

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут комп'ютерних систем
Кафедра інформаційних систем

КУРСОВА РОБОТА

з дисципліни «Технології створення програмних продуктів»

за темою

«Розробка web-ресурсу WishList»

Пояснювальна записка до етапів визначення вимог до програмного продукту та
планування процесів розробки програмного продукту

Виконали:
студенти 3-го курсу
групи ЛАІ-194
Бицкало І. І.
Дем`яненко І. Р.
Перевірив:
Блажко О. А.

Одеса-2022

Анотація

В курсовій роботі розглядається процес створення програмного продукту «Розробка web-ресурсу WishList» на етапах визначення вимог до програмного продукту та планування процесів розробки.

Робота виконувалась в команді з двох учасників: Бицкало І. І. та Дем'яненко І. Р..

Робота пов'язана з такими матеріальними потребами споживача, як формування варіантів подарунків для будь-яких днів та свят. Аналіз вказаних потреб визначив інформаційну потребу – отримати інформацію для кращого розуміння, який подарунок людина хоче та коли.

При визначені ступеня готовності існуючих програмних продуктів до вирішення інформаційної потреби проаналізовано наступні програмні продукти: Gift Whaleh, My Wishlist, Gifster.

Поточну версію пояснювальної записки до результатів роботи розміщено на *GitHub*-репозиторії за адресою:

https://github.com/jolynutella/SW_Bytskalo_Demyanenko_LAI194

Перелік скорочень

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

ПП – програмний продукт

UML – уніфікована мова моделювання

Зміст

- 1 Вимоги до програмного продукту
 - 1.1 Визначення потреб споживача
 - 1.1.1 Ієрархія потреб споживача
 - 1.1.2 Деталізація матеріальної потреби
 - 1.2 Бізнес-вимоги до програмного продукту
 - 1.2.1 Опис проблеми споживача
 - 1.2.1.1 Концептуальний опис проблеми споживача
 - 1.2.1.2 Опис цільової групи споживача
 - 1.2.1.3 Метричний опис проблеми споживача
 - 1.2.2 Мета створення програмного продукту
 - 1.2.2.1 Проблемний аналіз існуючих програмних продуктів
 - 1.2.2.2 Мета створення програмного продукту
 - 1.2.3 Назва програмного продукту
 - 1.2.3.1 Гасло програмного продукту
 - 1.2.3.2 Логотип програмного продукту
 - 1.3 Вимоги користувача до програмного продукту
 - 1.3.1 Історія користувача програмного продукту
 - 1.3.2 Діаграма прецедентів програмного продукту
 - 1.3.3 Сценаріїв використання прецедентів програмного продукту
 - 1.4 Функціональні вимоги до програмного продукту
 - 1.4.1. Багаторівнева класифікація функціональних вимог
 - 1.4.2 Функціональний аналіз існуючих програмних продуктів
 - 1.5 Нефункціональні вимоги до програмного продукту
 - 1.5.1 Опис зовнішніх інтерфейсів
 - 1.5.1.1 Опис інтерфейсів користувача
 - 1.5.1.1.1 Опис INPUT-інтерфейсів користувача
 - 1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача
 - 1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

- 1.5.1.3 Опис програмних інтерфейсів
- 1.5.1.4 Опис інтерфейсів передачі інформації
- 1.5.1.5 Опис атрибутів продуктивності
- 2 Планування процесу розробки програмного продукту
 - 2.1 Планування ітерацій розробки програмного продукту
 - 2.2 Концептуальний опис архітектури програмного продукту
 - 2.3 План розробки програмного продукту
 - 2.3.1 Оцінка трудомісткості розробки програмного продукту
 - 2.3.2 Визначення дерева робіт з розробки програмного продукту
 - 2.3.3 Графік робіт з розробки програмного продукту
 - 2.3.3.1 Таблиця з графіком робіт
 - 2.3.3.2 Діаграма Ганта
- 3 Проектування програмного продукту
 - 3.1 Концептуальне та логічне проектування структур даних програмного продукту
 - 3.1.1 Концептуальне проектування на основі *UML*-діаграми концептуальних класів
 - 3.1.2 Логічне проектування структур даних
 - 3.2 Проектування програмних класів
 - 3.3 Проектування алгоритмів роботи методів програмних класів
 - 3.4 Проектування тестових сценаріїв верифікації роботи програмних модулів
 - 3.4.1 Проектування тестових сценаріїв верифікації функціональних вимог
 - 3.4.2 Проектування тестових сценаріїв верифікації нефункціональних вимог
 - 3.4.3 Створення матриці відповідності вимог до програмного продукту
- 4 Конструювання програмного продукту
 - 4.1 Особливості конструювання структур даних

4.2 Особливості конструювання програмних модулів

4.2.1 Конструювання програмної структури з урахуванням спеціалізованого *Framework* для *FrontEnd*-компонент архітектури (за наявності)

4.2.2 Конструювання програмних класів (за наявності об'єктно-орієнтованого програмування)

4.2.3 Конструювання алгоритмів методів програмних класів або процедур/функцій

4.2.4 Особливості використання спеціалізованих програмних бібліотек та API (за наявності)

4.3 Модульне тестування програмних модулів

5 Верифікація програмного продукту

5.1 Тестування апаратно-програмних інтерфейсів програмного продукту

5.2 Тестування інтерфейсу користувача програмного продукту

5.3 Тестування часу реакції програмного продукту на дії користувача

6 Розгортання та валідація програмного продукту

6.1 Інструкція з встановлення системного програмного забезпечення

6.2 Інструкція з використання програмного продукту

6.3 Результати валідації програмного продукту

Висновки

1 Вимоги до програмного продукту

1.1 Визначення потреб споживача

1.1.1 Ієрархія потреб споживача

Відомо, що в теорії маркетингу потреби людини можуть бути представлені у вигляді ієрархії потреб ідей американського психолога Абрахама Маслоу включають рівні:

- фізіологія (вода, їжа, житло, сон);
- безпека (особиста, здоров'я, стабільність),
- приналежність (спілкування, дружба, любов),
- визнання (повага оточуючих, самооцінка),
- самовираження (вдосконалення, персональний розвиток).

На рисунку 1.1 представлено одну ієрархію потреби споживача, яку хотілося б задовольнити, використовуючи майбутній програмний продукт.

Потребою споживача є необхідність у самовираженні через подарунок, та у прояві уваги до іншої людини.



Рис. 1.1 – Ієрархія потреби споживача

1.1.2 Деталізація матеріальної потреби

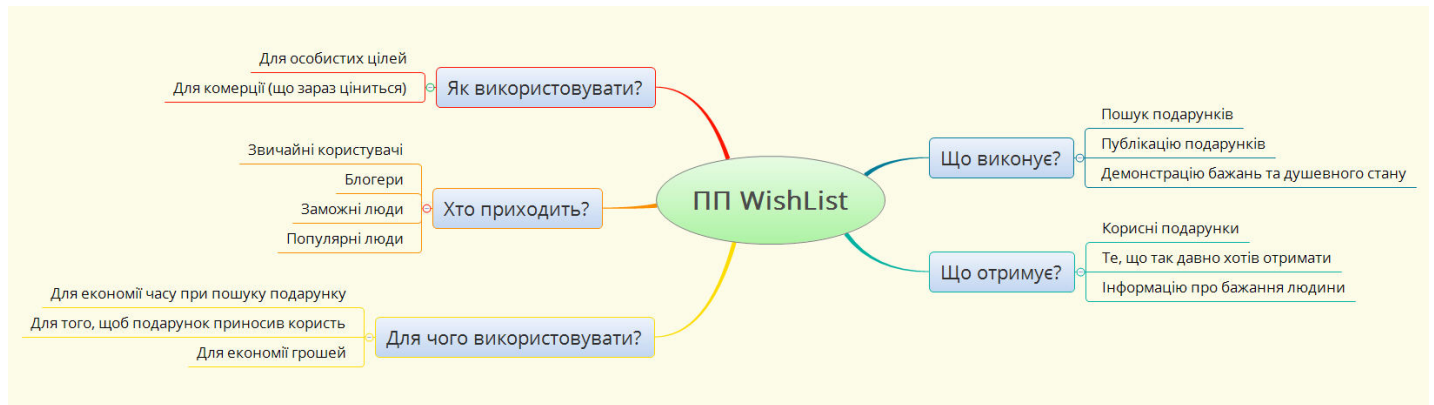


Рис. 1.2 – Mind map ПП «WishList»

1.2 Бізнес-вимоги до програмного продукту

1.2.1 Опис проблеми споживача

1.2.1.1 Концептуальний опис проблеми споживача

«Wishlist» – термін, який з’явився завдяки західним вченням про матеріалізації і візуалізації бажань і означає список, в якому прописані всі бажання людини, починаючи від дрібниць і необхідних речей і закінчуючи такими глобальними мріями як шикарний будинок, багатство, щастя і т.п.

Даний список може бути оформлений як завгодно: на аркуші паперу, на стіні кімнати, на магнітній дошці. З розвитком інтернету список побажань все частіше представляють на спеціалізованих wishlist-сайтах і, відповідно, термін «wishlist» слід більше відносити до інтернет-терміну і співвідносити його з сайтами, за допомогою яких можна створювати свої списки бажань.

Таким чином, можна формувати свої бажання онлайн і підказувати близьким, що саме подобається створювачу списку. Тому було обрано таку тему, яка є дуже актуальною зараз.

1.2.1.2 Опис цільової групи споживачів

ЦА можна розбити на 3 групи:

- Молоді люди та студенти 18-27 років, які створюють основну масу користувачів сервісу;
- Середній вік 30-55 років, створюють середню масу користувачів;
- Літні люди 55-70 років, найменша маса користувачів сервісу.

Цільовою аудиторією будуть молоді люди в віці 18-27 років, тому, що в цьому віці люди ознайомлені із технікою та різними сервісами якісніше, частіше ніж люди похилого віку.

1.2.1.3 Метричний опис проблеми споживача

Складність вибору = $(P_a * P_p) * 100$,

де P_a – користувачі, які вибирають подарунок;

P_p – Подарунки, які можна вибрати.

Анкетування проводилося на 15 студентах потоку AI, опитування показало, що більшість того, хто використовуватиме, це зазвичай дівчата. Тому, що вони хвилюються на рахунок подарунку більше ніж хлопці.

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

Таблиця 1.1 – Аналіз існуючих програмних продуктів і ПП, що розробляється

Характеристика системи	Gift Whale	My Wishlist	Giftster	ІС, що розроблюється
Функція закритого та відкритого списку	+	+	+	+
Функція рекомендацій списків друзів	-	-	-	+
Додатковий мобільний додаток	+	-	+	-
Функція пріоритетності подарунку	-	-	+	+
Мобільна адаптація	+	-	-	+
Пріоритетність за ціною / бажанням / назвою	-	+	+	+
Функція бронювання подарунків	-	-	-	+

1.2.2.2 Мета створення програмного продукту

Метою роботи є розробка веб-сервісу для формування побажань для свят, на якому користувач зможе розміщувати свої бажання подарунків і будь-яка людина, яка перейде за посиланням, зможе виконати його мрію і подарувати йому дійсно корисний подарунок, який буде ним використовуватися, також метою є скорочення часу на пошук подарунків.

1.2.3 Назва програмного продукту

«WishList»

1.2.3.1 Гасло програмного продукту

«Зробимо ваше свято ціннішим та простішим»

1.2.3.2 Логотип програмного продукту



Рис. 1.3 – Логотип ПП «WishList»

1.3 Вимоги користувача до програмного продукту

1.3.1 Пригодницька історія користувача програмного продукту (за бажанням членів проектної команди)

1.3.2 Історія користувача програмного продукту

Як гість я хочу:

- Мати змогу завести новий профіль
- Мати доступ до перегляду різних профілів друзів
- Мати змогу виставити фільтри для пошуку серед подарунків
- Мати змогу редагувати свій профіль
- Вести переписку з іншими користувачами

Як користувач я хочу:

- Як користувач, Я хочу швидко відправляти і публікувати свій wishlist, щоб економити час.
- Як користувач, Я хочу бачити приклад подарунків, щоб швидше вибирати і мати наочний приклад.

- Як користувач, Я хочу мати можливість посилатися на інші сайти, щоб прискорити процес пошуку.
- Як користувач, Я хочу бачити хто з моїх друзів виставив свої бажання для того, щоб бути в курсі їх побажань.
- Як користувач, Я хочу мати функцію відзначати який подарунок я не хочу отримати, і які у мене є, щоб не отримати непотрібні речі.

