

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи автоматів
самообслуговування на базі сенсорних терміналів платіжної
системи EasyPay”

КБПЗ - 2024

Виконав здобувач вищої освіти
IV курсу, групи КІ-20
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Квач Д.Р.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Дресєв О.М.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Квачу Данілу Руслановичу

(прізвище, ім'я, по батькові)

- Тема роботи Програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay
- Керівник роботи Дресєв Олександр Миколайович, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 131-02 від 01.04.2024 року
- Строк подання студентом роботи до захисту 23.05.2024 р.
- Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.
 - Перегляд аналогічних існуючих систем.
 - Опис і обґрунтування проектних рішень.
 - Етапи програмування системи.
 - Впровадження системи в промислову експлуатацію.
 - Висновки
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

<u>Структурна схема системи</u>	<u>1 аркуш</u>
<u>Функціональна схема системи</u>	<u>1 аркуш</u>
<u>Діаграма процесів</u>	<u>1 аркуш</u>
<u>Блок-схема алгоритму роботи додатку</u>	<u>2 аркуша</u>

7. Дата видачі завдання « 17 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання
« 17 » січня 2024 р.

Підпис керівника

Дреєв О.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2024 р.

Підпис здобувача

Квач Д.Р.
(прізвище та ініціали)

АНОТАЦІЯ

Квач Д.Р. Програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Метою розробки є програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Результат роботи – програмна реалізація системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.1.

Ключові слова: комп'ютерна інженерія, EasyPay

ABSTRACT

Kvach D.R. Self-service vending machine software based on sensor terminals of the EasyPay payment system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the first (bachelor) level of higher education, software was developed, which is intended for the system of self-service machines based on sensor terminals of the EasyPay payment system.

The purpose of the development is the software of the self-service machine system based on the sensor terminals of the EasyPay payment system.

The result of the work is the software implementation of the system of self-service machines based on sensor terminals of the EasyPay payment system.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Delphi 10.4.1 environment.

Keywords: computer engineering, EasyPay

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	32
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	53
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 ОСНОВНІ ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65

					ВКРБ-123.24.0005.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Квач Д.Р.				Програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay	Літ.	Аркуш	Аркушів
Перев.	Дресв О.М.					Б	1	71
Н.контр.	Коваленко А.С.				ЦНТУ КІ-20			
Затв.	Смірнов О.А.							

ВСТУП

Актуальність теми. Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay – це нова модульна платформа, яку можна налаштувати для різноманітних додатків самообслуговування, включаючи лояльність, реєстрацію, продаж квитків, нескінченні проходи тощо. Корпус оснащений 21,5-дюймовим сенсорним екраном PCAP для чіткого та сучасного інтерфейсу користувача.

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay доступна у вигляді окремої підставки або стільниці, що забезпечує кілька варіантів розгортання в одній стандартизованій сім'ї. Корпус можна налаштувати за допомогою бажаних компонентів, загальних для більшості транзакцій самообслуговування – від платіжних пристроїв EMV, сканерів, принтерів квитанцій/квитків/карток тощо. Як і всі конструкції, автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay налаштований для простого обслуговування. Він має всі передні дверцята для простого доступу до компонентів. Конструкція, сумісна з ADA, забезпечує доступ до самообслуговування для всіх ваших клієнтів.

Більшість високонадійних компонентів у Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay виробляються на Тайвані. Компоненти відомі своїм довгим життєвим циклом і високою надійністю, що знижує витрати на обслуговування протягом усього терміну експлуатації.

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay – це модульний кіоск, який можна налаштувати, щоб доповнити параметри розгортання та посилити бренд компанії. Створений спеціально для різноманітних додатків самообслуговування, цей кіоск можна інтегрувати з декількома компонентами транзакцій, забезпечуючи універсальну продуктивність, щоб допомогти власникам бізнесу підвищити залучення клієнтів.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay доступний у вигляді настільного кіоску або підлогового кіоску з п'єдесталом, що забезпечує модульні варіанти розгортання в одній стандартизованій сім'ї.

Усі двері доступу відкриваються спереду, щоб забезпечити легкий доступ для швидкого обслуговування та обслуговування. Немає необхідності відсувати кіоск від стін або демонтувати основний корпус кіоску, щоб видалити компоненти для обслуговування.

Дозволяє вибрати із ряду периферійних пристроїв, включаючи сканер штрих-кодів, принтер квитанцій/квитків, купюроприймач і диспенсер карток, щоб задовольнити ваші вимоги щодо транзакцій.

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay, розроблена для загального доступу, підходить для широкого спектру додатків самообслуговування, включаючи: роздрібні точки продажу, програми лояльності, самостійне замовлення ресторану / фуд-корту, самообслуговування готелю вхід, реєстрація пацієнтів у лікарні, самостійний продаж квитків у театр/музей та контроль доступу.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.
- Дослідження системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.
- Програмна реалізація системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ_2024

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

EasyPay – платіжний сервіс, призначений для оплати різних повсякденних послуг, від мобільного зв'язку й ЖКГ до банківських кредитів.

Особливість сервісу EasyPay полягає в тому, що оплата може вироблятися як готівкою в мережі платіжних терміналів EasyPay, так і через інтернет-сервіс і додаток для мобільних пристроїв. Фактично користувач може робити оплату зі свого рахунку в системі, перебуваючи в зручному для нього місці й у зручний час.

Операції в Україні здійснюються в національній валюті – гривні.

Автомати самообслуговування (ОСМП) – апаратно-програмне рішення, призначене для проведення платежів через систему процесингу. Для роботи автомата не потрібно додаткового персоналу. Широка лінійка автоматів самообслуговування дозволяє розмістити їх як на вулиці, так і в будь-якому приміщенні, а програмне забезпечення дозволяє клієнтам швидко оплатити потрібну послугу.

1.2 Область застосування

Сьогодні за допомогою платіжного сервісу EasyPay можна робити платежі на користь більш ніж 5000 компаній – операторів різних послуг, серед них:

- оператори мобільного зв'язку;
- електронні платіжні системи;
- контент-провайдери й агрегатори;
- інтернет-провайдери, оператори послуг IP-телефонії, комерційного телебачення;
- компанії, що надають послуги ЖКГ і фіксованого зв'язку;

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

- інтернет-магазини;
- і інші.

Крім перерахованих сервісів, стала доступна оплата послуг туристичних операторів, внесення пожертвувань на користь благодійних фондів, а також з'явилася можливість здійснювати грошові перекази UNIstream і здобувати віртуальні карти VISA. Оплата різних послуг здійснюється в мережі платіжних терміналів під брендом EasyPay, для цього досить в інтерфейсі терміналу вибрати послугу, що цікавить, і, слідуючи підказкам, зробити платіж. Якщо користувач бажає оплачувати послуги через Інтернет і стільниковий телефон, необхідно зареєструвати аккаунт для доступу до «EasyPay гаманця» і додатку «EasyPay у мобільному».

Основна якість бренду EasyPay – це орієнтованість на кінцевого споживача. EasyPay створив для своїх користувачів унікальну екосистему з прийому платежів, яка об'єднала в собі майже всі платіжні технології, створені у світі, які дають можливість просто, швидко і зручно оплачувати максимальний набір послуг у будь-який час і в будь-якій точці світу. Персоналізація платників, передплачені рахунки, альтернативні інтерфейси, альтернативні джерела коштів – ось перелік максимально сучасних сервісів, які потрібні користувачам по всьому світу.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Проведемо огляд автоматів самооплати.

Платіжний термінал МЕДІА

Опис:

1. Платіжний термінал «МЕДІА» відмінно працює в місцях з високої прохідності.

2. За рахунок другого монітора, на якому можна включити барвисті рекламні ролики, він зауважує до себе більше уваги.

«Медія» увібрав у себе всі переваги «Стандарту» – це надійний корпус із двохміліметрової сталі. Міцні ригельні замки. Глянсове фарбування корпуса, який не можуть похвастатися термінали з корпусами низької якості (на гладкому глянсовому корпусі видні всі вади – тому корпуса «середньої» якості красять шорсткуватою шагренню).

Апарат дуже надійний, тому що в ньому не використовуються китайські комплектуючі. У ньому встановлений перевірений роками й тисячами терміналів купюроприймач Cash Code (Канада). Довговічний термопринтер Custom VKP 80 II (Італія) або Citizen CT S2000 (Японія), 4G модем Siemens mc35i (Німеччина).

Надійна комплектація дозволяє апарату працювати без збоїв 24 години на день, 7 днів у тиждень.

Комплектація:

- Вандалостійкий металевий корпус.
- Мультимедіа монітор 19" широкоформатний.
- Сенсорний монітор (Acer 17", Kее Touch).

					VKPB-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- Купюроприймач Cash Code SM 1500 Cash Code MVU інкасаційний мішок на 3000 купюр.
- Термопринтер Custom VKP-80 II Citezen CT-S2000.
- 4G Siemens MC 35i.
- Промисловий комп'ютер Intel D945GCLF Atom 230/512Мб/80Gb/LAN100.



Рисунок 2.1 – Платіжний термінал МЕДІА

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Платіжний термінал СТАНДАРТ

Напольний термінал для приміщень. Моделі з корпусами лінійки "Стандарт" дуже добре зарекомендували себе на ринку як надійні й зручні апарати з високою "збиранням". Займають приблизно половину вітчизняного ринку платіжних терміналів.

Найбільш зручний, захищений і ергономічний корпус у сукупності з різними варіантами комплектацій роблять цей термінал просто незамінним. Модель "Ультра" – це максимальна комплектація платіжного терміналу. У ній використовуються самий надійний купюроприймач і термопринтер.

Платіжний термінал "Стандарт" – це, насамперед, сучасний і зручний автомат по прийому грошей за послуги. Його головна зовнішня перевага – це яскравий і помітний дизайн. Навіть у малогабаритних приміщеннях, напевно знайдеться місце для платіжного терміналу "Стандарт".

Власникові такого апарата не варто турбуватися за його схоронність, оскільки корпус пристрою виготовлений з міцної 2-3 мм сталі, термінал оснащений ригельним замком, а гроші – в окремому сейфі.

Комплектація:

- Вандалостійкий металевий корпус.
- Сенсорний монітор (Acer 17", Kee Touch).
- Купюроприймач Cash Code SM с касетою на 1500 купюр / Cash Code MVU + мішок на 3000 купюр.
- Термопринтер Custom VKP80 II / Citizen 2000.
- 4G модем Siemens MC35i.
- Промисловий комп'ютер 2000MHz/1024Mb/320Gb/LAN100/Audio/Video.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10



Рисунок 2.2 – Платіжний термінал СТАНДАРТ

Платіжний термінал ЗУПИНКОВИЙ

Вуличні апарати, як правило, ставляться біля зупинкових кіосків, якщо кіоск цілодобовий, а в продавщиці тривожна кнопка – переживати не коштує. А от якщо кіоск уночі не працює – те можна поставити термінал «Зупинковий» і попросити продавця закривати його перед відходом. Надійні рольставні товщиною 2 міліметри.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Надійні комплектуючі, система клімат контролю, морозостійке й вандалостійке сенсорне скло Zytronic товщиною більше 10 мм дозволять вашому апарату працювати 24 години на добу 7 днів у тиждень. У жару +40 і холод -40, у дощ і сніг.



Рисунок 2.3 – Платіжний термінал ЗУПИНКОВИЙ

Комплектація:

- Вандалостійкий металевий корпус.
- Монітор Acer 17””.
- Вандалостійке сенсорне скло – вуличне, Zytronic 17”, триплекс.
- Купюроприймач Cash Code SM с касетою на 1500 купюр.
- Термопринтер Custom VKP80 II.
- 4G модем Siemens MC35i Terminal.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Промисловий комп'ютер Intel D945GCLF Atom
230/512Мб/80Gb/LAN100.

- Клімат контроль. Система клімат контролю з термодатчиком.
- Рольставні для захисту в темний час доби.

Серія кіосків Expanse

Серія кіосків Expanse була створена, щоб забезпечити сімейство стандартизованих корпусів, які можна легко адаптувати для розміщення широкого спектру транзакційних компонентів, таким чином обслуговуючи численні вертикальні ринкові програми. Усвідомлюючи, що кожне розгортання кіоску має певні унікальні вимоги до транзакцій, KIOSK застосував модульний підхід до проектування, щоб поєднати можливість налаштування компонентів зі стандартними конструкціями моделей і металами. Кіоск Expanse складається з трьох розмірів із додатковими шафами для розміщення початкового, середнього або дуже складного набору компонентів.

Гнучкість, запропонована в серії Expanse, вирішує багато поширених потреб у налаштуванні, включаючи параметри розміру/орієнтації дисплея та можливість розміщення більше одного набору компонентів у розгортанні. Поєднання простих і сучасних варіантів дизайну в різних варіантах використання зменшує гонорари за індивідуальне проектування та прискорює час виходу на ринок. Невеликі коригування можна легко внести, не проектуючи індивідуальний корпус повністю з нуля.

Платформа кожного розміру має спільні елементи дизайну:

- Параметри розміру та орієнтації РК-дисплея. Сенсорні дисплеї Brilliant Projective Capacitive (PCAP) кріпляться за стандартом VESA та мають універсальність відображення у вертикальній або альбомній орієнтації. Вибір розміру дисплея включає 32", 27" або 21".

- Програмоване світлодіодне освітлення для акцентів передньої панелі (підсвічується на компонентах і основі), що забезпечує сучасний вигляд і покращену видимість.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

– Чітке розміщення компонентів для легкого доступу до кожного пристрою (деякі встановлені під поверхнею корпусу, деякі встановлені на дверцятах, з більшими компонентами на полицях або висувних напрямних).

Професіонал із продажів KIOSK може підказати вам розмір шафи серії Expanse, який підійде для роботи, виходячи з вашого конкретного набору компонентів.

В загальному:

– Великі корпуси були б ідеальними для розміщення багатьох компонентів, необхідних для таких додатків, як реєстрація авіакомпаній/маркування багажу, транзитні квитки або платіжні платформи як готівкою, так і монетами.

– Корпус середнього розміру підійде для багатофункціональних програм для оплати рахунків, лояльності та безпеки.

– Невеликий корпус підійде для легких квитків і оплати, нескінченних проходів, додатків типу введення замовлень.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкістю. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCL, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

масштабується під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений. Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ_2024

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Проведемо опис частин автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Існує багато типів систем самообслуговування, таких як автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. Вони прості у використанні та впроваджуються в різних галузях промисловості. У цьому розділі розглянемо, що таке автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay та платіжні системи, які галузі виграють від використання цієї технології та різні типи машин.

Автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay та платіжні системи

Майбутнє платіжних систем тут. Просунутих систем EPOS уже недостатньо у світі після COVID. Безконтактне самообслуговування забезпечує підвищену безпеку з меншими накладними витратами.

Для чого використовуються автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay

Автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay мають кілька переваг: від оплати оренди до купівлі квитків на улюблену групу, яка приїжджає до міста.

З точки зору бізнесу, такі системи, як автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay, зменшують витрати, одночасно підвищуючи ефективність. Уявіть, що більше не стоять довгі черги в барі, а замовлення надходять напряму та вже оплачені. Це має потенціал для значного збільшення обороту та гарантує отримання оплати за послуги.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Як вони використовуються

Автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay – це автоматизовані машини, які не вимагають відвідування. Завдяки цьому ви можете використовувати їх для будь-яких сценаріїв. Зазвичай автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay сплачують клієнтом за отримання товару.

Однак функціональність також поширюється на доставку. Розумні шафки Amazon є чудовим прикладом. Як кур'єрська служба, ви можете передати товар для отримання клієнтом, використовуючи безпечний унікальний пароль для доступу.

