Issue 6

https://doi.org/10.23939/sisn2019.02.041

УДК 004.056:681.5.042

ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА “РОЗУМНИЙ ЗАМОК” ДЛЯ ЗАХИСТУ ПРИМІЩЕНЬ

Л. Я. Рибак, П. О. Кравець

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж,
rybaklevko@ukr.net, ORCID: 0000-0002-5915-5520
Petro.O.Kravets@lpnu.ua, ORCID: 0000-0001-8569-423X

Розглянуто інтелектуальну інформаційну систему “Розумний замок” для захисту приміщень. Здійснено загальний опис розумного дому та актуальності їх використання. Сформульовано проблему, яка виникає під час використання розумних пристроїв для дому. Наведено приклади двох відомих таких моделей: SmartLock та Schlage Sense Smart Lock. Проаналізовано їхні переваги та недоліки для врахування під час розроблення інтелектуального замка. Подано короткий опис актуальної інформації щодо методики розпізнавання обличчя. На основі цих даних сформульовано цілі, які досягнуто у цій статті. Наведено діаграму функціонування підсистем інтелектуальної інформаційної системи. Виконано детальний опис кожної зі складових інформаційної системи, а саме: серверної частини, програми для розпізнавання обличчя, бази даних та фізичного пристрою “розумний замок”. Здійснено поетапний опис процесу створення кожного із цих модулів системи. Обґрунтовано вибір мови програмування, вебзасобів, пристроїв для запуску, операційної системи, системи управління базами даних та програмних бібліотек для розпізнавання обличчя. Наведено детальний опис пристрою, на якому працюватиме програма, та пристрою, який використано для налагодження програмних елементів системи. Пояснено підхід до вибору типу інтеграції інтелектуальної інформаційної системи, описано принцип роботи інформаційної системи, продемонстровано концептуальну модель системи. Наведено приклад використання цієї інформаційної системи в реальних умовах. Надано інструкції щодо типового використання інформаційної системи користувачем. Проаналізовано результати та сформовано висновки щодо актуальності створення інформаційної системи та її практичного застосування.

Ключові слова: інтелектуальна інформаційна система, “розумний замок”, розпізнавання обличчя, система керування базами даних, безпека інтеграції, горизонтальна інтеграція, концептуальна модель, функціональна модель.

Вступ

Тема розроблення розумних пристроїв для дому є дуже актуальною в сучасному світі, оскільки важливість їх застосування зростає у зв’язку зі збільшенням кількості самих розумних домів та спробами максимально автоматизувати вирішення побутових проблем.

Розумний дім – це резиденція, яка використовує підключені до Інтернету пристрої для віддаленого моніторингу та управління приладами і системами, наприклад, такими як освітлення та опалення.

Технологія розумного дому, яку також часто називають домашньою автоматизацією, забезпечує його власникам безпеку, комфорт, зручність та енергоефективність, даючи їм змогу

керувати розумними пристроями за допомогою додатків на своїх смартфонах або інших мережевих пристроях. Домашня автоматизація створює можливості для контролю предметів навколо дому простим натисканням кнопки чи голосовою командою. Частина системи Інтернету речей (IoT, Internet of Things) та пристрої розумного дому часто працюють разом, обмінюючись даними між собою та автоматизуючи дії на основі особливостей їхніх власників [1, 2].

Існує багато категорій “розумних” товарів для дому, й одним із них є “розумний замок”. Створення системи “розумних замків” доцільне і для України, бо з кожним днем кількість “розумних” домів зростає і, відповідно, збільшується потреба у їх захисті. Окрім цього, цей пристрій можна використовувати і як самостійну одиницю для звичайних будинків. Розроблена система створена передусім для вирішення проблеми із домашніми крадіжками.

Постановка проблеми

Під час роботи із “розумними замками” виникають певні проблеми, а саме: наявність доступу до відео та аудіо через дзвінок у двері полегшує хакеру шпигування за власниками дому та будь-якими іншими співмешканцями. Для прикладу, зловмисник може підставити власне зображення як відоме для системи або ж, навпаки, підставити замість власного зображення те, що належить реально зареєстрованим користувачам, щоб дистанційно відчинити двері. Іншою проблемою розроблених “розумних замків” є висока вартість, складність у використанні та прив’язаність до великої комплексної системи.

У цій статті проаналізовано проблеми вже відомих систем, детально описано та частково продемонстровано роботу інформаційної системи “Розумний замок” для захисту приміщень, яка створена на противагу системам, які вже існують.

Аналіз останніх досліджень та публікацій

Спершу розглянемо такий приклад розумного замка, як “Smart lock”, розроблений компанією Indeema Software.

