

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра прикладної математики

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КУРСОВОЇ РОБОТИ

з дисципліни “Бази даних та інформаційні системи”

на тему: КУПІВЛЯ ПОДАРУНКІВ

Студентки IV курсу, групи КМ-41
напряму підготовки 6.040301 –
прикладна математика
ТОКАР С.Ю,

Викладач
ТЕРЕЩЕНКО І.О.

Оцінка: ____ балів

Київ – 2017

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

В ході курсової роботи потрібно розробити інформаційну систему для автоматизації та оптимізації процесу вибору подарунків. Для реалізації програмного продукту потрібно:

- 1) провести передпроектне дослідження, тобто чітко визначити цілі проекту, визначити граничні умови для розроблюваної системи, провести аналіз можливих груп користувачів інформаційної системи та визначити їх ролі, продумати основний функціонал системи;
- 2) провести Scrum-планування проекту з описом всіх стадій розробки проекту, враховуючи визначені бізнес-правила функціонування системи;
- 3) для кожної групи користувачів описати дії та можливості при користуванні системою у формі Use Case діаграм, більш детально описати дії користувачів за допомогою діаграм послідовностей (Sequence Diagrams);
- 4) виділити основні компоненти системи та спроектувати взаємодію між цими компонентами у формі діаграми компонент (Component Diagram);
- 5) розробити семантичну модель даних, що відображає інформаційний зміст розроблюваної системи, тобто виділити основні сутності системи, їх атрибути, зв'язки між ними та спроектувати логічну та фізичну структуру бази даних системи;
- 6) розробка клієнтського застосунку та клієнт-серверної взаємодії, провести валідацію даних на клієнті та на сервері.

АНОТАЦІЯ

Метою даної роботи є автоматизація та оптимізація процесу складання та вибору подарунків. Для досягнення поставленої мети в ході роботи був проведений аналіз можливих груп користувачів, розроблено бізнес-правила для цієї системи, проведено проектування процесу розробки системи.

Дана робота складається з 5 розділів. Перший розділ містить передпроектне дослідження, де визначено основні цілі проекту, проведено аналіз існуючих систем. На підставі семантичного аналізу розробляється система, яка зможе покращити та оптимізувати процес купівлі подарунків в порівнянні з існуючими системами.

У другому розділі сформульовані основні вимоги до розроблюваної системи, сформовано ділову модель, яка описує функціонування системи та визначає основні бізнес-правила системи.

У третьому розділі проведено моделювання та планування процесу розробки програмного застосунку, визначено основні компоненти системи та зв'язки між ними, визначено основні групи користувачів та їх ролі.

В четвертому розділі отримано семантичну модель даних, що відображає інформаційний зміст розроблюваної системи.

П'ятий розділ містить даталогічне проектування, в результаті якого отримуємо логічну та фізичну моделі системи.

Робота виконана на 72 сторінках, містить 17 таблиць, 5 рисунків, 18 літературних джерел, 8 додатків.

РЕФЕРАТ

Курсова робота обсягом 40 аркуші складається із вступу, п'яти розділів, висновків та переліку посилань із 13 джерел. Містить 16 рисунків та 3 таблиці.

Метою підготовки курсової роботи є закріплення знань, набутих при вивченні дисципліни «Бази даних та інформаційні системи», а основним завданням курсової роботи є проектування та розробка інформаційної системи для полегшення процесу купівлі подарунків.

Об'єктом дослідження курсової роботи є процес вибору, складання та покупки подарунків.

Предметом дослідження є стадії відповідного процесу та етапи створення автоматизованої системи, яка буде реалізовувати даний процес та допоможе оптимізувати його.

В даній курсовій роботі було проведено проектування потрібної системи та визначено етапи її реалізації, для чого було вивчено технології та методи планування розробки інформаційних систем. Крім того було проведено аналіз можливих груп користувачів системи, їхніх потреб, відповідно до цього були описані бізнес-правила функціонування системи та спроектовано необхідний функціонал.

Дана курсова робота присвячена плануванню та розробці інформаційних систем та проектуванню необхідної бази даних для функціонування цієї системи. Ключові слова: Scrum-планування, групи користувачів, Use Case діаграми, діаграма послідовностей, діаграма компонент, концептуальна модель системи, даталогічне проектування, логічна модель, фізична модель, база даних, сутність, атрибути, зв'язки між сутностями.

ЗМІСТ

1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ.....	7
2 ПОСТАНОВКА ЗАДАЧІ	9
3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ	14
4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ.....	28
5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ	29
ВИСНОВКИ	32
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	33

ВСТУП

Сучасний стан інформаційних систем та технологій можна охарактеризувати наступними тенденціями:

1. Наявність великої кількості промислово функціонуючих баз даних великого обсягу, що містять інформацію практично по всіх видах діяльності суспільства.

2. Створення технологій, що забезпечують інтерактивний доступ масового користувача до цих інформаційних ресурсів.

3. Розширення функціональних можливостей інформаційних систем, що забезпечують паралельну одночасну обробку баз даних з різноманітною структурою даних, мультиоб'єктних документів, гіперсередовища, в тому числі реалізують технології створення та ведення гіпертекстових баз даних.

4. Включення в інформаційні системи елементів інтелектуалізації інтерфейсу користувача, експертних систем, систем машинного перекладу, автоіндексування та інших технологічних засобів.

В наслідок чого, інформаційний продукт у вигляді програмних засобів, баз даних і служб експертного забезпечення набуває стратегічного значення.

З ростом значущості інформаційного продукту можливість провести ідеальний обмін цим продуктом між комп'ютером і людиною або між інформаційними системами набуває значення ведучої технологічної проблеми. Фірми можуть за допомогою інформаційних технологій вести справи де завгодно, отримуючи вичерпну інформацію. Глобалізація ринку інформаційного продукту націлена на отримання переваг за рахунок розподілу постійних і напівпостійних витрат на більш широкий географічний регіон.

1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ

Для просування розроблюваного програмного продукту на ринку необхідно: розробити бізнес-план для фінансового обґрунтування створення веб-сервісу; створити потужну динамічну базу даних з можливістю легкого внесення змін та її доповнення; сформулювати умови співпраці з офлайн торговельними мережами.

Програмний продукт буде мати модуль для реалізації «вибору подарунків в онлайн режимі», моніторингу оновлення товарів в системі, рух товарів за допомогою служб доставки. Було проведено SWOT-аналіз розроблюваного продукту та сформовано шаблон бізнес моделі (таблиця. 1.1).

Сили	Слабкості
<ul style="list-style-type: none">• Програмний продукт, що є веб-сервісом;• Зручний, зрозумілий інтерфейс;• Локалізація системи для користувачів з України;• Отримання інвестицій за рахунок реклами товарів магазинів.	<ul style="list-style-type: none">• Відсутність інвесторів;• Вимагає підключення до інтернет мережі;• Наявність закордонних аналогів.
Можливості	Загрози
<ul style="list-style-type: none">• Створити кошик для подарунків;• Додати вибрану кількість товарів до кошика;• Переглядати вміст кошика;• Замовити кошик.	<ul style="list-style-type: none">• Погане покриття мережі;• Обмежений асортимент товарів;• Нема товарів в наявності (несинхронізовані дані).

Таблиця 1.1 – SWOT – аналіз розроблюваної системи

Виконаний SWOT-аналіз дозволяє виявити сильні та слабкі сторони, основні можливості, необхідність розробки та ризику використання програмного продукту.

Головна мета програмного продукту - це економія часу. Він дозволить оптимізувати пошукові процеси та є основою для автоматизації процесу складання подарунків.

Практичне значення одержаних результатів – даний програмний продукт є ефективним щодо спрощення процесу вибору та складання подарунків.

2 ПОСТАНОВКА ЗАДАЧІ

На даному етапі мають бути визначені цілі та призначення автоматизованої інформаційної системи, сформульовані основні вимоги до розроблюваної системи та перелік основних функцій.

Для реалізації даної системи було обрано створення інтернет-магазину (веб-сайт) для створення і покупки подарунків. Метою даної системи є автоматизація і оптимізація процесу вибору подарунків, а саме повинна бути розроблена можливість скласти подарунок з представлених товарів і вибрати спосіб упакування; можливість замовити обраний подарунок з доставкою. Відмінністю від інших систем є те, що дана система має опцію автоматичного складання подарунку (генерація можливих варіантів подарунку) на основі відповідей користувача в опитуванні з розумною формою (нові запитання показуються респондентам в залежності від їхніх попередніх відповідей).

2.1. Граничні умови

Передбачено граничні умови та обмеження в сервісі. Система розробляється з розрахунком на користувачів з України. В системі є доступними конкретні товари, що містяться на складі магазину на час замовлення.

2.2. Категорії користувачів системи

В системі, що розробляється, визначено наступні групи користувачів: гість (неавторизований користувач), авторизований користувач (після реєстрації та авторизації в системі), адміністратор системи. Відповідно до

аналізу потреб кожної з груп було визначено ролі користувачів та можливості роботи в системі:

1) користувачу-гостю доступні такі дії: перегляд всіх доступних товарів в системі, перегляд детальної інформації про певний товар, можливість реєстрації на сайті;

2) зареєстрований користувач після авторизації в системі має наступні можливості: перегляд всіх доступних товарів в системі, можливість пройти опитування, можливість зробити замовлення, можливість залишити відгук чи побажання);

3) адміністратору доступні такі дії: додавання нових товарів в базу системи, редагування інформації вже існуючих в базі товарів, менеджмент замовлень, доступ до всіх даних на сайті окрім даних процесу перерахування коштів, що здійснюється за рахунок банківської системи Приват 24, перегляд побажань користувачів щодо покращення якості системи і щодо внесення до бази певних товарів; додавання і редагування питань опитування, а також схеми вибору наступного питання.

2.3. Основний функціонал та класи даних

Далі наведено детальний опис деяких основних функцій системи:

1) Додавання товарів в систему (час виконання – діапазон $< 1\text{хв}$; джерело інформації – адміністратор; структура для збереження даних – таблиця в базі даних): параметрами функції є дані про товар, який додається в систему (код, назва, опис, ціна, категорія, коментар, постачальник, кількість товару на складі), ідентифікатор адміністратора, який виконує операцію (кодове ім'я); результатом виконання функції є збереження даних в таблицю доступних в системі товарів.

2) Додавання товару до подарунку (час виконання – діапазон $< 10\text{сек}$; джерело інформації – база даних товарів; структура для збереження даних –

таблиця бази даних): параметрами функції є ідентифікатор користувача, який додає товар (номер, присвоєний користувачу під час реєстрації), ідентифікатор замовлення, до якого додають товар (код замовлення/ код кошика), ідентифікатор товару, який додається (код товару); результатом виконання функції є внесення даних в таблицю замовлених товарів.

