НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.9:37.014.6:37.018.43

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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Система контролю кліматичних параметрів у складському приміщенні

Спеціальність _____122 Комп'ютерні науки____

_____

(код і назва)

Освітня програма ____Комп'ютерний еколого-економічний моніторинг

_____

(назва)

Орієнтація освітньої програми _____

(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

к.т.н., доцент кафедри комп'ютерних наук _____ Басараб Р.М.

(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи

_____к.ф.-м.н., доцент_____ _____Лялецький О.В._

(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ _____Прийдун М.І._____

(підпис) (ПІБ студента)

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри _____

_____ _____ _____
(науковий ступінь, вчене звання) (підпис) (ПІБ)
“ ” _____ 20 ____ року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Прийдуну Михайлу Івановичу
(прізвище, ім’я, по батькові)

Спеціальність Комп’ютерні науки

Освітня програма Комп’ютерний еколого-економічний моніторинг

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Система контролю кліматичних параметрів у складському приміщенні

затверджена наказом ректора НУБіП України від “29” листопада 2021 р. №2025“С”

Термін подання завершеної роботи на кафедру 30 листопада 2021 р.

Вихідні дані до магістерської кваліфікаційної роботи: дані щодо оцінки якості освітніх ресурсів

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Аналіз предметної області	30.10.2020-10.11.2020	
2.	Аналіз вже існуючих рішень	15.11.2020-25.11.2020	
3.	Визначити методи та технології моніторингу кліматичних параметрів	12.12.2020-20.01.2020	

4.	Проектування архітектури системи	02.02.2021-05.03.2021
5.	Дослідження динаміки змін параметрів моніторингу відносно визначених критеріїв	15.03.2021-20.07.2021
6.	Аналіз результатів дослідження та підведення підсумків	01.09.2021-14.11.2021
7.	Попередній захист	30.11.2021
8.	Захист	14.12.2021

Дата видачі завдання “29”_____жовтня_____2020 р.

Керівник магістерської кваліфікаційної роботи _____ __Лялецький О.В.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ __Прийдун М.І.
(підпис) (прізвище та ініціали студента)

НУБіП України

Зміст

ВСТУП 5

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 6

Особливості розроблюваної системи (табл. 1) 6

1.1 Постановка задачі 6

1.2 Огляд інформаційних джерел та існуючих рішень 8

1.3 Моделювання предметної області 10

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ 17

2.1 Побудова логічної моделі даних 17

2.2 Вибір інструментарію для створення БД та роботи з нею 21

2.3 Побудова структури БД 23

3 ПРИКЛАДНЕ ПРОГРАМНЕ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ 24

3.1 Обгрунтування технічного забезпечення системи 24

3.2 Вибір інструментарію для створення програмного забезпечення 34

4 РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ 45

4.1 Тестування системи 45

4.2.2 Вимоги до апаратного забезпечення 47

4.2.2.1 Вимоги до апаратного забезпечення сервера 47

4.2.2.2 Вимоги до апаратного забезпечення клієнту користувача 47

4.2.3 Вимоги до програмного забезпечення 47

4.2.3.1 Вимоги до системного програмного забезпечення 47

4.2.3.2 Склад інсталяційного пакету для встановлення розробленої системи 48

5 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ 49

5.1 Оцінка, час реалізації та календарний план 49

5.2 Розрахунок витрат на розробку проекту 57

ВИСНОВКИ 61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 62

НУБіП України

ВСТУП

В наш час, стала вельми популярна концепція Інтернет речей. Одним з варіантів реалізації цієї ідеї стала можливість створення «Розумних будинків». Технології «розумного будинку» на сьогоднішній день вже перестали сприйматися як якась «розкіш». Навпаки, можливості «розумного будинку» стали нагальною потребою практично в будь-якому сучасному помешканні. Адже сучасні будинки - це не просто поєднання красивого інтер'єру з функціональними меблями. Це ще й місце взаємодії сучасних комунікаційних і інженерних систем, які потребують зручної та ефективної системи управління - системи «розумного будинку».

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

1 СИСТМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Особливості розроблюваної системи (табл. 1)

Функції ІС	Реалізація виконання
Безпечна передача даних	Spring
Безпечне збереження даних	MySQL
Відправлення команд від клієнта до пристрою	Spring
Відображення даних в зручному вигляді	HTML,CSS,JS
Інформування клієнта про позаштатні ситуації	JavaMail API
Виконання сценаріїв -створених клієнтом	Java
Можливість розпізнавання команд від нових пристроїв.	Spring
Зчитування показників	Arduino
Відправлення команд на виконання від ІС до мікроконтролеру	RabbitMQ

Таблиця 1. Особливості розроблюваної системи.

1.1 Постановка задачі

Завдання полягає в розробці інформаційної системи для керування пристроями та моніторингу даних у межах об’єкту на якому вони встановлені.

ІС має забезпечити виконання спектра послуг, що необхідні для комфортного проживання або проінформує власника про позаштатну ситуацію та мати тришарову архітектуру.

Реалізувати доступ мікроконтролеру до інформаційних ресурсів розроблюваної системи за допомогою REST підходу. Дані повинні

передаватися у вигляді невеликої кількості стандартних форматів (наприклад, HTML, XML, JSON) ,повинно підтримуватися кешування, відсутність залежності від мережевого прошарку, не повинна зберігатися інформація про стан між парами «запит-відповідь».

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

1.2 Огляд інформаційних джерел та існуючих рішень

Myhome Legrand

Система була презентована в 2011 році і прийшла на зміну системі Розумний будинок InOne by Legrand.Компанія "Интелсити" є авторизованим інсталятором системи MyHome Legrand.

Переваги використання:

- Керування сценаріями
- Бездротові пристрої радіо ZigBee
- Додавання простих функцій
- Звукова і відео трансляція
- Технічна сигналізація
- Домова і охоронна сигналізація
- Програмування та контроль споживання електроенергії, води, газу
- Мультимедійна панель - централізоване управління всіма функціями My Home
- Віддалений контроль і управління
- Додаток для віддаленого управління системою My Home за допомогою iPhone і iPodTouch

Home Assistant

Платформа домашньої автоматизації із відкритим вихідним кодом, що працює на Python 3.

Переваги використання :

- Безкоштовне та відкрите джерело
- Оптимізовано для вбудованих пристроїв, таких як Raspberry Pi
- 100% локальна домашня автоматизація

НУБіП України

- Легка установка та оновлення (на базі ResinOS та Docker)
- Управління користувальницьким інтерфейсом, інтегрованим в Home Assistant
- Легко створювати і відновлювати повні резервні копії всієї вашої конфігурації
- Легко встановіть багато популярних надбудов, включаючи Google Assistant, шифрування за допомогою Let's Encrypt та динамічного DNS через Duck DNS.

CLAP - розумний будинок українського виробництва

CLAP складається з координаційного центру «Хаба», панелі управління, об'єднаної з IP-відеодомофоном, терморегулятором для контролю опалювальних приладів і набору датчиків, які вимірюють температуру в квартирі, реагують на затоплення, сигналізують про рух і аналізують інші параметри. Більшість пристроїв використовують контролери STM різних модифікацій.

Стек технологій:

Було використано: Java, Spring (MVC, Boot, Integration, Security), JMS, реляційні та нереляційні бази даних, Docker, Selenium, Cucumber. На сервері - мікросервісная архітектура.

Переваги використання :

- Замість безлічі габаритних приладів із дротами, які нервують та вимагають уваги, – єдина, безперебійно діюча, непомітна система.
- CLAP легко встановлюється, не пошкоджуючи простір та не привертаючи до себе надто багато уваги.
- CLAP працює абсолютно непомітно, функціонуючи за допомогою бездротового покриття.

НУБіП України

- Можливість змінювати, доповнювати, удосконалювати систему за власним бажанням зі свого смартфона.
- Технологія забезпечує безперешкодну сумісність та ідеальну синхронізацію усіх пристроїв, підключених до системи.
- Широкий діапазон зв'язку, резервні акумулятори та своєчасні реакції забезпечують надвисоку надійність системи.
- Контроль аварійних ситуацій
- Технологія інтелектуального керування CLAP контролює, раціонально розподіляє та зменшує споживання енергії.

1.3 Моделювання предметної області

1.3.1 Методології системного аналізу.

Системний підхід – форма методологічного знання, зв'язана із дослідженням та створенням об'єктів у вигляді систем.

Ієрархічність пізнання, що базується на таких поняттях як абстракція та декомпозиція , що вимагає багаторівневого вивчення предмета:

- вивчення самого предмета – "власний" рівень;
- вивчення цього ж предмета як елемента більш широкої системи – "вищестоящий" рівень;
- вивчення цього предмета в співвідношенні зі складовими даний предмет елементами – "нижчий" рівень.

Системний підхід вимагає розглядати проблему не ізольовано, а в єдності зв'язків з системою , частиною якої являється об'єкт , в якому було виявлено данну проблему , осягати сутність кожного зв'язку та окремого елемента, розмежувати цілі на загальні та приватні.

1.3.2 Структурно-функціональне моделювання

Методологія SADT розроблена Дугласом Россом .

На її основі розроблена, зокрема, відома методологія IDEF0 (Icam DEFinition), яка є основною частиною програми ICAM (Інтеграція комп'ютерних та промислових технологій), що проводиться за ініціативою ВПС США.

Під методологією SADT мається на увазі набір необхідних методів, правил і процедур, що застосовуються для побудови функціональної моделі об'єкта будь-якої предметної області. Функціональна модель SADT відображає об'єкт у вигляді його функціональної структури, тобто вироблені їм дії та їх взаємодію у вигляді зв'язків.