1.3.3 Діаграма прецедентів програмного продукту

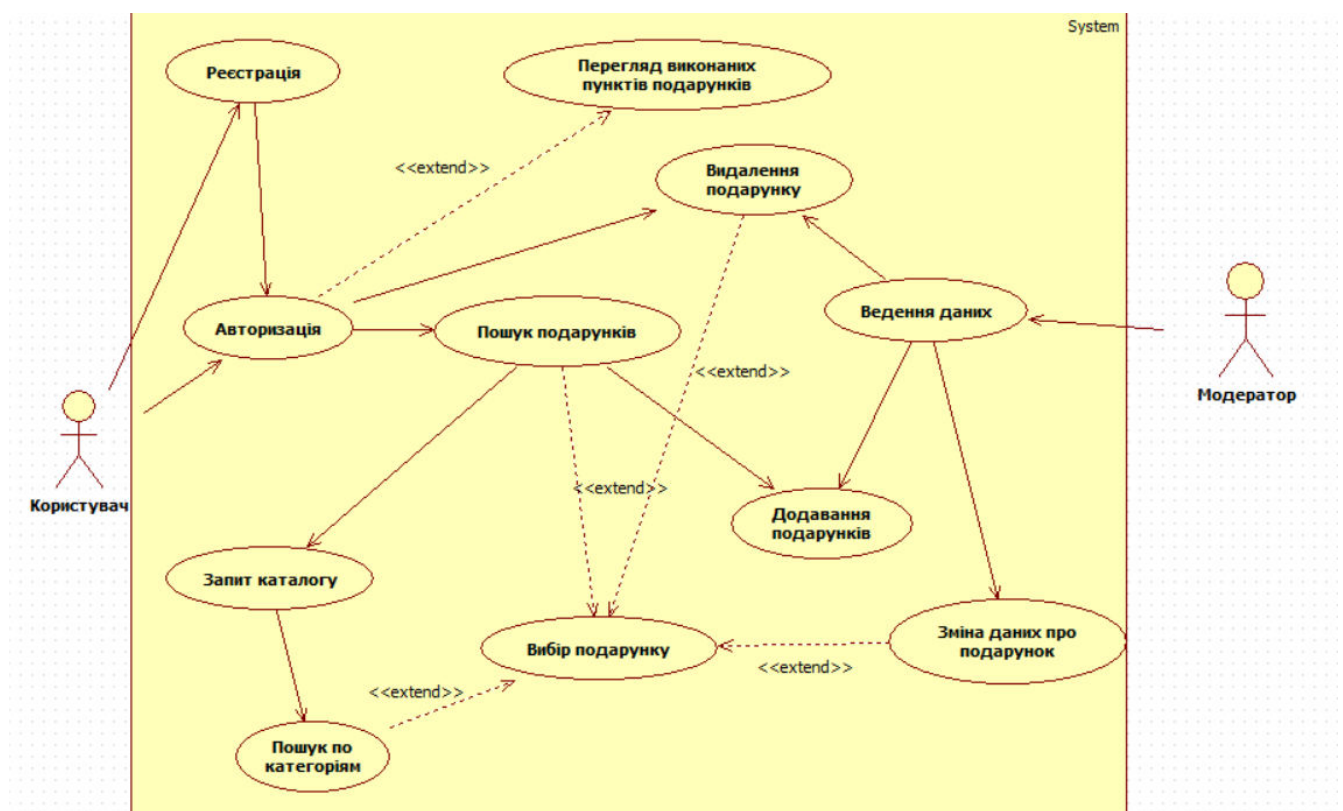


Рис. 1.4 – Діаграма прецедентів програмного продукту

1.3.4 Сценарії використання прецедентів програмного продукту

Таблиця 1.2 – 1. Прецедент ПП «Реєстрація»:

Пункт	Опис
Прецедент	Реєстрація
Передумова початку виконання прецеденту	«Відкриття сайту»
Актори як зацікавлені особи у виконанні прецеденту	Користувач
Актор-основна зацікавлена особа як ініціатор початку прецеденту	Користувач
Гарантія успіху	Реєстрація на сайті

ри
кл
ад
ос
но
вн
ог
о

успішного сценарію прецеденту «Реєстрація»:

1. При потраплянні на сайт, програма виведе діалогове вікно «Реєстрації»;
2. Користувачу необхідно ввести реєстраційні дані (ФІО, емейл, дату народження, пароль);
3. Після перевірки і підтвердження коректності даних, користувач отримує персональну сторінку.

Приклад альтернативного сценарію для успішного прикладу основного сценарію прецеденту «Реєстрація»:

1. При потраплянні на сайт, програма виведе діалогове вікно «Реєстрації»;
2. Користувачу необхідно ввести реєстраційні дані (ФІО, емейл, дату народження, пароль);
3. Після перевірки даних, виводиться повідомлення, що за цими даними не можливо зареєструватися;
4. Після виведення повідомлення про помилку, програма повертає користувача до кроку 1.

Таблиця 1.3 – 2. Прецедент ПП «Авторизація»:

Пункт	Опис
Прецедент	Авторизація
Передумова початку виконання прецеденту	«Відкриття сайту» або «реєстрація»
Актори як зацікавлені особи у виконанні прецеденту	Користувач
Актор-основна зацікавлена особа як ініціатор початку прецеденту	Користувач
Гарантія успіху	Отримати доступ до свого профіля та всіх його можливостей

П

ри
кл
ад
ос
но
вн
ог
о
ус

пішного сценарію прецеденту «Авторизувати користувача»:

1. Програма запитує у гостя параметри авторизації (адреса ел. пошти та пароль);
2. Гість вводить свої власні параметри (адреса ел. пошти та пароль) ;
3. Програма надає авторизованому користувачу доступ до прецедентів.

Приклад альтернативного сценарію для успішного прикладу основного сценарію прецеденту «Авторизувати користувача»:

1. Програма запитує у гостя його параметри авторизації (адреса ел.пошти та пароль);
2. Гість вводить свої власні параметри (адреса ел. пошти та пароль);
3. Програма виявляє, що параметри (адреса ел.пошти та пароль) користувача не є вірними
4. Програма видає повідомлення про помилку і повертається на виконання 1 кроку.

Таблиця 1.4 – 3. Прецедент ПП «Пошук подарунків»:

Пункт	Опис
Прецедент	Пошук подарунків, вибір подарунків, перегляд виконаних пунктів подарунків, запит каталогу
Передумова початку виконання прецеденту	«Пошук по категоріям», «Вибір подарунків», «Видалення подарунку», «Додавання подарунків», «Зміна даних про подарунок»
Актори як зацікавлені особи у виконанні прецеденту	Користувач
Актор-основна зацікавлена особа як ініціатор початку прецеденту	Користувач, модератор
Гарантія успіху	Знаходження подарунку

ри
кл
ад
ос
но
вн
ог
о
ус
пі
шн
ог
о

сценарію прецеденту «Пошук подарунку»:

1. Користувач заходить на сайт та відкриває вікно пошуку,
2. Користувач вводить необхідні для нього фільтри і запускає функцію пошуку;
3. Програма робить обробку по фільтрам та видає результат, якого чекає користувач.

Приклад альтернативного сценарію для успішного прикладу основного сценарію прецеденту «Пошук подарунку»:

1. Користувач заходить у програму та відкриває вікно пошуку;
2. Користувач вводить необхідні для нього фільтри і запускає функцію пошуку;
3. Програма виявляє, що по даним фільтрам не має ні одного подарунку;
4. Програма видає повідомлення про пустий запит і просить перевірити введені дані, повертаючись до пункту 1 або 2.

Пункт	Опис
Прецедент	Додавання подарунків
Передумова початку виконання прецеденту	«Успішна авторизація», «Ведення даних»
Актори як зацікавлені особи у виконанні прецеденту	Користувач, модератор
Актор-основна зацікавлена особа як ініціатор початку прецеденту	Користувач, модератор
Гарантія успіху	Додавання подарунку на сайт

аб
ли
ця
1.5
—
4.
Пр
ец
ед
ен
т

ПП «Додавання подарунків»:

Приклад основного успішного сценарію прецедента «Додавання подарунків»:

1. Користувач заходить в свій аккаунт та на форму додавання подарунків;
2. Через форму, додає подарунки на аккаунт;
3. Програма перевіряє на коректність даних, та після модерації побажання з'являється на аккаунті.

Приклад альтернативного сценарію для успішного прикладу основного сценарію прецеденту «Додавання подарунків»:

1. Користувач заходить в свій аккаунт та на форму додавання подарунків;
2. Через форму, додає подарунки на аккаунт;
3. Програма перевіряє на коректність даних, та після модерації не з'являється на аккаунті;
4. Програма повідомляє про невдалу спробу додавання та повертає до пункту 2.

1.4 Функціональні вимоги до програмного продукту

1.4.1. Багаторівнева класифікація функціональних вимог

Таблиця 1.6 – Багаторівнева класифікація функціональних вимог

Ідентифікатор функції	Функціональні залежності	Вплив на досягнення мети	Пріоритет функції
FR1-Реєстрація користувача	-	20%	M
FR1.1-Ввод параметрів реєстрації користувача	-		
FR1.2-зберігання параметрів реєстрації	-		
FR1.3-перевірка параметрів реєстрації	FR1.2		
FR2-Авторизація	-	20%	M
FR2.1-перевірка параметрів авторизації	FR1.2		
FR3-Внесення подарунків	FR2	30%	M
FR4-Пошук людей та побажань	FR2	15%	M
FR5-Вибір подарунку	FR2	15%	C
FR5.1-Вибір з каталогу	FR5		

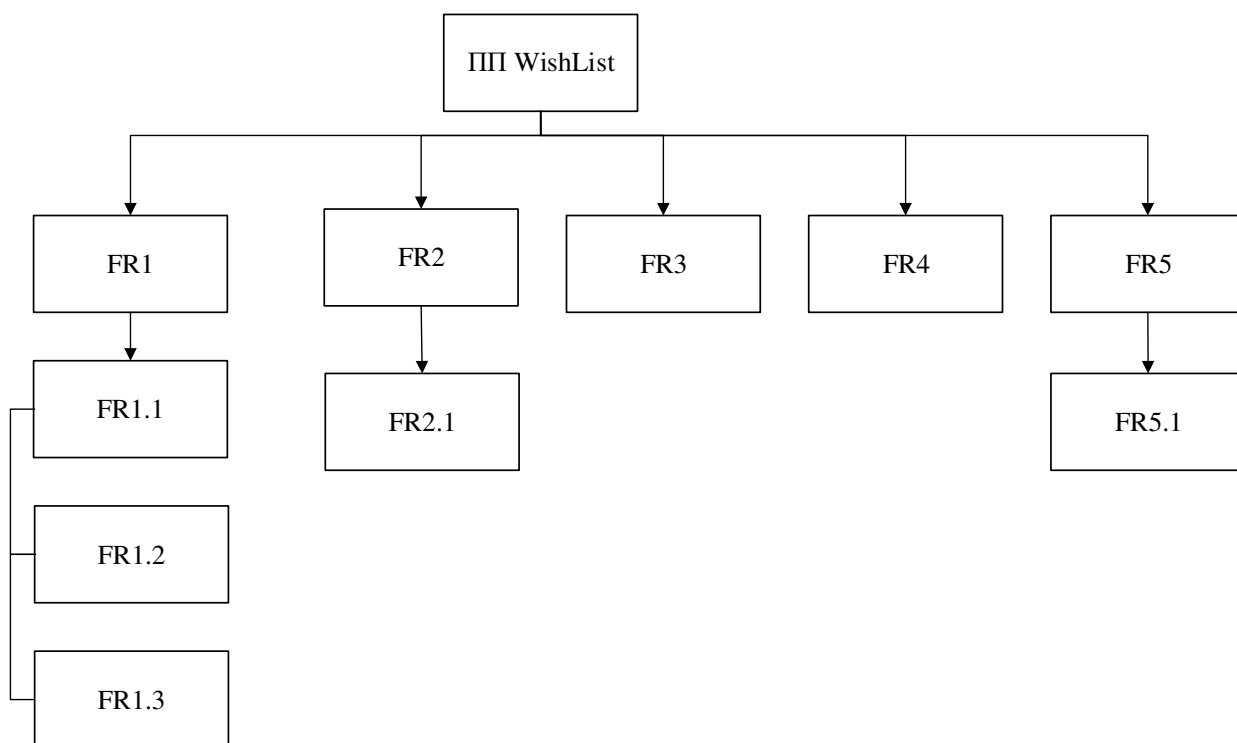


Рис. 1.5 – Ієрархічна WBS-структура багаторівневої класифікації функціональних вимог

1.4.2 Функціональний аналіз існуючих програмних продуктів

Таблиця 1.7 – Функціональний аналіз існуючих програмних продуктів

A	B	C	D
Ідентифікатор функції (назва)	ПП1	ПП2	ПП розроблюваний
FR1 (Реєстрація користувача)	+	-	+
FR2 (Авторизація)	-	+	+
FR3 (Внесення подарунків)	+	-	+
FR4 (Пошук людей та побажань)	-	+	+
FR5 (Вибір подарунку)	+	+	+
Безкоштовне розповсюдження	+	-	+

