

Politechnika Krakowska

Wydział Inżynierii Elektrycznej i Komputerowej

Dokumentacja

Systemy Baz Danych

Projekt:Serwis Komputerowy

Paweł Wądołny
Marcin Zieliński

Spis treści

1. Cel i zakres projektu:	4
2. Charakterystyka użytkowników	4
3. Główne funkcje produktu	4
4. Wymagania Funkcjonalne	4
5. Wymagania Niefunkcjonalne	4
6. Architektura i technologie	4
7. Instalacja Oracle Database	5
8. Konfiguracja wysyłania emaili z localhost	5
9. Konfiguracja reCaptcha	6
10. Główne funkcje strony internetowej	6
10.1 Strona startowa	6
10.1.1 Przypomnienie hasła	7
10.1.2 Rejestracja	7
10.2 Strona serwis	8
10.2.1 Serwis – dodawanie zgłoszenia	8
10.2.2 Serwis – podgląd zgłoszeń	9
10.2.3 Serwis – szczegóły zgłoszenia	10
10.2.4 Edycja danych osobowych – zmiana hasła	11
11. Główne funkcje aplikacji serwisu	12
11.1 Okno logowania	12
11.2 Panel pracownika	13
11.2.1 Panel pracownika – nowe zgłoszenia	14
11.2.2 Panel pracownika – zarządzanie aktualnymi naprawami	15
11.2.3 Panel pracownika – Kasjer	19
11.3 Panel administratora	20
11.3.1 Panel administratora – zarządzanie kontami pracowników	21
11.3.2 Panel administratora – zarządzanie cenami usług	23
11.3.3 Panel administratora – zarządzanie magazynem	25
12. Proces powstawania bazy danych	26
13. Kod SQL tworzący bazę i opis struktur bazodanowych	28
13.1 Tabela Klienci	28
13.2 Tabela Pracownicy	29
13.3 Tabela Umowy	30
13.4 Tabela Zamowienie_naprawy	30

13.5 Tabela Komputery	31
13.6 Tabela Płatności.....	31
13.7 Tabela Prace_naprawcze	31
13.8 Tabela Cennik.....	32
13.9 Tabela Prace_naprawcze_czesci.....	32
13.10 Tabela Czesci_zamienne	33
13.11 Tabela Kategorie.....	33
13.12 Tabela Producenci	33
13.13 Tabela Magazyn	34
14. Kod PL/SQL.....	34
14.1 ADD_ACCOUNT	34
14.2 ADD_ITEMS.....	35
14.3 ADD_NEW_ITEM.....	36
14.4 ADD_NEW_SERVICE.....	36
14.5 DELETE_ACCOUNT.....	37
14.6 DISMISS_WORKER	37
14.7 DOSERVICE.....	38
14.8 END_REPAIR.....	38
14.9 EXCHANGEPART	39
14.10 TAKE_WORK	39
15. Przykładowe zapytania do bazy	40
16. Plany wykonania bardziej złożonych zapytań	40
17. Wnioski	41

1. Cel i zakres projektu:

Celem projektu jest stworzenie strony internetowej dla serwisu komputerowego, oraz aplikacji na komputer stacjonarny dla administratora i pracowników serwisu. Aplikacja będzie działała w następujący sposób: Na serwerze umieszczamy bazę danych Oracle DataBase 12c. Strona internetowa serwisu jest umieszczona na serwerze z bazą danych, a aplikacja na komputer stacjonarny znajduje się na komputerach serwisu.

Zakres projektu obejmuje stworzenie strony i aplikacji z intuicyjnym GUI.

2. Charakterystyka użytkowników

- **Administrator** – jego zadaniem jest przydzielenie zadań serwisowych do danych pracowników.
- **Pracownik serwisu** – ma możliwość zmiany statusu zlecenia oraz potwierdzenie wykonania naprawy. Podczas diagnozy komputera wpisuje potrzebne czynności do bazy danych.
- **Użytkownik** – ma możliwość rejestracji i dodawania nowych zgłoszeń naprawy oraz możliwość podglądu szczegółów już istniejących zgłoszeń

3. Główne funkcje produktu

- a. Serwer bazy danych (Oracle DataBase 12c)
- b. Aplikacja serwisu
- c. Strona kliencka

4. Wymagania Funkcjonalne

- a. Łączność strony internetowej z bazą danych
- b. Łączność aplikacji serwisu z bazą danych
- c. Dodawanie zleceń
- d. Zmiana statusu zleceń
- e. Informacja o płatności

5. Wymagania Niefunkcjonalne

- a. Stabilność – aplikacja musi działać niezawodnie
- b. Intuicyjność – aplikacja powinna być przyjazna dla użytkownika; niedoświadczony klient nie może mieć problemów z obsługą
- c. Łatwość instalacji – system powinien być możliwy do wdrożenia w jeden dzień roboczy

6. Architektura i technologie

- a. Do stworzenia strony internetowej zostanie użyty język php oraz html.
- b. Bazą danych będzie Oracle DataBase 12c
- c. Aplikacja serwisu zostanie wykonana w języku Java wersja 8.

7. Instalacja Oracle Database

- 1) Zainstalować Oracle Database 12c – przy instalacji SID bazy ustawić na: orcl
<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>
- 2) Pobrać serwer Apache np. XAMPP
<https://www.apachefriends.org/ro/download.html>
- 3) Pobrać InstantClient12_2 (dla Oracle database 12c)
<https://www.oracle.com/technetwork/topics/winsoft-085727.html>
- 4) Po instalacji bazy i serwera apache wypakować instantclient np. na dysku C:/
- 5) W zmiennych środowiskowych systemu dodać do parametru PATH (w zmiennych systemowych) ścieżkę do folderu z instantclient (np. C:/instantclient_12_2)
- 6) W pliku php.ini (domyślnie C:\xampp\php\php.ini) odkomentować (usunąć ;) z linii extension=oci8_12c
- 7) W bazie danych utworzyć użytkownika login: serwis pass: serwis
- 8) Zaimportować bazę danych
- 9) Skopiować pliki ze stroną do folderu C:\xampp\htdocs\nazwa-strony (gdzie nazwa-strony to nazwa np. serwis lub serwis-komputerowy)
- 10) Uruchomić serwer Apache (w XAMPP)
- 11) W przypadku chęci zmiany użytkownika, hasła, należy zmienić dane logowania w pliku
setup-connect.php . Po zmianie danych, można uruchomić skrypt testujący z pliku
basetest.php
- 12) W przeglądarce wpisać localhost/nazwa-strony (gdzie nazwa strony to nazwa folderu z plikami w folderze htdocs)

8. Konfiguracja wysyłania emaili z localhost

W celu działania funkcji „przypomnij hasło” korzystając z localhost niezbędna jest konfiguracja pakietu XAMPP:

- 1) W pliku C:/xampp/sendmail/sendmail.ini wpisujemy
(Jeżeli nie mamy zainstalowanego pakietu sendmail razem z xampp należy go ręcznie doinstalować do folderu C:/xampp/sendmail)

```
smtp_server=smtp.gmail.com
smtp_port=25
error_logfile=error.log
debug_logfile=debug.log
auth_username=<username>
auth_password=<password>
force_sender=<e-mail username>@gmail.com
```

gdzie <username> to nazwa użytkownika poczty gmail, a <password> to hasło.
Jeżeli mamy podwójną autoryzację w poczcie (token) to musimy utworzyć „hasło do aplikacji”

<https://security.google.com/settings/security/apppasswords>

- 2) W pliku C:/xampp/php/php.ini należy odkomentować i edytować linie:

```
[mail function]
; For Win32 only.
SMTP = smtp.gmail.com
smtp_port = 25

; For Win32 only.
sendmail_from = <e-mail username>@gmail.com

sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"
```

- 3) Edytować plik setup-mail.php w folderze strony
Należy w nim wpisać mail, z którego będą wysyłane wiadomości.

9. Konfiguracja reCaptcha

W celu zapewnienia funkcjonalności ochrony przed botami, w formularzu rejestracyjnym został dodany mechanizm reCaptcha. Jest on domyślnie ustawiony na serwer localhost. Jeżeli chcemy wstawić stronę na serwer publiczny, należy wygenerować nowe klucze reCaptcha na stronie:

<https://www.google.com/recaptcha>

W zakładce „getreCaptcha”.

Następnie należy zmienić klucz publiczny i prywatny w pliku setup-recaptcha.php .

10. Główne funkcje strony internetowej

Strona internetowa jest minimalistyczna. Zapewnia ona podstawowe funkcjonalności, które są wymagane od serwisu. Przede wszystkim rejestrację użytkownika, dodawanie zgłoszeń i podgląd statusu. Logo strony jest minimalistyczne (SBD – Serwis Komputerowy) po jego kliknięciu zostajemy zawsze przeniesieni na stronę główną lub jeżeli jesteśmy zalogowani to zostajemy przeniesieni na stronę serwisu.

10.1 Strona startowa

SBD - Serwis Komputerowy

Login:

Hasło:

[Zarejestruj Się!](#)
[Nie pamiętam hasła](#)

Na stronie startowej możemy się zalogować do strony, zarejestrować użytkownika lub przypomnieć hasło.

10.1.1 Przypomnienie hasła

SBD - Serwis Komputerowy

Login:

Email:

W celu przypomnienia hasła użytkownika należy podać login i email użytkownika, jeżeli zostaną wprowadzone poprawne dane na adres email zostanie wysłane nowo wygenerowane hasło użytkownika (8 losowych znaków alfabetu).

10.1.2 Rejestracja

SBD - Serwis Komputerowy

Wypełnij formularz rejestracyjny:

Login:

E-mail:

Twoje hasło:

Powtórz hasło:

Imię:

Nazwisko:

Adres:

Kod pocztowy (bez -):

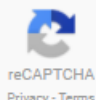
Miasto:

Województwo:

☒ Akceptuje [regulamin](#)



I'm not a robot



Formularz rejestracyjny posiada niezbędne pola odpowiadające tym w bazie danych. Każde pole przed wstawieniem do bazy jest sprawdzane. Dodatkowo zostało dodane zabezpieczenie

reCaptcha w celu ochrony przed botami próbującymi utworzyć wiele fałszywych kont. Ponadto formularz ma zabezpieczenie przed wstrzykiwaniem SQL – atak na bazę danych.

10.2 Strona serwis

SBD - Serwis Komputerowy

Witaj sbdserwis!

Email: sbdserwis@sbd.pl [\[Wyloguj się \]](#)

[\[Dodaj nowe zgłoszenie \]](#)

[\[Zobacz swoje zgłoszenia \]](#)

[\[Edytuj dane osobowe \]](#)

Po zalogowaniu się mamy możliwość przejścia na podstrony serwisu. Oferują kolejno dodawanie nowych zgłoszeń, podgląd już utworzonych zgłoszeń oraz edycję danych osobowych.

