

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Проектування та розробка інтерфейсу інформаційної
системи керування елементами обладнання внутрішньої інженерії розумного
будинку»

КУХНЮК АНАСТАСІЯ ОЛЕГІВНА

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Проектування та розробка інтерфейсу інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку»

Виконала: Кухнюк А.О.
(прізвище та ініціали)

Спеціальності: 122 «Комп'ютерні науки

Спеціалізація: «Інформаційні управляючі системи і технології»
(шифр і назва напрямку підготовки, спеціальності)

Групи: КН – 20-1

Керівник: к.т.н., доц. Саченко І. А.
(прізвище та ініціали)

Київ – 2024

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації й інформаційних технологій

Випускова кафедра: інформаційних технологій

Освітній ступінь: бакалавр за ОП

Спеціальність: Комп'ютерні науки

Освітня програма: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ

„___” _____ 2024 року

ЗАВДАННЯ

**ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА
ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР**

Кухнюк Анастасія Олегівна

(прізвище, ім'я та по батькові здобувача)

1. Тема роботи: Проектування та розробка інтерфейсу інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку

затверджена наказом ректора КНУБА № __ від «__» _____ 202__ року

2. Керівник роботи: Саченко Ілля Анатолійович вич, к.т.н, доцент
кафедри інформаційних технологій проектування та прикладної математики

(прізвище, ім'я та по батькові, науковий ступінь, вчене звання)

3. Строк подання Здобувачем роботи до захисту _____

4. Зміст пояснювальної записки за розділами:

P.1. Аналіз предметної області та постановка задачі _____

P.2. Алгоритмічне та математичне забезпечення _____

P.3. Розробка програмного забезпечення _____

P.4. Техніко-економічне обґрунтування розробки _____

5. Графічний матеріал за розділами:

P.1 _____
P.2 _____
P.3 _____
P.4 _____

6. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1	20.04.2024
Розділ 2	01.05.2024
Розділ 3	20.05.2024
Розділ 4	30.05.2024
Остаточне оформлення роботи	01.06.2024
Направлення роботи для перевірки на плагіат	03.06.2024
Попередній захист роботи на випусковій кафедрі	
Направлення роботи на рецензування	

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1			
Розділ 2			
Розділ 3			
Розділ 4			

8. Дата видачі завдання _____

Зав. кафедри _____
(підпис) (прізвище та ініціали)

Керівник _____
(підпис) (прізвище та ініціали)

Здобувач _____
(підпис) (прізвище та ініціали)

РЕЗЮМЕ (SUMMARY) до атестаційної випускної роботи Здобувач:	Сидоренко Владислав Володимирович Vladislav Sidorenko		
ЗВО	Київський національний університет будівництва і архітектури		
Тема (українською та англійською)	Проектування та розробка інтерфейсу інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку Design and Development of the Interface for the Information System for Managing Equipment Elements of Internal Engineering in a Smart House		
Освітній ступінь	Бакалавр		
Факультет	Автоматизації і інформаційних технологій		
Випускаюча кафедра	Інформаційних технологій		
Спеціальність	122 «Комп'ютерні науки»		
Освітня програма	Інформаційні управляючі системи та технології		
Керівник	Гончаренко Тетяна Андріївна		
Обсяг роботи:	поясню вальна записка, стор.	розділів	креслень формату А
	89	4	4
Ключові слова: Keywords:	Розумний будинок, автоматизована система, інтерфейс, веброзробка, проектування, підсистема Smart House, Automated System, Interface, Web Development, Design, Subsystem		

У роботі розроблено інтерфейс інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку. З метою розробки інформаційної системи для розумного будинку створено інтерфейс сайту для автоматизації комплексної системи, яка складається з трьох основних підсистем: загального та аварійного освітлення, відеоспостереження та протипожежного захисту.

Здобувач: _____ / Анастасія КУХНЮК /

Керівник: _____ / Тетяна ГОНЧАРЕНКО /

“ ___ ” _____ 2024 р

АНОТАЦІЯ

Кухнюк А.О. Проектування та розробка інтерфейсу інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку.

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», спеціалізація: «Інформаційні управляючі системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2020.

Робота присвячена процесу проектування та розробки інтерфейсу інформаційної системи для розумного будинку. Основна увага приділяється трьом підсистемам: освітленню, протипожежному захисту та відеоспостереженню. В рамках дослідження було проаналізовано сучасні технології веброзробки, які використовувалися для створення інтуїтивно зрозумілого та зручного інтерфейсу користувача.

Ключові слова: розумний будинок, автоматизована система, інтерфейс, веброзробка, проектування, підсистема.

SUMMARY

Kukhniuk A.O. Design and Development of the Interface for the Information System for Managing Equipment Elements of Internal Engineering in a Smart Home.

Bachelor's Graduation Thesis in Specialty: 122 "Computer Science", Specialization: "Information Control Systems and Technologies". – Kyiv National University of Construction and Architecture. – Kyiv, 2020.

The work is dedicated to the process of designing and developing the interface for the information system for a smart home. The main focus is on three subsystems: lighting, fire protection, and video surveillance. The study analyzed modern web development technologies used to create an user-friendly interface.

Keywords: smart home, automated system, interface, web development, design, subsystem.

ЗМІСТ

Вступ	9
Перелік умовних позначень	10
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ..	11
1.1. Опис предметної області	11
1.2 Аналіз об'єкта дослідження	14
1.3 Опис предмету дослідження	18
1.4 Аналіз актуальності	24
1.5 Стан вже існуючих рішень	26
1.6 Визначення цілей дослідження та постановка задачі	31
2. АЛГОРИТМІЧНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	34
2.1 Математичне забезпечення системи	34
2.2 Опис проєктних рішень з математичного забезпечення	37
2.3 Проєктування системи	40
2.4 Алгоритмічне забезпечення програми	49
3. ПРОЕКТНІ РІШЕННЯ. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	54
3.1 Опис проєктних рішень з програмного забезпечення	54
3.2 Розробка компонентів програмного забезпечення	60
3.2.1 Підготовка вхідних даних	60
3.2.2 Опис	63
3.2.2 Опис	класів
3.3 Опис тестового прикладу	66
4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ	72
4.1 Опис проєктованого продукту	72

4.2	Оцінка ринку збуту та аналіз конкурентів	74
4.3	Маркетингова стратегія	77
4.4	План виробництва	80
4.5	Фінансовий план та стратегія фінансування	82
4.6	Оцінка ризику та страхування	83
	ВИСНОВКИ.....	87
	Список використаних джерел	88

Вступ

У сучасному світі стрімкий розвиток технологій і впровадження концепції "розумного будинку" стають ключовими чинниками підвищення безпеки в житлових приміщеннях. "Розумний будинок" — це інтеграція автоматизованих підсистем, що дозволяють дистанційно керувати різними аспектами життєдіяльності будинку, такими як освітлення, протипожежний захист, відеоспостереження та іншими інженерними підсистемами. Одним з важливих елементів цієї концепції є створення інтерфейсу, який забезпечує зручне та інтуїтивне керування.

Тема дипломної роботи "Проектування та розробка інтерфейсу інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку" є актуальною і важливою з кількох причин. По-перше, ефективний інтерфейс користувача сприяє більш зручному та швидкому управлінню комплексною системою, таким чином, що підвищує рівень безпеки мешканців. По-друге, максимально інтуїтивний та ергономічний дизайн і функціональність інтерфейсу відповідно впливають на швидкість реагування підсистем та миттєві зміни в роботі та управлінні станом їх складових, як наслідок - значно підвищується оптимізація роботи всіх інженерних систем будинку. Також проєкт відповідає сучасним тенденціям та потребам ринку, і приносить реальну користь користувачам.

У даній роботі розглянуто основні етапи проектування та розробка інтерфейсу інформаційної системи, починаючи від аналізу вимог користувачів і закінчуючи тестуванням готового продукту.

Основна мета роботи полягає у розробці інтуїтивного та ефективного інтерфейсу, який відповідатиме сучасним стандартам і вимогам до систем керування елементами обладнання внутрішньої інженерії розумного будинку.

Таким чином, результати цього дослідження можуть бути корисними як для фахівців у галузі розробки програмного забезпечення, так і для кінцевих користувачів, які прагнуть покращити свій досвід взаємодії з розумними системами керування будинком.

Перелік умовних позначень

ІС – Інформаційна система

БД – База даних

UI - User interface

UX – User experience

ІТ – Інформаційні технології

ЖЦ – Життєвий цикл

ПЗ – Програмне забезпечення

ТЗ – Технічне завдання

УП – Управління проєктом

UCD – User-Centered Design

JS - JavaScript

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Опис предметної області

Предметною областю даної дипломної роботи є автоматизовані підсистеми управління в розумному будинку, які забезпечують підвищений рівень комфорту, безпеки та ефективного використання ресурсів у житлових приміщеннях.

«Smart house» - це термін насамперед, який описує здатність розпізнавати певні ситуації й відповідним чином на них реагувати, при цьому будівля повинна проектуватися так, щоб всі системи максимально ефективно консолідувались між собою, а їхнє обслуговування здійснювалося централізовано та фінансово необтяжливо.

Розумні будинки являють собою інтегровані системи, що використовують новітні технології для автоматизації та дистанційного керування різними аспектами житла. Вони забезпечують високий рівень комфорту та безпеки, покращуючи якість життя мешканців. В даній роботі особливу увагу приділено трьом ключовим системам: освітленню, протипожежному захисту та відеоспостереженню.

Функціональність таких типів будинків полягає у здатності фіксувати конкретні ситуації, що виникають всередині будівлі, і реагувати на них відповідно. Одна система може надавати інформацію, що обумовлює зміни в поведінці інших відповідно, використовуючи попередньо налаштовані алгоритми. Тобто можемо визначити, що основний аспект - об'єднання окремих підсистем у цілісний комплекс. Для забезпечення ефективної інтеграції різних підсистем необхідно використовувати стандартизовані інтерфейси та протоколи зв'язку.

Система управління дає власникам можливість створювати будь-які за складністю процедури функціонування, оскільки всі виконавчі підсистеми можуть працювати узгоджено і спільно. Тобто можемо стверджувати, що це веде до впровадження численних ресурсозаощаджувальних процедур, а саме:

- Віддалений контроль доступом, адже інформаційні канали є цифровими;
- Забезпечення безпеки;
- Контроль всіх параметрів системи;
- Комплексне і миттєве реагування на зміни в системі.

Важливо зазначити що дана система передбачає майже повне усунення обслуговчого персоналу для управління будинком. Тобто система самостійно або за допомогою контролю користувача програми реагує на зміну параметрів датчиків.

Для подальшого вивчення цієї теми визначимо ключові поняття, зокрема:

Автоматизована система – це система здатна самостійно збирати, обробляти та аналізувати дані, а також приймати рішення на основі запрограмованих алгоритмів. Автоматизовані системи використовуються для підвищення ефективності, точності та надійності процесів.

Програмна інтеграція – це процес об'єднання різних програмних компонентів, додатків та систем в єдину, взаємодійну інфраструктуру. Вона забезпечує безперебійну взаємодію між різними програмними продуктами, дозволяючи їм обмінюватися даними й функціонувати як єдине ціле.

Основні вимоги до системи – зручність, інформативність та інтуїтивність користувача. На сьогодні немає більш комфортної системи управління домашньою технікою, ніж «розумний будинок». Зручність у проєктах інтелектуальних будівель робить повсякденне життя простішим, дозволяючи мешканцям легко керувати освітленням, протипожежним захистом, безпекою та іншими системами через один інтерфейс. Це зменшує кількість зусиль, необхідних для виконання рутинних завдань.

Для бездоганної роботи цієї технології потрібно забезпечити ефективне і надійне функціонування окремих систем, що досягається правильним вибором і налаштуванням електронних компонентів та алгоритмів керування. Розробка

цілісної системи, яка включає підсистеми, стикається з рядом значних викликів, що включають декілька аспектів:

- Сумісність апаратних засобів. Необхідність уніфікації інтерфейсів для забезпечення коректної взаємодії між підсистемами освітлення, відеоспостереження та протипожежного захисту.
- Програмна інтеграція. Складність у створенні єдиного програмного забезпечення, яке б ефективно керувало всіма підсистемами й забезпечувало інтуїтивно зрозумілий інтерфейс для користувачів.
- Гнучкість. Налаштування різних сценаріїв роботи для задоволення специфічних потреб користувачів.

Створення даного проєкту вимагає ретельного планування і вирішення численних технічних та організаційних питань для забезпечення ефективної та безпечної інтеграції всіх підсистем.

1.2 Аналіз об'єкта дослідження

Об'єктом дослідження даної дипломної роботи є інформаційна система керування елементами обладнання внутрішньої інженерії «розумного будинку». Дослідження спрямоване на аналіз наявних систем, виявлення їхніх сильних та слабких сторін, а також на розробку і впровадження ефективного та інтуїтивно зрозумілого інтерфейсу для керування системами внутрішньої інженерії «розумного будинку». Це обумовлено прагненням підвищити рівень комфорту та безпеки житлових приміщень. Тому в сучасному світі технології автоматизації та інтелектуального управління будівель набувають все більшого значення.

Зважаючи на те, що екологічна ситуація, особливо у великих містах, залишає бажати кращого, а аварії на державних комунікаціях стають звичайним явищем і часто не забезпечують належного постачання ресурсів, питання створення «розумного будинку» набуває особливої актуальності. Саме тому люди ще з минулого століття намагалися автоматизувати побутові процеси. Для більш детального аналізу, пропонуємо розглянути етапи появи технологій розумного будинку та поступовий розвиток кожного з них.

Етапи появи технологій розумного будинку:

1. Перші ідеї:

Ідея автоматизації житлових приміщень виникла задовго до появи сучасних технологій. Перші згадки про автоматизовані системи можна знайти в науковій фантастиці початку 20 століття. Відомий письменник-фантаст Рей Бредбері у своїй розповіді "There Will Come Soft Rains" (1950) описав будинок, який самостійно виконував побутові завдання без участі людини.

2. Перші кроки:

Реальні технології автоматизації почали з'являтися в середині 20 століття. У 1960-х роках компанії почали розробляти перші системи автоматичного управління освітленням і опаленням. Наприклад, у 1966 році Джим Сазерленд створив систему "ЕСНО IV", яка була однією з перших домашніх комп'ютерних систем, здатних керувати домашніми приладами.



Рисунок 1.1 Джим Сазерленд та система "ЕCHO IV"

3. Розвиток у 1980-х:

1980-ті роки стали періодом активного розвитку технологій «розумного будинку». З'явилися перші комерційні системи автоматизації, такі як X10 – стандарт для комунікації між електронними пристроями в будинку, який використовував проводку для передачі сигналів управління. Цей стандарт дозволив керувати освітленням, опаленням та іншими приладами через центральний пульт.

4. Революція 1990-х:

У 1990-х роках технології «розумного будинку» продовжили розвиватися завдяки появі Інтернету. З'явилися перші системи, які могли бути підключені до Інтернету для віддаленого керування. Однією з таких систем стала система "SmartHouse" від компанії IBM, яка пропонувала інтеграцію різних домашніх приладів в одну мережу.

5. Поява сучасних технологій:

2000-ті роки стали періодом бурхливого розвитку бездротових технологій і мобільних пристроїв. З'явилися нові стандарти зв'язку, такі як Zigbee (бездротовий стандарт передачі даних, який формує основну топологію дерева мережі й може зв'язуватися з іншими мережами) та Z-Wave (протокол бездротового зв'язку, який використовується переважно для домашньої

автоматизації), які дозволили створювати більш надійні та енергоефективні системи автоматизації. В цей час з'явилися такі пристрої, як програмовані термостати та системи відеоспостереження, які могли бути інтегровані в єдину систему управління.

6. Ера голосових помічників:

У 2010-х роках з'явилися голосові помічники, такі як Amazon Alexa, Google Assistant та Apple Siri, які відкрили нові можливості для управління «розумним будинком». Тепер користувачі могли керувати освітленням, безпекою та іншими системами за допомогою голосових команд. Голосові помічники значно спростили взаємодію з системами автоматизації та зробили їх більш доступними для широкого кола користувачів.



Рисунок 1.2 Голосовий помічник «Amazon Alexa»

7. Інтернет речей:

На сьогодні технології «розумного будинку» стали невіддільною частиною сучасного життя. Інтернет речей (IoT) дозволив інтегрувати величезну кількість пристроїв у єдину систему, що забезпечує високий рівень комфорту. Сучасні системи можуть самостійно навчатися й адаптуватися до потреб користувачів завдяки штучному інтелекту та машинному навчанню. Саме на цьому етапі люди почали звертати увагу не лише на функціональність програми, але і на зручність інтерфейсу програм для управління системами домашньої автоматизації. Інтерфейс повинен бути інтуїтивно зрозумілим та адаптивним (таким, що пристосовується до різних розмірів екрана, роздільної

здатності та орієнтацій пристроїв), щоб користувачі могли легко взаємодіяти з різними елементами системи. Розробка одного адаптивного інтерфейсу замість кількох окремих версій для різних пристроїв дозволяє економити час і ресурси на розробку та підтримку.

Перспективи розвитку технологій «розумного будинку» пов'язані з подальшим розвитком штучного інтелекту, вдосконаленням бездротових технологій і підвищенням рівня безпеки даних. У майбутньому очікується, що «розумні будинки» стануть ще більш автономними й здатними забезпечувати максимально комфортні умови для життя своїх мешканців.

Аналіз об'єкта дослідження показав, що наявні системи управління «розумним будинком» мають значний потенціал для підвищення комфорту та ефективності використання житлових приміщень. Однак, для досягнення максимального результату необхідно розробити інтерфейс, який буде поєднувати інтуїтивність, налаштованість, мобільність, безпеку та сумісність. Це дозволить створити оптимальну систему керування елементами обладнання внутрішньої інженерії.

1.3 Опис предмету дослідження

В даній дипломній роботі є відразу два предмети дослідження: проєктування та розробка інтерфейсу інформаційної системи.

Проєктування – процес створення плану або концепції для реалізації певної системи. Тобто це комплекс робіт який складається з пошуку, досліджень, розрахунків та розрахування з метою отримання опису достатнього для створення нового об'єкту, а також включає аналіз вимог, розробку концепції, створення моделей та прототипів, тестування та вдосконалення кінцевого продукту.

Під поняттям «система» мається на увазі сукупність взаємодіючих компонентів і зв'язків між ними, які разом утворюють властивості, що відсутні у кожного з цих елементів окремо.