Інтерфейс користувача

Інтерфейс може відрізнитися залежно від типу автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. Як правило, більшість сучасних платіжних автоматів самообслуговування використовують сенсорний інтерфейс. Проте потреба в традиційних системах залишається.

Наприклад, банківські банкомати все ще пропонують такі ж чудові послуги, як і завжди, з орієнтованою на кнопку взаємодією. Проте оновлення стають стандартними, оскільки безконтактні платежі мають численні переваги .

Внутрішнє обладнання

Існує багато компонентів автомату самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. Хоча системи інтегровані складним чином, вони є простими пристроями введення та виведення. Насправді вони мало чим відрізняються від споживчих пристроїв, таких як ПК, смартфони та планшети. Стандартні інтерфейси включають клавіатури, введення PIN-коду та фіксовані сканери.

Як працює програмне забезпечення

Програмне забезпечення є життєво важливим для використання обладнання. Драйвери та спеціальні пакети, доступні через комплекти розробки

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

програмного забезпечення (SDK), дозволяють взаємодіяти з внутрішніми компонентами. SDK дозволяють швидко розгортати програмні системи з повнофункціональними спеціальними елементами, унікальними для вашої компанії.

Наприклад, більшість ресторанів потребують однакових систем, таких як апарати для напоїв із сенсорним екраном або автоматичне обслуговування столиків. Але доступні варіанти, ціни та спеціальні акції відрізнятимуться від закладу до закладу. Проте всі SDK пропонують зручні для користувача системи.

Різні типи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay

Існує багато типів автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. Зазвичай вони розроблені відповідно до призначеної функції та пропонують функції як з точки зору бізнесу, так і з точки зору клієнта.

Видача та отримання

Можливо, ваш бізнес не обов'язково працює цілодобово, без вихідних, і платити працівникам додаткові гроші за зміну просто неможливо. На щастя, автоматизованій машині для висадки та забору не потрібно їсти чи спати, і вона може обслуговувати клієнтів у будь-який час дня чи ночі.

Припустімо, що ви керуєте автостоянкою, і клієнту потрібно забрати свій автомобіль у неробочий час. У такому випадку кіоск видачі та видачі може в будь-який час керувати їхніми ключами від автомобіля. Або може статися так, що клієнтам потрібно перерахувати гроші на банківські рахунки у зручний для них час.

Автоматизовані квитки

Квитки є загальноприйнятою формою підтвердження оплати. Вони також можуть підтвердити ідентифікацію. Стандартним використанням може бути участь у приватних заходах або підтвердження оплати за паркування. Нюанси, необхідні для квитків, означають, що відстеження індивідуальної інформації

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

може бути виснажливим.

Проте, працюючи з серверною базою даних і надійним SDK, ви можете легко змінювати деталі та відстежувати замовлення, використовуючи унікальні посилання. Крім того, квиткові автомати значно полегшують увесь процес для клієнтів, які просто хочуть заплатити та отримати квитки легко.

Розумні шафки

Розумні шафки роблять революцію в кур'єрській галузі. Ви можете скористатися різноманітною безпекою унікальних кодів як для доставки, так і для отримання. Кур'єру більше не потрібно турбуватися про доставку за неправильною адресою, що спричинить проблеми з логістикою та скарги клієнтів.

Ви можете повністю інтегрувати системи в транзакції електронної комерції, щоб товари доставлялися безпосередньо в розумні шафки, або ви можете безпечно зберігати активи компанії для використання вашими співробітниками, використовуючи гібридний офіс.

Інтерактивні вивіски

Будь-який бізнес чи послуга завжди потребують точних методів комунікації, нової реклами та нових методів просування. Такі системи, як InterAct, можуть інтегрувати численні взаємодії, орієнтовані на клієнта.

Наприклад, цифрові вивіски можуть рекламувати ваш майбутній сценічний виступ. У той же час клієнти можуть взаємодіяти з вашою цифровою рекламою, щоб вибрати бронювання, оплатити та отримати квитки. По суті, цифрові інтерактивні вивіски пропонують комплексний досвід, оптимізуючи кілька послуг в одній, що полегшує роботу вам і вашим клієнтам.

Кіоски для оплати рахунків

Одним із найпоширеніших способів використання автомату самообслуговування на базі сенсорних терміналів платіжної системи EasyPay є рахунки. Клієнти можуть платити практично будь-що в безпечному автоматі самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Стандартне прийняття включає орендну плату та іпотечні платежі, навчання в коледжі та оплату комунальних послуг, таких як газ і електроенергія.

Багато з цих автоматизованих систем мають усе необхідне для безпечної та надійної транзакції. Пристрої включають сканери штрих-кодів, принтери квитанцій, зчитувачі дебетових карток і готівкові депозити. Це чудова зручність для клієнтів, які не мають доступу до онлайн-об'єктів.

Різні типи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay пропонують широкий спектр послуг. До популярних систем належать автомати видачі та видачі, розумні шафки та універсальні станції оплати рахунків.

Які галузі виграють від автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay?

Автомати самообслуговування на базі сенсорних терміналів платіжної системи EasyPay бувають різних форм і розмірів. Але, мабуть, найкориснішим є сенсорний екран. Платіжна система з сенсорним екраном корисна для різних комерційних галузей.

Ресторан і замовлення їжі

Схоже, що інтерфейси сенсорних екранів знайшли свій дім у ресторанах і секторі замовлення їжі. Великі бренди, такі як McDonald's, розгорнули ці практичні системи в усіх своїх ресторанах. Функціональність і простота використання автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay допомагають усім, хто бере участь, від оплати клієнта до доставки.

Крім того, банк із сенсорним екраном може заощадити гроші, зменшивши потребу в дорогих співробітниках, які потребують навчання та часто роблять помилки. Автоматизована система просто виконує свою роботу краще.

Банківський сектор

Технологічні вдосконалення безпосередньо сприяють безпеці банків для клієнтів і клієнтів. Тільки у Великій Британії кількість банківських крадіжок

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

скоротилася більш ніж на 90% порівняно з початком 1990-х років.

Крім безпеки, банківський сектор виграє від платіжних систем самообслуговування та сенсорних екранів як з точки зору клієнтів, так і з точки зору бізнесу. Наприклад, ви можете скористатися інтерфейсом банку, щоб вибрати послугу, яку можна виконати тут же. Або отримати талон із зазначенням часу для спрощення процесу очікування.

Роздрібна торгівля та покупки

Технології розповсюджуються в роздрібній торгівлі, щоб зробити покупки ще більш захоплюючими. Однією з основних причин є скорочення витрат на ведення бізнесу. Звичайно, початкові авансові інвестиції заощадять гроші згодом.

Але з точки зору клієнта, інтерфейс із сенсорним екраном пропонує легший спосіб знайти те, що він шукає, і новий спосіб перегляду ваших товарів. Щоб спростити весь процес покупки, ви можете надати системи продажу квитків для збору та додати пристрої для зчитування кредитних і дебетових карток на пристрій для оплати.

Державні послуги

Збалансування бухгалтерського обліку міста чи міста мало чим відрізняється від бізнесу. І витрати, пов'язані з виконанням громадських обов'язків, таких як HR, стають дорогими. Однак за допомогою автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay муніципальні послуги можуть бути автоматизованими та інтерактивними. І ви можете розгорнути технологію прямо в усьому державному секторі.

Такі види використання включають інформацію про громадську безпеку, інтерактивні карти, опитування щодо вдосконалення, маршрути надання послуг та інформацію про голосування. Тим не менш, ви також можете значно скоротити час очікування в державних установах, таких як ратуші та горезвісне DMV.

Готелі та туризм

За останнє десятиліття готельна та туристична індустрія повністю охопила автомати самообслуговування на базі сенсорних терміналів платіжної системи

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

EasyPay та сенсорні екрани. Від оригінальних готелів типу «ліжко та сніданок» до розкішних п'ятизіркових готелів – сенсорні екрани значно спрощують реєстрацію заїзду та виїзду. Це чудове доповнення до традиційного обслуговування, яке спрощує весь процес, щоб звільнити персонал і порадувати гостей.

Крім того, ви можете використовувати ці системи, щоб пропонувати додаткові послуги клієнтам, наприклад обслуговування номерів. З точки зору туризму, індивідуальне програмне забезпечення може забезпечити всі автоматизовані системи кількома мовами.

Пристрої з сенсорним екраном пропонують виняткову функціональність у всіх секторах. Однак деякі з найбільших переваг стосуються гостинності, комерційної роздрібною торгівлі, а також громадських і фінансових послуг.

Висновок

Платіжні системи самообслуговування використовуються для різноманітних послуг. Вони працюють за допомогою апаратного та програмного забезпечення, спрямованого на створення простих у використанні інтерфейсів з точки зору бізнесу та клієнта. Цілий ряд галузей, наприклад банківська справа та роздрібна торгівля, використовують таку технологію з різними кіосками. Різні типи включають автоматизовані системи продажу квитків та інтерактивні вивіски.

3.2 Розробка структурної схеми

На рисунку 3.1 показана структурна схема програмного забезпечення, розглянемо її зверху долілиць.

Структурна схема включає в себе 2 блоки:

- Апаратну частину.
- Програмну частину.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Апаратна частина

Комплектація:

- Вандалостійкий металевий корпус.
- Вандалостійкий сенсорний монітор (Acer 17", Kee Touch).
- Купюроприймач Cash Code SM с касетою на 1500 купюр / Cash Code MVU + мішок на 3000 купюр.
- Термопринтер Custom VKP80 II / Citizen 2000.
- 4G модем Siemens MC35i.
- Промисловий комп'ютер Intel D945GCLF Atom 2000MHz/1024Mb/320Gb/LAN100/Audio/Video.

Програмна частина

Вибір секції на сенсорному екрані, включає в себе оплату операторів різних послуг, серед них:

- оператори мобільного зв'язку;
- електронні платіжні системи;
- контент-провайдери й агрегатори;
- інтернет-провайдери, оператори послуг IP-телефонії, комерційного телебачення;
- компанії, що надають послуги ЖКГ і фіксованого зв'язку;
- інтернет-магазини;
- і інші.

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay – це новий сучасний інтерактивний кіоск, який чудово підходить для швидкого обслуговування та ресторанів, а також роздрібних магазинів. Він є природним розширенням існуючих POS-терміналів, допомагаючи збільшити продажі та прискорити потік клієнтів у години пік.

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay, розроблена з урахуванням функціональності та естетики, оснащена яскравим 32-дюймовим сенсорним екраном PCAP високої чіткості та гладким, міцним корпусом, що забезпечує сучасний та складний інтерфейс

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

користувача. Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay – це модульна конструкція, у якій можна розмістити різноманітні варіанти платіжних пристроїв. Модульна конструкція робить встановлення, обслуговування та заміну компонентів простими та зручними.

Більшість високонадійних компонентів у автоматі самообслуговування на базі сенсорних терміналів платіжної системи EasyPay виробляються материнською компанією на Тайвані. Компоненти відомі своїм довгим життєвим циклом і високою надійністю, що знижує витрати на обслуговування протягом усього терміну експлуатації. Інвентаризація та остаточна конфігурація виконуються в Україні для найшвидшого часу виконання. Унікальний ланцюг постачання цієї серії дозволяє нам забезпечувати виняткову якість за дуже привабливою ціною. Цю модульну модель автомату самообслуговування на базі сенсорних терміналів платіжної системи EasyPay можна розгорнути як настінну, одно- або двосторонню конфігурацію, що забезпечує максимальну гнучкість. Він добре підходить для підвищення продажів різноманітних готельних послуг, де клієнти можуть самостійно орієнтуватися в пропозиціях. Деякі з його популярних застосувань – показ реклами, замовлення та оплата та електронне бронювання. Він слугує бездоганним розширенням існуючих POS-терміналів, який підтримує персонал у пік сезону. За допомогою цього автомату самообслуговування на базі сенсорних терміналів платіжної системи EasyPay клієнти тепер можуть замовити собі масаж у спа-салоні, дізнатися розклад тренажерних залів і навіть знайти ресторани, доступні в готелі або поблизу. Це лише кілька прикладів того, як готелі можуть найкращим чином використовувати автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. Завжди є простір для налаштування, додавши більше периферійних пристроїв до пристрою, наприклад NFC, 2D-сканер, принтер чеків і багато іншого.

Універсальні функціональні можливості

Включення широкого спектру периферійних пристроїв у програми самообслуговування, включаючи 2D-сканер, принтер чеків, NFC, індикатор стану, кронштейн для платіжних пристроїв EMV тощо.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Апаратна частина

1. Вандалостійкий металевий корпус.
2. Вандалостійкий сенсорний монітор (Acer 17", Kee Touch).
3. Купюроприймач Cash Code SM с касетою на 1500 купюр / Cash Code MVU + мішок на 3000 купюр.
4. Термопринтер Custom VKP80 II / Citizen 2000.
5. 4G модем Siemens MC35i.
6. Промисловий комп'ютер:
 - Процесор (Intel D945GCLF Atom).
 - Материнська плата (VIA P4M800).
 - Модуль пам'яті (DDR III 1024Mb).
 - Сторожовий таймер.
 - Блок живлення (300W).
 - Накопичувач (HDD WD 320Gb).
 - Кабель (монітор-комп'ютер)
 - Блок живлення.

Програмна частина



Рисунок 3.1 – Структурна схема системи

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРБ-123.24.0005.00.00.ПЗ

Арк.

31

Стильний, сучасний і створений для інтуїтивно зрозумілого інтерактивного використання

Автомат самообслуговування на базі сенсорних терміналів платіжної системи EasyPay, розроблена з урахуванням як функціональності, так і естетики, має яскравий 32-дюймовий сенсорний екран PCAP високої чіткості в гладкому міцному корпусі, що забезпечує сучасний і складний автоматизований інтерфейс користувача.

Модульний для зручності обслуговування

Модульна конструкція робить установку, технічне обслуговування та заміну компонентів простими, швидкими та зручними, а також прискорює повернення інвестицій за рахунок зниження вартості ремонту.

Гнучка конфігурація

Варіанти кріплення включають настінне кріплення, односторонню або двосторонню роботу на підставці, що забезпечує гнучке розгортання

3.3 Розробка функціональної схеми

На рисунку 3.2 показана функціональна схема роботи системи. У ній можна чітко простежити основні компоненти терміналу. Як видно з рисунка користувач сприймає термінал як 3 основні компоненти: сенсорний екран; купюроприймач; монетоприймач (є не у всіх версіях). Адміністраторові доступні всі налаштування й дії стосовно автомата при наявності спеціального ключа й знання спеціального системного номера, що вводить апарат у режим налаштування.

Кінцева дія апарата видача чека поповнення мобільного рахунку є одним з найважливіших компонентів терміналу. Чек є юридичним папером і при виникненні проблем з ним можна звертатися в різні інстанції.

Програмне забезпечення повинне забезпечувати наступні функції:

1. Первісне налаштування.
2. Сервісне обслуговування автоматів.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

3. Установка й налаштування таймера.
4. Налаштування додаткових параметрів через командний рядок.
5. Моніторинг функціонування терміналів.
6. Робочий режим.



Рисунок 3.2 – Функціональна схема системи

Для коректної роботи ПЗ необхідно, щоб на автоматі самообслуговування було попередньо встановлене наступне програмне забезпечення:

- Операційна система WINDOWS 10/11.

– EDGE.

Крім того, перед початком роботи треба переконатися, що в операційній системі автомата настроєне з'єднання з Інтернет. У якості Інтернет-з'єднання може використовуватися як канал 4G, так і будь-який інший доступний спосіб з'єднання. Для доступу в Інтернет ПЗ використовує налаштування Internet Explorer, тому у випадку використання проху-сервера, при підключенні автомата в локальну мережу, необхідно додати проху у налаштування «Internet Explorer».

Розглянемо перераховані вище функції більш докладно.

1. Первісне налаштування.

