МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**

з курсу: «Об’єктно-орієнтоване програмування»

на тему: «Розроблення програмного забезпечення з використанням

об’єктно-орієнтованої парадигми»

Виконав: студент (ка) 2 курсу групи 621п

Спеціальність 121 – Інженерія програмного забезпечення

(код та найменування)

Гусаренко Д. С.

(прізвище й ініціали студента(ки))

Керівник: Лучшев. П. О.

(прізвище й ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_ Лучшев П.О.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_ Дем’яненко В.А.

(підпис) (прізвище та ініціали)

Харків – 2024

**ТИПОВЕ ЗАВДАННЯ НА КУРСОВУ РОБОТУ**

Варіант 9. Замовлення доставки їжі (Glovo)

|  |  |  |
| --- | --- | --- |
| **Критерій оцінювання** | **Бали** | **Штрафи** |
| **Покрокове виконання курсової роботи:** | | |
| Розділ 1(крайній термін – **3 тиждень**) | 0..15 | -3 |
| Розділ 2(крайній термін – **6 тиждень**) | 0..15 | -3 |
| Розділ 3(крайній термін – **9 тиждень**) | 0..15 | -3 |
| Розділ 4(крайній термін – **12 тиждень**) | 0..15 | -3 |
| Розділ 5(крайній термін – **13 тиждень**) | 0..5 | -1 |
| Оформлення пояснювальної записки (термін – **14 тиждень**) | 0..3 | -1 |
| **Підготовка до захисту курсової роботи:** | | |
| доповідь з презентацією (термін – **15 тиждень**) | 0..1 |  |
| відеоролик з демонстрацією роботи програми  (термін – **15 тиждень**) | 0..1 |  |
| **Захист курсової роботи:** | | |
| доповідь з презентацією | 0..5 |  |
| демонстрація роботи програми і github-репозиторію | 0..5 |  |
| відповіді на питання | 0..5 |  |
| використання\*:   1. інкапсуляції 2. статичних членів класів 3. інтерфейсів 4. абстрактних класів 5. спадкування 6. поліморфізму 7. .NET-delegates і events 8. сереалізації/десереалізації об’єктів 9. Collections.Generic 10. LINQ   \* мають бути представленні на захисті курсової роботи у презентації | 0..15 |  |
| **Всього за курсову роботу:** | **0..100** | **0..14** |

**Зміст**

[СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ 2](#_Toc157882173)

[ВСТУП 2](#_Toc157882174)

[1 ФУНКЦІОНАЛЬНІ ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 2](#_Toc157882175)

[1.1 Аналіз функціональності програмних аналогів 2](#_Toc157882176)

[1.2 Побудова Mind Map для заданої предметної області 2](#_Toc157882177)

[1.3 Виділення ролей користувачів і формулювання функціональних вимог 2](#_Toc157882178)

[1.4 Проєктування інтерфейсу користувача 2](#_Toc157882179)

[1.5 Розроблення функціональних тестів 2](#_Toc157882180)

[1.5.1 Функціональні тести для ролі користувача «Адміністратор» 2](#_Toc157882181)

[1.5.2 Функціональні тести для ролі користувача «Зареєстрований користувач» 2](#_Toc157882182)

[1.5.3 Функціональні тести для ролі користувача «Гість» 2](#_Toc157882183)

[2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ 2](#_Toc157882184)

[2.1 Виділення і опис класів предметної області 2](#_Toc157882185)

[2.2 Встановлення зв’язків між класами 2](#_Toc157882186)

[3 ПРОГРАМНА РЕАЛІЗАЦІЯ КЛАСІВ ПРЕДМЕТНОЇ ОБЛАСТІ І ЇХ ТЕСТУВАННЯ 2](#_Toc157882187)

[3.1 Структура проєкту з реалізацією класів предметної області 2](#_Toc157882188)

[3.2 Реалізація інтерфейсів і каркасів класів предметної області 2](#_Toc157882189)

[3.3 Розроблення unit-тестів для класів предметної області 2](#_Toc157882190)

[3.4 Повна реалізація класів предметної області 2](#_Toc157882191)

[3.5 Результати unit-тестування класів предметної області 2](#_Toc157882192)

[4 ПРОЄКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ КЛАСІВ ІНТЕРФЕЙСУ КОРИСТУВАЧА 2](#_Toc157882193)

[4.1 Структура проєкту з реалізацією класів інтерфейсу користувача 2](#_Toc157882194)