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейсів користувача

1.5.1.1.1 Опис INPUT-інтерфейсів користувача

Таблиця 1.8 – Опис INPUT-інтерфейсів користувача

A	B	C
Ідентифікатор функції (назва)	Засіб INPUT-потoku	Особливості використання
FR1 (Реєстрація користувача)	стандартна комп'ютерна клавіатура; 2/3-кнопочний маніпулятор типу "миша"	
FR2 (Авторизація)	стандартна комп'ютерна клавіатура; 2/3-кнопочний маніпулятор типу "миша"	
FR3 (Внесення подарунків)	2/3-кнопочний маніпулятор типу "миша"	
FR4 (Пошук людей та побажань)	2/3-кнопочний маніпулятор типу "миша"	Використання колеса миші для прокрутки та вибору
FR5 (Вибір подарунку)	2/3-кнопочний маніпулятор типу "миша"	Використання колеса миші для прокрутки та вибору

1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача

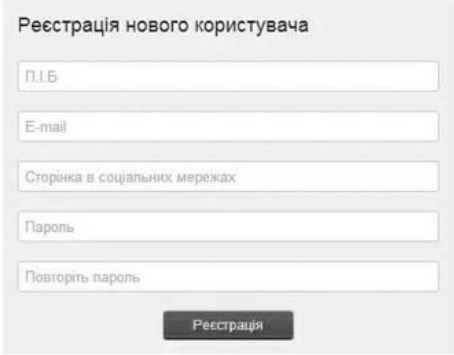
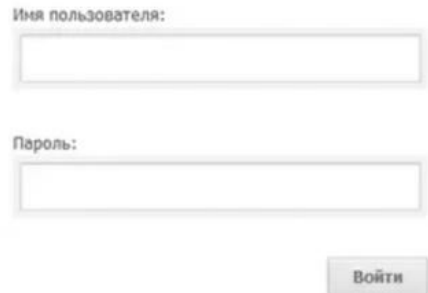
Ідентифікатор функції (назва)	Засіб OUTPUT - потоку	Особливості використання
FR1 (Реєстрація користувача)	Графічний інтерфейс	
FR2 (Авторизація)	Графічний інтерфейс	

Рис. 1.6 – Опис OUTPUT-інтерфейсів користувача

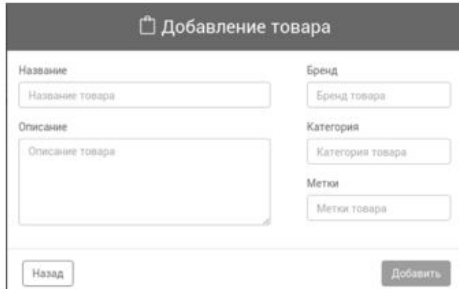

FR3 (Внесення подарунків)	Графічний інтерфейс	
FR4 (Вибір подарунку)	Графічний інтерфейс	

Рис. 1.7 – Опис OUTPUT-інтерфейсів користувача

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

Так, як програма типу RWA, рекомендується швидке підключення до інтернету (дротове або без дротове), та пристрій на якому встановлено браузер.

1.5.1.3 Опис програмних інтерфейсів

Програма може працювати всюди де є браузер з останньою версією оновлення.

1.5.1.4 Опис інтерфейсів передачі інформації

1.5.1.5 Опис атрибутів продуктивності

При використанні звичайних функцій, затримка не має бути більше ніж пару секунд.

Єдині функції ПП, що можуть викликати більшу затримку:

Таблиця 1.9 – Максимальний час реакції ПП на дії користувача

A	B
Ідентифікатор функції (назва)	Максимальний час реакції ПП на дії користувачів, секунди
FR1 (Реєстрація користувача)	5
FR2 (Авторизація)	5
FR3 (Внесення подарунків)	5
FR4 (Пошук людей та побажань)	5
FR5 (Вибір подарунку)	10

Одночасно, буде обслуговуватися не більше 10000 клієнтів, якщо більше, то буде затримка на відклик та повідомлення на сайті.

2 Планування процесу розробки програмного продукту

2.1 Планування ітерацій розробки програмного продукту

З метою забезпечення вимог таких рекомендацій IEEE-стандарту, як необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, визначено функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП. Результати представлено в таблиці 2.1

Таблиця 2.1 – приклад опису функцій з наданням унікальних ієрархічних ідентифікаторів

Ідентифікатор функції	Функціональні залежності	Вплив на досягнення мети	Пріоритет функції
FR1-Реєстрація користувача	-	20%	M
FR1.1-Ввод параметрів реєстрації користувача	-		
FR1.2-зберігання параметрів реєстрації	-		
FR1.3-перевірка параметрів реєстрації	F1.2		
FR2-Авторизація	-	20%	M
FR2.1-перевірка параметрів авторизації	F1.2		
FR3-Внесення подарунків	F2	30%	M
FR4-Пошук людей та побажань	F2	15%	M
FR5-Вибір подарунку	F2	15%	C
FR5.1-Вибір з каталогу	F5		

2.2 Концептуальний опис архітектури програмного продукту

Rich WEB Application (RWA) - спеціальний тип WEB ПП, який ґрунтується на HTML5+JavaScript, що виконується, як правило на стороні клієнта і

використовує його процесорний час і ресурси та досить часто складається з двох «половинок» - серверне та клієнтське ПЗ.

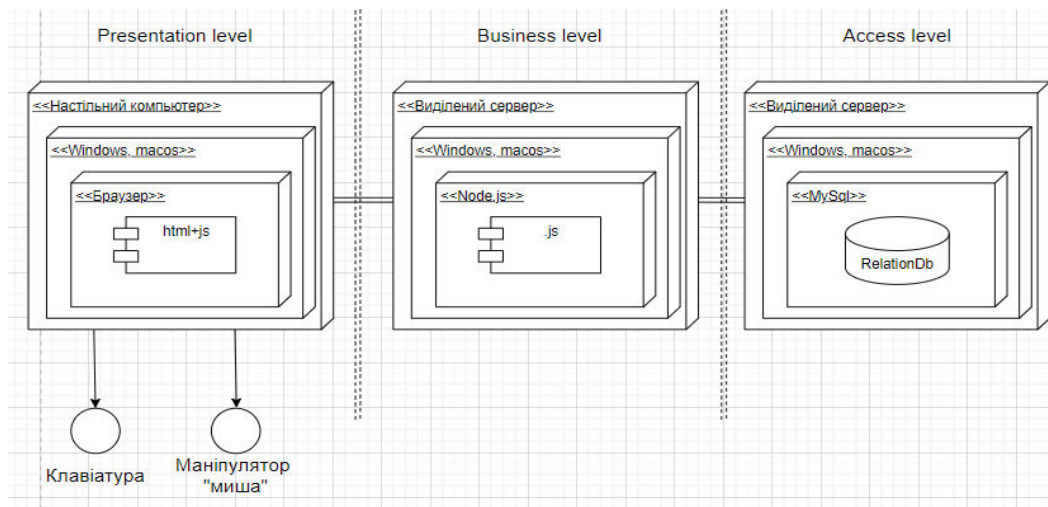


Рис. 2.1 – Концептуальний опис архітектури програмного продукту

2.3 План розробки програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

Таблиця 2.2 – Оцінка трудомісткості розробки ПП

Прецедент	Кількість кроків сценарію	Ваговий коефіцієнт
Користувач	9	3
Реєстрація	4	10
Авторизація	2	5
Перегляд виконаних пунктів подарунків	1	5
Видалення подарунку	3	5
Пошук подарунків	4	10
Запит каталогу	1	5
Пошук по категоріям	1	5
Додавання подарунків	3	5
Вибір подарунку	5	10
Модератор	5	3

Введення даних	3	5
Видалення подарунку	1	5
Додавання подарунків	1	5
Зміна даних про подарунок	1	5

$$UC = 10*5 + 3*10 + 2*3 = 86$$

2.3.2 Визначення дерева робіт з розробки програмного продукту

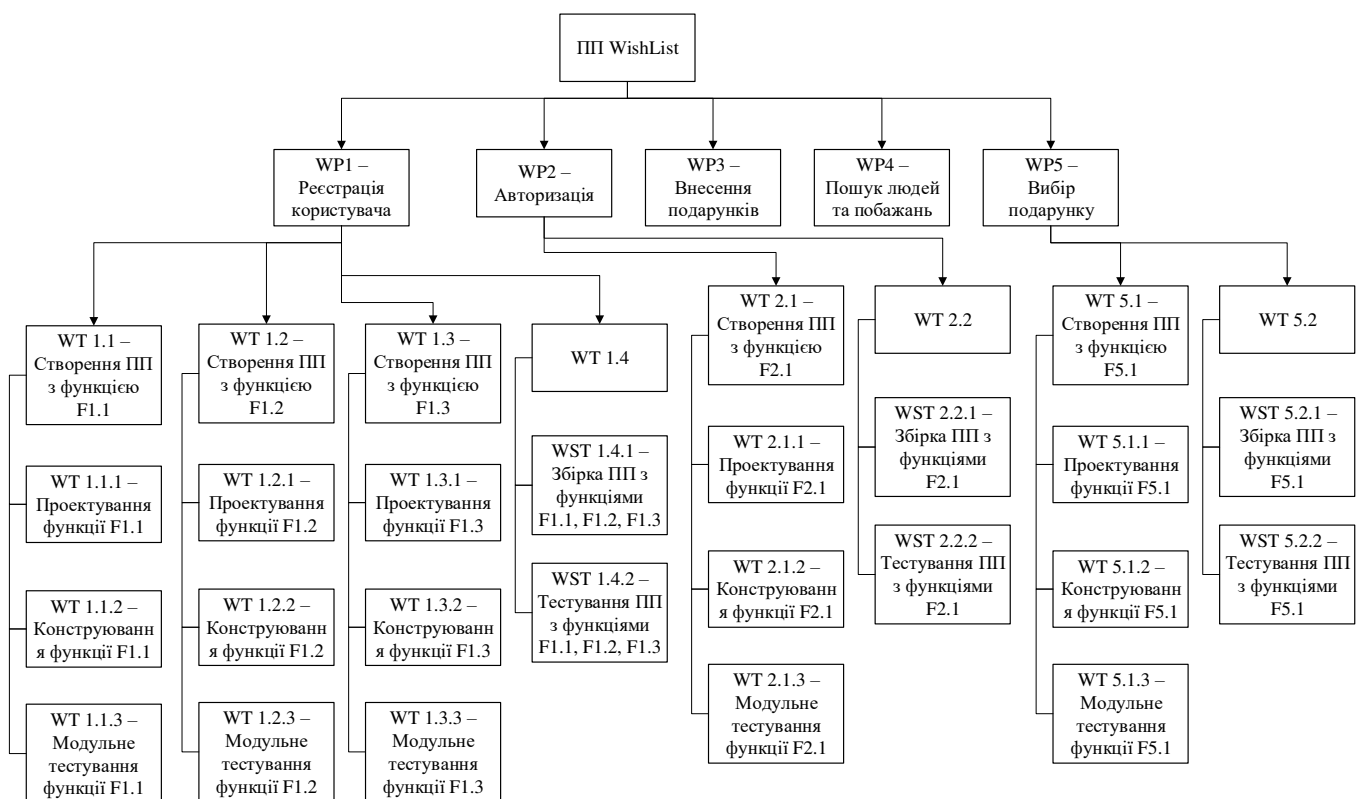


Рис. 2.2 – Дерево робіт з розробки ПП

2.3.3 Графік робіт з розробки програмного продукту

2.3.3.1 Таблиця з графіком робіт

Таблиця 2.3 – таблиця з графіком робіт

WST	Дата початку	Дні	Дата завершення	Виконавець
1.1.1	01.10.2021	2	02.10.2021	Бицкало І. Дем'яненко Р.
1.2.2	02.10.2021	2	04.10.2021	Бицкало І. Дем'яненко Р.
1.3.3	04.10.2021	3	07.10.2021	Бицкало І. Дем'яненко Р.
1.2.1	07.10.2021	1	08.10.2021	Бицкало І. Дем'яненко Р.
1.2.2	08.10.2021	3	11.10.2021	Бицкало І. Дем'яненко Р.
1.2.3	11.10.2021	3	14.10.2021	Бицкало І. Дем'яненко Р.
1.3.1	14.10.2021	4	18.10.2021	Бицкало І. Дем'яненко Р.
1.3.2	18.10.2021	2	20.10.2021	Бицкало І. Дем'яненко Р.
1.3.3	20.10.2021	3	23.10.2021	Бицкало І. Дем'яненко Р.
1.4.1	23.10.2021	2	25.10.2021	Бицкало І. Дем'яненко Р.
1.4.2	25.10.2021	3	28.10.2021	Бицкало І. Дем'яненко Р..
2.1.1	28.10.2021	4	01.11.2021	Бицкало І. Дем'яненко Р.
2.1.2	01.11.2021	5	06.11.2021	Бицкало І. Дем'яненко Р.
2.1.3	06.11.2021	5	11.11.2021	Бицкало І. Дем'яненко Р.
2.2.1	11.11.2021	5	16.11.2021	Бицкало І. Дем'яненко Р.
2.2.2	16.11.2021	5	21.11.2021	Бицкало І. Дем'яненко Р.
5.1.1	21.11.2021	5	26.11.2021	Бицкало І. Дем'яненко Р.
5.1.2	26.11.2021	5	01.12.2021	Бицкало І. Дем'яненко Р.
5.1.3	01.12.2021	4	05.12.2021	Бицкало І. Дем'яненко Р.

