Т. 2, № 1(2), 2020

УДК 004.7

Г. І. Влах-Вигриновська, О. О. Іванюк, М. А. Вигриновський,
Н. Р. Малиновський
Національний університет "Львівська політехніка",
кафедра комп'ютеризованих систем автоматики

РОЗРОБЛЕННЯ ДОМАШНЬОЇ СИСТЕМИ БЕЗПЕКИ НА ОСНОВІ ТЕХНОЛОГІЙ ІНТЕРНЕТУ РЕЧЕЙ

https://doi.org/10.23939/amm2020.01.078

Пропонується платформа економічно ефективної розумної домашньої системи безпеки на базі Інтернету речей, що надає дані у хмарі для додатків. Розумна система безпеки включає мережу зондування пристроїв моніторингу стану будинку через бездротовий зв'язок в діапазоні ISM, контролера управління системою безпеки будинку з підключенням до мережі Інтернет за допомогою бездротового з'єднання Wifi, а резервний спосіб оповіщення користувача реалізовано SMS повідомленням через стільникову мережу GSM, хмарного сервера та додатку для мобільних телефонів на операційній системі Android. Розроблений додаток поєднує в собі застосування хмарного сервера, клієнтських датчиків та бази даних. Також з'ясовано апаратну та програмну реалізацію домашньої системи безпеки на основі IoT та їх відносні функції.

Ключові слова: Інтернет речей, Arduino, хмарний сервер, MQTT, мобільний додаток.

This article proposes a cost effective IoT smart home security platform that delivers data in the cloud for applications. The smart security system includes a network for sensing home monitoring devices via wireless ISM communication, a home security system control controller with an Internet connection using a Wifi connection, and a backup method of notifying the user is implemented by SMS message via GSM cellular network, cloud server and applications for mobile phones on the Android operating system. The developed application combines the application of a cloud server, client sensors and a database. It also clarified the hardware and software implementation of the IoT-based home security system and their relative functions.

Key words: Internet of Things, Arduino, Cloud Server, MQTT, Mobile Application.

Вступ

Інтернет речей (IoT) – це доповнення чинних засобів Інтернету для забезпечення зв'язку, з'єднання для роботи в мережі Інтернет між різними пристроями та фізичними об'єктами, також відомими як «Речі». З бурхливим розвитком Інтернету речей (IoT) безпека та конфіденційність систем розумного будинку, заснованих на IoT, стають дедалі популярнішими. А нещодавні досягнення смартфонів та доступних апаратних платформ з відкритим кодом дозволили розробити недорогі архітектури для систем безпеки з підтримкою Інтернету речей (IoT). Ці системи зазвичай складаються з сенсорного та виконавчого шару, який складається з датчиків, таких як пасивні інфрачервоні датчики, також відомі як датчики руху; датчики температури; датчики диму та веб-камери для нагляду за безпекою. Ці датчики, розумні електроприлади та інші пристрої IoT підключаються до Інтернету через домашній шлюз [1].

Пристрої IoT, як правило, виробляють велику кількість даних зондування, які передаються у різних форматах даних і зберігаються на хмарних серверах. Ці дані обробляються на хмарному

сервері різними способами, такими як доступ до файлів, обмін даними та спільний доступ до ресурсів. Крім того, хмарний сервер функціонує як контролер реального часу, передаючи дані за допомогою аналізу великих даних та обчислення даних. Віртуальні хмарні сервери широко використовуються для реалізації цих розроблених функцій. Завдяки своїй низькій вартості, високій швидкості та хорошій стабільності хмарний сервер може легко виконувати адміністрування мережі та керувати передачею даних від датчиків до серверів. Швидша передача даних відбувається за допомогою технології Wi-Fi, що допомагає користувачеві контролювати та контролювати системи в усьому світі.

Аналіз останніх досліджень та публікацій

В системі домашньої безпеки, основаної на платформі IoT, основний акцент на захисті наших близьких і нашого майна в дома. Сьогодні на ринку представлена велика кількість систем домашньої безпеки на базі Інтернету. Згідно з оглядом літературних джерел і ринку, загальними вимогами до систем домашньої безпеки є наявність платформи IoT для цілодобового спостереження та виявлення зловмисників, економічність та точність системи повідомлень в реальному часі запропоновані різними дослідниками. Нижче приведено вклад різних дослідників в області Інтернет-речей.

В роботі [2] йдеться про розробку та впровадження системи домашньої безпеки на основі IoT, яка забезпечує виявлення несанкціонованого або небажаного вторгнення або рух за допомогою різних апаратних та програмних засобів і через електронну пошту чи голосовим сповіщенням інформує уповноважену особу. У цій роботі контролер безпеки реалізовано одноплатному комп'ютері Raspberry Pi.