Основні елементи цієї методології базуються на наступних концепціях:

- Графічне представлення блочного моделювання.
- Строгість та точність.

Правила SADT включають:

- Обмеження кількості блоків на кожному рівні декомпозиції (правило 3-6 блоків);
- Діаграми повинні бути зв’язаними (номери блоків);
- Назви та мітки - унікальні(відсутність повторюваних імен);
- Синтаксичні правила для графіки (блоків і дуг);
- Поділ входів та управлінь (правило визначення ролі даних);
- Відділення функції від організації, мається на увазі забезпечення виключення впливу організаційної структури на функціональну модель.

1.3.3 Об`єктно-орієнтоване моделювання

ООМ - це методологія , що з'єднує в собі процес об'єктної декомпозиції і прийоми подання фізичних, логічних, динамічних та статичних моделей проектованої системи.

НУБіП України

Дану методологію застосовують :

- **UML**
- **Патерни (шаблони)проектування**

Патерни проектування– це підходи, алгоритми та правила вирішення основних задач ,при створенні програмного забезпечення, що були перевірені часом . Вони не залежать від певної мови програмування і можуть бути застосованими в основному незалежно від неї. Патерни відрізняються за рівнем складності та рівнем деталізованості проектуємої системи. На додачу, їх можна поділити на три групи, щодо розв'язуваних проблем.

UML (уніфікована мова моделювання) – це стандартна нотація візуального моделювання програмних систем. На сьогоднішній день, вона підтримується багатьма об'єктно-орієнтованими інструментальними системами (CASE-системами). UML – уніфікована мова, вона: не залежить від методології, що використовується при розробці проекту; може підтримувати будь-яку об'єктно-орієнтовану мову програмування. В UML можна змістовно описувати класи, об'єкти й компоненти з різних галузей, що можуть істотно відрізнятись між собою.

Діаграма прецедентів – це UML діаграма , на якій відображається взаємодія прецедентів та акторів. Виступає основою для подальшої декомпозиції системи у вигляді різних логічних та фізичних моделей.

НУБіП України

В розробленій діаграмі акторами виступають користувачі системи (Мешканець , Адміністратор) а також мікроконтролер (плакат №1).

- Адміністратор - керування даними по клієнтам та пристроям.
- Мешканець – Моніторинг даних , створення команд на виконання ,отримання повідомлень про критичні ситуації.
- Мікроконтролер - виконання команд , зчитування та передача показників .

Усі прецеденти , що були відображені на плакаті , певним чином пов'язані з якимось із акторів. Наприклад, перед тим як мешканець зможе отримати необхідні дані для моніторингу , мікроконтролер , повинен зчитати дані у реальному часі з датчика та відправити їх .

Сценарій прецеденту «Відправлення показників»:

- Передумови:
  - Показник був отриманий з датчика мікроконтролером
- Головний потік:
  - Перевірка з’єднання з веб сервером A1;
  - Створення HTTP Post запиту.
  - Відправлення запиту
  - Отримання запиту сервером.
  - Валідація A2
  - Відправлення даних до сервіс прошарку
  - Додавання до даних дати та часу
  - Відправлення даних до прошарку маніпуляції з даними
  - Збереження даних у БД .
- Альтернативні потоки:
  - A1 – з’єднання відсутнє.
  - A2 – дані не пройшли валідацію.

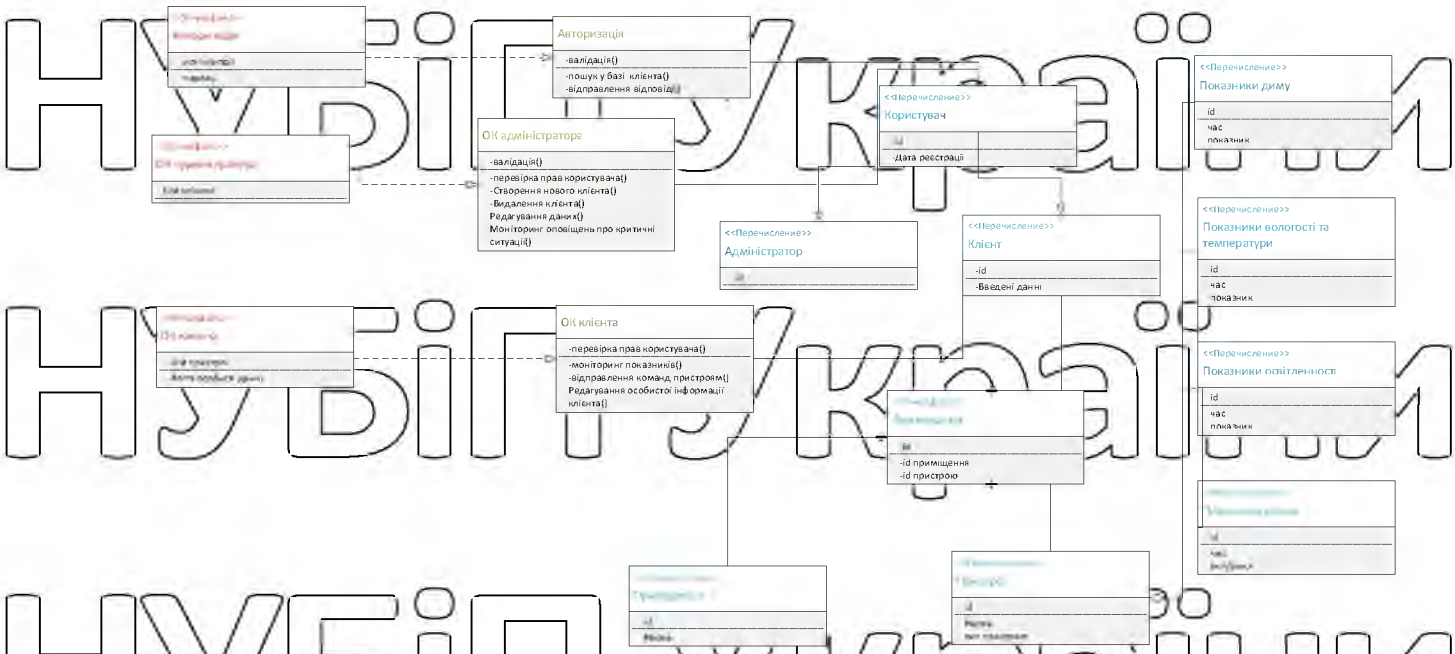
НУБіП України

Діаграма класів

Діаграма класів - це тип UML діаграм, які настіше зазвичай використовуються для моделювання об'єктно-орієнтованих систем. Елементи діаграми пов'язані структурними зв'язками. Цю діаграму обирають для відображення статичної структури створюємої моделі системи у вигляді класів об'єктно – орієнтованого програмування .

На діаграмі зазвичай зображують такі типи класів :

- Інтерфейсиї,
- Контролюючі,
- Сутності



Опис діаграми класів. В діаграмі класів (рис.) виділено такі класи:

1. Класи сутності:

- Пристрої - користувацький тип даних для збереження інформації по пристроям

- Показники - користувацький тип даних для збереження інформації по показникам
- Помешкання - користувацький тип даних для збереження інформації по помешканням
- Користувач - користувацький тип даних для збереження інформації по користувачам
- Адміністратор - користувацький тип даних для збереження інформації по адміністраторам

2. Контролюючі класи :

- Авторизація - забезпечення виконання функціоналу авторизації
- ОК адміністратора, користувача – забезпечення виконання функціоналу інтерфейсу.

3. Інтерфейсні класи:

- Авторизація – форма авторизації
- ОК клієнта - інтерфейс взаємодії користувача з системою
- ОК адміністратора - інтерфейс взаємодії адміністратора з системою

Класи перебувають між собою у різних відношеннях. А саме:

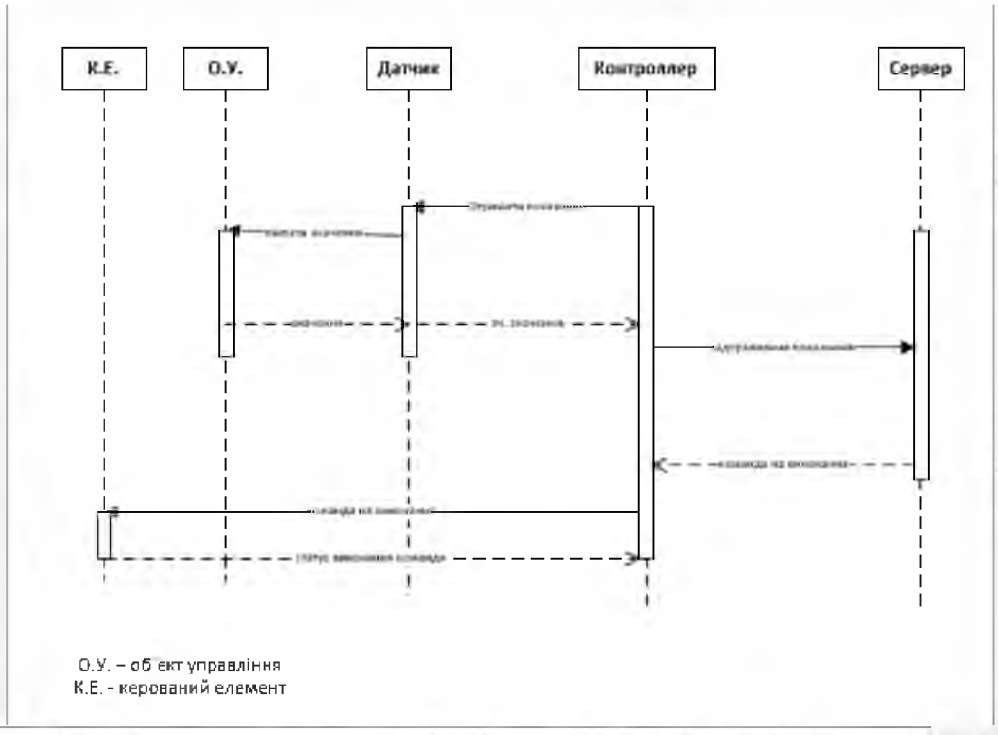
Асоціація – взаємна залежність між об'єктами різних класів, кожен з яких є рівноправним членом залежності. Для асоціації може позначатися кількість екземплярів об'єктів кожного класу, які беруть участь у зв'язку (0 - якщо жодного, 1 - якщо один, * - якщо багато). Можуть вказуватися мінімальна й максимальна кількість, наприклад, 0..1..* означає, що на відповідному кінці асоціації може не бути жодного екземпляра, бути один або багато.