5.2.1	05.12.2021	1	06.12.2021	Бицкало І. Дем'яненко Р.
5.2.2	06.12.2021	3	09.12.2021	Бицкало І. Дем'яненко Р.

2.3.3.2 Діаграма Ганта

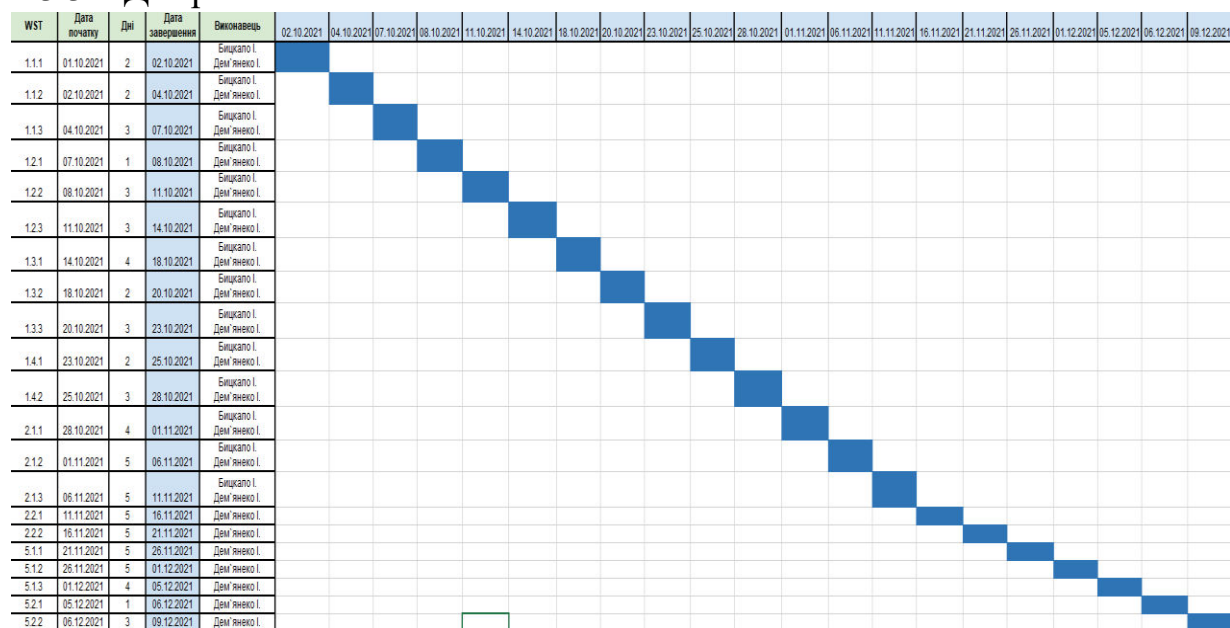


Рис 2.3 – Діаграма Ганта

3 Проектування програмного продукту

3.1 Концептуальне та логічне проектування структур даних програмного продукту

3.1.1 Концептуальне проектування на основі UML-діаграми концептуальних класів

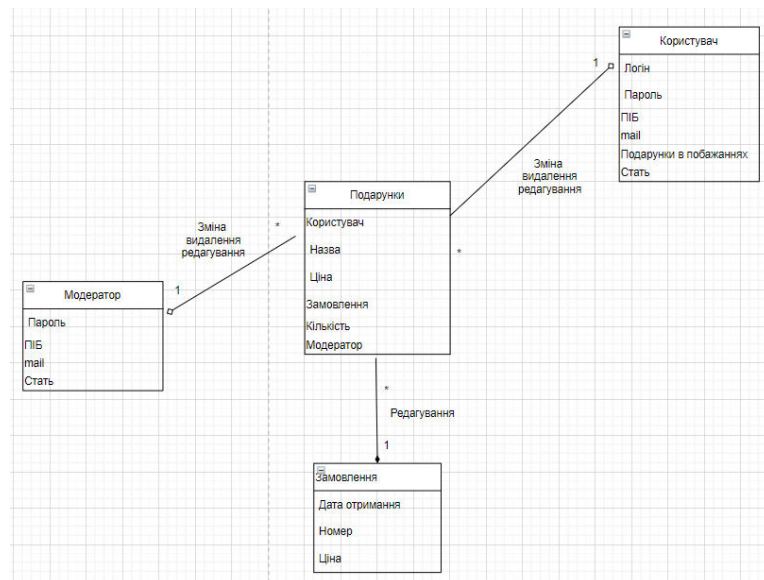


Рис. 3.1.1.1 – Концептуальне проектування

3.1.2 Логічне проектування структур даних

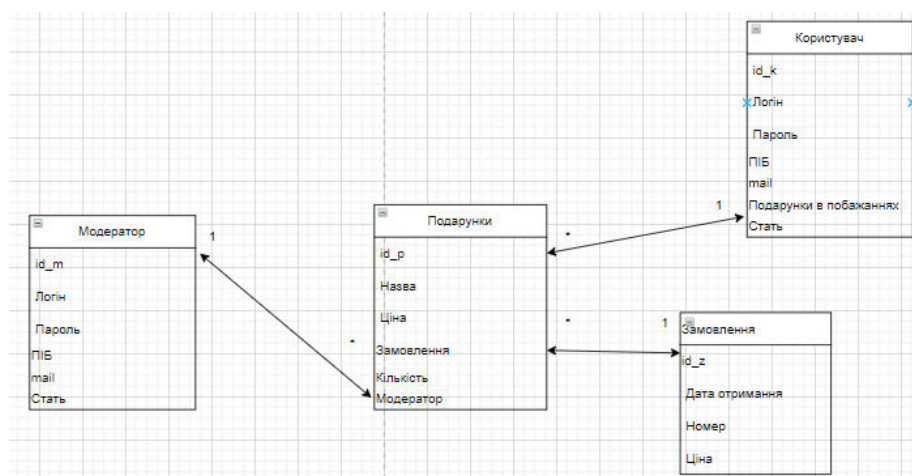


Рис. 3.1.2.1 – Логічне проектування структур даних

3.2 Проектування програмних класів

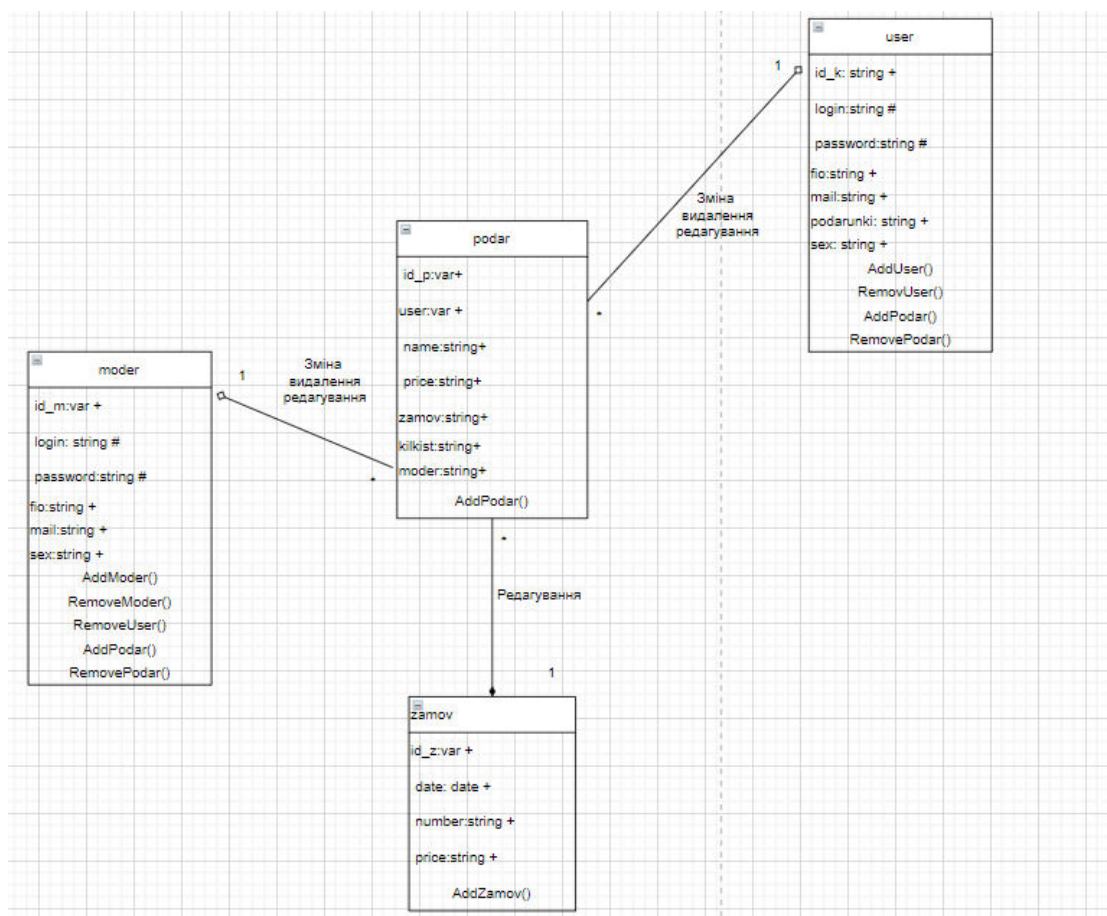


Рис. 3.2 – Проектування програмних класів

3.3 Проектування алгоритмів роботи методів програмних класів

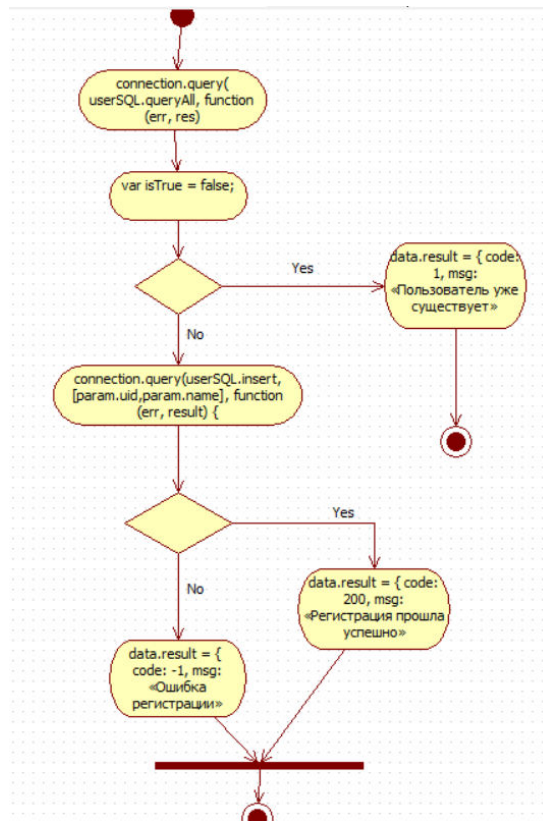


Рис. 3.3.1 – UML-діаграма активності «reest_user(), save_user(), autoriz_user()»

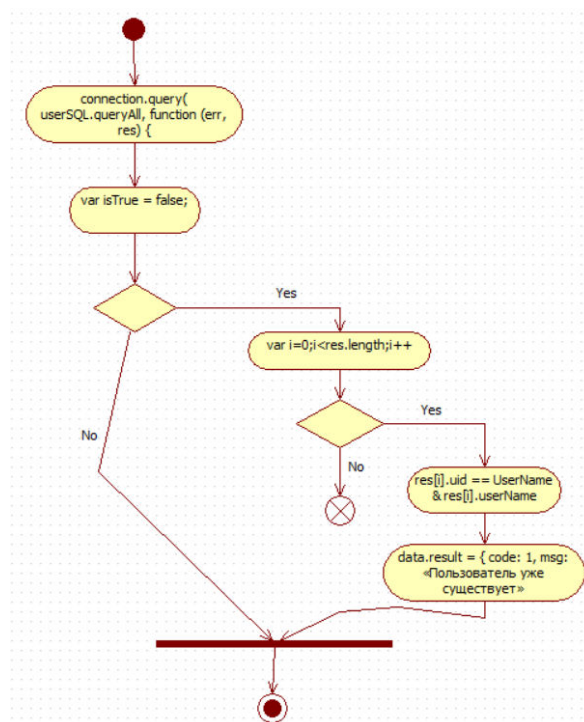


Рис. 3.3.2 – UML-діаграма активності «perevir_reestr_user()»

