

Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: **Informatyka techniczna (ITE)**
Specjalność: **Inżynieria systemów informatycznych (INS)**

PRACA DYPLOMOWA
INŻYNIERSKA

**Budowa narzędzia wspomagającego
zarządzanie sprzętem komputerowym**

**Building a tool to support the
management of computer equipment**

Maurycy Niewczas

Opiekun pracy
dr inż. Tomasz Kubik

Słowa kluczowe: inwentaryzacja sprzętu, loteria sprzętu, aplikacja internetowa

WROCŁAW, 2024

Spis treści

1. Wstęp	8
1.1. Wprowadzenie	8
1.2. Cel pracy	9
1.3. Zakres pracy	9
1.4. Układ pracy	10
2. Wykorzystane narzędzia i technologie	11
2.1. Warstwa logiki biznesowej	11
2.1.1. Java	11
2.2. Frameworki Java	12
2.2.1. Spring oraz Spring Boot	12
2.2.2. Hibernate	13
2.3. Warstwa interfejsu użytkownika	13
2.3.1. Javascript i React	13
2.3.2. Arkusze stylów	13
2.4. Środowisko deweloperskie	13
2.4.1. Maven	13
2.4.2. IntelliJ IDEA	14
2.4.3. SQL Server Management Studio	14
2.4.4. Postman	14
2.4.5. Lighthouse	14
3. Analiza wymagań	15
3.1. Wymagania funkcjonalne	15
3.2. Wymagania niefunkcjonalne	21
3.2.1. Ogólnodostępność	21
3.2.2. Niezawodność	22
3.2.3. Bezpieczeństwo	22
3.2.4. Wydajność	22
3.2.5. Koszty	22
4. Baza danych	23
4.1. Model bazy danych	23
4.1.1. Opisy encji urządzeń	23
4.1.2. Dziedziczenie tabel	24
4.2. Tworzenie bazy danych	25
5. Implementacja	27
5.1. Architektura	27
5.1.1. Struktura interfejsu REST	27
5.2. Aplikacja serwerowa	29
5.2.1. Struktura projektu	30
5.2.2. Fragmenty implementacji	31

5.3. Aplikacja kliencka	34
5.3.1. Struktura projektu	35
5.3.2. Fragmenty implementacji	37
5.3.3. Instrukcja użytkowania	38
6. Testy aplikacji	44
6.1. Testowanie restowych zapytań przy pomocy Postmana	44
6.1.1. Wybrane scenariusze testowe	45
6.2. Testowanie jakości strony internetowej systemu przy użyciu Lighthouse	48
Literatura	49
A. Instrukcja wdrożeniowa	50
A.1. Wymagane narzędzia	50
A.2. Konfigurowanie środowiska	50

Spis rysunków

4.1. Diagram związków encji	23
4.2. Tworzenie bazy danych SSMS a) model, b) definiowanie relacji	25
4.3. Generowaniu kodu na podstawie tabeli computer przy użyciu IntelliJ IDEA	26
5.1. Struktura projektu: a) ogólna struktura, b) kontrolery, c) dto, d) encje, e) repozytoria, f) serwisy	30
5.2. Struktura projektu: a) ogólna struktura, b) Logowanie i rejestracja, c) pasek nawigacyjny, d) formularze komponentów, e) formularze sprzętów, f) widoki stron	36
5.3. Widok W-1, Logowanie i Rejestracja: a) logowanie, b) rejestracja	38
5.4. Widok W-2, Strona domowa: a) Administrator bez filtrowania, b) Administrator z filtrowaniem, c) Pracownik bez filtrowania	39
5.5. Widok W-3 dodawania urządzenia a) wariant komputera: b) wariant tabletu, c) wariant innego urządzenia	40
5.6. Widok W-4 formularzy modyfikacji: a) komputera: b) tabletu, c) innego urządzenia	41
5.7. Widok W-5 formularzy informacji: a) o komputerze, b) o tablecie, c) o innym sprzęcie	41
5.8. Widok W-6, Informacje o użytkownikach: a) Zarządzanie użytkownikami przez administratora, b) Przeglądanie uczestników losowania	42
5.9. Widok W-7, Historia loterii pracownika	42
5.10. Widok W-8, Widok komponentów komputera	43
5.11. Widok W-9, Formularze komponentów: a) procesor, b) ram, c) dysk pamięci	43
6.1. Struktura testów a) po id, b) wszystkie urządzenia, c) sortowanie po biurze, d) dodawanie, e) aktualizacja, f) usuwanie, g) związane z loterią, h) związane z użytkownikiem	44
6.2. Scenariusz testowy 1, Pobieranie informacji o urządzeniu: a) komputer, b) inne urządzenie	45
6.3. Scenariusz testowy 2, Dodawanie i modyfikowanie komputera: a) dodawanie, b) modyfikacja	46
6.4. Scenariusz testowy 3, Usuwanie urządzenia po ID	46
6.5. Scenariusz testowy 3, Losowanie zwycięzcy: a) powodzenie, b) niepowodzenie	47
6.6. Scenariusz testowy 4, Rejestracja użytkownika	47
6.7. Test Lighthouse strony domowej, a) administratora, b) pracownika	48
A.1. Konfiguracja TCP/IP	50
A.2. Konfiguracja konta a) uprawnienia, b) zmiana hasła, c) typ uwierzytelniania	51

Spis tabel

2.1. Zestawienie wykorzystanych narzędzi i technologii wraz z ich wersjami i licencjami	11
5.1. Tabela prezentująca strukturę REST (cz.1)	28
5.2. Tabela prezentująca strukturę REST (cz.2)	29
5.3. Zestawienie ścieżek url dostępnych w aplikacji klienckiej	36
A.1. Loginy i hasła aplikacji klienckiej	51

Spis listingów

5.1. Klasa nadrzędna reprezentująca rdzeń sprzętu: DeviceCore	31
5.2. Klasa potomna: Computer, reprezentująca komputer	31
5.3. Generyczne repozytorium sprzętu komputerowego: GenericDeviceRepository . . .	32
5.4. Klasa nadrzędna kontrolera sprzętu: GenericDeviceController	32
5.5. Klasa potomna reprezentująca ogólną postać sprzętu	33
5.6. Klasa nadrzędna serwisu sprzętu GenericDeviceService	33
5.7. Klasa potomna serwisu tabletu: TabletService	34
5.8. Zdefiniowane ścieżki url widoków systemu	37
5.9. Obsługa formularzy	37
5.10. Przykładowe zapytanie do serwera dla formularza komputera	38

Skróty

SSMS (àng *SQL Server Managment Studio*)

JPA (àng *Java Persistance API*)

FR (àng *Functional Requirements*)

ORM (àng *Object Relational Mapping*)

REST (àng *Representational State Transfer*)

HTTP (àng *Hypertext Transfer Protocol*)

CSS (àng *Cascading Style Sheets*)

API (àng *Application Programming Interface*)

SEO (àng *Search Engine Optimization*)

Rozdział 1

Wstęp

1.1. Wprowadzenie

Dzięki dynamicznemu rozwojowi technologii informacyjnych oraz znaczącemu postępowi w projektowaniu i wytwarzaniu użytkowej elektroniki na rynku powiązanych z nimi usług i produktów pojawiają się coraz to nowsze i ciekawsze oferty. Dzieje się to w sposób niemal ciągły, zapewniający postęp cywilizacyjny. Temu zjawisku towarzyszą jednak skutki uboczne. Do takich należą, między innymi, problemy szybkiego starzenia się sprzętu komputerowego. Uwidaczniające się one szczególnie dotkliwie w firmach z branży IT. Dążąc do zwiększenia zysków oraz obniżenia kosztów własnych menadżerowie tych firm muszą zapewnić pracownikom odpowiedni warsztat pracy (w tym komputery, laptopy itp.), co pociąga za sobą konieczność dokonywania okresowej wymiany sprzętu oraz jego utylizacji.

Wymiana i utylizacja sprzętu zwykle jest uwarunkowana jego zdolnością do użytkowania. Przy czym nie chodzi tu jedynie o fizyczną sprawność poszczególnych urządzeń. Często wymieniany i utylizowany sprzęt jest w pełni funkcjonalny, ale z uwagi na zmiany w obszarze technologii, nie da się go już wykorzystać na tzw. „produkcji”. Taki sprzęt szkoda poddawać automatycznej kasacji. Dużo lepszym pomysłem jest jego odsprzedaż.

Pomysł odsprzedaży wysłużonego sprzętu można zrealizować na różne sposoby. Na przykład uruchamiając loterię adresowaną do pracowników danej firmy. Sposób ten daje możliwość osiągnięcia kilku korzyści: uzyskania częściowego zwrotu kosztów poniesionych na zakup sprzętu, zwiększenia poziomu integracji członków firmy, wdrożenia mechanizmów motywacyjnych itp.

Wyniki inwentaryzacji sprzętu zwykle zapisywane są w postaci tabelarycznej. Do ich obsługi (edycji) wykorzystuje się narzędzia typu Office, a mówiąc dokładniej – edytory arkuszy kalkulacyjnych. W podstawowym scenariuszu aby przygotować i przeprowadzić loterię wystarczy sięgać do odpowiedniego arkusza kalkulacyjnego i nanieść w nim odpowiednie zmiany. Zdarza się jednak, że informacje dotyczące danej loterii jest rozproszona pomiędzy kilkoma arkuszami. Dlatego lepszym rozwiązaniem byłoby uruchomienie osobnej aplikacji, działającej zgodnie z przyjętymi w firmie regułami. Chęć zaimplementowania takiej aplikacji stała się motywem do zdefiniowania tematu oraz celu niniejszej pracy.

Co prawda na rynku istnieją rozwiązania, które pozwalają zarządzać sprzętem komputerowym. Jednymi z ciekawszych są **OCS Inventory** oraz **Microsoft System Center Configuraton Manager**. Jednak oba te rozwiązania stworzono z myślą o zarządzaniu sprzętem, a nie o jego odsprzedaży. Ponadto z uwagi na mnogość oferowanych funkcji, skorzystanie z tych narzędzi jest zaledwie skomplikowane, by opłacało się je wykorzystywać do sprzedaży urządzeń wewnątrz firmy. Warto jednak spojrzeć na ich możliwości przed przystąpieniem do implementacji własnej aplikacji.

OCS Inventory jest darmowym oprogramowaniem które po instalacji jest w stanie pobrać informacje o sprzęcie. Informacje te dotyczą procesora, pamięci RAM, dysku twardego podłączonych urządzeń peryferyjnych, oprogramowania, kart sieciowych.

Schemat działania

- serwer bazodanowy przechowujący informacje o inwentarzu(mysql),
- serwer komunikacyjny, który obsługuje komunikacje HTTP między serwerem bazy danych a agentami (Apache, perl i mod_perl),
- konsola administracyjna, która pozwala administratorom odpytywać serwer bazy danych za pomocą przeglądarki(Apache, php),
- serwer wdrażania, który przechowuje wszystkie konfiguracje pakietów (Apache, ssl).

Rozwiązanie jest wieloplatformowe, które działa na systemach: Windows, Mac, Linux oraz Android. Dodatkowo istnieje płatna wersja narzędzia, która pozwala na pomoc administratorów OCS Inventory. Narzędzie pozwala na zbieranie danych w postaci tabelarycznej oraz generowania raportów dotyczących sprzętu.

Microsoft System Center Configuration Manager jest bardzo dużym i płatnym rozwiązaniem dostarczającym przez firmę Microsoft. Pozwala ono na zarządzanie dużą grupą komputerów. Do możliwości narzędzia należą między innymi: zdalny dostęp, menadżer aktualizacji oraz monitorowania stanu technicznego urządzeń. Bardzo dużo możliwości dostarczanych przez ten produkt sprawia, że cena jest nieadekwatna gdy firma chce wykorzystywać to narzędzie tylko do inwentaryzacji sprzętu. Dodatkowe opłaty mogą zostać również wygenerowane przez to, że firma musi wyszkolić pracowników, którzy umieli by zarządzać tym oprogramowaniem.

1.2. Cel pracy

Celem pracy jest stworzenie kompleksowego narzędzia umożliwiającego ocenę stanu technicznego urządzeń wycofywanych z użytku. Narzędzie to powinno pomóc w stworzeniu efektywnego systemu wyceny wartości tych urządzeń oraz w procesie ich odsprzedaży.

Projekt zakłada przeprowadzenie inwentaryzacji sprzętu, umożliwiając zapisanie stanu technicznego urządzeń oraz ewentualnych usterek, które zazwyczaj wymagają fizycznych oględzin. Po wdrożeniu, system powinien być dostępny na wybranej platformie internetowej, umożliwiając zalogowanym użytkownikom wykonywanie działań zgodnie z nadanymi uprawnieniami. Przewidziane akcje obejmują przeglądanie, modyfikowanie i usuwanie urządzeń oraz zarządzanie loteriami i użytkownikami. Dodatkowo, użytkownicy powinni mieć możliwość uczestniczenia w loteriach oraz przeprowadzania losowań.

Rozwiązanie ma charakteryzować się przyjaznym interfejsem użytkownika, zapewniając łatwą nawigację nawet dla pracowników bez specjalistycznej wiedzy technicznej. Proces przeglądania sprzętu komputerowego powinien być intuicyjny, usprawniając monitorowanie stanu technicznego i statusu loterii. Prostota systemu ma wpłynąć na ogólną wygodę użytkowników oraz przyczynić się do redukcji kosztów związanych z procesem inwentaryzacji.

1.3. Zakres pracy

Projektowany system składać się będzie z dwóch kluczowych komponentów: **aplikacji klienckiej** oraz **aplikacji serwerowej**. Aplikacja serwerowa, będąca centralnym elementem systemu, ma odpowiadać za komunikację z bazą danych oraz dostarczanie interfejsu REST konsumowanego przez aplikację kliencką. Aplikacja kliencka dostarczy ma interfejs użytkownika, poprzez który możliwe będzie przeglądanie, usuwanie oraz korekta danych dotyczących urządzeń, histo-

rii przeprowadzonych i aktualnych loterii. Dodatkowo, aplikacja kliencka umożliwi przeprowadzenie losowań oraz jednoznaczne wyłonienie zwycięzcy.

Projekt ma powstać z myślą o elastyczności i łatwości zarządzania danymi. Będzie umożliwiał dynamiczne dodawanie, usuwanie i modyfikację danych, zapewniając tym samym pełną funkcjonalność oraz dostosowanie do zmieniających się potrzeb użytkowników. W celu zapewnienia bezpieczeństwa dane wrażliwe przetwarzane w systemie takie jak hasła, będą szyfrowane.

1.4. Układ pracy

Praca składa się z 6 rozdziałów:

- Rozdział 1 – skupia się na zagadnieniach związanych z celem i zakresem pracy. Posiada też przegląd istniejących rozwiązań które mają zbliżone funkcjonalności do projektowanego systemu,
- Rozdział 2 – skupia się na opisie wykorzystywanych narzędzi i technologii,
- Rozdział 3 – posiada wymagania funkcjonalne i нефункционалне systemu. Wymagania funkcjonalne szczegółowo opisują jak ma działać system,
- Rozdział 4 – skupia się na zaprojektowaniu, stworzeniu i wdrożeniu bazy danych,
- Rozdział 5 – posiada szczegółowe informacje na temat struktury projektu aplikacji klienckiej i serwerowej. Opisane zostały w nim listingi kodu oraz wygląd i sposób działania systemu,
- Rozdział 6 – skupia się na przeprowadzeniu testów przy użyciu Postman i Lighthouse,
- Dodatek A – jest instrukcją skonfigurowania i uruchomienia systemu. Zawiera również dostęp do aplikacji klienckiej.

Rozdział 2

Wykorzystane narzędzia i technologie

Rozwijana aplikacja powstanie w oparciu o architekturę klient-serwer. Podczas implementacji tego typu aplikacji wyróżnia się technologie wykorzystywane do realizacji logiki biznesowej oraz technologie służące do budowy interfejsu użytkownika. Ponadto sama implementacja odbywa się w środowisku deweloperskim, zapewniającym dostęp do niezbędnych narzędzi. Zdarza się jednak i tak, że te same technologie i narzędzia pojawiają się w na każdym etapie i przy implementacji różnych części aplikacji. W tabeli 2.1 wymieniono główne narzędzia i technologie wykorzystanych podczas realizacji pracy. W kolejnych podrozdziałach podano szczegóły.

Tab. 2.1: Zestawienie wykorzystanych narzędzi i technologii wraz z ich wersjami i licencjami

Technologia	Wersja	Licencja
Java	Oracle OpenJDK 17.0.8 17	NFTC
Maven	3.9.5	Apache License
Springboot	3.1.4	Apache 2.0
Hibernate	6.1.7.Final	GNU Lesser General Public License
Javascript	v18.17.0	MIT
React	18.2.0	MIT
SQL Server Managment Studio	19 Standard Edition	Microsoft Software License
Lighthouse	100.0.0.3	Apache License 2.0

2.1. Warstwa logiki biznesowej

2.1.1. Java

Java to termin, który w dziedzinie informatyki jest kojarzony z platformą do budowy aplikacji (ang. *Java Platform*), obiektywnym językiem programowania (ang. *Java language*) oraz z wirtualną maszyną (ang. *Java Virtual Machine*). Technologia ta została opracowana przez firmę Sun Microsystems w roku 1995. Jest rozwijana aż do dzisiaj i doczekała się wielu wersji.

