

Попов А. А. КС-231(2)

Лабораторна робота №3

Тема: розробка кнопочних елементів управління




Мета: засвоїти основні принципи розробки компонентів управління людино-машинної взаємодії типу “командна кнопка”, отримати практичні навички програмної реалізації командних кнопок із заданими властивостями.

Порядок виконання роботи:

Завдання 1. Розробити сценарій діалогу для одного-двох кроків людино-машинної системи заданого типу. Передбачити максимальну можливу кількість.

Розроблено сценарій взаємодій «командна кнопка» для вибору способу доставки. Реалізовано перші два кроки: вибір варіанту доставки та активація кнопки підтвердження вибору («Обрати доставку»).

Крок 1:

При першому запуску користувач бачить напис "Оберіть спосіб доставки:", список із трьох емоджі-кнопок, які позначають способи доставки: " Кур'єр", " Пошта" та " Самовивіз", а також кнопку «Обрати доставку» внизу вікна. Кожна з цих кнопок має свій стан.

Кнопка доставки має активний стан, коли вона готова до взаємодії. Усі кнопки мають однаковий розмір, стиль, шрифт та іконку. Вони виглядають нейтрально, жодна не виділяється серед інших. При наведенні миші не відбувається масштабування або анімаційного ефекту, але змінюється колір рамки (border) кнопки — це сигналізує користувачеві, що кнопка може бути натиснута. Курсор при цьому змінюється з «Default» на «Hand», підкреслюючи інтерактивність елемента. Крім того, при наведенні через затримку приблизно 2 секунди з'являється підказка з описом способу доставки, наприклад, «Отримання у відділенні пошти».


Після вибору однієї з кнопок доставки, вона візуально виділяється кольором (фон або обводка), а решта залишаються в неактивному стані. Неактивні кнопки не блокуються повністю, але стають менш помітними. Проте з ними можна повторно взаємодіяти: наведення знову підсвічує рамку, і натискання робить таку кнопку новим активним вибором, при цьому попередня повертається у неактивний стан.

Таким чином, для кожної кнопки доставки реалізовано три візуальні стани: нормальний (активний), наведений (зміна обводки) та неактивний (коли вибрана інша кнопка).

Крок 2:

Кнопка «Обрати доставку» є залежною від стану вибору способу доставки. Поки користувач не зробив жодного вибору, кнопка залишається неактивною: вона має сірий фон, текст менш контрастний, і курсор при наведенні змінюється на заборонений символ — перекреслене червоне коло. У цьому стані кнопка є повністю недоступною для натискання, навіть якщо користувач намагається взаємодіяти з нею.

Як тільки користувач обирає один із способів доставки, кнопка переходить у активний стан. Її фон змінюється на яскравий зелений колір, текст стає білим, і курсор при наведенні змінюється на «Hand», що вказує на готовність до взаємодії. На відміну від кнопок вибору способу доставки, при наведенні на кнопку «Обрати доставку» не змінюється розмір або форма, але може візуально підкреслюватися активність за рахунок контрастного кольору або тіні (залежно від реалізації).

Після натискання на кнопку «Обрати доставку» з вибраним варіантом, відкривається спливаюче вікно з підтвердженням вибору. У вікні вказується спосіб доставки, який обрав користувач, наприклад: "Ви обрали спосіб доставки:  Пошта", та кнопка для закриття повідомлення.

Загалом, для цієї кнопки реалізовано чотири стани: неактивна (до вибору), активна (після вибору), наведена (візуально підсвічена рамкою або фоном), і натиснута (після підтвердження).

Завдання 2. Розробити дизайн командних кнопок (не менше двох) з графічним елементом та написом.

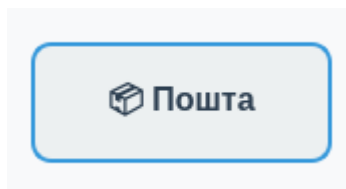


Рис. 1 – дизайн кнопки вибору доставки представленого текстом з емоджі

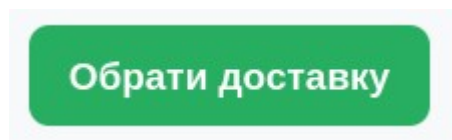


Рис. 2 - дизайн кнопки вибору в активному стані

Завдання 3. Розробити програмний код командних кнопок для панелі з властивістю Resize.