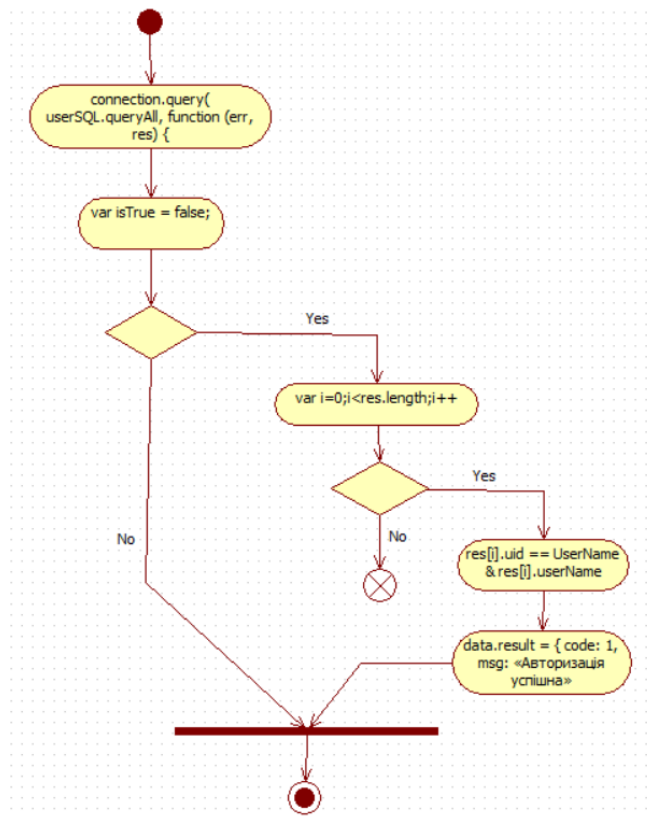


Рис. 3.3.3 – UML-діаграма активності «perevir_authorized_user()»

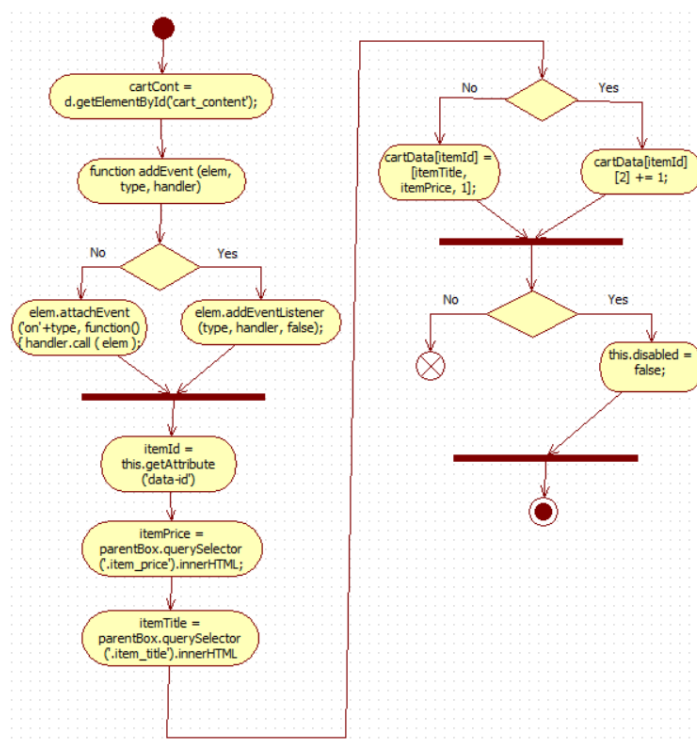


Рис. 3.3.4 – UML-діаграма активності «vnes_podar(), vibir_podar()»

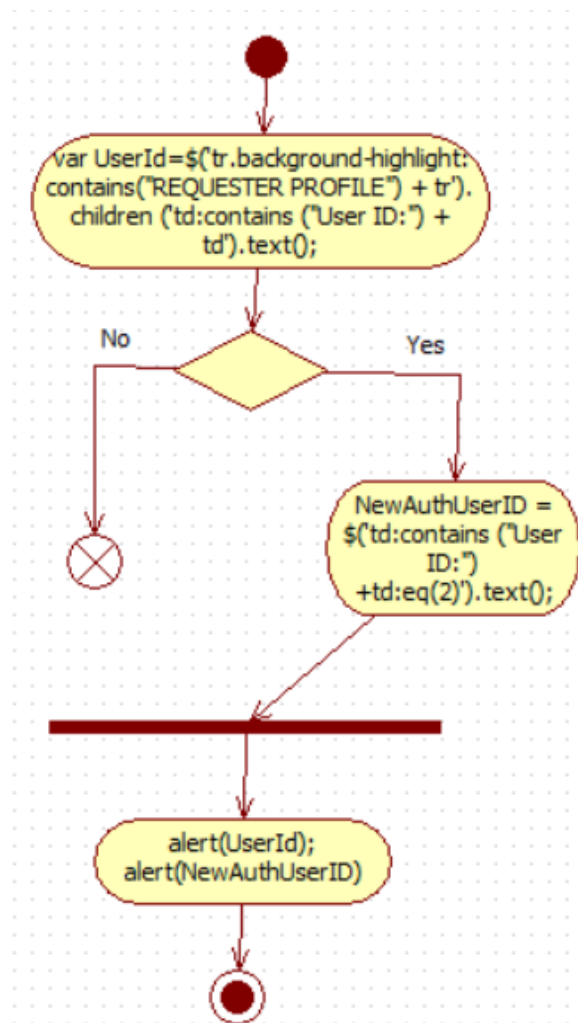


Рис. 3.3.5 – UML-діаграма активності «search_people()»

3.4 Проектування тестових сценаріїв верифікації роботи програмних модулів

3.4.1 Проектування тестових сценаріїв верифікації функціональних вимог

Таблиця 3.4.1.1 – Верифікація функціональних вимог

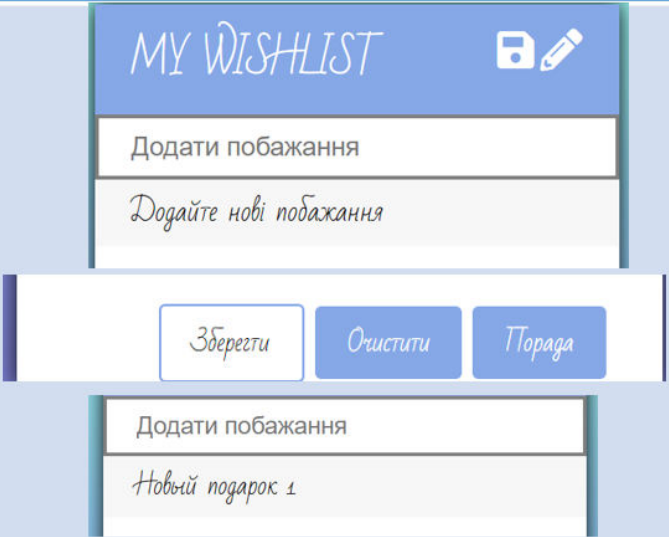
FR ID	Test Case ID	Опис значень вхідних даних до методу (процедури, функції)	Опис очікуваних значень результату виконання методу
FR1	TC1	Input = «ф,к»	Повідомлення: «Параметр «Назва» має містити не менше 3-х букв»
FR1	TC2	Input = «1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9,1,2,3,4,5,4»	Повідомлення: «Параметр «Назва» має містити не більше 23-х символів»
FR1	TC3	Input = «Null»	Повідомлення: «Заповніть Ім'я подарунку»

Продовження таблиці 3.4.1.1

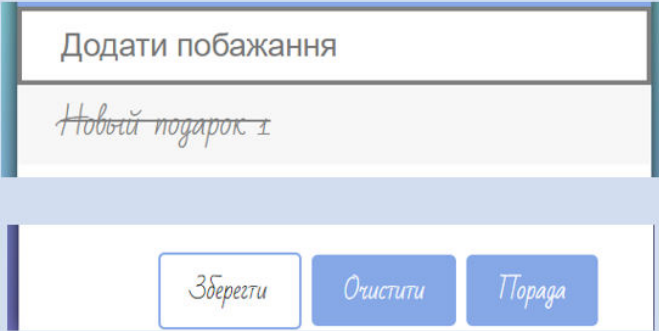
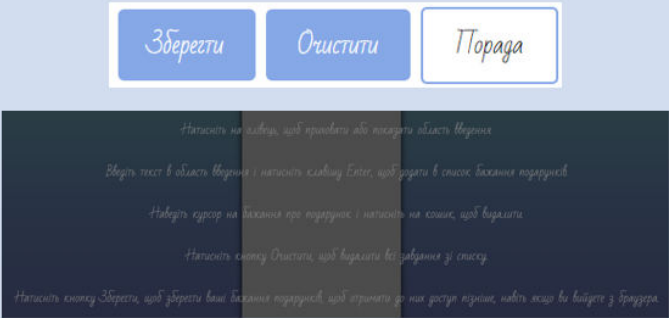
FR ID	Test Case ID	Опис значень вхідних даних до методу (процедури, функції)	Опис очікуваних значень результату виконання методу
FR1	TC4	Input = (empty)	Повідомлення: «Ім'я та опис не може бути пустим»
FR1	TC5	Input = (текст, що перевищує 100 символів)	Повідомлення: «Не можна перевищувати допустиму величину тексту в 100 знаків»
FR1	TC6	Input = (Нова інформація)	Повідомлення «Дані оновлено»
FR1	TC7	Input = (Некоректне введення)	Повідомлення: «Помилка з введенням, будь ласка, перезавантажте сторінку та повторіть спробу»
FR1	TC8	Input = (Введення коректного тексту)	Повідомлення: «Успішне додавання подарунку»

3.4.2 Проектування тестових сценаріїв верифікації нефункціональних ВИМОГ

Таблиця 5.2.2 – Опис тестових сценаріїв для інтерфейсу користувача

NFR ID	Test Case ID	Опис дій з ПП на рівні інтерфейсу користувача	Опис очікуваних результатів взаємодії з ПП на рівні інтерфейсу користувача
NFR1	TC9	1) В поле «Додати побажання» ввести будь-яке значення 2) Натиснути Enter 3) Натиснути кнопку «Зберегти»	

Продовження таблиці 5.2.2

NFR ID	Test Case ID	Опис дій з ПП на рівні інтерфейсу користувача	Опис очікуваних результатів взаємодії з ПП на рівні інтерфейсу користувача
NFR2	TC10	1) Натиснути на «Новый подарок 1», для того, щоб відмітити виконаний подарунок 2) Натиснути кнопку «зберегти»	
NFR3	TC11	1) Натиснути кнопку «Порада» 2) Ознайомитися з інструкцією	

3.4.3 Створення матриці відповідності вимог до програмного продукту

FR ID	NFR ID	TC ID
FR1	NFR 1	TC 1
FR1	NFR 1	TC 2
FR1	NFR 1	TC 3
FR1	NFR 1	TC 4
FR1	NFR 1	TC 5
FR1	NFR 1	TC 6
FR1	NFR 1	TC 7
FR1	NFR 1	TC 8
FR5	NFR 1	TC 9
FR2	NFR 2	TC 10
FR3	NFR 3	TC 11

Рис. 3.4.3.1 – Матриця відповідності вимог

4 Конструювання програмного продукту

4.1 Особливості конструювання структур даних

Нереляційна база даних

```
<?xml version="1.0" encoding="UTF-8"?>

<project version="4">
  <component name="ChangeListManager">
    <list      default="true"      id="45ea9f55-bc7c-4f9a-8c04-e7a724a47557"
name="Default Changelist" comment="" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option      name="HIGHLIGHT_NON_ACTIVE_CHANGELIST"
value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="ProjectId" id="21kes47i15l7v39DJgtgywuJIvv" />
  <component name="ProjectViewState">
    <option name="hideEmptyMiddlePackages" value="true" />
    <option name="showLibraryContents" value="true" />
  </component>
  <component name="PropertiesComponent">
    <property name="WebServerToolWindowFactoryState" value="false" />
    <property name="dart.analysis.tool.window.visible" value="false" />
    <property name="last_opened_file_path" value="$PROJECT_DIR$/.." />
  </component>
  <component name="SvnConfiguration">
    <configuration />
  </component>
```

```

<component name="TaskManager">

  <task active="true" id="Default" summary="Default task">

    <changelist id="45ea9f55-bc7c-4f9a-8c04-e7a724a47557" name="Default
Changelist" comment="" />

    <created>1638489715748</created>

    <option name="number" value="Default" />

    <option name="presentableId" value="Default" />

    <updated>1638489715748</updated>

    <workItem from="1638489740438" duration="91000" />

  </task>

</servers />

</component>

<component name="TypeScriptGeneratedFilesManager">

  <option name="version" value="2" />

</component>

<component name="WindowStateProjectService">
  <state      x="549"      y="167"      key="FileChooserDialogImpl"
timestamp="1638489838457">
    <screen x="0" y="0" width="1536" height="824" />
  </state>
  <state      x="549"      y="167"
key="FileChooserDialogImpl/0.0.1536.824@0.0.1536.824"
timestamp="1638489838457" />

