МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ

ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Циклова комісія програмної інженерії

**КУРСОВИЙ ПРОЕКТ**

з навчальної дисципліни

"ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ"

на тему: ««Реєстратура. Талон» замовлення талонів до лікарів»

(вказати тему курсового проекту)

Студента IV курсу ПЗ-19-1 групи

напряму підготовки 6.050103

Програмна Інженерія

спеціальності 121 Інженерія

програмного забезпечення

Вакуленко О.Д.

|  |
| --- |
| (прізвище та ініціали студента) |

Керівник викладач Гапоненко Н.В.

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

Кількість балів: \_\_\_\_\_\_ Оцінка ECTS:\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. ДНІПРО

2022 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ

ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Предметна комісія програмної інженерії

|  |  |
| --- | --- |
|  | **ЗАТВЕРДЖУЮ**  Голова предметної комісії ПІ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.С.Ланська  "\_\_"\_\_\_\_\_ 2022 р. |

**ЗАВДАННЯ**

**на виконання курсового проєкту**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| з дисципліни | | | Об’єктно-орієнтоване програмування | | | | | | | |
| студенту |  | | | | | | | | | |
|  | (прізвище, ім'я та по батькові) | | | | | | | | | |
| Відділення | | Комп'ютерної та програмної інженерії | | | | | | | | |
| Спеціальність | | | | 121 Інженерія програмного забезпечення | | | | | | |
| Курс | IV | | | | | |  | Група (шифр) | | ПЗ-19-1 |
| 1 Тема проекту | | | | | Програмний модуль «Фоторепортер» обліку замовлень фотосесій | | | | | |
|  | | | | | | | | | | |
| 2 Початкові дані: | | | | | |  | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
| Розглянуто і ухвалено на засіданні циклової комісії | | | | | | | | | Програмної інженерії | |
| Протокол № 2 від 16.09.2020 р. | | | | | | | | | | |

Керівник КП \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Н.В.Гапоненко\_\_\_

(підпис) (ініціали та прізвище)

Завдання до виконання

одержав студент \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (ініціали та прізвище)

Дата видачі

Термін виконання

Зміст

[Вступ 4](#_Toc863371350)

[2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ 6](#_Toc135276671)

[2.1 Опис середовища програмування 6](#_Toc1557310971)

[2.2 Опис мови програмування 7](#_Toc194799961)

[2.3 Опис СКБД 8](#_Toc1237866499)

[2.4 Опис основних принципів ООП 9](#_Toc676557)

[3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ 11](#_Toc633475264)

[3.1 Опис бази даних 11](#_Toc1742323869)

[3.2 Проектування користувацького інтерфейсу 13](#_Toc150255791)

[3.3 Контроль вхідних даних програми 17](#_Toc2055794112)

[Висновок 18](#_Toc867243801)

# Вступ

У сучасному світі повсюдно необхідна автоматизація належного зберігання та обліку цих даних. Майже кожна компанія має власне програмне забезпечення, яке допомагає співробітникам виконувати свою роботу.

Студії мають багато програмного забезпечення, але ці програми важко освоїти та використовувати. Велика частина цих програм працює в Інтернеті, що не завжди зручно.

Програма має бути простою у використанні, швидкою, легкою для вивчення та придатною для багатьох малих і великих підприємств. Користуватися програмою можуть навіть недосвідчені користувачі ПК. Його може використовувати як одна людина (наприклад, керівник невеликої компанії), так і група людей (наприклад, у великій компанії робота розбита на кілька посад).

1 ПОСТАНОВКА ЗАДАЧІ

Необхідно розробити програмний продукт який дозволить заказати послуги різного напрямку. Забронювати послугу на конкретний час а отримати приблизний термін виконання.

Наприкінці отримати відповідну форму з інформацією про послугу термін, час виконання та ціну.

