МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

кафедра програмних засобів

Звіт

з лабораторної роботи № 3

з дисципліни «Людино-машинна взаємодія»

на тему:

«Етапи проектування користувальницького інтерфейсу. Вибір елементів керування. Текст і числа»

Виконав

Ст. гр. КНТ-137 В.В. Козлов

Прийняв

Ассистент Ж.К. Камінська

1. Мета роботи

Освоїти третій етап проектування користувальницького інтерфейсу, в ході якого необхідно вивчити основи дизайну форм і елементи керування.

1. Завдання до роботи

Ознайомитися з теоретичними відомостями та конспектом лекцій.

Для обраного в першій лабораторній роботі об'єкту (каси самообслуговування) продумати такий дизайн форм, щоб користувач міг легко працювати з інтерфейсом.

Програмно реалізувати інтерфейс що містить наступні елементи керування: елементи для вибору, числові, текстові, дати і час.

Оформити звіт по роботі.

Відповісти на контрольні питання.

1. Результати виконання роботи

Після розробки структури форми та макета екрану інтерфейсу каси самообслуговування, необхідно визначити перелік елементів керування, необхідних для успішного виконання поставлених завдань.

* 1. Головна форма

У верхній частині головного екрану розмістимо інформацію про поточну дату та час (рис. 3.1 №1). Для цього обираємо елемент керування – label (мітка). Щохвилини оновлюємо значення яке виводить цей елемент керування:

private void ClockTimer\_Tick(object sender, EventArgs e)

{

CurrentTime.Text = DateTime.Now.ToString("dd MMMM, HH:mm"); //Властивості Text мітки CurrentTime передаємо значення поточної дати та часу у форматі “день місяць, година:хвилини”

}

Перш за все користувач (покупець) має додати обрані товари до чеку, тому у центральній частині головного екрану розмістимо таблицю з даними про відскановані товари (рис. 3.1 №2), а для додавання товару застосуємо керуючу кнопку, яка викликає вікно додавання товару до списку (рис. 3.1 №3).

private void AddNewProduct\_Click(object sender, EventArgs e) //Оброблюємо ситуацію натискання кнопки «Додати новий товар»

{

UpdateLastActionTime(); //Оновлюємо час останньої дії користувача. Це необхідно для того, щоб система розуміла, чи справді на даний момент користувач використовує її. У негативному випадку система через хвилину бездіяльності переходить у режим очікування

BarcodeReadingWindow window = new BarcodeReadingWindow(this);

BarcodeReading = window;

window.ShowDialog(); //Створюємо окно введення інформації про додаваємий товар, і показуємо його

BarcodeReading = null;

}

Для визначення кількості одиниць товару або його видалення зі списку для кожного товару застосуємо керуючі кнопки (рис. 3.1 №4).

Для того, щоб користувач міг викликати асистента і отримати допомогу, додамо керуючу кнопку для виклику асистента (рис. 3.1 №5).

private void CallAdmin\_Click(object sender, EventArgs e) //Оброблюємо натискання кнопки виклику асистента

{

AssistantWaitingWindow window = new AssistantWaitingWindow(false, this);

IsCheckActivity = false;

window.ShowDialog(); //Створюємо вікно з інформаціїним повідомленням «Очікування Асистента…» і блокуємо роботи каси.

UpdateLastActionTime(); //Оновлюємо час останньої дії користувача, як час приходу асистента

IsCheckActivity = true;

}

Для наладчика додамо керуючу кнопку яка викликає вікно авторизації до системного меню (рис. 3.1 №6).

Для того, щоб користувач (покупець) міг оцінити вартість усієї покупки, додамо елемент керування label (мітка), який буде містити інформацію про загальну вартість покупки (рис. 3.1 №7)

Для сплати коштів за покупку додамо керуючу кнопку «Сплатити» (рис. 3.1 № 8).

private void PayMoney\_Click(object sender, EventArgs e) //Обробка натискання кнопки «Сплатити»

{

UpdateLastActionTime();

if (ListOfProducts.Count == 0) return; //У випадку коли користувач (покупець) натиснув кнопку, попередньо не додавши товарів, то зрозуміло, що сплачувати йому немає чого. Отже на натискання кнопки не реагуємо

PaymentWindow window = new PaymentWindow(this);

Payment = window;

window.ShowDialog(); //Створюємо вікно сплати коштів за покупку та виводимо його на екран