Діаграма послідовності.

На діаграмі послідовності зображають взаємо обмін повідомленнями між об'єктами, упорядкованих по тимчасовій осі. На діаграмі послідовності зображені часові послідовності повідомлень.

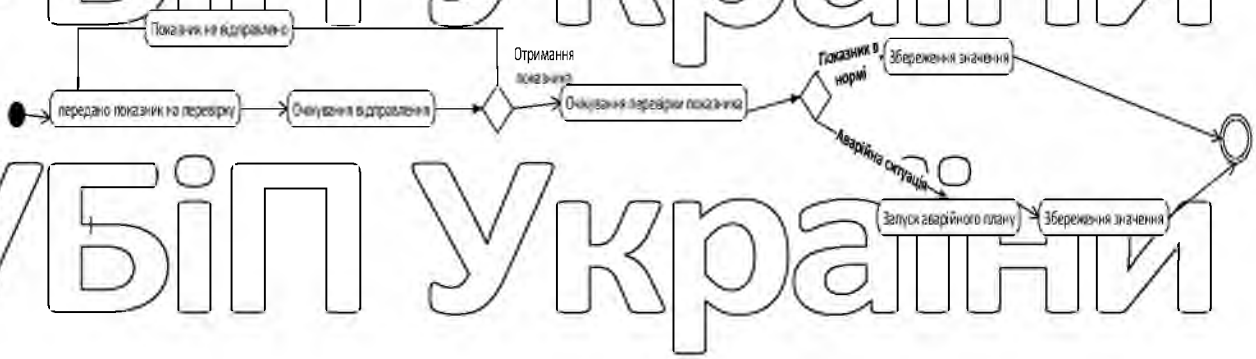
НУБіП України

Діаграма послідовності може описувати всі можливі сценарії або ж описувати один конкретний сценарій. На рис. представлено діаграму послідовностей,



Діаграма станів

Діаграма станів - відображає зміну станів об'єкта у часі. Являється деякими видом графу, що представляє собою автомат. Дуги графа слугують для позначення переходів зі стану в стан.



2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Побудова логічної моделі даних

2.1.1 Загальні відомості про ER-діаграму

ER – діаграма(Діаграма сутностей і зв’язків) – це графічне зображення множин сутностей , перелік їх атрибутів та зв’язків між даними сутностями.

Сутність – Це клас однотипних об'єктів, інформація про яких враховується під час створення моделі.

Атрибут сутності –це характеристика яка виділена назвою, що є частиною стану сутності.

Зв'язок – це певна асоціація між двома сутностями.

Під час побудови моделі даних за допомогою ERD, її елементам належать такі властивості:

- Усі сутності повинні мати унікальні назви в межах моделі;
- сутність має один, або декілька атрибутів, що або належать сутності, або має посилання на іншу сутність даної моделі;
- кожен атрибут сутності повинен мати унікальну назву в межах сутності, до якої він належить;
- атрибути з різних сутностей можуть мати однакові імена, але ці атрибуті повинні інтерпритуватися однаково , і атрибути з разною назвою не можуть інтерпритуватися однаково
- сутність володіє одним або декількома атрибутами, що однозначно ідентифікують кожний екземпляр сутності;
- Будь-яка сутність може мати довільну кількість взаємозв’язків з іншими сутностями моделі.

НУБіП України

2.1.2 Побудова ER-діаграми.

Для моделювання ER-діаграми, було використано програмний засіб що має назву – Erwin Data Modeler

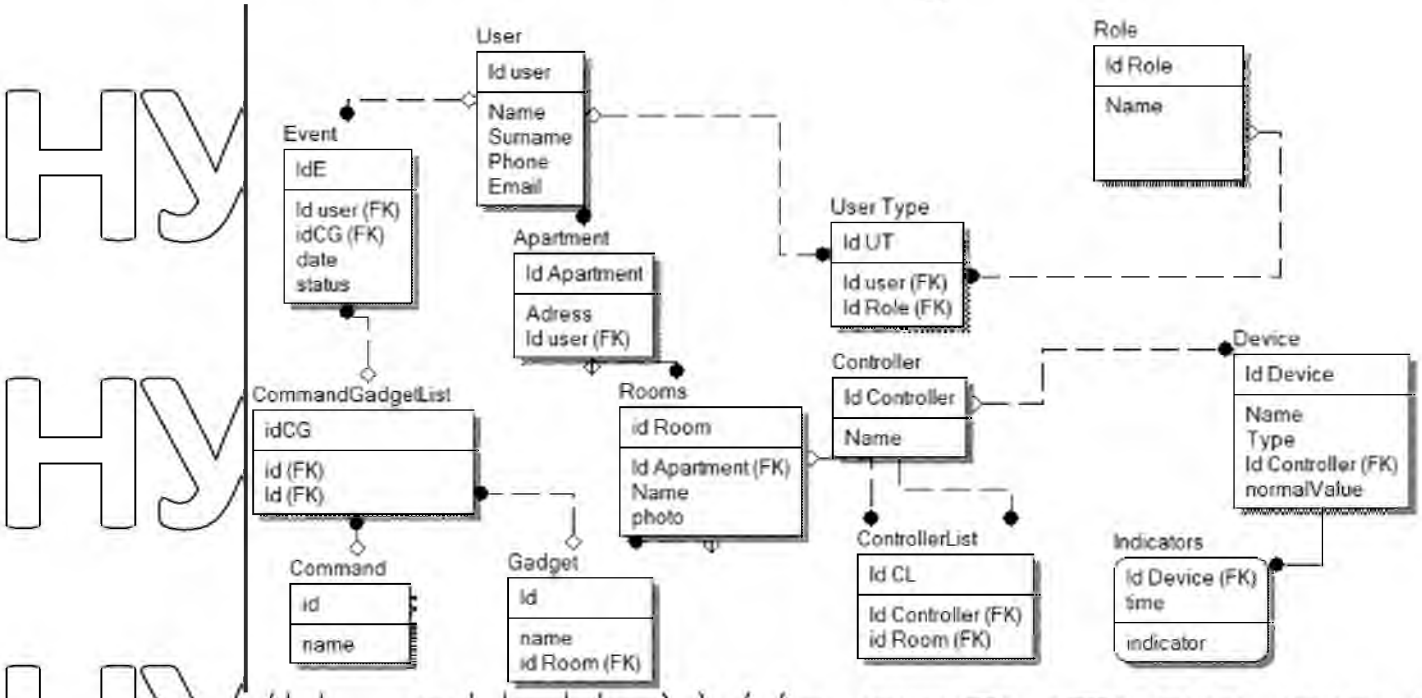


Рис. Модель «сутність-зв'язок»

Розглянемо сутності та їх атрибути у побудованій логічній моделі даних:

Сутності:

1. Користувач
2. Роль
3. Список користувачів та їх ролей
4. Подія
5. Команда
6. Пристрій керування
7. Список команд та пристроїв керування
8. Помешкання
9. Кімнати
10. Мікроконтролери

НУБіП України

11.Список кімнат та мікроконтролерів

12.Пристрої моніторингу

13.Показники пристроїв моніторингу

Атрибути:

Користувач

1. Ідентифікаційний номер далі ІН (Long)
2. Ім’я (String)
3. Прізвище (String)
4. Email(String)
5. Телефон(String)

Роль

1. ІН (Long)
2. Назва

Список користувачів та їх ролей

1. ІН (Long)
2. ІН користувача (Long)
3. ІН ролі (Long)

Подія

1. ІН (Long)
2. ІН Списку команд та пристроїв керування(Long)
3. Статус виконання (Boolean)
4. Дата виконання події (Date)

Список команд та пристроїв керування

1. ІН (Long)
2. ІН пристрою керування (Long)

НУБіП України

3. ІН команди (Long)

Команда

1. ІН (Long)
2. Назва команди (String)

Пристрій керування

1. ІН(Long)
2. Назва
3. ІН кімнати

Помешкання

1. ІН(Long)
2. Адреса (String)
3. ІН користувача(Long)

Кімната

1. ІН(Long)
2. Назва(String)
3. Фотографія(Blob)

Контролер

1. ІН(Long)
2. Назва(String)
3. Ip адреса(String)

Пристрій моніторингу

1. ІН(Long)
2. Назва(String)
3. Тип(String)
4. Мінімально допустимий показник(Int)
5. Максимально допустимий показник(Int)

НУБіП України

6. ІН Контролеру(Long)

Показник

1. ІН(Long)
2. ІН контролеру(Long)
3. Дата (Date)
4. Значення показника(double)

Список кімнат та мікроконтролерів

1. ІН (Long)
2. ІН контролеру(Long)
3. ІН кімнати(Long)

2.2 Вибір інструментарію для створення БД та роботи з нею

2.2.1 Вибір СУБД

MySQL — вільна система керування реляційними базами даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам.

Основні переваги MySQL.