Рисунок 1.3. Концептуальне зображення системи «розумний будинок»

Концептуальне зображення системи «розумний будинок» на рисунку 1.3. демонструє кілька ключових елементів, які забезпечують інтегроване управління різними аспектами житлового приміщення. Ось опис основних елементів, представлених на зображенні:

- Центральний контролер (пристрій на якому встановлена програма з інтерфейсом управління) – дозволяє керувати всіма підключеними компонентами будинку.

- Бездротове з'єднання (Wi-Fi) – забезпечує комунікацію між центральним контролером і всіма підключеними пристроями в будинку.
- Підсистеми керування освітленням, протипожежним захистом, відеоспостереженням і т.д.
- Хмарні сервіси – зберігання даних, резервного копіювання та віддаленого доступу до системи.

Ці елементи разом створюють цілісну систему «розумний будинок».

В контексті інформаційних систем, проєктування охоплює декілька ключових аспектів:

- Аналіз вимог: Визначення потреб та очікувань користувачів, а також технічних вимог до системи. Це включає збір даних через опитування, інтерв'ю, вивчення наявних систем і аналіз ринку.
- Розробка концепції: Створення початкових ідей та концепцій на основі зібраних вимог. Це може включати ескізи, сценарії використання та концептуальні моделі.
- Створення моделей та прототипів: Візуалізація концепції через створення моделей, макетів або прототипів, які дозволяють оцінити функціональність та зручність використання майбутнього продукту.
- Тестування: Перевірка моделей та прототипів на відповідність вимогам користувачів та технічним характеристикам. Тестування може проводитись із залученням кінцевих користувачів для отримання зворотного зв'язку.
- Вдосконалення: Аналіз результатів тестування та внесення необхідних змін для покращення функціональності та зручності використання системи.

Проєктування є комплексним процесом, що містить в собі ряд етапів, кожен з яких спрямований на створення ефективного та зручного продукту.

Розробка інтерфейсу інформаційної системи (ІС) – це процес створення користувацького інтерфейсу, який дозволяє користувачам взаємодіяти з інформаційною системою зручно, ефективно та безпечно. Цей

процес охоплює в себе кілька етапів, які забезпечують створення інтуїтивно зрозумілого, функціонального та привабливого інтерфейсу.

Ефективний користувацький інтерфейс поєднує в собі дві ключові складові: привабливий дизайн та зручність у використанні. Для створення такого інтерфейсу важливо розуміти основи: як користувач взаємодіє з UI, які в нього очікування та що може викликати проблеми.^[1]

Візуально привабливий дизайн може викликати позитивні емоції у користувачів, що сприяє загальному задоволенню від використання продукту. Це може зробити досвід роботи з інтерфейсом більш приємним. Також це демонстрація професіоналізму й уваги до деталей, що може підвищити рівень довіри користувачів до продукту. Звичайно, що такий дизайн допомагає організувати інформацію так, щоб користувачі легко могли знайти потрібні елементи. Чітка візуальна ієрархія та добре продумані візуальні підказки сприяють кращій навігації.

На сучасному ринку, де багато подібних продуктів, красивий дизайн може стати конкурентною перевагою, допомагаючи продукту виділитися серед інших. Адже правильно підібрані елементи, такі як кольори, типографіка, іконки та графіка, можуть покращити користувацький досвід (User experience), якщо вони використані для підкреслення важливих функцій і забезпечення зручності використання.

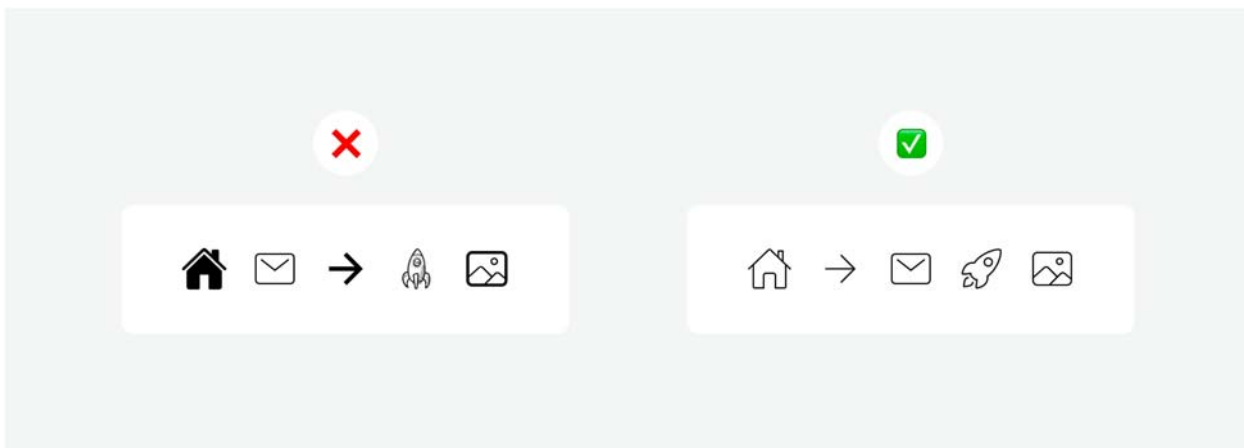


Рисунок 1.4. Приклад правильно підібраних елементів (іконок)

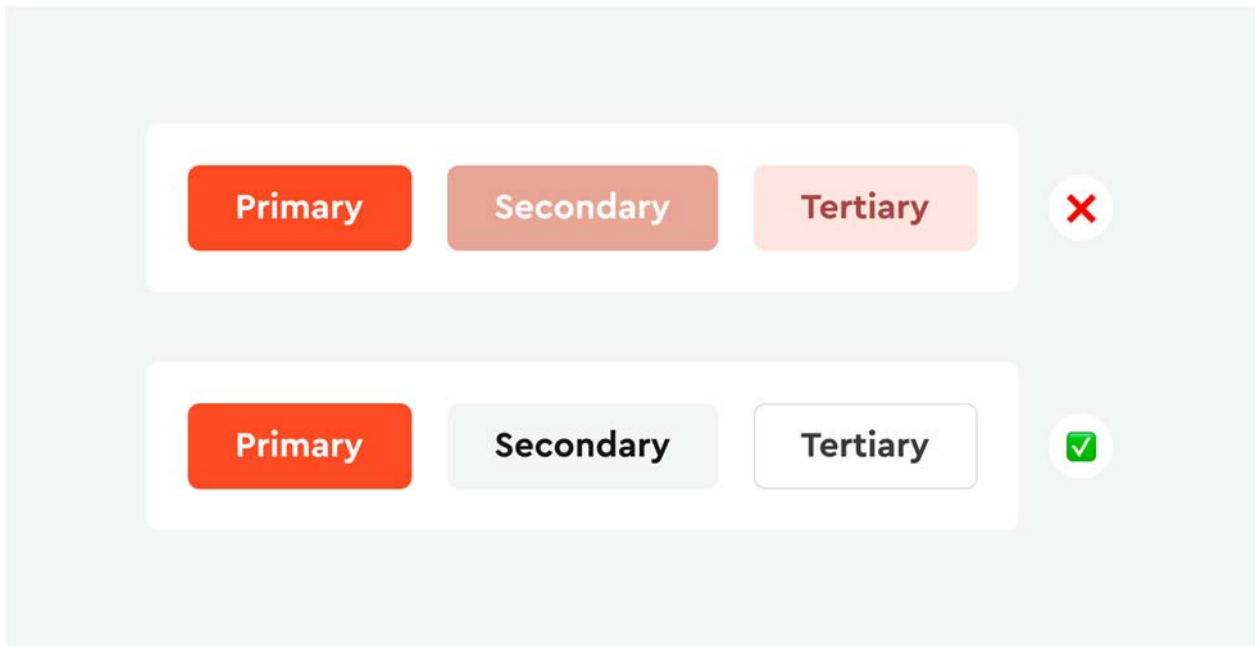


Рисунок 1.5. Приклад правильно підібраних елементів (кнопок)

Якісний дизайн починається з вирішення основних завдань, а потім переходить до складних рівнів. На жаль, у світі продуктів та інтерфейсів існує багато прикладів, де пріоритетом було відрізнятись від інших, але при цьому не були вирішені базові завдання. Насправді інтерфейс користувача об'єднує всі елементи та компоненти програми, які впливають на взаємодію користувача з програмним забезпеченням. Це охоплює сукупність завдань, які користувач вирішує через систему; навігацію між блоками системи; елементи управління; візуальний (і не тільки) дизайн екранів; пристрої та технології введення/виведення даних; засоби відображення інформації; порядок використання програми та документацію до неї.

Користувачі оцінюють програмний засіб не тільки за його функціональні можливості, але й з урахуванням зручності інтерфейсу, ергономічності, відповідності їх вимогам. Ці вимоги можуть бути різними й залежати від виконуваних завдань, уподобань, специфіки представлення інформації в різних предметних галузях, умов та середовища використання, рівня підготовленості користувача тощо.

Інтерфейс користувача грає досить важливу роль в кожній інформаційній системі. Звичайно ідеальних інтерфейсів немає, проте для даної роботи було відділено кілька аспектів добре продуманих інтерфейсів:

1. Адаптивність. Користувач повинен мати можливість легко розрізняти елементи, які мають високий або низький пріоритет. Ієрархічне упорядкування створює можливість для користувача індивідуально аналізувати зміст. Використання однієї колонки у макеті, а не декількох, є доцільнішим, оскільки це забезпечує кращу навігацію читачу, дозволяючи йому легше пройти від початку до кінця. Рекомендується групувати схожий контент за призначенням і об'єднувати стилі оформлення, рамки та інші елементи. Занадто багато стовпців може відвернути користувача від основної мети сторінки.

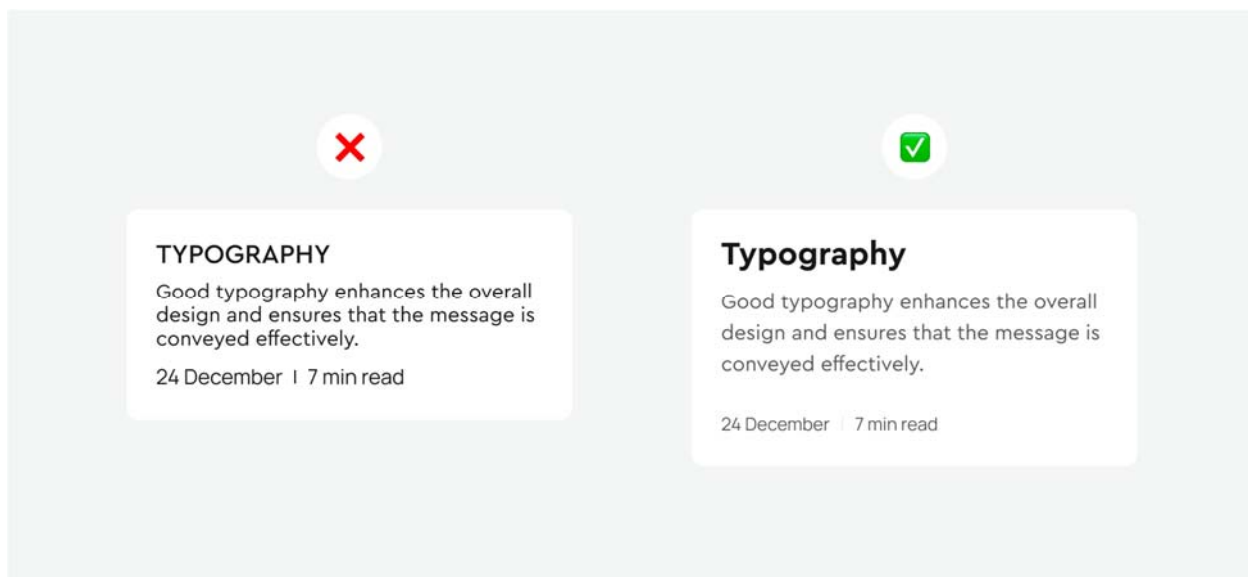


Рисунок 1.6. Приклад ієрархічного упорядкування

2. Юзабіліті (від англ. usability — «використовність») ^[2] — характеристика якості користувацького інтерфейсу, що визначає легкість і ефективність його використання. Це містить оптимізацію робочих процесів і мінімізацію непотрібних дій.
3. Чіткість та лаконічність. Простий і зрозумілий інтерфейс робить програму більш доступною для користувачів будь-якого рівня навичок. Це дозволяє користувачам швидше здійснювати необхідні дії без витрати зайвого часу на пошук потрібних функцій чи інформації. Основна перевага для розробника - мінімізація зайвих складнощів та

деталей в інтерфейсі, що зменшує ймовірність помилок та непорозумінь з боку користувача.

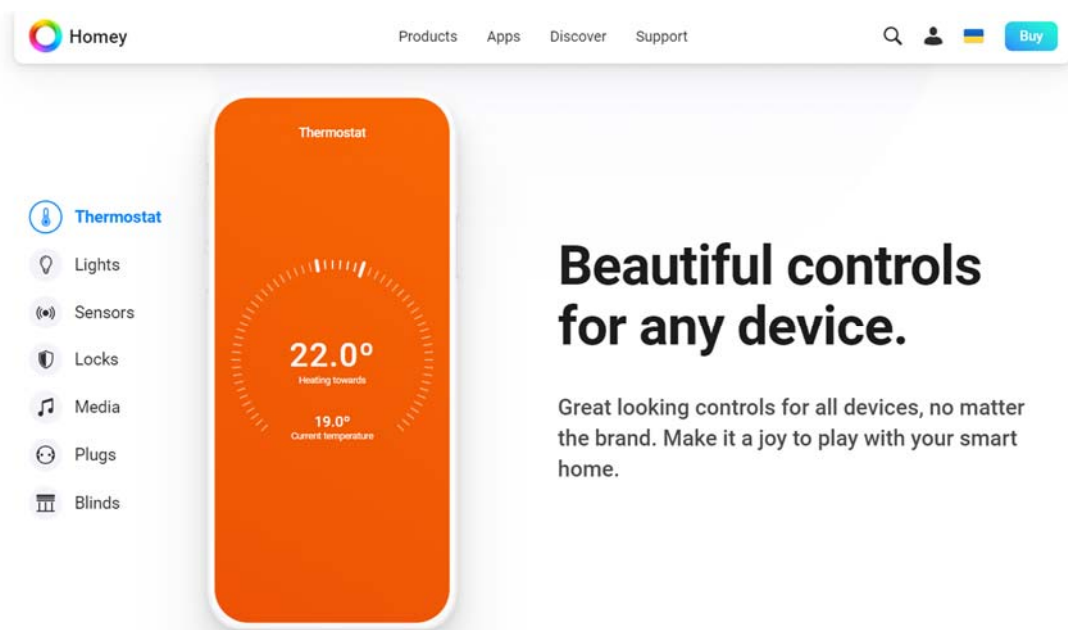


Рисунок 1.7. – Приклад грамотного інтерфейсу

Отже, успішна розробка інтерфейсу передбачає уважне врахування потреб користувачів, створення простого та зрозумілого дизайну, ергономічність, естетичність та постійне вдосконалення на основі зворотного зв'язку від користувачів.

1.4 Аналіз актуальності

Сфера інтелектуальних будівель починалася з програмно-апаратних рішень, що дозволяли керувати окремими пристроями через бездротові протоколи передачі даних. З часом ці системи ставали складнішими й функціональнішими, що призводило до виникнення нових вимог до управління та створювало додаткові виклики для розробників.

«Розумний будинок» – це сучасний засіб для покращення комфорту та якості життя, завдяки автоматизованому управлінню більшістю процесів за допомогою електроніки. Це підкреслює важливість дослідження та вдосконалення таких технологій.

У сучасному світі стрімко зростає тенденція до автоматизації різних процесів, що значно полегшує життя людям. Автоматизація охоплює всі аспекти людського життя: від оптимізації робочих завдань і автоматизації рутинних процесів до управління великими системами, такими як житлові будинки. Підсистеми автоматизації процесів у житлових приміщеннях, або технології розумних будинків, стають все більш популярними у сучасному суспільстві завдяки спрощенню побутових завдань. Для деяких категорій людей ці технології стають абсолютно необхідними для нормального функціонування в їхніх оселях.

Останнім часом серед молодих людей зростає інтерес до розумних будинків, які спрощують життя, автоматизують рутинні задачі й наділяються інтелектуальними можливостями. Водночас все більш популярними є системи розумних будинків для людей з обмеженими фізичними можливостями. Через функціональні обмеження їхнього тіла такі особи часто не можуть самостійно виконувати рутинні побутові задачі, такі як контроль над освітленням, водою або рулонними шторами на вікнах, які зазвичай потребують значної фізичної активності. Щобільше, серед новітніх інновацій існують розумні будинки, які можна керувати голосовими командами, не потрібно натискати кнопки, тримати смартфон або мати доступ до комп'ютера. Це робить життя значно простіше і зручніше.

Швидкий розвиток технологій, зростання інтересу до інтернету речей та збільшення зручності управління побутовими системами робить розумні будинки дедалі популярнішими серед споживачів.

Ось деякі фактори, які підтримують актуальність теми:

1. Зручність і ефективність: Розумні системи дозволяють спростити життя. Вони дозволяють контролювати різні аспекти побуту за допомогою мобільних пристроїв, що робить життя більш ефективним та комфортним.
2. Економія ресурсів: Розумні системи дозволяють зменшити витрати на електроенергію, воду та інші ресурси через оптимізацію їх використання. Наприклад, автоматичне вимикання світла або регулювання температури може значно знизити споживання енергії.
3. Безпека: Розумні системи також допомагають підвищити рівень безпеки в будинку. Вони можуть включати системи відеоспостереження, датчики пожежі та системи безпеки, які автоматично сповіщають власників про потенційні загрози.
4. Доступність для різних груп населення: Розумні системи стають особливо корисними для людей з обмеженими можливостями, які можуть мати складнощі зі здійсненням певних побутових завдань. Такі системи дозволяють їм жити більш незалежним життям.

Загалом, з розвитком технологій і збільшенням свідомості про зручність та ефективність розумних систем, ця тема залишається дуже актуальною і продовжує здобувати популярність у суспільстві.

1.5 Стан вже існуючих рішень

Розумні будинки перетворюються на стиль сучасного життя, де технологія зручні інтерфейси стають невід'ємною частиною нашого повсякденного існування.

Сучасні інтерфейси програм для керування розумними будинками являють собою складні, інтегровані системи, що забезпечують користувачів можливістю легко та ефективно керувати різними аспектами свого житла. Ці інтерфейси постійно вдосконалюються для підвищення зручності, функціональності та безпеки. Розглянемо основні рішення, які вже існують на ринку, і їхні ключові особливості.