Payment = null;

}

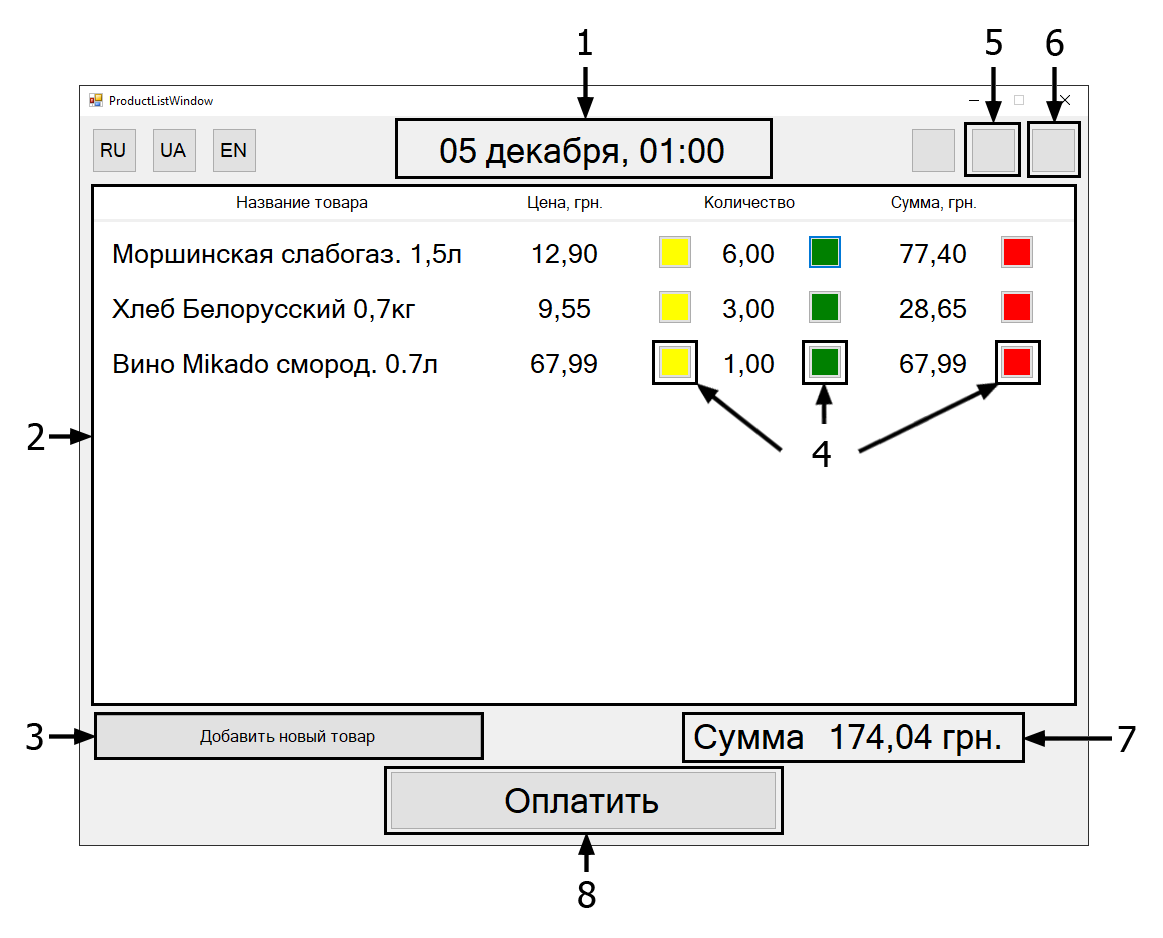


Рисунок 3.1 – Головна форма

* 1. Фома додавання товару

За допомогою мітки виведемо повідомлення «Штрихкод». За допомогою іншої мітки будемо виводити на екран поточний введений штрихкод (рис. 3.2 №1).

Для того щоб користувач міг повернутися у попереднє меню, додамо керуючу кнопку виходу з форми (рис. 3.2 №2).

private void Cancel\_Click(object sender, EventArgs e) //Обробка натискання кнопки назад

{

ProductList.UpdateLastActionTime();

this.Close(); //Закриваємо поточну форму введення штрихкоду товара

}

Для того, щоб користувач (покупець) міг підтвердити введений ним штрихкод додамо керуючу кнопку для підтвердження дії (рис. 3.2 №3).