В роботі [3] пропонується система безпеки для середовища IoT, яка запобігає проникненню додому, банку, чи будь-якому іншому місці системи безпеки. Щоб виявити зловмисну діяльність або порушення конфіденційності, запропоновано систему виявлення несанкціонованого входу людей (UHEDS), яка виявляє будь-яке вторгнення або порушення та, як правило, повідомляє власника будинку. Проєкт включає техніку на основі Anomaly для несанкціонованого виявлення входу та аналізу підписів з використанням алгоритму розпізнавання обличчя, що працює на хмарі AWS, для розмежування між уповноваженою особою та зловмисником і, таким чином, покращуючи точність авторизації законної особи та забезпечуючи доступ до приватної/особистої зони, тим самим зменшуючи ризик надсилання помилкових попереджень/тривог.

В роботі [4] запропоновано використати додаток IoT Домашня безпека для створення недорогої системи безпеки як для будинку, так і для промислового використання. Система проінформує власника про будь-яке несанкціоноване проникнення або про те, що двері відкриваються відправкою повідомлення користувачу. Отримавши повідомлення, користувач може вжити необхідних заходів. Основним вузлом системи безпеки є мікроконтролер Arduino Uno для взаємодії між компонентами, магнітний геркон для моніторингу статусу, зумер для подачі сигналу тривоги та модуль WiFi ESP8266 для підключення до мережі Інтернет. Основні переваги такої системи включають простоту настройки, більш низькі витрати та низькі експлуатаційні витрати.

Мета роботи

Мета роботи - розробити недорогу, розширювану, гнучку бездротову домашню систему безпеки на основі Інтернету речей із мобільною підтримкою, доступ до якої можна отримати у будь-якій точці світу.

Виклад основного матеріалу

Згідно з дослідженнями ринку, загальними параметрами або характеристиками домашньої системи безпеки є: простота підключення, надійність, ефективна, швидка та точна система оповіщення, зручний інтерфейс користування.

Врахувавши всі перераховані властивості була створена наступна система безпеки. Рисунок 1. демонструє структуру домашньої системи безпеки на базі IoT, яка умовно розділена на три частин. Перша частина мікроконтролерний блок, друга – веб-сервер, третя – користувацький інтерфейс.

Розглянемо детальніше взаємодію між блоками схеми. Система безпеки може знаходитися у двох станах: Увімкнений/Вимкнений.

Мікроконтролерна частинаскладається з:

- давачів та охоронного обладнання, які відповідають за цілодобовий моніторинг приміщення спостереження, якщо система увімкнута. Забезпечує виявлення небажаного переміщення людей у будинку.
- протокол передачі даних (ППД) - являє собою інтерфейсом взаємодії між блоками давачів та контролером.
- контролер безпеки відповідає за аналіз даних отриманих з давачів, передачу інформації про стан, безпосереднє сповіщення системи про втручання на приватну власність у випадку відхилення даних від норми.