Rozwój Javy odbywa się w ramach JCP (ang. *Java Community Process*). Każdy użytkownik może brać udział w recenzowaniu i dostarczaniu informacji zwrotnej dla publikowanych specyfikacji JSR (ang. *Java Specification Request*). Specyfikacje składają się z powiązanych ze sobą dokumentów, w tym specyfikacji platformy, języka i maszyny wirtualnej. Na podstawie tych specyfikacji wydawcy tworzą własne implementacje Javy.

Język Java – jest współbieżnym, opartym na klasach, obiektowym językiem programowania. Java wykorzystuje tworzenie programów źródłowych kompilowanych do kodu bajtowego. Kod bajtowy jest interpretowany przez maszynę wirtualną Javy.

Maszyna Wirtualna Javy – jest zestawem aplikacji napisanych na tradycyjne urządzenia i systemy operacyjne. Jest środowiskiem zdolnym do wykonywania kodu bajtowego Javy. Oferuje też automatyczne zarządzanie pamięcią przez *garbage collector*. Implementacja wirtualnej maszyny jest napisana w C++. Dostępna publicznie specyfikacja umożliwiła różnym producentom na tworzenie własnych wirtualnych maszyn przez co istnieje wiele różnych niezależnych wersji. Łączy je wszystkie zgodność ze specyfikacją języka Java.

Platforma Java – oferuje programy i narzędzia, które wspierają budowę oraz uruchomienie aplikacji. Aplikacja napisana w ramach pracy dyplomowej została stworzona w wersji Java 17. Wersja ta domyślnie została zainstalowana w środowisku programistycznym. Jest to wersja stabilna, o długim okresie wsparcia.

2.2. Frameworki Java

Framework to platforma programistyczna, która jest szkieletem do budowy aplikacji. Dostarcza gotowe narzędzia, biblioteki i komponenty do wykonywania określonych zadań. Korzystając z frameworków można uprościć pisanie aplikacji, ponieważ narzędzia, biblioteki i komponenty dostarczają gotowe rozwiązania. Programista skupia się wtedy na specyficznych zadaniach, bez konieczności implementacji rozwiązań, które dostarcza framework.

2.2.1. Spring oraz Spring Boot

Spring Framework jest platformą stworzoną w celu uproszczenia tworzenia oprogramowania dla platformy Java co dobrze opisano w literaturze [10]. Oferuje on kontener, który jest określony mianem kontekstu springa. Do jego zadań należą tworzenie i zarządzanie komponentami aplikacji. Komponenty te są powiązane ze sobą tworząc spójną całość. Sposób w jakim są łączone ze sobą komponenty (tzw. ziarenka, ang. *beans*) to wstrzykiwanie zależności (ang. *dependency injection*). Wstrzykiwanie zależności polega na tym, że zamiast obsługi życiowych komponentów *bean* następuje zdefiniowanie oddzielnej encji(kontenera), która tworzy i przechowuje wszystkie komponenty. Przechowywane komponenty będą wstrzykiwane do potrzebujących ich komponentów. Dzieje się to za sprawą argumentów konstruktora lub metod akcesora właściwości.

Spring Boot jest rozszerzeniem Springa, który eliminuje konieczność konfiguracji środowiska charakterystyczną dla samego Springa. Oferuje on także wiele ciekawych funkcji, takich jak: analiza wewnętrznego sposobu działania aplikacji w środowisku uruchomieniowym, elastyczna specyfikacja właściwości środowiska oraz dodatkowe możliwości w zakresie obsługi testów. Więcej o Springu opisane jest w oficjalnej dokumentacji [3]

Spring Data – Spring Data pozwalają na zdefiniowanie repozytorium danych aplikacji w postaci interfejsów Javy. Wykorzystując te interfejsy, dostarcza metody dostępu do danych. Umożliwia pracę z wieloma rodzajami baz danych przy użyciu JPA (ang. *Java Persistence API*)

Spring Web – warstwa sieciowa Springa składa się z modułów Web, Web-Servlet, Web-Struts i Web-Portlet. Moduł Web w ramach frameworku Spring dostarcza podstawowe funkcje integracyjne zorientowane na sieć. Oferuje moduły do implementacji funkcji związanych z warstwą

prezentacji, obsługą żądań HTTP, kontrolerami, widokami a także integracją z innymi modułami Springa.

2.2.2. Hibernate

Hibernate jest frameworkiem służącym do realizacji warstwy dostępu do danych. Zapewnia on przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiekowym co opisuje literatura [6]. Hibernate pozwala na automatyczne mapowanie obiektów Javy na wiersze z bazy danych oraz odczytywać rekordy z bazy i automatycznie tworzyć z nich obiekty. Wykorzystując Hibernate nie trzeba poświęcać uwagi na zapytania SQL do bazy danych bo framework robi to sam.

2.3. Warstwa interfejsu użytkownika

2.3.1. Javascript i React

JavaScript – jest skryptowym językiem programowania, który jest powszechnie stosowany do tworzenia stron internetowych. Pozwala on na dynamiczne dodawanie funkcji dla stron internetowych. Wykonywany jest na przeglądarce internetowej, to znaczy, że kod JavaScript może być uruchomiany na różnych platformach i systemach operacyjnych.

React – jest biblioteką dla JavaScript. Wykorzystywana jest do tworzenia interfejsów graficznych aplikacji internetowych. Pozwala na budowanie dynamicznych i efektywnych aplikacji front-endowych. React pozwala na tworzenie interaktywnego interfejsu użytkownika, który reaguje na zmiany danych i efektywne odświeżanie widoku użytkownika co opisano w literaturze [9].

2.3.2. Arkusze styli

Arkusze styli definiują sposób prezentacji i stylizacji dokumentów HTML lub XML. Używane są do oddzielenia treści strony do jej wyglądu. Pozwalają na zarządzanie wyglądem witryny.

CSS (ang. *Cascading Style Sheets*) – jest najpopularniejszym językiem arkuszy styli. Wykorzystano go również w projektowanym systemie.

2.4. Środowisko deweloperskie

2.4.1. Maven

Apache Maven jest narzędziem oferującym automatyczną budowę oprogramowania na platformę Java. Poszczególne funkcje Mavena są realizowane poprzez wtyczki. Proces budowania kończy się osiągnięciem wybranego przez budującego celu. Cele umożliwiają między innymi na kompilację, zbudowanie paczki dystrybucyjnej oraz na uruchomienie testów automatycznych. Maven korzysta z pliku POM (ang. *Project Object Model*). POM to dokument XML-owy o nazwie `pom.xml`, który opisuje projekt. W pliku tym można zdefiniować zależności które będą wykorzystane w projekcie.

2.4.2. IntelliJ IDEA

IntelliJ IDEA jest zintegrowanym środowiskiem programistycznym który oferuje wiele narzędzi pomagających w programowaniu. W wersji ultimate która dostępna jest na licencji studenckiej występuje wsparcie do integracji i zarządzania bazą danych.

2.4.3. SQL Server Managament Studio

SSMS jest aplikacją wydaną przez Microsoft. Służy ona zarządzania, konfigurowania i administrowania bazą danych. Umożliwia między innymi łączenie z serwerami baz danych SQL Server, projektowanie tabel i indeksów, zarządzanie relacjami, kontrolowanie uprawnień dostępu do danych. Szczegółowa konfiguracja środowiska jest opisana w dokumentacji Microsoftu [7]

2.4.4. Postman

Postman jest narzędziem do testowania interfejsów API. Umożliwia on wysyłanie zapytań HTTP dla różnych metod, takich jak GET, POST, PUT, PATCH, DELETE do serwera i analizowanie odpowiedzi. Pozwala on także zdefiniować tokeny których wymaga serwer w celu uwierzytelniania. Zwraca on także statusy HTTP wykonanych zapytań.

2.4.5. Lighthouse

Lighthouse jest otwartym narzędziem do analizy jakości stron internetowych. Pomaga deweloperom, projektantom i administratorom stron internetowych w ocenie i poprawie różnych aspektów stron internetowych.

Rozdział 3

Analiza wymagań

3.1. Wymagania funkcjonalne

Projektowany system powinien zapewnić szereg funkcji do potencjalnego wykorzystania przez jego użytkowników. Poniżej wypisano te funkcje w podziale z uwagi na ich zastosowanie. Dodatkowo dostarczono opis podstawowych przypadków ich użycia.

FR-1 Zarządzanie użytkownikami:

- FR-1.1 rejestracja użytkownika,
- FR-1.2 logowanie użytkownika,
- FR-1.3 usuwanie pracownika,
- FR-1.4 wylogowywanie użytkownika,
- FR-1.5 przeglądanie zarejestrowanych pracowników.

FR-2 Zarządzanie sprzętem komputerowym:

- FR-2.1 przeglądanie sprzętu komputerowego,
- FR-2.2 dodawanie nowego sprzętu komputerowego,
- FR-2.3 modyfikacja sprzętu komputerowego,
- FR-2.4 usuwanie sprzętu komputerowego,
- FR-2.5 wyświetlanie szczegółowych informacji dotyczących sprzętu komputerowego.

FR-3 Zarządzanie komponentami komputera:

- FR-3.1 przeglądanie komponentów komputera,
- FR-3.2 dodawanie komponentu(procesora, ram, dysku) dla komputera,
- FR-3.3 modyfikacja komponentu komputera,
- FR-3.4 usuwanie komponentu komputera.

FR-4 Zarządzanie loterią wybranego sprzętu komputerowego

- FR-4.1 zmiana gotowości sprzętu do losowania,
- FR-4.2 zmiana statusu sprzętu na sprzedane lub niesprzedane,
- FR-4.3 wyświetlanie listy uczestników biorących udział w losowaniu,
- FR-4.4 losowanie zwycięzcy loterii.

FR-5 Zarządzanie uczestnictwem w loterii

- FR-5.1 przeglądanie historii loterii wybranego użytkownika,
- FR-5.2 zapisywanie się na loterię urządzenia,
- FR-5.3 wypisywanie się z loterii urządzenia.

FR-1.1 Rejestracja użytkownika

Opis: Rejestracja użytkownika ze standardową rolą, przy użyciu maila, nazwy użytkownika, imienia, nazwiska, hasła i potwierdzenia hasła.

Aktorzy: Pracownik

Warunki początkowe: Brak.

Przebieg podstawowy:

1. Podanie w formularzu maila, nazwy użytkownika, imienia, nazwiska, hasła i potwierdzenia hasła.
2. Kliknięcie przycisku zarejestruj.
3. Przekierowanie do okna logowania.

Przebieg alternatywny 1 (odrzućenie próby logowania):

5. Użytkownik wprowadził 2 różne hasła w polach hasło i powtórz hasło przez co rejestracja nie powiodła się.
6. Użytkownik o zadanym mailu lub nazwie użytkownika istnieje, przez co rejestracja nie powiodła się.

FR-1.2 Logowanie użytkownika

Opis: Logowanie użytkownika przy pomocy email lub nazwy użytkownika oraz hasła.

Aktorzy: Pracownik, Administrator

Warunki początkowe: Użytkownik został wcześniej zarejestrowany w systemie.

Przebieg podstawowy:

1. Podanie w formularzu nazwy użytkownika lub email oraz podanie hasła.
2. Kliknięcie przycisku zaloguj.
3. Nadanie użytkownikowi odpowiednich uprawnień.
4. Przekierowanie na stronę domową.

Przebieg alternatywny 1 (odrzućenie próby logowania):

4. Podane dane w formularzu są nieprawidłowe, przez co nie następuje logowanie i pozostaje na stronie logowania.

FR-1.3 Usuwanie pracownika

Opis: Administrator jako zarządca systemu może usunąć użytkownika.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Include:

1. FR-1.5 Przeglądanie zarejestrowanych pracowników.

Przebieg podstawowy:

1. Kliknięcie przycisku usuń na wierszu reprezentującym pracownika.
2. Usunięcie z bazy danych pracownika oraz jego historii uczestnictwa w loteriach.

FR-1.4 Wylogowywanie użytkownika.

Opis: Wylogowywanie z systemu

Aktorzy: Pracownik, Administrator

Warunki początkowe: Użytkownik został wcześniej zalogowany.

Przebieg podstawowy:

1. Kliknięcie przycisku wyloguj.
2. Przejście na stronę logowania.

FR-1.5 Przeglądanie zarejestrowanych pracowników

Opis: Możliwe jest wyświetlanie wszystkich użytkowników z rolą Pracownik.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada

uprawnienia administratora.

Przebieg podstawowy:

1. Kliknięcie przycisku zarządzaj użytkownikami.
2. Wyświetlenie widoku użytkowników.

FR-2.1 Przeglądanie sprzętu komputerowego

Opis: Administrator widzi wszystkie sprzęty dodane w systemie, natomiast pracownik tylko te które są gotowe do loterii.

Aktorzy: Pracownik, Administrator

Warunki początkowe: Użytkownik jest zalogowany w systemie.

Extend:

1. Filtrowanie jaki rodzaj sprzętu ma być wyświetlany na widoku lub wybranie wszystkich rodzajów.
2. Filtrowanie sprzętu po biurze w jakim się znajdują.

Przebieg podstawowy:

1. Wyświetlenie widoku sprzętów komputerowych na stronie domowej, z dostępnymi operacjami zarządzania sprzętem i loteriami tych sprzętów.

Przebieg alternatywny:

- 1b Użytkownik o dostępie pracownika nie ma dostępu do operacji zarządzania sprzętem.

FR-2.2 Dodanie nowego sprzętu komputerowego

Opis: Administrator może dodawać 3 różne rodzaje urządzeń zdefiniowanych w systemie.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Extend:

1. FR-3.2 Dodawanie komponentu dla komputera.

Przebieg podstawowy:

1. Kliknięcie przycisku odpowiedzialnego za dodanie urządzenia.
2. Wybranie z listy rozwijanej typu urządzenia, które będzie dodawane.
3. System wyświetla formularz z odpowiednim rodzajem urządzenia.
4. Wpisanie oraz wybranie odpowiednich parametrów reprezentujących urządzenie.
5. Dodanie urządzenia do systemu.

Przebieg alternatywny:

4. Administrator nie znalazł w formularzu komputera; procesora, ram, lub dysku i postanowił dodać taki klikając odpowiedni przycisk: dodaj procesor, dodaj ram lub dodaj pamięć.
5. Następuje teraz wykonanie przypadku FR-3.2 Dodawanie komponentu dla komputera.
6. Kontynuacja uzupełniania formularza.
7. Dodanie nowego komputera.

FR-2.3 Modyfikacja sprzętu komputerowego

Opis: Możliwa jest korekta urządzeń lub podanie brakujących parametrów urządzenia.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku modyfikuj na odpowiednim wierszu reprezentującym urządzenie.

2. System wykrywa odpowiedni rodzaj urządzenia i wyświetla formularz z poprzednimi danymi urządzenia.
3. Wpisanie oraz wybranie odpowiednich parametrów reprezentujących urządzenie.
4. Modyfikacja urządzenia.

Przebieg alternatywny:

5. Administrator nie znalazł w formularzu komputera; procesora, ram, lub dysku i postanowił dodać taki klikając odpowiedni przycisk: dodaj procesor, dodaj ram lub dodaj pamięć.
6. Następuje teraz wykonanie przypadku FR-3.2 Dodawanie komponentu dla komputera.
7. Kontynuacja uzupełniania formularza.
8. Modyfikacja komputera.

FR-2.4 Usuwanie sprzętu komputerowego

Opis: Administrator może usuwać urządzenia.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku usuń na odpowiednim wierszu reprezentującym urządzenie.
2. Usunięcie sprzętu z systemu.

FR-2.5 Wyświetlanie szczegółowych informacji dotyczących sprzętu

Opis: Możliwe jest szczegółowe sprawdzenie dotyczące parametrów sprzętów w specjalnym formularzu.

Aktorzy: Administrator, Pracownik

Warunki początkowe: Użytkownik został zalogowany do systemu.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku informacji o urządzeniu na odpowiednim wierszu reprezentującym urządzenie.
2. System wykrywa odpowiedni rodzaj urządzenia i wyświetla formularz urządzenia z zablokowanymi polami.

FR-3.1 Przeglądanie komponentów komputera

Opis: Do lepszego oszacowania ceny sprzętu Administrator ma widok komponentów, które może posiadać komputer.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Extend:

1. Filtrowanie komponentów po rodzaju: procesor, ram, dysk pamięci.

Przebieg podstawowy:

1. Kliknięcie przycisku odpowiedzialnego za widok komponentów.
2. Wyświetlenie widoku na stronie.

FR-3.2 Dodawanie nowego komponentu komputera

Opis: Możliwe jest dodawanie komponentu, na który składa się cena i nazwa.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Przebieg podstawowy:

1. Wykonanie przypadku FR-3.1 Przeglądanie komponentów(procesory, ram, dyski) komputera.
2. Kliknięcie przycisku dodaj procesor, dodaj RAM lub dodaj dysk.
3. System wyświetla formularz komponentu.
4. Wpisanie nazwy i ceny komponentu.
5. Dodanie komponentu do bazy danych.