Код, яка відповідає за зміну розміру та форми кнопок при взаємодії:

```
void ConfirmButton::setupButton()
```

```

{

    setMinimumSize(150, 50);
    setSizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
    setMaximumSize(250, 50);

    setFont(QFont("Arial", 14, QFont::Bold));
    updateButtonStyle();
}

void ConfirmButton::setupAnimation()
{

    m_hoverAnimation = new QPropertyAnimation();
    m_hoverAnimation->setDuration(200);
    m_hoverAnimation->setEasingCurve(QEasingCurve::OutCubic);
}


void ConfirmButton::updateButtonStyle()
{
    QString activeStyle =
        "QPushButton {"
        "  background-color: #27ae60;"
        "  border: none;"
        "  border-radius: 10px;"
        "  color: white;"
        "  font-weight: bold;"

```

```

" padding: 12px;"
" transition: all 0.2s ease;"
"}"
"QPushButton:hover, QPushButton[hovered=\"true\"] {"
" background-color: #2ecc71;"
" border-radius: 15px;"
" transform: scale(1.05);"
"}"
"QPushButton[pressed=\"true\"] {"
" transform: scale(0.95);"
"}";

```

QString inactiveStyle =

```

"QPushButton {"
" background-color: #95a5a6;"
" border: none;"
" border-radius: 10px;"
" color: #7f8c8d;"
" font-weight: bold;"
" padding: 12px;"
"}";

```

```

setStyleSheet(m_isActive ? activeStyle : inactiveStyle);
}

```

```

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent), m_selectedDelivery(nullptr)
{

```

```
    setupUI();  
    connectSignals();  
    updateConfirmButton();  
}
```

```
MainWindow::~MainWindow()  
{  
}
```

```
void MainWindow::onDeliveryButtonClicked()  
{  
    DeliveryButton *clickedButton = qobject_cast<DeliveryButton*>(sender());  
    if (!clickedButton) return;  
  
    for (DeliveryButton *button : m_deliveryButtons) {  
        button->setSelected(false);  
    }  
  
    clickedButton->setSelected(true);  
    m_selectedDelivery = clickedButton;  
  
    updateConfirmButton();  
}
```

```
void MainWindow::onConfirmButtonClicked()  
{  
    if (m_selectedDelivery) {
```

```
QString selectedText = m_selectedDelivery->text();
```

```
QMessageBox msgBox;
```

```
msgBox.setWindowTitle("Вибір доставки");
```

```
msgBox.setText(QString("Ви обрали спосіб доставки: %1").arg(selectedText));
```

```
msgBox.setStandardButtons(QMessageBox::Ok);
```

```
msgBox.setStyleSheet(
```

```
    "QMessageBox {"
```

```
    "  background-color: #ecf0f1;"
```

```
    "  font-size: 14px;"
```

```
    "}"
```

```
    "QMessageBox QPushButton {"
```

```
    "  background-color: #3498db;"
```

```
    "  color: white;"
```

```
    "  border: none;"
```

```
    "  border-radius: 5px;"
```

```
    "  padding: 8px 16px;"
```

```
    "  font-weight: bold;"
```

```
    "min-width:300 px;"
```

```
    "min-height:80px;"
```

```
    "}"
```

```
);
```

```
msgBox.exec();
```

```
}
```

```
}
```

```
void MainWindow::setupUI()
```

```
{
```

```
setWindowTitle("Оберіть спосіб доставки");
```

```
setMinimumSize(500, 400);
```

```
resize(500, 400);
```

```
setStyleSheet("QMainWindow { background-color: #f8f9fa; }");
```

```
m_centralWidget = new QWidget(this);
```

```
setCentralWidget(m_centralWidget);
```

```
QVBoxLayout *mainLayout = new QVBoxLayout(m_centralWidget);
```

```
mainLayout->setSpacing(30);
```

```
mainLayout->setContentsMargins(50, 50, 50, 50);
```

```
QLabel *titleLabel = new QLabel("Оберіть спосіб доставки:");
```

```
titleLabel->setAlignment(Qt::AlignCenter);
```

```
titleLabel->setFont(QFont("Arial", 18, QFont::Bold));
```

```
titleLabel->setStyleSheet("color: #2c3e50; margin-bottom: 20px;");
```

```
titleLabel->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Fixed);
```

```
mainLayout->addWidget(titleLabel);
```

```
QHBoxLayout *deliveryLayout = new QHBoxLayout();
```

```
deliveryLayout->setSpacing(20);
```

```
m_courierButton = new DeliveryButton("🚚 Кур'єр", "Доставка кур'єром до дверей");
```



```
m_postButton = new DeliveryButton("📦 Пошта", "Отримання у відділенні  
пошти");  
  
m_pickupButton = new DeliveryButton("🏠 Самовивіз", "Самостійне отримання в  
магазині");  
  
m_deliveryButtons = {m_courierButton, m_postButton, m_pickupButton};  
  
deliveryLayout->addStretch();  
deliveryLayout->addWidget(m_courierButton);  
deliveryLayout->addWidget(m_postButton);  
deliveryLayout->addWidget(m_pickupButton);  
deliveryLayout->addStretch();  
  
mainLayout->addLayout(deliveryLayout);  
  
mainLayout->addStretch();  
  
m_confirmButton = new ConfirmButton("Обрати доставку");  
QHBoxLayout *confirmLayout = new QHBoxLayout();  
confirmLayout->addStretch();  
confirmLayout->addWidget(m_confirmButton);  
confirmLayout->addStretch();  
  
mainLayout->addLayout(confirmLayout);  
}
```

