

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Komputerowe systemy zarządzania
procesami produkcyjnymi (ARS)

PRACA DYPLOMOWA
INŻYNIERSKA

System informatyczny wspomagający planowanie
spotkań biznesowych

IT system supporting planning business meetings

AUTOR:

Małgorzata Wojnarowska

PROWADZĄCY PRACĘ:

dr inż. Jarosław Pempera

OCENA PRACY:

Spis treści

1	Wstęp	3
2	Cel i zakres pracy	4
3	Wymagania projektowe	5
3.1	Opis działania systemu	5
3.2	Wymagania funkcjonalne	5
3.3	Wymagania niefunkcjonalne	6
4	Projekt systemu	8
4.1	Projekt relacyjnej bazy danych	8
4.2	Interfejs użytkownika	11
4.3	Diagram projektowy	16
5	Implementacja systemu baz danych	18
5.1	Tworzenie tabel	18
5.2	Definiowanie indeksów oraz ograniczeń	19
5.3	Nadawanie uprawnień użytkownikom	21
5.4	Testowanie bazy danych	21
6	Implementowanie i testowanie aplikacji	23
6.1	Instalacja i konfigurowanie systemu	23
6.2	Instrukcja użytkowania aplikacji	23
6.3	Użycie funkcji systemu	25
6.3.1	Zabezpieczenie stron dla niezalogowanych użytkowników	25
6.3.2	Tworzenie spotkań i klientów	25
6.3.3	Edycja/usuwanie notatki	26
6.3.4	Rejestracja użytkownika	27
6.3.5	Przeglądanie zaplanowanych spotkań oraz notatek	28
6.4	Wybrane funkcje - opis programowania	29
6.4.1	Implementacja dostępu do bazy danych	29

SPIS TREŚCI	2
6.4.2 Implementacja wybranych funkcjonalności systemu	30
6.4.3 Implementacja mechanizmów bezpieczeństwa	32
7 Podsumowanie	37
Bibliografia	38

Rozdział 1

Wstęp

Niektórym ludziom wydaje się, że planowanie nie jest niezbędnym elementem życia. Tacy ludzie twierdzą, że należy żyć chwilą. I rzeczywiście, wielu rzeczy można nie planować. Jednak, kiedy mamy pracę, studia, rodzinę czy po prostu więcej innych obowiązków - bez planowania daleko nie zajdziemy, a szybko zacznie się nam wydawać, że doba jest zbyt krótka.

Szczególnie ważnym aspektem jest planowanie spotkań biznesowych. Przy kontakcie z klientem, zarówno przy dłuższej współpracy, jak i potencjalnym, nie można działać spontanicznie. Ciężko wyobrazić sobie prezesa dużej firmy, który nie miałby zaplanowanych spotkań na - co najmniej - kilka dni do przodu. Aby się w tym wszystkim nie pogubić warto mieć narzędzie, które będzie odpowiadało naszym potrzebom w stu procentach.

Takie narzędzie jak planer czy kalendarz towarzyszy nam codziennie, szczególnie w pracy. Warto, aby taki aspekt życia jak planowanie sprawiał nam maksymalnie dużo przyjemności, a jeśli i tak nie jest to ulubione zajęcie - aby planowanie odbywało się możliwie szybko oraz intuicyjnie.

W dzisiejszych czasach ludzie coraz częściej odkładają papierowe planery na rzecz telefonu czy komputera, które zawsze mają przy sobie. Dlatego w pracy przedstawiony będzie system do planowania spotkań biznesowych, który może być używany na komputerze. Dzięki temu planowanie nie będzie wymagało dodatkowego sprzętu w plecaku czy torebce, ponieważ dostęp do systemu będziemy mieć z każdego miejsca na świecie, a podczas wyjazdów biznesowych nie będzie trzeba pamiętać o niczym więcej niż laptop.

Aby czerpać maksymalnie możliwą przyjemność z planowania proponuję narzędzie, które jest minimalistyczne, a także proste i intuicyjne w obsłudze. System pomoże zaplanować nad czasem oraz biznesowymi obowiązkami. Stworzenie odpowiedniego planu działania to podstawa organizacji.

Rozdział 2

Cel i zakres pracy

Celem pracy jest zaprojektowanie i wykonanie systemu informatycznego, który ułatwi planowanie spotkań biznesowych. System ma pozwalać na intuicyjne zarządzanie spotkaniami, a także osobistymi notatkami.

Realizację pracy podzielono na następujące etapy:

- stworzenie relacyjnej bazy danych,
- stworzenie aplikacji do planowania i zarządzania spotkaniami,
- stworzenie systemu do zarządzania notatkami,
- wdrożenie systemu, testowanie oraz uruchomienie.

Na początku pracy zostanie przeprowadzona analiza wymagań do stworzenia systemu. Zostaną przedstawione wymagania zarówno funkcjonalne, czyli projekt funkcji systemu, a także нефункционалне, takie jak użyte technologie, narzędzia czy zaprojektowane zabezpieczenia systemu.

W kolejnym etapie zostanie przedstawiony projekt bazy danych, w której będą znajdować się dane dotyczące klientów, pracowników, a także ich spotkań i notatek. Ponieważ baza danych jest relacyjna, zostaną pokazane także relacje pomiędzy poszczególnymi tabelami oraz uprawnienia grup użytkowników.

Następnie zostanie pokazany interfejs użytkownika, jego przykładowe funkcjonalności i zabezpieczenia. W kolejnym rozdziale znajduje się implementacja baz danych, czyli stworzenie tabel, ich indeksów oraz ograniczeń. Na końcu rozdziału przedstawione jest testowanie bazy danych. Ostatni etap pracy to implementacja i testy samej aplikacji, opis oraz programowanie wybranych funkcji systemu.

Rozdział 3

Wymagania projektowe

3.1 Opis działania systemu

System jest przeznaczony dla pracowników firmy. Każdy z nich ma możliwość rejestracji i zalogowania się. Mogą dowolnie planować swoje spotkania biznesowe, a także dodawać i edytować notatki. Każde spotkanie może zostać zaplanowane na dany dzień i godzinę. Notatka może być dodana do dowolnego dnia lub bez daty, dzięki czemu można dowolnie nimi zarządzać, ułatwiając organizację spotkań. Dodatkowo w systemie jest możliwość edycji oraz usuwania notatek i spotkań, a także przeglądania ich w zależności od dnia, miesiąca i roku. Każdy pracownik może także dodać klienta do bazy danych i umówić spotkanie z każdym klientem.

3.2 Wymagania funkcjonalne

Każde spotkanie ma ustaloną datę oraz godzinę, kiedy się odbędzie. Na dany dzień można też zaplanować jakieś działanie lub dodać informację za pomocą notatki. Notatka może również nie mieć określonej daty, a być jedynie przypomnieniem. Dostęp do systemu planowania mogą mieć wyłącznie zalogowani pracownicy. Użytkownik to każda osoba, która przegląda stronę, a pracownik to każda zarejestrowana osoba.

Użytkownik

System informatyczny ma dostarczyć następujące funkcjonalności dla użytkownika:

- możliwość rejestracji,
- możliwość zalogowania,
- możliwość przeglądania strony głównej.

Pracownik

Liczba funkcjonalności dla pracownika jest znacznie większa:

- planowanie spotkania,
- stworzenie notatki,
- dodanie klienta,
- przeglądanie notatek i spotkań,
- dowolne zarządzanie notatkami i spotkaniami,
- możliwość zalogowania,
- możliwość rejestracji.

3.3 Wymagania niefunkcjonalne

Zaprojektowany system powinien być wygodny i intuicyjny w użytkowaniu dla każdego pracownika. System do planowania spotkań biznesowych ma być użyteczny oraz prosty w obsłudze.

Technologie i narzędzia

Aby stworzyć aplikację zostaną użyte różne technologie i narzędzia. Do projektu i implementacji części front-end wykorzystano HTML5 oraz CSS3 [3]. Relacyjna baza danych stworzona zostanie w MySQL [1], [2], a następnie połączona ze stroną za pomocą PHP [1], [2]. Back-end systemu zostanie wykonany w języku PHP. Projekt zostanie wykonany na localhost, gdzie zostanie użyty serwer XAMPP/Apache.

Do stworzenia systemu użyte zostały następujące technologie i narzędzia:

- system operacyjny: MacOS,
- szablon: HTML5 oraz CSS3,
- relacyjna baza danych: MySQL,
- programowanie oraz połączenie z bazą danych: PHP,
- serwer: Apache.

Bezpieczeństwo systemu

Aby zachować maksymalne bezpieczeństwo systemu, zostaną zastosowane następujące zabezpieczenia:

- możliwość przeglądania systemu jedynie dla pracowników,
- szyfrowanie haseł,
- potwierdzenie rejestracji za pomocą wiadomości e-mail.

Przyjęte założenia projektowe

Do realizacji projektu przyjęto następujące założenia projektowe:

- stworzenie wygodnego systemu do planowania spotkań biznesowych,
- stworzenie systemu do zarządzania notatkami,
- system z możliwością logowania i rejestracji.

Rozdział 4

Projekt systemu

4.1 Projekt relacyjnej bazy danych

Baza danych spełnia podstawowe zasady projektowe, jest to baza znormalizowana. Dzięki normalizacji możemy organizować dane, tak aby ich ilość była optymalna oraz eliminować anomalie, powtarzające się dane. Atomowość danych zapewnia, że każda tabela przechowuje dane tak, aby w jednym polu znalazła się jedna informacja, np w tabeli 'Pracownicy' osobno są kolumny 'Imie' oraz 'Nazwisko'. Każda tabela przechowuje dane dotyczące tylko obiekty właściwe dla niej. W tabeli 'Spotkania' nie ma informacji o kliencie innych niż 'IdKlienta'. Resztę danych, jak imię i nazwisko można wyciągnąć już z tabeli 'Klienci'. Żadne z kolumn we wszystkich 4 tabelach nie zależą od innej kolumny w danej tabeli. Znaczy to, że baza danych jest znormalizowana do trzeciej postaci normalnej. Dane znajdujące się w bazie przechowuje się w czterech tabelach:

- Klienci,
- Spotkania,
- Pracownicy,
- Notatki.

Tabela 'Klienci' zawiera opis klientów firmy, w szczególności imię, nazwisko, nazwę firmy, w której pracuje oraz e-mail. Połączona jest z tabelą 'Spotkania' przez unikalny klucz 'IdKlienta'.