```

```

    <state      x="431"      y="145"      key="search.everywhere.popup"
timestamp="1638489814271">
    <screen x="0" y="0" width="1536" height="824" />
  </state>

  <state      x="431"      y="145"
key="search.everywhere.popup/0.0.1536.824@0.0.1536.824"
timestamp="1638489814271" />

</component>
</project>

```

```

var input = document.querySelector( selectors: "input[type = 'text']");
var ul = document.querySelector( selectors: "ul");
var container = document.querySelector( selectors: "div");
var lists = document.querySelectorAll( selectors: "li");
var spans = document.getElementsByTagName( qualifiedName: "span");
var pencil = document.querySelector( selectors: "#pencil");
var saveBtn = document.querySelector( selectors: ".save");
var clearBtn = document.querySelector( selectors: ".clear");
var tipsBtn = document.querySelector( selectors: ".tipBtn");
var closeBtn = document.querySelector( selectors: ".closeBtn");
var overlay = document.getElementById( elementId: "overlay")

```

Рис. 4.1.1 – Виконання CRUD-операцій доступу до даних

4.2 Особливості конструювання програмних модулів

```

function deleteTodo(){
  for(let span of spans){
    span.addEventListener ("click",function (){
      span.parentElement.remove();
      event.stopPropagation();
    });
  }
}

//function to load todo if list is found in local storage.
function loadTodo(){
  if(localStorage.getItem( key: 'todoList')){
    ul.innerHTML = localStorage.getItem( key: 'todoList');
    deleteTodo();
  }
}

//event listener for input to add new todo to the list.
input.addEventListener( type: "keypress", listener: function(keyPressed :Event ){
  if(keyPressed.which === 13){
    //creating lists and span when enter is clicked
    var li = document.createElement( tagName: "li");
    var spanElement = document.createElement( tagName: "span");
    var icon = document.createElement( tagName: "i");

    var newTodo = this.value;
    this.value = " ";

    icon.classList.add('fas', 'fa-trash-alt');
    spanElement.append(icon);
    ul.appendChild(li).append(spanElement,newTodo);

    deleteTodo();
  }
});

```

Рис. 4.2.1 – Конструювання програмних модулів NodeDao

4.2.1 Конструювання програмної структури з урахуванням спеціалізованого *Framework* для *FrontEnd*-компонент архітектури (за наявністю)

Canvas – елемент HTML5, призначений для створення растрового двовимірного зображення за допомогою скриптів, зазвичай мовою JavaScript. Початок відліку блоку знаходиться зліва зверху. Від нього і будується кожен елемент блоку. Розмір простору координат не обов'язково відображає розмір фактичної відображуваної площі. За замовчуванням його ширина дорівнює 300 пікселям, а висота 150.

```
/*  
 * html2canvas 1.3.3 <https://html2canvas.hertzen.com>  
 * Copyright (c) 2021 Niklas von Herten <https://hertzen.com>  
 * Released under MIT License  
 */  
!function(A,e){"object"==typeof exports&&"undefined"!=typeof module?module.exports=e():"function"==typeof define&&define  
.amd?define(e):(A="undefined"!=typeof globalThis?globalThis:A||self).html2canvas=e()}(this,function(){  
/*  
 * Copyright (c) Microsoft Corporation.  
 *  
 * Permission to use, copy, modify, and/or distribute this software for any  
 * purpose with or without fee is hereby granted.  
 *  
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH  
 * REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY  
 * AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,  
 * INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM  
 * LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR  
 * OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR  
 * PERFORMANCE OF THIS SOFTWARE.  
 */  
*****  
 *Var B=function(A,e){return(B=Object  
.setPrototypeOf({__proto__:[]})instanceof Array&&function(A,e){A.__proto__=e}||function(A,e){for(var t in e)Object.prototype  
.hasOwnProperty.call(e,t)&&(A[t]=e[t]))(A,e)};function A(A,e){if("function"!=typeof e&&null!==e)throw new TypeError("Class  
extends value "+String(e)+" is not a constructor or null");function t(){this.constructor=A}B(A,e).prototype=null===e?Object  
.create(e):t.prototype=e.prototype,new t)}var h=function(){return(h=Object.assign||function(A){for(var e,t=1,B=arguments  
.length;t<B;t++)for(var r in e=arguments[t])Object.prototype.hasOwnProperty.call(e,r)&&(A[r]=e[r]);return A}).apply(this,  
arguments)};function a(A,s,o,i){return new(o=Promise)(function(t,e){function B(A){try{n(i.next(A))}catch(A){e(A)}}function  
r(A){try{n(i.throw(A))}catch(A){e(A)}}function n(A){var e:A.done?(A.value):(e=A.value)instanceof o?new o(function(A){A(e)})  
.then(B,r)):n((i=i.apply(A,s||[])).next())}}function H(t,B){var r,n,s,o={label:s,sent:function(){if(!s[0])throw s[1];return  
s[1]},trys:[],ops:[],A:{next:e(B),throw:e(1),return:e(2)}};return"function"==typeof Symbol&&A[Symbol.iterator]=function()  
{return this}}A:function e(){return function(A){return function(e){if(e)throw new TypeError("generator is already executing.")
```

Рис. 4.2.1.1 – Конструювання програмної структури з урахуванням спеціалізованого Framework «Html2canvas»

Використовується, як правило, для відтворення графіків для статей і ігрового поля в деяких браузерних іграх. Але також може використовуватися для вбудовування відео в сторінку і створення повноцінного плеєра.

4.2.2 Конструювання програмних класів (за наявністю об'єктно-орієнтованого програмування)

Немає

4.2.3 Конструювання алгоритмів методів програмних класів або процедур/функцій

```
function deleteTodo(){
  for(let span of spans){
    span.addEventListener ("click",function (){
      span.parentElement.remove();
      event.stopPropagation();
    });
  }
}

//function to load todo if list is found in local storage.
function loadTodo(){
  if(localStorage.getItem( key: 'todoList')){
    ul.innerHTML = localStorage.getItem( key: 'todoList');
    deleteTodo();
  }
}
```

Рис. 4.2.3.1 – Конструювання алгоритмів методів програмних класів

```
//event listener for input to add new todo to the list.
input.addEventListener( type: "keypress", listener: function(keyPressed :Event ){
  if(keyPressed.which === 13){
    //creating lists and span when enter is clicked
    var li = document.createElement( tagName: "li");
    var spanElement = document.createElement( tagName: "span");
    var icon = document.createElement( tagName: "i");

    var newTodo = this.value;
    this.value = " ";

    icon.classList.add('fas', 'fa-trash-alt');
    spanElement.append(icon);
    ul.appendChild(li).append(spanElement,newTodo);

    deleteTodo();

  }
});
```

Рис. 4.2.3.2 – Конструювання алгоритмів методів програмних класів

4.2.4 Особливості використання спеціалізованих програмних бібліотек та API (за наявності)

Немає

4.3 Модульне тестування програмних модулів

Таблиця 4.3.1 – Модульне тестування програмних модулів

FR ID	Test Case ID	Опис значень вхідних даних до методу (процедури, функції)	Опис очікуваних значень результату виконання методу
FR1	TC1	Input = «ф,к»	Повідомлення: «Параметр «Назва» має містити не менше 3-х букв»
FR1	TC2	Input = «1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9,1,2,3,4,5,4»	Повідомлення: «Параметр «Назва» має містити не більше 23-х символів»
FR1	TC3	Input = «Null»	Повідомлення: «Заповніть Ім'я подарунку»
FR1	TC4	Input = (empty)	Повідомлення: «Ім'я та опис не може бути пустим»
FR1	TC5	Input = (текст, що перевищує 100 символів)	Повідомлення: «Не можна перевищувати допустиму величину тексту в 100 знаків»
FR1	TC6	Input = (Нова інформація)	Повідомлення «Дані оновлено»
FR1	TC7	Input = (Некоректне введення)	Повідомлення: «Помилка з введенням, будь ласка, перезавантажте сторінку та повторіть спробу»
FR1	TC8	Input = (Введення коректного тексту)	Повідомлення: «Успішне додавання подарунку»

FR1	TC1	Input = «ф,к»	Повідомлення: «Параметр «Назва» має містити не менше 3-х букв»
-----	-----	---------------	--

```

pencil.addEventListener( type: 'click', listener: function(){
    input.classList.toggle( token: 'display');
    var input = "ф,к"
});

```

Test Results	976 ms
SearchServiceImplTest	976 ms
getResourceByTitleTest()	976 ms

Рис. 4.3.1 – Test case №1

Рис. 4.3.5 – Test case №5

FR1	TC6	Input = (Нова інформація)	Повідомлення: «Дані оновлено»
-----	-----	---------------------------	-------------------------------

```

ul.addEventListener( type: 'click', listener: function(ev :MouseEvent ) {
  if (ev.target.tagName === 'LI') {
    ev.target.classList.toggle( token: 'checked');
  }
  var input_name = "new information";
}, options: false
);

```

- Test Results 903 ms
- SearchServiceImplTest 903 ms
- getResourceByTitleTest6() 903 ms

Рис. 4.3.6 – Test case №6

FR1	TC7	Input = (Некоректне введення)	Повідомлення: «Помилка з введення, будь ласка, перезавантажте сторінку та повторіть спробу»
-----	-----	-------------------------------	---

```

function loadTodo(){
  if(localStorage.getItem( key: 'todoList')){
    ul.innerHTML = localStorage.getItem( key: 'todoList');
    deleteTodo();
    var input = "awdadwd";
  }
}

```

- Test Results 630 ms
- SearchServiceImplTest 630 ms
- getNodesTest() 630 ms

Рис. 4.3.7 – Test case №7

FR1	TC8	Input = (Введення коректного тексту)	Повідомлення: «Успішне додавання подарунку»
-----	-----	--------------------------------------	---

```

function loadTodo(){
  if(localStorage.getItem( key: 'todoList')){
    ul.innerHTML = localStorage.getItem( key: 'todoList');
    deleteTodo();
    var input = input;
  }
}

```

- Test Results 711 ms
- SearchServiceImplTest 711 ms
- getNodesTest2() 711 ms

Рис. 4.3.8 – Test case №8

5 Верифікація програмного продукту

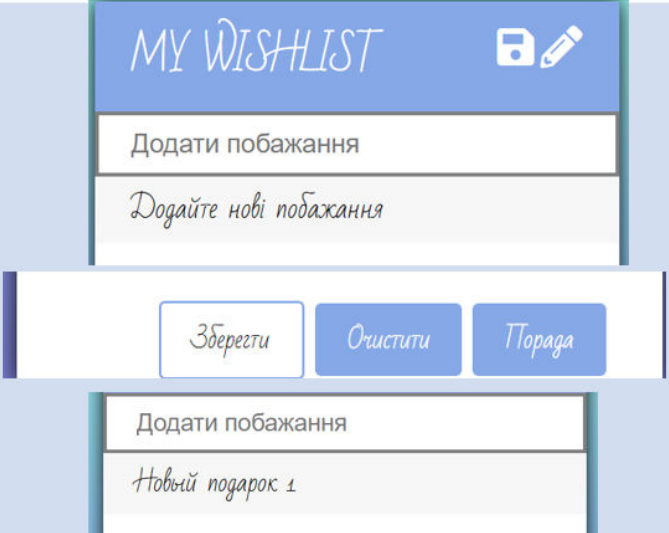
5.1 Тестування апаратно-програмних інтерфейсів програмного продукту

Таблиця 5.1.1 – Опис умов перевірки апаратно-програмних інтерфейсів ПП.

