НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій

УДК 004.9:37.014.6:37.018.43

«ПОГОДЖЕНО»	**«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»**
Декан факультету інформаційних технологій	Завідувач кафедри комп'ютерних наук
Глазунова О.Г., д.п.н., професор	Голуб Б.Л., к.т.н., доцент
___ _____ **2021 р.**	___ _____ **2021 р.**

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Програмне забезпечення системи аналізу і візуалізації динаміки змін біометричних показників

Спеціальність _____121 Інженерія програмного забезпечення ____

Освітня програма ___Програмне забезпечення інформаційних систем

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми

_____ _____ Голуб Б.Л.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи

_____д.п.н., професор_____ _____ Бондаренко В.Є.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ _____Сагалатий Д.В._____
(підпис) (ПІБ студента)

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук
Голуб Б.Л., к.т.н., доцент
_____ _____ _____
(науковий ступінь, вчене звання)
(підпис) (ПІБ)
"___" _____20___ року

З А В Д А Н Н Я
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
СТУДЕНТУ

____Сагалатому Дмитру Васильовичу_____
(прізвище, ім'я, по батькові)

Спеціальність ____121 Інженерія програмного забезпечення

Освітня програма ___ Програмне забезпечення інформаційних систем

Орієнтація освітньої програми_____освітньо-професійна_____

Тема магістерської кваліфікаційної роботи: Програмне забезпечення системи аналізу і візуалізації динаміки змін біометричних показників
_____
затверджена наказом ректора НУБіП України від "29" ___жовтня_____2020 р. №1636"С"

Термін подання завершеної роботи на кафедру _____30 листопада 2021 р._____

Вихідні дані до магістерської кваліфікаційної роботи: дані щодо оцінки якості освітніх ресурсів_____
_____

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Аналіз предметної області	30.10.2020-10.11.2020	
2.	Аналіз вже існуючих рішень	15.11.2020-25.11.2020	

3.	Визначити методи та технології для аналізу і візуалізації біометричних показників	12.12.2020-20.01.2020	
4.	Проектування архітектури системи	02.02.2021-05.03.2021	
5.	Дослідження динаміки біометричних показників	15.03.2021-20.07.2021	
6.	Аналіз результатів дослідження та підведення підсумків	01.09.2021-14.11.2021	
7.	Попередній захист	30.11.2021	
8.	Захист	14.12.2021	

Дата видачі завдання “29”_____жовтня_____2020 р.

Керівник магістерської кваліфікаційної роботи _____ Бондаренко В.Є.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Сагалатий Д.В.
(підпис) (прізвище та ініціали студента)

ЗМІСТ

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IoT –Internet of Things, Інтернет речей

ПС – програмна система

ПЗ – програмне забезпечення

UML – Universal Modeling Language, мова програмування

ВСТУП

З розвитком та впровадженням інформаційних технологій у життя людини, зокрема «Internet of Things», зростає потреба й у підвищенні рівня безпеки та конфіденційності. Таким чином, для доступу до персональних даних користувача все частіше використовуються його біометричні данні: починаючи від відносно простого сканування овалу лиця, закінчуючи скануванням унікальних для кожної людини відбитків пальців або рогівки ока. Такі види захисту доступу є не тільки більш захищеними, а й зручнішими. Найближчим часом разом з все більшим впровадженням IoT у життя людей, прості ключі від дому можуть і будуть замінені електронними, у тому числі біометричними. Тому розробка системи керування механізмами, інтегрованими у загальну систему безпеки є досить актуальною, зважаючи на темп та напрямок розвитку інтегрованих інформаційних технологій.

Мета розробки програмного забезпечення управління електронним замком з використанням біометричних даних користувача полягає в удосконаленні елементів підвищення рівня безпеки та конфіденційності у життя людини, зокрема «Internet of Things».

Методи та технології, які використовуються при розробці програмного додатку

РОЗДІЛ 1
СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Предметною областю для дослідження є програмне забезпечення управління електронним замком з використанням біометричних даних користувача. Захист і зручність – ось основні вимоги, які споживач пред'являє до сучасних систем замикання та контролю доступу.

1.1 Опис предметної області

Інтернет речей (Internet of Things – IoT) – концепція обчислювальної мережі фізичних об'єктів ("речей"), оснащених вбудованими технологіями для взаємодії один з одним або із зовнішнім середовищем, що розглядає організацію таких мереж як явище, яке здатне перебудувати економічні та суспільні процеси та виключити з частини дій та операцій необхідність участі людини.

Інтернет речей проникає у всі сфери життя і забезпечує роботу багатьох систем: від суто споживацьких (таких як різноманітні побутові сенсори, носима електроніка, розумні будинки тощо) до індустріальних (керування і моніторинг виробничих процесів, розумні енергосистеми, розумні міста, автономні автомобілі тощо).

За допомогою «речей» додана вартість збільшуватиметься за рахунок:

- покращення клієнтського досвіду (customer experience);
- зменшення часу, що проходить від задуму продукту до його появи у продажу (time-to-market);
- поліпшення ланцюжків постачання та логістики;
- збільшення продуктивності працівників;
- більш ефективного використання активів (зменшення витрат).

Розвиток інтернету речей стимулює прогрес у багатьох галузях інженерних наук. Очевидно, що зростання попиту на апаратне забезпечення для IoT сприяє інноваціям в галузі електроніки, яка забезпечує індустрію інтернету речей електронними платами, сенсорами, акумуляторами тощо.

Специфіка портативних пристроїв накладає певні обмеження на програмне забезпечення, яке керує ними. Обмеженість обчислювальних можливостей компактних плат, а також вимоги щодо використання енергії (особливо для "речей", які живляться від акумулятора), вимагає економного ставлення до ресурсів, а також переносу майже всіх операцій по обробці даних на бік сервера. Серверні платформи, які обробляють показання сенсорів, також повинні обов'язково враховувати те, що дані неминуче містять похибки, а час від часу пристрої можуть виходити з ладу та функціонувати зовсім неправильно.

Питання безпеки також набуває все більшого значення в світі, де велика кількість пристроїв збирає, обробляє та передає різноманітні дані, які мають певну індустріальну чи бізнесову цінність, або містять персональну інформацію про користувачів. Також необхідно реагувати на виклики, пов'язані з постійно зростаючою кількістю інформаційних загроз.

Сьогодні все більше сучасних офісів представляють собою розгалужену систему точок проходу. Найчастіше розмежування доступу забезпечується всередині приміщень. У звичному для нас розумінні, контроль над прохідністю будь-якої точки, відбувається за допомогою замикаючого пристрою. На допомогу приходять новітні технології, які покликані зробити життя будь-якого офісу чи житлового будинку зручніше та безпечніше. Це дверні замки, які замість механічного ключа використовують відбиток пальця, – так звані біометричні, або точніше дактилоскопічні.

Біометрія – сукупність автоматизованих методів і засобів ідентифікації людини, заснованих на її фізіологічній або поведінковій характеристиці.

Біометрична ідентифікація – засіб підтвердження:

а) особи;

б) належності паспорта його власникові – шляхом розпізнавання і зіставлення біометричних даних (кольору очей, малюнка сітківки ока, відбитків пальців, геометрії руки, рис обличчя тощо), що зафіксовані носіями цих даних, з особистими даними власника.

Існує безліч методів біометричної ідентифікації, які можна розділити на дві великі групи: статистичні та динамічні. Статистичні методи грунтуються на фізіологічній (статистичній) характеристиці людини, тобто унікальній властивості, даному йому від народження і невід'ємне від нього.

1.2 Аналіз вимог до програмної системи

Кожна програмна система – це перетворювач, функцією якого є визначене оброблення даних і вивід отриманих результатів. З метою побудови програмної системи до неї, насамперед, формулюються вимоги до умов виконання функції і обробки даних. Ці вимоги є предметом практичного контракту між замовником і розробником системи.

У загальному випадку під вимогами до ПС розуміють властивості, які повинна мати система для виконання запропонованих замовником функцій.

Прикладами таких функцій можуть бути бізнес-функції, документообіг, керування даними і структурою інформації, що необхідна для прийняття системних рішень, та ін.

Змістовна сторона системних вимог – опис функцій, даних і умов функціонування. Методологія формування вимог за допомогою прецедентів реалізована в середовищі Rational Rose (www.rational.com.uml) і передбачає побудову ряду моделей на їхній основі. Прецеденти відіграють визначену їм роль у кожному з основних процесів проектування: розроблення вимог, аналіз і проектування, виконання й випробування системи. Екземпляр прецеденту у реалізації відображає послідовність дій, виконуваних системою, і спостережень за одержанням результату.

У керованому прецедентами проекті розробляються два зображення системи – зовнішнє і внутрішнє. Зовнішнє зображення ПрО визначає, що повинно відбуватися в системі, щоб забезпечити замовнику необхідні результати. Після подання цілей системи прецедентами розробляються принципи взаємодії системи і її суб'єктів.

Вимоги до програмного забезпечення – набір вимог щодо властивостей, якості та функцій програмного забезпечення, що буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки вимог.

Розробка вимог до програмної системи може бути розділена на декілька етапів: знаходження вимог (збір, визначення потреб зацікавлених осіб та систем); аналіз вимог (перевірка цілісності та закінченості); специфікація (документування вимог); тестування вимог.

Вимоги до функціональних характеристик: Програма повинна забезпечувати виконання наступних функцій:

- відмикання замка, у разі розпізнавання авторизованого користувача;
- налаштування режиму роботи: введення паролю, відбитку пальця, розпізнавання овалу обличчя або комбінований;
- розподіл груп користувачів (адміністрування, користування тощо);
- реєстрація нових користувачів та їх даних;
- ведення журналу вдалих та невдалих спроб авторизації з можливістю його перегляду авторизованими користувачами;
- перегляд журналу вдалих та невдалих спроб авторизації.

Вхідні дані 1:

a) ім’я користувача та пароль для авторизації;

b) відбиток пальця користувача;

c) графічний знімок обличчя користувача.

Вихідні дані 2:

a) стан поточного доступу: дозволено/заблоковано;

b) відмітки про спроби авторизації: вдалі та невдалі.

Вимоги до надійності системи:

- передбачити різні права доступу та керування системою для різних груп користувачів;
- передбачити блокування некоректних дій користувача при роботі з системою.

Вимоги до складу і параметрів технічних засобів:

- система повинна працювати під управлінням операційних системах Windows або Linux;
- керування фізичними об'єктами системи має бути виконано за допомогою мікропроцесора Arduino;
- вхідні дані можуть бути внесені у систему як через окремі фізичні елементи підсистеми Arduino (ip-камера, сканер відбитків пальців), так і через інтерфейси комп'ютера (веб-камера, сканер відбитків пальців).

1.3 Огляд інформаційних джерел та існуючих рішень

Рання каталогізація відбитків пальців датується 1891 р., коли Хуан Вучетіч розпочав збірник відбитків пальців злочинців у Аргентині [1].

Джош Елленбоген та Ніцан Лебовіч стверджували, що біометрія виникла в ідентифікаційних системах злочинної діяльності, розроблених Альфонесом Бертиллоном (1853–1914) і розробленою теорією відбитків пальців і фізіономії Френсіса Гальтона. Джош Елленбоген, "Обгрунтовані та необгрунтовані зображення: За даними Лебовича, робота Гальтона «привела до застосування математичних моделей до відбитків пальців, френології та особливостей обличчя», як частина «фотографії Бертіллона, Гальтона та Марея» (University Park, PA, 2012) [2].

Відповідно, «біометрична система і це абсолютна політична зброя нашої ери „і форма“ м'якого контролю» [3]. Теоретик Девід Ліон показав, що

протягом останніх двох десятиліть біометричні системи проникли на цивільний ринок і розмити лінії між урядовими формами контролю та приватним корпоративним контролем [4].

Келлі А. Гейтс визначила 11 вересня як поворотний момент для культурної мови нашого сучасності: «на мові культурних досліджень наслідки 11 вересня були момент артикуляції, де об'єднуються об'єкти або події, які не мають необхідного зв'язку, і формується нова дискурсна формація: автоматичне розпізнавання обличчя як технологія безпеки батьківщини» [5].

Статистичні методи біометрії грунтуються на фізіологічній (статистичній) характеристиці людини, тобто унікальній властивості, даному йому від народження і невід'ємне від нього.

За відбитком пальця. Найпоширеніший метод біометричної ідентифікації, в основі цього методу лежить унікальність для кожної людини малюнка папілярних візерунків на пальцях. Зображення відбитка пальця, отримане за допомогою спеціального сканера, перетвориться в цифровий код (згортку) і порівнюється з раніше введеним шаблоном (еталоном) або набором шаблонів (у випадку ідентифікації).

За формою долоні. Цей метод побудований на розпізнаванні геометрії кисті руки. З допомогою спеціального пристрою, що дозволяє отримувати тривимірний образ кисті руки, виходять вимірювання, необхідні для унікальної цифрової згортки, що ідентифікує людини.

За розташуванням вен на тильній стороні долоні. За допомогою інфрачервоної камери зчитується малюнок вен на тильній стороні долоні або кисті руки, отримана картинка обробляється, і за схемою розташування вен формується цифрова згортка.

За сітківкою ока. Вірніше, це спосіб ідентифікації за малюнком кровоносних судин очного дна. Для того, щоб малюнок став видно, людині треба подивитися на віддалену світлову точку, і підсвічується таким чином очне дно сканується спеціальною камерою.

За райдужною оболонкою ока. Метод заснований на унікальності малюнка райдужної оболонки ока. Для реалізації методу необхідні спеціальна камера і відповідне програмне забезпечення, що дозволяє виділити з отриманого зображення малюнок райдужної оболонки ока, за якою будується цифровий код.

За формою обличчя. У даному методі ідентифікації будується дво- або тривимірний образ обличчя людини. За допомогою камери і спеціалізованого програмного забезпечення на зображенні виділяються контури очей, брів, носа, губ і т. д., обчислюються відстані між ними. За цими даними будується образ, що перетворюється в цифрову форму для порівняння.