Звичайний електромагнітний дверний замок використовується як стандартне рішення у багатьох офісних будівлях. Зазвичай він замкнений, і щоб відкрити його, треба скористатись або RFID-картою ззовні (саму ключову карту, яку розробники часто залишають вдома), або кнопку з внутрішньої сторони. До кнопки підключається просте реле, щоб замок відкрився на деякий час. Реле повинно управлятися мікроконтролером (або мікрокомп’ютером), який надсилає відповідні сигнали. Raspberry Pi вибрано як мікроконтролер, що повинен контролювати реле після перевірки запитів клієнтської програми. Основна суть у такому робочому алгоритмі – програма, що працює як з операційними системами iOS, так і з Android. Отже, весь проєкт смарт-блокування передбачає складне апаратно-програмне рішення IoT, яке об’єднує різні технології, такі як веббекенд, клієнтську частину для iOS та Android, вбудовану частину в Linux, а також різні спеціальні фрейми бібліотеки Raspberry Pi.

Щоб не перевантажувати додаток зайвими функціями, для його користувальницького інтерфейсу розроблено лише три екрани: головний екран, екран входу та налаштування. Додаток – це серверно-клієнтське рішення, яке використовує сторонні аутентифікації за допомогою облікових даних компанії Redmine. Після входу в систему окремий ключ API користувача Redmine можна створити для надсилання на сервер. Сервер перевіряє конкретний ключ API, з’ясовуючи, чи може він отримати доступ до компанії Redmine, тобто можна відкрити дверний замок чи ні. Наприклад, клієнтська програма для iOS – проста програма, написана на Swift, у якій задіяно 3D Touch для полегшення доступу. Функція, яка запобігає випадковому відкриванню дверей, коли користувач далеко від офісу, надає додаткове вікно, яке спливає і в якому користувач може підтвердити намір відкрити двері. Версія програми Android не відрізняється від версії iOS щодо загальної

функціональності. Отже, остаточне рішення IoT надає користувачам можливість входити в офіс лише за допомогою кількох дотиків на своїх смартфонах.

Python вибрано мовою програмування серверної частини програми, що мотивовано кількома міркуваннями, пов'язаними як із доволі тісним 24-годинним періодом Hackathon, так і з ARM-процесором вбудованої цільової платформи. Останній трохи поступається процесорам x86 щодо компіляції та розгортання. Ось чому Python як інтерпретовану мову програмування високого рівня, що дає змогу редагувати код як на цільовій платформі, так і на хост-сервері, було застосовано для негайного запуску програми для тестування. Крім того, додаток Python працює як засіб Linux, коли операційна система надає різні можливості, такі як запуск програми, перезапуск у разі збоїв програми, виконання програми у вигляді фонового процесу тощо. Отже, переваги Python передбачають розгортання додатка до цільової платформи без попередньої компіляції вихідного коду в машинні інструкції [3].

Створюваний додаток не передбачає аутентифікації за допомогою розпізнавання обличчя, а лише за допомогою платформи Redmine і внутрішнього токена цього сервісу забезпечує дозвіл чи заборону проходу користувача. Оскільки додаток встановлюється на мобільні пристрої, то це створює проблему із захистом у випадку втрати цього пристрою. Іншою загрозою є також можливі проблеми із самим сервісом, а це призведе до того, що особа в принципі не зможе увійти у будівлю або її потрапляння туди дуже ускладниться. Перевагами цієї системи є простота у роботі, швидкість реалізації та повторне використання засобу захисту системи.

Іншим розглянутим "розумним замком" є Schlage Sense Smart Lock для Android. У Smart Lock від Schlage вдалося досягти правильного поєднання простоти, прогресивної технології та якості. Він не настільки відверто оцифрований, як інші варіанти, але має всі функції, що дають можливість вважати його "розумним замком". Однією з особливостей є наявність сенсорної панелі для доступу до дверей.

Подальше розширення можливостей Schlage забезпечити комфорт для своїх користувачів полягає у тому, що доступ до блокування можна отримати за допомогою резервного ключа. Це може вирішити проблему для користувачів, які забувають паролі та коди для розблокування дверей.

Ще одна відмітна особливість цього товару – різні гарантії, які його супроводжують. Schlage постачається із трирічною електронною гарантією та довічною механічною гарантією. Це значно знижує загальну вартість товару в довгостроковій перспективі [4].

І навіть більше, цей "розумний замок", що працює на батареях, вміє попереджувати про низький рівень заряду акумулятора, хоч сам акумулятор забезпечує довготривалу роботу. Вартість замка висока, однак надійність, що забезпечує довговічність, принесе й економічну вигоду з часом.

Переваги цього замка такі:

- Вбудована у смарт-замок сигналізація про втручання гарантує, що жоден зловмисник не зможе зламати систему.
- Його легко встановити.
- Керувати замком можна за допомогою голосового інтерфейсу Siri.
- Блокування однаково сумісне із пристроями Android та iOS.

Недоліками є:

- Для керування замком за допомогою пульта потрібен телевізор Apple.
- У продукту невеликий діапазон Bluetooth.

Завдяки штучному інтелекту та блокчейну розпізнавання облич, безумовно, є важливим цифровим викликом для всіх компаній, організацій та урядових установ.