Przebieg podstawowy:

- 1b Administrator wykonuje dodanie z poziomu formularza dodawania komputera.
- 4 Administrator podał nazwę komponentu która istnieje w bazie danych.
- 5 Następuje modyfikacja podanej ceny istniejącego komponentu zamiast dodania nowego.

FR-3.3 Modyfikacja komponentu komputera

Opis: Możliwe jest modyfikowanie komponentu, na który składa się cena i nazwa.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Include:

1. FR-3.1 Przeglądanie komponentów(procesory, ram, dyski) komputera.

Przebieg podstawowy:

1. Kliknięcie przycisku modyfikuj na odpowiednim wierszu reprezentującym komponent.
2. Modyfikacja komponentu w bazie danych.

FR-3.4 Usuwanie komponentu komputera

Opis: Możliwe jest usuwanie komponentów wchodzącym w skład komputera.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora.

Include:

1. FR-3.1 Przeglądanie komponentów(procesory, ram, dyski) komputera.

Przebieg podstawowy:

1. Kliknięcie przycisku usuń na odpowiednim wierszu reprezentującym komponent.
2. Usunięcie komponentu z bazy, oraz ustawienie wszystkich odwołań w komputerach do danego komponentu na null.

FR-4.1 Zmiana gotowości sprzętu do losowania

Opis: Administrator zarządza urządzeniem i określa jego gotowość do loterii.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora, urządzenie nie zostało wcześniej wylosowane w loterii.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku gotowości do losowania na odpowiednim wierszu reprezentującym urządzenie.
2. Zmiana gotowości losowania na przeciwny do poprzedniego.

FR-4.2 Zmiana statusu sprzętu na sprzedane lub niesprzedane

Opis: Administrator kontroluje, które sprzęty zostały już sprzedane i dostarczone do pracownika

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora, urządzenie zostało wylosowane w loterii.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego

Przebieg podstawowy:

1. Kliknięcie przycisku odpowiedzialnego za zmianę statusu sprzedane na odpowiednim wierszu reprezentującym urządzenie.
2. Aktualizacja statusu sprzedane urządzenia w systemie.

FR-4.3 Wyświetlanie listy uczestników biorącym udział w losowaniu

Opis: Administrator sprawdza jacy użytkownicy biorą udział w losowaniu sprzętu.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora, Status urządzenia został ustawiony na gotowe do losowania.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku odpowiedzialnego za wyświetlanie listy uczestników na odpowiednim wierszu reprezentującym urządzenie.
2. Przekierowanie do widoku użytkowników biorącym udział w losowaniu wybranego sprzętu.
3. Wyświetlanie widoku uczestników losowania, oraz ich statusu dotyczącego czy już wygrali loterię.

FR-4.4 Losowanie zwycięzcy loterii

Opis: Administrator losuje zwycięzcę spośród tych pracowników, którzy zapisali się a loterię.

Aktorzy: Administrator

Warunki początkowe: Administrator został wcześniej zarejestrowany w systemie i posiada uprawnienia administratora, status urządzenia został ustawiony na gotowe do losowania, przynajmniej jeden pracownik jest zapisany na losowanie.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku odpowiedzialnego za losowanie zwycięzcy na odpowiednim wierszu reprezentującym urządzenie.
2. Ustawienie daty losowania na dzisiejszą.
3. Ustawienie wylosowanego pracownika jako zwycięzce losowania.

FR-5.1 Przeglądanie historii loterii wybranego użytkownika

Opis: W systemie istnieje filtrowanie dotyczących historii loterii.

Aktorzy: Administrator, Pracownik

Warunki początkowe: Użytkownik został zalogowany w systemie.

Extend:

1. Filtrowanie loterii po statusie uczestnika: wygrana, przegrana, trwająca.
2. FR-2.5 Wyświetlanie szczegółowych informacji dotyczących sprzętu komputerowego.

Przebieg podstawowy:

1. Kliknięcie przycisku pokaż historie na odpowiednim wierszu reprezentującym użytkownika.
2. Wyświetlenie sprzętu, daty loterii oraz statusu użytkownika w odniesieniu do bieżącej loterii.

Przebieg alternatywny:

- 1b Kliknięcie przycisku pokaż historie użytkownika który jest właśnie zalogowany.

FR-5.2 Zapisywanie się na loterię urządzenia

Opis: Pracownik przeglądając specyfikacje sprzętu może być zainteresowany wzięciem udziału w loterii.

Aktorzy: Pracownik

Warunki początkowe: Pracownik został wcześniej zalogowany w systemie i posiada uprawnienia pracownika, sprzęt nie został wcześniej wylosowany, pracownik wcześniej nie zapisał się na loterię dotyczącego tego sprzętu.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Pracownik klika przycisk weź udział na odpowiednim wierszu reprezentującym urządzenie.
2. Do systemu zostaje dodane uczestnictwo w loterii wybranego urządzenia.

FR-5.3 Wypisanie się z loterii

Opis: Pracownik może chcieć się wypisać z loterii.

Aktorzy: Pracownik

Warunki początkowe: Pracownik został wcześniej zalogowany w systemie i posiada uprawnienia pracownika, sprzęt nie został wcześniej wylosowany, pracownik wcześniej zapisał się na loterię dotyczącego wybranego sprzętu.

Include:

1. FR-2.1 Przeglądanie sprzętu komputerowego.

Przebieg podstawowy:

1. Pracownik klika przycisk wypisz się na odpowiednim wierszu reprezentującym urządzenie.
2. Z systemu zostaje usunięte uczestnictwo w loterii wybranego urządzenia.

3.2. Wymagania нефункциональные

Etapem projektowania systemu jest zdefiniowanie wymagań нефункциональных. Poniżej wypisano listę podstawowych wymagań tego typu.

3.2.1. Ogólnodostępność

- **Kryterium** Ogólnodostępność – system powinien być dostępny dla każdego użytkownika mającego dostęp do internetu. Powinien być zgodny z przeglądarkami internetowymi.
- **Mierzalne kryteria**
 1. czas odpowiedzi dla żądania użytkownika nie przekracza 2 sekund,
 2. kompatybilność z przeglądarkami: Google Chrome, Mozilla Firefox, Microsoft Edge,
 3. responsywność interfejsu użytkownika na różnych urządzeniach (desktop, smartphone).

Wykorzystanie biblioteki React (tabela 2.1) umożliwi spełnieniu kryterium ogólnodostępności.

3.2.2. niezawodność

- **Niezawodność** – system powinien być niepodatny na awarie.
- **Mierzalne kryteria**
 1. średni czas pomiędzy zaobserwowanymi awariami powinien wynosić co najmniej 30 dni,
 2. średni czas naprawy w przypadku awarii nie powinien przekraczać 1 godziny.

Niezawodność jest związana z działaniem aplikacji na środowisku chmurowym.

3.2.3. Bezpieczeństwo

- **Bezpieczeństwo** – system powinien zabezpieczać dane użytkowników i posiadać mechanizmy uwierzytelniania i autoryzacji.
- **Mierzalne kryteria**
 1. wrażliwe dane użytkowników przechowywane są w formie zaszyfrowanej,
 2. mechanizmy sprawdzania bezpieczeństwa haseł użytkowników.

Wykorzystanie logiki napisanej w aplikacji serwerowej w języku JAVA (tabela 2.1) powinno umożliwiać szyfrowanie i sprawdzenie bezpieczeństwa haseł.

3.2.4. Wydajność

- **Wydajność** – system powinien zapewniać satysfakcjonujący czas oczekiwania na wykonanie operacji.
- **Mierzalne kryteria**
 1. większość operacji użytkownika powinno być wykonane poniżej 3 sekund,
 2. skalowalność systemu: możliwość obsłużenia równocześnie zalogowanych 100 użytkowników.

Java, JavaScript umożliwiają szybkie wykonywanie operacji po stronie serwerowej i klientkiej.

3.2.5. Koszty

- **Koszty** – system powinien mieć niskie koszty utrzymania oraz być łatwy w utrzymaniu.
- **Mierzalne kryteria**
 1. miesięczne koszty utrzymania nie powinny przekraczać określonego budżetu,
 2. wprowadzenie zmian przez deweloperów w systemie nie powinno generować dużych kosztów.

Koszty utrzymania są związane z wdrożeniem systemu na środowisko produkcyjne. Dokumentacja systemu umożliwia szybką identyfikację potencjalnych problemów co powinno zredukować koszty.

Rozdział 4

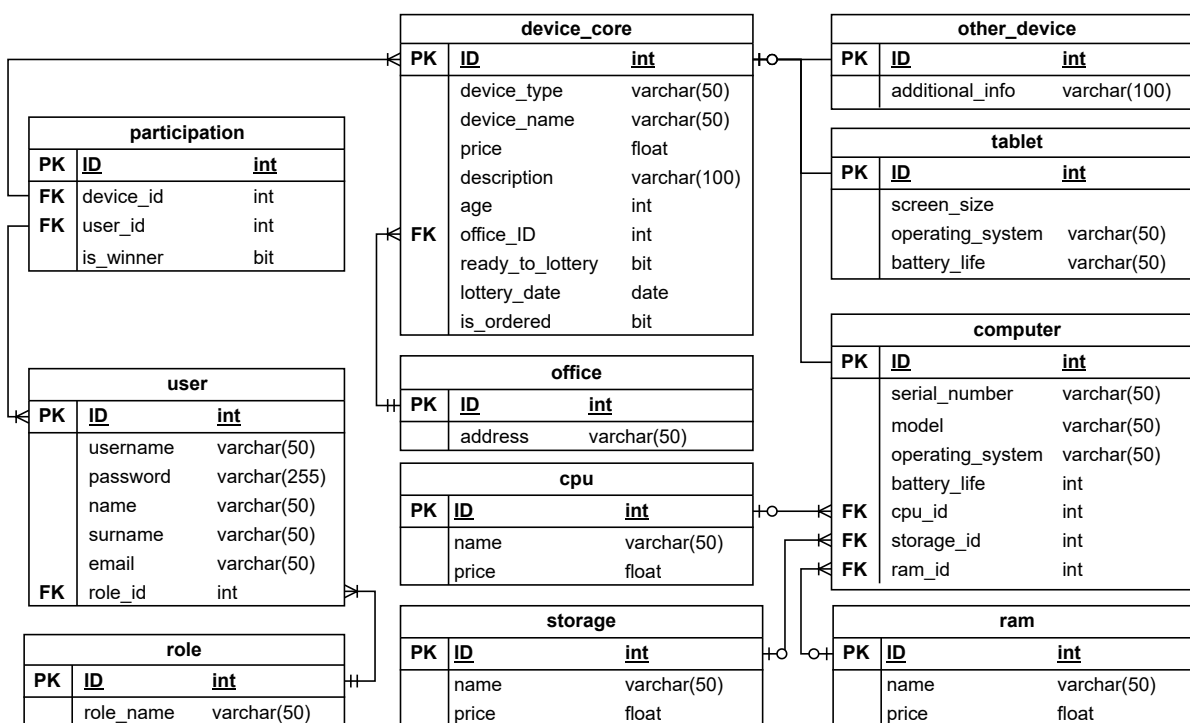
Baza danych

Bazy danych pełnią kluczową rolę w strukturze wielu aplikacji, umożliwiając składowanie, organizację i efektywne zarządzanie danymi. Dzięki nim możliwe jest elastyczne dodawanie nowych informacji oraz rozwijanie funkcji systemu. Struktury baz danych są zoptymalizowane pod kątem efektywnego przeszukiwania i pobierania danych, co przyczynia się do wydajnego działania aplikacji. Dodatkowo, istnieje szereg narzędzi, takich jak **Spring Boot** i **Hibernate** (tabela 2.1), które ułatwiają integrację z bazami danych, usprawniając tym samym proces ich obsługi.

4.1. Model bazy danych

4.1.1. Opisy encji urządzeń

W celu zdefiniowania struktury bazy danych stworzono diagram związków encji ERD (ang. *entity-relationship diagram*) jak na rysunku 4.1. Na diagramie tym zilustrowano, jakie typy danych są przechowywane w poszczególnych tabelach, a także jakie relacje zachodzą między



Rys. 4.1: Diagram związków encji

tymi tabelami. Warto zaznaczyć, że ze względu na brak domyślnego wsparcia dla dziedziczenia tabel w relacyjnych bazach danych, zdecydowano się skorzystać z mechanizmu dziedziczenia dostępnego w Hibernate. Aspekt ten omówiono w dalszej części rozdziału.

W systemie istnieją różne rodzaje urządzeń, z których każdy jest reprezentowany przez odpowiadającą mu tabelę w bazie danych. Trzy główne kategorie urządzeń to: `computer`, `tablet` oraz `other_device`. Wszystkie te kategorie dziedziczą swoje cechy od wspólnej tabeli o nazwie `device_core`, która stanowi rdzeń dla wszystkich urządzeń.

Wspólny rdzeń, reprezentowany przez tabelę `device_core`, zawiera parametry, które są dziedziczone przez wszystkie rodzaje urządzeń. Dzięki tej strukturze, każda tabela reprezentująca konkretny typ urządzenia posiada te same podstawowe atrybuty, ułatwiając jednolite zarządzanie nimi w systemie. Parametry wspólne dla wszystkich typów urządzeń obejmują:

- `ID` – identyfikator urządzenia,
- `device_type` – typ urządzenia ułatwiający zarządzanie nimi w aplikacji klienckiej,
- `device_name` – nazwa urządzenia,
- `price` – cena urządzenia,
- `age` – wiek urządzenia,
- `office_id` – identyfikator biura, w którym znajduje się urządzenie,
- `ready_to_lottery` – informacja czy sprzęt jest gotowy do przeprowadzenia loterii,
- `lottery_date` – data loterii, w której odbyło się losowanie, jeżeli się nie odbyło to wartość jest pusta,
- `is_ordered` – informacja czy sprzęt został już odebrany przez pracownika.

Pozostałe tabele reprezentujące urządzenia rozszerzają informacje dostarczane przez rdzeń. Komputer dodatkowo posiada informacje odnośnie:

- `serial_number` – numer seryjny komputera,
- `model` – model komputera,
- `operatinng_system` – system operacyjny zainstalowany na komputerze,
- `battery_life` – żywotność baterii,
- `cpu_id` – identyfikator reprezentujący procesor posiadany przez komputer,
- `storage_id` – identyfikator reprezentujący dysk posiadany przez komputer,
- `ram_id` – identyfikator reprezentujący pamięć RAM posiadaną przez komputer.

Tablet posiada dodatkowo informacje odnośnie:

- `screen_size` – parametry wyświetlacza,
- `operatinng_system` – system operacyjny zainstalowany na tablecie,
- `battery_life` – żywotność baterii.

Inne urządzenie natomiast posiada tylko pole `additional_info`, które pozwala na podanie szczególnych informacji urządzenia.

4.1.2. Dziedziczenie tabel

Projektowany system powstał z myślą możliwości łatwego rozszerzania. Wykorzystanie tabeli `device_core` ułatwi dodanie ewentualnych tabel potomnych. Kod tabeli nadrzędnej `device_core` umożliwi wykonanie ogólnych operacji, co zapobiegnie powielaniu kodu oraz umożliwi dodanie tabel rozszerzających zakres danych zapisanych już w tabeli `device_core`.

Ważnym aspektem projektu jest zapewnienie unikalności kluczy w kontekście schematu dziedziczenia, szczególnie w przypadku użycia Hibernate z modelem `TABLE_PER_CLASS` opisanym w artykule [4]. W tym podejściu każda klasa dziedzicząca, tak jak `computer`, `tablet`, czy `other_device`, ma swój własny identyfikator, ale także dziedziczy z klasy nadrzędnej. Hibernate programowo zarządza unikalnością kluczy dla całej hierarchii dziedziczenia, eliminując ry-

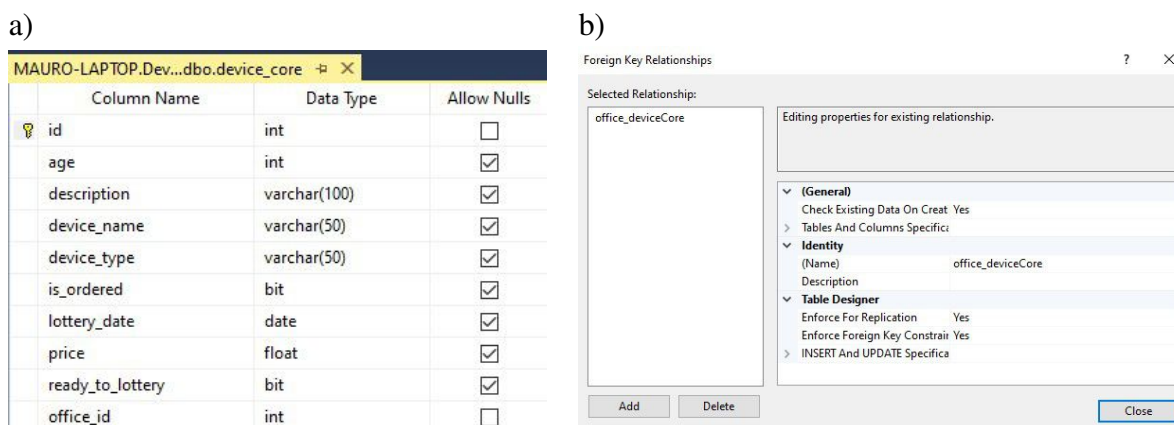
zyko konfliktów kluczy pomiędzy różnymi klasami. Dzieje się to na poziomie aplikacji. W praktyce, w bazie danych dla każdej klasy dziedziczącej, a także dla samej klasy `device_core`, zostaną utworzone oddzielne tabele wykorzystując zapytania SQL generowane przez Hibernate. Tabele te zawierają wszystkie pola związane z daną klasą, a Hibernate zarządza relacjami między nimi. W przypadku klasy `device_core`, tabela zawiera podstawowe atrybuty wspólne dla wszystkich typów urządzeń, takie jak `device_name`, `price`, `age` itp. Przykład implementacji takiego dziedziczenia znajduje się w listingach kodu 5.1 i 5.2.