Інтерфейси для керування розумними будинками зазвичай працюють через браузер і надають користувачам можливість доступу до системи з будь-якого пристрою, підключеного до Інтернету. Вони забезпечують високу гнучкість і не потребують установки додаткового програмного забезпечення. Основні характеристики інтерфейсів включають:

- 1) Кросплатформеність – це здатність програмного забезпечення працювати на різних платформах і пристроях без необхідності значної модифікації. У контексті інтерфейсів для керування розумними будинками це означає, що користувачі можуть отримати доступ до системи за допомогою будь-якого пристрою, який має веб-браузер.
- 2) Дистанційний доступ – це можливість керувати системою розумного будинку з будь-якої точки світу за допомогою Інтернету.
- 3) Адаптивний дизайн

Адаптивний дизайн (англ. Responsive Design) – це підхід до створення інтерфейсів, які автоматично підлаштовуються під різні розміри екранів і характеристики пристроїв. Це досягається шляхом використання гнучких макетів, зображень і CSS медіа-запитів. ^[3]

Кросплатформеність, дистанційний доступ та адаптивний дизайн є ключовими аспектами сучасних інтерфейсів для систем розумних будинків. Вони забезпечують зручність, універсальність та ефективність, що дозволяє користувачам легко керувати своїм житлом, незалежно від їхнього місця знаходження та використовуваного пристрою.

Пропонуємо ознайомитися з прикладами існуючих інтерфейсів програм, щоб детально зрозуміти переваги та недоліки сучасних систем розумного будинку та в подальшому внести корективи до власних проєктних рішень.

Перший приклад – інтерфейс програми від компанії Pergenio (міжнародна технологічна компанія, що спеціалізується на розробці та впровадженні рішень для розумного будинку та офісу).

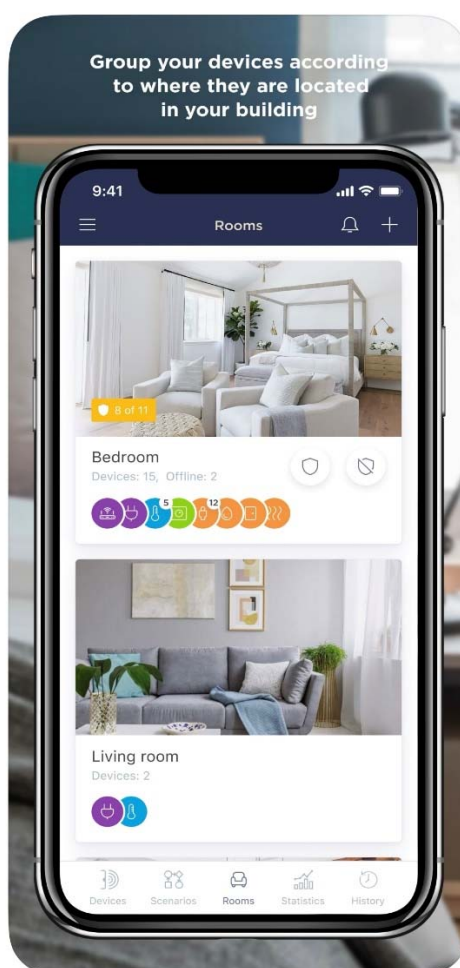


Рисунок 1.8. – Приклад інтерфейсу від компанії Pergenio

Нижче наведено детальний аналіз основних переваг та недоліків представленого інтерфейсу.

Основні переваги:

- 1) Зручна навігація: Візуальні іконки та зображення кімнат забезпечують інтуїтивно зрозумілу навігацію.
- 2) Інформаційна насиченість: Користувач може одразу бачити кількість підключених пристроїв та їх статус (онлайн/офлайн).
- 3) Використання кольорових іконок для різних типів пристроїв допомагає швидко зрозуміти, які саме пристрої встановлені в кімнаті.

Вагомі недоліки:

- 1) Відображення великої кількості інформації на одному екрані може бути складним для нових користувачів.
- 2) Іконки та текст можуть здаватися дрібними на екрані смартфона, що ускладнює їх сприйняття.

Наступним прикладом – система розумного будинку від компанії MSmartHome (технічна компанія, яка займається розробкою системи розумного будинку для забезпечення ефективного управління всіма аспектами домашнього життя.).

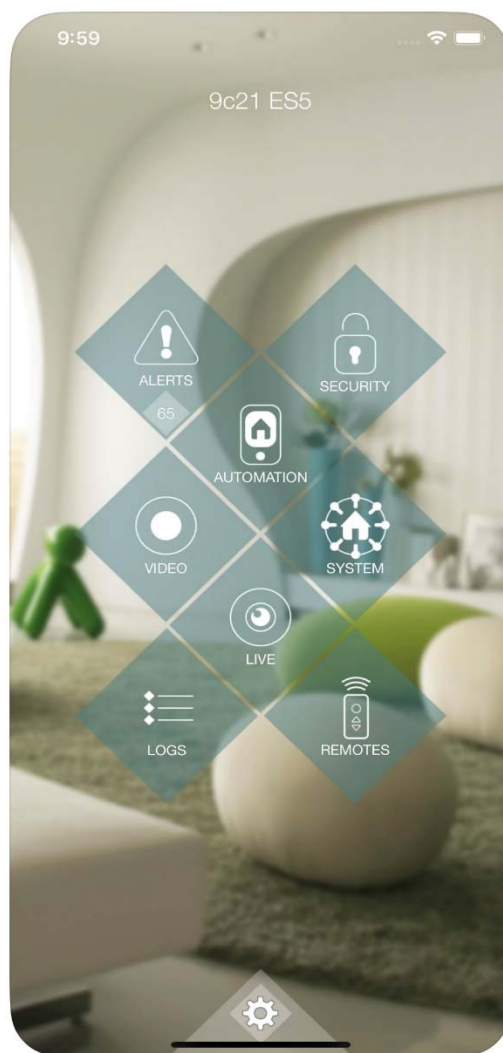


Рисунок 1.9. – Приклад інтерфейсу від компанії Rosohrana

Переваги даного інтерфейсу:

- 1) Зрозуміле розташування іконок: Виглядає організовано та зручно для швидкого доступу до різних функцій.
- 2) Іконки з підписами: Кожна іконка має підпис, що допомагає користувачам швидко зрозуміти, яка функція за що відповідає.
- 3) Інтуїтивно зрозумілий дизайн: Зрозумілі іконки, такі як "безпека", "відео" та "автоматизація", дозволяють користувачам легко орієнтуватися в додатку.
- 4) Централізоване керування: Додаток забезпечує доступ до різних систем будинку (безпека, відео, автоматизація) з одного екрану, що зручно для користувачів.

Основні недоліку інтерфейсу:

- 1) Відсутність текстового меню: Виключно іконки можуть бути не настільки інформативними для деяких користувачів, особливо для тих, хто вперше використовує додаток.
- 2) Недостатня контрастність: Деякі користувачі можуть мати проблеми з баченням іконок або тексту через недостатню контрастність між фоном і елементами інтерфейсу.
- 3) Незрозумілі індикатори стану: В інтерфейсі немає чітких індикаторів стану (наприклад, чи активована система безпеки), що може спричинити плутанину для користувачів.

Розробляючи інтерфейс власної програми, важливо враховувати вищеперелічені переваги та недоліки прикладів інтерфейсів програм від технічних компаній.

1.6 Визначення цілей дослідження та постановка задачі

Мета дослідження роботи на тему полягає в створенні ефективного, інтуїтивно зрозумілого та надійного інтерфейсу для управління різними елементами внутрішньої інженерії розумного будинку та проєктування даної системи. Це дозволить користувачам легко контролювати та налаштовувати системи освітлення, протипожежного захисту та відеоспостереження, забезпечуючи зручність використання та високу функціональність.

Для чіткого візуального представлення створене дерево цілей, яке дозволяє краще зрозуміти зв'язки між різними етапами та компонентами роботи, а також наочно побачити, як кожне завдання сприяє досягненню загальної мети. На рисунку 1.8 представлений приклад дерева цілей.

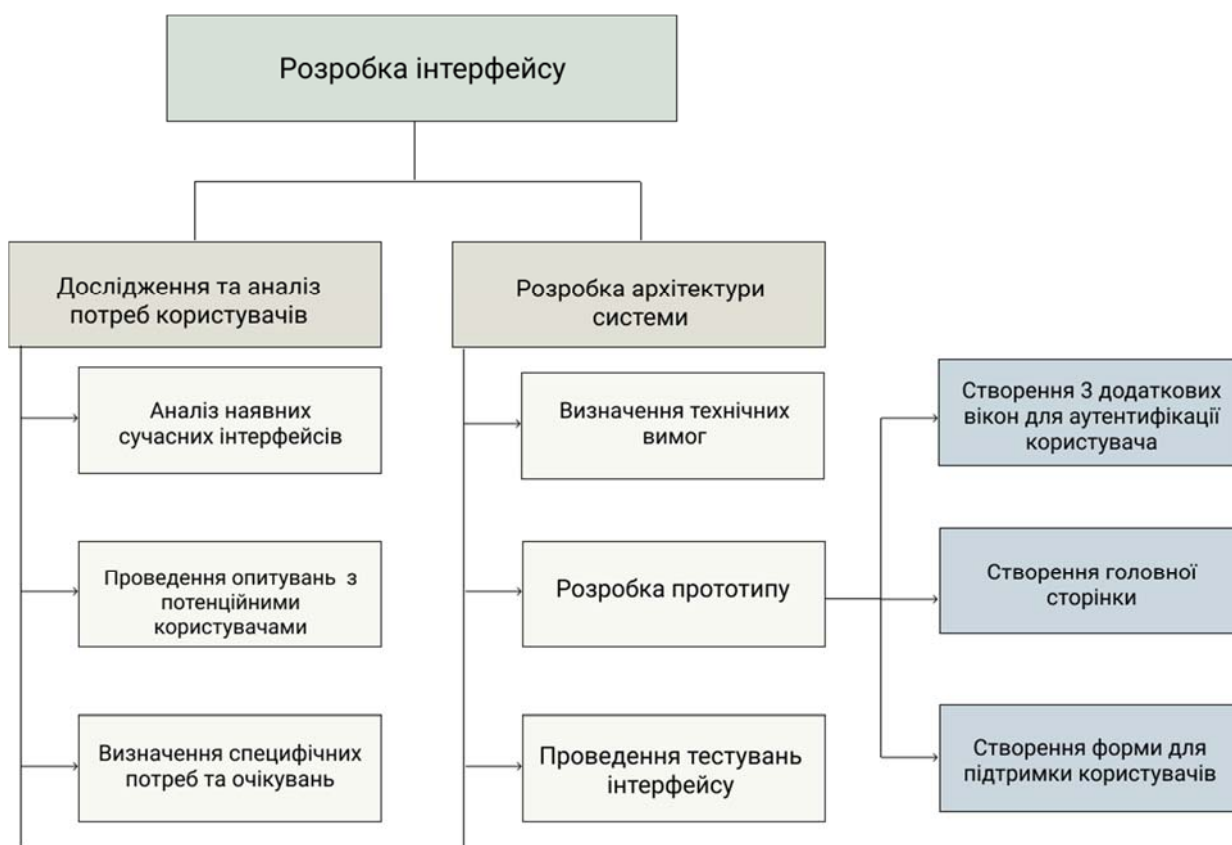


Рисунок 1.8. – Дерево основних розробки інтерфейсу програми

Для реалізації загальної мети розділимо дерево на дві загальні задачі – «Дослідження та аналіз потреб користувачів» й «Розробка архітектури системи».

Завдання «Дослідження та аналіз потреб користувачів» містить:

- Аналіз наявних сучасних інтерфейсів: Вивчення існуючих рішень на ринку, їх переваг та недоліків, щоб зрозуміти, які аспекти можна впровадити або вдосконалити.
- Проведення опитувань з потенційними користувачами: Збір даних від майбутніх користувачів щодо їхніх уподобань, проблем та побажань. Це дозволяє краще зрозуміти, що саме їм потрібно від інтерфейсу.
- Визначення специфічних потреб та очікувань: Аналіз отриманої інформації для визначення конкретних вимог користувачів до інтерфейсу. Це може включати специфічні побажання щодо дизайну, зручності використання тощо.

Завдання «Розробка архітектури системи» містить:

- Визначення технічних вимог: Формулювання вимог до системи на основі зібраних даних.
- Розробка прототипу: Створення початкової версії інтерфейсу, яка демонструє основні функції та структуру. Прототип дозволяє оцінити, як система буде виглядати та працювати. Цей етап розбито на 3 основних підетапи, а саме:
 - 1) Створення 3 додаткових вікон для аутентифікації користувача: Розробка додаткових інтерфейсних елементів для забезпечення безпеки доступу.
 - 2) Створення головної сторінки: Розробка основного екрану інтерфейсу, з якого користувач буде взаємодіяти з усіма функціями розумного будинку.
 - 3) Створення форми для підтримки користувачів: Розробка інтерфейсних елементів для забезпечення зворотного зв'язку та підтримки користувачів.

- Проведення тестувань інтерфейсу: Тестування прототипу з користувачами для отримання зворотного зв'язку. Це допомагає виявити недоліки та можливі покращення перед фінальною розробкою.

Об'єктом дослідження роботи є інформаційна система для керування елементами обладнання внутрішньої інженерії "розумного будинку".

Предметом дослідження є процеси та методи проектування і розробки інтерфейсу користувача для цієї інформаційної системи.

Як можемо проаналізувати, в даній сфері вже є дослідження з вагомими результатами, проте тема все ще має багато напрямків для розвитку – зокрема, удосконалення зручності та інтуїтивності інтерфейсу. Подальший розвиток системи дозволить більш ефективно керувати елементами обладнання розумного будинку.

2. АЛГОРИТМІЧНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Математичне забезпечення системи

Математичні моделі є основою для розуміння та аналізу складних систем, таких як розумний будинок. Вони дозволяють формалізувати поведінку системи та прогнозувати її реакцію на різні події.

В даній роботі особливу увагу приділено функції, яка імітує датчик освітлення, визначає потужність освітлення залежно від часу доби та наявного природного освітлення. Ця функція враховує зміни природного освітлення протягом доби і коригує інтенсивність штучного освітлення в кімнаті для забезпечення комфортного рівня освітленості.

Природне освітлення залежить від часу доби t . Для спрощення розрахунків використаємо синусоїдальну функцію для моделювання зміни освітленості протягом доби:

$$E_{\text{нат}}(t) = E_{\text{max}} \cdot \sin\left(\frac{\pi t}{T}\right)$$

Де:

- $E_{\text{нат}}(t)$ – освітленість від природнього світла в даний момент часу t .
- E_{max} – максимальна природна освітленість.
- T – тривалість світлового дня (в годинах).

Освітленість в кімнаті $E_{\text{кім}}$ залежить як від природного, так і від штучного освітлення. Завдання полягає в тому, щоб за наявного природного освітлення забезпечити заданий рівень загальної освітленості $E_{\text{заг}}$:

$$E_{\text{кім}} = E_{\text{нат}}(t) + E_{\text{шт}}$$

Де:

- $E_{\text{шт}}$ – освітленість від штучного освітлення, яку необхідно налаштувати.

Для забезпечення комфортного рівня освітленості в кімнаті $E_{\text{заг}}$, необхідно врахувати поточну природну освітленість $E_{\text{нат}}(t)$ і відповідним чином налаштувати штучне освітлення $E_{\text{шт}}$:

$$E_{\text{шт}} = E_{\text{заг}} - E_{\text{нат}}(t)$$

При цьому, якщо $E_{\text{нат}}(t) \geq E_{\text{заг}}$, то штучне освітлення не потрібне і $E_{\text{шт}} = 0$

Для автоматичного налаштування освітлення система повинна виконувати наступні кроки:

- 1) Вимірювання природного освітлення. За допомогою датчика визначається поточний рівень природного освітлення $E_{\text{нат}}(t)$.
- 2) Визначення необхідної освітленості. На основі заданого рівня загальної освітленості $E_{\text{заг}}$ і виміряного значення $E_{\text{нат}}(t)$ обчислюється необхідний рівень штучного освітлення $E_{\text{шт}}$:

$$E_{\text{шт}} = \max(0, E_{\text{заг}} - E_{\text{нат}}(t))$$

- 3) Відповідно до обчисленого значення $E_{\text{шт}}$, система налаштовує потужність ламп освітлення. Потужність освітлення P може бути визначена пропорційно до необхідного рівня освітленості:

$$P = \frac{E_{\text{шт}}}{E_{\text{шт_макс}}} \cdot P_{\text{макс}}$$

Де:

- $E_{\text{шт_макс}}$ - максимальний рівень штучного освітлення, який може забезпечити лампа.
- $P_{\text{макс}}$ - максимальна потужність лампи.

Для кращого розуміння надаємо приклад розрахунку.

Припустимо:

- Максимальна природна освітленість: $E_{\text{max}} = 1000$ люкс
- Загальний рівень освітленості: $E_{\text{заг}} = 500$ люкс
- Тривалість світлового дня $T = 12$ годин
- Максимальний рівень штучного освітлення: $E_{\text{шт_макс}} = 500$ люкс
- Максимальна потужність лампи $P_{\text{макс}} = 100$ Вт

Для часу $t = 8$ годин (зранку):

$$E_{\text{нат}}(8) = 1000 \cdot \sin\left(\frac{\pi \cdot 8}{12}\right) = 1000 \cdot \sin\left(\frac{2\pi \cdot 8}{24}\right) = 1000 \cdot \sin\left(\frac{2\pi}{3}\right) \approx 866 \text{ люкс}$$

$$E_{\text{шт}} = \max(0, 500 - 866) = 0 \text{ люкс}$$

Таким чином, штучне освітлення не потрібно.

Для часу $t = 18$ годин (вечір):

$$E_{\text{нат}}(18) = 1000 \cdot \sin\left(\frac{\pi \cdot 18}{12}\right) = 1000 \cdot \sin\left(\frac{3\pi}{2}\right) = 1000 \cdot (-1) = -1000 \text{ люкс}$$

(оскільки синусоїдальна функція набуває негативних значень, враховуємо тільки позитивні освітленості, тому $E_{\text{нат}}(18) = 0$)

$$E_{\text{шт}} = \max(0, 500 - 0) = 500 \text{ люкс}$$

$$P = \frac{500}{500} \cdot 100 = 100 \text{ Вт}$$

Отже, ввечері для досягнення рівня освітленості $E_{\text{заг}} = 500$ люкс необхідно увімкнути лампі на повну потужність $P_{\text{макс}} = 100$ Вт

Розроблене математичне забезпечення дозволяє автоматично регулювати рівень штучного освітлення залежно від часу доби та наявного природного освітлення, забезпечуючи комфортні умови для користувачів розумного будинку.