Розпізнавання обличчя – це процес виявлення або перевірки обличчя особи. Програма фіксує, аналізує та порівнює візерунки на основі деталей обличчя людини. Процес розпізнавання обличчя є

важливим кроком, оскільки виявляє та знаходить обличчя людини на зображеннях та відео. Процес захоплення обличчя перетворює аналогову інформацію (зображення обличчя) на набір цифрової інформації (даних) на підставі особливостей обличчя людини. Процес встановлення відповідності обличчя перевіряє, чи належать два зображення обличчя одній людині.

Сьогодні це вважається найприроднішим із усіх біометричних вимірювань, оскільки люди впізнають одне одного за обличчям, а не за відбитками пальців чи іншими параметрами. У випадку використання штучного інтелекту ця технологія уможливлює віддалене застосування (людині не треба торкатися безпосередньо до камери аби зчитувати контури обличчя), фіксування багатьох людей одночасно у публічних місця та вищу безпеку системи.

Біометричні дані використовують для ідентифікації та автентифікації людини за допомогою набору впізнаваних та перевірених даних, унікальних та специфічних для цієї особи.

Ідентифікація відповідає на запитання: “Хто ти такий?”, аутентифікація – на запитання: “Ви справді є тим, за кого себе видаєте?”.

Що стосується біометрики обличчя, то 2D або 3D-датчик спочатку “захоплює” обличчя. Потім він оцифровує дані, застосовуючи алгоритм, перш ніж порівняти зняте зображення із зображеннями, що зберігаються в базі даних.

Ці автоматизовані системи можуть бути використані для виявлення або перевірки особи лише за кілька секунд, на підставі особливостей обличчя: відстані між очима, форми носа, контуру губ, вух, підборіддя тощо [5].

Формулювання цілі статті

Мета статті – описання інтелектуальної інформаційної системи “Розумний замок” для захисту приміщень, розгляд її поетапного створення та інтеграції (об’єднання різних підсистем у єдину систему).

У результаті буде обґрунтовано теоретичні положення та продемонстровано практичне застосування для удосконалення роботи із захистом “розумного” дому.

Виклад основного матеріалу

Для реалізації такої комплексної системи необхідно розділити її на дрібніші підсистеми (рис. 1).

Рис. 1. Функціонування підсистем інтелектуальної інформаційної системи

Спочатку потрібно розробити програму інтелектуального замка, яка займатиметься розпізнаванням облич. Для цього використано вже відомий програмний продукт face_recognition (https://github.com/ageitgey/face_recognition). Ця бібліотека написана мовою програмування Python і надає можливості для виявлення облич на картинці, пошуку та здійснення різних маніпуляцій із зображеннями облич і просто знаходження певних особливостей на їх зображенні. Точність розпізнавань дуже висока і становить 99,38 %, за оцінкою Label Laces in Wild [5]. Бібліотека також може використовуватися в комбінації з OpenCV для графічного позначення облич на зображеннях чи відео, зокрема в реальному часі.

OpenCV [OpenCV] – це бібліотека комп'ютерного зору із відкритим кодом. Бібліотека написана на C і C ++ і працює під Linux, Windows та Mac OS X. Активна розробка на інтерфейсах для Python, Ruby, Matlab та інших мов. Однією з цілей OpenCV є створення простої у користуванні інфраструктури комп'ютерного зору, що допомагає людям швидко створювати доволі складні програми для зору. Бібліотека OpenCV містить понад 500 функцій, що охоплюють багато областей: комп'ютерний зір, медичні знімки, безпеку, інтерфейс користувача, калібрування камери, стереобачення та робототехніку. OpenCV також містить повну бібліотеку машинного навчання загального призначення (MLL). Суббібліотека орієнтована на статистичне розпізнавання шаблонів та кластеризацію. MLL дуже корисний для завдань, пов'язаних із зором, покладених в основу OpenCV, але він є загальним для використання у будь-якій проблемі машинного навчання [6].

Розроблена програма буде використовувати можливості бібліотеки OpenCV для фіксації та відображення обличчя за прямої трансляції відео, яка може здійснюватись з адміністративної панелі управління (рис. 2).

Ще одна із функцій, які будуть використані з бібліотеки для розпізнавання облич, – виявлення наявності тієї чи іншої людини на зображенні. Для цього на самому пристрої передбачена відеофіксація зображення, яке надалі порівнюватиметься із зображеннями із бази даних. У випадку розпізнавання особи буде повертатися успішний результат, наслідком чого є відкривання дверей.