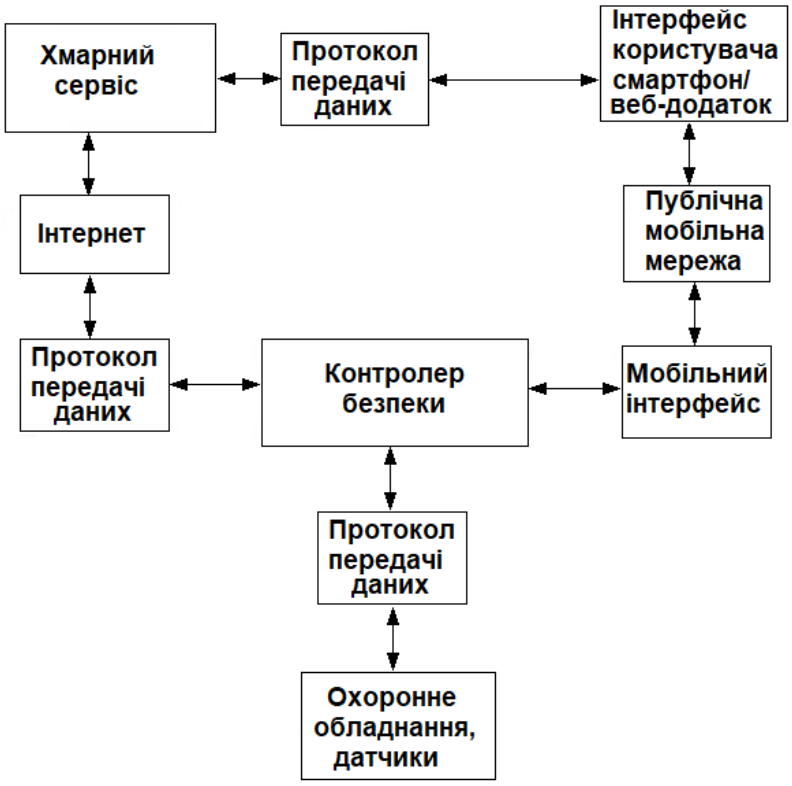


Рис. 1. Структурна схема системи безпеки розумного будинку

Структурна схема контролера управління системою безпеки представлена на рис. 2.

Сьогодні на ринку присутня безліч пропозицій від виробників датчиків та охоронного обладнання, таких як інфрачервоні датчики руху; магнітоконтактні, вібраційні, акустичні комбіновані та веб-камери для нагляду за безпекою. Інтерфейсом взаємодії між блоками датчиків та мікроконтролером обрано приймально/передавальний радіомодуль NRF24L01, який здатний об'єднати до семи датчиків з в загальну радіомережу за топологію зірка зі швидкістю передачі даних від 250 кбіт/с до 2 Мбіт/с та радіусом дії 100 метрів. Завдяки NRF24L01 стає можливим розв’язувати технічні проблеми по збору даних з декількох датчиків одночасно. За допомогою радіомодуля, зондуються дані з давачів, та упаковуються і у вигляді пакетів і надсилаються на контролер управління – це мозок моніторингової частини системи. На роль контролера було вибрано мікроконтролер Arduino

Uno для подальшого аналізу та реагування на події. Однак у мікроконтролера через свої компактні розміри відсутні інтерфейси Ethernet, Wi-Fi. Тому було прийнято рішення для організації зв'язку між мікроконтролером та хмарним сервером по протоколу прикладного рівня MQTT використати модуль ESP8266 з інтерфейсом Wi-Fi, забезпечивши вихід в Інтернет для передачі сигналу тривоги уповноваженим особам про втручання зловмисника на приватну власність. У випадку, коли з певних причин користувач або мікроконтролер не мають можливості під 'єднатися до мережі Інтернет, спрацьовує додаткова гілка структури, а саме блок мобільної мережі. За допомогою даного блоку можлива одностороння взаємодія між мікроконтролерною частиною та інтерфейсом користувача. Дана послідовність передбачена як додатковий спосіб оповіщення користувача у вигляді SMS повідомлення, що підвищує можливість попередження крадіжки майна. Ця можливість стає доступною при підключенні GSM модуля SIM800L [5]. Для того, щоб контролеру управління системою безпеки мати змогу відправляти дані на сервер за допомогою MQTT, потрібно виконати налаштування MQTT клієнта [6]. Для цього використовуємо чинну бібліотеку для роботи з MQTT. Такою бібліотекою є Arduino - MQTT.

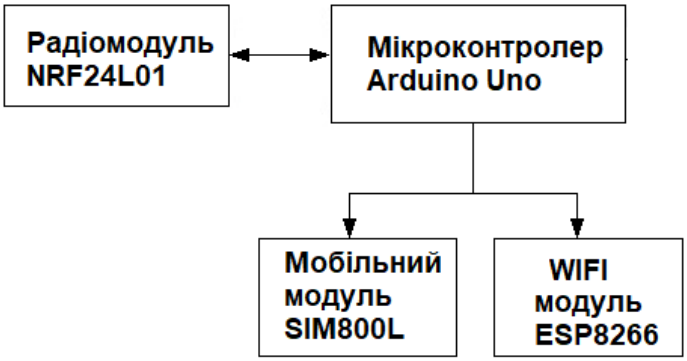


Рис. 2. Структурна схема контролера управління системою безпеки

Публікація даних на сервері. Після ініціалізації, віддалений контролер управління системою безпеки (видавець) може відправляти повідомлення за допомогою методу ***publish(param1, param2),*** де param1 та param2 текстові значення теми за якою публікується повідомлення та яке саме повідомлення, відповідно. Контролер своєю чергою отримує дані за допомогою методу ***onMessage(),*** який викликається при запуску програми [6].

Веб - серверна частина складається з:

- База даних зберігає деталі усіх домашніх пристроїв та їх поточного стану. Дані користувачів та їх ключі автентифікації.
- Протоколи передачі даних – набір угод інтерфейсу логічного рівня, які визначають обмін даними через інтернет між веб-серверною та мікроконтролерною частинами.

