



**Politechnika
Śląska**

Dokumentacja projektowa

Programowanie Obiektowe i Graficzne

System Inwentaryzacyjny

Kierunek: Informatyka

Członkowie zespołu:

Bartłomiej Janoszka

Dominik Meisner

Mikołaj Molenda

Piotr Dusiński

Gliwice, 2024/2025

Spis treści

| | | |
|----------|---|-----------|
| 1 | Wprowadzenie | 2 |
| 1.1 | Role w projekcie | 2 |
| 1.2 | Cel projektu | 2 |
| 2 | Założenia projektowe | 3 |
| 2.1 | Opis wymagań | 3 |
| 2.2 | Część dotycząca aplikacji oraz wzorca MVC | 3 |
| 2.3 | Część dotycząca Bazy Danych | 3 |
| 2.3.1 | Realizacja połączenia z baza | 3 |
| 2.3.2 | Encje | 4 |
| 2.3.3 | Związki | 4 |
| 2.3.4 | Diagram związków encji | 5 |
| 2.3.5 | Model relacyjny | 6 |
| 3 | Instrukcja obsługi aplikacji - system inwentaryzacyjny | 7 |
| 4 | Wnioski | 14 |
| 4.1 | Podsumowanie projektu | 14 |
| 4.2 | Potencjał rozwoju | 14 |
| 5 | Link do repozytorium GitHub | 14 |

1 Wprowadzenie

1.1 Role w projekcie

- Bartłomiej Janoszka - tworzenie dokumentacji projektu, projekt koncepcyjny bazy danych, stworzenie użytkowników, stworzenie koncepcji działania aplikacji, stworzenie UI aplikacji
- Dominik Meisner - tworzenie dokumentacji projektu, projekt koncepcyjny bazy danych, tworzenie struktur bazy, hosting aplikacji na serwerze Mikr.us, stworzenie podstaw aplikacji Django
- Piotr Dusiński - stworzenie struktury bazy, testowanie poprawności działania aplikacji, uzupełnienie bazy danymi, stworzenie UI aplikacji, praca nad aplikacją
- Mikołaj Molenda - tworzenie dokumentacji projektu, uzupełnienia bazy danymi, testowanie poprawności działania aplikacji, stworzenie UI aplikacji, praca nad aplikacją

1.2 Cel projektu

Celem projektu było stworzenie systemu inwentaryzacyjnego sprzętu IT. Aplikacja byłaby przeznaczona dla firm oraz przedsiębiorstw, w celu monitorowania, porządkowania oraz kontrolowania sprzętu IT. Baza danych połączona z aplikacją przechowuje informacje o użytkownikach, projektach, sprzęcie komputerowym (od laptopów i komputerów PC po drukarki i routery) oraz licencjach na oprogramowanie. Dzięki odpowiednim relacjom mamy możliwość podejrzenia do jakiego projektu/użytkownika przypisana jest dana licencja bądź urządzenie. W aplikacji będzie zastosowany wzorzec projektowy MVC (*Model-View-Controller*), a całość napisana będzie w Pythonie przy pomocy wysokopoziomowego framework'a Django. Na końcu aplikacja będzie wdrożona na serwer Mikr.us w celu umożliwienia dostępu do niej z dowolnego miejsca na ziemi.

2 Założenia projektowe

2.1 Opis wymagań

- Stworzenie działającej aplikacji systemu inwentaryzacji sprzętu IT
- Zastosowanie wzorca MVC
- Realizacja połączenia z baza danych
- Stworzenie prostego, czytelnego i przejrzystego interfejsu użytkownika
- Udostępnienie aplikacji w internecie

2.2 Część dotycząca aplikacji oraz wzorca MVC

Aplikacja zbudowana jest w oparciu o wzorzec architektoniczny MTV (Model-Template-View), będący interpretacją klasycznego MVC w Django. System został logicznie podzielony na trzy niezależne aplikacje Django: `hardware`, `licenses` oraz `users`. Każda z nich odpowiada za osobny fragment domeny biznesowej i posiada własne modele danych (`models.py`), logikę kontrolerów (`views.py`), lokalny routing (`urls.py`) oraz szablony HTML. Taka modułarna architektura ułatwia rozwój, testowanie i skalowanie aplikacji.

| Klasyczny MVC | Django (MTV) | Funkcja |
|---------------|--------------|---|
| Model | Model | Odpowiada za strukturę danych i logikę warstwy ORM |
| View | Template | Generuje HTML na podstawie danych przekazanych z widoku |
| Controller | View | Obsługuje żądania HTTP i pełni rolę kontrolera |