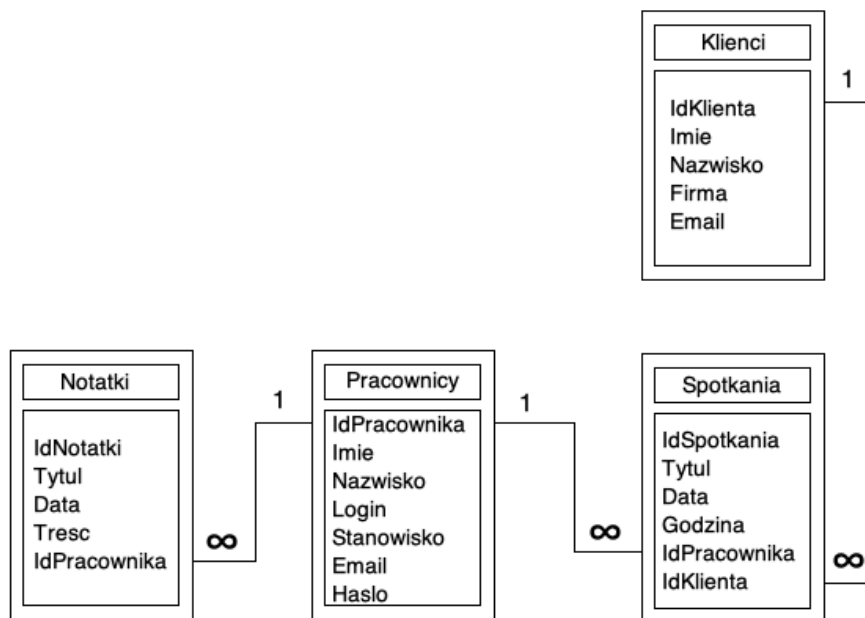
Tabela 'Spotkania' przechowuje dane dotyczące spotkań: opis spotkania (pole 'Tytuł'), datę, godzinę oraz klucze identyfikujące klienta i pracownika, którzy odbywają spotkanie. Każde spotkanie może mieć przypisanego jednego klienta oraz jednego pracownika.

Informacje dotyczące pracowników przechowywane są w tabeli 'Pracownicy', zawierają się tam takie informacje jak imię, nazwisko, stanowisko, na którym pracuje, e-mail, hasło

w zaszyfrowanej postaci, login oraz unikalny indeks 'IdPracownika', który jest kluczem łączącym tabelę 'Spotkania' oraz 'Notatki'.

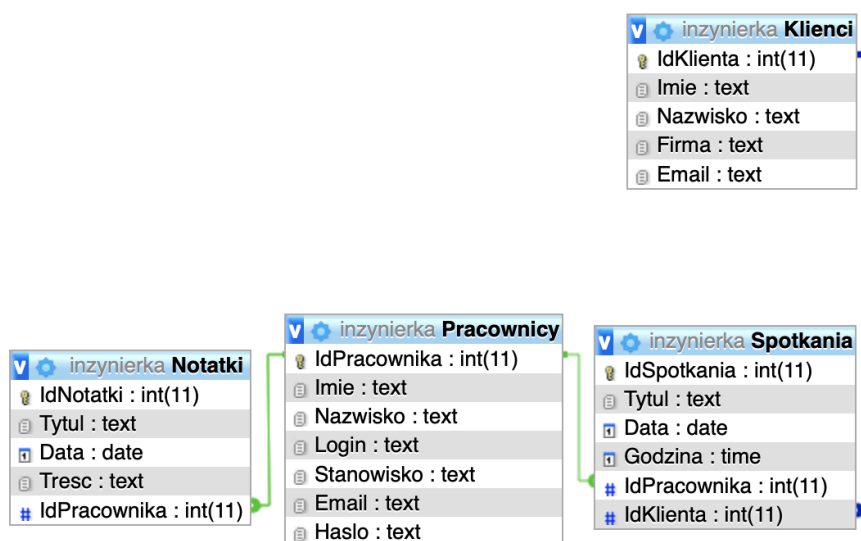
Ostatnią z tabel jest tabela 'Notatki'. W tabeli oprócz samej treści notatki przechowywany jest też jej tytuł oraz data. Kluczem głównym tabeli jest 'IdNotatki'. Do każdej notatki przypisany jest tylko jeden pracownik.

Model koncepcyjny struktury bazy danych przedstawiony jest na Rysunku 4.1.



Rysunek 4.1: Model koncepcyjny

Model fizyczny bazy danych został przedstawiony na Rysunku 4.2. Tabele 4.1 - 4.4 przedstawiają szczegółowy opis pól poszczególnych tabel w bazie danych.



Rysunek 4.2: Model gotowej bazy danych

Tabela 4.1: Tabela pracowników

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo
1	IdPracownika	int(11)			Nie	Brak		AUTO_INCREMENT
2	Imie	text	utf8_polish_ci		Nie	Brak		
3	Nazwisko	text	utf8_polish_ci		Nie	Brak		
4	Login	text	utf8_polish_ci		Nie	Brak		
5	Stanowisko	text	utf8_polish_ci		Nie	Brak		
6	Email	text	utf8_polish_ci		Nie	Brak		
7	Haslo	text	utf8_polish_ci		Nie	Brak		

Tabela 4.2: Tabela klientów

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo
1	IdKlienta	int(11)			Nie	Brak		AUTO_INCREMENT
2	Imie	text	utf8_polish_ci		Nie	Brak		
3	Nazwisko	text	utf8_polish_ci		Nie	Brak		
4	Firma	text	utf8_polish_ci		Tak	Brak		
5	Email	text	utf8_polish_ci		Nie	Brak		

Tabela 4.3: Tabela notatek

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo
1	IdNotatki	int(11)			Nie	Brak		AUTO_INCREMENT
2	Tytul	text	utf8_polish_ci		Nie	Brak		
3	Data	date			Tak	Brak		
4	Tresc	text	utf8_polish_ci		Nie	Brak		
5	IdPracownika	int(11)			Nie	Brak		

Tabela 4.4: Tabela spotkań

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo
1	IdSpotkania	int(11)			Nie	Brak		AUTO_INCREMENT
2	Tytul	text	utf8_polish_ci		Nie	Brak		
3	Data	date			Nie	Brak		
4	Godzina	time			Nie	Brak		
5	IdPracownika	int(11)			Nie	Brak		
6	IdKlienta	int(11)			Nie	Brak		

Mechanizmy bezpieczeństwa

W celu zachowania bezpieczeństwa ograniczono dostęp do danych przez mechanizm uprawnień. Uprawnienia poszczególnych użytkowników przedstawiono w Tabeli 4.5.

Tabela 4.5: Uprawnienia użytkowników do poszczególnych tabel w bazie danych

	Użytkownik	Pracownik	Administrator
Pracownicy	C	RU	CRUD
Klienci	-	CRU	CRUD
Notatki	-	CRUD	CRUD
Spotkania	-	CRUD	CRUD

4.2 Interfejs użytkownika

Na stronie głównej interfejsu użytkownika znajduje się kafelkowe menu, w którym są następujące opcje do wyboru:

- Dzień - wyświetlenie bieżącego dnia, razem z zaplanowanymi spotkaniami oraz notatkami. Za pomocą strzałek przy dacie można zmieniać dzień na poprzedni i kolejny. Po naciśnięciu na wybraną notatkę lub spotkanie można dokonać jej edycji.
- Miesiąc - wyświetla trwający miesiąc z podświetlonym dniem dzisiejszym, po prawej stronie są wypisane notatki na dany miesiąc. Dzięki strzałkom przy aktualnym miesiącu można zmieniać miesiąc w przód oraz w tył. Po naciśnięciu na notatkę można ją edytować. Przy wybraniu konkretnego dnia, aplikacja przeniesie pracownika do spotkań i notatek zaplanowanych na ten dzień (jak w zakładce 'Dzień').
- Rok - pokazuje bieżący rok, który można zmieniać analogicznie jak miesiąc i dzień, a także wybierać konkretny miesiąc, który przeniesie nas do zakładki 'Miesiąc' z wybranym miesiącem i rokiem. Po prawej stronie można zobaczyć notatki na wybrany rok i również wybrać jedną i dokonać edycji, a także dodać nową.
- Notatki - wyświetla spis wszystkich notatek, razem z tymi, które nie mają zaplanowanej konkretnej daty. Tu także jest możliwość dodania i edycji notatki.
- Zaloguj/Wyloguj - pierwsza opcja przenosi niezalogowanego użytkownika do panelu logowania i rejestracji, a druga pokazuje się jedynie użytkownikowi zalogowanemu. Po wybraniu tej opcji pracownik zostanie wylogowany.
- Dodaj spotkanie - wyświetla dwa formularze, jeden do dodania spotkania z wybranym klientem, a drugi do dodania klienta, żeby móc umówić z nim spotkanie.
- Dodaj notatkę - umożliwia dodanie nowej notatki.
- W stopce aplikacji znajdują się dane osoby, która ją stworzyła oraz prawa autorskie.

Na Rysunku 4.3 przedstawiono stronę główną aplikacji dla niezalogowanego użytkownika, natomiast na Rysunku 4.4 wygląd strony głównej po zalogowaniu.



Rysunek 4.3: Strona główna dla niezalogowanego użytkownika

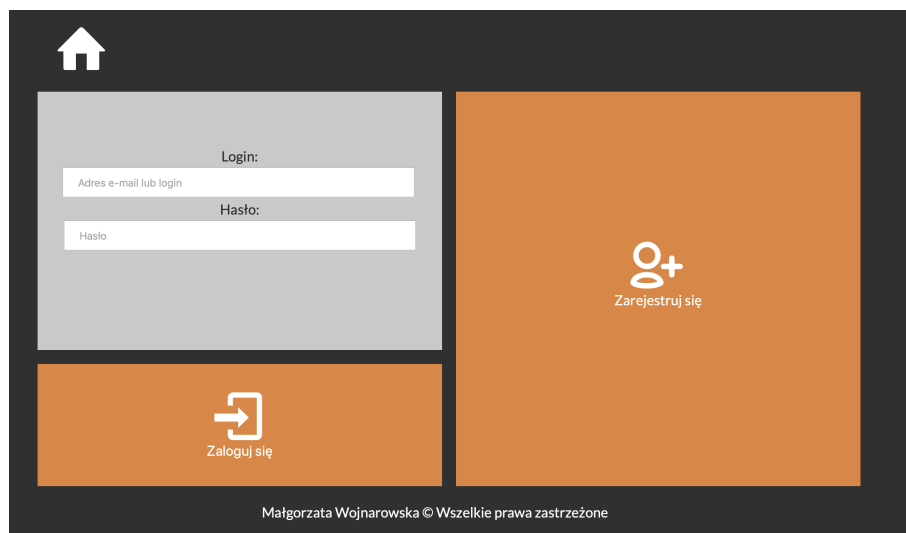


Rysunek 4.4: Strona główna dla zalogowanego pracownika

Funkcja rejestracji nowego użytkownika

Przed zalogowaniem, kliknięcie dowolnego pola przeniesie użytkownika do panelu logowania. Tam nowy użytkownik może stworzyć konto po naciśnięciu przycisku "Zarejestruj się". W tym celu zostanie otworzona nowa podstrona z formularzem do rejestracji. Po wypełnieniu wszystkich pól i stworzeniu konta, użytkownik zostanie przeniesiony na stronę, która wyświetla komunikat o poprawnej rejestracji oraz przycisk do panelu logowania. Od teraz pracownik będzie mógł się zalogować przez wybrany login i hasło.

