МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут електронних та інформаційних технологій

Кафедра інформаційних та комп’ютерних систем

**Допущено до захисту**

Завідувач кафедри

к.е.н., доцент Базилевич В.М.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2020 р.

ВИПУСКНА КВАЛІФІКЦІЙНА РОБОТА

за освітньо-професійною програмою бакалавра

РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ З ВИКОРИСТАННЯМ ІДЕНТИФІКАЦІЙНИХ КАРТ

Спеціальність 123 – Комп’ютерна інженерія

Галузь знань 12 – Інформаційні технології

Виконавець:

студент гр. КІ–161

Трухан Антон Володимирович

*(підпис)*

Керівник:

|  |  |  |
| --- | --- | --- |
| *доцент* |  | *к.т.н., доцент* |
| **–––––––––––––––––––––––––––––** |  | **–––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––** |
| *(посада)* |  | *(науковий ступінь, вчене звання)* |
|  |  |  |

Риндич Євгеній Володимирович

*(підпис)*

Чернігів 2020

**Індивідуальне завдання на випускну кваліфікаційну роботу (проекту)**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут електронних та інформаційних технологій

Кафедра інформаційних та комп’ютерних систем

ЗАТВЕРДЖУЮ:

Зав. кафедри

к.е.н., доцент Базилевич В.М.

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_2020г.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ  
 ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ БАКАЛАВРА

Трухана Антона Володимировича

Тема роботи: РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ З ВИКОРИСТАННЯМ ІДЕНТИФІКАЦІЙНИХ КАРТ

Тему затверджено наказом ректора

від " " 2020р. № с\_

1. **Вхідні дані до роботи:**

Кількість найменувань товарів до 1000. Кількість постачальників до 100. Кількість клієнтів до 100. Операційна система Linux. Є робітники із вадами зору (дальтонізм).

1. **Зміст розрахунково-пояснювальної записки:**

Кваліфікаційна робота складається з вступу, основної частини та висновків. Основна частина складається з трьох розділів: «Аналіз задачі створення системи», «Розробка системи» та «Реалізація системи».

1. **Демонстраційні матеріали:**

7 слайдів для презентації роботи

АНОТАЦІЯ

TODO

ЗМІСТ

[ВСТУП 5](#_Toc39342560)

[1 АНАЛІЗ ЗАДАЧІ СТВОРЕННЯ СИСТЕМИ 6](#_Toc39342561)

[1.1 Аналіз систем контролю та управління доступом 6](#_Toc39342562)

[1.1.1 Огляд систем контролю і управління доступом 6](#_Toc39342563)

[1.1.2 Основні різновиди систем контролю та управління доступом 6](#_Toc39342564)

[1.2 Аналіз вимог до системи контролю доступу 6](#_Toc39342565)

[1.3 Постановка задачі на розробку КС 6](#_Toc39342566)

[2 РОЗРОБКА СИСТЕМИ 7](#_Toc39342567)

[2.1 Вибір технічних засобів побудови системи 7](#_Toc39342568)

[2.1.1 Технічні засоби для побудови пристрою контролю доступу 7](#_Toc39342569)

[2.1.2 Технічні засоби для побудови серверного застосунку 8](#_Toc39342570)

[2.1.3 Технічні засоби для побудови графічного інтерфейсу 10](#_Toc39342571)

[2.2 Архітектура системи контролю доступу 11](#_Toc39342572)

[2.2.1 Архітектура 11](#_Toc39342573)

[2.3 Розробка програмної підсистеми 11](#_Toc39342574)

[3 РЕАЛІЗАЦІЯ СИСТЕМИ 12](#_Toc39342575)

[ВИСНОВКИ 13](#_Toc39342576)

[ПЕРЕЛІК ПОСИЛАНЬ 14](#_Toc39342577)

ВСТУП

TODO

1 АНАЛІЗ ЗАДАЧІ СТВОРЕННЯ СИСТЕМИ

* 1. Аналіз систем контролю та управління доступом

1.1.1 Огляд систем контролю і управління доступом

Система контроля доступу – це сукупність різних технічних або апаратних видів захисту, яка має мету обмеження и реєстрування фізичних або технічних об’єктів на певній території через пункти контролю: двері, ворота, тощо.

TODO

1.1.2 Основні різновиди систем контролю та управління доступом

TODO

* 1. Аналіз вимог до системи контролю доступу
  2. Постановка задачі на розробку КС

1. РОЗРОБКА СИСТЕМИ
   1. Вибір технічних засобів побудови системи
      1. Технічні засоби для побудови пристрою контролю доступу

Для складення списку технічних засобів апаратної частини необхідно виділити основні вимоги для пристрою контролю доступу.

