

Kierunek: IST

ZESPOŁOWE PRZEDSIĘWZIĘCIE INFORMATYCZNE

Scan and Store System do Zarządzania Inwentarzem z Użyciem Kodów Kreskowych i Kodów QR

Adam Klementowski

Adam Rudnicki

Adam Skowron

Wojciech Skrzypiec

Opiekun pracy: dr inż. Rafał Palak

SPIS TREŚCI

1 Wykaz symboli, oznaczeń i akronimów (opcja)	3
1.1 Symbole	3
1.2 Oznaczenia	3
1.3 Akronimy	3
2 Cel i zakres przedsięwzięcia	4
2.1 Cel	4
2.2 Zakres	4
3 Słownik pojęć (opcja)	5
4 Stan wiedzy w obszarze przedsięwzięcia (opcja)	6
5 Założenia wstępne	7
5.1 Dobór technologii	7
5.2 Ograniczenia	7
6 Specyfikacja wymagań na produkt programowy	8
6.1 Baza danych	8
6.2 Historyjki	8
7 Projekt produktu programowego	10
7.1 Baza danych	10
7.2 Architektura	10
7.3 Pozostałe schematy i diagramy	10
8 Implementacja (opcja)	12
8.1 Warstwa użytkownika - frontend	12
8.2 Warstwa systemowa - backend	13
8.3 Architektura chmurowa	14
9 Demonstracja produktu programowego	16
9.1 Wstęp	16
9.2 Widoki - komputer stacjonarny	16
9.3 Widoki - telefon komórkowy	16
9.4 Opis aplikacji	16

1 WYKAZ SYMBOLI, OZNACZEŃ I AKRONIMÓW (OPCJA)

1.1 Symbole

1.2 Oznaczenia

1.3 Akronimy

- SNS - Scan and Store, Nazwa robocza projektu
- AWS - Amazon Web Services
- PWA - Progressive Web App

2 CEL I ZAKRES PRZEDSIĘWZIĘCIA

2.1 Cel

Celem projektu jest zbadanie dojrzałości technologii Progressive Web Application (PWA) poprzez opracowanie i wdrożenie systemu do zarządzania inventarzem. System będzie wykorzystywał kody kreskowe oraz kody QR do szybszego dostępu do informacji o zasobach i ewentualnej edycji. W ramach przedsięwzięcia zostanie przeprowadzona wstępna analiza technologii PWA, jej możliwości oraz ograniczeń, a także ocena jej przydatności w praktycznych zastosowaniach, takich jak zarządzanie stanem magazynowym.

2.2 Zakres

W fazie planowania projektu ustalono, iż aplikacja końcowa powinna korzystać z funkcjonalności oferowanych przez technologię PWA, co oznacza, że będzie dostępna z poziomu przeglądarki internetowej, ale jednocześnie umożliwiająca instalację na urządzeniu jako aplikacja natywna. Aplikacja ma również umożliwiać zarządzanie stanem magazynu w podstawowym zakresie.

Z funkcjonalności PWA jakie powinny zostać zaimplementowane wybrano dostęp do kamery urządzenia aby skanować kody kreskowe i kody QR w celu szybszego dostępu do informacji na temat produktu lub stanu magazynu. Dostęp do geolokacji podyktowany jest chęcią udoskonalenia i zmniejszenia na podatność błędów wprowadzania odpowiednich danych do danego magazynu. Kolejnym ważnym punktem technologii PWA jest wykorzystanie Service Workera do obsługi trybu online oraz przyśpieszenia wczytywania zasobów (szczególnie zdjęć). Responsywność aplikacji również jest ważnym zagadnieniem. Technologia PWA daje możliwość korzystania z aplikacji nie tylko poprzez przeglądarkę ale również możliwe jest zainstalowanie aplikacji bezpośrednio na smartfonie (Android lub iOS), na komputerze jak i również tablecie.

3 SŁOWNIK POJĘĆ (OPCJA)

- framework - biblioteka narzędzi wspomagających implementację danego rozwiązania w danym języku
- API - interfejs programowy aplikacji (ang. application programming interface) - zestaw poleceń udostępnianych osobom korzystającym z danej aplikacji
- REST - reprezentacyjny transfer stanu (ang. representational state transfer) - zbiór zasad i wytycznych o tym jak budować API aplikacji webowych
- REST API - API trzymające się zasad REST

4 STAN WIEDZY W OBSZARZE PRZEDSIĘWZIĘCIA (OPCJA)

5 ZAŁOŻENIA WSTĘPNE

5.1 Dobór technologii

Najważniejszą technologią dla projektu jest **Progressive Web App** (PWA).

Aby wybrać framework do implementacji warstwy użytkownika (fronendu) zwrócono szczególną uwagę na dobre wsparcie dla technologii PWA i wybrano framework **Angular**.

Warstwę systemową (backend) zaimplementowano przy użyciu frameworku **Spring Boot**, który ma duże wsparcie dla zabezpieczeń (Spring Security) oraz prosty do zaimplementowania zestaw punktów końcowych (endpointów) jako REST API.

Jako bazę danych wybrano **PostgreSQL**.

Całość architektury aplikacji została wdrożona przy pomocy **Terraform HashiCorp** na serwery **Amazon Web Services** (AWS), co niskim kosztem pozwala na przetestowanie różnych rozwiązań bez konieczności samodzielnego utrzymywania infrastruktury.

5.2 Ograniczenia

W fazie wstępnej planowania projektu przyjęto wyobrażonego klienta, który ustalił następujące założenia:

Klient

- Aplikacja wytwarzana jest dla jednego klienta
- Klient posiada wiele magazynów
- Klient chce być w stanie sprawdzić stan każdego z magazynów
- Klient chce dać swoim pracownikom możliwość dodawania artykułów do stanu magazynu
- Klient chce mieć wyłączny dostęp do dodawania i modyfikowania artykułów, magazynów i kategorii

Aplikacja

- Aplikacja musi wspierać technologię PWA
- Aplikacja musi być instalowalna na różnych urządzeniach
- Aplikacja musi wspierać skanowanie kodów kreskowych i kodów QR
- Aplikacja musi wspierać autoryzację użytkowników

Inne

- Infrastruktura musi być skalowalna

6 SPECYFIKACJA WYMAGAŃ NA PRODUKT PROGRAMOWY

6.1 Baza danych

- Magazyn posiada nazwę i adres: państwo, miasto, ulica, kod pocztowy oraz opcjonalnie numer domu
- Magazyn może posiadać opis
- Każdy magazyn może mieć artykuły
- Artykuły posiadają nazwę i kategorię
- Artykuły mogą posiadać zdjęcie, opis i kod EAN13
- Kategorie posiadają nazwę
- Kategorie mogą posiadać opis
- Kategorie mogą być przypisane do więcej niż jednego artykułu
- Jeden artykuł może mieć nie więcej niż jedną kategorię
- Artykuł może się znajdować w wielu magazynach w różnych ilościach

6.2 Historyjki

Użytkownik

- Jako użytkownik chcę mieć możliwość zalogowania się na swoje konto
- Jako użytkownik chcę mieć możliwość wylogowania się ze swojego konta
- Jako użytkownik chcę mieć możliwość sprawdzenia listy przedmiotów
- Jako użytkownik chcę mieć możliwość sprawdzenia listy kategorii
- Jako użytkownik chcę mieć możliwość sprawdzenia listy magazynów
- Jako użytkownik chcę mieć możliwość sprawdzenia listy przedmiotów w magazynie
- Jako użytkownik chcę mieć możliwość dodania istniejącego przedmiotu do magazynu
- Jako użytkownik chcę mieć możliwość zwiększenia liczby przedmiotów na stanie magazynu
- Jako użytkownik chcę mieć możliwość zmniejszenia liczby przedmiotów na stanie magazynu
- Jako użytkownik chcę mieć możliwość skanowania kodów kreskowych w celu szybkiego dostępu do szczegółów danego artykułu
- Jako użytkownik chcę mieć możliwość skanowania kodów QR w celu szybkiego dostępu do stanu danego przedmiotu w danym magazynie
- Jako użytkownik chcę mieć możliwość wyboru domyślnego magazynu