3) Автоматичний підбір можливих варіантів подарунку на основі відповідей з розумної форми опитування (час генерації – діапазон < 2 хв; джерело інформації – таблиця, де відповіді користувача на запитання в опитуванні; структура для збереження даних – таблиця бази даних): параметрами функції є ідентифікатор користувача, який проходив опитування (номер, присвоєний користувачу під час реєстрації), ідентифікатор опитування (унікальний код опитування, яке пройшов користувач), результатом виконання функції є генерація можливих варіантів подарунку для даного користувача в результаті проходження даного опитування (варіанти зберігаються в таблиці можливих замовлень (таблиця згенерованих кошиків) або в таблиці замовлень (загальна таблиця кошиків) з коментарем «можливе замовлення»).

4) Менеджмент замовлення – адміністратор вибирає замовлення для подальшого опрацювання (час виконання – діапазон < 1 хв; джерело інформації – загальна таблиця замовлень/або таблиця замовлень, які потрібно опрацювати; структура збереження даних – загальна таблиця замовлень/ або таблиця замовлень, які опрацьовуються): параметрами функції є ідентифікатор адміністратора, який буде опрацьовувати замовлення (кодове ім'я), ідентифікатор замовлення (код замовлення); результатом виконання функції є зміна статусу замовлення на «опрацьовується» і внесення кодового ім'я адміністратора як людини, яка займається опрацюванням конкретного замовлення, до інформації про замовлення в загальну таблицю замовлень (або збереження інформації в таблиці замовлень, що опрацьовуються).

2.4. Бізнес-правила

Бізнес логіка — система зв'язків та залежностей елементів бізнес-даних та правил обробки цих даних відповідно до особливостей ведення окремої діяльності (бізнес-правил), яка встановлюється при розробці програмного забезпечення, призначеного для автоматизації цієї діяльності.

Бізнес логіка описує бізнес-правила реального світу, які визначають способи створення, представлення та зміни даних. Бізнес-правила розглядаються як вимоги до моделей справжніх бізнес-об'єктів.

В даній системі були визначені такі бізнес-правила:

- 1) лише авторизовані користувачі системи можуть створювати подарункові кошики та додавати в них вибрані товари, причому додати можна таку кількість одиниць товару, що не перевищує кількість одиниць на складі магазину;
- 2) для того, щоб замовити складений кошик, користувач повинен надати свої адресні дані або адресу, на яку повинна бути здійснена доставка;
- 3) для подальшого опрацювання замовлення користувач повинен оплатити вартість складеного подарункового кошика;
- 4) після підтвердження оплати обрані товари упаковуються відповідно до побажань користувача та відправляються службою доставки (це виконує обслуговуючий персонал за вказівками адміністратора системи);
- 5) можлива експрес-доставка (на яку встановлено додаткову ціну) або можливий самовивіз за бажанням користувача;
- 6) повернення коштів можливе лише у випадку втрати товарами їх товарного вигляду чи функціональності за час, що визначений адміністратором для кожного з товарів;
- 7) дані користувача (адреса електронної пошти) можуть використовуватись для розсилок про оновлення на складі, акції, знижки і т.д. (за попередньою згодою користувача).

2.5. Матриця елементарних подій

Матриця елементарних подій представлена в таблиці 2.1.

№	Опис події	Тип події	Реакція на подію
1	Користувач хоче переглянути всі доступні товари	N	Надати список всіх доступних товарів
2	Користувач хоче сформувати замовлення	N	Надати відповідну форму для вводу. Зберегти запис.
3	Користувач хоче оплатити замовлення	N	Сформувати замовлення та повернути результат
4	Користувач хоче переглянути свої замовлення	N	Надати список замовлень користувача
5	Адміністратор хоче редагувати інформацію про товар в системі	NN	Надати форму для редагування і зберегти результат
6	Адміністратор хоче видалити товар з системи	NN	Видалити вибраний товар
7	Адміністратор хоче редагувати замовлення в базі даних	NN	Надати форму для редагування і зберегти результат
8	Адміністратор хоче видалити замовлення в базі даних	NN	Видалити вибране замовлення

Таблиця 2.1. Матриця елементарних подій

3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

3.1. Scrum-планування етапів розробки програмного продукту

Для реалізації системи купівлі подарунків було сплановано 4 спринти: «Create Basis for System», «Users», «Simple Gift Making» та «Managing Orders». В першому спринті «Create Basis for System» повинні бути реалізовані основні функції, за допомогою яких буде створено базу для системи. Перш за все - це створення ролі адміністратора (його авторизація – рис.3.1, рис.3.2), який зможе працювати з інформацією в системі, а саме - додавати новий товар в систему (рис.3.3, рис.3.4, рис.3.5).

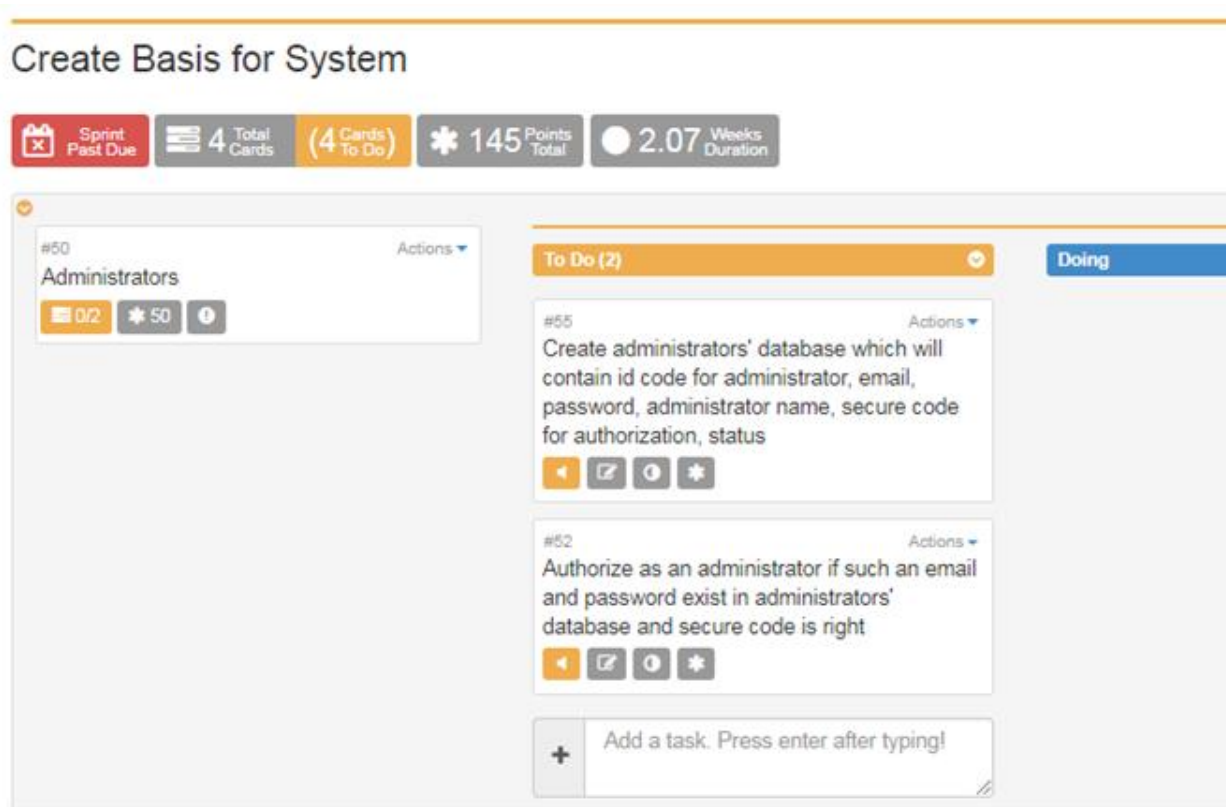


Рисунок 3.1 – Планування створення та авторизації адміністратора.

Task Description (#52)

Administrator's authorization fields (interface requirements):

1. Email field: string (email field); max-length of string = 100 symbols; allowed alphanumeric characters and "_.@"; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=25%).
2. Password field: string (password field); max-length of string = 25 symbols, min-length of string = 10 symbols; allowed alphanumeric + special symbols; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=30%).
3. Secure code field: string; defined length of string = 10 symbols; allowed alphanumeric + special symbols; editable field; size (width=15% screen width, height=4% of screen height); position (left=40%, top=50%).

Button "Sign In": size (width=5% screen width, height=4% of screen height), position (left=60%, top=55%).

When the button "Sign In" is clicked and filled information is right (record with such information exists in administrators' database), then redirect to the main page, register in the system as "online" (change status field in administrators' database for appropriate administrator), all pages are opened with options for administrators.

Else show pop-up message that signing in wasn't successful (message appears on the top of the page for 1 minute; size (width=50% of screen width, height=6% of screen height), position (left=25%, top=15%); and then the message disappears.

Close

Task Description (#55)

Requirements for administrators' database fields:

1. ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; unique for each record.
2. Email field: string (email field); max-length of string = 100 symbols; allowed alphanumeric characters and "_.@".
3. Password field: string (password field); max-length of string = 25 symbols, min-length of string = 10 symbols; allowed alphanumeric + special symbols.
4. Administrator name field: string (text field); max-length of string = 20 symbols; allowed alphabetic characters.
5. Secure code field: string; defined length of string = 10 symbols; allowed alphanumeric + special symbols; editable field - this field is changed every week; every code is unique.
6. Status field: boolean (two options: "online" and "offline"; set "offline" as a default); is changed when appropriate administrator authorizes.

Close

Рисунок 3.2 – Опис полів для форми авторизації адміністратора та відповідних полів в таблиці.

