**Specyfikacja Implementacji eAd v.1.0**

Spis Tresci

Inhaltsverzeichnis

[1.Architektura systemu 4](#_Toc490953904)

[1.1 Ogolny zarys systemu 4](#_Toc490953905)

[1.2 Schemat BE 7](#_Toc490953906)

[1.3 Schemat FE 8](#_Toc490953907)

[2.BE 9](#_Toc490953908)

[2.1 Zalozenia, standardy, wykorzystane technologie i metodologie 9](#_Toc490953909)

[2.2 Zarzadzanie sesja 9](#_Toc490953910)

[2.3 Polaczenie do bazy danych 9](#_Toc490953911)

[2.3 Komunikacja z FE (Kontrolery) 9](#_Toc490953912)

[2.4 Eteczka.main 9](#_Toc490953913)

[3.FE 10](#_Toc490953914)

[4. Deployment (Uruchomienie programu na serwerze) 11](#_Toc490953915)

[5. Specyfikacja BE 15](#_Toc490953916)

[5.1. Widok Glowny 15](#_Toc490953917)

[5.2 Pracownicy 16](#_Toc490953918)

[5.3 Wprowadzanie pliku 17](#_Toc490953919)

[5.4 Administrowanie wydzialy 18](#_Toc490953920)

[5.5 Administrowanie rejony 19](#_Toc490953921)

[5.6 Administrowanie podwydzialy 20](#_Toc490953922)

[5.7 Administrowanie konta5 21](#_Toc490953923)

[5.8 Administrowanie funkcje admina uzytkownicy 22](#_Toc490953924)

[5.9 Administrowanie funkcje admina rodzaje dokumentow 23](#_Toc490953925)

[5.10 Administrowanie firma 23](#_Toc490953926)

[6. Specyfikacja FE 24](#_Toc490953927)

[6.1. Widok Glowny 24](#_Toc490953928)

[6.2 Pracownicy 25](#_Toc490953929)

[6.3 Wprowadzanie pliku 26](#_Toc490953930)

[6.4 Administrowanie wydzialy 27](#_Toc490953931)

[6.5 Administrowanie rejony 27](#_Toc490953932)

[6.6 Administrowanie podwydzialy 28](#_Toc490953933)

[6.7 Administrowanie konta5 28](#_Toc490953934)

[6.8 Administrowanie funkcje admina uzytkownicy 29](#_Toc490953935)

[6.9 Administrowanie funkcje admina rodzaje dokumentow 30](#_Toc490953936)

[6.10 Administrowanie firma 31](#_Toc490953937)

[7. Podsumowanie 31](#_Toc490953938)

# 1.Architektura systemu

## 1.1 Ogolny zarys systemu

System eAd (Eteczka) jest aplikacja typu WebApplication. Modelem architektonicznym jest MVC (Model – View – Controller). Oznacza to, ze Model (Baza danych, XML, eteczka.main) posiada pewien stan. Stan ten ulega zmianie tylko poprzez dzialania kontrolera (BE – C#). Kontroler odczytuje i zapisuje stan, sam w sobie nie posiadajac zadnej pamieci (zmiennych globalnych, danych „w pamieci“). View – widok – na rzadanie uzytkownika (klikniecie przycisku, wybranie opcji menu) komunikuje sie z kontrolerem i prosi o aktualny stan danych. Zaleta tego modelu jest to, ze wielu uzytkownikow moze rownoczesnie korzystac z systemu i dostac spojne dane. Wada – troszeczke trudniejsza implementacja.

|  |
| --- |
|  |
| Rys.1 Schemat ogolny systemu |

Nasz system jest aplikacja typu klient-serwer. Serwerem calego systemu jest komputer z wgranym systemem operacyjnym Windows lub Windows Server z zainstalowanym kontenerem IIS **(**[**https://pl.wikipedia.org/wiki/Internet\_Information\_Services**](https://pl.wikipedia.org/wiki/Internet_Information_Services)**)**. IIS jest serwerem dla eAd, windows serwer dla calej naszej platformy (eAd + Fox). Gdy uruchamiamy eAd w Visual Studio, uruchamiamy tak naprawde mini-wersje IIS – IIS Express, mniejsza wersje serwera do developmentu (Rys.2)

|  |
| --- |
| d:\Usr\Tokarz\private\eteczka.rel\eteczka\Design\Obrazy\iis express.png |
| Rys. Nr 2 – IIS Express |

Na rys. 1 widac glowne komponenty skladowe systemu oraz sposob i protokol (jezyk) komunikacji miedzy komponentami. Czesc eteczki dzialajacej na serwerze IIS konfigurujemy za pomoca glownego pliku konfiguracyjnego Web.config. Znajduje sie on w projekcie Eteczka

|  |
| --- |
|  |

Plik ten jest konfiguracja naszej aplikacji. Zawiera definicje konfiguracyjne serwera i ustawienia polaczen BE-FE. Dodatkowo zawiera parametry systemu, ktore ustawiamy sami, takie jak connection string czy nazwa zmiennej srodowiskowej do katalogu roboczego eteczki.