Administrator

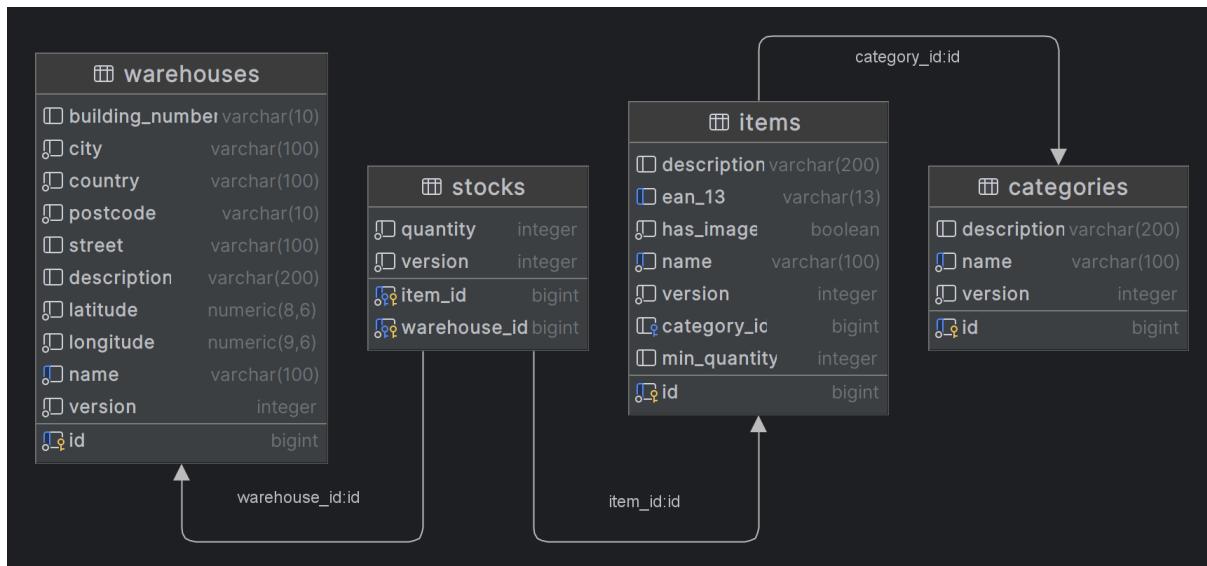
- Jako administrator chcę mieć możliwość zalogowania się na swoje konto
- Jako administrator chcę mieć możliwość wylogowania się ze swojego konta
- Jako administrator chcę mieć możliwość dodania nowego przedmiotu
- Jako administrator chcę mieć możliwość dodania nowej kategorii
- Jako administrator chcę mieć możliwość dodania nowego magazynu
- Jako administrator chcę mieć możliwość dodania istniejącego przedmiotu do magazynu
- Jako administrator chcę mieć możliwość modyfikacji istniejącego przedmiotu
- Jako administrator chcę mieć możliwość modyfikacji istniejącej kategorii

- Jako administrator chcę mieć możliwość modyfikacji istniejącego magazynu
- Jako administrator chcę mieć możliwość zwiększenia liczby przedmiotów na stanie magazynu
- Jako administrator chcę mieć możliwość zmniejszenia liczby przedmiotów na stanie magazynu
- Jako administrator chcę mieć możliwość usunięcia istniejącego przedmiotu
- Jako administrator chcę mieć możliwość usunięcia istniejącej kategorii
- Jako administrator chcę mieć możliwość usunięcia istniejącego magazynu
- Jako administrator chcę mieć możliwość usunięcia istniejącego przedmiotu z magazynu

7 PROJEKT PRODUKTU PROGRAMOWEGO

7.1 Baza danych

Baza danych została zaimplementowana w PostgreSQL według diagramu na Rysunku 1.



Rysunek 1: Diagram fizyczny bazy danych

7.2 Architektura

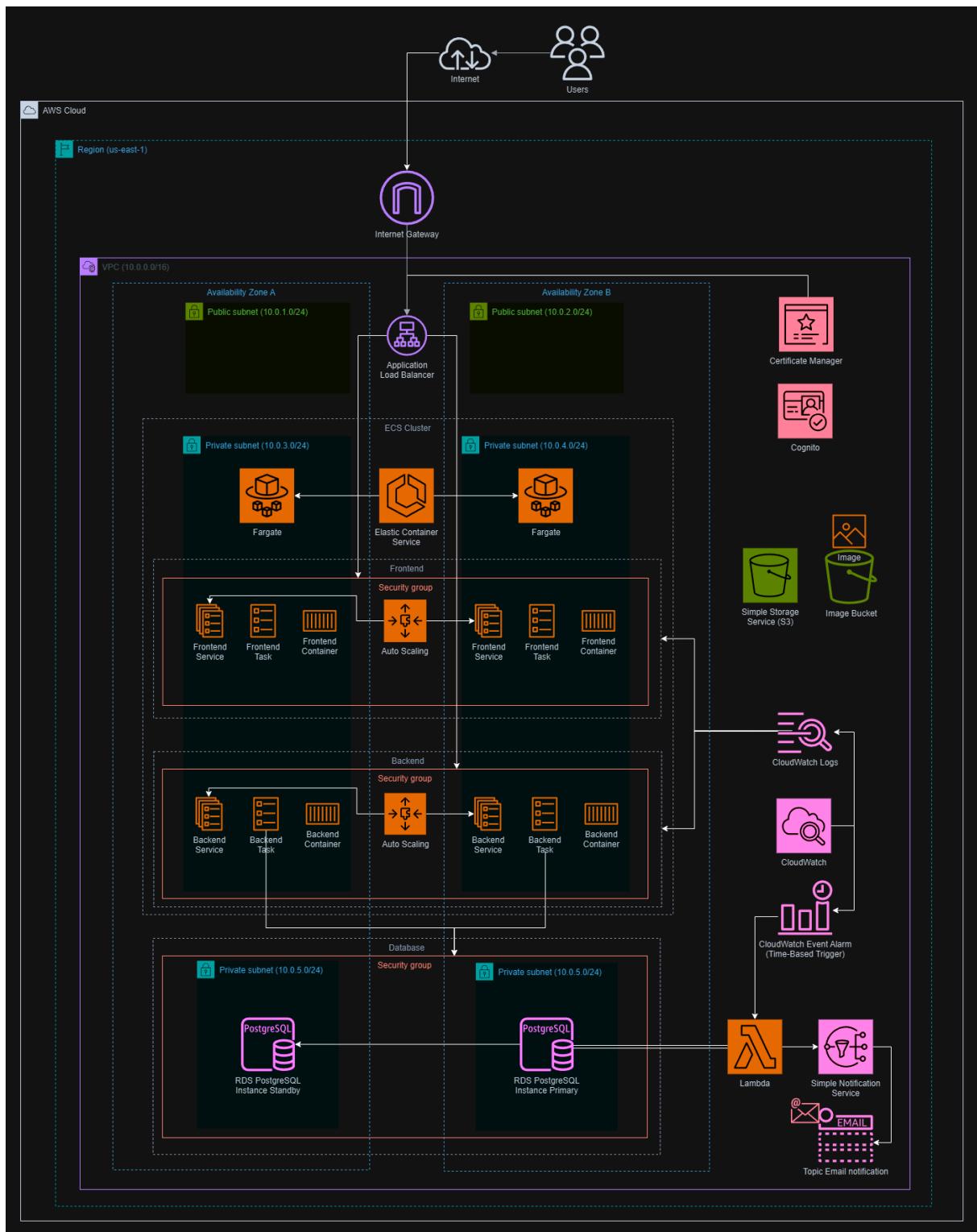
Architektura aplikacji została wdrożona przy pomocy narzędzia HashiCorp Terraform na platformie AWS. Schemat zawierający najważniejsze elementy architektury przedstawiony jest na Rysunku 2

