

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

« ___ » _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Програмне забезпечення
інформаційно-комунікаційних систем»
спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Застосунок для керування проєктами»

Виконала:

студентка IV курсу, групи ІТ-61

Гаршіна Ксенія Антонівна _____

Керівник:

Ст. викладач каф. АУТС,

Тимофєєва Юлія Сергіївна _____

Рецензент:

Доц. каф. ОТ,

Волокита Артем Миколайович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.
Студентка _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматичного управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

ЗАВДАННЯ

на дипломний проєкт студентці

Гаршій Ксенії Антонівні

1. Тема проєкту «Застосунок для керування проєктами», керівник проєкту Тимофєєва Юлія Сергіївна, ст. викладач каф. АУТС, затверджені наказом по університету від «7» травня 2020 р. №1081-с
2. Термін подання студентом проєкту «9» червня 2020 року
3. Вихідні дані до проєкту: керування проєктами на початковому, основному та кінцевому етапах організації з використанням засобів: Канбан-дошка, SWOT-аналіз та Team Canvas
4. Зміст пояснювальної записки
 - 1) Загальні положення: опис предметного середовища, огляд наявних аналогів
 - 2) Опис функціональності застосунку: вхідні дані, вихідні дані, опис структури бази даних, опис процесу діяльності, діаграма використання
 - 3) Програмне та технічне забезпечення: засоби розробки, архітектура програмного забезпечення

4) Розробка компонентів керування проектами: Канбан-дошка, SWOT-аналіз, Team Canvas, оцінка організації проекту, бухгалтерія, вхід до застосунку

5) Огляд застосунку зі сторони користувача: керівництво користувача, тестування застосунку

5. Перелік графічного матеріалу

- 1) Діаграма бази даних
- 2) Діаграма використання
- 3) Діаграма класів
- 4) Діаграма компонентів

6. Дата видачі завдання «7» лютого 2020 року

Календарний План

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	10.02.2020	
2.	Аналіз існуючих методів розв'язання задачі	18.02.2020	
3.	Постановка та формалізація задачі	24.02.2020	
4.	Розробка інформаційного забезпечення	2.03.2020	
5.	Алгоритмізація задачі	16.03.2020	
6.	Обґрунтування використовуваних технічних засобів	23.03.2020	
7.	Розробка програмного забезпечення	20.04.2020	
8.	Налагодження програми	11.05.2020	
9.	Виконання графічних документів	15.05.2020	
10.	Оформлення пояснювальної записки	22.05.2020	
11.	Подання ДП на основний захист	9.06.2019	

Студентка

Ксенія ГАРШІНА

Керівник

Юлія ТИМОФЄЄВА

АНОТАЦІЯ

Гаршіна К.А. Застосунок для керування проєктами. КПІ ім. Ігоря Сікорського, Київ, 2020.

Проєкт містить 61 с. тексту, 17 рисунків, 1 таблиця, посилання на 18 літературних джерел, додатки та 4 конструкторських документів.

Ключові слова: керування проєктами, Канбан-дошка, SWOT-аналіз, засіб Team Canvas, логістика.

Об'єктом розробки є застосунок для керування проєктами.

Мета розробки — оптимізація процесу організації великих проєктів.

У дипломному проєкті розроблено застосунок для керування проєктами, а саме такі його компоненти як: початковий, основний та кінцевий етап організації. До початкового етапу входять такі засоби керування проєктами: Канбан-дошка, SWOT-аналіз та засіб Team Canvas. До основного етапу відносять ведення логістики та ведення звітності членами команди. До кінцевого етапу входять такі компоненти як: загальна оцінка організації проєкту та заповнення форми передачі знань. Ці засоби було реалізовано у формі зручного для користування інтерфейсу користувача. Застосунок призначений для користування членами команди та головними організаторами проєктів різного типу організацій. Значну увагу було приділено аналізу та вибору основних засобів керування проєктами, розробки зручного інтерфейсу користувача та розробки бази даних застосунку.

Отримані результати можуть бути корисними волонтерським, неприбутковим або іншим організаціям, а також користувачам для особистого використання.

SUMMARY

Harshina K.A. Project management application. Igor Sikorsky KPI, Kyiv, 2020.

The project contains 61 pages of text, 17 figures, 1 table, links to 18 literary sources, annexes and 4 design documents.

Keywords: project management, Kanban-board, SWOT-analysis, Team Canvas, logistics.

The object of development is an application for managing projects.

The purpose of the development — optimization of the process of organizing large projects.

In the scope of the graduation project was developed an application for managing projects, more specifically components such as: beginning, middle and final stages of project organization. Beginning stage contains these methods for project management: Kanban-board, SWOT-analysis and Team Canvas. Middle stage contains project logistics and team reports. Final stage contains such methods as overall project evaluation and knowledge transfer. These methods were realized in the form of an easy to use user interface. Application is suited for the usage of team members and head organizers for different types of organizations. Much attention was given to the analysis and selection of the main project management techniques, user interface development and the data base development.

Achieved results can be useful to volunteer, non-profit or other organizations and also for personal use.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	4
ВСТУП.....	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	7
1.1 Опис предметного середовища.....	7
1.2 Огляд наявних аналогів	12
1.3 Постановка задачі.....	16
Висновок до розділу	17
2 ОПИС ФУНКЦІОНАЛЬНОСТІ ЗАСТОСУНКУ.....	18
2.1 Вхідні дані.....	18
2.2 Вихідні дані.....	19
2.3 Опис структури бази даних.....	19
2.3 Опис процесів діяльності	24
2.4 Діаграма Використання	26
Висновок до розділу	27
3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	28
3.1 Засоби розробки	28
3.2 Архітектура програмного забезпечення.....	30
3.2.1 Діаграма класів.....	30
3.2.2 Діаграма компонентів.....	33
3.2.3 Специфікація функцій	36

					IT61.090БАК.004 ПЗ			
		<i>Прізвище</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		Гаршіна К.А..			Застосунок для керування проєктами.	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Перевірів.</i>		Тимофєєва Ю.С.				2	61	
<i>Н. кон.</i>					КПІ ім. Сікорського ФІОТ група IT-61			
<i>Затв.</i>								
					Пояснювальна записка.			

Висновок до розділу	41
4 РОЗРОБКА КОМПОНЕНТІВ КЕРУВАННЯ ПРОЄКТАМИ.....	42
4.1 Канбан-дошка	42
4.2 Team Canvas	44
4.3 SWOT-аналіз.....	46
4.4 Оцінка організації проєкту.....	47
4.5 Бухгалтерія.....	48
4.6 Вхід до застосунку	50
Висновок до розділу	51
5 ОГЛЯД ЗАСТОСУНКУ ЗІ СТОРОНИ КОРИСТУВАЧА.....	52
5.1 Керівництво користувача	52
5.2 Тестування застосунку	56
Висновок до розділу	57
ВИСНОВКИ.....	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТОК А.....	Ошибка! Закладка не определена.

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

БД — База даних

Діаграма Ганта — це діаграма, яка ілюструє розклад проєкту

ОС — Операційна система

Канбан-дошка — це засіб управління роботою на індивідуальному або колективному рівні

API (Application Programming Interface) — це набір визначень засобів для створення програмного забезпечення

EF Core (Entity Framework Core) — це технологія для доступу до даних

ER-діаграма (Entity-Relationship) — це модель даних для опису концептуальних схем

MVC (Model-view-controller) — це архітектурний шаблон для проєктування програмного забезпечення

SWOT-аналіз — це техніка стратегічного планування

Team Canvas — це бізнес модель для командної роботи

					IT61.090БАК.004 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дат		

ВСТУП

Дипломний проєкт присвячений розробці застосунку для керування проєктами. Тема дипломного проєкту була обрана з власного досвіду авторки організації великих проєктів у міжнародній студентській волонтерській організації «Board of European Students of Technology», де вона була активною протягом трьох років. Ця організація відповідальна за створення таких проєктів як хакатон «Int20h» з кількістю учасників близько 150, інженерні змагання «Engineering BEST European Competition», академічні курси, які відвідують 25 студентів з різних країн Європи та інші освітні заходи та проєкти, направлені на розвиток технічних навиків студентів. Найбільші проєкти займають багато часу (підготовка займає майже рік) та потребують багато людських ресурсів (над великим проєктом зазвичай працює команда з 15 людей, якій допомагають інші члени організації, кількість яких може складати близько 70-100 людей). На власному досвіді авторка дипломного проєкту побувала в командах як член команди, а також вона побувала головним організатором великого проєкту, тож авторка має розуміння які засоби потребує команда та головний організатор для організації великих проєктів.

Ідея для створення застосунку для керування проєктами прийшла до авторки дипломного проєкту у той час, коли вона була головним організатором академічних курсів, які відвідали 25 іноземних студентів та 15 студентів КПІ. Авторка зрозуміла, що на той момент не було додатку, який би повністю відповідав вимогам веденню великих проєктів. Усі додатки, які були доступні, надавали засоби лише для основного етапу організації проєктів, тобто опирались на делегацію завдань команді, але для більш доцільної організації проєктів слід користуватися трьома етапами організації: початковим, основним та кінцевим. На початковому етапі команда повинна поставити власні цілі по майбутній організації проєкту, визначити свої слабкі та сильні сторони як команди, поставити конкретну мету та ін. Основний етап відповідає делегуванню завдань

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		5

команді, розробки графічних матеріалів, ведення логістики проєкту. Кінцевий етап складається з опису загальної оцінки проєкту та заповнення форми передачі знань членами команди. На досвіді авторки під час організації проєктів не було єдиного засобу, де були б зосереджені усі етапи організації проєкту, що приводило до втрачання важливої інформації, яка могла би полегшити роботу команді. Звичайні засоби, якими користуються члени організацій: описання початкового етапу організації проєкту на листах паперу, делегування завдань команді через месенджери або індивідуально, ведення логістики в Excel Spreadsheets та ін. Отже, часто важлива інформація втрачається через те, що вона знаходиться в різних місцях. Саме тому ідеєю було створити застосунок, де усі етапи організації проєкту були зосереджені в одному місці.

Існує багато організацій, які мають схожу структуру керування проєктами. Сама організація «Board of European Students of Technology» має 93 локальні групи у 34 країнах Європи [1]. Отже цей застосунок буде корисним багатьом волонтерам по всьому світі. Також застосунок потенційно може бути використаний на персональному рівні.

					IT61.090БАК.004 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дат		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

При організації проєктів використовуються різні засоби для покращення роботи команди та оптимізації процесу підготовки проєкту. Типи проєктів, на які буде орієнтуватися застосунок, — це проєкти, організовані неприбутковими організаціями, так як авторка дипломного проєкту має безпосередній досвід у цій сфері діяльності та знає методики, які використовуються, але застосунок при бажанні можна використовувати для організації будь-яких проєктів. Взагалі етапи організації проєкту можна поділити на: початковий, основний та кінцевий. Ці етапи мають дуже важливу роль в організації та не повинні нехтуватися командою організаторів.

На початковому етапі організації проєкту команда та головний організатор проєкту повинні зібратися разом та прийняти низку важливих рішень, які стосуються проєкту та команди загалом. Більш конкретно, команда повинна поставити свої спільні та індивідуальні цілі, оцінити індивідуальний досвід кожного члена команди, зробити оцінку можливих сильних та слабких сторін, та ін. Існують декілька засобів, які слід використовувати щоб доцільно відповісти на поставлені запитання та оптимізувати початкову стадію організації проєкту.

Перший такий засіб — це засіб Team Canvas. Засіб Team Canvas — це бізнес модель для роботи в команді. Цей засіб є безкоштовним інструментом для лідерів, фасилітаторів та консультантів, щоб вивести членів команди на спільну сторону, вирішувати конфлікти та побудувати продуктивну культуру. Цей засіб корисний для лідерів чи фасилітаторів в моменти, коли потрібно зібрати членів команди разом, уточнити свої цілі, з'ясувати їхні мотивації та допомогти їм бути більш продуктивними. Колективи, які застосовують Team Canvas, відзначають більш високе розуміння між собою із меншою кількістю конфліктів [2]. Зображення засобу Team Canvas можна побачити на рисунку 1.1 [2]. Команда по черзі перелічує та записує своє розуміння задач поставлених засобом .

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		7

Засіб Team Canvas дозволяє членам команди та головному організатору проаналізувати свої цінності, правила роботи, цілі та навички, знання яких буде дуже корисним при подальшій роботі над проєктом.

The Team Canvas Basic

Version 1.0 | English | theteamcanvas.com

Most important things to talk about in the team to make sure your work as a group is productive, happy and stress-free

TEAM NAME _____ DATE _____

<p>PEOPLE & ROLES</p> <p>What are our names and the roles we have in the team?</p>	<p>VALUES</p> <p>What do we stand for? What are guiding principles? What are our common values that we want to be at the core of our team?</p>
<p>PURPOSE</p> <p>Why are we doing what we are doing in the first place?</p>	
<p>GOALS</p> <p>What we want to achieve as a group? What are our key goals that are feasible, measurable and time-bounded?</p>	<p>RULES & ACTION POINTS</p> <p>What are the rules we want to introduce after doing this session? How do we communicate and keep everyone up to date? How do we make decisions? How do we execute and evaluate what we do?</p>

The Team Canvas by TheTeamCanvas.com
Alexey Ivanov.

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License.
To view a copy of this license, visit: <http://creativecommons.org/licenses/by-sa/3.0/>

Рисунок 1.1 — Засіб Team Canvas

Наступний засіб керування проєктами — це SWOT-аналіз. SWOT-аналіз широко застосовується у процесі стратегічного планування, що полягає в розділенні чинників і явищ на чотири категорії: сильних (англ. Strengths) і слабких (англ. Weaknesses) сторін проєкту, можливостей (англ. Opportunities), що відкриваються при його реалізації, та загроз (англ. Threats), пов'язаних з його здійсненням [3]. Використання SWOT-аналізу допомагає команді об'єктивно оцінити майбутній шлях організації проєкту та підготувати себе до можливих ускладнень. Таблиця 1.1 описує основні чотири особливості SWOT-аналізу. На рисунку 1.2 зображено приклад використання SWOT-аналізу [3].