2.3 Część dotycząca Bazy Danych

2.3.1 Realizacja połączenia z baza

1. Konfiguracja połączenia

Połączenie z bazą danych definiuje się w pliku `settings.py` w zmiennej `DATABASE`. Oto nasza konfiguracja:

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.mysql",
        "NAME": "inventory_shop_system_db",
        "USER": "root",
        "PASSWORD": "admin",
        "HOST": "db",
    }
}
```

2. Danymi z bazy manipulujemy za pomocą ORM-u (Object-Relational Mapper) Django, który pozwala na interakcję bez konieczności pisania zapytań SQL:

```
from .models import Hardware
hardware_list = Hardware.objects.all()
```

ORM umożliwia zmianę silnika bazy danych, bez konieczności wprowadzania dużych zmian w kodzie.

3. Migracje

Django śledzi zmiany w modelach i zapisuje je w formie migracji do wykonania na bazie danych. W celu wykonania migracji posługujemy się dwiema komendami:

```
python manage.py makemigrations
python manage.py migrate
```

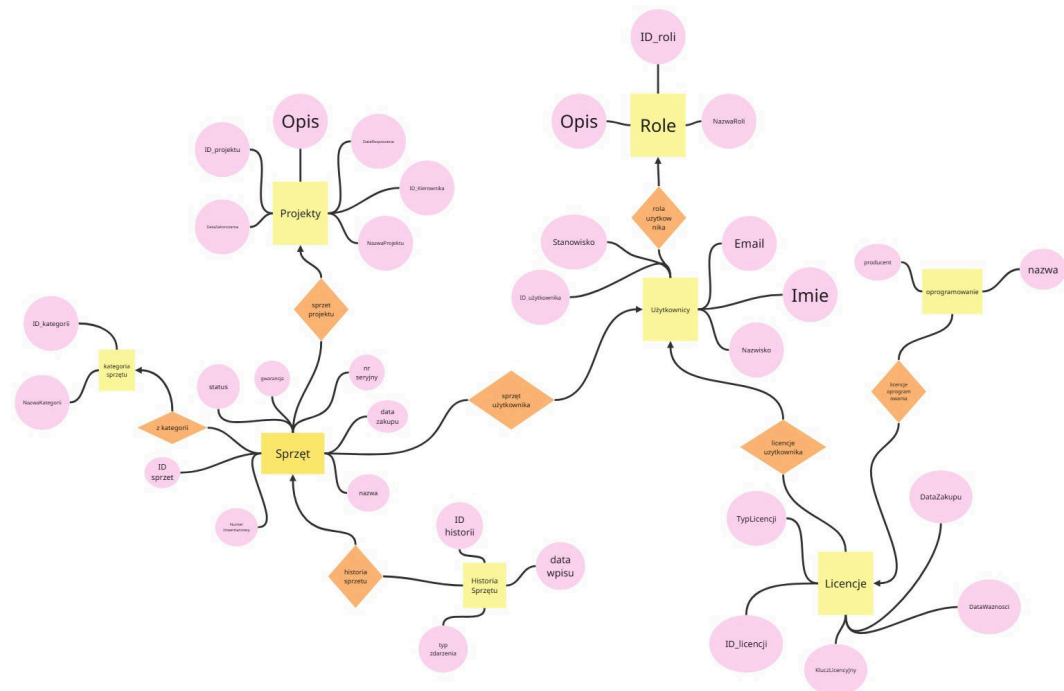
2.3.2 Encje

1. **Użytkownik** - ID_Uzytkownika[PK], Imie, Nazwisko, Email, Stanowisko
2. **Rola** - ID_Roli[PK], Nazwa, Opis
3. **Projekt** - ID_Projektu[PK], Nazwa, Opis, DataRozpoczecia, DataZakonczenia
4. **Sprzet** - ID_Sprzetu[PK], NumerInwentarzowy, Nazwa, NumerSeryjny, DataZakupu, GwarancjaDo, Status
5. **KategoriaSprzetu** - ID_Kategorii[PK], Nazwa
6. **HistoriaSprzetu** - ID_Historii[PK], DataZdarzenia, TypZdarzenia, Opis
7. **Licencje** - ID_Licencji[PK], KluczLicencyjny, DataZakupu, DataWaznosci, TypLicencji
8. **Oprogramowanie** - ID_Oprogramowania[PK], Nazwa, Producent

2.3.3 Związki

1. **Rola użytkownika** - 1 : n