7.3 Pozostałe schematy i diagramy

Diagram przypadków użycia

Diagram klas

Schemat przebiegu aplikacji



Rysunek 2: Diagram architektury AWS

8 IMPLEMENTACJA (OPCJA)

8.1 Warstwa użytkownika - frontend

8.1.1 Struktura plików

Warstwa użytkownika została zaimplementowana przy pomocy frameworku Angular [3] opartego o język TypeScript (TS). Strukturę plików przedstawia Rysunek 3.

```

    < src
      < app
        < core
          > guards
          > interceptors
          > services
          > utils
        < features
          > category
          > item
          > page-landing
          > scanner
          > services
          > settings
          > stock
          > warehouse
        > mocks
        > shared
      TS app-routing.module.ts
      # app.component.css
      < app.component.html
      TS app.component.spec.ts
      TS app.component.ts
      TS app.config.ts
      TS app.module.ts
      > assets
      < environment
      TS environment.ts
      ★ favicon.ico
      < index.html
      TS main.ts
      { } manifest.webmanifest
      # styles.css
  
```

(a) Ogólna struktura plików

```

    < item
      < add-view-component
        < submodules
          < item-edit-add-view
            < item-edit-add-view.component.html
            < item-edit-add-view.component.scss
            TS item-edit-add-view.component.ts
          < select-field
            < dropdown-menu.component.html
            < dropdown-menu.component.scss
            TS dropdown-menu.component.ts
          # item-add.component.css
          < item-add.component.html
          TS item-add.component.ts
          TS item-add.module.ts
        < details-view-component
          < submodules\item-detail-view
            < item-detail-view.component.html
            < item-detail-view.component.scss
            TS item-detail-view.component.ts
          # item-details.component.css
          < item-details.component.html
          TS item-details.component.ts
          TS item-details.module.ts
        < grid-view
          < submodules\card-item
            < card-item.component.html
            < card-item.component.scss
            TS card-item.component.ts
            < item-grid-view.component.html
            < item-grid-view.component.scss
            TS item-grid-view.component.ts
            TS item-grid-view.module.ts
        < services
          TS item.service.spec.ts
          TS item.service.ts
          TS itemImageUrl.service.spec.ts
          TS itemImageUrl.service.ts
          TS items-routing.module.ts
          TS items.module.ts
  
```

(b) Dokładna struktura jednego z komponentów

Rysunek 3: Struktura plików frontendu

Aplikacja jest tworzona przy pomocy komponentów. Każdy komponent składa się ze skryptu TS, pliku stylu SCSS oraz pliku układu HTML. Dodatkowo niektóre komponenty korzystają z serwisów napisanych w języku TypeScript. Komponenty graficzne zazwyczaj mogą być wykorzystywane wiele razy i są one umieszczone w folderze *shared*. Komponenty jednokrotne są umieszczone w folderze *features* i zawierają one konkretne implementacje poszczególnych widoków.

8.1.2 Widoki

8.1.3 Routing

8.1.4 Testy

8.1.5 Konteneryzacja - Docker

8.2 Warstwa systemowa - backend

- 8.2.1 Struktura plików**
- 8.2.2 Konfiguracja**
- 8.2.3 Kontrolery**
- 8.2.4 Inne**
- 8.2.5 Testy**
- 8.2.6 Konteneryzacja - Docker**

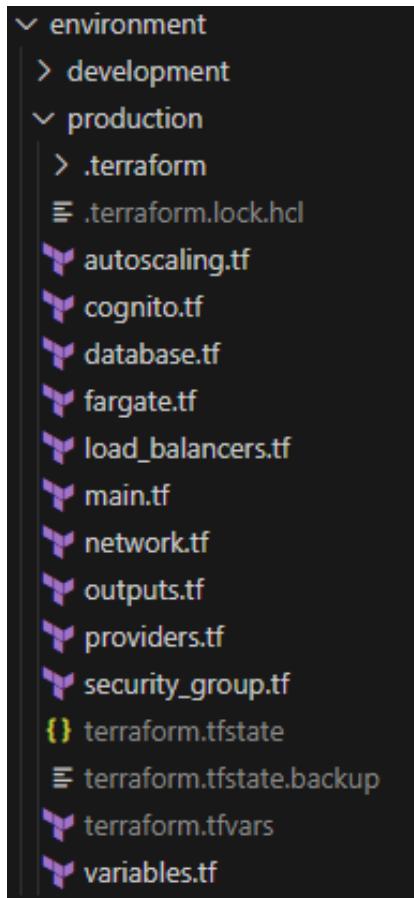
8.3 Architektura chmurowa

8.3.1 Wstęp

Architektura chmurowa została napisana w oprogramowaniu HashiCorp Terraform i wdrożona na serwery Amazon Web Services (AWS) [1]. HashiCorp Terraform (albo krótko Terraform) [4] to oprogramowanie Infrastructure as Code (IaC) pozwalające na skryptowe wdrażanie infrastruktury na serwery dostawcy.

8.3.2 Struktura plików

Struktura plików została podzielona ze względu na środowisko produkcyjne i testowe. Na Rysunku 4 przedstawione jest tylko środowisko produkcyjne ze względu na tą samą strukturę systemu plików. Z tego samego względzu omówione zostanie tylko środowisko produkcyjne.



Rysunek 4: Struktura plików Terraform

Pliki `.tf` zawierają opis zasobów wdrażanych na serwery. Plik `.tfstate` przechowuje dokładny spis zasobów już wdrożonych wraz z ich unikalnymi identyfikatorami, sekretami, itd. Opis pozostałych plików:

- `main.tf` - wersje oprogramowania wykorzystywane w projekcie
- `providers.tf` - konfiguracja dostawcy AWS
- `variables.tf` - spis wszystkich zmiennych wraz z ich opisami i niektórymi wartościami domyślnymi
- `terraform.tfvars` - niewersjonowany plik zawierający wartości niektórych zmiennych; przede wszystkim przechowuje sekrety
- `outputs.tf` - zbiór przydatnych wartości wyjściowych, np. nazwa DNS Application Load Balancera
- `network.tf` - zasoby sieciowe: VPC, brama sieciowa, tablice routingu, podsieci
- `security_groups.tf` - grupy bezpieczeństwa (działają jak zapora ogniodziałająca - ang. firewall) konfigurujące przychodzący i wychodzący ruch sieciowy
- `cognito.tf` - dostawca tożsamości Amazon Cognito i jego konfiguracja

- `load_balancers.tf` - Application Load Balancer, zasoby nasłuchujące na portach 80, 443 i 8080 oraz grupy celów
- `fargate.tf` - definicje zadań wykonywanych przy pomocy AWS Fargate oraz serwisy które nimi zarządzają
- `autoscaling.tf` - zestaw reguł pozwalających na automatyczne zmniejszanie lub zwiększenie liczby działających instancji modułów (frontendu i backendu)
- `database.tf` - konfiguracja bazy danych RDS

8.3.3 Wykorzystane zasoby

Amazon Virtual Private Cloud (VPC): Amazon VPC pozwala na tworzenie izolowanej, logicznie odseparowanej sieci w chmurze AWS. Umożliwia pełną kontrolę nad konfiguracją sieci, w tym zakresami adresów IP, podsieciami, trasami, bramami i konfiguracją reguł zapory sieciowej.