Baza danych wybrana do systemu to PostgreSQL i serwer bazy danych zarzadzany przez aplikacje pgAdmin. Serwer ten dziala w tle (jako proces)

|  |
| --- |
|  |

Domyslnie jest dostepny na hoscie lokalnym pod numerem portu 5432 (localhost:5432). Uruchamia sie przy starcie systemu. Do zdefiniowania przez nas pozostaja dane dostepowe u klienta.

## 1.2 Schemat BE

|  |
| --- |
|  |
| Rys. Nr X Schemat Back End Eteczki |

BE sklada sie obecnie z 2 glownych projektow : BE (Backend) i DB (Database). DB jest odpowiedzialny za warstwe polaczenia z baza danych. BE korzysta z DB za pomoca obiektow DAO (Data-Access Object). Idea takiego obiektu jest udostepnienie klasy, ktora w jezyku obiektowym udostepni jakis zasob bazy danych. Przyklad:

Klasa KatLoginDAO, metoda

*public KatLoginy WczytajPracownikaPoNazwieIHasle(string username, string password)*

Zwraca nam ona obiekt z bazy danych dla danego username i password. Nie interesuje nas na tym etapie, jaka to jest baza danych i jak ma wygladac do niej kwerenda. Mozemy obiektowo wywolac logike dostepu do bazy danych. Formatem wymiany danych miedzy BE a DB jest klasa typu DTO (Data-Transfer Object). Jest to obiekt reprezentujacy wynik kwerendy. Czasami jest on widokiem jednej tabeli czasami kilku (gdy Select uzywa operacji Join i laczy kilka tabel). To samo DTO kierowane jest do FE jako dane dostepne dla uzytkownika.

Glowna warstwa komunikacyjna miedzy FE a BE sa klasy typu Kontroler. Sa to zwykle klasy, zakonczone slowem Controller, ktore dziedzicza po klasie bazowej *System.Web.Mvc.Controller* .

Posiadaja metody zwracajace typ *ActionResult.* Z naszego punktu widzeniajest to nic innego jak JSON.

Kontrolery nie powinny posiadac zbyt wiele logiki, ich zadaniem jest odebranie rzadania z FE i oddelegowanie zadan do „Serwisow“. Serwisy zwracaja wartosc, ktora nastepnie kontrolery przekazuja spowrotem do FE w formacie JSON.

Serwisy to glowne miejsce implementacji logiki. Korzystaja one z klas pomocniczych, takich jak Utils, Model, Mappers, Factories, itd. Ich zadaniem jest wywolanie odpowiednich klas pomocniczych w celu pobrania i obrobki danych dla Kontrolera. Lacza sie one rowniez z DAO zeby uzyskac dane z bazy danych.

DAO sluza do pobierania danych z bazy. Korzystaja one z interfejsu *IDbConnectionFactory*

Ktory za pomoca interfejsu IConnection tworzy connection string do naszej bazy i zwraca obiekt IdbConnection ktory moze wykonywac zapytania do bazy danych. W wersji 1.2 bedzie on rowniez wspieral transakcje. Klasy w przestrzeni nazw „Entities“ sa to klasy odpowiadajace 1:1 tabelom w bazie. Ich nazwy pol i typy powinny pokrywac sie z definicjami tabel w SQL.

## 1.3 Schemat FE

|  |
| --- |
|  |

Glownym interfejsem pomiedzy FE a BE jest serwis httpService. Posiada on definicje metod GET i POST do komunikacji z BE. Powinien on stanowic jedyne polaczenie pomiedzy kontrolerami Angulara a BE. Ujednolici to przede wszystkim sposob komunikacji i ulatwi testowanie. Serwis httpService korzysta z serwisu sessionService do zarzadzania sesja uzytkownika. SessionService tworzy i przechowuje sesja na caly czas pracy z eAd oraz zamyka sesje na koniec pracy. Sesja zostaje stworzona w momencie w ktorym uzytkownik zaloguje sie do serwisu i zostaje zamknieta, kiedy nastapi wylogowanie badz zamkniecie przegladarki.