За термограмою особи. В основі цього методу лежить унікальність розподілу на обличчі артерій, які постачають кров'ю шкіру і виділяють тепло. Для отримання зображення використовуються спеціальні камери інфрачервоного діапазону.

Інші статистичні методи. Існують ще такі унікальні способи як ідентифікація за ДНК, підшкірним шаром шкіри, формою вуха, запахом тіла тощо.

Динамічні методи – ґрунтуються на поведінковій (динамічній) характеристиці людини, тобто враховують особливості, характерні для підсвідомих рухів у процесі відтворення якої-небудь дії.

За рукописним почерком. Для цього методу використовується підпис людини (іноді написання кодового слова). Цифровий код формується за динамічними характеристиками написання, тобто будується згортка, в яку входить інформація щодо графічних параметрів, часових характеристик нанесення підпису та динаміки натиску на поверхню тощо.

За клавіатурним почерком. Метод аналогічний вищеописаному, але замість підпису використовується кодове слово. Основна характеристика, за якою будується згортка – динаміка набору кодового слова.

За голосом. Існує багато способів побудови коду ідентифікації за голосом, як правило, це різні поєднання частотних і статистичних характеристик голосу.

Інші динамічні методи. Для даної групи методів описані вище тільки найпоширеніші, існують такі унікальні методи як ідентифікація за рухом губ, за динамікою повороту ключа в дверному замку тощо.

1.3.1 Біометричні дані: збір і захист у Європі, США та Україні.

Найвідомішими *в Європі* правилами є Директива 95/46/ЄС Європейського Парламенту і Ради «Про захист фізичних осіб при обробці персональних даних і про вільне переміщення таких даних» від 24.10.1995 р. (далі – Директива), яка застосовувалася до GDPR, а також GDPR (The General Data Protection Regulation, 2016/Загальний регламент захисту даних для європейських держав-членів), який застосовується з 25.05.2018 р. Організації, які не є членами ЄС, підпадають під дію GDPR, коли вони обробляють персональні дані суб'єктів даних ЄС. Фактично це робить GDPR глобальним законом.

GDPR (§1 ст. 9) спеціально виділяє біометричні дані як «чутливу» категорію особистої інформації, гарантуючи надійний захист. GDPR визначає біометричні дані достатньо широко, в багатьох випадках вимагає оцінки впливу на конфіденційність для їх обробки та надає державам-членам можливість здійснювати різні варіанти захисту біометричних даних.

На відміну від Директиви, GDPR спеціально визнає біометричні дані як підмножину чутливих персональних даних, які вважаються «чутливою категорією персональних даних». Зокрема, GDPR визначає біометричні дані як «персональні дані, отримані в результаті специфічної технічної обробки, що стосується фізичних, фізіологічних чи поведінкових характеристик фізичної особи, які дозволяють або підтверджують унікальну ідентифікацію цієї фізичної особи, наприклад, зображення обличчя або дактилоскопічні дані (відбитки пальців)».

Загалом, GDPR було встановлено дві категорії біометричних даних. По-перше, інформація, що стосується тілесних характеристик (фізичні або

фізіологічні особливості людини: обличчя, відбитки пальців, сканування райдужної оболонки ока тощо). По-друге, інформація, що стосується поведінки людини (будь-які поведінкові характеристики людини, які є унікальними, завдяки чому є можливість ідентифікації людини). На жаль, конкретизація у правилах відсутня.

Важливим добутком GDPR є те, що суб'єкт даних має право відкликати свою згоду в будь-який час, тобто має право бути забутим. Стрімкий розвиток біометричних технологій, невідповідність, пов'язана з поводженням з біометричними даними GDPR, а також можлива розбіжність підходів держав-членів ЄС до біометричних даних вимагатиме більшої уваги та обережності контролерів даних.

Варто зазначити, що у квітні цього року Європарламент схвалив створення однієї з найбільших у світі біометричної бази даних. Нова база даних називатиметься Common Identity Repository (CIR) і буде призначена для об'єднання записів понад 350 млн осіб.

У *Сполучених Штатах Америки* не існує єдиного, всебічного федерального закону, який регулював би збір та використання біометричних даних. Однак, окрім директив Президента, існує достатньо велика кількість законів, які регулюють зазначені правовідносини в США:

Закон «Про захист користувачів кабельних мереж» (The Cable Television Consumer Protectionand Competition Act, 1992);

Закон «Про захист конфіденційності відеоматеріалів» (The Video Privacy Protection Act, 1980);

Закон «Про конфіденційність» (The Privacy Act, 1974);

Закон «Про надання кредитної інформації про покупця» (The Fair Credit Reporting Act, 1970);

Закон «Про право на фінансову приватність» (The Rightto Financial Privacy Act, 1978);

Закон «Про свободу інформації» (The Freedom of Information Act, 1996);

BIPA (Biometric Information Privacy Act, passed by Illinoys (2008), Texas, Washington, Michigan, New Hampshire, Alaska, Montana) — найсуворіший у США закон, який вимагає, щоб люди та організації обробляли біометричні дані, як і всі персональні дані, із заходами безпеки, відповідними шкоді, яку може заподіяти втрата цих даних;

CCPA (California Consumer Privacy Act, 2018/Каліфорнійський закон про конфіденційність споживачів) набуває чинності 01.01.2020 р.

Станом на червень 2019 р. у 47 штатах (окрім штатів з BIPA, де існує заборона з метою комерційного використання) законно використовувати програмне забезпечення для визначення особи, застосовуючи зображення, зроблені без згоди, поки вони знаходяться на публіці.

CCPA часто подається як потенційна модель закону про конфіденційність даних США. Фактично, за рівнем впливу CCPA може стати як другий GDPR. Проте цей закон підвищує права на конфіденційність та захист споживачів для жителів штату Каліфорнія. Можна сказати, що тлумачення біометричних даних CCPA більш розширене, якщо порівнювати з GDPR. Це «фізіологічні, біологічні та поведінкові особливості індивіда, включаючи ДНК індивіда, які можуть використовуватися окремо або у поєднанні один з одним чи з іншими ідентифікаційними даними для встановлення особи індивіда».

Якщо коротко, то права, надані CCPA споживачам штату Каліфорнія на захист їхньої особистої інформації та біометричних даних, стосуються можливості видалення отриманих даних (право бути забутим); доступу до даних (право на розголошення або доступ); отримання даних (можливість перенесення даних, тобто дані повинні бути отримані в поширеному загальновживаному форматі); вимоги до компанії не продавати свою особисту інформацію; відмови від участі («Погодження» є основним стандартом згоди, дозволеним європейським GDPR); застосування дії (застосування штрафних санкцій).

Варто звернути увагу на низку постанов у Сан-Франциско (штат Каліфорнія, травень 2019 р.), Сомервілі (штат Массачусетс, червень 2019 р.), Окленді (штат Каліфорнія, 2019 р.), відповідно до яких міста вирішили заборонити застосування технології розпізнавання облич, у тому числі поліцією.

Неможливо не згадати, що найбільша біометрична база даних у світі — в Індії (до найбільшої у світі біометричної бази даних входить понад 90% населення країни, а біометрія використовується всюди) — UIDAI (Unique Identification Authority of India). Унікальний персональний номер, що присвоюється системою, має назву AADHHAAR.

В Україні ще донедавна продовжувалася боротьба з «міткою звіра», доки в цьому питанні не була поставлена крапка рішенням Верховного суду у складі колегії суддів Касаційного адміністративного суду 26.03.2018 р. у справі №806/3265/17.

Загалом, питання збору, використання та обробки біометричних даних регулюється низкою нормативних актів:

- Цивільний кодекс України;
- Закон України «Про Єдиний державний демографічний реєстр та документи, що підтверджують громадянство України, посвідчують особу чи її спеціальний статус» (далі — Закон ЄДДР);
- Закон України «Про захист персональних даних» (далі — Закон ЗПД);
- Закон України «Про інформацію»;
- Закон України «Про правовий статус іноземців і осіб без громадянства»;
- Закон України «Про біженців та осіб, які потребують додаткового або особливого захисту»;
- Положення про національну систему біометричної верифікації та ідентифікації громадян України, іноземців та осіб без громадянства, затверджене постановою КМУ від 27.12.2017 р. №1073 (далі — Положення);

- Інструкція про порядок фіксації біометричних даних (параметрів) іноземців та осіб без громадянства посадовими особами Державної міграційної служби України, її територіальних органів і територіальних підрозділів, затверджена наказом МВС України від 23.11.2018 р. №944 (зареєстрована в МЮ України 17.12.2018 р. за №1428/32880).

Українське законодавство значно вужче захищає права українських громадян, якщо порівнювати з GDPR, BIPA і CCPA, в результаті чого можливі зловживання.

Хоча біометричні технології спрощують автентифікацію та мали б посилювати захист, фактичний збір і зберігання даних створюють нові загрози для нашої безпеки та водночас є причиною виникнення нових проблем (конфіденційності щодо збору та зберігання біометричних даних). Все частіше в ЗМІ ми чуємо про випадки викрадення та/або втрати біометричних даних.

1.3.2 Біометричні електрозамки на ринку України. Дверний замок зі зчитувачем відбитків пальців є найпоширеніщим типом біометричних електрозамків. Головною перевагою використання замків із вбудованим зчитувачем відбитків пальців є безпека.

У наведеному нижче списку ми розглянемо кращі дверні замки, які відкриваються при розпізнаванні зареєстрованих в системі відбитків пальців.

Біометричний замок Ardwolf A20 – це замок на батарейках, який оснащений ригелем для зниження ймовірності злому. Цей замок також надає домовласникам і власникам комерційних підприємств безліч засобів доступу: за допомогою відбитків пальців, кодів, механічних ключів та їх комбінацій для більш надійної подвійної аутентифікації. Іншою перевагою цього замку є його ціна, яка дозволяє багатьом домовласникам і власникам малого бізнесу використовувати біометричні технології, не виходячи за межі своїх бюджетів.

Біометричний замок Adel 3398 дуже схожий на замок Ardwolf, який був описаний вище, і з легкістю виконує більшість тих самих функцій. Крім того, ціна дверного замка робить його ще привабливішим для домовласників.

Розробниками запропоновано три методи доступу: відбитки пальців, пароль і традиційний – використання механічного ключа. У пристрої може зберігатися 100 відбитків пальців і тільки один пароль. Це може здатися недоліком, але не для всіх.

Більшість господарів будинків не будуть використовувати пароль, тому що легше керувати замком за допомогою відбитків пальців. Такого ж принципу можуть дотримуватися керівники компаній і також використовувати замок без ключа. Офісне приміщення буде в більшій безпеці завдяки тому, що співробітники компанії будуть відкривати двері своїми відбитками пальців.

Електрозамок U-tec Ultraloq UL3 – відносно дешеве рішення. Низька ціна пояснюється тим, що замок призначений для житлових будинків і невеликих комерційних приміщень. Житлові будинки потребують тих же рішень управління доступом, що і більшість комерційних організацій, тому Ultraloq прагне забезпечити правильний функціональний баланс.

U-tec Ultraloq найкраще підходить для власників малого бізнесу, які хочуть використовувати дверний замок із відбитками пальців на внутрішніх дверях. Саме з цієї причини цей замок входить до нашого переліку як один із кращих замків, які підвищують безпеку. Замок має рейтинг вологозахищеності і пилозахищеності IP65, а також вдосконалену систему зчитування відбитків пальців, в якій можуть зберігатися ідентифікатори 90 користувачів.

Врізний електрозамок Yale YDM4109. Одна з найяскравіших особливостей цього замку – його зовнішній вигляд. Yale YDM4109 пропонує кілька способів доступу, хоча і позиціонується як пристрій, який відкривається за допомогою відбитків пальців. Дверний замок Yale YDM4109 відкривається одним дотиком пальця, використовуючи прогресивне сканування. Прогресивне сканування – це корисна технологія, що допомагає підвищити вашу безпеку, а також підвищує простоту доступу. Функція прогресивного сканування точніше фіксує відбитки пальців. Цей замок має кілька функцій,

які роблять його одним із кращих дверних замків із функцією зчитування відбитків пальців, що підвищують вашу безпеку. Замок оснащений високонадійним врізним циліндровим механізмом.

Крім того, замок YDM4109 має сигналізацію, яка спрацьовує під час злому або пошкодження механізму. Єдиним недоліком цього замку є те, що він має резервний механічний ключ, як і більшість замків із відбитками пальців.

Marks USA 175 BIO Series. Виробник замку 175 BIO заявив про усунення ризику несанкціонованого доступу. Це те, чого прагне кожен виробник біометричних замків, але далеко не всі з них можуть цього досягти. У дверному замку є пристрій для зчитування відбитків пальців, який може зберігати дані 100 користувачів, також в ньому ведеться журнал на 4000 подій. Це зменшує ймовірність непомітного несанкціонованого доступу.

Електрозамок Westinghouse RTS. Замок зі зчитувачем відбитків пальців RTS, в першу чергу, орієнтований на користувачів систем контролю доступу з високими вимогами до безпеки. Він поставляється з оптичним датчиком відбитків пальців, який працює в поєднанні з пін-кодом і механічним ключем. Дверний замок використовує оптичний датчик CMOS, який служить для перетворення світла в електричний сигнал.

Електрозамок ZKAccess TF1700. У більшості замків, про які ми говорили раніше, є одна вада - вона полягає в наявності замкової щілини. Як відомо, будь-який замок можна зламати, особливо, якщо в ньому є замкова щілина. ZKAccess TF1700 виділяється серед інших замків, перш за все, приділяючи пріоритетну увагу безпеці. Цей замок добре підходить як для зовнішнього використання, так і для внутрішнього використання. TF1700 має рейтинг IP65 водонепроникності і захисту від атмосферних впливів. Замок ZKAccess має вражаючу швидкість розпізнавання користувачів за 1,5 секунди.

Біометричний замок Samsung SHS-P718 (SHS) є одним із кращих рішень, що з'явилися на ринку. Не варто очікувати, що цей замок буде бездоганним. Однак, безсумнівно, він зручний і надійний у використанні. Він

може бути встановлений не тільки в організаціях, але і в житлових будинках, проте, його ціна сподобається не всім домовласникам.

