



POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

Systemy Baz Danych

Projekt: Aplikacja wspomagająca pracę gabinetu kosmetycznego
Prowadzący - prof. Dr hab. Inż. Sergii Telenyk

Wydział	WIEiK	
Kierunek	Informatyka	
Zespół	Wąsik Hubert	33i
	Pawlikowska Anna	32i
	Piskorz Paweł	33i

Spis treści

1	Cel i zakres projektu	3
2	Charakterystyka użytkowników	3
3	Główne funkcje produktu	3
4	Wykorzystywane technologie	3
5	Wymagania funkcjonalne i нефункционалне	4
5.1	Wymagania funkcjonalne	4
5.2	Wymagania нефункционалне	5
6	Diagram ERD przed normalizacją	6
7	Diagramy DFD	7
7.1	Diagram kontekstowy	7
7.2	Diagram systemowy	8
8	Diagramy STD	9
8.1	Proces logowania użytkownika i rejestracji klienta	9
8.2	Proces dla osoby niezalogowanej	9
8.3	Proces zalogowanego klienta	10
8.4	Proces zalogowanego pracownika	10
8.5	Proces zalogowanego administratora	10
9	Diagram klas	11
10	Baza danych	12
10.1	Proces normalizacji schematu bazy danych	12
10.2	Schemat ERD po przeprowadzeniu normalizacji	13
10.3	Skrypt bazy danych	14
10.3.1	Tabela klient	14
10.3.2	Tabela pracownik	14
10.3.3	Tabela konto	15
10.3.4	Tabela wizyta	15
10.3.5	Tabela raport_odpadów	15
10.3.6	Tabela aktualność	15
10.3.7	Tabela wydarzenie	16
10.3.8	Tabela usługa	16
10.3.9	Tabela promocja	16
10.4	Procedury utworzone dla bazy danych	17
10.4.1	Procedura tworząca konto klienta	17
10.4.2	Procedura tworząca konto pracownika	17
10.4.3	Procedura tworząca usługę	18
10.4.4	Procedura tworząca wydarzenie (szkolenie / kongres / targi)	18
10.4.5	Procedura tworząca przegląd sprzętu	18
10.4.6	Procedura tworząca nieobecność pracownika	19
10.4.7	Procedura tworząca wiadomość od użytkownika do użytkownika	19
10.4.8	Procedura dodająca produkt do danej promocji	19
10.5	Schemat bazy danych w systemie	20

1 Cel i zakres projektu

Celem projektu jest utworzenie aplikacji, której odbiorcą jest zarówno właściciel gabinetu kosmetycznego, jak i pracownik oraz klient. Aplikacja służy do zarządzania gabinetem kosmetycznym oraz umożliwia potencjalnym klientom w łatwy sposób przeglądać ofertę gabinetu, zapisać się na konkretny zabieg bądź zgłosić reklamację dotyczącą wykonanej usługi.

Aplikacja musi spełniać wypisane niżej funkcjonalności niezbędne do działania gabinetu kosmetycznego, a których zakres skonsultowany został z właścicielem gabinetu.

2 Charakterystyka użytkowników

Administrator (Właściciel) – odpowiedzialny jest za zarządzanie istniejącymi kontami, usługami udostępnianymi przez gabinet oraz zasobami gabinetu. Posiada dostęp do danych klientów i pracowników, gdzie ma możliwość ich edytowania.

Użytkownik (Pracownik, Klient) – konto posiadające dwa typy – pracownik bądź klient. W zależności od typu użytkownika posiada dostęp do określonych funkcji systemu, takich jak zapisywanie się na zabieg czy zgłaszanie reklamacji.

3 Główne funkcje produktu

- 1) Logowanie do systemu.
- 2) Przedstawianie zakresu usług gabinetu.
- 3) Wspomaganie zarządzania zasobami ludzkimi.
- 4) Wspomaganie zarządzania zasobami gabinetu typu produkty/urządzenia.
- 5) Umożliwienie komunikacji pomiędzy klientem a pracownikiem.
- 6) Wspomaganie kierowania gabinetem.

4 Wykorzystywane technologie

- 1) Java oraz JavaServer
- 2) MySQL
- 3) JavaScript
- 4) HTML oraz CSS
- 5) Bootstrap Framework

5 Wymagania funkcjonalne i нефункционалне

5.1 Wymagania funkcjonalne

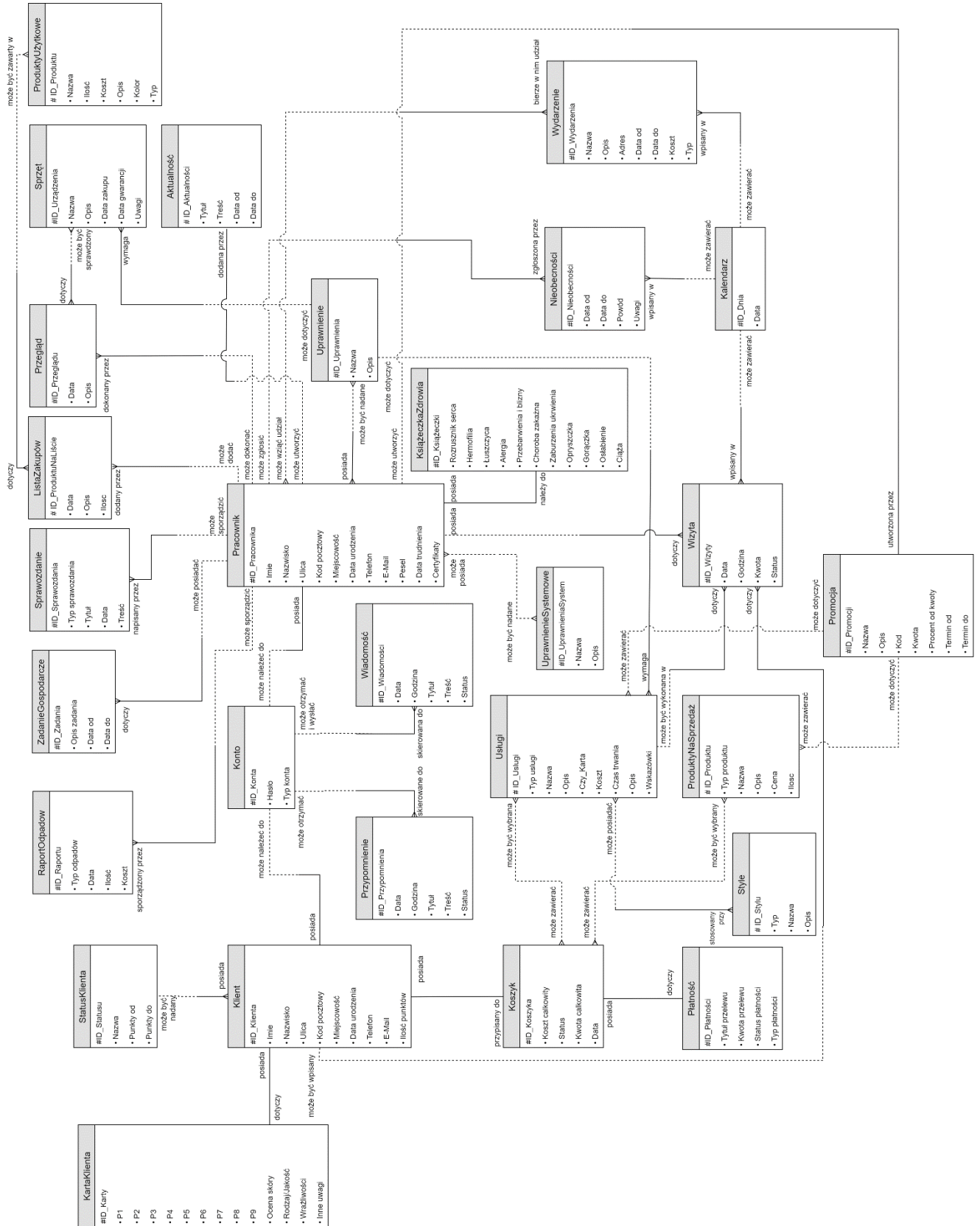
- 1) Logowanie i wylogowanie użytkowników – klient / administrator / pracownik.
- 2) Tworzenie kont pracowników, możliwość edycji danych kont
- 3) Zarządzanie uprawnieniami usługowymi przez administratora.
- 4) Dodawanie przez administratora terminu szkoleń i kongresów kosmetycznych, informacji o ich zakresie, wyświetlanie ich w systemie oraz możliwość zapisu przez pracownika.
- 5) Dodawanie przez administratora bądź zalogowanego w gabinecie pracownika klienta wraz z zamówioną usługą do grafiku (kalendarza), możliwość jego edycji i podglądu.
- 6) Tworzenie „Kart Informacyjnych” zawierające wskazania i przeciwwskazania do zabiegów oraz alergię i nietolerancję organizmu klienta możliwych do edytowania przez klienta.
- 7) Możliwość prowadzenia książeczki zdrowia pracowników.
- 8) Dodawanie zatwierdzonych osiągnięć – ukończone szkolenia oraz otrzymane certyfikaty – przez administratora do profili pracowników oraz ich prezentacja.
- 9) Dodawanie sprawozdania zawierającego przebieg ostatnio wykonanego przeglądu technicznego (wewnętrzny), serwisowego (zewnętrzny) oraz sporadyczne kontrolne (medyczne) wraz z datami ich przeprowadzenia.
- 10) Możliwość dodawania wybranych sprzętów użytkowych, informacji o nich oraz ich wyświetlanie.
- 11) Możliwość dodawania produktów sprzedażowych przez administratora.
- 12) Zarządzanie odpadami – utylizacja – generowanie sprawozdania odbioru zarejestrowanych odpadów (data oraz ilość), ustawienie okresowego przypomnienia o zbliżającym się terminie odbioru odpadów oraz generowanie rocznego bilansu utylizacji.
- 13) Rejestrowanie wizyt kontrolnych, takich jak sanepid, ZUS czy państwowa inspekcja pracy.
- 14) Promowanie social-medium gabinetu – dostęp do Facebook’a, Instagrama oraz Messengera gabinetu.
- 15) Dodawanie przez administratora informacji o aktualnych ofertach, pakietach sezonowych, okolicznościowych, świątecznych i konkursach oraz ich udostępnienie klientowi.
- 16) Możliwość prowadzenia spisu wykonanych usług, kto ich wykonał oraz kwota otrzymanej zapłaty.
- 17) Możliwość podglądu przez pracownika w swoim profilu informacji o wysokości nadchodzącej wypłaty.
- 18) Zgłaszanie prośby o przyznanie urlopu bądź poinformowanie o otrzymaniu zwolnienia lekarskiego przez pracownika administratorowi.
- 19) Możliwość wprowadzenia zakresu zadań gospodarczych poszczególnym pracownikom.
- 20) Dodawanie w systemie CV potencjalnego pracownika (w przypadku gdy poszukiwany jest pracownik) oraz przesłania go na mail administratora.
- 21) Możliwość konsultacji online poprzez chat online pomiędzy klientem a pracownikiem za pośrednictwem aplikacji Messenger.
- 22) Możliwość przeglądania oferty usług udostępnianych przez gabinet.
- 23) Możliwość wstępnej rezerwacji terminu wykonania wybranej usługi u wybranego pracownika.
- 24) Możliwość składania formularza reklamacyjnego za pośrednictwem e-mail.
- 25) Złożenie zamówienia przez klienta dotyczącego kupna produktu bądź usługi znajdującego się w ofercie gabinetu.

