

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

**ВЕБ-ПЛАТФОРМА ДЛЯ РЕАЛІЗАЦІЇ МОДЕЛІ "ПОТРІЙНОЇ
СПІРАЛІ" (ВЗАЄМОДІЇ УНІВЕРСИТЕТІВ, ІТ-КОМПАНІЙ ТА
ВЛАДНИХ СТРУКТУР)**

**Текстова частина до магістерської роботи
за спеціальністю «Інженерія програмного забезпечення»**

Виконав студент 2-го року навчання
Денисенко Ігор
Михайлович

Керівник Ковалюк Т. В.
доцент, к.н.

Магістерська робота захищена
з оцінкою «_____»

Секретар ДЕК _____
«__» _____ 2023 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
доц., канд. фіз.-мат. наук
С. С. Гороховський

_____ (підпис)

1 листопада 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на магістерську роботу

студенту Денисенку І. М. факультету інформатики 2-го курсу МП
Тема: «Веб-платформа для реалізації моделі "потрійної спіралі"»

Зміст ТЧ до дипломної роботи:

Індивідуальне завдання

Вступ

1. Дослідження

2. Розробка власної системи

3. Практичне застосування

Висновки

Список літератури

Дата видачі листопад 2023 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Календарний план виконання роботи:

№ п/п	Назва етапу дипломної роботи	Термін виконання	Примітки
1	Отримання завдання на дипломну роботу	Листопад 2022	
2	Огляд теоретичного матеріалу за темою	Січень 2023	
3	Опис теоретичної частини	Лютий 2023	
4	Продумування вимог до практичної частини	Березень 2023	
5	Проектування	Квітень 2023	
6	Написання пояснювальної роботи	Травень 2023	
7	Попередній захист	Травень 2023	
8	Аналіз та корегування	Травень 2023	
9	Створення слайдів для доповіді та написання доповіді	Червень 2023	
10	Захист	Червень 2023	

Анотація	5
Вступ.....	6
1 Дослідження.....	8
1.1 Концепція.....	8
1.2 Критерії оцінки проекту.....	12
1.2.1 Наявність даних.....	14
1.2.2 Конкурентоспроможність.....	15
1.2.3 Новизна.....	17
1.2.4 Сприйняття.....	19
1.2.5 Повторне використання	22
1.2.6 Основа проекту	23
1.2.7 Часові межі	24
1.2.8 Прибутковість	26
1.2.9 Досвід.....	27
1.2.10 Фінанси	29
1.2.11 Інші засоби реалізації	31
1.2.12 Ризики	32
1.3 Висновки до частини 1.....	33
2. Реалізація.....	34
2.1 Розробка сайту	34
2.1.1 Основний інструментарій.....	34
2.1.2 Розміщення проекту	36
2.2 База проекту - backend.....	38
2.3 База проекту - frontend.....	40
2.4 Висновки до частини 2.....	42
3. Практичне використання.....	43
3.1 Інтерфейс та основний бізнес-процес.....	43
3.2 Висновки до частини 3.....	47
Висновки	48
Список використаних джерел.....	49

Анотація

Дипломна робота реалізує можливість взаємодії трьох різних структур, тобто уряду, університетів та компаній у реалізації проектів уряду, процесу їх аналізу, розробки та імплементації проектів, що вирішуються суспільні чи будь які інші проблеми.

Перший розділ містить дослідження бізнес процесів, пов'язаних з основною темою роботи, потенційних шляхів реалізації, проблематику системи «потрійної спіралі» та планування вирішення поставлених задач, також і шляхи інтеграції чи живучість ідеї, як бізнес проекту.

Другий розділ описує реалізацію власного веб-ресурсу за допомогою сучасних бібліотек та використання CRM платформ. У цьому розділі показано процес створення клієнт-серверного застосунку, описано його структуру, компоненти та використані інструменти та технології.

Третій розділ демонструє практичне використання створеного застосунку для різних ролей та їх бізнес-процеси, які є частиною «потрійної спіралі». У даному розділі описано користувацький інтерфейс для різних типів користувачів, та взаємодію з ним.

Вступ

У сучасному світі розвиток інформаційних технологій відіграє ключову роль у взаємодії різних суспільних секторів, в даному випадку університетів, ІТ-компаній та владних структур. Ці сектори здатні працювати разом, створюючи взаємовигідні партнерства та сприяючи розвитку суспільства в цілому. Однак, існують певні виклики, пов'язані з ефективною організацією цієї взаємодії.

На сьогоднішній день існує потреба у створенні цілісної веб-платформи, яка забезпечуватиме ефективну комунікацію, обмін даними та співпрацю між університетами, ІТ-компаніями та владними структурами. Особливо важливою є реалізація моделі "потрійної спіралі", яка уособлює взаємодію між цими секторами та сприяє збагаченню розвитку усіх трьох перелічених галузей.

Метою даної дипломної роботи є розробка та реалізація веб-платформи для моделі "потрійної спіралі". Ця платформа буде пропонувати інструменти та реалізовувати процеси для покращення комунікації, обміну даними та співпраці між університетами, ІТ-компаніями та владними структурами. Платформа буде забезпечувати можливість владним структурам робити запити до університетів а університетам та відповідальним особам ІТ-компаній реалізувати проекти, запропоновані владними структурами.

Ця дипломна робота має важливе значення для розвитку сфери взаємодії між університетами, ІТ-компаніями та владними структурами. Реалізація цієї веб-платформи для моделі "потрійної спіралі" відкриє нові можливості для покращення спільної роботи та співпраці між цими секторами. Завдяки цьому університети зможуть ефективніше реалізовувати свої академічні досягнення та наукові розробки в проектах владних структур, що сприятиме швидкому трансферу технологій та створенню інноваційних продуктів.

ІТ-компанії, з свого боку, зможуть набути доступу до актуальних наукових досліджень та розробок, що сприятиме їхньому розвитку та конкурентоспроможності на ринку. Крім того, вони зможуть взаємодіяти з університетами та владними структурами для реалізації спільних проектів, що дозволить їм отримувати нові ідеї та перспективи для свого бізнесу.

Владні структури, зокрема органи місцевого самоврядування та управління, також зможуть зрозуміти потреби університетів та ІТ-компаній, що дозволить їм краще планувати та реалізовувати стратегії розвитку. Така взаємодія сприятиме залученню нових технологій, забезпеченню науково-технічного прогресу та підвищенню якості послуг для громадян.

Завершення цієї дипломної роботи матиме позитивний вплив на суспільство в цілому. Веб-платформа для моделі "потрійної спіралі" стане ефективним інструментом для спільної роботи, обміну знаннями та інноваційними розробками між усіма трьома секторами.

1 Дослідження

1.1 Концепція

Метою роботи автора є створення повноцінного програмного проекту з багатьох потенційних ідей, що можуть бути реалізовані. Почнемо з аналізу бізнес процесів, аналізу специфіки роботи кожної із сторін, задіяних у роботі «потрійної спіралі». На успіх проекту впливає те, наскільки він відповідатиме реаліям взаємодії усіх трьох сфер, їх можливостей та того, наскільки потенційний програмний застосунок зможе дійсно вирішити питання реалізації проектів для покращення життя суспільства.

Для того, аби проект був актуальним для сьогодення необхідно сконцентруватись на сучасних методах і засобах розробки, сучасних та актуальних SAAS, PAAS платформах, базах даних та веб сервісах. Також для клієнської частини потрібно докласти зусиль аби реалізувати необхідні процеси зі зрозумілим та інтуїтивним UX-дизайном, аби потенційні користувачі не відчували незручності у користуванні системою.

Для визначеності в межах дослідження та реалізація задуманого введемо певні поняття, на які будемо спиратись у аналізі та процесі розробки. По-перше, проектом будемо вважати весь процес планування, аналізу та реалізації та експлуатування певної ідеї, втіленої у чіткій реалізації проекту і ця реалізація і буде баченням автора щодо потенційно прибуткового проекту, спланованого і реалізованого для різних сфер, задіяних у концепції «triple helix model».

Тому треба сформулювати певні рамки, в яких ми реалізуємо цей проект. Зрозуміло, що реалізацій на дану тему можуть бути стільки, скільки є авторів подібних проектів, тому це конкретне рішення, над яким працюватиме автор буде реалізовувати певні принципи, закладені на етапі аналізу моделі і потенційної реалізації.

Беручи до уваги все вищезазначене, нам необхідно визначити набір процесів які реалізовуватиме програмний продукт. Після годин пошуків та формування ідеї у щось більш прагматичне і зрозуміле автор прийшов дійшов висновку, що ми маємо можливість реалізувати повний цикл розробки програмного забезпечення для процесу, коли уряд матиме можливість зробити запит на вирішення певної проблеми, а університети у складі окремих користувачів та команд будуть мати можливість конкурувати між собою за гранти та брати на себе відповідальність у реалізації проектів, за усім цим процесом будуть наглядати відповідальні особи з уряду, а представники ІТ-компаній будуть у ролі наглядачів, кураторів цих проектів, маючи великий досвід у реалізації різноманітних ІТ-продуктів.

Таким чином, ми маємо реалізувати систему для замкнутого циклу розробки продуктів і втілення їх у життя зі сторони бачення усіх трьох її учасників.

Беручи до уваги все вищезазначене, програмне рішення цього циклу розробки може бути вирішене багатьма способами, оскільки часу на повну реалізацію з максимальним набором функцій може бути витрачено тисячі, то для початку логічно було б реалізувати MVP даного проекту з основним набором функцій а далі, за допомогою демо-версії проекту вже шукати інвестиції або потенційних замовників, і далі аналізувати і розвивати проект відповідно до вимог замовника, коригуючи початковий план під реальні процеси урядової системи.