Серед основних переваг MySQL відзначають наступні:

- *Масштабованість*. MySQL пыдтримує роботу з масивами даних великих розмірів. Відповідно до документаці MySQL, деякі БД, що використовуються компанією, зберігають до 50 млн. записів.
- *Переносність*. MySQL працює на різних платформах, в число яких входять Linux, Windows, Unix, Solaris, OS/2, Mac ОС. Окрім того, MySQL працює на платформах різних типів.
- *Зв'язаність*. MySQL має мережеву структуру. До MySQL можна отримати доступ кільком користувачам одночасно. MySQL має цілий ряд програмних інтерфейсів додатків (Application Programming Interface

–API), які надають можливість встановлювати з’єднання з MySQL із додатків, написаних на таких мовах як Java,C, C++, Perl, PHP, Python.

- *Безпека.* MySQL має систему контроля доступу до даних, шифрує дані при їх передачі
- *Швидкість функціонування.*
- *Відкритий код.*

Оскільки для написання диплому був обраний Java стек технологій який забезпечує кросплатформеність доцільно обрати MySQL як систему керування базою даних.

2.2.2 Мова маніпулювання даними SQL.

Згідно із 5 правилом Кодда (Правило повноти підмови маніпулювання даними):

Система управління реляційними базами даних має підтримувати хоча б одну реляційну мову, яка

а) має лінійний синтаксис,

б) може використовуватись інтерактивно і в прикладних програмах,

в) підтримує операції визначення даних, визначення уявлень, маніпулювання даними (інтерактивні та програмні), обмежувачі цілісності, управління доступом та операції управління транзакціями (begin, commit і rollback).

Саме такою мовою виступає SQL.

SQL (Structured query language — мова структурованих запитів) — декларативна мова програмування для створення взаємодії між користувачем та базою даних, що застосовується для створення та виконання запитів, модифікації і керування реляційними БД, створення схеми бази даних і її підтримка, система контролю доступу до бази даних.

2.3 Побудова структури БД

Hibernate — засіб відображення між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java. Hibernate є вільним програмним забезпеченням, яке поширюється на умовах GNU Lesser General Public License. Hibernate надає легкий для використання каркас (фреймворк) для відображення між об'єктно-орієнтованою моделлю даних і традиційною реляційною базою даних. Hibernate дозволяє розробляти стійкі класи об'єктно-орієнтованих ідіом з успадкуванням, поліморфізмом, асоціацією, композицією.

```
@Entity
@Table
public class Appartment {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="aId")
    private long appId;
    @ManyToOne
    @JoinColumn(name="uId")
    private User user;

    @Column(name = "adress")
    private String adress;

    @Column(name = "city")
    private String city;

    @OneToMany (cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    @JoinColumn(name="aId")
    private List<Room> roomList= new ArrayList<>();

    public Appartment(){};

    public Appartment(User user, String adress) {
        this.user = user;
        this.adress = adress;
    }
```

Рис. Опис таблиці «Помешкання»

3 ПРИКЛАДНЕ ПРОГРАМНЕ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування технічного забезпечення системи

Інформаційна система «розумний дім » має певну кількість модулів, які включають технічну інфраструктуру, програмні компоненти та інформаційну базу даних з показниками моніторингу. Система має клієнт-серверну архітектуру з веб-клієнтом. Датчики підключаються до мікроконтролеру , який в свою чергу взаємодіє із контейнером сервлетів (Tomcat) в який завантажено Spring додаток.

Рис. Схема взаємодії мікроконтрлера з сервером

Spring Framework — фреймворк з відкритим кодом та контейнери з підтримкою інверсії управління для платформи Java.

Розроблений Spring веб додаток дозволяє зчитувати показники в реальному часі , керувати пристроями , отримувати звіт у форматі excel . При необхідності, функціонал системи може бути збільшено.

Для відправлення показників мікроклімату було використано плату Ардуіно з Ethernet шилдом .

Для отримання команд с серверу та їх виконання було використано плату RobotDyn.

НУБіП України

Для запуску ІС «Розумний дім» було вирішено використати одноплатні комп'ютери, серед яких найкраще себе зарекомендували Orange pi lite (під керуванням Armbian) та Raspberry pi 3 (під керуванням Raspbian)

Arduino - апаратна обчислювальна платформа, основними компонентами якої є плата вводу/виводу та середовище розробки на мові Processing/Wiring. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері

Arduino складається з мікроконтролера Atmel, елементів обв'язки для програмування та інтеграції з іншими пристроями. На багатьох платах наявний лінійний стабілізатор напруги +5В або +3,3В. Тактування здійснюється на частоті 84, 16 або 8 МГц кварцовим резонатором. У мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен.

Програми(скетчі) Arduino пишуться на мові програмування C++. Arduino IDE завантажується разом із бібліотекою програм, ,яка дозволяє робити стандартні операції вводу/виводу набагато простіше. Користувачам слід описати дві функції.

Start – функція , що виконується перед початком програми

Loop – функція, що виконується циклічно весь час роботи скетчу.

Плати Arduino нараховують певну кількість різних мікроконтролерів тож для написання даної роботи було обрано Arduino як саму оптимальну

Arduino Uno R3, виконана на базі процесора ATmega328P, чіп FTDI було змінено на ATmega16U2. Це дозволило збільшити швидкість передачі даних; під Linux або Mac не потрібні драйвера (inf-файл для Windows необхідний, який входить до складу Arduino IDE), і дозволяє під час

перепрошивки використовувати контролер як клавіатуру, мишу або джойстик.

Рис. Arduino Uno R3

Характеристики

- Мікроконтролер ATmega328P
- Частота 16 МГц
- 32Кб Flash Memory
- 14 цифрових входів / виходів (6 з них ШІМ)
- 6 аналогових входів
- Напруга живлення 7-12 В
- Розмір: 68 х 53 х 15 мм

Оскільки Arduino UNO не має самостійного підключення до інтернету. До неї розроблено плату розширення w5100 (Ethernet Shield)

НУБіП України

Плата розширення Arduino Ethernet побудована на базі Ethernet-контролера Wiznet W5100 ,який дозволяє Ардуіно підключатися до мережі Інтернет. Wiznet W5100 підтримує мережеві протоколи (IP) і надає можливість використовувати як з TCP, так і з UDP-протоколами. При цьому можливо керувати до чотирьох відкритих сокет-з'єднань одночасно.

Рис W5100

Характеристики:

- Робоча напруга - 5В (подається від плати Ардуіно)
- Ethernet-контролер: W5100 зі вбудованим буфером об'ємом 16 КБ
- Швидкість з'єднання: 10/100 Мбіт / с
- Взаємодія з Ардуіно здійснюється через інтерфейс SPI
- Остання версія плати розширення підтримує стандартний вигляд, прийнятий в моделі Arduino UNO R3.
- Плата розширення Ethernet підтримує технологію "Power over Ethernet" і має стандартний роз'єм RJ-45 з вбудованою гальванічною розв'язкою.

DHT11 — датчик температури і вологості, комплекс з каліброваним цифровим виходом сигналу. Даний датчик дозволяє вимірювати вологість

резистивного типу і компонент вимірювання NTC температури, завдяки відключенню до 8-розрядного мікроконтролеру, тим самим пропускаючи швидке реагування та задовільну якість.

Рис. DHT 11

Характеристики:

- Визначення вологості 20-95% з 5% точністю
- Визначення температури 0-50 град. з точністю 2 град.
- Частота опитування не більше 1 Гц (не більше одного разу в 1 сек.)
- Розміри 15.5мм х 12мм х 5.5мм
- вивеження з відстанню між ніжками 0.1

MQ2 – датчик диму:

НУБіП України

Рис. Датчик диму mq2

Характеристики

- Газ - Горючий газ, дим
- Діапазон чутливості 300-10000 ppm
- Rs (опір чутливого елемента) 1 ... 20 кОм 50ppm толуол
- Газ, для якого нормується датчик Ізобутан, 1000ppm
- Час відгуку ≤ 10с
- Чутливість (R в повітрі) / (R в присутності характерного газу) ≥ 5с
- Rh (опір нагрівача) 31Ω ± 3Ω
- Ih (струм нагрівача) ≤180мА
- Vh (напруга нагрівача) 5В ± 0,2 В
- Ph (потужність нагрівача) ≤900мВт
- Vc (напруга схеми) ≤24В
- Стандартні робочі умови
  - Температура: -10 ~ + 50 ° C,
  - вологість: ≤95% RH,
  - концентрація кисню: 21% (стандартні умови)
- Умови зберігання
  - Температура: -20 ~ + 70 ° C,

- вологість: <70% RH
- Конфігурація A або B (металевий або пластиковий корпус)

Фотоелектричний аналоговий сенсор GL5516

Рис. Фотоелектричний аналоговий сенсор GL5516

Характеристики:

- Максимальна напруга: 150 В
- Максимальна потужність: 90 мВт
- Світловий опір (10 люкс): 5-10 кому
- Нічний опір (0 люкс): 0.5 МОм
- Спектральний пік: 540 нм
- Час відгуку: 30 мс
- Робоча температура: від -30 до +70 ° С

RobotDyn – плата, яка включає в себе мікроконтролер ATmega328P і

Wi-Fi контролер ESP8266, флеш-пам'ять 8 Мбіт. Це дозволяє використовувати даний MCU як класичну UNO, а також як "модернізовану" - разом з Wi-Fi, який теж може працювати самостійно. Плата виконана за схемою Arduino UNO. Підключити до ПК плату можна за допомогою microUSB роз'єму, забезпечує підключення USB-UART конвертер CH340. Сама плата в форм-

факторі UNO, що робить її ще більш універсальною для підключення модулів.