2. **Licencje użytkownika** - 1 : n
3. **Licencje oprogramowania** - 1 : n
4. **Sprzęt użytkownika** - 1 : n
5. **Sprzęt projektu** - 1 : n
6. **Z kategorii** - 1 : n
7. **Historia Sprzętu** - 1 : n

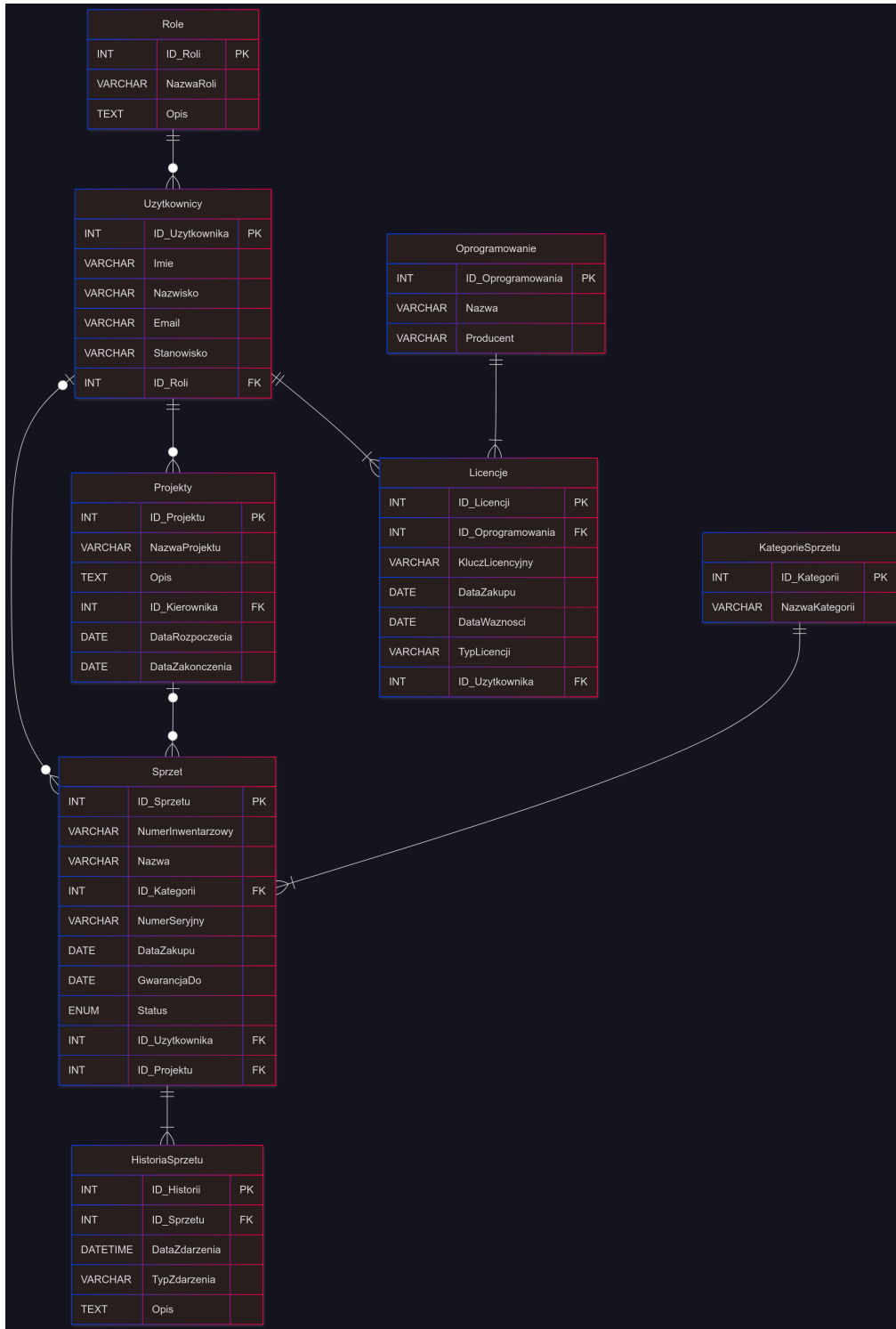
2.3.4 Diagram związków encji



Rysunek 1: Diagram związków encji

2.3.5 Model relacyjny

Zaznaczając relacje w poniższym modelu posłużyliśmy się notacją kruczej stopki.



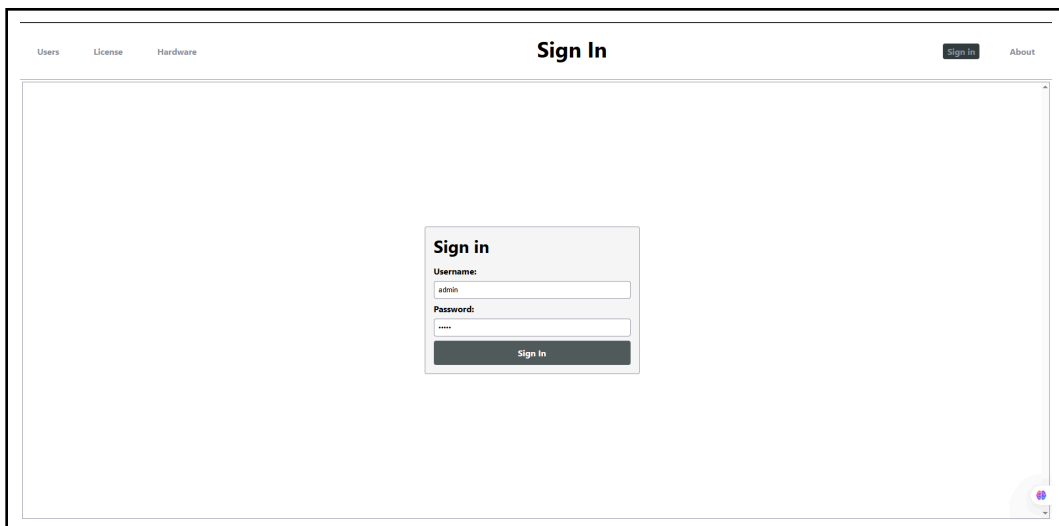
Rysunek 2: Model relacyjny

3 Instrukcja obsługi aplikacji - system inwentaryzacyjny

Użytkownicy

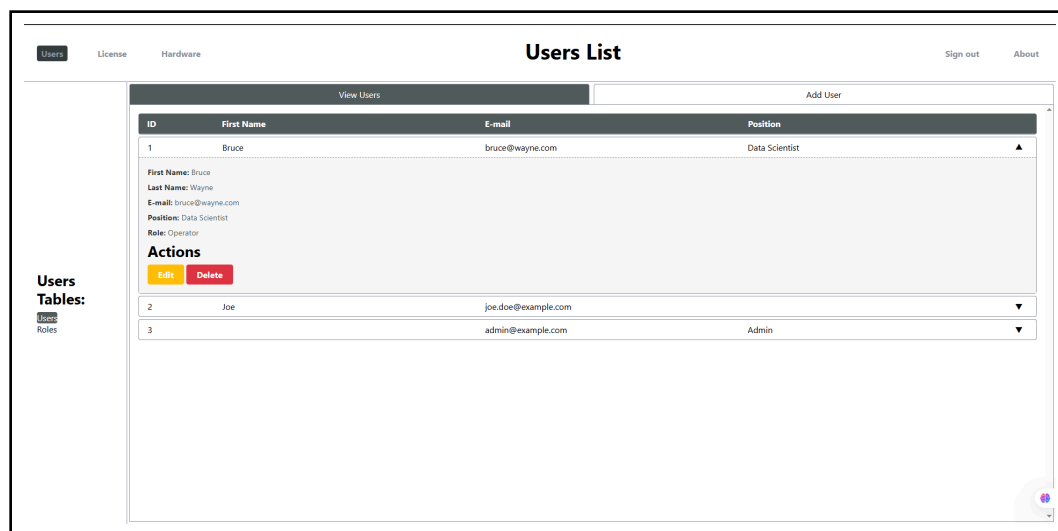
- **Administrator** - przykładowy użytkownik - login: **bwayne**, hasło: **gotham**
Administrator ma dostęp do całej bazy danych natomiast jego główną rolą jest dodawanie, edytowanie oraz usuwanie użytkowników.
- **Operator** - przykładowy użytkownik - login: **pparker**, hasło: **spidey**
Jego rolą jest dodawanie i zarządzanie sprzętem, oprogramowaniem, projektami oraz licencjami w bazie danych.

Aby móc zarządzać sprzętem IT za pomocą aplikacji użytkownik musi się najpierw zalogować na swoje konto. Przykładowe dane logowania zostały podane powyżej.



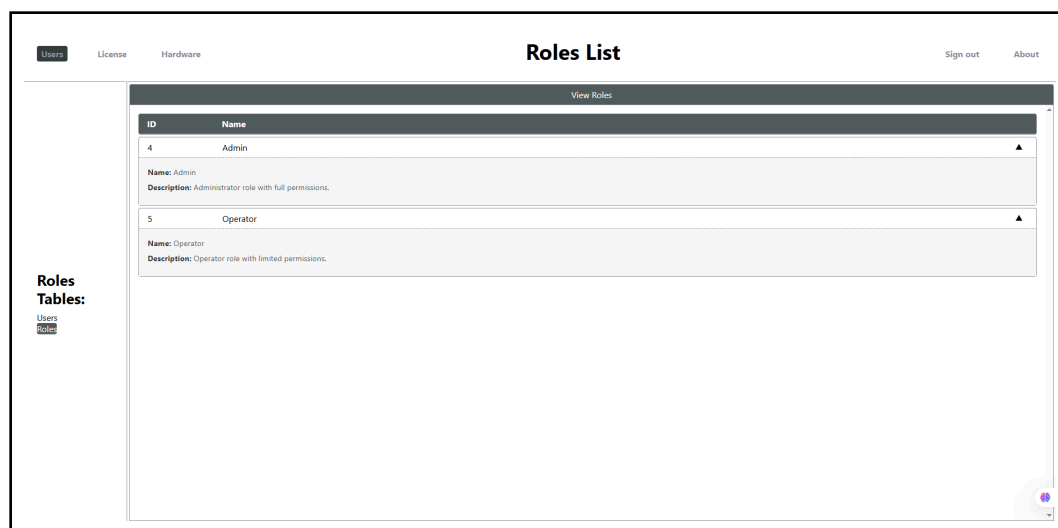
Rysunek 3: Ekran logowania

Po zalogowaniu się jako **Administrator** naszym oczom ukaże się lista użytkowników. W tym miejscu możemy nimi zarządzać.



Rysunek 4: Lista użytkowników

Możemy również wybrać z bocznego paska listę dostępnych ról zaimplementowanych w naszej aplikacji. Na chwilę obecną są to jedynie **Administrator** i **Operator**.



Rysunek 5: Lista Ról