2.2 Опис проєктних рішень з математичного забезпечення

В даній дипломній роботі математичне забезпечення системи спрямоване саме на реалізацію алгоритмів, які забезпечують автоматичне регулювання освітлення в приміщенні відповідно до змін рівня природного освітлення протягом доби. Основним завданням є підтримка комфортного рівня освітленості для користувачів, що досягається шляхом динамічного коригування потужності штучного освітлення.

Для реалізації даного функціоналу були визначені такі константи:

1) `const recommendedLux = 300;`

`recommendedLux`: Рекомендований рівень загальної освітленості в приміщенні, встановлений на 300 люкс. Цей рівень забезпечує комфортні умови для перебування та виконання різних видів діяльності.

2) `const maxArtificialLux = 500;`

`maxArtificialLux`: Максимальний рівень освітленості, який може бути забезпечений штучним освітленням, встановлений на 500 люкс. Ця величина визначає граничні можливості системи штучного освітлення.

```
function getCurrentNaturalLightLevel() {
  const now = new Date();
  const hour = now.getHours();
  let naturalLightLevel;

  if (hour >= 6 && hour < 12) {
    // Morning: 6 AM - 12 PM
    naturalLightLevel = 200;
  } else if (hour >= 12 && hour < 18) {
    // Afternoon: 12 PM - 6 PM
    naturalLightLevel = 500;
  } else if (hour >= 18 && hour < 22) {
    // Evening: 6 PM - 10 PM
    naturalLightLevel = 100;
  } else {
    // Night: 10 PM - 6 AM
    naturalLightLevel = 10;
  }

  return naturalLightLevel;
}
```

Рисунок 2.1. – Код, який описує функцію `getCurrentNaturalLightLevel`

В коді прописана функція `getCurrentNaturalLightLevel`, що обчислює поточний рівень природного освітлення на основі часу доби. Ця функція використовує об'єкт `Date` для отримання поточного часу та визначає рівень природного освітлення на основі годин:

- Ранок (6 AM - 12 PM): Освітленість становить 200 люкс.
- День (12 PM - 6 PM): Освітленість становить 500 люкс.
- Вечір (6 PM - 10 PM): Освітленість становить 100 люкс.
- Ніч (10 PM - 6 AM): Освітленість становить 10 люкс.

Розрахунок необхідної потужності штучного освітлення здійснюється наступним чином:

```

if (requiredLuxFromArtificialLight <= 0) {
    lightingPowerPercentage = 0;
} else if (requiredLuxFromArtificialLight >= maxArtificialLux) {
    lightingPowerPercentage = 100;
} else {
    lightingPowerPercentage = (requiredLuxFromArtificialLight / maxArtificialLux) * 100;
}

```

Рисунок 2.2. – Код, який описує потужність освітлення

Функція `requiredLuxFromArtificialLight` розраховує кількість люксів, яку повинне забезпечити штучне освітлення. Визначається як різниця між `recommendedLux` та поточним рівнем природного освітлення (визначеним за допомогою функції `getCurrentNaturalLightLevel`).

Якщо `requiredLuxFromArtificialLight` менше або дорівнює 0, потужність штучного освітлення встановлюється на 0% (освітлення не потрібне).

Якщо `requiredLuxFromArtificialLight` більше або дорівнює `maxArtificialLux`, потужність штучного освітлення встановлюється на 100% (максимальне освітлення).

В іншому випадку, потужність штучного освітлення обчислюється як частка від `maxArtificialLux`, помножена на 100 для отримання відсоткового значення.

Загальний алгоритм роботи:

- 1) Вимірювання природного освітлення. Викликається функція `getCurrentNaturalLightLevel`, яка на основі поточного часу доби повертає значення природного освітлення.
- 2) Обчислення необхідного рівня штучного освітлення. Визначається кількість люксів, яку потрібно забезпечити за допомогою штучного освітлення, віднімаючи поточний рівень природного освітлення від рекомендованого рівня `recommendedLux`.
- 3) Встановлення потужності штучного освітлення. На основі розрахованого значення `requiredLuxFromArtificialLight`, встановлюється відповідний відсоток потужності для ламп штучного освітлення.

Приклад реалізації:

Розглянемо приклад реалізації системи для різних часових інтервалів:

Для часу $t = 8$ годин (ранок):

- Поточний рівень природного освітлення: 200 люкс.
- Необхідний рівень штучного освітлення: $300 - 200 = 100$ люкс.
- Відсоток потужності освітлення: $\frac{100}{500} \cdot 100 = 20\%$

Для часу $t = 18$ годин (вечір):

- Поточний рівень природного освітлення: 100 люкс.
- Необхідний рівень штучного освітлення: $300 - 100 = 200$ люкс.
- Відсоток потужності освітлення: $\frac{200}{500} \cdot 100 = 40\%$

Запропоновані математичні моделі та алгоритми забезпечують автоматичне налаштування рівня штучного освітлення залежно від природного освітлення, що змінюється протягом доби. Це дозволяє підтримувати комфортний рівень освітленості в приміщенні, економити енергію та підвищувати ефективність використання системи розумного будинку.

2.3 Проєктування системи

Для проєктування інтерфейсу інформаційної системи керування елементами обладнання внутрішньої інженерії розумного будинку було вибрано методологію проєктування – User-Centered Design (UCD).

Методологія User-Centered Design (UCD) зосереджується на потребах, бажаннях і обмеженнях кінцевих користувачів на кожному етапі процесу проєктування. Цей підхід має ряд переваг, що роблять його ефективним і корисним для створення інтерфейсів, орієнтованих на користувачів. Нижче розглянемо детальніше основні переваги UCD.

- 1) Підвищення задоволеності користувачів. Завдяки постійному залученню користувачів на різних етапах проєктування забезпечується висока задоволеність кінцевих користувачів продуктом. Це досягається через глибоке розуміння користувачів, а саме аналіз поведінки, мотивацій та потреб користувачів, та ітеративне тестування прототипів із залученням користувачів, що дозволяє вчасно виявляти та усувати проблеми.
- 2) Зменшення ризиків. Метод UCD допомагає зменшити ризики, пов'язані з розробкою продукту, шляхом раннього виявлення потенційних проблем та перевірки рішень на практиці. Це включає: раннє виявлення помилок та перевірка припущень. Тестування прототипів допомагає перевірити гіпотези про поведінку користувачів, що зменшує ризик невідповідності очікуванням користувачів.
- 3) Підвищення ефективності та продуктивності. Інтерфейси, розроблені з використанням UCD, зазвичай більш інтуїтивні та зручні, що підвищує ефективність і продуктивність користувачів. Адже дизайн, орієнтований на користувачів, забезпечує легкість і зрозумілість використання продукту. Інтуїтивний інтерфейс скорочує час, необхідний для навчання користувачів роботі з продуктом.

Як можемо переконатися щодо обраної методології User-Centered Design забезпечує численні переваги. Використання UCD допомагає створювати інтерфейси, що відповідають потребам користувачів, що є ключовим фактором успішності сучасних програмних продуктів.

Розглянемо детальніше основні етапи та методи UCD, які були використані в дипломній роботі.

1. Дослідження користувачів (User Research)

На цьому етапі проводяться різноманітні дослідження для розуміння потреб та поведінки користувачів. Основні методи включають:

- 1) Аналіз конкурентів. Вивчення наявних рішень на ринку для розуміння їхніх переваг та недоліків, що було розписано більш детально в підрозділі 1.6.
- 2) Інтерв'ю з користувачами. Глибокі розмови з потенційними користувачами для виявлення їхніх потреб, бажань та проблем.

Комплексну систему розумного будинку розбили на три підсистеми, щоб більш точно виділити потреби та побажання потенційних користувачів.

Підсистема загального та аварійного освітлення:

- Забезпечення швидкого доступу до функцій аварійного освітлення з можливістю вручну ввімкнути/вимкнути аварійне освітлення у випадку надзвичайних ситуацій.
- Інтуїтивно зрозумілий дизайн, який не потребує спеціальних навичок для управління освітленням. Користувачі повинні мати можливість швидко зрозуміти, як користуватися інтерфейсом.
- Висока швидкість реакції на команди користувача та надійна робота без збоїв. Інтерфейс повинен бути стабільним та працювати без затримок.
- Наявність функції, яка визначає інтенсивність світла залежно від часу доби та наявного природного освітлення.

Підсистема відеоспостереження:

- Можливість увімкнути/вимкнути датчики диму відразу у всіх секціях будинку одразу.

- Можливість спостереження за рівнем диму у будинку та виявлення пожежі чи протікання газу.

Підсистема відеоспостереження:

- Підтримка адаптивного дизайну для коректного відображення інтерфейсу на різних пристроях (комп'ютерах, планшетах, смартфонах).
- Відображення відеопотоків у реальному часі з мінімальною затримкою.
- Можливість самостійно вимикати у разі потреби.

2. Аналіз вимог (Requirements Analysis)

Після збору даних проводиться їх аналіз для визначення основних вимог до продукту. Було виявлено, що вимоги поділяються на дві основні категорії: функціональні та нефункціональні.

Функціональні вимоги для підсистеми освітлення:

1) Вмикання та вимикання освітлення:

- General Light Switch: Кожна кімната має можливість вмикати та вимикати загальне освітлення за допомогою перемикача.
- Emergency Light Switch: Додаткова можливість вмикати та вимикати аварійне освітлення.

2) Регулювання інтенсивності освітлення. Інтерфейс дозволяє користувачам використовувати рекомендовану інтенсивність освітлення, яка автоматично підлаштовується під умови освітлення в кімнаті (наприклад, 58%).

3) Кімнатно-орієнтоване управління. Інтерфейс має окремі контролі для різних кімнат (кухня, кімната, ванна кімната, коридор), що дозволяє налаштовувати освітлення індивідуально для кожного приміщення.

Функціональні вимоги для підсистеми протипожежного захисту:

1) Інтерфейс повинен показувати роботу датчиків диму та температури для виявлення ознак пожежі.

- 2) Інтерфейс повинен містити графік, який в реальному часі показує рівень диму та температури, що дозволяє користувачам відстежувати стан приміщень.
- 3) Перемикач (ALL ROOMS), який дозволяє вмикати або вимикати систему протипожежного захисту для всіх приміщень одночасно.

Функціональні вимоги для підсистеми відеоспостереження:

- 1) Інтерфейс відображає відео з камери в реальному часі, що дозволяє користувачам бачити поточний стан в коридорі. Це важливо для оперативного моніторингу та прийняття рішень.
- 2) Система повинна мати функцію обмеження доступу до відеоспостереження для захисту приватності користувачів.

Нефункціональні вимоги для всіх трьох підсистем одразу:

- 1) Інтерфейс повинен бути зрозумілим та легким у використанні, забезпечуючи простий доступ до всіх функцій.
- 2) Дизайн інтерфейсу має бути естетично привабливим, що сприяє кращому користувацькому досвіду.
- 3) Система повинна працювати безперебійно, забезпечуючи постійний доступ до керування освітленням.
- 4) Інтерфейс повинен швидко реагувати на команди користувача без затримок.
- 5) Інтерфейс повинен забезпечувати можливість обмеження доступу до управління освітленням для уникнення несанкціонованого використання.
- 6) Інтерфейс має бути доступним на різних пристроях (смартфонах, планшетах, комп'ютерах) і сумісним з різними операційними системами.
- 7) Інтерфейс має підтримувати можливість оновлення та додавання нових функцій у майбутньому.

3. Розробка дизайну (Design)

Етап розробки дизайну є одним з ключових етапів у процесі створення інтерфейсу користувача для будь-якої системи. На цьому етапі створюються ескізи, каркаси (wireframes) та прототипи інтерфейсу ^[4]. Основні методи включають:

1) Створення концептуального дизайну

На початковому етапі розробки дизайну формується загальна концепція інтерфейсу. Тобто створюються перші ескізи та низькодеталізовані прототипи, які дозволяють побачити загальне розташування елементів на екрані та їх взаємодію.^[6]

Для візуального уявлення концептуального дизайну представлено один з блоків інтерфейсу на рисунку 2.3, де зображено ескізи трьох підсистем, кожна з яких має однакову структуру для зручності користування.

Структура інтерфейсу:

Заголовок (TITLE 1): Загальний заголовок, який містить загальну назву системи розумного будинку.

Зображення: Великий блок, зарезервований для візуального представлення підсистем.

Підзаголовки (TITLE 2): Менший заголовок під зображенням, який відповідає за назву конкретної підсистеми.

Текстовий блок: Блок тексту, що містить детальну інформацію про підсистему.

Кнопка (button): Інтерактивний елемент для взаємодії з підсистемою (наприклад, для увімкнення/вимкнення функцій або налаштування параметрів).

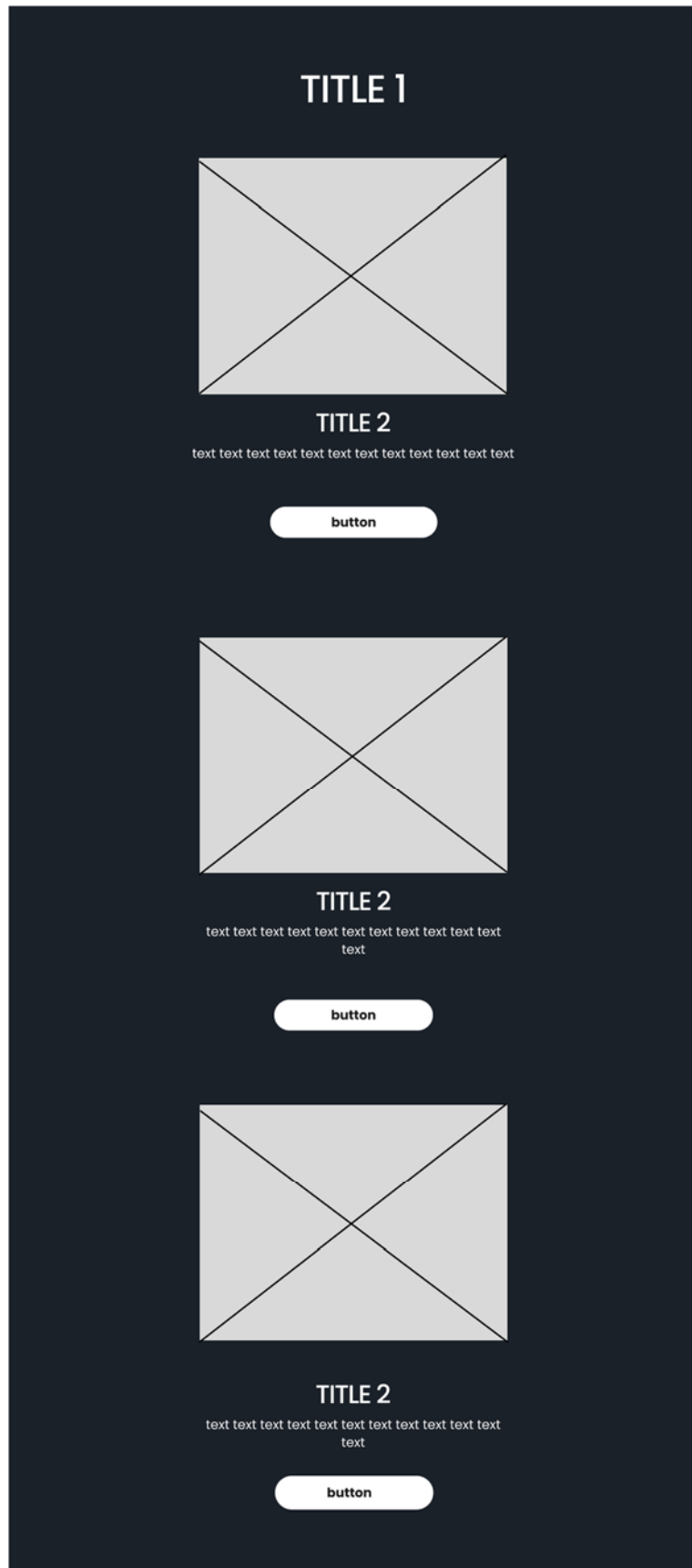


Рисунок 2.3 – Концептуальний дизайн одного із блоків інтерфейсу програми

2) Створення інформаційної архітектури.

На цьому етапі визначається структура та організація контенту. Визначаються основні секції та підсекції інтерфейсу, створюється логічна і зручна структура навігації. Створюється карта сайту, яка відображає всі сторінки та їх взаємозв'язки.

На рисунку 2.4 продемонстровано підсекцію на якій розміщено кнопку, якщо ми натискаємо на неї, то нас перекидає у відповідну секцію в блоці сторінки.

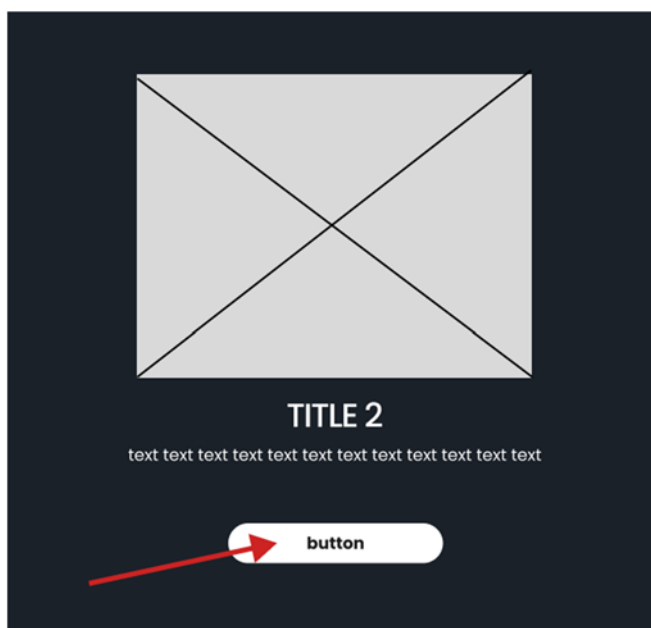


Рисунок 2.4 – Зображення кнопки в підсекції, де можна перейти на конкретну секцію

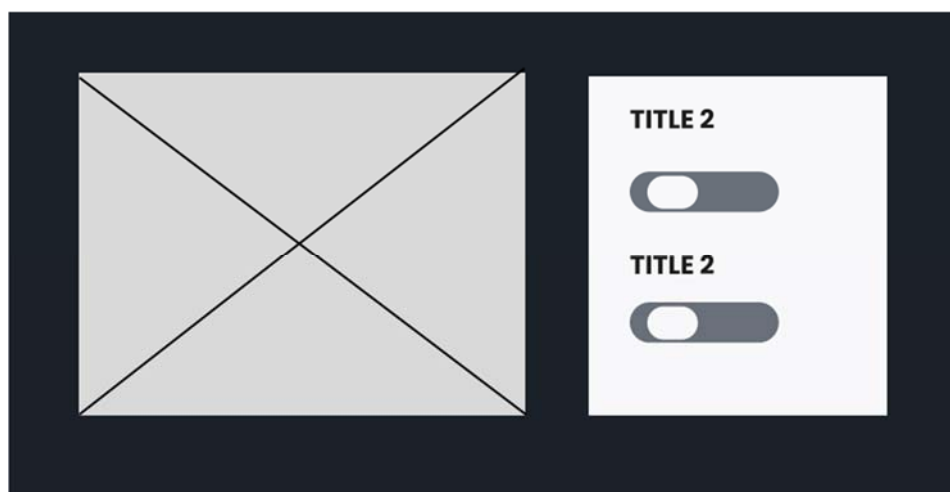


Рисунок 2.5 – Секція, на яку перекидає кнопка з рис. 2.4

3) Створення детального дизайну інтерфейсу.