Amazon Cognito: Amazon Cognito umożliwia zarządzanie tożsamościami i autoryzacją użytkowników w aplikacjach internetowych oraz mobilnych. Zapewnia obsługę logowania użytkowników, synchronizacji danych oraz integracji z dostawcami tożsamości.

Amazon Relational Database Service (RDS): Amazon RDS to zarządzana usługa baz danych, która umożliwia łatwe konfigurowanie, działanie i skalowanie relacyjnych baz danych. Obsługuje popularne silniki, takie jak MySQL, PostgreSQL, MariaDB, Oracle oraz SQL Server.

Amazon Application Load Balancer (ALB): Amazon ALB to rodzaj load balansera, który automatycznie rozdziela ruch aplikacji na wiele celów, takich jak kontenery, instancje EC2 lub funkcje Lambda. Jest zoptymalizowany do obsługi ruchu HTTP/HTTPS oraz oferuje zaawansowane funkcje, takie jak routing oparte na zawartości czy target groups.

Amazon Elastic Container Service (ECS): Amazon ECS to zarządzana usługa orkiestracji kontenerów, która umożliwia uruchamianie, zatrzymywanie i zarządzanie kontenerami Docker na klastrach instancji EC2 lub w ramach AWS Fargate (serverless). Jest w pełni zintegrowana z innymi usługami AWS.

Amazon Certificate Manager (ACM): Amazon ACM umożliwia łatwe zarządzanie certyfikatami SSL/TLS. Automatyzuje procesy wydawania, odnawiania i wdrażania certyfikatów dla aplikacji działających w AWS, eliminując potrzebę ręcznej konfiguracji.

Amazon CloudWatch: Amazon CloudWatch to usługa monitorowania zasobów i aplikacji AWS. Umożliwia zbieranie, przeglądanie i analizowanie metryk, logów oraz ustawianie alarmów. Zapewnia również możliwość automatycznej reakcji na zmiany wydajności systemu.

Amazon Simple Storage Service (S3): Amazon S3 to skalowalna usługa magazynowania obiektów, która pozwala na przechowywanie dowolnej ilości danych. Jest idealna do tworzenia kopii zapasowych, archiwizacji, hostowania zasobów statycznych i integrowania z aplikacjami chmurowymi.

AWS Auto Scaling: AWS Auto Scaling automatycznie dostosowuje liczbę zasobów obliczeniowych (np. kontenerów Fargate) w odpowiedzi na zmieniające się obciążenie. Zapewnia elastyczność, wysoką dostępność i optymalizację kosztów poprzez skalowanie w górę lub w dół w oparciu o zdefiniowane zasady.

9 DEMONSTRACJA PRODUKTU PROGRAMOWEGO

9.1 Wstęp

W ramach demostracji produktu programowego stworzono przykładowe artykuły, kategorie, magazyny i stany magazynów korzystając z tekstur gry Minecraft. Wszystkie tekstury pochodzą z portalu minecraft.fandom.com [2]

9.2 Widoki - komputer stacjonarny

Podstawowe widoki wyświetlane na komputerze stacjonarnym zostały przedstawione na Rysunkach 5, 7, 9, 14, 19, 24, 27, 28.

9.3 Widoki - telefon komórkowy

Podstawowe widoki wyświetlane na telefonie komórkowym zostały przedstawione na Rysunkach 6, 8, 10, 15, 20, 25, 29, 30.

9.4 Opis aplikacji

9.4.1 Podstawowe widoki

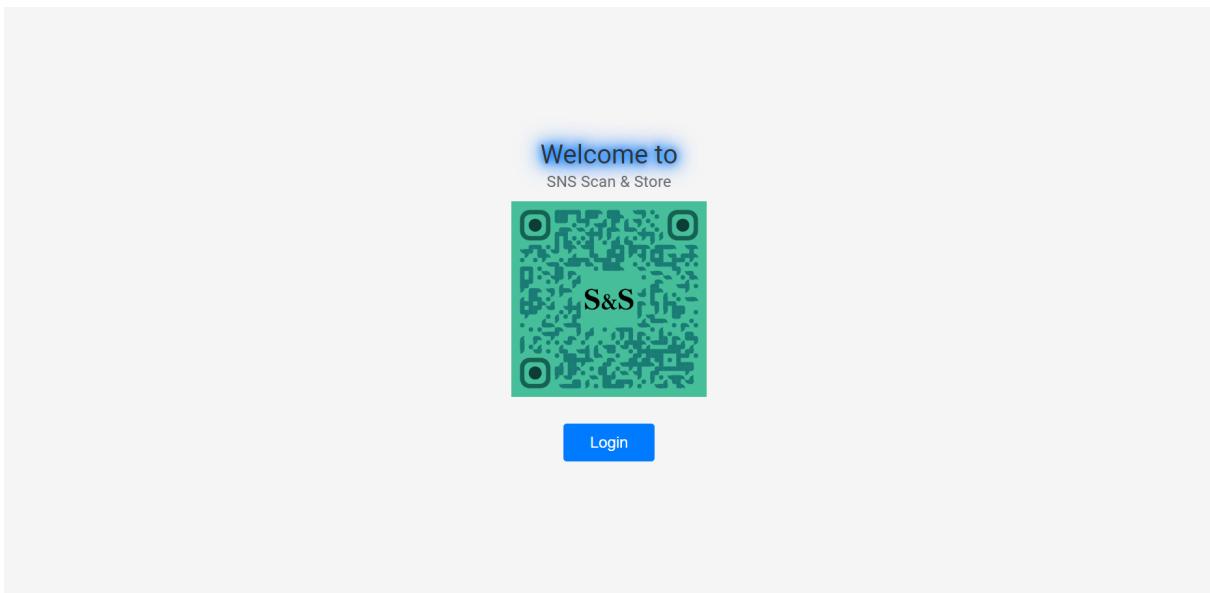
Aplikacja zawiera 7 podstawowych widoków:

- stronę startową (Rysunki 5 i 6),
- artykuły (Rysunki 9 i 10),
- kategorie (Rysunki 14 i 15),
- magazyny (Rysunki 19 i 20),
- ustawienia (Rysunki 24 i 25)
- stany magazynów (Rysunki 28 i 30),
- skaner kodów kreskowych i kodów QR.

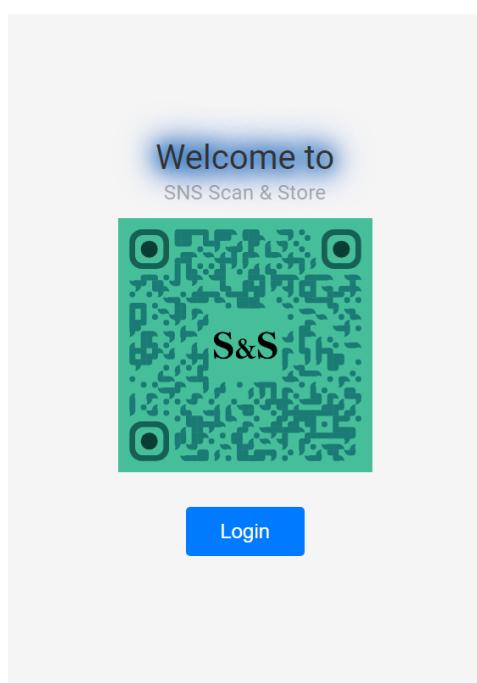
9.4.2 Strona startowa

Strona startowa może przyjąć 2 stany: użytkownik niezalogowany (Rysunek 5) i użytkownik zalogowany (Rysunek 7). W przypadku użytkownika niezalogowanego jedyną funkcjonalnością jest przekierowanie do logowania. W przypadku użytkownika zalogowanego istnieją dwie funkcjonalności: wylogowanie lub przejście do aplikacji.