Po powrocie do zakładki Users mamy możliwość dodanie użytkownika po przez naciśnięcie przycisku „Add User”. Następnie ukazuje nam się formularz do wypełnienia wraz z przyciskiem „Add User”, który to wysyła ów formularz.

The screenshot shows a web application interface with a top navigation bar containing 'Users', 'License', and 'Hardware'. On the right of the top bar are links for 'Sign out' and 'About'. The main content area has a tabbed interface with 'View Users' and 'Add User' tabs. The 'Add User' tab is active, displaying a form with the following fields: 'First name:' (text input), 'Last name:' (text input), 'Position:' (text input), 'Role:' (dropdown menu), 'Email address:' (text input), 'Username:' (text input), and 'Password:' (text input). A 'Add User' button is at the bottom of the form. On the left side of the main area, there is a sidebar with the text 'Users Tables:' and a 'Roles' link. The bottom right corner features a small circular icon with a plus sign.

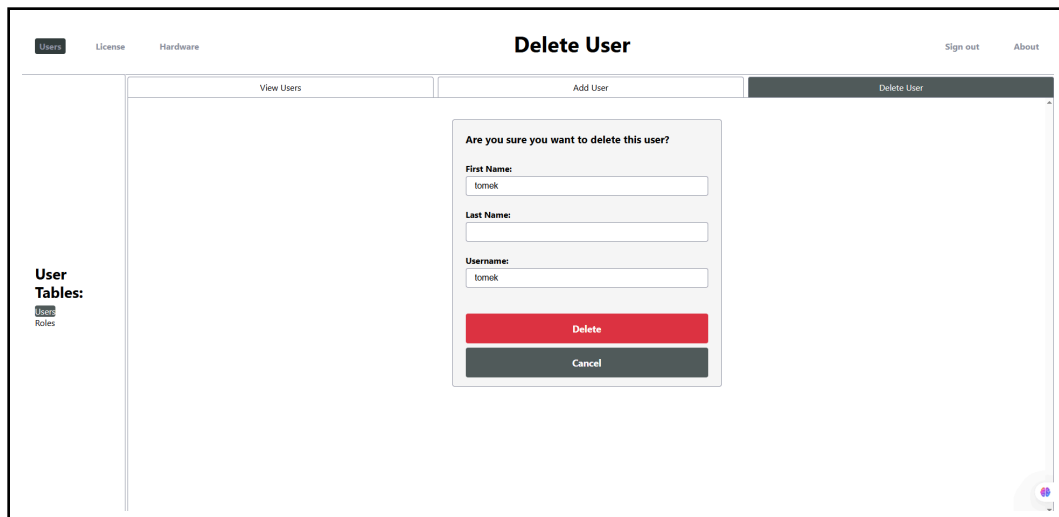
Rysunek 6: Dodawanie użytkownika

Użytkowników również możemy edytować, a robimy to poprzez rozwinięcie jego kafelka oraz kliknięcie przycisku „Edit”. Następnie ponownie otworzy się formularz identyczny jak ten, który wykorzystywany był w celu dodania nowego użytkownika. Zmieniamy interesujące nas dane, a następnie potwierdzamy za pomocą „Save Changes”.

The screenshot shows the 'Edit User' form, which is identical in layout to the 'Add User' form. The top navigation bar and sidebar are the same. The 'Edit User' tab is active. The form fields are pre-filled with the following data: 'First name:' (Bruce), 'Last name:' (Wayne), 'Position:' (Data