# 2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

## 2.1 Опис середовища програмування

Qt — це кросплатформний набір для розробки програмного забезпечення для мови програмування C++. Дозволяє запускати програмне забезпечення, написане на ньому, у більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної ОС, не змінюючи вихідний код. Містить усі базові класи, які можуть знадобитися для розробки прикладного програмного забезпечення, від елементів графічного інтерфейсу до мереж, баз даних, класів OpenGL, SVG і XML. Бібліотека забезпечує потоковий, мережевий і міжплатформний доступ до файлів. Qt також доступний для багатьох інших мов програмування: Ada (QtAda), C# (Qyoto/Kimono), Java (Qt Jambi), Qt Jambi, Node.js, Pascal, Perl, PHP (PHP-Qt), Ruby (QtRuby) і Python (PyQt, PySide).

Qt 5 був випущений у грудні 2012 року з його модульною структурою та перемістіть увагу на написання програм за допомогою інструментів декларативний опис інтерфейсу, який визначає логіку взаємодії з користувачем н мові JavaScript, тоді як програми C++ позиціонуються для реалізації критичні для виконання або надто складні частини програми, а також с творіть новий модульний сервер для Qt Quick. Хоча багато основні вдосконалення та зміни, Qt 5 зберігає базову зворотну сумісність із минулим версія, яка повністю підтримує інструменти для створення програм Qt цією мовою C++ і включає майже всі компоненти Qt 4 (давно застаріли елементи), більшість модулів колишнього Qt Mobility та деякі експериментальні модулі Проект від Qt Labs.

З моменту створення в 1996 році комерційна версія бібліотеки Qt потрапила в розряд основа для тисяч успішних проектів по всьому світу.

Крім того, Qt є фундаментом популярного робочого середовища KDE, що входить до складу багатьох дистрибутивів GNU/Linux.

## 2.2 Опис мови програмування

Для реалізації цього проекту були обрані такі мови програмування: - C++ (діалект Qt/MOC і автоматизований QMake) - QML (бібліотека QStyle). C++ — мова програмування високого рівня, яка підтримує кілька парадигм Програмування: об'єктно-орієнтоване, узагальнене та процедурне. розроблений Б'ярне Страуструп з AT&T Bell Labs

При створенні С++ прагнули зберегти сумісність з мовою С. Більшість програм на С справно працюватимуть і з компілятором С++. С++ має синтаксис, заснований на синтаксисі С.

Нововведеннями С++ порівняно з С є:

− підтримка об'єктно-орієнтованого програмування через класи;

− підтримка узагальненого програмування через шаблони;

− доповнення до стандартної бібліотеки;

− додаткові типи даних;

− обробка винятків;

− простори імен;

− вбудовані функції;

− перевантаження операторів;

− перевантаження імен функцій;

− посилання і оператори управління вільно розподіленою пам'яттю.

QML (Qt Meta Language або Qt Modeling Language) - декларативна мова програмування, в основі якої лежить мова JavaScript. QML використовується для розробки додатків, які роблять основний упор на призначений для користувача інтерфейс і, в цілому, на дизайн графічної частини. Є частиною Qt Quick, середовища розробки призначеного для користувача інтерфейсу, поширюваної разом з Qt. Часто використовується для створення додатків, орієнтованих на мобільні пристрої з сенсорним управлінням.

QML-документ являє собою дерево елементів. QML елемент, так само, як і елемент Qt, являє собою сукупність блоків: графічних (таких, як rectangle, image) і поведінкових (таких, як state, transition, animation). Ці елементи можуть бути об'єднані, щоб побудувати комплексні компоненти, починаючи від простих кнопок і повзунків і закінчуючи повноцінними додатками, що працюють з інтернетом. QML елементи можуть бути доповнені стандартними для JavaScript вставками шляхом вбудовування .js файлів. Також вони можуть бути розширені C++ компонентами через Qt framework.