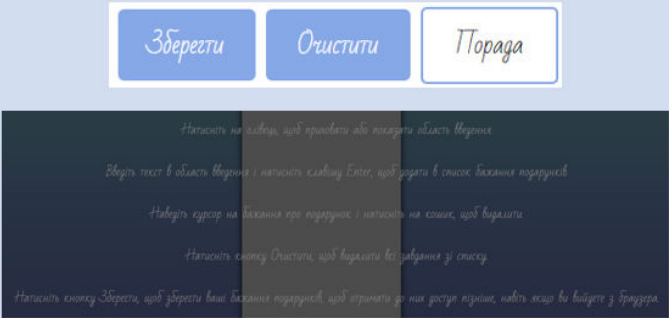
Тип умови (<i>Hard/Soft</i>)	Опис вимог	Опис реальних вимог
Hard	CPU: frequency $\geq 1\text{Hz}$ Memory: size $\geq 200\text{Mb}$	CPU: Intel Pentium G4400 Memory: 1Gb
Soft	OS: Кросплатформенна	OS: Кросплатформенна
Soft	Web browser: Google Chrome, Microsoft Edge, Mozilla Firefox, Opera	Web browser: Будь-який останньої версії

5.2 Тестування інтерфейсу користувача програмного продукту

Таблиця 5.2.2 – Опис тестових сценаріїв для інтерфейсу користувача

NFR ID	Test Case ID	Опис дій з ПП на рівні інтерфейсу користувача	Опис очікуваних результатів взаємодії з ПП на рівні інтерфейсу користувача
NFR1	TC9	1) В поле «Додати побажання» ввести будь-яке значення 2) Натиснути Enter 3) Натиснути кнопку «Зберегти»	

Продовження таблиці 5.2.2

NFR ID	Test Case ID	Опис дій з ПП на рівні інтерфейсу користувача	Опис очікуваних результатів взаємодії з ПП на рівні інтерфейсу користувача
NFR2	TC10	<ol style="list-style-type: none"> 1) Натиснути на «Новый подарок 1», для того, щоб відмітити виконаний подарунок 2) Натиснути кнопку «зберегти» 	
NFR3	TC11	<ol style="list-style-type: none"> 1) Натиснути кнопку «Порада» 2) Ознайомитися з інструкцією 	

5.3 Тестування часу реакції програмного продукту на дії користувача

Після проведення двох експериментів з часом виконання функції отримано два результати: 0,9 сек, 0.7 сек. Середнє значення = $(0.9+0.7)/2 = 0.8$

$$\delta = ((0.9-0.7) / 0.8) * 100\% = 25\%$$

Такий результат, вказує на необхідність продовження експериментів, поки δ не стане менше порогового значення $< 15\%$

Таблиця 5.3.3 – Результат тестування часу реакції ПП на дії користувача

FR (назва)	Максимальний час реакції ПП на дії користувачів, секунди	Кількість проведених тестів	Результат тестування, секунди	Відносна похибка вимірювань, %
FR1	2с	5	20с	7%
FR2	1,5с	4	30с	5%
FR3	1,2с	6	45с	8%
FR4	1с	8	24с	9%
FR5	0,7с	9	18с	4%

6 Розгортання та валідація програмного продукту

6.1 Інструкція з встановлення системного програмного забезпечення

Отримати оригінал (завантажитися в папку eternity):

```
git clone https://github.com/jolynutella/SW_Bytskalo_Demyanenko_LAI194
```

Отримати оригінал в папку custom:

```
git clone https://github.com/jolynutella/SW\_Bytskalo\_Demyanenko\_LAI194 custom
```

Перевірити оновлення (при умові, що скачано в папку eternity):

```
cd eternity
```

```
git remote -v update
```

Отримати оновлення:

```
git pull
```

Розроблений ПП може працювати на будь-якому комп'ютері де є інтернет та веб-браузер.

Послідовність кроків для забезпечення роботи ПП:

- 1 Для клієнта потрібне встановлення Html, JS, веб-браузер;
- 2 Для встановлення ПП потрібно архів с файлами розархівувати в встановлену папку eternity;
- 3 Запустити за допомогою файлу index.html програму.

6.2 Інструкція з використання програмного продукту

ПП надає можливість користувачу ролі «гість» вводити назву подарунку та його характеристики, як показано на рисунку 1.

The screenshot shows a web form titled "MY WISHLIST" in a blue header bar. To the right of the title are icons for a floppy disk and a pencil. Below the header is a white input field containing the text "Подарунок (Його властивості)".

Рис. 6.2.1 – Екранна форма введення подарунку

При цьому необхідно пам'ятати про обмеження тексту: не менше 1 та не більше 100 букв або символів.

Для опублікування подарунку та внесення в список, потрібно натиснути клавішу Enter, як показано на рисунку 2.

This screenshot shows the same "MY WISHLIST" form. The input field is now empty, and the text "Подарунок (Його властивості)" has been moved to a separate row in the list below, indicating it has been successfully added.

Рис. 6.2.2 – Екранна форма опублікування подарунку

ПП надає можливість користувачу ролі «гість» зберігати список подарунків натиснувши клавішу зберегти, як показано на рисунку 3.

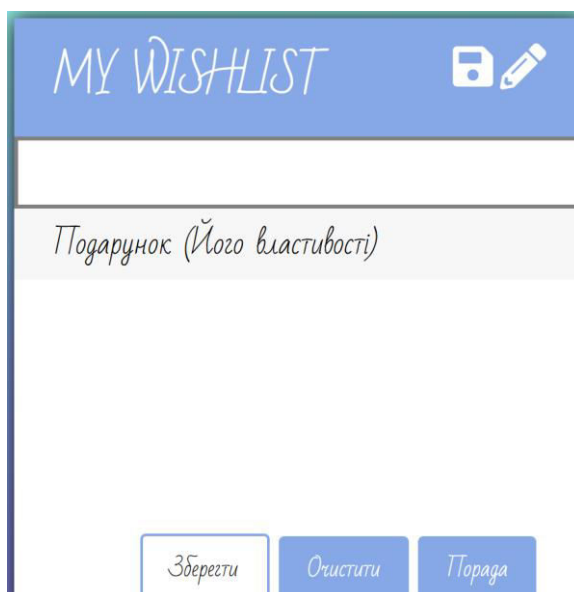
The screenshot shows a web interface titled "MY WISHLIST" in a blue header bar. To the right of the title is a save icon (floppy disk) and an edit icon (pencil). Below the header is a white rectangular input field. Underneath the input field is a light gray bar containing the text "Подарунок (Його властивості)". At the bottom of the interface, there are three buttons: "Зберегти" (Save) in a white box with a blue border, "Очистити" (Clear) in a solid blue box, and "Порада" (Advice) in a solid blue box.

Рис. 6.2.3 – Екранна форма збереження списку

При цьому необхідно пам'ятати про натиснення кнопки «Зберегти», інакше при завантаженні або натисненні кнопки очистити все буде знищене.

ПП надає можливість користувачу ролі «гість» ознайомитися із інтерфейсом сайту, натиснувши кнопку «Порада», як показано на рисунку 4 та 5.

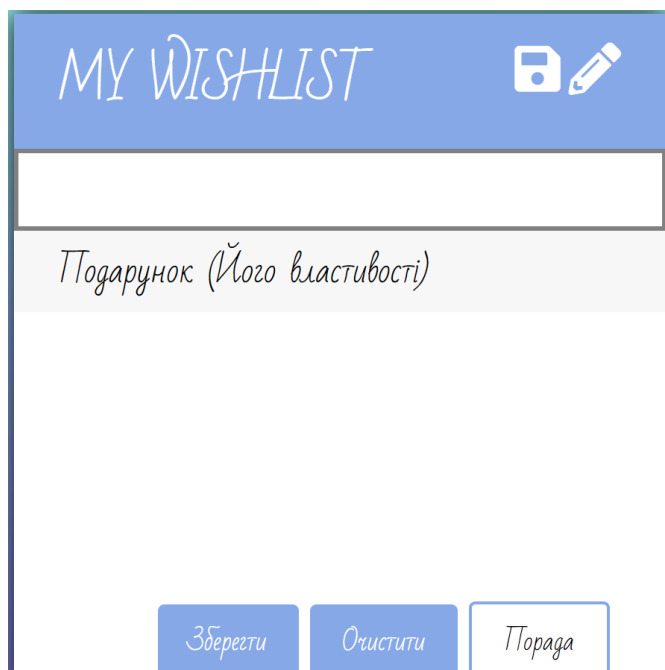


Рис. 6.2.4 – Екранна форма кнопки «Порада»

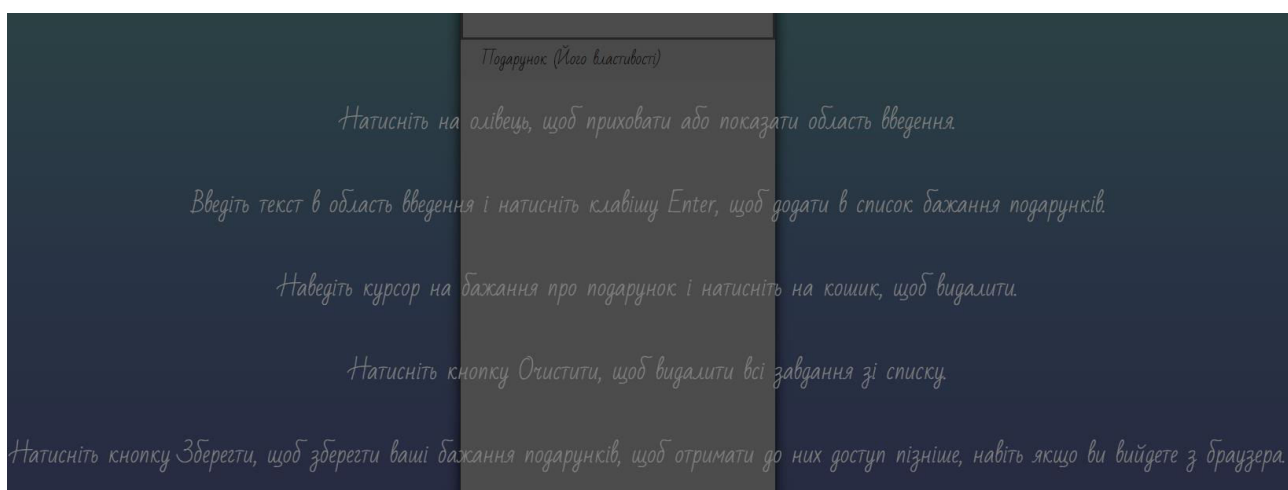


Рис. 6.2.5 – Екранна форма виведення значення кнопки «Порада»

ПП надає можливість користувачу ролі «гість» сховати поле вводу подарунків для комфортного перегляду та «скріншоту» списку, натиснувши кнопку «Олівець», як показано на рисунку 6.



Рис. 6.2.6 – Екранна форма кнопки «Олівець»

Для виконання користувач повин натиснути на зображення кнопки «Олівця», як показано на рисунку 7.

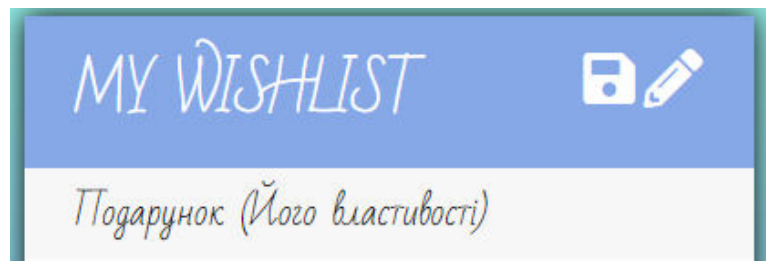


Рис. 6.2.7 – Екранна форма демонстрації виконання кнопки «Олівець»

ПП надає можливість користувачу ролі «гість», зберегти свій створений список подарунків натиснувши кнопку «Дискети», як показано на рисунку 8.

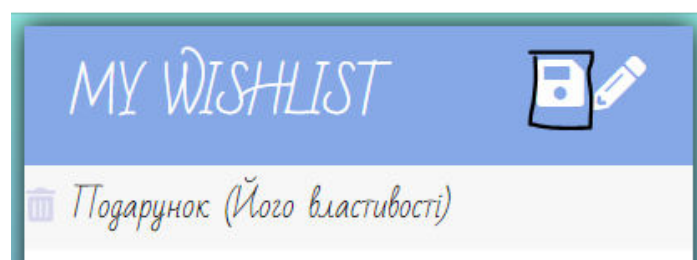


Рис. 6.2.8 – Екранна форма демонстрації кнопки «Дискета»

Для виконання користувач повин натиснути на зображення кнопки «Дискета», як показано на рисунку 9.

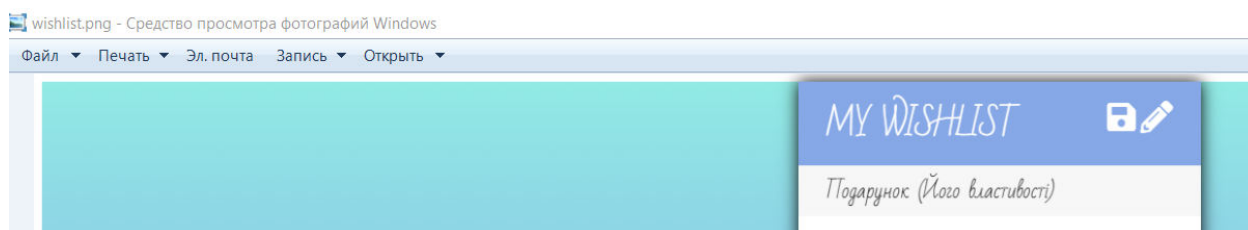


Рис. 6.2.9 – Екранна форма демонстрації виконання кнопки «Дискета»

ПП надає можливість користувачу ролі «гість» за допомогою комбінацій клавіш **ctrl+p** зробити «скріншот» списку, як показано на рисунку 10.