[4.2 Виділення класів для реалізації інтерфейсу користувача 2](#_Toc157882195)

[4.3 Програмна реалізація класів інтерфейсу користувача 2](#_Toc157882196)

[5 ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 2](#_Toc157882197)

[5.1 Функціональне тестування для ролі користувача «Адміністратор» 2](#_Toc157882198)

[5.2 Функціональне тестування для ролі користувача «Зареєстрований користувач» 2](#_Toc157882199)

[5.3 Функціональне тестування для ролі користувача «Гість» 2](#_Toc157882200)

[ВИСНОВКИ 2](#_Toc157882201)

[ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ 2](#_Toc157882202)

[ДОДАТОК А. Лістинг класів предметної області 2](#_Toc157882203)

[ДОДАТОК Б. Лістинг класів інтерфейсу користувача 2](#_Toc157882204)

[ДОДАТОК В. Лістинг класів unit-тестів 2](#_Toc157882205)

[ДОДАТОК Г. Назва додатку 2](#_Toc157882206)

# СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

|  |  |  |
| --- | --- | --- |
| CRUD | – | (*англ*. Create Read Update Delete) основні функції управління даними «створення, читання, оновлення і вилучення». |
| ООП | – | об’єктно-орієнтоване програмування |
| … |  | … |

# ВСТУП

Чому тема є актуальною?

Замовлення доставки їжі, як тема, актуальна через зручність обслуговування для людей, які знаходяться поза домівкою чи працюють тривалий час. Обширне меню, безпечність в умовах війни/пандемії, підтримка місцевих бізнесів – переваги в цій галузі сфери обслуговування.

Хто є потенційним користувачем?

Потенційними користувачами сервісів замовлення доставки їжі є:

* люди, у яких обмаль часу на приготування страв,
* мандрівники,
* студенти,
* мешканці великих міст.

Навести лідерів серед програмних аналогів.

В Україні лідерами серед програмних аналогів є такі додатки як:

* Glovo,
* Rocket (до 2022 через банкротство та припинення існування),
* Bolt Food,
* додатки інших ресторанів (McDonalds, PizzaDay тощо).

Зазначити стек-технологій, який буде використаний.

Не забуваємо наводити правильно оформлені посилання. Для ручного оформлення посилань звернутися до [1], для автоматичного – до [2] із «Перелік джерел і посилань».

Вступ має займати не більше 1 стор.

# ФУНКЦІОНАЛЬНІ ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Аналіз функціональності програмних аналогів

Проаналізуємо функціонал аналогічних програмних систем:

* **Bolt Food** (посилання на Google Play **[3]**) (рисунок 1.1 – 1.4):

|  |  |
| --- | --- |
| Рисунок 1.1 | Рисунок 1.2 |
| Рисунок 1.3 | Рисунок 1.4 |

Сервіс **Bolt Food** дозволяє зробити замовлення із запропонованих закладів харчування (ресторани, фаст-фуд тощо.), магазинів (АТБ, Сільпо, ЕкоМаркет) та обрати адресу доставки (рис. 1.1.). Присутній пошук за категорією для фільтрації результатів (рис 1.2.). Також є можливість передивитись актуальні та раніше сформовані замовлення в окремому пункті меню (рис. 1.3.). Присутні інші налаштування, а саме: спосіб оплати, промокоди, профіль, параметри, про нас та допомога (рис. 1.4.).

* **Glovo** (посилання на Google Play **[4]**) (рисунок 1.5 – 1.8)

|  |  |
| --- | --- |
| Рисунок 1.5 | Рисунок 1.6 |
| Рисунок 1.7 | Рисунок 1.8 |

Сервіс **Glovo** дозволяє зробити замовлення із запропонованих закладів харчування (ресторани, фаст-фуд тощо.), магазинів (АТБ, Сільпо, ЕкоМаркет), здоров’я та краси (АНЦ, АОЦ тощо.), послугу кур’єрської доставки (рисунок 1.7). Присутній зручний пошук адреси для замовлення у вигляді мапи з маркером (рисунок 1.6). Присутні інші налаштування, а саме: замовлення, спосіб оплати, промокоди, профіль, параметри, про нас та допомога.

**Загальні характеристики:**

* замовлення із запропонованих варіантів;
* обрання адреси доставки;
* відстеження замовлення;
* налаштування профілю, промокоди, тех. підтримка.