## 2.3 Опис СКБД

MySQL — безкоштовна система керування реляційними базами даних. MySQL розроблена компанією «ЦХ» для підвищення швидкості обробки великих баз даних. Ця система керування базами даних з відкритим кодом (СУБД) була створена як заміна комерційним системам. З самого початку MySQL була дуже схожа на mSQL, але з часом вона розширилася, і зараз MySQL є однією з найпопулярніших систем керування базами даних. Він в основному використовується для створення динамічних веб-сторінок через чудову підтримку кількох мов програмування.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання. MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

## 2.4 Опис основних принципів ООП

Інкапсуляція

Вирішальним фактором у проектуванні компонента програми є приховування внутрішніх даних компонента та деталей його реалізації від інших компонентів програми, а також надання набору методів (API) для взаємодії з ними. Цей принцип є одним із трьох фундаментальних принципів ООП і називається інкапсуляцією.

Наслідування

Наслідування є одним із найважливіших принципів об’єктно-орієнтованого програмування, оскільки воно дозволяє створювати ієрархії об’єктів. Використовуючи успадкування, ви можете створити загальний клас

Поліморфізм

Розглядаючи поліморфізм, слід пам’ятати, що цей принцип нерозривно пов’язаний з іншим принципом об’єктно-орієнтованого програмування – імітацією, який допомагає досягти поліморфізму. Давайте візьмемо приклад абстрактного класу «Автомобіль», за яким слідують два конкретних класи — «Спортивний автомобіль» і «Вантажівка».

# 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Опис бази даних

Для збереження великих об’ємів інформації потрібно розробити низку правил та відношень між цими даними. Відношення між таблицями показані на ER-діаграмі. Зміст таблиць бази даних 3.1 - 3.4

Таблиця 3.1 - Користувач(users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| Users\_ID | Унікальний код | Числовий | Макс. значеня –  2147483647 | Первинний |
| Users\_name | Імʼя лікаря | Символьний | Макс. Кількість символів - 45 |  |
| Users\_surname | Прізвище лікаря | Символьний | Макс. Кількість символів - 45 |  |
| Users\_patronymic | По-батькові | Символьний | Макс. Кількість символів - 45 |  |
| Phone\_number | Номер телефону | Символьний | Макс. Кількість символів - 12 |  |

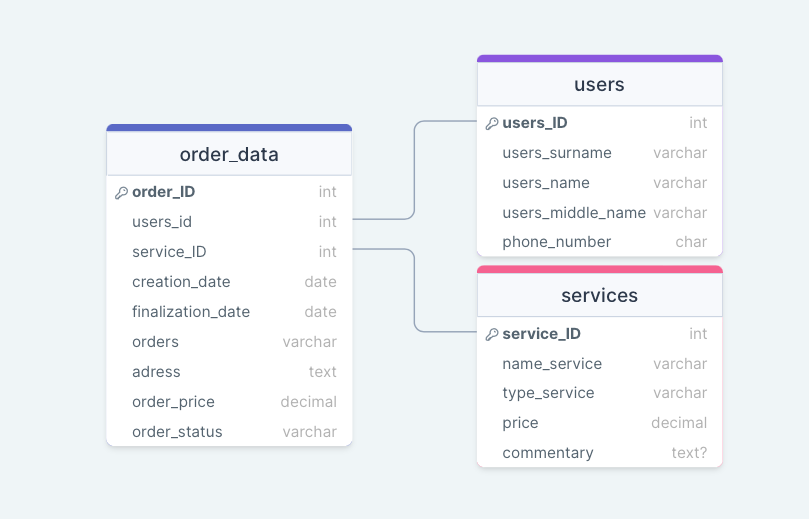
Таблиця 3.2 - Послуги(іservices)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| Service\_ID | Унікальний код | Числовий | Макс. значеня –  2147483647 | Первинний |
| Service\_name | Назва послуги | Символьний | Макс. Кількість символів - 45 |  |
| Service\_type | Тип послуги | Символьний | Макс. Кількість символів - 45 |  |
| Service\_price | Ціна послуги | Числовий | Макс. значеня –  2147483647 |  |
| commentary | Коментар до послуги | Символьний | Макс. Кількість символів - 255 |  |