10.2.1 Serwis – dodawanie zgłoszenia

SBD - Serwis Komputerowy

Wypełnij formularz zgłoszenia usterki:

Producent komputera: (dla komputera stacjonarnego wpisz producenta płyty głównej)

Numer seryjny: (dla komputera stacjonarnego wpisz numer seryjny płyty głównej)

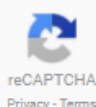
Rok Produkcji:

Opis usterki:

Komputer się zepsuł. Nie reaguje na przycisk włączający. Wyświetla czarny ekran.



I'm not a robot



Przy dodawaniu zgłoszenia musimy podać wymienione powyżej dane. Jeżeli w bazie istnieje już komputer o podanym numerze seryjnym jego producent i rok produkcji zostają zmienione na wprowadzone dane, po takiej zmianie użytkownik dostanie informację, że jego komputer znajdował się już w bazie danych. Opis usterki może wynosić maksymalnie 4000 znaków, jeżeli użytkownik przekroczy tę wartość, na ekranie wyświetli się odpowiedni komunikat.

10.2.2 Serwis – podgląd zgłoszeń

SBD - Serwis Komputerowy

Twoje aktualne zgłoszenia:

Numer Zlecenia	ID Klienta	ID Komputera	Status	Data dodania	Data zakończenia	Opis Usterki	Zobacz szczegóły
62	61	42	nowy	18/01/06		Komputer ma zbitą matrycę, przy uruchomieniu słychać pischczenie "3x długie 1x krótkie"	Zobacz
61	61	41	nowy	18/01/06		Komputer się zepsuł. Nie reaguje na przycisk włączający. Wyświetla czarny ekran.	Zobacz

W podglądzie zdarzeń widnieją wszystkie utworzone przez nas zgłoszenia. Aby przejść do szczegółów danego zgłoszenia wystarczy kliknąć przycisk „Zobacz”. Zostaniemy przeniesieni na automatycznie wygenerowaną podstronę.

10.2.3 Serwis – szczegóły zgłoszenia

[\[Powrót do listy Twoich zgłoszeń \]](#)

Szczegóły Twojego zamówienia nr 62 :

Numer Zlecenia	ID Klienta	ID Komputera	Status	Data dodania	Data zakończenia	Opis Usterki
62	61	43	naprawiony	18/01/07	18/01/07	Znowu nie działa

Opis statusów:

nowy - Zgłoszenie utworzone przez użytkownika, oczekiwanie na dostawę urządzenia

w naprawie - Sprzęt został zdiagnozowany i rozpoczęła się jego naprawa, postępy możesz śledzić w szczegółach

naprawiony - Sprzęt został naprawiony, oczekujemy na wpłatę kwoty podanej w szczegółach

zakończony - Otrzymałeś Twoją wpłatę, komputer został wysłany lub oczekuje na odbiór, jeżeli dostarczyłeś go osobiście

Jeżeli twój komputer posiada status nowy, powinieneś go dostarczyć do naszego serwisu, osobiście lub wysłać na adres:

SBD - Serwis Komputerowy

ul. Warszawska 24

31-155 Kraków

Nr. rachunku ING 17 1111 1111 2222 2222 2222 2222

Szczegóły komputera nr 43 :

ID Komputera	Producent	Rok Produkcji
43	AAtrox	2017

Szczegóły wykonanych prac dla komputera 43 :

Numer usługi	Nazwa usługi	Cena	Numer pracownika
101	usługa testowa	2	3
100	wymiana części	10	3
12	Włącz/ Wylącz komputer	5	3
SUMA	-	17	-

Szczegóły wymienionych części w komputerze nr 43 :

Numer Części	Nazwa modelu części	Nazwa kategorii	Nazwa producenta	Cena zakupu
3	4400	bateria	dell	50
1	model1	karta graficzna	AMD	100
2	model2	karta graficzna	AMD	200
SUMA	-	-	-	350

Szczegóły Płatności:

Numer Faktury	Do zapłaty	Zapłacono
4	367	0

Jeżeli twój komputer posiada status naprawiony, oczekujemy na Twoją wpłatę na nasze konto:

SBD - Serwis Komputerowy

ul. Warszawska 24

31-155 Kraków

Nr. rachunku ING 17 1111 1111 2222 2222 2222 2222

W szczegółach zgłoszenia widnieją wszystkie niezbędne informacje co zostało naprawione w komputerze oraz ile wynosi rachunek. Strona jest generowana automatycznie w PHP przy użyciu metody GET w pasku adresu. Wpisując adres: http://localhost/Serwis-komputerowy/szczegoly-zgloszenia.php?zgloszenie=numer_zgloszenia gdzie

numer_zgloszenia to odpowiedni numer zostaniemy przeniesieni do szczegółów danego zgłoszenia, aczkolwiek, jeżeli nie jesteśmy posiadaczem danego zgłoszenia (zgłoszenie należy do innego konta), to otrzymamy komunikat : „Zamówienie nr numer_zgloszenia nie należy do Ciebie” lub jeżeli zamiast numeru wpisujemy jakieś znaki, otrzymamy komunikat: „NIEPOPRAWNY NUMER ZGŁOSZENIA”. Jak widać zadbano o odpowiednie zabezpieczenie bazy przed ewentualnymi błędami. Przy każdym odświeżeniu strony PHP pobiera z bazy spis części i wykonanych usług dla danego zlecenia oraz sprawdza fakturę. Jeżeli dane w fakturze i w obliczonej sumie wykonanych usług się różnią to kolumna do_zapłaty jest aktualizowana.

10.2.4 Edycja danych osobowych – zmiana hasła

SBD - Serwis Komputerowy

[\[Zmiana hasła \]](#)

Twoje aktualne dane:

Imię: SBD
Nazwisko: Serwis
Email: sbdserwis@sbd.pl
Adres: Warszawska 24
Kod pocztowy: 31155
Miasto: Kraków
Województwo: małopolskie

Zmień swoje dane:

E-mail:

Imię:

Nazwisko:

Adres:

Kod pocztowy (bez -):

Miasto:

Województwo:

Użytkownik ma możliwość zmiany wszystkich swoich danych osobowych za wyjątkiem loginu. Adres email można zmieniać na dowolny inny, o ile nie został on już użyty przez innego użytkownika (jeżeli wpisujemy identyczny jak inny użytkownik, otrzymamy stosowny komunikat). Dane w formularzu przy pierwszym uruchomieniu podstrony są wpisywane automatycznie, w celu szybkiej ich zmiany.

SBD - Serwis Komputerowy

[\[POWRÓT \]](#)

Zmień swoje hasło:

Aktualne hasło:

Nowe hasło:

Powtórz nowe hasło:

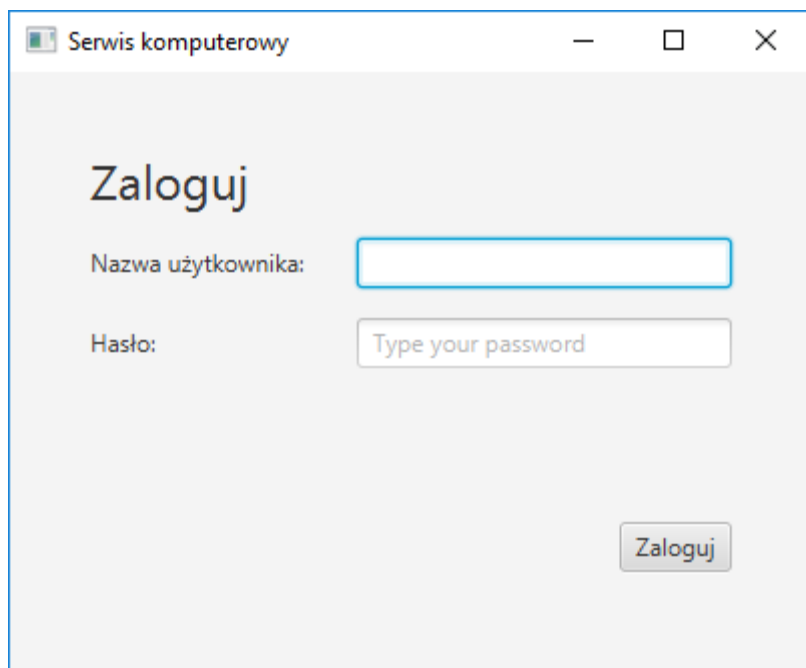
Użytkownik może wygodnie zmienić hasło, po wpisaniu poprzedniego i dwukrotnie nowego hasła w bazie zostanie ono zmienione. Hasło w bazie jest przechowywane w postaci

zahashowanej przez co nie ma możliwości jego odtworzenia, a także nie można go wysłać w przypomnieniu do użytkownika, tylko jest generowane nowe – losowe.

11. Główne funkcje aplikacji serwisu

11.1 Okno logowania

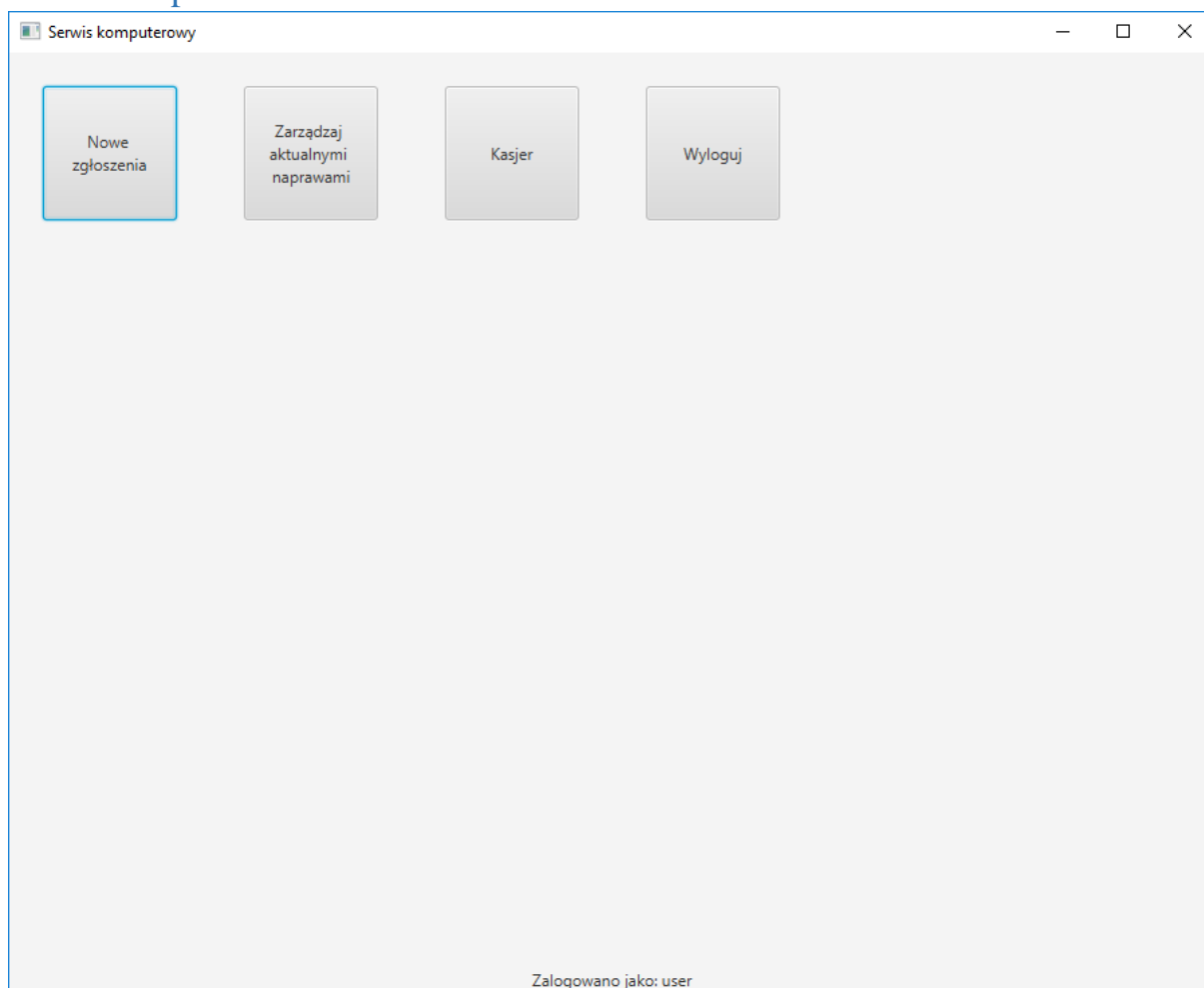
Po uruchomieniu aplikacji ukazuje się okno logowania, w którym pracownik musi wpisać swój login i hasło. Konta mogą mieć uprawnienia administratora lub zwykłego użytkownika (pracownik). Tworzone są w panelu administratora.