Dla klas dziedziczących, takich jak `computer`, `tablet` czy `other_device`, Hibernate generuje tabele z dodatkowymi kolumnami specyficznymi dla danej klasy, takimi jak `serial_number` czy `screen_size`. W efekcie każda tabela reprezentuje pełne zestawienie danych związanych z danym rodzajem urządzenia.

4.2. Tworzenie bazy danych

W tworzeniu baz danych istnieją dwa podejścia: *code first* i *database first*. W implementacji systemu wykorzystane zostało podejście *database first*. Daje to w początkowym etapie projektowania większą przejrzystość danych. Po stworzeniu modelu można potem skorzystać z gotowych narzędzi, które pozwolą automatycznie wygenerować kod odzwierciedlający model danych.

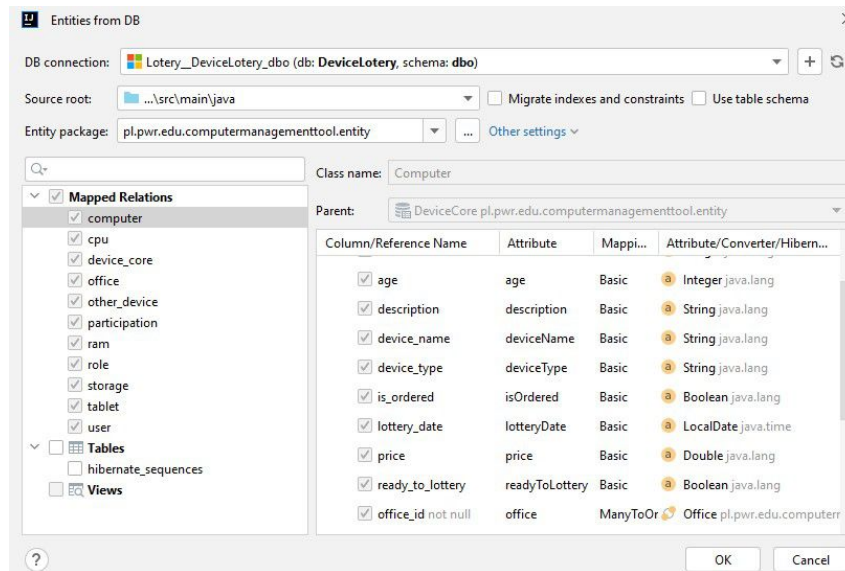
Przy tworzeniu bazy danych wykorzystywanym narzędziem jest SSMS (tabela 2.4.3). Na początek stworzono tabele na podstawie diagramu związków encji 4.1 oraz określono jej relacje. Przykład tworzenia encji oraz jej relacji znajduje się na rysunku 4.2



Rys. 4.2: Tworzenie bazy danych SSMS a) model, b) definiowanie relacji

Po stworzeniu wszystkich tabel możliwe jest wygenerowanie kodu. Wykorzystano do tego plugin JPA Buddy oraz wbudowane narzędzia Intelij. Upraszcza to proces implementacji systemu. Na rysunku 4.3 przedstawiono sposób generowania kodu encji dla klasy `computer`.

Tworzenie w ten sposób początkowej struktury projektu jest bardzo pomocne. Jednak kiedy definiuje się sposób dziedziczenia tabel, to jedynym sposobem jest zdefiniowanie relacji wykorzystując Hibernate. O sposobie dziedziczenia tabel przez Hibernate napisano na początku tego rozdziału 4.1.2. Szczegóły implementacji dotyczące klasy `device_core` opisano w podrozdziale 5.1. Odpowiednia konfiguracja projektu umożliwia aktualizowanie struktury bazy danych. Możliwe dzięki temu jest wyeliminowanie nadmiarowego kodu oraz skorzystanie abstrakcji dostarczanej przez dziedziczenie. Gdy Hibernate zaktualizuje strukturę bazy danych, w SSMS automatycznie powstają odpowiednie tabele i relacje klas sprzętu. Zarządzanie unikalnością kluczy dla tych klas też jest możliwa tylko z poziomu kodu.



Rys. 4.3: Generowaniu kodu na podstawie tabeli computer przy użyciu IntelliJ IDEA

Rozdział 5

Implementacja

5.1. Architektura

5.1.1. Struktura interfejsu REST

Tabele 5.1 oraz 5.2 prezentują jakie zapytania mogą być odbierane przez aplikację serwerową. Z racji tego, że w aplikacji wykorzystywane są typy generyczne w klasach kontrolerach urządzeń i komponentów komputera (implementacja 5.4), uproszczono tabele by pokazać ogólną formę zapytań. Wszystkie zapytania posiadające „devices” w ścieżce umożliwiają alternatywne wywołanie zapytania na podstawie typu sprzętu. Ogólna zasada jest taka aby zastąpić „devices” odpowiadającemu typowi sprzętu: „computers”, „tablets”, „other-devices”. Powoduje to, że zapytanie wykonywane jest przez kontroler tej klasy, a co za tym idzie ma określony typ. Możliwe jest też wykonanie zapytania nie zmieniając ścieżki. Wtedy takie zapytanie wykonywane jest na ogólnym rodzaju sprzętu. Przykładowo „devices/all” pobiera wszystkie sprzęty nie zwracając uwagi na typ, natomiast „computers/all” pobiera tylko sprzętu typu „Computer”. Podobne rozwiązanie wykorzystano w komponentach komputera. Dotyczy to zapytań mających w ścieżce: „component”. Tutaj należy zamienić „component” na jeden z trzech dostępnych typów komponentu: „rams”, „cpus”, „storages”. W tym wypadku nie ma możliwości wywołania zapytania zawierającego „component”.

Tab. 5.1: Tabela prezentująca strukturę REST (cz.1)

Ścieżka	Metoda	Parametry	Opis
/devices/all	GET	-	Pobranie danych sprzętów na podstawie określonego typu
/devices/id	GET	id urządzenia	Pobranie danych wszystkich sprzętów
/devices/id	DELETE	id urządzenia	Usunięcie sprzętu
/computers/update/id	PATCH	id sprzętu oraz zmienione dane sprzętu	Aktualizacja danych komputera
/tablets/update/id	PATCH	id sprzętu oraz zmienione dane sprzętu	Aktualizacja danych tabletu
/other-device/update/id	PATCH	id sprzętu i zmienione dane sprzętu	Aktualizacja danych innego urządzenia
/computers/add	POST	dane komputera	Dodanie nowego komputera
/tablets/add	POST	dane tabletu	Dodanie nowego urządzenia
/other-device/add	POST	dane innego urządzenia	Dodanie nowego urządzenia
/devices/by-office	GET	id biura	Pobranie danych sprzętu znajdujących się w biurze
/devices/all-ready-to-lottery	GET	-	Pobranie danych sprzętów, które są gotowe do losowania
/devices/all-ready-to-lottery-by-officeId/id	GET	id biura	Pobranie danych sprzętów, które są gotowe do losowania i znajdują się w danym biurze
/devices/set-ready-to-lottery/id	PUT	id sprzętu	Ustawienie gotowości sprzętu do losowania
/devices/set-not-ready-to-lottery/id	PUT	id sprzętu	Ustawienie braku gotowości sprzętu do losowania
/devices/set-ordered/id	PUT	id sprzętu	Ustawienie statusu sprzętu na dostarczone do pracownika
/devices/set-not-ordered/id	PUT	id sprzętu	Ustawienie statusu sprzętu na dostarczone do pracownika
/components/all	GET	-	Pobranie danych wszystkich procesorów, ram lub dysków pamięci
/components/id	GET	id komponentu	Pobranie danych procesora, ram lub dysku pamięci
/components/id	DELETE	id komponentu	Usunięcie procesora, ram lub dysku pamięci
/components/update/id	PUT	id sprzętu oraz zmienione dane sprzętu	Aktualizacja danych komputera
/offices/all	GET	-	Pobranie wszystkich biur w systemie
/users/all	GET	-	Pobranie wszystkich użytkowników w systemie

Tab. 5.2: Tabela prezentująca strukturę REST (cz.2)

Ścieżka	Metoda	Parametry	Opis
/users/all-standard	GET	—	Pobranie wszystkich użytkowników z rolą pracownik w systemie
/users/id	GET	—	Pobranie użytkownika
/users/id	DELETE	—	Usunięcie użytkownika
/participation/add	POST	id użytkownika oraz id urządzenia	Dodanie uczestnictwa użytkownika w loterii do wybranego sprzętu
/participation/all	POST	—	Wyświetlenie wszystkich uczestników dla każdego sprzętu
/participation/id	DELETE	—	Usunięcie uczestnictwa o danym id
/participation/delete-by-user_id-and-device_id	DELETE	id użytkownika i id sprzętu	Usunięcie uczestnictwa spełniającego zadane kryteria
/participation/all-wins	GET	—	Wyświetlenie wszystkich zwycięskich uczestników
/participation/user-lottery-history/id	GET	id użytkownika	Wyświetlenie historii uczestnictwa użytkownika o danym id
/participation/user-win-history/id	GET	id użytkownika	Wyświetlenie historii zwycięstw użytkownika o danym id
/participation/user-lose-history/id	GET	id użytkownika	Wyświetlenie historii porażek użytkownika o danym id
/participation/user-pending-lottery/id	GET	id użytkownika	Wyświetlenie uczestnictwa, które czeka na losowanie
/participation/device-participation/id	GET	id urządzenia	Wyświetlenie wszystkich użytkowników zapisanych na losowaniu urządzenia o danym id
/participation/check-if-user-in-lottery	GET	id urządzenia oraz id użytkownika	Sprawdzenie czy użytkownik bierze udział w losowaniu
/participation/select-random-winner/id	GET	id urządzenia	Losowanie zwycięzcy loterii dla urządzenia o danym id

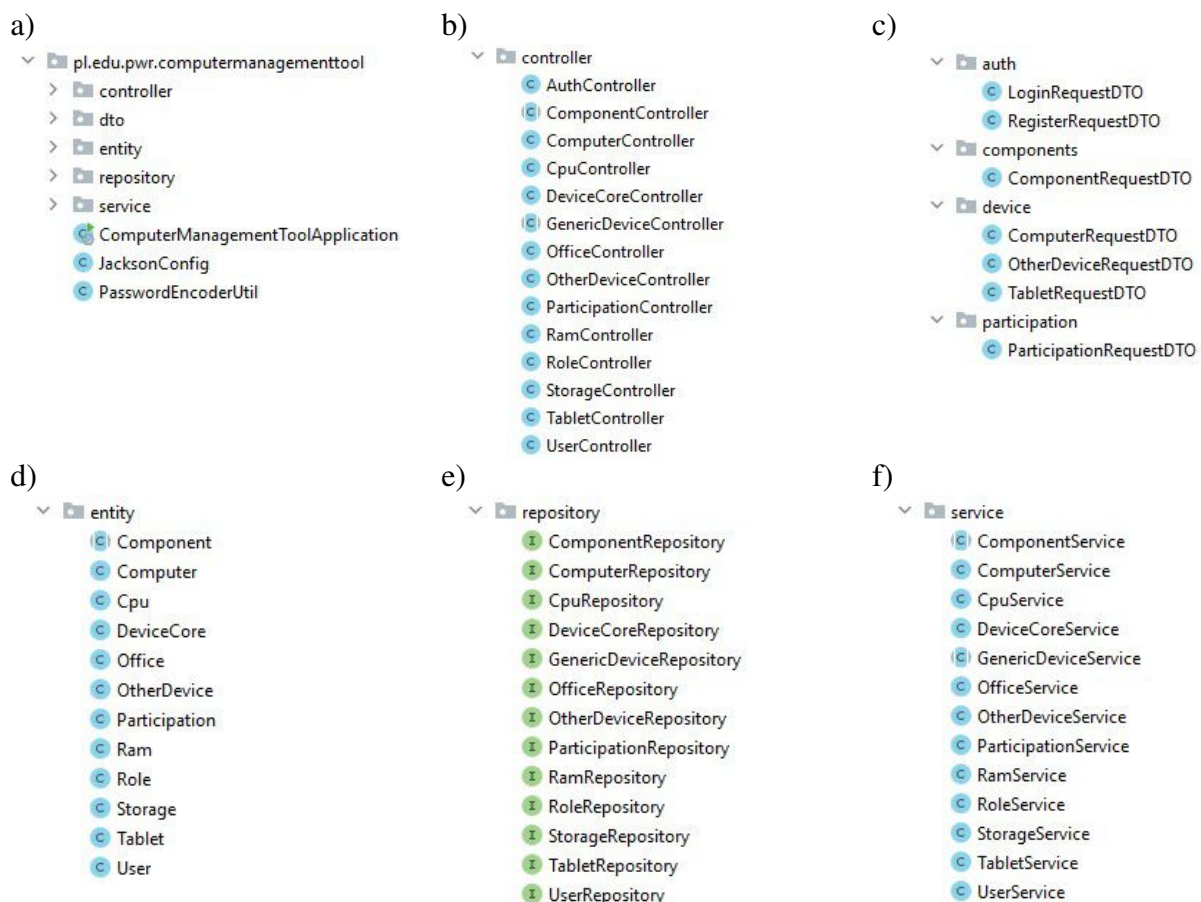
5.2. Aplikacja serwerowa

Do zarządzania danymi wykorzystywana jest aplikacja serwerowa. Przetwarza ona żądania aplikacji klienckiej. Dostarcza ona funkcjonalność uwierzytelniania i autoryzacji. W wyniku zapytań wysłanych przez aplikację kliencką wyświetlane są w aplikacji odpowiednie dane. Możliwe jest też dodawanie i usuwanie odpowiednich rekordów. Implementacja odbywa się z wykorzystaniem języka Java z frameworkami Spring oraz Hibernate 2.1 z wykorzystaniem dobrych praktyk programowania przedstawionych w literaturze [5]. Graficzną reprezentację struktury projektu aplikacji serwerowej pokazano na rysunku 5.1

5.2.1. Struktura projektu

Ścieżka pakietowa wykorzystana jest odwróconą ścieżką domenową Politechniki Wrocławskiej oraz nazwy projektu: `pl.edu.pwr.computermanagementtool`. Następnie pakiety, które są dołączone do tej ścieżki odpowiadają funkcjom które pełnią klasy. Wyróżnia się tutaj pakiety:

- `pl.edu.pwr.computermanagementtool` - pakiet ogólny przechowujący pozostałe pakiety. Klasa `ComputerManagementToolApplication` posiada metodę `Main` odpowiedzialną za uruchomienie aplikacji. Klasa `JacksonConfig` jest konfiguracją odpowiedzialną za serializację formatu JSON. Natomiast klasa `PasswordEncoderUtil` jest odpowiedzialna za szyfrowanie haseł użytkowników korzystających z systemu,
- `controller` – pakiet przechowujący w sobie klasy kontrolerów odpowiedzialne za wykonywanie zapytań aplikacji za pomocą interfejsu REST, implementacje kontrolerów odbywały się z pomocą materiałów [1, 8].
- `dto` – pakiet który jest odpowiedzialny za przechowywanie klas służących do przechowywania danych i ułatwiania ich przesyłania między różnymi częściami systemu. Wykorzystywany jest w klasach kontrolera,
- `entity` – pakiet przechowujący definicje encji reprezentowanych w bazie danych,
- `repository` – pakiet przechowujący repozytoria służące do komunikacji z bazą danych,
- `service` – pakiet przechowujący serwisy które są odpowiedzialne za warstwę logiki biznesowej w systemie.



Rys. 5.1: Struktura projektu: a) ogólna struktura, b) kontrolery, c) dto, d) encje, e) repozytoria, f) serwisy

5.2.2. Fragmenty implementacji

Fragment kodu klas encji wykorzystującej schemat dziedziczenia

Na listingu 5.1 kodu klasy `DeviceCore` w linii 3 określono schemat dziedziczenia `TABLE_PER_CLASS`. O sposobie dziedziczenia napisano w podrozdziale 4.1.2. Linia 6 określa strategię generowania klucza. Hibernate tworzy wtedy specjalną tabelę w bazie danych, która odpowiedzialna jest za przechowywanie unikalnych kluczy głównych dla encji sprzętu w aplikacji. W kodzie pokazanym na listingu 5.2 nie trzeba wtedy definiować klucza.