Наступною складовою нашої системи є хмарний сервіс. Хмарні сервіси (public cloud services) – це програми та платформи, які «живуть» та працюють на серверах хмарних провайдерів. Головна особливість хмарних сервісів полягає в тому, що створюючи обліковий запис на такій платформі, людина зможе отримувати доступ до власної інформації з будь-якого гаджета в будь-якій точці світу. Для нашого проєктного рішення було обрано публічний хмарний сервер CloudMQTT для IoT пристроїв. CloudMQTT – ідеальне рішення для обміну повідомленнями "Інтернету речей" між датчиками, пристроями IoT та мобільними пристроями, використовує механізм видавець/підписник.

В нашому проєкті хмарний сервер CloudMQTT у разі отримання сповіщення тривоги з контролера управління системою безпеки, виконає запис необхідних даних та здійснить відправку оповіщення на користувацький інтерфейс.

Користувацький інтерфейс. Користувач (підписник), який віддалено отримує доступ до свого будинку, може запитувати інформацію про стан пристрою з бази даних та змінювати її через брокер CloudMQTT. Користувацькі інтерфейси забезпечують взаємодію користувача із системою домашньої безпеки. Розуміння переваг користувача та взаємодії користувача з пристроєм має вирішальне значення для забезпечення домашньої безпеки.

Роль користувацького інтерфейсу в нашій системі відіграє додаток для мобільних смартфонів на базі операційної системи Android. Android – це мобільна операційна система, заснована на модифікованій версії ядра Linux та іншого програмного забезпечення з відкритим кодом, розробленого головним чином для мобільних пристроїв із сенсорним екраном, таких як смартфони та планшети.

Створення Android додатку. Додатки, що розширюють функціональність пристроїв, записуються за допомогою набору програм для розробки програмного забезпечення Android (SDK), в основному, за допомогою мови програмування Java. Java може поєднуватися з C/C ++, разом з вибором режимів невиконання за замовчуванням, які дозволяють покращити підтримку C ++. Мова програмування Go також підтримується, хоча з обмеженим набором інтерфейсів прикладного програмування (API). У травні 2017 року Google оголосила про підтримку розробки додатків для Android мовою програмування Kotlin. Android Studio, заснований на IntelliJ IDEA, є основним середовищем розробки програм.

Архітектура додатку. Застосованою методологією став шаблон архітектури Model-View-ViewModel (MVVM). Його концепція полягає в відділенні логіки представлення даних від бізнес-логіки шляхом винесення її в окремий клас для кращого розмежування [7].

Що означає кожна з трьох частин в назві:

- Model - це логіка, яка пов'язана з даними додатка. Іншими словами – це POJO, класи роботи з API, базою даних та ін.
- View - власне, це логіка інтерфейсу, в якому розташовуються всі необхідні віджети для відображення інформації.
- ViewModel - об'єкт, в якому описується логіка поведінки View в залежності від результату роботи Model. Можна назвати його моделлю поведінки View. Це може бути як форматування тексту, так і логіка управління видимістю компонентів або відображення станів, таких як завантаження, помилка, порожні екрани та інше. Також в ній описується поведінка, яка була ініційована користувачем (введення тексту, натискання на кнопку, свайп та ін.).

Ініціалізація MQTT клієнта та методи передачі даних. Користувач відкриває додаток, на початку кожного запуску програма ініціалізує MQTT клієнт необхідний для передачі даних між смартфоном та веб-сервером. Для цього використовується бібліотека ***Eclipse Paho Android Service.***

Для забезпечення отримання даних з веб-серверу, клієнт повинен підписатися на відповідні теми повідомлення.

Публікація повідомлення забезпечується методом ***publishMessage(topic, message),*** де ***topic*** – тема повідомлення, ***message*** - саме повідомлення.

Класова діаграма. На рис. 3 зображена класова діаграма роботи додатку. При запуску програми створюється екземпляр класу Application в якому реалізована ініціалізація MQTT клієнта, та локальної бази даних. Об’єкт ***MqttClient*** передається в конструктор об’єкта ***MessageManager*** який відповідає за публікування повідомлень та підписку на теми повідомлень. Об’єкт Database виконує функції збереження даних користувачів, повідомлень, та інформації про будинок. Класи з суфіксом ***Fragment*** – це компоненти системи Андроїд які являються класами опису графічного інтерфейсу екрану.

До прикладу, об’єкт типу ***UsersFragment*** (екран “мешканці”) відповідає за відображення інформації користувачів та показ повідомлень у разі виникнення тих чи інших помилок. Для кожного фрагмента реалізується прив’язаний до нього клас, який працює з даними, позначений з суфіксом ***ViewModel.*** Так клас ***UsersViewModel*** забезпечує завантаження даних користувачів з

хмари, відправку їхньої інформації до локальної бази даних та оновлення даних користувачів. Саме тому дані класи працюють з об'єктами ***MessageManager*** та ***Database***.