Цей етап включає розробку детальних макетів інтерфейсу. Створюються високодеталізовані прототипи, які показують точний зовнішній вигляд інтерфейсу, включаючи колірну гаму, типографіку, іконки та інші елементи дизайну. Створюються всі необхідні дизайн графічних елементів, такі як кнопки, іконки, фони, логотип, які будуть використовуватися в інтерфейсі.



Рисунок 2.6 – Створення необхідних графічних елементів

4) Визначення інтерактивних елементів та анімацій.

Цей етап спрямований на створення інтерактивних елементів та анімацій, які покращують користувацький досвід. Визначаються інтерактивні елементи, такі як кнопки, форми, меню, та їх поведінка при взаємодії з користувачем (натискання, наведення, введення даних). Створюються анімації, які допомагають користувачам орієнтуватися в інтерфейсі та покращують візуальне сприйняття (наприклад, плавні переходи між екранами, анімація завантаження).

На ристунку 2.7 продемонстровано взаємодію користувача з кнопкою «ALL ROOMS», яка вмикає датчик диму для визначення рівня диму в приміщенні.

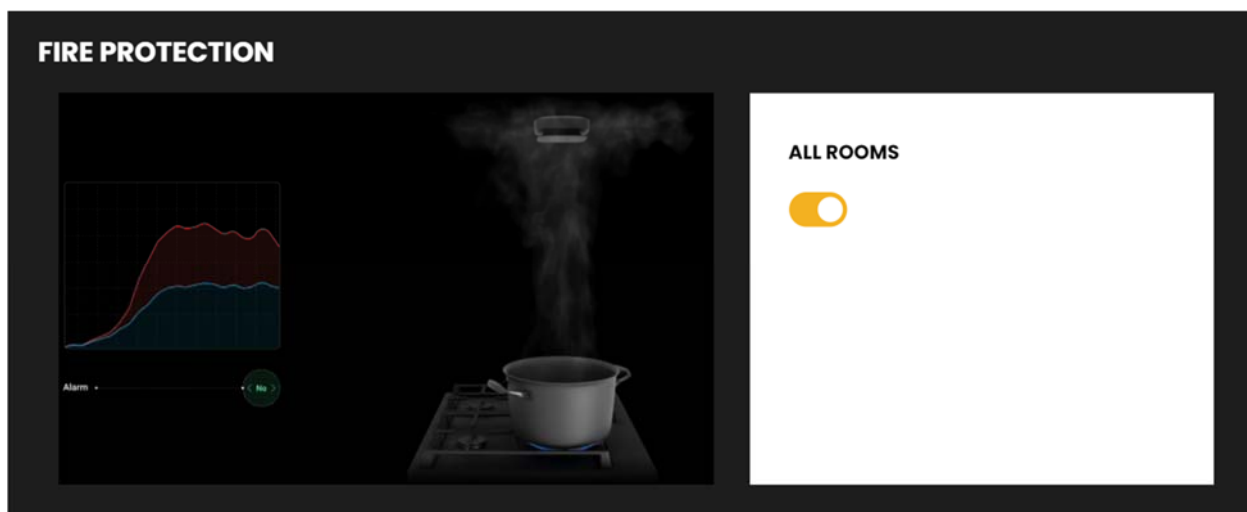


Рисунок 2.7 – Взаємодія користувача з кнопкою «ALL ROOMS»

5) Внесення коригувань та фіналізація дизайну.

Остаточний етап включає внесення змін на основі зібраного зворотного зв'язку. Аналізуються результати користувацького тестування, виявляються проблемні місця. Вносяться необхідні зміни та покращення в дизайн та функціонал інтерфейсу.

Етап проєктування дизайну та функціоналу є критично важливим для створення інтерфейсу, який буде зручним, інтуїтивно зрозумілим та привабливим для користувачів. Містить кілька підетапів, від створення концептуальних ескізів до фіналізації дизайну після користувацького тестування. Цей процес дозволяє створити інтерфейс, який максимально відповідає потребам користувачів та забезпечує позитивний досвід взаємодії з системою.

2.4 Алгоритмічне забезпечення програми

В даному розділі розглянуто алгоритмічні рішення, які реалізовані в програмному коді для забезпечення роботи інформаційної системи.

Перш за все реалізовано функції відкриття та закриття бокового меню. Меню відкривається при натисканні на значок, закривається при натисканні на кнопку закриття або посилання з меню. Закриття меню відбувається з невеликою затримкою для плавності анімації, і при натисканні на посилання відбувається перенаправлення до відповідного розділу.

```
menuItem.addEventListener('click', function(e) {
  e.preventDefault();
  sidebarMenu.classList.add('visible');
});

closeMenu.addEventListener('click', function(e) {
  e.preventDefault();
  closeSidebarMenuWithDelay();
});

links.forEach((element) => {
  element.addEventListener("click", function(e){
    e.preventDefault();
    const targetSection = this.getAttribute('href');
    closeSidebarMenuWithDelay(targetSection);
  });
});

function closeSidebarMenuWithDelay(targetSection = null) {
  // Закрити меню з затримкою
  setTimeout(function() {
    sidebarMenu.classList.remove('visible');
    if (targetSection) {
      window.location.href = targetSection;
    }
  }, 100);
}
```

Рисунок 2.8 – Реалізація функції відкриття та закриття бокового меню

Далі прописують алгоритми для кожної підсистеми окремо. Пропонуємо розглянути першу підсистему – загальне та аварійне освітлення.

Для управління рівнем освітлення використовується функція, яка обчислює поточний рівень природного освітлення на основі часу доби. На основі цього рівня визначається необхідна потужність штучного освітлення

для досягнення рекомендованого рівня освітленості. Розрахована потужність відображається на вебсторінці, і функція оновлюється кожну хвилину.

```
function getCurrentNaturalLightLevel() {
  const now = new Date();
  const hour = now.getHours();
  let naturalLightLevel;

  if (hour >= 6 && hour < 12) {
    naturalLightLevel = 200; // Ранок: 6:00 - 12:00
  } else if (hour >= 12 && hour < 18) {
    naturalLightLevel = 500; // День: 12:00 - 18:00
  } else if (hour >= 18 && hour < 22) {
    naturalLightLevel = 100; // Вечір: 18:00 - 22:00
  } else {
    naturalLightLevel = 10; // Ніч: 22:00 - 6:00
  }

  return naturalLightLevel;
}
```

Рисунок 2.9 – Алгоритм визначення поточного рівня природного освітлення

```
function updateLightingPower() {
  const currentNaturalLightLevel = getCurrentNaturalLightLevel();
  const requiredLuxFromArtificialLight = recommendedLux - currentNaturalLightLevel;

  let lightingPowerPercentage;

  if (requiredLuxFromArtificialLight <= 0) {
    lightingPowerPercentage = 0; // Немає необхідності у штучному освітленні
  } else if (requiredLuxFromArtificialLight >= maxArtificialLux) {
    lightingPowerPercentage = 100; // Необхідна максимальна потужність штучного освітлення
  } else {
    lightingPowerPercentage = (requiredLuxFromArtificialLight / maxArtificialLux) * 100;
  }

  document.getElementById('lightingPower').innerText = `Use recommended lighting power: ${lightingPowerPercentage}%`;
  document.getElementById('lightingPower1').innerText = `Use recommended lighting power: ${lightingPowerPercentage}%`;
}
```

Рисунок 2.9 – Алгоритм оновлення потужності освітлення

Функція управління рівнем освітлення дозволяє автоматично регулювати штучне освітлення залежно від часу доби та поточного рівня природного освітлення. Це забезпечує комфортні умови для користувачів та ефективне використання енергії в розумному будинку.

Для увімкнення та вимкнення відеоспостереження використовуються чек бокси. При зміні стану чек боксу відео відповідно починає відтворюватися або ставиться на паузу. Також змінюється клас відео для управління його стилем (освітленням).

```
videoCheckBox.addEventListener('change', function(e){
  if(this.checked){
    video1.play();
    video1.classList.remove('darken');
  }else{
    video1.pause();
    video1.classList.add('darken');
  }
});
```

Рисунок 2.10 – Використання чек боксів для увімкнення та вимкнення відеоспостереження

Аналогічно до відеоспостереження, реалізовано управління відео для підсистеми протипожежного захисту через окремий чек бокс.

```
fireCheckBox.addEventListener('change', function(e){
  if(this.checked){
    video2.play();
    video2.classList.remove('darken');
  }else{
    video2.pause();
    video2.classList.add('darken');
  }
});
```

Рисунок 2.11 – Реалізація управління відео для підсистеми протипожежного захисту

Також реалізовано обробку форми зворотного зв'язку. При натисканні кнопки посилання відбувається перехоплення події та збирання даних з полів форми. Зібрані дані формуються у вигляді поштового посилання mailto.

Обробка форми виконується за допомогою JavaScript. Коли користувач натискає кнопку "Send", перехоплюється подія submit, і виконується відповідна функція.

Функція addEventListener додає обробник події submit до форми з ідентифікатором contactForm. Використовується метод preventDefault(), щоб запобігти стандартній поведінці браузера (надсиланні форми та перезавантаженню сторінки).

```
document.getElementById('contactForm').addEventListener('submit', function(event){  
    event.preventDefault();  
});
```

Рисунок 2.12 – Реалізація функції addEventListener

Значення полів форми витягуються за допомогою методу getElementById та зберігаються у відповідних змінних (name, email, message).

```
var name = document.getElementById('name').value;  
var email = document.getElementById('email').value;  
var message = document.getElementById('message').value;
```

Рисунок 2.13 – Реалізація методу getElementById

Змінна mailtoLink містить посилання mailto, яке формується на основі введених даних:

- Адреса електронної пошти, на яку буде відправлено повідомлення (example@example.com).
- Параметр subject визначає тему повідомлення, яка включає ім'я користувача.
- Параметр body містить текст повідомлення, який включає ім'я, електронну пошту та саме повідомлення. Використовується функція encodeURIComponent, щоб уникнути проблем із символами, які можуть мати спеціальне значення в URL.

```
var mailtoLink = 'mailto:example@example.com' +  
    '?subject=' + encodeURIComponent('Повідомлення від ' + name) +  
    '&body=' + encodeURIComponent('Ім'я: ' + name + '\n' +  
        'Електронна пошта: ' + email + '\n' +  
        'Повідомлення: ' + message);
```

Рисунок 2.13 – Формування посилання mailto

Використовується властивість `location.href` об'єкта `window` для перенаправлення браузера до згенерованого посилання `mailto`.

```
window.location.href = mailtoLink;
```

Рисунок 2.14 – Перенаправлення браузера до згенерованого посилання `mailto`

Дана форма зворотного зв'язку дозволяє користувачам легко відправляти повідомлення адміністраторам чи технічній підтримці. Такий підхід забезпечує простоту інтеграції та зручність використання, оскільки не потребує налаштування додаткових серверних компонентів для обробки повідомлень.

3. ПРОЕКТНІ РІШЕННЯ. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис проектних рішень з програмного забезпечення

У рамках дипломної роботи було створено інтерфейс, що дозволяє користувачам ефективно взаємодіяти з розумними пристроями в їхньому будинку. Для досягнення цієї мети було обрано використання технологій HTML, CSS та JavaScript, які є стандартом для створення інтерфейсів.

HTML використовується для створення структури вебсторінки. В основі проекту лежить розмітка, яка визначає розташування та організацію елементів інтерфейсу. Використання семантичних тегів HTML5 забезпечує не тільки чітку структуру, але й покращує доступність та SEO-оптимізацію сайту.

Також HTML забезпечує розширені можливості для створення інтерактивних форм, які є основою для введення та обробки даних користувачами. Використання таких елементів, як `<form>`, `<input>`, `<select>`, `<button>`, дозволяє створювати зручні та функціональні інтерфейси для збору інформації. До прикладу, в проекті за допомогою елементів `<form>` секції контактів для збору даних користувача. Елементи `<input>` використовуються для введення текстових даних, email та чек боксів для керування різними системами. За допомогою `<textarea>` створена текстова область, яка використовується для введення повідомлення користувача у формі контактів.

Дана технологія легко інтегрується з іншими, такими як CSS для стилізації та JavaScript для динамічної взаємодії. Це дозволяє створювати комплексні вебдодатки з високим рівнем інтерактивності та привабливим дизайном.

CSS було обрано як основну технологію для стилізації інтерфейсу інформаційної системи. Насамперед CSS дозволяє відокремити презентаційний стиль від структурної розмітки HTML. Це забезпечує чистоту та організованість коду, спрощує його підтримку та оновлення. Завдяки цьому, зміни у стилях можуть бути здійснені без впливу на структуру HTML.

Використання CSS медіазапитів дозволяє створювати адаптивні вебсторінки, які виглядають та функціонують належним чином на різних пристроях та екранах. Це важливо для забезпечення зручного користувацького досвіду на мобільних телефонах, планшетах та настільних комп'ютерах.

Однією з переваг даної технології – модульність та повторне використання. CSS дозволяє створювати модульні стилі, які можуть бути повторно використані на різних частинах вебсторінки або у різних проєктах. Це сприяє економії часу та ресурсів під час розробки та підтримки.

Під час роботи було використано анімації та переходи, для цього ідеально підійшла технологія CSS3, яка включає потужні можливості для створення анімацій та плавних переходів між станами елементів. Приклад коду, який представлений на рисунку 3.1, є фрагментом, який забезпечує плавний перехід для фільтра яскравості відео. Коли до елемента відео додається клас `darken`, яскравість плавно змінюється протягом 0.3 секунди.

```
video {
  transition: filter 0.3s ease;
}

.darken {
  filter: brightness(20%);
}
```

Рисунок 3.1 – Фрагмент коду CSS, який забезпечує плавний перехід для фільтра яскравості відео

Також прикладом (рисунок 3.2) є плавний перехід для зміни кольору фону та переміщення перемикача (`slider`) при зміні стану чек бокса. При зміні стану чек бокса, фон та позиція елемента `slider:before` плавно змінюються протягом 0.4 секунди.

```

.slider {
  position: absolute;
  cursor: pointer;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: #777;
  transition: .4s;
}

.slider:before {
  position: absolute;
  content: "";
  height: 26px;
  width: 26px;
  left: 4px;
  bottom: 4px;
  background-color: #ffffff;
  transition: .4s;
}

input:checked + .slider {
  background-color: #f3b121;
}

input:checked + .slider:before {
  transform: translateX(26px);
}

```

Рисунок 3.2 – Фрагмент коду CSS для зміни кольору фону та переміщення перемикача

JavaScript було обрано як основну технологію для додавання динамічної поведінки до інтерфейсу інформаційної системи. Адже ця мова програмування дозволяє змінювати та оновлювати контент на вебсторінці без необхідності її перезавантаження. Це забезпечує більш інтерактивний та чуйний інтерфейс, який реагує на дії користувача в реальному часі. Як приклад, до проєкту була додана важлива функція за допомогою JS (рисунок 3.4), яка дозволяє відтворювати або зупиняти відео залежно від того, чи вибрано відповідний чек бокс. [5]

```

videoCheckBox.addEventListener('change', function(e){
  if(this.checked){
    video1.play();
    video1.classList.remove('darken');
  }else{
    video1.pause();
    video1.classList.add('darken');
  }
});

```

Рисунок 3.3 – Фрагмент коду JS, що дозволяє відтворювати або зупиняти відео

JavaScript дозволяє здійснювати маніпуляції з Document Object Model (DOM), що забезпечує динамічну зміну структури та стилів елементів сторінки. Це критично важливо для реалізації інтерактивних функцій, таких як модальні вікна, динамічні форми та інші елементи. Насамперед для проєкту потрібно було створити форму для надсилання електронного листа на основі введених користувачем даних, тому ідеальним рішенням було використати JS (рисунок 3.3).

```

document.getElementById('contactForm').addEventListener('submit', function(event) {
  event.preventDefault();

  var name = document.getElementById('name').value;
  var email = document.getElementById('email').value;
  var message = document.getElementById('message').value;

  var mailtoLink = 'mailto:example@example.com' +
    '?subject=' + encodeURIComponent('Повідомлення від ' + name) +
    '&body=' + encodeURIComponent('Ім'я: ' + name + '\n' +
      'Електронна пошта: ' + email + '\n' +
      'Повідомлення: ' + message);

  window.location.href = mailtoLink;
});

```

Рисунок 3.4 – Фрагмент коду JS для створення форми

JavaScript забезпечує можливість обробки подій, таких як кліки, наведення миші, введення даних, що дозволяє реалізовувати інтерактивні функції, які покращують користувацький досвід.

Поєднання HTML, CSS та JavaScript є потужним інструментом для створення інтерфейсу проєкту розумного будинку завдяки їх сумісності,

багатофункціональності та здатності забезпечувати інтерактивність і динамічність. Використовуючи ці технології, було створено зручний, привабливий та ефективний користувацький інтерфейс.

Для розробки програмного забезпечення було обрано спіральну модель життєвого циклу. Цей підхід дозволяє проводити багаторазові ітерації через чотири основні фази: планування, аналіз ризиків, інженерія (конструювання) і оцінка. Дана модель життєвого циклу дозволяє врахувати всі вимоги та ризики, пов'язані з розробкою трьох ключових підсистем. Для розробки інтерфейсу програми розумного будинку, що включає підсистеми загального та аварійного освітлення, протипожежного захисту та відеоспостереження, спіральна модель забезпечує гнучкість та адаптивність до змінних вимог і технологічних умов.