Panel logowania do aplikacji pokazany jest na Rysunku 4.5.



Rysunek 4.5: Panel logowania

Funkcja dodawania i edycji klienta lub spotkania

Każdy zalogowany użytkownik ma możliwość dodania własnego spotkania oraz stworzenia klienta. Aby to zrobić należy wypełnić znajdujący się po prawej stronie formularz, który ukaże się po naciśnięciu przycisku "Dodaj spotkanie". Następnie za pomocą rozwijanej listy zalogowany pracownik ma możliwość zapisania spotkania z danym klientem za pomocą wyboru z listy rozwijanej. Aby dodać notatkę należy wypełnić analogiczny formularz po wyborze opcji "Dodaj notatkę".

Na Rysunku 4.6 przedstawione jest dodawanie spotkania i klienta do bazy danych.

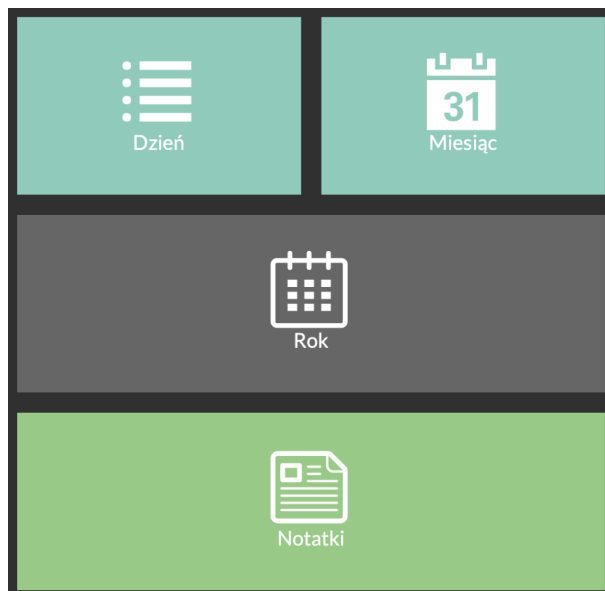


Rysunek 4.6: Dodawanie spotkania lub klienta

Funkcja przeglądania kalendarza, zapisanych spotkań oraz notatek

Pracownik może także przeglądać zapisane notatki oraz spotkania, segregując je za pomocą daty: konkretnego dnia, miesiąca oraz roku, a także wyświetlając wszystkie notatki bez względu na datę. Pracownik może za pomocą kliknięcia wybrać konkretny miesiąc oraz dzień.

Dostępne funkcje przeglądania notatek oraz spotkań pokazane są na Rysunku 4.7.

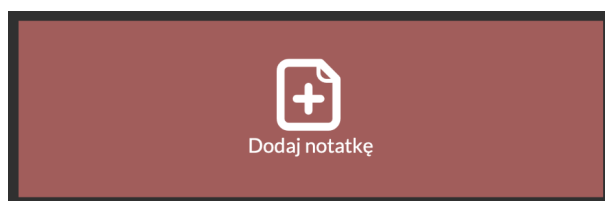


Rysunek 4.7: Przeglądanie kalendarza

Funkcja dodawania oraz edycji notatki

Po zalogowaniu do systemu pracownik może także dowolnie dodawać oraz edytować spotkania i notatki. Aby to zrobić należy wybrać daną notatkę, która wyświetla się po prawej stronie dnia, miesiąca lub roku, a także po wyświetleniu listy wszystkich notatek.

Po wybraniu przycisku 'Dodaj notatkę' (Rysunek 4.8) otwiera się formularz, który pozwala na stworzenie notatki.



Rysunek 4.8: Dodawanie notatki

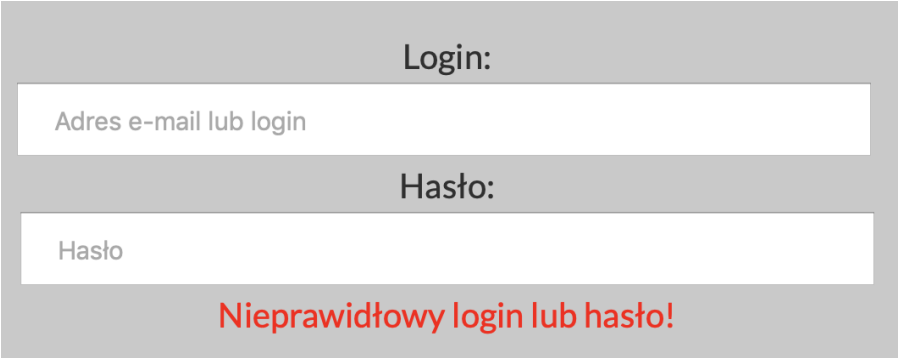
Zabezpieczenia

W aplikacji znajdują się pewne zabezpieczenia, które mają na celu zwiększenie bezpieczeństwa systemu. Zabezpieczenia są zarówno na poziomie bazy danych, jak i przy programowaniu aplikacji. Są to między innymi:

- ograniczenie uprawnień dla poszczególnych użytkowników,
- przy rejestracji zabezpieczenie akceptacji regulaminu,
- przy wypełnianiu formularza adres e-mail musi mieć poprawną formę,

- podczas rejestracji login nie może zawierać znaków specjalnych,
- ograniczenie długości znaków podczas wypełniania formularzy,
- zabezpieczenie przed otwarciem podstron przez niezalogowanych użytkowników.

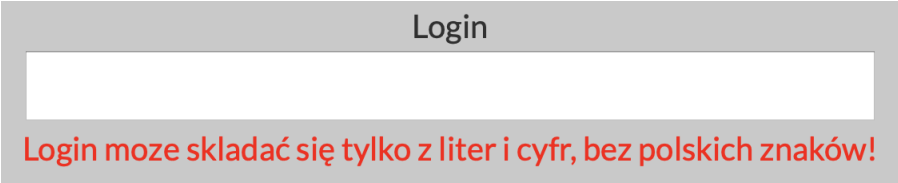
Podczas logowania wpisanie błędnego loginu lub hasła skutkuje błędem logowania, który przedstawiony jest na Rysunku 4.9.



The image shows a login form with a grey background. At the top, the word "Login:" is centered. Below it is a white input field containing the placeholder text "Adres e-mail lub login". Underneath the input field, the word "Hasło:" is centered. Below that is another white input field containing the placeholder text "Hasło". At the bottom of the form, the error message "Nieprawidłowy login lub hasło!" is displayed in red text.

Rysunek 4.9: Błąd logowania

Formularz rejestracji należy poprawnie wypełnić. Loginu, musi się składać jedynie z liter oraz cyfr, nie może zawierać polskich znaków. Login powinien mieć od 3 do 20 znaków. Jeśli ten warunek nie jest spełniony, to przy formularzu pojawi się błąd, który jest pokazany na Rysunku 4.10. Ponieważ login musi być unikalny, błąd pojawi się także przy próbie wybrania loginu, który już znajduje się w bazie danych. Każdy z przypadków błędnego wpisania loginu skutkuje innym komunikatem o niepoprawności wypełnienia formularza.



The image shows a login form with a grey background. At the top, the word "Login" is centered. Below it is a white input field. At the bottom of the form, the error message "Login może składać się tylko z liter i cyfr, bez polskich znaków!" is displayed in red text.

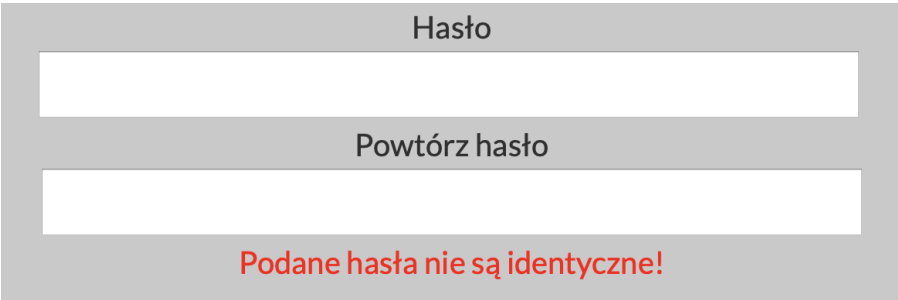
Rysunek 4.10: Błąd loginu

Podczas rejestracji sprawdzana jest także poprawność wpisanego e-maila. Ponieważ jeden e-mail może być przypisany do jednego konta, tutaj również sprawdzane jest, czy wybrany e-mail nie znajduje się już w bazie danych. W zależności od tego, który z tych warunków nie został spełniony, wyświetlany jest odpowiedni komunikat o błędzie. Jeśli użytkownik pomylił się w formie e-maila - na przykład nie wystąpił w nim znak '@' lub nic się po nim nie znajduje - zostanie wyświetlona wiadomość o błędzie z Rysunku 4.11.

A screenshot of a web form with a grey background. At the top, the text "Adres e-mail" is centered. Below it is a white rectangular input field. At the bottom, the text "Podaj poprawny adres e-mail" is centered in red.

Rysunek 4.11: Błąd e-maila

W formularzu rejestracji podane są dwa okna do wpisania hasła. Jedno z nich występuje po to, aby potwierdzić, że użytkownik nie pomylił się przy jego wpisywaniu. Dla bezpieczeństwa sprawdzamy, czy oba hasła podane przez użytkownika są takie same. Hasła muszą mieć od 3 do 20 znaków. Jeśli się nie zgadzają zostanie wyświetlony komunikat pokazany na Rysunku 4.12.

A screenshot of a web form with a grey background. At the top, the text "Hasło" is centered. Below it is a white rectangular input field. Below that, the text "Powtórz hasło" is centered. Below that is another white rectangular input field. At the bottom, the text "Podane hasła nie są identyczne!" is centered in red.