Jedynymi obiektami odnoszacymi sie do struktury DOM bezposrednio, powinny byc dyrektywy, wewnatrz metody link. Kontrolery i serwisy nie powinny odnosic sie do obiektow DOM besposrednio.

# 2.BE

## 2.1 Zalozenia, standardy, wykorzystane technologie i metodologie

BE powstaje w mysl zalozen MVC. Do tworzenia obiektow wykorzystujemy kontener DI (klasa UnityConfig.cs). Kazda klasa powinna zostac przetestowana. Poczatkowym punktem systemu jest klasa Global.asax w projekcie Eteczka. Startuje ona aplikacje oraz konfiguruje wszystkie parametry systemu. Ze wzgledu bezpieczenstwa, zeskanowane pliki znajduja sie w folderze roboczym wskazywanym przez zmienna srodowiskowa EAD\_DIR. Domyslnie powinien byc to katalog C:eteczka.main

## 2.2 Zarzadzanie sesja

Sesja powstaje podczas zalogowania sie do systemu. Odpowiedzalna za to jest klasa *SesjaController.* Sesja powinna miec timeout i po czasie dluzszej bezczynnosci powinno nastapic automatyczne wylogowanie i zamkniecie sesji. Obiektem kontrolujacym sesje jest klasa *StanSesji.*

## 2.3 Polaczenie do bazy danych

Polaczenie do bazy danych wywolujemy za pomoca sterownika ODBC. Jest to metoda wykorzystujaca connection string do nawiazania polaczenia. Dane bazy zdefiniowane sa w pliku Web.config a connection string powstaje w klasie *Connection* po wywolaniu metody

*public string GetConnectionString(IConnectionDetails connectionDetails).*

Klasa opakowujaca cala funkcjonalnosc to *IDbConnectionFactory.* To ona powinna stanowic jedyny sposob uzyskiwania polaczenia do DB.

## 2.3 Komunikacja z FE (Kontrolery)

Framework .NET rozwiazuje za nas problem mapowania FE do BE. Dzieje sie to za naszymi plecami poprzez wykorzystanie tzw. Kontrolerow. Sa to klasy dziedziczace po typie *Controller* i zwracajace typ *ActionResult* do FE. Ich zasada dzialania wspiera interfejs REST. Jego idea polega na pobieraniu rezultatu zapytania do BE na zasadzie zasobu. Rzadanie zasobu to adres url:

*localhost/ead/NaszaKlasaController/NazwaMetody/Parametr*

Tak sformulowane zapytanie w oknie przegladarki powinno zwrocic JSON z danymi danego parametru. Ten adres url zostaje wywolane przez serwis httpService w kodze FE.

## 2.4 Eteczka.main

Kazdy katalog, ktory znajduje sie wewnatrz katalogu inetpub/wwwroot jest automatycznie dostepny „w sieci“. Trzeba zabezpieczac kazdy zasob na serwerze przed niepowolanym dostepem. Jest to skomplikowane i narazone na bledy. Dlatego dane klienta powinny znajdowac sie poza serwerem. W naszej architekturze, miejsce skladowania plikow i narzedzi pomocniczych pelni katalog eteczka.main. jego struktura w wersji 1.0 wyglada tak:

|  |
| --- |
|  |

Zawiera ona fizyczne kopie skanow, archiwizowane dane, jsony do importu z Systemu Ksiegowego Foxa oraz logi systemowe. Sciezka do tego katalogu wskazywana jest przez zmienna srodowiskowa EAD\_DIR.

# 3.FE

Glownym zalozeniem konstrukcji FE jest tzw. SPA (Single Page Application). Nasz framework do implementacji tego zalozenia to AngularJS (w wersji przed 2.0). Do layoutu i tworzenia responsywnej warstwy prezentacyjnej korzystamy z HTML5 oraz CSS3 (rezygnujemy z frameworka Bootstrap). Glownym plikiem konfiguracyjnym FE jest plik *app.js*, definiujacy zaleznosci i moduly calego systemu. Miesci sie on wewnatrz projektu Eteczka w podkatalogu app.

|  |
| --- |
|  |

Definiujemy w nim zaleznosci naszego glownego modulu FE oraz definicje i sciezki widokow.

Dodatkowo korzystamy z frameworka JQuery.