The screenshot shows a Windows-style window titled "Serwis komputerowy". Inside, the word "Zaloguj" is prominently displayed. Below it, there are two input fields: one for the username labeled "Nazwa użytkownika:" and one for the password labeled "Hasło:". The password field has a placeholder text "Type your password". At the bottom right of the window is a button labeled "Zaloguj".

Po kliknięciu „zaloguj” aplikacja sprawdza w bazie danych poprawność wprowadzonych danych i uprawnienia konta. Następnie przenosi użytkownika do panelu administratora lub panelu pracownika w zależności od typu konta. W przypadku podania błędnych danych, wyświetla odpowiedni komunikat.

11.2 Panel pracownika



Po zalogowaniu na konto pracownika, wyświetla się jego panel. Użytkownik może przeglądać nowe zgłoszenia, zarządzać naprawami, płatnościami lub się wylogować. Na dole widnieje jego login.

Serwis komputerowy

Nowe zgłoszenia

Zarządzaj aktualnymi naprawami

Kasjer

Wyloguj

Nowe naprawy:

Id naprawy	Opis usterki:	Status naprawy:	Data zgłoszenia:
62	Nie włącza się	nowy	2018-01-07 19:15:20.0

Rozpocznij naprawę:

Id naprawy

Podjemij

Zalogowano jako: user

Wartość wpisana przez pracownika jest weryfikowana w bazie, tak więc w przypadku wpisania błędnego Id, zostanie wyświetlony komunikat o błędzie.

11.2.2 Panel pracownika – zarządzanie aktualnymi naprawami

The screenshot shows a web application window titled "Serwis komputerowy". At the top, there are four buttons: "Nowe zgłoszenia", "Zarządzaj aktualnymi naprawami" (highlighted with a blue border), "Kasjer", and "Wyloguj". Below these buttons, there is a section "Wybierz nr zlecenia:" with a dropdown menu labeled "Id:" and an "OK" button. To the right of this is a section "Wymienione części:" with a table structure. Below the "Wybierz nr zlecenia:" section, there is a section "Wykonane usługi:" with a table structure. Below the "Wymienione części:" section, there is a section "Wymień części:" with three dropdown menus and three "OK" buttons. At the bottom center, there is a button labeled "Zakończ naprawę". At the very bottom, it says "Zalogowano jako: user".

Usługa:	Cena:
No content in table	

Rodzaj	Producent	Model	Cena:
No content in table			

Wymień części:

W tym menu pracownik może wypisywać wykonane przez siebie usługi, a także wymienione podzespoły.

Na początku należy wybrać z listy numer zlecenia i zatwierdzić przyciskiem OK. W liście „Wybierz nr zlecenia” będą dostępne zlecenia, które mają status „w naprawie” i zostały wybrane w menu „nowe zgłoszenia” przez danego pracownika. Nie ma możliwości, aby inny użytkownik wykonywał operacje na zleceniu, które do niego nie należy.

Nowe zgłoszenia

Zarządzaj aktualnymi naprawami

Kasjer

Wyloguj

Wybierz nr zlecenia:

Id:

62

OK

Wykonane usługi:

Usługa:	Cena:
Przyjęcie do serwisu	15.0
Czyszczenie komputera	15.0

Wykonaj usługę:

Czyszczenie komputera

OK

Wymienione części:

Rodzaj	Producent	Model	Cena:
No content in table			

Wymień części:

OK

OK

Zakończ naprawę

Zalogowano jako: user

Po zatwierdzeniu numeru zlecenia, uaktywnia się rozwijana lista z dostępnymi usługami.

W tabeli „wykonane usługi” będą widoczne wszystkie wykonane usługi danego pracownika w aktualnym zleceniu. Automatycznie dodawana jest również cena z cennika, który modyfikowany może być jedynie przez administratora.

Strona 16 z 42

Nowe zgłoszenia

Zarządzaj aktualnymi naprawami

Kasjer

Wyloguj

Wybierz nr zlecenia:

Id:

62

OK

Wykonane usługi:

Usługa:	Cena:
Przyjęcie do serwisu	15.0
Czyszczenie komputera	15.0
wymiana czesci	10.0

Wymienione części:

Rodzaj	Producent	Model	Cena:
No content in table			

Wykonaj usługę:

wymiana czesci

OK

Wymień części:

bateria

dell

OK

OK

Zakończ naprawę

Zalogowano jako: user

Po wybraniu usługi „wymiana czesci” pracownik może doliczyć daną część do rachunku klienta. W tym celu należy wybrać z pierwszej rozwijanej listy po prawej stronie rodzaj podzespołu (np.: bateria, karta graficzna, płyta główna), z kolejnej listy nazwę producenta i zatwierdzić przyciskiem OK, znajdującym się obok. W tym momencie zostanie sprawdzony stan magazynu i w kolejnej rozwijanej liście dostępne będą tylko podzespoły, które znajdują się na stanie.

Strona 17 z 42

Nowe zgłoszenia

Zarządzaj aktualnymi naprawami

Kasjer

Wyloguj

Wybierz nr zlecenia:

Id:

62

OK

Wykonane usługi:

Usługa:	Cena:
Przyjęcie do serwisu	15.0
Czyszczenie komputera	15.0
wymiana czesci	10.0

Wymienione części:

Producent	Rodzaj	Model	Cena:
bateria	dell	4400	50.0

Wykonaj usługę:

OK

Wymień części:

bateria

dell

4400

OK

OK

Zakończ naprawę

Zalogowano jako: user

Po wybraniu konkretnego modelu dostępnego podzespołu, przyciskiem OK, można dopisać go do listy wymienionych części. Po tej czynności w tabeli „Wymienione części” zapisane zostaną informacje o producencie, typie i modelu podzespołu, a także o jego cenie.

Po wykonaniu wszystkich czynności pracownik może zakończyć naprawienie sprzętu, klikając na przycisk „Zakończ naprawę”. Spowoduje to zmianę statusu zlecenia na „naprawiony”, a także wyliczenie sumy, jaką klient musi zapłacić.

Strona 18 z 42

11.2.3 Panel pracownika – Kasjer

W tym oknie znajdują się wszystkie ukończone zlecenia, oczekujące na opłacenie przez klienta.

Nowe zgłoszenia Zarządzaj aktualnymi naprawami **Kasjer** Wyloguj

Zakończone naprawy do opłacenia:

ID	Imię	Nazwisko	Data rozpoczęcia	Data zakończenia	Status	Do zapłaty
62	Jan	Kowalski	2018-01-07 19:15:20.0	2018-01-07 20:06:03.0	naprawiony	90.0
61	mojeimie	mojenazwisko	2018-01-07 17:28:33.0	2018-01-07 17:39:53.0	naprawiony	125.0

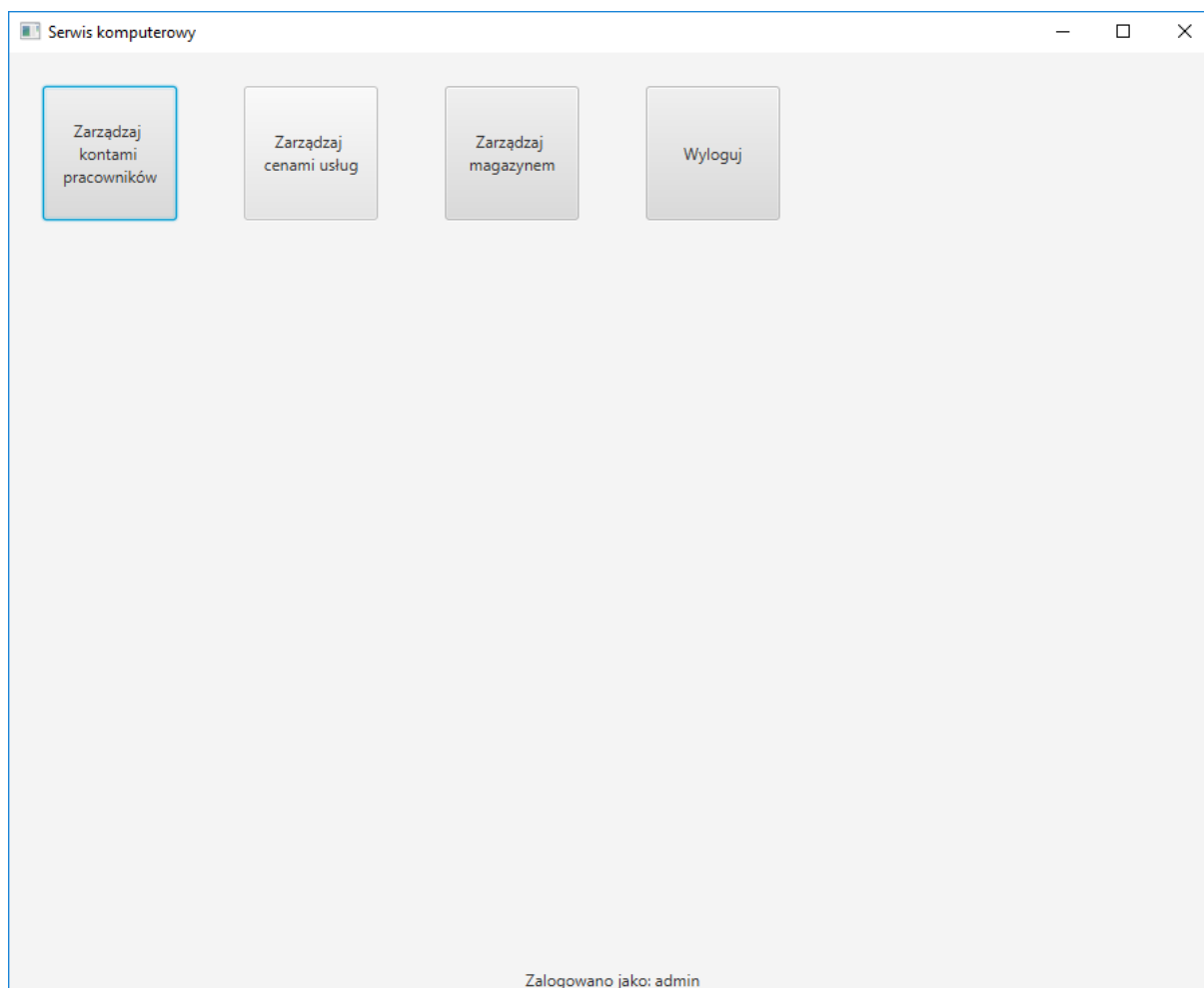
Potwierdź płatność:

ID

Zalogowano jako: user

W tabeli wyświetlone zostają: numer zlecenia, dane klienta oraz kwota do zapłaty. W przypadku, gdy klient zapłaci gotówką za usługę, pracownik wypisuje ID jego zlecenia, zatwierdza przyciskiem „Potwierdź”, co spowoduje zmianę statusu zlecenia na „zakończony” oraz wpisanie opłaconej kwoty do bazy.

11.3 Panel administratora



Po zalogowaniu się na konto z uprawnieniami administratora, użytkownik może wybrać zarządzanie kontami innych pracowników, zmianę cennika, zarządzanie magazynem lub się wylogować.

11.3.1 Panel administratora – zarządzanie kontami pracowników

Serwis komputerowy

Zarządzaj kontami pracowników

Zarządzaj cenami usług

Zarządzaj magazynem

Wyloguj

Dodaj konto:

Imię:

Miasto:

Login:

☐ Konto admina

Nazwisko:

Województwo:

Hasło:

Adres:

Nr tel:

Początek umowy:

RR/MM/DD

Kod pocztowy:

Pensja:

Koniec umowy:

RR/MM/DD

Dodaj

Id pracownika:	Numer umowy:	Imię:	Nazwisko:	Adres:	Kod pocztowy:	Miasto:	Województwo:	
1	1	admin	admin	admin	0	admin	admin	0
2	2	user	user	uuser	0	user	user	0
3	3	imie	nazwisko	adres	0	miasto	wojewodztwo	123
4	4	aa	bb	aaabb	111	gg	hh	12345
5	5	y	y	y	11	y	y	3

Zwolnij pracownika:

Id pracownika:

Zwolnij

Usuń konto:

Id pracownika:

Usuń

Zalogowano jako: admin

Po wybraniu tej opcji, administrator ma szereg możliwości. W formularzu ma możliwość dodania nowego konta dla zwykłego pracownika, lub administratora (gdy CheckBox „konto admina” jest zaznaczony). Po wypełnieniu formularza i kliknięciu „Dodaj” zostanie dodane nowe konto, co zostanie uwidocznione w tabeli poniżej. W tabeli znajdują się wszystkie konta użytkowników w systemie.

Zarządzaj kontami pracowników

Zarządzaj cenami usług

Zarządzaj magazynem

Wyloguj

Dodaj konto:

Imię:

Miasto:

Login:

☐ Konto admina

Nazwisko:

Województwo:

Hasło:

Adres:

Nr tel:

Początek umowy:

RR/MM/DD

Kod pocztowy:

Pensja:

Koniec umowy:

RR/MM/DD

Dodaj

Id pracownika:	Numer umowy:	Imię:	Nazwisko:	Adres:	Kod pocztowy:	Miasto:	Województwo:	
1	1	admin	admin	admin	0	admin	admin	0
2	2	user	user	uuser	0	user	user	0
3	3	imie	nazwisko	adres	0	miasto	wojewodztwo	123
4	4	aa	bb	nowy adres	111	gg	hh	12345
5	5	y	y	y	11	y	y	3

Zwolnij pracownika:

Id pracownika:

Zwolnij

Usuń konto:

Id pracownika:

Usuń

Zalogowano jako: admin

Istnieje również możliwość edycji danych. Po kliknięciu na pole w tabeli stanie się ono edytowalne. Można wtedy wprowadzić zmiany i po zatwierdzeniu ich klawiszem „Enter” dane konto zostanie uaktualnione w bazie.

Kolejna opcja to zwolnienie pracownika. Po wpisaniu jego Id i zatwierdzeniu przyciskiem „Zwolnij” zostaną odebrane mu uprawnienia do logowania do systemu, a także zostanie rozwiązania z nim umowa (zostanie wpisana aktualna data jako data zakończenia umowy).

Ostatnia możliwość to usunięcie konta pracownika. Za pomocą formularza w prawym dolnym rogu. Wtedy konto zostanie bezpowrotnie usunięte z bazy. Ta opcja nie jest zalecana, ponieważ zostaną utracone dane o konkretnym pracowniku.

11.3.2 Panel administratora – zarządzanie cenami usług

The screenshot shows a web application window titled "Serwis komputerowy". At the top, there are four buttons: "Zarządzaj kontami pracowników", "Zarządzaj cenami usług", "Zarządzaj magazynem", and "Wyloguj". Below these, under the heading "Dodaj usługę:", there is a form with two input fields: "Nazwa usługi" (containing "Reinstalacja systemu") and "Cena" (containing "50"). A "Dodaj" button is positioned below the "Cena" field. Under the heading "Edytuj ceny:", there is a table with the following data:

ID ▲	Nazwa usługi	Cena	
1	Przyjęcie do serwisu	15.0	
2	HARD RESET DYSKU	600.0	
11	Czyszczenie komputera	15.0	
12	Włącz/ Wylącz komputer	5.0	
100	wymiana części	10.0	
101	usługa testowa	2.0	

At the bottom of the window, it says "Zalogowano jako: admin".

W tym oknie użytkownik może dodać nową usługę wpisując jej nazwę oraz cenę i akceptując przyciskiem „Dodaj”. Nowa usługa zostanie dodana do bazy, co zostanie ukazane w tabeli poniżej.

Zarządzaj kontami pracowników

Zarządzaj cenami usług

Zarządzaj magazynem

Wyloguj

Dodaj usługę:

Nazwa usługi

Reinstalacja systemu

Cena

50

Dodaj

Edytuj ceny:

ID	Nazwa usługi	Cena	
102	Reinstalacja systemu	100	
101	usługa testowa	2.0	
1	Przyjęcie do serwisu	15.0	
100	wymiana części	10.0	
11	Czyszczenie komputera	15.0	
12	Włącz/ Wyłącz komputer	5.0	
2	HARD RESET DYSKU	600.0	

Zalogowano jako: admin

Tabela „Edytuj ceny” również ma możliwość zmiany wartości takich jak nazwa oraz cena. Po kliknięciu na daną komórkę, edycji oraz zatwierdzeniu przyciskiem „Enter” baza danych zostanie zaktualizowana.

11.3.3 Panel administratora – zarządzanie magazynem

Serwis komputerowy

Zarządzaj kontami pracowników Zarządzaj cenami usług Zarządzaj magazynem Wyloguj

Dodaj nowy podzespół

Nazwa kategorii:

Producent:

Model:

Cena:

Dodaj podzespół

ID:

Liczba sztuk:

Lista podzespołów w magazynie:

ID	Nazwa kategorii	Producent	Model	Cena	Na magazynie
3	bateria	dell	4400	50.0	9
1	karta graficzna	AMD	model	100.0	41
2	karta graficzna	AMD	model2	200.0	11
4	bateria	Asus	4600	50.0	10
5	karta graficzna	Sapphire	HD7850	250.0	10
6	karta sieciowa	TP-LINK	TF-3239DL	50.0	10