Таблиця 3.3 - Данні прийому(order\_data)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| Order\_id | Унікальний код | Числовий | Макс. значеня –  2147483647 | Первинний |
| Creation\_date | Дата створення | DATE | Стандарт ISO | Зовнішній |
| Fianlization\_date | Дата прийому | DATE | Стандарт ISO |  |
| img | Посилання на зображення | Символьний | Стандарт ISO |  |
| addres | Адреса | Символьний | Стандарт ISO |  |
| Order\_price | ціна | Числовий |  |  |
| Order\_status | статус | BOOL |  |  |
| Users\_ID | Унікальний код | Числовий | Макс. значеня –  2147483647 | Зовнішній |
| Service\_ID | Унікальний код | Числовий | Макс. значеня –  2147483647 | Зовнішній |

ER-діаграма зв’язку таблиць бази даних приведена нижче.

Рисунок 3.1 - ER-діаграма зв’язку таблиць бази даних

## 3.2 Проектування користувацького інтерфейсу

На початку програми необхідно здійснити підключення до бази даних.

Потім користувача зустрічає головне вікно, в якому можна додати нове замовлення, видалити старе або чи якось змінити обране.

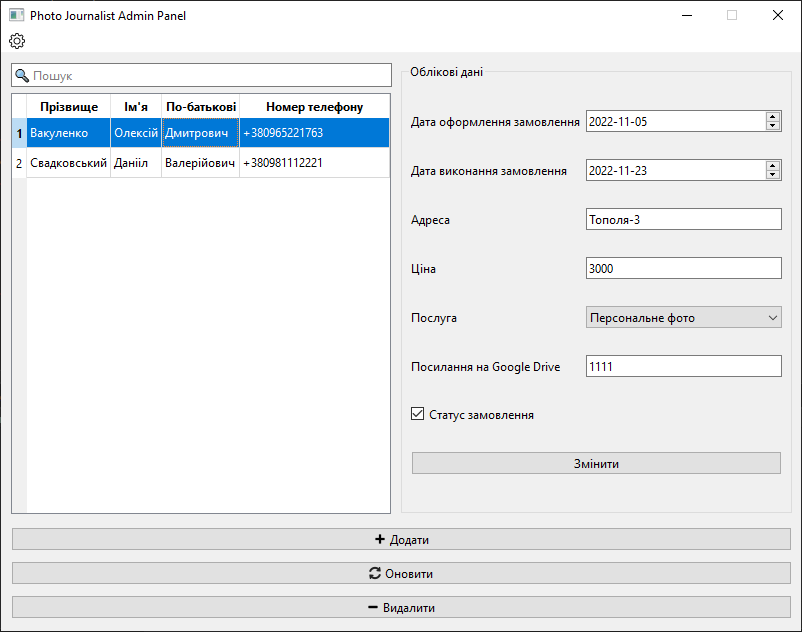


Рисунок 3.2 - Головне вікно програми

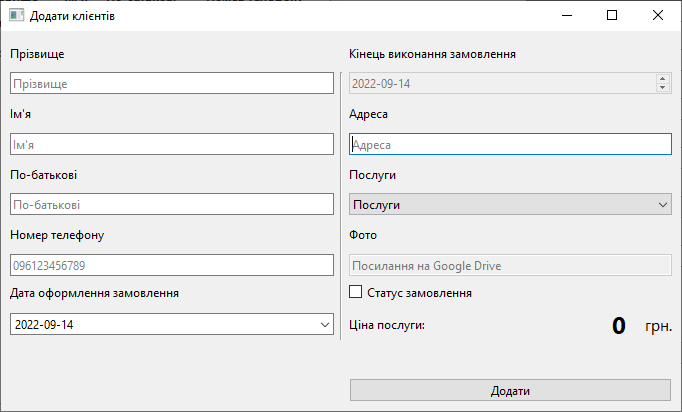


Рисунок 3.3 - Вікно додавання послуги

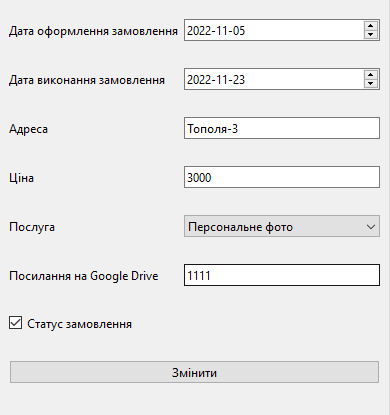


Рисунок 3.4 - Поле для зміни даних для обраного замовлення

Також програма дозволяє змінювати послуги та додавати нові.