При першому запуску системи вам необхідно налаштувати дані для авторизації автомата в системі оплати послуг мобільних операторів. Спершу треба перейти в параметри авторизації, тобто ввести значення параметрів:

– Номер терміналу – ідентифікатор терміналу в системі оплати послуг у автоматі самообслуговування на базі сенсорних терміналів платіжної системи EasyPay (EasyPay). У модулі адміністратора системи EasyPay повинен бути зареєстрований термінал, під яким будуть проводитися платежі. Терміналу повинен бути заданий тип Автомат EasyPay. Неприпустиме використання того самого номера терміналу декількома пристроями. Для кожного автомата, реєструйте в системі новий термінал:

– Ім'я користувача – логін користувача, від імені якого будуть відбуватися платежі.

– Пароль – пароль користувача, від імені якого будуть відбуватися платежі.

– Підтвердження пароля – у дане поле необхідно повторно ввести пароль користувача для підтвердження його коректності.

У якості логіна й пароля повинні використовуватися дані персони, зареєстрованої в системі EasyPay із правами «Продавець». Припустимо використання однакових даних користувача в різних автоматах.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Після цього необхідно налаштувати параметри сервісного доступу автомата, тобто ввести налаштування для входу в режим обслуговування й інкасації автомата:

– Секретний номер телефону – задається номер телефону, що дозволяє перевести автомат у сервісний режим і потрапити в панель налаштування.

– Ім'я користувача – логін користувача, що буде здійснювати сервісне обслуговування автомата.

– Пароль – пароль користувача, що буде здійснювати сервісне обслуговування автомата.

– Підтвердження пароля – у дане поле необхідно повторно ввести пароль користувача для підтвердження його коректності.

Користувач, під ім'ям якого буде здійснюватися сервісне обслуговування автомата не повинен бути зареєстрований у системі EasyPay. Тому, для логіна й пароля ви можете використовувати будь-яку комбінацію символів. Реєструвати в EasyPay даного користувача не потрібно.

Після цього етапу відбувається налаштування системи Параметри з'єднання з Інтернет і опції автомата:

– З'єднання з Інтернет – у даному полі прапором відзначте те з'єднання, через яке автомат буде здійснювати комунікацію із системою EasyPay.

– Задати рядок ініціалізації модему – дозволяє вказати нестандартний рядок ініціалізації модему для поточного обраного з'єднання.

– Включити стиск даних, переданих через Інтернет – дозволяє включити додатковий стиск даних, переданих через Інтернет з метою економії трафіку.

– Включити мультиканальний режим – включення мультиканального режиму.

– Опції автомата – панель додаткових опцій автомата:

– Зупиняти автомат при помилках купюроприймача – даний прапор дозволяє припинити прийом платежів при виявленні яких-небудь проблем у роботі купюроприймача.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

– Зупиняти автомат при помилках принтера – даний прапор дозволяє припинити прийом платежів при виявленні яких-небудь проблем у роботі принтера чеків. При знятому прапорі автомат буде продовжувати приймати платежі, навіть якщо в принтері закінчився папір для печаті чеків.

– Зупиняти автомат при відсутності засобів на рахунку агента – даний прапор дозволяє припинити прийом платежів на автоматі при відсутності засобів в агента.

– Зупиняти автомат при відсутності зв'язку – даний прапор дозволяє припинити прийом платежів при відсутності зв'язку з автоматом протягом зазначеного періоду часу (за замовчуванням, 15 минут).

– Не перезавантажувати програмно модем у випадку відсутності сторожового таймера – відзначте прапор, щоб відключити опцію програмного перезавантаження модему у випадку відсутності сторожового таймера.

Після цього автомат переходить у робочий режим. Якщо в автомат не завантажені файли інтерфейсу, автомат скачує їх автоматично із сервера EasyPay. Даний процес може зайняти досить тривалий час (хвилин 30-40 при використанні 4G з'єднання), при цьому на екрані буде показане повідомлення «Вибачте, автомат тимчасово не працює». У робочому режимі в деяких випадках можлива поява наступного напису: «Вибачте, автомат тимчасово не працює» Даний напис означає, що були виявлені проблеми з купюроприймачом або принтером чеків.

2. Сервісне обслуговування автоматів.

Воно складається з наступних підрозділів:

- Перехід у режим обслуговування й інкасації.
- Сервісне меню.
- Зміна провайдеру 4G зв'язку.
- Мультиканальний режим роботи модему.

Розглянемо їх більш докладно.

Перехід у режим обслуговування й інкасації. Для цього необхідно вибрати опцію Оплата послуг, ввести номер телефону, що задали при налаштуванні

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

автомата й розблокувати сервісний режим за допомогою імені користувача й пароля.

Сервісне меню. Містить такі дані як:

- Версія інтерфейсу.
- Кількість купюр у купюроприймачі й суму.
- Статус купюроприймача.
- Статус принтера.
- Статус з'єднання.
- Статус сторожового таймера.
- Статус платежів.

За допомогою кнопок у вікні сервісного меню можна виконувати також наступні дії:

- Змінити номер терміналу, логін і пароль EasyPay.
- Змінити параметри входу в сервісний режим.
- Змінити параметри Інтернет і опції автомата.
- Налаштування інтерфейсу.
- Налаштування e-mail оповіщень.
- Налаштування Multi SIM підключення.
- Видалити файл конфігурації й запустити знову програму.
- Переглянути лог.
- Запустити відновлення.
- Видалити файл конфігурації й запустити знову програму.
- Вийти й запустити знову програму.
- Налаштувати параметри безпеки при роботі з додатком.

Крім цього, дозволяє виконувати й контролювати наступні дії:

- Проведення інкасації.
- Налаштування параметрів принтера.
- Рівень сигналу GSM мережі.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

- Налаштування одержання балансу.
- Налаштування оповіщень.
- Налаштування параметрів сторожового таймера.
- Налаштування інтерфейсу.
- Режим підтримки двох SIM-карт.
- Безпека.
- Режим блокування системних комбінацій клавіатури.

Зміна провайдеру 4G зв'язку. Призначено для введення параметрів передачі даних по мережі GSM, операторів, які не прописані в автоматі EasyPay.

Мультиканальний режим роботи модему. Даний режим дозволяє здійснювати одночасно наступні дії:

- передача даних по Інтернет;
- відправлення SMS;
- одержання рівня сигналу й інших даних про стан GSM мережі;
- відправлення USSD запитів балансу SIM карти.

Таким чином, використання даного режиму дозволяє, не розриваючи модемне з'єднання з Інтернет, одержувати безупинно дані про баланс SIM карти, коливання рівня сигналу й дані про стан GSM мережі. Підключення мультиканального режиму відбувається в кілька етапів:

- Установка драйвера для мультиканального режиму.
- Установка стандартного драйвера модему.
- Створення модемного з'єднання.
- Налаштування ПЗ автомата.

3. Установка й налаштування таймера.

Таймер дозволяє здійснювати дві основні функції:

– Перезавантаження модему, при відсутності відгуку від модему протягом деякого часу. Даний режим роботи включений за замовчуванням.

– Перезавантаження комп'ютера при відсутності сигналу від комп'ютера протягом 30 минут.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Таким чином, за допомогою плати відслідковується «зависання» комп'ютера або модему й коректуються дані проблеми шляхом перезавантаження модему або комп'ютера. Період в 30 мінут для перезавантаження комп'ютера обраний з метою виключення ситуацій помилкового спрацьовування: при відновленні даних із сервера й т.п.

4. Налаштування додаткових параметрів через командний рядок.

Зо допомогою командного рядка ви можете вказати параметри, відмінні від параметрів за замовчуванням для принтера й купюроприймача. Вони підрозділяються на:

- Параметри принтера й купюроприймача.
- Додаткові параметри.

5. Моніторинг функціонування терміналів.

Для зручності відстеження роботи ваших автоматів самообслуговування, у системі EasyPay передбачена можливість віддаленого моніторингу в реальному часі. Дана функціональність доступна користувачам із правами головного менеджера, і містить у собі:

- Моніторинг через Інтернет.
- Моніторинг за допомогою мобільного телефону.

6. Робочий режим.

Його робота полягає у виконанні трьох основних функцій:

- Прийом платежів. Інтерфейс для прийому платежів розроблений таким чином, щоб максимально спростити процес звернення клієнта до автомату і зробити прийом платежів за допомогою автомата інтуїтивно зрозумілим без додаткових інструкцій.

- Відновлення ПЗ. Фахівці компанії ОСМП постійно працюють над удосконаленням програмного забезпечення автоматів самообслуговування. Після виходу нової версії ПЗ, автомат автоматично завантажує відновлення, не припиняючи роботу. Також автоматично відбувається відновлення зовнішнього

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

вигляду інтерфейсу автомата: додавання можливості оплати послуг нових провайдерів і т.п.

– Режим автоматичного включення автомата. Призначений для того, щоб налаштувати автоматичне включення автомата після збоїв електрики в мережі.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3. Після початку роботи ПЗ ми потрапляємо до основного блоку ПЗ, звідкіля через обробник помилок ми можемо потрапити до інтерфейсу користувача чи інтерфейсу адміністратора. Це залежить від прав доступу користувача терміналу.



Рисунок 3.3 – Діаграма взаємодії процесів

При роботі з інтерфейсом користувача проходить вибір оператора оплати та обробка запитів користувача. Якщо є повноваження роботи з інтерфейсом адміністратора, ми можемо проводити роботу зі всіма функціями терміналу, через виклик відповідних API функцій опцій терміналу. Крім цього проводити налаштування параметрів інкасації та налагодження терміналу.

КБПЗ_2024

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Вся блок-схема складається з виконання основного алгоритму розробленого ПЗ (рис 4.1) та виконання підпрограм обробки запиту користувача (рис 4.2) та підпрограм обробки запиту адміністратора (рис 4.3).

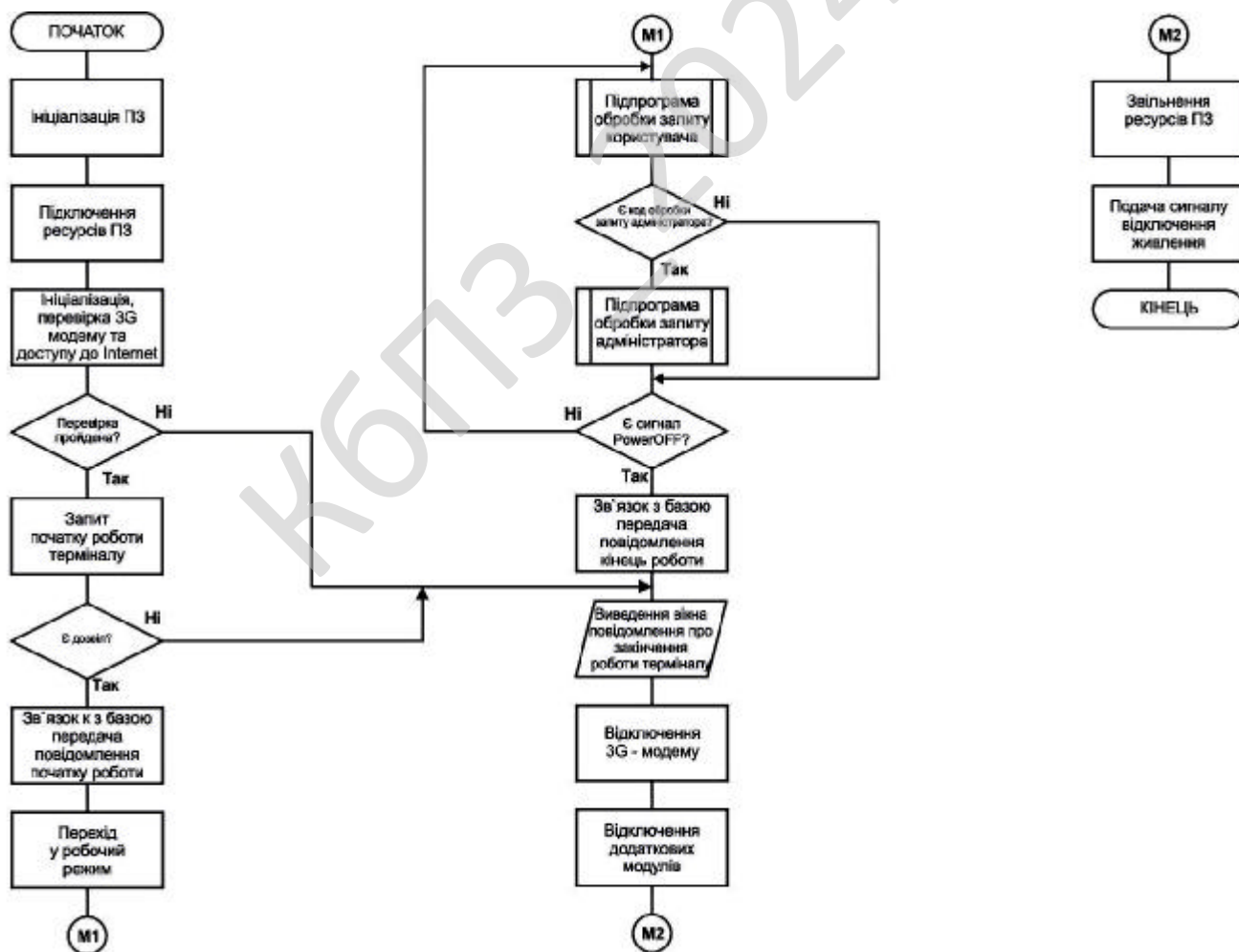


Рисунок 4.1 – Блок-схема основної програми

Розглянемо детально як працює розроблене ПЗ покроково. В першу чергу проходить ініціалізація ПЗ, підключення ресурсів ПЗ, та ініціалізація, перевірка 3G модему та доступу до Internet. Після чого перевіряється результат та при позитивному проходженні тесту проходить запит початку роботи терміналу і при наявності дозволу проводиться зв'язок к з базою і передача повідомлення початку роботи.

Далі проходить перехід у робочий режим з викликом підпрограми (рис. 4.2) обробки запиту користувача. Далі при введенні коду обробки запиту адміністратора проходить виконання підпрограми обробки запиту адміністратора (рис. 4.3). Далі якщо поступає сигнал PowerOFF проводиться зв'язок з базою з передачею повідомлення кінця роботи терміналу. Далі проводиться виведення вікна повідомлення про закінчення роботи терміналу, відключення 3G-модему та відключення додаткових модулів з звільненням ресурсів ПЗ і подачею сигналу відключення живлення.

Опис протоколу взаємодії.

Розглянемо по крокам розроблену систему та методику оплати послуг мобільного телефону.

Повна реалізація платіжного шлюзу повинна забезпечувати обробку наступних запитів від терміналу:

- Перевірка можливості платежу з певними реквізитами (Check).
- Проведення платежу з певним реквізитами (Pay).
- Перевірка стану платежу (Status).

Check

Перевірка можливості здійснення платежу виконується терміналом після збору необхідної інформації безпосередньо перед прийомом грошей. Термінал генерує ідентифікатор сесії (випадкова величина) і передає її разом із запитом. Якщо в системі сесія з даним номером не закріплена за іншим терміналом і може використовуватися, то шлюз виконує перевірку можливості проведення платежу і при успішному завершенні створює сесію з вказаним номером.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Ідентифікатор сесії використовується для проведення платежу і перевірки статусу платежу.

Рау

Здійснення платежу виконується терміналом після прийому грошей. При проведенні платежу використовується той же ідентифікатор сесії, який був отриманий на етапі перевірки можливості платежу. У випадку помилки виконання платежу формується «зліпок» платежу і після деякого проміжку часу буде зроблена спроба повторного платежу.