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		8

Таблиця 1.1 — Візуалізація SWOT-аналізу

	Позитивний вплив	Негативний вплив
Внутрішнє середовище	Strengths (сильні сторони)	Weaknesses (слабкі сторони)
Зовнішнє середовище	Opportunities (можливості)	Threats (загрози)

Strengths What do you do well? What unique resources can you draw on? What do others see as your strengths?	Weaknesses What could you improve? Where do you have fewer resources than others? What are others likely to see as weaknesses?
Opportunities What opportunities are open to you? What trends could you take advantage of? How can you turn your strengths into opportunities?	Threats What threats could harm you? What is your competition doing? What threats do your weaknesses expose to you?

Рисунок 1.2 — SWOT-аналіз

Наступний засіб початкового етапу організації проекту — це Канбан-дошка. Канбан — це графічна система для управління роботи під час її руху через певний процес. Канбан візуалізує як процес (робочий процес), так і саму роботу, що проходить через цей процес. Метод Канбан — це процес поступового вдосконалення певної поставленої задачі — будь то розробка програмного забезпечення, ІТ, підбір персоналу, маркетинг та продажі, закупівлі тощо. Практично будь-яка бізнес-функція може отримати користь від застосування

принципів Канбан методики. Метод Канбан дотримується набору принципів та практик управління та вдосконалення потоку роботи. Це метод, який сприяє поступовому вдосконаленню процесів організації. Якщо дотримуватися цих принципів і практик, можна успішно використовувати метод Канбан для отримання максимального вдосконалення будь-якого процесу — покращити потік роботи, скоротити час циклу, збільшити цінність для клієнта, — все це має вирішальне значення для будь-якого бізнесу сьогодні. [4].

Канбан-дошка є невід’ємним засобом при організації проєктів. Вона допомагає візуалізувати роботу команди, кількість роботи, яка вже зроблена та кількість роботи, що ще залишилося зробити. На рисунку 1.3 можна побачити приклад зображення Канбан-дошки [4].

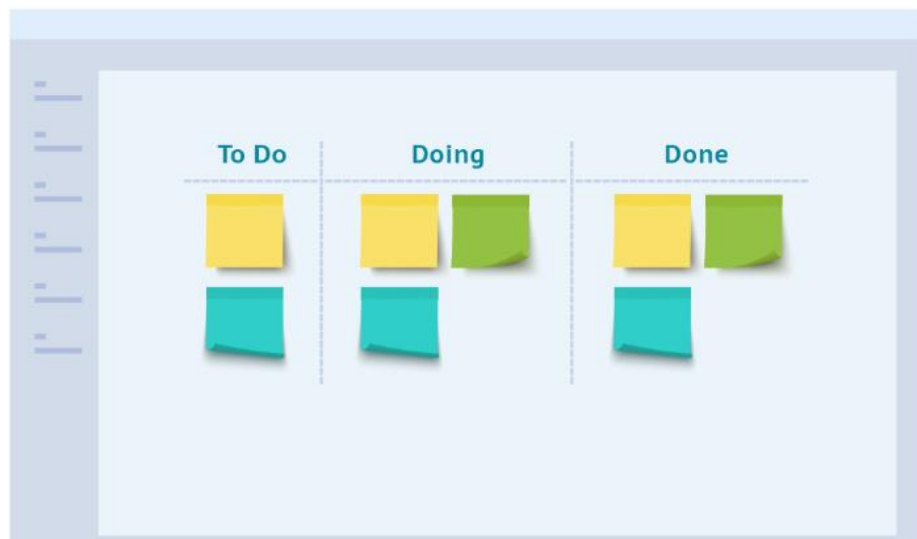


Рисунок 1.3 — Приклад Канбан-дошки

Також ще одним корисним та популярним засобом для організації роботи над проєктом є діаграма Ганта. Діаграма Ганта — це засіб, який часто використовується в управлінні проєктами, він є одним з найпопулярніших і корисних способів відображення діяльності (завдань чи подій), за допомогою плину часу. Зліва від діаграми відображається перелік завдань, а вгорі — відповідна шкала часу. Кожне завдання представлена смужкою.

Положення та довжина смужки відображає дату початку, тривалість та дату закінчення завдання [5]. Це дозволяє легко побачити:

- які існують різні завдання;
- коли кожне завдання починається і закінчується;
- скільки часу планує тривати кожне завдання;
- де завдання перетинаються з іншими видами діяльності і на скільки;
- початкові та кінцеві дати всього проекту.

Підводячи підсумок, діаграма Ганта показує, що потрібно зробити (завдання) та коли (графік). Структура діаграми Ганта зображена на рисунку 1.4 [5]. Діаграма Ганта виходить за рамки функціональності даного застосунку, але при подальшій роботі над ним, є можливість реалізації цього засобу.

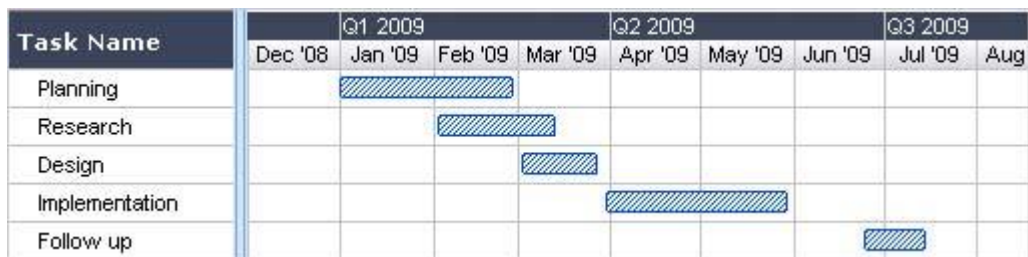


Рисунок 1.4 — Діаграма Ганта

При успішному використанні перерахованих вище засобів завершується початковий етап підготовки до організації проекту. У такому разі команда може переходити до основного етапу організації проекту.

До основного етапу організації проекту можна віднести: оновлення результатів роботи членів команди у Канбан-дошці (чи діаграмі Ганта), регулярне звітування про виконану роботу членами команди та головним організатором проекту, створення графічних матеріалів для користуванням у публічних чи корпоративних відносинах, ведення бухгалтерії проекту, тобто ведення переліку усіх витрат та прибутків та їх вплив на загальний бюджет проекту, а також загальне ведення комунікації між членами команди онлайн та у реальному житті.

Коли сам проєкт завершився, це не означає, що робота команди організаторів на цьому зупиняється. Залишається ще два важливих пункти, які є обов'язковими для успішного результату роботи над проєктом. Разом вони описують кінцевий етап організації проєкту.

Перший пункт — це оцінка виконаної роботи та перелік вдалих та невдалих нововведень, які команда використовувала при організації проєкту. Цей процес допоможе наступній команді, яка буде працювати над схожим проєктом, не допустити тих самих помилок та повторно використати усі вдалі засоби роботи минулої команди.

Другий пункт кінцевого етапу організації проєкту — це передача досвіду членами команди. Якщо при складанні оцінки виконаної роботи команди підбиваються загальні висновки про організацію проєкту, то при передачі досвіду кожен організатор, який відповідає за конкретну частину організації проєкту (логістика, дизайн, корпоративні зв'язки та ін.) описує свій індивідуальний досвід роботи над проєктом та тим самим передає свої знання наступній людині, яка буде займати ту саму посаду у майбутньому колективі.

Отже, задача зводиться до створення єдиного застосунку, який буде містити у собі усі перелічені методики та тримати у собі інформацію початкового, проміжного та кінцевого етапів організації проєкту.

1.2 Огляд наявних аналогів

Перед початком розробки застосунку, було розглянуто вже існуючі рішення керування проєктами для того щоб зробити висновки щодо можливої оптимізації рішень та створити унікальний продукт. Було розглянуто такі застосунки по керуванню проєктами: «Trello», «Basecamp», «Asana», «Casual», «OmniPlan» та «Podio».

Спочатку розглянемо застосунок «Trello». «Trello» — це безкоштовна багатоплатформна система управління проєктами. Застосунок «Trello»

					ІТ61.090БАК.004 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дат		

спирається на принцип Канбан, тобто метод візуалізації робочих процесів для забезпечення огляд проєкту від початку до кінця. Це робиться за допомогою спеціальних дошок, списків та карток. Дошка, як правило, зосереджується на конкретному проєкті, такому як запуск нового веб-сайту, або завдань, що ґрунтуються на процесах, таких як наймання працівника [6]. Кожна дошка містить списки, які можуть, наприклад, вказувати на прогрес організації проєкту. Картки у списках містять інформацію про конкретне завдання, і після його завершення, воно може бути переміщено зі списку в список. Застосунок «Trello» — це не лише інструмент управління бізнес-завданнями. «Trello» також широко використовується для особистих справ, включаючи будь-що від планування подорожей до весіль [6]. Інтерфейс застосунку «Trello» зображений на рисунку 1.5 [7].

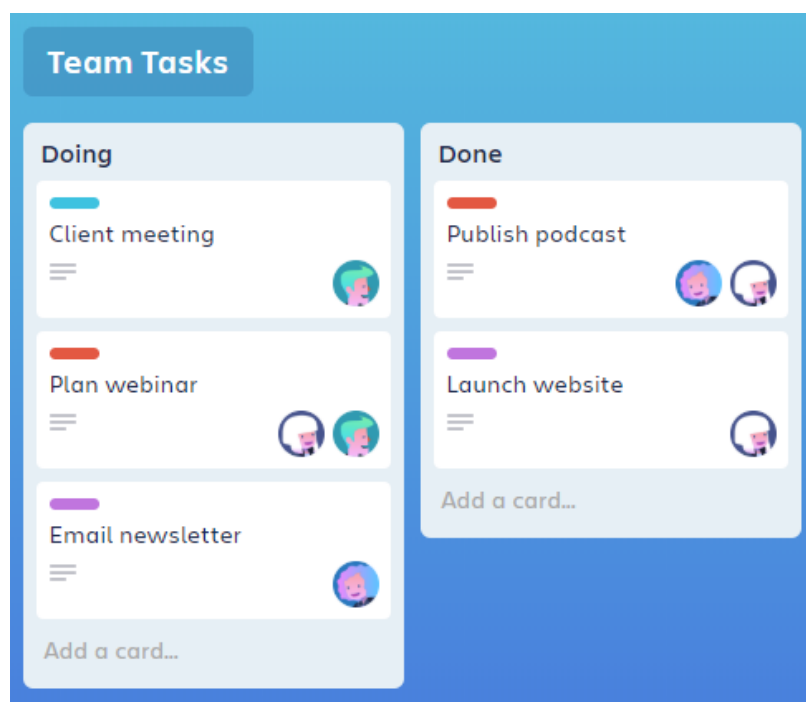


Рисунок 1.5 — Застосунок «Trello» використовує методику Канбан

Отже, «Trello» базується на методиці Канбан. Це зручний та корисний засіб для делегування завдань членам команди на початковому етапі організації

проекту. Цей засіб теж знадобиться при реалізації даного застосунку, але поруч і з іншими засобами, які відповідають усім етапам організації проекту.

Розглянемо далі застосунок «Basecamp». «Basecamp» — це онлайн-інструмент для управління проектами, спільної роботи і постановки завдань до проектів. Його основні особливості — це списки справ, віхове керування проектами, обмін повідомленнями на форумі, обмін файлами [8]. Обмін файлами це досить корисна функціональність для основного етапу організації проектів, але обмін повідомленнями виходить за рамки функціональності при розробці даного застосунку. Інтерфейс застосунку «Basecamp» зображений на рисунку 1.6 [8].

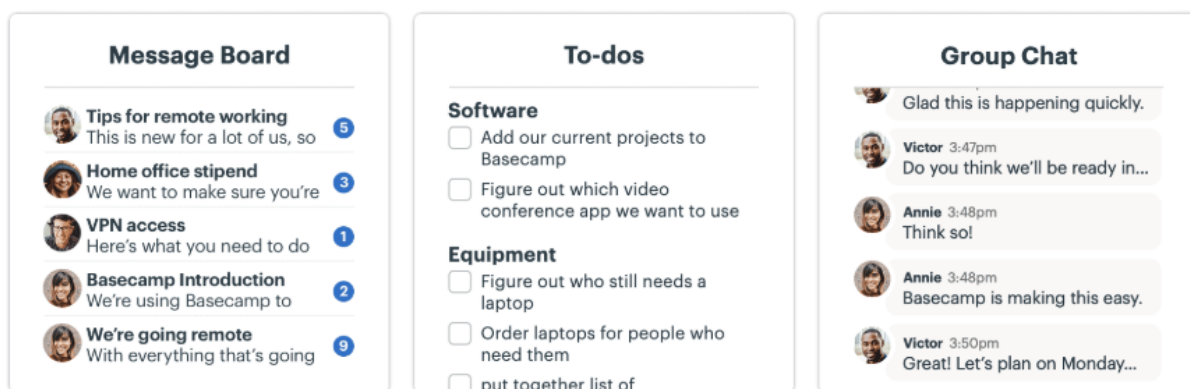


Рисунок 1.6 — Застосунок «Basecamp»

Наступний застосунок, який було розглянуто — це «Asana». «Asana» — програмне забезпечення для веб і мобільних пристроїв, призначене для спільної роботи над проектами без електронної пошти. Команди, які користуються застосунком «Asana», можуть створювати проекти, призначати роботу членам команди, визначати дедлайни та спілкуватися про завдання безпосередньо в застосунку [9]. Сюди також входять засоби звітності, вкладення файлів, календарі тощо [9]. Застосунок «Asana» містить багато різної функціональності, яка потрібна для керування проектами, але цьому додатку не вистачає використання засобів, які би допомогли виконувати керування проектом від його початку до завершення. Інтерфейс застосунку зображений на рисунку 1.7 [9].