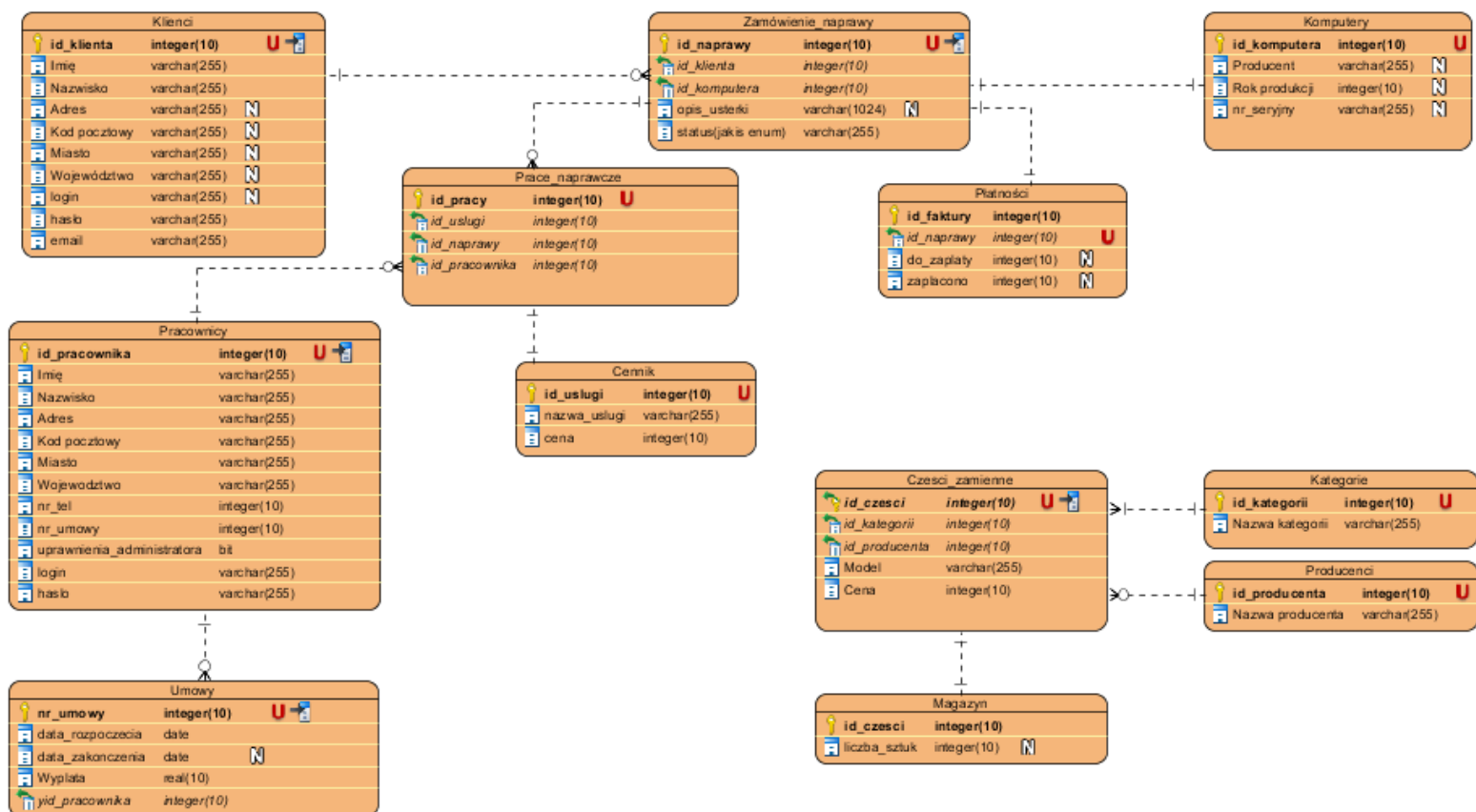
Zalogowano jako: admin

W tym oknie za pomocą formularza „Dodaj nowy podzespół” użytkownik może dodać nowy przedmiot, którego jeszcze nie posiadał na stanie. Jeśli dany producent bądź kategoria jeszcze nie istniała, również zostanie dodana do bazy do odpowiednich tabel. Nowy podzespół zostanie dodany z wartością „na magazynie” równą 0, która odpowiada stanowi magazynowemu. Operacja ta zostanie uwidoczniiona w tabeli „Lista podzespołów na magazynie”. Aby wprowadzić dostawę części, zwiększając ich liczbę w magazynie, należy wypełnić formularz po prawej stronie „Dodaj podzespół”. Spowoduje to zwiększenie liczby sztuk na stanie o wprowadzoną wartość.

12. Proces powstawania bazy danych

Projektowanie bazy danych rozpoczęliśmy od analizy wymagań serwisu komputerowego.

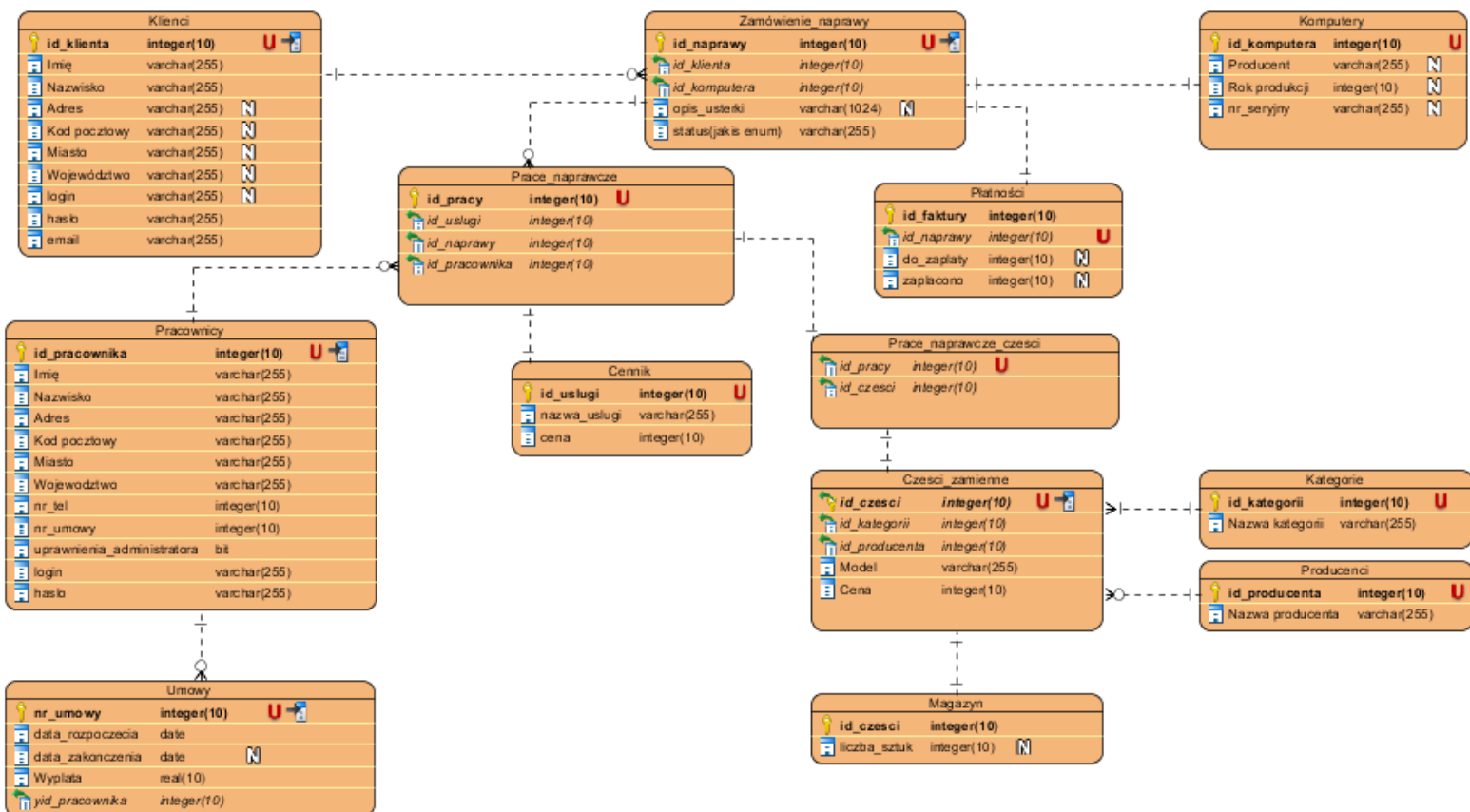
„Co będzie potrzebne?”. Na początku zajęliśmy się podstawowym zarysem tabel, tworząc je razem na kartce. Po stworzeniu podstawowych tabel i relacji przerysowaliśmy wszystko do komputera:



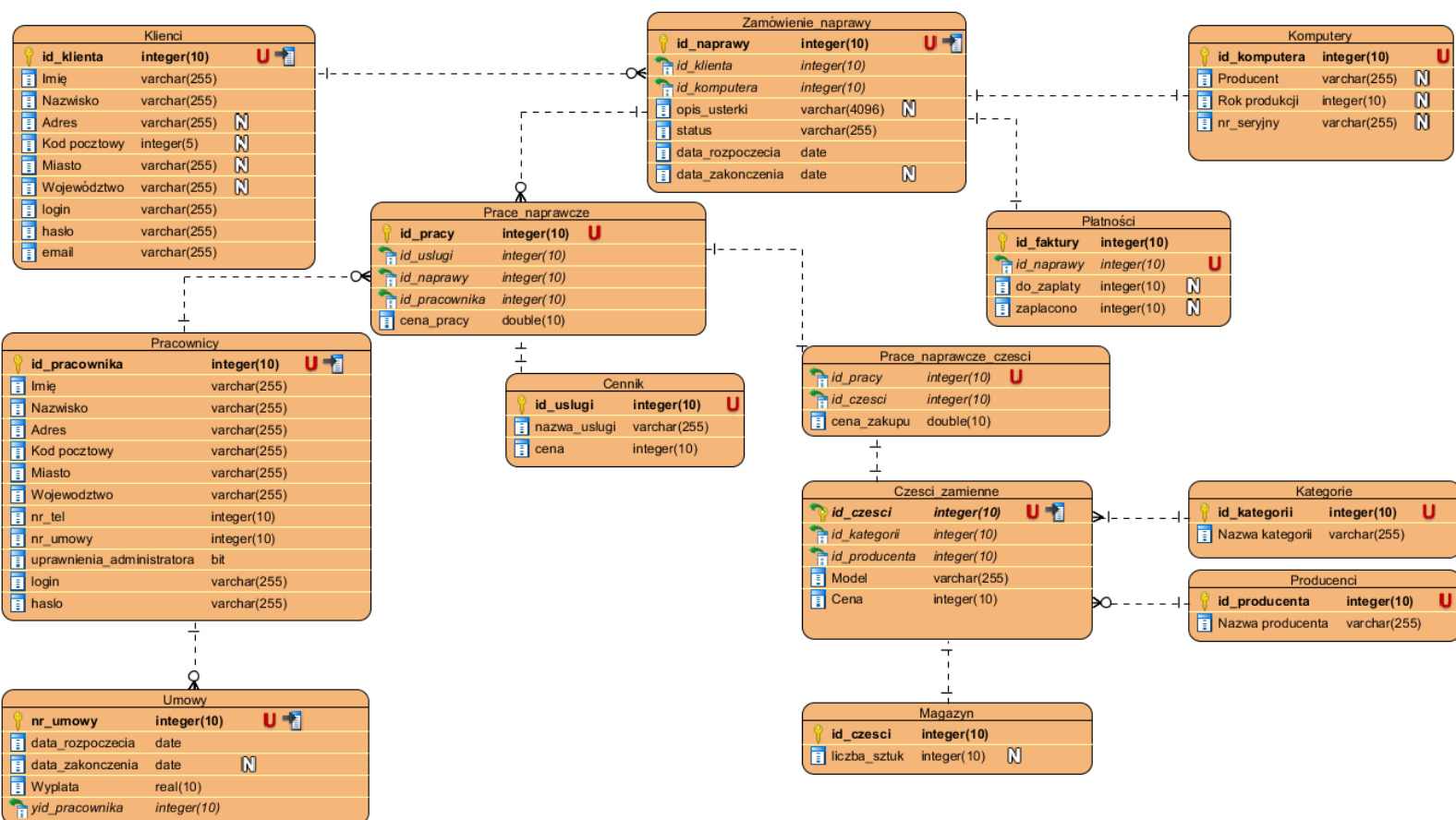
Jak widać na powyższym screenie baza danych została podzielona na 2 części – tą dotyczącą operowaniu na zamówieniach i tą związaną z dostawą i częściami zamiennymi.

W początkowym koncepcie bazy danych użytkownicy i pracownicy mieli logować się przy użyciu unikalnego id oraz hasła, jednak nie było to zbyt wygodne, gdyż użytkownicy musieliby pamiętać swoje numery, więc dodaliśmy loginy, które są znacznie lepsza opcja niż id. Do tabeli KLIENCI dodaliśmy także email, w celu kontaktu z klientem.