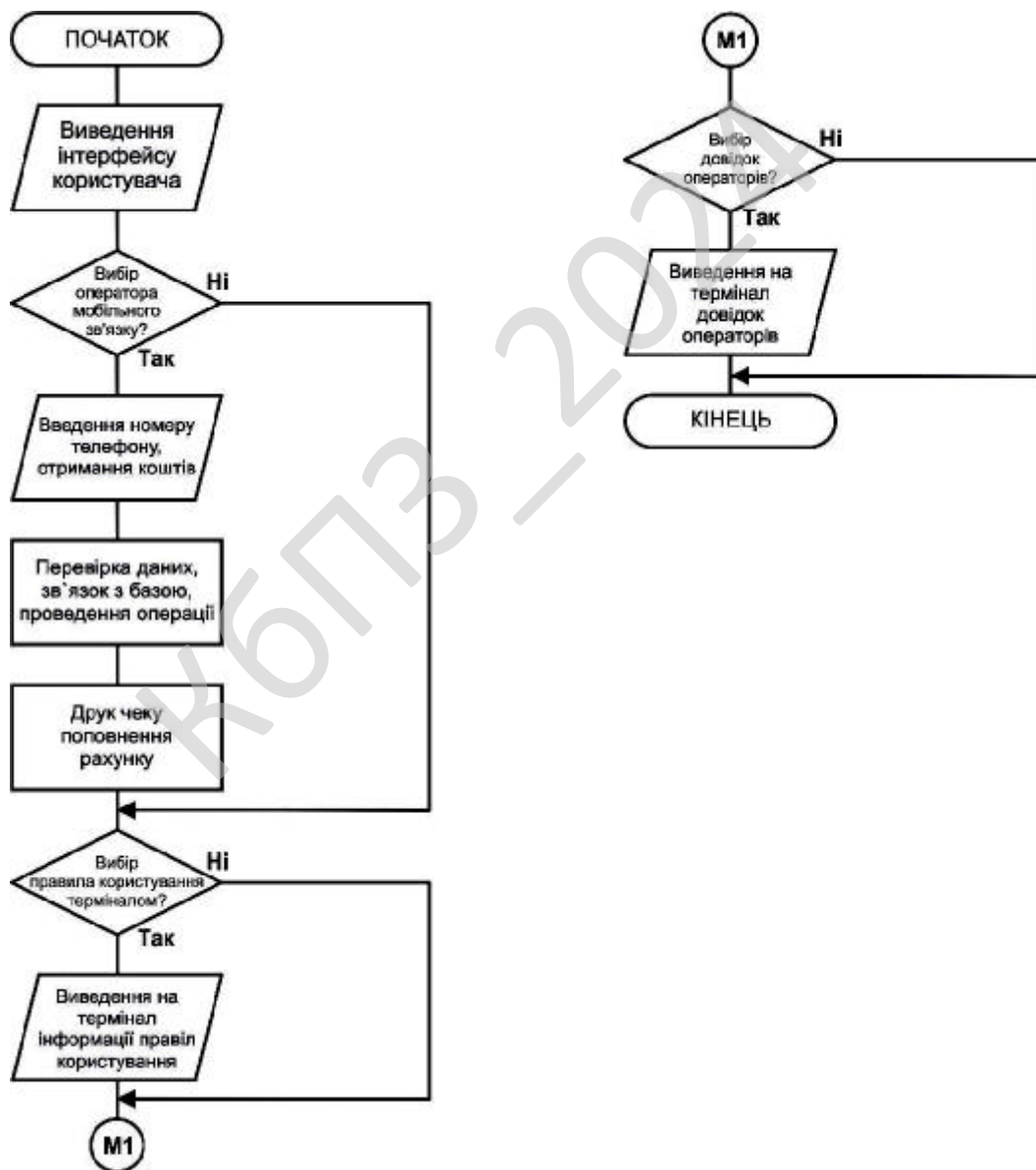


Рисунок 4.2 – Блок-схема підпрограми обробки запиту користувача

Status

Статус платежу запитується терміналом перед спробою повторної відсилання платежу в платіжний шлюз. Повторна досилання буде здійснюватися тоді і тільки тоді, коли запит на статус привів до відповіді шлюзу, що можливість проведення платежу перевірена (виконана операція Check), а запит на платіж не виконаний (операція Pay не виконана або завершилася з помилкою).

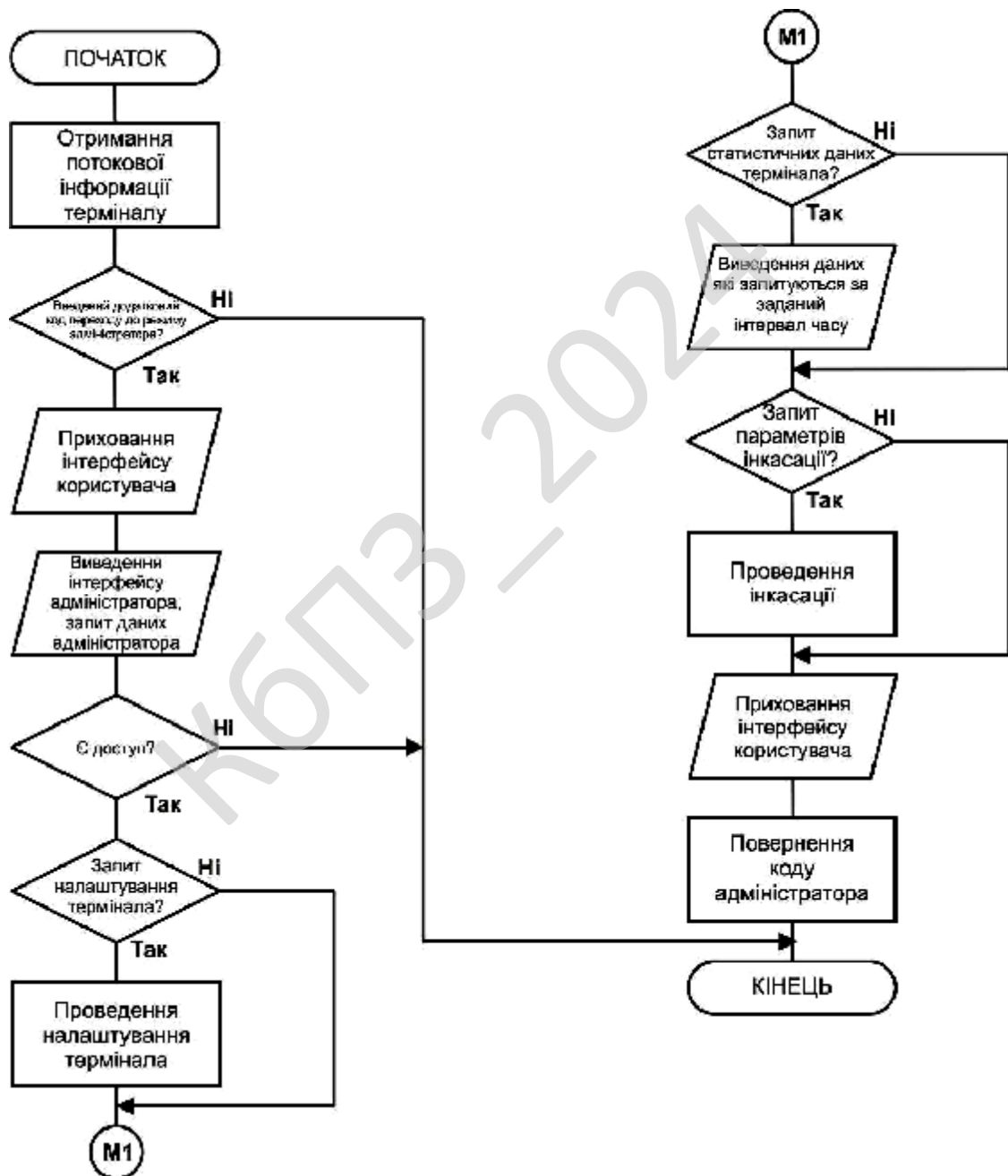


Рисунок 4.3 – Блок-схема підпрограми обробки запиту адміністратора

Формат передачі даних

Для обміну інформацією з шлюзом використовується протокол HTTP або HTTPS (з використанням SSL).

Параметри платежу передаються методом POST у вигляді параметрів. Відповідь шлюзу, являє собою аналогічним чином оформлений список параметрів безпосередньо в тілі документа.

Параметр запиту Content-Length вказується відповідно до реальної довжиною повідомлення. Стандартні параметри, автоматично передаються з кожним запитом.

Стандартні параметри запиту

SessionID (рядок до 32 символів) – ідентифікатор сесії (створений терміналом при операції Check або раніше збережений при інших операціях).

MachineMark (рядок до 32 символів) – ідентифікатор терміналу, з якого здійснюється платіж (константа, обумовлена в налаштуваннях)ю

ClearingNumber (рядок до 32 символів) – Номер для взаєморозрахунків для забезпечення поділу груп терміналів на платіжному шлюзі (константа, обумовлена в налаштуваннях)

Hashcode (рядок) – хеш-код всіх значень передаваних з терміналу і константи, визначеної в налаштуваннях. Для формування хеш-коду використовується алгоритм SHA-1. Результуючий бінарний масив кодується з використанням Base64.

Стандартні параметри відповіді

SessionID (рядок до 32 символів) – ідентифікатор сесії (створений терміналом при операції Check або раніше збережений при інших операціях).

ErrorCode (ціле) – результат виконання операції. Зумовлені коди наведені нижче.

ClearingNumber (рядок до 32 символів) – номер для взаєморозрахунків для забезпечення поділу груп терміналів на платіжному шлюзі (константа, обумовлена в налаштуваннях)

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Hashcode (рядок) – хеш-код всіх значень передаваних з терміналу і константи, визначеної в налаштуваннях. Для формування хеш-коду використовується алгоритм SHA-1. Результуючий бінарний масив кодується з використанням Base64.

Стандартні коди помилок

Код помилки 0 (операція завершилася успішно) – повертається шлюзом для всіх запитів, обробка яких повністю виконалася.

Код помилки 1 (сесія вже існує) – повертається шлюзом в тому випадку, якщо в запиті на перевірку можливості платежу переданий номер сесії (SessionID), який вже використовувався раніше.

Код помилки 2 (сесія не знайдена) – повертається шлюзом в тому випадку, якщо зазначений у запиті на платіж або перевірку стану платежу номер сесії (SessionID) не визначений в системі (з використанням запиту Check).

Код помилки 3 (термінал не зареєстрований) – повертається шлюзом в тому випадку, якщо переданий ідентифікатор терміналу (MachineMark) не виявлений в базі даних платіжного шлюзу.

Код помилки 4 (контрагент не зареєстрований) – повертається шлюзом в тому випадку, якщо переданий ідентифікатор для взаєморозрахунків (ClearingNumber) не виявлений в базі даних платіжного шлюзу.

Код помилки 5 (термінал не відповідає контрагенту) – повертається шлюзом в тому випадку, якщо термінал, відповідний переданому ідентифікатору (MachineMark) належить контрагенту, відмінному від відповідного ідентифікатору (ClearingNumber).

Код помилки 6 (термінал заблокований) – повертається шлюзом в тому випадку, якщо обробка запитів від зазначеного терміналу припинена (по будь-якої причини).

Код помилки 7 (недостатньо коштів) – повертається шлюзом в тому випадку, якщо при проведенні платежу не вистачило суми депозиту для його виконання.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Код помилки 8 (помилка хеш-коду) – повертається шлюзом при неспівпаданні переданого і розрахованого хеш-коду для запиту. Ймовірно невідповідність секретної рядки на терміналі і на сервері.

Код помилки 9 (помилка запиту) – повертається шлюзом при неможливості розібрати запит (не вказані необхідні параметри).

Код помилки 10 (загальна помилка шлюзу) – повертається шлюзом при загальній непрацездатності (немає зв'язку з БД, некоректна конфігурація і т.п.)

Код помилки 11 (шлюз одержувача недоступний) – повертається шлюзом в тому випадку, якщо шлюз одержувача недоступний.

Код помилки 12 (параметри платежу невірні) – повертається шлюзом в тому випадку, якщо шлюз одержувача відмовив у прийнятті платежу із зазначеними параметрами (наприклад, невірний номер телефону, неіснуючий особовий рахунок і т.п.).

Код помилки 13-99 (зарезервовані) – коди зарезервовані для можливого використання. Реалізовані користувачем шлюзи повинні повертати специфічні коди помилок починаючи з коду 100.

Код помилки 100-1000 (користувальницькі коди помилок).

Check

При виконанні запиту на перевірку передаються всі визначені поля інтерфейсу, налаштовані в конфігурації терміналу у відповідності з відображенням полів на параметри запиту.

При отриманні відповіді всі додаткові параметри розбираються у відповідності з налаштуваннями і додаються для відображення в інтерфейсі терміналу.

Запит на перевірку можливості платежу:

<MachineMark>, <ClearingNumber>, 123456, <MachineSecret>

Тут в якості додаткового параметра передається номер особового рахунку (AccountNumber).

Відповідь на запит (успіх):

0, <ClearingNumber>, user1, <MachineSecret>

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Тут в якості відповіді (властивості для відображення) клієнту передається параметр Login, який після перевірки буде передано в інтерфейс.

Відповідь на запит (помилка):

```
SessionID = AAAAAAAAAA & ErrorCode = 12 & ClearingNumber = <ClearingNumber>
& Hashcode = SHA-1 (AAAAAAAAAA, 12, <ClearingNumber>, <MachineSecret>)
```

Тут була отримана відповідь з кодом помилки 12 («Параметри платежу невірні»). Важливо, що додаткові параметри (як «Login» з попереднього прикладу) у разі одержання повідомлення про помилку не розбираються клієнтом.

Pay

При виконанні запиту на проведення платежу передаються всі визначені поля інтерфейсу, налаштовані в конфігурації терміналу у відповідності з відображенням полів на параметри запиту. Параметри відповіді на запит не розбираються і не використовуються.

При проведенні платежу доцільно передавати суму платежу до зарахування. Рекомендується визначати цю властивість з назвою Amount.

Запит на проведення платежу:

```
<MachineMark>, <ClearingNumber>, "123456", "user1", "100.0"
<MachineSecret>)
```

Відповідь на запит (успіх і помилка)

```
SessionID = AAAAAAAAAA & ErrorCode = <ErrorCode> & ClearingNumber =
<ClearingNumber> & Hashcode = SHA-1 (AAAAAAAAAA, <ErrorCode>, <ClearingNumber>,
<MachineSecret>)
```

Status

При запиті на стан платежу передаються лише стандартні параметри. З відповіді обробляється код помилки для прийняття рішення про необхідність виконання платежу.

Дії в залежності від коду помилки

Код 0 (операція завершилася успішно) – платіж вже проведений, але інформація про це не надійшла ПЗ терміналу. Відкладений платіж скасовується.

Код 1 (сесія вже існує) – сесія створена і існує можливість проведення платежу. ПЗ терміналу виконає спробу повторного проведення платежу.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Код 3-1000 (інші помилки) – перевірка стану в поточний момент неможлива. Спроба проведення платежу відкладається.

Запит на стан платежу:

```
SessionID=AAAAAAAA&MachineMark=<MachineMark>&ClearingNumber=<ClearingNumber>&HashCode=SHA-1(AAAAAAAAA, <MachineMark>, <ClearingNumber>, <MachineSecret>)
```

Відповідь на запит (успіх і помилка):

```
SessionID=AAAAAAAA&ErrorCode=<ErrorCode>&ClearingNumber=<ClearingNumber>&HashCode=SHA-1(AAAAAAAAA, <ErrorCode>, <ClearingNumber>, <MachineSecret>)
```

Для забезпечення роботи термінала також був розроблений API інтерфейс із врахуванням особливостей програмування сенсорних панелей.