Наступне рішення, яке було розглянуто це — застосунок «Casual». Застосунок «Casual» — це візуальний менеджер завдань із інструментом управління робочими потоками. Якщо проєкти включають і вимагають повторюваних процесів, користувачі зможуть клонувати і перезапускати кожен робочий процес, створений за допомогою застосунку «Casual» [10]. Додаток «Casual» надає зручний спосіб візуалізації управління проєктами, але його основною функцією є делегування завдань членам команди. Даний застосунок повинен мати додатково ще багато іншої функціональності, яка відповідає організації проєкту на початковому, основному та кінцевому етапах.

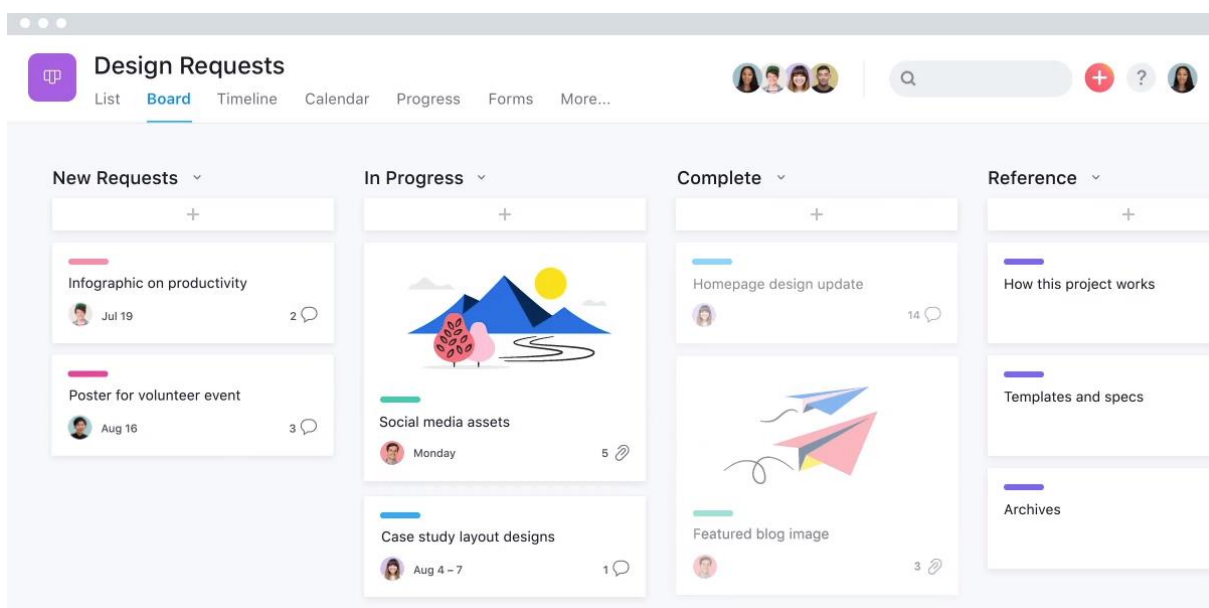


Рисунок 1.7 — Застосунок «Asana»

Застосунок «OmniPlan» дає можливість користувачам керувати складними проєктами, координувати командою та постійно вдосконалювати процеси роботи [11]. Застосунок «OmniPlan» має багато різних засобів керування проєктами, але основний використовуваний засіб — це діаграма Ганта. Застосунок доступний лише для О.С. Apple та має досить високу ціну, що робить його не придатним для використання волонтерськими організаціями та великою кількістю інших потенційних користувачів, які не зможуть дозволити собі придбати даний застосунок.

Наступний розглянутий застосунок — це застосунок «Podio». Гранульовані можливості адміністратора дозволяють користувачам застосунку контролювати, хто має доступ до робочих просторів застосунку «Podio» та можливість швидко вносити корективи. Користувачі також можуть приводити своїх клієнтів, фрілансерів та інших зовнішніх учасників у застосунок «Podio» [12]. Індивідуальність процесів для відповідності кожного кроку та етапу, який є унікальним для команди, забезпечує ефективність даного застосунку. Застосунок має багату функціональність, але як і додаток «OmniPlan» має високу ціну, яка не підходить для волонтерських організацій.

Роблячи аналіз цих рішень, було знайдено багато спільних атрибутів між цими додатками. А саме: використання Канбан-дошки, діаграми Ганта, обмін файлами, повідомленнями та ін. Проте не було використання усіх можливих засобів керування проєктами таких як Team Canvas, SWOT-аналіз та інших засобів, які використовуються неприбутковими або волонтерськими організаціями при керуванні проєктами та командами. Також при розробці даного застосунку береться до уваги не тільки основна стадія організації проєкту як делегування завдань членам команди, а також початкова та кінцева стадія, що не було присутнє в розглянутих вище рішеннях. Отже, при розробці даного застосунку є можливість створити унікальний продукт, який буде корисний для великої кількості людей та команд, а саме для волонтерських неприбуткових організацій або організацій іншого типу, а також для персонального користування.

1.3 Постановка задачі

Метою дипломного проєкту є оптимізація керування складними проєктами над якими працює велика кількість людей. Це можуть бути проєкти, наприклад, волонтерських неприбуткових організацій чи організацій будь-якого іншого типу.

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		16

Для досягнення поставленої мети мають бути вирішені такі задачі:

- можливість використовувати засоби керування проектом на початковому етапі організації (засіб Team Canvas та SWOT-аналіз);
- ведення логістики проекту;
- додавання до проекту потрібних матеріалів та файлів;
- ведення звітності роботи членів команди;
- використання засобу Канбан-дошка для делегування завдань членам команди;
- можливість керування проектом зі сторони головного організатора та зі сторони члена команди;
- заповнення форми передачі досвіду та оцінки організації на кінцевому етапі організації проекту.

Висновок до розділу

Цей розділ був присвячений загальним відомостям про дипломний проект. Був зроблений опис предметного середовища застосунку, який містив у собі відомості про основні засоби керування проектами та загальну інформацію про етапи організації великих проектів. Було виконано огляд наявних аналогів застосунку, а саме таких застосунків: «Trello», «Asana», «Basecamp», «Casual», «OmniPlan» та «Podio». Також у цьому розділі була виконана постановка задачі розробки застосунку.

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		17

2 ОПИС ФУНКЦІОНАЛЬНОСТІ ЗАСТОСУНКУ

2.1 Вхідні дані

У застосунку вхідні дані надходять від головного організатора проєкту та від членів команди. Обмін даними між членами команди — це одна з основних функцій застосунку. Основну кількість вхідних даних застосунок отримує від головного організатора проєкту.

Перелік вхідних даних, які можуть надавати або головний організатор проєкту, або член команди:

- електронну пошту та пароль для створення акаунту;
- особисту інформацію для власного акаунту;
- інформацію про прибутки або витрати для ведення логістики проєкту;
- звітування про виконану роботу;
- повідомлення на головній сторінці застосунку;
- форму передачі досвіду знань;
- позитивні або негативні відгуки до оцінки організації проєкту.

Головний організатор проєкту відповідає за делегацію завдань членам команди та устанавлення стратегії спільної підготовки. Головний організатор має можливість додавати більшу кількість даних до системи, ніж член команди, а саме: створювати проєкт та заповнювати інформацію про основні засоби керування проєктами..

Перелік вхідних даних, які надає головний організатор:

- дані для засобу Team Canvas, а саме: цінності, цілі, навички та активності команди;
- дані для SWOT-аналізу, а саме: сильні та слабкі сторони проєкту та можливості;
- загальну інформацію про проєкт, а саме: дати проведення проєкту, опис,
- кількість учасників проєкту та сума коштів для організації проєкту;
- завдання для делегування членам команди (Канбан-дошка).

					ІТ61.090БАК.004 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дат		

2.2 Вихідні дані

Вихідні дані проекту доступні як для головного організатора, так і для членів команди.

Вихідними даними для користувачів є:

- Канбан-дошка з трьома списками завдань для членів команди;
- SWOT-аналіз;
- засіб Team Canvas;
- звіти про виконану роботу команди;
- повідомлення;
- оцінка організації проекту;
- список проєктів певної організації та інформацію про дані проєкти;
- список форм передачі досвіду членів команди.

2.3 Опис структури бази даних

На кресленіку ІТ61.090БАК.004 Д1 наведена ER-діаграма для застосунку. База даних застосунку має 12 основних сутностей, які повністю описують функціональність застосунку для керування проєктами. У розробленій базі даних присутні такі зв'язки як «один до одного», «один до багатьох» та «багато до одного». Сутності, які присутні у базі даних: «Проєкт», «Користувач», «Оцінка», «Файл», «Звіт», «Завдання», «Форма передачі знань», «Витрата», «Прибуток», «Team Canvas» та «SWOT-аналіз». Сутності «Team Canvas» та «SWOT-аналіз» насправді представляють групи сутностей. Тобто сутність «Team Canvas» представляє сутності: «Цінність», «Правила», «Ціль» та «Мета». Ці сутності разом описують функціональність засобу керування командою Team Canvas, який більш детально описаний у розділі 1. Сутність «SWOT-аналіз» представляє сутності: «Сильна Сторона», «Слабка Сторона», «Можливість» та «Загроза». Ці сутності разом описують функціональність засобу керування командою SWOT-

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		19

аналіз, яка теж більш детально описана у розділі 1. Сутність «Витрати» разом з сутністю «Прибуток» допомагають слідкувати за бюджетом проєкту та налаштовані на допомогу члену команди, який відповідає за ведення логістики проєкту. Сутності «Позитивна оцінка» та «Негативна оцінка» відповідають за створення загальної оцінки проєкту для подальшої передачі знань наступним командам. Сутність «Користувач» створює сутність «Проект», яка є основною сутністю у системі і від якої залежать усі інші сутності, які описують засоби керування проєктом та командою.

Таблиця бази даних «Projects» зберігає дані про усі проєкти, створені головним організатором. Атрибути таблиці «Projects»: «Id» (тип даних — int), «Name» (тип даних — nvarchar), «StartDate» (тип даних — datetime2), «EndDate» (тип даних — datetime2), «Description» (тип даних — nvarchar), «NumPax» (тип даних — int) та «ProfitGoal» (тип даних — int). Атрибути «StartDate» та «EndDate» описують початкові та кінцеві дати проєкту. Атрибут «NumPax» описує кількість учасників проєкту. Атрибут «ProfitGoal» позначає бажану кількість прибутків для вдалої організації проєкту.

Таблиця бази даних «Reports» зберігає дані про звіти, залишені членами команди та головним організатором певного проєкту, про виконану роботу. Атрибути таблиці «Reports»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Date» (тип даних — datetime2), «Description» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена команди проєкту, який робить звіт. Атрибут «Date» позначає дату створення звіту членом команди. Атрибут «ProjectId» позначає id проєкту якому належить звіт. Сутність «Звіт», зв'язана з сутністю «Проект» у співвідношенні «один до багатьох».

Таблиця бази даних «Users» зберігає дані про користувачів, які зареєструвалися у застосунку. Система реєстрації та авторизації додатку розроблена за допомогою технології ASP.NET Identity. З її допомогою також були створені такі таблиці: UserRoles, Roles, Logins, UserClaims. Атрибути

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		20

таблиці «Users»: «Id» (тип даних — int), «UserName» (тип даних — nvarchar), «Email» (тип даних — nvarchar), «EmailConfirmed» (тип даних — bit) та «Password» (тип даних — nvarchar). Атрибут «UserName» позначає ім'я зареєстрованого користувача. Атрибут «Email» позначає електронну пошту користувача, а атрибут «EmailConfirmed» позначає чи підтвердив користувач свою реєстрацію у застосунку через електронну пошту. Атрибут «Password» позначає пароль користувача. Для реєстрації у застосунку користувачу потрібно надати свою електронну пошту та пароль та підтвердити реєстрацію через свою електронну пошту.

Таблиця бази даних «KTs» зберігає дані про досвід членів команди проєкту, який буде переданий наступній команді, яка буде працювати над схожим проєктом. КТ — це скорочення від Knowledge transfer, що в перекладі з англійської — передача знань або досвіду. Атрибути таблиці «KTs»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Position» (тип даних — nvarchar), «PositionExp» (тип даних — nvarchar), «TeamExp» (тип даних — nvarchar), «ImprovementPoints» (тип даних — nvarchar), «ThingsToKeep» (тип даних — nvarchar), «FinalAdvice» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена команди проєкту, який виконує передачу досвіду. Атрибут «Position» позначає позицію, яку член команди займає у проєкті (головний організатор, дизайнер, відповідальний за корпоративні відносини та ін.). Атрибут «PositionExp» описує досвід члена команди проєкту відносно його позиції. Атрибут «TeamExp» позначає досвід члена команди проєкту роботи з іншими членами команди. Атрибут «ImprovementPoints» позначає можливі аспекти покращення роботи людини, яка займає дану позицію у команді. Атрибут «ThingsToKeep» позначає перелік аспектів роботи, які варто зберегти для організації проєктів схожого типу в майбутньому. Атрибут «FinalAdvice» позначає загальні поради наступній людині, яка буде займати дану позицію в проєкті. Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність.