Усі дії, пов'язані зі змінами у проекті мають бути направлені на оптимізацію та підвищення ефективності етапів планування та реалізації проектів від уряду.

Зазначимо, що множина етапів, з яких складатиметься проект, може буде необмеженою, тому що проект може не мати завершального терміну.

Для прикладу можна навести випадки, коли виходять оновлення на платформу PaaS, оновлення браузеру чи бібліотек, що використовуються у проекті в різних його частинах, як на frontend так і backend.

Ефективне розподілення ресурсів розробників та бюджетів для даної теми роботи є ключовим етапом в процесі реалізації проекту. Нижче наведені кілька рекомендацій щодо оптимального розподілу ресурсів для веб-платформи моделі "потрійної спіралі":

1. Аналіз потреб і пріоритетів: Перш за все, необхідно провести детальний аналіз потреб університетів, IT-компаній та владних структур щодо веб-платформи. Визначити основні функціональні вимоги та пріоритети, що дозволить визначити необхідність розподілу ресурсів.
2. Технічний план розробки: Створення детального технічного плану розробки допоможе визначити, які компоненти та функції потребують більшої уваги та витрат ресурсів. Допоможе врахувати реалістичні терміни розробки, розподіливши ресурси на окремі етапи проекту.
3. Команда розробників: Бажано мати компетентну команду розробників, включаючи програмістів, дизайнерів та тестувальників, здатних ефективно працювати над реалізацією веб-платформи. Залучення фахівців зі спеціалізацією відповідно до потреб проекту допоможе забезпечити оптимальне використання розробницьких ресурсів. Проте на даному етапі розробки, дане дослідження та реалізація залежатиме лише від автора роботи.
4. Гнучкість та ітеративний підхід: Рекомендується використовувати гнучкі методології розробки, такі як Scrum або Agile, що дозволяють гнучко реагувати на зміни, встановлюють короткі ітерації розробки та фокусуються на пріоритетах. Це дозволить оптимізувати використання ресурсів та забезпечити постійний прогрес у реалізації

- проекту.
5. Бюджетний план: Важливо створити реалістичний бюджетний план, враховуючи потреби витрат на розробку веб-платформи. Оцінити необхідні витрати на програмне забезпечення, інфраструктуру та інші складові проекту. Потрібно планувати бюджет таким чином, щоб він задовольняв потреби проекту, уникати недофінансування або надмірних витрат, в тому числі і використання часу.
 6. Управління ресурсами: Ефективне управління ресурсами є ключовим фактором успіху проекту. Потрібно встановити механізми контролю та звітності, щоб слідкувати за використанням ресурсів, контролювати витрати та забезпечувати їх оптимальне використання. Регулярно оцінювати прогрес проекту та вносити необхідні коригування для забезпечення ефективного розподілу ресурсів.
 7. Партнерства та спонсорство: Аби ідея мала можливість в майбутньому Розглянути можливості партнерства з університетами, ІТ-компаніями та владними структурами для спільного фінансування проекту. Залучення спонсорів та грантових організацій може забезпечити додаткові ресурси для успішної реалізації веб-платформи.

1.2 Критерії оцінки проекту

Автор даної дипломної роботи встановлює наступні критерії оцінки проекту з метою визначення його успішності і досягнення поставлених цілей:

1. **Функціональність:** Проект буде оцінюватись на його здатність задовольняти потреби університетів, IT-компаній та владних структур у взаємодії та обміну інформацією. Важливо, щоб веб-платформа забезпечувала ефективне функціонування моделі "потрійної спіралі" і включала необхідні функції, інтерфейси та можливості для користувачів які можуть бути реалізовані у MVP цього продукту.
2. **Надійність:** Оцінка проекту буде враховувати його надійність і стабільність. Веб-платформа повинна працювати без збоїв, забезпечуючи неперервний доступ до інформації і функцій для користувачів. Наявність механізмів резервного копіювання даних і відновлення роботи системи також будуть оцінюватись.
3. **Ефективність:** Проект буде оцінюватись на його ефективність у використанні ресурсів, включаючи обсяги пам'яті, швидкість роботи та використання мережевих ресурсів. Важливо, щоб веб-платформа працювала оптимально і забезпечувала швидку та ефективну взаємодію між учасниками моделі "потрійної спіралі".
4. **Зручність використання:** Проект буде оцінюватись на його зручність використання для різних категорій користувачів. Веб-платформа повинна мати інтуїтивно зрозумілий інтерфейс, який дозволить користувачам легко доступатися до потрібної інформації та використовувати функціонал платформи без зайвих зусиль чи відчуття того, що немає розуміння що робити на кожному кроці.
5. **Безпека:** Оцінка проекту включатиме аспекти безпеки. Веб-

платформа повинна мати відповідні заходи безпеки для захисту конфіденційної інформації, обмеження несанкціонованого доступу та запобігання можливим загрозам безпеки даних. Також у різних категорій користувачів буде різний рівень доступу до функцій та даних системи.

6. Масштабованість: Проект буде оцінюватись за його здатність до масштабування. Веб-платформа повинна бути готовою розширюватися і впроваджувати нові функції або обслуговувати більшу кількість користувачів при необхідності без значних змін в архітектурі або коді конкретної реалізації продукту.
7. Інноваційність: Проект буде оцінюватись за його інноваційний потенціал. Веб-платформа повинна пропонувати новаторські рішення, які допомагають покращити взаємодію університетів, IT-компаній та владних структур, сприяють ефективному обміну інформацією та співпраці.

Враховуючи ці критерії, автор дипломної роботи зможе здійснити оцінку ефективності та успішності проекту з використанням веб-платформи для реалізації моделі "Потрійної спіралі" та її подальшого використання.

1.2.1 Наявність даних

Незалежно від того, що над проектом працює лише один розробник, який виконує різні ролі у проекті важливо зрозуміти, що для успішної реалізації проекту потрібно мати належну кількість інформації.

Необхідно вивчити модель "Потрійної спіралі" взаємодії університетів, ІТ-компаній та владних структур та провести дослідження відповідної предметної області. Якщо відсутні спеціалісти, які можуть надати консультації, розробник повинен вкласти додаткові зусилля для здобуття необхідних знань.

Це може включати самостійне дослідження, вивчення літератури, пошук інформації в Інтернеті та зв'язок з експертами з даної галузі. Важливо мати чітке розуміння того, що розробник створює і яке призначення має розроблюваний продукт.

Працюючи самостійно, розробник також повинен враховувати можливі проблеми та складнощі, які можуть виникнути під час розробки та підтримки проекту. Необхідно враховувати можливість затримок в графіку реалізації проекту, фінансових обмежень та необхідність проведення фіксів та оновлень у майбутньому.

Розробник, працюючи самостійно, повинен бути настільки досконалим у розумінні моделі "Потрійної спіралі" та предметної області, щоб ефективно реалізувати проект. Він має бути самодостатнім у збиранні інформації, вивченні нових технологій та вирішенні проблем, які можуть виникнути під час процесу розробки.

Також дуже важливо вивчити наявність та ключові моделі, за якими працюють конкуренти, аби на основі існуючих рішень, допрацювати і реалізувати своє рішення даної проблематики.

1.2.2 Конкуренентоспроможність

На основі моделі "Triple Helix Model" було розроблено різноманітні програмні продукти, які впроваджують концепцію взаємодії університетів, ІТ-компаній та владних структур. Ось кілька прикладів компаній, які використали цю модель для розробки своїх продуктів:

1. Google: Компанія Google використовує модель "Triple Helix" для співпраці з університетами та владними органами у розвитку нових технологій та розв'язанні складних завдань. Програмні продукти Google, такі як пошукова система Google, Gmail, Google Maps та інші, були створені на основі цієї моделі.
2. Microsoft: Компанія Microsoft активно співпрацює з університетами та владними органами для розробки програмних продуктів. Вони використовують концепцію моделі "Triple Helix" для створення інноваційних рішень та покращення своїх продуктів, таких як операційна система Windows, офісні програми Office, хмарні сервіси Azure та багато інших.
3. IBM: Компанія IBM активно співпрацює з університетами та владними структурами для розробки програмних продуктів і розвитку нових технологій. Вони використовують модель "Triple Helix" для створення інноваційних рішень в галузі штучного інтелекту, кібербезпеки, аналітики даних та інших областей.
4. Siemens: Компанія Siemens, яка спеціалізується на розробці програмного забезпечення та рішень в галузі промисловості та енергетики, використовує модель "Triple Helix" для співпраці з університетами та владними органами. Вони спільно розробляють інноваційні програмні продукти та технології.
5. Apple: Apple є ще однією компанією, яка використовує модель "Triple Helix" у своїй діяльності. Вони співпрацюють з університетами та владними органами для розробки інноваційних програмних продуктів, таких як

операційна система iOS, програмне забезпечення macOS, мобільні пристрої iPhone та інші.

6. Facebook: Facebook також використовує модель "Triple Helix" для співпраці з університетами та владними органами у розробці своїх соціальних медіа-платформ. Вони спільно працюють над вдосконаленням своїх алгоритмів, безпеки даних та функціональності продукту.

Усі наведені компанії давно співпрацюють з університетами та владою через систему грантів, або інших програм, які мають на меті вирішення певних проблем і реалізацію бізнес ідей.

Кожна з них має величезні фінансові, людські і технологічні ресурси, в цьому конкурувати з ними просто неможливо, тому ідеєю проекту буде реалізація хабу для місцевих органів влади, сам застосунок дозволить багатьом невеликим клієнтам мати можливість співпрацювати по моделі «потрійної спіралі» не маючи великих бюджетів, і при цьому реалізовувати певні проекти.