# 4. Deployment (Uruchomienie programu na serwerze)

Serwer IIS zlokalizowany jest w systemach Windows Professional i Windows Server na dysku c: pod nazwa inetpub (c:/inetpub). Gdy chcemy uruchomic „deployment“ naszej aplikacji, wybieramy w Visual Studio opcje publish (prawy przycisk myszki na projekcie glownym – u nas to jest projekt Eteczka)

|  |
| --- |
| d:\Usr\Tokarz\private\eteczka.rel\eteczka\Design\Obrazy\publish.png |
| d:\Usr\Tokarz\private\eteczka.rel\eteczka\Design\Obrazy\publish dalej.png |

Gdy proces ten wykona sie bez bledow, w katalogu inetpub powstanie (lub uaktualni sie) katalog ead

|  |
| --- |
| d:\Usr\Tokarz\private\eteczka.rel\eteczka\Design\Obrazy\ead nkatalog.png |

W srodku jego struktura wyglada tak:

|  |
| --- |
|  |

Katalog bin posiada skompilowany BE (pliki dll ). Reszta to pliki FE i ustawienia aplikacji (Web.config). Teraz musimy sie upewnic ze IIS jest uruchomiony:

|  |
| --- |
|  |

Teraz nasza aplikacja dostepna jest na serwerze pod adresem:

localhost/ead/index.html

|  |
| --- |
|  |

# 5. Specyfikacja BE

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| Ogolne | - | Szyfrowanie plikow | 16h |
| Ogolne | - | https | 40h |
| Ogolne | - | Skrypty SQL (selecty) | 40h |
| Ogolne | - | Skrypt instalacyjny | 16h |

## 5.1. Widok Glowny

|  |
| --- |
|  |
| |  |  |  |  | | --- | --- | --- | --- | | Grupa | Klasa | Opis | Czasochlonnosc | | - | - | - | - | |
|  |

## 5.2 Pracownicy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZP | PracownicyKontroller | Pobierz wszystkich pracownikow | 0 (zrobione) |
| ZP | PracownicyKontroller | Rozszerz PracownikDTO o miejsca pracy. Dodaj metode PobierzPracownikaDlaId | 4h |
| ZP | PracownicyKontroller | Dodaj metode WyszukajPracownikow(string search) | 8h |
| ZP | PracownicyDAO | Dodaj metode WyszukajPracownikowPoTekscie. Napisz kwerende SQL uzywajac pola %LIKE% | 8h |

## 5.3 Wprowadzanie pliku

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZPlik | PlikiKontroler | Metoda PobierzPlikiDoDodania. Powinna pobierac z bazy informacje o plikach i zwrocic te, ktore maja status do dodania | 8h |
| ZPlik | PlikiDTO | Rozszerz PlikiDTO by pasowaly do widoku (analiza) | 4h |
| ZPlik | PlikiService | Dodaj metode kodujaca i dekodujaca plik pdf przed wywolaniem podgladu | 24h |
| ZPlik | PlikiController | Dodaj Metode DodajZmienPliki ktora zapisze zmiany w DB po kliknieciu przycisku zatwierz | 16h |
| ZPlik | PlikiController | Stworz metode PobierzHistoriePliku | 8h |

## 5.4 Administrowanie wydzialy

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZF | FirmyKontroller | Stworz kontroler. Dodaj metode PobierzWszystkieFirmy | 8h |
| ZF | FirmyKontroller | Stworz metode PobierzWydzialy dla wybranej firmy | 8h |

## 5.5 Administrowanie rejony

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZR | RejonyKontroller | Stworz kontroler. Dodaj metode pobierzWszystkieRejony | 4h |
| ZR | RejonyKontroller | Stworz metode PobierzRejony dla wybranegj firmy | 8h |

## 5.6 Administrowanie podwydzialy

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZPW | WydzialyKontroller | Stworz kontroler. Dodaj metode pobierzWszystkiePodwydzialy | 4h |
| ZPW | RejonyKontroller | Stworz metode PobierzPodwydzialy dla wybranej firmy | 8h |
| ZPW | RejonyDTO | Stworz DTO | 2h |

## 5.7 Administrowanie konta5

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZK5 | KontaKontroller | Stworz kontroler. Dodaj metode pobierzWszystkieKonta | 4h |
| ZK5 | KontaKontroller | Stworz metode PobierzKonta5 dla wybranej firmy | 8h |
| ZKS | KontaDTO | Stworz DTO | 2h |

## 5.8 Administrowanie funkcje admina uzytkownicy

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZU | UsersKontroller | Stworz kontroler. Dodaj metode PobierzUzytkownkow | 4h |