Користувачі можуть відчинити замок за допомогою відбитків пальців, RFID-карти або пін-коду. Основний метод доступу – це відбиток пальця, але цей замок має режим подвійної аутентифікації, який дає вам можливість об'єднати відбиток пальця з іншим методом доступу. Крім того, SHS дає можливість зберігати 30 пін-кодів і 100 відбитків пальців, що робить його ідеальним рішенням для малих підприємств, які хочуть досліджувати різні рішення для контролю доступу.

На відміну від деяких інших своїх колег, SHS не використовує механічні ключі як метод доступу. Однак, в комплект входять 5 механічних ключів, які можуть використовуватися в екстреному випадку, якщо розрядиться батарея або виникне будь-яка інша пов’язана з електроживленням проблема і порушить його нормальне функціонування.

На даний момент сканерів відбитків пальців існує декілька різновидів: ємнісні (capactive); оптичні (optical); ультразвукові (ultrasound); термічні (termal).

Ємнісні датчики вважаються найпрактичнішими у використанні та працюють за рахунок масиву магнітних конденсаторів. Оскільки шкіра людини є доволі хорошим провідником, це дає можливість забезпечити контакт з індивідуальним ємнісним елементом на масиві. Підвищення на пучці пальця, будучи ближче до конденсатора, має вище ємність, а впадини – мають нижчу ємність. На деякі з цих датчиків подають невелику напругу до пальця, щоб посилити сигнал, таким чином створюється кращий контраст зображення.

Шкіра людини є достатньо провідною і здатна забезпечити ємнісний зв'язок у поєднанні з індивідуальним ємнісним елементом на масиві. Фізичні гребні відбитків пальців знаходяться ближче до пластин конденсатора і мають більшу ємність, тоді як впадини відбитків пальців, тобто глибший шар, мають меншу ємність.

1.4 Постановка завдання

У процесі опрацювання наукової інформації та результатів інноваційної діяльності передових підприємств у ІТ сфері, було визначено завдання розробити програмний продукт «Програмне забезпечення управління електронним замком з використанням біометричних данних користувача».

У процесі розробки розглянути наступні питання:

1. Аналіз предметної області, а саме процесу формування розкладу
2. Моделювання предметної області
3. Проектування баз даних, яка буде забезпечувати збереження необхідної для роботи системи інформації
4. Реалізація бази даних в обраній СУБД
5. Розробка алгоритмів обробки необхідних даних
6. Розробка модулів програмної системи відповідно до розроблених алгоритмів
7. Тестування програмної системи
8. Впровадження системи

1.5 Моделювання предметної області

Сучасне виробництво в усіх сферах характеризується високою складністю та різноманіттям технологічних процесів. Воно вже неможливе без створення систем керування, які забезпечують його ефективність, надійність та безпеку.

Система – це сукупність взаємопов'язаних між собою складових частин, яка характеризується спільною метою функціонування.

Проектування систем керування – складний багатоетапний процес, у якому беруть участь фахівці різного профілю і кваліфікації. Тому одна з головних проблем проектування – забезпечення спільної мови фахівців, яка дозволяє однозначно, чітко і зрозуміло сформулювати основні концепції проекту. Така мова була створена в ході розв'язання задач управління проектами і автоматизації розробки програмного забезпечення, але її основні засоби можуть бути застосовані для проектування будь-яких комп'ютеризованих систем, які містять як програмні, так і апаратні засоби. Тому ця мова отримала назву UML – Universal Modeling Language.

Наразі мова UML стала міжнародним стандартом проектування комп'ютеризованих систем. Зрозумілість і прозорість проектів, які описані мовою UML забезпечується використанням графічних засобів – так званих UML-діаграм. Такий підхід виправляє розповсюджений у останні десятиріччя стиль проектування з мінімізацією використання графічних зображень.

Процес проектування з використанням UML-діаграм ґрунтується на двох базових принципах: орієнтації на об'єктне подання системи (тобто уявленні про систему як сукупність окремих об'єктів, які взаємодіють один з одним) і ітераційності проектування (тобто відмові від намагання з першого кроку передбачити усі функції, властивості і характеристики системи, а уточнення і розвиток системи в процесі розробки).

Діаграма – графічне уявлення набору елементів, найчастіше зображеного у вигляді зв'язного графа вершин (сутностей) і шляхів

(відношень/зв'язків). Діаграми використовуються для візуалізації системи з різних точок зору, тому окрема діаграма – це проекція системи.

Діаграма дає певний згорнутий погляд про елементи системи. Один і той же елемент (сутність/відношення) може з'являтися або в усіх діаграмах, або в деяких. Теоретично діаграма може містити в собі будь-яку комбінацію сутностей і відношень/зв'язків.

На практиці використовується лише невелике число загальних комбінацій. Перші п'ять із цих діаграм найчастіше застосовуються для побудови архітектурних виглядів програмних систем.

Виходячи з цього, подаємо короткий опис 6 видів найчастіше використовуваних для розроблення моделей та архітектурних виглядів ПЗ UML -діаграм.

Діаграма класів (Class diagram) показує набір класів, інтерфейсів і кооперацій, а також їх зв'язки. Діаграми цього виду найчастіше використовуються для моделювання об'єктно-орієнтованих систем і призначені для статичного зображення системи. Діаграми класів, що включають активні класи, описують статичний вигляд процесів системи. Діаграми компонентів є різновидом діаграм класів.

Діаграма об'єктів (Object diagram) показує набір об'єктів і їх зв'язки. Діаграми об'єктів представляють статичні копії станів екземплярів сутностей, описаних у діаграмі класів. Також представляють статичний вигляд проектування або процесів системи (як і діаграми класів, але з погляду реальних або прототипних ситуацій).

Діаграма компонентів (Component diagram) демонструє інкапсульовані класи та їх інтерфейси, порти і внутрішні структури, що складаються із вкладених компонентів і коннекторів. Діаграми компонентів описують статичний вигляд з точки зору реалізації системи. Вони важливі при побудові складних систем. Поряд з діаграмою компонентів UML дозволяє створювати діаграму складеної структури (composite structure diagram), що застосовуються до будь-якого класу.

Діаграма варіантів використання (Use case diagram) демонструє набір ВВ і діючих осіб-акторів (актантів) (які є спеціальним видом класів), а також їх зв'язки. Вони описують статичний вигляд ВВ системи і важливі для організації та моделювання поведінки системи.

Діаграма взаємодії (Interaction diagram) показує взаємодію, що складається з набору об'єктів і виконуваних ними ролей, включаючи повідомлення, які можуть передаватися між ними. Вони призначені опису динамічного стану системи. Діаграми взаємодії поділяються на діаграми послідовностей і діаграми комунікацій.

Діаграма послідовності (sequence diagram) – різновид діаграми взаємодії, що показує тимчасову послідовність повідомлень.

Діаграма діяльності (Activity diagram) показує структуру процесу або інших обчислень як покроковий потік керування і даних. Діаграми діяльності описують динамічний вигляд системи, є важливі при моделюванні функцій системи, виділяючи потік керування між об'єктами.

Діаграма розміщення (Deployment diagram) показує конфігурацію вузлівпроцесорів, а також розташовані на них компоненти. Діаграми розміщення дають статичний вигляд розміщення архітектури. Вузли містять один або кілька артефактів.

Діаграма пакетів (Package diagram) показує декомпозицію моделі на організаційні одиниці та їх залежності

1.5.1 Діаграми UML для предметної області. *Діаграма Прецедентів* візуально зображає різноманітні сценарії взаємодії між акторами (користувачами) і прецедентами (випадками використання); описує функціональні аспекти системи (бізнес логіку) (рис. 1.1).

Діаграми Прецедентів відіграють важливу роль не тільки у комунікації між збирачами вимог до проекту і потенційними користувачами. Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання.

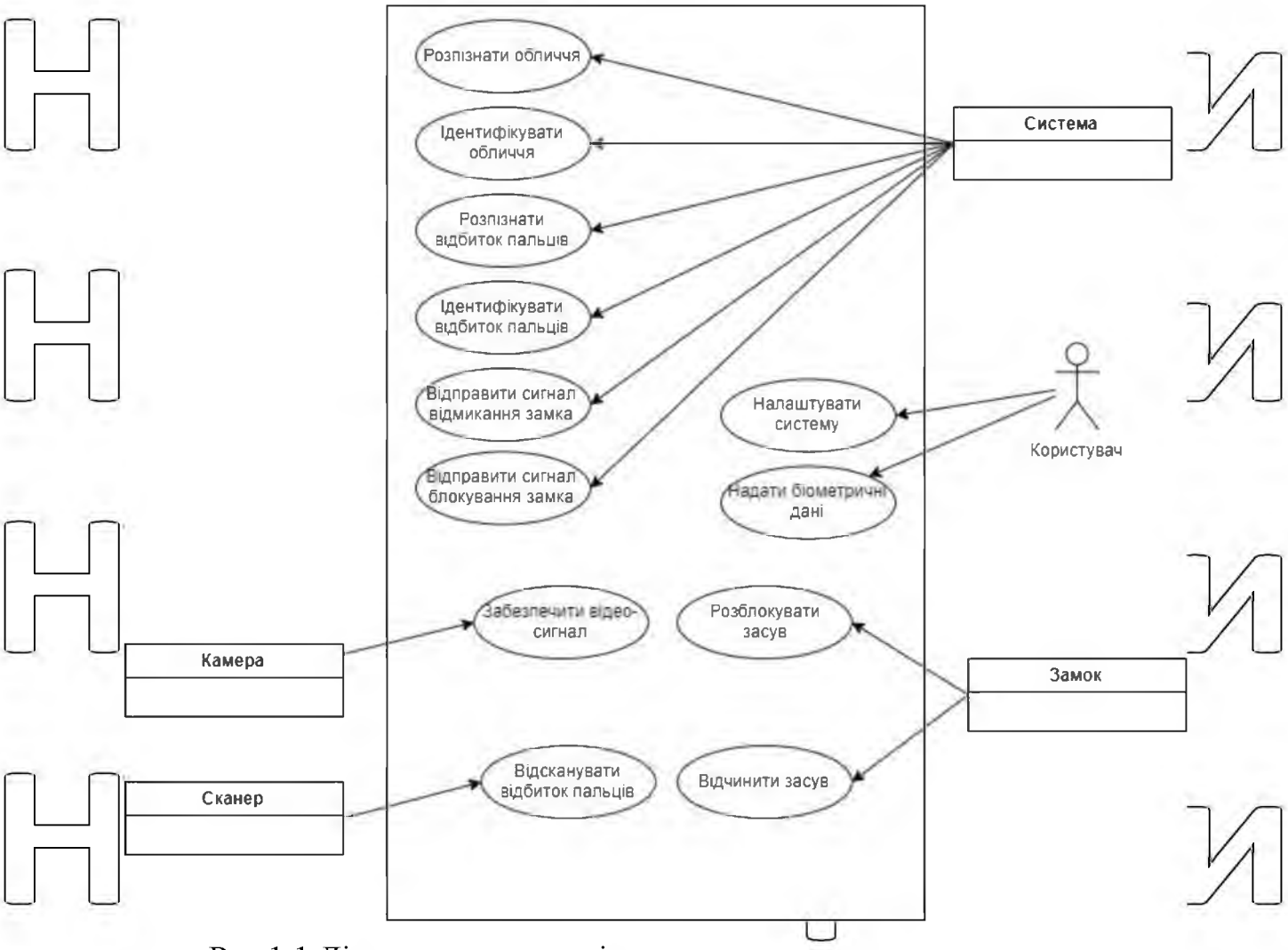


Рис.1.1 Діаграма прецедентів

Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Діаграми Прецедентів доповнені бізнес логікою і детальними специфікаціями прецедентів, як джерельна інформація, успішно використовують учасники розробки проекту на всіх його фазах (зародження, дизайн, програмування, тестування, документування). Добре продумані і завершені специфікації прецедентів легко перевтілюються у тестові випадки.

Діаграма послідовності – відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність

відправлених повідомлень (рис. 1.2). Іншими словами, діаграма послідовностей відображає часові особливості передачі і прийому повідомлень об'єктами.

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертаються результати, які пов'язані з повідомленнями.

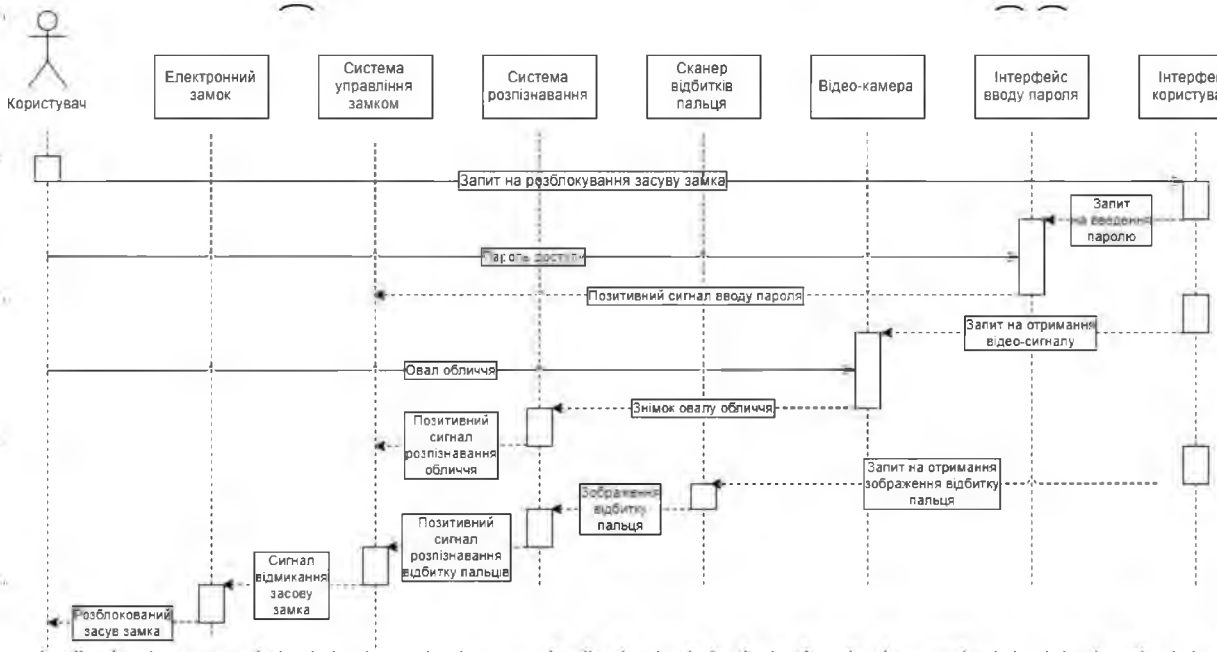


Рис. 1.2. Діаграма послідовностей

На діаграмі послідовності зображуються тільки ті об'єкти, які безпосередньо беруть участь у взаємодії.