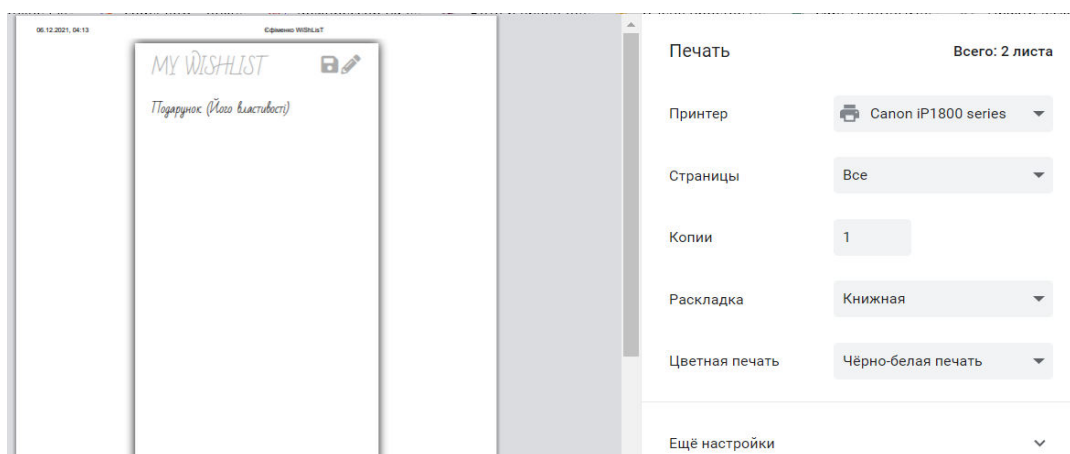


Рис. 6.2.10 – Екранна форма «Скріншоту» списку подарунків



Рис. 6.2.11 – Демонстрація списку виконаного «скріншоту»

ПП надає можливість користувачу ролі «гість» видаляти подарунки по одному або очищувати список повністю, натиснувши кнопку «Очистити», як показано на рисунку 12.

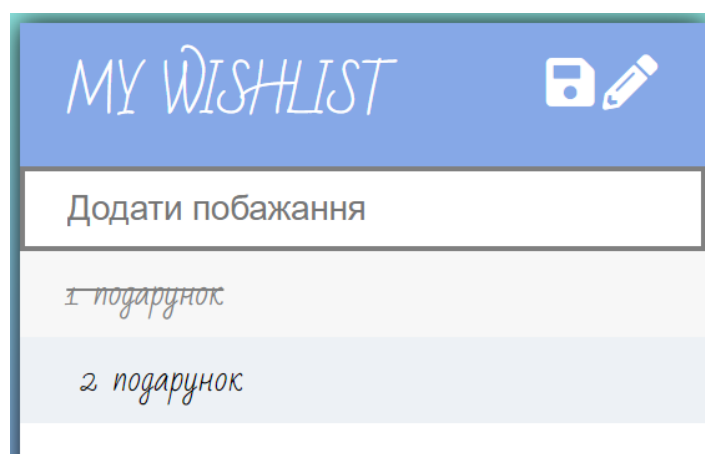


Рис. 6.2.12 – Екранна форма демонстрації виконаного пункту подарунку методом викреслювання

Для виконання повного видалення пункту потрібно натиснути на зображення кнопки «Корзини для сміття», як показано на рисунку 13.

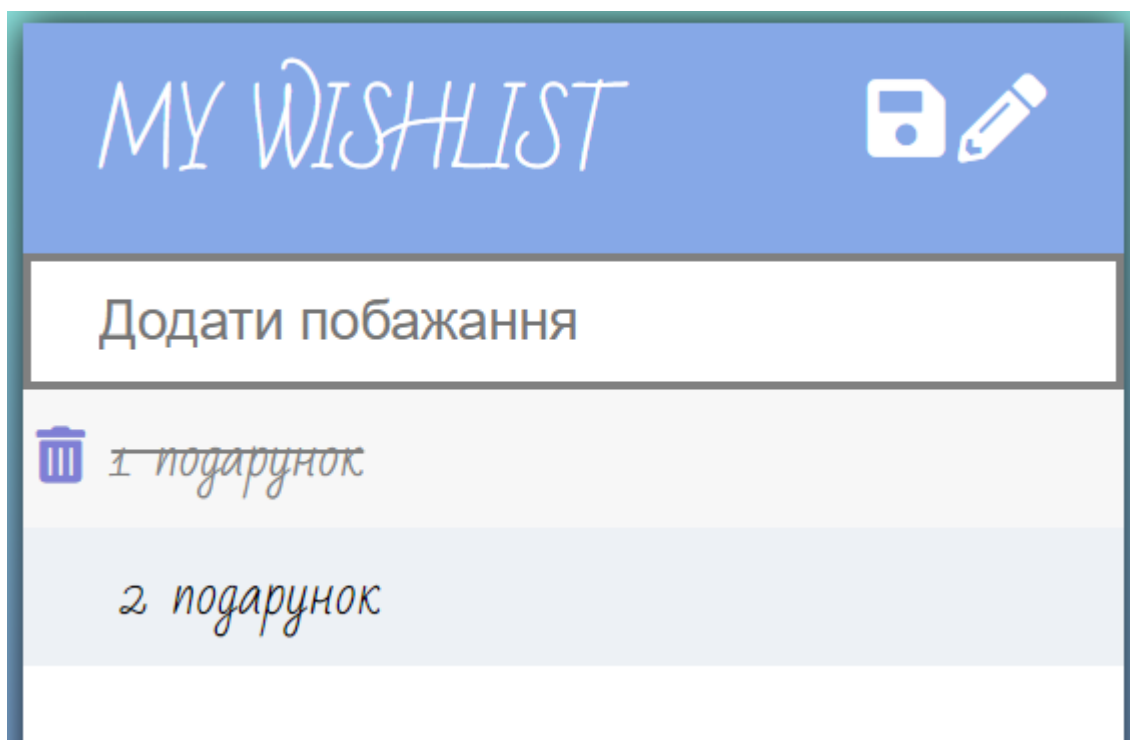


Рис. 6.2.13 – Екранна форма повного видалення пункту меню за допомогою кнопки «Корзина для сміття»

Для виконання повного очищення списку потрібно натиснути на кнопку «Очистити», як показано на рисунку 14.

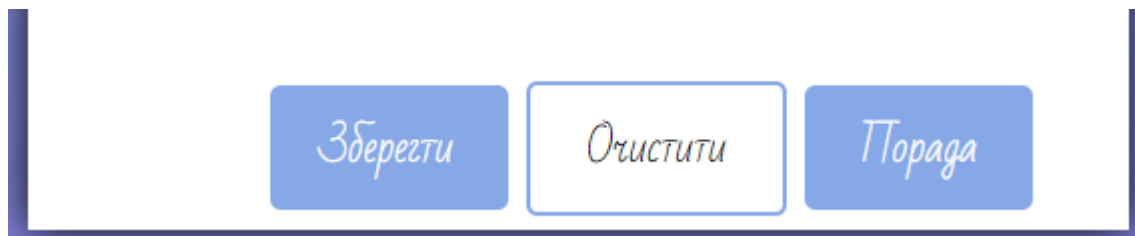


Рис. 6.2.14 – Приклад екранної форми кнопки «Очистити»

6.3 Результати валідації програмного продукту

Метою створення ПП, є скорочення часу на пошук подарунків, за допомогою розробки веб-сервісу для формування побажань подарунків та корисних речей, на якому користувач зможе розміщувати свої бажання подарунків і будь-яка людина, яка перейде за посиланням або отримає список, зможе виконати його мрію і подарувати йому дійсно корисний подарунок та корисну річ, яка буде ним використовуватися.

1. Експеримент. Авторизація та створення WishList (повний цикл створення)

Раніше потрібно було зробити приблизно до ~80 не потрібних та зайвих натискань клавіш і кліків, враховуючи реєстрацію на веб-ресурсі та інтуїтивно не зрозумілого інтерфейсу без підказок.

Існуючі ПП, кроки:

- 1) Пошук сервісу приблизно ~10-15 натискань кліків миші та клавіш клавіатури;
- 2) Реєстрація на веб-ресурсі до ~30 натискань кліків миші та клавіш клавіатури;
- 3) Ознайомлення з інтуїтивно не зрозумілим інтерфейсом та надто нагромадженою сторінкою ~25 натискань кліків миші та клавіш клавіатури;
- 4) Заповнення списку подарунків ~10+ натискань кліків миші та клавіш клавіатури;
- 5) Збереження списку бажань ~5-10 натискань кліків миші та клавіш клавіатури;
- 6) Поділитися списком подарунків ~5-10 натискань кліків миші та клавіш клавіатури.

Створений ПП, кроки:

1) Пошук сервісу приблизно ~10-15 натискань кліків миші та клавіш клавіатури;

2) Ознайомлення з інтуїтивно зрозумілим інтерфейсом ~5 натискань (або 1 клік миші по «Пораді») кліків миші та клавіш клавіатури;

3) Заповнення списку подарунків ~5-10+ натискань кліків миші та клавіш клавіатури;

4) Збереження списку бажань ~5 натискань кліків миші та клавіш клавіатури;

5) Поділитися списком подарунків ~3 натискання кліків миші та клавіш клавіатури.

Як, можна побачити з експерименту, пункт 2 взагалі не потрібен, а всі інші пункт можна оптимізувати.

Тому, було створено зручний веб-сервіс, для того, щоб швидко створити WishList. Створений ПП дозволяє в 2-2,5 рази швидше створити Wishlist та зекономити час, для інших корисних речей.

2. Експеримент. Створення WishList (Цикл створення WishList)

Існуючі ПП, кроки:

Веб-сервіс №1

1) Пошук сервісу приблизно ~10-15 натискань кліків миші та клавіш клавіатури;

2) Роздруківка на принтері WishList ~10 натискань кліків миші та клавіш клавіатури (заповнення в Word або пункт 3);

3) Заповнюємо від руки ~30 хвилин;

4) Створення фотографії та її поширення ~25-35 натискань кліків миші та клавіш клавіатури.

Веб-сервіс №2

- 1) Пошук сервісу приблизно ~10-15 натискань кліків миші та клавіш клавіатури;
- 2) Створення WishList ~20-30 натискань кліків миші та клавіш клавіатури;
- 3) Заповнення побажаннями ~5+ натискань кліків миші та клавіш клавіатури;
- 4) Створення посилання та її поширення ~25-35 натискань кліків миші та клавіш клавіатури.

Створений ПП, кроки:

- 1) Пошук сервісу приблизно ~10-15 натискань кліків миші та клавіш клавіатури;
- 2) Створення WishList ~5 натискань кліків миші та клавіш клавіатури (Можливість роздрукування та заповнення від руки);
- 3) Заповнення побажаннями ~5+ натискань кліків миші та клавіш клавіатури;
- 4) Створення посилання та її поширення ~5-10 натискань кліків миші та клавіш клавіатури.

Можна зробити висновок на основі 2 експерименту, що розроблений веб-сервіс WishList, економить час на створення має більший функціонал, потребує менше натискань та часу на вивчення інтерфейсу.

Висновки

В результаті виконання етапів курсової роботи було створено програмне забезпечення процесу швидкого пошуку побажань користувачів.

Створене програмне забезпечення досягло наступної мети його споживача: розробка веб-сервісу для формування побажань подарунків та корисних речей, на якому користувач зможе розміщувати свої бажання подарунків і будь-яка людина, яка перейде за посиланням або отримає список, зможе виконати його мрію і подарувати йому дійсно корисний подарунок та корисну річ, яка буде ним використовуватися, також метою є скорочення часу на пошук подарунків.

Доказом цього є наступні факти:

- 1) Зменшення кількості кліків для пошуку та додавання подарунків;
- 2) зменшення кількості кліків для пересилання списків;
- 3) спрощення інтерфейсу керування.

Вказані факти перетворили програмне забезпечення на програмний продукт.

В процесі створення програмного продукту виникли такі труднощі (організаційні, проблеми відсутності досвіду, знань, потрібних в різних етапах):

- 1) Брак досвіду та знань;
- 2) брак часу.

Через вищеописані непередбачені труднощі, а також через обмежений час на створення програмного продукту, залишаються нереалізованими такі прецеденти або їх окремі кроки роботи:

- 1) Реєстрація (1 крок);
- 2) авторизація (2 крок);
- 3) каталог (5 крок).

Зазначені недоробки планується реалізувати в майбутніх курсових роботах з урахуванням тем дисциплін наступних семестрів.