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		21

Таблиця «Expenses» зберігає дані про витрати, які належать певному проєкту. Атрибути таблиці «Expenses»: «Id» (тип даних — int), «Item» (тип даних — nvarchar), «Quantity» (тип даних — int), «PricePer1» (тип даних — int), «Price» (тип даних — int), «Purpose» (тип даних — nvarchar), «Unit» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «Item» позначає витрату певного проєкту. Атрибут «Quantity» позначає кількість таких витрат, які потрібно зробити під час організації або проведення проєкту. Атрибут «PricePer1» позначає ціну за одну таку витрату. Атрибут «Price» позначає загальну кількість грошей, які потрібно буде виділити на дану витрату. Атрибут «Purpose» описує призначення даної витрати. Атрибут «Unit» позначає одиницю виміру даної витрати (кг, літри, шт. та ін.). Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Витрата» зв'язана з сутністю «Проект» у співвідношенні «один до багатьох».

Таблиця «Profits» зберігає дані про прибутки, які належать певному проєкту. Атрибути таблиці «Profits»: «Id» (тип даних — int), «Resource» (тип даних — nvarchar), «Money» (тип даних — int), «Description» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «Resource» описує ресурс даного прибутку (грант, компанія, фонд та ін.). Атрибут «Money» описує кількість грошей, отриманих від певного ресурсу. Атрибут «Description» описує деталі про даний прибуток (хто з членів команди веде комунікацію, контакти, перспективи майбутньої співпраці). Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Прибуток» зв'язана з сутністю «Проект» у співвідношенні «один до багатьох».

Таблиці «PositivePoints» та «NegativePoints» разом утворюють загальну оцінку організації певного проєкту. Таблиця «PositivePoints» зберігає дані про позитивні аспекти організації проєкту. Атрибути таблиці «PositivePoints»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Description» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена команди проєкту, який додає позитивну оцінку організації.

Атрибут «Description» описує зміст оцінки користувача. Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Позитивна оцінка» зв'язана з сутністю «Проект» у співвідношенні «один до багатьох».

Таблиці «Strengths», «Weaknesses», «Opportunities» та «Threats» разом утворюють SWOT-аналіз певного проєкту. Більш детально про цей засіб можна дізнатися у розділі 1. Таблиця «Opportunities» зберігає дані про можливості певного проєкту, тобто можливості, які можуть створитися при організації даного проєкту. Атрибути таблиці «Opportunities»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Description» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена команди проєкту, який додає можливість до SWOT-аналізу. Атрибут «Description» описує зміст можливості. Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Можливість» засобу SWOT-аналіз, зв'язана з сутністю «Проект» у співвідношенні «один до багатьох».

Таблиці «Goals», «Values», «Activities» та «Skills» разом утворюють засіб Team Canvas певного проєкту. Більш детально про цей засіб можна дізнатися у розділі 1. Таблиця «Goals» зберігає дані про цілі організації певного проєкту, тобто цілі, які ставить собі команда на початковому етапі організації проєкту. Атрибути таблиці «Goals»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Description» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена команди проєкту, який додає ціль до засобу Team Canvas. Атрибут «Description» описує зміст цілі. Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Ціль», зв'язана з сутністю «Проект» у співвідношенні «один до багатьох».

Таблиця «KanbanTasks» зберігає дані про завдання, які належать засобу Канбан-дошка. Більш детально про цей засіб можна дізнатися у розділі 1. Атрибути таблиці «KanbanTasks»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Task» (тип даних — nvarchar), «TaskOrder» (тип даних — int), та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена

команди проєкту, який додає завдання до Канбан-дошки. Атрибут «Task» описує зміст завдання. Атрибут «TaskOrder» позначає порядок завдання у певному списку. Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Завдання», зв'язана з сутністю «Проєкт» у співвідношенні «один до багатьох».

Таблиця «Files» зберігає дані про існуючі матеріали певного проєкту. Атрибути таблиці «Files»: «Id» (тип даних — int), «NameOfUser» (тип даних — nvarchar), «Name» (тип даних — nvarchar), «Path» (тип даних — nvarchar) та «ProjectId» (тип даних — int). Атрибут «NameOfUser» позначає ім'я члена команди проєкту, який додає файл до проєкту. Атрибут «Name» описує назву файлу. Атрибут «Path» описую шлях до файлу. Атрибут «ProjectId» позначає id проєкту, якому належить дана сутність. Сутність «Файл» зв'язана з сутністю «Проєкт» у співвідношенні «один до багатьох».

2.3 Опис процесів діяльності

Застосунок має три основних процеси діяльності: початковий етап організації проєкту зі сторони головного організатора, основний етап організації проєкту зі сторони головного організатора або члена команди та кінцевий етап організації проєкту зі сторони головного організатора або члена команди. Разом ці процеси забезпечують керування певним проєктом від початку до кінця його організації.

Для початку розглянемо початковий етап організації проєкту. Перші кроки в усіх трьох процесах — це реєстрація та авторизація користувача, після чого, якщо користувач є головним організатором, він має можливість створити новий проєкт. Після створення проєкту наступний крок — це заповнення Канбан-дошки завданнями для делегування їх іншим членам команди та відстеження загального прогресу організації проєкту. Користувач може помістити завдання у три різних зв'язаних списки: «Потрібно виконати», «У процесі виконання» та

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докum.	Підпис	Дат		24

«Виконано». Коли завдання успішно розміщені у списках у головного організатора є можливість створити SWOT-аналіз та Team Canvas. Більш детально про ці засоби можна дізнатися у розділі 1. Використання цих засобів робиться на огляд користувача відносно величини проєкту, але є вкрай рекомендоване для проєктів кількість членів команди яких перевершує 6 людей. Після вирішення та додавання основних функцій до проєкту, головний організатор проєкту вирішує чи потрібно додавати додаткову інформацію чи файли до проєкту — це останній пункт в даному процесі. По завершенню користувач виходить зі свого акаунту.

Наступний процес — основний етап організації проєкту зі сторони головного організатора або члена команди. Після успішної авторизації чи реєстрації, користувач, який може бути як головним організатором так і членом команди, має можливість додати завдання до Канбан-дошки чи, наприклад, перенести завдання зі списку «У процесі виконання» до «Виконано». Також користувач має можливість додати до проєкту файл, який може бути, наприклад, або презентація про партнерську пропозицію чи дизайн матеріали для соціальних мереж. Після цього користувач може залишити звіт про виконану ним роботу, щоб інші члени команди могли бачити індивідуальний прогрес кожного члена команди. Після цього, якщо у бюджет проєкту були внесені зміни, наприклад знадобилося придбати декілька кілограм печива для офіційної церемонії відкриття проєкту, цю інформацію можна занести до розходів проєкту. Бухгалтерія описує інформацію про витрати та прибутки проєкту. За допомогою цієї інформації підраховується загальний бюджет проєкту. Укінці користувач може залишити повідомлення для команди на головній сторінці застосунку, якщо бажає. Після цього користувач виходить зі свого акаунту.

Останній процес — кінцевий етап організації проєкту зі сторони головного організатора або члена команди. На цьому етапі користувач може залишити оцінку щодо конкретного аспекту організації проєкту (позитивну або негативну). Наприклад, було прийнято рішення запрошувати на захід студентів з різних міст

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		25

України і це дало гарні результати для видимості проєкту та організації. У такому випадку можна залишити позитивний відгук про це рішення для того, щоб цю практику можна було продовжувати у майбутньому. Також на кінцевому етапі організації проєкту потрібно задокументувати індивідуальний досвід кожного члена команди, для того щоб наступній людині, яка буде займати цю ж посаду, було простіше адаптуватися. Тож після заповнення форми передачі досвіду, користувач може залишити повідомлення для команди на головній сторінці застосунку та, за бажанням, вийти зі свого акаунту.

2.4 Діаграма Використання

Даний застосунок призначений для користування командами, які організовують великі проєкти, наприклад у різних неприбуткових чи волонтерських організаціях або для особистого користування. У командах завжди присутній лідер чи головний організатор, який делегує завдання членам команди, веде звітність перед командою та організацією, фасилітує зустрічі. Головний організатор також повинен слідкувати за веденням бухгалтерії проєкту. Отже, у даному застосунку буде дві дійові особи (актора): головний організатор та член команди.

Діаграма використання для головного організатора зображена на кресленику ІТ61.090БАК.004 Д2. Вона описує усі можливі дії, які може виконувати головний організатор застосунку. Основна функція, яку може виконувати головний організатор — це створення нового проєкту. Створення нового проєкту передбачає початок першого етапу організації проєкту. Початковий етап організації проєкту передбачає використання трьох основних засобів: заповнення Канбан-дошки завданнями (список «Потрібно виконати»), наповнення інформацією SWOT-аналіз про сильні та слабкі сторони проєкту, а також можливості та загрози та наповнення інформацією про цілі, цінності, навички та активності команди засіб Team Canvas.

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		26

Більш детально функціональність цих засобів описана у розділі 1. Інші можливості користувачів є спільними.

Вхід до застосунку виконується однаково обома типами користувачів. Якщо потрібний проєкт вже існує, головний організатор чи член команди можуть виконати різні функції над цим проєктом, а саме: редагувати Канбан-дошку, залишити звіт про виконану роботу, додавання файлів до проєкту, внесення витрат або прибутків для підрахунку загального бюджету, заповнення форми передачі досвіду, внесення оцінки організації проєкту і також відправлення повідомлення на головну сторінку застосунку. Діаграма використання для члена команди зображена на кресленнику ІТ61.090БАК.004 Д2.

Висновок до розділу

У цьому розділі були визначені вхідні та вихідні дані застосунку. Була описана база даних застосунку та описані її основні сутності. Було описано три основних процеси застосунку: початковий, основний та кінцевий етап організації проєкту. Також проаналізовано користування застосунком зі сторони головного організатора та члена команди.

					ІТ61.090БАК.004 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дат		

3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Перед початком реалізації застосунку було обрано різні технології, які будуть перелічені у цьому розділі. Технології обирались зважаючи на власний досвід користування, на актуальність технологій та на можливості виконання потреб розробки застосунку, а саме розробки застосунку з зручним користувацьким інтерфейсом.

Першим чином було обрано середовище розробки застосунку, а саме ASP.NET Core. ASP.NET Core — це вільне та відкрите програмне забезпечення каркаса веб застосунків, з продуктивністю вищою ніж у ASP.NET. Середовище розробки ASP.NET Core було розроблено корпорацією Microsoft. Фреймворк б'єднує раніше окремі ASP.NET MVC та ASP.NET Web API у єдину програмувальну модель [13]. Не зважаючи на те, що ASP.NET Core є новим фреймворком, побудованим на новому веб-стеку, ASP.NET Core має високий ступінь сумісності концепцій з ASP.NET MVC, який об'єднує функціональність MVC, Web API та Web Pages [13]. Отже, ASP.NET Core — це сучасна технологія, яка поєднує у собі усі можливості попередніх ASP.NET, але і багато нових. Це робить цю технологію ідеальним варіантом для розробки даного застосунку.

Для роботи з базою даних була обрана технологія Entity Framework Core. Entity Framework Core (EF Core) являє собою об'єктно-орієнтовану технологію від компанії Microsoft для доступу до даних [14]. EF Core є ORM-інструментом (object-relational mapping — відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати з базами даних, але дає більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних і її таблиць і працювати з даними незалежно від типу сховища [14]. Якщо на фізичному рівні розробники оперують таблицями, індексами, первинними і зовнішніми ключами, але на концептуальному рівні, який пропонує Entity Framework, розробники вже працюють з об'єктами [14].

					ІТ61.090БАК.004 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дат		

Для швидкого та зручного процесу створення різних представлень була обрана технологія Razor Pages. Технологія Razor Pages дозволяє створювати сторінки з кодом Razor, які можуть обробляти запити користувачів.

Для розробки зручної системи реєстрації та авторизації користувачів була використана технологія ASP.NET Identity. Дана система дозволяє користувачам створювати облікові записи, авторизуватися, управляти обліковими записами або використовувати для входу на сайт облікові записи зовнішніх провайдерів, таких як Facebook, Google, Microsoft, Twitter та інші [15].

Шаблон MVC розділяє застосунок на три компоненти — модель, представлення та контролер.

Модель описує дані. Об'єкти моделі зберігають дані, отримані з бази даних.

Представлення у шаблоні MVC — це інтерфейс користувача. Представлення відображає дані моделі користувачу, а також дозволяє їх змінювати. Представлення у ASP.NET MVC — це мова HTML, CSS та спеціальний синтаксис (Razor), який робить легкою комунікацію моделі з контролером [16].

Контролер обробляє запити користувача. Зазвичай користувач використовує представлення даних і здійснює HTTP-запит, який буде оброблятися контролером. Контролер обробляє запит і повертає відповідне представлення. На рисунку 3.1 [16] користувач вводить URL-адресу у браузері, вона переходить на веб-сервер і перенаправляється до контролера. Контролер зіставляє відповідне представлення та модель для цього запиту, створює відповідь і відсилає її назад у браузер.

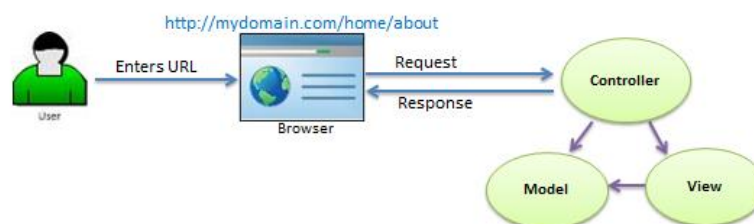


Рисунок 3.1 – Шаблон MVC

Сучасне веб-програмування та створення веб-сайтів неможливо уявити без використання мови JavaScript. В даний час, все частіше використовується JavaScript-фреймворки і бібліотеки. Однією з таких бібліотек є jQuery. Поняття "jQuery" об'єднує цілу екосистему бібліотек, побудовану навколо базової бібліотеки. Фреймворк jQuery дозволяє не тільки управляти елементами DOM, відправляти запити AJAX, але представляє інструмент для побудови призначеного для користувача інтерфейсу за допомогою віджетів [17]. Для роботи з інтерфейсом призначена спеціальна бібліотека jQuery UI, яка і буде використовуватися при побудові даного застосунку.