Відповідно до існуючих стандартів функції іменуються та виконують відповідні дії: CloseDevice; CreateDevice; DoCalibration; DoDrawTest; DoLinearization; EnableAutoRightButton; EnableConstTouch; EnableTouch; EnumerateTouchInstance; GetCalInfo; GetConstTouchInfo; GetEdgeInfo; GetLinzInfo; GetMonitorInfo32; GetMouseMode; GetTouchBeepInfo; RegisterDeviceRemovalHook; RegisterTPNotifyMessage; RestartFirmware; SetCalInfo; SetConstTouchInfo; SetEdgeInfo; SetConstTouchInfo; SetLinzInfo; SetMonitorInfo32; SetMouseMode; SetTouchBeepInfo; StartDeviceThread; StopDeviceThread; UpdateControllerCalInfo; UpdateControllerLinzInfo; VerifyEEPROM.

Структура та принцип роботи з модулем отримання довідкових даних.

У модулі даних застосовуються збережені довідники, які періодично оновлюються з «сервера системи», для швидкого доступу до даних без запиту даних з «сервера системи» при використанні користувачем.

Для забезпечення економії трафіку і зменшення обсягу передачі даних, відбувається завантаження тільки тих довідників, в яких є зміни.

Запит на отримання інформації по довіднику:

```
<? Xml version = "1.0" encoding = "utf-8"?>  
<request>  
<protocol-version> 1.00 </ protocol-version> // версія протоколу обміну  
  <extra name="id"> 1 </ extra> // ID платіжної системи
```

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

```
<extra name="password-md5"> 72e10b083b0d3423122f1d40a64793f6 </ extra>
// хеш пароля *
<request-type> 4 </ request-type>
// тип запиту * (4 - отримання інформації по довіднику)
<table-name> categories </ table-name> // ім'я таблиці (довідника)
</ Request>
```

Авторизація тут і далі можлива також з передачею пароля у відкритому вигляді. Тоді замість <extra name="password-md5"> використовується <extra name="password ">

Відповідь сервера з інформацією по довіднику:

```
<? Xml version = "1.0" encoding = "utf-8"?>
<response>
  <protocol-version> 1.00 </ protocol-version> // версія протоколу обміну
  <request-type> 4 </ request-type>
  // тип запиту * (4 - отримання інформації про довіднику)
  <result-code> 0 </ result-code>
  // 0 - успішно
  <table-name> categories </ table-name>
  // ім'я таблиці (довідника)
  <table-version> 2 </ table-version> // версія (реліз) поточної таблиці
  // Дата і час зміни поточної таблиці у форматі Timestamp
  <date> 123456789 </ date>
  <rows> 38 </ rows>
  // число записів у таблиці
  <table>
  // структура таблиці
    <row>
      <id> </ id>
      <parent_id> </ parent_id>
      <name_ru> </ name_ru>
    </ Row>
  </ Response>
```

Запит на отримання довідника:

```
<? Xml version = "1.0" encoding = "utf-8"?>
<request>
  <protocol-version> 1.00 </ protocol-version>
  // версія протоколу обміну
  <extra name="id"> 1 </ extra>
  // ID платіжної системи
  <extra name="password-md5"> 72e10b083b0d3423122f1d40a64793f6 </ extra>
```

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

```
// хеш пароля
  <request-type> 5 </ request-type>
// тип запиту (5 - отримання довідника)
  <table-name> categories </ table-name>
// ім'я таблиці (довідника)
</ Request>
```

Відповідь сервера з вмістом довідника:

```
<? Xml version = "1.0" encoding = "utf-8"?>
<response>
  <protocol-version> 1.00 </ protocol-version>
// версія протоколу обміну
  <request-type> 5 </ request-type>
// тип запиту (5 - отримання довідника)
  <result-code> 0 </ result-code>
// 0 - успішно
  <table-name> categories </ table-name>
// ім'я таблиці (довідника)
  <table-version> 2 </ table-version>
// версія (реліз) поточної таблиці
// Дата і час зміни поточної таблиці у форматі Timestamp
  <date> 123456789 </ date> <rows> 38 </ rows>
// число записів у таблиці
  <table>
    <row>
      <id> 1 </ id>
      <parent_id> 0 </ parent_id>
      <name_ru> A</ name_ru>
    </ Row>
    ...
  </ Table>
</ Response>
```

Запит на отримання загальної інформації:

```
<? Xml version = "1.0" encoding = "utf-8"?>
<request>
  <protocol-version> 1.00 </ protocol-version>
// версія протоколу обміну
  <extra name="id"> 1 </ extra>
// ID платіжної системи
  <extra name="password-md5"> 72e10b083b0d3423122f1d40a64793f6 </ extra>
// хеш пароля
  <request-type> 6 </ request-type>
```

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

// тип запиту (6 - отримання загальної інформації)
</ Request>

Відповідь сервера з вмістом загальної інформації:
<? Xml version = "1.0" encoding = "utf-8"?>
<response>
  <protocol-version> 1.00 </ protocol-version>
  // версія протоколу обміну
  <request-type> 6 </ request-type>
  // тип запиту (6 - отримання загальної інформації)
  <result-code> 0 </ result-code>
  // 0 - успішно
  <date> 123456789 </ date>
  // дата і час поточної інформації у форматі Timestamp
  <rows> 3 </ rows>
  // число записів
  <table>
  ...
  </ Table>
</ Response>

```

4.2 Захист розробленого програмного забезпечення

Дані у програмному забезпеченні я захищаю за допомогою MISTY1. MISTY1 – блоковий алгоритм шифрування, створений для компанії Mitsubishi Electric криптологом Міцуро Мацуї. Назва є аббревіатурою Mitsubishi Improved Security Technology. Алгоритм був розроблений в 1995-1996 рр. Відомі також дві модифікації алгоритму MISTY1: MISTY2 і KASUMI

Шифр став переможцем на Європейському конкурсі NESSIE. У результаті аналізу алгоритму експерти зробили вивід, що ніяких серйозних уразливостей даний алгоритм не має (переважно, завдяки вкладеним мережам Фейстеля, що суттєво утрудняє криптоаналіз). У нього високий запас криптостійкості, алгоритм має високу швидкість шифрування й досить ефективний для апаратної реалізації.

Алгоритм був розроблений на основі теорії «підтвердженої безпеки» проти диференціального й лінійного криптоаналізу. Цей алгоритм був спроектований, щоб протистояти криптоатакам, відомим на момент створення.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

З моменту публікації MISTY1 було проведено багато досліджень, щоб оцінити його рівень безпеки.

Диференціальний і неможливий диференціальний криптоаналіз високого порядку ефективно застосовується до блокових шифрів з малим ступенем. Найкращі результати для обох варіантів були отримані для 5-рівневого алгоритму MISTY1 без FL функцій.

Саме FL функції й широкобітні AND/OR операції в сильно утрудняють використання диференціального криптоаналізу, що не заважає проведенню в цьому напрямку всі нових досліджень і досягненню усе більш близьких до розв'язку результатів.

Параметри вихідних даних

MISTY1 – це шифр на основі вкладених мереж Фейстеля з вар'юємим числом раундів. Рекомендоване використання 8-раундової версії, але може використовуватися будь-яка кількість раундів, кратне 4-м. Розмір блоку вихідного тексту – 64 біта, розмір ключа – 128 біт.

Для роботи алгоритму також попередньо виконується процедура розширення ключа, яка для 8-мі раундів обчислює 1216 бітів ключової інформації з 128-бітного ключа шифрування.

Структура алгоритму

Для задоволення вимогам конкурсу NESSIE, а також для задоволення завдання мультиплатформеності, в алгоритмі MISTY1 використовувалися наступні методи шифрування:

- Логічні операції.
- Арифметичні операції.
- Операції зрушення.
- Таблиці перестановок.

Як говорилося вище, алгоритм MISTY1 заснований на «вкладених» мережах Фейстеля. Спочатку блок вихідного тексту розбивається на два 32-бітних субблоки, після чого виконується r раундів наступних перетворень[1]:

- У кожному непарному раунді обоє субблоки обробляються операцією FL
- Над оброблюваним субблоком виконується операція FO.
- Результат цих операцій накладається логічною операцією «, що виключає або» (XOR) на неопрацьований субблок.
- Субблоки міняються місцями. Після заключного раунду обоє субблоки ще раз обробляються операцією FL.

Операція FL

Оброблюваний 32-бітний субблок розбивається на два 16-бітних фрагмента, до яких застосовуються операції, де:

- L і R – вхідні значення лівого й правого фрагментів відповідно;
- L' і R' – вихідні значення;
- i – фрагменти j-го підключа i-го раунду для функції FL (процедура розширення ключа докладно описана далі);
- i – побітві логічні операції «і» і «або» відповідно.

Операція FO

Саме ця функція є вкладеною мережею Фейстеля. Тут, як і раніше, виконується розбивка вхідного значення на два 16-бітних фрагмента, що проходять 3 раунду наступних перетворень:

- На лівий фрагмент операцією XOR накладається фрагмент ключа, де k – номер раунду функції FO.
- Лівий фрагмент обробляється операцією FI.
- На лівий фрагмент накладається операцією XOR значення правого фрагмента.
- Фрагменти міняються місцями.

Після третього раунду операції FO на лівий фрагмент накладається операцією XOR додатковий фрагмент ключа.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

2. Формуються значення: у якості використовується результат обробки значення функцією FI, яка в якості ключа (тобто сукупності необхідних 7- і 9-бітного фрагментів) використовує значення(якщо індекс n фрагмента ключа перевищує 8, то замість нього використовується індекс n-8).

Необхідні фрагменти розширеного ключа «набираються» у міру виконання перетворень із відповідних масивів і згідно з відповідними таблицями 16-бітний фрагмент ділиться на 7-бітний фрагмент і 9-бітний .

Розшифрування

Розшифрування проводиться виконанням тих же операцій, що й при зашифруванні, але з наступними змінами:

- фрагменти розширеного ключа використовуються у зворотній послідовності,
- замість операції FL використовується зворотна їй операція – FLI.

Схеми виконання функції FLI і процедури розшифрування наведено на малюнках 6 і 7 відповідно:

Методи аналізу

Як говорилося на початку розділу, диференціальний і неможливий диференціальний аналізи виявилися ефективні лише до версій шифру з меншою кількістю раундів і без операції FL [2][3]. Проте, на даний момент цей напрямок аналізу, особливе використання слабких ключів, найбільше перспективно, тому що наближене до реальних можливих допущень при використанні алгоритму.

Так само, ученим з Японії був проведений інтегральний аналіз повного алгоритму, використовуючи відкритих текстів зі складністю обчислення, рівної [4].

Лінійний аналіз дав результати тільки для 7-раундової версії шифру, і також без операції FL[5].

Так як MISTY1 створювався, у тому числі, з розрахунку на апаратну реалізацію, має сенс диференціальний аналіз, заснований на використанні атаки по помилках обчислень, що в цьому випадку наближене до реальності.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Висновок

Таким чином, була докладно описана структура алгоритму шифрування MISTY1 і розглянуті методи його аналізу, найбільш прагматичні напрямки дослідження. Далі має бути створення програмної реалізації для більш детального розгляду алгоритму й набір статистичних даних для повного дослідження й пошуку оптимального підходу до аналізу MISTY1.

КБПЗ_2024

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене ПЗ ґрунтується на терміналі EasyPay рисунок 5.1. Це зручний сервіс для оплати повсякденних послуг в Україні. Платіжний сервіс EasyPay входить до складу холдингу EasyPay поряд з платіжними сервісами EasyPay Гаманець, EasyPay Банк, комунікаційною системою EasyPay Реклама, постачальником фіскального устаткування РАУ Kiosk і компанією-інтегратором лотерей на платіжних терміналах Лото Інтегратор.



Рисунок 5.1 – Загальний вигляд термінала

За допомогою EasyPay можна оплачувати послуги більше 450 компаній-провайдерів: від мобільного зв'язку та ЖКП до банківських кредитів, а також купувати авіа-і ж/д квитки, оплачувати покупки в Інтернет магазинах і багато іншого.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59



Рисунок 5.2 – Головне вікно терміналу

В Україні EasyPay пропонує своїм користувачам широкий набір платіжних інструментів: мережу з більш ніж 6000 терміналів EasyPay, Електронний гаманець. EasyPay дає можливість оплачувати послуги в соціальних мережах.

Всі платіжні інструменти EasyPay працюють в режимі 7 днів на тиждень, 24 години на добу, при цьому в більшості випадків оплата надходить на рахунок провайдера в режимі реального часу.

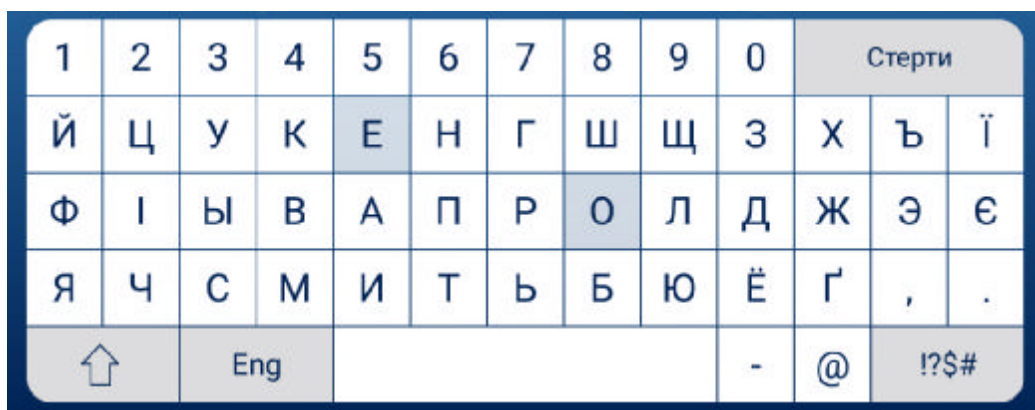


Рисунок 5.3 – Екранна клавіатура, введення даних користувача

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

КБПЗ – 2024

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

– Досліджена система автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

– На основі отриманих результатів досліджень створена програмна реалізація системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10.4.1. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм MISTY1.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
2. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
3. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
4. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
5. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
6. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
7. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
8. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
9. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
10. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
11. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
12. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
13. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

14. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

15. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

16. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

17. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

18. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

21. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

22. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

23. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

24. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

25. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

26. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

27. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

28. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

29. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

30. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

31. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів ІЕС60880 та ІЕС62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

32. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

33. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

34. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

35. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

36. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

37. Смірнов О.А., Усік П.С., Миронєць І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

38. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

39. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнотраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

40. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

41. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу

відновлення та зміцнення поверхонь деталей. Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

42. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

43. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

44. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

45. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

46. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSН-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

47. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA - 2017.

48. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

управління, навігації та зв'язку. – Випуск 4 (44). - Полтава: ПолтНТУ. - 2017. - С. 112-115.

49. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

50. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). - Полтава: ПолтНТУ. - 2016. - С. 98-103.

51. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). - Харків: ХУПС. - 2016. - С.96-100.

52. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". - Випуск 8 (145). - Х.: ХУПС - 2016. - С. 77-80.

53. Смірнов О.А., Лисенко І.А. Розробка впорядкованих каскадних таблиць рішень із використанням матриць слідування. Збірник наукових праць "Системи обробки інформації". - Випуск 6 (143). - Х.: ХУПС - 2016. - С. 216-220.

54. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод кількісної оцінки ризиків розроблення програмного забезпечення. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). - Харків: ХУПС. - 2016. - С. 128-133.

					ВКРБ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Перелік документів, що розробляються.....	5
8	Етапи розробки.....	6
9	Порядок контролю та приймання.....	6

					ВКРБ-123.24.0005.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Квач Д.Р.				Програмне забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay	Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.					Б	1	6
Н. Контр.	Коваленко А.С				ЦНТУ КІ-20			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

2 Підстава для розробки

Підставою для розробки служить завдання на випуск кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 131-02 від 01.04.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-123.24.0005.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи автоматів самообслуговування на базі сенсорних терміналів платіжної системи EasyPay;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.24.0005.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					ВКРБ-123.24.0005.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 71 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					ВКРБ-123.24.0005.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 7.06.2024 р.

					ВКРБ-123.24.0005.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти
_____ Дреєв О.М.

*Програмне забезпечення системи автоматів самообслуговування на базі
сенсорних терміналів платіжної системи EasyPay*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 24

Літера: РП

Кропивницький – 2024 року

**Файл проекту автомату самообслуговування на базі сенсорних терміналів
платіжної системи EasyPay**