Następnie przystąpiliśmy do implementacji aplikacji serwisu i strony klienckiej. Po zaimplementowaniu podstawowych funkcjonalności okazało się, że zapomnieliśmy o jednej ważnej rzeczy, a mianowicie w przypadku wymiany części zamiennej w komputerze nie mieliśmy żadnego śladu po tej operacji w bazie danych. Musieliśmy więc dodać kolejną tabelę do bazy PRACE_NAPRAWCZE_CZESCI, która łączy nam prace naprawcza wraz z użytymi w niej częściami. Mogliśmy w ten sposób wyświetlić klientowi na stronie listę części, które zostały wymienione. Poniższy screen przedstawia wprowadzone zmiany:



Gdy zbliżaliśmy się już do końca pisania aplikacji i strony, napotkaliśmy jeszcze jeden problem. Ceny części i usług mogą się zmieniać. Więc jeżeli nie przechowujemy historii cen w bazie, to gdy klient wejdzie na stronę i zobaczy swoje zgłoszenie sprzed pół roku, to zobaczy całkowicie inne ceny niż były w dniu naprawy. W naszych głowach zrodziło się proste rozwiązanie. Wystarczyło do tabeli PRACE_NAPRAWCZE oraz PRACE_NAPRAWCZE_CZESCI dodać kolumnę z ceną w dniu wykonania usługi / zakupu części. Dzięki temu przy dodawaniu danej pracy czy wymianie danej części wystarczy skopiować daną cenę z cennika do właśnie utworzonej kolumny. Dzięki temu możemy wyświetlać klientowi cenę z dnia zakupu/ wykonania usługi, a nie aktualną, która może się zmieniać nawet codziennie. Tym sposobem uzyskaliśmy naszą finalną bazę danych, która zmieniała się do ostatnich chwil powstawania aplikacji. Efekt końcowy jest widoczny poniżej:



13. Kod SQL tworzący bazę i opis struktur bazodanowych

13.1 Tabela Klienci

Klienci		
	id_klienta	integer(10)
	Imię	varchar(255)
	Nazwisko	varchar(255)
	Adres	varchar(255)
	Kod pocztowy	integer(5)
	Miasto	varchar(255)
	Województwo	varchar(255)
	login	varchar(255)
	hasło	varchar(255)
	email	varchar(255)

Tabela zawiera wszystkie dane osobowe klientów, a także login i hasło (w postaci zahashowanej) niezbędne do logowania.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."KLIENCI"
2. (
3.     "ID_KLIENTA" NUMBER(10,0),
4.     "IMIE" VARCHAR2(255 CHAR),
5.     "NAZWISKO" VARCHAR2(255 CHAR),
6.     "ADRES" VARCHAR2(255 CHAR),
7.     "KOD_POCZTOWY" VARCHAR2(255 CHAR),
8.     "MIASTO" VARCHAR2(255 CHAR),
9.     "WOJEWODZTWO" VARCHAR2(255 CHAR),
10.    "HASLO" VARCHAR2(255 CHAR),
11.    "LOGIN" VARCHAR2(255 CHAR),
12.    "EMAIL" VARCHAR2(255 CHAR)
13. )
```

13.2 Tabela Pracownicy







Pracownicy	
 id_pracownika	integer(10) 
 Imię	varchar(255)
 Nazwisko	varchar(255)
 Adres	varchar(255)
 Kod pocztowy	varchar(255)
 Miasto	varchar(255)
 Wojewodztwo	varchar(255)
 nr_tel	integer(10)
 nr_umowy	integer(10)
 uprawnienia_administratora	bit
 login	varchar(255)
 haslo	varchar(255)

Tabela zawiera dane osobowe zatrudnionych pracowników oraz login i hasło niezbędne do zalogowania w aplikacji serwisu. Za pomocą zmiennej uprawnienia_administratora określamy czy pracownik jest zwykłym użytkownikiem czy adminem.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."PRACOWNICY"
2. (
3.     "ID_PRACOWNIKA" NUMBER(10,0),
4.     "IMIE" VARCHAR2(255 CHAR),
5.     "NAZWISKO" VARCHAR2(255 CHAR),
6.     "ADRES" VARCHAR2(255 CHAR),
7.     "KOD_POCZTOWY" VARCHAR2(255 CHAR),
8.     "MIASTO" VARCHAR2(255 CHAR),
9.     "WOJEWODZTWO" VARCHAR2(255 CHAR),
10.    "NR_TEL" NUMBER(10,0),
11.    "NR_UMOWY" NUMBER(10,0),
12.    "HASLO" VARCHAR2(255 CHAR),
13.    "UPRAWNIENIA_ADMIN" NUMBER(3,0),
14.    "LOGIN" VARCHAR2(255 CHAR)
15. )
```

13.3 Tabela Umowy









Umowy			
 nr_umowy	integer(10)	U	
 data_roz poczenia	date		
 data_zakonczenia	date	N	
 Wyplata	real(10)		
 yid_pracownika	integer(10)		

Zawiera informacje o umowach z pracownikami.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."UMOWY"  
2. ( "NR_UMOWY" NUMBER(10,0),  
3.   "DATA_ROZPOCZECIA" DATE,  
4.   "DATA_ZAKONCZENIA" DATE,  
5.   "WYPLATA" FLOAT(126),  
6.   "ID_PRACOWNIKA" NUMBER(10,0)  
7. )
```

13.4 Tabela Zamowienie_naprawy





Zamówienie naprawy			
 id_naprawy	integer(10)	U	
 id_klienta	integer(10)		
 id_komputera	integer(10)		
 opis_usterki	varchar(4096)	N	
 status	varchar(255)		
 data_roz poczenia	date		
 data_zakonczenia	date	N	

Przechowuje informację o złożonych zleceniach naprawy. Data_roz poczenia jest ustawiana automatycznie na bieżący czas przy dodawaniu zgłoszenia. Do tej tabeli trafia opis_usterki z formularza ze strony internetowej.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."ZAMOWIENIE_NAPRAWY"  
2. ( "ID_NAPRAWY" NUMBER(10,0),  
3.   "ID_KLIENTA" NUMBER(10,0),  
4.   "ID_KOMPUTERA" NUMBER(10,0),  
5.   "OPIS_USTERKI" VARCHAR2(4000 CHAR),  
6.   "DATA_ROZPOCZECIA" DATE,  
7.   "DATA_ZAKONCZENIA" DATE,  
8.   "STATUS" VARCHAR2(255 BYTE)  
9. )
```

13.5 Tabela Komputery





Komputery			
	id_komputera	integer(10)	U
	Producent	varchar(255)	N
	Rok produkcji	integer(10)	N
	nr_seryjny	varchar(255)	N

Zawiera informację o naprawianym komputerze.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."KOMPUTERY"  
2. (  
3.     "ID_KOMPUTERA" NUMBER(10,0),  
4.     "PRODUCENT" VARCHAR2(255 CHAR),  
5.     "ROK_PRODUKCJI" NUMBER(10,0),  
6.     "NR_SERYJNY" VARCHAR2(30 BYTE)  
7. )
```

13.6 Tabela Płatności






Płatności			
	id_faktury	integer(10)	
	id_naprawy	integer(10)	U
	do_zapłaty	integer(10)	N
	zapłacono	integer(10)	N

Przechowuje numery faktur i kwoty do zapłacenia. Po zaksięgowaniu płatności pracownik serwisu potwierdza płatność za naprawę i do danej komórki zostaje wpisana kwota jaką zapłacono.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."PLATNOSCI"  
2. (  
3.     "ID_FAKTURY" NUMBER(10,0),  
4.     "ID_NAPRAWY" NUMBER(10,0),  
5.     "DO_ZAPLATY" FLOAT(126),  
6.     "ZAPLACONO" FLOAT(126)  
7. )
```

13.7 Tabela Prace_naprawcze




Prace_naprawcze			
	id_pracy	integer(10)	U
	id_uslugi	integer(10)	
	id_naprawy	integer(10)	
	id_pracownika	integer(10)	
	cena_pracy	double(10)	

Przechowuje spis wszystkich czynności jakie wykonano w danym sprzęcie. Przechowuje także cenę z dnia wykonania czynności.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."PRACE_NAPRAWCZE"  
2.   (  
3.     "ID_PRACY" NUMBER(10,0),  
4.     "ID_USLUGI" NUMBER(10,0),  
5.     "ID_NAPRAWY" NUMBER(10,0),  
6.     "ID_PRACOWNIKA" NUMBER(10,0),  
7.     "CENA_PRACY" FLOAT(63)  
8.   )
```

13.8 Tabela Cennik




Cennik		
	id_uslugi	integer(10) U
	nazwa_uslugi	varchar(255)
	cena	integer(10)

Przechowuje nazwy i aktualne ceny usług.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."CENNIK" ( "ID_USLUGI" NUMBER(10,0), "NAZWA_USLUGI" VARCHAR2(255 CHAR),  
2.   "CENA" FLOAT(126))
```

13.9 Tabela Prace_naprawcze_czesci






Prace_naprawcze_czesci		
	id_pracy	integer(10) U
	id_czesci	integer(10)
	cena_zakupu	double(10)

Wiąże prace naprawcze i części jakich do nich użyto. Przechowuje także cenę zakupu danej części z dnia jej użycia.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."PRACE_NAPRAWCZE_CZESCI"  
2.   (  
3.     "ID_PRACY" NUMBER(10,0),  
4.     "ID_CZESCI" NUMBER(10,0),  
5.     "CENA_ZAKUPU" FLOAT(63)  
6.   )
```


13.10 Tabela Czesci_zamienne

Czesci_zamienne		
 id_czesci	integer(10)	U
 id_kategorii	integer(10)	
 id_producenta	integer(10)	
 Model	varchar(255)	
 Cena	integer(10)	

Przechowuje Model danej części oraz jej aktualną cenę. W celu uniknięcia redundancji danych Kategoria części jak i nazwa producenta jest przechowywana w osobnej tabeli.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."CZESCI_ZAMIENNE"  
2. (  
3.     "ID_CZESCI" NUMBER(10,0),  
4.     "ID_KATEGORII" NUMBER(10,0),  
5.     "ID_PRODUCENTA" NUMBER(10,0),  
6.     "MODEL" VARCHAR2(255 CHAR),  
7.     "CENA" FLOAT(126)  
8. )
```

13.11 Tabela Kategorie



Kategorie		
 id_kategorii	integer(10)	U
 Nazwa kategorii	varchar(255)	

Przechowuje nazwy kategorii części.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."KATEGORIE"  
2. (  
3.     "ID_KATEGORII" NUMBER(10,0),  
4.     "NAZWA_KATEGORII" VARCHAR2(255 CHAR)  
5. )
```

13.12 Tabela Producenci



Producenci		
 id_producenta	integer(10)	U
 Nazwa producenta	varchar(255)	

Przechowuje nazwy producentów części. W kolejnych wersjach jest możliwość łatwego uzupełnienia tabeli o dodatkowe kolumny np. Adres producenta czy telefon.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."PRODUCENCI"  
2.   ( "ID_PRODUCENTA" NUMBER(10,0),  
3.     "NAZWA_PRODUCENTA" VARCHAR2(255 CHAR)  
4.   )
```

13.13 Tabela Magazyn

Magazyn			
	id_czesci	integer(10)	
	liczba_sztuk	integer(10)	N

W jest przechowywany aktualny stan magazynu dla danej części.