JqueryUI — це потужна бібліотека Javascript, побудована на бібліотеці JavaScript jQuery. Інтерфейс призначений для користувальницького інтерфейсу. Це набір плагінів для jQuery, який додає нових функціональних можливостей до основної бібліотеки jQuery. Набір плагінів у JqueryUI включає взаємодію інтерфейсів, ефекти, анімацію, віджети та теми, побудовані поверх jQuery JavaScript Library [18].

3.2 Архітектура програмного забезпечення

3.2.1 Діаграма класів

Так як застосунок має досить велику кількість класів, була розроблена діаграма для відображення наявних класів та їх зв'язків між собою. Діаграму можна побачити на кресленику ІТ61.090БАК.004 ДЗ. Бачимо, що на діаграмі присутні 12 основних класів. Зв'язки, які присутні в діаграмі класів: «один до одного», «один до багатьох», «багато до одного». Клас User застосовується для реєстрації та авторизації користувача в застосунку. Після реєстрації користувач має можливість (якщо він отримає права головного організатора) створити новий проєкт. За дії над проєктом відповідає клас Project, він є основним класом в системі, від якого залежать усі інші класи. Ці класи надають усі можливості керування проєктами, які забезпечує застосунок, а саме: створення завдань для

					ІТ61.090БАК.004 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дат		

Канбан-дошки, створення SWOT-аналізу, створення засобу Team Canvas, додавання файлів до проєкту, додавання оцінки організації та ведення логістики проєкту.

При створенні нового об'єкту класу «Project», користувач заповнює певні властивості, які відповідають за опис даного проєкту: «Назва», «Дата початку проєкту», «Дата кінця проєкту», «Опис» та id об'єкту «User», який є відповідальним за створення проєкту. Клас Project також містить списки об'єктів класів, які залежать від нього. Ці списки допомагають створенню потрібних нам засобів керування проєктами. Перелік цих списків: «Завдання», «Звіти», «Сильні сторони», «Слабкі сторони», «Цінності», «Активності», «Правила», «Можливості», «Загрози», «Позитивні відгуки», «Негативні відгуки», «Витрати», «Прибутки» та «Форми передачі знань».

Один з основних засобів застосунку — це делегування завдань членам команди за допомогою Канбан-дошки. Клас, який відповідає за цю функціональність, називається «Kanban Task». Властивості, які має цей клас: «Опис» та id об'єкту «Project», якому належить завдання. Користувач створює новий об'єкт класу «Kanban Task» та робить опис завдання, яке буде належати цьому класу. Список завдань певного проєкту знаходиться в об'єкті класу «Project». За допомогою цього списку може бути створена функціональність Канбан-дошки. Список завдань проєкту завантажується у представлення, яке відображає користувачу Канбан-дошку. Користувач бачить три зв'язані між собою списки: «потрібно виконати», «у процесі виконання» та «виконано». Список завдань проєкту відображається спочатку у списку «потрібно виконати». За допомогою можливостей бібліотеки jQuery UI, користувач може пересувати завдання між іншими двома списками та змінювати послідовність завдань у певному списку. Цей процес відтворює функціональність Канбан-дошки. Більш детально його буде описано у розділі 4. До початкового етапу організації проєкту також відносяться засоби «Team Canvas» та «SWOT-аналіз», функціональність яких описана більш детально у розділі 1.

Мета засобу «Team Canvas» — це початок налагодження взаєморозуміння поміж членами команди. Засіб «Team Canvas» складається з чотирьох класів: «Skill», «Activity», «Value» та «Goal», які відповідно описують цінності, навички, цілі та активності команди певного проєкту. Кожен з цих класів має дві властивості: «Опис» та id об'єкту «Project», якому належить засіб «Team Canvas». Списки об'єктів цих класів належать об'єкту «Project». Засіб «Team Canvas» відображається користувачеві за допомогою представлення, куди завантажуються об'єкти, що відповідають за зміст даного компоненту. Представлення також використовує бібліотеку jQuery UI для інтерфейсу користувача, але у даному випадку користувач не може взаємодіяти зі списками. Знову цей процес описаний більш детально у розділі 4.

Схожу функціональність можна побачити в засобі «SWOT-analysis». Мета засобу «SWOT-аналіз» — це побудова майбутньої стратегії керування проєктом з розумінням усіх майбутніх аспектів організації. Засіб «SWOT-analysis» складається з чотирьох класів: «Strength», «Weakness», «Opportunity» та «Threat», які описують сильні та слабкі сторони проєкту, потенціальні можливості та загрози. Кожен з цих класів має дві властивості: «Опис» та id об'єкту «Project», якому належить «SWOT-аналіз». Списки об'єктів цих класів належать об'єкту «Project». «SWOT-аналіз» відображається користувачеві за допомогою представлення, куди завантажуються об'єкти, що відповідають за зміст даного засобу. Представлення також використовує бібліотеку jQuery UI для інтерфейсу користувача, у даному випадку користувач теж не може взаємодіяти зі списками. Отже основна функція засобів «Team Canvas» та «SWOT-аналіз» — це переглядання інформації членами команди.

Мета класу «Report» — ведення звітів членами команди певного проєкту. Клас «Report» має чотири властивості: «Дата залишення звіту», «Ім'я члена команди, який заповнює звіт», «Опис звіту» та та id об'єкта «Project», якому належить звіт. Звіти переглядаються користувачем у вигляді списку. Звіти можуть залишати як члени команди, так і головні організатори певного проєкту.

Мета засобу «Evaluation» — відображення оцінки організації певного проєкту. Засіб «Evaluation» складається з двох класів: «Positive Point» та «Improvement Point». Кожен з цих класів має дві властивості: «Опис» та ід об'єкту «Project», якому належить оцінка. Списки об'єктів цих класів належать об'єкту «Project».

Мета засобу «Logistics» — підрахунок загального бюджету проєкту та допомога з аналізом інформації члену команди, відповідального за ведення бухгалтерії. Засіб «Logistics» складається з двох класів: «Profit» та «Expense», які відповідно описують прибуток та витрати певного проєкту та утворюють загальний бюджет. Основні властивості цих класів: «Назва», «Ціна», «Опис» та ід об'єкту «Project».

Мета класу «Knowledge Transfer» — заповнення користувачем форми передачі досвіду для наступної людини, яка буде займати ту ж позицію на майбутньому проєкті. Клас «Knowledge Transfer» має шість властивостей: «Ім'я члена команди», «Досвід на позиції», «Досвід у команді», «Шляхи покращення роботи», «Позитивні аспекти роботи на позиції» та ід об'єкту «Project», якому належить форма знань. Форми передачі знань переглядаються користувачем у вигляді списку.

Мета класу «Message» — залишення користувачем повідомлень (коментарів) на головній сторінці застосунку. Клас «Message» має дві властивості: «Ім'я користувача» та «Зміст». Залишені користувачами повідомлення можна побачити на головній сторінці додатку.

3.2.2 Діаграма компонентів

Діаграма компонентів зображена на кресленику IT61.090БАК.004 Д4. Застосунок містить 12 компонентів. Компоненти поєднують групи класів базуючись на спільній меті. Компоненти спілкуються за допомогою інтерфейсів. Компоненти, які входять до діаграми: «Реєстрація та Авторизація Користувача»,

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		33

«Проект», «Перша Стадія Організації», «Друга Стадія Організації» та «Третя Стадія Організації». До компоненту «Перша Стадія Організації» входять такі компоненти: «SWOT-аналіз» та «Team Canvas». До компоненту «Друга Стадія Організації» входять такі компоненти: «Канбан-дошка», «Матеріали проекту» та «Логістика». До компоненту «Третя Стадія Організації» входять компоненти: «Оцінка» та «Форма передачі знань».

Компонент «Реєстрація та Авторизація Користувача» включає до себе такі функції застосунку як: реєстрація та авторизація користувачів. Класи, які входять до цього компоненту створені за допомогою технології ASP.NET Identity. Це такі класи як: User та UserManager, Role та RoleManager. Від компоненту «Реєстрація та Авторизація Користувача» виходить інтерфейс «Інформація про проект». Тобто авторизований користувач при створенні проекту надає потрібну інформацію. Цей інтерфейс надається компоненту «Проект».

Компонент «Проект» описує основний клас застосунку від якого залежні усі інші класи та компоненти. До компоненту «Проект» надходить інтерфейс з інформацією про створений проект від авторизованого користувача. Від компоненту «Проект» надходять сім різних інтерфейсів.

Компонент «Перша Стадія Організації» описує початкову стадію організації проекту, а саме створення SWOT-аналізу та засобу Team Canvas. Ці засоби керування проектами описані більш детально у розділі 1. Саме ці компоненти входять до компоненту «Перша Стадія Організації».

Компонент «SWOT-аналіз» входить до компоненту «Перша Стадія Організації». Він описує класи, які відповідають за сильні та слабкі сторони проекту, загрози та можливості організації. До компоненту «SWOT-аналіз» надходить інтерфейс, який надає компоненту «Проект» інформацію про заповнений «SWOT-аналіз». Компонент «Team Canvas» також входить до компоненту «Перша Стадія Організації». Він описує класи, які відповідають за навички, цілі, активності та правила команди проекту. До компоненту «SWOT-

аналіз» надходить інтерфейс, який надає компоненту «Проект» інформацію про заповнений «Team Canvas».

Компонент «Друга Стадія Організації» описує основну стадію організації проекту, а саме Канбан-дошка, логістика та матеріали проекту. Ці компоненти входять до компоненту «Друга Стадія Організації». Засіб Канбан-дошка більш детально описаний у розділі 1.

Компонент «Канбан-дошка» входить до компоненту «Друга Стадія Організації». Він відповідальний за делегування завдань членам команди. Від компоненту «Канбан-дошка» надходять два інтерфейси: один надає компоненту «Проект» список завдань, які містить Канбан-дошка, другий порядок у списках «у процесі виконання», «потрібно виконати» та «виконано».

Компонент «Логістика» входить до компоненту «Друга Стадія Організації». Він описує прибутки та витрати проекту його загальний бюджет. До компоненту «Логістика» надходить інтерфейс з інформацією про ціни та прибутки від компоненту «Проект». Також від компоненту «Логістика» надходить інтерфейс з інформацією про загальний бюджет проекту.

Компонент «Матеріали проекту» входить до компоненту «Друга Стадія Організації». Він відповідальний за утримання списку матеріалів, які належать певному проекту (наприклад зображення). До компоненту «Матеріали проекту» надходить інтерфейс з файлами, які будуть належати списку матеріалів проекту. Також від компоненту «Матеріали проекту» надходить інтерфейс, який містить список матеріалів проекту .

Компоненти «Форма передачі знань» та «Оцінка проекту» входить до компоненту «Третя Стадія Організації». Вони описують кінцевий етап організації проекту.

Компонент «Оцінка проекту» описує оцінку організації проекту після його завершення та складається з позитивних та негативних відгуків членів команди.

До компоненту «Оцінка проекту» надходить інтерфейс з позитивними та негативними відгуками.

До компоненту «Форма передачі знань» надходить інтерфейс з заповненою членом команди формою передачі досвіду, а від компоненту «Форма передачі знань» надходить список заповнених користувачами форм.

3.2.3 Специфікація функцій

Застосунок має велику кількість компонентів, отже досить багато функцій, які будуть детально описані у цьому підрозділі.

Функція `ProjectIndex ()` призначена для повернення користувачу представлення зі списком усіх наявних у застосунку проєктів. Функція немає параметрів. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту.

Функція `ProjectDetails (int? id)` призначена для отримання деталей про існуючий проєкт. При виконанні функції користувачу повертається представлення з даними про певний проєкт, а також усі доступні засоби керування даним проєктом. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається параметр `id` проєкту, якщо параметр відсутній, функція повертає список усіх існуючих проєктів.

Функція `ProjectCreate()` призначена для відкриття форми користувачу для створення нового проєкту. Ця функція доступна лише для користувачів з правами головного організатора проєкту. При створенні проєкту головний організатор заповнює інформацію про початкові та кінцеві дати проєкту, кількість учасників та бажану суму прибутків, які потрібні для вдалої організації проєкту. Функція немає параметрів.

Функція `ProjectCreate(Project project)` призначена для створення нового проєкту. На відміну від минулої функції, яка мала HTTP запит GET, дана функція має HTTP запит POST. Це означає, що ця функція використовується для відправлення даних до серверу. Функція здійснюється після заповнення

користувачем форми створення нового проєкту та натисканням на кнопку «Submit». Функції передається параметр project, що є об'єктом класу Project.

Функція ProjectEdit(int? id) призначена для відкриття форми користувачу для редагування певного проєкту. З допомогою цієї функції користувач, за бажанням, може змінити основну інформацію про певний проєкт. Ця функція доступна лише для користувачів з правами головного організатора проєкту. До функції передається параметр id проєкту, якщо параметр відсутній, функція повертає список існуючих проєктів.

Функція ProjectEdit(int id, Project project) призначена для змінення інформації про певний проєкт. Ця функція доступна лише для користувачів з правами головного організатора проєкту. На відміну від минулої функції, яка мала HTTP запит GET, дана функція має HTTP запит POST. Це означає, що ця функція використовується для відправлення даних до серверу. Функція здійснюється після заповнення користувачем форми створення нового проєкту та натисканням на кнопку «Submit». Функції передається параметр id проєкту та параметр project, що є об'єктом класу Project.