Rysunek 4.12: Błąd hasła - podane hasła się różnią

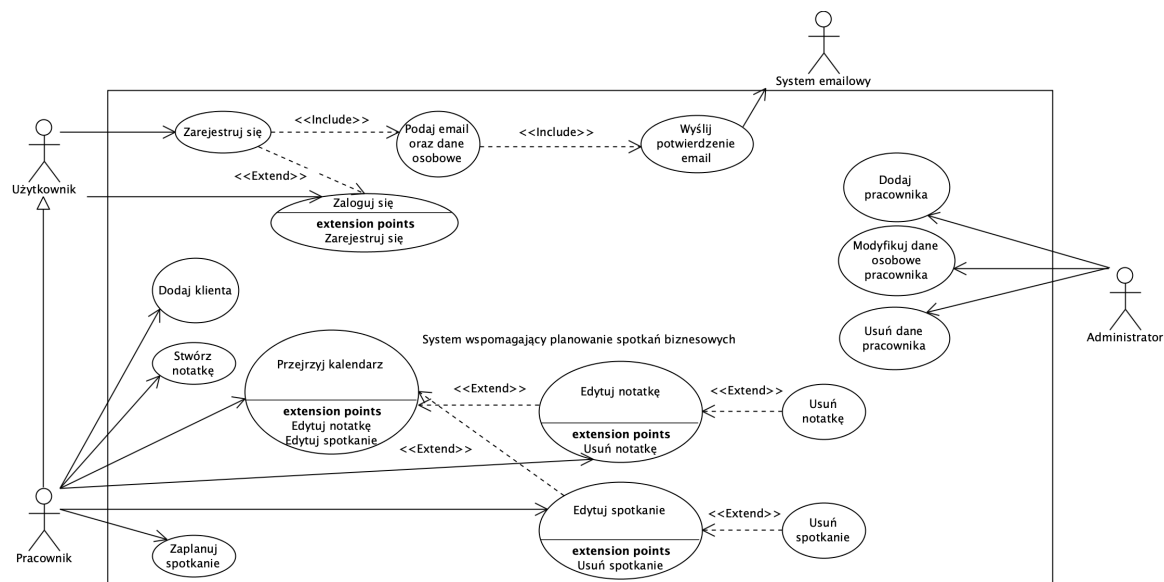
Aby uniknąć prawnych zażaleń użytkownik musi zaakceptować regulamin korzystania z aplikacji. Jest to także kontrola przed sztucznymi kontami, tworzonymi przez zaprogramowane algorytmy. Bez akceptacji konto pracownika nie zostanie stworzone, a w formularzu wyświetli się komunikat z Rysunku 4.13.

A screenshot of a web form with a grey background. At the top left, there is a small white square checkbox. To its right, the text "Akceptuję regulamin" is centered. Below it, the text "Zaakceptuj regulamin!" is centered in red.

Rysunek 4.13: Błąd regulaminu - brak zaakceptowania

4.3 Diagram projektowy

Korzystając z wymagań funkcjonalnych (patrz 3.2) został stworzony diagram prezentujący interakcję pomiędzy poszczególnymi grupami użytkowników, a aplikacją. Diagram ten znajduje się na Rysunku 4.14.



Rysunek 4.14: Diagram przypadków użycia

Rozdział 5

Implementacja systemu baz danych

5.1 Tworzenie tabel

Aby zaprogramować użytkową aplikację, należy utworzyć relacyjną bazę danych, gdzie będą mogły być przechowywane informacje na temat klientów, pracowników, a także ich spotkań i osobistych notatek. Na początku tworzymy potrzebne nam tabele za pomocą języka SQL:

- tabela 'Spotkania', gdzie będą się znajdować informacje na temat spotkań pracowników, takie jak ich temat ('Tytuł'), data oraz godzina spotkania, a także dane pracownika oraz klienta, z którym to spotkanie się odbędzie,
- tabela 'Notatki' zawiera informacje na temat jej tytułu, treści, unikalnego indeksu 'IdPracownika', który stworzył tę notatkę, a także opcje daty, do której można, ale nie trzeba jej przypisać. Oznacza to, że notatka nie musi mieć daty,
- tabela 'Klienci' przechowuje dane klientów. Znajdują się tam takie informacje jak imię i nazwisko klienta, firma, w której pracuje (może nie pracować w żadnej), a także jego e-mail do celów kontaktowych,
- tabela 'Pracownicy', w której znajdują się ich dane, takie jak imię, nazwisko, stanowisko, e-mail a także indywidualny login i hasło, za pomocą których pracownik może zalogować się do systemu.

Wszystkie tabele mają zapisane kodowanie utf8, dzięki czemu w aplikacji można używać znaków polskich. Tworzenie poszczególnych tabel pokazane jest na Listingach 5.1 - 5.4.

Listing 5.1: Stworzenie tabeli spotkań

```
CREATE TABLE 'Spotkania' (  
  'IdSpotkania' int(11) NOT NULL,  
  'Tytul' text COLLATE utf8_polish_ci NOT NULL,  
  'Data' date NOT NULL,  
  'Godzina' time NOT NULL,  
  'IdPracownika' int(11) NOT NULL,  
  'IdKlienta' int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
```

Listing 5.2: Stworzenie tabeli notatek

```
CREATE TABLE 'Notatki' (  
  'IdNotatki' int(11) NOT NULL,  
  'Tytul' text COLLATE utf8_polish_ci NOT NULL,  
  'Data' date DEFAULT NULL,  
  'Tresc' text COLLATE utf8_polish_ci NOT NULL,  
  'IdPracownika' int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
```

Listing 5.3: Stworzenie tabeli klientów

```
CREATE TABLE 'Klienci' (  
  'IdKlienta' int(11) NOT NULL,  
  'Imie' text COLLATE utf8_polish_ci NOT NULL,  
  'Nazwisko' text COLLATE utf8_polish_ci NOT NULL,  
  'Firma' text COLLATE utf8_polish_ci ,  
  'Email' text COLLATE utf8_polish_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
```

Listing 5.4: Stworzenie tabeli pracowników

```
CREATE TABLE 'Pracownicy' (  
  'IdPracownika' int(11) NOT NULL,  
  'Imie' text COLLATE utf8_polish_ci NOT NULL,  
  'Nazwisko' text COLLATE utf8_polish_ci NOT NULL,  
  'Login' text COLLATE utf8_polish_ci NOT NULL,  
  'Stanowisko' text COLLATE utf8_polish_ci NOT NULL,  
  'Email' text COLLATE utf8_polish_ci NOT NULL,  
  'Haslo' text COLLATE utf8_polish_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
```

5.2 Definiowanie indeksów oraz ograniczeń

Aby relacyjna baza danych była funkcjonalna, tworzymy relacje pomiędzy poszczególnymi tabelami. W tabeli 'Notatki' dodajemy klucz obcy 'IdPracownika', który wiąże tę tabelę z tabelą 'Pracownicy'. Dzięki temu, każdy pracownik ma swoje notatki i nie ma dostępu do notatek współpracowników.

Analogicznie dodajemy klucz obcy 'IdPracownika' do tabeli 'Spotkania'. Tam dodajemy także klucz obcy 'IdKlienta', żeby łatwo rozpoznać z kim pracownik umówił dane spotkanie.

Tworzenie indeksów dla tabel 'Klienci', 'Notatki', 'Pracownicy' oraz 'Spotkania' pokazane jest kolejno na Listingu 5.5, natomiast ograniczenia tabel 'Notatki' i 'Spotkania' znajdują się na Listingu 5.6.

Listing 5.5: Tworzenie indeksów dla tabel

```

—
— Indeksy dla tabeli 'Klienci'
—
ALTER TABLE 'Klienci'
  ADD PRIMARY KEY ('IdKlienta');

—
— Indeksy dla tabeli 'Notatki'
—
ALTER TABLE 'Notatki'
  ADD PRIMARY KEY ('IdNotatki'),
  ADD KEY 'napisana przez' ('IdPracownika');

—
— Indeksy dla tabeli 'Pracownicy'
—
ALTER TABLE 'Pracownicy'
  ADD PRIMARY KEY ('IdPracownika');

—
— Indeksy dla tabeli 'Spotkania'
—
ALTER TABLE 'Spotkania'
  ADD PRIMARY KEY ('IdSpotkania'),
  ADD KEY 'z' ('IdKlienta'),
  ADD KEY 'spotyka sie' ('IdPracownika') USING BTREE;

```

Listing 5.6: Ograniczenia tabel

```

—
— Ograniczenia dla tabeli 'Notatki'
—
ALTER TABLE 'Notatki'
  ADD CONSTRAINT 'napisana przez' FOREIGN KEY ('IdPracownika') REFERENCES 'inzynierka'.'. '
    Pracownicy' ('IdPracownika') ON DELETE CASCADE ON UPDATE CASCADE;

—
— Ograniczenia dla tabeli 'Spotkania'
—
ALTER TABLE 'Spotkania'
  ADD CONSTRAINT 'spotyka sie' FOREIGN KEY ('IdPracownika') REFERENCES 'inzynierka'.'. '
    Pracownicy' ('IdPracownika') ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT 'z' FOREIGN KEY ('IdKlienta') REFERENCES 'inzynierka'.'. 'Klienci' ('
    IdKlienta') ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

```

Automatyczne numerowanie indeksów (auto_increment) sprawia, że użytkownik bazy danych nie musi się martwić o numerację rekordów. Jest to atrybut klucza głównego, który musi być unikalny dla każdego rekordu w tabeli. Implementacja kolejno dla tabel

'Klienci', 'Notatki', 'Pracownicy' oraz 'Spotkania' jest przedstawiona na Listingu 5.7.

Listing 5.7: Automatyczne numerowanie indeksów

```
—  
— AUTO_INCREMENT dla tabeli 'Klienci'  
—  
ALTER TABLE 'Klienci'  
  MODIFY 'IdKlienta' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;  
  
—  
— AUTO_INCREMENT dla tabeli 'Notatki'  
—  
ALTER TABLE 'Notatki'  
  MODIFY 'IdNotatki' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;  
  
—  
— AUTO_INCREMENT dla tabeli 'Pracownicy'  
—  
ALTER TABLE 'Pracownicy'  
  MODIFY 'IdPracownika' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;  
  
—  
— AUTO_INCREMENT dla tabeli 'Spotkania'  
—  
ALTER TABLE 'Spotkania'  
  MODIFY 'IdSpotkania' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=23;
```

5.3 Nadawanie uprawnień użytkownikom

Na podstawie Tabeli 3.5 nadano uprawnienia użytkownikom. Przykłady nadawanych uprawnień znajdują się na Listingach 5.8 oraz 5.9.

Listing 5.8: Nadanie uprawnień w całej bazie danych dla administratora

```
GRANT ALL PRIVILEGES ON 'inzynierka'.* TO 'admin'@'localhost';
```

Listing 5.9: Przykładowe uprawnienia do tabel Notatki i Klienci dla pracownika

```
GRANT ALL PRIVILEGES ON 'Notatki' TO 'pracownik'@'localhost';  
GRANT SELECT, INSERT, UPDATE PRIVILEGES ON 'Klienci' TO 'pracownik'@'localhost';
```

5.4 Testowanie bazy danych

W ramach testów dodajemy przykładowe dane klienta za pomocą zapytania SQL (Listing 5.10), a następnie wyświetlamy tabelę 'Klienci' (Listing 5.11), aby sprawdzić, czy klient został dodany poprawnie. Tabela 'Klienci' po wykonaniu dwóch powyższych zapytań wyświetla się jak na Tabeli 5.1.