Плюсом даної системи буде також не складна інтеграція її в більшість бізнес систем США, тому за приклад і потенційних клієнтів я взяв місцеві органи влади США та їх потенційну зацікавленість у такого роду проектах і можливість отримати просту та зручну реалізацію продукту, яка дасть можливість співпрацювати з університетами та місцевими, або глобальними ІТ-компаніями.

1.2.3 Новизна

Для потенційної успішності проекту ми маємо гарантувати, що користувач, тобто в першу чергу уряд, отримає нові можливості, або отримає покращений додаток того, що він вже має і значно дешевше. Важно буде перевести людей, що працюють у цілих структурах на щось нове, по тій причині, що більшість людей не хочуть вчитись, тому переваги дійсно мають бути вагомими, аби в уряді з'явилися лоббісти переходу на нову систему.

Якщо перебрати всі потенційні пункти, за якими можна було б охарактеризувати програмне забезпечення як нове, то в моєму випадку, уряд, як замовник програмного забезпечення отримує перевагу у тому, що їм не доведеться в певному обсязі змінювати систему в якій вони працюють, а лише додадуть програмний пакет до існуючого хмарного програмного забезпечення і зможуть використовувати його за призначенням.

Також позитивними сторонами буде вартість програмного забезпечення та сучасність інтерфейсу, які безумовно вирізнятимуть проект серед існуючих реалізацій подібних продуктів.

Проте головною перевагою серед інших проектів, буде переосмислення процесів, покладених у модель «потрійної спіралі», а саме, створення хабу ідей та проектів на локальному рівні та рівні штатів, що дозволить краще контролювати процеси та мати звіти по будь яким даним, що будуть внесені в систему та їх категоризацію. Це дозволить уряду ще краще та оптимізованіше використовувати кошти а також прогнозувати майбутні витрати та аналізувати розвиток і процес реалізації проектів у системі.

В даному випадку автор концентрується на новизні в плані підходу до основної теми, а саме моделі «потрійної спіралі» та її осмислення.

Проект має зазнавати змін у ході спілкування з представниками усіх трьох галузей, що задіяні в процесах системи, такий підхід дасть можливість покращити існуючий додаток, внести зміни в плани розробки на наступні цикли.

На думку автора використання моделі "потрійної спіралі" в середовищі Salesforce також має великий потенціал новизни та переваг. Salesforce є платформою, спрямованою на створення та розгортання хмарних додатків для управління взаємодією з клієнтами та управління бізнес-процесами.

Новизна використання моделі "потрійної спіралі" у середовищі Salesforce полягає в створенні екосистеми співпраці, де університети, ІТ-компанії та владні структури працюють разом для створення інноваційних рішень. Це може призвести до появи нових функцій, розширення можливостей платформи та поліпшення її впровадження в різних галузях та організаціях. Крім того, такий підхід може сприяти розвитку екосистеми розробників, дослідників та інноваторів, що активно використовують платформу Salesforce.

1.2.4 Сприйняття

Впровадження проекту, заснованого на моделі "потрійної спіралі" взаємодії університетів, ІТ-компаній та владних структур, може мати різні наслідки для користувачів. Ось деякі позитивні та негативні аспекти, які вони можуть сприймати:

Позитивні сторони:

1. Інноваційні рішення: Користувачі можуть отримати доступ до новаторських продуктів та послуг, які поєднують в собі академічні знання, технологічну експертизу та регуляторну підтримку. Це може сприяти вдосконаленню існуючих процесів та наданню унікальних можливостей.
2. Поліпшена функціональність: Взаємодія між університетами, ІТ-компаніями та владними структурами може привести до розробки продуктів, що відповідають потребам користувачів. Застосування передових технологій та набуття нових знань може призвести до створення більш ефективних та інтуїтивно зрозумілих інструментів.
3. Доступ до ресурсів: Користувачі можуть мати користь від доступу до розширених ресурсів, таких як академічні дослідження, експертна думка, фінансування та підтримка владних структур. Це може підтримати їхні власні проекти, збільшити ефективність та знизити витрати на розробку програмного забезпечення для владних структур.
4. Створення системи на основі вже існуючої системи в уряді дасть можливість уряду простіше переходити на нові рішення у сфері теми

роботи, замість того, аби використовувати нове, або додаткове програмне забезпечення, чи мігрувати дані з системи в систему, уряд отримує можливість лише встановити додаток до їхньої CRM системи та використовувати за призначенням.

Негативні сторони:

1. Складність інтеграції в інші системи: Взаємодія між різними структурами може створити проблеми щодо інтеграції різних технологічних платформ, стандартів та процесів, які вже працюють десятиліттями. Це може потребувати додаткового зусилля та ресурсів для забезпечення сумісності і взаємодії між різними системами. Користувачі можуть зіткнутися зі складнощами під час налаштування та інтеграції різних компонентів, що може призвести до затримок та технічних проблем.
2. Потенційні проблеми конфіденційності та безпеки: Взаємодія між університетами, ІТ-компаніями та владними структурами може вимагати обміну конфіденційною інформацією та доступу до даних. Це може підвищити ризик витоку даних, порушення конфіденційності та збільшити загрозу кібератак. Забезпечення високого рівня захисту даних та забезпечення відповідності з регуляторними вимогами стануть важливими аспектами в реалізації такого проекту.
3. Залежність від рішень третіх сторін: У разі використання програмних продуктів, розроблених на основі "потрійної спіралі", користувачі можуть стати залежними від рішень та оновлень, які залежать від активності університетів, ІТ-компаній та владних структур. Це може вплинути на швидкість і доступність оновлень, а також на підтримку та обслуговування продуктів, але всі ці процеси мають регулюватись відповідальними особами з метою підтримання планки якості та

вчасного завершення проєктів.

4. Також, працюючи одноосібно над проєктом дуже складно оцінити потенційні ризики з усіх сторін, лише набуття більшої кількості співробітників, менеджерів, бізнес-аналістів дасть можливість більш явно оцінити ризики проєкту.

1.2.5 Повторне використання

Разом з розвитком технологій наша потенційна система повинна розвиватись в тому ж напрямку, що і основні лідери галузі, тому в майбутньому наш застосунок може стати мультиплатформенним, оскільки вже клієнтська частина застосунку є незалежною від платформи.

Системи повинні адаптуватися до постійних змін та тенденцій, оскільки старі програмні засоби стають застарілими і можуть бути витіснені більш конкурентоспроможними рішеннями. Розробники мають передбачити способи оновлення своїх продуктів, щоб не втратити свою цільову аудиторію.

Існує кілька способів розвитку проекту в концепції «потрійної спіралі»:

1. Додавання нових функцій та можливостей до системи.
2. Оновлення допоміжного коду, такого як плагіни та бібліотеки, щоб забезпечити швидшу та стабільнішу роботу.
3. Адаптація інтерфейсу до сучасного стилю та тенденцій.
4. Перехід на нові платформи, наприклад, порт на нові операційні системи.
5. Створення різних версій системи для різних середовищ, таких як веб-версії або мобільні додатки для різних платформ.

Історія вже показала, що порт системи на різні платформи може оживити забуті проекти та залучити нову аудиторію. Створення версій для мобільних пристроїв та веб-клієнтів стає стандартом для великих систем, крім цього можна додавати інтеграції з іншими існуючими системами, це покращить як і використання так і перехід на систему.

Якщо проект не передбачає оновлення та розширення, варто звернути на це увагу і задуматись про можливості покращення у майбутньому. Інакше існує ризик того, що проект буде забутий, не розкриє свій потенціал, а

розробники не отримують бажаних результатів. Тому все має розвиватись покроково, аби корегувати рух у розвитку проекту в правильному напрямку, в більш вірогідному напрямку для отримання позитивних результатів під час розробки проекту та реалізації основного потенціалу.

1.2.6 Основа проекту

Проект можна починати повністю з нуля, а можна створювати на основі вже наявних наробіток. Як було зазначено у попередньому пункті, значна частина коду повторюється від одного проекту до іншого і повторювану функціональність є сенс виділяти в окремі модулі, звідки слідує, що для створення нового проекту було б корисно використати вже наявний код. Що більший відсоток коду можна буде взяти зі сторонніх рішень, то менше часу доведеться витратити на написання одних і тих же функцій, і відповідно, тим швидше можна буде перейти до експлуатації.

Наразі майже жоден серйозний проект не створюється без участі інших бібліотек або фреймворків. Поширення відкритого коду і його багаторазове використання в різних системах є основою ідеології відкритого коду (open source), проте використовуючи такий код у своїх роботах, варто звертати увагу на ліцензійні та авторські питання. Відкритість коду не завжди означає можливість його безоплатного використання, що вже було досліджено в одному з попередніх пунктів, утім якщо розробник упевнений, що інструментарій (за умови, що він існує), який він планує використати, підходить для проекту, це є перевагою, яку варто враховувати при аналізі.

Окремо стоїть питання якості сторонніх бібліотек чи рішень, які ми включаємо в свій проект, в першу чергу важлива наявність гарної документації, відсутність якої, може навпаки сповільнити робочий процес або ж взагалі призвести до необхідності пошуку нових сторонніх рішень, із чим на практиці розробники постійно зустрічаються, наприклад через те, що розробники бібліотек чи фреймворків перестають підтримувати свої розробки.

В даному випадку автор створює своє рішення на базі PaaS платформи та фронтенд бібліотек, що дає змогу побудувати сервіс на основі нових та в той же час гарно відтестованих технологіях, що гарантуватиме стабільність роботи, оптимізацію та надійність кожного з компонентів клієнт-серверного застосунку.