Scientist), 'Role:' (Operator), 'Email address:' (bruce@wayne.com), 'Username:' (bwayne), and 'Password:' (a hint: 'Leave blank to keep the current password.'). A 'Save Changes' button is at the bottom of the form. The sidebar on the left shows 'User Tables:' and a 'Roles' link. The bottom right corner has the same circular icon.

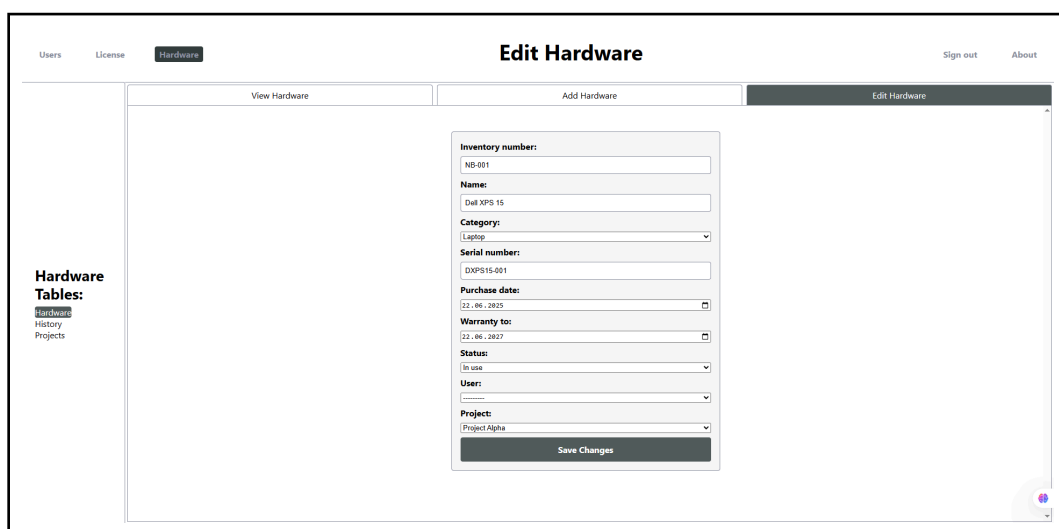
Rysunek 7: Edycja użytkownika

Oprócz edycji i dodawania, mamy również możliwość usuwania użytkowników. Robimy to po przez rozwinięcie jego kafelka oraz kliknięcie przycisku „Delete”. Następnie możemy anulować nasz zamiar po przez naciśnięcie „Cancel”, bądź potwierdzić po przez „Delete”



Rysunek 8: Usuwanie użytkownika

Dla sprzętu, licencji, oprogramowania oraz projektów zarządzanie wygląda dokładnie tak samo. Jedynym wyjątkiem jest edycja sprzętu, gdzie zmiana statusu na inny niż In use powoduje automatyczne usunięcie przypisanego do niego użytkownika i/lub projektu. Jednocześnie przypisanie użytkownika i/lub projektu do sprzętu z dowolnym statusem innym niż In use, powoduje automatyczną zmianę statusu sprzętu na In use.



Rysunek 9: Edycja sprzętu

UsersLicenseHardware

Sign outAbout

Hardware List

Hardware Tables:

HardwareHistoryProjects

View Hardware

Add Hardware

| ID | Inv. Number | Name | Category | Status |
|--|-------------|--------------------------|----------|------------|
| 1 | NB-001 | Dell XPS 15 | | In use |
| 2 | MON-001 | Dell 27inch 4K | | In storage |
| 3 | KEY-001 | Logitech MX Keys | | In use |
| 4 | NB-002 | Apple Macbook Pro | | In use |
| 5 | MON-002 | LG 34inch Ultrawide | | In repair |