- 26) Usługa lojalnościowa – klient otrzymuje status stałego klienta po skorzystaniu z określonej ilości usług zatwierdzonych przez pracownika. Takiemu klientowi przysługują bonusy określone przez właściciela gabinetu.
- 27) Automatyczne wysyłanie przypomnienia klientowi o zatwierdzeniu wizyty na adres e-mail.
- 28) Informowanie klienta o procedurze przygotowania się do zabiegu.

5.2 Wymagania niefunkcjonalne

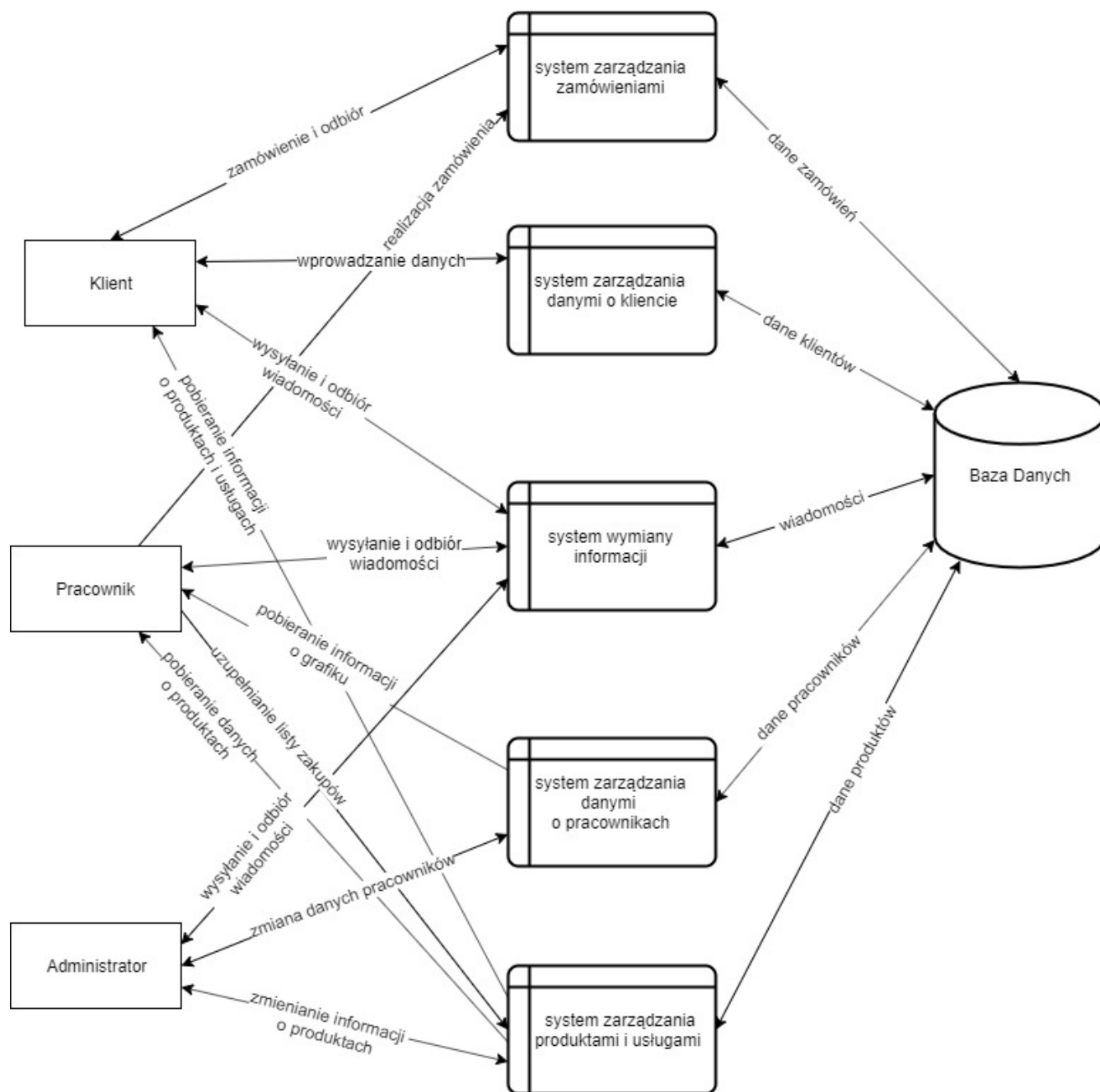
- 1) Aplikacja ma być niezawodna i bezbłędna, tzn. realizować swoje zadania w ściśle określonym przedziale czasu i robić to poprawnie.
- 2) Aplikacja ma być bezpieczna - rozumiemy przez to zabezpieczenia przed niepowołanym dostępem do aplikacji.
- 3) Aplikacja ma być niezależna od platformy - posiadać możliwość działania na każdym urządzeniu bez względu na to, na jakim systemie operacyjnym zostaje uruchomiona jak i posiadanej mocy obliczeniowej urządzenia.
- 4) Aplikacja ma być prosta w obsłudze - interfejs będzie intuicyjny, przejrzysty i czytelny.
- 5) Aplikacja ma stosować hierarchię dostępu - wynika to z podziału funkcjonalności: użytkownik zarejestrowany jako klient nie może mieć dostępu do funkcji administratora.
- 6) Aplikacja nie powinna udostępniać żadnych danych osobowych użytkowników osobom do tego nieuprawnionym.
- 7) Aplikacja nie może pozwalać na wprowadzanie istotnych zmian pracownikowi w systemie jeśli nie jest zalogowany za pośrednictwem sieci znajdującej się w gabinecie.