1.2.7 Часові межі

Кожному проекту необхідно встановити терміни та часові рамки його реалізації. У випадку довгострокових або циклічних проектів, потрібно визначити приблизні межі для введення першої версії в експлуатацію. Часові рамки є надважливим фактором при аналізі проекту. Іноді варто зробити MVP, запустити його в production і почати отримувати фідбек від клієнтів, ніж витратити ресурси на роки розробки програмного застосунку, а лише після цього дізнатись, що даний програмний продукт нікому не потрібен. Швидкість сучасної розробки змушує програмістів працювати швидше, оскільки запізнення з випуском продукту на ринок може призвести до появи конкурентних рішень, що будуть більш привабливими, тому завжди потрібно тримати руку на пульсі.

Інший сценарій, але з аналогічним результатом, це коли проект пов'язаний з певною подією, після якої сама програма стає неактуальною. Наприклад, під час війни дуже актуальним є створення різних програмних

застосунків, або різних технологічних рішень, пов'язаних з безпілотними апаратами, де розробники мають розумітись як на апаратній частині так і програмній, також мати знання і бажано досвід того, як працювати з протоколами захищеного зв'язку.

Часові обмеження є важливим фактором, який покладає обмеження на проект, оскільки недотримання їх зменшує цінність і актуальність розробки і може призвести до появи конкурентних аналогів, які будуть швидше завойовувати ринок. Зазвичай, розробка менше року вважається коротким терміном, від одного до двох років - середнім, а більше двох років - довгим. Очевидно, що чим швидше виконується розробка, тим більше шансів на успіх проекту.

Передбачити, чи проект окупить себе наперед неможливо, тому оцінка очікуваного терміну окупності є важливим завданням під час планування. Чим коротший термін окупності, тим менше ризиків пов'язаних з розробкою проекту. Наприклад, команда розробляє інтернет-магазин з мінімальними рекламними витратами. Так само менше часу на розробку дозволить скоротити витрати, що пов'язані безпосередньо з розробкою застосунку.

Цей проект є типовим і може бути реалізований швидко, використовуючи готовий шаблон і завершуючи роботу над серверною частиною. Проте, після запуску проходить певний час, перш ніж клієнти почнуть ставати партнерами системи і користуватись нею по місячній підписці. Без активної реклами популяризація проекту буде повільною, а розробники продовжуватимуть отримувати зарплату та оплачувати витрати на підтримку проекту, та його оновлень.

Якщо активність покупців триватиме занадто довго, це може призвести до невдачі проекту і накопичення зобов'язань. Для переконання у вигідності

термінів окупності, потрібно звернути увагу на наступні фактори:

1. Що впливає на прибуток продукту (активність користувачів, платні покупки, перегляд реклами, кількість користувачів за платною підпискою тощо).
2. Чи залежить проект від тимчасової події (програми, пов'язані з одноразовими або періодичними подіями, які обмежені у часі, або проект, що актуальний лише сезонно).

Чим коротший термін розробки, до першого MVP, який вже зможе приносити перші прибутки, тим більше шансів на успіх в нього є.

1.2.8 Прибутковість

Проект, заснований на моделі "потрійної спіралі" (також відомий як модель розробки продукту), може бути прибутковим за умови правильної реалізації і ефективного впровадження. Основна ідея цієї моделі полягає в поєднанні ітераційного циклу розробки, залучення користувачів та накопичення знань та досвіду.

Основні кроки моделі "потрійної спіралі" включають:

1. Формулювання цілей і вимог: Визначення чітких цілей проекту та вимог, які мають суттєво відрізнити цей продукт від продукту конкурента та які повинен задовольняти продукт в цілому. Це включає аналіз потреб користувачів і визначення вимог до функціональності, продуктивності, безпеки тощо.
2. Оцінка альтернативних рішень: В цьому етапі розглядаються різні підходи і технології для реалізації проекту. Розробляються прототипи різних рішень, які потенційно на етапі тестування можуть показати себе набагато краще в плані живучості концепції та після цього проводяться

оцінки для вибору оптимального рішення з погляду вартості, термінів, технічних можливостей тощо.

3. Реалізація та тестування: Наступним етапом є етап програмування, розробка та перенесення продукту в production environment. Також проводиться тестування для перевірки функціональності, надійності та відповідності вимогам.
4. Оцінка ризиків: Здійснюється оцінка можливих ризиків та небезпек, пов'язаних з проектом, і приймаються заходи для їх запобігання. Це допомагає зменшити вплив негативних факторів і забезпечити стабільну реалізацію проекту.

Проект автора в загальному тісно пов'язаний з урядовими установами, які є найвищим щабелем управління у моделі «потрійної спіралі», тому успіх залежатиме в більшості від того, наскільки місцевий уряд кожного з штатів США буде готовий для співробітництва та взаємного розвитку проекту, на що, і є орієнтир в першу чергу для автора, хоча маючи локалізацію і для інших країн, можна запроваджувати такий самий застосунок в різних куточках світу.

1.2.9 Досвід

Швидкість виконання проекту, що базується на моделі "потрійної спіралі", прямо залежить від організованості команди, яка, у свою чергу, залежить від досвіду керівників проекту у подібних проектах. Незважаючи на неможливість передбачити всі можливі проблеми, які можуть виникнути під час розробки, завжди є фактор випадковості.

Якщо проблема є новою, потрібно знову шукати рішення, що може сповільнити роботу. Однак повторне вирішення задачі пришвидшує роботу, оскільки команда знає, де шукати рішення. Досвід може бути особистим або

набутим від інших. Чужий досвід також може бути корисним, особливо якщо проблема вже виникала у багатьох і була вирішена.

Вибір технологій для проекту також важливий. Популярні технології, які використовуються широкою спільнотою розробників, мають більшу ймовірність мати вже існуючі рішення для проблем, що виникають. Досвід не обмежується тільки технічними аспектами, а й навченістю команди, що повинна мати досвід та теоритечну базу, пов'язану з інструментарієм, який буде використовуватись у даному проекті.

Організаційні аспекти, такі як вибір методології управління проектом і підхід до комунікації з командою, також впливають на ефективність та продуктивність. Чим більше досвіду у керівництва проекту з організації робочого процесу, тим краще співпрацює команда, і тим більше ймовірність завершити проект вчасно.

Крім того, наявність досвідчених спеціалістів є ключовим моментом для успішної реалізації ідеї. Важливо враховувати навички команди при виборі технологій. Іноді вибір більш застарілого, але знайомого команді інструменту може бути кориснішим, оскільки новіші технології можуть вимагати часу на вивчення і тестування і можуть бути не стабільними у використанні, або бути несумісними з іншими технологіями, які використовуються у проекті.

Загалом, досвід технічних та не-технічних аспектів розробки є важливим для успіху проекту. Він охоплює знання, аналітичні навички, здатність до спілкування та співпраці з командою. Без наявності досвідчених фахівців створення якісного продукту і підтримка його стають складними.

1.2.10 Фінанси

Однією з дуже важливих основних пунктів у розробці проекту є фінансова вигода, яку потенційно можуть отримати власники та розробники від реалізації програмного застосунку та успішного його просування. Адже робота повинна мати фінансове винагородження, оскільки наш проект є комерційним, існуючий бюджет має бути правильно розподілений на частини, зазвичай половина бюджету йде на маркетинг а інша половина на розробку, підтримку та адміністрування проекту під час розробки та запуску проекту.

Проект, навіть з MVP версії потрібно починати проектувати якісно, плануючи архітектуру заздалегідь, оскільки технічний борг пізніше виправляти набагато дорожче ніж, продовжувати слідувати правильно побудованим патернам. Якісно виконану роботу, як правило, роблять повільно, а нашвидкуруч зроблений проект якісним навряд чи буде. Проту, за наявності досвідчених спеціалістів можна виконувати проект і швидко і достатньо якісно.

Після того, як певний час проект буде утримуватись за рахунок команди, що працює над цим проектом, мають з'явитись перші клієнти за рахунок результатів маркетингової компанії, яка буде запущена після релізу першої версії проекту. Якщо коштів на реалізацію проекту автора недостатньо, то можна залучити кошти інвесторів, або через краудфаундинг, проте не для всіх моделей майбутньої монетизації проекту це можливо.

В даному випадку нашими клієнтами є великі інституції, і отримати співробітництво з ними, або отримати певні владні структури, як своїх клієнтів може бути досить складно, тому ми маємо сконцентруватись на тому, аби виробити максимально зручну систему для керування процесами моделі «потрійної спіралі», аби отримати грант, чи бюджет на реалізацію даного застосунку.

Ціна застосунку, запланованого автором, буде сталою, оскільки основними клієнтами є представники владних структур, і ми визначемо ціну за користувача чи організацію в системі і від цього залежатиме дохідність проекту за певний період часу (підписка на сервіс).

При цьому варто не забувати про те, що існуючий проект може бути повністю профінансованим владними структурами, якщо вдасться дійти певних домовленостей, і в цьому випадку для такого роду проекту це було дуже хороше співробітництво, оскільки певна організація зможе мати вплив на розробку та в кінці розробки отримати максимально підходячий проект під задачі цієї організації, при цьому розробники матимуть постійні кошти на розробку та покращення застосунку.

Для успішного продукту важливо мати конкурентоспроможну ціну. Якщо на ринку є аналог з подібними можливостями, але з нижчою ціною, клієнти можуть вибрати його. Ціна продукту визначається витратами на його розробку, існує мінімальна вартість, яку можна встановити. При визначенні ціни важливо враховувати конкурентну ситуацію на ринку і порівнювати її з цінами конкурентів, щоб переконатися, що встановлена ціна є конкурентноспроможною для клієнтів.