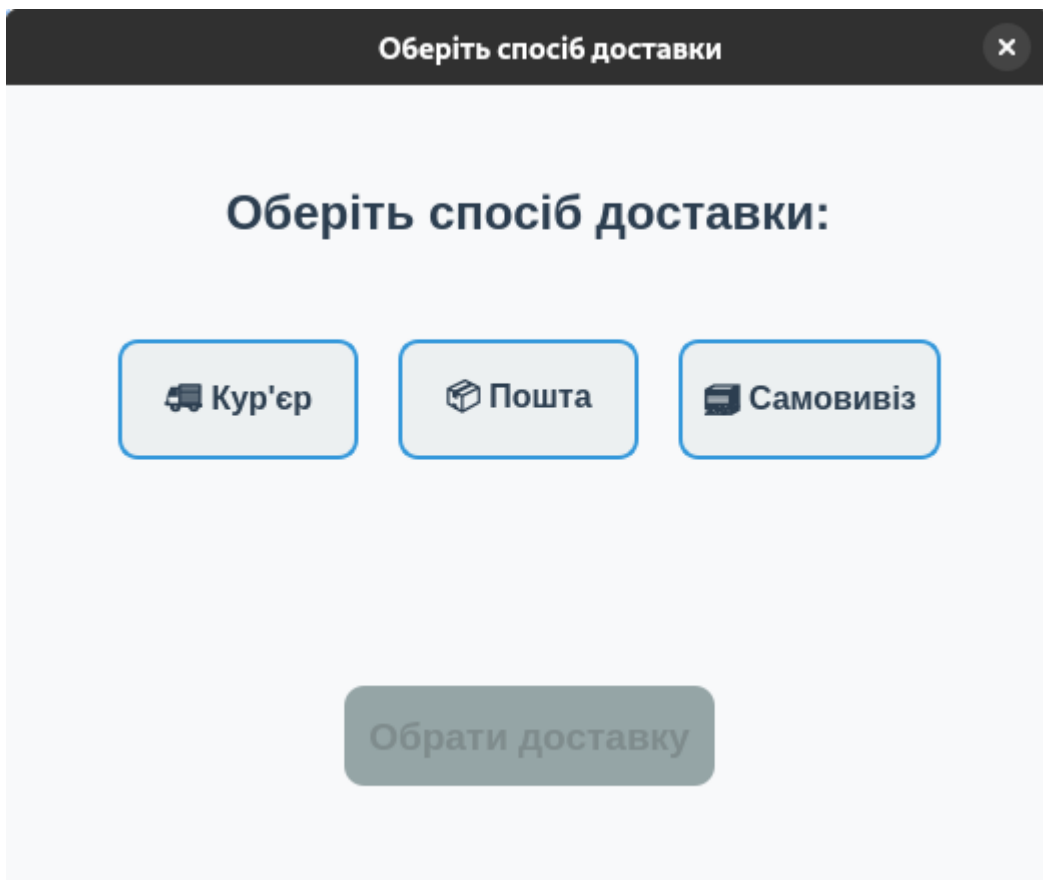


Рис. 3 – вигляд кнопок у вікні з незмінним розміром

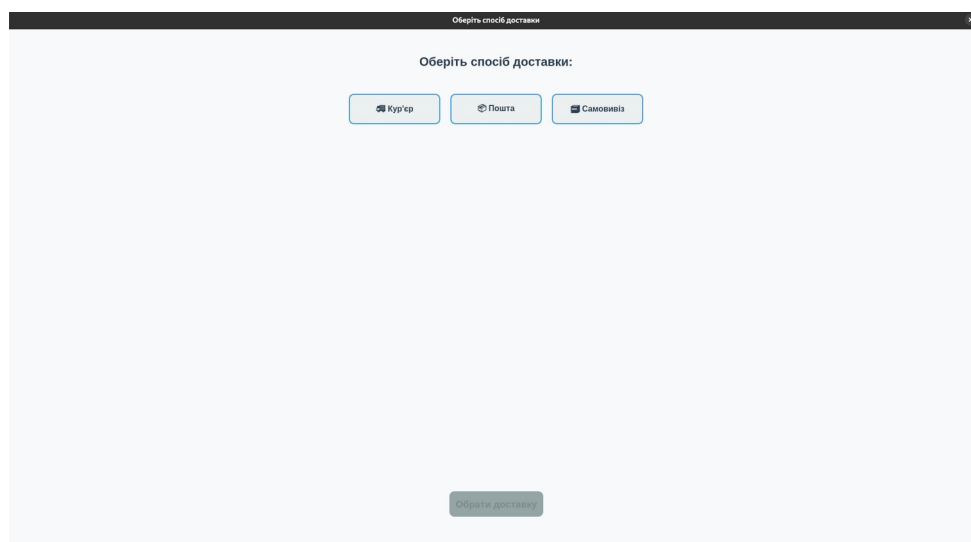


Рис. 4 – вигляд кнопок у вікні зі зміненим розміром

Завдання 4. Провести експеримент з перевірки відображення командних кнопок у різних станах.