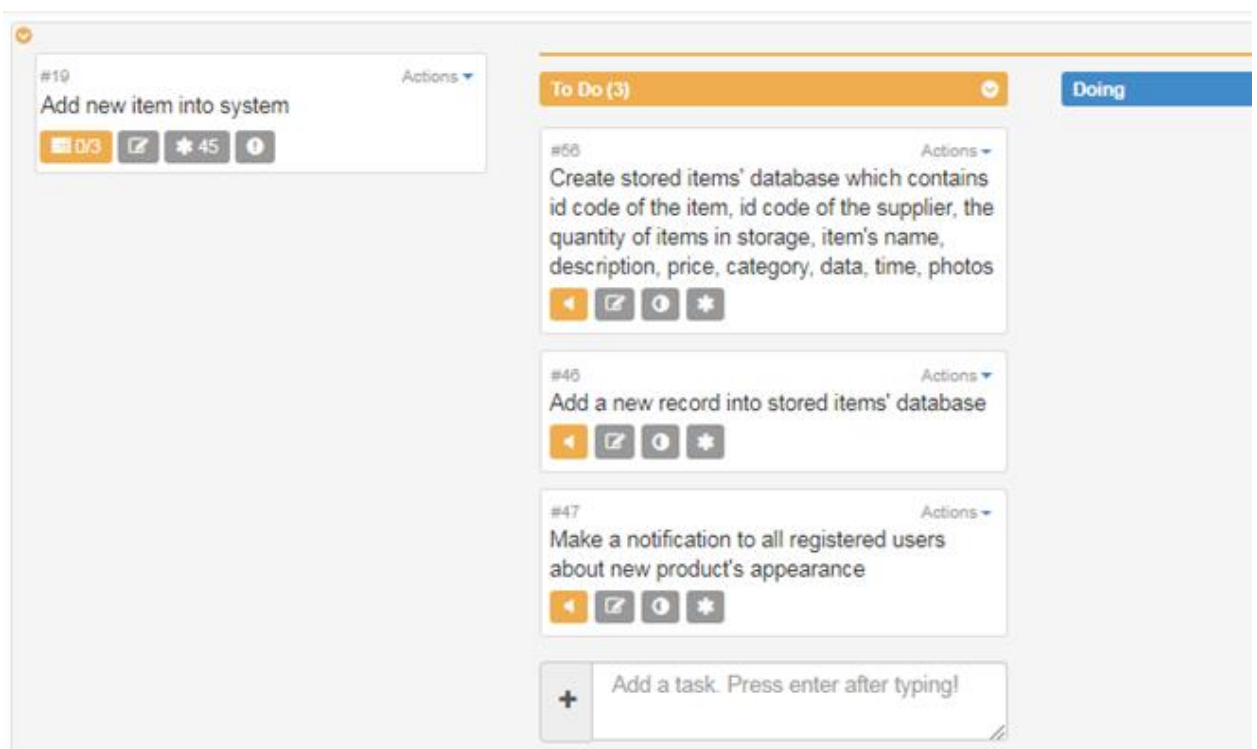


Рисунок 3.3 – Планування функції додавання нового товару в систему.

Task Description (#47)

Requirements for the interface:

1. List of all products that were added or edited recently (define them by date in the stored items' database): contains items' names; size (width=15% of screen width, height=10% of screen height), position (left=40%, top=20%).
2. Button "Select": if the button is clicked then add all information about the selected item (from stored items' database) in the message (or add an item's page URL); size (width=6% of screen width, height=4% of screen height), position (left=40%, top=35%).
3. List of users' emails, that could be selected (when email is pressed, then add it to the emails to which the message should be sent); size (width=75% of screen width, height=10% of screen height), position (left=10%, top=40%).
4. Button "Select all" that will add all users' emails to the emails to which the message should be sent; size (width=6% of screen width, height=4% of screen height), position (left=15%, top=55%).
5. Message field: string (text field, consist of URL addresses of items pages), max-length of string = 70 symbols; allowed alphanumeric characters and special symbols; editable field; size (width=75% of screen width, height=29% of screen height), position (left=15%, top=60%).
6. Button "Send": when button is clicked, send a requested message to selected users; size (width=10% of screen width, height=4% of screen height), position (left=70%, top=90%).

Close

Task Description (#56)

Requirements for stored items' database fields:

1. Item ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; unique for each record; not editable field.
2. Supplier ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters.
3. Quantity of items in storage field: integer (numeric field); allowed only numeric symbols; editable field.
4. Item's name field: string (text field); max-length of string = 30 symbols; allowed alphabetic characters.
5. Description field: string (text field); max length of string = 1000 symbols; allowed alphanumeric + special symbols; editable field.
6. Price field: float (numeric field); allowed only float numbers in range from 0 to 10000; editable field.
7. Category field: string (text field); max length of string = 30 symbols; allowed alphabetic characters; editable field.
8. Date field: string (data format); default format "dd/mm/yyyy"; allowed numerical symbols and "/"; not editable field; filled automatically by the system.
9. Time field: string (time format); default format "hh:mm:ss"; allowed numerical symbols and ":"; not editable field; filled automatically by the system.
10. Photos: file, max size = 2 Mb; allowed formats: .jpg, .png, .gif; editable field.

Рисунок 3.4 – Опис полів таблиці, де зберігатиметься інформація про товари.

Task Description (#46)

Button "Add": size (width=5% of screen width, height=4% of screen height), position (left=70%, top=85%).

The result is adding an item to stored items' database, creating the page for this item with information about it, redirecting on the just created item's page and show pop-up message that adding was successful (message appears on the top of the page for 1 minute; size (width=50% of screen width, height=6% of screen height), position (left=25%, top=15%); and then the message disappears.

Adding an item fields (interface requirements):

1. Item ID code field: filled automatically by the system; size: (width=15% of screen width, height=4% of screen height), position: (left=40%, top=20%).
2. Supplier ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=25%).
3. Quantity of items in storage field: integer (numeric field); allowed only numeric symbols; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=30%).
4. Item's name field: string (text field); max-length of string = 30 symbols; allowed alphabetic characters; editable field; size (width=15% of screen width, height=4% of screen height); position: (left=40%, top=35%).
5. Description field: string (text field); max length of string = 1000 symbols; allowed alphanumeric + special symbols; editable field; size (width=15% of screen width, height=14% of screen height); position: (left=40%, top=40%).

6. Price field: float (numeric field); allowed only float numbers in range from 0 to 10000; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=55%).
7. Category field: string (text field); max length of string = 30 symbols; allowed alphabetic characters; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=60%).
8. Date field: string (data format); default format "dd/mm/yyyy"; allowed numerical symbols and "/"; filled automatically by the system after button "Add" is pressed; this field isn't shown on the page.
9. Time field: string (time format); default format "hh:mm:ss"; allowed numerical symbols and ":"; filled automatically by the system after button "Add" is pressed; this field isn't shown on the page.
10. Photos: file, max size = 2 Mb; allowed formats: .jpg, .png, .gif; editable field; size (width=30% of screen width, height=19% of screen height); position (left=40%, top=65%).

Close

Рисунок 3.5 – Опис полів інтерфейсу для додавання нового товару в систему.

В першому спринті також передбачені функції редагування та видалення товарів адміністратором (рис.3.6, рис.3.7, рис.3.8).

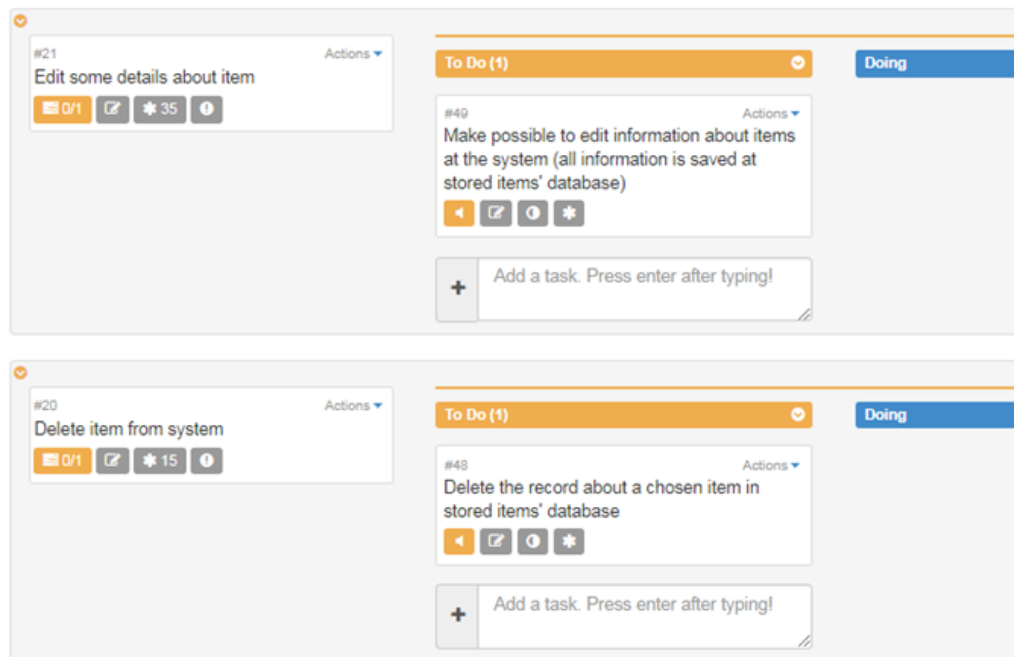


Рисунок 3.6 – Планування функцій редагування та видалення.

Task Description (#49)

An item which needs to be changed must be selected previously from the list of all products.

Button "Edit" (appears for each item on its page when it is opened/selected (only for administrators)): open page for editing; size (width=5% of screen width, height=4% of screen height), position (left=40%, top=20%).

Editing an item fields (interface requirements):

- Item ID code field (can't be edited): filled automatically by the system; size (width=15% of screen width, height=4% of screen height), position (left=40%, top=20%).
- Supplier ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=25%).
- Quantity of items in storage field: integer (numeric field); allowed only numeric symbols; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=30%).
- Item's name field: string (text field); max-length of string = 30 symbols; allowed alphabetic characters; editable field; size (width=15% of screen width, height=4% of screen height); position: (left=40%, top=35%).
- Description field: string (text field); max length of string = 1000 symbols; allowed alphanumeric + special symbols; editable field; size (width=15% of screen width, height=14% of screen height); position: (left=40%, top=40%).
- Price field: integer (numeric field); allowed only numeric symbols; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=55%).
- Category field: string (text field); max length of string = 30 symbols; allowed alphabetic characters; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=60%).
- Date field (must be changed if smth is edited): string (data format); default format "dd/mm/yyyy"; allowed numerical symbols and "/"; filled automatically by the system after button "Save" is pressed; this field isn't shown on the page.
- Time field (must be changed if smth is edited): string (time format); default format "hh:mm:ss"; allowed numerical symbols and ":"; filled automatically by the system after button "Save" is pressed; this field isn't shown on the page.
- Photos: file, max size = 2 Mb; allowed formats: .jpg, .png, .gif; editable field; size (width=30% of screen width, height=50% of screen height); position (left=50%, top=65%).