Сценарії подій

Оптимістичний сценарій

А. Користувач намагається відімкнути сейф, зачинений електронним замком з трирівневим захистом: код доступу, сканер овалу лиця, сканер відбитків пальців.

Б. Користувач вводить код доступу, поміщає обличчя у зону огляду камери та прикладає палець до сканеру відбитків.

B. Система розпізнає та підтверджує особистість користувача та надає йому доступ.

Г. Електронний замок відмикає засув.

Д. Система робить запис у журнал про вдалу спробу отримання доступу.

Е. Користувач зачиняє сейф.

Є. Електричний замок блокує засув.

Ж. Система переходить у режим очікування.

Прагматичний сценарій

Умова 1. При спробі отримання доступу, користувач зробив помилку при введенні коду доступу.

В1. Показати повідомлення про неправильне введення коду доступу.

В2. Якщо користувач вводить код доступу правильно, перейти до пункту В.

В3. Якщо користувач вводить код доступу неправильно другий раз, вивести повідомлення про помилку, та попередження про те, що залишилось 1 спроба введення.

В4. Якщо користувач вводить код доступу правильно, перейти до пункту В.

В5. Якщо користувач вводить код доступу неправильно, заблокувати доступ до системи на 5 хвилин.

В6. Після закінчення блокування, надати користувачу ще три спроби.

В7. Якщо користувач вводить код доступу правильно, перейти до пункту В.

В8. Якщо користувач вводить код доступу неправильно ще 3 рази, заблокувати систему до введення ключа адміністратора

Діаграма взаємодії показує потік повідомлень між об'єктами системи й основні асоціації між ними й по суті, як уже було сказано вище, є альтернативою діаграми послідовностей. Діаграма об'єктів показує статику, якийсь знімок системи, зв'язок між об'єктами в цей момент часу, діаграма ж

взаємодії, як і діаграма послідовностей, показує взаємодію об'єктів у часі, тобто у динаміці (рис. 1.3).

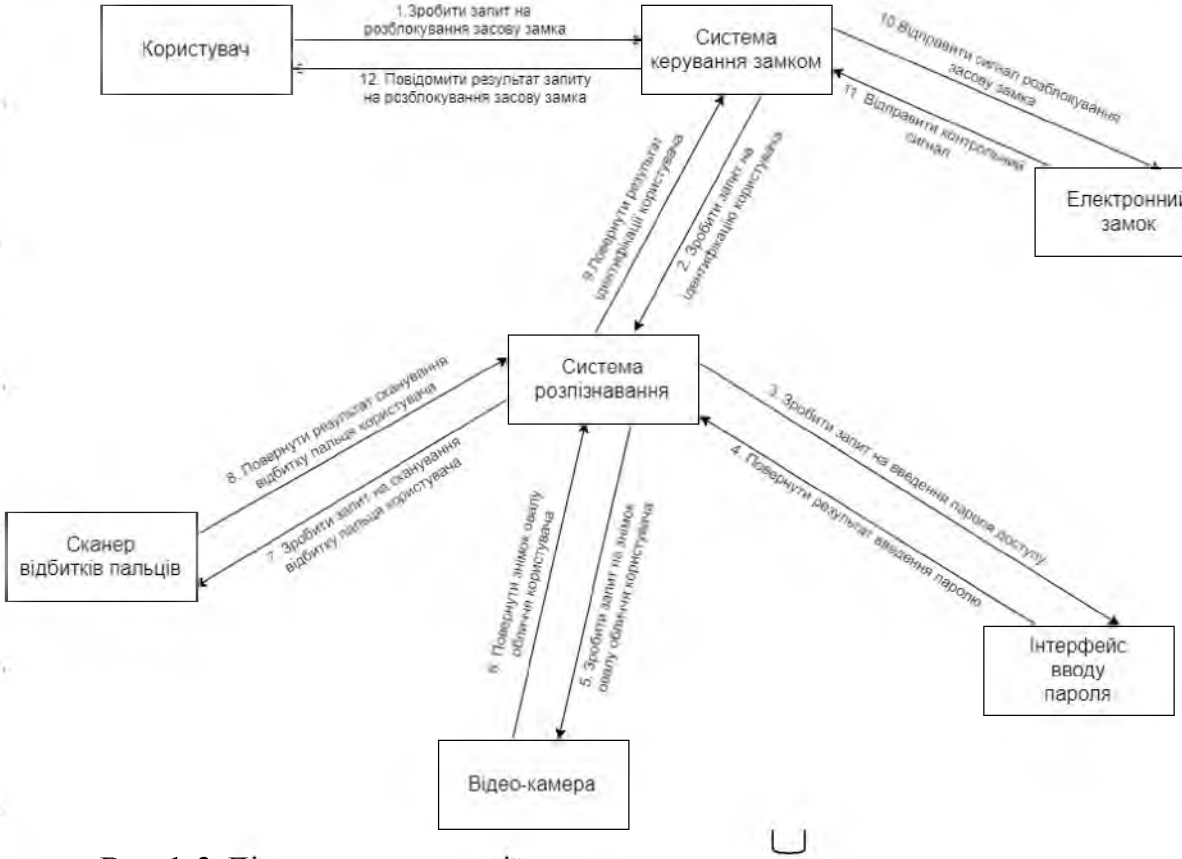


Рис.1.3 Діаграма взаємодії

Слід зазначити, що використання діаграми послідовності або діаграми взаємодії – особистий вибір кожного проектувальника й залежить від індивідуального стилю проектування. На позначеннях, застосовуваних на діаграмі взаємодії не варто зупинятися докладно. Тут все стандартно: об'єкти позначаються прямокутниками з підкресленими іменами (щоб відрізнити їх від класів), асоціації між об'єктами вказуються у вигляді з'єднуючих їхніх ліній, над ними може бути зображена стрілка із вказівкою назви повідомлення і його порядкового номера. Необхідність номера повідомлення пояснюється дуже просто - на відміну від діаграми послідовностей, час на діаграмі взаємодії не показується у вигляді окремого виміру. Тому послідовність передачі повідомлень можна вказати тільки за допомогою їхньої нумерації.

Діаграма активності (діяльності) – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій (1.4).

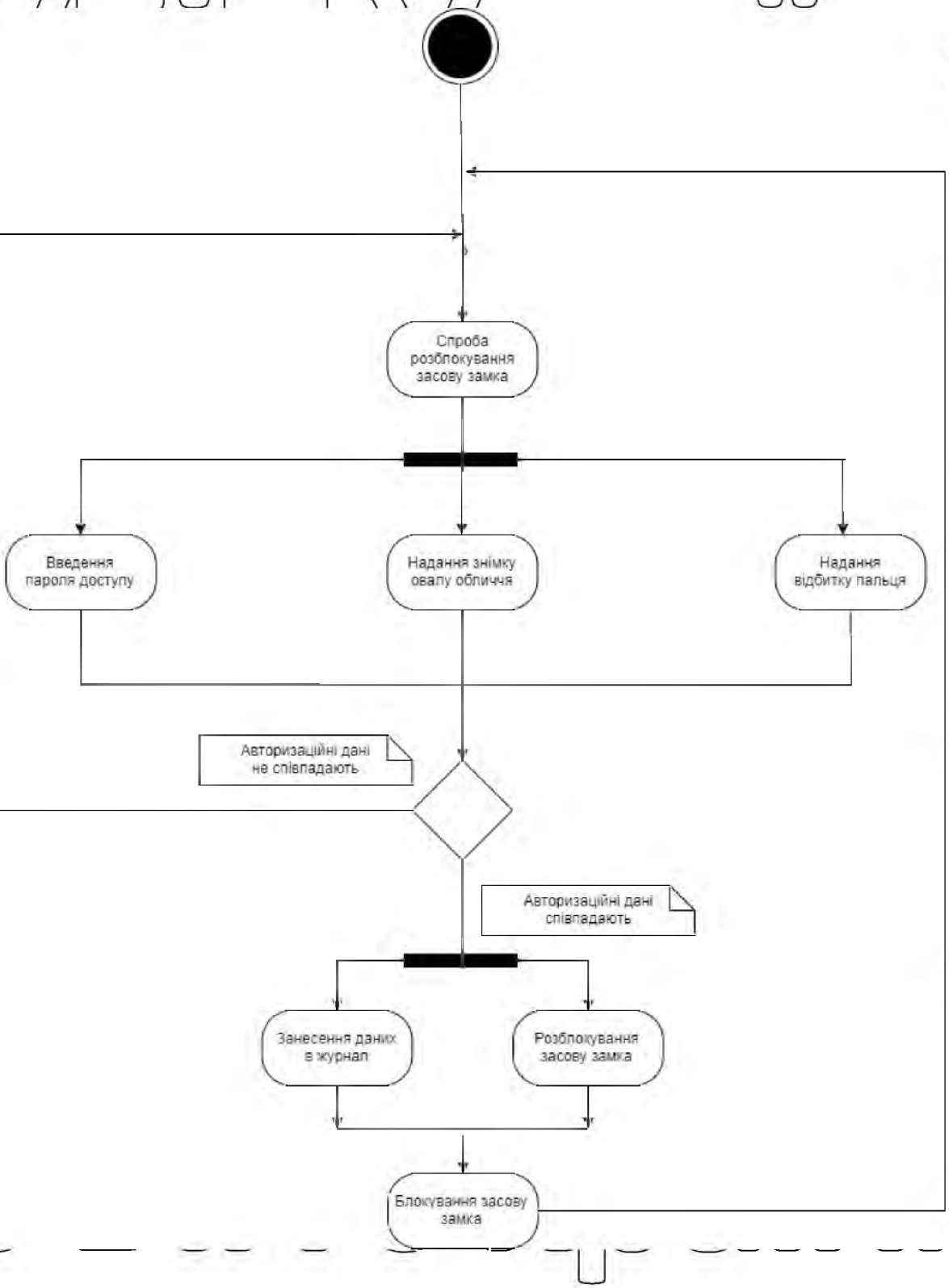


Рис.1.4 Діаграма активності

Дія (активність) є фундаментальною одиницею визнання поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків.

1.5.2. Моделювання програмної системи. *Діаграма пакетів* – відображає залежності між пакетами, з яких і складається модель (рис. 1.5).

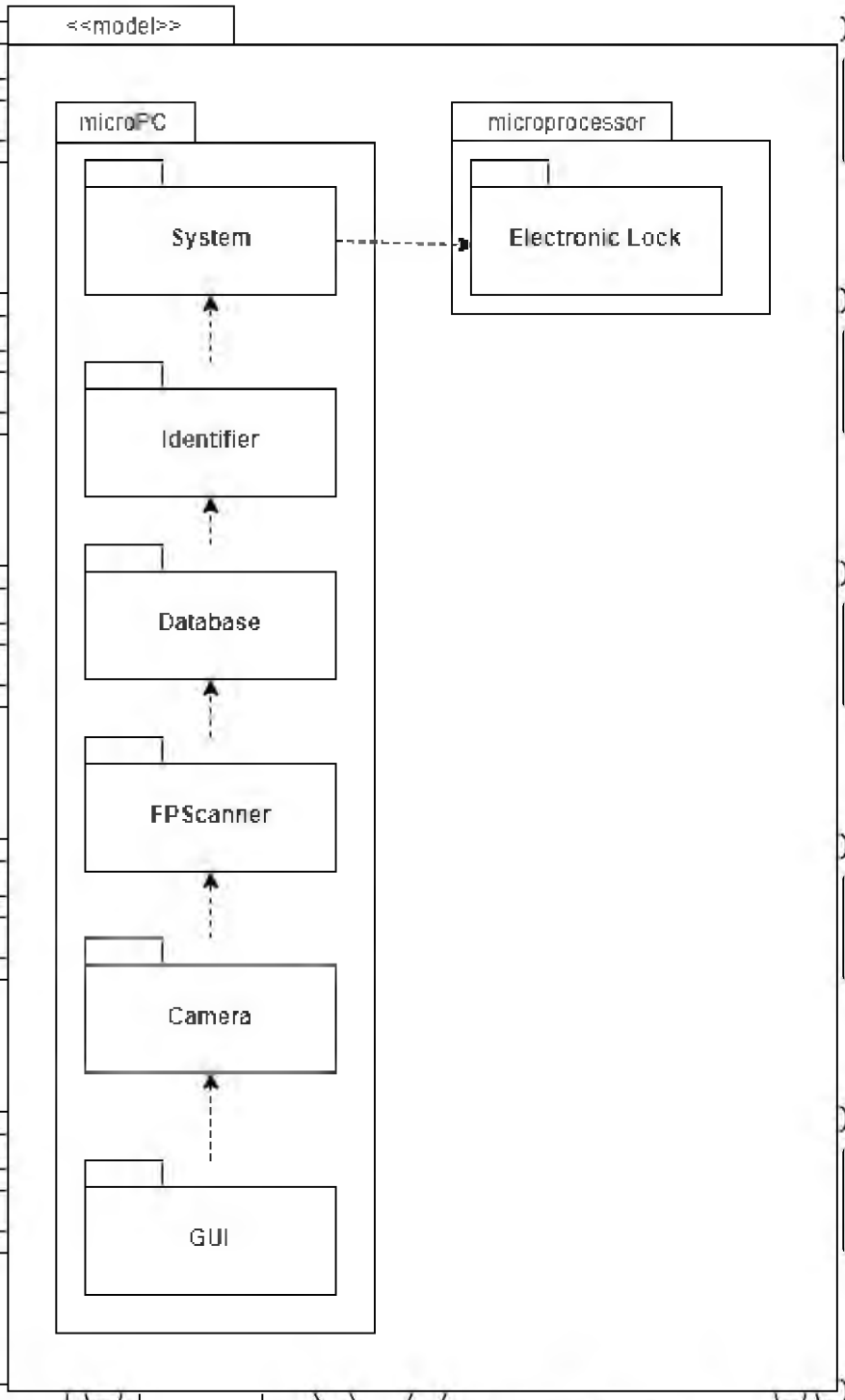


Рис. 1.5 Діаграма пакетів

Пакет (package) – елемент моделі, який використовують для групування інших елементів моделі. Елементи моделі, які входять у склад певного пакету, називаються членами пакету. Пакет володіє усіма своїми членами. Про членів пакету кажуть, що вони є у власності пакету, тобто належать йому. Якщо певний пакет видаляється з моделі, то з неї також видаляються усі члени, що знаходяться у власності цього пакету.

