

Premia APP – Dokumentacja Techniczna

Konopacki Aleks, Lidia Sobiech

1. Założenia Projektowanej Aplikacji

Aplikacja ma na celu usprawnienie procesu rozliczania premii wewnątrz organizacji. Premia stanowi x% wartości każdej faktury wystawionej na rzecz klienta za:

- Licencje
- Zmiany wprowadzane w systemie

2. Grupa Docelowa

Aplikacja skierowana jest do następujących grup użytkowników:

- **Pracownicy działów wsparcia i wdrożeń**, którzy rozliczają premie zgodnie z określonymi zasadami.
- **Menadżerowie działów**, odpowiedzialni za weryfikację dokumentów i przyznawanie premii.
- **Dyrektor**, który zatwierdza przyznane premie.
- **Dział HR**, który przygotowuje niezbędne dokumenty i wypłaca premie.

3. Opis Funkcjonalności

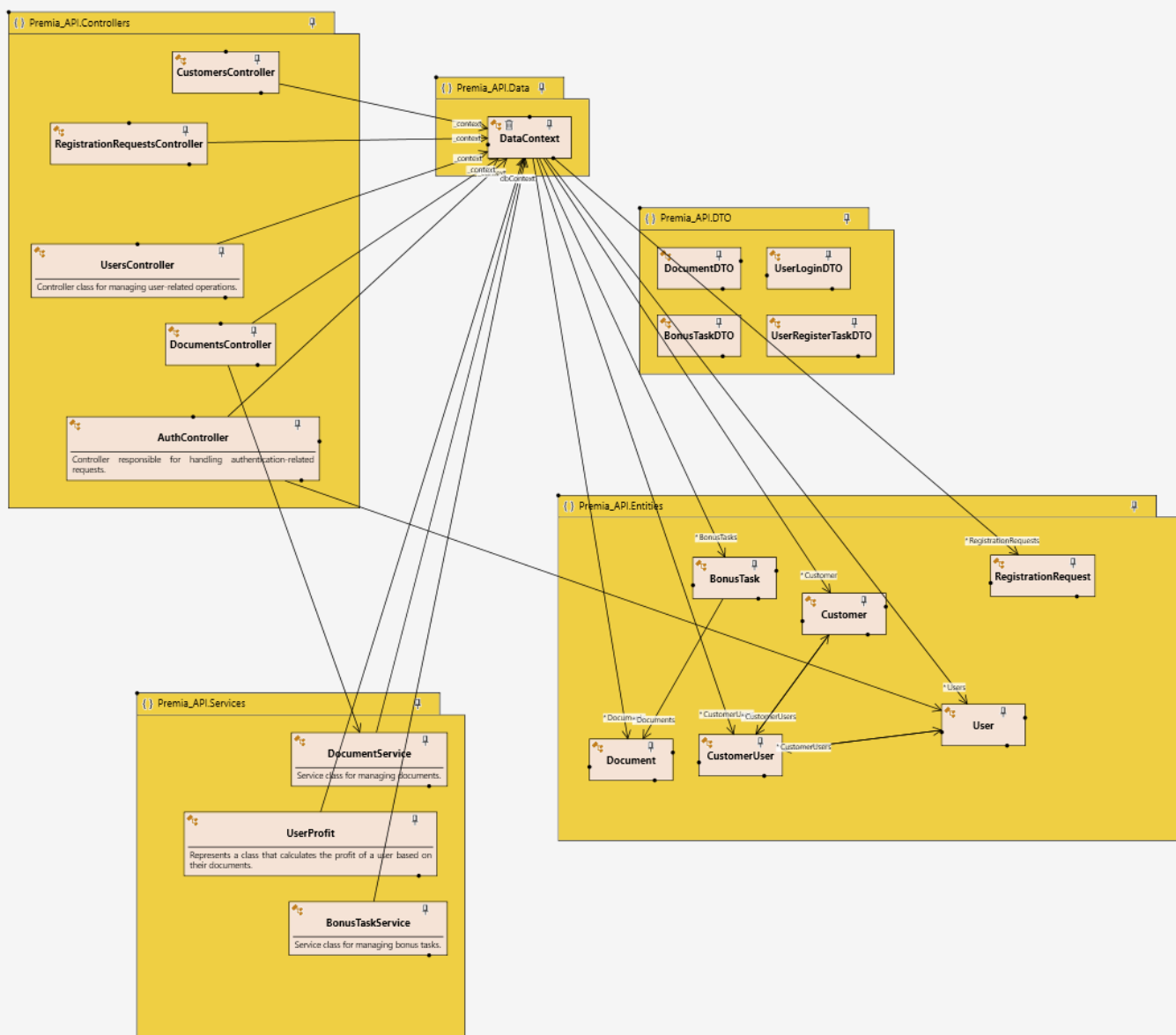
Szczegółowy opis funkcjonalności aplikacji dostępny jest pod adresem:

<https://drive.google.com/file/d/1LT4u7BJNK3BjWmvJJFg2ReL-Nh9LXQbm/view?usp=sharing>

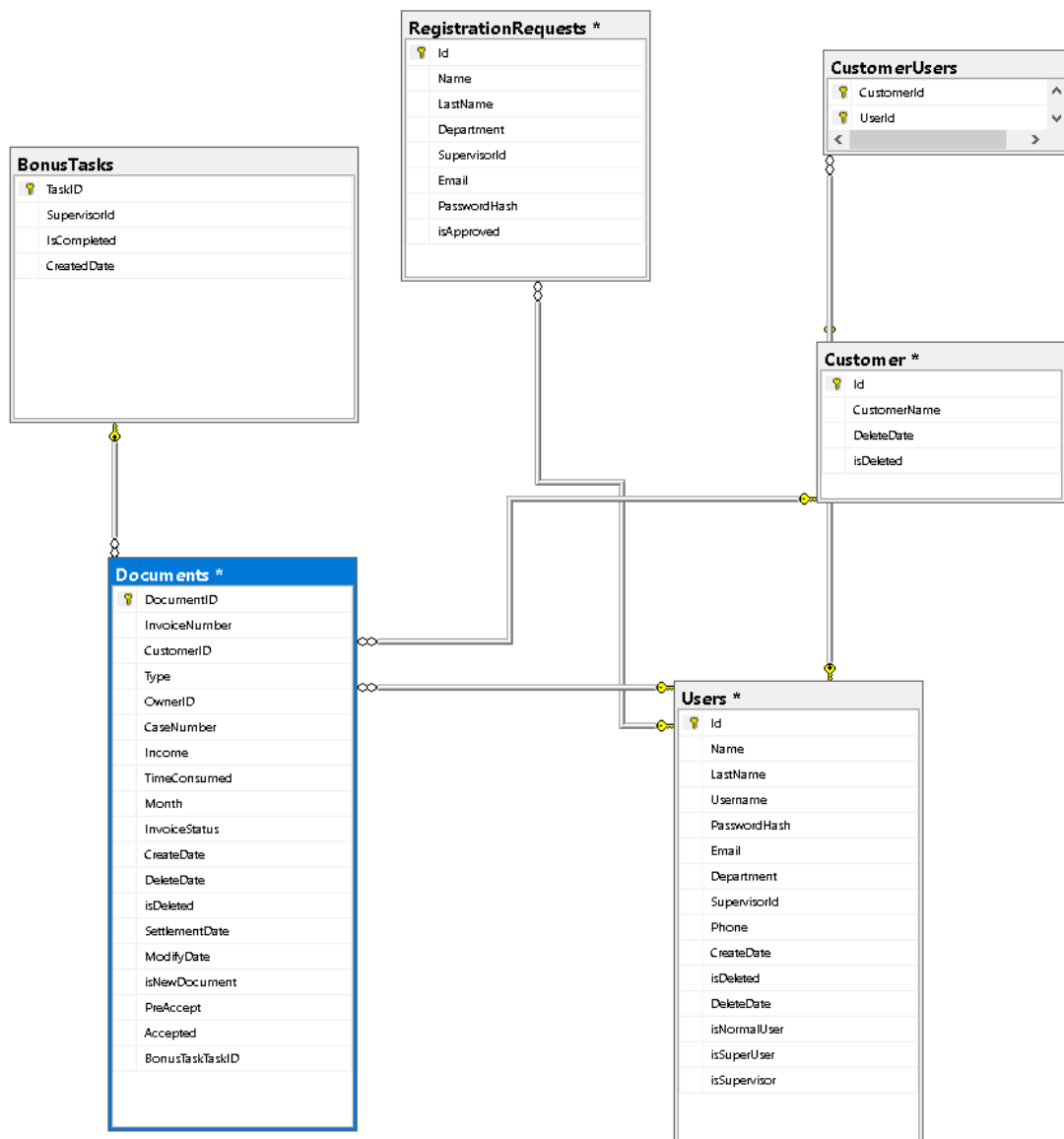
4. Architektura Systemu

4.1 Backend

- Diagram klas



- Schemat bazy danych



- Schemat metod API / Kontrolery

Auth		^
POST	/api/Auth/register	▼
POST	/api/Auth/login	▼
Documents		^
GET	/api/Documents	▼
POST	/api/Documents	▼
GET	/api/Documents/{id}	▼
PUT	/api/Documents/{id}	▼
DELETE	/api/Documents/{id}	▼
POST	/api/Documents/accept/{id}	▼
GET	/api/Documents/supervisor/{id}	▼
RegistrationRequests		^
GET	/api/RegistrationRequests	▼
POST	/api/RegistrationRequests	▼
PUT	/api/RegistrationRequests/{id}	▼
Users		^
GET	/api/Users	▼
POST	/api/Users	▼
GET	/api/Users/{id}	▼
PUT	/api/Users/{id}	▼
DELETE	/api/Users/{id}	▼
GET	/api/Users/supervisors/names	▼

- **Auth:** Kontroler do logowania i rejestracji, z możliwością utworzenia konta administratora.
- **Documents:** Kontroler z podstawowymi operacjami CRUD oraz funkcjonalnością tworzenia zadań dla przełożonych (obecnie niezaimplementowaną po stronie frontendu).
- **Registration Request:** Kontrolery do zarządzania zadaniami, ich pobierania i akceptowania.
- **Users:** Kontroler z operacjami CRUD oraz funkcją zwracania informacji o przełożonych.

4.2 Instrukcja Pierwszego Uruchomienia

Wersja deweloperska aplikacji dostępna jest pod adresem:

<https://premiafront.z36.web.core.windows.net/>

Pliki konfiguracyjne wraz z hasłami zostały udostępnione na platformie Moodle.

4.3 System Użytkowników

Każdy użytkownik posiada jedną z czterech ról:

1. Normalny użytkownik (isNormalUser)
2. Superużytkownik (isSuperUser)