Kod tworzący:

```
1. CREATE TABLE "SERWIS"."MAGAZYN"  
2.   ( "ID_CZESCI" NUMBER(10,0),  
3.     "LICZBA_SZTUK" NUMBER(10,0)  
4.   )
```

14. Kod PL/SQL

14.1 ADD_ACCOUNT

```
create or replace PROCEDURE "ADD_ACCOUNT"  
(  
  pFirstName IN varchar2,  
  pLastName IN varchar2,  
  pAddress IN varchar2,  
  pPostCode IN NUMBER,  
  pCity IN varchar2,  
  pState IN varchar2,  
  pPhone IN NUMBER,  
  pAdmAcc IN NUMBER,  
  pContractStart IN DATE,  
  pContractEnd IN DATE,  
  pLogin IN varchar2,  
  pPass IN varchar2,  
  pSalary IN FLOAT  
)  
IS  
vContractNo NUMBER;  
vAccNo NUMBER;  
BEGIN  
  
  SELECT (max(NR_UMOWY)+1) INTO vContractNo FROM SERWIS.UMOWY;  
  SELECT (max(ID_PRACOWNIKA)+1) INTO vAccNo FROM SERWIS.PRACOWNICY;  
  
  INSERT INTO SERWIS.UMOWY (  
    DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, WYPLATA, NR_UMOWY, ID_PRACOWNIKA)  
  VALUES (  
    pContractStart, pContractEnd, pSalary, vContractNo, vAccNo);  
  
  INSERT INTO SERWIS.PRACOWNICY (  
    IMIE, NAZWISKO, ADRES, KOD_POCZTOWY, MIASTO, WOJEWODZTWO, NR_TEL,  
    LOGIN, HASLO, UPRAWNIENIA_ADMIN, ID_PRACOWNIKA, NR_UMOWY)  
  VALUES (  
    pFirstName, pLastName, pAddress, pPostCode,  
    pCity, pState, pPhone, pLogin, pPass, pAdmAcc, vAccNo, vContractNo);  
  
commit;  
END;
```

Procedura ta dodaje konto nowego użytkownika, przyjmując jako parametry jego dane. Uaktualnia tabelę PRACOWNICY oraz UMOWY.

14.2 ADD_ITEMS

```
create or replace PROCEDURE "ADD_ITEMS" (  
  pPartId NUMBER,  
  pPiecesNo NUMBER  
)  
IS  
  vPiecesBefore NUMBER;  
  vPiecesAfter NUMBER;  
BEGIN  
  SELECT LICZBA_SZTUK INTO vPiecesBefore FROM MAGAZYN WHERE ID_CZESCI = pPartId;  
  vPiecesAfter := vPiecesBefore + pPiecesNo;  
  
  UPDATE MAGAZYN  
  SET LICZBA_SZTUK = vPiecesAfter  
  WHERE ID_CZESCI = pPartId;  
commit;  
END;
```

Procedura ADD_ITEMS dodaje liczbę sztuk danego podzespołu o podanym ID do magazynu, aktualizując tabelę MAGAZYN.

14.3 ADD_NEW_ITEM

```
create or replace PROCEDURE "ADD_NEW_ITEM" (  
    pCategory VARCHAR2,  
    pProducer VARCHAR2,  
    pModel VARCHAR2,  
    pPrice NUMBER  
)  
IS  
    vId NUMBER;  
    vCategoryId NUMBER;  
    vProducerId NUMBER;  
BEGIN  
    SELECT max(ID_CZESCI)+1 INTO vId FROM CZESCI_ZAMIENNE;  
    BEGIN  
        SELECT ID_KATEGORII INTO vCategoryId FROM KATEGORIE  
        WHERE UPPER(NAZWA_KATEGORII) LIKE UPPER(pCategory); -- znajdz id kategorii  
    EXCEPTION WHEN NO_DATA_FOUND THEN vCategoryId := null;  
    END;  
    -- jesli nie ma dodaj  
    IF vCategoryId IS NULL THEN  
        SELECT max(ID_KATEGORII)+1 INTO vCategoryId FROM KATEGORIE;  
        INSERT INTO KATEGORIE(ID_KATEGORII, NAZWA_KATEGORII)  
        VALUES (vCategoryId, pCategory);  
    END IF;  
    BEGIN  
        SELECT ID_PRODUCENTA INTO vProducerId FROM PRODUCENCI  
        WHERE UPPER(NAZWA_PRODUCENTA) LIKE UPPER(pProducer); -- znajdz id producenta  
    EXCEPTION WHEN NO_DATA_FOUND THEN vProducerId := null;  
    END;  
    -- jesli nie ma dodaj  
    IF vProducerId IS NULL THEN  
        SELECT max(ID_PRODUCENTA)+1 INTO vProducerId FROM PRODUCENCI;  
        INSERT INTO PRODUCENCI(ID_PRODUCENTA, NAZWA_PRODUCENTA)  
        VALUES (vProducerId, pProducer);  
    END IF;  
  
    INSERT INTO CZESCI_ZAMIENNE (ID_CZESCI, ID_KATEGORII, ID_PRODUCENTA, MODEL, CENA)  
    VALUES (vId, vCategoryId, vProducerId, pModel, pPrice);  
  
    INSERT INTO MAGAZYN (ID_CZESCI, LICZBA_SZTUK)  
    VALUES (vId, 0);  
    commit;  
END;
```

Procedura ADD_NEW_ITEM dodaje nowy podzespól do bazy do tabeli CZESCI_ZAMIENNE, generując nowe ID_CZESCI, wpisując cenę oraz znajdując identyfikatory producenta i kategorii w tabelach PRODUCENCI i KATEGORIE. Jeśli dana kategoria lub producent nie istniała wcześniej w bazie, wychwytuje ten stan za pomocą wyjątku NO_DATA_FOUND i dodaje nowe wartości do tych tabel.

14.4 ADD_NEW_SERVICE

```
create or replace PROCEDURE "ADD_NEW_SERVICE" (  
    pName VARCHAR2,  
    pPrice FLOAT  
)  
IS  
    vId NUMBER;  
BEGIN  
    SELECT max(ID_USLUGI)+1 INTO vId FROM CENNIK;  
    INSERT INTO CENNIK (ID_USLUGI, NAZWA_USLUGI, CENA)  
    VALUES (vId, pName, pPrice);  
    commit;  
END;
```

Procedura ADD_NEW_SERVICE dodaje nową usługę do cennika, wprowadzając nową wartość do tabeli CENNIK.

14.5 DELETE_ACCOUNT

```
create or replace PROCEDURE "DELETE_ACCOUNT"
(
  pAccId IN NUMBER
)
IS
BEGIN
  DELETE FROM SERWIS.PRACOWNICY WHERE ID_PRACOWNIKA = pAccId;
  DELETE FROM SERWIS.UMOWY WHERE ID_PRACOWNIKA = pAccId;
commit;
END;
```

Procedura DELETE_ACCOUNT umożliwia bezpowrotne usunięcie konta pracownika, kasując wpisy o danym id konta z tabel PRACOWNICY oraz UMOWY.

14.6 DISMISS_WORKER

```
create or replace PROCEDURE "DISMISS_WORKER"
(
  pAccId IN NUMBER
)
IS
vDismissDate DATE;
BEGIN
  vDismissDate := SYSDATE;

  UPDATE SERWIS.PRACOWNICY
  SET UPRAWNIENIA_ADMIN = -1
  WHERE ID_PRACOWNIKA = pAccId;

  UPDATE SERWIS.UMOWY
  SET DATA_ZAKONCZENIA = vDismissDate
  WHERE ID_PRACOWNIKA = pAccId;
commit;
END;
```

Procedura DISMISS_WORKER umożliwia zwolnienie pracownika i uniemożliwienie mu logowania poprzez ustawienie flagi uprawnień na -1 oraz wpisanie aktualnej daty, jako daty rozwiązania umowy.

14.7 DOSERVICE

```
create or replace PROCEDURE "DOSERVICE" (  
pServiceId NUMBER, -- nr usługi z cennika  
pRepairId NUMBER, -- nr pracy (ktory sprzet)  
pWorkerId NUMBER -- nr pracownika  
)  
IS  
vWorkNo NUMBER;  
vPrice NUMBER;  
BEGIN  
SELECT MAX(ID_PRACY)+1 INTO vWorkNo FROM PRACE_NAPRAWCZE;  
SELECT CENA INTO vPrice FROM CENNIK WHERE ID_USLUGI = pServiceId;  
INSERT INTO PRACE_NAPRAWCZE (ID_PRACY, ID_USLUGI, ID_NAPRAWY, ID_PRACOWNIKA, CENA_PRACY)  
VALUES (vWorkNo, pServiceId, pRepairId, pWorkerId, vPrice);  
commit;  
END;
```

Procedura DOSERVICE umożliwia wykonanie danej usługi na rzecz klienta. Odbywa się to poprzez wprowadzenie wartości do tabeli PRACE_NAPRAWCZE. W niej zostają powiązane usługi z procesem naprawy danego sprzętu.

14.8 END_REPAIR

```
create or replace PROCEDURE "END_REPAIR" (  
pWorkNo NUMBER  
)  
IS  
vBillNo NUMBER;  
vToPay NUMBER;  
vToPayServices NUMBER;  
vToPayParts NUMBER;  
BEGIN  
SELECT max(ID_FAKTURY)+1 INTO vBillNo FROM PLATNOSCI;  
SELECT SUM(CENA_PRACY) INTO vToPayParts FROM PRACE_NAPRAWCZE WHERE ID_NAPRAWY = pWorkNo;  
SELECT SUM(CENA_ZAKUPU) INTO vToPayServices FROM PRACE_NAPRAWCZE CZESCI  
NATURAL JOIN PRACE_NAPRAWCZE WHERE ID_NAPRAWY = pWorkNo;  
  
vToPay := vToPayParts + vToPayServices;  
UPDATE ZAMOWIENIE_NAPRAWY  
SET STATUS = 'naprawiony',  
DATA_ZAKONCZENIA = SYSDATE  
WHERE ID_NAPRAWY = pWorkNo;  
  
INSERT INTO PLATNOSCI(  
ID_FAKTURY, ID_NAPRAWY, DO_ZAPLATY, ZAPLACONO)  
VALUES (vBillNo, pWorkNo, vToPay, 0);  
commit;  
END;
```

Procedura END_REPAIR umożliwia zakończenie procesu naprawy danego komputera. Zmienia status w tabeli ZAMOWIENIE_NAPRAWY na „naprawiony” oraz przypisuje aktualną datę, jako DATA_ZAKONCZENIA. Ponadto w tabeli PLATNOSCI wpisuje łączną kwotę, jaką klient musi zapłacić za naprawę, tj. kwota usług i części.