Стан кнопок при запуску програми: кнопки вибору настрою в активному стані (однакового розмірі, використаний border синього кольору, текст з емоджі чорного кольору, курсор у вигляді руки)
кнопка «Обрати доставку» в неактивному стані (сіра, текст чорного кольору)

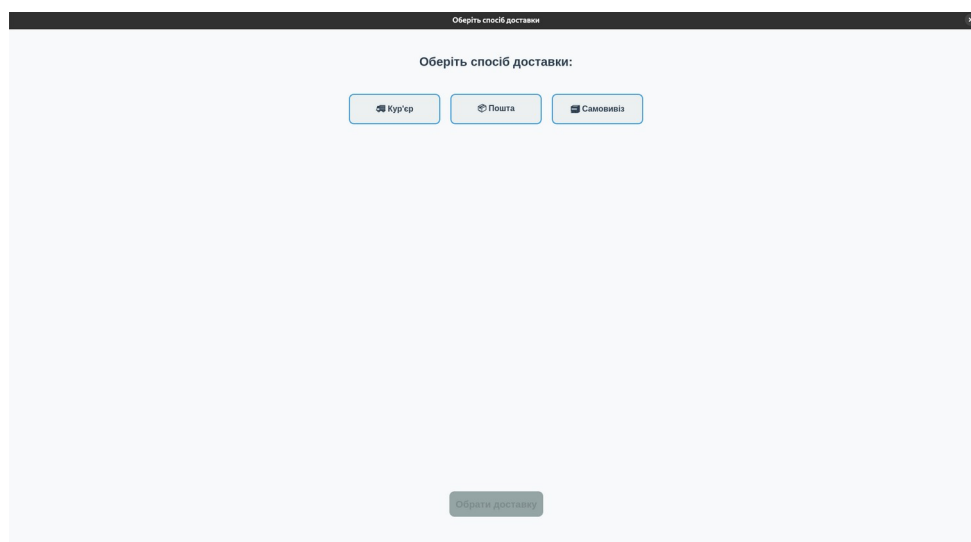


Рис. 5 – кнопки вибору настрою в активному стані.