## Побудова Mind Map для заданої предметної області

Виходячи з аналізу функціональності програмних аналогів, було розроблено Mind Map мобільного додатку доставки їжі (рисунок 1.9):

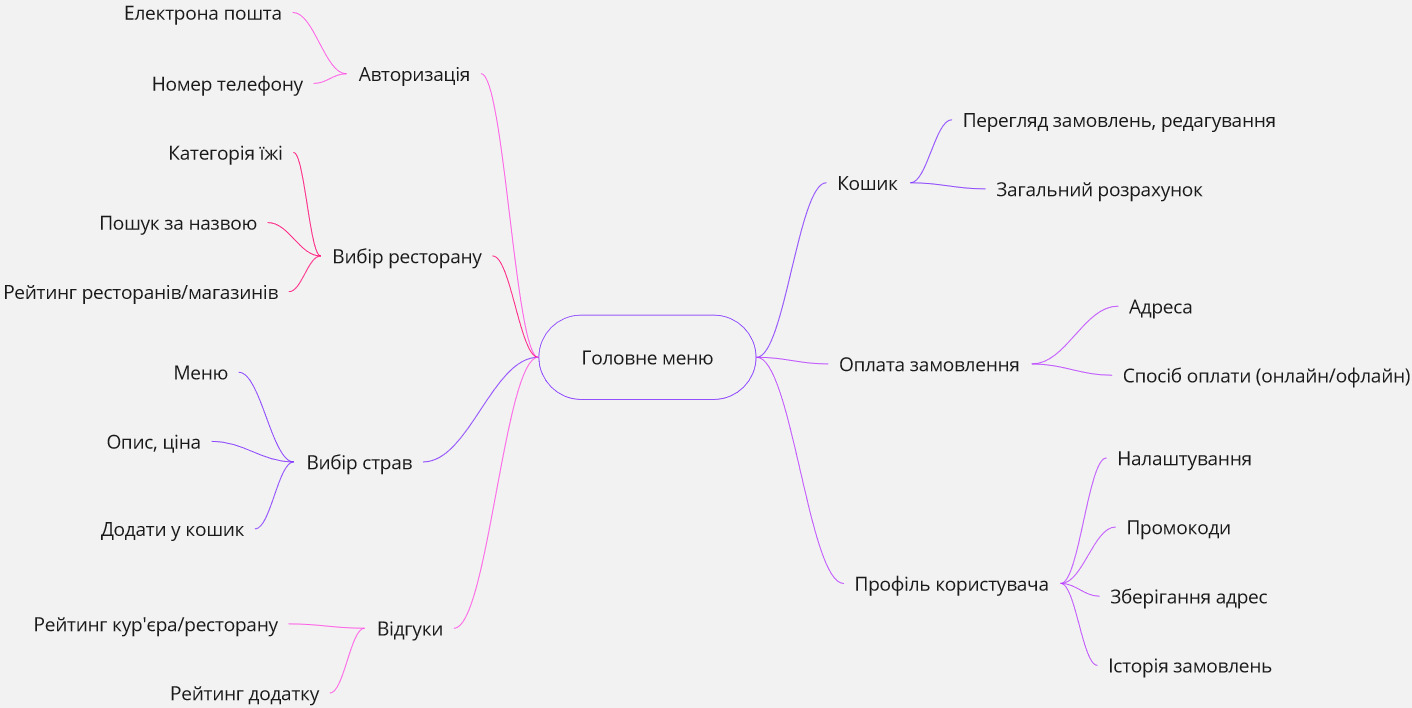


Рисунок 1.9 – Mind Map мобільного додатку доставки їжі

## Виділення ролей користувачів і формулювання функціональних вимог

Були виділені три ролі користувачів: «Адміністратор», «Зареєстрований користувач», «Гість».

Опишемо функціональні вимоги кожного типу користувачів.