14.9 EXCHANGEPART

```
create or replace PROCEDURE "EXCHANGEPART" (
pIdNaprawy NUMBER,
pIdCzesci NUMBER
)
IS
-- numer pracy naprawczej gdy wymieniano czesc (100- kod wymiany czesci)
-- nie bierze pod uwage przypadku, ze wpisano usluga wymiana czesci i nie ma vPracyNaprawczej
vPracaNaprawcza NUMBER;
vPiecesBefore NUMBER;
vPiecesAfter NUMBER;
vPrice NUMBER;
BEGIN
SELECT CENA INTO vPrice FROM CZESCI_ZAMIENNE WHERE ID_CZESCI = pIdCzesci;
SELECT max(ID_PRACY) INTO vPracaNaprawcza FROM PRACE_NAPRAWCZE WHERE ID_NAPRAWY=pIdNaprawy AND ID_USLUGI = 100;

INSERT INTO PRACE_NAPRAWCZE_CZESCI (ID_PRACY, ID_CZESCI, CENA_ZAKUPU)
VALUES (vPracaNaprawcza, pIdCzesci, vPrice);

SELECT LICZBA_SZTUK INTO vPiecesBefore FROM MAGAZYN WHERE ID_CZESCI = pIdCzesci;
vPiecesAfter := vPiecesBefore - 1;

UPDATE MAGAZYN
SET LICZBA_SZTUK = vPiecesAfter
WHERE ID_CZESCI = pIdCzesci;

commit;
END;
```

Procedura EXCHANGEPART służy do zapisania informacji o wymienionym podzespolu podczas naprawy. Pobiera cenę danej części i wpisuje ją do tabeli PRACE_NAPRAWCZE_CZESCI razem z identyfikatorem części i danej pracy. Ponadto aktualizuje stan magazynu, zmniejszając liczbę sztuk o 1.

14.10 TAKE_WORK

```
create or replace FUNCTION "TAKE_WORK"
(
pUserId NUMBER,
pReportNo NUMBER -- nr zgłoszenia wybrany przez pracownika z tabeli zamowienie_naprawy
)
RETURN NUMBER
IS
PRAGMA AUTONOMOUS_TRANSACTION;
vFlag NUMBER;
vWorkNo NUMBER;
vPrice NUMBER;
BEGIN
SELECT max(ID_PRACY)+1 INTO vWorkNo FROM PRACE_NAPRAWCZE;
SELECT CENA INTO vPrice FROM CENNIK WHERE ID_USLUGI = 1; -- cena przyjęcia do serwisu
SELECT CASE WHEN EXISTS (SELECT * FROM ZAMOWIENIE_NAPRAWY WHERE ID_NAPRAWY = pReportNo AND STATUS = 'nowy')
THEN 1 ELSE 0 END INTO vFlag FROM dual;
IF vFlag = 1 THEN
INSERT INTO PRACE_NAPRAWCZE (ID_PRACY, ID_USLUGI, ID_NAPRAWY, ID_PRACOWNIKA, CENA_PRACY)
-- id usugi = 1 - przyjęcie do serwisu
VALUES (vWorkNo, 1, pReportNo, pUserId, vPrice);

UPDATE "SERWIS"."ZAMOWIENIE_NAPRAWY"
SET STATUS = 'w naprawie'
WHERE ID_NAPRAWY = pReportNo;
commit;
ELSE
vFlag := 0;
END IF;

RETURN vFlag;
END;
```

Funkcja TAKE_WORK umożliwia pracownikowi rozpoczęcie realizacji danego zlecenia. Sprawdza, czy pracownik podał prawidłowy identyfikator zlecenia. Jeśli nie, nic nie robi, tylko zwraca flagę 0. Jeśli identyfikator jest poprawny, dodaje nowe wartości do tabeli

PRACE_NAPRAWCZE oraz aktualizuje status naprawy w tabeli ZAMOWIENIE_NAPRAWY na „w naprawie”

Automatycznie nalicza klientowi usługę przyjęcia do serwisu.

15. Przykładowe zapytania do bazy

```
CALL ADD_ACCOUNT('Jan','Kowalski','adres',00000,'Kraków','małopolskie',123,0,'18/01/05','18/01/10','login','haslo',2000);
```

-wywołuje procedurę dodania użytkownika, jako parametry przyjmuje kolejno: imię, nazwisko, adres, kod pocztowy, miejscowość, województwo, numer telefonu, uprawnienia (0-user, 1-admin), daty rozpoczęcia i zakończenia umowy, login i hasło do serwisu oraz pensję.

```
select ID_NAPRAWY FROM PRACE_NAPRAWCZE p NATURAL JOIN ZAMOWIENIE_NAPRAWY  
| z WHERE ID_PRACOWNIKA = 1 AND STATUS NOT LIKE 'naprawiony' GROUP BY ID_NAPRAWY;
```

- wyświetla aktualne naprawy danego pracownika

```
select NAZWA_KATEGORII, NAZWA_PRODUCENTA, MODEL, CENA FROM PRACE_NAPRAWCZE_CZESCI  
NATURAL JOIN CZESCI_ZAMIENNE NATURAL JOIN PRACE_NAPRAWCZE NATURAL JOIN KATEGORIE  
NATURAL JOIN PRODUCENCI WHERE ID_NAPRAWY = 1;
```

- wyświetlenie wymienionych części w danej pracy naprawczej

```
SELECT c.ID_CZESCI, MODEL, CENA FROM CZESCI_ZAMIENNE c LEFT JOIN MAGAZYN m ON  
m.id_czesci = c.ID_CZESCI WHERE c.ID_KATEGORII = 1 AND c.ID_PRODUCENTA = 1 AND m.LICZBA_SZTUK > 0
```

- wyszukanie dostępnych części danego typu i danego producenta w magazynie

```
select CZESCI_ZAMIENNE.ID_CZESCI, NAZWA_KATEGORII, NAZWA_PRODUCENTA, MODEL, CENA, LICZBA_SZTUK  
from (CZESCI_ZAMIENNE NATURAL JOIN PRODUCENCI NATURAL JOIN KATEGORIE) LEFT JOIN MAGAZYN  
ON magazyn.id_czesci = CZESCI_ZAMIENNE.ID_CZESCI
```

- wypisanie stanu magazynu

```
select ID_NAPRAWY, IMIE as "IMIE_KLIENTA", NAZWISKO as "NAZWISKO_KLIENTA",  
DATA_ROZPOCZECIA, DATA_ZAKONCZENIA, STATUS, DO_ZAPLATY FROM ZAMOWIENIE_NAPRAWY  
NATURAL JOIN KLIENCI NATURAL JOIN PLATNOSCI WHERE STATUS = 'naprawiony';
```

- wypisanie napraw do opłacenia

16. Plany wykonania bardziej złożonych zapytań

Zapytanie wypisujące wszystkie wymienione części dla danego zlecenia naprawy.

```
SELECT * FROM PRACE_NAPRAWCZE p LEFT JOIN  
PRACE_NAPRAWCZE_CZESCI pc ON p.ID_PRACY = pc.ID_PRACY LEFT JOIN  
CZESCI_ZAMIENNE c ON pc.ID_CZESCI = c.ID_CZESCI LEFT JOIN  
PRODUCENCI pr ON c.ID_PRODUCENTA = pr.ID_PRODUCENTA LEFT JOIN  
KATEGORIE k ON c.ID_KATEGORII = k.ID_KATEGORII WHERE p.ID_NAPRAWY = '$a_naprawa';
```


Plan zapytania:

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		3	14
HASH JOIN (OUTER)		3	14
Access Predicates			
C.ID_PRODUCENTA=PR.ID_PRODUCENT			
NESTED LOOPS (OUTER)		3	14
STATISTICS COLLECTOR			
HASH JOIN (OUTER)		3	11
Access Predicates			
C.ID_KATEGORII=K.ID_KATE			
NESTED LOOPS (OUTER)		3	11
STATISTICS COLLECTOR			
HASH JOIN (OUTER)		3	8
Access Predicates			
PC.ID_CZESCI=			
NESTED LOOPS (OUT)		3	8
STATISTICS COL			
HASH JOIN (3	5
Access			
P.ID			
TABLE ACCESS (NAPRAWCZE)		3	2
INDID_NAPRAWY		3	1
TABLE ACCESS (NAPRAWCZE_CZESCI)		2	3
TABLE ACCESS (CZESCI_ZAMIENNE)		1	1
INDEX (UNIQUEPRIMARY_7)		1	0
Access			
PC.I			
TABLE ACCESS (FULLCZESCI_ZAMIENNE)		1	1
TABLE ACCESS (BY INDEX ROKATEGORIE)		1	1
INDEX (UNIQUE SCAN) PRIMARY_8		1	0
Access Predicates			
C.ID_KATEGORI			
TABLE ACCESS (FULL)	KATEGORIE	1	1
TABLE ACCESS (BY INDEX ROWID)	PRODUCENCI	1	1
INDEX (UNIQUE SCAN) PRIMARY_3		1	0
Access Predicates			
C.ID_PRODUCENTA=PR.ID_f			
TABLE ACCESS (FULL)	PRODUCENCI	1	1

Jak zapytanie jest złożone. Nie obeszło się bez dużej liczby joinów. Jeżeli chcemy przedostać się z numeru_zlecenia z tabeli ZAMOWIENIA_NAPRAWY do konkretnych części, które zostały wymienione musimy się liczyć wymaganym połączeniem wielu tabel, gdyż przechowujemy powtarzające się dane takie jak nazwy producentów oraz kategorie części w osobnych tabelach, w celu uniknięcia redundancji danych. Mając konkretny numer zamówienia naprawy jesteśmy w stanie w miarę szybko uzyskać pożądane wyniki. Połączenia tabel odbywają się za pomocą indexowanych kluczy głównych, co usprawnia całość zapytania.

17. Wnioski

Na etapie projektowania bazy pominieliśmy kilka znaczących w późniejszej implementacji aplikacji aspektów. Pominięcie loginów w bazie danych zostając przy samych hasłach nie było dobrym rozwiązaniem. Późniejsze dodanie tabeli PRACE_NAPRAWCZE_CZESCI zaowocowało możliwością wyświetlania na stronie szczegółów zamówienia spisu części, za które zapłacił klient. Lecz przy tworzeniu tej tabeli również nie uwzględniliśmy tego, że ceny części mogą ulec zmianie, a klient chciałby zobaczyć cenę, w dniu, kiedy dana część została zakupiona. Sprytnie to rozwiązaliśmy dodając do tej tabeli kolumnę CENA_ZAKUPU. Przy dodawaniu danej części do zlecenia wystarczy skopiować cenę z tabeli CZESCI_ZAMIENNE i tym oto prostym sposobem mamy historię cen.

Niektóre z powyższych problemów były ciężkie do przewidzenia, kiedy skupialiśmy się na innych ważnych aspektach m. in. Jak zaimplementować dane funkcjonalności w aplikacji i stronie. Jednak naszym zdaniem finalny efekt jest zadowalający.