Safe Home

NURE Students

Oleksandr Kozhanov

Nikita Lisovyi

Vladyslav Bilous

**Software Requirements Specification**

**Document**

**Version: 1.0** **Date: 06/05/2025**

**Table of Contents**

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations. 3

1.4 References 3

1.5 Overview 4

2. The Overall Description 4

2.1 Product Perspective 4

2.1.1 System Interfaces 4

2.1.2 Interfaces 4

2.1.3 Hardware Interfaces 4

2.1.4 Software Interfaces 5

2.1.5 Communications Interfaces 5

2.1.6 Memory Constraints 5

2.1.7 Operations 5

2.1.8 Site Adaptation Requirements 6

2.2 Product Functions 6

2.3 User Characteristics 6

2.4 Constraints 6

2.5 Assumptions and Dependencies 7

2.6 Apportioning of Requirements. 7

3. Specific Requirements 7

3.1 External Interfaces 7

3.2 Functions 8

3.3 Performance Requirements 8

3.4 Logical Database Requirements 9

3.5 Design Constraints 9

3.5.1 Standards Compliance 9

3.6 Software System Attributes 9

3.6.1 Reliability 9

3.6.2 Availability 9

3.6.3 Security 9

3.6.4 Maintainability 9

3.6.5 Portability 10

3.7 Organizing the Specific Requirements 10

3.8 Additional Comments 10

# 1. Introduction

## 1.1 Purpose

*Цей документ описує вимоги до програмного забезпечення для системи Safe Home, призначеної для забезпечення безпеки житла шляхом інтеграції IoT-пристроїв, таких як магнітні датчики для дверей та вікон. Документ призначений для розробників, тестувальників, менеджерів проекту та зацікавлених сторін, щоб забезпечити чітке розуміння функціональності системи та її вимог.*

## 1.2 Scope

*Назва продукту: Safe Home*

*Опис: Safe Home — це цифрова платформа для створення власної системи безпеки житла. Вона дозволяє користувачам підключати та налаштовувати магнітні датчики, отримувати сповіщення в реальному часі про несанкціоновані дії, переглядати історію подій та керувати системою через мобільний або веб-додаток. Система не включає інтеграцію з іншими розумними пристроями (освітлення, термостати) у першому релізі.*

*Переваги та цілі:*

1. *Зниження витрат на охорону порівняно з традиційними охоронними компаніями.*
2. *Забезпечення зручного та доступного рішення для моніторингу безпеки оселі.*
3. *Надання гнучкості в інтеграції сторонніх датчиків.*

## 1.3 Definitions, Acronyms, and Abbreviations.

* *IoT: Інтернет речей (Internet of Things).*
* *API: Інтерфейс програмування додатків (Application Programming Interface).*
* *FCM: Firebase Cloud Messaging — сервіс для надсилання push-сповіщень.*
* *AES-256: Стандарт шифрування даних.*
* *TLS: Transport Layer Security — протокол для безпечної передачі даних.*
* *RESTful API: Архітектурний стиль для створення веб-сервісів.*

## 1.4 References

1. *Vision and Scope Document для Safe Home, версія 1.0, 20.04.2025.*
2. *Google OAuth 2.0 Documentation:* [*https://developers.google.com/identity/protocols/oauth2*](https://developers.google.com/identity/protocols/oauth2)
3. *Firebase Cloud Messaging Documentation:* [*https://firebase.google.com/docs/cloud-messaging*](https://firebase.google.com/docs/cloud-messaging)
4. *Stripe API Documentation:* [*https://stripe.com/docs/api*](https://stripe.com/docs/api)

## 1.5 Overview

*Документ складається з трьох основних розділів:*

* *Розділ 1: Вступ, мета, сфера застосування та визначення.*
* *Розділ 2: Загальний опис продукту для зацікавлених сторін.*
* *Розділ 3: Детальні вимоги для розробників і тестувальників.*

# 2. The Overall Description

## 2.1 Product Perspective

*Safe Home — це незалежна платформа, яка інтегрує IoT-пристрої для моніторингу безпеки житла. Система не є частиною більшої системи, але взаємодіє з зовнішніми сервісами, такими як Google OAuth, Firebase Cloud Messaging та Stripe для обробки платежів.*

### 2.1.1 System Interfaces

*Система взаємодіє з:*

1. *Google OAuth: Для автентифікації користувачів.*
2. *Firebase Cloud Messaging (FCM): Для надсилання push-сповіщень.*
3. *Stripe API: Для обробки платежів за преміум-підписку.*

### 2.1.2 Interfaces

1. *Мобільний додаток: Розроблений на Kotlin + Android Studio, підтримує автентифікацію, моніторинг датчиків у реальному часі, налаштування режимів охорони, перегляд історії подій.*
2. *Веб-додаток: Розроблений на React, включає адмін-панель для управління користувачами, підписками та статистикою.*

*Інтерфейс оптимізований для користувачів без технічних знань, відповідає вимогам ADA (доступність для людей з обмеженими можливостями).*