Таблиця 1.1 – Функціональні вимоги користувача з роллю «Адміністратор»

|  |  |
| --- | --- |
| **Іденти-фікатор** | **Функціональні вимоги** |
| А.1. | Адміністратор може додавати, редагувати та видалити ресторани та їхні меню.  ***Примітка:***  Є можливість |
| А.2. | Адміністратор може керувати замовленнями, відстежувати стан доставки  ***Примітка:***  Є можливість забезпечити взаємодії з кур'єрами та надання додаткової інформації для клієнтів про статус замовлення. |
| А.3. | Адміністратор може переглядати та аналізувати звіти з продажу та популярності страв, продуктів, ресторанів  ***Примітка:***  Є можливість додати аналітичні інструменти для аналізу продажів, зокрема популярності страв, ресторанів за різними періодами часу. |

Таблиця 1.2 – Функціональні вимоги користувача з роллю «Зареєстрований користувач»

|  |  |
| --- | --- |
| **Іденти-фікатор** | **Функціональні вимоги** |
| ЗК.1. | Зареєстрований користувач може авторизуватися.  ***Примітка:***  не можна авторизуватися з невірним номером телефону або кодом підтвердження (реєстрації). |
| ЗК.2. | Зареєстрований користувач може переглядати меню ресторанів та складати замовлення.  ***Примітка:***  Є можливість залишати рейтинг закладу. |
| ЗК.3. | Зареєстрований користувач може відстежувати стан замовлення та інформацію про доставку  ***Примітка:***  Є можливість сповіщення користувача про зміни у статусі замовлення та точний час доставки. |

Таблиця 1.3 – Функціональні вимоги користувача з роллю «Гість»

|  |  |
| --- | --- |
| **Іденти-фікатор** | **Функціональні вимоги** |
| Г.1. | Гість може авторизуватись у додатку для подальшого використання.  ***Примітка:***  не можна користуватись повноцінним функціоналом додатку без авторизації. |

## Проєктування інтерфейсу користувача

Для реалізації замовником було обрано mobile-застосунок.

На основі сформульованих функціональних вимог були розроблені екранні форми, які наведені на рис. 1.10 – 1.19).

|  |  |
| --- | --- |
| Рисунок 1.10 – Макет інтерфейсу «Авторизація» | Рисунок 1.11 – Макет інтерфейсу «Головне меню» |

|  |  |
| --- | --- |
| Рисунок 1.12 – Макет інтерфейсу «Контекстне меню» | Рисунок 1.13 – Макет інтерфейсу «Огляд асортименту магазину АТБ» |
| Рисунок 1.14 – Макет інтерфейсу «Детальна інформація про продукт» | Рисунок 1.15 – Макет інтерфейсу «Кошик» |
| Рисунок 1.16 – Макет інтерфейсу «Форма замовлення» | Рисунок 1.17 – Макет інтерфейсу «Замовлення оформлено» |
| Рисунок 1.18 – Макет інтерфейсу «Історія замовлень» | Рисунок 1.19 – Макет інтерфейсу «Вибір адреси доставки» |

## Розроблення функціональних тестів

### Функціональні тести для ролі користувача «Адміністратор»

На основі опису предметної області, а також функціональних вимог і екранних форм були розроблені функціональні тести для ролі користувача «Адміністратор» (див. табл. 1.4).

Таблиця 1.4 – Функціональні тести для ролі користувача «Адміністратор»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Іденти-фікатор тесту** | **Послідовність дій користувача** | | **Очікуваний результат** | |
| *А.1. Адміністратор може додавати, редагувати та видалити ресторани та їхні меню.* | | | | |
| А.1.1 | 1. Адміністратор переходить до секції управління закладами.  2. Адміністратор додає новий заклад вводить необхідну інформацію про новий ресторан та його меню.  3. Зберігає зміни і перевіряє, що новий ресторан з'явився в списку. | | Заклад доданий та відображається у списку закладів | |
| А.1.2 | 1. Адміністратор обирає заклад що потрібно відредагувати.  2. Адміністратор зміню інформацію про заклад та оновлює асортимент, при необхідності.  3. Перевіряє, що зміни збережено та коректно відображаються на сторінці закладу. | | Зміни в інформації та асортимент закладу успішно внесені | |
| А.1.3 | 1. Адміністратор обирає опцію видалення закладу зі списку.  2. Підтверджує видалення закладу.  3. Перевіряє, що заклад більше не відображається в списку. | | Успішне видалення закладу | |
| А.1.4 | 1. Адміністратор обирає заклад та переходить до секції "Редагувати асортимент".  2. Додає, редагує та видаляє страви в асортименту закладу.  3. Зберігає зміни та перевіряє, що меню коректно відображається у додатку. | | Зміни відбулись в системі, оновлений асортимент закладу. | |
| *А.2 Адміністратор може керувати замовленнями, відстежувати стан доставки* | | | | |
| А.2.1 | 1. Увійти в обліковий запис адміністратора.  2. Перейти до секції "Замовлення" або аналогічної.  3. Вибрати конкретне замовлення для керування. | | Адміністратор має можливість редагувати та видаляти замовлення, змінювати статус та інформацію доставки. | |
| А.2.2 | Перевірити можливість відстеження стану доставки кожного замовлення в режимі реального часу. | | Адміністратор отримує точну інформацію про доставку. | |
| *А.3. Адміністратор може переглядати та аналізувати звіти з продажу та популярності страв, продуктів, ресторанів* | | | | |
| А.3.1 | | 1. Увійти в обліковий запис адміністратора.  2. Перейти до секції "Аналітика" або аналогічної. | | Адміністратор повинен мати доступ до звітів про продажі. |
| А.3.2 | | Обрати конкретний період часу або параметри для аналізу. | | Відображення графіків, діаграм і таблиць динаміки продажів та популярності. |
| А.3.3 | | Перевірити можливість сортування та фільтрації результатів. | | Адміністратор впорядковує та обирає дані для аналізу. |