Діаграми пакетів можуть використовувати пакети, які містять прецеденти для демонстрації функціональності програмного забезпечення системи. Діаграми можуть використовувати пакети, які показують різноманітні шари програмного комплексу для ілюстрації його архітектури, що складається з різних шарів. Залежності між цими пакетами можуть бути наділені позначками , щоби вказати механізм зв'язку між шарами.

Діаграма компонентів – в UML, діаграма, на якій відображаються компоненти, залежності та зв'язки між ними (рис. 1.6).

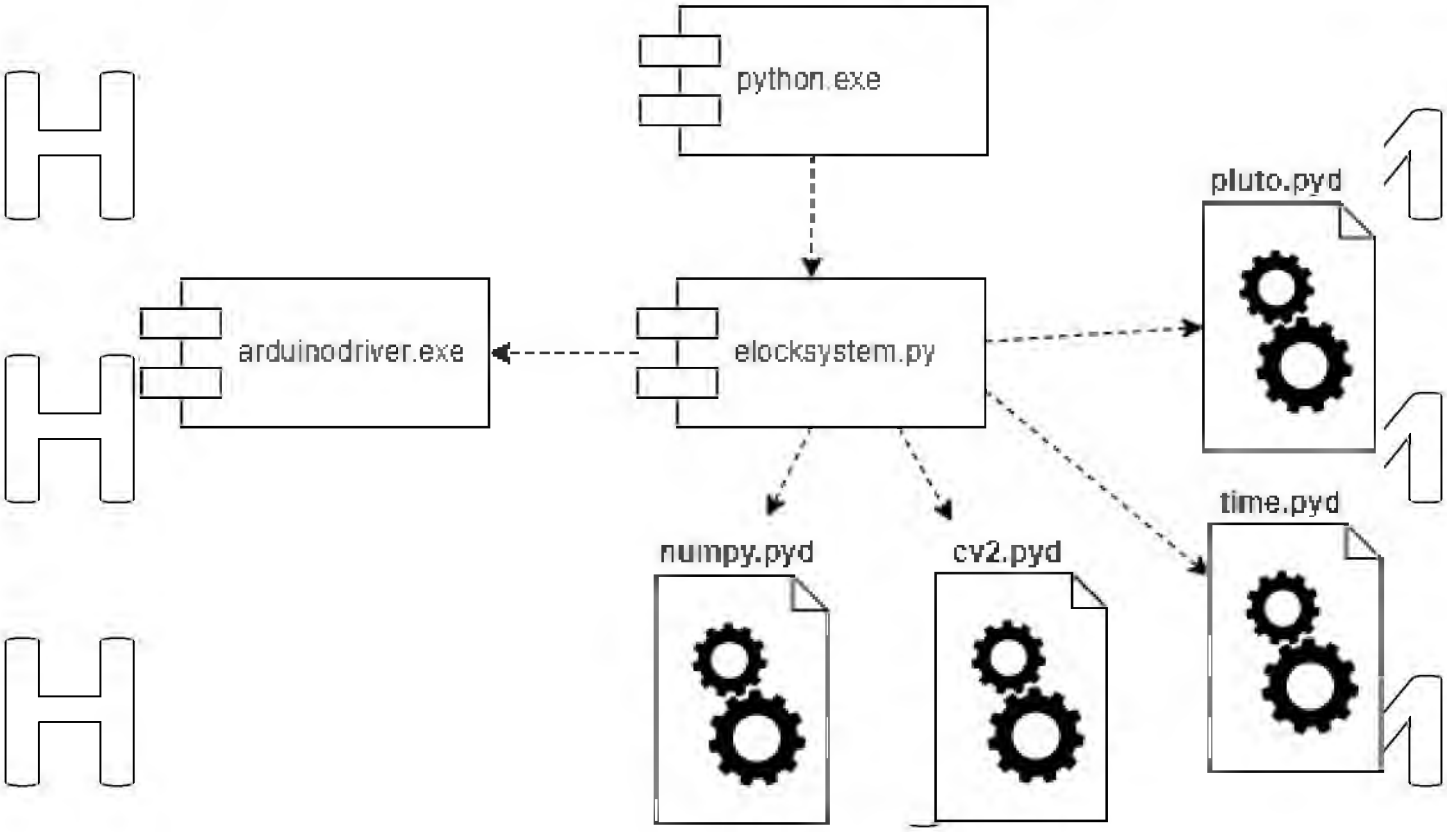


Рис. 1.6. Діаграма компонентів

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми. Діаграма компонентів відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Діаграма класів відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення (рис. 1.7).

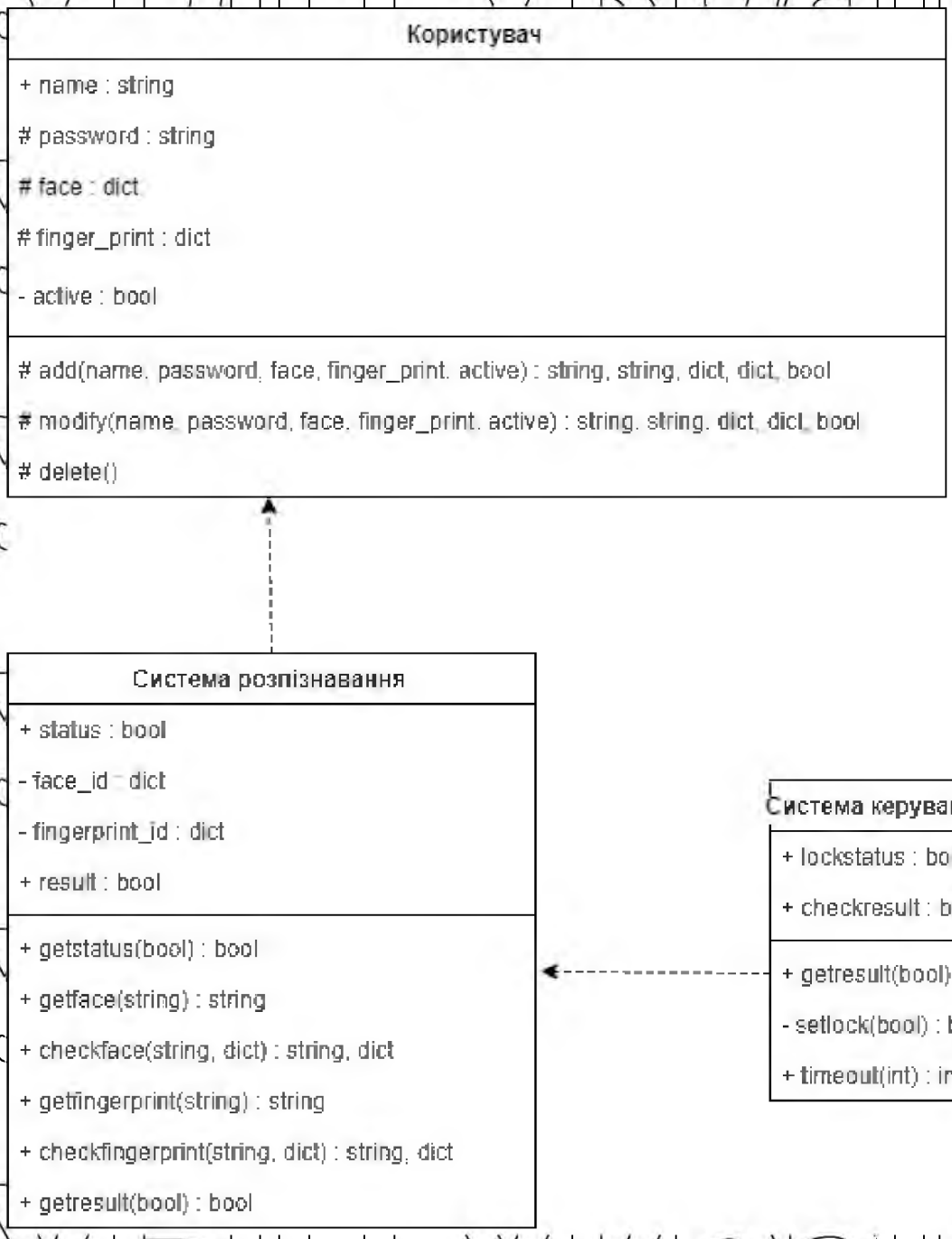


Рис. 1.7. Діаграма класів

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів

об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Діаграма об'єктів – це діаграма, що показує об'єкти і їх стосунки в певний момент часу. Діаграма об'єктів графічно представлена у вигляді графів з вершинами і ребрами (рис. 1.8).

Діаграма об'єктів є по суті екземпляром діаграми класів або статична частина діаграми. У будь-якому випадку, вона містить об'єкти та зв'язки, і зосереджена на конкретних екземплярах, або екземплярах прототипів.

Проектуючи діаграми класів, компонентів або розгортання, описуються групи абстракцій і розкривається в цьому контексті їхня семантика та відносини з іншими абстракціями у групі. Ці діаграми відображають лише потенційні можливості. Наприклад, якщо клас А зв'язаний з класом В асоціацією типу «один до багатьох», то один екземпляр класу А може мати п'ять екземплярів класу В, а інший – лише один. Крім того, в будь-який момент часу екземпляр класу А та пов'язані з ним екземпляри класу В будуть мати конкретні значення своїх атрибутів.

Якщо уявити собі хвилину в житті системи, що моделюється, можна знайти набір об'єктів, кожен з яких є в певному стані і пов'язаний конкретними відносини з іншими об'єктами. Діаграма об'єктів дозволяє візуалізувати, специфікувати, проектувати та документувати структуру, що складається з цих об'єктів. Вона особливо корисна для моделювання складних структур даних.

Діаграма розгортання відображає обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах (рис. 1.9).

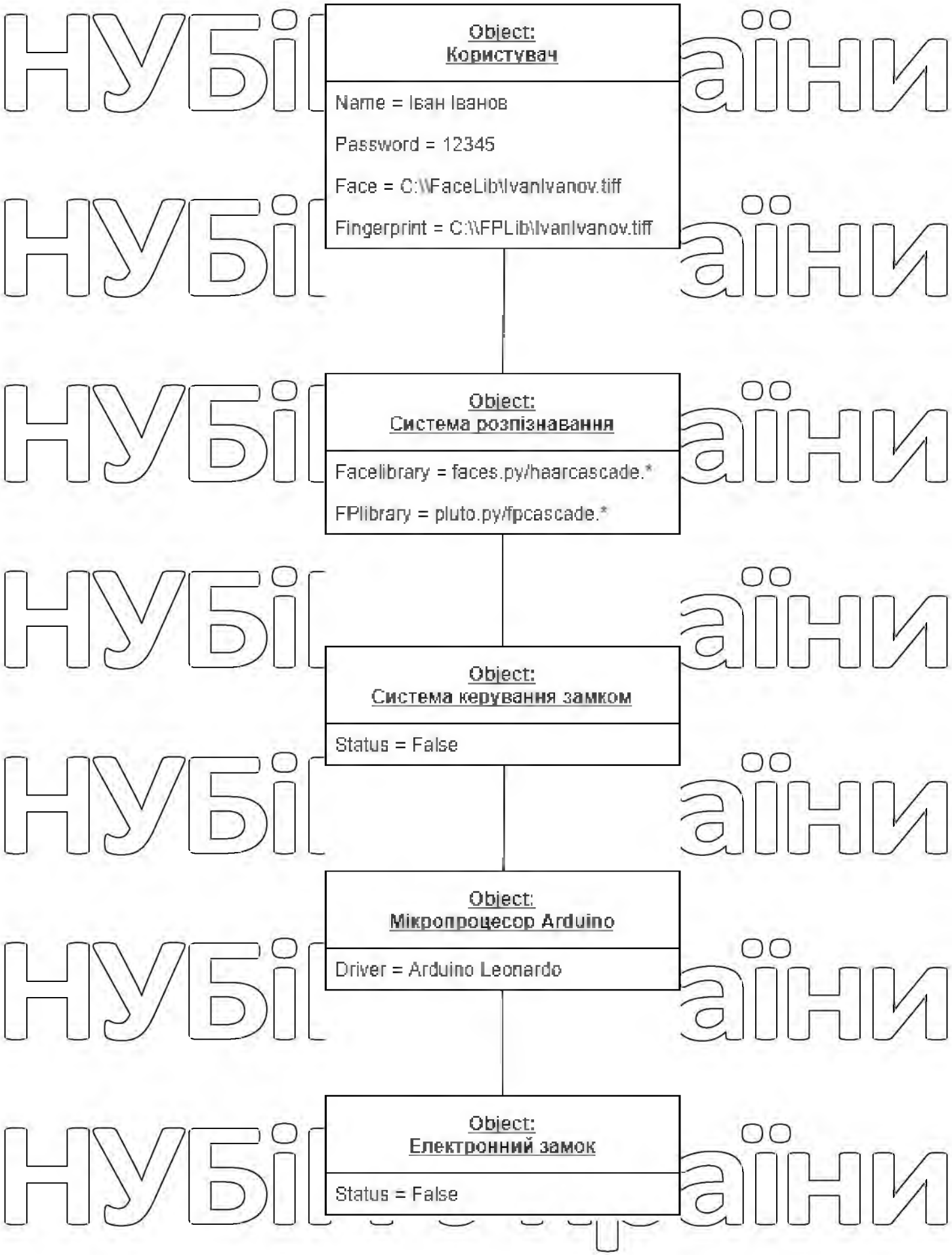

Object:
Користувач
Name = Іван Іванов
Password = 12345
Face = C:\\FaceLib\IvanIvanov.tiff
Fingerprint = C:\\FPLib\IvanIvanov.tiff
Object:
Система розпізнавання
Facelibrary = faces.py/haarcascade.*
FPlibrary = pluto.py/fpcascade.*
Object:
Система керування замком
Status = False
Object:
Мікропроцесор Arduino
Driver = Arduino Leonardo
Object:
Електронний замок
Status = False