```
program Daniil Kvach_Automat_samoobslugovuvanya_EasyPay;

uses
  Forms,
  Unit1 in 'Ter_Unit1.pas' {Form1},
  Unit2 in ' Ter_Unit2.pas' {Form2},
  Unit3 in ' Ter_Unit3.pas' {Form3};
  Unit4 in ' Ter_Unit4.pas' {Form4};
  Unit5 in ' Ter_Unit5.pas' {Form5};
  Unit5 in ' Ter_Unit6.pas' {Form6};

{$R *.RES}

begin
  Form6:=TForm5.Create(Application);
  Form6.Show;
  Form6.Update;
Try
  Application.HintPause:=200;
  Application.HintHidePause:=7000;
  Application.HintShortPause:=25;
  Application.Initialize;
  Application.Title := 'Daniil Kvach_Automat_samoobslugovuvanya_EasyPay';
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm4, Form4);
  Application.CreateForm(TForm5, Form5);
  finally
  Form6.Free;
end;
Application.Run;
end.
```

Файл проекту Ter_Unit4.pas - заголовки внутрішніх змінних бібліотеки IPHLPAPI

```

//*****//
// Testuvalos`a u: Delphi 10.4.1 //
// OS: WINDOWS 10/11 //
//Testova versiya automatu samoobslugovuvanya //
// coded by: Daniil Kvach //
//*****//
unit Unit4
//IPHLPAPI;

interface
uses
  Windows, winsock;
const
  VERSION = '1.3';
const
  ANY_SIZE = 1;
  MAX_ADAPTER_DESCRIPTION_LENGTH = 128; // arb.
  MAX_ADAPTER_NAME_LENGTH = 256; // arb.
  MAX_ADAPTER_ADDRESS_LENGTH = 8; // arb.
  DEFAULT_MINIMUM_ENTITIES = 32; // arb.
  MAX_HOSTNAME_LEN = 128; // arb.
  MAX_DOMAIN_NAME_LEN = 128; // arb.
  MAX_SCOPE_ID_LEN = 256; // arb.

  BROADCAST_NODETYPE = 1;
  PEER_TO_PEER_NODETYPE = 2;
  MIXED_NODETYPE = 4;
  HYBRID_NODETYPE = 8;
  ATypes : array[0..8] of string[20] =
  ( 'UNKNOWN', 'BROADCAST', 'PEER_TO_PEER', '', 'MIXED', '', '', '', 'HYBRID');
  IF_OTHER_ADAPTERTYPE = 0;
  IF_ETHERNET_ADAPTERTYPE = 1;
  IF_TOKEN_RING_ADAPTERTYPE = 2;
  IF_FDDI_ADAPTERTYPE = 3;
  IF_PPP_ADAPTERTYPE = 4;
  IF_LOOPBACK_ADAPTERTYPE = 5;
  IF_SLIP_ADAPTERTYPE = 6;
  //
  AdaptTypes : array[0..6] of string[10] =
  ( 'other', 'ethernet', 'tokenring', 'FDDI', 'PPP', 'loopback', 'SLIP' );
  MAX_INTERFACE_NAME_LEN = 256; { mrapi.h }
  MAXLEN_PHYSADDR = 8; { iprtmib.h }
  MAXLEN_IFDESCR = 256; { --"--- }
  //-----
type
  TMacAddress = array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte;

  PTIP_ADDRESS_STRING = ^TIP_ADDRESS_STRING;
  TIP_ADDRESS_STRING = array[0..15] of char;
  PTIP_ADDR_STRING = ^TIP_ADDR_STRING;
  TIP_ADDR_STRING = packed record // for use in linked lists
    Next: PTIP_ADDR_STRING;
    IpAddress: TIP_ADDRESS_STRING;
    IpMask: TIP_ADDRESS_STRING;
    Context: DWORD;
  end;

  PTFixedInfo = ^TFixedInfo;
  TFixedInfo = packed record
    HostName: array[0..MAX_HOSTNAME_LEN + 4] of char;
    DomainName: array[0..MAX_DOMAIN_NAME_LEN + 4] of char;
    CurrentDNSServer: PTIP_ADDR_STRING;
    DNSServerList: TIP_ADDR_STRING;
    NodeType: UINT;
    ScopeID: array[0..MAX_SCOPE_ID_LEN + 4] of char;
    EnableRouting: UINT;
    EnableProxy: UINT;

```

```

    EnableDNS: UINT;
end;

PTMibIfRow = ^TMibIfRow;
TMibIfRow = packed record
    wszName: array[1..MAX_INTERFACE_NAME_LEN] of WCHAR;
    dwIndex: DWORD;
    dwType: DWORD;
    dwMTU: DWORD;
    dwSpeed: DWORD;
    dwPhysAddrLen: DWORD;
    bPhysAddr: array[1..MAXLEN_PHYSADDR] of byte;
    dwAdminStatus: DWORD;
    dwOperStatus: DWORD;
    dwLastChange: DWORD;
    dwInOctets: DWORD;
    dwInUcastPkts: DWORD;
    dwInNUCcastPkts: DWORD;
    dwInDiscards: DWORD;
    dwInErrors: DWORD;
    dwInUnknownProtos: DWORD;
    dwOutOctets: DWORD;
    dwOutUCastPkts: DWORD;
    dwOutNUCcastPkts: DWORD;
    dwOutDiscards: DWORD;
    dwOutErrors: DWORD;
    dwOutQLen: DWORD;
    dwDescrLen: DWORD;
    bDescr: array[1..MAXLEN_IFDESCR] of char; //byte;
end;

TMIBIfArray = array of TMIBIFRow;

PTMibIfTable = ^TMIBIfTable;
TMibIfTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIfRow;
end;

TTIME_T = array[1..325] of byte;

PTIP_ADAPTER_INFO = ^TIP_ADAPTER_INFO;
TIP_ADAPTER_INFO = packed record
    Next: PTIP_ADAPTER_INFO;
    ComboIndex: DWORD;
    AdapterName: array[1..MAX_ADAPTER_NAME_LENGTH + 4] of char;
    Description: array[1..MAX_ADAPTER_DESCRIPTION_LENGTH + 4] of char;
    AddressLength: UINT;
    Address: array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte;
    Index: DWORD;
    aType: UINT;
    DHCPEnabled: UINT;
    CurrentIPAddress: TIP_ADDR_STRING;
    IPAddressList: TIP_ADDR_STRING;
    GatewayList: TIP_ADDR_STRING;
    DHCPserver: TIP_ADDR_STRING;
    HaveWINS: BOOL;
    PrimaryWINSServer: TIP_ADDR_STRING;
    SecondaryWINSServer: TIP_ADDR_STRING;
    LeaseObtained: TTIME_T; //??
    LeaseExpires: TTIME_T; //??
end;

PTMibTCPRow = ^TMibTCPRow;
TMibTCPRow = packed record
    dwState: DWORD;
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
    dwRemoteAddr: DWORD;

```

```

    dwRemotePort: DWORD;
end;
//
PMibTCPTable = ^TMibTCPTable;
TMibTCPTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..0] of TMibTCPRow;
end;
//
PMibTCPStats = ^TMibTCPStats;
TMibTCPStats = packed record
    dwRTOAlgorithm: DWORD;
    dwRTOMin: DWORD;
    dwRTOMax: DWORD;
    dwMaxConn: DWORD;
    dwActiveOpens: DWORD;
    dwPassiveOpens: DWORD;
    dwAttemptFails: DWORD;
    dwEstabResets: DWORD;
    dwCurrEstab: DWORD;
    dwInSegs: DWORD;
    dwOutSegs: DWORD;
    dwRetransSegs: DWORD;
    dwInErrs: DWORD;
    dwOutRsts: DWORD;
    dwNumConns: DWORD;
end;

PMibUDPRow = ^TMibUDPRow;
TMibUDPRow = packed record
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
end;
//
PMibUDPTable = ^TMIBUDPTable;
TMIBUDPTable = packed record
    dwNumEntries: DWORD;
    UDPTable: array[0..ANY_SIZE - 1] of TMibUDPRow;
end;
//
PMibUdpStats = ^TMIBUdpStats;
TMIBUdpStats = packed record
    dwInDatagrams: DWORD;
    dwNoPorts: DWORD;
    dwInErrors: DWORD;
    dwOutDatagrams: DWORD;
    dwNumAddrs: DWORD;
end;

PMibIPNetRow = ^TMibIPNetRow;
TMibIPNetRow = packed record
    dwIndex: DWord;
    dwPhysAddrLen: DWord;
    bPhysAddr: TMacAddress;
    dwAddr: DWord;
    dwType: DWord;
end;

PMibIPNetTable = ^TMibIPNetTable;
TMibIPNetTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIPNetRow;
end;

PMibIPStats = ^TMibIPStats;
TMibIPStats = packed record
    dwForwarding: DWORD;
    dwDefaultTTL: DWORD;
    dwInReceives: DWORD;

```

```

dwInHdrErrors: DWORD;
dwInAddrErrors: DWORD;
dwForwDatagrams: DWORD;
dwInUnknownProtos: DWORD;
dwInDiscards: DWORD;
dwInDelivers: DWORD;
dwOutRequests: DWORD;
dwRoutingDiscards: DWORD;
dwOutDiscards: DWORD;
dwOutNoRoutes: DWORD;
dwReasmTimeOut: DWORD;
dwReasmReqds: DWORD;
dwReasmOKs: DWORD;
dwReasmFails: DWORD;
dwFragOKs: DWORD;
dwFragFails: DWORD;
dwFragCreates: DWORD;
dwNumIf: DWORD;
dwNumAddr: DWORD;
dwNumRoutes: DWORD;
end;
//
PMibIPAddrRow = ^TMibIPAddrRow;
TMibIPAddrRow = packed record
    dwAddr: DWORD;
    dwIndex: DWORD;
    dwMask: DWORD;
    dwBCastAddr: DWORD;
    dwReasmSize: DWORD;
    Unused1,
    Unused2: WORD;
end;

TMibIPAddrArray = array of TMIBIPAddrRow;

PMibIPAddrTable = ^TMibIPAddrTable;
TMibIPAddrTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIPAddrRow;
end;

PMibIPForwardRow = ^TMibIPForwardRow;
TMibIPForwardRow = packed record
    dwForwardDest: DWORD;
    dwForwardMask: DWORD;
    dwForwardPolicy: DWORD;
    dwForwardNextHop: DWORD;
    dwForwardIFIndex: DWORD;
    dwForwardType: DWORD;
    dwForwardProto: DWORD;
    dwForwardAge: DWORD;
    dwForwardNextHopAS: DWORD;
    dwForwardMetric1: DWORD;
    dwForwardMetric2: DWORD;
    dwForwardMetric3: DWORD;
    dwForwardMetric4: DWORD;
    dwForwardMetric5: DWORD;
end;
//
PMibIPForwardTable = ^TMibIPForwardTable;
TMibIPForwardTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE - 1] of TMibIPForwardRow;
end;

PMibICMPStats = ^TMibICMPStats;
TMibICMPStats = packed record
    dwMsgs: DWORD;

```

```

    dwErrors: DWORD;
    dwDestUnreachs: DWORD;
    dwTimeEcxcds: DWORD;
    dwParmProbs: DWORD;
    dwSrcQuenchs: DWORD;
    dwRedirects: DWORD;
    dwEchos: DWORD;
    dwEchoReps: DWORD;
    dwTimeStamps: DWORD;
    dwTimeStampReps: DWORD;
    dwAddrMasks: DWORD;
    dwAddrReps: DWORD;
end;

PTMibICMPInfo = ^TMibICMPInfo;
TMibICMPInfo = packed record
    InStats: TMibICMPStats;
    OutStats: TMibICMPStats;
end;
//-----+-----
function GetAdaptersInfo( pAdapterInfo: PTIP_ADAPTER_INFO;
    pOutBufLen: PULONG ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetNetworkParams( FixedInfo: PTFixedInfo; pOutPutLen: PULONG ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetTcpTable( pTCPTable: PTMibTCPTable; pDWSize: PDWORD;
    bOrder: BOOL ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetTcpStatistics( pStats: PTMibTCPStats ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetUdpTable( pUdpTable: PTMibUDPTable; pDWSize: PDWORD;
    bOrder: BOOL ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetUdpStatistics( pStats: PTMibUdpStats ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetIpStatistics( pStats: PTMibIPStats ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetIpNetTable( pIpNetTable: PTMibIPNetTable;
    pdwSize: PULONG;
    bOrder: BOOL ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetIpAddrTable( pIpAddrTable: PTMibIPAddrTable;
    pdwSize: PULONG;
    bOrder: BOOL ): DWORD;
stdcall; external 'IPHLPAPI.DLL';

function GetIpForwardTable( pIPForwardTable: PTMibIPForwardTable;
    pdwSize: PULONG;
    bOrder: BOOL ): DWORD;
stdCall; external 'IPHLPAPI.DLL';

function GetIcmpStatistics( pStats: PTMibICMPInfo ): DWORD;
stdCall; external 'IPHLPAPI.DLL';

function GetRTTAndHopCount( DestIPAddress: DWORD; HopCount: PULONG;
    MaxHops: ULONG; RTT: PULONG ): BOOL;
stdCall; external 'IPHLPAPI.DLL';

function GetIfTable( pIfTable: PTMibIfTable; pdwSize: PULONG;
    bOrder: boolean ): DWORD;
stdCall; external 'IPHLPAPI.DLL';

```

```
function GetIfEntry( pIfRow: PTMibIfRow ): DWORD;  
stdCall; external 'IPHLPAPI.DLL';
```

```
implementation
```

```
end.
```

К6П3_2024

Файл проекту Ter_Unit1.pas - організація захищеного стеку автомату самообслуговування на базі сенсорних терміналів платіжної системи EasyPay

```

//*****//
// Testuvalos`a u:   Delphi 10.4.1 //
// OS: WINDOWS 10/11 //
//Testova versiya automatu samoobslugovuvanya //
// coded by: Daniil Kvach //
//*****//
unit Unit1;
//Automat_samoobslugovuvanya_EasyPay_Traffic;

interface

uses SysUtils, Windows, IPHelper, IPHLPAPI;

type
TAutomat_samoobslugovuvanya_EasyPayTraffic = Class;

TNewInstanceEvent = procedure(Sender
:TAutomat_samoobslugovuvanya_EasyPayTraffic) of object;
TFreezeEvent = procedure(Sender :TAutomat_samoobslugovuvanya_EasyPayTraffic) of
object;

TAutomat_samoobslugovuvanya_EasyPayTraffic = Class
private
FIP: string;
FMac: string;
FInPerSec: Dword;
FInTotal: Dword;
FPeakInPerSec: Dword;
FInterfaceIndex: DWord;
FActiveCountIn: Dword;
FSecondsActive: Cardinal;
FPrevCountIn: DWord;
FDescription: string;
FOutTotal: Dword;
FPeakOutPerSec: Dword;
FOutPerSec: Dword;
FPrevCountOut: DWord;
FActiveCountOut: Dword;
FAverageInPerSec: Dword;
FAverageOutPerSec: Dword;
FStartedAt: TDateTime;
FRunning: boolean;
FOnFreeze: TFreezeEvent;
FOnUnFreeze: TFreezeEvent;
FConnected: boolean;
FFound: boolean;
FSpeed: DWord;

function GetIPFromIFIndex(InterfaceIndex: Cardinal): string;
public
property Found : boolean read FFound write FFound;
property Connected : boolean read FConnected;
property Running : boolean read FRunning;
property InterfaceIndex : DWord read FInterfaceIndex;
property IP : string read FIP;
property Mac : string read FMac;
property Description : string read FDescription;
property StartedAt : TDateTime read FStartedAt;
property SecondsActive : Cardinal read FSecondsActive;
property Speed : DWord read FSpeed;
property ActiveCountIn : Dword read FActiveCountIn;
property PrevCountIn : DWord read FPrevCountIn;

```