Listing 5.1: Klasa nadrzędna reprezentująca rdzeń sprzętu: `DeviceCore`

```

1 @Entity
2 @Table(name = "device_core")
3 @Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
4 public class DeviceCore {
5     @Id
6     @GeneratedValue(strategy = GenerationType.TABLE)
7     @Column(name = "id", nullable = false)
8     private Integer id;
9
10    @Column(name = "device_type", length = 50)
11    private String deviceType;
12
13    @Column(name = "device_name", length = 50)
14    private String deviceName;
15
16    // Pozostała część klasy

```

Na listingu 5.2 w linii 3 określone zostało dziedziczenie po klasie nadrzędnej reprezentującej rdzeń sprzętu. Pola które należy określić w tej klasie są unikalnymi polami tabeli `computer`. W linii 5 istnieje statyczna zmienna `DEVICE_TYPE` która pomaga określić jakiego rodzaju dany sprzęt. Wykorzystywane jest to w aplikacji klienckiej.

Listing 5.2: Klasa potomna: `Computer`, reprezentująca komputer

```

1 @Entity
2 @Table(name = "computer")
3 public class Computer extends DeviceCore{
4
5     public static final String DEVICE_TYPE = "COMPUTER";
6     @Column(name = "serial_number", length = 50)
7     private String serialNumber;
8
9     @Column(name = "operating_system", length = 50)
10    private String operatingSystem;
11
12    @Column(name = "battery_life", length = 50)
13    private String batteryLife;
14
15    // Pozostała część klasy

```

Przykład kodu repozytorium dla urządzeń dziedziczącym po rdzeniu sprzętu

Repozytorium sprzętu komputerowego napisano z użyciem wyrażeń generycznych (patrz listing 5.3). Takie podejście przyczynia się do uproszczenia kodu oraz łatwiejszej jego rozbudowy. Definicja klas generycznych powstała przy pomocy artykułu [2]. Adnotacja `@NoRepositoryBean` z linii 1 jest używana do oznaczenia interfejsów, które nie mają mieć swojej instancji. Oznacza to, że nie jest on przeznaczony do utworzenia instancji repozytorium

w trakcie uruchamiania aplikacji. W linii 4 i 5 poprzez zdefiniowanie metody możliwe jest wykonanie konkretnego zapytania SQL do bazy danych. Z racji, że klasy sprzętu komputerowego mają zbliżone funkcje, nie trzeba dostarczać tych samych interfejsów we wszystkich repozytoriach, tylko zastosować schemat dziedziczenia.

Listing 5.3: Generyczne repozytorium sprzętu komputerowego: GenericDeviceRepository

```

1 @NoRepositoryBean
2 public interface GenericDeviceRepository<T extends DeviceCore> extends
    ↳ JpaRepository<T, Integer> {
3
4     List<T> findAllByReadyToLotteryIsTrue();
5     List<T> findAllByOfficeId(int officeId);}

```

Fragmenty implementacji klasy kontrolerów sprzętu

Kontrolery pełnią istotną funkcję w systemie. Umożliwiają one definiowanie zapytań REST które będą wysyłane do serwera. Wykorzystując abstrakcje oraz wyrażenia generyczne możliwe jest wykonanie niektórych zapytań bez konieczności ich implementacji w klasach potomnych. Zaimplementowane kontrolery bazują na klasie abstrakcyjnej pokazanej na listingu 5.4.

Listing 5.4: Klasa nadrzędna kontrolera sprzętu: GenericDeviceController

```

1 public abstract class GenericDeviceController<T extends DeviceCore> {
2
3     protected final GenericDeviceService<T> genericDeviceService;
4     protected final GenericDeviceRepository<T> genericRepository;
5
6     protected GenericDeviceController(GenericDeviceService<T> genericDeviceService,
    ↳ GenericDeviceRepository<T> genericRepository) {
7         this.genericDeviceService = genericDeviceService;
8         this.genericRepository = genericRepository;
9     }
10    @GetMapping("/{id}")
11    @CrossOrigin(origins = "*")
12    T getOneBasicDevice(@PathVariable int id){
13        return genericDeviceService.getDeviceById(id);
14    }
15    @GetMapping("/all")
16    @CrossOrigin(origins = "*")
17    List<T> getAllBasicDevices(){
18        return genericDeviceService.getAllDevices();
19    }
20    @GetMapping("/all-ready-to-lottery")
21    @CrossOrigin(origins = "*")
22    List<T> getAllReadyToLotteryDevice(){
23        return genericDeviceService.getAllReadyToLotteryDevices();
24    }
25    // Pozostała część klasy
26
27    @PutMapping("/set-ready-to-lottery/{id}")
28    @CrossOrigin(origins = "*")
29    public ResponseEntity<T> setReadyToLottery(@PathVariable int id){
30        try{
31            T updatedBasicDevice = genericDeviceService.setReadyToLottery(id);
32            return new ResponseEntity<>(updatedBasicDevice, HttpStatus.OK);
33        } catch (RuntimeException e){
34            return new ResponseEntity<>(HttpStatus.NOT_FOUND);}
35
36    // Pozostała część klasy

```


Klasa pokazana na listingu 5.5 nie posiada własnych metod. Umożliwia jednak wykonywanie ogólnych operacji dotyczących sprzętu. Możliwe jest między innymi pobranie informacji dotyczące sprzętu bez względu na jego typ co idealnie obrazuje test 6.2.

Listing 5.5: Klasa potomna reprezentująca ogólną postać sprzętu

```

1 @RestController
2 @RequestMapping("/devices")
3 public class DeviceCoreController extends GenericDeviceController<DeviceCore>{
4
5     protected DeviceCoreController(DeviceCoreService deviceCoreService,
6         ↪ GenericDeviceRepository<DeviceCore> genericRepository) {
7         super(deviceCoreService, genericRepository);
8     }
9 }

```

Fragmenty implementacji klas serwisów sprzętu

Serwisy są odpowiedzialne za warstwę logiki biznesowej systemu. Implementację ich klasy bazowej pokazano na listingu 5.6.

Listing 5.6: Klasa nadrzędna serwisu sprzętu GenericDeviceService

```

1
2 public abstract class GenericDeviceService<T extends DeviceCore> {
3
4     protected final GenericDeviceRepository<T> genericDeviceRepository;
5     protected final OfficeRepository officeRepository;
6
7     public GenericDeviceService(GenericDeviceRepository<T> genericRepository,
8         ↪ OfficeRepository officeRepository) {
9         this.genericDeviceRepository = genericRepository;
10        this.officeRepository = officeRepository;
11    }
12    public T getDeviceById(int id) {
13        Optional<T> basicDeviceOptional = genericDeviceRepository.findById(id);
14        return basicDeviceOptional.orElseThrow(() -> new RuntimeException("Device not
15            ↪ found with id: " + id));
16    }
17    public List<T> getAllDevices() {
18        return genericDeviceRepository.findAll();
19    }
20    // Pozostała część klasy
21
22    protected DeviceCore addDevice(Class<? extends DeviceCore> deviceClass, String
23        ↪ deviceName, Double price, String description, Integer age, Boolean
24        ↪ readyToSell, Integer officeId) {
25
26        if(officeId == null){
27            throw new RuntimeException("Office required");
28        }
29        Optional<Office> officeOptional = officeRepository.findById(officeId);
30        Office office = officeOptional.orElseThrow(() -> new RuntimeException("
31            ↪ Office not found with id: " + officeId));
32
33        DeviceCore deviceCore;
34        try {
35            deviceCore = deviceClass.getDeclaredConstructor().newInstance();
36        } catch (InstantiationException | IllegalAccessException |
37            ↪ NoSuchMethodException | InvocationTargetException e) {
38            throw new RuntimeException("Error creating device", e);
39        }
40    }
41 }

```

```

33     }
34
35     deviceCore.setDeviceName(deviceName);
36     deviceCore.setPrice(price);
37     deviceCore.setDescription(description);
38     deviceCore.setAge(age);
39     deviceCore.setReadyToSell(readyToSell);
40     deviceCore.setOffice(office);
41
42     return deviceCore;
43 }
44 // Pozostała część klasy

```

W linii 12 widnieje metoda, która pozwala na pobranie sprzętu. W linii 17 natomiast jest możliwe pobranie wszystkich sprzętów. Odbywa się to przy pomocy sparametryzowanych typów danych T. Metoda zdefiniowana w linii 22 pozwala uprościć dodawanie sprzętu, implementując część logiki ustawiając parametry charakterystyczne dla rdzenia sprzętu. Jednak aby utworzyć odpowiedni typ niezbędne jest określenie jakiego typu będzie dodawane urządzenie. Dlatego w linii 22 metoda jako parametr przyjmuje `Class<? extends DeviceCore>`, który oznacza typ sprzętu. Tak zdefiniowana metoda może zostać użyta do dodania sprzętu w klasach potomnych co pokazano w listingu 5.7.

Listing 5.7: Klasa potomna serwisu tabletu: `TabletService`

```

1  @Service
2  public class TabletService extends GenericDeviceService<Tablet>{
3
4      public TabletService(TabletRepository tabletRepository, OfficeRepository
          ↪ officeRepository){
5          super(tabletRepository, officeRepository);
6      }
7      public Tablet addTablet(String deviceName, Double price, String description,
8          Integer age, String officeAddress, String screenSize,
9          String operatingSystem, String batteryLife){
10
11          Tablet tablet = (Tablet) addDevice(Tablet.class, deviceName, price,
          ↪ description, age, officeAddress);
12
13          tablet.setScreenSize(screenSize);
14          tablet.setOperatingSystem(operatingSystem);
15          tablet.setDeviceType(Tablet.DEVICE_TYPE);
16          tablet.setBatteryLife(batteryLife);
17
18          return genericDeviceRepository.save(tablet);
19      }
20      // Pozostała część klasy

```

Serwis klasy `TabletService` dziedziczy i parametryzuje metody klasy `GenericDeviceService`. Wykorzystuje on w linii 13 metodę klasy nadrzędnej co uprasza implementację metody dodającej sprzęt (linia 9).

5.3. Aplikacja kliencka

Aplikacja kliencka jest aplikacją internetową, dostarczającą przyjazny interfejs użytkownika. Napisano ją w języku JavaScript, z wykorzystaniem biblioteki React (tabela 2.1). Istnieją zdefiniowane zapytania REST do aplikacji serwerowej. Wykorzystując te zapytania na widoku widocznym dla użytkownika zostają podjęte specjalne akcje, które ten widok aktualizują. Formu-

larze pokazywane są w takiej formie, że po kliknięciu odpowiedniego przycisku ukazuje się formularz na danym widoku. Zestawienie widoków, ścieżek i roli pokazano w tabeli 5.3.

5.3.1. Struktura projektu

Ogólna struktura projektu powstała na skutek wywołania komendy tworzącej aplikację Reacta: `npx create-react-app frontend`. Następnie utworzono folder `components`, który w sobie zawiera foldery przechowujące pliki JavaScript na podstawie pełnionych funkcji w projekcie. Foldery i odpowiadające im funkcje są następujące:

- **auth** przechowuje pliki odpowiedzialne za obsługę i stylizację widoków logowania i rejestracji:
 - `Login.js` – skrypt odpowiedzialny za widok i funkcjonalność logowania,
 - `Register.js` – skrypt odpowiedzialny za widok i funkcjonalność rejestracji,
 - `Login.css` – arkusz styli odpowiedzialny za stylizację widoków logowania i rejestracji.
- **bar** przechowuje skrypty odpowiedzialne za paski nawigacyjne poszczególnych widoków:
 - `AdminHomeBar.js` – skrypt odpowiedzialny za pasek nawigacyjny na widoku domowym administratora: `Home.js`,
 - `ComputerComponentsBar.js` – skrypt odpowiedzialny za pasek nawigacyjny dla widoku komponentów: `ComputerComponents.js`,
 - `ManageUserBar.js` – skrypt odpowiedzialny za pasek nawigacyjny widoku zarządzania użytkownikami: `Users.js`,
 - `UserHomeBar.js` – skrypt odpowiedzialny za pasek nawigacyjny na widoku domowym pracownika: `Home.js`,
 - `UserLotteryHistory` – skrypt odpowiedzialny za pasek nawigacyjny na widoku historii loterii użytkownika: `UserLotteryHistory.js`,
 - `UsersInLottery.js` – skrypt odpowiedzialny za pasek nawigacyjny na widoku użytkowników biorących udział w loterii: `UsersInLottery.js`.
- **computer_components** przechowuje skrypty formularzy komponentów komputera, które mogą dodawać lub modyfikować te komponenty:
 - `CpuForm.js` – formularz procesora,
 - `RamForm.js` – formularz pamięci RAM,
 - `StorageForm.js` – formularz pamięci dyskowej.
- **device_form** przechowuje skrypty oraz arkusz styli odpowiedzialne za logikę formularzy dodawania, modyfikacji urządzeń:
 - `AddNewFormPopup.js` – formularz umożliwiający wybór formularza który dodaje komputer, tablet, lub inne urządzenie,
 - `ComputerForm.js` – formularz komputera,
 - `DeviceForm.css` – arkusz styli odpowiedzialny za stylizację formularzy,
 - `FormPopup.js` – skrypt odpowiedzialny za logikę i wybór rodzaju formularza (dodawanie, modyfikacja, informacja) urządzenia,
 - `OtherDeviceForm` – formularz innego urządzenia,
 - `TabletForm.js` – formularz tabletu.
- **view** przechowuje w sobie skrypty oraz arkusz styli odpowiedzialne za widoki aplikacji:
 - `ComputerComponets.js` – widok dostępny tylko dla administratora. Zawiera w sobie informacje o komponentach komputera w postaci tabeli,
 - `Home.css` – arkusz styli odpowiedzialny za stylizację widoków,
 - `Home.js` – widok strony domowej, ma w sobie dwa warianty: Administratora i Pracownika,

- `UserLotteryHistory.js` – widok historii loterii użytkownika o zadanym ID. Dostępny dla pracownika i administratora,
- `Users.js` – widok pracowników zarejestrowanych w systemie. Dostępny tylko dla administratora,
- `UsersInLottery` – widok pracowników biorących udział w loterii danego sprzętu. Dostępny tylko dla administratora.

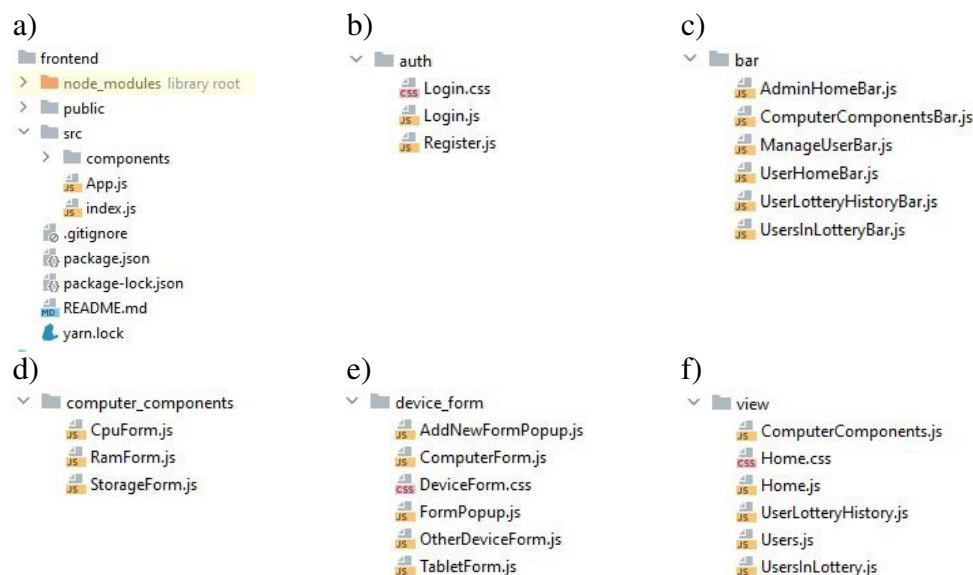
Równolegle do folderu `components` w folderze `src` istnieją jeszcze dwa skrypty, które są niezbędne do funkcjonowania aplikacji.

- `App.js` – zawiera w sobie ścieżki url pod którymi dostępne są poszczególne widoki,
- `index.js` – definiuje funkcję główną programu.

Strukturę projektu pokazano na rysunku 5.2.

Tab. 5.3: Zestawienie ścieżek url dostępnych w aplikacji klienckiej

Ścieżka	Widok	Skrypt	Rola
<code>/auth/login</code>	Logowanie	<code>Login.js</code>	Przed nadaniem roli
<code>/auth/register</code>	Rejestracja	<code>Register.js</code>	Przed nadaniem roli
<code>/home</code>	Strona domowa z tabelą sprzętów	<code>Home.js</code>	Administrator lub Pracownik
<code>/users</code>	Tabela pracowników	<code>Users.js</code>	Administrator
<code>/components</code>	Tabela komponentów komputera	<code>ComputerComponents.js</code>	Administrator
<code>/users-in-lottery</code>	Tabela użytkowników biorących udział w loterii	<code>UsersInLottery.js</code>	Administrator lub Pracownik
<code>/users-lottery-history</code>	Tabela zawierająca historię loterii	<code>UsersLotteryHistory.js</code>	Administrator lub Pracownik



Rys. 5.2: Struktura projektu: a) ogólna struktura, b) Logowanie i rejestracja, c) pasek nawigacyjny, d) formularze komponentów, e) formularze sprzętów, f) widoki stron

5.3.2. Fragmenty implementacji

Ścieżki url dla widoków aplikacji

Na listingu kodu 5.8 pokazano definicje ścieżek url, pod którymi dostępne są poszczególne widoki wymienione w tabeli 5.3.