Listing 5.10: Przykład dodawania klienta

```
INSERT INTO 'Klienci' ('IdKlienta', 'Imie', 'Nazwisko', 'Firma', 'Email') VALUES (NULL, 'Katarzyna', 'Nowak', 'Atos', 'katarzyna.nowak@atos.com');
```

Listing 5.11: Wyświetlanie tabeli 'Klienci'

```
SELECT * FROM 'Klienci';
```

Tabela 5.1: Wynik zapytań na tabeli 'Klienci'

IdKlienta	Imie	Nazwisko	Firma	Email
1	Krzysztof	Kowalski	KK Production	krzysztof.kowalski@kkproduction.com
2	Natalia	Sadowska	Airos	n.sadowska@airos.com
3	Mariusz	Romanowski	Atos	mariusz.romanowski@atos.pl
4	Aldona	Rybak	Nokia	aldonarybak@nokia.com
5	Katarzyna	Nowak	Atos	katarzyna.nowak@atos.com

Kolejnym testem jest wyświetlenie tabeli 'Notatki' dla pracownika o określonym indeksie. Wynik zapytania z Listingu 5.12 znajduje się w Tabeli 5.2.

Listing 5.12: Wyświetlanie tabeli 'Notatki' dla pracownika o indeksie 4

```
SELECT * FROM 'Notatki' WHERE 'IdPracownika'='4';
```

Tabela 5.2: Tabela 'Notatki' dla pracownika o IdPracownika='4'

IdNotatki	Tytul	Data	Tresc	IdPracownika
16	Notatka z dzisiaj	2018-11-24	Moja notatka	4

Zapytanie SELECT z Listingu 5.13 wyświetla wybrane kolumny z tabeli 'Pracownicy'. W tym przypadku są to kolumny indeksu pracownika, jego login oraz hasło w zaszyfrowanej postaci. Wynik zapytania przedstawia Tabela 5.3.

Listing 5.13: Wyświetlanie wybranych kolumn z tabeli 'Pracownicy'

```
SELECT 'IdPracownika', 'Login', 'Haslo' FROM 'Pracownicy';
```

Tabela 5.3: Wynik zapytania 'SELECT' na tablicy 'Pracownicy'

IdPracownika	Login	Haslo
1	admin	\$2y\$10\$kr6pkXx6bUafLWVn1Dc/3uenjDT/YxNmJyhBCU1opQ8...
2	monisia	\$2y\$10\$V1TXMzMwHfVBXkrVpKcx.r3erJbUBL.YRsK2X3d0Ez...
3	krzysiek	\$2y\$10\$0/Pyggj6SB5gUACc25ix6OP/pqBbS/M9KqDAI54aIUQ...
4	admin1	\$2y\$10\$Q5oHsaSqAcW3MLAvNAmlPembmqF8pDG.qCIWdPn9KKw...

Dodawanie oraz wyświetlanie rekordów działa prawidłowo. Dane są poprawnie odczytywane oraz kodowane. Hasło pracownika jest zaszyfrowane ze względów bezpieczeństwa.

Rozdział 6

Implementowanie i testowanie aplikacji

6.1 Instalacja i konfigurowanie systemu

Do uruchomienia aplikacji należy pobrać oraz uruchomić serwer lokalny XAMPP, a następnie odpalić moduł Apache oraz MySQL. Następnie otworzyć przeglądarkę i wejść na stronę localhost/phpmyadmin, gdzie należy importować bazę danych. Wszystkie pliki aplikacji powinny znajdować się w folderze /htdocs. Po wykonaniu tych akcji należy wpisać w okno przeglądarki adres localhost/aplikacja.

6.2 Instrukcja użytkowania aplikacji

Aplikacja została stworzona tak, aby była prosta w obsłudze. Jej kafelkowy wygląd sprawia, że obsługa jest intuicyjna dla każdego użytkownika. Instrukcje przykładowych funkcji znajdują się poniżej.

Instrukcja rejestracji pracownika

Rejestrację może wykonać każdy użytkownik, który odwiedza stronę. Aby stworzyć nowe konto, należy:

1. wejść w panel logowania,
2. wypełnić formularz ze swoimi danymi, takimi jak imię, nazwisko, a także login i hasło,
3. zaakceptować regulamin,
4. potwierdzić chęć rejestracji za pomocą wybrania przycisku 'Zarejestruj się'.

Po wykonaniu tej instrukcji użytkownikowi zostanie wyświetlona informacja o poprawnie dokonanej rejestracji. Następnie będzie mógł się zalogować na swoje konto za pomocą wybranego loginu oraz hasła.

Instrukcja dodania klienta

Instrukcja przeznaczona jest dla zalogowanego pracownika. Aby dodać klienta, należy:

1. zalogować się za pomocą loginu (lub e-maila) oraz hasła,
2. wejść w zakładkę 'Dodaj spotkanie',
3. wypełnić formularz z danymi klienta,
4. potwierdzić dane klienta wybierając opcję 'Dodaj klienta'.

Po wykonaniu tych czynności klient zostanie dodany do bazy danych. Pracownikowi wyświetli się komunikat o poprawnie dodanym kliencie. Następnie pracownik będzie mógł wybrać opcję 'Dodaj spotkanie' i umówić spotkanie z tym klientem.

Instrukcja edycji notatki

Edycja notatki jest dostępna jedynie dla zalogowanego pracownika. Aby mieć dostęp do tej funkcji, pracownik musi wcześniej mieć stworzoną chociaż jedną notatkę. Aby edytować istniejącą notatkę, należy:

1. zalogować się na swoje konto pracownika,
2. wejść w jedną z wymienionych zakładek: 'Dzień', 'Miesiąc', 'Rok' lub 'Notatki',
3. nacisnąć na tytuł wybranej notatki,
4. wybrać opcję 'Edytuj', która się pojawi po wybraniu notatki,
5. edytować wybrane dane notatki,
6. potwierdzić zmiany za pomocą przycisku 'Zapisz'.

Po edycji notatki zostanie wyświetlona podstrona z komunikatem o poprawnej edycji. Pracownik może edytować notatki dowolną liczbę razy.

Instrukcja usuwania spotkania

Aby usunąć spotkanie trzeba być zalogowanym pracownikiem, a także mieć stworzone przynajmniej jedno spotkanie. Aby je usunąć, należy:

1. zalogować się z panelu logowania,
2. wejść w zakładkę 'Rok', 'Miesiąc' lub 'Dzień',
3. przejść do wybranej daty,
4. wybrać odpowiednie spotkanie naciskając tytuł spotkania,
5. wybrać przycisk 'Edytuj', który pojawi się zamiast opcji 'Dodaj spotkanie',
6. nacisnąć przycisk 'Usuń', nie wypełniając formularza, który się pojawi.

Po wykonaniu tych czynności zostanie wyświetlony komunikat o poprawnym usunięciu spotkania. Usuniętego spotkania nie da się przywrócić.

6.3 Użycie funkcji systemu

6.3.1 Zabezpieczenie stron dla niezalogowanych użytkowników

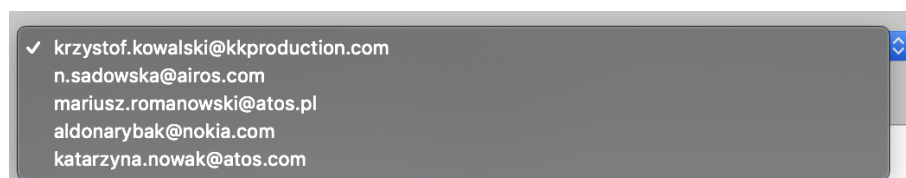
Niezalogowany użytkownik nie ma dostępu do żadnych funkcji aplikacji, poza stroną główną, rejestracji i logowania. Przy próbie dostania się na dowolną inną podstronę, automatycznie zostaje przekierowany na stronę logowania.

6.3.2 Tworzenie spotkań i klientów

Na stronie głównej, a także na podstronach 'Dzień', 'Miesiąc' i 'Rok' znajduje się przycisk prowadzący do zakładki, która umożliwia tworzenie spotkań z klientami. Wygląd podstrony pokazany jest na Rysunku 6.1. Jeśli klient nie istnieje jeszcze w bazie danych, można go dodać po prawej stronie zakładki. Po dodaniu go do bazy danych, pojawi się on w liście rozwijanej przy wyborze klienta do umówienia spotkania (Rysunek 6.2).



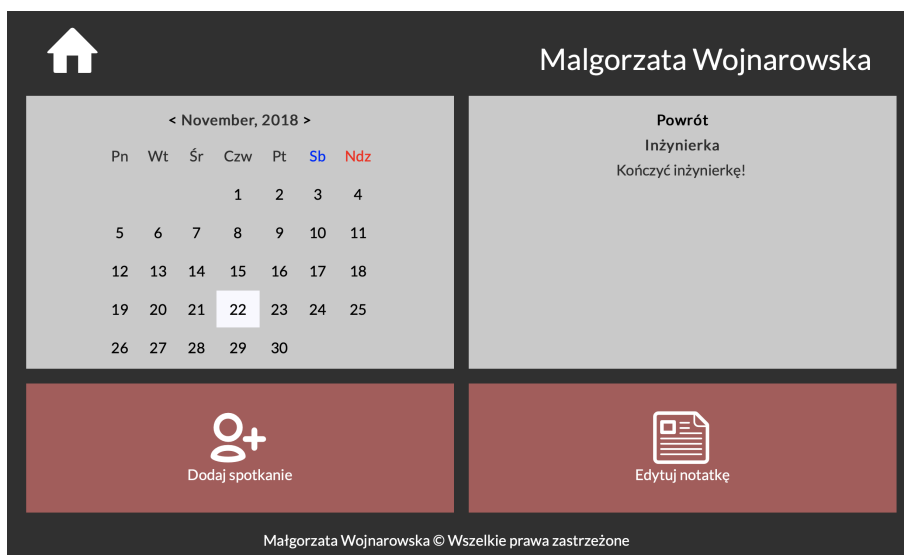
Rysunek 6.1: Dodawanie spotkania lub klienta



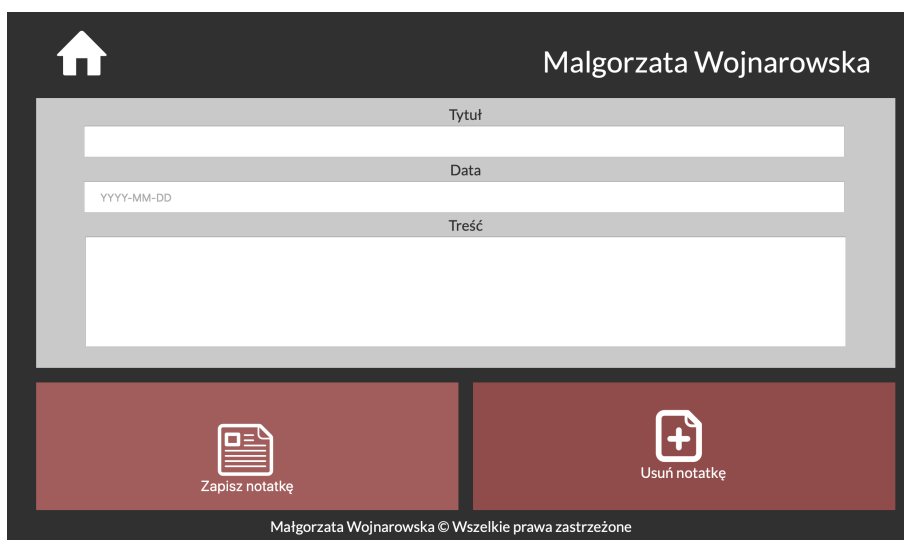
Rysunek 6.2: Wybór klienta z listy rozwijanej

6.3.3 Edycja/usuwanie notatki

Spotkanie i notatki można edytować, a także usunąć. Notatki można przeglądać sortując je według dnia, miesiąca, roku, a także wybrać wszystkie, również takie, które nie mają przypisanej daty. Widok przy sortowaniu według miesiąca przedstawia Rysunek 6.3. Po przejściu do wybranej notatki/spotkania, można na nią nacisnąć, aby wyświetlić szczegóły. Po naciśnięciu przycisku 'Edytuj' pojawi się panel edycji (Rysunek 6.4), gdzie możemy edytować wybraną notatkę, a następnie kliknąć przycisk 'Zapisz', aby ją nadpisać lub 'Usuń', aby całkowicie ją usunąć.