| 6 | KEY-002 | Razer Blackwidow | | Utilized |
| 7 | NB-003 | Lenovo Thinkpad | | In use |
| <div>Serial Number: LTP-001</div> <div>Purchase Date: June 22, 2025</div> <div>Warranty To: June 22, 2027</div> <div>Assigned User: bwayne</div> <div>Assigned Project: Project Alpha</div> <div>Actions</div> <div>EditDelete</div> | | | | |
| 8 | DS-001 | CalDigit TS4 | | In use |
| 9 | PRN-001 | HP LaserJet Pro | | In storage |
| 10 | NB-004 | Microsoft Surface Laptop | | In use |

Rysunek 10: Lista sprzętu

UsersLicenseHardware

Sign outAbout

Projects List

Hardware Tables:

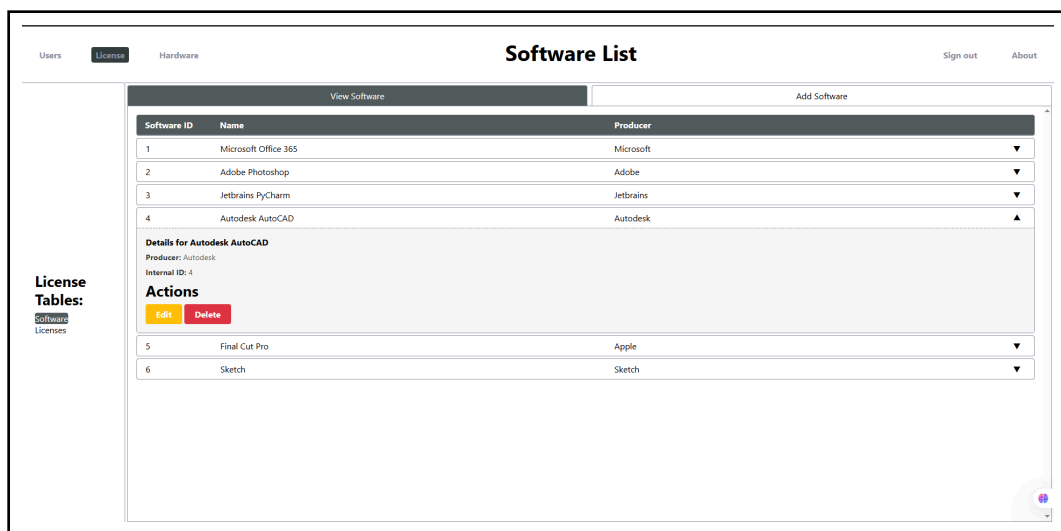
HardwareHistoryProjects

View Projects

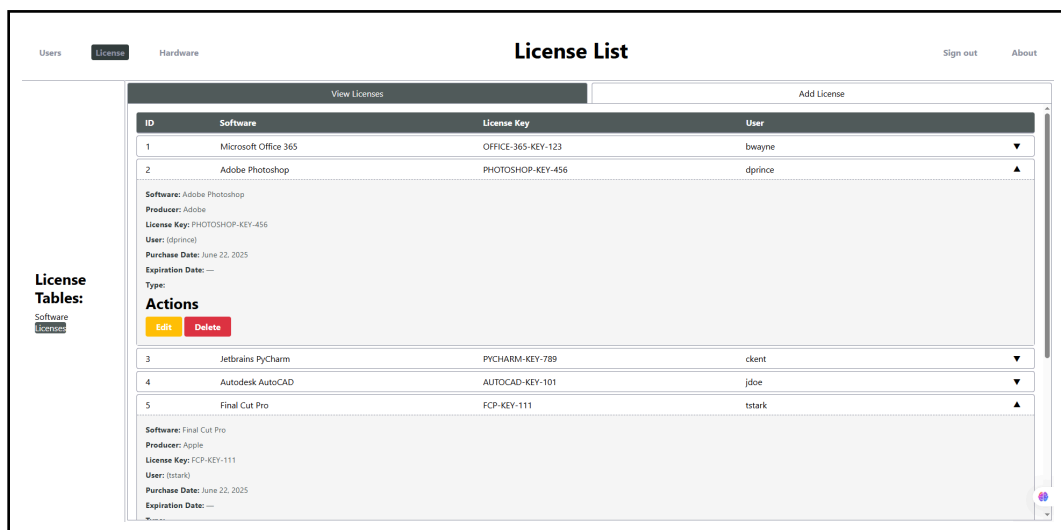
Add Project

| ID | Name | Description | Supervisor |
|---|---------------|-------------|------------|
| 1 | Project Alpha | | |
| 2 | Project Beta | | |
| <div>Full Description:</div> <div>Start Date: June 22, 2025</div> <div>End Date:</div> <div>Actions</div> <div>EditDelete</div> | | | |
| 3 | Project Gamma | | |

Rysunek 11: Lista Projektów



Rysunek 12: Lista oprogramowania



Rysunek 13: Lista licencji

Każda zmiana statusu sprzętu czy też użytkownika i/lub projektu do jakiego jest on przypisany, jest automatycznie zapisywana w historii.

| Date | Hardware | Event Type |
|---|-----------------------------------|---------------------------------|
| June 22, 2025 | Razer Blackwidow (KEY-002) | Created |
| June 22, 2025 | Lenovo Thinkpad (NB-003) | Assigned to user and/or project |
| June 22, 2025 | Microsoft Surface Laptop (NB-004) | Assigned to user and/or project |
| June 22, 2025 | Microsoft Surface Laptop (NB-004) | Created |