Рис. 1.8. Діаграма об'єктів

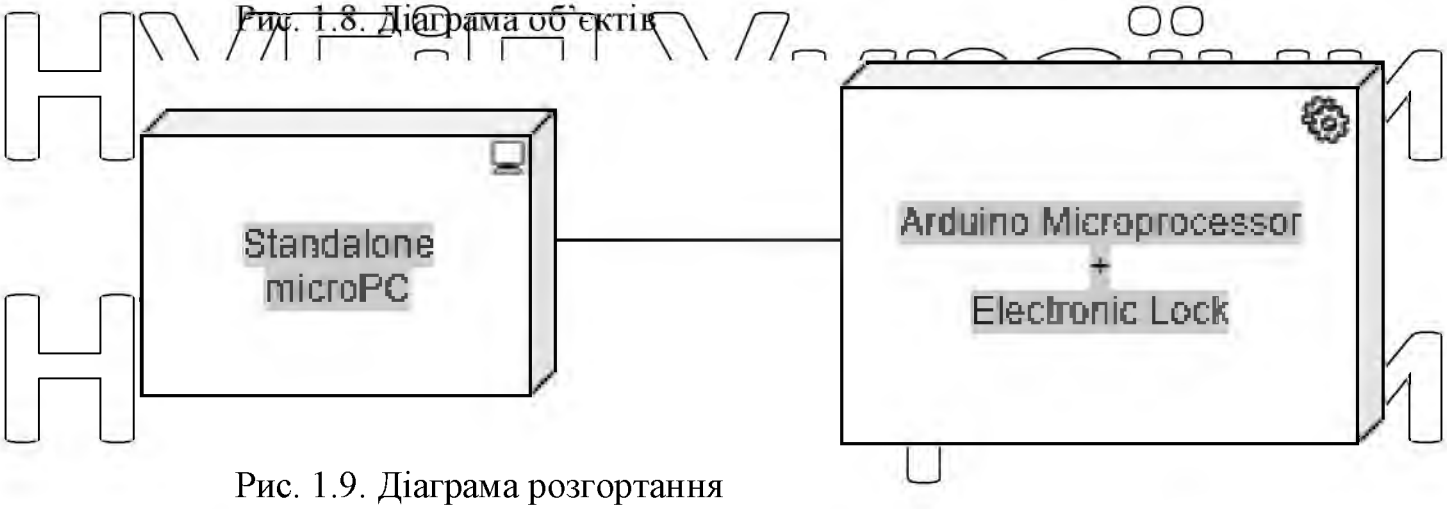


Рис. 1.9. Діаграма розгортання

Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються, натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

У результаті виконання першого розділу дипломного проекту було досліджено предметну область, вивчено основні принципи роботи систем розпізнавання овалу обличчя та відбитків пальців, а також механізм та принцип роботи електронного замка.

На підставі знань про предметну область було визначено основну функціональність системи, необхідну для використання біометричних даних користувача для керування електронним замком. Було проведено моделювання предметної та програмної області, за для поліпшення в майбутньому розробки самого програмного продукту.

РОЗДІЛ 2
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних

Логічна модель даних, або логічна схема – модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання (фізична модель даних), але в термінах структур даних, таких як реляційні таблиці та колонки, об'єктно-орієнтовані класи чи теги XML. Вона є протилежністю концептуальній моделі даних, яка описує семантику організації без посилання на технологію.

Логічні моделі даних подають абстрактну структуру області інформації. Вони часто мають схематичний характер і найтиповіше використовуються у бізнес-процесах, які прагнуть захопити речі, що мають важливе для організації значення, та як вони відносяться одна до одної. Одного разу перевірена та схвалена, логічна модель даних може стати основою фізичної моделі даних і сформувати дизайн бази даних.

Логічні моделі даних повинні засновуватися на структурах, визначених у попередній концептуальній моделі даних, оскільки вона описує семантику інформаційного контексту, яку логічна модель повинна також відображати. Навіть так, оскільки логічна модель передбачає реалізацію на конкретній обчислювальній системі, вміст логічної моделі даних коригується для досягнення певної ефективності.

Термін «логічна модель даних» іноді використовується як синонім «моделі предметної області» або як її альтернатива. Тоді як два поняття тісно пов'язані та мають цілі, що перекриваються, модель предметної області більше зосереджена на захопленні понять у предметній області, ніж на структурі даних, пов'язаній із цією областю.

Логічне проектування бази даних – процес конструювання інформаційної моделі підприємства на основі існуючих конкретних моделей даних, не залежних від використовуваної СКБД і інших фізичних умов реалізації.

Незалежно від типу біометричного ідентифікатора, який застосовується системою, загальний алгоритм функціонування системи біометричної ідентифікації може бути наданий у вигляді, показаному на рис. 2.1, а спрощена структурна схема системи біометричної ідентифікації показана на рис. 2.2.

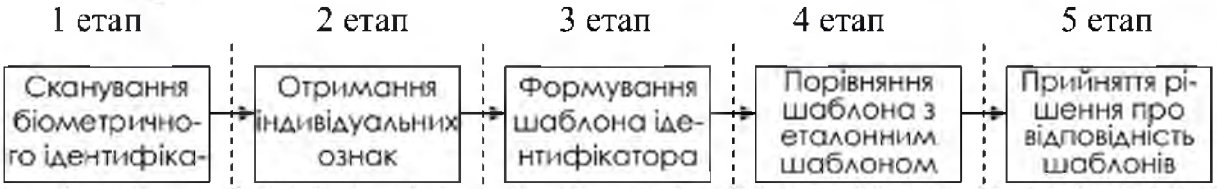


Рис. 2.1. Алгоритм функціонування систем біометричної ідентифікації

Рис. 2.2. Спрощена структурна схема системи біометричної ідентифікації

Логічне проектування бази даних (для реляційної моделі) наступне:

1. Побудова і перевірка локальної логічної моделі даних на основі уявлення про предметну область кожного з типів користувачів.
2. Перетворення локальної концептуальної моделі даних у локальну логічну модель.
3. Визначення набору відношень виходячи зі структури локальної логічної моделі даних.
4. Перевірка моделі за допомогою правил нормалізації.

5. Перевірка моделі у відношенні транзакцій користувачів.
6. Створення діаграм "сутність-відношення".
7. Визначення вимог підтримки цілісності даних.
8. Обговорення розроблених локальних логічних моделей даних з кінцевими користувачами.
9. Створення і перевірка глобальної логічної моделі даних.
10. Злиття локальних логічних моделей даних у єдину глобальну модель даних.
11. Перевірка глобальної логічної моделі даних.
12. Перевірка можливостей розширення моделі в майбутньому.
13. Створення остаточного варіанта діаграми "сутність-відношення".
14. Обговорення глобальної логічної моделі даних з користувачами.

2.2 Принципи функціонування системи біометричної ідентифікації

У загальному випадку системи біометричної ідентифікації працюють за наступним принципом. Усі системи спочатку працюють у режимі реєстрації, тобто спочатку система повинна отримати та зберегти певний біометричний ідентифікатор, за допомогою якого надалі буде здійснюватися ідентифікація користувача. В залежності від типу системи вона може використовувати декілька біометричних ідентифікаторів (наприклад, якщо здійснюється ідентифікація за відбитками пальців, або за параметрами ока).

Після отримання біометричного ідентифікатора система перетворює його за допомогою відповідних засобів в електронний вигляд. Ця стадія роботи системи біометричної ідентифікації називається реєстрація, тобто система отримує первісну інформацію, необхідну для її подальшої роботи. Звичайно система біометричної ідентифікації не зберігає зображення відбитка пальця, сітківки ока, геометрії долоні і т. ін.

У системі зберігається так званий шаблон ідентифікатора, який являє собою одну або декілька цифрових послідовностей, які були отримані під час

оброблення біометричного ідентифікатора. Тобто, біометричний ідентифікатор, який надав користувач через спеціальний пристрій-реєстратор перетворюється в електронний вид, який потім проходить декілька стадій оброблення за різними алгоритмами (тип алгоритму та кількість обробок залежить від типу біометричного ідентифікатора), внаслідок чого отримується шаблон, за допомогою якого потім здійснюється безпосередньо процедура ідентифікації користувача.

Після того, як процес реєстрації здійснено, система здатна проводити процес ідентифікації, тобто встановлення відповідності особи та визначення її прав на виконання тих чи інших дій. На рис. 2.3 наданий загальний алгоритм роботи системи біометричної ідентифікації.

Рис. 2.3. Загальний алгоритм роботи системи біометричної ідентифікації

Слід зазначити, що процес ідентифікації у біометричних системах в цілому поділяється на два види – ідентифікацію та верифікацію. Звичайно різниця між цими двома поняттями надто тонка і досить часто один процес плутають з іншим.

Ідентифікація – це порівняння типу "один-до-багатьох", тобто здійснюється порівняння наданого біометричного ідентифікатора з усіма шаблонами біометричних ідентифікаторів, які є у базі. У результаті цього

порівняння виявляється декілька найбільш схожих шаблонів (ті, які мають найбільшу вірогідність відповідності), а потім за допомогою будь-якого математичного критерію приймається рішення про найбільш ідентичний шаблон.

Верифікація – це порівняння типу “один-до-одного”, тобто здійснюється порівняння наданого ідентифікатора з відповідним шаблоном з бази. Однак в даному випадку необхідно надати додатковий ідентифікатор, який дозволить обрати з бази відповідний шаблон. Наприклад, спочатку вводиться логін користувача, а потім надається відповідний біометричний ідентифікатор. У даному випадку система відповідає на питання: «Чи дійсно ви та людина за яку себе видаєте?». У цьому режимі система ідентифікації працює набагато швидше та в повністю автоматичному режимі.

Слід зазначити, що такий принцип роботи систем ідентифікації дозволяє розділити їх за сферами використання. Наприклад, систему у режимі ідентифікації використовують криміналісти для встановлення, або системи безпеки на вокзалах та аеропортах, системи у режимі верифікації використовують у системах контролю доступу до ресурсів комп’ютерних мереж, до ресурсів електронних платіжних системи і т. ін.

Цілком зрозуміло, що система біометричної ідентифікації здатна працювати у двох режимах, проте більшість систем працює саме у режимі верифікації. Необхідно зазначити, що може виникнути ситуація, коли еталонний зразок біометричного ідентифікатора, який зберігається у базі, не буде збігатися зі знов запропонованим біометричним ідентифікатором.

Це може бути пов’язано, в першу чергу, з тим, що при повторному наданні біометричного ідентифікатора користувач трохи змінив геометричні умови його надання, наприклад при ідентифікації за відбитками пальців він міг докласти пальця до сканера під іншим кутом, або при ідентифікації за сітківкою ока він міг нахилити голову і таким чином змінити кут падіння світла в око.

На рис. 2.4 надано загальну схему роботи системи біометричної ідентифікації з урахуванням особливостей процесу ідентифікації

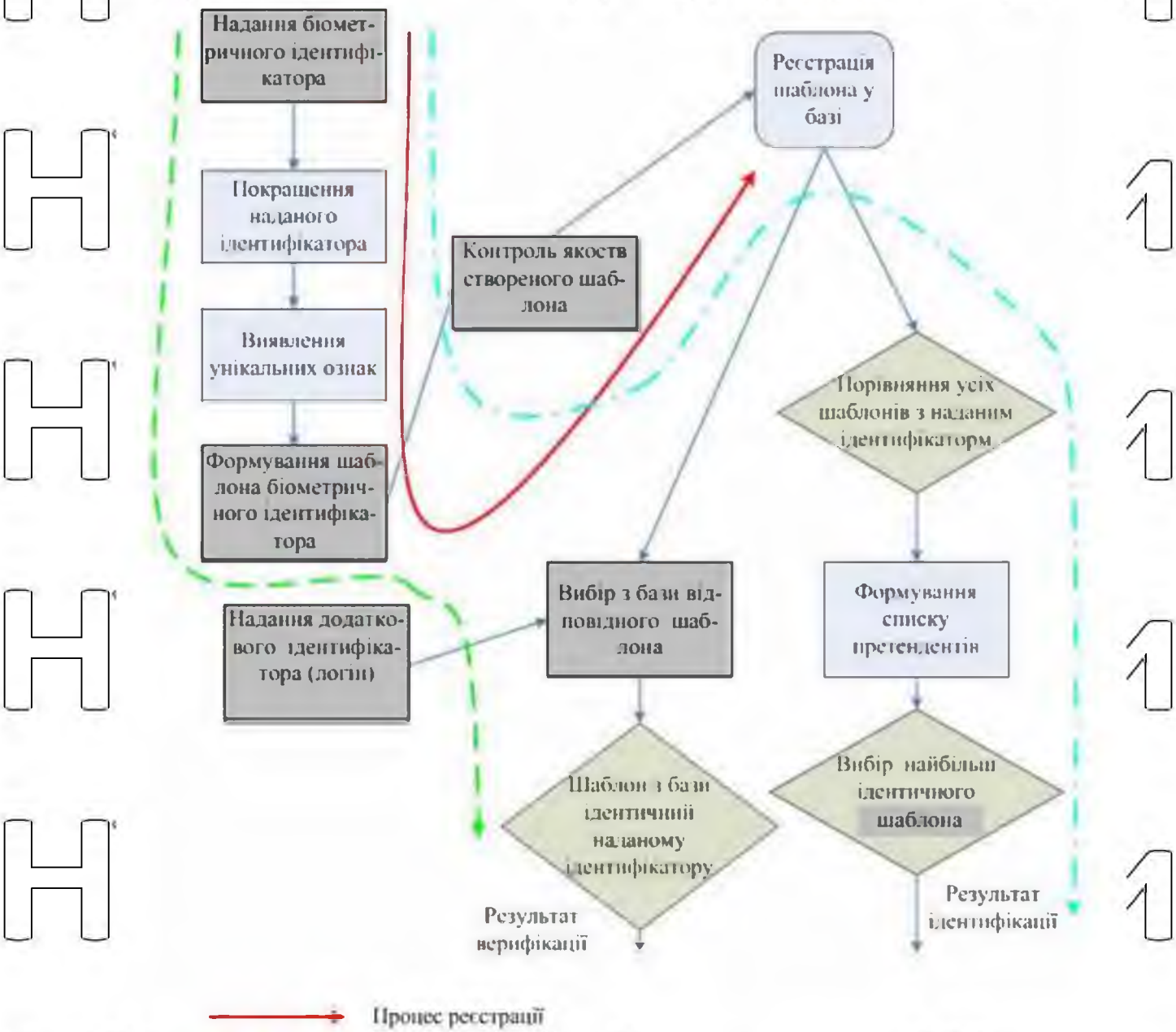


Рис.2.4 Загальна схема функціонування системи біометричної ідентифікації

Отже, щоб уникнути невдалої ідентифікації через подібні обставини кожна система біометричної ідентифікації має у своєму складі пристрій, який забезпечує покращення електронного зображення біометричного ідентифікатора до рівня встановленого стандартом, тільки після цього

розпочинається процес відокремлення унікальних ознак біометричного ідентифікатора.