### 2.1.3 Hardware Interfaces

*Система взаємодіє з магнітними датчиками для дверей та вікон через IoT-протоколи. Використовується симулятор Wokwi для тестування інтеграції з датчиками.*

### 2.1.4 Software Interfaces

1. *Flask: RESTful API для серверної частини.*

* *Версія: 2.x*
* *Джерело:* [*https://flask.palletsprojects.com*](https://flask.palletsprojects.com)
* *Мета: Обробка запитів від клієнтських додатків.*

1. *PostgreSQL: База даних для зберігання інформації.*

* *Версія: 15.x*
* *Джерело:* [*https://www.postgresql.org*](https://www.postgresql.org)
* *Мета: Зберігання даних користувачів, осель, датчиків та подій.*

1. *SQLAlchemy: ORM для інтеграції з PostgreSQL.*

* *Версія: 2.x*
* *Джерело:* [*https://www.sqlalchemy.org*](https://www.sqlalchemy.org)

1. *Google OAuth: Для автентифікації.*
2. *Firebase Cloud Messaging: Для push-сповіщень.*
3. *Stripe: Для обробки платежів.*

### 2.1.5 Communications Interfaces

1. *HTTPS: Для безпечної передачі даних між клієнтом і сервером (TLS).*
2. *IoT-протоколи: Для зв’язку з магнітними датчиками (залежить від специфікацій датчиків).*
3. *WebSocket: Для моніторингу в реальному часі.*

### 2.1.6 Memory Constraints

*Система розрахована на роботу на пристроях із щонайменше 2 ГБ оперативної пам’яті для мобільного додатка та 4 ГБ для веб-додатка. Серверна частина на Azure повинна підтримувати масштабування для обробки до 10 000 активних користувачів.*

### 2.1.7 Operations

*Режими роботи:*

* + *Інтерактивний режим: користувачі взаємодіють через мобільний або веб-додаток.*
  + *Автоматизований режим: система автоматично надсилає сповіщення при порушенні режиму охорони.*

*Резервне копіювання: дані зберігаються в хмарі Azure з щоденним резервним копіюванням.*

*Відновлення: система підтримує відновлення даних після збою з втратою не більше 5 хвилин даних.*

### 2.1.8 Site Adaptation Requirements

* *Магнітні датчики повинні бути встановлені на дверях/вікнах користувача до активації системи.*
* *Необхідне стабільне інтернет-з’єднання (щонайменше 1 Мбіт/с).*

## 2.2 Product Functions

1. *Реєстрація та автентифікація користувачів (включаючи Google OAuth).*
2. *Управління профілями осель (створення, редагування, видалення, архівування).*
3. *Інтеграція та управління магнітними датчиками.*
4. *Моніторинг стану датчиків у реальному часі.*
5. *Налаштування режимів охорони ("вдома", "поза домом", "нічний режим").*
6. *Надсилання push-сповіщень та email-повідомлень про порушення безпеки.*
7. *Перегляд історії подій.*
8. *Управління преміум-підпискою через Stripe.*
9. *Адмін-панель для управління користувачами, підписками та статистикою.*

## 2.3 User Characteristics

1. *Кінцеві користувачі: власники осель, батьки, які прагнуть контролювати безпеку житла або дітей. Рівень технічної підготовки: базовий.*
2. *Адміністратори: системні адміністратори з досвідом роботи з веб-інтерфейсами.*
3. *Розробники: технічний персонал із знаннями Flask, React, PostgreSQL.*

## 2.4 Constraints

1. *Залежність від стабільного інтернет-з’єднання.*
2. *Підтримка лише магнітних датчиків у першому релізі.*
3. *Максимальний час відповіді системи — 5 секунд.*
4. *Підтримка лише англійської та української мов у першому релізі.*
5. *Обмеження API датчиків може вплинути на сумісність.*

*.*

## 2.5 Assumptions and Dependencies

* *Припущення:*
  1. *Користувачі мають стабільне інтернет-з’єднання.*
  2. *Користувачі довіряють системі щодо безпеки даних.*
  3. *Базовий рівень технічної підготовки для налаштування датчиків.*
* *Залежності:*
  1. *Сумісність із магнітними датчиками.*
  2. *Надійність хмарного середовища Azure.*
  3. *Доступність Google OAuth, FCM, Stripe.*

## 2.6 Apportioning of Requirements.

* *Перший реліз: реєстрація, автентифікація, управління оселями/датчиками, моніторинг, сповіщення, адмін-панель (MF1–MF20).*
* *Наступні релізи: підтримка інших типів датчиків, біометрична автентифікація, офлайн-режим, багатомовність, автоматизації.*

# 3. Specific Requirements

## 3.1 External Interfaces

***Вхідні дані****:*

* *Дані від магнітних датчиків ("відчинено"/"зачинено").*
* *Дані користувача (email, пароль, Google OAuth токен).*
* *Налаштування режимів охорони.*