Рис. RobotDyn

Характеристики:

- Мікроконтролер: ATmega328P
- USB-UART перетворювач: CH340G
- Wi-Fi контролер: ESP8266
- Флеш-пам'ять: 8 Mbit
- Робоча напруга: 5V
- Вхідна напруга: 7 - 12V
- Вхідна напруга (рекомендований): 7 - 9V
- Цифрові канали введення / виводу: 14
- Аналогові канали введення: 6
- Постійний струм через вхід / вихід: 40 mA
- Постійний струм для виведення 3,3 В: 50 mA
- ППЗУ (Flash Memory) 32 КБ з яких 0,5 КБ використовується завантажувачем

НУБіП України

- ОЗУ (SRAM): 2кб
- ПЗУ (EEPROM): 1кб
- Тактова частота: 16MHz

Orange Pi Lite – це одноплатний комп'ютер який працює на базі 4-х ядерного процесора Allwinner H3, має графічний процесор Mali400MP2 і 512Mb оперативної пам'яті DDR3. Новинка має вбудований Wi-Fi з можливістю підключення зовнішньої антени, IR портом. Має 2 USB порту 2.0, microUSB OTG, HDMI роз'єм, мікрофон, кнопку ввімкнення, світлодіоди відображення статусу роботи. Для нормальної роботи необхідно 5V2A.

Рис. Orange Pi Lite

Характеристики:

- CPU: Allwinner H3 ARM Cortex-A7 Quad Core 1.2 GHz
- Кількість ядер процесора: 4

НУБіП України

- GPU:Mali400MP2
- RAM:512 MB
- Сховище:microSD
- Мережеві можливості: WiFi 2.4G 150 mb / s
- Відео:HDMI
- USB порти:2
- Аудіо: HDMI
- GPIO:40 pins
- Живлення:5V 2A
- Габарити:69 x 48 мм

На даний одноплатний комп’ютер було встановлено систему Armbian: Armbian- це дистрибутив Linux, призначений для самих різних пристроїв ARM.

Архітектура ARM (спочатку *Advanced RISC Machine* — поліпшена RISC машина, попередник *Acorn RISC Machine*) — 32-бітна RISC архітектура процесорів, яку розробила компанія ARM Limited. Широко застосовується у розробці портативних пристроїв. Головною причиною цього є використання енергозберігаючих технологій. Саме тому ця архітектура домінує у пристроях, головною ідеєю яких є енергозбереження

Raspberry Pi 3 Model B має 64-х бітовим Soc BCM2837 з 4 ядрами Cortex-A53, що працює на частоті 1.2GHz (900MHz у попередній версії). Поєднання збільшення тактової частоти на 33% з різними архітектурними поліпшеннями забезпечить збільшення продуктивності на 50-60% в 32-бітному режимі в порівнянні з Raspberry Pi 2, або приблизно в десять разів більше в порівнянні з Raspberry Pi 1. Для зв'язку на плату були додані Wi-Fi і Bluetoth 4.1 модулі. Поріг вхідного струму збільшений до 2.5A, що дозволить підключати енергоємні пристрої до USB портів Raspberry. Форм-фактор Raspberry Pi 3 Model B практично ідентичний Raspberry B + / 2B, що дозволить

використовувати корпусу, аксесуари і іншу периферію від цих моделей. Єдина відмінність - зміна розташування світлодіодів, які перемістилися на сторону слота SD карти.

Рис. Raspberry Pi 3 Model B

3.2 Вибір інструментарію для створення програмного забезпечення

Java — об'єктно-орієнтована мова програмування, створена у 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року розробкою мовою займається «Oracle», придбала «Sun Microsystems». В офіційній реалізації Java класи компілюються у байт-код, який після цього інтерпретується Java JVM (віртуальна машина) для кожної конкретної платформи чим і забезпечується кросплатформеність. Станом на 2018 рік використовується 8 версія JDK

Зміни у 8 JDK:

- Lambda Expressions, нова функція мови. Можливість функціонального стилю програмування Лямбда-вирази дозволяють виразити екземпляри

«одномодових» інтерфейсів (називаються функціональними інтерфейсами) більш компактно.

- Методи за замовченням дозволяють додавати нові функціональні можливості до інтерфейсів бібліотек та забезпечувати двонарну сумісність з кодом, написаним для старих версій цих інтерфейсів.
- Stream API – дозволяє працювати з колекціями у функціональному стилі

- Нові алгоритми шифрування даних
- Atomic типи даних для роботи з багатопоточністю
- Інші зміни.

Spring – java фреймворк для побудови додатків.

Основні особливості Spring Framework можуть бути використані будь-яким додатком Java, але є розширення для створення веб-додатків на платформі Java EE.

Spring Framework складається з певних модулів, які надають можливість скористатися широким спектром послуг:

- Контейнер Інверсії управління. Завдяки певній конфігурації додатків управління життєвим циклом об'єктів Java, здійснюється через Інверсію управління
- Аспектно-орієнтоване програмування:
- Доступ до даних: робота з реляційною СУБД на платформі Java з використанням JDBC драйверу і об'єктно-реляційного відображення .
- Управління транзакціями: об'єднання кількох API, керування транзакціями та координинації операцій для Java-об'єктів
- MVC (Model-View-Controller): програмний каркас на основі HTTP сервлета, що забезпечує створення веб-додатків і веб-служб RESTful.

НУБіП України

- Аутентифікація і авторизація: налаштовувані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за допомогою підпроекту Spring Security
- Віддалене керування: конфігураційний вплив і управління Java-об'єктами для місцевої (локальної) або віддаленої конфігурації через JMX
- Тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів

Фреймворк Spring MVC забезпечує архітектуру паттерна Model - View - Controller за допомогою слабо пов'язаних готових компонентів. Патерн MVC розділяє аспекти додатку (логіку введення, бізнес-логіку і логіку UI), забезпечуючи при цьому вільний зв'язок між ними.

- Model (Модель) інкапсулює (об'єднує) дані програми, в цілому вони будуть складатися з POJO (Java-об'єктів, або бінов).
- View (Відображення, Вид) відповідає за відображення даних Моделі, - як правило, генеруючи HTML.
- Controller (Контролер) обробляє запит користувача, створює відповідну Model і передає її для відображення в View.

Вся логіка роботи Spring MVC побудована навколо DispatcherServlet, який приймає і обробляє всі HTTP-запити (з UI) і відповіді на них.

Послідовність подій, відповідна вхідному HTTP-запиту:

- Після отримання HTTP-запиту DispatcherServlet звертається до інтерфейсу HandlerMapping, який визначає, який Контролер повинен бути викликаний, після чого, відправляє запит в потрібний Контролер.
- Контролер приймає запит і викликає відповідний службовий метод, заснований на GET або POST. Викликаний метод визначає дані Моделі, засновані на певній бізнес-логікою і повертає в DispatcherServlet ім'я View.

- За допомогою інтерфейсу ViewResolver DispatcherServlet визначає, який View потрібно використовувати на підставі отриманого імені.
- Після того, як View створений, DispatcherServlet відправляє дані Моделі у вигляді атрибутів в View, який в кінцевому підсумку відображається в браузері.

Spring Security - це Java / JavaEE framework, що надає механізми побудови систем аутентифікації та авторизації, а також інші можливості забезпечення безпеки для корпоративних додатків, створених за допомогою Spring framework.

Ключові об'єкти контексту Spring Security:

- SecurityContextHolder, в ньому міститься інформація про поточний контекст безпеки програми, який включає в себе детальну інформацію про користувача (Principal),який в даний час працює над додатком. За замовчуванням SecurityContextHolder використовує ThreadLocal для зберігання інформації, що означає, що контекст безпеки завжди доступний для методів що виконуються в межах одного потоку.
- SecurityContext, містить об'єкт Authentication і в разі необхідності інформацію системи безпеки, пов'язану із запитом від користувача.
- Authentication представляє користувача (Principal) з точки зору Spring Security.
- GrantedAuthority відображає права доступу які можуть базуватися на певній ролі, що були видані користувачу в масштабі всього додатку, такі дозволи як наприклад ROLE_ANONYMOUS, ROLE_USER, ROLE_ADMIN.
- UserDetails надає необхідну інформацію для побудови об'єкта Authentication з DAO об'єктів додатка або інших джерел даних системи безпеки. Об'єкт UserDetails має такі атрибути як: ім'я користувача, пароль, логічні атрибути: isAccountNonExpired, isAccountNonLocked,

НУБіП України

isCredentialsNonExpired, isEnabled і Collection - прав (ролей) користувача.

HTML —мова розмітки гіпертексту веб-сторінок. Більшість веб-сторінок було створено за допомогою цієї мови(або XHTML). Документ HTML оброблюється браузером та показується на екрані у зручному для користувача вигляді. Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів (у оригіналі element), що забезпечують функціональність документа: текстові фрагменти із заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання і т.д. Елементи HTML оголошуються за допомогою команд розмітки, званих тегами (від англійського tag - ярлик). Всі HTML-теги, що зустрічаються в тексті документа інтерпретуються браузером при відображенні документа.

CSS— спеціальна мова, що використовується для відображення сторінок, написаних мовами розмітки даних. Найбільш часто CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. CSS використовується для того щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS). Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори та ін.

JavaScript — мова програмування для створення інтерактивних Web-сторінок. динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення

сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

JSP (**Java Server Pages**) — технологія, що дозволяє веб-розробникам динамічно генерувати HTML, XML та інші веб-сторінки. Робота над JSP розпочалась в 1997 році. Згодом JSP було включено у склад Java EE — програмної платформи для програмування веб-додатків. Технологія дозволяє вставляти Java-код, в статичний вміст сторінки. Також можуть використовуватись бібліотеки JSP тегів для вставки їх в JSP-сторінки. Сторінки компілюються JSP-компілятором в сервлети, які є Java-класами, і виконуються на сервері. Сервлети також можуть бути написані розробником, не використовуючи JSP-сторінки. Ці технології доповнюють одна одну.