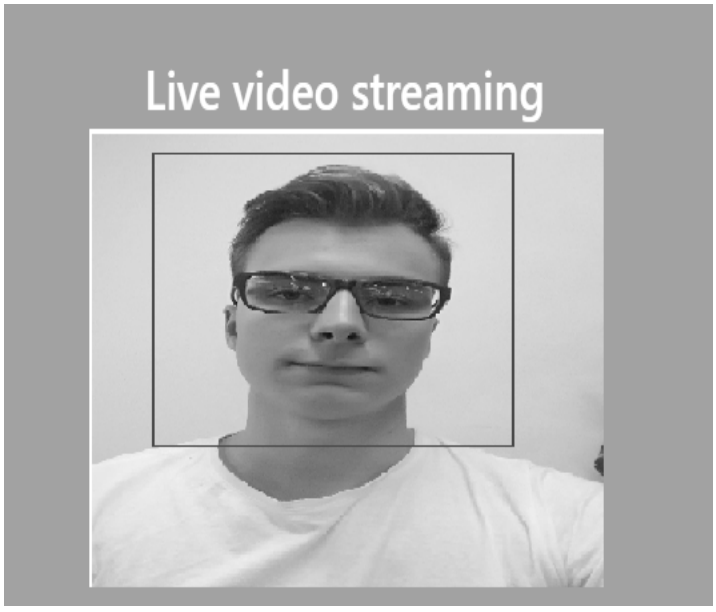

Рис. 2. Використання фіксації обличчя у ході прямої трансляції

Після цього необхідно розробити серверну частину, яка дасть змогу конфігувати пристрій та вносити оновлення у базу даних. Для реалізації серверної частина використано вебзасіб Flask. За

замовчуванням Flask не містить шар абстракції бази даних, перевірку форми чи інше, де вже існують призначені для цього бібліотеки. Натомість Flask підтримує розширення для додавання такої функціональності у програму, як вона була б реалізована у самій Flask. Численні розширення забезпечують інтеграцію баз даних, перевірку форм, оброблення завантажень, різні технології відкритої аутентифікації тощо.

На Flask є багато значень конфігурації із розумними типовими налаштуваннями та декількома умовами на початку роботи. За цими умовами шаблони та статичні файли зберігаються у підкаталогах у вихідному дереві програми Python із шаблонами імен та статичними даними відповідно. Хоча їх можна змінити, зазвичай цього не потрібно, особливо під час роботи [7].

У випадку створюваної серверної частини Flask надаватиме набір API для надсилання запитів до сервера. Відповідно за його допомогою дані пересилатимуться зі встановленого “розумного замка” на бекенд частину. Для прикладу, надсилатиметься зображення користувача, яке аналізуватиметься на серверній частині, а результат перевірки на предмет наявності такої людини в базі даних буде відправлено клієнту. Також за його допомогою будуть генеруватися вебсторінки для адміністративної сторінки.

Ще одним елементом системи буде база даних, яка використовуватиметься для зберігання зображень, інформації про клієнтів та пристрої. Її структура достатньо примітивна і містить лише три таблиці: Devices, Users і Images. Як реляційна система управління базами даних буде використовуватися SQLite. Це внутрішня бібліотека, яка реалізує автономну, нульову конфігурацію, безсерверний, транзакційний двигун бази даних SQL. Вихідний код для SQLite існує у відкритому доступі, безкоштовний як для приватних, так і для комерційних цілей. SQLite має прив’язки до декількох мов програмування, таких як C, C ++, C#, Python, Java, Delphi та інших. Система SQLite відповідає атомарності, консистенції, ізоляції, міцності (Atomicity, Consistency, Isolation, Durability). Ця вбудована система управління реляційними базами даних міститься у невеликій бібліотеці програмування C і є невід’ємною частиною клієнтських додатків. Бібліотека SQLite використовує динамічний синтаксис SQL і забезпечує багатозадачність для одночасного читання і запису, які здійснюються безпосередньо у звичайні файли диска. Бібліотека SQLite динамічна, а прикладні програми використовують функціональність SQLite через прості виклики функцій, зменшуючи затримку в доступі до бази даних. Ці програми зберігають цілі бази даних як єдині файли на хост-машинах для міжплатформних застосувань [8]. Для створюваної системи актуальним є використання простої за будовою системи керування, оскільки сама база даних буде нескладною за структурою, доступ до неї здійснюватиметься з одного потоку і від одного користувача. Подібна одиниця бази даних буде використана для кожного “розумного замка”.

Програмні засоби встановлено на одноплатний комп’ютер Raspberry Pi 3 Model B+. Це поліпшена версія Raspberry Pi 3 Model B, основана на системі-на-чипі (System-on-a-chip) BCM2837B0, яка містить 1,4 ГГц чотириядерний ARMv8 64-бітний процесор і потужний відеопроцесор VideoCore IV. Комп’ютер Raspberry Pi може працювати із повним спектром дистрибутивів ARM GNU/Linux, зокрема із Ubuntu Snappy Core, Debian, Fedora і Arch Linux, а також Microsoft Windows 10 IoT Core.

Raspberry Pi 3 Model B є першим Raspberry Pi, який має бездротовий Bluetooth-зв’язок. Операційною системою слугуватиме Ubuntu Mate 18.04, що забезпечує повне, звичне для робочого стола середовище, яке можна використовувати для базових обчислень настільних ПК. Вона прекрасно підходить для створення IoT-пристроїв на основі ARMv7 або ARMv8 [9]. У цій моделі є вхід, до якого буде під’єднано камеру CSI Camera Port, та інтерфейс 40 GPIO Pin, через який надсилатиметься сигнал на електронний замок. Також важливою була підтримка з боку операційної системи потрібної версії мови програмування Python 3.6 для використання бібліотеки

face_recognition та вебзасобу Flask останньої версії, оскільки серверна частина та програма для розпізнавання обличчя будуть розгорнуті безпосередньо на цій платі.