6 Diagram ERD przed normalizacją

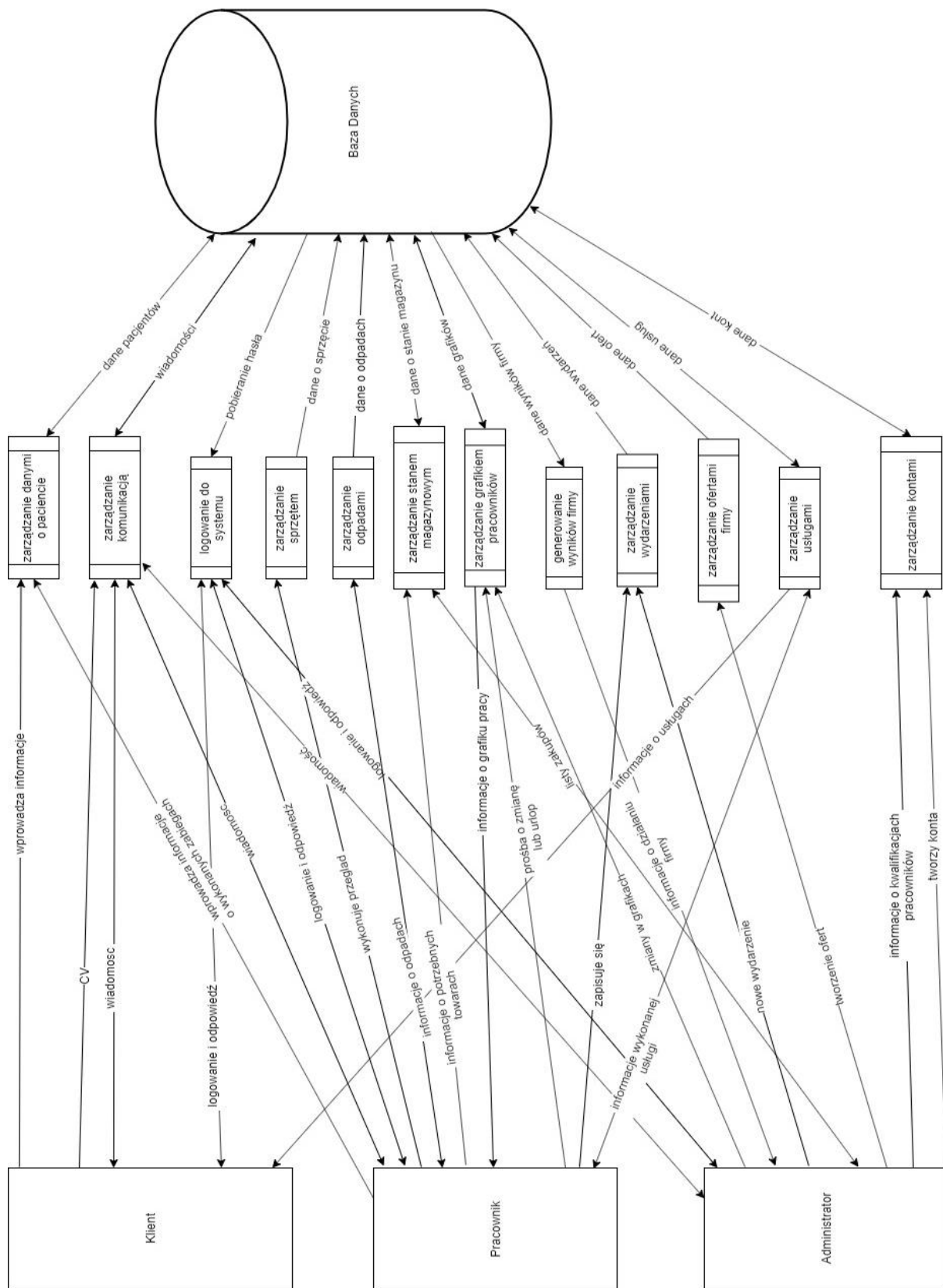


7 Diagramy DFD

7.1 Diagram kontekstowy



7.2 Diagram systemowy

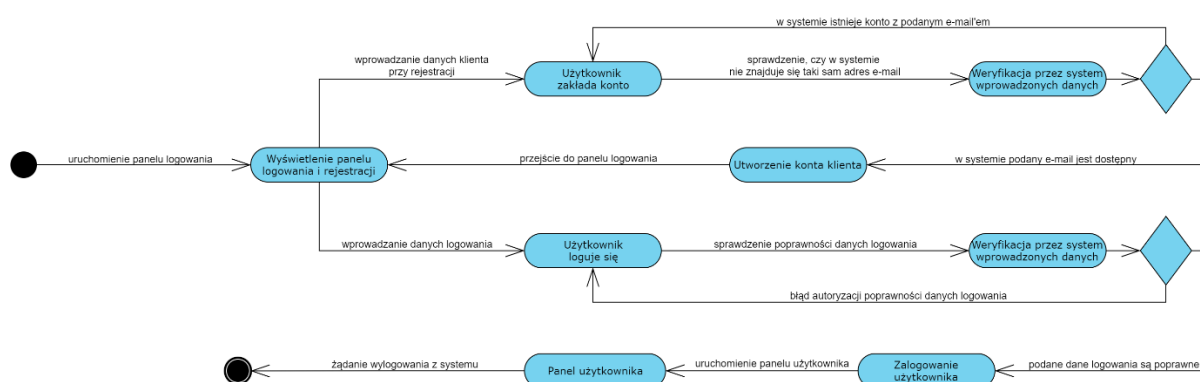


8 Diagramy STD

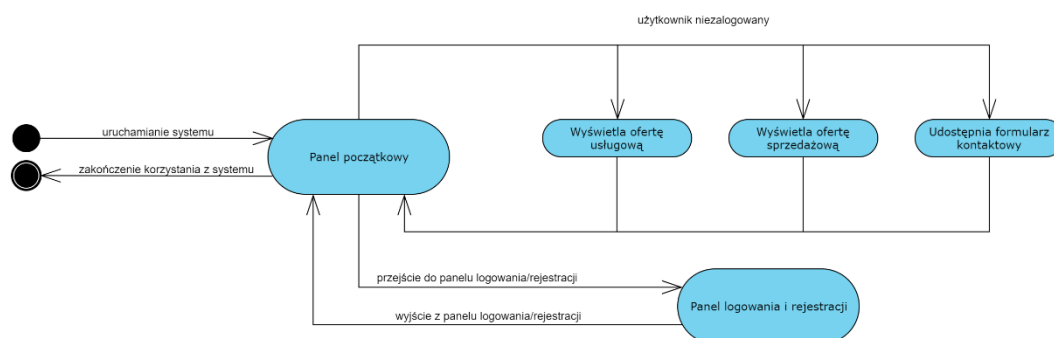
Poniżej zamieszczone zostały diagramy maszyn stanów dla poszczególnych funkcjonalności aplikacji:

- proces logowania użytkownika i rejestracji klienta
- proces dla osoby niezalogowanej w systemie
- proces dla zalogowanego w systemie klienta
- proces dla zalogowanego w systemie pracownika
- proces dla zalogowanego w systemie administratora

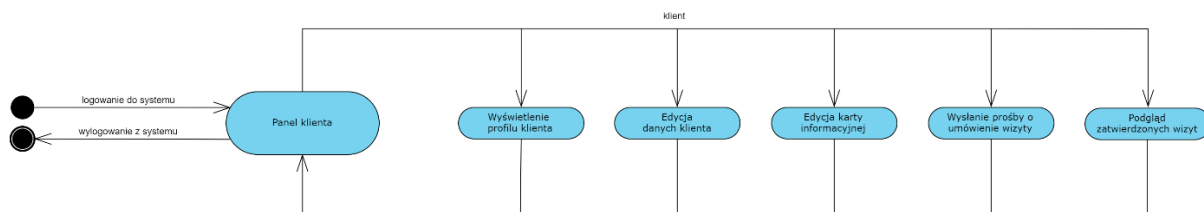
8.1 Proces logowania użytkownika i rejestracji klienta



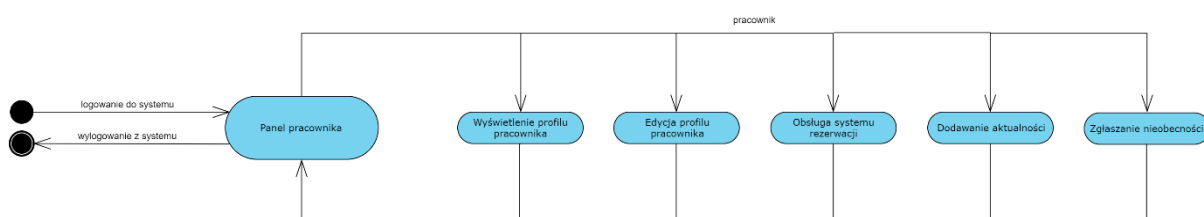
8.2 Proces dla osoby niezalogowanej



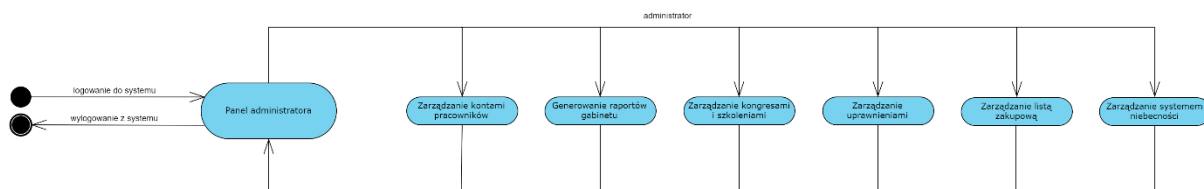
8.3 Proces zalogowanego klienta



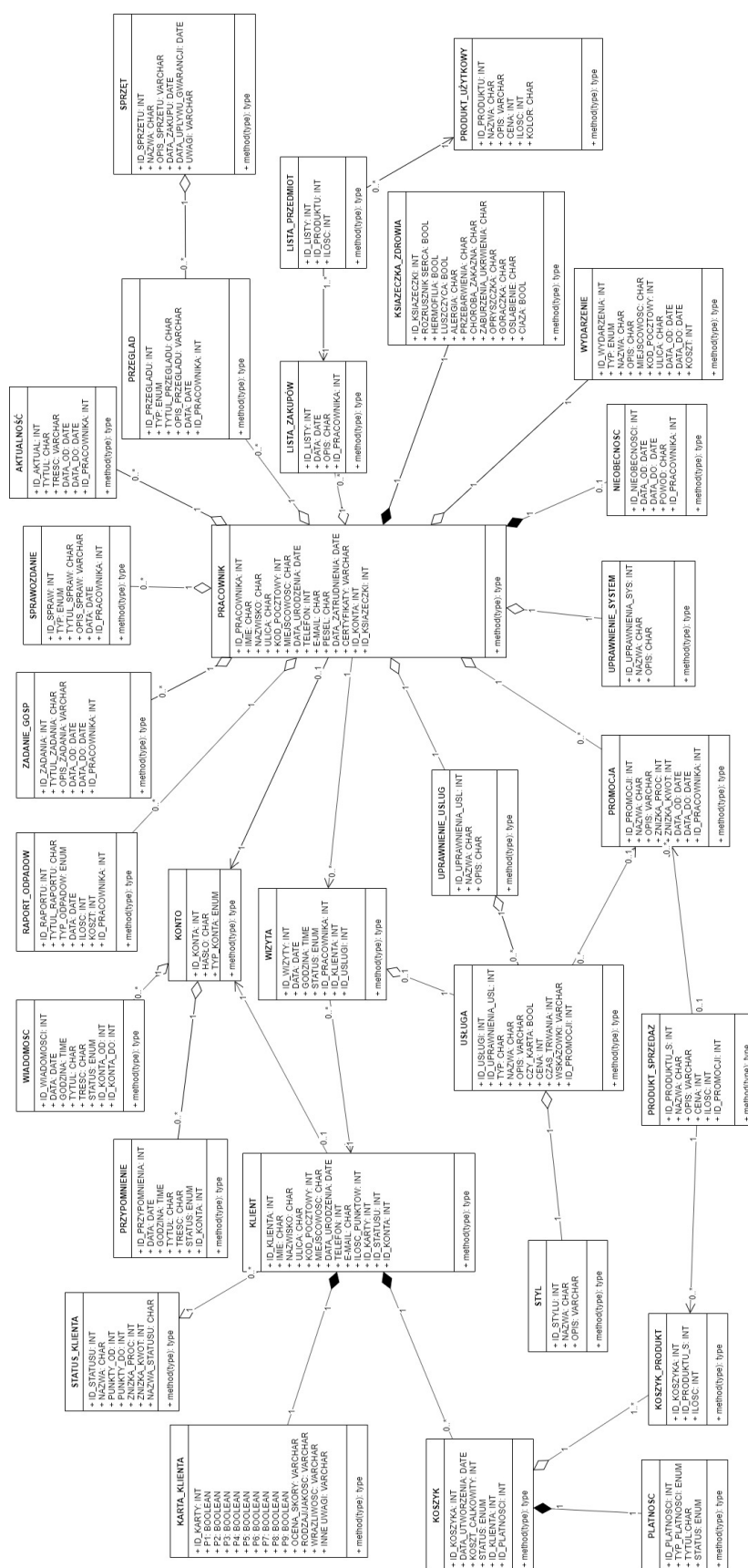
8.4 Proces zalogowanego pracownika



8.5 Proces zalogowanego administratora



9 Diagram klas



10 Baza danych

10.1 Proces normalizacji schematu bazy danych

Przedstawiony na pierwszym schemacie model bazy danych przekształciliśmy do 1NF, gdzie należało uwzględnić to, by każde pole przechowywało jedną informację oraz to, by każda encja mogła być identyfikowana za pośrednictwem własnego, oryginalnego i niepowtarzalnego klucza głównego.

W procesie normalizacji zauważyliśmy, iż:

- encja „kalendarz” jest encją zbędną i niepotrzebną, by spełnić jedno z założeń projektu jakim jest obsługa rezerwacji wizyt online. Zakładając, że w gabinecie każdy z pracowników powinien być obecny od godziny 10:00 do 18:00 (gabinet, dla którego tworzona jest aplikacja nie posiada zmian porannych/wieczorowych pracy pracowników) można w łatwy sposób stwierdzić, czy pracownik jest do dyspozycji w wybrany dzień o wybranej godzinie. Wystarczy jedynie sprawdzić, czy nie posiada wpisanej „nieobecności” (zatwierdzonego przez administratora systemu zwolnienia) oraz czy nie posiada o wybranej godzinie wizyty.