JSP — одна із високопродуктивних технологій, оскільки весь код сторінки транслюється в java-код сервлету за допомогою компілятора JSP сторінок (напр. Jasper), а потім компілюється в байт-код віртуальної машини java (JVM).

JSTL (Стандартна бібліотека тегів JSP) - розширення специфікації JSP, що додає бібліотеку JSP тегів для загальних потреб і містить умовна обробка, розбір XML даних, підтримка інтернаціоналізації та ін.

НУБіП України

AJAX — не концепція використання декількох споріднених технологій. AJAX-підхід до розробки, використовується користувачами інтерфейсів, та об'єднує декілька основних методів і прийомів:

- Використання DHTML для динамічної зміни змісту сторінки.
- Використання XMLHttpRequest для звернення до сервера «на льоту», не перезавантажуючи всю сторінку повністю
- альтернативний метод — динамічне підвантаження коду JavaScript в тег <SCRIPT> з використанням DOM, що здійснюється із використанням формату JSON)
- динамічне створення дочірніх фреймів

Використання цих підходів дозволяє створювати набагато зручніші веб-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем. AJAX — асинхронний, тому користувач може переглядати далі контент сайту, поки сервер все ще обробляє запит. Браузер не перезавантажує web-сторінку і дані посилаються на сервер без візуального підтвердження (крім випадків, коли ми самі захочемо показати процес з'єднання з сервером).

Apache Maven - фреймворк для автоматизованого збирання проектів на основі опису їх структури в файлах на мові POM, що є підмножиною XML.

Життєвий цикл Maven проекту - це список фаз, що визначають порядок виконання дій при побудові. Життєвий цикл Maven містить три незалежних фази виконання

clean - життєвий цикл для очищення проекту. Містить наступні фази:

- pre-clean
- clean
- post-clean

НУБіП України

default - основний життєвий цикл, що містить наступні фази:

- validate - виконується перевірка, чи є структура проекту повну й правильну.
- generate-sources
- process-sources
- generate-resources
- process-resources
- compile - компілюються вихідні тексти.
- process-test-sources
- process-test-resources
- test-compile
- test - зібраний код тестується заздалегідь підготовленим набором тестів.
- package - упаковка відкомпільованих класів та інших ресурсів. Наприклад, в JAR-файл.
- integration-test - програмне забезпечення повністю або певні його великі модулі перевіряються за допомогою інтеграційного тестування. Перевіряється взаємодія між складовими частинами програмного продукту.
- install - установка програмного забезпечення в локальному Maven-репозиторії, щоб зробити його доступним для інших проектів користувача.
- deploy - стабільна версія програмного забезпечення поширюється на віддалений Maven-репозиторій, щоб зробити його доступним для інших користувачів.

Site —життєвий цикл генерації звітної документації. Нараховує такі фази:

1. pre-site
2. site
3. post-site

НУБіП України

4. site-deploy

Maven базується на plugin-архітектурі, яка дозволяє застосовувати плагіни для різних завдань (compile, test, build, deploy, cheekstyle, pmd, scp-transfer) для даного проекту, без необхідності їх в явному вигляді інсталювати.

Apache POI - це набір кросплатформених API для Java, що надають доступ до читання/запису файлів у форматах таких офісних додатків Microsoft Office, як Word, PowerPoint, Excel, Outlook, Visio і Publisher. Вихідний код Apache POI поширюється під ліцензією Apache License 2.0.

Останній реліз дозволяє працювати з файлами OLE2 (xls, .doc, .ppt) і OOXML (.xlsx, .docx, .pptx), надає низькорівневий API для підтримки Open Packaging Conventions (за допомогою openxml4j), високорівневі API для електронних таблиць Excel, документів Word і презентацій PowerPoint, підтримку повідомлень і вкладених файлів в Outlook, конвертери для документів Excel і Word в формати HTML і XSLF-FO.

Apache Tomcat — контейнер сервлетів, створений Apache Software Foundation. Повністю написаний на мові програмування Java та використовує специфікацію сервлетів та Java Server Pages від Oracle, що є стандартом при розробці веб-застосунків на Java.

Tomcat надає можливість запускати додатки розроблені під веб платформу, містить ряд програм для самоконфігурації.

Tomcat може використовуватися як: незалежний, самостійний веб-сервер, як сервер контенту в поєднанні з веб-сервером Apache HTTP Server, також в серверах додатків JBoss і GlassFish його використовують як контейнер сервлетів.

Компоненти:

Tomcat версії 4.x був випущений з Jasper (перепроектований механізм JSP), Catalina (перепроектований контейнер сервлетів) і Coyote (стек HTTP).

- Catalina - контейнер сервлетів Tomcat'a, який реалізує специфікацію сервлетів Servlet API. Servlet API є базовим поняттям для всіх технологій Java, що застосовуються у Web та надає можливість динамічної генерації будь-якого web-контенту та дозволяє використовувати усі бібліотеки, що доступні для java.
- Coyote - частина стеку HTTP Tomcat'a, який підтримує протокол HTTP 1.1 ,що використовує веб-сервери або контейнер додатків. Coyote прослуховує вхідні з'єднання ,що надходять на певний TCP порту сервера, пересилає запити в механізм Tomcat для обробки запитів та надсилає відповідь назад клієнту.
- Jasper - технологія JSP Tomcat. Tomcat 5.x використовує Jasper 2, який є реалізацією специфікації JavaServer Pages 2.0 Sun Microsystems. Jasper аналізує JSP-файли, для подальшої їх компіляції в Java код, у вигляді сервлетів (які можуть бути оброблені за допомогою Catalina). Під час виконання, Jasper може автоматично виявляти зміни JSP-файлу та перекомпілювати його.

RabbitMQ — платформа, що реалізує систему обміну повідомленнями між компонентами програмної системи на основі стандарту AMQP (Advanced Message Queuing Protocol). Також можливе використання стандарту MQTT.
Основні відомості

- Використовує стандарт AMQP (Advanced Message Queuing Protocol).
- Підтримується горизонтальне маштабування для побудови кластерної архітектури.
- Підтримує збереження даних на диск
- Підтримки протоколів HTTP, XMPP та STOMP
- Є реалізація клієнтів для доступу до RabbitMQ для ряду мов програмування: Java, .NET, Perl, Python, Ruby, PHP та ін.

НУБіП України

- Існують різноманітні плагіни (такі як плагін для моніторингу та управління через HTTP або веб-інтерфейс або плагін «Shovel» для передачі повідомлень між брокерами)

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

4 РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ

4.1 Тестування системи

Для тестування програмного продукту використовується програма Postman:

Спроба авторизації без вказаного параметру _csrf-token (рис).

Рис. Помилка авторизації

Тестування OpenWeatherApi , яке дозволяє отримати погоду у місті за вказаним помешканням (рис.):

Рис.18 Тестування OpenWeatherApi

Тестування запиту для введення дати відліку щоб отримати звітню інформацію за певний проміжок часу та відповідь (рис.19):

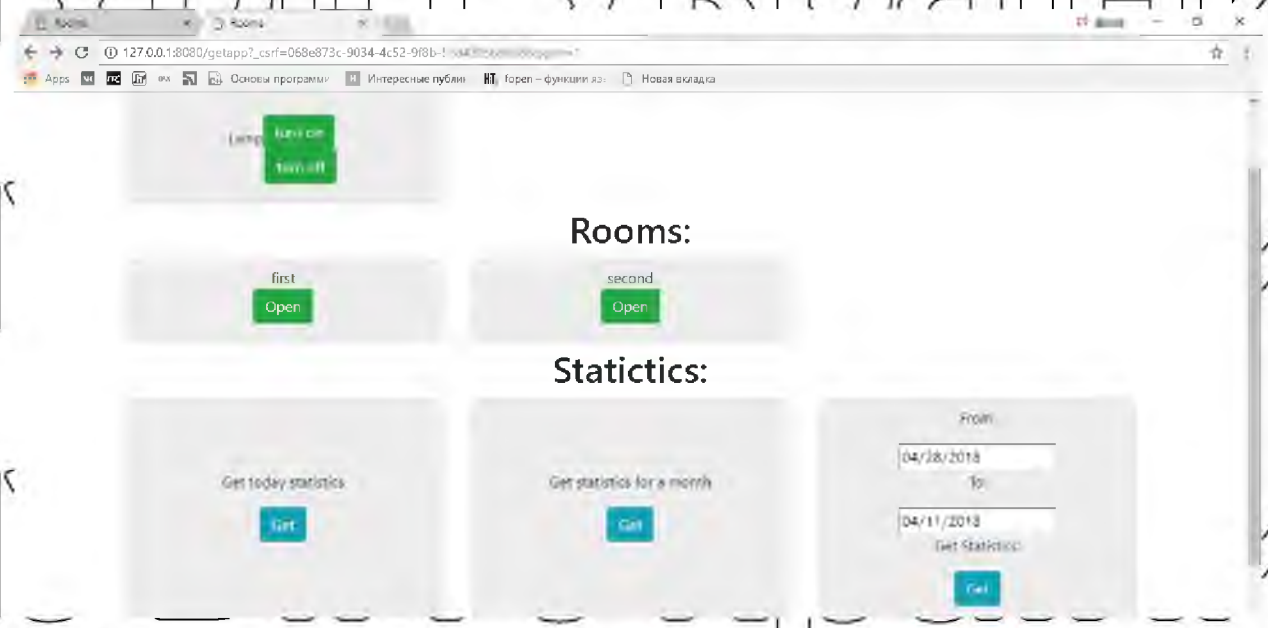


Рис. Введення дати , яка не настала на момент тестування:

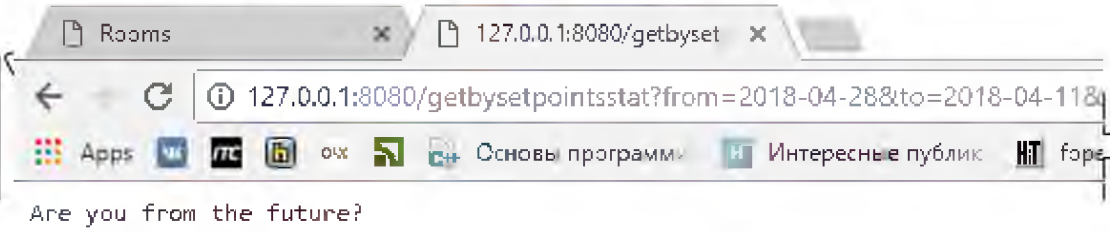


Рис. Відповідь сервера на неправильно введену дату.