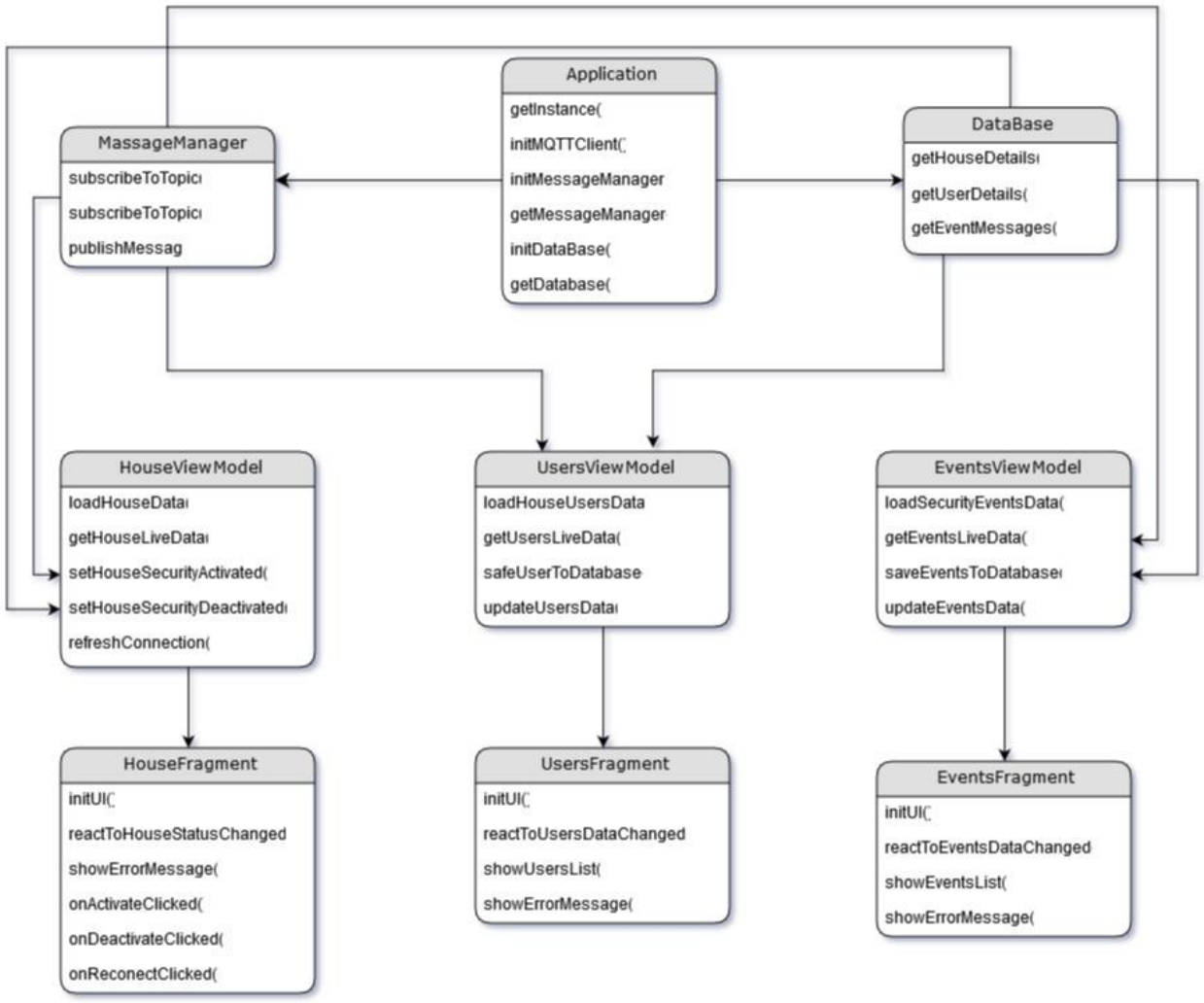


Рис. 3. Класова діаграма

*Алгоритм роботи та навігація додатку***.** На початковому екрані програми – рис. 4 (А), алгоритм роботи якого зображено на рис. 5 (А), клієнту пропонується автентифікуватися шляхом введення логіну та паролю. Ці дані відправляються на веб-сервер та проходять перевірку, після чого веб-сервер відсилає відповідь про успішну або не успішну автентифікацію. У разі підтвердження особи користувача програма отримує дані про користувача та зберігає їх в локальній базі даних. При наступному відкритті додатку початковий екран відображатиме тільки поле вводу паролю або можливість зміни користувача, для зручності користування – рис. 4 (Б).