Для надання можливості користувачу ввести штрихкод товару додамо набір з кнопок перемикачів (рис. 3.2 №4).

У випадку спроби підтвердження не існуючого штрихкоду, виведемо повідомлення «Невірний штрихкод».

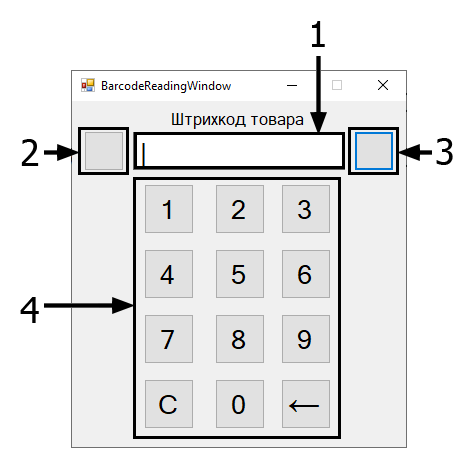


Рисунок 3.2 – Форма додавання товару

* 1. Форма сплати

Щоб сплатити вартість обраних товарів користувачу необхідно вказати дані своєї карти. Для цього призначено вікно сплати.

У верхній частині вікна сплати розмістимо мітку з інформацією про загальну вартість покупки (рис. 3.3 №1).

Для введення номера карти та паролю використаємо набір з кнопок перемикачів (рис. 3.3 №2), а для відображення введених даних використаємо інформаційні мітки (рис. 3.3 №3).

Для того щоб користувач міг повернутися у попереднє меню та змінити список товарів, додамо керуючу кнопку виходу з форми (рис. 3.3 №4).

Для підтвердження даних про карту додамо керуючу кнопку підтвердження (рис. 3.3 №5).

У випадку введення неправильної інформації, виведемо повідомлення «Неправильний номер карти або пароль», а у випадку недостатньої кількості коштів «Не вистачає коштів на рахунку».

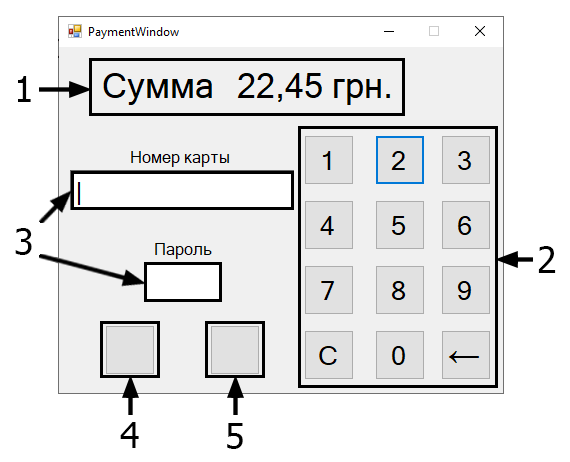


Рисунок 3.3 – Форма сплати

* 1. Форма авторизації до системного меню

Щоб до системного меню не міг зайти звичайний користувач, додамо форму з авторизацією до меню (рис. 3.4).

Додамо до форми два текстових поля, для введення логіну та паролю (рис. 3.4 №1).

Для того, щоб користувач (покупець) міг підтвердити введені ним дані для авторизації додамо керуючу кнопку для підтвердження дії (рис. 3.4 №2).

private void Accept\_Click(object sender, EventArgs e) //Обробка натискання кнопки підтвердження

{

ProductList.UpdateLastActionTime();

if (Login.Text == "admin" && Password.Text == "admin") //На даному етапі розробки для доступу достатньо щоб користувач увів логін «admin» та пароль «admin».

{

SettingsWindow window = new SettingsWindow(); //Створюємо і відкриваємо вікно з налаштуваннями

window.ShowDialog();

ProductList.Close(); //Закриваємо всі форми і повертаємося до вікна очікування

this.Close();

}

else //У випадку введення не правильного логіну або паролю виведемо повідомлення «Помилка доступу»

{

MessageWindow window = new MessageWindow("Ошибка доступа");

Message = window;

window.ShowDialog();

Message = null;

}

}

Для того щоб користувач міг повернутися у попереднє меню, додамо керуючу кнопку виходу з форми (рис. 3.4 №3).