1.2.11 Інші засоби реалізації

Зазвичай, реалізація проекту може різнитись від того, в якій предметній області знаходиться проект, над яким має працювати команда. В даному випадку ми не маємо альтернативної реалізації даного програмного продукту у іншому вигляді, оскільки наше завдання – це автоматизувати співробітництво владних структур, університетів та ІТ-компаній.

Звісно, в сучасному світі великими компаніями вже налагоджені шляхи взаємодії між цими трьома сутностями нашої бізнес моделі. Проте, автор пропонує іншу концепцію, де ми не будемо тісно зв'язані з конкретними компаніями, чи державними інституціями, а створимо платформу, що буде адаптована для розширення в майбутньому, та буде мати функціонал, пов'язаний з основними бізнес-процесами, що мають реалізовувати цикл від запиту на вирішення проблеми і до повної реалізації проекту за бюджет країни, чи штату, чи окремо взятого міста.

Також, в залежності від предметної області проекту, розробники мають добре розумітись на проблематиці певної області, розбиратись в документації та процесах, за якими вже працюють сфери, задіяні в цій бізнес моделі «потрійної спіралі».

1.2.12 Ризики

При правильному ризик-менеджменті ми маємо визначити певні характеристики, за якими ми можемо вважати успішним. Крім цього, ми можемо визначити декілька рівнів успіху, або так само рівнів того, наскільки проект може бути провальним.

Також, під час розробки ми будемо вносити певні корективи, і це впливатиме на те, чи буде проект успішним не менш, ніж початкове технічне завдання.

Хоча невиконання всіх елементів плану проекту не є катастрофічним, будь-яке відхилення від плану призводить до отримання іншого продукту, ніж було задумано спочатку. Це означає, що сприйняття продукту його потенційними користувачами буде відрізнятись, і це може нашкодити нашим планам задовольнити їх очікування. Це може призвести до зниження продажів, недосягнення повернення інвестицій та можливих фінансових втрат.

1.3 Висновки до частини 1

Автором було досліджено можливості різні теоретичні аспекти того, якими шляхами можливо реалізувати технічно проект. Було розглянуто різні технічні деталі, деталі стосовно предметної області, та ми намагались дати потенційну оцінку проекту на успіх, розрахувати вигоду кожної зі сторін моделі «потрійної спіралі» та дати оцінку можливості реалізувати проект.

Було розглянуто формальні підходи до визначення поняття проекту та оцінки ризиків. Також було визначено технології, якими буде найбільш оптимально реалізовувати даний проект.

У кінці було досліджено проблематику, ризики та інші явища, та запропоновано шляхи їх усунення. Було визначено бізнес-процеси, над якими працюватиме автор. Також було оцінено конкурентів на ринку, та запропоноване універсальне програмне рішення для усіх сфер діяльності, які є частиною нашої моделі «потрійної спіралі».

2. Реалізація

2.1 Розробка сайту

2.1.1 Основний інструментарій

Наш програмний застосунок, являтиме собою веб-сервіс, базований на PaaS Salesforce CRM, який ми будемо використовувати як бекенд і базу даних. При цьому різні ролі наших користувачів матимуть різний інтерфейс з різними можливостями всередині.

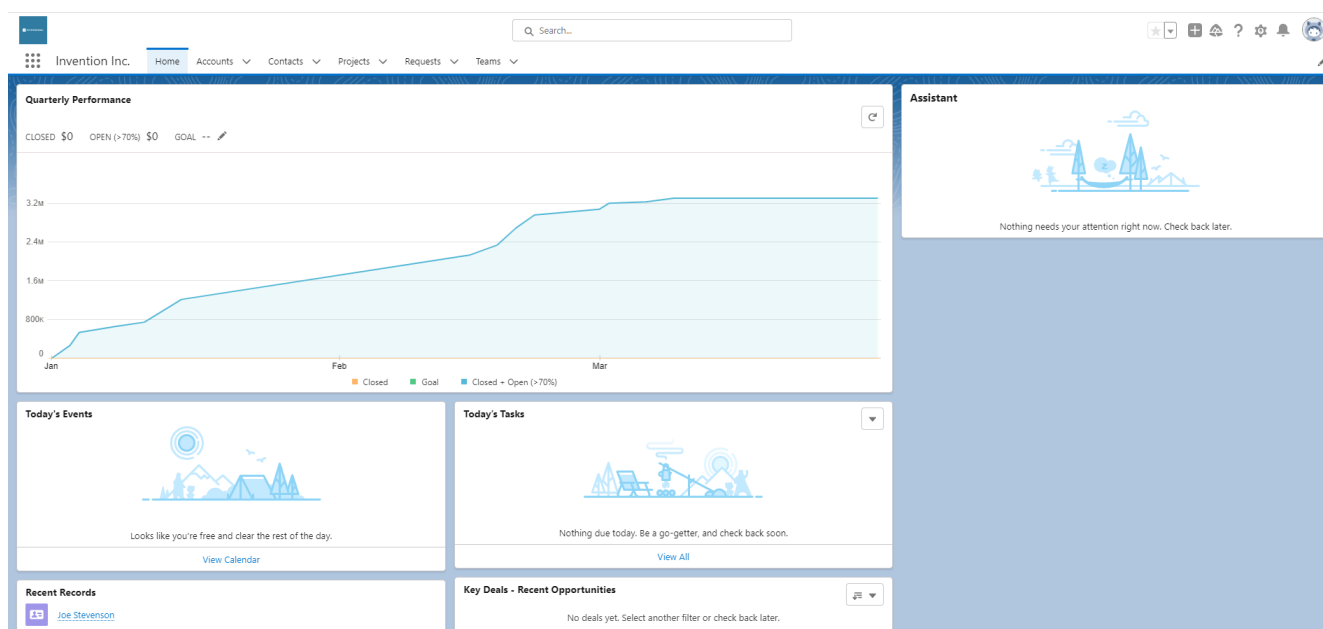


Рисунок 2.1.1.1 — приклад інтерфейсу CRM

У нашій системі адміністраторами виступають саме представники владних структур, які мають повний доступ до системи, та адмініструють дані та усіх користувачів з клієнтської частини яку надає Salesforce.

У нашій системі, адміністратори мають можливість створювати запити до проектів, створювати акаунти партнерів, тобто університетів, або ІТ-компаній. Також управляти даними по проектах, по їх версіях, та будуть мати можливості створювати різні автоматизації, такі як блокування певних записів для підтвердження менеджером, або нагадування про певні події за допомогою Email Alerts, чи автоматизація підрахунку даних за допомогою Flows, чи Apex Triggers.

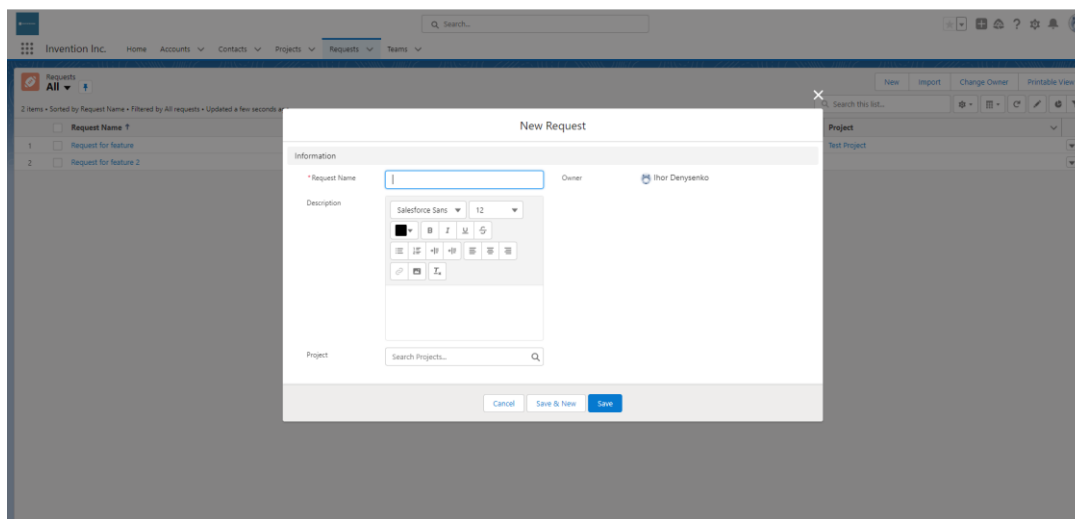


Рисунок 2.1.1.2 — приклад створення запиту від представника владних структур на вирішення певної проблеми громадськості

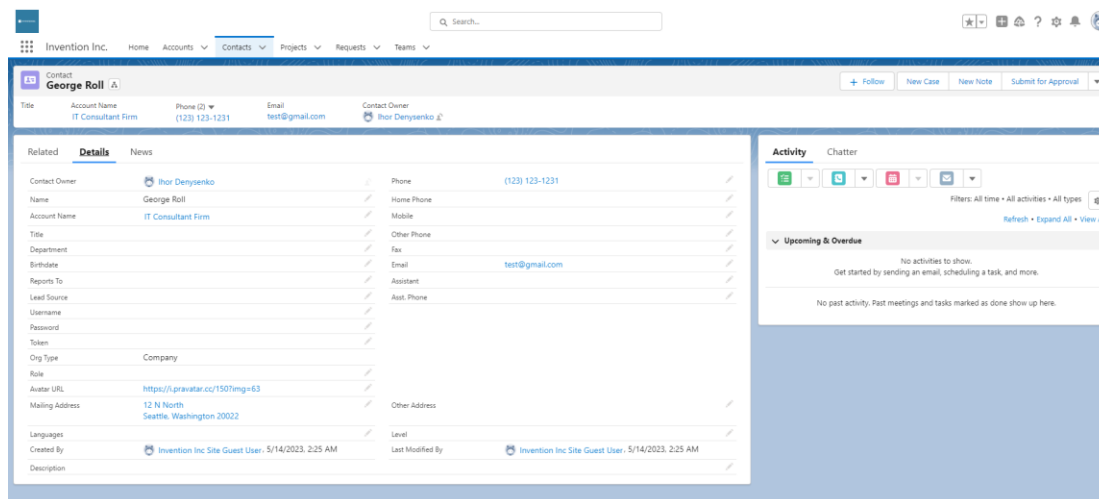


Рисунок 2.1.1.3 — приклад даних про представника ІТ-компанії, що зберігаються в системі

2.1.1 Розміщення проекту

Наш проект має бути доступним для придбання та встановлення потенційними користувачами і тут платформа Salesforce надає нам великі переваги в цьому. Оскільки ми маємо можливість зібрати Managed Package і розмістити його на AppExchange, що дасть нам можливість продавати наш додаток до платформи усім охочим, хто має доступ до платформи Salesforce. Тому будь яка урядова установа, що вже використовує, або планує використовувати Salesforce, зможе придбати за місячною підпискою наш сервіс і використовувати його за призначенням.