Фази спіральної моделі:

1. Планування:

- 1) Цілі: Визначення цілей проєкту, загальних вимог до інтерфейсу та підсистем.
- 2) Вимоги: Збір та документування вимог користувачів для кожної підсистеми.
 - Загальне та аварійне освітлення: Вимоги до управління освітленням, автоматизації та аварійних сценаріїв.
 - Протипожежний захист: Вимоги до моніторингу рівня диму в приміщенні.
 - Відеоспостереження: Вимоги до перегляду відео в реальному часі.

2. Аналіз ризиків

- 1) Ідентифікація ризиків: Виявлення можливих ризиків для кожної підсистеми, включаючи технічні, операційні та ризики користувацького досвіду.
- 2) Оцінка ризиків: Аналіз і пріоритезація ризиків за їх впливом та ймовірністю.

- Загальне та аварійне освітлення: Наприклад, можливість збою сенсорів та їх резервування.
- Протипожежний захист: Надійність датчиків диму та систем оповіщення.
- Відеоспостереження: Безпека даних та конфіденційність.

3. Інженерія (конструювання)

1) Прототипування: Створення прототипів інтерфейсу для кожної підсистеми з урахуванням вимог та ризиків.

- Загальне та аварійне освітлення: Прототип інтерфейсу для управління освітленням.
- Протипожежний захист: Прототип інтерфейсу для моніторингу датчиків диму.
- Відеоспостереження: Прототип інтерфейсу для перегляду відео в реальному часі.

2) Розробка та інтеграція: Реалізація функціональності інтерфейсу на основі прототипів та інтеграція підсистем в єдиний інтерфейс.

3) Тестування: Проведення тестування функціональності та юзабіліті інтерфейсу.

4. Оцінка

1) Зворотний зв'язок: Збір відгуків від користувачів.

2) Аналіз результатів: Оцінка зібраних даних для виявлення недоліків та можливостей покращення.

Спіральна модель життєвого циклу є ідеальним підходом для розробки інтерфейсу програми розумного будинку. Вона дозволяє поетапно вдосконалювати інтерфейс, забезпечуючи гнучкість і адаптивність до змінних вимог. Завдяки ітераційному підходу та акценту на управлінні ризиками, спіральна модель сприяє створенню надійного та ефективного інтерфейсу для керування підсистемами.

3.2 Розробка компонентів програмного забезпечення

3.2.1 Підготовка вхідних даних

При розробці компонентів програмного забезпечення для сайту важливо забезпечити зрозумілий процес входу та реєстрації користувачів, який буде ідентифікувати, щоб спростити доступ до облікових записів та забезпечити безпеку даних. Це дозволяє користувачам легко взаємодіяти з сайтом, знижує кількість помилок при введенні інформації та забезпечує надійний захист особистих даних. Нижче описано підготовку вхідних даних для створення цих компонентів, а також процес їх реалізації.

Головна сторінка з вибором дії. Коли користувач заходить на сайт, йому надається можливість обрати між входом у систему або створенням нового облікового запису. Це реалізовано у вигляді двох кнопок: "Log in" та "Create account", розміщених на головній сторінці. Візуальне представлення цього інтерфейсу наведено на рисунку 3.5.

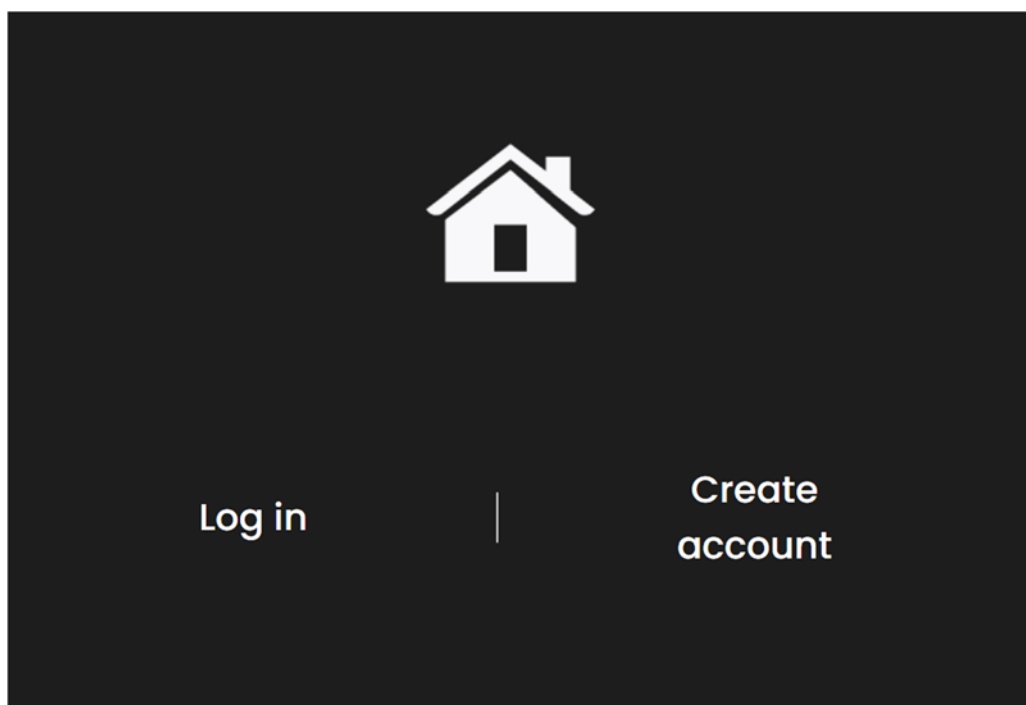


Рисунок 3.5 – Сторінка з вибором дії "Log in" або "Create account"

Якщо користувач вже зареєстрований, він може перейти до форми входу, натиснувши на кнопку "Log in". У формі входу користувач має ввести

своє ім'я та електронну пошту. Після заповнення цих полів користувач натискає кнопку "Log in", щоб увійти в систему. Візуальне представлення форми входу наведено на рисунку 3.6.

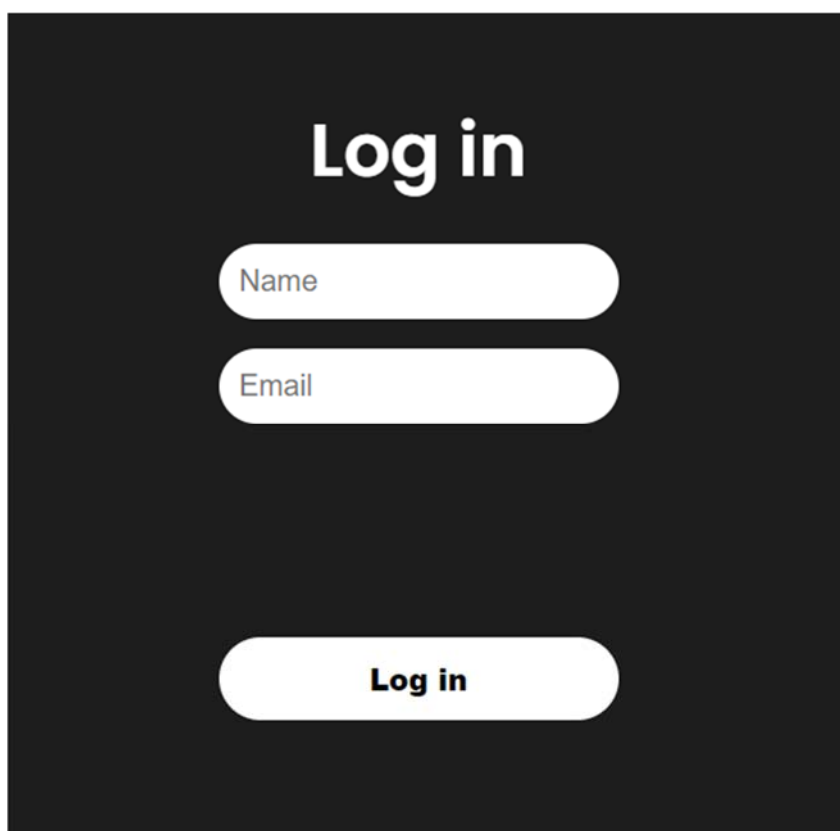
The image shows a dark-themed login interface. At the top, the text "Log in" is displayed in a large, white, sans-serif font. Below this, there are two white, rounded rectangular input fields. The first field is labeled "Name" and the second is labeled "Email". At the bottom of the form, there is a white, rounded rectangular button with the text "Log in" in a bold, black font.

Рисунок 3.6 – Сторінка "Log in"

Якщо користувач не зареєстрований, він обирає опцію "Create account". На сторінці реєстрації користувач має заповнити кілька полів, таких як ім'я, електронна пошта, назва вулиці, номер будинку та номер квартири. Після заповнення всіх полів користувач натискає кнопку "Create" для створення нового облікового запису. Візуальне представлення форми реєстрації наведено на рисунку 3.7.

The image shows a dark-themed user interface for creating an account. At the top, the text "Create account" is displayed in a large, white, sans-serif font. Below this, there are five white, rounded rectangular input fields stacked vertically, each containing a label: "Name", "Email", "Street name", "House number", and "Apartment number". At the bottom of the form is a single, wide white button with rounded corners, labeled "Create" in a bold, black, sans-serif font.

Рисунок 3.7 – Сторінка "Create account"

Процес реєстрації та входу дозволяє забезпечити конфіденційність та захист персональних даних користувачів, що є критично важливим у сучасному цифровому середовищі. Форма реєстрації дозволяє точно ідентифікувати кожного користувача, що забезпечує індивідуалізований доступ до послуг та інформації на сайті. Це допомагає уникнути несанкціонованого доступу та зловживання.

3.2.2 Опис класів

В даному проєкті реалізовані класи, що дозволяють організувати код у логічні модулі, що полегшує його розуміння. Кожен клас відповідає за конкретну функціональність, що полегшує знаходження, редагування та розширення коду. Для роботи програми розумного будинку важливо групувати дані та методи, що пов'язані між собою, в одній структурі. Це дозволяє приховати внутрішню реалізацію і забезпечує легкий доступ до функціональності, не розголошуючи всі деталі.

Для реалізації інтерфейсу програми інформаційної системи було виділено основні класи:

1. Menu – відповідає за інтерактивність меню в програмі. Забезпечує можливість відкриття та закриття меню, а також затримання закриття меню для виконання певних операцій.

Основні методи:

- openMenu(): Відкриває бічне меню.
- closeMenu(): Закриває бічне меню.
- closeMenuWithDelay(targetSection): Закриває бічне меню з затримкою.

2. VideoControl – відповідає за керування відтворенням відео та зміною його стилів залежно від стану чек боксів. Використовується для взаємодії з відеопрогравачами та забезпечення користувачу контролю над відтворенням.

Основні методи:

- playVideo1(): Відтворює перше відео.
- pauseVideo1(): Призупиняє відтворення першого відео.
- playVideo2(): Відтворює друге відео.
- pauseVideo2(): Призупиняє відтворення другого відео.

3. LightingControl – відповідає за контроль освітлення в приміщенні. Повертає поточний рівень природного світла та оновлює інформацію

про використану потужність освітлення залежно від умов освітлення та вимог.

Основні методи:

- `getCurrentNaturalLightLevel()`: Повертає поточний рівень природного світла.
- `updateLightingPower()`: Оновлює інформацію про використану потужність освітлення.

4. `ContactForm` – відповідає за взаємодію з формою зв'язку. Надає можливість надсилання повідомлення електронною поштою з введеними користувачем даними.

Основний метод: `sendEmail(name, email, message)`, що відправляє повідомлення електронною поштою.

Кожен клас має свій набір методів, які забезпечують необхідну функціональність. Використання класів дозволяє чітко розділити відповідальності між частинами програми. Кожен клас відповідає за свою конкретну задачу.

Для візуалізації взаємозв'язків між класами та їхню функціональність у межах інформаційної системи представлено діаграму класів (рисунок 3.8).

Діаграма класів – це тип діаграми, який використовується для візуалізації структури системи на рівні класів. На діаграмі представлена сукупність декларативних або статичних елементів моделі, таких як класи з атрибутами й операціями, а також відношення, що їх з'єднують.

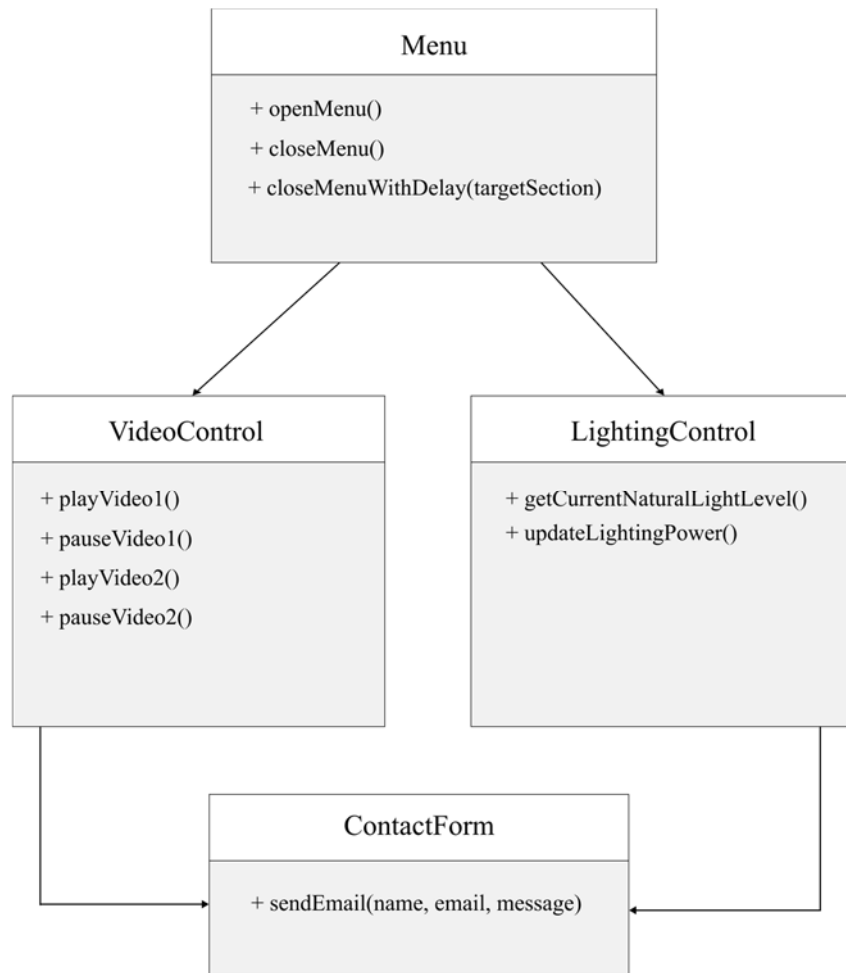


Рисунок 3.8 – Діаграма класів системи

Ця діаграма відображає основні класи системи та їх методи, показуючи, як організовані різні функціональні блоки інформаційної системи.

3.3 Опис тестового прикладу

Для того, щоб користувач увійшов у свій профіль програми йому потрібно обрати з-поміж двох дій "Log in" або "Create account". Якщо користувач вже раніше був зареєстрований він заходить за своїми даними вводячи своє ім'я та електронну пошту (рисунок 3.9). Якщо ні – користувачу потрібно зробити реєстрацію заповнюючи поля (рисунок 3.10).

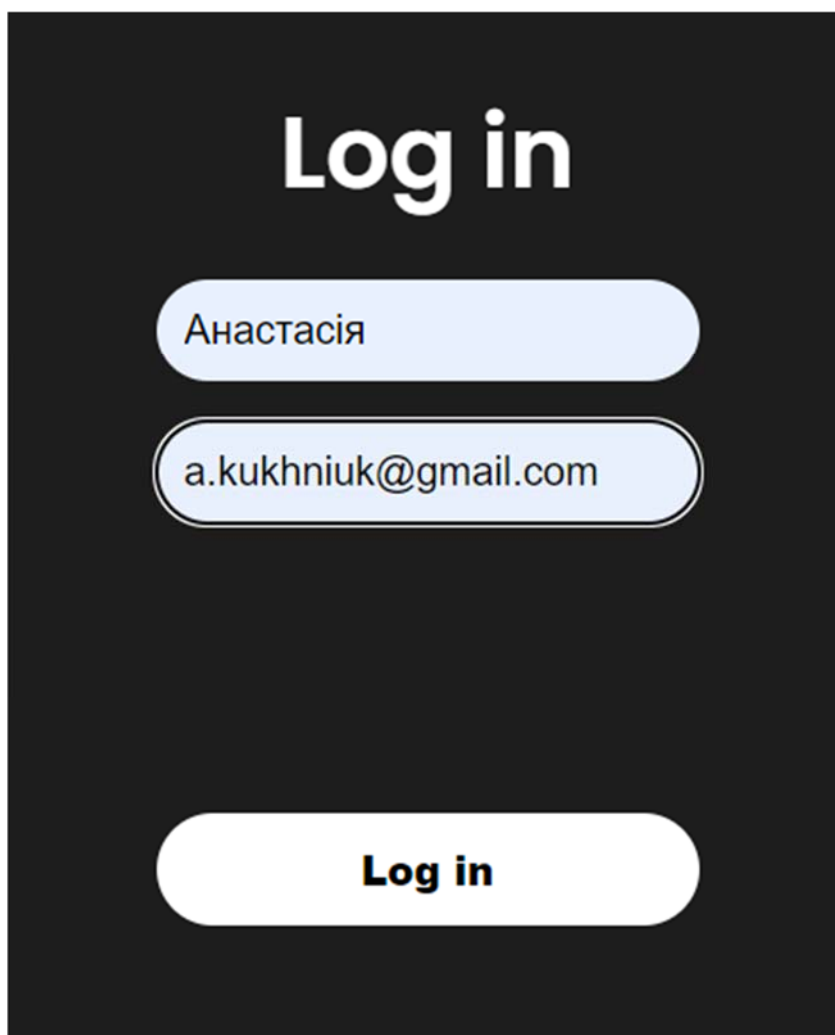
The image shows a dark-themed login interface. At the top, the text "Log in" is displayed in a large, white, sans-serif font. Below this, there are two light blue, rounded rectangular input fields. The first field contains the name "Анастасія" and the second field contains the email address "a.kukhniuk@gmail.com". At the bottom of the form, there is a white, rounded rectangular button with the text "Log in" in a bold, black, sans-serif font.