Кнопка «Обрати доставку» в неактивному стані.

При наведенні мишкою на кнопку вибору доставки (курсор переходить в стан «Hand») вона змінює border(рамку) на більш темний колір.

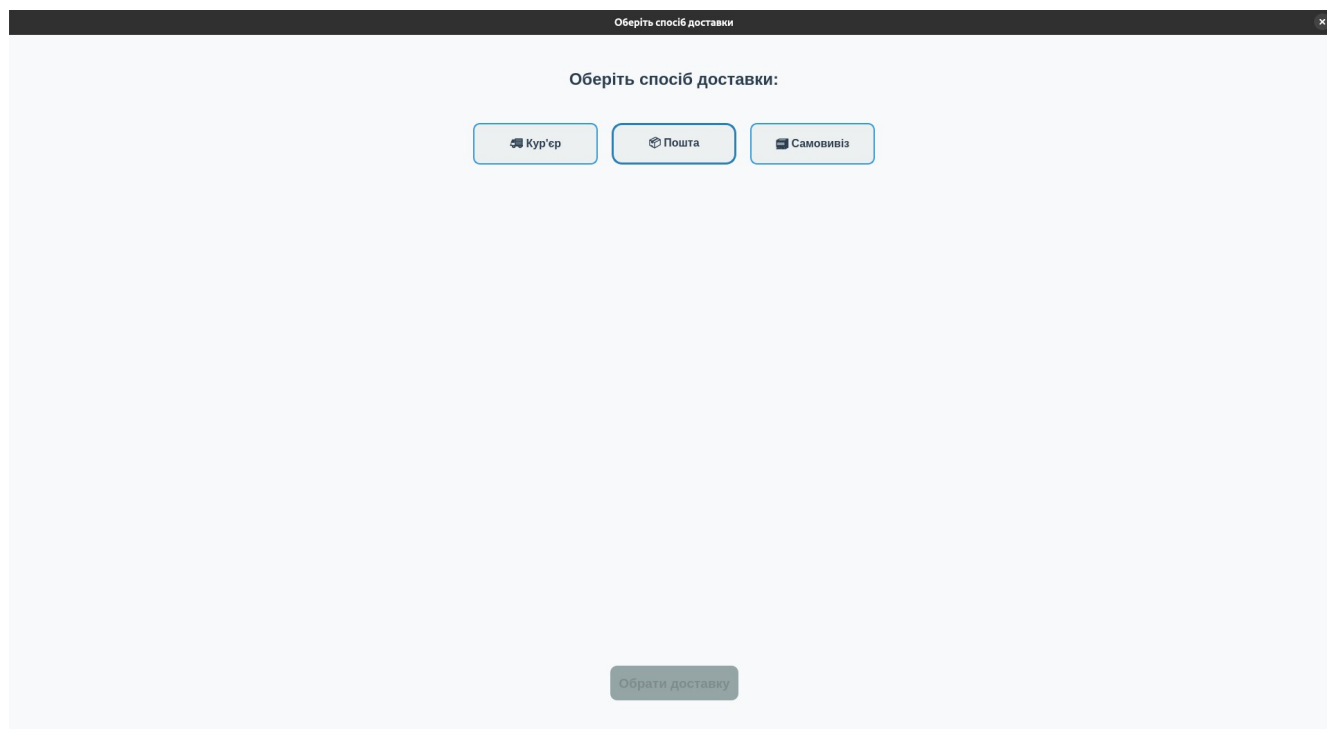


Рис. 6 – дизайн кнопок вибору настрою в активному стані при наведенні

Після вибору доставки і натискання на кнопку, всі інші кнопки залишаються в активному стані (це дозволяє змінити вибір користувача, якщо вибір зроблено

неправильно). Обрана кнопка змінює колір на яскраво-синій, та колір тексту стає білим.

Також при цьому кнопка «Обрати доставку» переходить в активний стан, змінюючи колір на яскравий, зелений колір. Колір тексту кнопки, при переходженні в активний стан, змінюється на білий.