У майбутньому передбачається також підтримка для інших моделей: Raspberry Pi 3 Model B, Raspberry Pi Model A+, Raspberry Pi 4 Model B.

Окрім цього, як пристрій для написання та налагоджування програми використано звичайний ПК із встановленою на ньому операційною системою Ubuntu 18.04 (або Ubuntu 16.04), щоб уникнути розбіжностей із операційною системою плати. Використано сумісні версії Python та застосованих бібліотек. Для введення відеоданих використано звичайну вебкамеру, інтегровану або підключену до комп'ютера. Для імітації GPIO (General-purpose input/output) використано бібліотеку virtual-GPIO.

Мовою програмування вибрано Python, що пояснюється кількома міркуваннями, пов'язаними як із використанням бібліотек, написаних на цій мові, так і з ARM-процесором вбудованої цільової платформи. Python взаємодіє із модулями C і, відповідно, із системними викликами операційної системи Linux. Також його необхідна версія встановлена як програма за замовчуванням на Ubuntu Mate 18.04, що забезпечить відсутність проблем зі збиранням, компіляцією та лінкуванням, які виникають у разі встановлення сторонніх продуктів для операційної системи. Python працює як послуга Linux, коли операційна система надає різні засоби, такі як виконання програми під час запуску комп'ютера, перезапуск у разі збоїв програми, виконання програми як фонового процесу тощо.

Наступним етапом розроблення інформаційної системи є її інтеграція. Системна інтеграція – це процес з'єднання різних підсистем (компонентів) в одну більшу систему. Мета використання системної інтеграції полягає у тому, щоб змусити різні елементи системи комунікувати між собою, щоб пришвидшити інформаційні потоки та зменшити операційні витрати.

Безпека інтеграції інформаційної системи забезпечується за рахунок обмеження доступу користувача до певних ресурсів, тобто користувачеві присвоюють певні права, відповідно до яких дозволено або заборонено локальний доступ до інформації, що зберігається на ПК, або віддалений доступ через комунікаційні посилання на інформацію, наявну на іншому ПК. У створюваній інформаційній системі у користувача буде доступ до адміністративної сторінки із використанням імені користувача та пароля, які прив'язані до електронної пошти, й безпосередньо доступ до фізичного пристрою "Розумного замка".

Виділяють чотири основні типи інтеграції: вертикальна, горизонтальна, зіркова та інтеграція загального формату даних. Для створюваної інформаційної системи буде використовуватися горизонтальна інтеграція, тобто така, у якій всі підсистеми відокремлені та взаємодіють між собою через задані інтерфейси, що збільшує розмір трансформації даних під час передавання їх від одного процесу до іншого [10]. Після інтеграції інформаційної системи очікується, що вона забезпечуватиме відокремлені автономні елементи, які матимуть зрозумілі та розширювані інтерфейси для передавання даних між різними процесами.

Весь принцип роботи інтелектуальної інформаційної системи можна подати за допомогою концептуальної моделі, зображеної на рис. 3. Користувач може звертатися або безпосередньо до фізичного пристрою (засобу введення/оброблення даних), або до адміністративної сторінки. Оскільки засоби введення та оброблення даних локалізовані в одному фізичному пристрої, то він лише обробляє фіксування обличчя користувача після натискання кнопки. За допомогою адміністративної сторінки є можливість налаштовувати увесь список пристроїв, користувачів та отримувати інформацію про додаток. Все це можливо за рахунок надсилання запиту на сервер, який обробляє ці дані та відправляє відповідний запит до бази даних. Система управління базою даних, своєю чергою, вибере необхідну інформацію та поверне її.