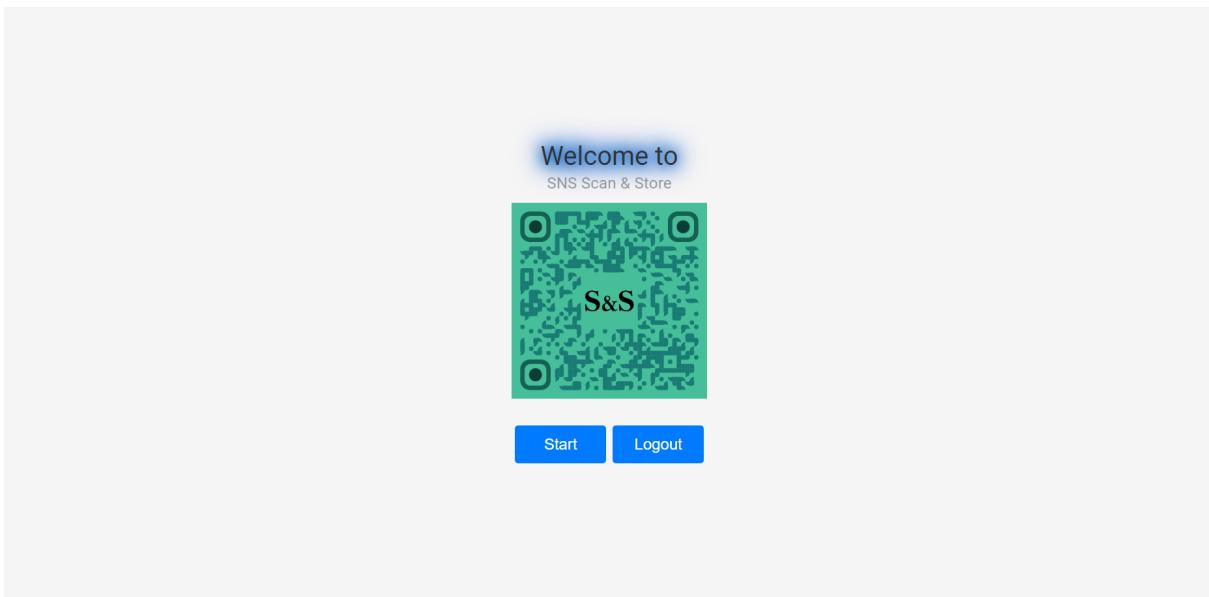
Po poprawnej próbie zalogowania użytkownik zostaje automatycznie przekierowany do aplikacji. Jeśli kiedykolwiek ponownie znajdzie się na stronie startowej to dostanie możliwość przejścia do aplikacji ręcznie lub możliwość wylogowania się. Po wylogowaniu użytkownik zostaje przekierowany z powrotem na stronę startową.



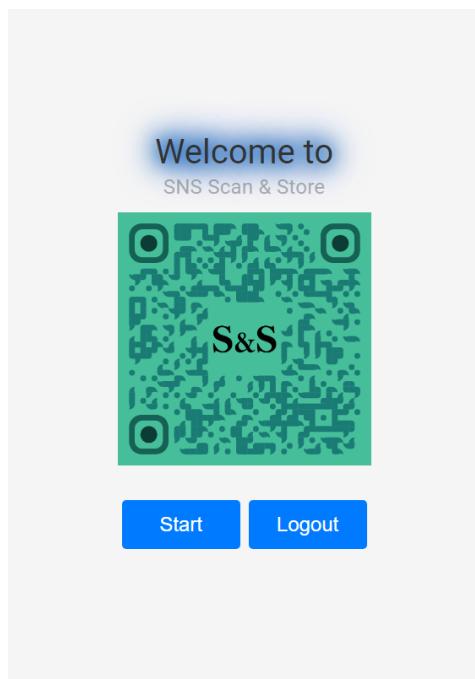
Rysunek 5: Strona startowa na komputerze stacjonarnym



Rysunek 6: Strona startowa na telefonie komórkowym



Rysunek 7: Strona startowa po zalogowaniu na komputerze stacjonarnym



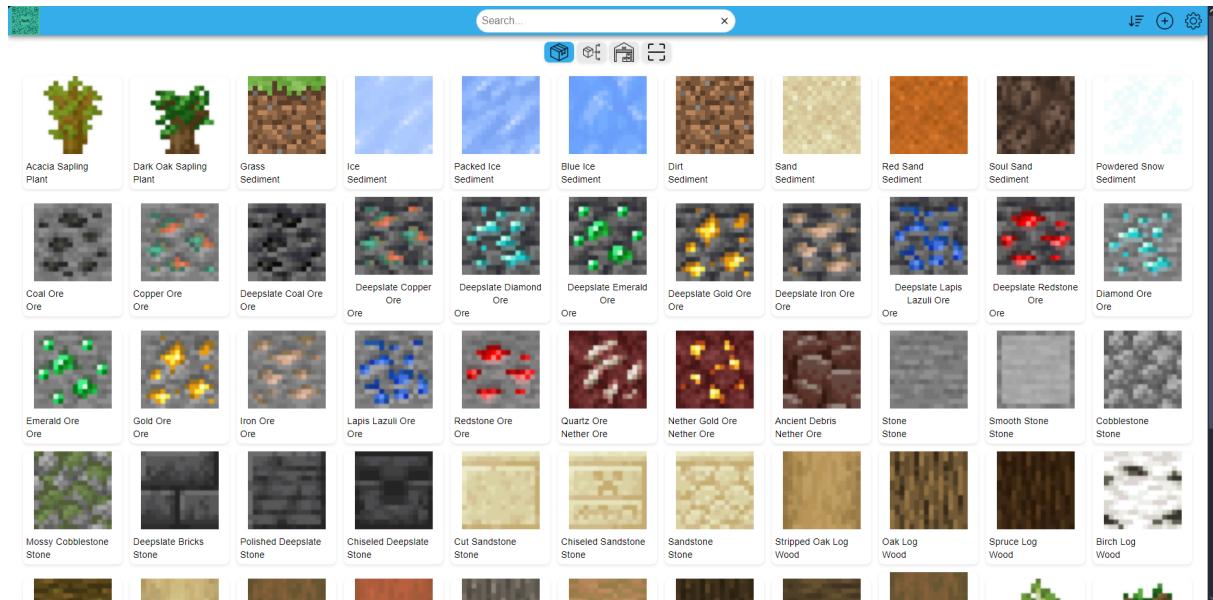
Rysunek 8: Strona startowa po zalogowaniu na telefonie komórkowym

9.4.3 Artykuły

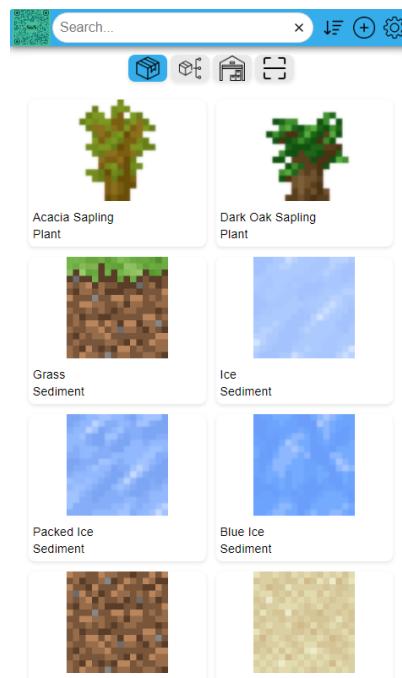
Strona artykułów przedstawiona na rysunku 9 służy do pełnej obsługi wszystkich dostępnych artykułów, które mogą znaleźć się w dowolonym magazynie. W górnym pasku znajdują się opcje pozwalające na kolejno od lewej:

- przejście z powrotem do strony głównej (element wspólny dla wszystkich widoków)
- wyszukanie konkretnego artykułu z uwzględnieniem nazwy i nazwy kategorii
- posortowanie artykułów względem nazwy
- dodanie nowego artykułu
- przejście do widoku ustawień (element wspólny dla wszystkich widoków)

Wybór konkretnego artykułu następuje poprzez naciśnięcie na konkretny wpis. Po wyborze użytkownik jest przeniesiony do strony ze szczegółami artykułu (Rysunek 11), a w górnym pasku otrzymuje możliwość edycji i usunięcia danego artykułu.