Сторінка для представників університетів та ІТ-компаній розміщується окремо, а саме на доменах Salesforce Sites, для цього автор зробив фронтенд частину проекту на фреймворці React.js, та приєднав зібраний проект на Salesforce, це дає можливість зробити хороший custom дизайн для проекту та приємний інтерфейс для кінцевих користувачів.

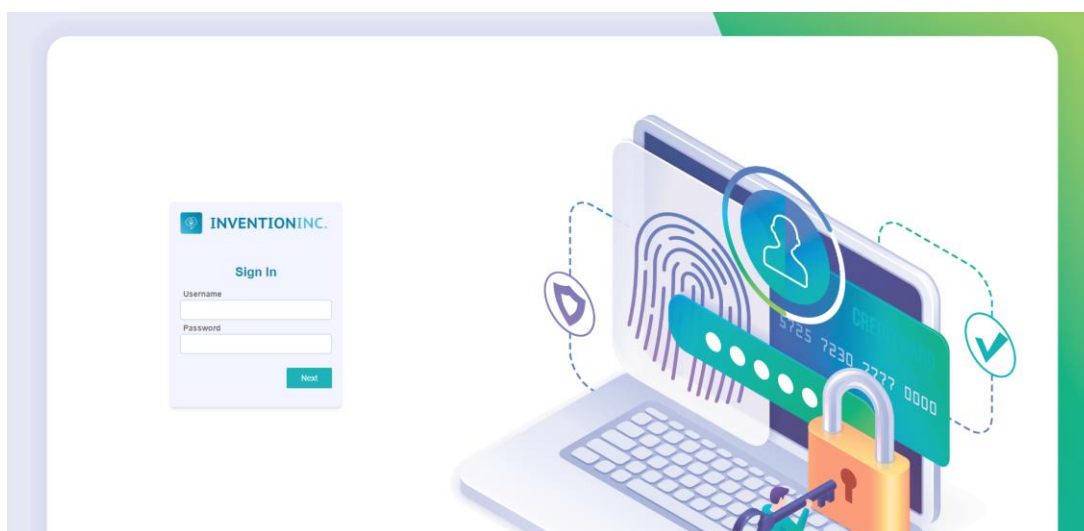


Рисунок 2.1.1.4 — авторизація для користувачів від університетів та представників владних структур

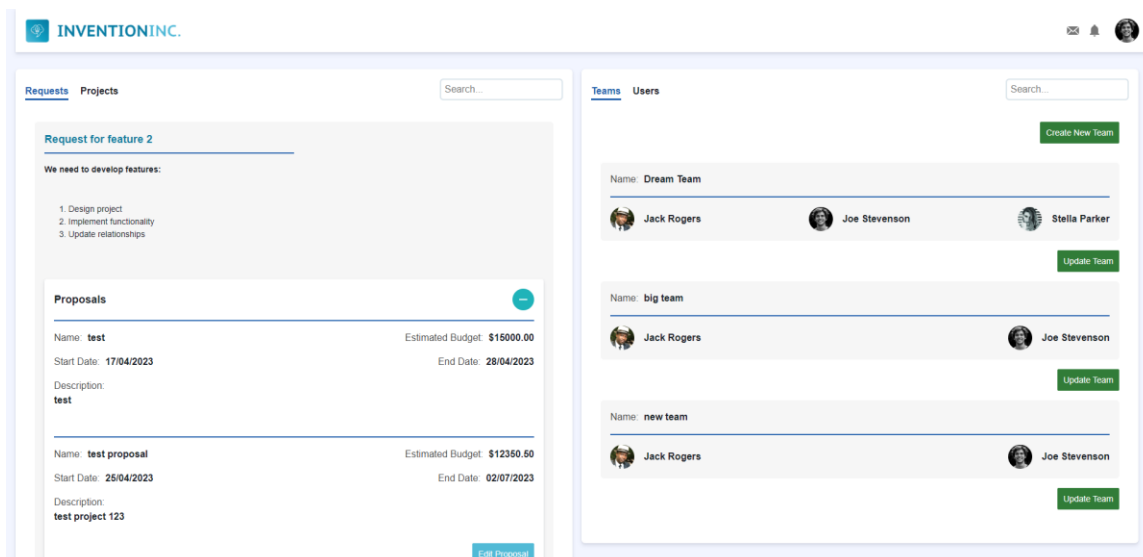


Рисунок 2.1.1.5 — приклад того, як виглядає система для користувачів з університетів та ІТ-компаній

2.2 База проекту - backend

Серверна частина написана мовою від Salesforce, що дуже схожа на Java, а саме Apex, також цією мовою реалізовані тригери до бази даних, яка використовується цією платформою.

Salesforce використовує базу даних Oracle і ми можемо діставати дані за допомогою SOQL запитів, що є аналогом SQL але в контексті платформи.

Також усі інші операції за даними реалізовані за допомогою DML-запитів, Salesforce підтримує наступні запити: insert, upsert, update, delete, undelete.

```

1 public class SiteController {
2
3     @RemoteAction
4     public static Map<String, Object> signIn(String username, String password) {
5         Map<String, Object> status = new Map<String, Object> { '__status' => 'error' };
6
7         try {
8             List<Contact> contacts = signInController.getContactsByCredentials(username, password);
9
10            if (!contacts.isEmpty()) {
11
12                String token = signInController.generateToken();
13
14                contacts[0].Token__c = token;
15
16                update contacts;
17
18                status.put('token', contacts[0].Token__c);
19                status.put('__status', 'success');
20
21            } else {
22                status.put('__msg', 'wrong credentials. ');
23            }
24
25        } catch (Exception e) {
26            System.debug(e.getMessage());
27        }
28
29        return status;
30    }
31
32     @RemoteAction

```

Рисунок 2.1.1.6 — приклад серверного коду, а саме функція логіну для користувачів

```
public static void updateProposal(ContactDAO contact, Map<String, Object> data) {  
  
    Proposal__c proposal = ProposalSelector.getProposalById((String) data.get('id'));  
  
    if (proposal != null && proposal.Created_By__c == contact.accountId) {  
  
        proposal.Name = (String) data.get('name');  
        proposal.Description__c = (String) data.get('description');  
        proposal.Estimated_Budget__c = (Decimal) data.get('estimatedBudget');  
  
        if ((String) data.get('startDate') != null) {  
            proposal.Start_Date__c = Date.valueOf((String) data.get('startDate'));  
        }  
  
        if ((String) data.get('endDate') != null) {  
            proposal.End_Date__c = Date.valueOf((String) data.get('endDate'));  
        }  
  
        update proposal;  
    }  
}
```

Рисунок 2.1.1.7 — приклад серверного коду для функції оновлення створеної пропозиції

2.3 База проекту - frontend

Клієнтська частина проекту створена за допомогою компонентно-орієнтовного фреймворку React.js, та бібліотеки UI-компонентів blueprint.js.

Зроблена якісна декомпозиція порталу на маленькі компоненти, які виконують свої функції.

На клієнтській частині ми реалізували можливість користувачів управління своїми командами, створення версій до проектів та створення пропозицій власних рішень до владних структур, також реалізований функціонал по управлінню профілями користувачів.