На головному екрані додатку – рис. 4 (В), алгоритм – рис. 5 (Б), відображаються основні дані будинку рис. 4 (Г, Д), статус системи безпеки будинку тобто чи захист активований чи ні. Доступна кнопка активації/дезактивації охорони будинку. Користувач натискає на кнопку активації, тим самим публікуючи повідомлення про активацію охорони, веб-сервер змінює статус охорони на *Активовано* та публікує повідомлення про зміну статусу. Мікроконтролер отримавши повідомлення, також змінює статус та починає працювати в режимі охорони будинку, та публікує повідомлення активації. Дане повідомлення сповіщає сервер та користувацький інтерфейс про успішне завершення активації. Після чого користувацький інтерфейс реагує відображенням активованого статусу та можливістю деактивувати охорону.

На екранні “Мешканці” – рис. 4 (Є), відображається інформація про активних мешканців тобто користувачів системою безпеки даного будинку. Перехід на даний екран доступний за допомогою навігаційної кнопки внизу екрану.

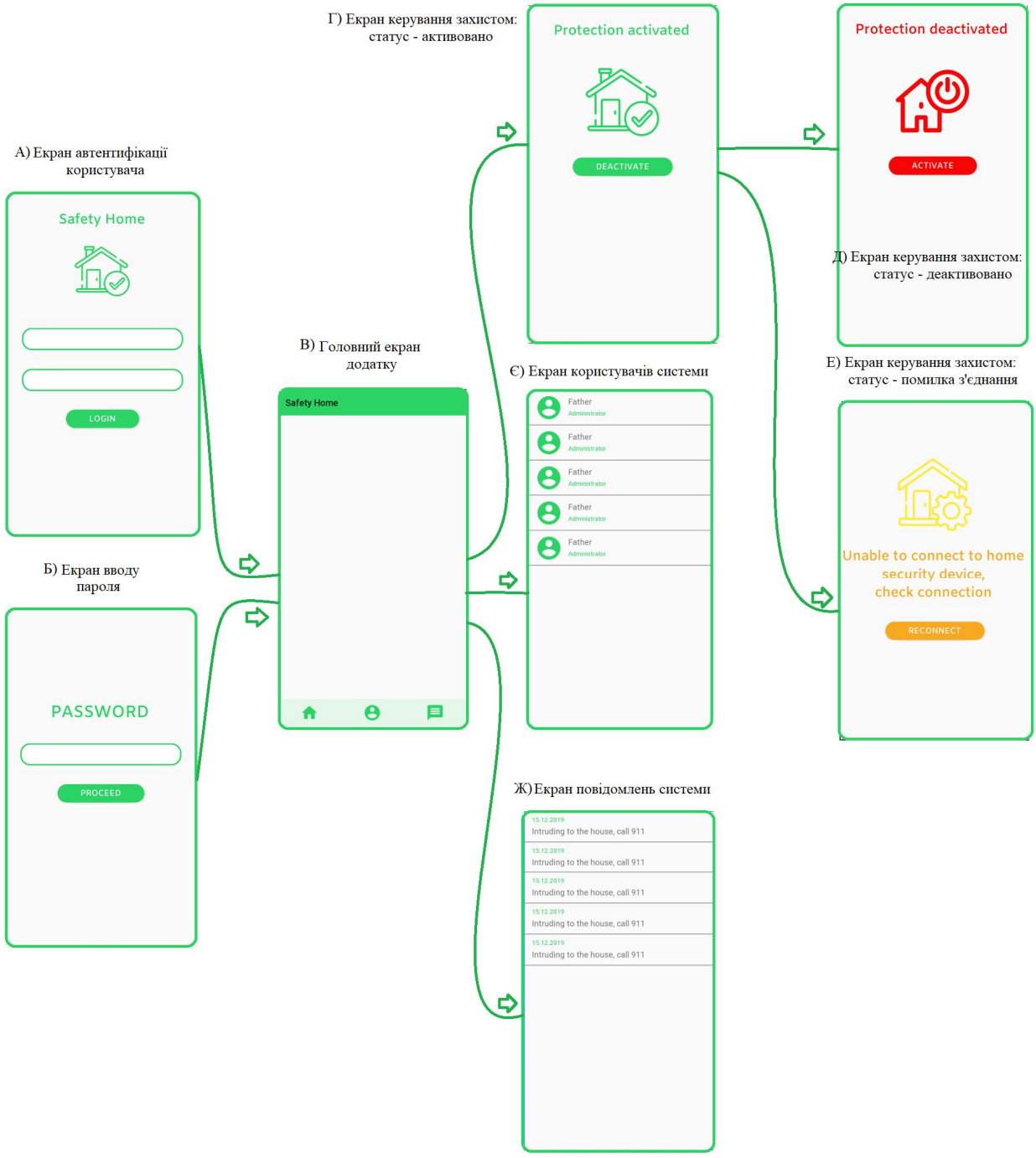
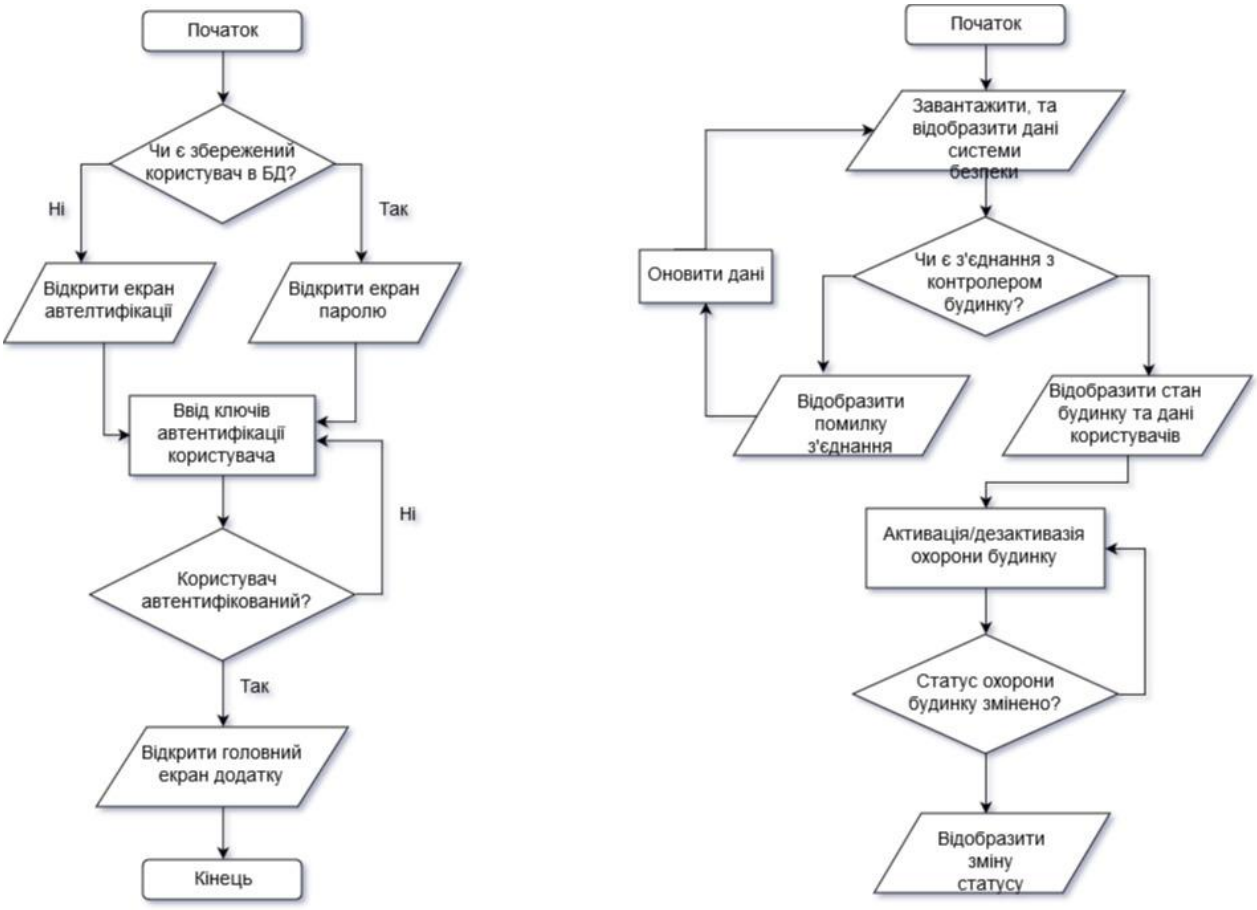


Рис. 4. Навігація екранів додатку

Не менш важливим є статус з’єднання веб-сервера з будинком, а точніше з контролером управління системою безпеки. Якщо веб-сервер не може з’єднатися з будинком, сервер публікую відповідне повідомлення про відсутність з’єднання. Отримавши дане повідомлення, на головному екрані додатку відображатиметься відповідне повідомлення про недоступність системи безпеки –

рис. 4 (Е). Така поведінка можлива, наприклад у випадку відключення контролера від електромережі. Після розв'язання проблем з контролером та успішного з'єднання з сервером, сервер публікує повідомлення про активність системи захисту будинку і додаток реагує на дану подію відображенням даних про будинок та навігацією керування системою будинку.