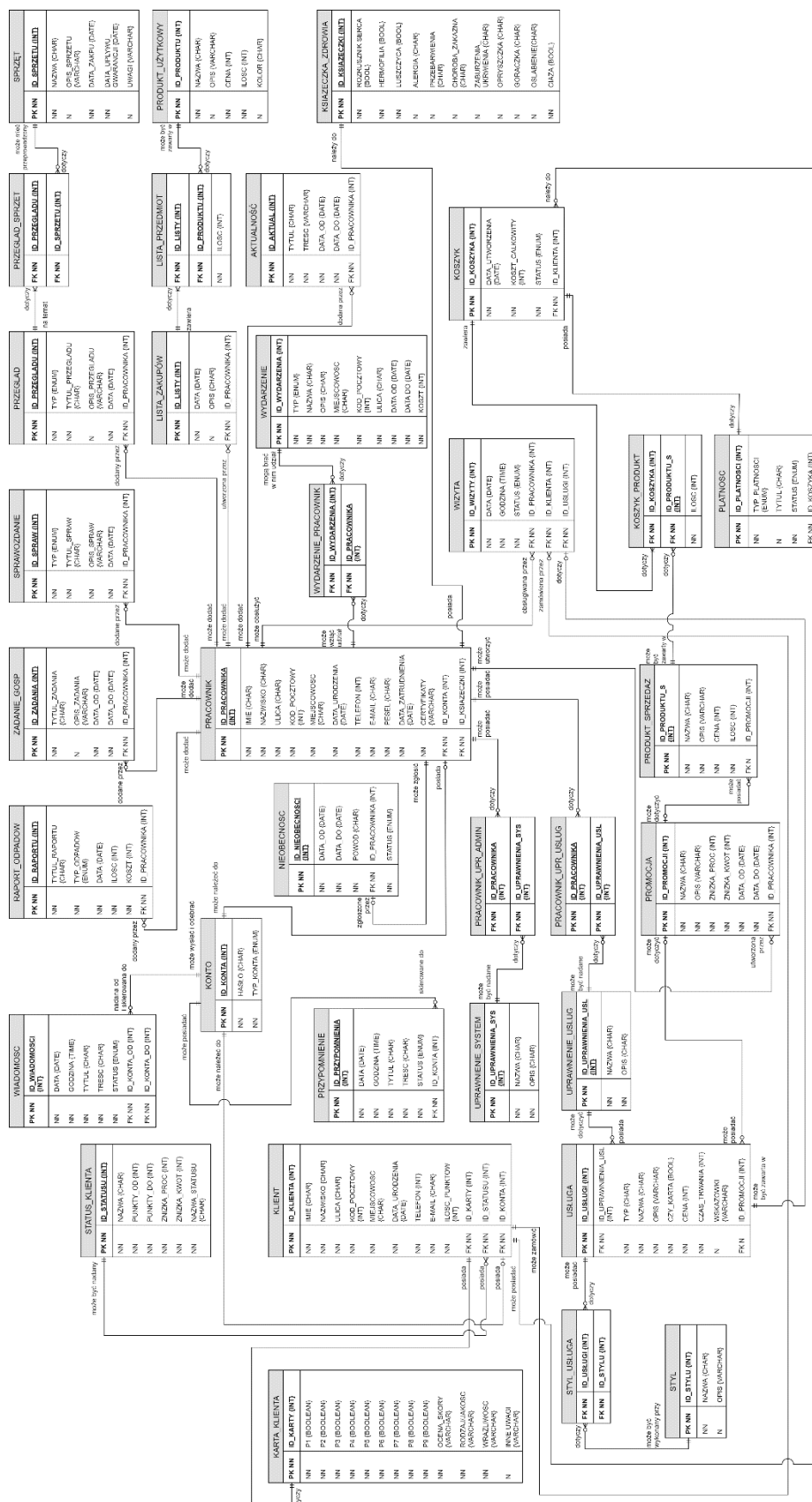
- encja „KONTO” może reprezentować zarówno konto pracownika, klienta jak i administratora. Rozróżniane jest to za pomocą ENUM przyjmujące wartości 'PRACOWNIK', 'ADMINISTRATOR' bądź 'KLIENT'.

- w encji „RAPORT_ODPADÓW” umieszczenie ENUM „TYP_ODPADÓW” umożliwił na scalenie encji reprezentujące różne typy odpadów: 150110 (opakowania, farby, lakiery w pojemnikach szklanych, opakowania zawierające pozostałości substancji niebezpiecznych), 150107 (zużyte opakowania szklane), 150102 (opakowania z tworzyw sztucznych - plastik), 160214 (zużyte urządzenia) oraz 80103 (odpady zakaźne - medyczne zawierające żywe kultury bakterii).

- dla związków M:M (wielu do wielu) utworzone zostały dodatkowe encje przechowywujące klucze główne encji zawierające taki związek pomiędzy sobą, na przykład encja „WYDARZENIE_PRACOWNIK” przechowywująca klucz obcy encji „PRACOWNIK” oraz klucz obcy encji „WYDARZENIE” dzięki czemu wiemy, który pracownik zapisał się na które wydarzenie (np.: pracownica „Agata” będzie mogła wziąć udział w szkoleniu odbywającym się w Poznaniu oraz w innym szkoleniu odbywającym się w Warszawie, a szkolenia te będą mogły posiadać wielu uczestników) bądź na przykład encja „KOSZYK_PRODUKT” informująca nas o tym, w jakiej ilości dany koszyk posiada dane przedmioty (oraz w jakiej ilości dane przedmioty należą do jakich koszyków).

W ostatnim kroku baza została znormalizowana do postaci 2NF, gdzie wymagało to częściowego funkcyjnego uzależnienia kolumn niekluczowych od wszystkich kluczy potencjalnych.

10.2 Schemat ERD po przeprowadzeniu normalizacji



10.3 Skrypt bazy danych

Poniżej przedstawiona została część skryptu odpowiedzialna za utworzenie w języku SQL bazy danych aplikacji wspomagającej pracę gabinetu kosmetycznego. Pełny skrypt wraz z skryptem tworzącym klucze obce znajduje się w oddzielnym pliku.

Zauważyć należy jedynie, iż nie ma możliwości utworzenia zmiennej typu BOOLEAN w systemie MySQL, dlatego wszystkie te wartości zastąpione zostały zmienną typu SHORTINT o długości 1, która przyjmuje wartości odpowiednio 1 bądź 0 dla wartości TRUE oraz FALSE.

We wszystkich tabelach zastosowane zostało również polecenie:
DEFAULT **characterSET** utf8mb4 **COLLATE** utf8mb4_unicode_ci;
umożliwiające obsługę polskich znaków w bazie danych jak i podczas wymiany danych między bazą danych a aplikacją.

10.3.1 Tabela klient

```
CREATE TABLE klient
(
    id_klienta      INT(8) PRIMARY KEY auto_increment,
    imie            CHAR(255) NOT NULL,
    nazwisko        CHAR(255) NOT NULL,
    ulica           CHAR(255) NOT NULL,
    kod_pocztowy    CHAR(10) NOT NULL,
    miejscowosc     CHAR(255) NOT NULL,
    data_urodzenia  DATE NOT NULL,
    telefon         INT(8) NOT NULL,
    e_mail          CHAR(255) NOT NULL,
    ilosc_punktow   INT(8) NOT NULL,
    id_karty        INT(8) NOT NULL,
    id_statusu      INT(8) NOT NULL,
    id_konta        INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.2 Tabela pracownik

```
CREATE TABLE pracownik
(
    id_pracownika   INT(8) PRIMARY KEY auto_increment,
    imie            CHAR(255) NOT NULL,
    nazwisko        CHAR(255) NOT NULL,
    ulica           CHAR(255) NOT NULL,
    kod_pocztowy    CHAR(10) NOT NULL,
    miejscowosc     CHAR(255) NOT NULL,
    data_urodzenia  DATE NOT NULL,
    telefon         INT(8) NOT NULL,
    e_mail          CHAR(255) NOT NULL,
    pesel           INT(8) NOT NULL,
    data_zatrudnienia DATE NOT NULL,
    certyfikaty     VARCHAR(5000) NULL,
    id_konta        INT(8) NOT NULL,
    id_ksiazeczki   INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.3 Tabela konto

```
CREATE TABLE konto
(
    id_konta INT(8) PRIMARY KEY auto_increment,
    haslo CHAR(255) NOT NULL,
    typ_konta ENUM('PRACOWNIK', 'ADMINISTRATOR', 'KLIENT')
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.4 Tabela wizyta

```
CREATE TABLE wizyta
(
    id_wizyty INT(8) PRIMARY KEY auto_increment,
    data DATE NOT NULL,
    godzina TIME NOT NULL,
    status ENUM('DO_ZATWIERDZENIA', 'ODRZUCONE', 'POTWIERDZONE',
    'OPLACONE') NOT
    NULL,
    id_pracownika INT(8) NOT NULL,
    id_klienta INT(8) NOT NULL,
    id_uslugi INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.5 Tabela raport_odpadow

```
CREATE TABLE raport_odpadow
(
    id_raportu INT(8) PRIMARY KEY auto_increment,
    tytul_raportu CHAR(255) NOT NULL,
    typ_odpadow ENUM('150110', '150107', '150102', '160214', '180103') NOT
    NULL,
    data DATE NOT NULL,
    ilos INT(8) NOT NULL,
    koszt INT(8) NOT NULL,
    id_pracownika INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.6 Tabela aktualność

```
CREATE TABLE aktualnosc
(
    id_aktualnosci INT(8) PRIMARY KEY auto_increment,
    tytul CHAR(255) NOT NULL,
    tresc VARCHAR(5000) NOT NULL,
    data_od DATE NOT NULL,
    data_do DATE NOT NULL,
    id_pracownika INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.7 Tabela wydarzenie

```
CREATE TABLE wydarzenie
(
    id_wydarzenia INT(8) PRIMARY KEY auto_increment,
    typ           ENUM('KONGRES', 'SZKOLENIE', 'TARGI_KOSMETYCZNE') NOT NULL,
    nazwa         CHAR(255) NOT NULL,
    opis          CHAR(255) NOT NULL,
    ulica         CHAR(255) NOT NULL,
    kod_pocztowy  CHAR(10) NOT NULL,
    miejscowosc   CHAR(255) NOT NULL,
    data_od       DATE NOT NULL,
    data_do       DATE NOT NULL,
    koszt         INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.8 Tabela usługa

```
CREATE TABLE usługa
(
    id_uslugi          INT(8) PRIMARY KEY auto_increment,
    id_uprawnienia_usl INT(8) NOT NULL,
    typ_uslugi         CHAR(255) NOT NULL,
    nazwa              CHAR(255) NOT NULL,
    opis               VARCHAR(5000) NOT NULL,
    czy_karta          BOOLEAN NOT NULL,
    cena               INT(8) NOT NULL,
    czas_trwania        INT(8) NOT NULL,
    wskazowki          VARCHAR(5000) NULL,
    id_promocji         INT(8) NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

10.3.9 Tabela promocja

```
CREATE TABLE promocja
(
    id_promocji      INT(8) PRIMARY KEY auto_increment,
    nazwa            CHAR(255) NOT NULL,
    opis             VARCHAR(5000) NOT NULL,
    zniżka_proc       INT(8) NULL,
    zniżka_kwot       INT(8) NULL,
    data_od           DATE NOT NULL,
    data_do           DATE NOT NULL,
    id_pracownika     INT(8) NOT NULL
)
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```


10.4 Procedury utworzone dla bazy danych

Poniżej przedstawiona została część skryptu odpowiedzialna za utworzenie w języku SQL procedur wykorzystywanych przez naszą aplikację. W chwili obecnej liczba utworzonych procedur osiągnęła liczbę 42, dlatego pełny skrypt procedur znajduje się w oddzielnym pliku.

10.4.1 Procedura tworząca konto klienta