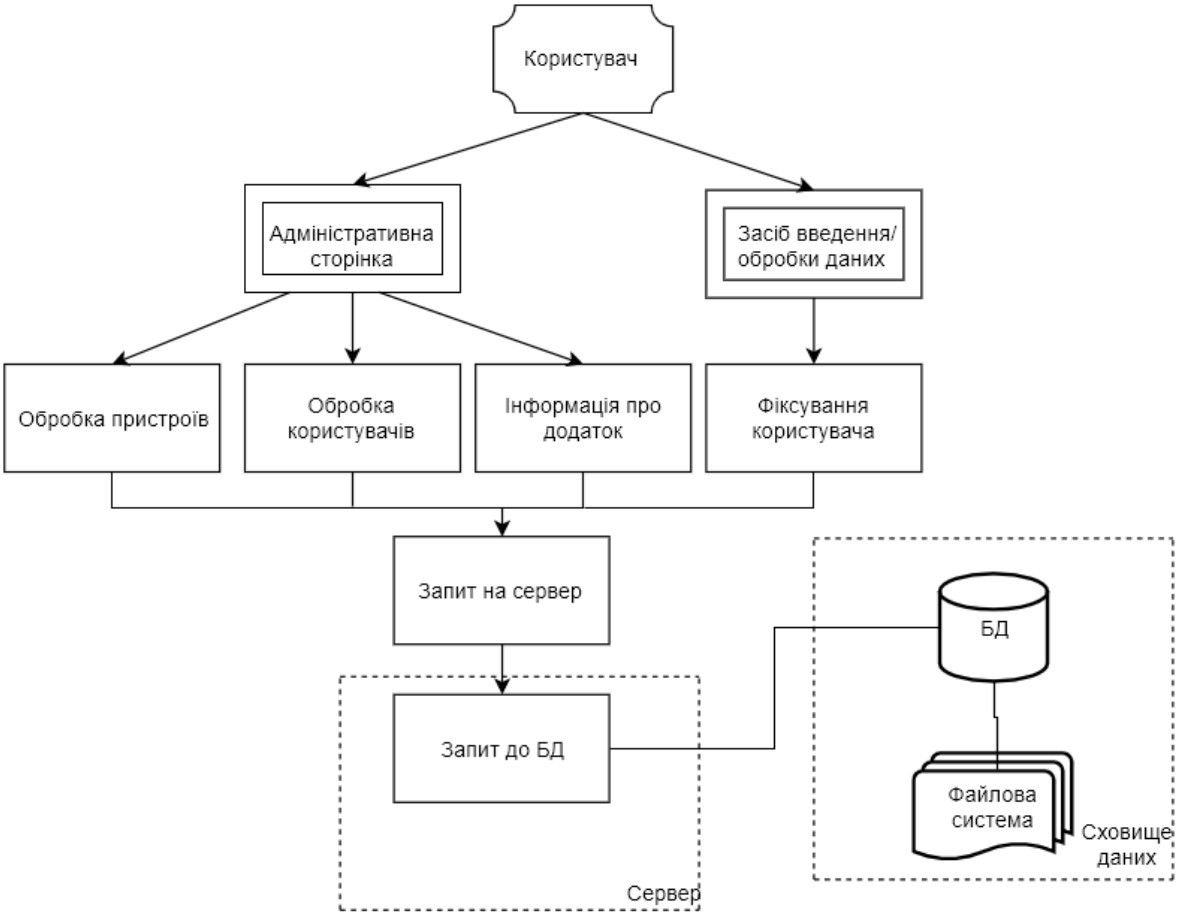

Рис. 3. Концептуальна модель

З практичного погляду це означає, що на одноплатний комп'ютер Raspberry Pi встановлюється серверна програма та програма для розпізнавання обличчя, а також база даних із системою керування SQLite. Відтак до пристрою фізично підмикаються камера та інтерфейс замка, які утворюють електричне коло. Після цього система вважатиметься встановленою та готовою до експлуатації.

Наступний етап – використання встановленої системи. Для цього користувачеві потрібно зареєструватися за допомогою електронної пошти, створити пароль та увійти в обліковий запис. У ньому можна перевірити коректність встановлення системи, перейшовши на вкладку Live Streaming (Пряма трансляція) і переглянувши відео у прямій трансляції. Після цього слід перейти на вкладку Devices (Пристрої) та додати один або декілька елементів інформаційної системи до облікового запису. Якщо реєстрація пристроїв успішна, можна перейти на вкладку Users (Користувачі) та додати користувачів із іменами та зображеннями, які будуть використовуватися для розпізнавання (рис. 4).

Тепер додані особистості розпізнаватиме пристрій і, відповідно, вони матимуть доступ до приміщення. Людей можна додавати або видаляти зі списку користувачів того чи іншого пристрою. Незареєстровані користувачі не зможуть пробратися в дім. Окрім цього, буде вестися цілодобове спостереження за вхідними дверима. Коли користувач наблизиться до дверей і натисне кнопку аутентифікації, його зображення буде зчитане за допомогою камери, надіслане на серверну частину, яка отримає зображення для цього пристрою із бази даних та виконає порівняння, використовуючи програму розпізнавання облич. Якщо буде виявлено збіг, то замок буде відкрито. Для цього логічна одиниця надсилається у порт електронного замка та на короткий час замикається електричне коло, що надасть доступ до приміщення.