***Вихідні дані****:*

* *Push-сповіщення та email-повідомлення.*
* *Історія подій у форматі JSON.*
* *Статистика для адмін-панелі.*

***Формати даних****: JSON для API, зашифровані дані (AES-256) для чутливої інформації.*

***Формати інтерфейсу****:*

* *Мобільний додаток: інтерактивний UI з підтримкою сенсорного введення.*
* *Веб-додаток: таблиці, графіки для адмін-панелі.*

## 3.2 Functions

1. *Реєстрація користувача (User Registration).*
2. *Підтвердження електронної пошти (Email Verification).*
3. *Реєстрація через Google (Google Registration).*
4. *Вхід у систему через логін та пароль (Login with Email and Password).*
5. *Вхід у систему через Google (Login with Google).*
6. *Зброс пароля (Reset Password)*
7. *Зміна паролю (Change Password)*
8. *Створення профілю оселі (Create Home Profile).*
9. *Редагування профілю оселі (Edit Home Profile).*
10. *Видалення профілю оселі (Delete Home Profile).*
11. *Архівування\розархівування оселі (Archive\Unarchivе Home)*
12. *Додавання датчика до оселі (Add Sensor to Home).*
13. *Видалення датчика (Delete Sensor).*
14. *Архівування\розархівування сенсору (Archive\Unarchivе Sensor)*
15. *Вибір режиму охорони (Select Security Mode).*
16. *Перегляд стану датчиків у реальному часі (Real-Time Sensor Monitoring).*
17. *Увімкнення та вимкнення конкретного датчика (Enable/Disable Specific Sensor).*
18. *Push-сповіщення про порушення, зміни в режимі охорони (Push Notifications).*
19. *Перегляд історії подій (View Event History).*
20. *Придбання преміум підписки з розширеною кількістю осель та датчиків. (Purchase a premium subscription)*
21. *Push-сповіщення та нотифікація на email при закінченні дії підписки (Notification for subscription)*
22. *Перегляд списку користувачів*
23. *Видалення профілю користувача*
24. *Перегляд історії підписок користувача*
25. *Відміна поточної підписки користувача*
26. *Перегляд списку адміністраторів*
27. *Реєстрація адміністратора, сповіщення про реєстрацію*
28. *Створення планів підписок*
29. *Редагування планів підписок*
30. *Перегляд адмін аудит логів*
31. *Перегляд статистики*

## 3.3 Performance Requirements

* *Система повинна підтримувати до 10 000 активних користувачів.*
* *Максимальний час відповіді: 5 секунд.*
* *99.9% безвідмовної роботи.*

## 3.4 Logical Database Requirements

* *Типи даних:*
  + *Дані користувачів (email, зашифрований пароль, Google ID).*
  + *Профілі осель (назва, адреса, статус).*
  + *Дані датчиків (ID, тип, стан, час події).*
  + *Історія подій (час, подія, датчик).*
* *Частота використання: реальний час для моніторингу, щоденний доступ до історії подій.*
* *Цілісність: унікальність email, зв’язок між оселями та датчиками.*
* *Тривалість зберігання: дані подій зберігаються протягом 1 року.*

## 3.5 Design Constraints

1. *Використання Flask, PostgreSQL, React, Kotlin.*
2. *Сумісність із магнітними датчиками через API.*

### 3.5.1 Standards Compliance

1. *Шифрування даних: AES-256, TLS.*
2. *Аудит безпеки: Відповідність ISO 27001.*
3. *Журнал аудиту: Запис усіх змін у профілях і підписках.*

## 3.6 Software System Attributes

### 3.6.1 Reliability

1. *MTBF (середній час між відмовами): 1000 годин.*
2. *Тестування: Автоматизовані тести навантаження.*

### 3.6.2 Availability

1. *99.9% безвідмовної роботи.*
2. *Відновлення після збою: не більше 5 хвилин втрати даних.*

### 3.6.3 Security

1. *Шифрування даних (AES-256, TLS).*
2. *Захист від SQL-ін’єкцій та XSS-атак.*
3. *Логування всіх дій адміністраторів.*

### 3.6.4 Maintainability

1. *Модульна архітектура (Flask, React).*
2. *Документація API та коду.*

### 3.6.5 Portability

1. *Система працює на Android (мобільний додаток) та будь-якому сучасному браузері (веб-додаток).*
2. *Серверна частина розгортається на Azure через GitHub Actions, клієнтська частина через Docker Azure Pipelines.*

## 3.7 Organizing the Specific Requirements

*Вимоги організовані за функціональною ієрархією:*

* *автентифікація та управління користувачами (MF1–MF7).*
* *управління оселями та датчиками (MF8–MF17).*
* *сповіщення та моніторинг (MF18–MF19).*
* *преміум-підписка (MF20–MF21).*
* *адмін-панель (MF22–MF31).*

## 3.8 Additional Comments

1. *Система підтримує інтеграцію з новими типами датчиків у майбутніх релізах.*
2. *Використання симулятора Wokwi для тестування IoT-інтеграції.*