Listing 5.8: Zdefiniowane ścieżki url widoków systemu

```

1  const App = () => {
2    return (
3      <Router>
4        <Routes>
5          <Route path="/auth/login" element={<Login />} />
6          <Route path="/auth/register" element={<Register />} />
7          <Route path="/home" element={<Home />} />
8          <Route path="/" element={<Navigate to="/auth/login" />} />
9          <Route path="/auth/*" element={<Navigate to="/auth/login" />} />
10         <Route path="/users" element={<Users/>}/>
11         <Route path="/components" element={<ComputerComponents/>}/>
12         <Route path="/users-in-lottery" element={<UsersInLottery/>}/>
13         <Route path="/user-lottery-history" element={<UserLotteryHistory/>}/>
14       </Routes>
15     </Router>

```

Sparytmetryzowany formularz urządzenia

Kod skryptu `FormPopup.js` przedstawiony na listingu 5.9 służy do wyboru formularza na podstawie typu sprzętu oraz operacji wykonywanej przez formularz. Przyjmuje 4 argumenty. Przekazywanie tych parametrów odbywa się w linii 4 z wykorzystaniem „props”. Zmienna `deviceType` odpowiada dla jakiego urządzenia dotyczy formularz. Może on dotyczyć komputera, tabletu lub innego urządzenia co pokazano w liniach 9-11. Zmienna `setTrigger` jest odpowiedzialna za to czy formularz jest aktualnie wyświetlany na widoku. Zmienna `formType` dotyczy operacji reprezentowanej przez formularz. Istnieją 3 operacje: dodawania nowego sprzętu, modyfikacji sprzętu oraz wyświetlania informacji o sprzęcie. Tak stworzona funkcja umożliwia zarządzanie formularzami, które zostały zdefiniowane w skryptach `ComputerForm.js`, `TabletForm.js` oraz `OtherDeviceForm.js`.

Listing 5.9: Obsługa formularzy

```

1  function FormPopup(props){
2
3    const {setTrigger, formType, deviceType, deviceId} = props;
4
5    return (props.trigger) ? (
6      <div className="popup">
7        <div>
8          {deviceType === 'COMPUTER' && <ComputerForm setTrigger={props.setTrigger}
9            ↪ formType={formType} deviceId={deviceId}/>}
10         {deviceType === 'TABLET' && <TabletForm setTrigger={props.setTrigger}
11           ↪ formType={formType} deviceId={deviceId}/>}
12         {deviceType === 'OTHER' && <OtherDeviceForm setTrigger={props.setTrigger}
13           ↪ formType={formType} deviceId={deviceId} />}
14         {props.children}
15       </div>
16     </div>
17   ) : "";

```

Na listingu 5.10 ukazano, w jaki sposób aplikacje kliencka komunikuje się z serwerem. Po podaniu odpowiednich danych w formularzu i kliknięciu przycisku następuje wysłanie żądania typu POST do serwera które dodaje nowy komputer.

Listing 5.10: Przykładowe zapytanie do serwera dla formularza komputera

```

1  const handleAddComputer = async (e) =>{
2    try{
3      const compData ={
4        "deviceName" : formData.deviceName,
5        "price" : formData.price,
6        "description" : formData.description,
7        "age" : formData.age,
8        "officeAddress" : formData.office,
9        "serialNumber" : formData.serialNumber,
10       "model" : formData.model,
11       "operatingSystem" : formData.model,
12       "batteryLife" : formData.batteryLife,
13       "cpuName" : formData.cpu,
14       "storageName" : formData.storage,
15       "ramName" : formData.ram};
16     const response = await axios.post('http://localhost:8080/computers/add',
      ↪ compData, {});
17   }catch (error){console.error('Błąd dodawania komputera', error)}}

```

5.3.3. Instrukcja użytkownika

Zdefiniowano widoki aplikacji i pogrupowano je ze względu na typ które pełnią.

- W-1 – widok logowania i rejestracji (rysunek 5.3),
- W-2 – widok strony domowej zawierający urządzenia oraz loterię tych urządzeń (rysunek 5.4),
- W-3 – widok formularzy dodawania urządzeń, z wariantami na podstawie typu urządzenia (rysunek 5.5),
- W-4 – widok formularzy modyfikacji urządzeń (rysunek 5.6),
- W-5 – widok formularzy informacji urządzeń (rysunek 5.7),
- W-6 – widok użytkowników i uczestników (rysunek 5.8),
- W-7 – widok historii loterii użytkownika (rysunek 5.9),
- W-8 – widok komponentów komputera (rysunek 5.10),
- W-9 – widok formularzy komponentów komputera (rysunek 5.11).

a)

Zaloguj się

Nazwa użytkownika

Hasło

Zaloguj

[Nie masz konta? Zarejestruj się](#)

b)

Zarejestruj się

Nazwa użytkownika

Email address

Imię

Nazwisko

Hasło

Powtórz Hasło

Zarejestruj

[Masz konto? Zaloguj się](#)

Rys. 5.3: Widok W-1, Logowanie i Rejestracja: a) logowanie, b) rejestracja

Początkowym widokiem aplikacji jest widok z widok W-1 z formularzem logowania i rejestracji. Każdy użytkownik musi się zalogować podając poprawne dane logowania (rysunek 5.3a). W przypadku braku konta możliwa jest rejestracja, klikając na odnośnik „Nie masz konta? Zarejestruj się”. Po kliknięciu odnośnika następuje przekierowanie na formularz rejestracji (rysunek 5.3b). Po pomyślnym zarejestrowaniu następuje przekierowanie do formularza logowania. Możliwe jest też kliknięcie odnośnika „Masz już konto? Zaloguj się” aby przejść do formularza logowania.

a)

Zarządzanie sprzętem komputerowym												
Wszystkie urządzenia Wszystkie biura Dodaj urządzenie Przeglądaj komponenty Zarządzaj użytkownikami Wyloguj												
ID	Nazwa	Cena	Opis	Wiek	Biuro	Czy sprzedany	Gotowy na losowanie	Usuń	Modifikuj	Informacje	Lista uczestników	Loteria
554	Dell 7470	500 zł	lekko używany	1	Grunwaldzka	Nie Zmien	Po losowaniu	Usuń	Modifikuj	Informacje	Wyświetl	2024-01-21
555	Lenovo 500E	299 zł	brak baterii	2	Politechnika	Nie Zmien	Tak Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Losuj
556	ASUS X515JA	300 zł		1	Grunwaldzka	Nie Zmien	Tak Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Losuj
557	LENOVO ThinkPad	300 zł	brak RAM	2	Grunwaldzka	Nie Zmien	Nie Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Oczekuje na zatwierdzenie
561	Myszka logitech	20 zł	Myszka USB	2	Grunwaldzka	Nie Zmien	Po losowaniu	Usuń	Modifikuj	Informacje	Wyświetl	2024-01-21
562	Myszka logitech	20 zł	Myszka USB	2	Grunwaldzka	Nie Zmien	Nie Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Oczekuje na zatwierdzenie
563	Myszka logitech	20 zł	Myszka bezprzewodowa	2	Grunwaldzka	Nie Zmien	Tak Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Losuj
564	Myszka logitech	20 zł	Myszka bezprzewodowa	2	Grunwaldzka	Nie Zmien	Nie Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Oczekuje na zatwierdzenie
559	TABLET BLOW PlatinumTAB8	420 zł	lekko porysowany	2	Grunwaldzka	Nie Zmien	Nie Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Oczekuje na zatwierdzenie
560	TABLET Lenovo Tab M10	120 zł	brak ładowarki	2	Politechnika	Nie Zmien	Nie Zmien	Usuń	Modifikuj	Informacje	Wyświetl	Oczekuje na zatwierdzenie

b)

Zarządzanie sprzętem komputerowym

Komputery

Biuro Grunwaldzka

Dodaj urządzenie

Przeglądaj komponenty

Zarządzaj użytkownikami

Wyloguj

ID	Nazwa	Cena	Opis	Wiek	Biuro	Numer seryjny	System	Bateria	Model	Procesor	Pamięć	Ram	Czy sprzedany	Gotowy na losowanie	Usuń	Modyfikuj	Informacje	Lista uczestników	Loteria	
554	Dell 7470	500 zł	lekko używany	1	Grunwaldzka	E123AS	Latitude 7480	10h	Latitude 7480	Intel Core i7-6600U	512 GB SSD	16 GB	Nie Zmien	Tak Zmien	<button>Usuń</button>	<button>Modyfikuj</button>	<button>Informacje</button>	<button>Wyświetl</button>	2024-01-21	
556	ASUS X515JA	300 zł		1	Grunwaldzka	AS1298	X515JA-BQ3326	12	X515JA-BQ3326	Intel Core i3-1005G1	256 GB SSD	4 GB	Nie Zmien	Tak Zmien	<button>Usuń</button>	<button>Modyfikuj</button>	<button>Informacje</button>	<button>Wyświetl</button>	<button>Losuj</button>	
557	LENOVO ThinkPad	300 zł	brak RAM	2	Grunwaldzka	FG123	ThinkPad T460s	12	ThinkPad T460s	Intel Celeron Quad Core	512 GB SSD	Brak	Nie Zmien	Nie Zmien	<button>Usuń</button>	<button>Modyfikuj</button>	<button>Informacje</button>	<button>Wyświetl</button>	Oczekuje na zatwierdzenie	

c)

Zarządzanie sprzętem komputerowym

Wszystkie urządzenia

Wszystkie biura

Przeglądaj swoje loterie

Wyloguj

ID	Nazwa	Cena	Opis	Wiek	Biuro	Czy sprzedany	Informacje	Loteria
554	Dell 7470	500 zł	lekko używany	1	Grunwaldzka	Nie	Informacja	2024-01-21
555	Lenovo 500E	299 zł	brak baterii	2	Politechnika	Nie	Informacja	Wypisz się
556	ASUS X515JA	300 zł		1	Grunwaldzka	Nie	Informacja	Weź udział w loterii
561	Myszka logitech	20 zł	Myszka USB	2	Grunwaldzka	Nie	Informacja	2024-01-21
563	Myszka logitech	20 zł	Myszka bezprzewodowa	2	Grunwaldzka	Nie	Informacja	Weź udział w loterii

Rys. 5.4: Widok W-2, Strona domowa: a) Administrator bez filtrowania, b) Administrator z filtrowaniem, c) Pracownik bez filtrowania

Na rysunku 5.4 pokazano wariantowy widok W-2 z tabelą urządzeń z loteriami. Dla każdego wariantu istnieje przycisk wyloguj odpowiedzialny za wylogowywanie i przekierowanie do widoku W-1a (rysunek 5.3). Na widoku W-2 możliwe jest filtrowanie urządzeń po typie urządzenia oraz po biurze, w którym się znajdują. Dla wariantu W-2b pokazane zostały tylko komputery, które znajdują się w biurze Grunwaldzka.

Wyświetlanie szczegółowych informacji o urządzeniu odbywa się na widoku formularzy W-5 pokazanym na rysunku 5.7. Wybierany jest odpowiedni formularz W-5a, W-5b, W-5c na podstawie typu urządzenia.

Widok W-2 jest widokiem administratora, oferującym wiele funkcji:

- Kliknięcie przycisku „Dodaj urządzenie” sprawia, że na ekranie wyświetlany jest formularz W-3. Początkowo wyświetlany jest wariant W-3a ale istnieje możliwość na zmianę typu formularza na wariant W-3b lub W-3c poprzez odpowiedni wybór z listy rozwijanej,
- kliknięcie przycisku „Przeglądaj komponenty” sprawia, że następuje przekierowanie do widoku komponentów W-8,

- kliknięcie przycisku „Zarządzaj użytkownikami” przekierowuje do widoku zarejestrowanych pracowników W-6a (rysunek 5.8),
- kliknięcie przycisku „Zmień” w kolumnie czy sprzedany ustawia status sprzętu na sprzedany lub nie. Jest to informacja czy sprzęt został dostarczony do pracownika,
- kliknięcie przycisku w kolumnie „Gotowy na losowanie” możliwe jest tylko wtedy gdy losowanie się nie odbyło. Sprawia ono, że operacja „Losuj” w kolumnie loteria jest możliwa. Sprzęty które mają status gotowe do losowania są wyświetlane na widoku pracownika W-2c,
- kliknięcie przycisku „Usuń” usuwa z systemu odpowiedni sprzęt,
- kliknięcie przycisku „Modyfikuj” wyświetla widok formularza modyfikacji W-4 (rysunek 5.6). Wyświetlany jest jeden z formularzy W-4a, W-4b, W-4c na podstawie typu modyfikowanego urządzenia,
- kliknięcie przycisku „Wyświetl” w kolumnie „Lista uczestników” przekierowuje do widoku pracowników biorących udział w loterii wybranego sprzętu W-6b (rysunek 5.8),
- kliknięcie przycisku „Losuj” w kolumnie loteria możliwe jest w przypadku gdy urządzenie jest gotowe do losowania oraz loteria się jeszcze nie odbyła. Następnie po kliknięciu przycisku ustawiana data losowania jest na dzisiejszą oraz losowany jest zwycięzca losowania. W przypadku braku uczestników losowanie nie odbędzie się,

Widok W-2c (rysunek 5.4) jest widokiem pracownika. Nie posiada on tak zaawansowanych funkcjonalności jak administrator,

- kliknięcie przycisku w kolumnie loteria umożliwia zapisywanie lub wypisywanie z uczestnictwa w losowaniu. Zapis na loterie możliwy jest wtedy kiedy losowanie się jeszcze nie odbyło,
- kliknięcie przycisku „Przeglądaj swoje loterie” przekierowuje do widoku W-7 (rysunek 5.9). Widok ten posiada historię loterii dla aktualnie zalogowanego użytkownika.

The image shows three variations of a web form for adding a device, labeled a), b), and c). Each form has a title 'Dodaj' followed by a dropdown menu indicating the device type.

- a) Dodaj Komputer:** This form includes fields for 'Nazwa', 'Cena', 'Opis', 'Wiek [Lata]', and 'Biuro' (with a dropdown menu). It also has a 'Numer seryjny' field. On the right side, there are sections for 'System operacyjny', 'Żywotność baterii', 'Model', 'Procesor' (with a dropdown menu and a 'Dodaj nowy procesor' button), 'Dysk pamięci' (with a dropdown menu and a 'Dodaj nowy dysk' button), and 'Ram' (with a dropdown menu and a 'Dodaj nowy ram' button). At the bottom, there are buttons for 'Dodaj komputer' and 'Zamknij'.
- b) Dodaj Tablet:** This form includes fields for 'Nazwa', 'Cena', 'Opis', 'Wiek [Lata]', and 'Biuro' (with a dropdown menu). It also has a 'Dodaj Tablet' button at the bottom left and a 'Zamknij' button at the bottom right.
- c) Dodaj Inne:** This form includes fields for 'Nazwa', 'Cena', 'Opis', and 'Wiek [Lata]'. It also has a 'Biuro' dropdown menu, a 'Dodatkowe informacje' field, and buttons for 'Dodaj urządzenie' and 'Zamknij' at the bottom.