| June 22, 2025 | HP LaserJet Pro (PRN-001) | Created |
| June 22, 2025 | CalDigit TS4 (DS-001) | Assigned to user and/or project |
| June 22, 2025 | CalDigit TS4 (DS-001) | Created |
| June 22, 2025 | Lenovo Thinkpad (NB-003) | Assigned to user and/or project |
| June 22, 2025 | Lenovo Thinkpad (NB-003) | Created |
| Description: Hardware created and added to storage via migration. | | |
| June 22, 2025 | Dell XPS 15 (NB-001) | Created |
| June 22, 2025 | LG 34inch Ultrawide (MON-002) | Created |
| June 22, 2025 | Apple Macbook Pro (NB-002) | Assigned to user and/or project |
| June 22, 2025 | Apple Macbook Pro (NB-002) | Created |
| June 22, 2025 | Logitech MX Keys (KEY-001) | Assigned to user and/or project |
| June 22, 2025 | Logitech MX Keys (KEY-001) | Created |

Rysunek 14: Historia

Klikając w zakładkę „About” użytkownik zostaje przeniesiony do strony z informacjami o projekcie oraz jego autorach.

| About the Project | | | |
|--|--|---|---|
| <h3>Inventory & Shop System</h3> <p>This project is a comprehensive inventory and license management system designed to help organizations track their hardware, software licenses, and assigned projects efficiently.</p> <p>Built with Django, this application provides a robust backend and a clean, user-friendly interface for managing assets and user roles.</p> | | | |
| <h3>Authors</h3> | | | |
| Dominik Meisner Lead Developer & Backend Architect Architect of the application's core logic. Responsible for designing the robust backend, creating API endpoints, and ensuring system scalability and performance. https://github.com/b14ucky | Piotr Dusiński Full Stack & Frontend