Spis treści

[1. Wstęp 2](#_Toc56902205)

[1.1 Wprowadzenie 2](#_Toc56902206)

[1.2 Cele pracy 2](#_Toc56902207)

[1.3 2](#_Toc56902208)

[2. Przegląd rynkowych rozwiązań 2](#_Toc56902209)

[3. Wymagania systemu 2](#_Toc56902210)

[3.1 Opis ogólny 2](#_Toc56902211)

[3.2 Wymagania funkcjonalne 2](#_Toc56902212)

[3.2.1 Zarządzanie aplikacją przez administratora 2](#_Toc56902213)

[3.2.2 Zarządzanie aplikacją przez użytkownika 3](#_Toc56902214)

[3.2.3 Rejestrowanie nowych urządzeń 3](#_Toc56902215)

[3.2.4 Informowanie o statusie urządzeń, oraz o przewidywanych zdarzeniach 3](#_Toc56902216)

[3.2.5 Sterowanie urządzeniami za pośrednictwem aplikacji. 4](#_Toc56902217)

[3.2.6 Funkcjonalność aplikacji 4](#_Toc56902218)

[3.3 Wymagania niefunkcjonalne 4](#_Toc56902219)

[4. Analiza technologiczna 4](#_Toc56902220)

[4.1 Azure Azure IoT Hub, Azure SQL Database, Storage account, Logic App. 4](#_Toc56902221)

[4.1.1 Azure IoT Explorer 5](#_Toc56902222)

[4.1.2 Azure Portal 5](#_Toc56902223)

[4.2 Technologie Webowe 5](#_Toc56902224)

[4.2.1 DOTNET 5](#_Toc56902225)

[4.2.2 JavaScript 5](#_Toc56902226)

[4.2.3 HTML 5](#_Toc56902227)

[4.3 Bazy danych 5](#_Toc56902228)

[4.3.1 Entity Framework 5](#_Toc56902229)

[4.3.2 Microsoft SQL Server Management Studio 5](#_Toc56902230)

[4.3.3 Azure Data Studio 5](#_Toc56902231)

[4.4 C# 5](#_Toc56902232)

[4.4.1 ASP.NET MVC 5](#_Toc56902233)

[4.4.2 LINQ 5](#_Toc56902234)

[4.4.3 5](#_Toc56902235)

[4. 5](#_Toc56902236)

# 1. Wstęp

## 1.1 Wprowadzenie

## 1.2 Cele pracy

(przykłądy, o co chodzi z iot bibliografia pokazać co wim w jakich systemach)

# 2. **Przegląd rynkowych rozwiązań**

# 3. Wymagania systemu

## 3.1 Opis ogólny

Aplikacja ma służyć do zarządzania ogromną liczbą urządzeń za pośrednictwem aplikacji webowej. Aplikacja ma być gotowa do zarządzania dużą ilością ekspresów do kawy wyposażonych czujniki umożliwiające zdalne łączenie, ale również powinna być na tyle skalowalna, aby można ją rozszerzyć do administrowania innymi rodzajami urządzeń. System ma być również wyposażony w symulator umożliwiający łączenie się z portalem Azure i wysyłać na niego dane telemetryczne z symulowanego urządzenia, potrzebny do testowania nowych urządzeń wprowadzanych do oferty.

## 3.2 Wymagania funkcjonalne

### 3.2.1 Zarządzanie aplikacją przez administratora

Administrator ma mieć dostęp do danych użytkownika szczególnie do danych kontaktowych, wyłączając hasła, hasła mają być szyfrowane utrudniając dostęp do nieautoryzowanych logowań. Administrator może modyfikować dane związane z użytkownikiem, takich jak ilość możliwych urządzeń, które użytkownik może zarejestrować w usłudze, dodawanie poszczególnych urządzeń w imieniu użytkownika oraz usuwanie ich. Pracownik z uprawnieniami administratora ma dostęp do danych statystycznych dotyczących zużycia zasobów w usłudze Azure.

### 3.2.2 Zarządzanie aplikacją przez użytkownika

Aplikacja ma umożliwiać użytkownikowi swobodne zarządzanie jego urządzeniami i otrzymywanie danych informujących o stanie urządzenia oraz prognozowanych koniecznych czynności w przyszłości.

### 3.2.3 Rejestrowanie nowych urządzeń

Użytkownik może rejestrować nowe urządzenia oraz je usuwać. Do zarządzania swoimi urządzeniami za pomocą wspomnianej usługi wystarczy poprawnie przeprowadzony proces rejestracji lub zalogowania w przypadku posiadania już wcześniej konta, oraz podanie odpowiednich informacji, o swoim urządzeniu

-nazwa modelu

-maksymalna możliwa ilość wody w zbiorniku

-maksymalna waga ziaren w pojemniku

-W przypadku, gdy ekspres nie posiada czujników odpowiadających za sprawdzanie stanu zapełnienia pojemników na

### 3.2.4 Informowanie o statusie urządzeń, oraz o przewidywanych zdarzeniach

Użytkownik ma mieć dostęp obecnego do statusu urządzenia, takich jak stan włączony/wyłączony, czy aktualne działania przeprowadzane przez urządzenia. W przypadku ekspresów do kawy użytkownik ma mieć wgląd do stanu działania przy robieniu nowej kawy, takich jak stan wykonania zadania. Powinna występować możliwość odczytania przyszłych zdarzeń, takich jak ilość cykli pracy urządzenia przed kolejnym opróżnieniem zbiorników na odpady lub uzupełnienia pojemników na wodę lub ziarno.

### 3.2.5 Sterowanie urządzeniami za pośrednictwem aplikacji.

Aplikacja umożliwia sterowanie działaniami urządzeń zarejestrowanych w usłudze. Z poziomu aplikacji możliwe jest włączenie/ wyłączenie urządzenia i wykonanie dostępnej przez urządzenie akcji, takiej jak rozpoczęcie procesu robienia kawy lub płukanie.

### 3.2.6 Funkcjonalność aplikacji

Aplikacja niezalogowanych użytkowników wita oknem umożliwiającym zalogowanie się lub rejestrację.

<schemat działania>

Po zalogowaniu użytkownik może zmienić dane dotyczące swojego konta takie jak nazwa, hasło, lub numer telefonu. Hasła użytkowników są szyfrowane, co oznacza że nikt nie ma dostępu do oryginalnego hasła, co znacząco utrudnia logowanie się na konto użytkownika przez osoby nieodpowiednie.

## 3.3 Wymagania niefunkcjonalne

System jest w stanie sprostać zarządzaniu milionów urządzeń IoT. System powinien przesyłać do platformy webowej oraz odbierać z niej dane telemetryczne w formacie JSON. System powinien korzystać z metod asynchronicznych do aktualizowania informacji w bazie danych dla szybszej reakcji systemu. Aplikacja powinna opierać się o CRUD. System ma być przejrzysty i nieskomplikowany, co ma przyśpieszyć jego działanie. Schemat działania został przedstawiony w diagramie <numer>

# 4. Analiza technologiczna

## 4.1 Azure Azure IoT Hub, Azure SQL Database, Storage account, Logic App.

### 4.1.1 Azure IoT Explorer

### 4.1.2 Azure Portal

## 4.2 Technologie Webowe

### 4.2.1 DOTNET

### 4.2.2 JavaScript

### 4.2.3 HTML

## 4.3 Bazy danych

### 4.3.1 Entity Framework

### 4.3.2 Microsoft SQL Server Management Studio

### 4.3.3 Azure Data Studio

## 4.4 C#

### 4.4.1 ASP.NET MVC

### 4.4.2 LINQ

### 4.4.3

# 4.