Rysunek 6.3: Widok miesiąca po wybraniu konkretnej notatki



Rysunek 6.4: Panel edycji i usuwania notatki

6.3.4 Rejestracja użytkownika

Każdy nowy użytkownik ma możliwość rejestracji w aplikacji. Aby się zarejestrować należy wejść w panel logowania, a następnie nacisnąć przycisk 'Zarejestruj się'. Po wykonaniu tej czynności pojawi się formularz rejestracji (Rysunek 6.5). Po prawidłowym wypełnieniu formularza zostanie on przekierowany do strony, która informuje go, że został on zarejestrowany poprawnie i od tej pory ma on możliwość zalogowania się w systemie (Rysunek 6.6).

Imię

Nazwisko

Stanowisko


Login

Adres e-mail

Hasło

Powtórz hasło

☐ Akceptuję regulamin

 Zarejestruj się

Małgorzata Wojnarowska © Wszelkie prawa zastrzeżone

Rysunek 6.5: Rejestracja nowego użytkownika



Dziękujemy za rejestrację w serwisie! Możesz się zalogować na swoje konto!

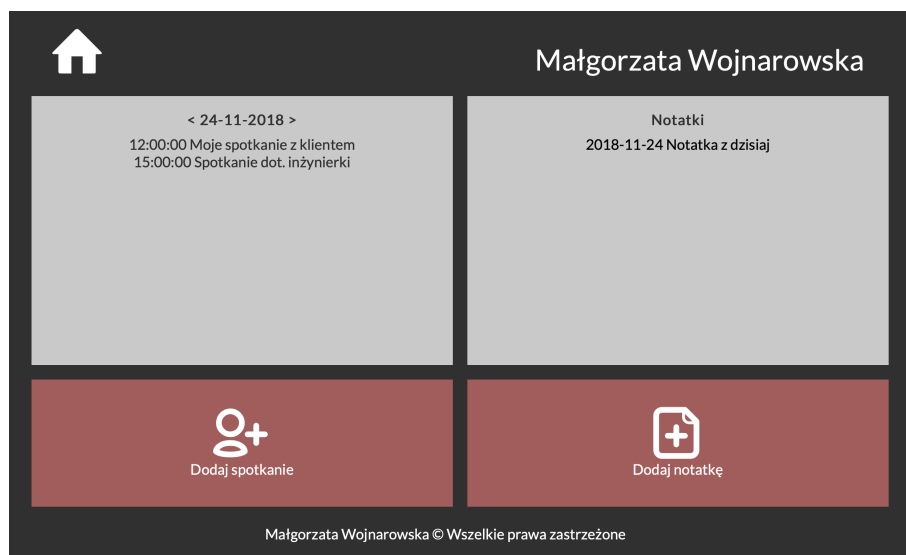
 Zaloguj się

Małgorzata Wojnarowska © Wszelkie prawa zastrzeżone

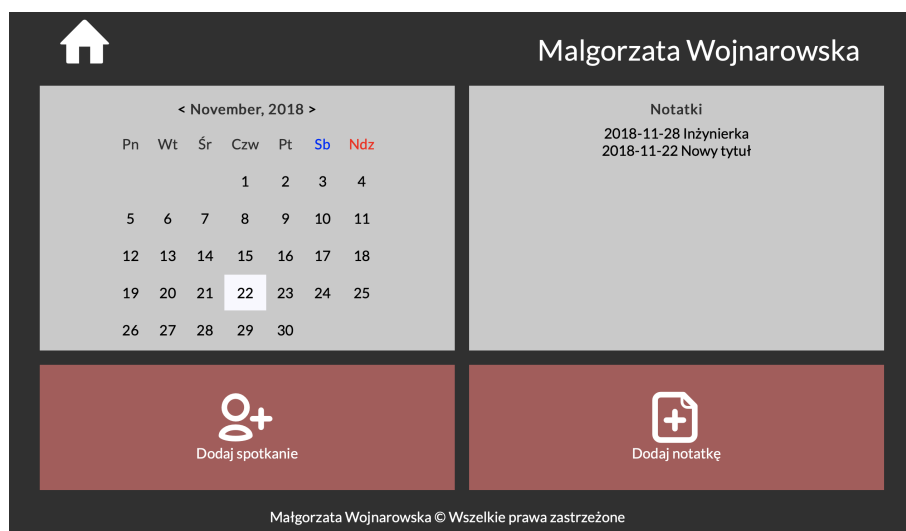
Rysunek 6.6: Ekran po udanej rejestracji

6.3.5 Przeglądanie zaplanowanych spotkań oraz notatek

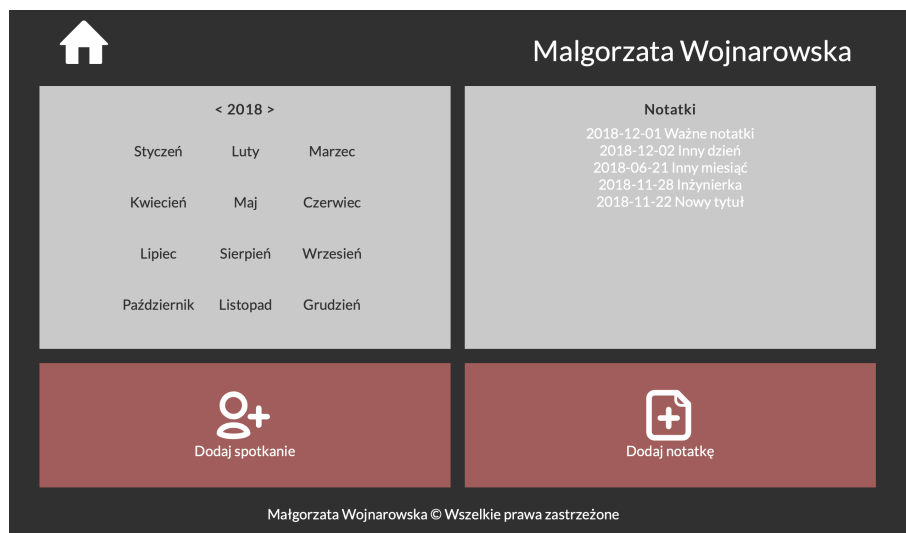
Wszystkie notatki oraz spotkania można przeglądać dowolnie, sortując je na dany dzień (Rysunek 6.7), miesiąc (Rysunek 6.8) oraz rok (Rysunek 6.9). Można także wyświetlić wszystkie notatki niezależnie od daty. Dni, miesiące i lata można przewijać za pomocą strzałek umieszczonych przy dacie. Po naciśnięciu na dany miesiąc w zakładce 'Rok' zostaniemy przeniesieni do danego miesiąca, a po kliknięciu w konkretny dzień - do widoku aplikacji i zaplanowanych czynności na ten dzień. Bieżący dzień jest podświetlany, co widać na Rysunku 6.8.



Rysunek 6.7: Widok bieżącego dnia



Rysunek 6.8: Widok bieżącego miesiąca z zaznaczoną aktualnie datą



Rysunek 6.9: Widok bieżącego roku

6.4 Wybrane funkcje - opis programowania

6.4.1 Implementacja dostępu do bazy danych

Do nawiązania połączenia z bazą danych należy wykorzystać rozszerzenie PHP o nazwie MySQLi używając następujących argumentów: (nazwa serwera mysql, login administratora bazy danych, hasło, nazwa bazy danych, z którą nawiązujemy połączenie). Połączenie z bazą danych przedstawione jest na Listingu 6.1. Dane te są zapisane w pliku 'connect.php' (Listing 6.2).

Listing 6.1: Połączenie z bazą danych

```
require_once "connect.php";
mysqli_report(MYSQLI_REPORT_STRICT);
try{
    $polaczenie = new mysqli($host, $db_user, $db_password, $db_name);
    if($polaczenie->connect_errno!=0){
        throw new Exception(mysqli_connect_errno());
    }
    else{
        %make code
        $polaczenie->close();
    }
}
catch(Exception $e){
    echo 'Error!';
}
```

Listing 6.2: Wnętrze pliku 'connect.php'

```
$host = "localhost";  
$db_user = "root";  
$db_password = "";  
$db_name = "aplikacja";
```

Do odczytania i edycji rekordów w bazie danych, należy użyć kolejnej funkcji oraz podać wybrane zapytanie SQL. Przykłady zapytań przedstawione są na Listingach 6.3 - 6.5.

Listing 6.3: Wybranie klientów o danym adresie e-mail (zapytanie SQL)

```
$rezultat = $polaczenie->query("SELECT IdKlienta FROM Klienci WHERE Email='$email'");
```

Listing 6.4: Dodanie klienta do bazy danych (zapytanie SQL)

```
if($polaczenie->query("INSERT INTO Klienci VALUES(NULL, '$imie', '$nazwisko', '$firma', '$email')")){  
    $_SESSION['udanedodanieklienta'] = true;  
    header("Location: dodanoklienta.php");  
}
```

Listing 6.5: Usunięcie wybranej notatki (zapytanie SQL)

```
if($polaczenie->query("DELETE FROM Notatki WHERE IdNotatki='$idnotatki'"))  
    $_SESSION['udaneusuniecieciannotatki'] = true;
```

6.4.2 Implementacja wybranych funkcjonalności systemu

Pobieranie danych z bazy dotyczących notatek oraz spotkań danego pracownika przedstawione jest na Listingach 6.6 oraz 6.7. Wygląd dnia dla przykładowego pracownika po wyświetleniu spotkań i notatek na dany dzień znajduje się na Rysunku 6.10.