```
DELIMITER //
CREATE PROCEDURE utworz_klienta(
    p_email          CHAR(255),
    p_haslo          CHAR(255),
    p_imie           CHAR(255),
    p_nazwisko       CHAR(255),
    p_ulica           CHAR(255),
    p_kod_pocztowy   INT(8),
    p_miejscowosc    CHAR(255),
    p_data_urodzenia DATE,
    p_telefon        INT(8))
BEGIN
    IF ((SELECT e_mail FROM klient WHERE e_mail = p_email) IS NULL) THEN
        INSERT INTO konto(haslo, typ_konta) VALUES (p_haslo, 'KLIENT');

        SET @id_konta := (SELECT MAX(id_konta) FROM konto WHERE haslo =
p_haslo);

        INSERT INTO klient(imie, nazwisko, ulica, kod_pocztowy, miejscowosc,
data_urodzenia, telefon, e_mail, ilosc_punktow, id_karty, id_statusu, id_konta)
VALUES (p_imie, p_nazwisko, p_ulica, p_kod_pocztowy, p_miejscowosc,
p_data_urodzenia, p_telefon, p_email, 0, 0, 1, @id_konta);
    END IF;
END //
```

10.4.2 Procedura tworząca konto pracownika

```
DELIMITER //
CREATE PROCEDURE utworz_pracownika(
    p_haslo          CHAR(255),
    p_imie           CHAR(255),
    p_nazwisko       CHAR(255),
    p_ulica           CHAR(255),
    p_kod_pocztowy   INT(8),
    p_miejscowosc    CHAR(255),
    p_data_urodzenia DATE,
    p_telefon        INT(8),
    p_e_mail         CHAR(255),
    p_pesel          INT(8),
    p_data_zatrudnienia DATE,
    p_certyfikaty     VARCHAR(5000),
    p_id_konta        INT(8),
    p_rozrusznik_serca BOOLEAN,
    p_hermofilia      BOOLEAN,
    p_luszczyca       BOOLEAN,
    p_alergia         CHAR(255),
    p_przebarwienie   CHAR(255),
    p_choroba_zakazna CHAR(255),
    p_zaburzenia_ukrwienia CHAR(255),
    p_opryszczka      CHAR(255),
    p_goraczka        CHAR(255),
    p_oslabienie      CHAR(255),
    p_ciaza           BOOLEAN)
)
BEGIN
    IF ((SELECT e_mail FROM pracownik WHERE e_mail = p_e_mail) IS NULL) THEN
        INSERT INTO konto(haslo, typ_konta) VALUES (p_haslo, 'PRACOWNIK');
        INSERT INTO ksiazeczka_zdrowia(rozrusznik_serca, hermofilia,
luszczyca, alergia, przebarwienie, choroba_zakazna, zaburzenia_ukrwienia,
opryszczka, goraczka, oslabienie, ciaza)
VALUES (p_rozrusznik_serca, p_hermofilia, p_luszczyca, p_alergia,
p_przebarwienie, p_choroba_zakazna, p_zaburzenia_ukrwienia, p_opryszczka,
p_goraczka, p_oslabienie, p_ciaza);

        SET @id_konta := (SELECT MAX(id_konta) FROM konto WHERE haslo =
p_haslo);
        SET @id_ksiazeczki := (SELECT MAX(id_ksiazeczki) FROM
ksiazeczka_zdrowia);
    END IF;
END //
```

10.4.3 Procedura tworząca usługę

```
DELIMITER //
CREATE PROCEDURE dodaj_usluge(
    P_ID_UPRAWNIENIA_USL INT(8),
    P_TYP_USLUGI CHAR(255),
    P_NAZWA CHAR(255),
    P_OPIS VARCHAR(5000),
    P_CZY_KARTA BOOLEAN,
    P_CENA INT(8),
    P_CZAS_TRWANIA INT(8),
    P_WSKAZOWKI VARCHAR(5000)
)
BEGIN
    IF ((SELECT MAX(ID_USLUGI) FROM uslugi) IS NULL) THEN
        SET @ID_USLUGI := 1;
    ELSE
        SET @ID_USLUGI := (SELECT MAX(ID_USLUGI) FROM uslugi) + 1;
    END IF;
    INSERT INTO uslugi(ID_USLUGI, ID_UPRAWNIENIA_USL, TYP_USLUGI, NAZWA, OPIS,
        CZY_KARTA, CENA, CZAS_TRWANIA, WSKAZOWKI) VALUES (@ID_USLUGI,
        P_ID_UPRAWNIENIA_USL, P_TYP_USLUGI, P_NAZWA, P_OPIS, P_CZY_KARTA, P_CENA,
        P_CZAS_TRWANIA, P_WSKAZOWKI);
END //
```

10.4.4 Procedura tworząca wydarzenie (szkolenie / kongres / targi)

```
DELIMITER //
CREATE PROCEDURE dodaj_wydarzenie(
    P_TYP ENUM('KONGRES', 'SZKOLENIE', 'TARGI_KOSMETYCZNE'),
    P_NAZWA CHAR(255),
    P_OPIS CHAR(255),
    P_MIEJSCOWOSC CHAR(255),
    P_KOD_POCZTOWY INT(5),
    P_ULICA CHAR(255),
    P_DATA_OD DATE,
    P_DATA_DO DATE,
    P_KOSZ INT(8),
    P_ID_PRACOWNIKA INT(8)
)
BEGIN
    INSERT INTO wydarzenie(TYP, NAZWA, OPIS, MIEJSCOWOSC, KOD_POCZTOWY, ULICA,
        DATA_OD, DATA_DO, KOSZ) VALUES (P_TYP, P_NAZWA, P_OPIS, P_MIEJSCOWOSC,
        P_KOD_POCZTOWY, P_ULICA, P_DATA_OD, P_DATA_DO, P_KOSZ);
    SET @ID_WYDARZENIA := (SELECT ID_WYDARZENIA FROM wydarzenie WHERE NAZWA =
        P_NAZWA);
    INSERT INTO wydarzenie_pracownik(ID_WYDARZENIA, ID_PRACOWNIKA) VALUES
        (@ID_WYDARZENIA, P_ID_PRACOWNIKA);
END //
```

10.4.5 Procedura tworząca przegląd sprzętu

```
DELIMITER //
CREATE PROCEDURE przeglad(
    P_TYTUL_PRZEGladu CHAR(255),
    P_OPIS_PRZEGladu VARCHAR(5000),
    P_DATA DATE,
    P_ID_PRACOWNIKA INT(8),
    P_ID_SPRZETU INT(8)
)
BEGIN
    INSERT INTO przeglad(TYTUL_PRZEGladu, OPIS_PRZEGladu, DATA, ID_PRACOWNIKA)
        VALUES (P_TYTUL_PRZEGladu, P_OPIS_PRZEGladu, P_DATA, P_ID_PRACOWNIKA);
    SET @ID_PRZEGladu := (SELECT ID_PRZEGladu FROM przeglad WHERE ID_PRZEGladu
        NOT IN(SELECT ID_PRZEGladu FROM PRZEGlad_SPRZET));
    INSERT INTO przeglad_sprzet(ID_PRZEGladu, ID_SPRZETU) VALUES
        (@ID_PRZEGladu, P_ID_SPRZETU);
END //
```

10.4.6 Procedura tworząca nieobecność pracownika

```
DELIMITER //
CREATE PROCEDURE zglos_nieobecnosc(
    P_DATA_OD DATE,
    P_DATA_DO DATE,
    P_POWOD CHAR(255),
    P_ID_PRACOWNIKA INT(8),
    P_STATUS ENUM('NIEPOTWIERDZONE', 'POTWIERDZONE')
)
BEGIN
    INSERT INTO nieobecnosc(DATA_OD, DATA_DO, POWOD, ID_PRACOWNIKA, STATUS)
VALUES (P_DATA_OD, P_DATA_DO, P_POWOD, P_ID_PRACOWNIKA, P_STATUS);
END //
```

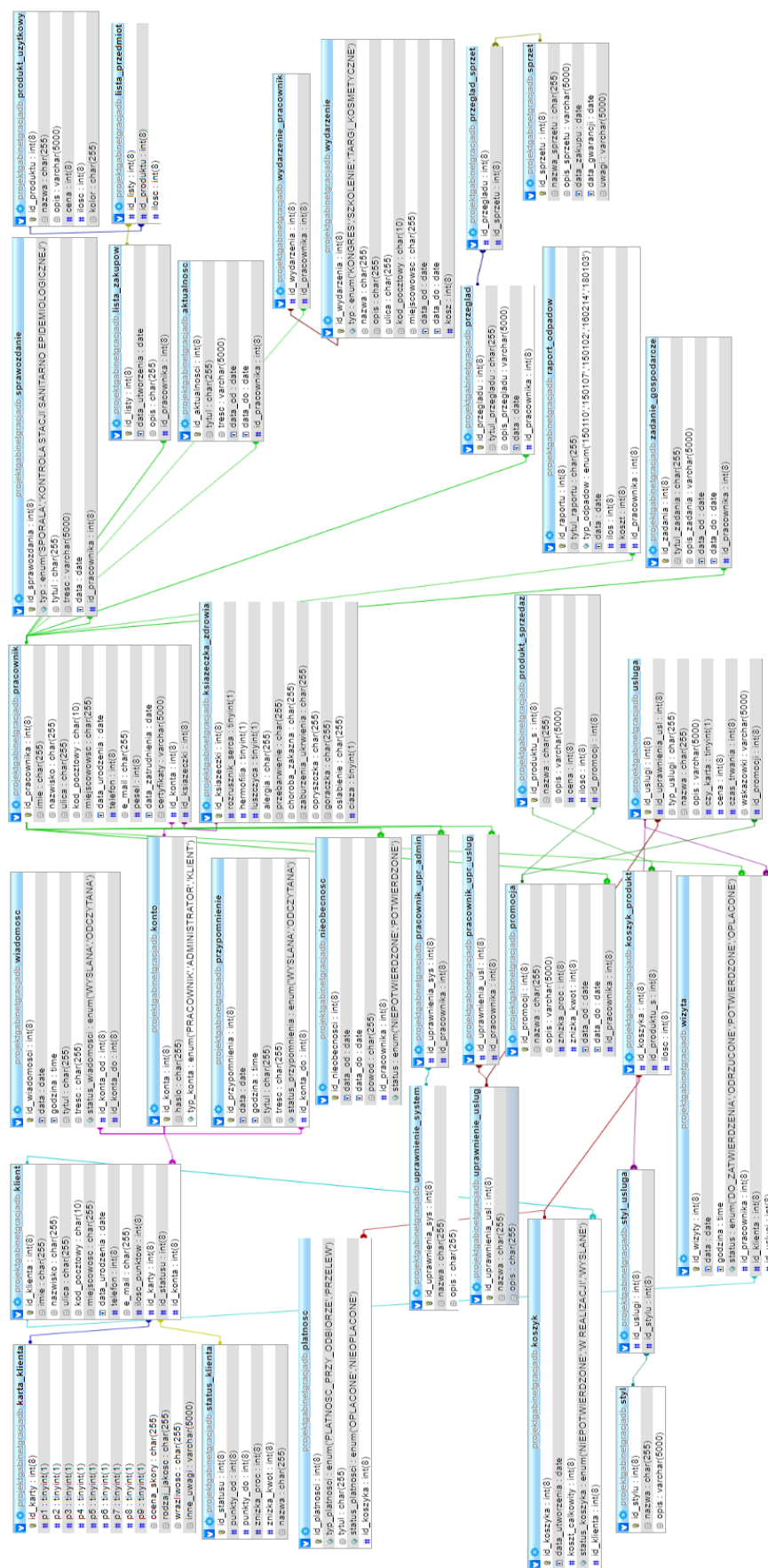
10.4.7 Procedura tworząca wiadomość od użytkownika do użytkownika