```

property InPerSec : Dword read FInPerSec; property AverageInPerSec : Dword read
FAverageInPerSec;
property InTotal : Dword read FInTotal;
property PeakInPerSec : Dword read FPeakInPerSec;
property ActiveCountOut : Dword read FActiveCountOut;
property PrevCountOut : DWord read FPrevCountOut;
property OutPerSec : Dword read FOutPerSec; property AverageOutPerSec : Dword
read FAverageOutPerSec;
property OutTotal : Dword read FOutTotal;
property PeakOutPerSec : Dword read FPeakOutPerSec;
procedure NewCycle(const InOctets, OutOctets, TrafficSpeed : Dword);
procedure Reset;
procedure Freeze;
procedure UnFreeze;
procedure MarkDisconnected;
function GetStatus : string;
function FriendlyRunningTime:string;
constructor Create(const AMibIfRow : TMibIfRow; OnNewInstance :
TNewInstanceEvent);
published
property OnFreeze :TFreezeEvent read FOnFreeze write FOnFreeze;
property OnUnFreeze :TFreezeEvent read FOnUnFreeze write FOnUnFreeze;
end;

function BytesToFriendlyString(Value : DWord) : string;
function BitsToFriendlyString(Value : DWord) : string;

implementation

function BytesToFriendlyString(Value : DWord) : string;
const
OneKB = 1024;
OneMB = OneKB * 1024;
OneGB = OneMB * 1024;
begin
if Value < OneKB then
Result := FormatFloat('#,##0.00 B',Value)
else
if Value < OneMB then
Result := FormatFloat('#,##0.00 KB', Value / OneKB)
else
if Value < OneGB then
Result := FormatFloat('#,##0.00 MB', Value / OneMB)
end; (*BytesToFriendlyString*)

function BitsToFriendlyString(Value : DWord) : string;
const
OneKB = 1000;
OneMB = OneKB * 1000;
OneGB = OneMB * 1000;
begin
if Value < OneKB then
Result := FormatFloat('#,##0.00 bps',Value)
else
if Value < OneMB then
Result := FormatFloat('#,##0.00 Kbps', Value / OneKB)
else
if Value < OneGB then
Result := FormatFloat('#,##0.00 Mbps', Value / OneMB)
end; (*BytesToFriendlyString*)

{ TAutomat_samoobslugovuvanya_EasyPayTraffic }
constructor TAutomat_samoobslugovuvanya_EasyPayTraffic.Create(const AMibIfRow:
TMibIfRow; OnNewInstance : TNewInstanceEvent);
var
Descr : string;
begin
inherited Create;

```

```

FRunning := True;
FConnected := True;

self.FInterfaceIndex := AMibIfRow.dwIndex;
self.FIP := GetIPFromIFIndex(self.InterfaceIndex);
self.FMac := MacAddr2Str(TMbAddress(AMibIfRow.bPhysAddr),
AMibIfRow.dwPhysAddrLen);

SetLength(Descr, Pred(AMibIfRow.dwDescrLen));
Move(AMibIfRow.bDescr, Descr[1], pred(AMibIfRow.dwDescrLen));
self.FDescription := Trim(Descr);

self.FPrevCountIn := AMibIfRow.dwInOctets;
self.FPrevCountOut := AMibIfRow.dwOutOctets;

self.FStartedAt := Now;

self.FSpeed := AMibIfRow.dwSpeed;

FActiveCountIn := 0;
FActiveCountOut := 0;
FInTotal := 0;
FOutTotal := 0;
FInPerSec := 0;
FOutPerSec := 0;
FPeakInPerSec := 0;
FPeakOutPerSec := 0;

if Assigned(OnNewInstance) then OnNewInstance(self);
end; (*Create*)

procedure TAutomat_samoobslugovuvanya_EasyPayTraffic.NewCycle(const InOctets,
OutOctets, TrafficSpeed: Dword);
begin
Inc(self.FSecondsActive);

If not Running then Exit;

FSpeed := TrafficSpeed;

//IN
self.FInPerSec := InOctets - self.PrevCountIn;
Inc(self.FInTotal, self.InPerSec);
if InPerSec > 0 then Inc(FActiveCountIn);
if InPerSec > PeakInPerSec then FPeakInPerSec := InPerSec;
try
self.FAverageInPerSec := InTotal div ActiveCountIn
except
self.FAverageInPerSec := 0
end;
FPrevCountIn := InOctets;

//OUT
self.FOutPerSec := OutOctets - self.PrevCountOut;
Inc(self.FOutTotal, self.OutPerSec);
if OutPerSec > 0 then Inc(FActiveCountOut);
if OutPerSec > PeakOutPerSec then FPeakOutPerSec := OutPerSec;
try
self.FAverageOutPerSec := OutTotal div ActiveCountOut
except
self.FAverageOutPerSec := 0
end;
FPrevCountOut := OutOctets;
end; (*NewCycle*)

```

```

function
TAutomat_samoobsslugovuvanya_EasyPayTraffic.GetIPFromIFIndex(InterfaceIndex:
Cardinal): string;
var
i: integer;
IParr : TMIBIPAddrArray;
begin
Result := '!not_found!';
Get_IPAddrTableMIB( IpArr );
if Length(IParr) > 0 then
for i := low(IParr) to High(IParr) do
if IParr[i].dwIndex = InterfaceIndex then
begin
Result := IPAddr2Str(IParr[i].dwAddr);
BREAK;
end;
end; (*GetIPFromIFIndex*)

procedure TAutomat_samoobsslugovuvanya_EasyPayTraffic.Reset;
begin
self.FPrevCountIn := InPerSec;
self.FPrevCountOut := OutPerSec;

self.FStartedAt := Now;
FSecondsActive := 0;

FActiveCountIn := 0;
FActiveCountOut:= 0;
FInTotal := 0;
FOutTotal := 0;
FInPerSec := 0;
FOutPerSec := 0;
FPeakInPerSec := 0;
FPeakOutPerSec := 0;
end; (*Reset*)

procedure TAutomat_samoobsslugovuvanya_EasyPayTraffic.Freeze;
begin
FRunning := False;
if Assigned(FOnFreeze) then OnFreeze(Self);
end; (*Freeze*)

procedure TAutomat_samoobsslugovuvanya_EasyPayTraffic.UnFreeze;
begin
FRunning := True;
if Assigned(FOnUnFreeze) then OnUnFreeze(Self);
end; (*UnFreeze*)

procedure TAutomat_samoobsslugovuvanya_EasyPayTraffic.MarkDisconnected;
begin
self.FConnected := False;
self.FRunning := False;
end; (*MarkDisconnected*)

function TAutomat_samoobsslugovuvanya_EasyPayTraffic.GetStatus: string;
begin
if self.Connected then
Result := 'Connected'
else
Result := 'NOT connected';

if self.Running then
Result := Result + ', 1'
else
Result := Result + ', 2';
end; (*GetStatus*)

function TAutomat_samoobsslugovuvanya_EasyPayTraffic.FriendlyRunningTime:
string;

```

```
var
H, M, S: string;
ZH, ZM, ZS: Integer;
begin
ZH := SecondsActive div 3600;
ZM := Integer(SecondsActive) div (60 - ZH * 60);
ZS := Integer(SecondsActive) - (ZH * 3600 + ZM * 60);
H := Format('%.2d', [ZH]);
M := Format('%.2d', [ZM]);
S := Format('%.2d', [ZS]);

Result := H + ':' + M + ':' + S;
end; (*FriendlyRunningTime*)

end.
```

K6П3_2024

Файл проекту Ter_Unit2.pas - перевірка апаратної частини

```

//*****
// Testuvalos`a u:   Delphi 10.4.1 //
// OS: WINDOWS 10/11 //
//Testova versiya automatu samoobslugovuvanya //
// coded by: Daniil Kvach //
//*****//
unit Unit2;
//Automat_samoobslugovuvanya_EasyPaySysInfoU;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, StdCtrls, Buttons, ImgList, Commctrl, ShellAPI, Menus;

type
TForm1 = class(TForm)
PageControl1: TPageControl;
TabSheet2: TTabSheet;
TabSheet1: TTabSheet;
ListView1: TListView;
StatusBar1: TStatusBar;
ListView2: TListView;
ImageList1: TImageList;
PopupMenu1: TPopupMenu;
Details1: TMenuItem;
KillProcess1: TMenuItem;
RefreshList1: TMenuItem;
PopupMenu2: TPopupMenu;
CloseWindow1: TMenuItem;
RefreshList2: TMenuItem;
SpeedButton1: TSpeedButton;
procedure PageControl1Change(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ListView1Click(Sender: TObject);
procedure ListView1DbClick(Sender: TObject);
procedure ListView1KeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure Details1Click(Sender: TObject);
procedure KillProcess1Click(Sender: TObject);
procedure RefreshList1Click(Sender: TObject);
procedure CloseWindow1Click(Sender: TObject);
procedure RefreshList2Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.DFM}
uses
TlHelp32, AdditU;
procedure ListProcesses;
var
hSnapshot: THandle;
lppe: TProcessEntry32;
hIcon: THandle;
Count: Integer;
procedure _FillList;
begin
with Form1.ListView1.Items.Add, lppe do
begin
hIcon:= ExtractIcon(hInstance, lppe.szExeFile, 0);
if hIcon = 0 then
hIcon:= LoadImage(0, IDI_WINLOGO, IMAGE_ICON, LR_DEFAULTSIZE,
LR_DEFAULTSIZE, LR_DEFAULTSIZE or LR_DEFAULTCOLOR or LR_SHARED);

```

```

ImageIndex:= ImageList_AddIcon(Form1.ImageList1.Handle, hIcon);
Caption:= ExtractFileName(szExeFile);
SubItems.Add(Format('$%x', [Th32ProcessID]));
SubItems.Add(Format('$%x', [Th32ParentProcessID]));
case pcPriClassBase of
4: SubItems.Add(Format('%d (Idle)', [pcPriClassBase]));
8: SubItems.Add(Format('%d (Norm)', [pcPriClassBase]));
13: SubItems.Add(Format('%d (High)', [pcPriClassBase]));
24: SubItems.Add(Format('%d (Real)', [pcPriClassBase]));
else SubItems.Add(Format('%d', [pcPriClassBase]));
end;
SubItems.Add(Format('%d', [cntThreads]));
SubItems.Add(Format('%d', [cntUsage]));
SubItems.Add(szExeFile);
end;
Inc(Count);
end;
begin
hSnapshot:= CreateToolHelp32Snapshot(TH32CS_SNAPPROCESS, 0);
if hSnapshot <> INVALID_HANDLE_VALUE then
begin
Form1.ListView1.Items.Clear;
lppe.dwSize:= SizeOf(lppe);
Count:= 0;
if Process32First(hSnapshot, lppe) then _FillList;
while Process32Next(hSnapshot, lppe) do _FillList;
Form1.StatusBar1.Panels[1].Text:= 'Total: ' + IntToStr(Count);
CloseHandle(hSnapshot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
end;
procedure ListWindows;
function EnumWindowsProc(hWnd: THandle; lParam: Integer): Boolean; stdcall;
var
Text, PName: string;
hIcon: THandle;
lpdwPID: PDWORD;
i: Integer;
begin
New(lpdwPID);
GetWindowThreadProcessID(hWnd, lpdwPID);
SetLength(Text, 255);
if GetWindowText(hWnd, PChar(Text), 255) <> 0 then
with Form1.ListView2.Items.Add, Form1.ListView1 do
begin
hIcon:= GetClassLong(hWnd, GCL_HICON);
ImageIndex:= ImageList_AddIcon(Form1.ImageList1.Handle, hIcon);
for i:= 0 to Items.Count - 1 do
if Cardinal(StrToInt(Items[i].SubItems[0])) = lpdwPID^ then
PName:= Items[i].Caption;
Caption:= Text;
SubItems.Add(Format('%d', [hWnd]));
SubItems.Add(Format('$%x', [lpdwPID^]));
SubItems.Add(PName);
Dispose(lpdwPID);
end;
Result:= true;
end;
begin
Form1.ListView2.Items.Clear;
EnumWindows(@EnumWindowsProc, 0);
end;

procedure ListModules(OwnerID: Cardinal);
var
hSnapshot: THandle;
lpme: TModuleEntry32;

```

```

procedure _FillList;
begin
with Form2.ListView1.Items.Add, lpme do
begin
Caption:= ExtractFileName(szModule);
SubItems.Add(Format('%d', [modBaseSize]));
SubItems.Add(Format('%p', [modBaseAddr]));
SubItems.Add(Format('%d', [ProccntUsage]));
SubItems.Add(Format('%d', [GlblcntUsage]));
SubItems.Add(szExePath);
end;
end;

begin
hSnapShot:= CreateToolHelp32Snapshot(TH32CS_SNAPMODULE, OwnerID);
if hSnapShot <> INVALID_HANDLE_VALUE then
begin
Form2.ListView1.Items.Clear;
lpme.dwSize:= SizeOf(lpme);
if Module32First(hSnapShot, lpme) then _FillList;
while Module32Next(hSnapShot, lpme) do _FillList;
CloseHandle(hSnapShot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
end;

procedure ListThreads(OwnerID: Cardinal);
var
hSnapShot: THandle;
lpte: TThreadEntry32;

procedure _FillList;
begin
if lpte.Th32OwnerProcessID =
Cardinal(StrToInt(Form1.ListView1.Selected.SubItems[0])) then
with Form2.ListView2.Items.Add, lpte do
begin
Caption:= Format('%x', [Th32ThreadID]);
case TpBasePri of
4: SubItems.Add(Format('%d (Idle)', [TpBasePri]));
8: SubItems.Add(Format('%d (Norm)', [TpBasePri]));
13: SubItems.Add(Format('%d (High)', [TpBasePri]));
24: SubItems.Add(Format('%d (Real)', [TpBasePri]));
else SubItems.Add(Format('%d', [TpBasePri]));
end;
case TpDeltaPri of
-15: SubItems.Add(Format('%d (Idle)', [TpDeltaPri]));
-2: SubItems.Add(Format('%d (Lowest)', [TpDeltaPri]));
-1: SubItems.Add(Format('%d (Low)', [TpDeltaPri]));
0: SubItems.Add(Format('%d (Normal)', [TpDeltaPri]));
1: SubItems.Add(Format('%d (High)', [TpDeltaPri]));
2: SubItems.Add(Format('%d (Highest)', [TpDeltaPri]));
15: SubItems.Add(Format('%d (Time Critical)', [TpDeltaPri]));
else SubItems.Add(Format('%d', [TpDeltaPri]));
end;
SubItems.Add(Format('%d', [cntUsage]));
end;
end;

begin
hSnapShot:= CreateToolhelp32Snapshot(TH32CS_SNAPTHREAD, OwnerID);
if hSnapShot <> INVALID_HANDLE_VALUE then
begin
Form2.ListView2.Items.Clear;
lpte.dwSize:= SizeOf(lpte);
if Thread32First(hSnapShot, lpte) then _FillList;
while Thread32Next(hSnapShot, lpte) do _FillList;
CloseHandle(hSnapShot);

```