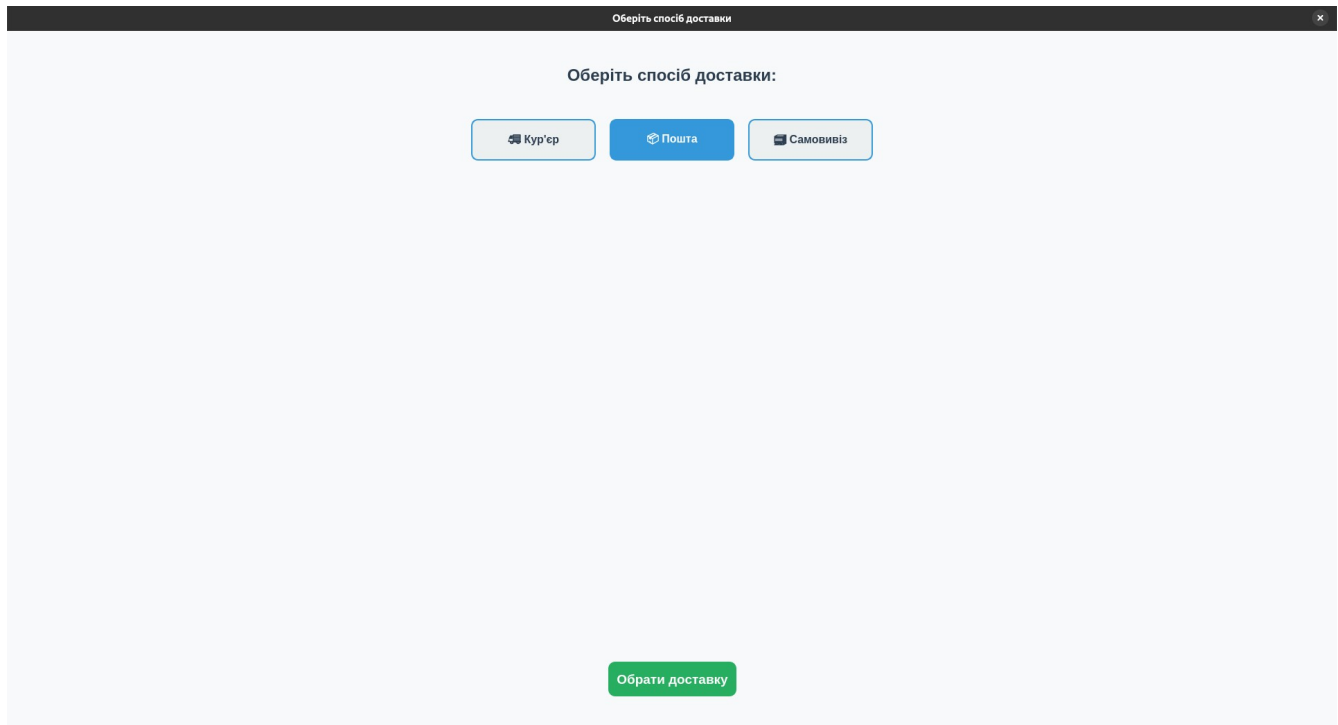


Рис. 7 – дизайн обраної кнопки вибору доставки змінено, та кнопка «Обрати доставку» в активному стані.

При наведенні мишкою на кнопку вибору «Обрати доставку» (курсор переходить в стан «Hand»), вона змінює колір на більш яскравий та заокруглює краї.