### Функціональні тести для ролі користувача «Зареєстрований користувач»

На основі опису предметної області, а також функціональних вимог і екранних форм були розроблені функціональні тести для ролі користувача «Зареєстрований користувач» (див. табл. 1.5).

Таблиця 1.5 – Функціональні тести для ролі користувача «Зареєстрований користувач»

|  |  |  |
| --- | --- | --- |
| **Іденти-фікатор**  **тесту** | **Послідовність дій користувача** | **Очікуваний результат** |
| *ЗК.1. Зареєстрований користувач може авторизуватися.* | | |
| ЗК.1.1 | Відкрити додаток до вікна Авторизації.  Ввести зареєстрований номер телефону та код підтвердження. | Авторизація через введення даних. |
| ЗК.1.2 | Спробувати авторизуватися з невірним номером телефону або кодом підтвердження (реєстрації). | Блокування спроб авторизації з невірними даними та повідомлення про помилку. |
| *ЗК.2. Зареєстрований користувач може переглядати меню ресторанів та складати замовлення.* | | |
| ЗК.2.1 | 1. Увійти в обліковий запис зареєстрованого користувача в мобільному додатку.  2. Перейти до головного меню | Система повинна показати користувачеві головне меню додатку. |
| ЗК.2.2 | 1. Вибрати певні страви/продукти та додати їх до кошика для оформлення замовлення.  2. Здійснити оформлення замовлення, вказавши необхідні дані для доставки. | Ефективне складання замовлень, вказання необхідної інформації. |
| ЗК.2.3 | Залишити рейтинг для закладу після завершення замовлення. | Система з рейтингом та коментарями для ресторану. |

|  |  |  |
| --- | --- | --- |
| *ЗК.3. Зареєстрований користувач може переглядати меню ресторанів та складати замовлення.* | | |
| ЗК.3.1 | 1. Увійти в обліковий запис зареєстрованого користувача.  2. Перейти до розділу "Мої замовлення" або аналогічного. | Система має оновлювати користувача про зміни у замовленні. |
| ЗК.3.2 | Перевірити точний час доставки та інші деталі замовлення. | Система має показати правильну і актуальну інформацію. |
| ЗК.3.3 | Перевірити можливість отримання сповіщень про будь-які зміни у статусі замовлення та точний час доставки. | Система дозволяє користувачам вибирати та отримувати сповіщення про замовлення. |

### Функціональні тести для ролі користувача «Гість»

На основі опису предметної області, а також функціональних вимог і екранних форм були розроблені функціональні тести для ролі користувача «Гість» (див. табл. 1.6).

Таблиця 1.6 – Функціональні тести для ролі користувача «Гість»

|  |  |  |
| --- | --- | --- |
| **Іденти-фікатор**  **тесту** | **Послідовність дій користувача** | **Очікуваний результат** |
| *Г.1. Гість може подивитися інформацію щодо наявних сеансів на обрану дату.* | | |
| Г.1.1 | 1. Відкрити додаток.  2. Спробувати використовувати додаток без авторизації. | Система має попередити гостя про авторизацію. |

# МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

## Виділення і опис класів предметної області

**Клас «User» – користувач**

Таблиця 2.1 – Виділення характеристик класу «User»

|  |  |  |
| --- | --- | --- |
| **Назва характеристики** | **Тип значення** | **Обмеження** |
| Електронна пошта | Рядок | Не пустий рядок, наявність символу ‘@’ |
| Ім’я | Рядок | Не пустий рядок, довжина: хоча б 1 символ |
| Прізвище | Рядок | Не пустий рядок, довжина: хоча б 1 символ |
| Дата народження | Дата | > 01.01.1900 |
| Телефон | Рядок | Не пустий рядок, відповідність формату:  +38(0XX)-XXXXXXX |

Таблиця 2.2 – Виділення поведінки класу «User»

|  |  |  |  |
| --- | --- | --- | --- |
| **Назва поведінки** | **Опис вхідних параметрів** | **Тип значення, що повертається** | **Бізнес-правила** |
| Оформити купівлю білетів | Фільм,  дата,  час,  кількість білетів | true/false – замовлення оформлено / помилка при оформленні замовлення;  номер замовлення  (у випадку true) | 1. Перевірити чи дійсні вхідні дані (фільм, дата, час). 2. Перевірити наявність необхідної кількості білетів. 3. Якщо пройдені перевірки, то оформити замовлення. |
| Повернути куплені білети | Номер замовлення | true/false – повернення оформлено / помилка при оформленні повернення. | 1. Перевірити чи дійсний номер замовлення. 2. Перевірити чи не відбувся вже сеанс. 3. Якщо до сеансу залишилося більше 24 годин, то повернути повну суму грошей, інакше – 50%. |
| … | … | … | … |

**Клас «Movie» – фільм**

Таблиця 2.3 – Виділення характеристик класу «Movie»

|  |  |  |
| --- | --- | --- |
| **Назва характеристики** | **Тип значення** | **Обмеження** |
| Назва фільма | Рядок | Не пустий рядок, довжина: хоча б 1 символ |
| Тривалість | Ціле число | >0 |
| Режисер | Рядок | Не пустий рядок, довжина: хоча б 1 символ |
| Актори | Рядок | Не пустий рядок, довжина: хоча б 1 символ |
| Мова | Рядок | Не пустий рядок, довжина: хоча б 1 символ |
| Субтітри | Булевське значення |  |
| Дата початку прокату | Дата | Не може бути меншою ніж поточна дата |
| Дата початку прокату | Дата | Має бути більшою за дату початку прокату |

Таблиця 2.4 – Виділення поведінки класу «Movie»

|  |  |  |  |
| --- | --- | --- | --- |
| **Назва поведінки** | **Опис вхідних параметрів** | **Тип значення, що повертається** | **Примітка** |
| … | … | … | … |
| … | … | … | … |
| … | … | … | … |

Описати всі виділені класи предметної області.

## Встановлення зв’язків між класами

На рисунку 0.3 наведена діаграма класів, яка була побудова з використанням сервісу <https://www.lucidchart.com>.

Рисунок 2.1 – Діаграма класів …

Описати всі зв’язки між класами.

Приділити увагу абстрактним класам.

Зробити акцент на власних і .NET-інтерфейсах.

# ПРОГРАМНА РЕАЛІЗАЦІЯ КЛАСІВ ПРЕДМЕТНОЇ ОБЛАСТІ І ЇХ ТЕСТУВАННЯ

## Структура проєкту з реалізацією класів предметної області

Навести файлову/модульну структуру проєкту, яка містить реалізацію класів предметної області.

Рисунок 3.1 – Структура проєкту з реалізацією предметної області

## Реалізація інтерфейсів і каркасів класів предметної області

**Інтерфейс «….» – користувач**

Навести програмний код інтерфейсу.

**Клас «…» – зареєстрований користувач**

Навести програмний код каркасу класу.

**Клас «…» – адміністратор**

Навести програмний код каркасу класу.

## Розроблення unit-тестів для класів предметної області

Розроблені unit-тести наведені у Додатку \_\_\_.

Таблиця 3.1 – Покриття unit-тестами класу «User»

|  |  |
| --- | --- |
| **Назва методу** | **Кількість розроблених unit-тестів** |
|  |  |
|  |  |
|  |  |
|  |  |

Таблиця 3.2 – Покриття unit-тестами класу «Movie»

|  |  |
| --- | --- |
| **Назва методу** | **Кількість розроблених unit-тестів** |
|  |  |
|  |  |
|  |  |
|  |  |

…

## Повна реалізація класів предметної області

У Додатку \_\_\_ наведено повну реалізацію програмного коду розроблених класів предметної області.