```
DELIMITER //
CREATE PROCEDURE wiadomosc(
    P_DATA DATE,
    P_GODZINA TIME,
    P_TYTUL CHAR(255),
    P_TRESC CHAR(255),
    P_STATUS ENUM('WYSLANA', 'ODCZYTANA'),
    P_ID_KONTA_OD INT(8),
    P_ID_KONTA_DO INT(8)
)
BEGIN
    INSERT INTO wiadomosc(DATA, GODZINA, TYTUL, TRESC, STATUS_WIADOMOSCI,
ID_KONTA_OD, ID_KONTA_DO) VALUES (P_DATA, P_GODZINA, P_TYTUL, P_TRESC,
P_STATUS, P_ID_KONTA_OD, P_ID_KONTA_DO);
END //
```

10.4.8 Procedura dodająca produkt do danej promocji

```
DELIMITER //
CREATE PROCEDURE promocja_produkt(
    P_ID_PROMOCJI INT(8),
    P_ID_PRODUKTU_S INT(8)
)
BEGIN
    UPDATE produkt_sprzedaz SET ID_PROMOCJI = P_ID_PROMOCJI WHERE ID_PRODUKTU
= P_ID_PRODUKTU;
END //
```

10.5 Schemat bazy danych w systemie



12 Przykładowe zapytania do systemu

12.1 Pobierz certyfikaty pracownika

```
SELECT certyfikaty FROM pracownik WHERE id_pracownika = 11;
```

W przykładowym zapytaniu pobierana jest zawartość zmiennej tekstowej certyfikaty, przypisana konkretnemu pracownikowi – w tym przypadku temu, którego ID wynosi 11.

12.2 Wygeneruj roczny bilans utylizacji

```
SELECT SUM(koszt*ilos) AS bilans FROM raport_odpadow WHERE data >= date(2019-01-01) AND data <= date(2019-12-31);
```

Zapytanie generuje bilans utylizacji w oparciu o raporty odpadów za rok 2019. W zapytaniu użyto formatu daty YYYY-MM-DD, który jest domyślny dla języka MySQL oraz rzutowania na typ DATE.

12.3 Pobierz dane dotyczące listy zakupowej

```
SELECT lz.*, pu.*, lp.ilosc AS ilosc_na_liscie FROM lista_zakupow lz JOIN lista_przedmiot lp ON lz.id_listy = lp.id_listy JOIN produkt_uzytkowy pu ON pu.id_produktu = lp.id_produktu WHERE lz.id_listy = 4;
```

Zapytanie w wyniku zwraca pozycje umieszczone na liście zakupowej gabinetu w oparciu o podane id listy, którą chcemy sprawdzić. Wypisywana jest liczba zamówionych przedmiotów jak również informacje o nich, zawarte w tabeli produkt_uzytkowy, za pośrednictwem złączenia JOIN.

12.4 Wypisz dane przelewu za koszyk danego klienta

```
SELECT p.* FROM koszyk k JOIN platnosc p ON k.id_koszyka = p.id_koszyka WHERE k.id_klienta = 5;
```

Zwracane są dane dotyczące płatności przelewem za wartość przypisanego klientowi koszyka w oparciu o numer ID klienta. Identyfikacja danych następuje za pośrednictwem złączenia JOIN.

12.5 Wypisz zadania gospodarcze przypisane do danego pracownika

```
SELECT * FROM zadanie_gospodarcze WHERE id_pracownika = 8;
```

Jedno z prostszych zapytań, wypisuje za jakie zadania gospodarcze odpowiedzialny jest pracownik o danym ID, w tym przypadku ID = 8.

12.6 Wypisz nieobecności pracownika

```
SELECT nieobecnos.*, pracownik.imie, pracownik.nazwisko FROM nieobecnos JOIN
pracownik ON nieobecnos.id_pracownika = pracownik.id_pracownika WHERE
nieobecnos.id_pracownika = 12;
```

Zapytanie zwraca wszystkie dane kiedykolwiek zgłoszonych nieobecności pracownika o podanym ID, tu 12, jak również imię i nazwisko pracownika, którego dotyczą nieobecności. Aby uzyskać ostatnie dwa, wykonano złączenie JOIN.

12.7 Pobierz wskazówki przygotowania do zabiegu

```
SELECT wskazowki FROM uslugi WHERE id_uslugi = 20;
```

Proste zapytanie, które wypisuje wskazówki dotyczące usługi o zadanym ID, wykorzystywane kiedy Klient zapisuje się na wykonanie usługi. W chwili zapisania się, system wyświetla wskazówki dotyczące przygotowania przed wizytą w chwili umówienia się na nią.

12.8 Pobierz dane o sprzęcie

```
SELECT * FROM sprzet WHERE ID_SPRZETU = 24;
```

Wypisywane są dane dotyczące konkretnego urządzenia będącego w wyposażeniu gabinetu, w oparciu o jego numer ID.

12.9 Wyświetl aktualności na dzisiaj

```
SELECT * FROM aktualnosc WHERE CURRENT_DATE > aktualnosc.data_od AND
CURRENT_DATE < aktualnosc.data_do;
```

Zapytanie wykorzystuje metodę CURRENT_DATE aby wykonywać porównania z datami przypisanymi aktualnościom i wybrać takie, które powinny być wyświetlane w bieżącym dniu. System powinien wysłać takie zapytanie raz dziennie, w celu odświeżania zestawu wyświetlonych aktualności.

12.10 Wypisz sprawozdania z przedziału czasowego

```
SELECT * FROM sprawozdanie WHERE data >= date(2018-06-09) AND data <=
date(2018-06-30);
```

Zapytanie służy do wypisywania wszystkich wpisów w tabeli sprawozdanie, które zostały wprowadzone do bazy danych w podanym przedziale czasowym. W tym przypadku wyszukiwane są sprawozdania dodane między 9 a 30 dniem czerwca.

13 Przykłady optymalizacji bazy danych

W poniższym rozdziale przedstawione zostały indeksy nałożone na odpowiednie kolumny w tabelach, które będą usprawniały najczęściej wykonywane zapytania wraz z tymi, które są jednymi z najbardziej złożonych obliczeniowo.

13.1 Index nazwisk klientów

```
CREATE INDEX idx_klient_nazw ON klient(nazwisko);
```

13.2 Index imion pracowników

```
CREATE INDEX idx_prac_imie ON pracownik(imie);
```

13.3 Index dat raportów

```
CREATE INDEX idx_raport_data ON raport_odpadow(data);
```

13.4 Index dat sprawozdań

```
CREATE INDEX idx_sprawozdanie_data ON sprawozdanie(data);
```

13.5 Kompozytowy index dat aktualności

```
CREATE INDEX idx_aktualnosc_data ON aktualnosc(data_od, data_do);
```

13.6 Kompozytowy index dat wydarzeń

```
CREATE INDEX idx_wydarzenie_data ON wydarzenie(data_od, data_do);
```

13.7 Kompozytowy index dat nieobecności