4.2.2 Вимоги до апаратного забезпечення

4.2.2.1 Вимоги до апаратного забезпечення сервера

Для забезпечення діяльності системи з боку серверної частини необхідно використовувати сервери моделі "IBM CICS/390" або ж "HP ProLiant 10" (4Гб ОЗУ та більше, 240Гб постійної пам'яті з швидкістю читання від 100Мбіт/с).

4.2.2.2 Вимоги до апаратного забезпечення клієнту користувача

Для пристрою користувача мінімальними є наступні вимоги до системи:

- Пристрій з можливістю встановлення браузера, з підтримкою HTML 5 -;

4.2.3 Вимоги до програмного забезпечення

4.2.3.1 Вимоги до системного програмного забезпечення

Для запуску сервера потрібно встановити наступне програмне забезпечення:

- Операційна система: Windows Server 2008 та новіші, Windows (XP, 7, 8, 10), Ubuntu, Mac OS;
- Jdk 8
- Tomcat 8
- Rabbit MQ
- MySQL

Для доступу до системи з боку клієнта достатньо встановити одну з операційних систем: Windows 7-8-10; Linux, Mac OS. Та один із web-браузерів: Google Chrome, Mozilla Firefox, IE 7 та новіші, Opera.

НУБіП України

4.2.3.2 Склад інеталяційного пакету для встановлення розробленої системи

- ❖ Файли для відображення сторінок на стороні клієнта:
  - Web/pages/*
- ❖ Файли для стилізації сторінок:
  - Web/assets/css/*
- ❖ Файли для клієнтської обробки сторінок та динамічного відображення:
  - Web/assets/js/*
- ❖ Файли серверної бізнес-логіки:
  - Src/main/java/controller/*
  - Src/main/java/service/*
  - Src/main/java/dao/*
  - Src/main/java/entity/*
  - Src/main/java/utils/*
  - Src/main/java/config/*
- ❖ Файли інтерфейсу з базою даних:
  - Src/main/java/config/AppConfig.class

НУБіП України

5 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

5.1 Оцінка, час реалізації та календарний план

5.1.1 Оцінка конкурентноспроможності

Для оцінки конкурентоспроможності продукту, що розробляється необхідно провести аналіз і порівняння з обраним аналогом за функціональним призначенням, основними технічними та експлуатаційними параметрами області застосування. Подібний аналіз здійснюється за допомогою оцінки експлуатаційно-технічного рівня продукту, що розробляється.

Експлуатаційно-технічний рівень розроблюваного продукту - це узагальнена характеристика його експлуатаційних властивостей , можливостей, ступеня новизни, що є основою якості продукту. Для визначення продукту можна використовувати індекс експлуатаційно-технічного рівня $J_{ЕТУ}$, який розраховується як сума приватних індексів , куди входять показники якості програмного продукту. Для обліку значимості окремих параметрів застосовується бально-індексний метод.

Тоді $J_{ЕТУ} = \sum_{j=1}^{n} B_j \times X_j$,де:

- $J_{ЕТУ}$ - - комплексний показник якості продукту за групою показників
- n - число розглянутих показників
- B_j - коефіцієнт вагомості j -го показника в долях одиниці, призначуваний відповідно до потреб організації-замовника програмного продукту
- X_j - відносний показник якості, встановлюваний експертним шляхом з обраної шкалою оцінювання

НУБіП України

Оцінка конкурентоспроможності в порівнянні з аналогами

Таблиця 1. Порівняльна характеристика наданих послуг

Послуги	Салон Умный дом	Home Smart Company	Secur	Smart House
Охорона і безпека	+	+	+	+
Керування мікрокліматом	+	+	+	-
Управління освітленістю	+	+	+	+
Медіа центр.	-	-	+	+
Гнучке керування і використання сценаріїв	+	-	+	+
Можливість розширення	+	-	+	+
Ціна ($)	1499-45000	Залежність від кількості замовлених модулей (Ціна одного модуля від 30)	Залежність від кількості замовлених модулей (Ціна одного модуля від 57)	Залежність від кількості замовлених модулей (Ціна одного модуля від 1)

Оберемо аналог який надає найбільшу кількість послуг(«Салон Умный дом»)

Оскільки послуги мають однакову важливість для розробки вирішальним критерієм є ціна:

Шкала оцінювання X_j [0;1]

Таблиця 2. Порівняльна характеристика проекту та головного аналогу

Показники якості	Коефіцієнт вагомості, B_j	Аналог		Проект	
		X_j	$B_j \times X_j$	X_j	$B_j \times X_j$
Охорона і безпека	0,1	1	0,1	0.7	0,07
Керування мікрокліматом	0,1	1	0,1	0.5	0,05

Управління освітленістю	0,1	1	0,1	1	0,1
Медіа центр.	0,1	0,7	0,07	0,3	0,03
Гнучке керування і використання сценаріїв	0.1	1	0,1	1	0,1
Можливість розширення	0,1	1	0,1	1	0,1
Ціна ($)	0,4	0,5	0,2	1	0,4

Спільний показник якості:
Проект: 0,78
Аналог:0,77

Коєф. Технічного рівня=0,78/0,77=1,013

Так як коефіцієнт більше 1, то розробка проекту з технічної точки зору виправдана.

3.1.2 Планування комплексу робіт розробки та оцінка трудоємності

Трудомісткість робіт визначається з урахуванням терміну закінчення робіт, обсягу виконуваних функцій, обраного середовища програмування.

Для визначення очікуваної тривалості роботи застосовується формула

$$T_{оч} = \frac{t_{мін} + 4t_{нв} + t_{макс}}{6} \quad (3.1)$$

де

- $t_{мін}$ - найкоротша тривалість даної роботи (оптимістична оцінка);
- $t_{макс}$ - найбільша тривалість роботи (песимістична оцінка);
- $t_{нв}$ - найбільш ймовірна тривалість роботи (реалістична оцінка).

Для розробки було задіяно три робітники: керівник проекту, інженер-програміст мобільних додатків та інженер-програміст.

В таблиці 3.2 наведено оцінку трудоємності робіт по проекту:

Таблиця 3 - Оцінка трудомісткості окремих видів робіт

Виды Работ	Оптимистическая оценка, t_{min}	Реалистическая оценка, $t_{нв}$	Пессимистическая оценка, t_{max}	Ожидаемая продолжительность работы, $T_{ож}$
1.1	15	17	18	17
1.2	7	8	9	8
1.3	12	14	16	14
2.1	2	3	5	3
2.2	5	7	8	7
3.1	8	14	16	13
3.2	19	21	23	21
3.3	2	5	7	5
3.4	3	4	8	5
4.1	2	3	4	3
4.2	3	4	5	4
4.3	11	16	17	15

Перелік та пояснення всіх видів робіт проекту наведено в таблиці 3:

Таблиця 4 – Комплекс робіт по розробці проекту

Зміст робіт	Виконавці	Тривалість Дні	Загрузка	
			дні	%
1. Підготовка процесу розробки і аналіз вимог				
1.1 Дослідження та обгрунтування розробки				
1.1.1 Постановка завдання	Керівник	1	1	100
	Програміст		1	100
1.1.2 Збір вихідних даних	Керівник	4	2	50
	Програміст		4	100
1.2 Пошук аналогів				
1.2.1 Аналіз існуючих методів вирішення задачі і програмних засобів	Керівник	6		
	Програміст		6	100
1.2.2 Обгрунтування принципової необхідності розробки	Керівник	2	2	50
	Програміст		2	100

1.3 Аналіз вимог				
1.3.1 Визначення і аналіз вимог до проектованої програми	Керівник	3	1	33
	Програміст		3	100
1.3.2 Визначення структури вхідних та вихідних даних	Керівник	5	1	20
	Програміст		5	100
1.3.3 Вибір технічних і програмних засобів реалізації	Керівник	3	1	33
	Програміст		3	100
1.3.4 Узгодження і затвердження технічного завдання	Керівник	3	1	33
	Програміст		3	100
Разом з етапу 1	Керівник	**27**	**9**	
	Програміст		**28**	
2. Проектування				
2.1 Проектування програмної архітектури	Програміст	4	4	100
2.2 Технічне проектування компонентів програми	Програміст	8	8	100
Разом з етапу 2	Програміст	**12**	**12**	
3. Програмування та тестування програмних модулів				
3.1 Програмування модулів в вибраному середовищі програмування	Програміст	13	13	100
3.2 Тестування програмних модулів	Керівник	21		
	Програміст		21	100
3.3 Складання і випробування програми	Керівник	8	4	50
	Програміст		8	100
3.4 Аналіз результатів випробувань	Керівник	5	1	20

НУБіП України

	Програміст		5	100
Разом з етапу 3	Керівник	47	**5**	
	Програміст		**47**	
4. Оформлення документації				
4.1 Проведення розрахунків показників безпеки життєдіяльності		3		
	Програміст		3	100
4.2 Проведення економічних розрахунків		3		
	Програміст		3	100
4.3 Оформлення пояснювальної записки	Керівник	15	15	100
Разом з етапу 4	Керівник	**22**	**20**	
	Програміст		**6**	
Разом з проекту	Керівник	**108**	**34**	
	Програміст		**108**	

НУБіП України

На основі даних таблиці розроблено календарний графік виконання робіт (табл. 5 , що показує послідовність і взаємозв'язок виконання комплексу робіт (з урахуванням неділі; можна брати без урахування неділі та святкових днів).