Функція ProjectDelete(int? id) призначена для видалення об'єкту проєкт. Ця функція доступна лише для користувачів з правами головного організатора проєкту. За допомогою цієї функції користувач, за бажанням, може видалити проєкт зі списку існуючих проєктів. HTTP запит функції є запит GET. До функції передається параметр id проєкту, який потрібно видалити, якщо параметр відсутній, функція повертає список проєктів.

Функція KanbanProjectTasks(int id) призначена для отримання представлення засобу керування проєктами Канбан-дошка. У засобі Канбан-дошка присутні три зв'язаних списки, що відповідають трьом типам завдань певного проєкту. Ці списки мають назви: «У процесі виконання», «Потрібно виконати» та «Виконано». Засіб Канбан-дошка відноситься до початкового етапу організації проєкту. Більш детально про цей засіб можна дізнатися у розділі 1. Ця функція доступна для користувачів з правами члена команди та з правами

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		37

головного організатора проєкту. До функції передається параметр id проєкту, до якого належать завдання.

Функція SWOT(int id) призначена для отримання представлення засобу SWOT-аналіз певного проєкту. Засіб SWOT-аналіз складається з чотирьох списків: сильні та слабкі сторони проєкту, можливості та загрози. Засіб SWOT-аналіз відноситься до початкового етапу організації проєкту. Більш детально про цей засіб можна дізнатися у розділі 1. До представлення, яке відображається при виконанні функції, передається дані чотирьох моделей з БД: «Strength», «Weakness», «Threat» та «Opportunity». Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається параметр id проєкту, до якого належать перелік властивостей засобу SWOT-аналіз.

Функція TeamCanvas(int id) призначена для отримання представлення з засобом Team Canvas певного проєкту. Засіб Team Canvas складається з чотирьох списків: цілі, цінності, правила та навик. Засіб Team Canvas відноситься до початкового етапу організації проєкту. Більш детально про цей засіб можна дізнатися у розділі 1. До представлення, яке відображається при виконанні функції, передається дані чотирьох моделей з БД: «Goal», «Activity», «Value» та «Skill». Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається параметр id проєкту, до якого належать перелік властивостей засобу Team Canvas.

Функція Evaluation (int id) призначена для отримання представлення з оцінкою організації проєкту на якому присутні два списки, що відображають позитивні та негативні аспекти організації проєкту. Оцінка організації проєкту відноситься до кінцевого етапу організації проєкту. До представлення, яке відображається при виконанні функції, передається дані двох моделей з БД: «Positive Point» та «Negative Point». Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається параметр id проєкту над яким проводиться оцінка.

Функція Logistics (int id) призначена для отримання представлення з витратами та прибутками проекту. Ця функція дозволяє зробити висновки про загальний бюджет проекту. Ведення бухгалтерії проекту відноситься до основного етапу організації. До представлення, яке відображається при виконанні функції, передається дані двох моделей з БД: «Expense» та «Profit». Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проекту. До функції передається параметр id проекту, бюджет якого потрібно розрахувати.

Функція ProjectReports (int id) призначена для повернення користувачу списку зі звітами, залишеними користувачами певного проекту. Звіт описує роботу над організацією проекту, яку було виконано членом команди певного. Ведення звітності проекту відноситься до основного етапу організації. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проекту. До функції передається параметр id проекту.

Функція CommentWall () призначена для зображення повідомлень, залишених користувачами на головній сторінці застосунку. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проекту. Функція немає параметрів.

Функція KTs (int id) призначена для повернення користувачу списку заповнених форм передачі досвіду членів команди певного проекту. Заповнення форми передачі досвіду відноситься до кінцевого етапу організації проекту. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проекту. До функції передається параметр id проекту.

Функція getFinalBudget (int id, int sumProfit, int sumExpense) призначена для отримання загального бюджету певного проекту. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проекту. До функції передається три параметри: параметр id проекту, параметр sumProfit, що відповідає сумі прибутків проекту та параметр sumExpense, що відповідає сумі витрат проекту.

Функція `getPricePerPerson(int id, int numPax, int sumExpense)` призначена для підрахунку витрат розрахована на одного учасника заходу. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається три параметри: параметр `id` проєкту, параметр `numPax`, який позначає кількість учасників проєкту та параметр `sumExpense`, який позначає загальні витрати певного проєкту.

Функція `getFiles(int id)` призначена для отримання представлення з списком файлів певного проєкту. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається параметр `id` проєкту до якого були завантажені файли.

Функція `countGoal (int id, int goal, int sumProfit)` призначена для підрахунку залишку потрібних коштів. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. До функції передається параметр `id` проєкту, параметр `goal`, який позначає ціль по прибуткам проєкту та параметр `sumProfit`, який, що відповідає сумі прибутків проєкту.

JavaScript функція `TCmovementFucntion ()` призначена для надання користувачеві можливості пересування елементів засобу `Team Canvas`. У цій функції використаний компонент `draggable` бібліотеки `JQuery UI`. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. Функція немає параметрів.

JavaScript функція `SWOTmovementFunction ()` призначена для надання користувачеві можливості пересування елементів засобу `SWOT-аналіз`. У цій функції використаний компонент `draggable` бібліотеки `JQuery UI`. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. Дана функція має структуру ідентичну структурі функції `TCmovementFucntion ()`. Функція немає параметрів.

JavaScript функція `kanbanControlFunction ()` призначена для надання користувачеві можливості змінювати положення елементів в зв'язаних списках

					ІТ61.090БАК.004 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дат		

Канбан-дошки. У цій функції використані компоненти `draggable` та `sortable` бібліотеки `jQuery UI`. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. Функція немає параметрів.

JavaScript функція `kanbanSaveOrder ()` призначена для збереження порядку елементів у списках Канбан-дошки. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. Функція немає параметрів.

JavaScript функція `evalPointControlFucntion ()` призначена для надання користувачеві можливості пересування елементів в Оцінці організації проєкту. У цій функції використаний компонент `draggable` бібліотеки `jQuery UI`. Ця функція доступна для користувачів з правами члена команди та з правами головного організатора проєкту. Функція немає параметрів.

Висновок до розділу

В даному розділі було розглянуто технології, за допомогою яких було реалізовано програмний продукт. Також описано детально діаграму класів з описом основних класів застосунку. Було також детально описано діаграму компонентів застосунку. Було наведено специфікацію до основних функцій системи.

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		41

4 РОЗРОБКА КОМПОНЕНТІВ КЕРУВАННЯ ПРОЄКТАМИ

4.1 Канбан-дошка

Канбан-дошка — це засіб для делегування завдань по керуванню проєкту членам команди. Більш детально цей засіб описано в розділі 1. Одним з завдань для даного проєкту було розробити функціональність Канбан-дошки, якою могли би користуватись члени команди та головний організатор проєкту, користуючись застосунком. Виконання цього завдання передбачає виконання наступних задач:

- аналіз функціональності Канбан-дошки та її особливостей;
- аналіз існуючих рішень ;
- розробка серверної частини засобу, тобто зберігання сутностей завдань та зберігання порядку завдань у дошках у БД ;
- розробка клієнтської частини засобу;
- розробка функціональності користуванням Канбан-дошкою для двох типів користувачів: членами команди та головним організатором.

При аналізі функціональності Канбан-дошки було прийнято рішення реалізувати її у вигляді трьох динамічних списків, які зв'язані між собою, тобто які можуть передавати між собою завдання з яких вони складається. Ці списки позначають завдання, які повинні виконатися командою під час організації проєкту та поділяються на такі категорії: «потрібно виконати», «у процесі виконання» та «виконано».

При аналізі існуючих рішень також були варіанти Канбан-дошок з більшою варіацією списків, але для даного проєкту буде достатньо реалізації трьох основних списків по організації певного проєкту.

Розробка серверної частини складається з двох етапів: створення сутності Завдання та зберігання порядку завдань у дошках у БД. Для початку, було створення модель Завдання та відповідна міграція, яка дозволила додати до БД відповідну таблицю. Після додаванні таблиці з Завданнями, був створений

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		42

контролер, який зіставляє запити користувача з операціями над БД. Саме тут прописується, що представлення, яке бачить користувач буде мати усі завдання, які відносяться до певного проєкту. Внесення порядку завдань у списках до БД відбувається за допомогою методу бібліотеки jQuery UI, яка належить компоненту sortable. Метод має назву serialize, його призначення є в тому, щоб надсилати інформацію про зміну позиції елемента списку до БД.

Клієнтська частина засобу Канбан-дошки розробляється за допомогою бібліотеки jQuery UI, так як вона містить усі потрібні компоненти. Компоненти jQuery UI, які були використані: sortable та droppable. Ці компоненти відповідають за створення послідовних списків та пересування одних елементів на інші відповідно.

Користування Канбан-дошкою відрізняється залежно від типу користувачів, тобто користування Канбан-дошкою головним організатором проєкту відрізняється від користуванням Канбан-дошкою членами команди. Користувач з правами головного організатора має можливість створювати завдання для Канбан-дошки, редагувати їх та видаляти. Члени команди мають можливість змінювати статус завдання, змінюючи список, якому воно належить («у процесі виконання», «потрібно виконати» або «виконано»).

Реалізацію засобу Канбан-дошка можна побачити на рисунку 4.1. Можна побачити, що Канбан-дошка виглядає як три зв'язані між собою списки. Елементи можуть змінювати списки залежно від готовності завдання.

KANBAN BOARD

Create New



Рисунок 4.1 — Реалізація клієнтської частини засобу Канбан-дошка

4.2 Team Canvas

Team Canvas — це засіб керування проектами на початковому етапі. Засіб Team Canvas відповідає за організацію команди на початковому етапі проекту та визначення цілей, мети, правил роботи та навиків. Більш детально цей засіб описаний в Розділі 1. Одним з завдань для дипломного проекту було розробити функціональність засобу Team Canvas, якою могли би користуватись головний організатор та члени команди проекту, користуючись застосунком.

Виконання цього завдання передбачає виконання наступних задач:

- аналіз функціональності засобу Team Canvas та його особливостей;
- аналіз існуючих рішень;
- розробка серверної частини засобу, тобто зберігання сутностей засобу (навики, активності, цілі, цінності);
- розробка клієнтської частини засобу, тобто користування Team Canvas головним організатором.

Засіб Team Canvas представляє собою комбінацію різних списків, які відповідають за організацію команди на початковому етапі проекту. При аналізі функціональності засобу було прийнято рішення реалізувати його у вигляді чотирьох основних списків «Активності», «Цілі», «Цінності» та «Навики» для простоти вигляду представлення застосунку. Зазвичай команди користуються засобом Team Canvas використовуючи великий лист паперу А3 або А2, який поділений на декілька частин, що відповідають характеристиками команди та проекту, які будуть досліджуватися головним організатором та членами команди. Вигляд засобу Team Canvas у застосунку буде схожий – він буде представляти собою лист розділений на чотири частини.

Розробка серверної частини складається з створення чотирьох сутностей: Цінність, Ціль, Активність та Навик. Було створено чотири моделі, що відповідають потрібним сутностям та відповідні міграції, які дозволили додати до БД відповідні таблиці. Також була створена ще одна модель ViewModel, яка

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		44

поєднує у собі списки відповідних сутностей для можливості відтворення засобу Team Canvas в одному представленні. Після додаванні таблиць, були створені чотири контролера, які зіставляють запити користувача з операціями над сутностями у БД. Також був створений ще один контролер, який відповідає за представлення моделі ViewModel, що і є представленням засобу Team Canvas.

Використання засобу Team Canvas здійснюється головним організатором проекту. Клієнтська частина засобу Team Canvas розробляється за допомогою CSS та бібліотеки jQuery UI. Бібліотека jQuery UI містить усі потрібні компоненти для бажаної реалізації засобу. Для відтворення функціональності засобу був використаний компонент бібліотеки jQuery UI — draggable.



Рисунок 4.2 — Реалізація клієнтської частини засобу SWOT-аналіз

Компонент draggable дозволяє пересувати елементи, використовуючи мишу. У даному випадку він використовується для пересування елементів чотирьох частин засобу, які відповідають за цілі, цінності, активності та навички команди. Компонент draggable дозволяє пересувати елементи, використовуючи мишу.

4.3 SWOT-аналіз

SWOT-аналіз — це засіб керування проєктами на початковому етапі організації. SWOT-аналіз відповідає за визначення сильних та слабких сторін, можливостей та погроз певного проєкту та відповідає за опис початкової стратегії організації проєкту. Більш детально ці засоби описані в розділі 1. Одним з завдань для даного проєкту було розробити функціональність SWOT-аналізу, якою могли би користуватись члени команди проєкту, користуючись застосунком. Виконання цього завдання передбачає виконання наступних задач:

- аналіз функціональності SWOT-аналізу та його особливостей;
- аналіз існуючих рішень;
- розробка серверної частини засобу, тобто зберігання сутностей засобу (сильні сторони проєкту, можливості, слабкі сторони проєкту та загрози);
- розробка клієнтської частини засобу, тобто користування SWOT-аналізу головним організатором проєкту.

Можна побачити, що перелік завдань для засобу SWOT-аналіз є схожим на перелік завдань засобу Team Canvas. Отже, можна зробити висновок, що їх реалізація теж може бути схожа. Так як SWOT-аналіз теж передбачає використання чотирьох списків (сильні та слабкі сторони, можливості та загрози), то одразу можна помітити, що до SWOT-аналізу можна застосувати ту саму функціональність.