Button "Save": size (width=5% of screen width, height=4% of screen height), position (left=70%, top=85%).

The result of clicking button "Save" is updating the information about the item in stored items' database, updating information about the item on its page (is done automatically, because information for item's page is taken from stored items' database), redirecting on the item's page and show pop-up message that editing was successful (message appears on the top of the page for 1 minute; size (width=50% of screen width, height=6% of screen height), position (left=25%, top=15%); and then the message disappears.

Button "Cancel": size (width=5% of screen width, height=4% of screen height), position (left=71%, top=85%). When this button is clicked, then return to list of all products.

Close

Рисунок 3.7 – Опис інтерфейсу для функції редагування інформації про товар.

Task Description (#48)



An item which needs to be deleted must be selected previously from the list of all products.

Button "Delete" (appears for each item on its page, when it is opened/selected (only for administrators)): size (width=5% of screen width, height=4% of screen height), position (left=46%, top=20%).

The result is deleting information about the selected item from stored items' database, from the list of all products and deleting item's page. Redirecting on the main page and show pop-up message that deleting was successful (message appears on the top of the page for 1 minute; size (width=50% of screen width, height=6% of screen height), position (left=25%, top=15%); and then the message disappears.

Button "Cancel": size (width=5% of screen width, height=4% of screen height), position (left=53%, top=20%). When this button is clicked, then return to list of all products.

Close

Рисунок 3.8 – Опис інтерфейсу для функції видалення товару з системи.

В наступному спринті «Users» реалізовано функції реєстрації та авторизації користувача в системі (рис.3.9, рис.3.10, рис.3.11).

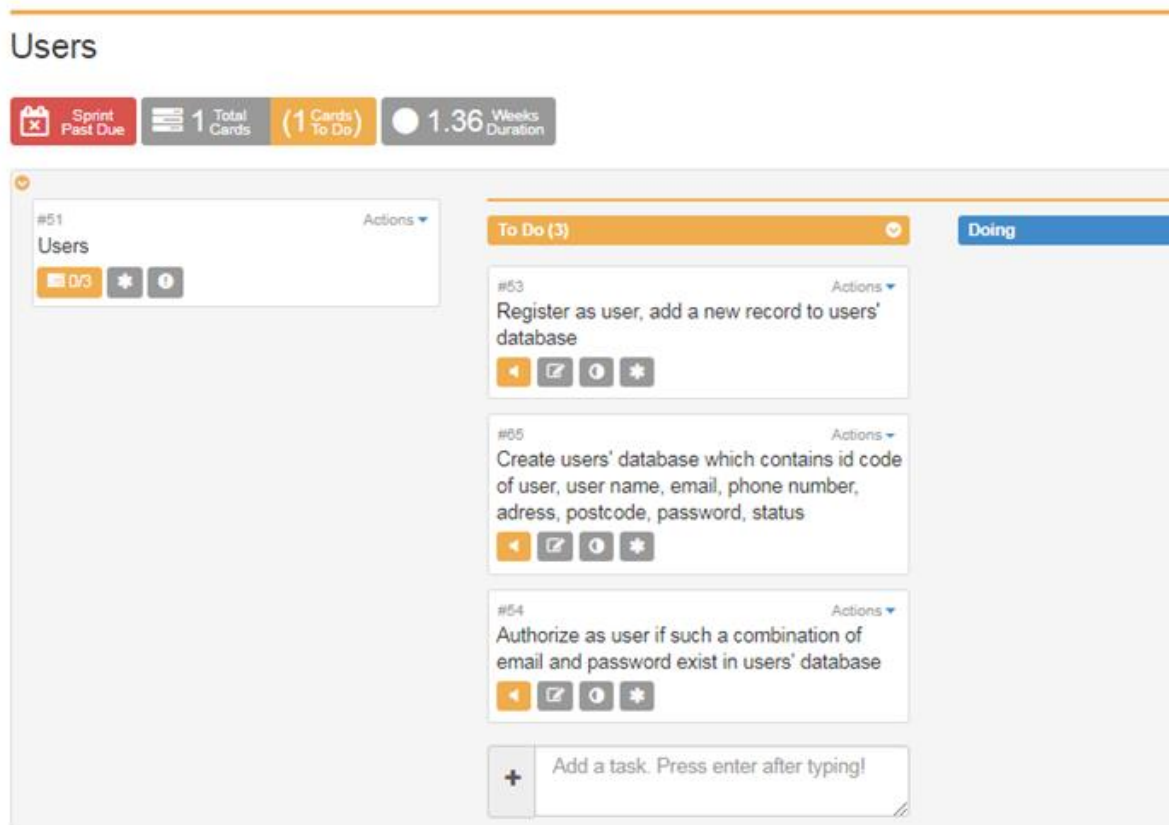


Рисунок 3.9 – Планування функцій реєстрації та авторизації користувачів.

Task Description (#53)



Button "Register": add all information from fields that are filled to users' database (is available only after filling in all required fields and checking the agreement with rules); size (width=5% of screen width, height=4% of screen height); position (left=50%, top=70%).

Requirements for users' registration fields:

1. User's name field: string (text field); max-length of string = 20 symbols; allowed alphabetic characters; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=25%).
2. Email field: string (email field); max-length of string = 100 symbols; allowed alphanumeric characters and "_@"; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=30%).
3. Password field: string (password field); max-length of string = 25 symbols, min-length of string = 10 symbols; allowed alphanumeric + special symbols; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=35%).
4. Phone number field: string (numeric field); defined length = 18 symbols, defined format +38(0XX)-XXX-XX-XX; allowed numeric characters and special symbols; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=40%).
5. Address field: string (text field); max-length = 50 symbols; allowed alphanumeric characters + all special symbols; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=45%).
6. Postcode field: string (number field); defined length of string = 6 symbols; allowed only numeric symbols; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=50%).
7. Agreement with rules: checkbox (if checked, then mske button "Register" available) + text field(default text: "I agree with all rules"); size of checkbox +text field (width=20% of screen width, height=2% of screen height), position (left=20%, top=55%); size of text field for rules (width=50% of screen width, height=10% of screen height), position (left=20%, top=55%);

Close

Рисунок 3.10 – Опис форми реєстрації.

Task Description (#65)



Task Description (#54)



Requirements for users' database fields:

1. ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; unique for each record; filled automatically by the system.
2. Name field: string (text field); max-length of string = 20 symbols; allowed alphabetic characters.
3. Email field: string (email field); max-length of string = 100 symbols; allowed alphanumeric characters and "_@".
4. Password field: string (password field); max-length of string = 25 symbols, min-length of string = 10 symbols; allowed alphanumeric + special symbols.
5. Phone number field: string (numeric field); defined length = 18 symbols, defined format +38(0XX)-XXX-XX-XX; allowed numeric characters and special symbols.
6. Address field: string (text field); max-length = 50 symbols; allowed alphanumeric characters + all special symbols.
7. Postcode field: string (number field); defined length of string = 6 symbols; allowed only numeric symbols.
8. Status field: boolean (two options: "online" and "offline"; set "offline" as a default); is changed when the appropriate user authorizes.

Close

User's authorization fields (interface requirements):

1. Email field: string (email field); max-length of string = 100 symbols; allowed alphanumeric characters and "_@"; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=30%).
2. Password field: string (password field); max-length of string = 25 symbols, min-length of string = 10 symbols; allowed alphanumeric + special symbols; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=35%).

Button "Sign In": size (width=5% screen width, height=4% of screen height), position (left=60%, top=45%).

When the button "Sign In" is clicked and filled information is right (record with such information exists in users' database), then redirect to the main page, register in the system as "online" (change status field in users' database for appropriate user), all pages are opened with options for users.

Else show pop-up message that signing in wasn't successful (message appears on the top of the page for 1 minute; size (width=50% of screen width, height=6% of screen height), position (left=25%, top=15%); and then the message disappears.

Close

Рисунок 3.11 – Опис полів таблиці для збереження даних користувачів та опис форми входу в систему.

В наступному спринті «Simple Gift Making» реалізовано функціонал складання подарунку: створення нового подарункового кошика (рис.3.12, рис.3.13), додавання товару у кошик (рис.3.14, рис.3.15), також можливість змінити деталі існуючого кошика, можливість залишити відгук на товар та можливість видалити вже існуючий подарунковий кошик (рис.3.16, рис.3.17).

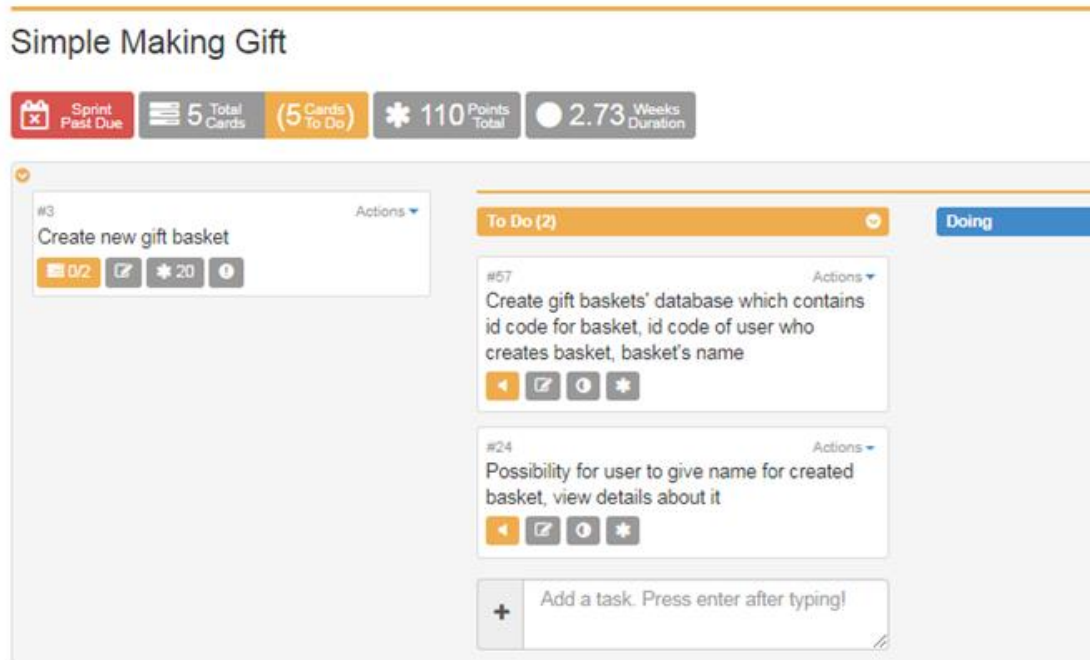


Рисунок 3.12 – Планування функцій створення подарункового кошика.