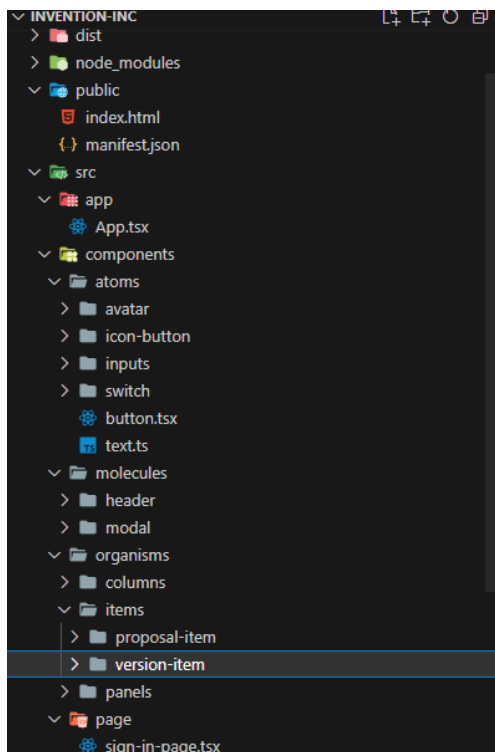


Рисунок 2.1.1.8 — структура проекту на React.js


```
interface SignInPageProps {
  onSignIn: (token: string) => void;
}

export function SignInPage({ onSignIn }: SignInPageProps) {

  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  return (
    <SignInContainer>
      <SignInForm>
        <Title src={LOGO_SRC} style={{ marginBottom: "50px" }} />

        <div style={{ textAlign: "center", marginBottom: "32px", color: "#0a85ab", fontSize: "22px", fontWeight: 600}}>Sign In</div>

        <div style={{ marginBottom: "24px" }}>
          <LabeledInput value={username} placeholder="" label="Username" onChange={setUsername} />
        </div>
        <div style={{ marginBottom: "24px" }}>
          <LabeledInput value={password} placeholder="" label="Password" onChange={setPassword} />
        </div>

        <div style={{ display: "flex", alignItems: "center", justifyContent: "flex-end" }}>
          <Button label="Next" buttonIntent="primary" onClick={async () => {
            const response: SignInDAO = await signIn(username, password) as SignInDAO;

            if (response.__status === "success") {
              onSignIn(response.token);
            } else {
              alert(response.__msg);
            }
          }} />
        </div>
      </SignInForm>
    </SignInContainer>
  );
}
```

Рисунок 2.1.1.9 — приклад функціонального компонента сторінки авторизації користувачів

2.4 Висновки до частини 2

Ми розглянули реалізацію сервісу з використанням сучасних фронтенд бібліотек, та серверної частини на базі платформи Salesforce, що дає нам багато цікавих та корисних шляхів у реалізації цього проекту як бізнесу.

Автор надав пояснення до використання саме цих інструментів та пояснив структуру його окремих частин та їх взаємодії, також було проведено повноцінне тестування усіх компонентів системи та бізнес-процесів.

Проект вже знаходиться на платформі PaaS Salesforce і вже знаходиться в мережі, тому демонстраційна версія проекту автора вже готова до публічного тестування при створенні профілів користувачів для потенційних клієнтів.

2. Практичне використання

2.1 Інтерфейс та основний бізнес-процес

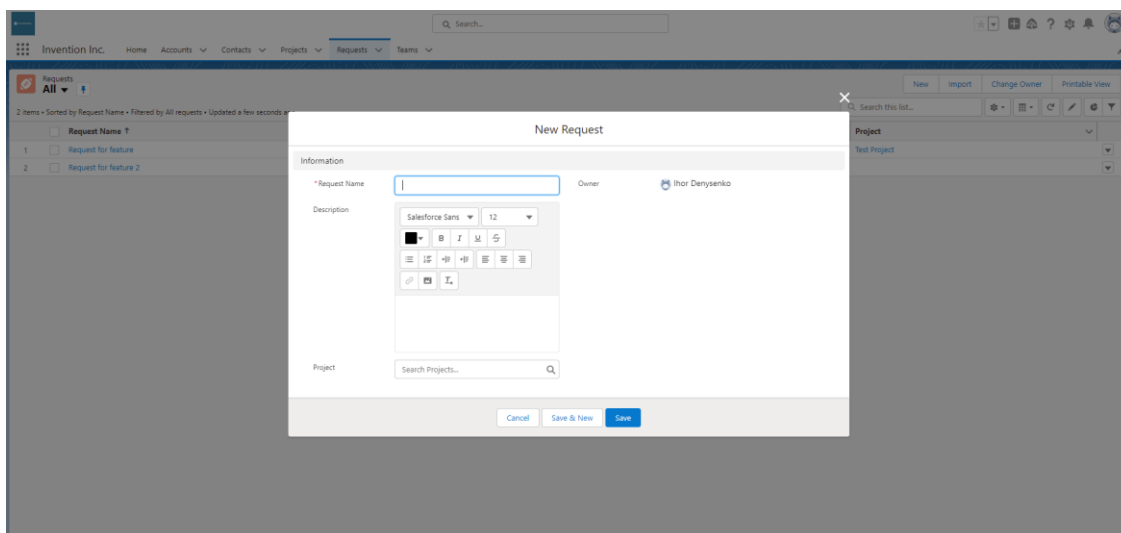


Рисунок 3.1.1.1 — сторінка створення запиту від уряду

Уряд створює запит зі свого адмінського порталу, після цього представники від університетів мають авторизуватись в системі та взяти участь у тендері.

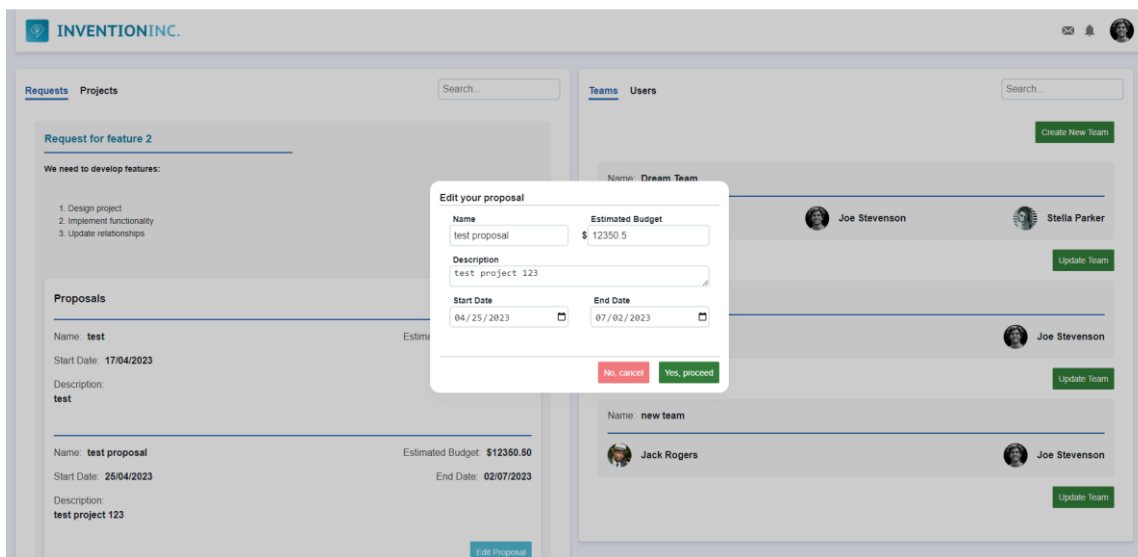


Рисунок 3.1.1.2 — інтерфейс та модальне вікно для користувачів від університетів, де вони створюють пропозицію вирішення проблеми у місті, штаті чи країні

Після цього адміністратор системи обирає найкраще рішення із запропонованих університетами і створює з нього проект.

Після створення проекту, адмін від університету має змогу побачити проект в системі, над яким він працює.

Також адмін від університету має змогу створювати версії проекту, аби усі користувачі мали змогу бачити прогрес у суспільних проектах від уряду, також обирати консультанта з ІТ-компанії, який буде допомагати у розробці рішення та консультувати команду з університету.

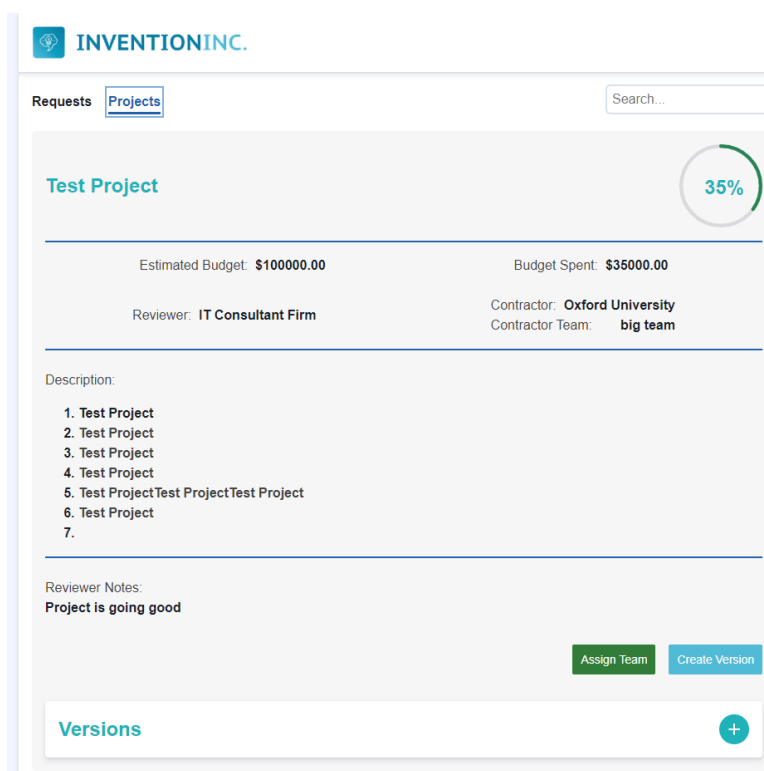


Рисунок 3.1.1.3 — інтерфейс для користувача університету, який бачить проекти, над яким працює його університет