Всі події які відбуваються з системою записуються в журналі на сервері та доступні для перегляду на екрані "Повідомлень" – рис. 4 (Ж).



а *б*

Рис. 5. Алгоритм роботи початкового (А) та головного (Б) екранів

Висновки

На основі проведених нами досліджень можна стверджувати наступне:

1. Для підвищення надійності роботи системи мікроконтролерний блок підключається до Інтернету за допомогою бездротового з'єднання Wi-Fi, а *мережа GSM* використовується як резервний канал зв'язку, причому обмін повідомленнями в реальному часі між мікроконтролерним блоком та користувацьким інтерфейсом реалізовано по протоколу MQTT через хмарний сервер CloudMQTT.

2. Для розроблення Android-додатку доцільно обирати шаблон проєктування MVVM, що дає змогу підвищити швидкість розробки та внесення змін системи. Для взаємодії додатку з хмарним сервером обрано бібліотеку Eclipse Paho Android Service.

3. Експерименти з верифікації показали, що середній час створення одного повідомлення за допомогою MQTT протоколу не перевищує 589 мікросекунд.

Список літератури

1. *Интернет вещей: Учебное пособие. А. В. Росляков, С. В. Ваняшин, А. Ю. Гребешков. Самара: ПГУТИ, 2015. – 200 с.*

2. *1IoT Based Home Security System Using Raspberry Pi with Email and Voice Alert. Reena Rani, S. Lavanya,. B.Poojitha Published Computer Science, 2018 pp 21-26.*

3. *Home Security System using IOT and AWS Cloud Services.Mahendra Mehra; Vedant Sahai; Pratik Chowdhury; Elvis Dsouza. International Conference on Advances in Computing, Communication and Control (ICAC3), 2019 pp 159-163.*

4. *Anitha A, "Home security system using internet of things", IOP Conf. Series: Materials Science and Engineering 263, 2017 pp 1-11.*

5. *Molugu Surya Virat, Aishwarya B, Bindu S M, Dhanush B N and Manjunath R Kounte. "Security and Privacy Challenges in Internet of Things", International Conference on Trends in Electronics and Informatics, 11-12, May 2018 pp 454-460.*

6. *OneM2M Architecture Based Secure MQTT Binding in Mbed OS. Ahsan Muhammad, Bilal Afzal, Bilal Imran, Asim Tanwir, Ali Hammad Akbar, Ghalib A. Shah. Computer Science. IEEE European Symposium on Security and Privacy, 2019 pp 48-56.*

7. *Электронная книга "Шаблоны корпоративного приложения". URL: https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/enterprise-application-patterns/introduction.*

Reference

1. *Internet veschey: uchebnoe posobie. A.V. Roslyakov, S.V. Vanyashin, A.Yu. Grebeshkov. – Samara: PGUTI, 2015, 200 c.*

2. *1IoT Based Home Security System Using Raspberry Pi with Email and Voice Alert. Reena Rani, S. Lavanya,. B.Poojitha Published Computer Science, 2018 pp 21-26.*

3. *Home Security System using IOT and AWS Cloud Services.Mahendra Mehra; Vedant Sahai; Pratik Chowdhury; Elvis Dsouza. International Conference on Advances in Computing, Communication and Control (ICAC3), 2019 pp 159-163.*

4. *Anitha A, "Home security system using internet of things", IOP Conf. Series: Materials Science and Engineering 263, 2017 pp 1-11.*
https://doi.org/10.1088/1757-899X/263/4/042026

5. *Molugu Surya Virat, Aishwarya B, Bindu S M, Dhanush B N and Manjunath R Kounte. "Security and Privacy Challenges in Internet of Things", International Conference on Trends in Electronics and Informatics, 11-12, May 2018 pp 454-460.*

6. *OneM2M Architecture Based Secure MQTT Binding in Mbed OS. Ahsan Muhammad, Bilal Afzal, Bilal Imran, Asim Tanwir, Ali Hammad Akbar, Ghalib A. Shah. Computer Science. IEEE European Symposium on Security and Privacy, 2019 pp 48-56.*

7. *Elektronnaya kniga "SHablony korporativnogo prilozheniya". URL: https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/enterprise-application-patterns/introduction*