Rys. 5.5: Widok W-3 dodawania urządzenia a) wariant komputera; b) wariant tabletu, c) wariant innego urządzenia

Formularze W-3 ukazane na rysunku 5.5 są dostępne tylko dla administratora. Wyświetlane są jako wyskakujące okno. Początkowym wariantem jest wariant W-3a służący za dodawanie komputera. Wariant można zmienić na dodawanie tabletu W-3b lub dodawanie innego urządzenia W-3c po przez wybór opcji z listy rozwijanej na formularzu. Na wariancie W-3a istnieje możliwość dodawania procesora, pamięci RAM i pamięci dyskowej po przez kliknięcie przycisku. Każdy przycisk powoduje otworzenie odpowiedniego formularza z widoku W-9. Waż-

nym aspektem takiego rozwiązania jest to, że jeżeli użytkownik nie znajdzie odpowiedniego procesora, RAM lub dysku to może szybko je dodać. Po takim dodaniu następuje powrót do formularza, formularz ten zachowuje poprzedni stan, a lista rozwijana dotycząca dodawanego komponentu od razu zostaje uzupełniona o dodatkowe dane. Podobną funkcjonalność ma widok W-4a modyfikacji komputera 5.6.

a)

Modyfikuj komputer

Nazwa

Cena

Opis

Wiek [Lata]

Biuro

Numer seryjny

System operacyjny

Żywotność baterii

Model

Procesor

Dysk pamięci

Ram

Modyfikuj komputer Zamknij

b)

Modyfikuj tablet

Nazwa

Cena

Opis

Wiek [Lata]

Biuro

Rozmiar ekranu

Żywotność baterii

System operacyjny

Modyfikuj Tablet Zamknij

c)

Modyfikuj urządzenie

Nazwa

Cena

Opis

Wiek [Lata]

Biuro

Dodatkowe informacje

Modyfikuj urządzenie Zamknij

Rys. 5.6: Widok W-4 formularzy modyfikacji: a) komputera; b) tabletu, c) innego urządzenia

Formularze na widoku W-4 (rysunek 5.6) są bardzo podobne do tych na widoku W-3 przedstawionych na rysunku 5.5. Zasadniczą różnicą jest jednak to, że przy modyfikacji urządzenia od razu określony jest jego typ, przez co nie ma możliwości zmiany na formularza na inny typ, tak jak to było możliwe przez listę rozwijaną w widoku W-3. Drugim aspektem jest to, że formularze są już wypełnione o odpowiednie dane, które zawiera modyfikowane urządzenie. Wymazanie takich danych sprawi to, że odpowiednie pola będą miały wartość null. W formularzu W-4a możliwe jest też dodanie odpowiedniego komponentu na podstawie jednego z trzech przycisków, co działa w ten sam sposób jak formularzu W-3a (rysunek 5.5). Formularze te również dostępne są tylko z poziomu administratora.

a)

Informacje o komputerze

Nazwa

Cena

Opis

Wiek [Lata]

Biuro

Numer seryjny

System operacyjny

Żywotność baterii

Model

Procesor

Dysk pamięci

Ram

Zamknij

b)

Informacje o tablecie

Nazwa

Cena

Opis

Wiek [Lata]

Biuro

Rozmiar ekranu

Żywotność baterii

System operacyjny

Zamknij

c)

Informacje o urządzeniu

Nazwa

Cena

Opis

Wiek [Lata]

Biuro

Dodatkowe informacje

Zamknij

Rys. 5.7: Widok W-5 formularzy informacji: a) o komputerze, b) o tablecie, c) o innym sprzęcie

Formularze na widoku W-5 (rysunek 5.7) podobne są do tych z widoków W-3 (rysunek 5.5) i W-4 (rysunek 5.6). Jednak w przeciwieństwie do nich nie dodają ani nie modyfikują danych. Z formularzy tych może korzystać również pracownik. Dane na takim formularzu są od razu uzupełnione, a możliwość ich edycji jest zablokowana.

a)

Panel administracyjny, zarządzanie użytkownikami						
Powrót do urządzeń Wyloguj						
ID	Nazwa użytkownika	Imię	Nazwisko	Email	Historia loterii	Usuń
10	mniewczas	Maurycy	Niewczas	maurycy.niewczas@gmail.com	Pokaż historię	Usuń użytkownika
11	jkowalski	Jan	Kowalski	jan.kowalski@gmail.com	Pokaż historię	Usuń użytkownika
12	lewy	Robert	Lewandowski	lewandowski@gmail.com	Pokaż historię	Usuń użytkownika
13	Mieszko	Mieszko	Pierwszy	mieszko@gmail.com	Pokaż historię	Usuń użytkownika
14	Zbigniew	Zbigniew	Zbyszowski	zbigniew@gmail.com	Pokaż historię	Usuń użytkownika
15	Aleksandra	Aleksandra	Olkowska	olusia@gmail.com	Pokaż historię	Usuń użytkownika

b)

Lista uczestników						
Powrót do urządzeń Wyloguj						
ID	User ID	Nazwa użytkownika	Imię	Nazwisko	Email	Czy wygrał
15	10	mniewczas	Maurycy	Niewczas	maurycy.niewczas@gmail.com	Nie
18	11	jkowalski	Jan	Kowalski	jan.kowalski@gmail.com	Nie
23	13	Mieszko	Mieszko	Pierwszy	mieszko@gmail.com	Tak

Rys. 5.8: Widok W-6, Informacje o użytkownikach: a) Zarządzanie użytkownikami przez administratora, b) Przeglądanie uczestników losowania

Widok W-6 (rysunek 5.8) posiada informacje o użytkownikach. W-6a jest widokiem dostępnym tylko dla administratora

- kliknięcie przycisku [powrót do urządzeń](#) przekierowuje do strony domowej W-2 (rysunek 5.4),
- kliknięcie przycisku „Pokaż historię” w kolumnie „Historia loterii” przekierowuje do widoku historii loterii W-7 (rysunek 5.9) dla odpowiedniego pracownika,
- kliknięcie przycisku „Usuń użytkownika” usuwa z systemu użytkownika.

Widok W-6b też jest widokiem tylko dostępnym dla administratora. Posiada on listę uczestników oraz informacje o wygranej dla każdego poszczególnego uczestnika wybranej loterii.

Historia loterii użytkownika								
Wszystkie Loterie użytkownika Powrót Wyloguj								
Wszystkie Loterie użytkownika								
Id urządzenia	Nazwa	Cena	Opis	Więcej	Wygrane Trwające	urządzeniu	Wypisz się	Status uczestnika
554	Dell 7470	500 zł	lekko używany	1	Grunwaldzka	Informacje	Po loterii	Przegrany
555	Lenovo 500E	299 zł	brak baterii	2	Politechnika	Informacje	Wypisz się	Oczekuje na losowanie
561	Myszka logitech	20 zł	Myszka USB	2	Grunwaldzka	Informacje	Po loterii	Przegrany

Rys. 5.9: Widok W-7, Historia loterii pracownika

Widok W-7 pozwala wyświetlić historię loterii użytkownika (rysunek 5.9). Pozwala on na pracownikowi przeglądanie własnej historii. Administrator może skorzystać z tego widoku aby sprawdzić historię loterii wybranego pracownika.

- możliwe jest tutaj filtrowanie loterii na podstawie statusu uczestnika. Są dostępne 3 opcje: „Wszystkie loterie użytkownika”, „Wygrana” i „Trwająca”,
- kliknięcie przycisku „Informacje” wyświetla formularz informacji z rysunku 5.7 na podstawie typu sprzętu.
- kliknięcie przycisku „Wypisz się” możliwe jest wtedy kiedy loteria się jeszcze nie odbyła. Po kliknięciu tego przycisku pracownik rezygnuje z uczestnictwa w losowaniu wybranego sprzętu.

Komponenty sprzętu komputerowego				
<div> Procesory Urządzenia Dodaj procesor Dodaj RAM Dodaj dysk Zarządzaj użytkownikami Wyloguj </div>				
ID	Nazwa		Cena	Usuń
8	Intel Core i7-6600U		0 zł	<button>Usuń</button>
9	Intel Celeron Quad COre		40 zł	<button>Usuń</button>
10	Intel Core i3-1005G1		0 zł	<button>Usuń</button>

Rys. 5.10: Widok W-8, Widok komponentów komputera

Widok W-8 jest dostępny dla administratora. Pozwala mu na oszacowanie ceny sprzętu patrząc jakie komponenty mają cenę. Możliwe jest wybranie z listy rozwijanej jakiego rodzaju komponenty mają być wyświetlane. Do wyboru są: procesory, pamięci RAM oraz pamięci dyskowe.

- kliknięcie przycisku „Dodaj procesor” wyświetli formularz W-9a (rysunek 5.11a) odpowiedzialny za dodanie procesora,
- kliknięcie przycisku „Dodaj RAM” Wyświetli formularz W-9b (rysunek 5.11b) odpowiedzialny za dodanie pamięci RAM,
- kliknięcie przycisku „Dodaj dysk” Wyświetli formularz W-9c (rysunek 5.11c) odpowiedzialny za dodawanie dysku pamięci,
- kliknięcie przycisku „Zarządzaj użytkownikami” przekieruje do widoku W-6a (rysunek 5.8),
- kliknięcie przycisku usuń spowoduje usunięcie komponentu. Komputer posiadający usuwany komponent ustawia identyfikator odpowiedniego komponentu na wartość „null”.

W przypadku dodaniu komponentu, którego nazwa już istnieje następuje modyfikacja ceny.

a)

Dodaj Procesor

Nazwa

Cena

Zamknij

Dodaj procesor

b)

Dodaj pamięć RAM

Nazwa

Cena

Zamknij

Dodaj ram

c)

Dodaj dysk

Nazwa

Cena

Zamknij

Dodaj dysk

Rys. 5.11: Widok W-9, Formularze komponentów: a) procesor, b) ram, c) dysk pamięci

Widok W-9 (rysunek 5.11) odpowiada za prezentację formularzy komponentów. Formularze te mogą zostać wyświetlone po kliknięciu odpowiedniego przycisku na widoku W-8 (rysunek 5.10) lub z przycisku dostępnego na formularzach dodawania i modyfikacji komputera (rysunek 5.5a i 5.6a)

Rozdział 6

Testy aplikacji

Do przetestowania aplikacji wykorzystano **Postmana** i **Lighthouse 2.1**. Postman pozwolił sprawdzić czy zapytania oraz logika działają prawidłowo. Lighthouse pozwolił przeanalizować jakość strony internetowej.

6.1. Testowanie restowych zapytań przy pomocy Postmana

Wykorzystując Postmana można na bieżąco monitorować stan aplikacji poprzez wysyłanie odpowiednich zapytań oraz obserwację odpowiedzi. Przetestowano każde z dostępnych zapytań zdefiniowane w tabeli 5.1 i 5.2. Testowanie odbywało się na bieżąco z rozwojem systemu. Umożliwiło to szybką identyfikację problemów.

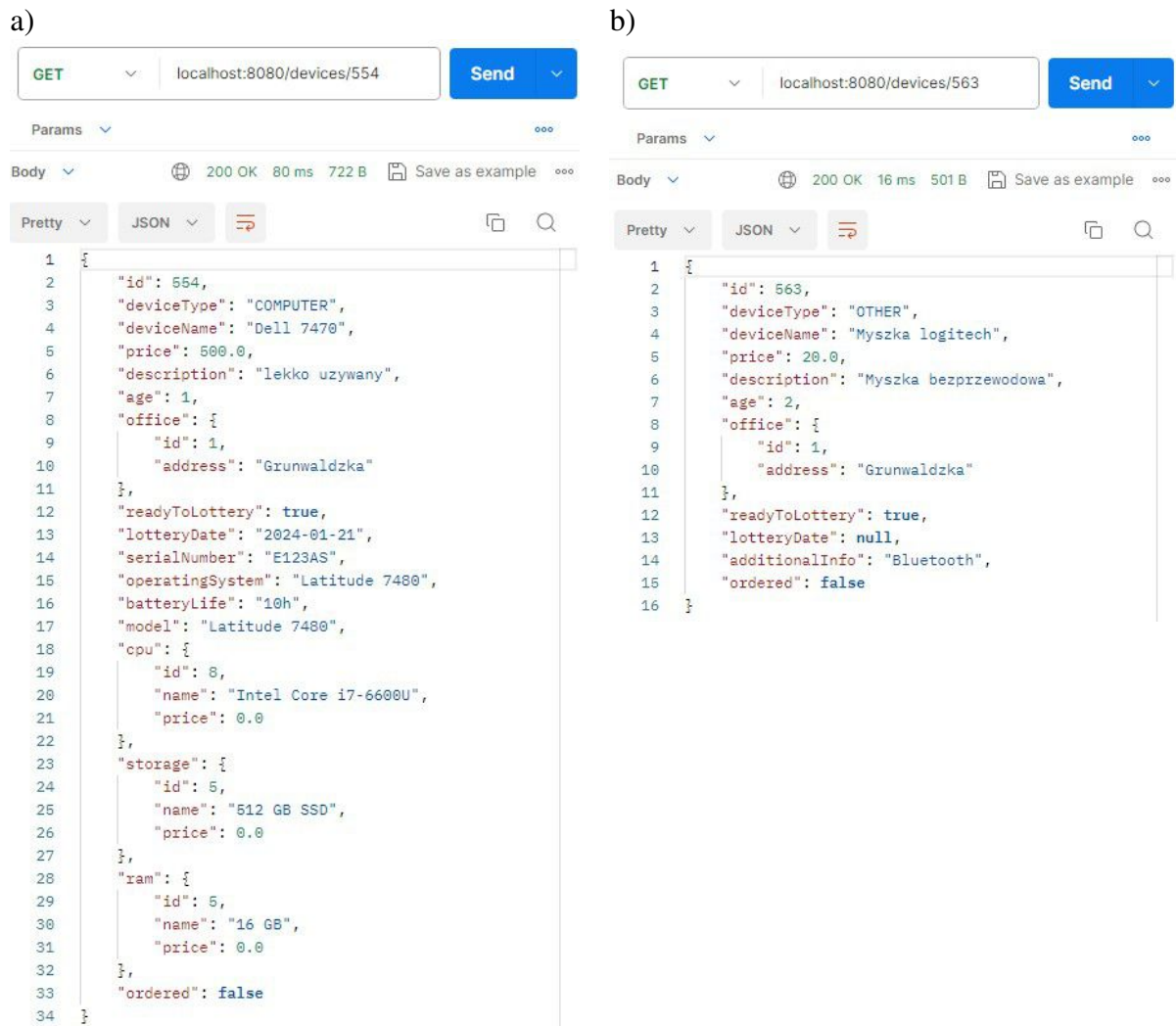


Rys. 6.1: Struktura testów a) po id, b) wszystkie urządzenia, c) sortowanie po biurze, d) dodawanie, e) aktualizacja, f) usuwanie, g) związane z loterią, h) związane z użytkownikiem

6.1.1. Wybrane scenariusze testowe

Pobieranie informacji o sprzęcie

Na scenariuszu testowym 1 (rysunek 6.2) przetestowano pobieranie informacji o urządzeniu. Wykorzystując zapytanie `localhost:8080/devices/id` metodą GET uzyskano na scenariuszu testowym 1a i 1b status 200 OK. Wyświetlane informacje świadczą o poprawności działania pobierania informacji o sprzęcie. W scenariuszu 1a zostały wyświetlone informacje o komputerze a w 1b o innym urządzeniu. Odpowiedź zapytania jest zależna od tego jakie urządzenie o zadanym ID ma typ. Możliwe jest coś takiego po wykorzystaniu schematu dziedziczenia, które dostarcza unikalnych kluczy dla poszczególnych typów sprzętów. Więcej na ten temat napisano w rozdziale 4.1.2.



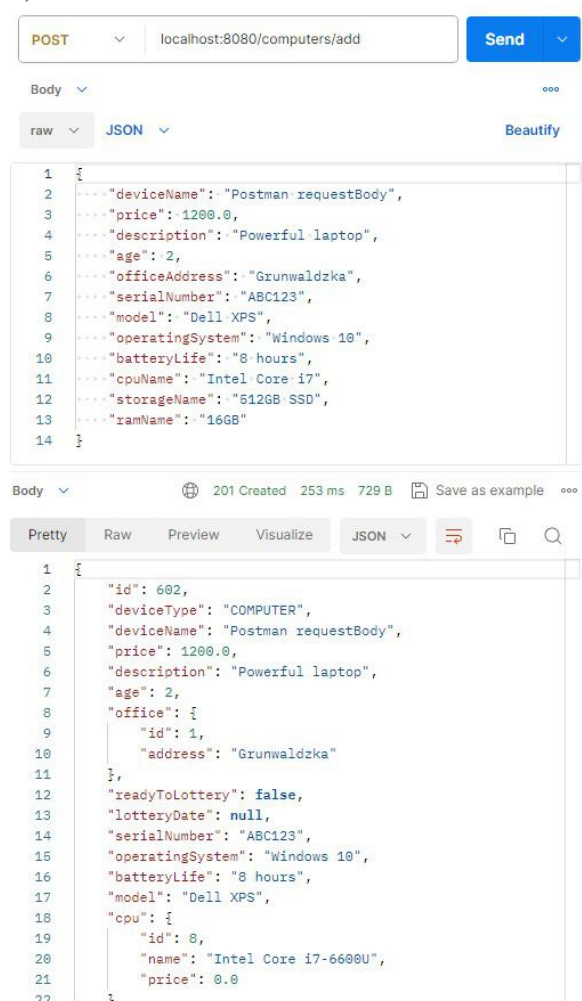
Rys. 6.2: Scenariusz testowy 1, Pobieranie informacji o urządzeniu: a) komputer, b) inne urządzenie

Dodawanie i modyfikowanie urządzenia

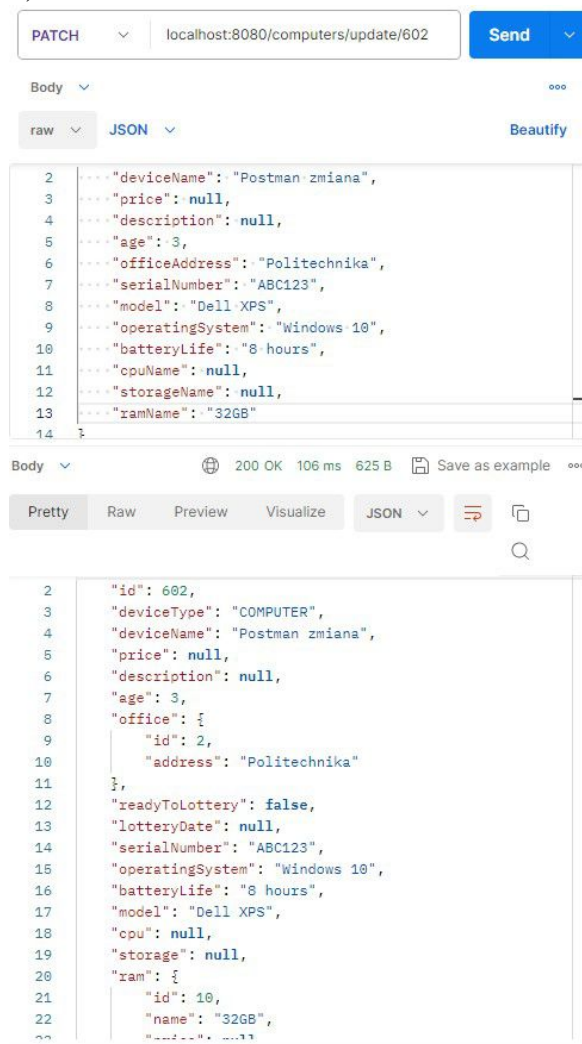
Na scenariuszu testowym 2 (rysunek 6.3) najpierw dodano a potem zmodyfikowano dodane urządzenie. W obu przypadkach dane zostały wysłane jako JSON. W 2a po wysłaniu żądania metodą POST uzyskano status HTTP 201 Created. Oznacza to że urządzenie zostało poprawnie dodane do systemu. W dolnej części 2a widać jakie urządzenie zostało stworzone i jakie ma parametry. Scenariusz 2b modyfikuje utworzone w 2a urządzenie. Potrzebuje do tego oprócz danych zmieniających sprzęt ID urządzenia. ID jest przekazywane przez ścieżkę url. Możliwa

jest modyfikacja pól w ten sposób by były miały wartość null. Skorzystano tutaj z metody PATCH która modyfikuje istniejący element.