Таблиця 5 – календарний графік виконання робіт

Содержание работы	Исполнители	Длительность, дни	График работ
1 Постановка задачі	Керівник	1	20.01.18-20.01.18
	Програміст	1	
2 Збір вхідних даних	Керівник	2	21.01.18-22.01.18
	Програміст	4	21.01.18-24.01.18
3 Аналіз існуючих методів вирішення задачі і програмних засобів	Програміст	6	25.01.18-1.02.18
4Обгрунтування принципової необхідності розробки	Керівник	2	2.02.018-3.02.18
	Програміст	2	2.02.018-3.02.18
5 Визначення і аналіз вимог до проектованої програми	Керівник	1	4.02.18-4.02.18
	Програміст	3	4.02.18- 6.02.18
6 Визначення структури вхідних та вихідних даних	Керівник	1	7.02.18-7.02.18
	Програміст	5	7.02.18-11.02.18
7 Вибір технічних і програмних засобів реалізації	Керівник	1	12.02.18-12.02.18
	Програміст	3	12.02.18-14.02.18
8 Узгодження і затвердження технічного завдання	Керівник	1	15.02.18 – 15.02.18
	Програміст	3	15.02.18 –17.02.18
9 Проектування програмної архітектури	Програміст	4	18.02.18 –21.02.18
10 Технічне проектування компонентів програми	Програміст	8	22.02.07– 29.02.18
11 Програмування модулів в вибраному середовищі програмування	Програміст	13	30.02.07 – 11.03.18
12 Тестування програмних модулів	Програміст	21	11.03.18 – 02.04.18
13 Складання і випробування програми	Керівник	2	03.04.18 – 04.04.18
	Програміст	5	03.04.18 – 07.04.18
14 Аналіз результатів випробувань	Керівник	1	08.04.18 – 08.04.18
	Програміст	5	08.04.18 – 12.04.18

15 Проведення розрахунків показників безпеки життєдіяльності	Программист	3	13.04.18 – 15.04.18
16 Проведення економічних розрахунків	Програміст	4	16.04.18 – 19.04.18
17 Оформлення пояснювальної записки	Керівник	15	20.04.18 – 4.05.18

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

5.2 Розрахунок витрат на розробку проекту

5.2.1 Капітальні вкладення на реалізацію проекту

Капітальні вклади, пов'язані з автоматизацією обробки інформації розраховуються за формулою

$$К=К_п+К_р, \quad (3.2)$$

де $К_п$ – капітальні вклади на проектування, грн.;

$К_р$ – капітальні вклади на реалізацію проекту, грн.

Предвиробничі затрати представляють з себе одночасні витрати на розробку забезпечуваних чи функціональних систем або елементів на всіх етапах проектування, а також витрати на їх вдосконалення, тобто на проведення обстеження і обробку матеріалів дослідження, розробку технічного завдання, розробку технічного і робочого проекту системи та її досвідченого інтегрування. Сюди включаються витрати на розробку алгоритмів і програм, вартість розробок по прив'язці типових проектних рішень (ТПР) і пакетів прикладних програм (ППП) до конкретного об'єкту автоматизації.

Сумарні витрати на проектування системи і її розробку і налагодження на комп'ютері визначаються за формулою

$$К_П = ((1+W_d)(1+W_c)+W_н)\sum_{i=1}^{m} З_{oi} + C_M + M_в, \quad (3.3)$$

де m – кількість працівників, які беруть участь в розробці проекту;

$З_{oi}$ – *витрати на основну заробітну плату працівника і-ї категорії*, грн.;

W_d – коефіцієнт, що враховує додаткову заробітну плату в частках до основної заробітної плати (W_d *= 0,4 і складається з коефіцієнту відпускних, рівного 0,1, і районного коефіцієнта - 0,3*);

W_c – коефіцієнт, що враховує відрахування на соціальні потреби, в частках до суми основної та додаткової заробітної плати розробників (W_c *=0,262*: *страхові внески в Пенсійний фонд в долях одиниці – 0,2, страхові внески в ФСС – 0,029, страхові внески в ФОМС – 0,031, страхові внески на виробничий травматизм– 0,002*);

$W_н$ – коефіцієнт, що враховує накладні витрати організації, в частках до основної заробітної плати розробників (*приймається по фактичним даним, $W_н$ =0,6*);

C_M – витрати на матеріали;

$M_в$ – витрати на матеріали машинного часу.

Витрати на основну заробітну плату працівника *і*-ї категорії:

$$З_{oi} = З_{днi} t_i, \quad (3.4)$$

де $З_{днi}$ – середньоденна заробітна плата працівника і-ї категорії, грн./дн.;

t_i – кількість днів, відпрацьованих працівником і-ї категорії.

Витрати часу на розробку системи по кожному виконавцю приймаються, виходячи з його завантаження за календарним графіком виконання робіт.

Розрахунок основної заробітної плати розробників проекту наведено в таблиці 3.5 з розрахунку, що в місяці в середньому 21 робочий день.

Таблиця 3.5 – Основна заробітна плата розробників

Таблиця 6 Витрати на робітників

Посада	Оклад, грн...	Середньо добова ставка, грн...	Ч.Г.	ОЗП, грн.
Керівник	8189,74	400	34	13600
Програміст	1717,95	100	108	10800
Разом				24 400

З огляду на те, що проектована інформаційна система повинна бути запрограмована і налагоджена за допомогою комп'ютерів, до сумарних витрат на розробку додаються витрати на використання машинного часу, що обчислюються як:

$$M_в = t_{мв} S_{мч} K_м, \qquad (3,5)$$

де $t_{мв}$ – машинний час комп'ютера, необхідний для розробки програмного продукту; $t_{мв}$ = 108 год.;

$S_{мч}$ – Вартість 1 години машинного часу: $S_{мч}$=12грн./год.;

$K_м$ – коефіцієнт мультипрограмності (показує частку машинного часу, що відводяться безпосередньо на роботу над проектом); $K_м$=1.

Додаємо розрахунок машинного часу:

108*8*12=10368 грн

Таблиця 7 – Витрати на матеріали

Матеріали	Одиниці виміру	Кількість	Ціна за одиницю	Сума, грн.
Монітор	шт.	1	2 545	2 545
Системний блок	шт.	1	10000	10000
Клавіатура	шт.	1	150	150
Миша дротова	шт	1	100	100
Разом				12795

Таблиця 8 – Витрати на розробку

Статьї затрат	Сумма, грн.
Основна ЗП	24 400
Бонуси	10000
Відрахування на соціальні потреби	6168,83
Витрати на матеріали	12795
Витрати на машинний час	10 368
Накладні витрати організації	10090,78
Разом	63 455

Для підтримки проекту необхідно найняти «support» програміста

Таблиця 9– Дані по зп спеціалістів

Посада	Оклад, грн...	Середньо добова ставка,	Ч.Г.	ОЗП, грн.
Программіст	3500	166,66	20	5889,1
Разом				5889,1

Силова енергія: ,2×800×2,25= 360 грн

Таблиця 10 Витрати на підтримку проекту у місяць

Витрати	Сумма, грн.
ЗП програмісту	5889,1
Бонуси	1000
Силова енергія	360
Разом	7249,1

НУБіП України

ВИСНОВКИ

Ставши популярною концепція Інтернет речей, система «розумний дім» стала привертати до себе інтерес громадськості. Проаналізувавши існуючі пропозиції мною був обраний необхідний стек технологій для реалізації більш доступної системи яка забезпечить виконання спектра послуг, що необхідні для комфортного проживання або проінформує власника про позаштатну ситуацію.

Було розроблено інформаційне забезпечення, що задовольняє інформаційні потреби системи. Побудовано модель даних та розроблена відповідна база даних.

Для реалізації інтерфейсу та бізнес-логіки було використано Spring framework.

Для зручності та ефективності динамічного завантаження даних з сервера використовується технологія асинхронного з'єднання з сервером AJAX.

Для графічного представлення сторінок користувача використовується мова гіпертекстової розмітки HTML5. Для оформлення зовнішнього вигляду сторінок та динамічного відображення елементів сторінок використовуються CSS (каскадні таблиці стилів) та JS (JavaScript).

НУБіП України

НУБіП України

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 Kevin Ashton. That ‘Internet of Things’ Thing. In the real world, things matter more than ideas. (англ.). RFID Journal (22 June 2009).

2 Гарингард Сэмюэл Интернет вещей. Будущее уже здесь / Сэмюэл Гарингард, 2017

3 Портал Хабрахабр [Электронный ресурс] – Режим доступа: https://habrahabr.ru/post/259243/

4 Портал Cisco [Электронный ресурс]. – Режим доступа: https://www.cisco.com/c/ru_ru/solutions/internet-of-things/overview.html

5 MySQL documentation [Електронний ресурс]. “MySQL.com” (Oracle 2018). Режим доступу: https://dev.mysql.com/doc/

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України