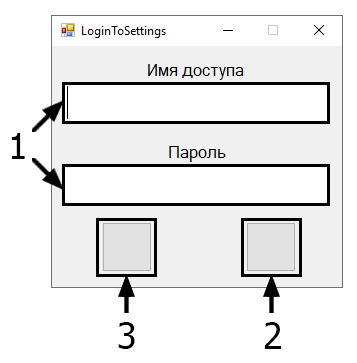


Рисунок 3.4 – Форма авторизації до системного меню

* 1. Форма системного меню

За допомогою мітки виведемо повідомлення «Системне меню каси» (рис. 3.5 №1).

Для того, щоб користувач міг закрити вікно системного меню додамо керуючу кнопку для виходу (рис. 3.5 №2).

Створимо панель на якій розмістимо всю інформацію про стан каси самообслуговування та керуючі кнопки (рис. 3.5 № 3).

Для відображення статистичної інформації додамо елемент пачка карток, з чотирма категоріями (рис. 3.5 №4).

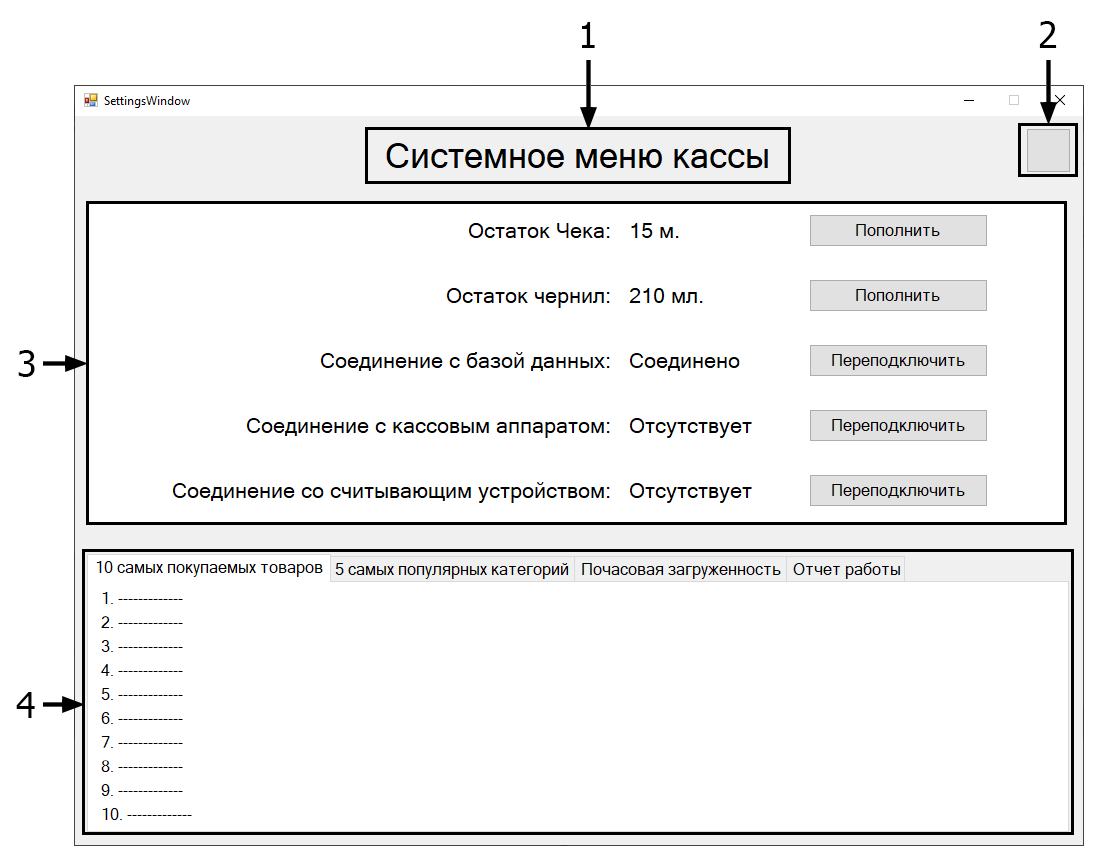


Рисунок 3.5 – Форма системного меню

Висновки

Протягом виконання цієї лабораторної роботи ознайомився з третім етапом проектування і створення користувальницького інтерфейсу, в ході якого вивчив основи дизайну форм і елементи керування.