## Результати unit-тестування класів предметної області

На рисунках 0.5 – 0.7 наведено скриншоти проходження розроблених unit-тестів для класів предметної області.

…

Рисунок 3.2 – Скриншот проходження unit-тестів для класу \_\_\_\_\_\_\_

…

Рисунок 3.3 – Скриншот проходження unit-тестів для класу \_\_\_\_\_\_\_

…

Рисунок 3.4 – Скриншот проходження unit-тестів для класу \_\_\_\_\_\_\_

# ПРОЄКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ КЛАСІВ ІНТЕРФЕЙСУ КОРИСТУВАЧА

## Структура проєкту з реалізацією класів інтерфейсу користувача

Навести файлову/модульну структуру проєкту, яка містить реалізацію класів інтерфейсу користувача.

Рисунок 4.1 – Структура проєкту з інтерфейсом користувача

## Виділення класів для реалізації інтерфейсу користувача

Описати виділені характеристики і методи для класів інтерфейсу користувача.

…

## Програмна реалізація класів інтерфейсу користувача

У Додатку \_\_\_ наведено повну реалізацію програмного коду розроблених класів інтерфейсу користувача.

# ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Функціональне тестування для ролі користувача «Адміністратор»

Для проведення функціонального тестування для ролі «Адміністратор» були використані розроблені у п. 1.4.1. функціональні тести. Результати проведення функціонального тестування наведені нижче.

**Ідентифікатор тесту:** А.1.1.

**Вхідні дані:**

1. Ввести дату сеансу: «20.01.2023».

2. Ввести час сеансу: «10:00».

3. Обрати зал у кінотеатрі: «2».

4. Обрати фільм: «Аватар: шлях води».

5. Натиснути кнопку «Додати».

**Очікуваний результат:** Успішне додавання сеансу

**Отриманий результат:**

Рисунок 5.1 – Скриншот виконання тесту А.1.1

**Ідентифікатор тесту:**

**Вхідні дані:**

**Очікуваний результат:**

**Отриманий результат:**

Рисунок 5.2 – Скриншот виконання тесту \_\_\_\_\_

## Функціональне тестування для ролі користувача «Зареєстрований користувач»

Для проведення функціонального тестування для ролі «Зареєстрований користувач» були використані розроблені у п. 1.4.2. функціональні тести. Результати проведення функціонального тестування наведені нижче.

**Ідентифікатор тесту:**

**Вхідні дані:**

**Очікуваний результат:**

**Отриманий результат:**

Рисунок 5.3 – Скриншот виконання тесту \_\_\_\_\_

## Функціональне тестування для ролі користувача «Гість»

Для проведення функціонального тестування для ролі «Гість» були використані розроблені у п. 1.4.3. функціональні тести. Результати проведення функціонального тестування наведені нижче.

**Ідентифікатор тесту:**

**Вхідні дані:**

**Очікуваний результат:**

**Отриманий результат:**

Рисунок 5.4 – Скриншот виконання тесту \_\_\_\_\_

# ВИСНОВКИ

У результаті виконання курсової роботи було розроблено програмне забезпечення з використанням об’єктно-орієнтованої парадигми для вказати предметну область.

В ході роботи було…. (коротко описати що саме було зроблено під час виконання курсової роботи).

…

….

**Примітка:**

Посилання на github-репозиторій: <https://github.com/husarenko/coursework>

Посилання на відеоролик: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ

1. Воробйов, Ю. А. Правила оформлення навчальних і науково-дослідних документів [Текст] : навч. посіб. / Ю. А. Воробйов, Ю. О. Сисоєв. – 4-те вид., випр. і доп. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2019. – 88 с.
2. Автоматичне оформлення джерел по ВАК України [Електронний ресурс] – Режим доступу до ресурсу: <http://vak.in.ua/do.php>.
3. Bolt Food - Google Play [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.bolt.deliveryclient&hl=ru.
4. Glovo - Google Play [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.glovo&hl=uk&gl=US.

# ДОДАТОК А. Лістинг класів предметної області

Власноруч створений код програми.

# ДОДАТОК Б. Лістинг класів інтерфейсу користувача

Власноруч створений код програми. Код, який було автоматично створено середовищем розробки, додавати не потрібно.

# ДОДАТОК В. Лістинг класів unit-тестів

Власноруч створений код програми.

# ДОДАТОК Г. Назва додатку

…