Рисунок 3.9 – Введення даних у форму "Log in"

The image shows a dark-themed 'Create account' form. At the top, the title 'Create account' is displayed in white. Below the title are five input fields, each with a light blue rounded rectangle containing the text: 'Анастасія', 'a.kukhniuk@gmail.com', 'Весняний провулок', '1', and '100'. At the bottom of the form is a white rounded rectangle with the text 'Create' in black.

Рисунок 3.10 – Введення даних у форму "Create account"

Далі користувач потрапляє на головну сторінку (рисунок 3.11), на головній сторінці відображається бокове меню (рисунок 3.12), яке спочатку приховане. Користувач натискає на іконку меню (`menuIcon`), щоб відкрити бокове меню.

Код JavaScript викликає метод `openMenu()`, який додає клас `visible` до елемента меню (`sidebarMenu`), роблячи його видимим. У боковому меню користувач бачить посилання ("Home" – повернутися на головну сторінку, "Video surveillance" – підсистема відеоспостереження, "Light" – підсистема освітлення, "Fire Protection" – підсистема протипожежного захисту, "+1234 5555-555" – номер телефону для зв'язку зі службою технічної підтримки).



Рисунок 3.11 – Головна сторінка

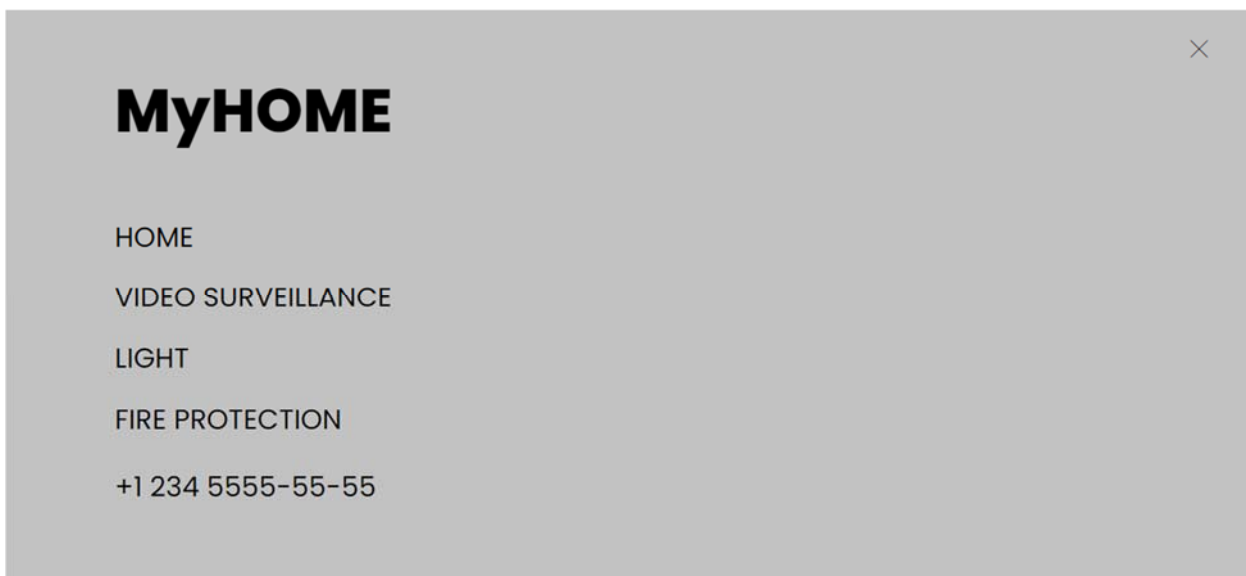


Рисунок 3.12 – Бокове меню

Також вибрати підсистему можна на головній сторінці (рисунок 3.13), де можна перейти за кнопкою «Go to».

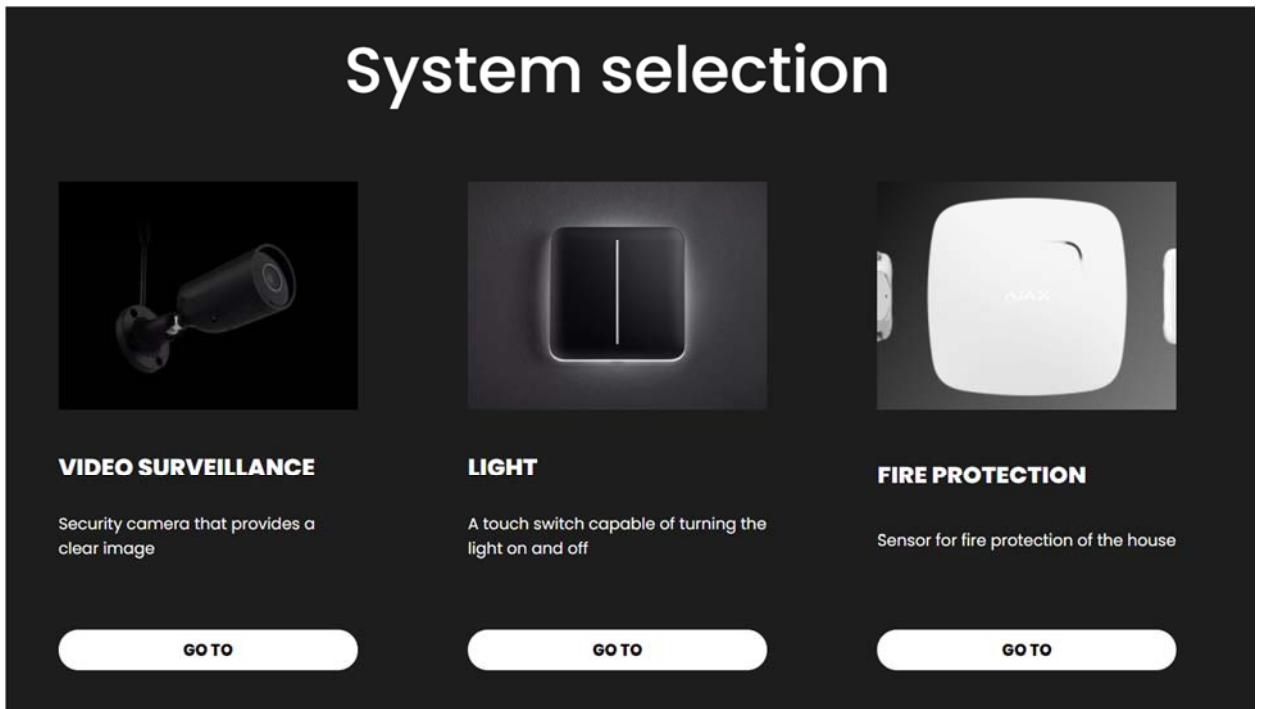


Рисунок 3.13 – Розділ сторінки, де обрання підсистеми

На сторінці розділу користувач бачить чек бокси для відеоспостереження (рисунок 3.14), загального та аварійного освітлення (рисунок 3.15) та протипожежної системи (рисунок 3.16).

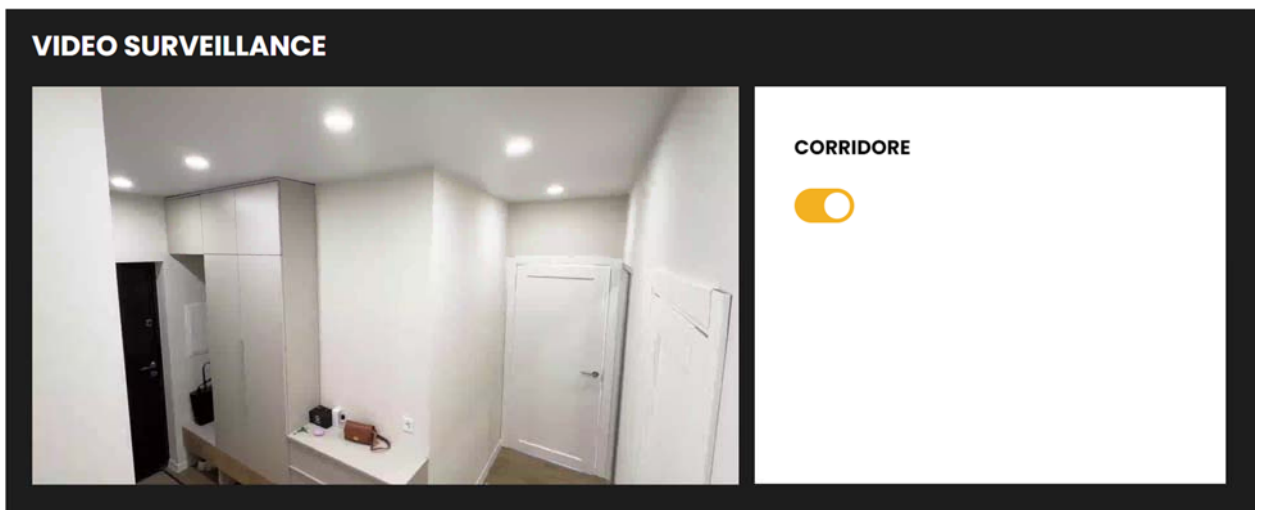


Рисунок 3.14 – Підсистема відеоспостереження

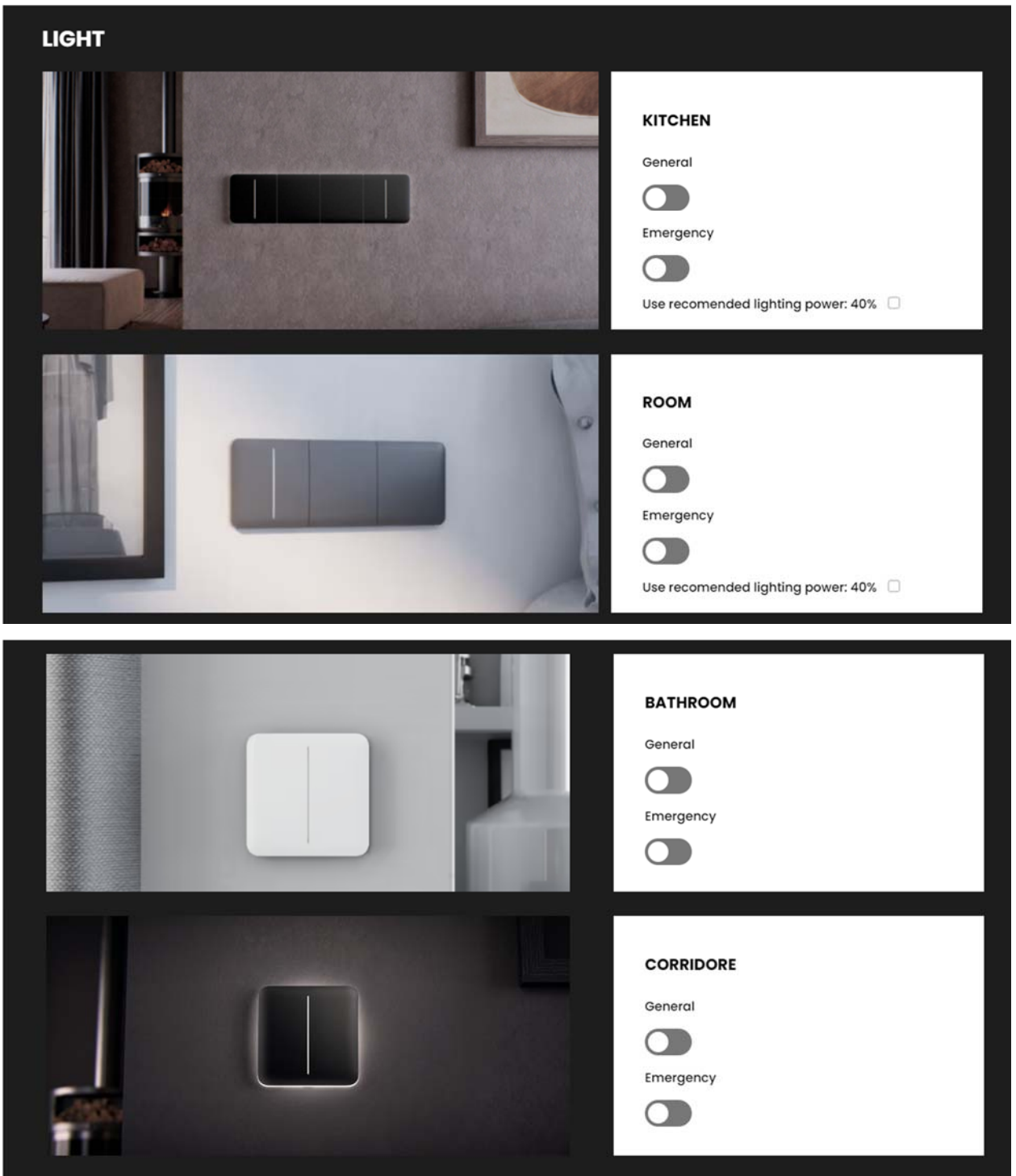


Рисунок 3.15 – Підсистема загального та аварійного освітлення

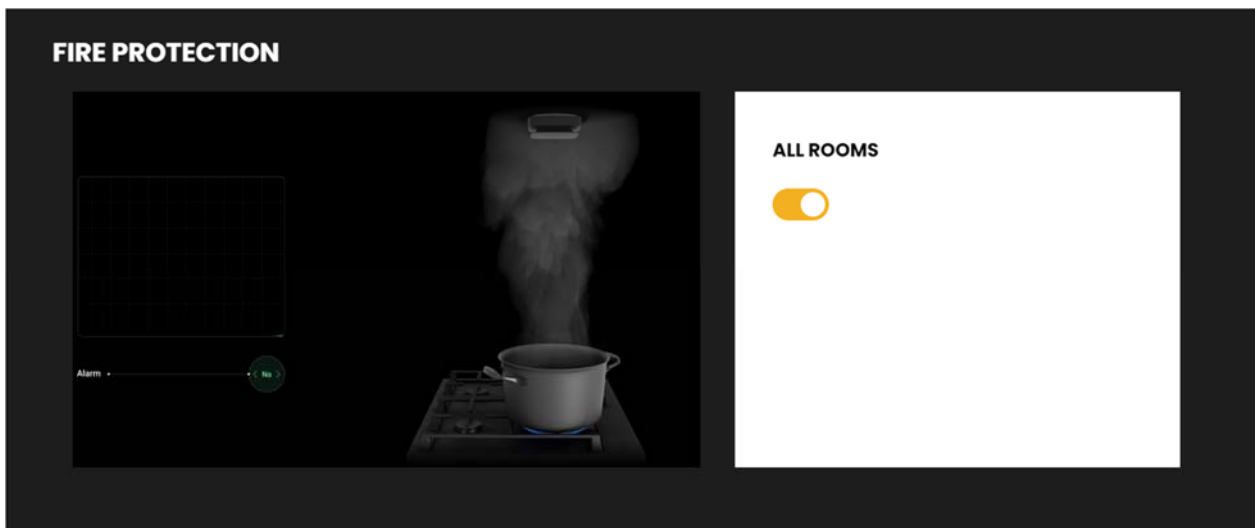


Рисунок 3.16 – Підсистема протипожежного захисту

Окрім цього, на головній сторінці користувач може знайти форму зворотного зв'язку (рисунок 3.17), де може залишити свої дані та повідомлення з питанням чи скаргою для служби підтримки.

Рисунок 3.16 – Форма зворотнього зв'язку

Отже, вище було показано тестовий приклад користування даною програмою, оглянувши основні розділи сторінки.

4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ

4.1 Опис проєктованого продукту

Інформаційна система розумного будинку призначена для автоматизації та інтеграції різних аспектів управління будинком, включаючи освітлення, протипожежний захист і відеоспостереження.

Метою розробки інтерфейсу інформаційної системи розумного будинку є створення зручного, інтуїтивно зрозумілого та ефективного засобу управління різними підсистемами будинку. Цей інтерфейс повинен забезпечити користувачам легкий доступ до функцій та налаштувань системи для підвищення комфорту та безпеки житла.

Основні цілі розробки інтерфейсу:

- 1) Підвищення зручності використання.
 - Інтерфейс, який буде зрозумілий користувачам з різним рівнем технічної підготовки.
 - Забезпечити простий та логічний доступ до всіх функцій та налаштувань системи.
 - Забезпечити оптимальну роботу інтерфейсу на різних пристроях, включаючи смартфони, планшети та комп'ютери.
- 2) Інтеграція всіх підсистем
 - Об'єднати управління всіх підсистем в одному інтерфейсі.
 - Надати можливість користувачам налаштовувати автоматичні сценарії для різних підсистем.
- 3) Підвищення ефективності
 - Швидкий доступ: Забезпечити можливість швидкого доступу до основних функцій, особливо в екстрених ситуаціях.
 - Оптимізація: Оптимізувати управління ресурсами для зниження споживання енергії та підвищення ефективності роботи системи.

4) Підвищення безпеки

- Надати користувачам можливість легко контролювати стан всіх підсистем і отримувати своєчасні повідомлення про будь-які відхилення чи загрози.
- Забезпечити безпеку персональних даних користувачів та захист системи від несанкціонованого доступу.

5) Забезпечення користувацького досвіду

- Надати користувачам можливість налаштовувати інтерфейс відповідно до їхніх потреб та вподобань.
- Забезпечити можливість отримання зворотного зв'язку від користувачів для подальшого вдосконалення системи.

Очікувані результати:

1. Зручність та ефективність: Інтерфейс, що дозволяє користувачам швидко та зручно керувати всіма аспектами розумного будинку.
2. Безпека та надійність: Система, яка забезпечує високий рівень безпеки та своєчасне інформування про можливі загрози.
3. Економія ресурсів: Оптимізоване використання енергії та інших ресурсів завдяки автоматизованим сценаріям та ефективному управлінню.
4. Задоволення користувачів: Позитивний користувацький досвід завдяки інтуїтивному та адаптивному інтерфейсу, що відповідає потребам і очікуванням користувачів.

Розробка такого інтерфейсу сприятиме підвищенню конкурентоспроможності інформаційної системи розумного будинку та її популярності серед споживачів.

4.2 Оцінка ринку збуту та аналіз конкурентів

Оцінка ринку збуту інформаційної системи розумного будинку є ключовим етапом для визначення потенціалу продажів та успіху продукту на ринку. Ця оцінка включає аналіз ринку, визначення цільової аудиторії, оцінку конкурентів та прогнозування продажів.

Розмір ринку.

За оцінками дослідницьких компаній, глобальний ринок розумних будинків швидко зростає. За даними MarketsandMarkets, ринок розумних будинків у 2021 році оцінювався приблизно в \$79,9 мільярда і прогнозується, що він досягне \$135,3 мільярда до 2025 року, з середньорічним темпом зростання (CAGR) близько 11%. [7]

Регіональний аналіз.