```
CREATE INDEX idx_nieobecnosc_data ON nieobecnosc(data_od, data_do);
```

14 Technologia i narzędzia programistyczne

14.1 Uzasadnienie wyboru technologii i narzędzia programistycznego

Podczas podejmowania decyzji dotyczącego wyboru odpowiednich technologii oraz narzędzi programistycznych propozycje wskazywały na Oracle, MySQL oraz ObjectDB dla baz danych, język obiektowy Java dla back-endu, JavaFX oraz HTML + CSS dla front-endu wraz z wykorzystywaniem technologii takich jak Bootstrap, Maven, Apache, Tomcat oraz Jenkins.

Ze względu na fakt, iż aplikacja, która miała zostać stworzona na podstawie tego projektu miała być webowa (uruchamiana przez przeglądarkę), stwierdziliśmy jednomyślnie, że najlepiej byłoby wykorzystać te technologie i narzędzia, które już znamy.

Z tego powodu wybór padł na Javę, HTML wraz z CSSem przy wykorzystaniu frameworku Bootstrap oraz na MySQL wraz z narzędziem XAMPP. Na decyzję wpłynął również fakt, że te technologie posiadają łatwy dostęp do swoich narzędzi oraz prosty sposób ich obsługi i wykorzystywania. Również dialekt języka SQL jest bardziej przystępny w systemie MySQL – optymalizacja wydajności naszej bazy danych została jednak przeprowadzona w programie SQLDeveloper firmy Oracle ze względu na ograniczone możliwości MySQL'a.

15 Planowanie, kontrola oraz analiza przedwdrożeniowa

Niniejszy rozdział przedstawia strukturę firmy klienta, dla którego wykonywana jest aplikacja w ramach projektu, analizę jego środowiska oraz analizę przedwdrożeniową wykonywaną przed wdrożeniem programu u klienta.

15.1 Dziedzina problemu

Rozwój informatyki w dzisiejszych czasach umożliwił wprowadzenie udogodnień i ulepszeń sposobu działania wielu dziedzin, jedną z których jest dziedzina usługowo-sprzedażowa. W celu polepszenia sposobu działania wielu podmiotów tej dziedziny wprowadza się system, który umożliwia pośredniczenie pomiędzy klientami usług a ich usługodawcami oraz pomiędzy właścicielem firmy a jej pracownikami.

System przedstawiony w dokumencie zapewnia korzyści wszystkim użytkownikom – klient gabinetu może złożyć prośbę o wizytę w gabinecie wybierając daną usługę firmy, pracownik gabinetu może taką prośbę potwierdzić bądź odrzucić, natomiast właściciel gabinetu może w wygodny sposób wprowadzać nową ofertę, aktualizacje, promocje czy raporty oraz mieć podgląd na wykonane w gabinecie usługi.

15.2 Interesariusze

Interesariusze naszego projektu obejmują właściciela Gabinetu Kosmetycznego „Gracja” Anna Wąsik, pracowników oraz klientów gabinetu – zarówno zarejestrowanych w systemie jak i tych nieposiadających konta w systemie.

Wdrożenie systemu pozwoli na skuteczniejszą możliwość promowania gabinetu kosmetycznego, łatwiejszego kontaktu pomiędzy pracownikami gabinetu a klientem oraz przyspieszy realizację zadań pracowników jak i właściciela co powinno pozytywnie wpłynąć na komfort wszystkich użytkowników.

15.2.1 Interesariusze wewnętrzni

- *Twórcy* – osoby tworzące system, odpowiedzialne za jego kod oraz kontakt z firmą gabinetu kosmetycznego.
- *Właściciel (Administrator)* – osoba odpowiedzialna za bezpośredni kontakt z twórcami aplikacji oraz osoba korzystająca z aplikacji, wprowadza, uaktualnia bądź usuwa dane z aplikacji.
- *Pracownicy gabinetu* – osoby korzystające z aplikacji i zawartych w niej danych, posiadają ograniczone możliwości właściciela.

15.2.1 Interesariusze zewnętrzni

- *Klienci* – osoby korzystające z aplikacji mogące w łatwy i szybki sposób przejrzeć ofertę gabinetu, dokonać zakupu produktu kosmetycznego bądź rezerwacji terminu wizyty.

15.3 Analiza ryzyka projektu

STRATY FINANSOWE	<ul style="list-style-type: none">• Błędne wyliczenia dotyczące finansowania projektu• Błędne oszacowanie terminu oddania serwisu skutkujące koniecznością zatrudnienia dodatkowych pracowników• Niezadowalające testy akceptacyjne• Konieczność skorzystania z zabezpieczeń dostarczanych przez inną firmę w przypadku braku pewności bezpieczeństwa danych systemu
STRATY WIZERUNKOWE	<ul style="list-style-type: none">• Błędna interpretacja funkcjonalności systemu dostarczanych przez klienta• Wystąpienie nieporozumienia podczas rozmów z klientem• Wystąpienie problemów podczas współpracy z klientem• Brak akceptacji ze strony klienta dotyczącej finalnego systemu
INNE	<ul style="list-style-type: none">• Niedokładność podczas tworzenia specyfikacji i wymagań systemów• Niepoprawna strategia finansowa• Awaria systemu i urządzeń firmy• Infekcja systemu przez osobę zewnętrzną, na przykład atak hakerski

15.4 Ograniczenia systemu

Aby móc korzystać w pełni z systemu, serwer na którym implementowany byłby system musi posiadać stały dostęp do internetu, serwer musi obsługiwać język Java, HTML, CSS, JavaScript oraz bazę danych utworzoną w języku SQL, posiadać co najmniej 4 GB pamięci RAM wraz z 4 rdzeniowym procesorem w celu optymalnego zarządzania systemem oraz umożliwienia płynnego działania aplikacji.

Użytkownik-klient, użytkownik-pracownik oraz administrator w celu skorzystania z systemu muszą posiadać jedno z dwóch urządzeń:

- *Urządzenie mobilne* posiadający system Android, iOS bądź Windows Mobile oraz kompatybilną aplikację przeglądarkową, na przykład Google Chrome lub Opera.
- *Komputer stacjonarny* posiadający dowolny system, na przykład Windows 7 bądź MacOS posiadający kompatybilną aplikację przeglądarkową, na przykład Google Chrome, Internet Explorer bądź Opera.

15.5 Architektura systemu

System wspomagający pracę gabinetu kosmetycznego składa się z aplikacji webowej uruchamianej po stronie serwera, w której zalogowany użytkownik może korzystać z funkcjonalności w zależności od typu konta, na które jest zalogowany – pracownik / klient / administrator. Back-end aplikacji został wykonany w oparciu o język Java – zaś cała komunikacja pomiędzy bazą danych i aplikacją odbywa się również za pośrednictwem Java.SQL. Frontend aplikacji natomiast został wykonany w oparciu o HTML oraz CSS z wykorzystaniem Bootstrap'a.

Klient, pracownik bądź administrator komunikuje się z systemem przez przeglądarkę, serwer hostujący aplikację odbiera te żądania, analizuje je, wykonuje, pobiera dane z bazy danych i zwraca wynik wyświetlany w oknie przeglądarki użytkownika.

15.6 Kontekst funkcjonowania

Głównym użytkownikiem systemu będzie właściciel gabinetu kosmetycznego – administrator. Będzie on posiadał możliwość tworzenia kont pracowników, zarządzania nimi, możliwość przeglądania zgłoszeń wizyt, możliwość generowania raportów, możliwość podglądu oraz zarządzania aktualnościami, ofertami usług gabinetu, ofertami sprzedażowymi, promocjami i wydarzeniami typu szkolenie, targi kosmetyczne bądź kongresy. Będzie decydował, który pracownik odpowiedzialny jest za wybraną część obowiązków gospodarczych. Dzięki temu zwiększy się jego tempo pracy.

Pracownik ma możliwość podglądu planowanych wizyt w danym dniu, możliwość ich zatwierdzenia bądź odrzucenia jak i również możliwość ich tworzenia dla niezarejestrowanych w systemie klientów, możliwość podglądu nadchodzącej wypłaty, możliwość prowadzenia książeczki zdrowia, możliwość zarządzania aktualnościami gabinetu, możliwość składania zlecenia zakupu danego produktu oraz możliwość obsługi zamówień online w sklepie internetowym gabinetu.

Klient korzystający z systemu będzie miał możliwość przeglądania oferty gabinetu, aktualności oraz oferty sklepu online, możliwość konsultacji online drogą mailową bądź przez aplikację Messenger, możliwość tworzenia wizyty oraz przeglądania bonusów – system lojalnościowy.

System będzie przechowywał historię wszystkich wizyt oraz sprzedaży w systemie, spis tworzonych raportów oraz aktualności wyświetlane w dniach określanych przez administratora bądź pracownika gabinetu jak i pozostałe dane wprowadzane do bazy danych.

15.7 Analiza struktury organizacji