Task Description (#24)	Task Description (#57)
<p>When the button "Add to the basket" (this button is on every item's page) is clicked, the user can choose the existing basket from the list (if the user already created some baskets) or the user can create a new basket.</p> <p>Creating a new basket fields (interface requirements):</p> <ol style="list-style-type: none"> 1. Basket's name field: string (text field); max-length of string = 30 symbols; allowed alphabetic characters; editable field; size (width=15% of screen width, height=4% of screen height); position (left=40%, top=25%). <p>Button "Done": when this button is clicked, add new record into gift baskets' database with appropriate information (basket id code is generated by the system; user id code refers to user who wants to create new basket; basket's name corresponds to the name user wants to give his basket); size (width=6% of screen width, height=4% of screen height); position (left=41%, top=30%).</p> <p>Button "Cancel": when this button is clicked, return to item's page where button "Add to the basket" was clicked; size (width=6% of screen width, height=4% of screen height); position (left=48%, top=30%).</p>	<p>Requirements for gift baskets' database fields:</p> <ol style="list-style-type: none"> 1. Basket ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; unique for each record; not editable field. 2. FK User ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters. 3. Basket's name field: string (text field); max-length of string = 30 symbols; allowed alphabetic characters. 4. Date field: string (data format); default format "dd/mm/yyyy"; allowed numerical symbols and "/"; not editable field; filled automatically by the system (filled in when the basket is created and changes when it is updating). 5. Time field: string (time format); default format "hh:mm:ss"; allowed numerical symbols and ":"; not editable field; filled automatically by the system (filled in when the basket is created and changes when it is updating).
Close	Close

Рисунок 3.13 – Опис полів таблиці, куди будуть записані дані про створений кошик, та опис інтерфейсу.

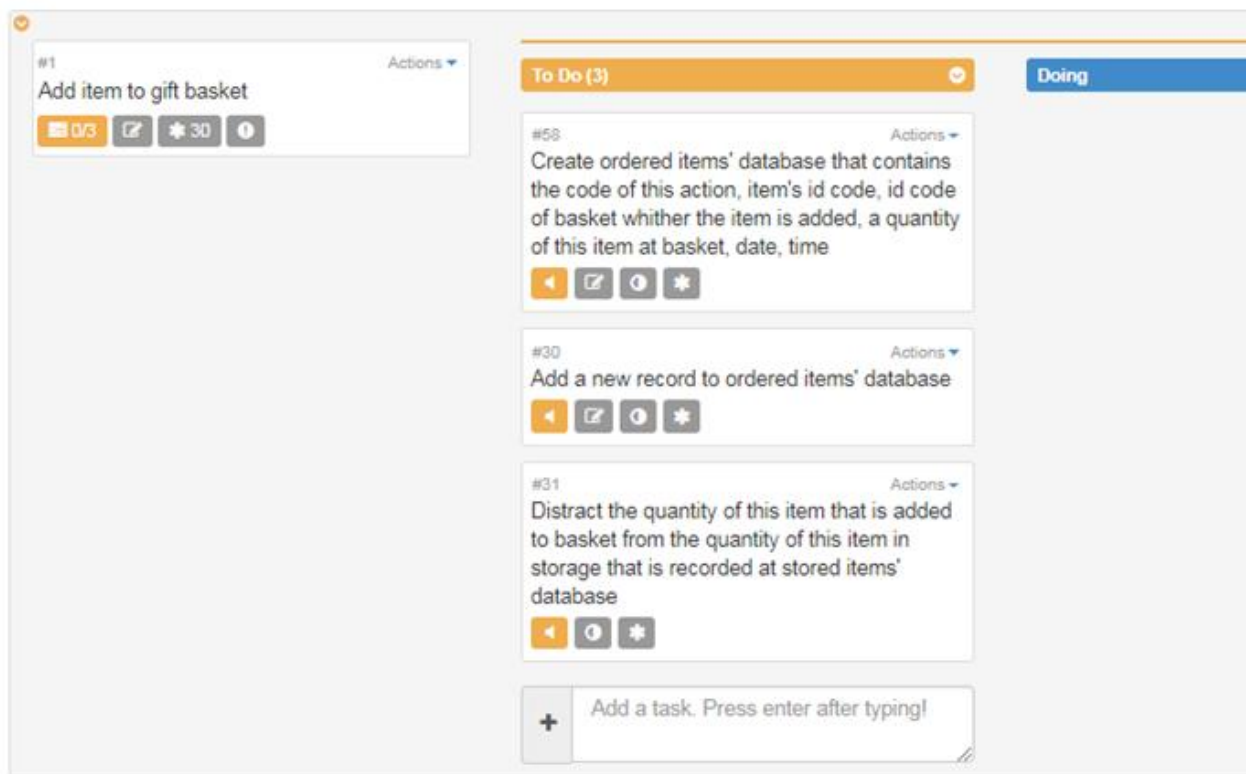


Рисунок 2.14 – Планування функції додавання товару у кошик.

Task Description (#58)

Requirements for ordered items' database fields:

1. Action code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; unique for each record; not editable field.
2. FK Item ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; not editable field.
3. FK Basket ID code field: string (text field); max-length of string = 10 symbols; allowed alphanumeric characters; not editable field.
4. Quantity of items in basket field: integer (numeric field); allowed only numeric symbols.
5. Date field: string (data format); default format "dd/mm/yyyy"; allowed numerical symbols and "/"; not editable field; filled automatically by the system (filled in when the item is added).
6. Time field: string (time format); default format "hh:mm:ss"; allowed numerical symbols and ":"; not editable field; filled automatically by the system (filled in when the item is added).

Close

Рисунок 3.15 – Опис полів таблиці, куди будуть записані дані про товари, що додані у відповідний кошик.

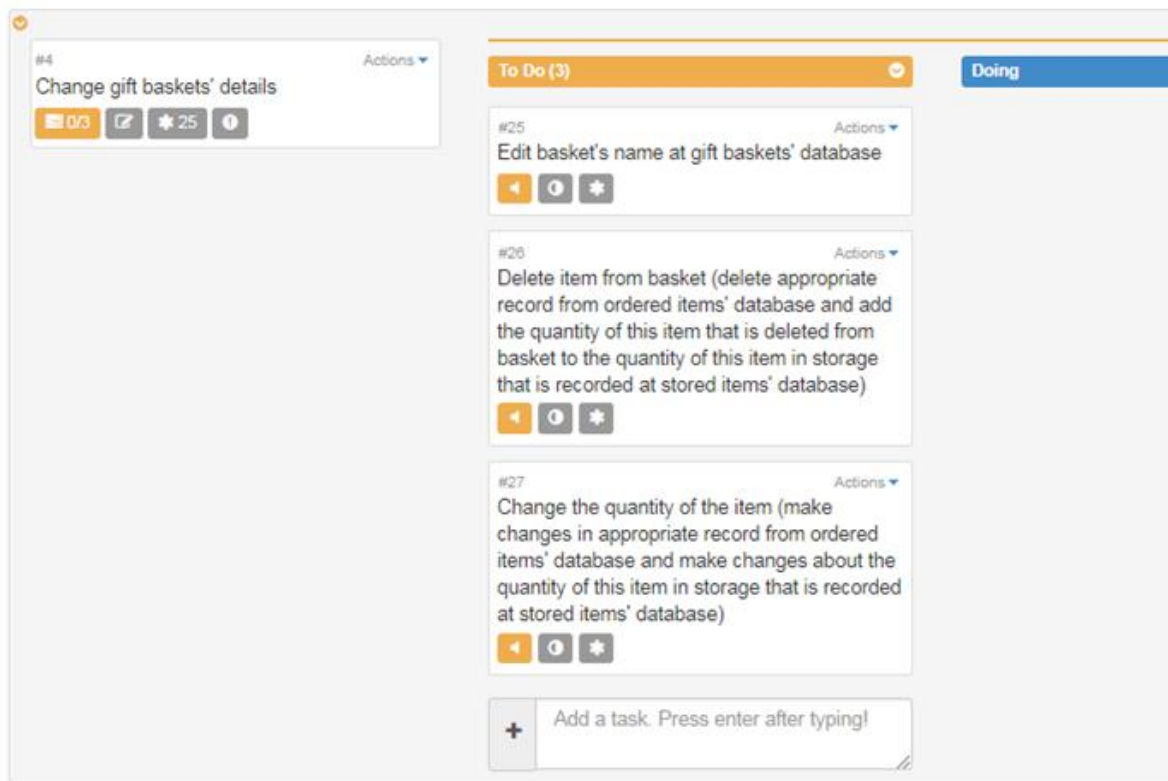


Рисунок 3.16 – Планування функції для редагування подарункового кошика.