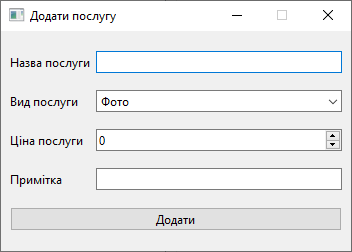


Рисунок 3.5 - Додавання послуги

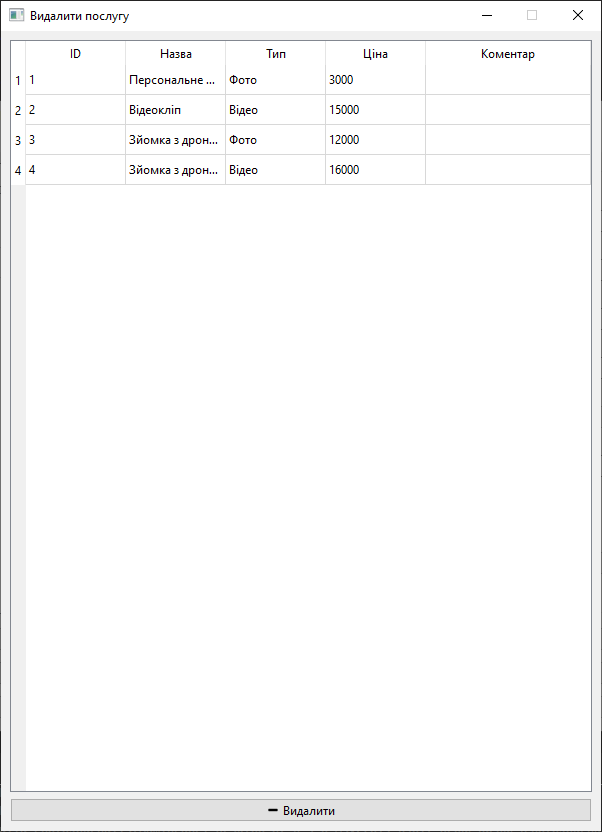


Рисунок 3.6 - Вікно для видалення або змінення обраної послуги

## 3.3 Контроль вхідних даних програми

У програмі для контролю вхідних даних було використано регулярні вираження класу QRegularExpression, приклад регулярного вираження та створення валідатору на введення даних лістинг 3.1.

Лістинг 3.1 – Приклад регулярного вираження для слів

QRegularExpression rx("[A-zА-я]+");

QValidator \*text\_validator = new QRegularExpressionValidator(rx, this);

Лістинг 3.2 – Приклад регулярного вираження для номеру телефону

rx = QRegularExpression("[0-9]{12}");

QValidator \*number\_validator = new QRegularExpressionValidator(rx, this);

Для виведення повідомлень про помилку або успіх було використано клас QMessageBox та види віконних повідомлень warning (для помилки) та information (для успіху). Приклад наведено у лістингу 3.2.

Лістинг 3.3 – Приклад виведення повідомлення помилки

QMessageBox::warning(this, "Ошибка", "Вы ввели не правильно номер телефона!");

Лістинг 3.3 – Приклад виведення повідомлення успіху

QMessageBox::information(this, "Успех", "Подключено!");

Лістинг 3.4 – Приклад виведення вікна з питанням про збереження

QMessageBox::question(this, "Сохранение", "Вы уверены, что хотите сохранить внесенные данные?", QMessageBox::Yes|QMessageBox::No)

# Висновок

Було розроблено програмний продукт який дозволить швидко створювати нове замовлення за однією з запропонованих послуг та змінювати інформацію вибраного замовлення.