На рисунку 4.3 можна побачити як засіб виглядає зі сторони користувача. Відомо, що більшість людей краще розуміють складний матеріал, коли він чітко та зрозуміло візуалізований. Можна побачити, що SWOT-аналіз має вигляд схожий на лист паперу, поділений на чотири частини. Це обумовлено тим, що такий формат допомагає візуалізації елементів SWOT-аналізу та інших засобів керування проєктами. Даний формат допоможе команді чітко проаналізувати усі елементи засобу. У розділі 5 можна побачити приклад заповненого головним організатором SWOT-аналізу.

SWOT ANALYSIS

Strenghts	Weaknesses
Opportunities	Threats

Рисунок 4.3 — Реалізація клієнтської частини засобу SWOT-аналіз

4.4 Оцінка організації проєкту

Загальна оцінка організації проєкту відповідає кінцевому етапу організації проєкту. Члени команди вказують позитивні та негативні аспекти організації проєкту для покращення роботи майбутніх команд. Збереження цієї інформації для майбутніх команд та проєктів є дуже важливою частиною передачі знань, що повинна бути присутньою в усіх організаціях.

В організаціях різного типу для оцінки організації проєкту відводиться окремий день, де команда та інші члени організації аналізують виконану роботу та перелічують позитивні аспекти організації та аспекти, які потребують покращення в майбутньому. Завдання зводиться до створення двох списків, що схоже на завдання для реалізації Канбан-дошки, але відрізняється тим, що списки не повинні бути зв'язані між собою.

Розробка серверної частини складається з створення двох сутностей: Позитивна Оцінка та Негативна Оцінка. Було створено дві моделі, що відповідають потрібним сутностям та відповідні міграції, які дозволили додати

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		47

до БД відповідні таблиці. Також була створена ще одна модель ViewModel, яка поєднує у собі списки відповідних сутностей для можливості відтворення позитивних та негативних якостей організації в одному представленні. Після додаванні таблиць, були створені два контролери, які заставляють запити користувача з операціями над сутностями у БД. Також був створений ще один контролер, який відповідає за представлення моделі ViewModel, що і є представленням засобу.

Клієнтська частина засобу розробляється за допомогою бібліотеки jQuery UI, так як вона містить усі потрібні компоненти. Компонент jQuery UI, який був використаний — sortable. Цей компонент відповідають за створення послідовних списків.

4.5 Бухгалтерія

Бухгалтерія є важливим аспектом організації проектів. При розрахунку логістики описуються усі розходи та прибутки проекту та вираховується загальний бюджет. Зазвичай для ведення логістики використовується Excel Spreadsheets, але в даному проекті буде реалізована альтернативна технологія. Одним з завдань для даного проекту було розробити таку функціональність, якою могли би користуватись члени команди, а особливо член команди, який відповідає за логістику проекту. Виконання цього завдання передбачає виконання наступних задач:

- визначення основних атрибутів для таблиці «Витрати»;
- визначення основних атрибутів для таблиці «Прибутки»;
- розробка серверної частини засобу;

Є багато можливих атрибутів, які можливо задіяти при веденні розходів проекту. Основною метою є доцільне розуміння загального бюджету проекту та розуміння наявності усіх потрібних закупок та витрат. Виходячи з цього було обрано основні атрибути, які будуть присутні у таблиці розходів проекту. До цих атрибутів входять: причина витрати, опис витрати, кількість, ціна за одиницю та

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		48

одиниця вимірювання . Знаючи ці атрибути можливо розрахувати загальну ціну за витрату та суму загальних витрат проекту.

У волонтерських організаціях зазвичай є люди, які відповідають за пошук партнерів та шляхів отримати додаткові кошти на проекти від різних фондів, грантів та інших організацій, які готові допомагати волонтерам. Це передбачає ведення ще одної таблиці. Ця таблиця буде відповідати прибуткам проекту. До атрибутів цієї таблиці будуть входити ресурс отриманих коштів та їх кількість. Знаючи ці атрибути можливо розрахувати суму прибутків проекту. Поєднуючи інформацію про суму витрат та прибутків можна зробити висновки про загальний бюджет проекту. Для того щоб ще краще розуміти матеріальне становище організації проекту можна на початковому етапі організації (наприклад при створенні Канбан-дошки) затвердити суму прибутків, на яку потрібно буде орієнтуватися головному організатору та члену команди, який відповідає за ведення логістики, для того щоб майбутній проект пройшов на достойному рівні.

Виходячи з встановлених атрибутів та потреб засобу, було прийнято рішення, реалізувати клієнтську сторону ведення логістики проекту за допомогою двох таблиць, які відображають усі перелічені атрибути витрат та прибутків певного проекту. Також повинна бути створена окрема таблиця, яка містила би загальну суму прибутків та витрат проекту, його загальний бюджет, суму прибутків, яка потрібна для організації проекту та інформація про те чи досяг поточний стан бюджету проекту поставленої мети.

Розробка серверної частини складається з створення двох сутностей: Витрата та Прибуток. Було створено дві моделі, що відповідають потрібним сутностям та відповідні міграції, які дозволила додати до БД відповідні таблиці. Також була створена ще одна модель ViewModel, яка поєднує у собі списки відповідних сутностей для можливості відтворення усіх витрат та прибутків проекту в одному представленні. Після додавання таблиць, були створені два контролера, які зіставляють запити користувача з операціями над сутностями у

					ІТ61.090БАК.004 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дат		

БД. Також був створений ще один контролер, який відповідає за представлення моделі ViewModel, що і є представленням засобу. Також були створені відповідні функції, які відповідають за підрахунок загальної суми прибутків та витрат проєкту та загального бюджету.

4.6 Вхід до застосунку

Реєстрація та авторизація користувача — це невід’ємний аспект розробки будь-якого застосунку. Дані користувача повинні зберігатися у надійному форматі та проходити валідацію належним чином. Застосунок реалізується на платформі ASP.NET Core, що дозволяє використовувати технологію ASP.NET Identity для реалізації функцій реєстрації та авторизації користувачів. На рисунку 4.4 зображено можливі функції, які доступні при користуванні технологією ASP.NET Identity. Серед них: реєстрація користувачів, авторизація, вихід з застосунку, підтвердження користувачами реєстрації за допомогою пошти, зміна паролю користувачів, реєстрація за допомогою зовнішніх ресурсів та інші функції. Для даного застосунку було обрано три основних функції: реєстрація та авторизація користувачів та можливість вийти зі свого акаунту. Ці функції відповідають поставленим завданням реалізації дипломного проєкту. В розділі 5 можна побачити реалізацію функцій технології ASP.NET Identity реєстрації та авторизації користувача.

Головна сторінка застосунку — це перша сторінка, яку бачить користувач. Вона повинна доцільно пояснювати шляхи користування усіма компонентами застосунку користувачеві та надавати базову теорію про засоби керування проєктами та як ці засоби потрібно використовувати зі сторони головного організатора та членів команди. Інтерфейс сторінки був реалізований за допомогою компоненту бібліотеки JQuery UI — accordion. Також після авторизації користувача на головній сторінці можливо залишати повідомлення для інших членів команди.

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		50

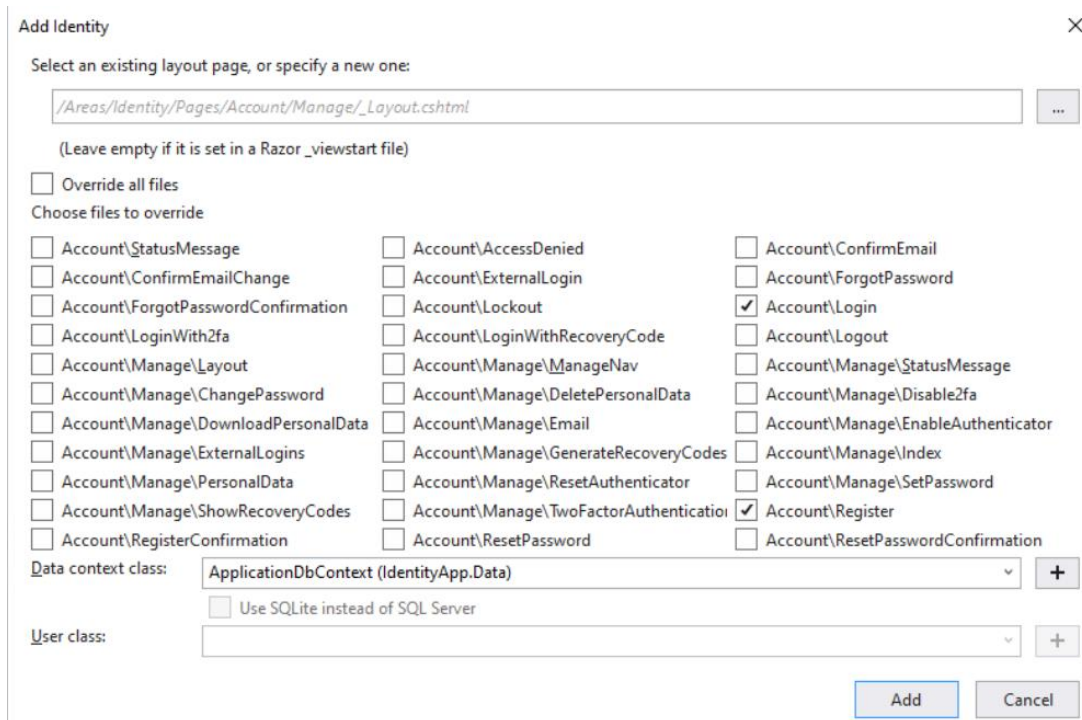


Рисунок 4.4 — Додавання технології ASP.NET Identity до проекту

Висновок до розділу

В даному розділі було описано розробку компонентів користувацького інтерфейсу. При описі розробки компонентів було проаналізовано функціональність засобів, описано розробку серверної та клієнтської частини засобів та описано розробка функціональності користуванням засобів зі сторони головного організатора проекту та членів команди.

5 ОГЛЯД ЗАСТОСУНКУ ЗІ СТОРОНИ КОРИСТУВАЧА

5.1 Керівництво користувача

У цьому підрозділі буде продемонстровано процес реєстрації користувача, користування Канбан-дошкою для слідкування за організацією завдань щодо проекту, користуванням SWOT-аналізом проекту, оцінкою організації проекту та веденням бухгалтерії проекту.

На рисунку 5.1 можна побачити вікно реєстрації та авторизації користувача. Користувачу потрібно ввести свої дані аби увійти до застосунку. Система відповідає правилам валідації даних.

Register

Create a new account.

- Passwords must have at least one digit ('0'-'9').
- Passwords must have at least one lowercase ('a'-'z').
- Passwords must have at least one uppercase ('A'-'Z').

Email

ksenia123@gmail.com

Password

.....

Confirm password

.....

Register

Log in

Use a local account to log in.

Email

kseniagarshina@gmail.com

Password

.....

Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)

Рисунок 5.1 — Демонстрація реєстрації користувача

Після виконання успішної авторизації користувач отримає доступ до списку проектів та можливостей застосунку та можливість вийти зі свого акаунту.

Ще до авторизації користувача, можна побачити головну сторінку застосунку. На ній здійснюється опис основних засобів організації проектів та

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		52

поради для користуванням застосунком. Після авторизації користувача на головній сторінці також можна побачити повідомлення, залишені членами команди. Також після авторизації користувач має можливість вийти з застосунку.

При виборі будь-якого проєкту зі списку, можна побачити відомості про цей проєкт та посилання на усі можливі компоненти. Переходячи по посиланню «Завдання», можна потрапити до засобу Канбан-дошка даного проєкту. Також можна побачити перелік завдань конкретного проєкту, які можна редагувати, додавати та видаляти з бази даних, якщо користувач має права головного організатора.

На рисунку 5.2 можна побачити Канбан-дошку проєкту з завданнями. Можна перетягувати завдання від одного списку в інший, в залежності від етапу опрацювання конкретного завдання («Потрібно виконати», «У процесі виконання», «Виконано»).

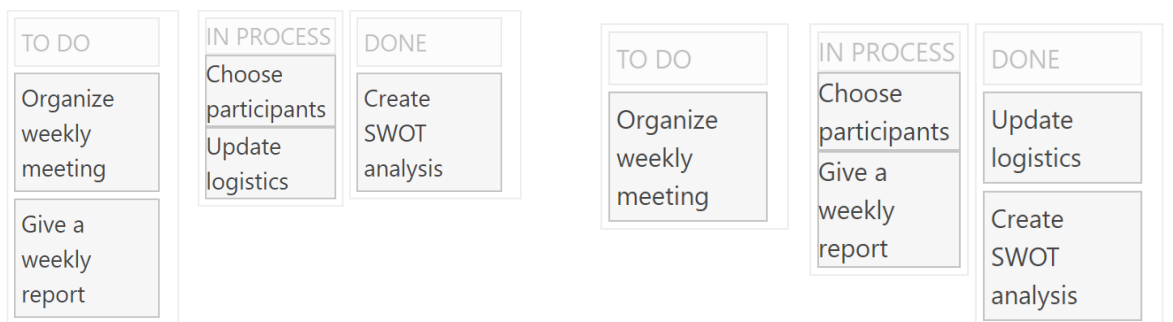


Рисунок 5.2 — Канбан-дошка проєкту

На рисунку 5.3 зображено SWOT-аналіз певного проєкту. SWOT-аналіз містить чотири різних типи списків: сильні та слабкі сторони проєкту, можливості та загрози. Усі ці списки можуть бути заповнені головним організатором переходячи до відповідних форм.

Схожий інтерфейс має засіб Team Canvas проєкту. Функціональність засобу Team Canvas схожа на функціональність засобу SWOT-аналіз. Засіб Team

Canvas входить до початкового етапу організації проєкту разом з заповненням Канбан-дошки та створенням SWOT-аналізу. Більш детально про ці три засоба можна дізнатися у розділі 1. Засіб Team Canvas має такі списки як: цілі, цінності, навички та активності. Усі ці списки можуть бути заповнені головним організатором, переходячи до відповідних форм. Елементи засобів SWOT-аналізу та Team Canvas можна вільно пересувати у границях відповідних списків. Також більш детально про розробку цих засобів можна дізнатися у розділі 5.