## 5.9 Administrowanie funkcje admina rodzaje dokumentow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZRD | DokumentyKontroller | Stworz kontroler. Dodaj metode PobierzWszystkieRodzajeDokumentow | 4h |
| ZRD | DokumentyKontroller | Stworz RodzajDokumentuDTO | 4h |

## 5.10 Administrowanie firma

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Klasa | Opis | Czasochlonnosc |
| ZAF | FirmaKontroller | Dodaj metode PobierzFirmePoNazwie | 4h |
| ZAF | FirmaKontroller | Dodaj metode DodajAktualizujFirme | 8h |

# 6. Specyfikacja FE

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| Og | Index.html | Stworz nowy layout:  Przenies widok do prawej czesci strony  Dodaj obrazy do katalogu Contents  Zaprojektuj responsywnosc (@media css)  Zdefiniuj kolory (najlepiej sass)  Zdefiniuj buttony i click-eventy  Zdefiniuj animacje  Panel glowny + opcje menu | 40h |
| Og | userIconDirective | Stworz okno rozwijane dla usera w prawym gornym rogu | 8h |

## 6.1. Widok Glowny

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Grupa | Plik | Opis | Czasochlonnosc |
| WG | mainView | Pokaz logo (moze animacja?) | 2h |
| KA | fileCart | Koszyk akt – html + click | 4h |

## 6.2 Pracownicy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Grupa | Plik | Opis | Czasochlonnosc |
| WP | employeesDirective | Layout + service call | 16h |
| WP | Dialog | Dialogi dodaj / edytuj usun | 24h |

## 6.3 Wprowadzanie pliku

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WDP | Dialog | Podglad pliku (juz jest, dodaj tylko haslo) | 4h |
| WDP | Dialog | Widok panelu dodaj pliki | 4h |
| WDP | Dialog | Widok panelu dodane pliki | 4h |
| WDP | Dialog | Click – eventy + dodanie komentarza + zatwierdz | 16h |

## 6.4 Administrowanie wydzialy

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAW | General | Widok rozszerzalnych opcji i podopcji | 16h |
| WAW | Dialog | Okna modalne dodaj edytuj usun | 8h |

## 6.5 Administrowanie rejony

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAR | Dialog | Jak 7.4, inne url httpService | 4h |

## 6.6 Administrowanie podwydzialy

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAPW | Dialog | Jak 7.4, inne url httpService | 4h |

## 6.7 Administrowanie konta5

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAK5 | Dialog | Jak 7.4, inne url httpService | 4h |

## 6.8 Administrowanie funkcje admina uzytkownicy

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAFA | Dialog | Jak 7.4, inne url httpService | 4h |

## 6.9 Administrowanie funkcje admina rodzaje dokumentow

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAFA | Dialog | Jak 7.4, inne url httpService | 4h |

## 6.10 Administrowanie firma

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Plik | Opis | Czasochlonnosc |
| WAF | Dialog | Widok firmy | 4h |
| WAF | Dialog | Opcje edytuj (pola klikalne), pola wymagane | 8h |

# 7. Podsumowanie

Laczna ilosc pracy do wykonania : **442h (55 dni po 8h pracy, czyli w idealnych warunkach i pelnej pracy kazdego z nas, zrobilismy calosc zadania w 14dni).** Szacunki robilem pesymistycznie, biorac pod uwage czas potrzebny na nauke i dostosowanie do systemu. Jak wiadomo, dziewiec kobiet nie urodzi dziecka w miesiac, wiec jest to tylko wielkosc pogladowa, sugerujaca skomplikowanie i trudnosc zadania. Im wiecej zadan bedziemy robic, tym szybciej i latwiej zrobimy kolejne, wiec biorac pod uwage termin 01.09.2017 uwazam go za realistyczny! Wazne, zebysmy nie dodali juz zbyt wiele funkcjonalnosci, bo nie damy rady osiagnac terminu.

Zadanie do wykonania:

Wypelnij tabele wedlug siebie:

|  |  |
| --- | --- |
| Programista | Deklarowany Czas dostepny (8h dziennie, 40h tygodniowo) |
| Paszczak |  |
| Michal |  |
| Burqin | 48 h (zakładam ok 6h dziennie w weekendy), może cos da rade więcej po godzinach, ale nie chcę deklarować bo nie wiem jak będzie |
| Maciek | 120h (wszystkie weekendy to ok 96h ale wiadomo ze nie cale 8h sie znajdzie, czas po godzinach i czesciowo w pracy) |