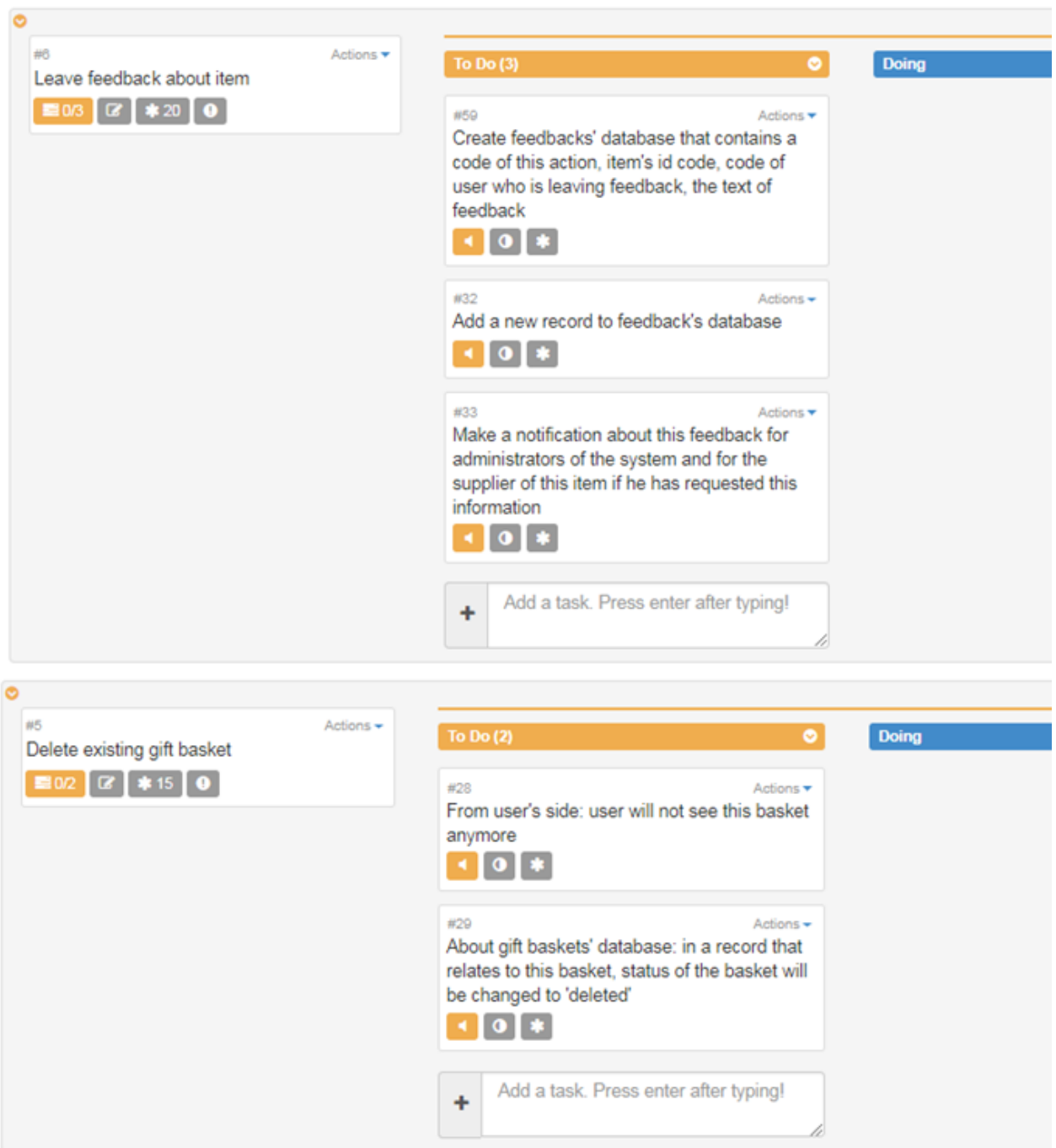


Рисунок 3.17 – Планування функцій для можливості залишити відгук про товар та можливості видалення існуючого кошика.

В останньому спринті «Managing Orders» буде реалізовано функціонал щодо менеджменту покупки створеного кошика (рис.3.18).

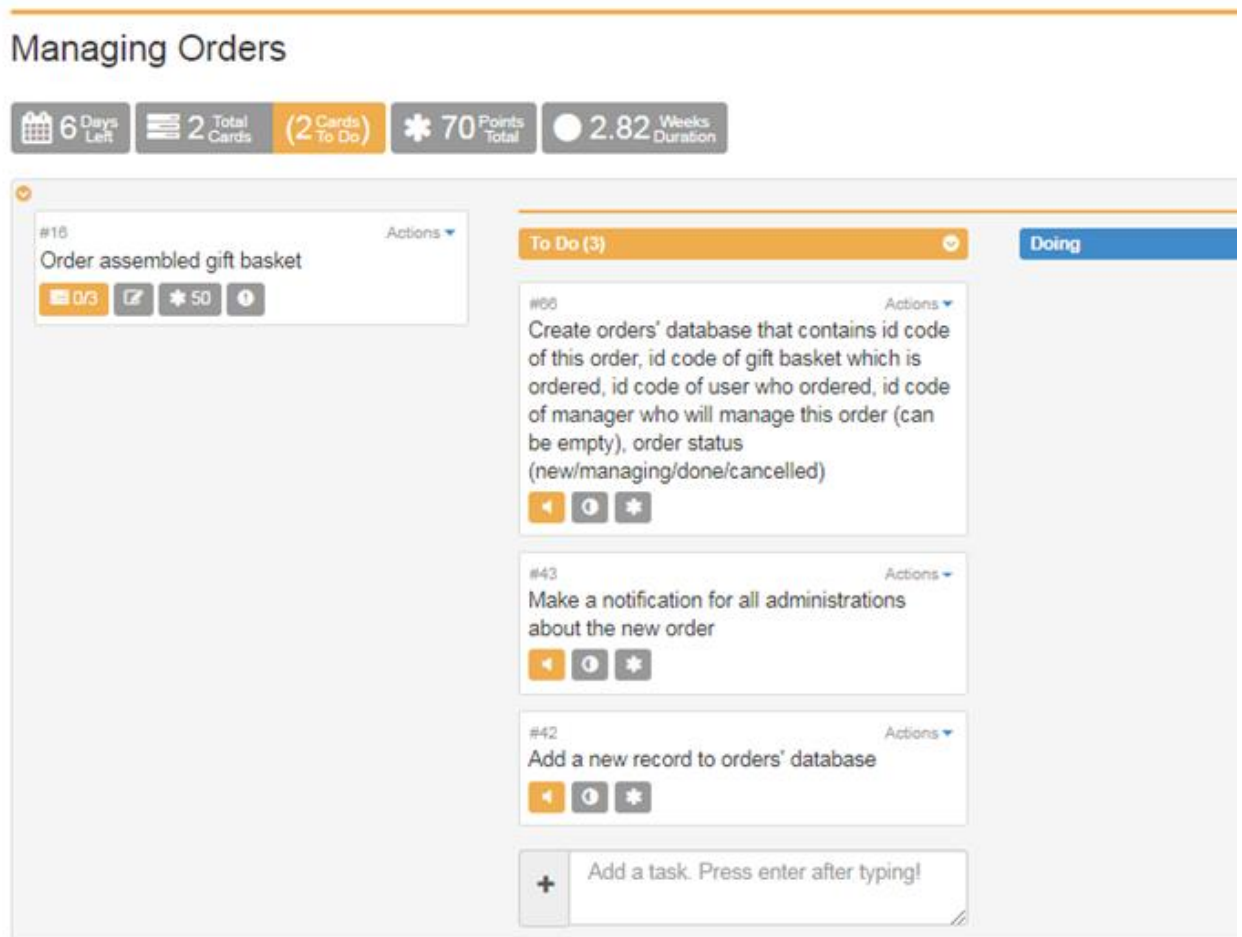


Рисунок 3.18 – Планування функції для менеджменту замовлення створеного користувачем кошика.

3.2. Побудова Use Case діаграм

Для опису поведінки та переходів між інтерфейсами сайту системи було запропоновано наступну Use Case діаграму, яка зображена на рисунках 3.19 (для ролі користувача). Зв'язок асоціації (association), позначений суцільною стрілкою означає, що актор - користувач, ініціює відповідний варіант використання сайту. Всі вершини діаграми - інтерфейси або процеси, що надають користувачу певний функціонал, а стрілки - переходи користувача між інтерфейсами та взаємозв'язок між інтерфейсами.

Вкладений перехід від одного інтерфейсу (процесу) до іншого інтерфейсу (процесу) означає, що другий є вкладеним підпроцесом першого (пуста стрілка). Вершина для вкладених переходів є деякою абстракцією, що реалізується через них, як, наприклад, інтерфейс «Main page» є абстракцією для вкладених, оскільки по своїй суті являє собою «List of all stored items».

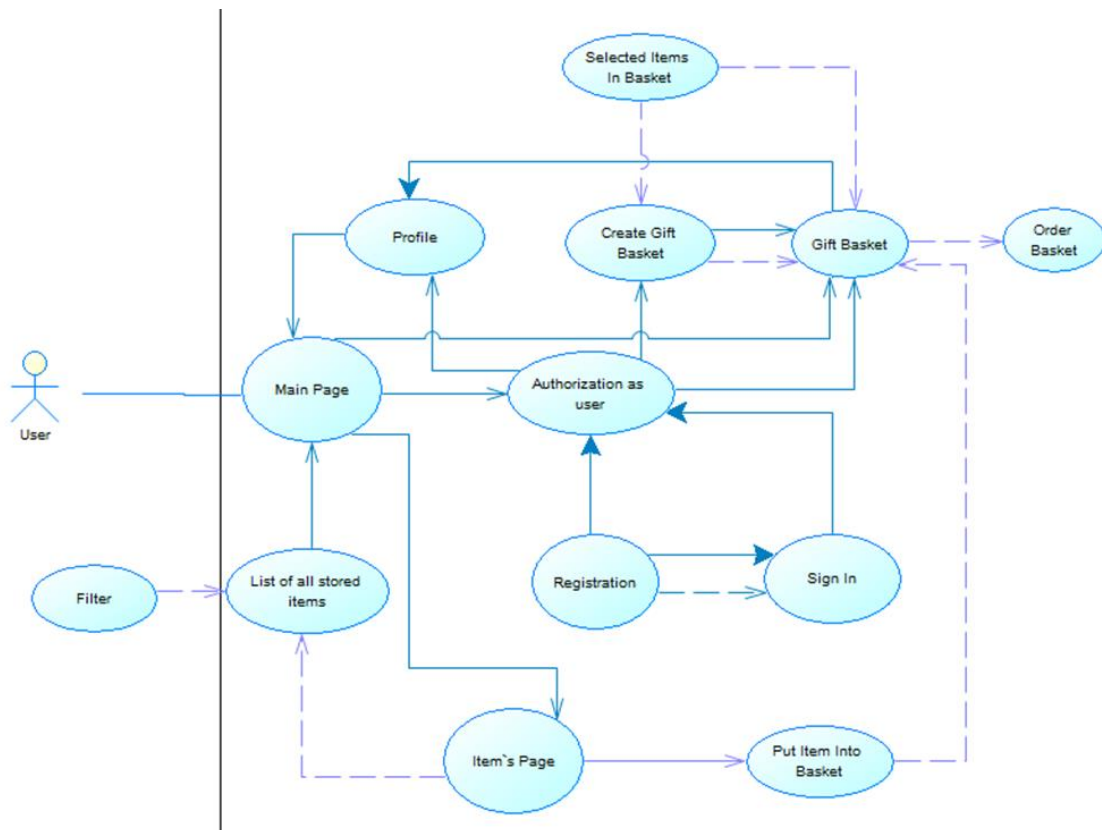


Рисунок 3.19 – Use case діаграма для користувача.

Розширений перехід від одного інтерфейсу (процесу) до іншого інтерфейсу (процесу) означає, що другий інтерфейс розширює функціонал першого, тобто для розширених переходів вершина існує як окремий функціонал (пунктирна стрілка). Наприклад, інтерфейс «Filter» розширюють функціонал інтерфейсу «List of all stored items», до того ж ці інтерфейси не можуть існувати без своєї вершини.