a)



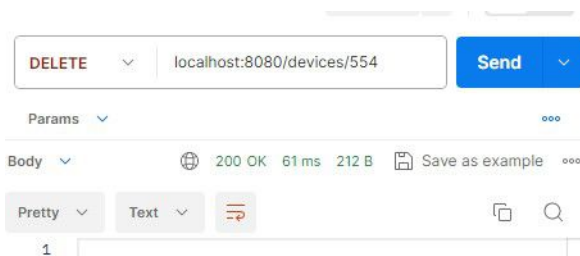
b)



Rys. 6.3: Scenariusz testowy 2, Dodawanie i modyfikowanie komputera: a) dodawanie, b) modyfikacja

Usuwanie urządzenia

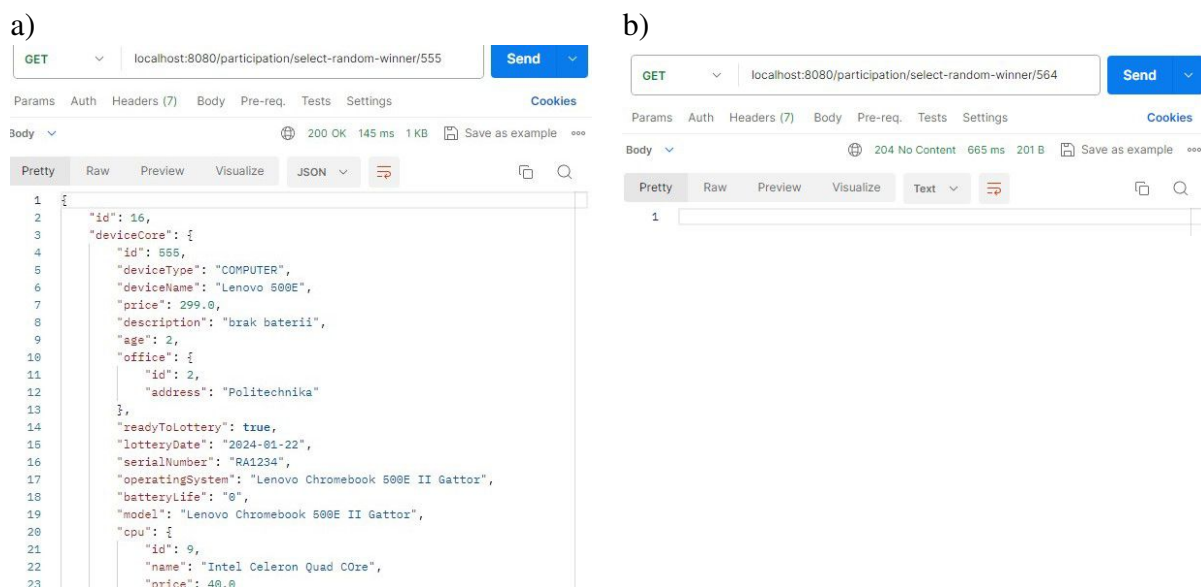
W scenariuszu testowym 3 (rysunek 6.4) pokazano w jaki sposób jest usuwane urządzenie. Do usunięcia urządzenia potrzebne jest ID, które przekazywane jest przez ścieżkę url. Po skorzystaniu z metody DELETE uzyskiwany jest status http 200 OK, który oznacza, że urządzenie zostało pomyślnie usunięte.



Rys. 6.4: Scenariusz testowy 3, Usuwanie urządzenia po ID

Losowanie zwycięzcy

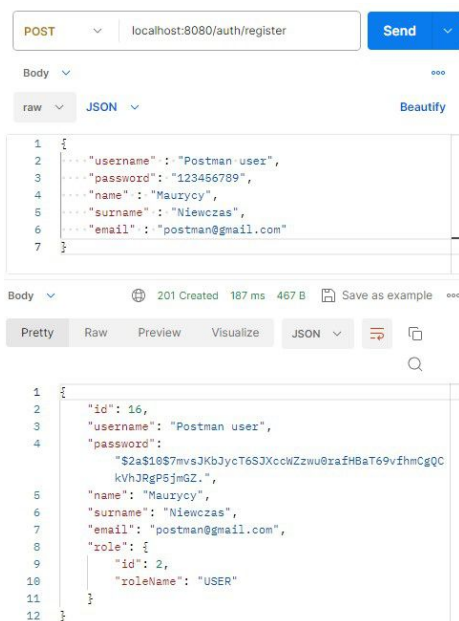
Jedną z głównych funkcjonalności aplikacji jest przeprowadzenie losowania. Na scenariuszu testowy 3 (rysunek 6.5) pokazano w jaki sposób jest przeprowadzane losowanie. W przypadku kiedy loteria posiada zapisanych użytkowników, losowany jest zwycięzca, i zwracany jest status HTTP 200 OK. W scenariuszu 3b pokazano przypadek kiedy na loterię nie został zapisany żaden pracownik lub loteria nie została otworzona. Wtedy zwracany status HTTP to 204 No Content.



Rys. 6.5: Scenariusz testowy 3, Losowanie zwycięzcy: a) powodzenie, b) niepowodzenie

Rejestracja użytkownika

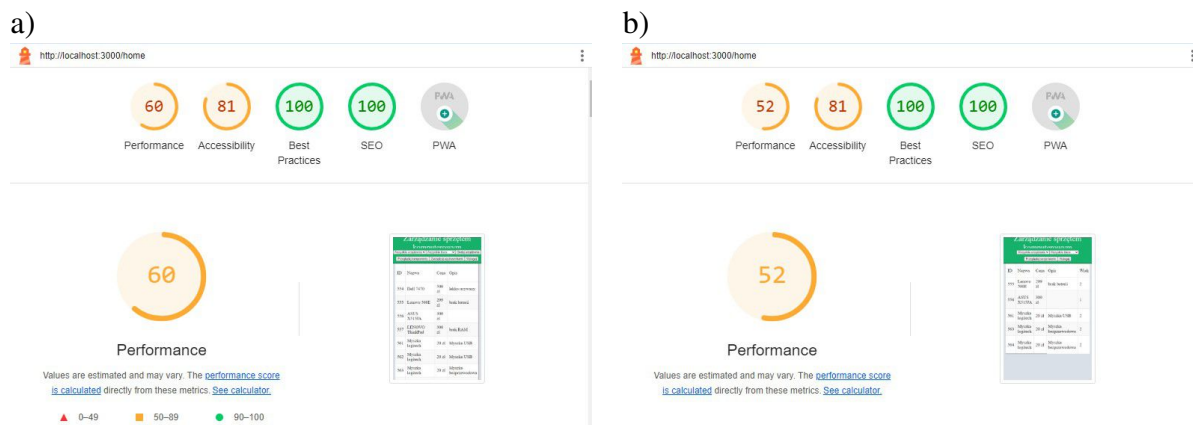
Na scenariuszu testowym 4 (rysunek 6.6) pokazano jak działa rejestracja użytkownika. Po utworzeniu takiego użytkownika zwracany jest status HTTP 201 Created. Użytkownik taki zapisywane hasło ma w postaci zaszyfrowanej.



Rys. 6.6: Scenariusz testowy 4, Rejestracja użytkownika

6.2. Testowanie jakości strony internetowej systemu przy użyciu Lighthouse

Wykorzystanie Lighthouse 2.1 pozwala na zbadanie wydajności, dostępności i dobrych praktyk strony internetowej. Testowany widok był strony domowej, ponieważ na niej dzieje się większość logiki systemu. Testowanie odbyło się w dwóch wariantach: strony domowej administratora i strony domowej pracownika 6.7



Rys. 6.7: Test Lighthouse strony domowej, a) administratora, b) pracownika

Wnioskując z testu widok aplikacji ma satysfakcjonujące parametry.

- **Wydajność (ang. *Performance*)** – jest na średnim poziomie. Pokazuje to że aplikacja mogła by być bardziej zoptymalizowana pod względem wydajności,
- **Dostępność (ang. *Accessibility*)** – jest na wysokim poziomie. Oznacza to, że osoby z różnymi niepełnosprawnościami nie miałyby większego problemu z posługiwaniem się aplikacją. Jednak tutaj jest możliwość optymalizacji by osiągnąć większy stopień zadowolenia tych użytkowników,
- **Dobre praktyki (ang. *Best Practices*)** – pokazuje, że test przeszedł test dobrych praktyk i strona jest zgodna ze standardami,
- **SEO** – pokazuje, że strona jest zoptymalizowana pod kątem wyszukiwarek internetowych.

Literatura

- [1] Building REST services with Spring. <https://spring.io/guides/tutorials/rest/> [odczyt dnia 23 stycznia 2024].
- [2] Generic controllers and services in Spring Boot Java. <https://ultimate.systems/web/blog/generic-controllers-and-services-in-spring-boot-java> [odczyt dnia 23 stycznia 2024].
- [3] Hibernate inheritance mapping. <https://www.baeldung.com/hibernate-inheritance> [odczyt dnia 23 stycznia 2024].
- [4] Spring framework documentation. <https://docs.spring.io/spring-framework/reference/index.html> [odczyt dnia 23 stycznia 2024].
- [5] J. Bloch. *Java. Efektywne programowanie*. Helion, wydanie 3, 2018.
- [6] G. G. Christian Bauer, Gavin King. *Java Persistence with Hibernate*. Manning, wydanie 2, 2015.
- [7] M. Ghanayem, R. West, J. Roth, D. Coulter, M. Bradley, K. Toliver, C. Guyer, M. Thomas. SQL Server Management Studio components and configuration, 2023. <https://learn.microsoft.com/en-us/sql/ssms/tutorials/ssms-configuration?view=sql-server-ver16> [odczyt dnia 23 stycznia 2024].
- [8] P. Kumar. Restful Web Services Tutorial in Java, Sier. 2022. <https://www.digitalocean.com/community/tutorials/restful-web-services-tutorial-java> [odczyt dnia 23 stycznia 2024].
- [9] S. Stefanov. *React: Up & Running*. O'Reilly, wydanie 2, 2021.
- [10] C. Walls. *Spring w Akcji*. Helion, wydanie 5, 2019.

Dodatek A

Instrukcja wdrożeniowa

A.1. Wymagane narzędzia

Aplikacja w formie minimalnego środowiska uruchomieniowego stworzona w pracy dyplomowej dostępna jest na repozytorium GitHub pod poniższym linkiem:

<https://github.com/Maurycjo/Praca-dyplomowa/tree/release>

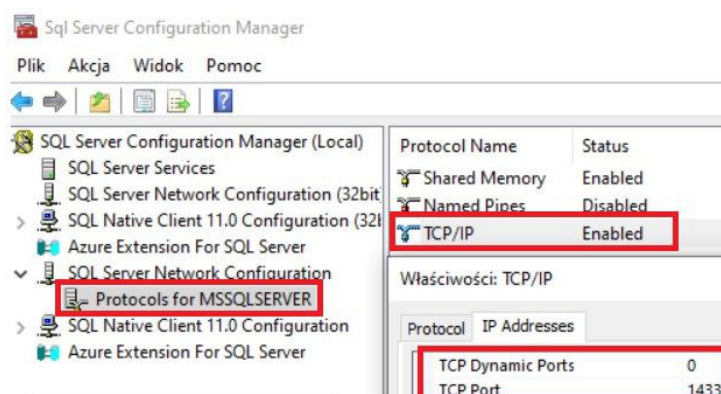
Do uruchomienia aplikacji niezbędne jest posiadanie następujących narzędzi (patrz tabela 2.1):

- Java 17
- Node v18.17.0
- SQL Server Management Studio
- SQL Server

SSMS i SQL Server dostępne są na stronie Microsoftu.

A.2. Konfigurowanie środowiska

Wykorzystując narzędzie SQL Server Configuration Manager (instalowane z SQL Server) należy odblokować **TCP/IP** dla protokołu **MSSQLSERVER**. W właściwościach protokołu **TCP/IP** należy ustawić: TCP Dynamics Ports: **0** i pozostawić domyślny TCP Port: **1433**. Pokazano to na rysunku A.1.

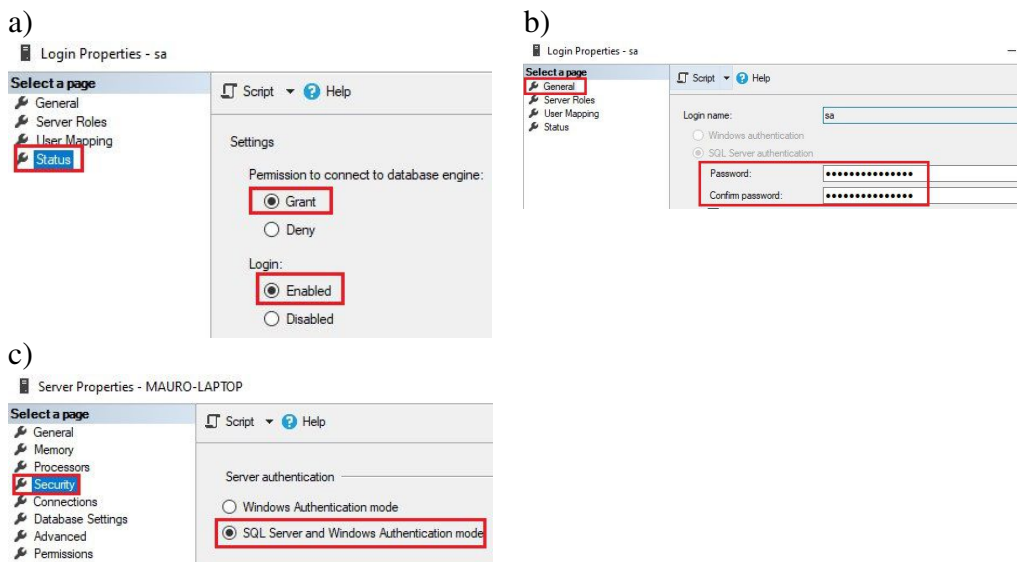


Rys. A.1: Konfiguracja TCP/IP

Konfiguracja konta:

- należy zalogować się połączyć z serwerem SQL z użyciem Windows Authentication,

- w folderze Security/Logins dla użytkownika „sa” należy ustawić możliwość logowania oraz dać możliwość łączenia się z bazą danych co pokazano na rysunku A.2a
- dla użytkownika „sa” należy również zmienić hasło na: „**maurycy123**” co pokazano na rysunku A.2b,
- dla właściwości serwera w zakładce „security” należy ustawić Server authentication na SQL Server and Windows Authentication co pokazano na rysunku A.2c,
- należy zrestartować komputer by ustawienia zadziałały,
- teraz można logować się w SSMS przy użyciu SQL Server Authentication



Rys. A.2: Konfiguracja konta a) uprawnienia, b) zmiana hasła, c) typ uwierzytelniania

Następnym krokiem jest importowanie bazy danych. Skrypt zawierający model i dane znajduje się w pobranym z repozytorium folderze Sql. Należy w SSMS utworzyć bazę danych o nazwie „**DeviceLotery**”. Następnie klikając prawym przyciskiem myszy na utworzoną bazę danych należy wybrać opcję „**New Query**”, przekopiować skrypt i kliknąć przycisk prawy przycisk myszy z opcją *execute*.

Uruchamianie aplikacji serwerowej: Należy z linii komend uruchomić komendę: `java -jar ComputerManagementTool-0.0.1-SNAPSHOT.jar`

Uruchamianie aplikacji klienckiej: Należy zainstalować pakiet serve z wykorzystaniem komendy: `npm install -g serve`. Następnie należy wykonać polecenie `serve -s build`

Logowanie do aplikacji klienckiej: W systemie istnieją konta z różnymi uprawnieniami. W tabeli A.1 zebrano istniejące loginy, hasła i role. Możliwa jest rejestracja pracownika na widoku W-1b (rysunek 5.3).

Tab. A.1: Loginy i hasła aplikacji klienckiej

Login	Hasło	Rola
admin	admin	Administrator
mniewczas	maurycy	Pracownik
jkowalski	maurycy	Pracownik
lewy	maurycy	Pracownik