Architect Master of the user-facing experience. Responsible for developing the responsive frontend, integrating it with the backend, and architecting the overall application flow. https://github.com/xPietrUX | Bartek Janoszka DevOps & Cybersecurity Guardian of the infrastructure and security. Manages the deployment pipeline, containerization with Docker, and implements security measures to protect the application. https://github.com/BartekU0110 | Mikołaj Molenda Database Manager & Database Architect Curator of the data. Designs and optimizes the database schema, manages data integrity, and ensures efficient data retrieval and storage. https://github.com/Misiek66 |

Rysunek 15: O aplikacji

4 Wnioski

4.1 Podsumowanie projektu

Udało nam się stworzyć działającą aplikację umożliwiającą zarządzanie sprzętem IT przy pomocy bazy danych. Skutecznie zaimplementowaliśmy w niej wzorzec MVC oraz połączenie z bazą danych. Aplikacja działa według naszych oczekiwań, jak również ma przejrzyste i czytelne UI. Udostępniliśmy aplikację na serwerze VPS *Mikr.us*: klik.

4.2 Potencjał rozwoju

- Dodanie opcji filtrowania
- Wdrożenie aplikacji na chmurę (AWS, Google Cloud, MS Azure)
- Wdrożenie opcji podglądu dla zwykłych użytkowników (użytkownik może po zalogowaniu się podejrzeć przypisany do niego sprzęt lub licencje)
- Połączenie użytkowników z AD, bądź AAD
- Rozbudowa bazy o dodatkowe parametry/kolumny np. dodanie do tabeli projekt miejsca realizacji, dodanie tabel do zarządzania numerami telefonicznymi oraz abonamentami/usługami

5 Link do repozytorium GitHub

- <https://github.com/xPietrUx/Inventory-Shop-System>