Для опису поведінки та переходів між інтерфейсами сайту адміністратора було запропоновано наступну Use Case діаграму, яка зображена на рисунку 3.20. Відмінністю являється те, що адміністратор для входу у систему має спочатку авторизуватись, а також були додані додаткові процеси: додавання, редагування та видалення товарів.

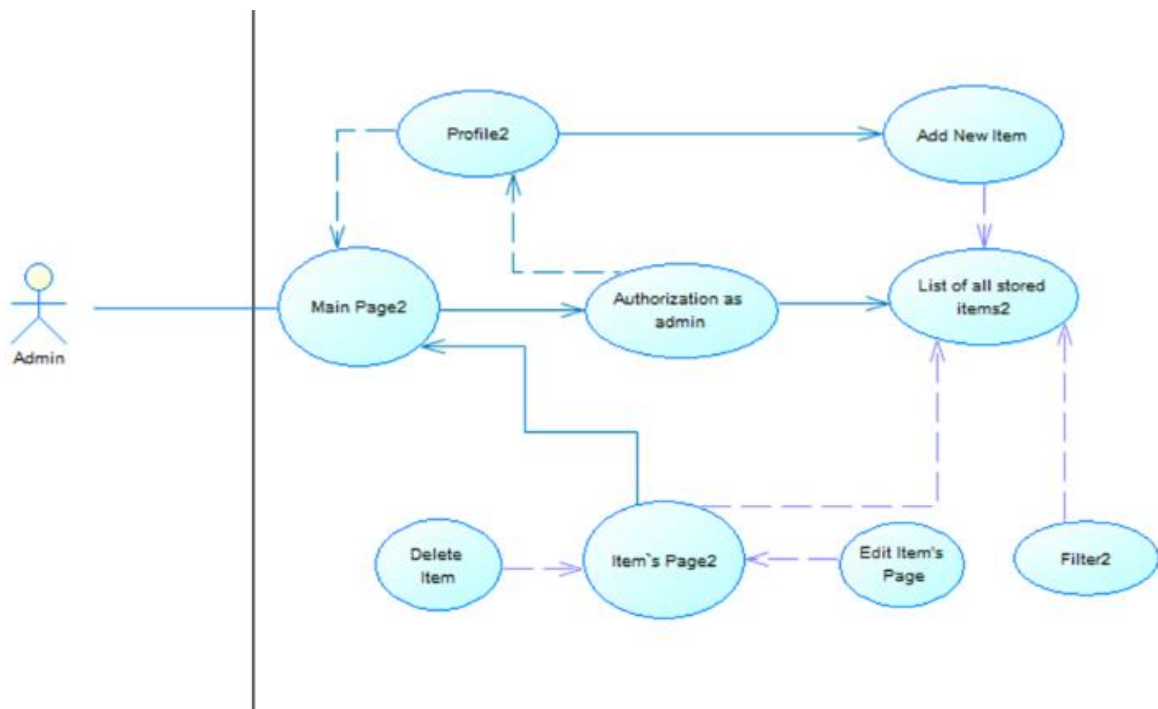


Рисунок 3.20 – Use case діаграма для адміністратора.

3.3 Побудова компонентної діаграми (Component Diagram)

Для даної системи було визначено основні компоненти – блоки функціоналу та модулі: блок авторизації/реєстрації, модуль бази даних, блок покупки та блок товару. Між даними компонентами було описано основні зв'язки, функції та процедури, які регулюють основні процеси системи та які мають забезпечувати роботу. Для розроблюваної системи компонентна діаграма матиме вигляд як зображено на рисунку 3.21.

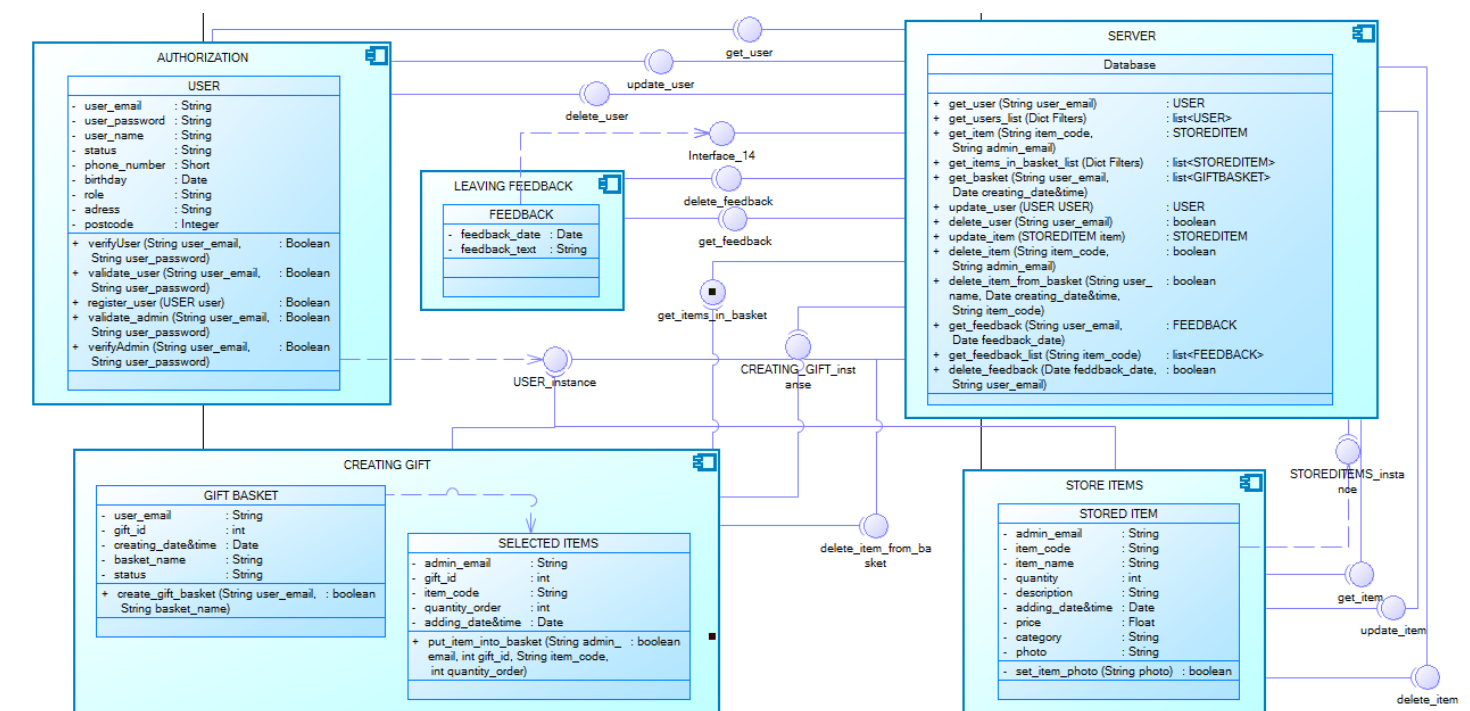


Рисунок 3.21 – Компонентна діаграма розроблюваної системи

4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Всі сутності та зв'язки між ними описані в концептуальній діаграмі (conceptual ERD), зображеній на рисунку 4.1.

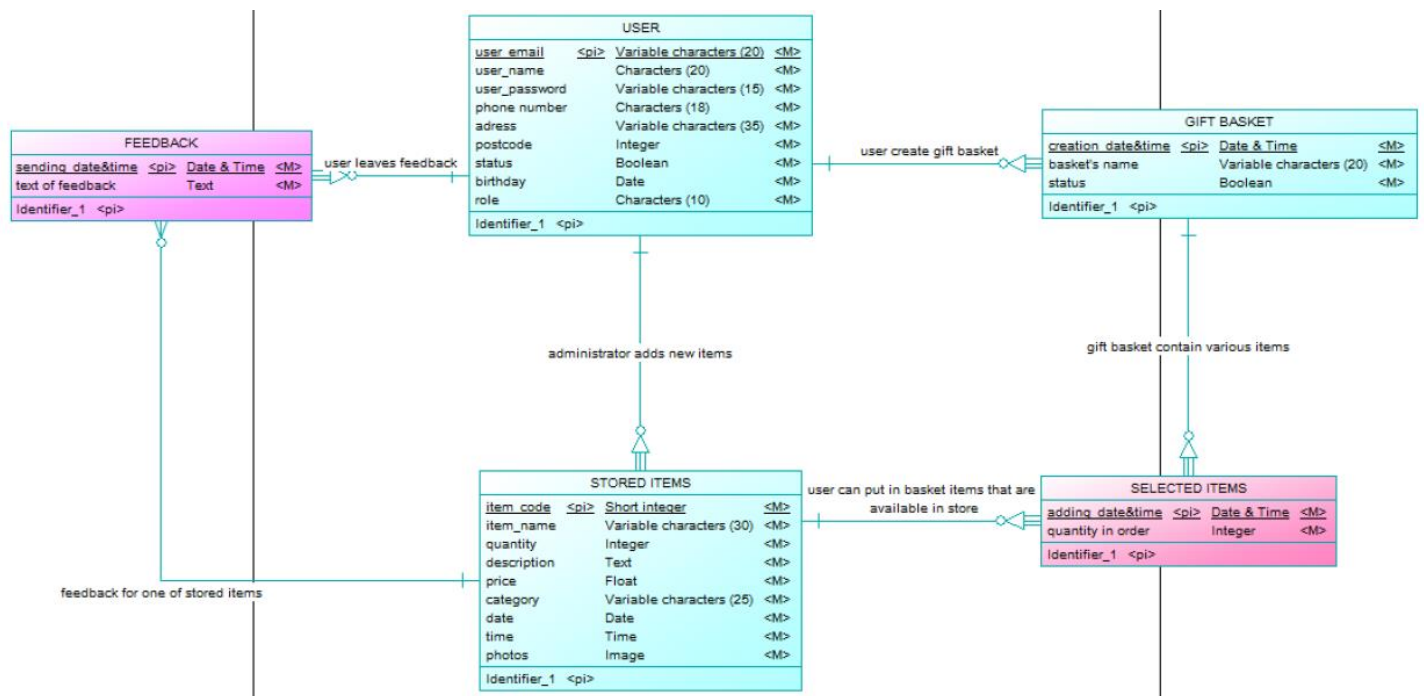


Рисунок 4.1 – Conceptual Data Model

5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

5.1 Логічна модель даних

Логічна модель бази даних являється модифікацією концептуальної діаграми: в ній у кожній сутності з'являються додаткові атрибути, які є зовнішніми ключами інших сутностей, між якими встановлено зв'язок «залежність по ключу». Логічна модель даних зображена на рисунку 5.1.