3. Przełożony (isSupervisor)
4. Użytkownik bez konta

Uwaga: Dostęp do systemu mają tylko zalogowani użytkownicy. Osoby bez konta mogą jedynie się zarejestrować lub skorzystać ze strony zapewniającej punkty. Widok interfejsu dostosowuje się do roli użytkownika, a parametry są przekazywane do frontendu za pomocą tokenów JWT.

5. Frontend

5.1 Serwisy

- **Auth:** Odpowiedzialny za logowanie.
- **Customer:** Pobiera dane klientów.
- **GetSupervisor:** Pobiera informacje o przełożonych.
- **LoaderService:** Obsługuje przeładowywanie strony.
- **RegistrationService:** Zarządza zadaniami.
- **User-Store:** Przechowuje tokeny i dane z tokenów.
- **UserService:** Zarządza użytkownikami.

5.2 Modele

Przechowują dane aplikacji.

5.3 Interceptory

- **Api Interceptor:** Umożliwia zmianę linków w zależności od środowiska (produkcyjne/deweloperskie).
- **Loading Interceptor:** Obsługuje animację przeładowywania.
- **Token Interceptor:** Odpowiada za uwierzytelnianie i autoryzację.

5.4 Routing

Odpowiada za nawigację w aplikacji.

5.5 Guard

- **AuthGuard:** Kontroluje dostęp do elementów strony.

6. Funkcjonalności

Jedną z kluczowych funkcjonalności jest możliwość przesyłania zadań między użytkownikami. Zadania są tworzone po wywołaniu odpowiedniego endpointu, a następnie pobierane przez frontend i archiwizowane po wykonaniu.

7. Technologie

7.1 Backend

- Framework: .NET (wersja 8)
- ORM: Entity Framework Core
- Baza danych: Microsoft SQL Server (MSSQL)

7.2 Frontend

- Framework: Angular (wersja 18)
- Biblioteki: Angular Material (opcjonalnie), NgRx (opcjonalnie), HttpClient, RxJS, Bootstrap
- Inne: HTML, CSS