Rysunek 6.10: Widok wybranego dnia

Listing 6.6: Pobranie spotkań z danego dnia i wyświetlenie ich

```
$sql = $polaczenie->query("SELECT * FROM Spotkania WHERE IdPracownika='".$praca['IdPracownika']."' AND MONTH(Data)='".$data->format('m')." AND DAY(Data)='".$data->format('d')." AND YEAR(Data)='".$data->format('Y')."");

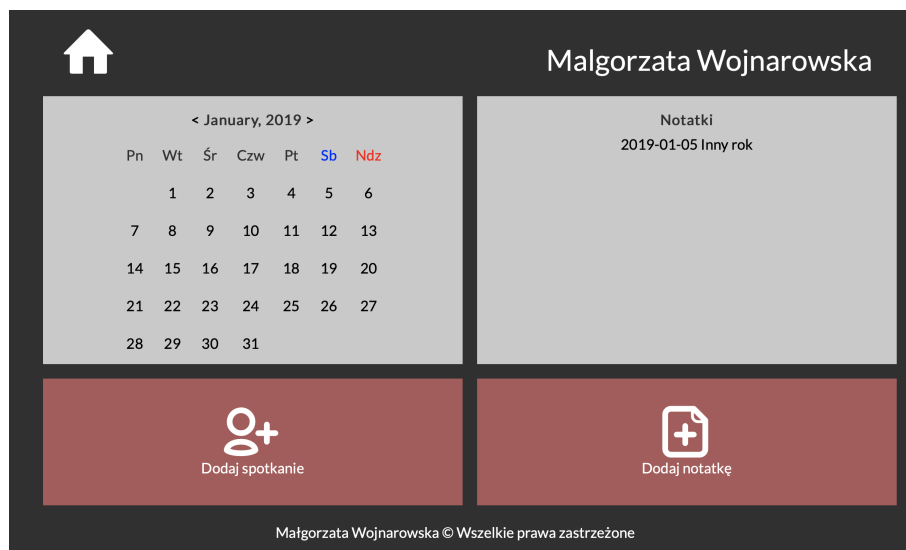
while($row = mysqli_fetch_array($sql))
    echo '<a href="dayspotkaniepokaz.php?n1='.$row['IdSpotkania'].'">'.$row['Godzina'].'<br/>'.$row['Tytul'].'</a>';
```

Listing 6.7: Pobranie notatek z danego dnia i wyświetlenie ich

```
$sql = $polaczenie->query("SELECT * FROM Notatki WHERE IdPracownika='".$praca['IdPracownika']."' AND MONTH(Data)='".$data->format('m')." AND DAY(Data)='".$data->format('d')." AND YEAR(Data)='".$data->format('Y')."");

while($row = mysqli_fetch_array($sql))
    echo '<a href="daypokaz.php?n1='.$row['IdNotatki'].'">'.$row['Data'].'<br/>'.$row['Tytul'].'</a>';
```

Przy dacie dnia, miesiąca oraz roku znajdują się strzałki w prawo oraz lewo do przesuwania daty. Przykład po naciśnięciu strzałki dwa razy w prawo (dwa miesiące od dnia bieżącego) przedstawiono na Rysunku 6.11. Implementacja tego mechanizmu znajduje się na Listingu 6.8.



Rysunek 6.11: Widok po dwukrotnym naciśnięciu strzałki przy nazwie miesiąca w prawo

Listing 6.8: Implementacja przesuwania miesiąca

```

$dzisiaj = date('Y-m-j');
$tytul = date('F, Y', $timestamp);
$poprzedni = date('Y-m', strtotime('-1 month', $timestamp));
$naplony = date('Y-m', strtotime('+1 month', $timestamp));
//Ilość dni w miesiącu
$day_count = date('t', $timestamp);
$str = date('N', $timestamp);
//Tablica dla kalendarza
$weeks = [];
$week = '';
$week .= str_repeat('<td></td>', $str - 1);
for ($day = 1; $day <= $day_count; $day++, $str++) {
    $date = $ym . '-' . $day;
    if ($dzisiaj == $date) {
        $week .= '<td class="today">';
    } else {
        $week .= '<td>';
    }
    $y = date('Y', $timestamp);
    $m = date('m', $timestamp);
    $week .= '<a href="month.php?dzien='.$day.'&miesiac=' . date('m', $timestamp) . '&rok='
        . date('Y', $timestamp) . '&yes=1">' . $day . '</a></td>';
    if ($str % 7 == 0 || $day == $day_count) {
        if ($day == $day_count && $str % 7 != 0) {
            $week .= str_repeat('<td></td>', 7 - $str % 7);
        }
        $weeks[] = '<tr>' . $week . '</tr>';
        $week = '';
    }
}
}

```

6.4.3 Implementacja mechanizmów bezpieczeństwa

Po naciśnięciu przycisku 'Zaloguj' na stronie głównej, użytkownik zostanie przeniesiony do panelu logowania. Na stronie znajduje się formularz, który należy wypełnić odpowiednimi danymi - loginem użytkownika oraz przypisanym do niego hasłem. Po wpisaniu i zatwierdzeniu danych odbywa się weryfikacja użytkownika oraz sprawdzenie, czy występuje on w bazie danych. Następnie sprawdzana jest poprawność hasła przypisanego do danego konta. Jeśli nie podano danych lub dane te są błędne zostaje wypisany błąd logowania. W przypadku, gdy uwierzytelnienie pracownika zakończyło się pomyślnie zostaje ustawiona zmienna sesyjna 'zalogowany', a pracownik jest przekierowywany do strony 'indexzalogowany', gdzie dostęp mają jedynie zalogowani użytkownicy. Dzięki temu, w prawym górnym rogu pojawia się jego imię i nazwisko. Na Listingu 6.9 znajduje się implementacja zmiennej sesyjnej.

Sprawdzenie zmiennej sesyjnej jest w każdym pliku oprócz logowania i rejestracji, dlatego próba dostępu do części dla zalogowanych użytkowników zostanie za każdym razem przekierowana do panelu logowania. Będzie ona wznawiała sesję rozpoczętą podczas

logowania, dopóki pracownik nie naciśnie przycisku 'Wyloguj', który wyzeruje zmienną 'zalogowany'.

Przy logowaniu hasło ma typ 'password', dzięki czemu nikt nie podejrzzy hasła, które jest wpisywane podczas logowania.

Listing 6.9: Zmienna sesyjna - sprawdzenie

```
session_start();

if(!isset($_SESSION['zalogowany'])){
    header('Location: logowanie.php');
    exit();
}
```

Jeśli użytkownik nie posiada konta, może się zarejestrować będąc na stronie logowania. W formularzu należy podać imię, nazwisko, stanowisko, email, login oraz hasło, a także zaakceptować regulamin. Wszystkie dane muszą być poprawnie wypełnione, a następnie zostaną przesłane do bazy do tabeli 'Pracownicy'. Implementacja formularza przedstawiona jest na Listingu 6.10.

Z uwagi na bezpieczeństwo formularz jest tak zaprojektowany, aby nie można było wykonać ataku komputerowego zwanego 'SQL injection' - czyli metody polegającej na wykorzystaniu luki w zabezpieczeniach polegającej na znajomości zapytań do bazy danych. Dzięki tym zabezpieczeniom nikt nie może się dostać na cudze konto dzięki znajomości zapytań SQL.

Listing 6.10: Implementacja formularza do rejestracji pracownika

```
<div class="rectangle" style="width: 1180px; margin: 10px;">
    <form method="post">
        <div class="square2">
            Imię<br/><input type="text" name="imie">
            <br/>Nazwisko<br/><input type="text" name="nazwisko">
            <br/>Stanowisko<br/><input type="text" name="stanowisko">
            <br/>Login<br/><input type="text" name="login">
            <?php
                if(isset($_SESSION['e_login'])){
                    echo '<div class="error">'. $_SESSION['e_login']. '</div>';
                    unset($_SESSION['e_login']);
                } ?>
        </div>
        <div class="square2">
            Adres e-mail<br/><input type="text" name="email">
            <?php
                if(isset($_SESSION['e_email'])){
                    echo '<div class="error">'. $_SESSION['e_email']. '</div>';
                    unset($_SESSION['e_email']);
                } ?>
            <br/>Hasło<br/><input type="password" name="haslo">
            <br/>Powtórz hasło<br/><input type="password" name="haslo2">
            <?php
                if(isset($_SESSION['e_haslo'])){
                    echo '<div class="error">'. $_SESSION['e_haslo']. '</div>';
```

```

        unset($_SESSION['e_haslo']);
    } ?>
<br/><label><input type="checkbox" name="regulamin"/> Akceptuje regulamin</label>
<?php
    if (isset($_SESSION['e_regulamin'])) {
        echo '<div class="error">'. $_SESSION['e_regulamin']. '</div>';
        unset($_SESSION['e_regulamin']);
    } ?>
</div>
<div style="clear: both"></div>
<button type="submit" class="tile11">
    <i class="icon-meeting"></i>
<br/>Zarejestruj sie
</button>
</form>
</div>

```

Jednym z takich sposobów jest zabezpieczenie loginu, aby nie mogło w nich być żadnych znaków specjalnych ani polskich znaków, a także kontrolowanie jego długości. Implementacja sprawdzenia poprawności loginu znajduje się na Listingu 6.11.

Ponieważ login musi być unikalny sprawdzamy też, czy nie znajduje się już taki w bazie danych (Listing 6.12).

Listing 6.11: Implementacja sprawdzenia poprawności loginu

```

//Poprawność loginu
$login = $_POST['login'];
//Sprawdzenie dlugosci loginu
if (strlen($login) < 3 || strlen($login) > 20) {
    $wszystko_ok = false;
    $_SESSION['e_login'] = "Login musi mieć od 3 do 20 znaków!";
}
//login bez znakow polskich, tylko alfanumeryczne
if (!ctype_alnum($login)) {
    $wszystko_ok = false;
    $_SESSION['e_login'] = "Login może składać się tylko z liter i cyfr, bez polskich znaków!";
}

```

Listing 6.12: Implementacja sprawdzenia czy login istnieje w bazie danych

```

//czy login już istnieje
$resultat = $polaczenie->query("SELECT IdPracownika FROM Pracownicy WHERE Login='$login'");
);

if (!$resultat) throw new Exception($polaczenie->error);

if ($resultat->num_rows > 0) {
    $wszystko_ok = false;
    $_SESSION['e_login'] = "Istnieje już konto z takim loginem!";
}

```

Kolejnym zabezpieczeniem jest forma adresu e-mail, korzystamy tu z gotowej funkcji PHP (filter_var), dzięki czemu eliminujemy podejrzane znaki z podanego adresu e-mail,

a następnie porównujemy czy jest taki sam, jak wpisany. Dzięki temu wiemy, czy takie znaki zostały użyte (Listing 6.13).