Rysunek 9: Widok przedmiotów na komputerze stacjonarnym



Rysunek 10: Widok przedmiotów na telefonie komórkowym

The screenshot shows the 'Details' view of a product. At the top left is a thumbnail image of a brown, textured object. To its right are five input fields: 'Name' (Dot), 'Category' (Sediment), 'Description' (empty), 'EAN13' (empty), and 'Minimum Quantity' (empty). Below these fields is a large, empty text area.

Rysunek 11: Widok szczegółów przedmiotu

W widoku edycji (Rysunek 12) użytkownik może zmienić dowolny element dotyczący artykułu, włącznie z dodaniem lub usunięciem zdjęcia.

The screenshot shows the 'Edit' view of the same product. It includes all the fields from the previous screenshot plus a 'Remove Image' button above the image area. The image area now has a placeholder 'No Image'.

Rysunek 12: Widok edycji przedmiotu

Po przejściu do widoku dodawania nowego przedmiotu otrzymamy widok, przedstawiony na Rysunku 13, który pozwala na dokładne te same funkcjonalności co edycja, lecz na początku nie posiada żadnych wypełnionych pól.

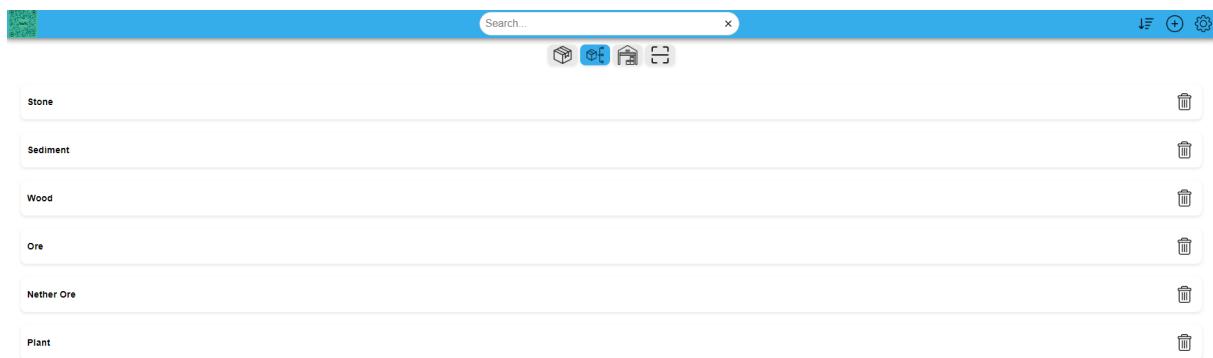
The screenshot shows the 'Add Item' view. It features a large 'No Image' placeholder where an image would normally be displayed. Below it are five empty input fields: 'Name' (Dot), 'Category' (Select a category), 'Description' (Enter text), 'EAN13' (empty), and 'Minimum Quantity' (empty).

Rysunek 13: Widok dodawania przedmiotu

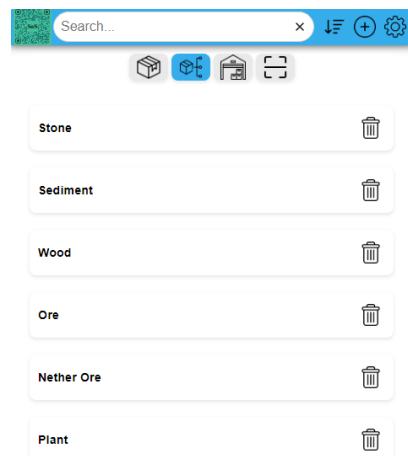
9.4.4 Kategorie

Strona kategorii przedstawiona na rysunku 14 pozwala na przegląd i dodawanie kategorii dla przedmiotów. Podobnie jak w przypadku artykułów w pasku górnym znajdują się te same opcje. Także analogicznie do artykułów wybór kategorii następuje poprzez nacisnięcie odpowiedniego elementu.

Widok szczegółów (Rysunek 16), edycji (Rysunek 17) i dodawania (Rysunek 18) nowej kategorii działa tak samo jak w przypadku artykułów.



Rysunek 14: Widok kategorii na komputerze stacjonarnym



Rysunek 15: Widok kategorii na telefonie komórkowym



Rysunek 16: Widok szczegółów kategorii

Edit

Name: Ore

Description: Ores that can be found in the Overworld

Submit

Rysunek 17: Widok edycji kategorii

Create

Name:

Description: Enter text...

Submit

Rysunek 18: Widok dodawania kategorii

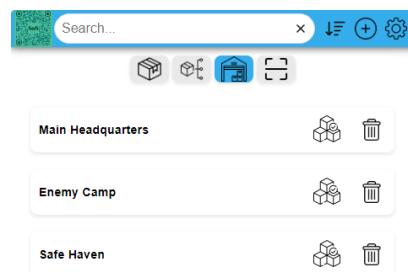
9.4.5 Magazyny

Strona magazynów przedstawiona na rysunku 19 pozwala na przegląd i dodawanie magazynów. Podobnie jak w przypadku artykułów i kategorii w pasku górnym znajdują się te same opcje. Także analogicznie do artykułów i kategorii wybór magazynów następuje poprzez naciśnięcie odpowiedniego elementu. Dodatkowo na każdym wpisie znajduje się przycisk, który przekierowuje do stanu danego magazynu.

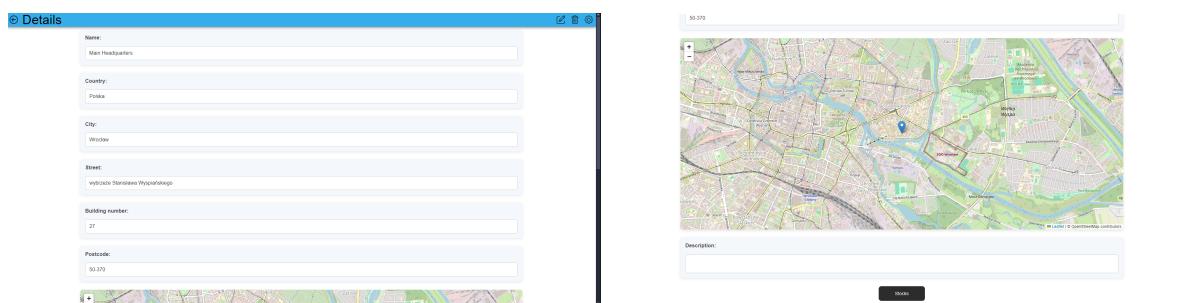
Widok szczegółów, edycji i dodawania nowej kategorii działa tak samo jak w przypadku artykułów i kategorii z dodatkiem interaktywnej mapy. Podczas wyświetlania szczegółów (Rysunek 21), mapa przedstawia wybraną lokalizację magazynu. Dodatkowo widok szczegółów zawiera przycisk, który przekierowuje bezpośrednio do stanu danego magazynu.