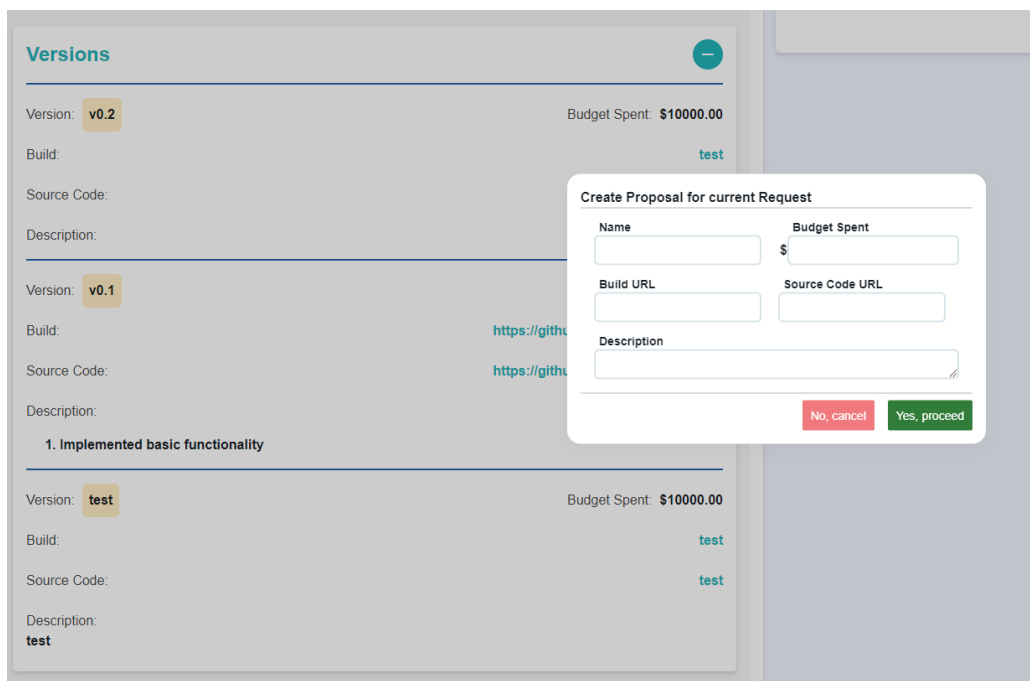


Рисунок 3.1.1.4 — модальне вікно, де після завершення роботи на новою версією проекту, користувач може занести посилання на код GitHub та посилання на готовий build проекту і вказати скільки коштів було витрачено

Також після витрачення усього бюджету, представники університету мають або показати повністю готове рішення, або скомунікуватись з представниками владних структур для уточнення подальших дій.

Також представники ІТ-компаній або представники з університетів мають можливість управляти командами та користувачами їх організацій.

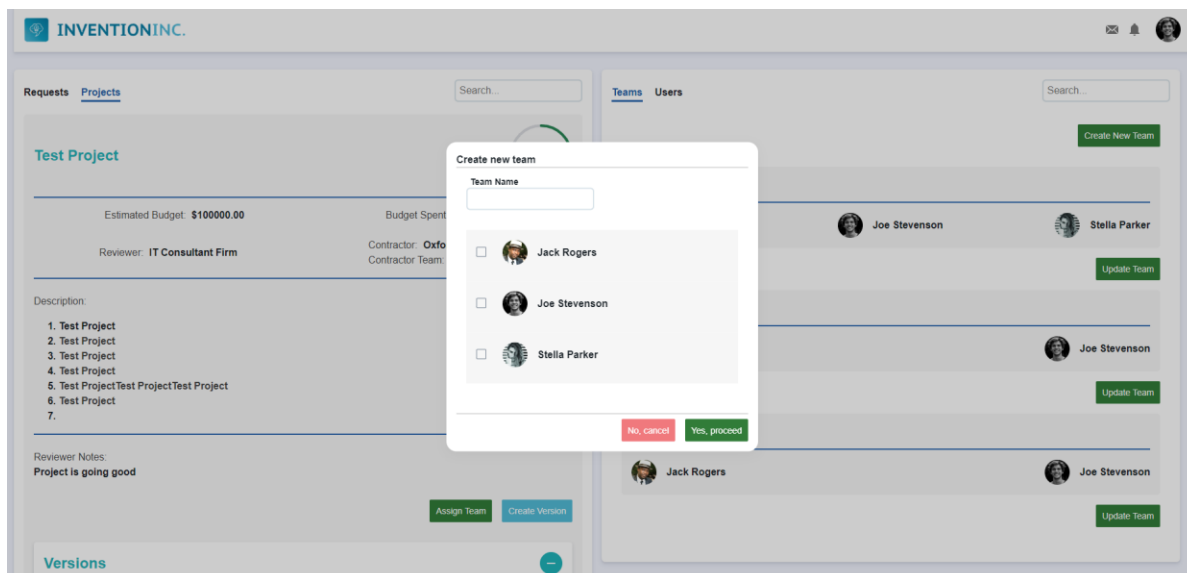


Рисунок 3.1.1.5 — модальне вікно створення нової команди

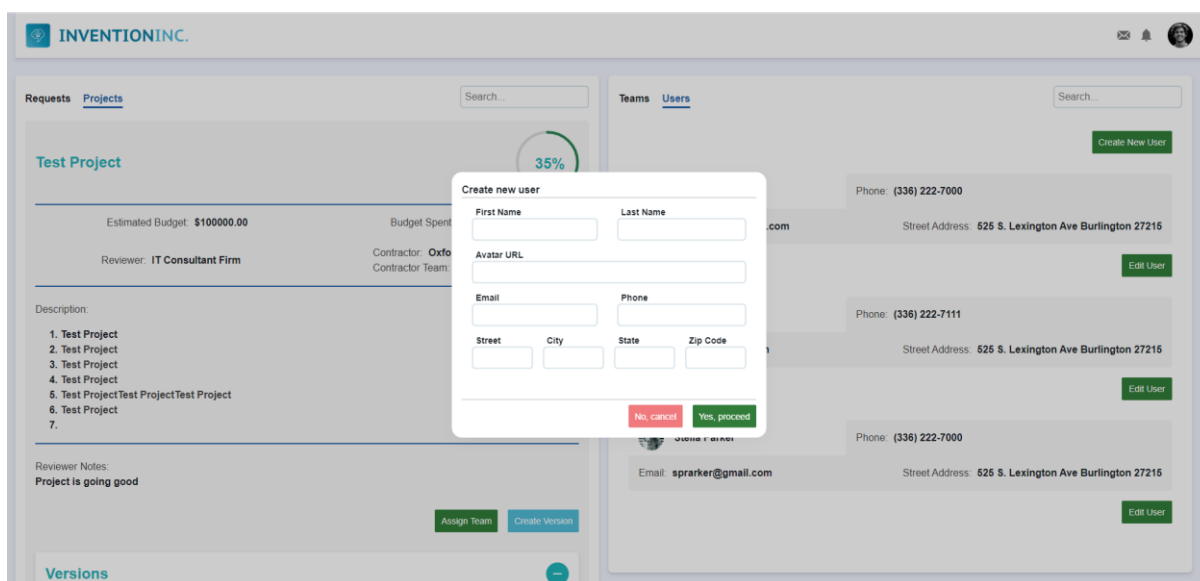


Рисунок 3.1.1.6 — модальне вікно створення нового користувача

3.2 Висновки до частини 3

Ми розглянули усі типи інтерфейсів для усіх видів користувачів, ми реалізували усі заплановані функції для демо-версії продукту. Ми маємо повний цикл розробки програмного забезпечення, від тендеру і до впровадження проекту в життя.

Було розглянуто також процес використання цього сервісу. Було показано увесь наявний функціонал там boiler-plates для функцій, що будуть додані у наступних версіях додатку. Сервіс побудований таким чином, щоб для усіх працівників сфер, що пов'язані з «потрійною спіраллю», зроблено максимально зручно на даний момент, була проведена робота по покращенню UX-дизайну для користувачів, аби спросити усі основні процеси та зробити їх інтуїтивно зрозумілими.

Висновки

У теоретичній частині роботи було усі аспекти побудови програмного забезпечення, базованого на моделі «потрійної спіралі», тобто усі можливі нюанси щодо цілей проекту, видів його реалізації та перспектив, було запропоновано рішення даної задачі.

Практична частина описує архітектуру клієнт-серверного застосунку, заглиблюючись в архітектурні рішення, прийняті у реалізації даного програмного продукту, та які були використані технології для досягнення мети.

В кінці теоретичної частини ми показали готове рішення, описали покроково увесь бізнес-процес, реалізований для спрощення та уніфікації взаємодії ІТ-компаній, університетів та владних структур у досягненні нових наукових, спільних рішень для реалізації громадських проектів.

Список використаних джерел

1. General FAQ documentation for Salesforce [Електронний ресурс] — Режим доступу до ресурсу: <https://salesforcefaqs.com/>

2. A JavaScript framework[Електронний ресурс] — Режим доступу до ресурсу: <https://reactjs.org/>

3. A UI library for React.js framework [Електронний ресурс] — Режим доступу до ресурсу: <https://blueprintjs.com/docs/>

4. A developer community forum for Salesforce development [Електронний ресурс] —Режим доступу до ресурсу: <https://salesforce.stackexchange.com/>

5. The Triple Helix Model and the Future of Innovation: A Reflection on the Triple Helix Research Agenda [Електронний ресурс] —Режим доступу до ресурсу: https://brill.com/view/journals/thj/8/2/article-p217_1.xml?language=en/

6. THE TRIPLE HELIX MODEL FOR INNOVATION: A HOLISTIC EXPLORATION OF BARRIERS AND ENABLERS [Електронний ресурс] —Режим доступу до ресурсу: <https://uwe-repository.worktribe.com/preview/840429/THE%20TRIPLE%20HELIX%20MODEL%20FOR%20INNOVATION%20A%20HOLISTIC%20EXPLORA>

7. Corporate training ecosystem for the development of competencies and capabilities of employees. [Электронный ресурс] —Режим доступа до ресурсу: <https://www.softserveinc.com/uk-ua/university>