Organizacja zlecająca wykonanie systemu posiada kilka typów użytkowników: administrator, pracownik oraz klient. Każdy użytkownik posiada inne uprawnienia i funkcje możliwe do wykonania.

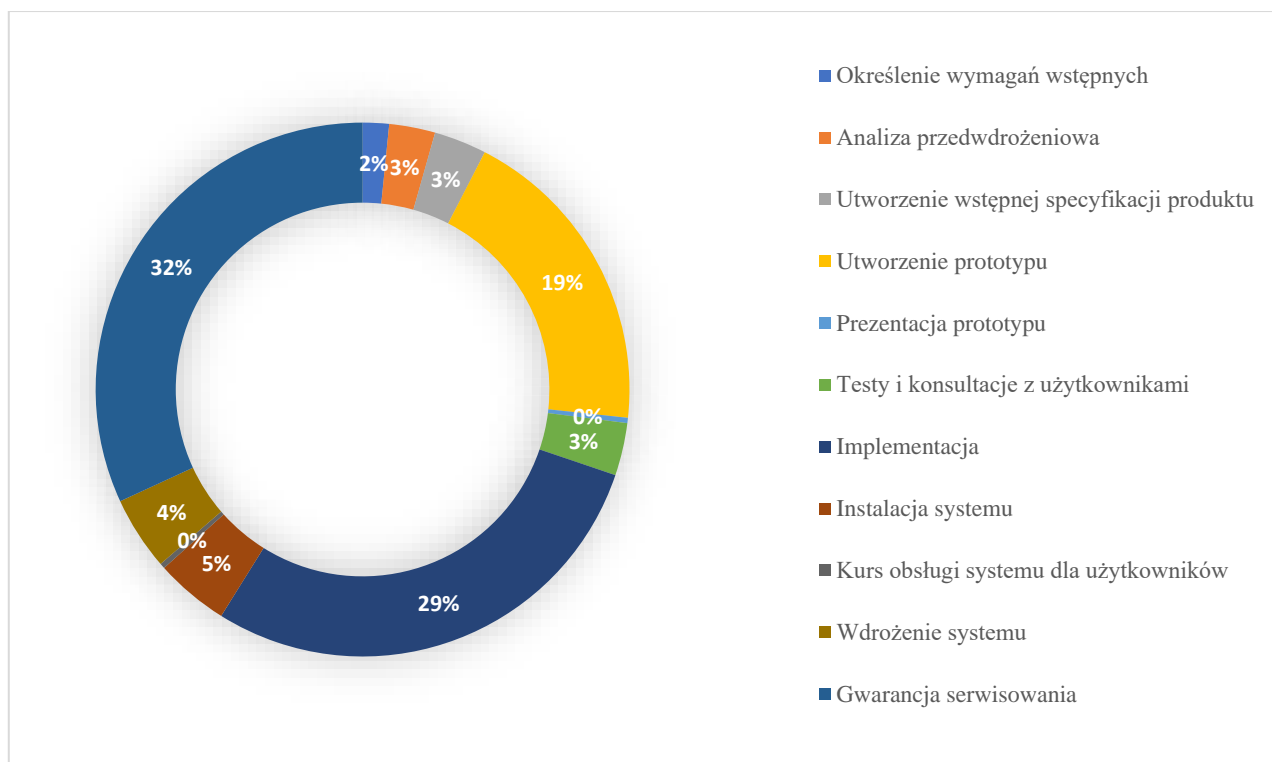
15.8 Analiza zaplecza informatycznego

Po analizie zaplecza informatycznego otrzymujemy następujący wynik – każdy z użytkowników posiada komputer pracujący na systemie Windows 10 wraz z przeglądarką Google Chrome. Nie ma więc potrzeby zakupu ani ulepszenia posiadanych komputerów by móc w pełni korzystać z tworzonego systemu.

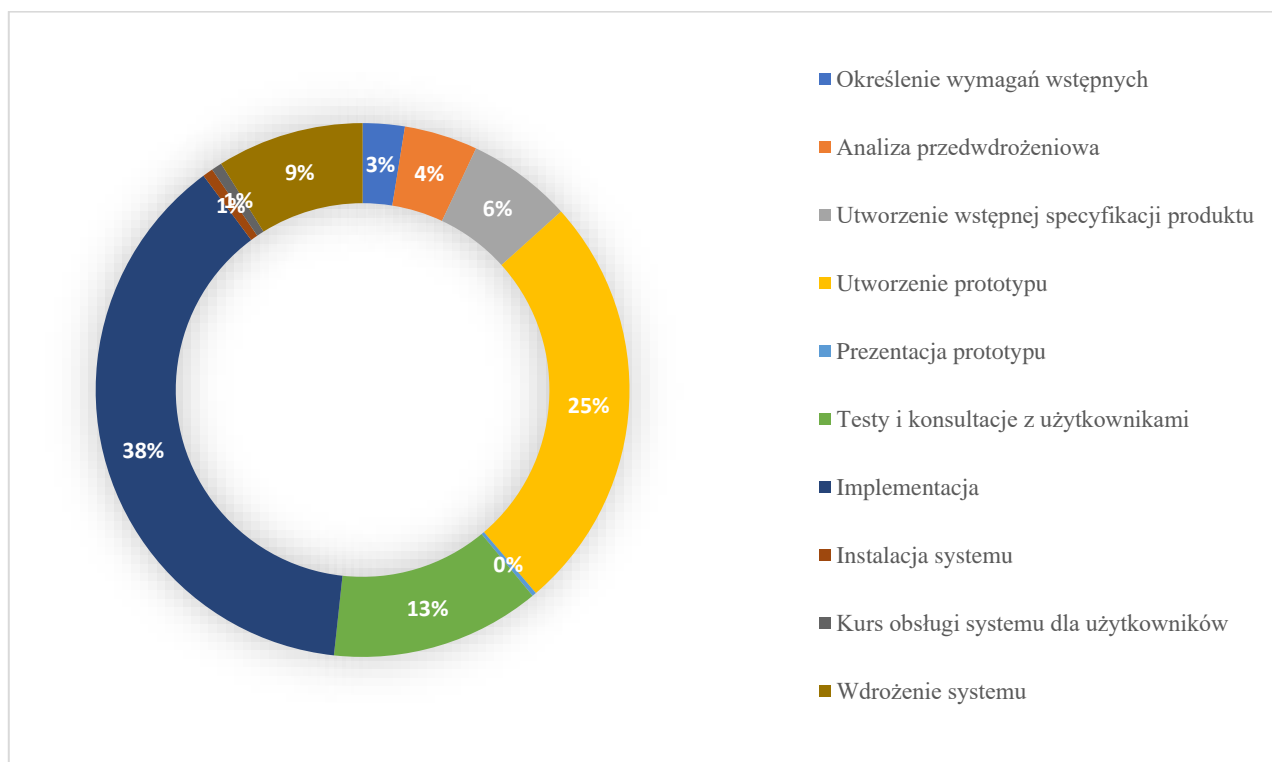
Właściciel firmy będzie musiał jedynie zakupić urządzenie UPS oraz serwer dla systemu – wybór modelu oraz firmy urządzeń pozostawiony został do decyzji klienta.

15.9 Harmonogram, koszt oraz szacowany czas wykonania prac

ETAP PROJEKTU	IŁOŚĆ GODZIN	STAWKA	KOSZT
Określenie wymagań wstępnych	20	25,00 zł	500,00 zł
Analiza przedwdrożeniowa	35	25,00 zł	875,00 zł
Utworzenie wstępnej specyfikacji produktu	50	20,00 zł	1 000,00 zł
Utworzenie prototypu	200	30,00 zł	6 000,00 zł
Prezentacja prototypu	2	50,00 zł	100,00 zł
Testy i konsultacje z użytkownikami	100	10,00 zł	1 000,00 zł
Implementacja	300	30,00 zł	9 000,00 zł
Instalacja systemu	5	15,00 zł	75,00 zł
Kurs obsługi systemu dla użytkowników	5	20,00 zł	100,00 zł
Wdrożenie systemu	70	20,00 zł	1 400,00 zł
Gwarancja serwisowania	-	10 000,00 zł	10 000,00 zł
SUMA	787	-	30 050,00 zł



Wykres przedstawiający jaki udział w kosztach pieniężnych mają poszczególne etapy pracy



Wykres przedstawiający jaki udział w kosztach czasu mają poszczególne etapy pracy

16 Słownik

- *Użytkownik* – każdy klient, pracownik bądź administrator posiadający konto w systemie.
- *Login, hasło* – dane wykorzystywane do uruchomienia systemu dla poszczególnych użytkowników.
- *ID klienta, ID pracownika* – identyfikator rozróżniający użytkowników systemu.
- *Użytkownik-klient* – osoba korzystająca z aplikacji z konta klienta gabinetu.
- *Użytkownik-pracownik* – osoba korzystająca z aplikacji z konta pracownika gabinetu.
- *Administrator* – osoba korzystająca z aplikacji z konta administratora gabinetu, posiada wszystkie uprawnienia.
- *Karta klienta* – karta informacyjna danego klienta, informuje o tym, czy dany klient posiada rozrusznik serca, alergie, czy ma zaburzenia ukrwienia, czy jest w ciąży itd.
- *P1, P2, P3, P4, P5, P6, P7, P8, P9* – zbiór poszczególnych pytań, określanych przez administratora systemu, zadawanych klientowi w celu założenia karty klienta.
- *Status klienta* – jest to etykieta nadawana każdemu klientowi gabinetu, określa, ile punktów z systemu lojalnościowego dany klient posiada oraz określa wartość zniżek przyznawanych danemu klientowi.
- *Nieobecność* – dana prośba o przyznanie zwolnienia dodawana przez pracownika a potwierdzana bądź odrzucana przez administratora systemu. Dzięki nieobecnościom przyznawane są przerwy związane z dniami świątecznymi, urlopy bądź zwolnienia lekarskie.
- *Wizyta* – zbiór informacji dotyczący tego, który klient w danym dniu skorzystał z danej usługi wykonanej przez danego pracownika oraz pobrana została dana kwota.
- *Usługi* – zbiór usług wykonywanych przez pracowników gabinetu.
- *Produkty użytkowe* – produkty wykorzystywane przez pracowników w gabinecie kosmetycznym w celu wykonania danej usługi.
- *Produkty sprzedażowe* – produkty oferowane w ofercie gabinetu kosmetycznego możliwe do zakupienia przez klientów gabinetu.
- *Oferta sprzedażowa* – zbiór produktów możliwych do zakupienia w gabinecie.
- *Oferta usługowa* – zbiór usług dostarczanych przez gabinet.
- *Książeczka zdrowia* – książeczka zdrowia danego pracownika informująca o tym, czy dany pracownik np.: posiada rozrusznik serca, ma alergię bądź jest w ciąży.
- *Sprawozdanie* – jest to dokument tworzony przez administratora, może dotyczyć sporali bądź kontroli stacji sanitarno epidemiologicznych dokonywanych w gabinecie.
- *Raport odpadów* – jest to dokument tworzony przez administratora, dotyczy wybranego typu odpadów generowanych przez gabinet.
- *Wydarzenie* – reprezentuje kongres, szkolenie bądź targi kosmetyczne na które zapisać może się dany pracownik gabinetu kosmetycznego.
- *Aktualność* – reprezentuje informację udostępnianą na wybranej stronie systemu prezentowaną klientom.
- *Zadanie gospodarcze* – prezentuje obowiązek danego pracownika dot. pilnowania porządku wybranego pomieszczenia gospodarczego.

17 Wnioski

W ramach przedstawionego powyżej projektu naszym zadaniem było przygotowanie analizy struktury firmy, przeprowadzenie analizy przedwdrożeńowej oraz tworzenia projektu, przygotowanie bazy danych współpracującej z utworzoną aplikacją wspomagającą pracę gabinetu kosmetycznego.

W pierwszej kolejności według zaleceń prowadzącego przedmiot stworzyliśmy projekt bazy danych, wypisaliśmy wszystkie wymagania jakie powinna ona spełniać oraz jej funkcjonalności i na podstawie tego stworzyliśmy w pierwszej kolejności schemat ERD. W następnym kroku przeprowadziliśmy normalizację naszej bazy danych tworząc kolejny schemat na podstawie którego omówiliśmy, w jakiś sposób wykonywane będą wszystkie nasze funkcjonalności za pomocą tej bazy danych.

Ze względu jednak na to, jak rozbudowane są funkcjonalności naszej aplikacji, nie jest możliwe jej pełne utworzenie i wdrożenie w tak krótkim czasie, dlatego postanowiliśmy grupowo, iż skupimy się na jej najważniejszych aspektach. Niemniej, udało nam się w pełni ukończyć naszą bazę danych, wprowadzić wszystkie procedury, które są wymagane podczas spełniania poszczególnych funkcjonalności oraz pomyślnie udało nam się przeprowadzić optymalizację jej wydajności.

W trakcie pracy nad projektem zauważyliśmy, że niektóre elementy nie powinny być usuwane z bazy danych, by nie tworzyć „czarnych dziur” w systemie aplikacji. Przykładem może być taka sytuacja: pracownik gabinetu chce się zwolnić, a tym samym – usunąć swoje konto. Niemniej, w przypadku usunięcia jego konta, z systemu znikną wszystkie usługi, jakie wykonał, a tym samym podczas generowania raportu finansowego, na przykład rocznego, dane nie będą prawidłowe. Dlatego niektóre procedury posiadające nazwę „usuń” tak naprawdę nie usuwają danych z naszej bazy danych, a jedynie zerują niewykorzystywane/niepotrzebne dane pozostawiając tylko te konieczne, takie jak właśnie imię i nazwisko pracownika kiedyś pracującego w gabinecie, dzięki czemu nadal będzie możliwy podgląd wykonanych w gabinecie usług sprzed na przykład 2 lat jeśli zaszłaby taka potrzeba.

Również podczas dalszych prac nad oprogramowaniem wprowadzane były jedynie niewielkie zmiany w bazie danych, takie jak dodawanie jednej lub dwóch dodatkowych atrybutów w niektórych encjach bazy dla dopełnienia danych bądź nieznaczna edycja procedur, na przykład chcąc sprawdzić usługi wykonywane przez danego pracownika procedura zwracała w pierwszej kolejności informacje o usłudze oraz ID pracownika. Niemniej użytkownik korzystający z aplikacji miałby problem, by stwierdzić, który dokładnie jest to pracownik jedynie posiadając jego ID. Dlatego procedura została zmodyfikowana, by zwracać również imię i nazwisko pracownika na podstawie tego ID przez polecenie JOIN ON.

Podsumowując, nasza baza danych składa się z ponad 30 tabel oraz z ponad 70 procedur umożliwiających wykorzystywanie bazy danych bez konieczności tworzenia w kodzie Javy poleceń SQL, a jedynie wywoływanie ich. Dzięki automatycznej inkrementacji ID we wszystkich tabelach nie zajdzie sytuacja, gdzie dwa obiekty bazy danych posiadają takie samo ID, a dzięki doświadczeniu oraz umiejętności wykorzystywania narzędzi, jakie nabyliśmy wszyscy podczas pracy nad tym projektem, bez problemu możemy rozszerzać funkcjonalności aplikacji w przypadku gdyby zaszła taka potrzeba, przykładem czego by było np. utworzenie w gabinecie usług fryzjerskich.