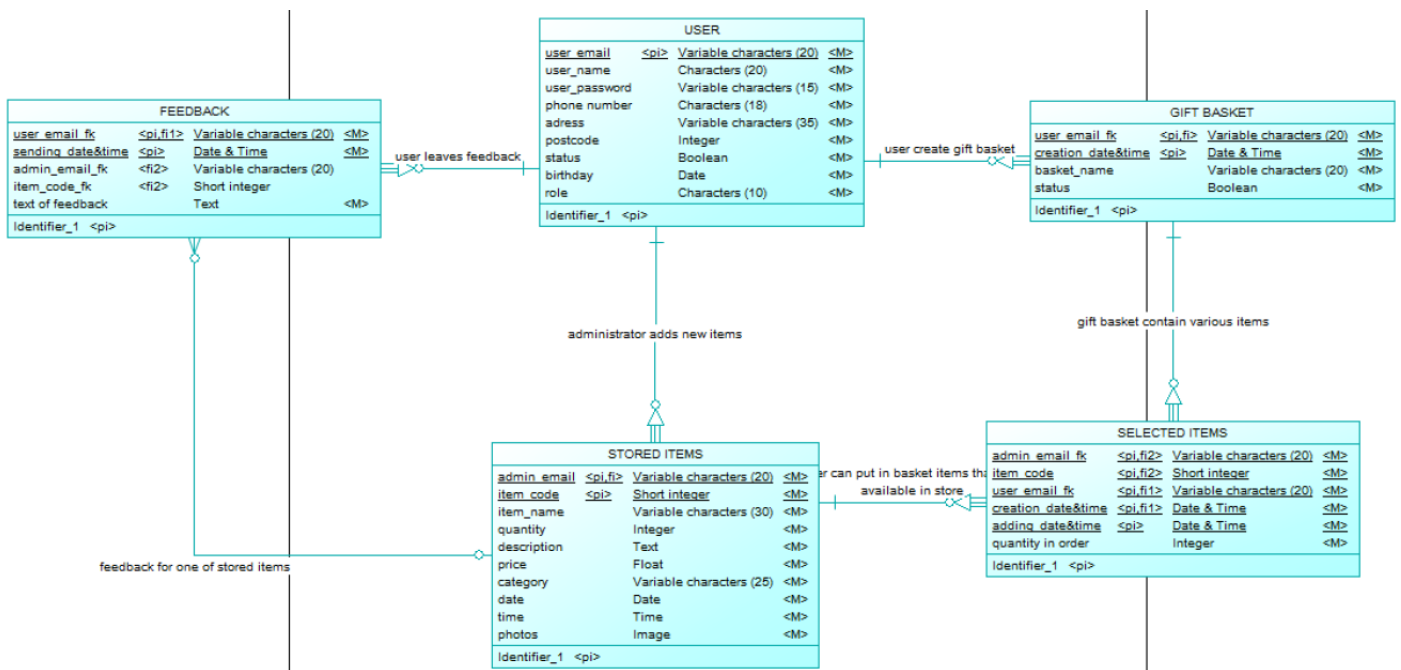


Рисунок 5.1 – Logical Data model

Логічну модуль даних було оптимізовано шляхом введення штучних primary key, щоб уникнути накопичення унікальних ключів у декількох сутностей. Оптимізована логічна модель даної бази даних зображена на рисунку 5.2.

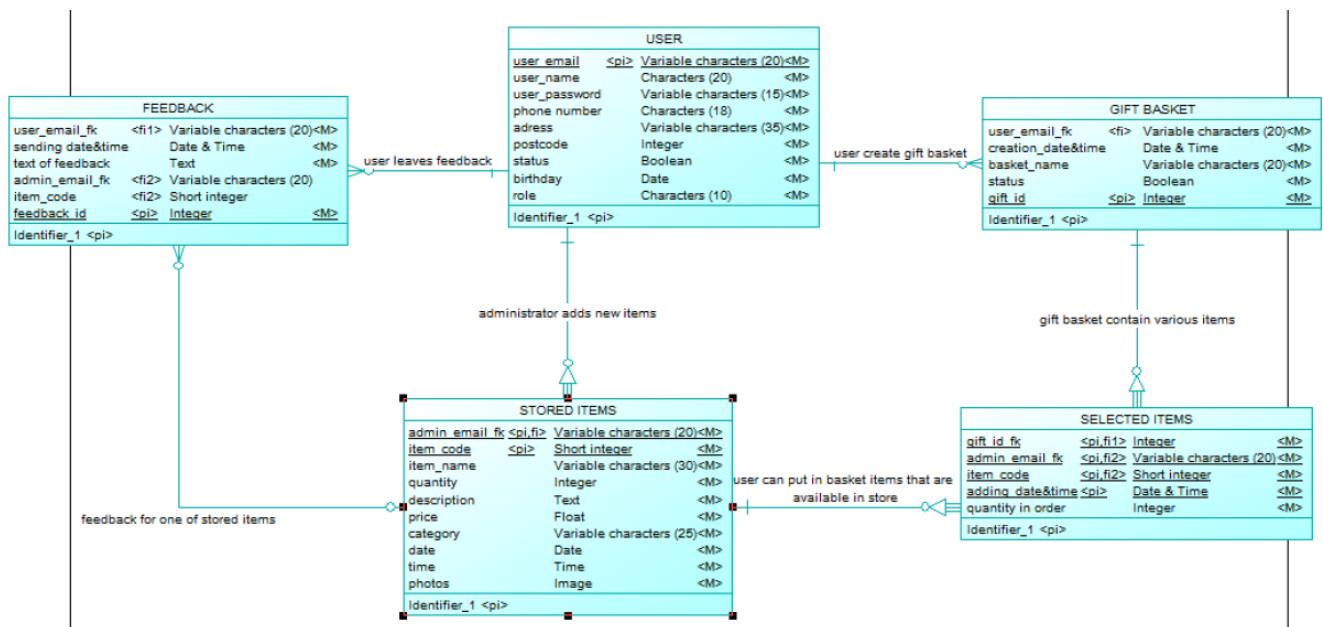


Рисунок 5.2 – Оптимізована Logical Data Model

5.2 Фізична модель бази даних

З логічної моделі даних автоматично було створено фізичну модель даних, для того щоб потім згенерувати кінцевий варіант бази даних у форматі SQL запитів. Фізична модель баз даних зображена на рисунку 5.3.

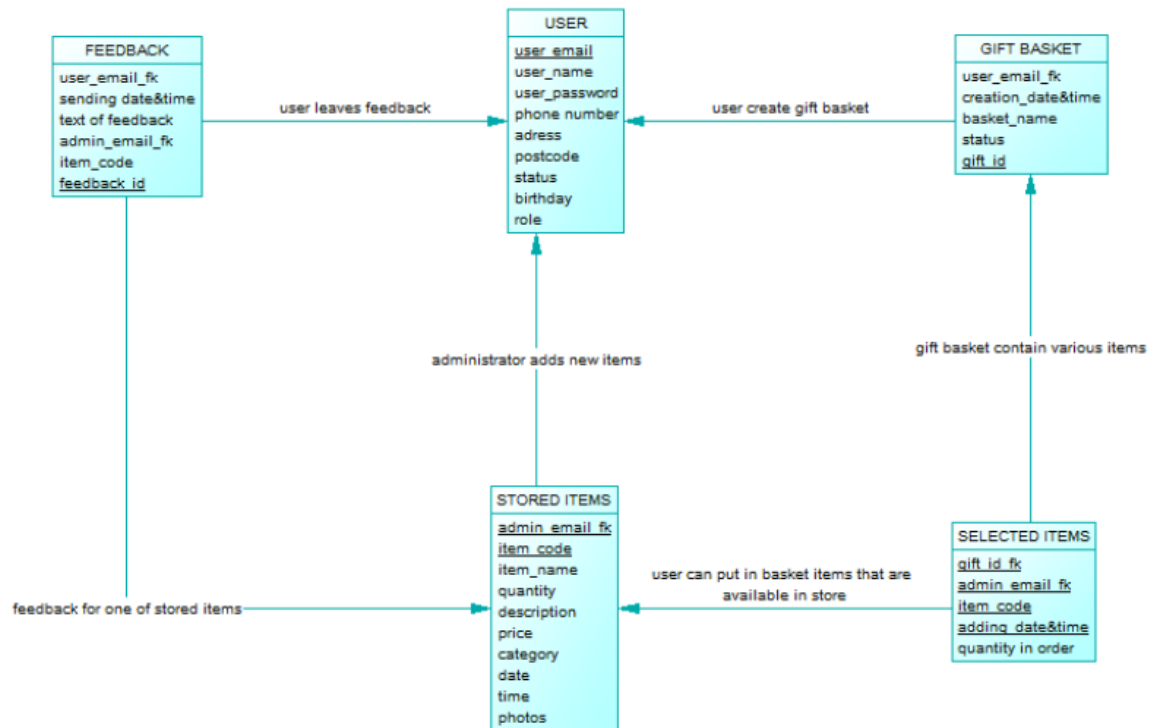


Рисунок 5.3 - Physical Data Model

ВИСНОВКИ

У ході виконання курсової роботи було розроблено інформаційну систему, що має клієнт-серверну архітектуру, призначена для спрощення купівлі та продажу квітів. Проект отримує і зберігає дані у БД Oracle 11g. У БД міститься інформація про зареєстрованих користувачів, інформація про квітки, які продаються, а також можливість робити замовлення. У системі реалізовано дві ролі: користувач та адміністратор. Кожен користувач відповідно до своєї ролі може використовувати певний функціонал. При роботі з системою користувач може обирати квітки та оформлювати замовлення. Перевагами використання розроблюваної системи є те, що вона дозволяє користувачам економити час, який би вони затратили на фізичну купівлю квітів, та дозволяє зберігати всю важливу інформацію в одному місці не переживаючи про її втрату та мати до неї доступ у будь-який час.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Using Python With Oracle Database 11g [Електронний ресурс] : [Веб-сайт]. – <http://www.oracle.com/technetwork/articles/dsl/python-091105.html>
2. Oracle Database Online Documentation 11g [Електронний ресурс] : [Веб-сайт]. – https://docs.oracle.com/cd/E11882_01/nav/portal_4.html
3. Building Oracle Database-backed Web Applications in Django [Електронний ресурс] : [Веб-сайт]. – <http://www.oracle.com/technetwork/articles/dsl/vasiliev-django-100257.html>