РОЗДІЛ 3
ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Організаційна структура програмного забезпечення

Будь-яка інформаційна система повинна мати мінімум три основні функціональні частини – модулі зберігання даних, їх обробки і інтерфейсу з користувачем. Кожна з яких може бути реалізована незалежно від двох інших.

Виходячи із вимог до розроблюваної автоматизованої системи найбільш доцільним є використання клієнт-серверної архітектури, що дозволить три основні функціональні частини розподілити по двох фізичних модулях.

Програмне забезпечення, що відповідає за зберігання даних, розташовується на сервері, інтерфейс з користувачем – на стороні клієнта, а обробку даних доводиться розподіляти між клієнтською і серверною частинами.

3.2 Вибір інструментарію для створення ППЗ

Прикладна програма – це будь-яка конкретна програма, що сприяє рішенню якої-небудь задачі в межах даної проблемної сфери.

Наприклад, там, де на комп'ютер покладено завдання контролю за фінансовою діяльністю якої-небудь фірми, прикладною буде програма підготовки платіжних відомостей. Прикладні програми можуть носити і загальний характер, наприклад, забезпечувати складання і друкування документів тощо.

Прикладні програми можуть використовуватися або автономно, тобто вирішувати поставлену задачу без допомоги інших програм, або в складі програмних комплексів або пакетів.

Для даної роботи необхідно було створити прикладну програму, яка б вирішувала поставлені завдання. Для написання коду програми – було обрано мову програмування С++. Перш за все, ця мова являється потужним інструментом, підтримуючим різноманітні підходи до розробки програм. С++ є однією з найпоширеніших мов програмування, що значно спрощує розробку і супровід програмного забезпечення. Вона дозволяє писати різноманітні програмні продукти для платформи Windows, на яку і орієнтовано дане програмне забезпечення. Не кажучи вже про те, що програмне забезпечення, розроблене на С++, має високу продуктивність [9].

Враховуючи загальні вимоги до сучасного програмного забезпечення, яке використовується і створюється у наш час. Вимоги до програмного забезпечення, яке потрібно створити для візуалізації даних по обліку, а також, обрану мову програмування – одним з найкращих варіантів був вибір візуального середовища розробки Borland C++ Builder 6.

Обране середовище: Borland C++ Builder 6 – дозволяє програмувати на мові С++ та дає змогу створювати програми з графічними інтерфейсами, використовуючи бібліотеку VCL. Це потужний та надійний інструмент, який не займає багато місця на жорсткому диску, а по своїм можливостям майже не поступається більш сучасним програмним продуктам. Крім того, він характеризується зручністю, надзвичайною легкістю в процесі розробки користувацького інтерфейсу, а в мережі Інтернет до нього можна знайти достатньо довідкової інформації.

3.3. Алгоритмізація та програмування програмних модулів

Алгоритмічне забезпечення, що застосовується до конкретного об’єкта, дозволяє визначити необхідні структуру і склад обчислювально-керуючого комплексу, інформаційного забезпечення, виробити вимоги до швидкодії, об’ємів пам’яті, надійності проектованої системи.

Алгоритмічне забезпечення даної інформаційної системи обліку полягає в моделюванні поведінки користувача та розробці алгоритмів до основних програмних функцій, в тому числі наглядного представлення розрахунків та формул у вигляді блок-схем. Для зображення процесу алгоритмізації спроектовано прості блок-схеми в редакторі Microsoft Visio. З їх врахуванням проектується інтерфейс користувача та здійснюється програмування окремих модулів самої системи. Результатом створення правильних алгоритмів буде вдала, продумана, безпомилкова робота системи.

РОЗДІЛ 4

РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ

4.1 Тестування системи

Тестування програмного забезпечення – це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки.

Може оцінюватись:

- відповідність вимогам, якими керувалися проектувальники та розробники;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з програмним забезпеченням та операційними системами;
- відповідність задачам замовника.

Оскільки число можливих тестів навіть для нескладних програмних компонент практично нескінченне, тому стратегія тестування полягає в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів. Як результат програмне забезпечення тестується стандартним виконанням програми з метою виявлення багів (помилок або інших дефектів). Тестування ПЗ може надавати об’єктивну, незалежну інформацію про якість ПЗ, ризики відмови, як для користувачів так і для замовників. Перевіряється, як виконуються функції програми, як приймаються вихідні дані, які виходять результати та як зберігається цілісність зовнішньої інформації.

4.2 Вимоги до апаратного та програмного забезпечення

Апаратне забезпечення (англ. hardware) – електронні і механічні частини обчислювального пристрою, що входять до складу системи чи мережі (програмне забезпечення і дані, які обробляє система, не є апаратним забезпеченням). До апаратного забезпечення належать: електронні схеми (арифметичні, логічні, цифрові і аналогові), реалізовані у вигляді різних електронних пристроїв і приладів, пристрої вводу-виводу, схеми і компоненти живлення (батареї, перетворювачі напруг і струмів), діагностична і тестувальна апаратура, пасивні компоненти (шасі, корпуси, стійки, комп'ютерні роз'єми і інше).

Вимоги до програмного забезпечення – набір вимог щодо властивостей, якості та функцій програмного забезпечення, що буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки вимог.

Розробка вимог до програмної системи може бути розділена на декілька етапів: знаходження вимог (збір, визначення потреб зацікавлених осіб та систем); аналіз вимог (перевірка цілісності та закінченості); специфікація (документування вимог); тестування вимог.

4.3 Склад інсталяційного пакету

Інсталяція (встановлення) – процес встановлення програмного забезпечення на комп'ютер кінцевого користувача. Виконується особливою програмою (пакетним менеджером), присутнім в операційній системі (наприклад, RPM і APT в Linux, Windows Installer в Microsoft Windows), або ж тим, що вже входить до складу самого програмного забезпечення засобом встановлення. В операційній системі GNU дуже поширене використання

системи GNU toolchain і її аналогів для компіляції програмного забезпечення безпосередньо перед встановленням.

Більшість програм постачаються для продажу та поширення в стисненному (упакованому) вигляді. Для нормальної роботи вони повинні бути розпаковані, а необхідні дані правильно розміщені на комп'ютері, враховуючи відмінності між комп'ютерами і налаштуваннями користувача. У процесі установки виконуються різні тести на відповідність заданим вимогам, а комп'ютер необхідним чином конфігурується (настроюється) для зберігання файлів і даних, необхідних для правильної роботи програми.

Установка, як правило, включає в себе розміщення всіх необхідних програмі файлів у відповідних місцях файлової системи, а також зміну та створення конфігураційних файлів. Пакетні менеджери також виконують при установці контроль залежностей, перевіряючи, чи є в системі необхідні для роботи даної програми пакети, а в разі успішної установки реєструючи новий пакет у списку доступних. Оскільки даний процес є різним для кожної програми і комп'ютера, то багато програм (включаючи операційні системи) поставляються разом з універсальним або спеціальним інсталятором — програмою, яка автоматизує більшу частину роботи, необхідної для їх установки.

Деякі комп'ютерні програми написані таким чином, що встановлюються простим копіюванням своїх файлів в потрібне місце, а самого процесу установки як такого немає. Про такі програми кажуть, що вони не вимагають установки. Це поширене серед програм для Mac OS X, DOS і Microsoft Windows. Існують операційні системи, які не вимагають установки, і, таким чином, можуть бути безпосередньо запущені з завантажувального CD, DVD або USB, не впливаючи на інші ОС, встановлені на комп'ютері користувача. Прикладом такої ОС є Knoppix або Mac OS 1-9. Цей термін також поширюється на плагіни, драйвери і програмні файли, які самі по собі не є програмами.

ВИСНОВКИ

В даній дипломній роботі було проведено аналіз концепції Internet of Things та властивостей з точки зору захисту персональних даних користувача . Було зроблено ретельний огляд функцій систем за подібною тематикою та визначено функцій та побудову системи.

За допомогою аналізу для системи розпізнавання об’єктів на основі графічної інформації було створено систему розпізнавання та верифікації овалу обличчя користувача на базі OpenCV.

Було розроблено програмне забезпечення, що дозволяє зберігати та оброблювати графічну інформацію, що містить біометричні дані користувача, а саме: знімок овалу обличчя та відбитку пальців користувача у реальному часі.

Під час розробки даного продукту я більш поглиблено вивчив як мову програмування Python, ознайомився з можливостями каскадного способу розпізнавання обличчя на базі OpenCV, які можна використувати за будь яким призначенням. Навчився створювати інтерфейс користувача з використанням бібліотеки Tkinter.

Планується продовження розробки. Представлений програмний продукт матиме можливість створювати базу даних користувачів, надавати та керувати різнорівневими правами доступу до системи, інтегрувати функціонал програмного забезпечення у підсистему мікропроцесора Arduino для подальшої імплементації у вигляді фізичного прототипу електронного замка.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вигерс К.И. Розробка вимог до ПЗ. – М.: Російська редакція Microsoft, 2004. 575 с.

2. Леонов И.В. Введення в методологію розробки програмного забезпечення за допомогою Rational Rose // Екеейп, 2004. 304 с.

3. Zave P., Jackson M. Four Dark Corners of Requirements Engineering // ACM Transactions on Software Engineering, January 1997. № 1.

4. Pinheiro Francisco A. C., Goguen Joseph A. An Object-Oriented tool for Tracing Requirements // Software. Mach 1996. №3.

5. Guckkenheimer S., Peter J. Software Engineering With Microsoft Visual Studio. Team System. – Adison Wesley, 2006. 273 p.

6. Історія відбитків пальців: http://onin.com/fp/fphistory.html

7. Ніцан Лебович, «Біометрія або сила радикального центру в критичному дослідженні» 41: 4 (літо, 2015), 841-868

8. Ніцан Лебович, «Біометрія або сила радикального центру в критичному дослідженні»41: 4 (літо, 2015), с. 853

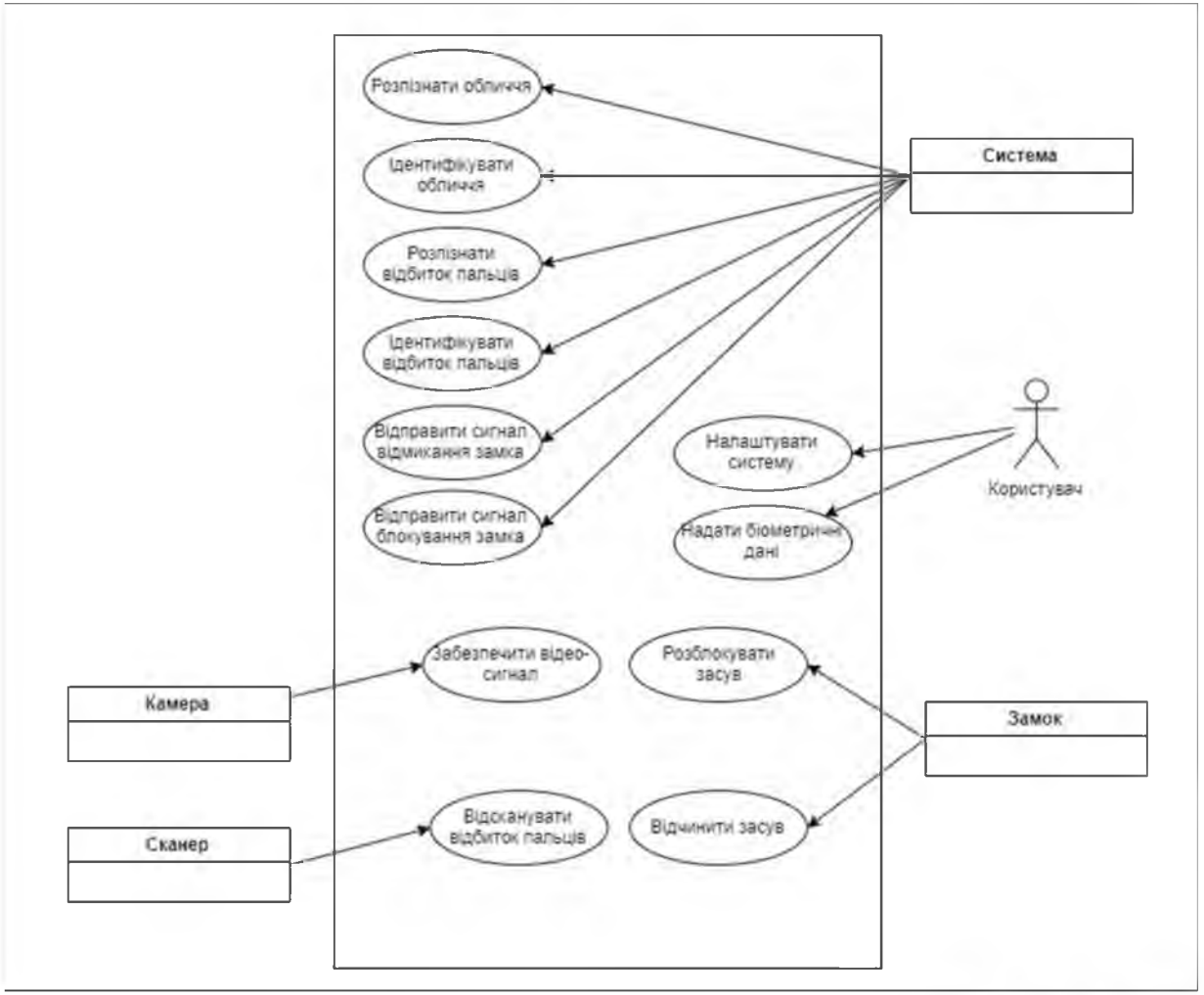
9. Девід Ліон, «Товариство спостереження: Моніторинг повсякденного життя»(Philadelphia, 2001)