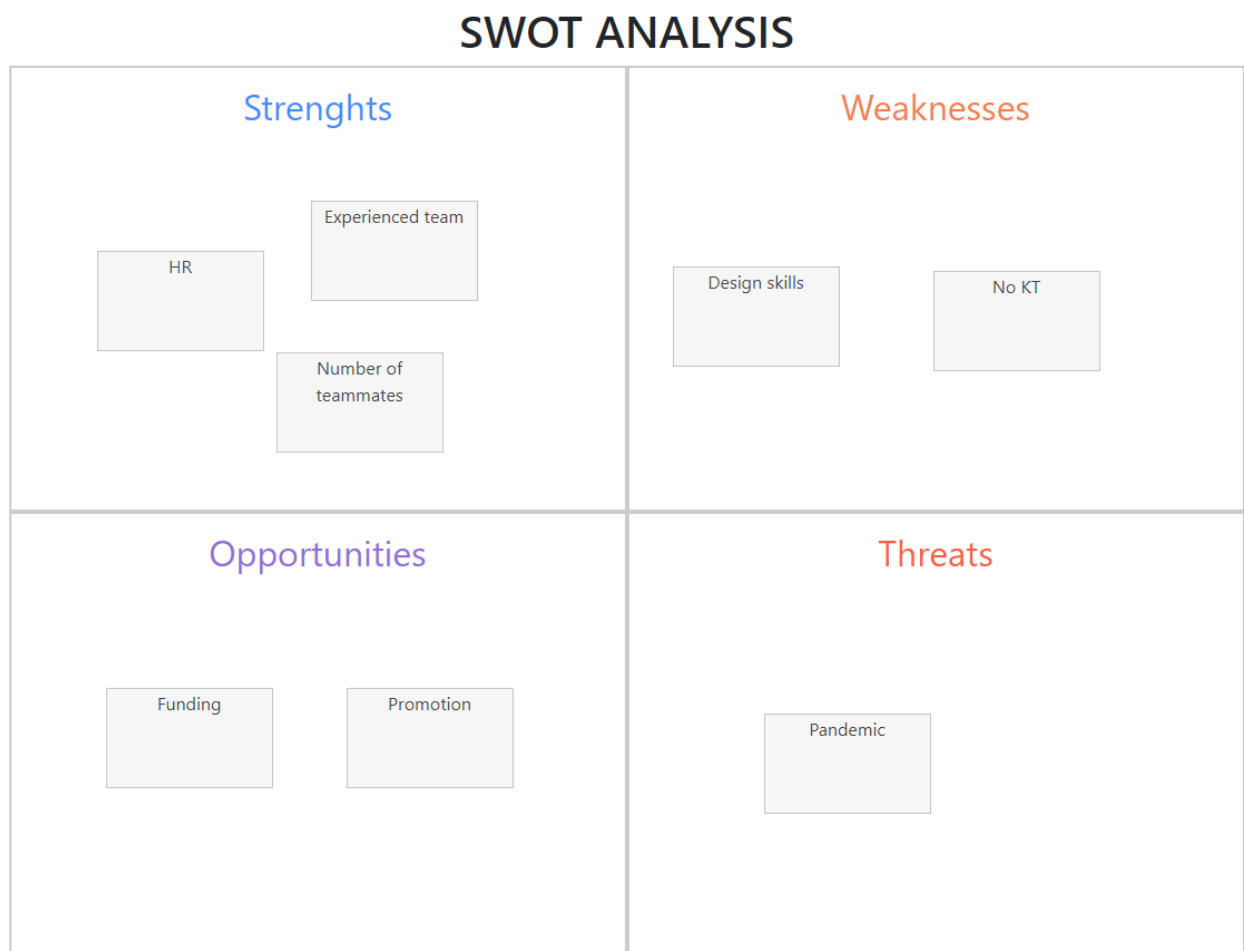


Рисунок 5.3 — SWOT-аналіз проєкту

Від відомостей про певний проєкт можна перейти до списку звітів залишених членами команди певного проєкту. Члени команди можуть ділитися своїм прогресом щодо організації проєкту. Створюючи новий звіт, користувач заповнює форму зі своїм ім'ям, датою заповнення звіту та змістом самого звіту.

Також від відомостей про певний проєкт можна перейти до списку форм передачі досвіду. Користувачі заповнюють форму, відповідаючи на питання про їх досвід на позиції в певному проєкті, досвід у команді, що потрібно виправити на наступний раз, що потрібно залишити та загальні поради людям, які будуть займати цю ж позицію у майбутньому.

Наступний засіб кінцевого етапу організації проєкту — це загальна оцінка організації проєкту. На рисунку 5.4 зображено приклад оцінки організації певного проєкту. Зеленим кольором позначені позитивні відгуки про організацію проєкту, а червоним — негативні.



Рисунок 5.4 — Оцінка організації проєкту

На рисунку 5.5 зображено дві таблиці, які описують розходи та прибутки певного проєкту. В таблиці «Розрахунки» перелічені такі атрибути як: певна витрата, ціна, кількість, спосіб використання, одиниця вимірювання та проєкт. Розрахунки, які використані у даному прикладі, включають транспортування та їжу. В таблиці «Прибутки» перелічені такі атрибути як: ресурс та сума коштів. За допомогою інформації про прибутки та розходи можна зробити висновки про загальний бюджет проєкту.

LOGISTICS

Item	Quantity	PricePerItem	Purpose	Unit	Project	
Bus Rent	2	700	Transportation on the weekend	None	Data Science Course	Edit Details Delete
Water	50	10	Water for participants during meals	Litres	Data Science Course	Edit Details Delete
Metro Passes	10	50	Transportation to lectures	None	Data Science Course	Edit Details Delete

Resource	Amount	Project	
IT Company	5000	Data Science Course	Edit Delete
Gov Fond	2000	Data Science Course	Edit Delete

Рисунок 5.5 — Таблиця прибутків та розходів певного проєкту

5.2 Тестування застосунку

Метою тестувань являється перевірка відповідності функцій застосунку по керуванню проєктам вимогам поставлених завдань. Розглянемо основні функції застосунку та перевіримо їх відповідність вимогам. Тестування застосунку можна поділити на три типи:

- тестування можливості реєстрації та авторизації користувача;
- тестування роботи користувача з даними;
- тестування інтерфейсу користувача.

До тестування можливості реєстрації та авторизації користувача входять такі тестування: тестування реєстрації користувача, тестування валідації даних при реєстрації, тестування авторизації користувача, тестування підтвердження реєстрації за допомоги електронної пошти, тестування виходу користувача з застосунку. Ці тестування можна побачити у додатку А. Результати тестувань: усі перелічені тестування пройшли успішно. Отже, можна вважати, що реєстрація та авторизація користувачів застосунку працює належним чином.

До тестування роботи користувача з даними входять такі тестування:

тестування створення нового проєкту головним організатором, тестування редагування інформації існуючого проєкту, тестування видалення існуючого проєкту головним організатором, тестування додавання нового завдання до проєкту користувачем, тестування додавання звіту про виконану роботу користувачем, тестування про додавання оцінки організації певного проєкту користувачем, тестування про додавання файлу до певного проєкту користувачем, тестування додавання форми передачі досвіду користувачем, тестування можливості користувача додавання інформації про нові розходи, тестування можливості користувача додавання інформації про нові прибутки . Ці тестування можна побачити у додатку А. Результати тестувань: усі перелічені тестування пройшли успішно. Отже, можна вважати, що робота користувача з даними у застосунку працює належним чином.

До тестування інтерфейсу користувача входять такі тестування: тестування зміни списку, до якого належить завдання, у засобі Канбан-дошка, тестування можливості пересування елементів користувачем у засобі SWOT-аналіз, тестування можливості пересування елементів користувачем у засобі Team Canvas. Ці тестування можна побачити у додатку А. Результати тестувань: усі перелічені тестування пройшли успішно. Отже, можна вважати, що інтерфейс користувача у застосунку працює належним чином.

Висновок до розділу

В даному розділі було сформоване керівництво користувача застосунку, де було визначено основні дії, що може здійснювати користувачі у застосунку. Було наведено п'ять рисунків, які показують як працює застосунок зі сторони користувача. Також були виконані та описан три типи тестування застосунку: тестування можливості реєстрації та авторизації, тестування роботи користувача з даними та тестування інтерфейсу користувача.

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		57

ВИСНОВКИ

У ході виконання дипломного проєкту були детально розглянуті питання, які виникають у процесі організації проєктів. Були виділені основні ключові етапи, сформовано предметне середовище, визначено та описано процеси діяльності, а також розглянуті наявні аналоги системи. Крім того, була сформована функціональна модель застосунку — описані користувачі системи, їх ролі та можливі дії у системі.

Застосунок був розроблений за допомогою середовища ASP.NET Core. Для збереження даних використовувалась технологія EF Core. Клієнтська сторона застосунку була розроблена за допомогою бібліотеки мови JavaScript, а саме бібліотеки jQuery UI.

Був реалізований застосунок для керування проєктами на початковому, основному та кінцевому етапах. Також були реалізовані наступні засоби керування проєктами: «Канбан-дошка», «SWOT-аналіз» та «Team Canvas».

Було детально описана реалізація таких основних компонентів інтерфейсу користувача застосунку: Канбан-дошка, засіб Team Canvas, SWOT-аналіз, оцінка організації проєкту, ведення логістики та головна сторінка застосунку.

Була наведена детальна інструкція користувача по використанню додатка для ведення проєктів. Проведено та описано тестування основних засобів застосунку.

Можливі аспекти покращення та розширення функціональності застосунку включають:

- розробка інтерактивної діаграми Ганта;
- розширення функціональності Канбан-дошки;
- підтримка відео чатів у застосунку;
- покращення графічного інтерфейсу користувача;
- розробка застосунку для використання на мобільній платформі;
- розробка відправки повідомлень з нагадуванням про виконання завдань

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		58

членам команди проекту на пошту.

Перелічені розширення та покращення застосунку зможуть надати більше зручності у користуванні та організації великих проєктів організаціям або на індивідуальному рівні. Діаграма Ганта є важливим методом керування процесами та зможе бути використана членами команди для слідкування за етапом готовності проєкту поруч з Канбан-дошкою. Також останнім часом відео чати набирають великої популярності, тому додавання цієї функціональності до застосунку стане великою перевагою серед інших застосунків для керування проєктами, що не мають цієї опції. Даний застосунок виконує усю потрібну функціональність для керування проєктами, проте можливий більш креативний підхід до реалізації графічного інтерфейсу користувача, що дозволить користувачам ще з більшою легкістю користуватися застосунком. Крім того, на сьогоднішній день більшість застосунків доступні на багатьох різних платформах, тож реалізація застосунку на мобільній платформі буде також великою перевагою.

Застосунок виконує усі поставлені задачі та етапи організацій проєктів та може бути використаний для керування великими проєктами організацій різного типу.

					ІТ61.090БАК.004 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ABOUT BEST. [Електронний ресурс]. — Режим доступу: <https://www.best.eu.org/aboutBEST/welcome.jsp>
2. Team Canvas - Get Your Team on the Same Page. [Електронний ресурс]. — Режим доступу: <http://theteamcanvas.com/>
3. SWOT Analysis: How to Develop a Strategy For Success. [Електронний ресурс]. — Режим доступу: https://www.mindtools.com/pages/article/newTMC_05.htm
4. What Is Kanban? An Overview Of The Kanban Method. [Електронний ресурс]. — Режим доступу: <https://www.digite.com/kanban/what-is-kanban/>
5. What is a Gantt Chart? Gantt Chart Software, Information, and History. [Електронний ресурс]. — Режим доступу: <https://www.gantt.com/>
6. What is Trello? A guide to Atlassian's collaboration and work management tool. [Електронний ресурс]. — Режим доступу: <https://www.computerworld.com/article/3226447/what-is-trello-a-guide-to-atlassians-collaboration-and-work-management-tool.html>
7. Trello Collaborate visually on any project. [Електронний ресурс]. — Режим доступу: <https://trello.com/>
8. Basecamp: Project Management & Team Communication [Електронний ресурс]. — Режим доступу: <https://basecamp.com/>
9. Manage your team's work, projects, & tasks online · Asana [Електронний ресурс]. — Режим доступу: <https://asana.com/>
10. Visual Project and Process Management. [Електронний ресурс]. — Режим доступу: <https://casual.pm/>
11. Project Management Software For Professionals - OmniPlan - The Omni Group. [Електронний ресурс]. — Режим доступу: <https://www.omnigroup.com/omniplan/>

					ІТ61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		60

12. Podio: Project Management and Collaboration Software. [Электронный ресурс]. — Режим доступа: <https://podio.com/site/en>
13. ASP.NET Core - новая эпоха в развитии ASP.NET. [Электронный ресурс]. — Режим доступа: <https://metanit.com/sharp/aspnet5/1.1.php>
14. Overview of Entity Framework Core. [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/en-us/ef/core/>
15. ASP.NET Core Identity. [Электронный ресурс]. — Режим доступа: <https://metanit.com/sharp/aspnet5/16.1.php>
16. ASP.NET MVC Architecture. [Электронный ресурс]. — Режим доступа: <https://www.tutorialsteacher.com/mvc/mvc-architecture>
17. Введение в jQuery. [Электронный ресурс]. — Режим доступа: <https://metanit.com/web/jquery/1.1.php>
18. JQueryUI — Overview. [Электронный ресурс]. — Режим доступа: https://www.tutorialspoint.com/jqueryui/jqueryui_overview.htm

					IT61.090БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		61