Пристрою контролю доступу повинен відповідати наступним вимогам:

* Пристрій повинен мати деякий контролер для оброблювання інформації
* Пристрій повинен мати доступ до локально мережі
* Пристрій повинен мати можливість відправляти запити на головний сервер
* Пристрій повинен мати базову авторизацію для внутрішнього веб серверу
* Пристрій повинен мати можливість приймати запити від головного серверу
* Пристрій повинен мати можливість зчитувати ідентифікаційну картку
* Пристрій повинен мати індикацію яка буду інформувати про те чи отримав користувач доступ чи ні
* Пристрій повинен мати можливість зберігати інформацію в флеш-пам’яті

Контролер пристрою повинен мати наступні модулі:

* Контролер
* Модуль доступу до локальної мережі
* Модуль зчитування RFID-міток
* Модуль індикація
* Модуль відмикання механізму замикання
* Модуль окремого живлення для механізму замикання

Виходячи с таких вимог пристрій контролю доступу може буди спроектований за допомогою наступних пристроїв, модулів:

1. Контролер: Nodemcu v3 Lolin.

Це плата яка дає можливість розробляти проекти на базі чипу ESP8266 [1], який представляє з себе WIFI модуль який не дуже вимогливий до живлення. Головна перевага цієї плата в тому що вона дозволяє спростити розробку для пристроїв які повинні мати доступ до мережі так як в неї вже вистроєний WIFI модуль та є внутрішня пам'ять для зберігання даних в розмірі до 4 Мбайт, а також є можливість підключати плату через USB для перепрограмування. Наступна перевага цієї плати є в тому що вона є повністю сумісною с усіма існуючими бібліотеками для Arduino, що дозволяє з легкістю програмувати цей контролер за допомогою вже існуючих засобів для програмування на Arduino.

Для програмування контролеру ESP8266 було мову програмування C++ так як ця мова дозволяє використовувати ООП для побудови системи, що спрощує розділення програмної частини на окремі рівні відповідальності.

1. Модуль зчитування RFID-міток: RFID-RC522

Це модуль який дозволяє з легкістю зчитувати RFID-мітки та передавати данні до контролеру. Модуль випромінює електромагнітні хвилі певної частоти, при зчитуванні мітка відправляє у відповідь інформація про унікальний ідентифікатор, який має кожна мітка, також данні пам’яті та інші данні.

1. Модуль індикація: модуль с світлодіодом SMD-5050 RGB з спільним катодом

Цей модуль дозволяє отримати різнокольорове світло яке необхідно для індикації. Перевагами модулю є те що в нього все катоди виводяться в один і для повного з’єднання с контролером необхідно лише 4 сполуки.

1. Модуль відмикання механізму замикання: модуль IRF520

Модуль представляє з себе силовий ключ з польовим транзистором, який дозволяє контролювати живлення для пристроїв з окремим живленням високої напруги.

1. Модуль окремого живлення для механізму замикання: модуль MT3608

Модуль представляє з себе регульований підвищувач напруги з вихідною напругою до 28В і максимальним струмом до 2А. Вхідна напруга завжди має бути нижче вихідного.

* + 1. Технічні засоби для побудови серверного застосунку

Серверний застосунок повинен відповідати наступним вимогам:

* Застосунок повинен працювати на операційних системах Windows та Linux
* Розробка застосунка повинна бути на мові програмування яка підтримує ООП модель проектування
* Застосунок повинен мати веб сервер який надає можливістю керувати всією систему
* Застосунок повинен мати можливість керувати пристроями контролю доступу

Виходячи з вимог для застосунку було вибрано асинхронне Nodejs оточення яке з легкістю дозволяє створювати масштабні застосунки [2]. Екосистема цього середовища дуже велика і це дуже спрощує створення застосунків, так як більшість інструментів вже є в наявності і не потрібно реалізовувати самостійно їх. Nodejs дозволяє створювати застосунки за допомогою мови javascript, але ця мова э не суворо типізованою мовою, а також можна з легкістю допустити помилки при форматуванні коду. Тож для мови написання було вибрано мову програмування typescript. Ця мова програмування є строго типізованою, можна створювати застосунки які базуються на ООП принципах, в порівнянні з javascript в цій мові можна створити біль жорстку архітектуру яка спростить тестування модулів проекту. Ще одна перевага э те що ця мова э тільки настройкою над js це дозволяє застосовувати всі бібліотеки які все є на js.