Північна Америка: Лідер ринку завдяки високому рівню впровадження новітніх технологій та інтересу споживачів до розумних рішень для будинку.

Європа: Значний ринок зі збільшеним попитом на енергоефективні рішення.

Азійсько-Тихоокеанський регіон: Швидке зростання завдяки урядовим ініціативам і збільшенню середньому класу.

Інші регіони: Включаючи Латинську Америку та Африку, також демонструють потенціал зростання, але з нижчою стартовою базою.

Аналіз цільової аудиторії.

Розуміння потреб кожної цільової аудиторії дозволяє розробити специфічні маркетингові стратегії та адаптувати функціонал системи під потреби конкретних груп користувачів. Це забезпечить високу привабливість продукту на ринку та сприятиме його успішному впровадженню.

Таблиця 4.1 – Аналіз цільової аудиторії

Категорія	Опис	Вік	Дохід	Локація
Першопрохідці та техно ентузіасти	Люди, які люблять новітні технології й готові інвестувати в розумні рішення для свого будинку.	25-45 років	Середній та високий	Міські та приміські райони з високою концентрацією молодих професіоналів
Сім'ї	Сім'ї, які шукають підвищеного комфорту, безпеки та енергоефективності для своїх домівок	30-55 років	Середній та вище середнього	Приміські та приватні будинки
Власники нерухомості	Власники великих будинків або компанії, які бажають підвищити вартість своєї нерухомості через інтеграцію розумних систем.	35-60 років	Високий	Міські райони, бізнес-центри, комерційні будівлі

Для кожної з категорій є базові важливі потреби та вимоги до інтерфейсу інформаційної системи розумного будинку. Для техно ентузіастів – інтуїтивно зрозумілий інтерфейс, який легко використовувати та широка сумісність з іншими розумними пристроями. Для людей в яких є діти важливою потребою є – безпека для дітей (контроль доступу до небезпечних зон, моніторинг дитячих кімнат). Для власників нерухомості – привабливий візуальний вигляд, який до того ж підвищить вартість нерухомості через інтеграцію розумних систем.

Аналіз конкурентів.

Аналіз конкурентів дозволяє оцінити поточний стан ринку, виявити сильні та слабкі сторони конкурентів, а також визначити конкурентні переваги власного продукту.

Під час аналізу було виявлено трьох основних конкурентів на ринку – Google Nest, ADT Pulse та Apple HomeKit.

1. Google Nest:

- Сильні сторони: Використання шифрування та інших технологій для захисту персональних даних користувачів.
- Слабкі сторони: Інтуїтивно не зрозумілий інтерфейс.
- Цільова аудиторія: Техно ентузіасти та сім'ї.

2. ADT Pulse

- Сильні сторони: Користувачі легко розуміють, як використовувати додаток без необхідності читання довгих інструкцій.
- Слабкі сторони: Невеликий вибір функцій та відсутність форми зворотного зв'язку.
- Цільова аудиторія: Сім'ї, власники нерухомості.

3. Apple HomeKit

- Сильні сторони: Висока безпека та конфіденційність, візуально привабливий інтерфейс.
- Слабкі сторони: відносно складний процес налаштування для початківців.
- Цільова аудиторія: сім'ї, техно ентузіасти.

Було виявлено сильні сторони наявних інтерфейсів компаній розробників, які спрямовані на забезпечення зручності, безпеки, продуктивності та інтеграції з іншими системами.

4.3 Маркетингова стратегія

Ефективна маркетингова стратегія охоплює різні аспекти, від позиціонування продукту до вибору маркетингових каналів та розробки рекламних кампаній.

Позиціонування: Інформаційна система розумного будинку, що забезпечує зручне, безпечне та інтуїтивно зрозуміле управління всіма аспектами вашого дому через єдиний інтегрований інтерфейс.

Унікальні торгові пропозиції (УТР):

- 1) Інтуїтивний інтерфейс: Простий у використанні, навіть для нетехнічних користувачів.
- 2) Висока сумісність: Підтримка широкого спектра пристроїв від різних виробників.
- 3) Доступна ціна: Конкурентна ціна без компромісів у якості.
- 4) Безпека: Високий рівень захисту даних та функції безпеки.

Маркетингові канали.

Проведений аналіз маркетингових каналів, а саме онлайн і офлайн методи, які можуть бути використані для залучення потенційних клієнтів, підвищення обізнаності про продукт та стимулювання продажів.

Таблиця 4.2 – Онлайн маркетингові канали

Соціальні мережі	Контент-маркетинг	Електронна пошта	SEO та SEM	Реклама в інтернеті
Facebook, Instagram, Twitter, LinkedIn, TikTok	Блоги, відео на YouTube, статті, подкасти	Розсилки новин, інформаційні бюлетені, персоналізовані пропозиції	Оптимізація для пошукових систем (SEO) та платна реклама в пошукових системах (SEM)	Google Ads, банерна реклама, контекстна реклама

Таблиця 4.3 – Офлайн маркетингові канали

Події та виставки	Поштові розсилки	Реклама на місцях продажу	Партнерські програми
Участь у галузевих виставках, конференціях, семінарах	Прямий маркетинг за допомогою друкованих листівок, каталогів	Рекламні матеріали в магазинах, стенди, вітрини	Співпраця з іншими компаніями, партнерами, дистриб'юторами

Для більш детального аналізу, який дозволить встановити, які фактори можуть впливати на ринок, які проблеми можуть виникнути в майбутньому і які ресурси доступні для досягнення цілей – проводиться SWOT-аналіз (табл. 4.4).

Таблиця 4.4 – SWOT-аналіз продукту

Сильні сторони	<ol style="list-style-type: none"> 1. Інтуїтивний інтерфейс, який є простішим у використанні порівняно з конкурента. 2. Підтримка широкого спектра пристроїв від різних виробників. 3. Наявність форми зворотного зв'язку для технічної підтримки користувача
Слабкі сторони	<ol style="list-style-type: none"> 1. Залежність від сторонніх платформ та пристроїв, що може впливати на стабільність та сумісність системи. 2. Наявність лише базових підсистем та функцій.
Можливості	<ol style="list-style-type: none"> 1. Впровадження нових функцій та інтеграція з додатковими пристроями для розширення можливостей системи. 2. Співпраця з виробниками розумних пристроїв та іншими технологічними компаніями для підвищення сумісності та функціональності.

Загрози	<ol style="list-style-type: none"> 1. Конкуренція з боку великих технологічних компаній з більшими маркетинговими бюджетами та впізнаваними брендами. 2. Швидкі технологічні зміни 3. Економічні коливання 4. Загрози з боку кіберзлочинців та витоку даних можуть вплинути на довіру споживачів до продукту.
----------------	---

SWOT-аналіз показує, що інформаційна система розумного будинку має багато сильних сторін, однак існують і слабкі сторони. Важливо також враховувати загрози, такі як сильна конкуренція та швидкі технологічні зміни, та розробляти стратегії для їх мінімізації.

4.4 План виробництва

План виробництва визначає ключові етапи розробки, виготовлення, тестування та впровадження інформаційної системи розумного будинку. Він включає всі необхідні процеси та часові рамки для ефективної реалізації проєкту. [8]

Таблиця 4.4 – План виробництва

Завдання	Тривалість	Дата початку	Дата завершення	Відповідальний
1. Аналіз вимог	15 робочих днів	03.06.2024	24.06.2024	Project-менеджер
1.1 Збір і аналіз вимог клієнта	5 робочих днів	03.06.2024	10.06.2024	Project-менеджер
1.2 Підготовка технічного завдання	7 робочих днів	10.06.2024	19.06.2024	Технічний спеціаліст
1.3 Підготовка документації	3 робочих дні	19.06.2024	24.06.2024	Технічний спеціаліст
2. Проєктування системи	40 днів	24.06.2024	19.08.2024	Frontend-розробник
2.1 Створення прототипів інтерфейсу	10 робочих днів	24.06.2024	08.07.2024	Frontend-розробник
2.2 Проєктування системи та архітектури	10 робочих днів	08.07.2024	22.07.2024	Frontend-розробник
2.3 Розробка інтерфейсу користувача	20 робочих днів	22.07.2024	19.08.2024	Frontend-розробник
3. Інтеграція та тестування	30 днів	19.08.2024	30.09.2024	Тестувальник

3.1 Інтеграція з апаратними компонентами	10 днів	19.08.2024	02.09.2024	Технічний спеціаліст
3.2 Тестування інтерфейсу	15 днів	02.09.2024	23.09.2024	Тестувальник
3.3 Виправлення помилок та оптимізація	5 днів	23.09.2024	30.09.2024	Frontend-розробник
4. Підготовка до запуску	5 днів	30.09.2024	07.10.2024	Технічний спеціаліст
4.1 Підготовка технічної документації	5 днів	30.09.2024	07.10.2024	Технічний спеціаліст
5. Запуск кінцевого продукту	1 день	07.10.2024	08.10.2024	Project-менеджер, frontend-розробник
5.1 Офіційний запуск продукту	1 день	07.10.2024	08.10.2024	Project-менеджер, frontend-розробник
5.2 Підтримка та оновлення	Постійний процес	08.10.2024	Постійний процес	Frontend-розробник

Тобто загальний дедлайн проєкту – 08.10.2024, що означає загальний термін від початку до кінцевого продукту – 4 місяці.

Цей план виробництва передбачає ретельну підготовку та виконання кожного етапу розробки інтерфейсу інформаційної системи розумного будинку. Дотримання встановлених термінів та ефективного управління проєктом забезпечить вчасне та якісне завершення проєкту.

4.5 Фінансовий план та стратегія фінансування

Фінансовий план включає детальний розрахунок витрат на кожен етап виробництва, починаючи з планування і закінчуючи запуском та підтримкою продукту. План базується на оцінці необхідних ресурсів, часу і людських витрат.

Таблиця 4.4 – План виробництва

Завдання	Витрати, \$
Аналіз вимог та ринку	2,500
Розробка технічного завдання	3000
Створення прототипів інтерфейсу	3000
Проектування системи та архітектури	6000
Розробка інтерфейсу користувача	10000
Інтеграція	7000
Тестування	4500
Виправлення помилок і оптимізація	2500
Підготовка технічної документації	2000
Запуск кінцевого продукту	1000
Підтримка (перший місяць)	2000
Всього	42,500

Загальний бюджет на розробку інтерфейсу інформаційної системи розумного будинку становить приблизно \$42,500. Цей план враховує всі ключові етапи від планування до запуску і підтримки продукту. Ефективне управління фінансами та ресурсами допоможе забезпечити успішне завершення проєкту в рамках встановлених термінів і бюджету.

4.6 Оцінка ризику та страхування

Ризик – невизначеність, що пов'язана з можливістю виникнення в ході реалізації бізнес-плану несприятливих умов, ситуацій та наслідків. Врахування ризиків у проєкті розробки інтерфейсу інформаційної системи розумного будинку є критично важливим для успіху проєкту з кількох причин. По-перше, це дозволяє прогнозувати та запобігати можливим проблемам, що можуть виникнути під час реалізації проєкту, таким чином мінімізуючи фінансові втрати та забезпечуючи стабільність графіка.

По-друге, врахування ризиків сприяє стабільності проєкту, дозволяючи уникати затримок і підтримувати продуктивність та злагоджену роботу команди. Це також підвищує конкурентоспроможність, дозволяючи вчасно вивести на ринок якісний продукт, зміцнює репутацію компанії та робить її більш привабливою для клієнтів й інвесторів.

Також важливим аспектом є забезпечення кібербезпеки. Врахування ризиків, пов'язаних з кібербезпекою, дозволяє краще захистити дані користувачів, знизити ризик кібератак та підтримувати конфіденційність інформації, що підвищує довіру користувачів.

Нижче наведено кілька категорій ризиків, які притаманні для теми проєкту:

1. Технічні ризики:

- Виявлення критичних неполадок під час або після тестування.
- Несумісність компонентів системи.

2. Ризики, пов'язані з персоналом:

- Нестача кваліфікованих фахівців.
- Низька продуктивність команди.

3. Фінансові ризики:

- Перевищення бюджету.
- Недостатнє фінансування на етапах розробки або запуску.

- Затримка у виплатах від інвесторів чи партнерів.
4. Ризики управління проектом:
- Зриви термінів виконання завдань.
 - Неефективне планування та управління ресурсами.
 - Зміни в вимогах клієнтів або ринку.
5. Ризики безпеки:
- Уразливості системи до кібератак.
 - Недостатній захист особистих даних користувачів.
6. Юридичні та регуляторні ризики:
- Невідповідність законодавчим вимогам.
 - Порушення авторських прав.

Аналіз ризиків.

Таблиця 4.5 – Ймовірність та вплив ризиків

Ризик	Ймовірність	Вплив	Примітка
Критичні неполадки	Середня	Високий	Може призвести до простою
Нестача фахівців	Середня	Середній	Затримка в термінах
Перевищення бюджету	Середня	Високий	Необхідність додаткових фінансів
Зрив термінів	Висока	Високий	Втрата довіри клієнтів
Кібератаки	Низька	Високий	Може вплинути на репутацію
Юридичні проблеми	Низька	Високий	Може призвести до штрафів

Управління ризиком означає взаємодію зі змінами та небезпеками під час виконання проєкту. Ключовим аспектом є постійний моніторинг ризиків, що містить стеження за ними протягом всього періоду проєкту. Якісний моніторинг забезпечує необхідну інформацію для прийняття ефективних рішень перед тим, як ризик стане актуальним.

Стратегії управління ризиками містять зменшення, уникнення, прийняття та передачу ризиків.

Зменшення ризиків здійснюється через:

- Впровадження автоматизованих тестів для виявлення неполадок.
- Регулярне оновлення програмного забезпечення безпеки.
- Резервування ключових компонентів системи.
- Підвищення кваліфікації персоналу через тренінги.

Уникнення ризиків здійснюється шляхом:

- Детального планування бюджету та ресурсів.
- Чіткого визначення вимог та контролю змін.
- Використання перевірених технологій та методологій розробки.

Для проєкту інтерфейсу інформаційної системи розумного будинку може бути використане страхування як стратегічний інструмент для мінімізації фінансових ризиків. Ось деякі аспекти, які можуть бути застосовані до страхування в рамках цього проєкту:

- 1) Страхування від технічних ризиків: Страхування може захистити проєкт від фінансових втрат, пов'язаних з технічними проблемами, такими як відмови апаратного забезпечення або програмного забезпечення, несправності компонентів системи тощо. Це може включати поліси на випадок відмови обладнання або послуги страхування від втрати даних.

- 2) Страхування від кібербезпеки: У зв'язку з тим, що системи розумного будинку зазвичай мають підключення до Інтернету та зберігають конфіденційну інформацію про користувачів, страхування від кібератак може бути важливим.
- 3) Страхування від відповідальності: Передбачення страхування від відповідальності може бути важливим для захисту проєкту від можливих судових позовів або відшкодування шкоди, що може виникнути через використання або експлуатацію інформаційної системи.

Страхування допомагає зменшити фінансові ризики, забезпечуючи певний рівень захисту від непередбачених обставин.

Отже, врахування ризиків є ключовим аспектом успішного управління проєктом, який дозволяє мінімізувати втрати, підвищити якість продукту, забезпечити стабільність і ефективність процесів та зміцнити репутацію компанії на ринку.

ВИСНОВКИ

У результаті проведеної атестаційної роботи було успішно розроблено інтерфейс інформаційної системи, який призначений для керування елементами обладнання внутрішньої інженерії розумного будинку. Цей інтерфейс створений з використанням ключових технологій веброзробки, зокрема HTML, CSS та JavaScript, що дозволило досягти високої функціональності та зручності використання.

Під час розробки було надано особливу увагу до деталей та потреб користувачів. Інтерфейс був спроектований таким чином, щоб забезпечити зручний та інтуїтивно зрозумілий доступ до всіх функціональних можливостей системи керування. Для цього були використані сучасні стандарти дизайну та веброзробки, що дозволило створити естетичний та легко використовуваний інтерфейс.

Окрім того, в процесі розробки були додані інноваційні функціональні можливості, які підвищують зручність та ефективність використання системи.

Загалом, розроблений інтерфейс відповідає сучасним вимогам технологій та відповідає потребам користувачів у сфері розумного будівництва. Його ефективність та зручність використання сприятимуть підвищенню комфорту та безпеки в майбутніх розумних будинках.

Список використаних джерел

1. Cooper, A., Reimann, R., Cronin, D., "About Face: The Essentials of Interaction Design", 650 с. [Електронний ресурс]. – URL: <https://www.wiley.com/en-us/About+Face%3A+The+Essentials+of+Interaction+Design%2C+4th+Edition-p-9781118766576>. [Дата звернення: 18.04.2024]
2. Nielsen, J., "Usability Engineering", 362 с. [Електронний ресурс]. – URL: <https://www.elsevier.com/books/usability-engineering/nielsen/978-0-12-518406-9>. [Дата звернення: 19.04.2024]
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 640 с. [Електронний ресурс]. – URL: <https://www.pearson.com/store/p/designing-the-user-interface/P100000385269>. [Дата звернення: 03.05.2024]
4. Norman, D., "The Design of Everyday Things: Revised and Expanded Edition", 384 с. [Електронний ресурс]. – URL: <https://www.basicbooks.com/titles/don-norman/the-design-of-everyday-things/9780465050659/>. [Дата звернення: 08.04.2024]
5. Freeman, E., Robson, E., Bates, B., Sierra, K., "Head First JavaScript Programming", 704 с. [Електронний ресурс]. – URL: <https://www.oreilly.com/library/view/head-first-javascript/9781449340124/>. [Дата звернення: 12.05.2024]
6. Garrett, J. J., "The Elements of User Experience: User-Centered Design for the Web and Beyond", 272 с. [Електронний ресурс]. – URL: <https://www.pearson.com/store/p/elements-of-user-experience-the/P100000082014>. [Дата звернення: 20.05.2024]
7. Research and Markets. Smart Home Market by Product, Lighting Control, Security & Access Control, - Global Forecast to 2025. [Електронний ресурс]. – URL:

<https://www.researchandmarkets.com/reports/5135225/smart-home-market-by-product-lighting-control>. [Дата звернення: 20.05.2024]

8. COBIT 2019 Framework: Governance and Management Objectives. [Електронний ресурс]. – URL: <https://www.isaca.org/bookstore/cobit/9781604207643>. [Дата звернення: 23.05.2024]