарним

показником та вищим за мінімальну ставку дисконтування, тому вкласти гроші у нашу наукову розробку вигідніше, ніж, наприклад, покласти на депозит у банку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій складе 2 роки, що є менше 5-ти і вказує швидку окупність вкладених інвестицій.

Такі показники дозволяють зробити висновок про доцільність розробки та впровадження, а також актуальність фінансування даної наукової розробки.

ВИСНОВКИ

У магістерській дипломній роботі було розроблено клієнтську частину програмного засобу для локалізації веб та мобільних додатків з метою полегшення процесу розширення цільової аудиторії для власників даних додатків, за рахунок надання зручної місця для локалізації їх інтерфейсу.

Було вирішено задачу розробки такого інтерфейсу програмного засобу, яка уникає недосконалостей існуючих наразі сервісів, та забезпечення модульності додатку, реалізацію авторизації користувачів та підтримки двох можливих типів клієнтів, забезпечення роботи сайту у режимі реального часу та надсилання актуальних сповіщень, які відбуваються у проектах або командах, до яких належать користувачі.

Створено дружній до користувача інтерфейс з адаптивним дизайном, який забезпечує зручність використання сайту, позитивний досвід його використання користувачем, полегшує процес локалізації для розробників і який сприяє поширенню регіонів використання застосунків для замовників.

Таким чином поставлена мета була досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Speak to Global Customers in Their Own Language [електронний ресурс] // Режим доступу: <https://hbr.org/2012/08/speak-to-global-customers-in-t> — Назва з екрану.
2. Азаров О. Д., Черняк О. І., Смольц Д. О., Мельник Ж. А. «Проблема локалізації веб та мобільних додатків» в Матеріали конференції «XLIX Науково-технічна конференція підрозділів Вінницького національного технічного університету (2020)», Вінниця, 2020. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allvntu/index/pages/view/zbirn2020>.
3. Локалізація в порівнянні з Інтернаціоналізацією [електронний ресурс] // Режим доступу: <https://www.w3.org/International/questions/qa-i18n.uk> — Назва з екрану.
4. What is localization and why is it so important in 2020? [електронний ресурс] // Режим доступу: <https://www.smartcat.com/blog/what-is-localization/> — Назва з екрану.
5. Черняк О. І., Залізецький В.В. «Вирішення проблем локалізації в мультимодульній структурі програмного забезпечення» у Х Міжнародній конференції «Молоді вчені 2019—від теорії до практики», Дніпро, 07 березня 2019 р.: 179-181.
6. Top 10 cultural adaptation tips from technology giants [електронний ресурс] // Режим доступу: <https://www.deseretnews.com/top/1554/0/Top-10-cultural-adaptation-tips-from-technology-giants.html> — Назва з екрану.
7. 7 principles that make your global content localisation stand out [електронний ресурс] // Режим доступу: <https://www.brightlines.co.uk/translationservices/localisation/7-principles-that-make-your-global-content-localisation-standout/> — Назва з екрану.
8. Website Localization: 3 Basic Principles [електронний ресурс] // Режим доступу: <https://uxpamagazine.org/website-localization/> — Назва з екрану.
9. Lost in Translation: 10 International Marketing Fails [електронний ресурс] // Режим доступу: <https://www.businessnewsdaily.com/5241-international-marketing->

[fails.html](#) — Назва з екрану.

10. 6 essential steps to planning a successful localization strategy [електронний ресурс] // Режим доступу: <https://blog.applingua.com/blog/localization-strategy> — Назва з екрану.

11. 5. Черняк О. І., Залізецький В.В. «Вирішення проблем локалізації в мультимодульній структурі програмного забезпечення» у Х Міжнародній конференція «Молоді вчені 2019—від теорії до практики», Дніпро, 07 березня 2019 р.: 179-181.

12. Language Technology [електронний ресурс] // Режим доступу: <https://www.gala-global.org/knowledge-center/about-the-industry/language-technology> — Назва з екрану.

13. Дем'янчук Ю. І. «Різновиди корпусу текстів у процесі перекладу документів офіційно ділового стилю» [Електронний ресурс]. Режим доступу: http://ddpu-filolvisnyk.com.ua/uploads/arkhiv-nomerov/2016/NV_2016_5-1/27.pdf.

14. What is Machine Translation? Rule Based Machine Translation vs. Statistical Machine Translation [електронний ресурс] // Режим доступу: <https://www.systransoft.com/systran/translation-technology/what-is-machine-translation/> — Назва з екрану.

15. What is GeoIP? [електронний ресурс] // Режим доступу: <https://help.nexcess.net/77285-other/what-is-geoip> — Назва з екрану.

16. Localization / Internationalization [електронний ресурс] // Режим доступу: <https://frontend.turing.io/lessons/module-4/localization.html> — Назва з екрану.

17. The Advantages and Disadvantages of JavaScript [електронний ресурс] // Режим доступу: <https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/> — Назва з екрану.

18. 10 Best JavaScript Frameworks to Use in 2020 [електронний ресурс] // Режим доступу: <https://hackr.io/blog/best-javascript-frameworks> — Назва з екрану.

19. Localizing your app [електронний ресурс] // Режим доступу: <https://angular.io/guide/i18n#steps-to-localize-your-app> — Назва з екрану.

20. The Good and the Bad of Firebase Backend Services [електронний ресурс] // Режим доступу: <https://www.altexsoft.com/blog/firebase-review-pros-cons->

[alternatives/](#) — Назва з екрану.

21. Why and When Should You Opt For Firebase? Features, Advantages, and Disadvantages Discussed [електронний ресурс] // Режим доступу: <https://kodytechnolab.com/features-benefits-firebase> — Назва з екрану.