Rysunek 19: Widok magazynów na komputerze stacjonarnym

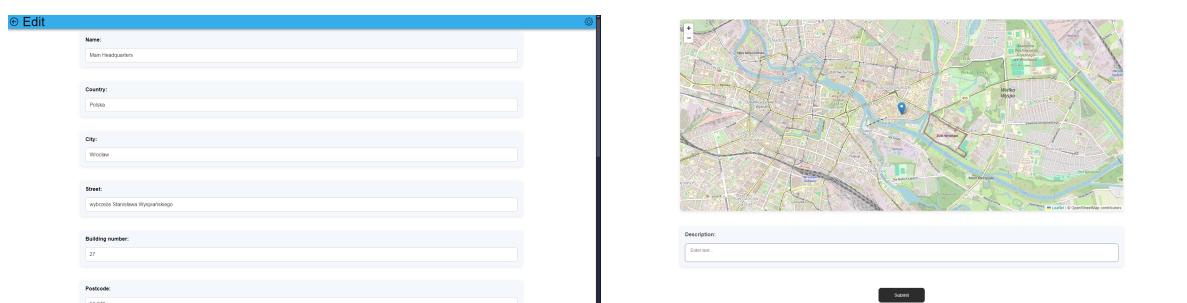


Rysunek 20: Widok magazynów na telefonie komórkowym



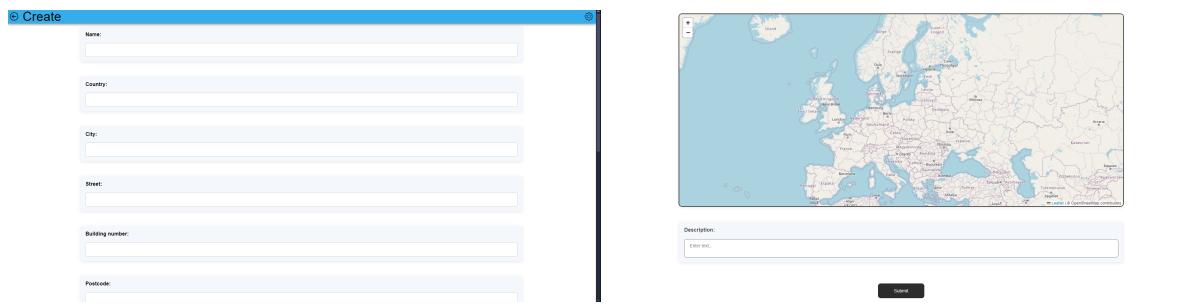
Rysunek 21: Widok szczegółów magazynu

Podczas edycji (Rysunek 22), mapa jest interaktywna i pozwala na wybór lokalizacji poprzez przeniesienie pinezki w wybrane miejsce. Wszystkie pola, które mogą być wypełnione, zostaną automatycznie zaktualizowane. Podobnie w przypadku wprowadzenia nowego adresu mapa zaktualizuje się aby odzwierciedlić zmiany.



Rysunek 22: Widok edycji magazynu

Podobnie jak w przypadku edycji, podczas dodawania nowego magazynu (Rysunek 23) mapa jest interaktywna i zachowuje się w ten sam sposób.



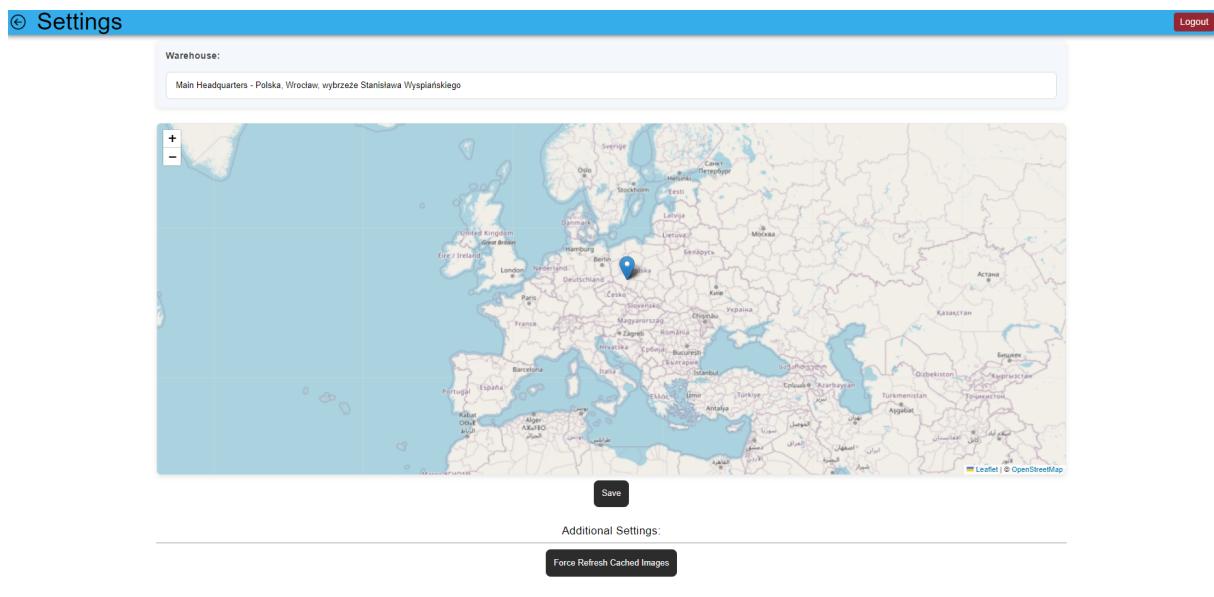
Rysunek 23: Widok dodawania magazynu

9.4.6 Ustawienia

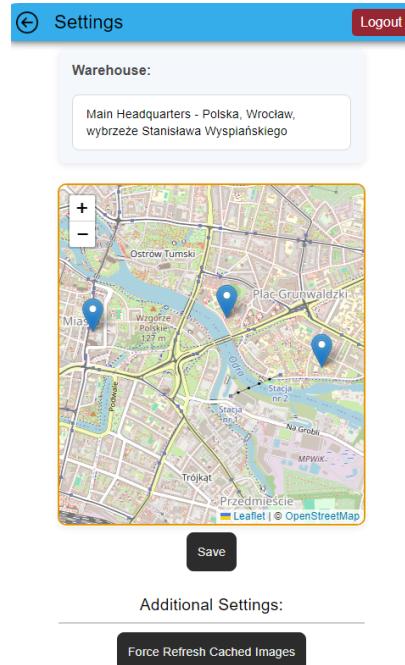
Widok ustawień (Rysunek 24) oferuje 3 funkcjonalności: ustalenie domyślnego magazynu, wylogowanie się z aplikacji oraz odświeżenie obrazków na wypadek jakichkolwiek problemów z funkcją cache. Przycisk wylogowania przekierowuje użytkownika na stronę startową i go wylogowuje. Odświeżenie obrazków powoduje wyczyszczenie wszystkich zapamiętanych obrazków i pozwala na ponowne pobranie ich z serwera.

Selekcja domyślnego magazynu przebiega poprzez wybranie jednego ze stworzonych wcześniej magazynu z listy, lub kliknięcie w odpowiednią pinezkę na mapie.