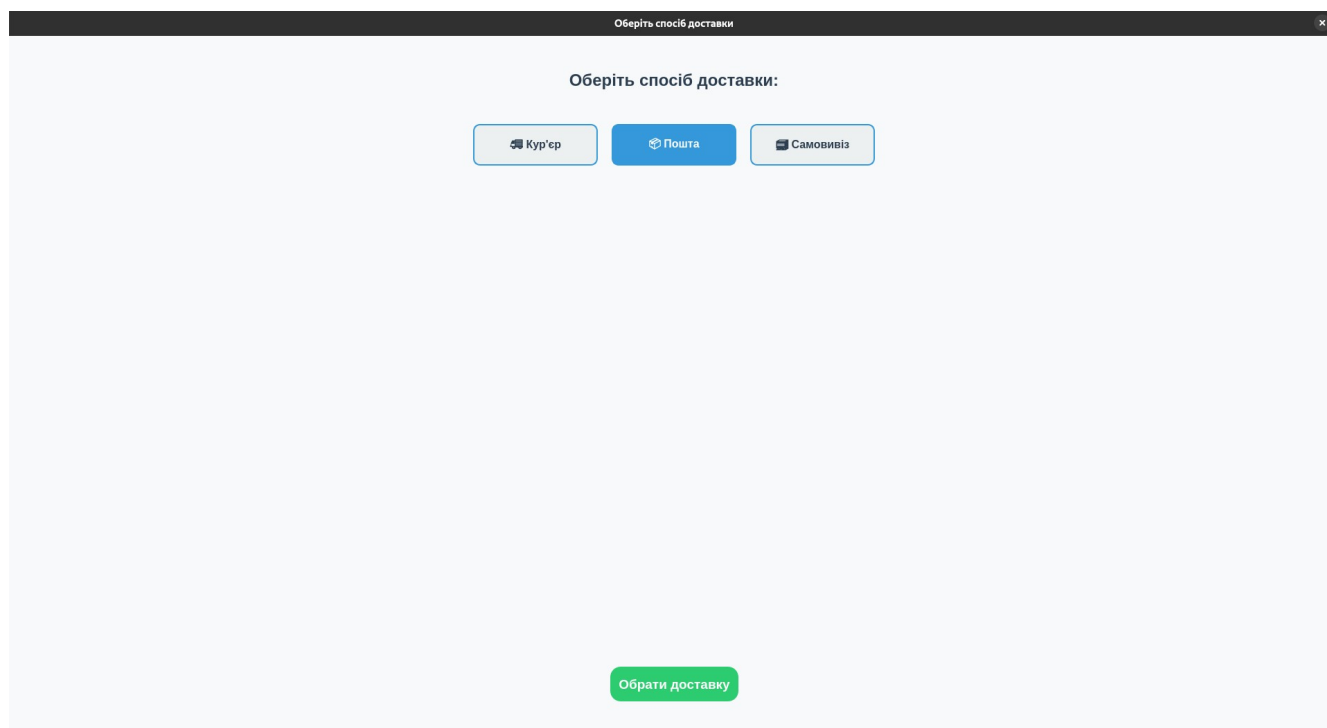


Рис. 8 – дизайн кнопки вибору в активному стані при наведенні

Завдання 5. Зробити висновки по роботі. У висновках відобразити оцінку трудових затрат на створення командних кнопок, інформаційну ефективність їхнього дизайну, можливі обмеження на використання.

У результаті виконання роботи було засвоєно основні принципи розробки компонентів управління людино-машинної взаємодії типу «командна кнопка», а також отримано практичні навички програмної реалізації таких кнопок із заданими властивостями у середовищі Qt.

У ході роботи було створено застосунок із набором командних кнопок для вибору способу доставки: «Кур'єр», «Пошта» та «Самовивіз». Було реалізовано зміну візуальних станів кнопок: активного, неактивного, наведення, а також логіку залежності підтвердження вибору. Загальна оцінка трудових затрат свідчить про помірну складність реалізації: базове налаштування функціоналу потребувало

близько 3 годин, з яких найбільше часу було витрачено на організацію візуального зворотного зв'язку, а саме — стилізацію рамок, кольорів, обробку станів та відображення повідомлень.

Створені кнопки виявилися інформаційно ефективними завдяки використанню зрозумілих візуальних символів (емодзі) та коротких підписів. Інтерфейс виявився інтуїтивним: зміна стилю при наведенні та виборі допомагає користувачу легко орієнтуватися серед варіантів доставки, а активність кнопки підтвердження забезпечує запобігання випадковим натисканням без вибору. Такий зворотний зв'язок значно підвищує зручність, швидкість взаємодії та позитивно впливає на загальний користувацький досвід.

Обмеження проєкту стосуються переважно нюансів дизайну. Наприклад, обведення замість масштабування потребує чіткої контрастності для помітності, а також адаптації для користувачів із вадами зору або моторики. Залежність від подій наведення й натискання вимагає точно налаштованих стилів, а за потреби — додаткових альтернативних способів взаємодії (клавіатурна навігація, підсвічування фокусом тощо).

Загалом реалізовані командні кнопки демонструють вдале поєднання естетики, функціональності та доступності, що робить їх ефективним елементом у побудові зручного й зрозумілого інтерфейсу вибору в людино-машинній системі.