Tutaj również sprawdzamy unikalność e-maila. Za jego pomocą również możemy się zalogować, dlatego tylko jeden e-mail może być przypisany do jednego konta (Listing 6.15).

Listing 6.13: Implementacja sprawdzenia poprawności e-maila

```
//Poprawnosc email
$email = $_POST['email'];
$email_safe = filter_var($email, FILTER_SANITIZE_EMAIL);
if(!filter_var($email_safe, FILTER_VALIDATE_EMAIL) || $email_safe != $email){
    $wszystko_ok = false;
    $_SESSION['e_email'] = "Podaj poprawny adres e-mail";
}
```

Listing 6.14: Implementacja sprawdzenia czy e-mail istnieje w bazie danych

```
//czy email istnieje
$resultat = $polaczenie->query("SELECT IdPracownika FROM Pracownicy WHERE Email='$email'");

if(!$resultat) throw new Exception($polaczenie->error);

if($resultat->num_rows > 0){
    $wszystko_ok = false;
    $_SESSION['e_email'] = "Istnieje już konto z takim adresem e-mail!";
}
```

W przypadku hasła musimy sprawdzić, czy ma ono określoną liczbę znaków, u nas od 3 do 20. Oprócz tego, ponieważ pracownik musi potwierdzić hasło i wpisać je dwa razy - trzeba sprawdzić, czy podane hasła są takie same. W zależności od tego, jaki jest błąd - system wyświetli odpowiedni komunikat, tak jak w poprzednich przypadkach. Na Listingu 6.15 znajduje się implementacja sprawdzenia poprawności wpisanego hasła.

Listing 6.15: Implementacja sprawdzenia poprawności hasła

```
//Poprawność hasła
$haslo = $_POST['haslo'];
$haslo2 = $_POST['haslo2'];
//Długość hasła
if(strlen($login)<3 || strlen($login)>20){
    $wszystko_ok = false;
    $_SESSION['e_haslo'] = "Hasło musi mieć od 3 do 20 znaków!";
}
//Identyczne hasła
if($haslo != $haslo2){
    $wszystko_ok = false;
    $_SESSION['e_haslo'] = "Podane hasła nie są identyczne!";
}
```

Dla bezpieczeństwa zostało także użyte haszowanie haseł pracowników (Listing 6.16), aby nikt nie miał do nich bezpośredniego dostępu - w tym administrator. Dzięki temu,

nawet po włamaniu do bazy danych, nikt nie odkryje haseł pracowników. W formularzu zmienna 'hasło' jest także ustawiona jako 'password', dzięki czemu przy wpisywaniu widoczne są jedynie kropki (nikt nie podejrzy hasła przy wpisywaniu). Taka zmienna występuje również przy logowaniu. Zahasowane hasła w tabeli 'Pracownicy' przedstawione są na Tabeli 6.1.

Listing 6.16: Haszowanie hasła

```
$haslo_hash = password_hash($haslo, PASSWORD_DEFAULT);
```

Tabela 6.1: Zahasowane hasła w tabeli 'Pracownicy'

Imię	Hasło
Malgorzata	\$2y\$10\$kr6pkXx6bUafLWVn1Dc/3uenjDT/YxNmJyhBCU1opQ8...
Monika	\$2y\$10\$V1TXMzMwHFfVBXkrVpKcx.r3erJbUBL.YRsK2X3d0Ez...
Krzysiu	\$2y\$10\$0/Pyggj6SB5gUACc25ix6OP/pqBbS/M9KqDAI54alUQ...

Ostatnim elementem bezpieczeństwa przy rejestracji jest akceptacja regulaminu. Dzięki temu eliminujemy konta tworzone przez sztuczne algorytmy, a także sprawdzamy, że dany użytkownik zgadza się na akceptację naszego regulaminu. Bez zaznaczenia tego checkboxa rejestracja nie zostanie ukończona. Implementacja akceptacji regulaminu znajduje się na Listingu 6.17.

Listing 6.17: Implementacja sprawdzenia czy regulamin został zaakceptowany

```
if (!isset($_POST['regulamin'])) {  
    $wszystko_ok = false;  
    $_SESSION['e_regulamin'] = "Zaakceptuj regulamin!";  
}
```

Rozdział 7

Podsumowanie

Stworzony system jest intuicyjny w użytkowaniu, a także posiada wszystkie zaplanowane funkcjonalności. Aplikacja ma estetyczny i prosty wygląd kafelkowy, który sprawia, że poruszanie się po niej jest proste i przyjemne. Dzięki planowaniu spotkań oraz tworzeniu notatek można w łatwy sposób zorganizować codzienny grafik. Dzięki notatkom, które można przypisywać do konkretnej daty, można mieć zapisane ważne rzeczy do zrobienia danego dnia, które niekoniecznie powiązane są ze spotkaniami.

Aplikacja została wykonana zgodnie z ustalonymi założeniami. Projekt spełnia postawione wymagania oraz przeszedł pozytywnie zaplanowane testy, które miały na celu sprawdzić poprawność działania aplikacji do planowania systemów biznesowych. Zaimplementowane funkcje działają zgodnie z oczekiwaniami, w tym planowania spotkań oraz notatek, a także zarządzanie nimi. Wybrane zabezpieczenia zarówno na poziomie relacyjnej bazy danych, a także aplikacji spełniają swoją rolę. Poprawność działania aplikacji, bazy danych oraz ich połączenia zostały sprawdzone przy pomocy testów, których wynik jest pozytywny.

System w przyszłości można rozwinąć tak, aby był maksymalnie funkcjonalny dla pracowników firmy. Można by poszerzyć uprawnienia prezesa firmy, który mógłby dodawać spotkania swoim pracownikom, a także pozwolić jego asystentce na planowanie spotkań prezesowi. Można także wyłączyć opcję rejestracji, a stworzyć dodawanie pracowników jedynie przez uprawnione osoby z kadr. Dzięki temu aplikacja byłaby jeszcze bardziej spersonalizowana do użytku dla firm. System mógłby także wysyłać powiadomienie e-mail do klienta, które poinformuje go o tym, że ktoś rzeczywiście zaplanował z nim spotkanie na konkretny dzień i godzinę.

System ma duże możliwości rozwoju, ale już teraz jest wystarczająco funkcjonalny, aby znacznie ułatwić i przyspieszyć planowanie spotkań biznesowych.

Bibliografia

- [1] Welling Luke, Thomson Laura, *PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty*. Helion, Gliwice, Wydanie czwarte, 2009.
- [2] Lis Marcin, *PHP i MySQL. Dla każdego*. Helion, Gliwice, Wydanie trzecie, 2017.
- [3] Duckett Jon, *HTML i CSS. Zaprojektuj i zbuduj witrynę WWW*. Helion, Gliwice, 2014

Spis rysunków

4.1	Model konceptualny	9
4.2	Model gotowej bazy danych	9
4.3	Strona główna dla niezalogowanego użytkownika	12
4.4	Strona główna dla zalogowanego pracownika	12
4.5	Panel logowania	13
4.6	Dodawanie spotkania lub klienta	13
4.7	Przeglądanie kalendarza	14
4.8	Dodawanie notatki	14
4.9	Błąd logowania	15
4.10	Błąd loginu	15
4.11	Błąd e-maila	16
4.12	Błąd hasła - podane hasła się różnią	16
4.13	Błąd regulaminu - brak zaakceptowania	16
4.14	Diagram przypadków użycia	17
6.1	Dodawanie spotkania lub klienta	25
6.2	Wybór klienta z listy rozwijanej	25
6.3	Widok miesiąca po wybraniu konkretnej notatki	26
6.4	Panel edycji i usuwania notatki	26
6.5	Rejestracja nowego użytkownika	27
6.6	Ekran po udanej rejestracji	27
6.7	Widok bieżącego dnia	28
6.8	Widok bieżącego miesiąca z zaznaczoną aktualnie datą	28
6.9	Widok bieżącego roku	29
6.10	Widok wybranego dnia	30
6.11	Widok po dwukrotnym naciśnięciu strzałki przy nazwie miesiąca w prawo	31

Spis listingów

5.1	Stworzenie tabeli spotkań	19
5.2	Stworzenie tabeli notatek	19
5.3	Stworzenie tabeli klientów	19
5.4	Stworzenie tabeli pracowników	19
5.5	Tworzenie indeksów dla tabel	20
5.6	Ograniczenia tabel	20
5.7	Automatyczne numerowanie indeksów	21
5.8	Nadanie uprawnień w całej bazie danych dla administratora	21
5.9	Przykładowe uprawnienia do tabel Notatki i Klienci dla pracownika	21
5.10	Przykład dodawania klienta	22
5.11	Wyświetlanie tabeli 'Klienci'	22
5.12	Wyświetlanie tabeli 'Notatki' dla pracownika o indeksie 4	22
5.13	Wyświetlanie wybranych kolumn z tabeli 'Pracownicy'	22
6.1	Połączenie z bazą danych	29
6.2	Wnętrze pliku 'connect.php'	30
6.3	Wybranie klientów o danym adresie e-mail (zapytanie SQL)	30
6.4	Dodanie klienta do bazy danych (zapytanie SQL)	30
6.5	Usunięcie wybranej notatki (zapytanie SQL)	30
6.6	Pobranie spotkań z danego dnia i wyświetlenie ich	31
6.7	Pobranie notatek z danego dnia i wyświetlenie ich	31
6.8	Implementacja przesuwania miesiąca	32
6.9	Zmienna sesyjna - sprawdzenie	33
6.10	Implementacja formularza do rejestracji pracownika	33
6.11	Implementacja sprawdzenia poprawności loginu	34
6.12	Implementacja sprawdzenia czy login istnieje w bazie danych	34
6.13	Implementacja sprawdzenia poprawności e-maila	35
6.14	Implementacja sprawdzenia czy e-mail istnieje w bazie danych	35
6.15	Implementacja sprawdzenia poprawności hasła	35
6.16	Haszowanie hasła	36
6.17	Implementacja sprawdzenia czy regulamin został zaakceptowany	36

Spis tabel

4.1	Tabela pracowników	10
4.2	Tabela klientów	10
4.3	Tabela notatek	10
4.4	Tabela spotkań	10
4.5	Uprawnienia użytkowników do poszczególnych tabel w bazie danych	10
5.1	Wynik zapytań na tabeli 'Klienci'	22
5.2	Tabela 'Notatki' dla pracownika o IdPracownika='4'	22
5.3	Wynik zapytania 'SELECT' na tablicy 'Pracownicy'	22
6.1	Zahaszowane hasła w tabeli 'Pracownicy'	36