Після вибору оточення треба вибрати фреймворк для створення веб додатку, було вибрано фреймворк fastify. Цей фреймворк має великі переваги перед іншими, він є дуже простим, а також він є одним із найшвидших фреймворків по швидкості оброблянні запитів і нарешті в нього велика бібліотека додаткових плагінів які можна використовувати разом з ним[3]. Як настройка під цей фреймворк була вибрана бібліотека fastify-decorators яка дозволяє дуже зменшити написання повторюваного коду використовуючи експериментальну функціональність декоратори (спосіб мета-програмування який дозволяє додати деяку функціональність на етапі виконання скриптів до класів або методів [4]) та зосередитися на написанні корисного коду. Ця настройка використовує шаблон проектування “ланцюжок відповідальностей” [5].

Веб сервер для керування системою контролю доступу потребую зберігання інформації про пристрої якими він керує та інформацію про самі ідентифікаційні картки. Для цього було вибрано sql базу даних Mysql 80, це звичайна база даних яка дозволяє зберігати звичайну інформацію та робити зручні вибірки для отримання даних.

Для комунікації серверу з базою даних потрібно створити деякий рівень взаємодії з нею. Щоб спростити реалізацію запитів до бази даних, а також автоматизувати всі дії з нею було вибрано Sequelize ORM (Object-Relational Mapping – з англ. Мови перекладається як об'єктно-реляційна проекція, технологія або метод доступу к реляційній базі даних з ООП мови, в якій для того щоб отримати доступ до даних треба створити клас який буду описувати модель таблиці, після цього ми зможемо з легкістю робити всі звичні операції без написання ручних sql запитів), ця бібліотека базується на так званих “Promise” які є одною із основних частин Nodejs середовища та мови програмування javascript, також ця бібліотека має гану підтримку транзакцій, відношень таблиць, пряме завантаження та ледаче завантаження даних і багато іншого [6]. Для кращої взаємодії бібліотеки Sequelize та мови програмування typescript було вибрано використовувати бібліотеку sequelize-typescript, він дозволяє створювати моделі таблиць бази даних описуючи їх в класах за допомогою декораторів.

* + 1. Технічні засоби для побудови графічного інтерфейсу

Графічний інтерфейс є одною із головних елементів сучасних інфраструктур і для розробки графічної системи треба вибирати інструменти які дозволять з легкістю масштабувати додаток, підтримувати та тестувати. Для створення WEB додатку будо використано фреймворк Angular(це середовище для розробки та платформа для розробки одно сторінкових додатків). Графічні елементи інтерфейсу такі як кнопки, поля вводу, панель навігації та інші компоненти взяті з бібліотеки графічних компоненті angular-material [9], вона має багато специфікованих компонентів поведінка яких відповідає вимогам Material Design(візуальна мова яка об’єднує класичні принципи доброго дизайну с внесками інноваційних технологій і наук[10]).

Мова програмування для веб інтерфейсу - typescript, вибір цієї мову був зроблений з цілю що серверний додаток буде писатися на цій мові і виходячи з цього підтримка та реалізація буде спрощена тим що мови однакові, а також можна робити модулі які будуть однакові як для серверного так і для графічного застосунків.

* 1. Архітектура системи контролю доступу
     1. Архітектура пристрою контролю доступу

Пристрій контролю доступу це пристрій який зчитую інформацію с ідентифікаційних карт(RFID-мітка)

* 1. Розробка програмної підсистеми

3 РЕАЛІЗАЦІЯ СИСТЕМИ

TODO

ВИСНОВКИ

ПЕРЕЛІК ПОСИЛАНЬ

1. Офіційна документація Nodemcu / [Електронний ресурс]. – Режим доступу: https://nodemcu.readthedocs.io
2. Посилання на документацію nodejs / [Електронний ресурс]. – Режим доступу:
3. Fastify official web site / [Електронний ресурс]. – Режим доступу: https://www.fastify.io
4. TypeScript Decorators overview / [Електронний ресурс]. – Режим доступу: https://www.typescriptlang.org/docs/handbook/decorators.html
5. Fastify decorators readme / [Електронний ресурс]. – Режим доступу: https://github.com/L2jLiga/fastify-decorators/blob/next/lib/README.md
6. Sequelize ORM / [Електронний ресурс]. – Режим доступу: https://sequelize.org
7. Sequelize typescript readme / [Електронний ресурс]. – Режим доступу: https://github.com/RobinBuschmann/sequelize-typescript/blob/master/README.md
8. Introduction to the Angular Docs / [Електронний ресурс]. – Режим доступу: https://angular.io/docs
9. Angular material components / [Електронний ресурс]. – Режим доступу: <https://material.angular.io/components/categories>
10. Material design introduction / [Електронний ресурс]. – Режим доступу: <https://material.io/design/introduction>