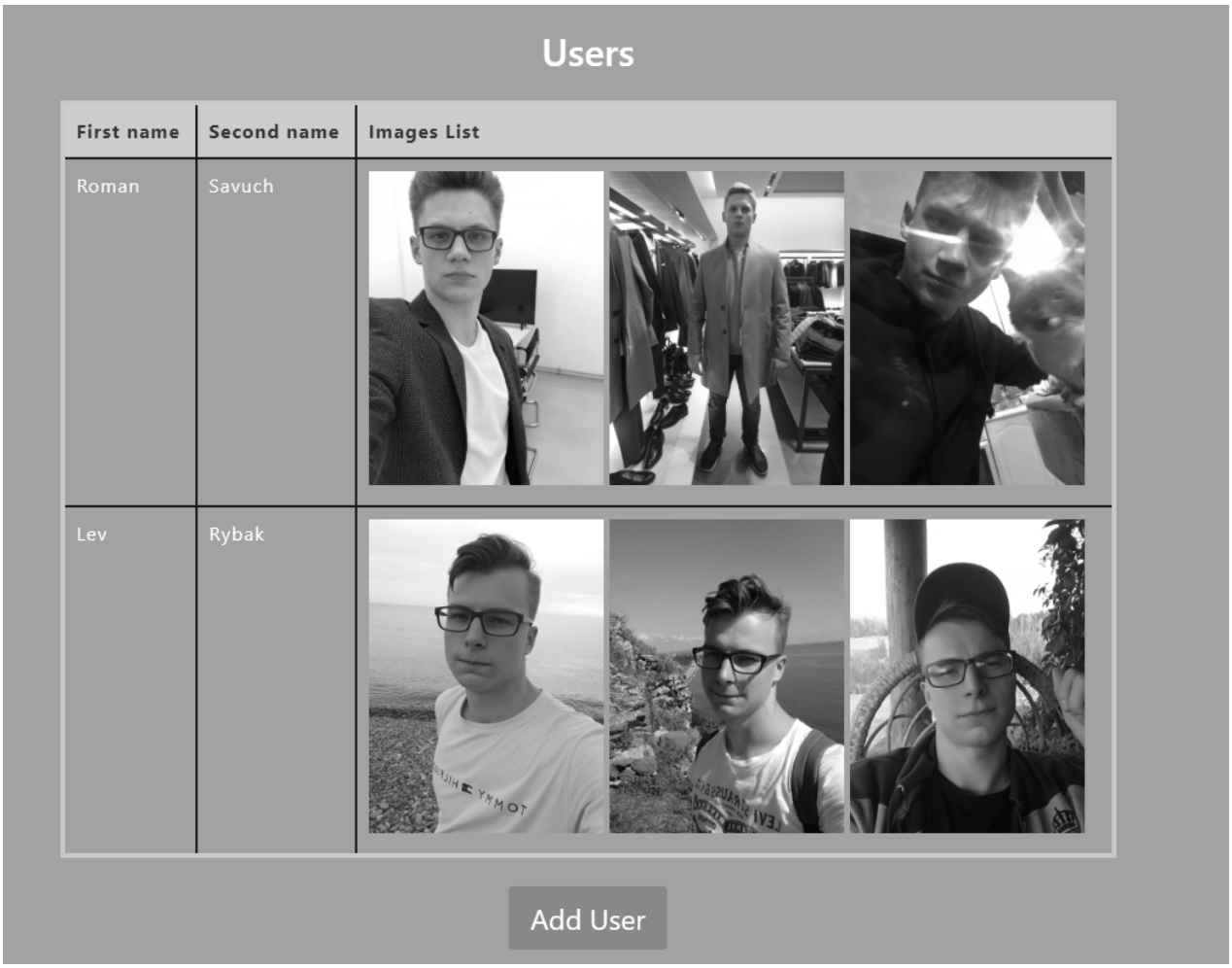

Рис. 4. Сторінка користувачів

Місцем використання цієї інформаційної системи можуть бути приватні будинки, підприємства, офісні будівлі тощо. Окрім цього, таку інформаційну систему можна використати з науковою метою для тестування алгоритмів розпізнавання обличчя у реальних умовах.

Висновки

У статті описано покрокове створення інтелектуальної інформаційної системи "Розумний замок", обґрунтовано використання конкретних засобів розроблення, а саме: мови програмування, вебзасобів, пристроїв, на які встановлено програмні засоби та операційної системи.

Здійснено аналітичний огляд літературних та інших джерел для збирання відомостей про системи-аналоги, найактуальніші методи, підходи, засоби й алгоритми для розв'язання подібних завдань. Проаналізовано вже відомі системи захисту. Виконано порівняльний аналіз переваг та недоліків систем, подібних до створюваної. Результатом став висновок про те, що запланована для розроблення система є актуальною, важливою і затребуваною сьогодні. Вона може забезпечити високий рівень захисту приміщень та простоту у використанні.

Практичне значення одержаних результатів полягає у розробленні захищенішої та досконалішої інформаційної системи для захисту дому. Запровадження цієї технології надає можливість уникнути крадіжок, спростити та зробити ефективнішим захист "розумних домів". Розробка може бути застосована для приватних власників, підприємств та для наукових досліджень.

Отже, визначено етапи розроблення інформаційної системи, проаналізовано систему з погляду практичного застосування, детально описано процес її створення та причини вибору програмних і технічних засобів, описано функціональність системи у вигляді інструкції для користувача, продемонстровано приклад працездатності програми. Беручи все це до уваги, можна дійти висновку, що розроблення інтелектуальної інформаційної системи "Розумний замок" є актуальною науково-практичною проблемою.