10. Келлі А. Гейтс, Наше Біометричне Майбутнє: Технологія розпізнавання обличчя та культура спостереження (Нью-Йорк, 2011 р.), Стор. & Nbsp; 100

11. Царьов Р.Ю. Біометричні технології: навч. посіб. / Р.Ю. Царьов, Т. М. Лемеха. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 140 с

ДОДАТОК Б

ГРАФІЧНА ЧАСТИНА

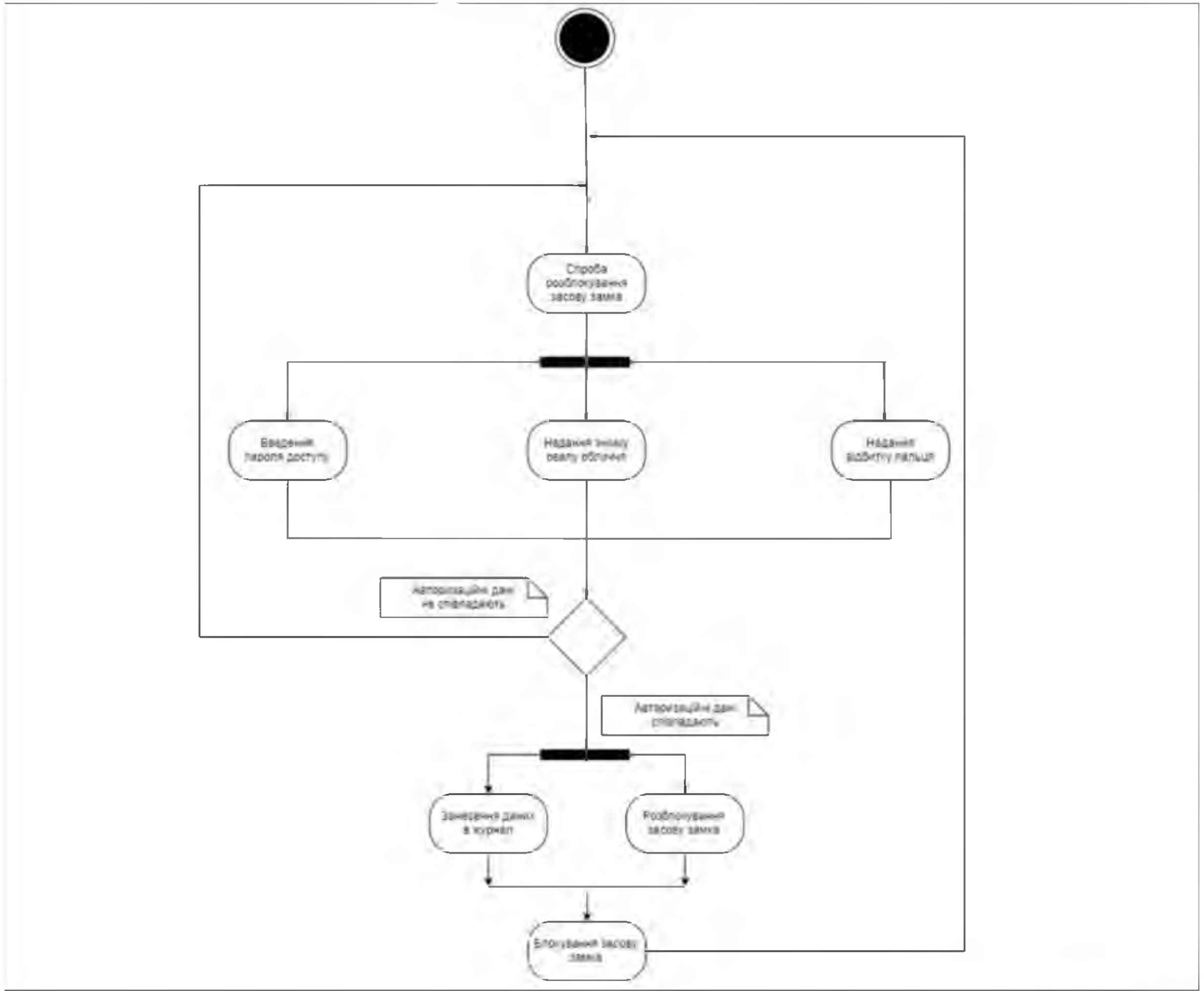


15.03 ДП.2399 "С" 19 12 06.016 ПЗ

Діаграма прецедентів

НУБіП України ПІ-16006 б

Рис. Б.1. – Діаграма прецедентів.

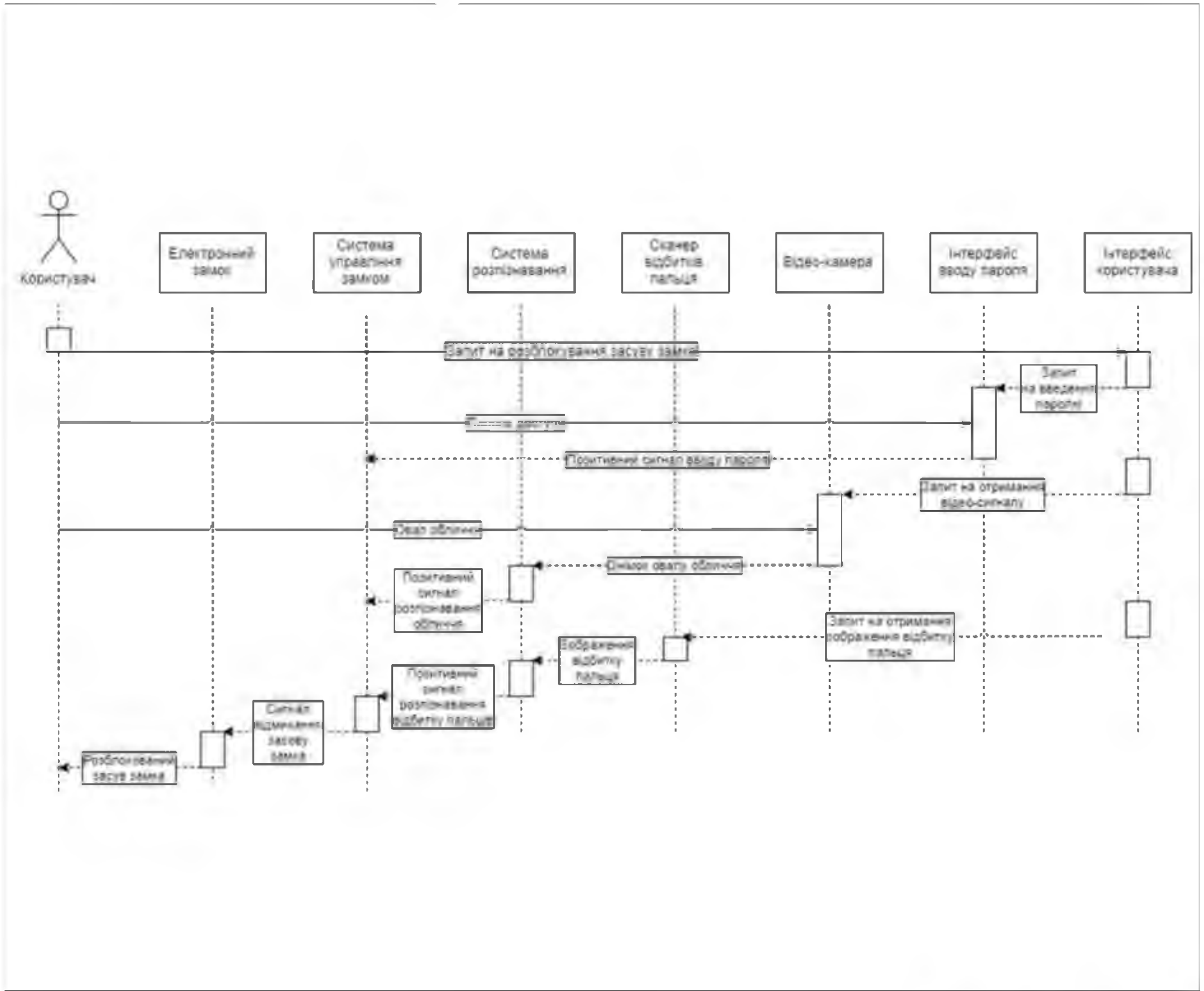


15.03 ДП.2399 "С" 19 12 06.016 ПЗ

Діаграма активності

НУБіП України
ПІ-16006 б

Рис. Б.2. – Діаграма активності.

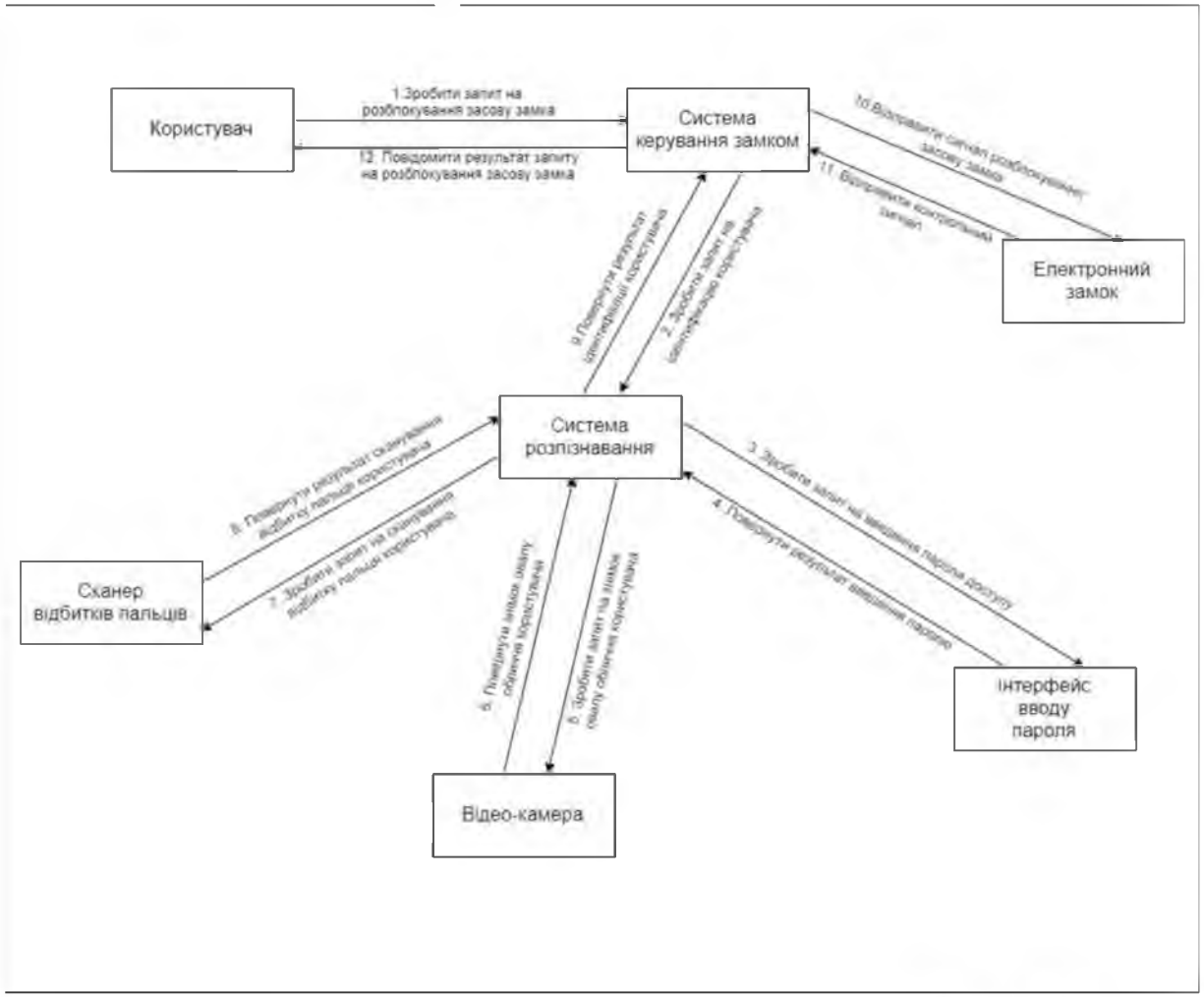


15.03 ДП.2399 "С" 19 12 06.016 ПЗ

Діаграма послідовності

НУБІП України
ПІ-18006 б

Рис. Б.3. – Діаграма послідовності.



						15.03 ДП.2399 "С" 19 12 06.016 ПЗ
Змн.	Арк.	№ докум.	Підпис	Дата		
Розроб.					Діаграма взаємодії	Літ. Арк. Аркушів
Керівник						НУБіП України ПІ-16006 б
Н. контр.						
Зав.каф.						

Рис. Б.4. – Діаграма взаємодії