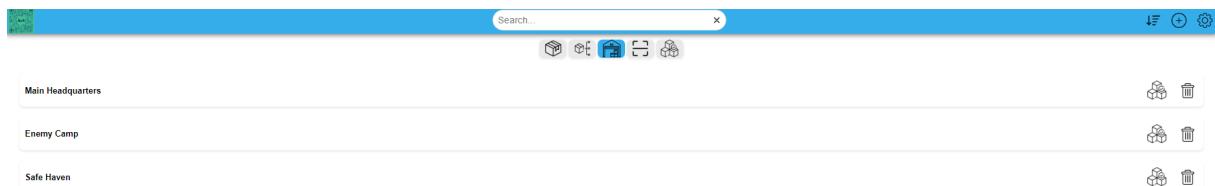
Po ustaleniu domyślnego magazynu w głównych widokach pojawi się nowa ikona oznaczająca stan domyślnego magazynu co przedstawia Rysunek 26.



Rysunek 24: Widok ustawień na komputerze stacjonarnym



Rysunek 25: Widok ustawień na telefonie komórkowym



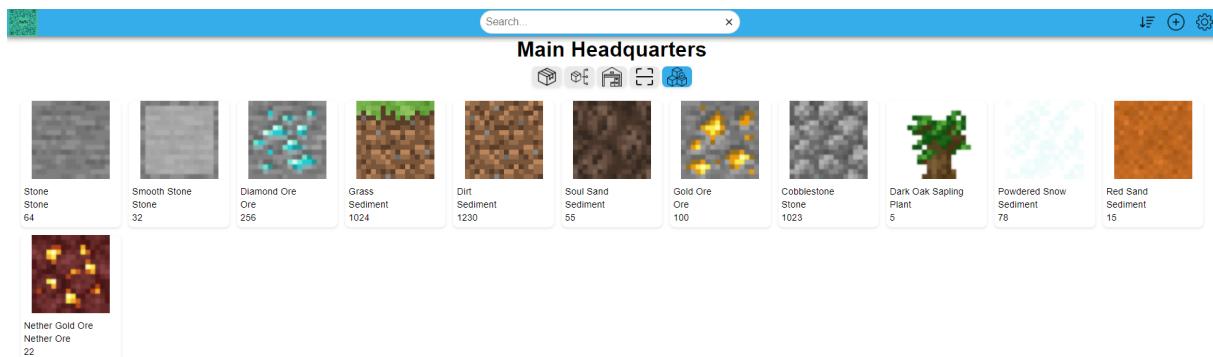
Rysunek 26: Widok magazynów po ustaleniu domyślnego magazynu

9.4.7 Stan magazynu

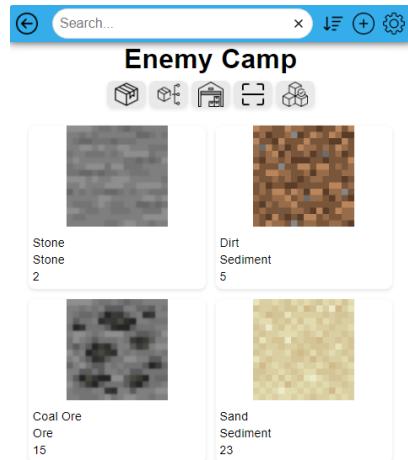
Widok stanu magazynu ma dwie postaci: dowolny magazyn (Rysunek 27) oraz magazyn domyślny (Rysunek 28), gdzie jedyną różnicą jest podświetlenie ikony domyślnego magazynu. Funkcje górnego paska są takie same jak w widoku artykułu, z tą różnicą, że przycisk dodawania pozwala na wprowadzenie nowego przedmiotu na stan magazynu. W widoku szczegółowym (Rysunek 31) wyświetlane są informacje o przedmiocie, takie jak obrazek, nazwa, kategoria, opis i kod EAN13, nazwa aktualnego magazynu minimalna liczba sztuk oraz aktualna liczba sztuk danego przedmiotu. Minimalna liczba sztuk jest określana w celu zawiadamiania odpowiednich użytkowników w momencie kiedy stan danego przedmiotu w magazynie spadnie poniżej wymaganej wartości.



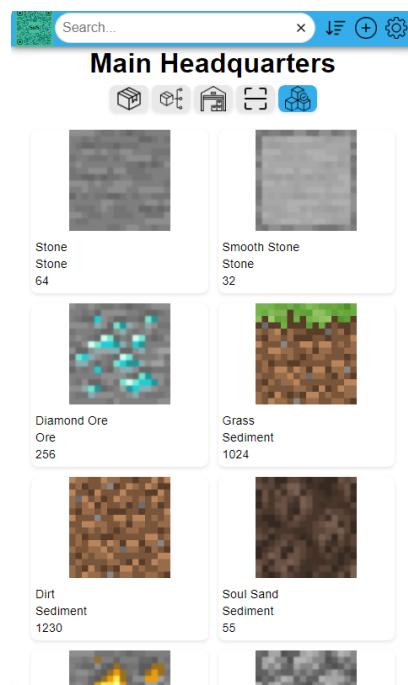
Rysunek 27: Widok stanu magazynu dla dowolnego magazynu na komputerze stacjonarnym



Rysunek 28: Widok stanu magazynu dla domyślnego magazynu na komputerze stacjonarnym



Rysunek 29: Widok stanu magazynu dla dowolnego magazynu na telefonie komórkowym



Rysunek 30: Widok stanu magazynu dla domyślnego magazynu na telefonie komórkowym

Details

Name: Dirt

Category: Sediment

Description:

EAN13:

Warehouse: Stan magazynu

Minimum quantity: 100

Quantity: 100

Rysunek 31: Widok szczegółów stanu magazynu

Edycja stanu magazynu przedstawiona na Rysunku 32 pozwala na określenie zmiany w liczbie sztuk danego przedmiotu. Zmiana może być dodatnia (wprowadzanie dodatkowych przedmiotów do magazynu) lub ujemna (wydawanie przedmiotów z magazynu). Zmiany można zatwierdzić lub anulować wybierając odpowiedni przycisk.

Edit

Name: Dirt

Category: Sediment

Description:

EAN13:

Warehouse: Stan magazynu

Minimum quantity:

Changes:

Current quantity: 100

Future quantity: 100

Cancel Save

Rysunek 32: Widok edycji stanu magazynu

Dodawanie przedmiotu do stanu magazynu działa analogicznie do edycji przedmiotu na stanie magazynu z tą różnicą, że wszystkie pola, poza nazwą aktualnego magazynu, są wstępnie puste.

Add

No Image

Name: Dirt a item

Category:

Description:

Changes:

Current quantity: 0

Future quantity: 0

Cancel Save

Rysunek 33: Widok dodawania stanu magazynu

9.4.8 Skaner

Widok skanera przedstawiony jest na Rysunku 34 i zawiera tylko obraz z kamery. Po zeskanowaniu odpowiedniego kodu użytkownik zostaje przekierowany do odpowiedniej strony: artykułu - jeśli zeskanował kod EAN13; stanu domyślnego magazynu - jeśli zeskanował kod QR z zawartym ID artykułu.



Rysunek 34: Widok skanera kodów kreskowych i kodów QR

LITERATURA

- [1] Matt Garman (CEO). Aws dokumentacja. <https://docs.aws.amazon.com>. Dostęp: 2024-11-30.
- [2] Perkins Miller (CEO). Strona internetowa z listą teksturow z gry minecraft. https://minecraft.fandom.com/wiki/List_of_block_textures. Dostęp: 2024-12-14.
- [3] Google. Angular dokumentacja. <https://angular.dev>. Dostęp: 2024-11-30.
- [4] HashiCorp. Hashicorp terraform dokumentacja. <https://developer.hashicorp.com/terraform/docs>. Dostęp: 2024-11-30.