Список літератури

1. John R. Patrick (2017). Home Attitude: Everything You Need To Know To Make Your Home Smart. California, US: The Appliance Studio, University Gate East CreateSpace Independent Publishing Platform.
2. Spivey D. (2015). Home Automation For Dummies. Framingham, Massachusetts, US: International Data Group, Inc, For Dummies.
3. Лисик С. (2019). Smart Lock: Why sloth is a driver of the IoT progress. Отриманий 22.02.2019 від https://habr.com/ru/post/441294/.
4. Rezan F. (2019). Best Smart Lock for Doors 2019. Отриманий 01.09.2019 від https://10hitech.com/best-smart-lock/.
5. Geitgey A. (2019). Face Recognition Documentation Release 1.2.3. Отриманий 29.08.2019 від https://buildmedia.readthedocs.org/media/pdf/face-recognition/latest/face-recognition.pdf .
6. Beyeler M. (2017). Machine Learning for OpenCV: Intelligent image processing with Python. USA: Packt Publishing.
7. Greenberg M. (2014). Flask Web Development: Developing Web Applications with Python. USA, CA: O'Reilly Media, Inc.
8. Jay A. Kreibich (2014). Using SQLite: Small. Fast. Reliable. Choose Any Three. USA, CA: O'Reilly Media, Inc.
9. Гололобов В. Н. (2019). Raspberry Pi для любознательных. М: Наука и техника.
10. Jellema L. (2019). Changing views on integration – from Enterprise Service Bus to API Gateway, Serverless and iPaaS. Отриманий 16.09.2019 від https://technology.amis.nl/2019/01/23/changing-views-on-integration-from-enterprise-service-bus-to-api-gateway-serverless-and-ipaas/.

References

1. John R. Patrick (2017). Home Attitude: Everything You Need To Know To Make Your Home Smart. California, US: The Appliance Studio, University Gate East CreateSpace Independent Publishing Platform.
2. Spivey D. (2015). Home Automation For Dummies. Framingham, Massachusetts, US: International Data Group, Inc, For Dummies.
3. Lisik S. (2019). Smart Lock: Why sloth is a driver of the IoT progress. Received 02.22.2019 from https://habr.com/en/post/441294/.
4. Rezan F. (2019). Best Smart Lock for Doors 2019. Received 09.01.2019 from https://10hitech.com/best-smart-lock/.
5. Geitgey A. (2019). Face Recognition Documentation Release 1.2.3. Received 29.08.2019 from https://buildmedia.readthedocs.org/media/pdf/face-recognition/latest/face-recognition.pdf.
6. Beyeler M. (2017). Machine Learning for OpenCV: Intelligent image processing with Python. USA: Packt Publishing.
7. Greenberg M. (2014). Flask Web Development: Developing Web Applications with Python. USA, CA: O'Reilly Media, Inc.
8. Jay A. Kreibich (2014). Using SQLite: Small. Fast. Reliable Choose Any Three. USA, CA: O'Reilly Media, Inc.
9. Gololobov V. N. (2019). Raspberry Pi for the curious. M: Science and Technology.
10. Jellema L. (2019). Changing views on integration - from Enterprise Service Bus to API Gateway, Serverless and iPaaS. Received 16.09.2019 from https://technology.amis.nl/2019/01/23/changing-views-on-integration-from-enterprise-service-bus-to-api-gateway-serverless-and-ipaas /.

INTELLIGENT INFORMATION SYSTEM "SMART LOCK" FOR THE PROTECTION OF APARTMENTS

Lev-Volodymyr Rybak, Petro Kravets

Lviv Polytechnic National University, Information Systems and Networks Department,
rybaklevko@ukr.net, ORCID: 0000-0002-5915-5520
Petro.O.Kravets@lpnu.ua, ORCID: 0000-0001-8569-423X

Intelligent information system "Smart lock" for the protection of apartments is considered and described. A general description of smart homes and the relevance of their use is made. Described an issue which appear when using smart home devices. Examples of two existing similar models are provided: SmartLock and Schlage Sense Smart Lock. An analysis of their advantages and disadvantages is made to take this information into account for the device that going to be created. A brief description of up-to-date information on face recognition techniques is provided. Based on this data, the goals that will be addressed in this article are formulated. The diagram of functioning of subsystems of the intellectual information system is given. A detailed description of each of the components of the information system is made, namely: server backend part, face recognition software, database and physical device "Smart lock". A step-by-step description of the process of creating each of these modules of the system is made. The choice of programming language, web tools, startup devices, operating system, database management system and face recognition software libraries is justified. A detailed description of the device on which the program will work and the device that was used to debug the program elements of the system. The approach to choosing the type of integration of the intelligent information system is explained. The principle of information system operation is described. The conceptual model of the system is demonstrated. The example of using this information system in real conditions is given. Instructions for the typical use of this information system by the user. The results were analyzed and conclusions were drawn on the relevance of the creation of the information system and its practical application.

Key words: intelligent information system, smart lock, face recognition, database management system, integration security, horizontal integration, conceptual model, functional model.