```

end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
end;

procedure ListHeaps(OwnerID: Cardinal);
var
hSnapShot: THandle;
lph1: THeapList32;
lphe: THeapEntry32;

procedure _FillList;
begin
if Heap32First(lphe, lph1.Th32ProcessID, lph1.Th32HeapID) then
repeat
with Form2.ListView3.Items.Add, lphe do
begin
Caption:= Format('$%x', [Th32HeapID]);
SubItems.Add(Format('%d', [dwBlockSize]));
SubItems.Add(Format('$%x', [dwAddress]));
case dwFlags of
LF32_FIXED: SubItems.Add('Fixed');
LF32_FREE: SubItems.Add('Free');
LF32_MOVEABLE: SubItems.Add('Moveable')
else SubItems.Add('Unknown');
end;
end
until not Heap32Next(lphe);
end;

begin
try
Form1.Caption:= 'SysInfo: PROCESSING... PLEASE WAIT';
hSnapShot:= CreateToolhelp32Snapshot(TH32CS_SNAPHEAPLIST, OwnerID);
if hSnapShot <> INVALID_HANDLE_VALUE then
begin
Form2.ListView3.Items.Clear;
lph1.dwSize:= SizeOf(lph1);
lphe.dwSize:= SizeOf(lphe);
if Heap32ListFirst(hSnapShot, lph1) then _FillList;
while Heap32ListNext(hSnapShot, lph1) do _FillList;
CloseHandle(hSnapShot);
end
else MessageBox(Form1.Handle, 'Internal error', 'Error', MB_OK or
MB_ICONERROR);
finally
Form1.Caption:= 'SysInfo';
end;
end;

procedure ListThreadWindows;
var
i: Integer;
function EnumThreadWindowsProc(hWnd: THandle; lParam: Integer): Boolean;
stdcall;
var
Text: string;
hIcon: THandle;
begin
SetLength(Text, 255);
if GetWindowText(hWnd, PChar(Text), 255) <> 0 then
with Form2.ListView4.Items.Add do
begin
hIcon:= GetClassLong(hWnd, GCL_HICON);
ImageIndex:= ImageList_AddIcon(Form1.ImageList1.Handle, hIcon);
Caption:= Text;
SubItems.Add(Format('%d', [hWnd]));
end;
Result:= true;

```

```

end;

begin
Form2.ListView4.Items.Clear;
for i:= 0 to Form2.ListView2.Items.Count - 1 do
EnumThreadWindows (Cardinal (StrToInt (Form2.ListView2.Items[i].Caption)), @EnumThreadWindowsProc, 0);
end;

{-----}
procedure TForm1.PageControl1Change(Sender: TObject);
begin
ListProcesses;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
ListProcesses;
ListWindows;
end;

procedure TForm1.ListView1Click(Sender: TObject);
begin
with (Sender as TListView) do
if Selected <> nil then
StatusBar1.Panels[0].Text:= Selected.SubItems[5] else
StatusBar1.Panels[0].Text:= '';
end;

procedure TForm1.ListView1DbClick(Sender: TObject);
begin
Details1Click(self);
end;

procedure TForm1.ListView1KeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if flag = true then flag:= false
else
begin
if Key = 13 then ListView1DbClick(ListView1);
if Key <> 0 then ListView1Click(ListView1);
end;
end;

procedure TForm1.Details1Click(Sender: TObject);
begin
with ListView1, Form2 do
if Selected <> nil then
begin
ListModules (StrToInt (Selected.SubItems[0]));
ListThreads (StrToInt (Selected.SubItems[0]));
ListHeaps (StrToInt (Selected.SubItems[0]));
ListThreadWindows;
Caption:= 'Details for ' + Selected.Caption;
Show;
end
'Information', MB_OK or MB_ICONINFORMATION);
end;

procedure TForm1.KillProcess1Click(Sender: TObject);
var
hProcess: THandle;
S: string;
ID: Cardinal;
begin
if ListView1.Selected <> nil then
begin
S:= ListView1.Selected.Caption;

```

```
ID:= StrToInt(ListView1.Selected.SubItems[0]);
hProcess:= OpenProcess(PROCESS_ALL_ACCESS, false, ID);
if hProcess <> INVALID_HANDLE_VALUE then
begin
if not TerminateProcess(hProcess, 0) then
MessageBox(0, PChar('Unable to kill process: ' + S),
'Error', MB_ICONWARNING or MB_OK);
CloseHandle(hProcess);
Sleep(500);
ListProcesses;
end;
end
MB_OK or MB_ICONINFORMATION);
end;

procedure TForm1.RefreshList1Click(Sender: TObject);
begin
ListProcesses;
end;

procedure TForm1.CloseWindow1Click(Sender: TObject);
begin
if ListView2.Selected <> nil then
begin
PostMessage(StrToInt(ListView2.Selected.SubItems[0]), WM_CLOSE, 0, 0);
Sleep(500);
ListWindows;
end
end;

procedure TForm1.RefreshList2Click(Sender: TObject);
begin
ListWindows;
end;

end.
```

Файл проекту Ter_Unit3.pas - Організація тимчасових проміжків, системні функції

```

//*****//
// Testuvalos`a u:   Delphi 10.4.1 //
// OS: WINDOWS 10/11 //
//Testova versiya automatu samoobslugovuvanya //
// coded by: Daniil Kvach //
//*****//
unit Unit3;
//FormUnit;
interface
uses
Windows, Graphics, ExtCtrls, Controls, StdCtrls, Buttons, Tabs,
ComCtrls, Classes, SysUtils, Forms, dialogs,
Automat_samoobslugovuvanya_EasyPayUnit;
type
TMainForm = class(TForm)
pnlMain: TPanel;
pnlBottom: TPanel;
pc: TPageControl;
tsAbout: TTabSheet;
tsTraffic: TTabSheet;
ExitButton: TButton;
TrafficTabs: TTabSet;
GroupBox: TGroupBox;
ledAdapterDescription: TLabelledEdit;
UnFreezeButton: TBitBtn;
FreezeButton: TBitBtn;
ClearCountersButton: TBitBtn;
ledMACAddress: TLabelledEdit;
gbIN: TGroupBox;
ledOctInSec: TLabelledEdit;
ledAvgInSec: TLabelledEdit;
ledPeakInSec: TLabelledEdit;
ledTotalIN: TLabelledEdit;
gbOUT: TGroupBox;
ledOctOUTSec: TLabelledEdit;
ledAvgOUTSec: TLabelledEdit;
ledPeakOUTSec: TLabelledEdit;
ledTotalOUT: TLabelledEdit;
Timer: TTimer;
gbTime: TGroupBox;
ledStartedAt: TLabelledEdit;
ledActiveFor: TLabelledEdit;
RemoveInactiveButton: TBitBtn;
StatusText: TStaticText;
cbOnTop: TCheckBox;
Panel3: TPanel;
ProductName: TLabel;
lblURL: TLabel;
Label3: TLabel;
ProgramIcon: TImage;
StaticText1: TStaticText;
ledSpeed: TLabelledEdit;
procedure TimerTimer(Sender: TObject);
procedure ClearCountersButtonClick(Sender: TObject);
procedure cbOnTopClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure TrafficTabsChange(Sender: TObject; NewTab: Integer;
var AllowChange: Boolean);
procedure ExitButtonClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FreezeButtonClick(Sender: TObject);
procedure UnFreezeButtonClick(Sender: TObject);
procedure RemoveInactiveButtonClick(Sender: TObject);
procedure lblURLClick(Sender: TObject);
procedure StaticText1Click(Sender: TObject);

```

```

procedure pcChange(Sender: TObject);
procedure ledAdapterDescriptionChange(Sender: TObject);
private
procedure HandleNewAdapter(ATraffic :
TAutomat_samoobslugovuvanya_EasyPayTraffic);
procedure HandleFreeze(ATraffic : TAutomat_samoobslugovuvanya_EasyPayTraffic);
procedure HandleUnFreeze(ATraffic :
TAutomat_samoobslugovuvanya_EasyPayTraffic);
function LocateTraffic(AdapterIndex : DWord) :
TAutomat_samoobslugovuvanya_EasyPayTraffic;
procedure ProcessMIBData;
procedure ClearDisplay;
procedure RefreshDisplay;
public
{ Public declarations }
end;
var
MainForm: TMainForm;
ActiveTraffic : TAutomat_samoobslugovuvanya_EasyPayTraffic;

implementation
{$R *.dfm}
procedure TMainForm.ClearDisplay;
var
j:integer;
begin
TrafficTabs.Tabs.Clear;
StatusText.Caption:='';
for j:= 0 to GroupBox.ControlCount-1 do
begin
if GroupBox.Controls[j] is TCustomEdit then
TCustomEdit(GroupBox.Controls[j]).Text := '';
end;
end;

procedure TMainForm.TimerTimer(Sender: TObject);
begin
Timer.Enabled := False;
ProcessMIBData;
Timer.Enabled := True;
end; (*TimerTimer*)

procedure TMainForm.ClearCountersButtonClick(Sender: TObject);
begin
ActiveTraffic.Reset;
RefreshDisplay;
end;

procedure TMainForm.cbOnTopClick(Sender: TObject);
begin
if cbOnTop.Checked = true then
FormStyle := fsSTAYONTOP
else
FormStyle := fsNORMAL;
end;

procedure TMainForm.FormDestroy(Sender: TObject);
var
i : integer;
begin
Timer.OnTimer := nil;
ActiveTraffic := nil;
for i:= 0 to -1 + TrafficTabs.Tabs.Count do
TrafficTabs.Tabs.Objects[i].Free;
end;
procedure TMainForm.TrafficTabsChange(Sender: TObject; NewTab: Integer; var
AllowChange: Boolean);
begin
if NewTab = -1 then

```

```

ActiveTraffic := nil
else
ActiveTraffic :=
TAutomat_samoobslugovuvanya_EasyPayTraffic(TrafficTabs.Tabs.Objects[NewTab]);
RefreshDisplay;
end;

procedure TMainForm.ExitButtonClick(Sender: TObject);
begin
Close;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
Timer.Interval := 1000;
ClearDisplay;
ActiveTraffic := nil;
pcChange(Sender);
Timer.Enabled := True;
end;

procedure TMainForm.RefreshDisplay;
begin
if not Assigned(ActiveTraffic) then
begin
ClearDisplay;
Exit;
end;

with ActiveTraffic do
begin

FreezeButton.Visible := Connected;
UnFreezeButton.Visible := Connected;
ClearCountersButton.Visible := Connected;
RemoveInactiveButton.Visible := not Connected;

FreezeButton.Enabled := Running;
UnFreezeButton.Enabled := not Running;

ledAdapterDescription.Text := Description;
ledMACAddress.Text := MAC;

ledSpeed.Text := BitsToFriendlyString(Speed);

ledOctInSec.Text := BytesToFriendlyString(InPerSec);
ledPeakInSec.Text := BytesToFriendlyString(PeakInPerSec);
ledAvgINSec.Text := BytesToFriendlyString(AverageInPerSec);
ledTotalIN.Text := BytesToFriendlyString(InTotal);

ledOctOUTSec.Text := BytesToFriendlyString(OutPerSec);
ledPeakOUTSec.Text := BytesToFriendlyString(PeakOutPerSec);
ledAvgOUTSec.Text := BytesToFriendlyString(AverageOutPerSec);
ledTotalOUT.Text := BytesToFriendlyString(OutTotal);

self.ledStartedAt.Text := DateTimeToStr(StartedAt);
self.ledActiveFor.Text := FriendlyRunningTime;

StatusText.Caption := GetStatus;
end;
end; (*RefreshDisplay*)

procedure TMainForm.ProcessMIBData;
var
MibArr : IpHlpAPI.TMIBIfArray;
i : integer;
ATraffic : TAutomat_samoobslugovuvanya_EasyPayTraffic;
begin

```

```

Get_IfTableMIB(MibArr);

for i:= 0 to -1 + TrafficTabs.Tabs.Count do
begin
ATraffic :=
TAutomat_samoobslugovuvanya_EasyPayTraffic(TrafficTabs.Tabs.Objects[i]);
if ATraffic.Connected then ATraffic.Found := False;
end;
ATraffic := nil;

//process
if Length(MibArr) > 0 then
begin
for i := Low(MIBArr) to High(MIBArr) do
begin
ATraffic := LocateTraffic(MIBArr[i].dwIndex);
if Assigned(ATraffic) then
begin
ATraffic.NewCycle(MIBArr[i].dwInOctets, MIBArr[i].dwOutOctets,
MIBArr[i].dwSpeed);
end
else
begin
ATraffic := TAutomat_samoobslugovuvanya_EasyPayTraffic.Create(MIBArr[i],
HandleNewAdapter);
ATraffic.Found := True;
ATraffic.OnFreeze := HandleFreeze;
ATraffic.OnUnFreeze := HandleUnFreeze;
end;
end;
end;

for i:= 0 to -1 + TrafficTabs.Tabs.Count do
if NOT
TAutomat_samoobslugovuvanya_EasyPayTraffic(TrafficTabs.Tabs.Objects[i]).Found
then
TAutomat_samoobslugovuvanya_EasyPayTraffic(TrafficTabs.Tabs.Objects[i]).MarkDis
connected;

RefreshDisplay;
end; (*ProcessMIBData*)

function TMainForm.LocateTraffic(AdapterIndex : DWord):
TAutomat_samoobslugovuvanya_EasyPayTraffic;
var
j : cardinal;
ATraffic : TAutomat_samoobslugovuvanya_EasyPayTraffic;
begin
Result := nil;
if TrafficTabs.Tabs.Count = 0 then Exit;

for j:= 0 to -1 + TrafficTabs.Tabs.Count do
begin
ATraffic :=
TAutomat_samoobslugovuvanya_EasyPayTraffic(TrafficTabs.Tabs.Objects[j]);
if ATraffic.InterfaceIndex = AdapterIndex then
begin
Result := ATraffic;
Result.Found := True;
Break;
end;
end;
end;

procedure TMainForm.HandleNewAdapter(ATraffic:
TAutomat_samoobslugovuvanya_EasyPayTraffic);
begin
TrafficTabs.Tabs.AddObject(ATraffic.IP, ATraffic);
TrafficTabs.TabIndex := -1 + TrafficTabs.Tabs.Count;

```

```

end; (*HandleNewAdapter*)

procedure TMainForm.FreezeButtonClick(Sender: TObject);
begin
ActiveTraffic.Freeze;
end;

procedure TMainForm.UnFreezeButtonClick(Sender: TObject);
begin
ActiveTraffic.UnFreeze;
end;

procedure TMainForm.HandleFreeze(ATraffic:
TAutomat_samoobslugovuvanya_EasyPayTraffic);
begin
self.FreezeButton.Enabled := ATraffic.Running;
self.UnFreezeButton.Enabled := not ATraffic.Running;
end;

procedure TMainForm.HandleUnFreeze(ATraffic:
TAutomat_samoobslugovuvanya_EasyPayTraffic);
begin
self.FreezeButton.Enabled := ATraffic.Running;
self.UnFreezeButton.Enabled := not ATraffic.Running;
end;

procedure TMainForm.RemoveInactiveButtonClick(Sender: TObject);
begin
If not ActiveTraffic.Connected then
begin
ActiveTraffic.Free;
ActiveTraffic := nil;
TrafficTabs.Tabs.Delete(TrafficTabs.TabIndex);
TrafficTabs.SelectNext(False);
end;

RefreshDisplay;
end;

procedure TMainForm.pcChange(Sender: TObject);
begin
pnlBottom.Visible := pc.ActivePage = tsTraffic;
end;

procedure TMainForm.ledAdapterDescriptionChange(Sender: TObject);
begin
ledAdapterDescription.Hint := ledAdapterDescription.Text;
ledAdapterDescription.ShowHint := Canvas.TextWidth(ledAdapterDescription.Text)
> ledAdapterDescription.ClientWidth;
end;

end.

```