|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dekoracyjny | | | | |
|  |  |  | |  |
| STUDIUM WYKONALNOŚCI ORAZ WYMAGANIA KLIENTA wraz z dokumentacją techniczną | | |
| PAWEŁ PALUCH | | PAWELPOLSKAPOLAK@WINDOWSLIVE.COM |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dekoracyjny | | | | |
|  |  |  | |  | |
|  | DO CZEGO ma służyć ten system?  System e-RWSS ma za zadanie połączyć środowiska rozproszone i połączyć je w ramach jednej aplikacji. Ma być ułatwieniem za równo dla członków RWSSu, pracowników dziekanatu, studentów i podmiotów zewnętrznych. Inteligentna struktura oraz zastosowane rozwiązania sieciowo-webowe mają za zadanie uatrakcyjnić wizerunek RWSSu, natomiast główną misją jest ujednolicenie całej komunikacji oraz skomulowanie całej struktury do jednej klastrowanej aplikacji. | |  | |

|  |  |
| --- | --- |
|  |  |
| opis głównych modułów systemue-Student Podsystem do obsługi zdalnej studenta. Każdy student ma możliwość zaproponowania nowych działań, sprawdzenia dostepności RWSS oraz zapewnienie bezpiecznej komunikacji z organem Samorządu Studentów działającym na Wydziale. Cały moduł ma mieć zintegrowane możliwości logowania za pomocą SSO uczelnianego oraz Węzła Krajowego. e-Dziekanat Moduł do zarządzania oraz odpowiedzi na pisma RWSSu. Każda sprawa ma mieć unikatowy Znak sprawy generowany w zależności od kodu zawartego w odpowiedniej kategoryzacji spraw. Moduł ma mieć możliwość dopisanie użytkowników do poszczególnych ról oraz pozwala na bezpieczną wymianę dokumentacji między jednostkami organizacyjnymi. | |
| Obraz przedstawiający wiele żarówek z tylko jedną zapaloną | |  |
|  | |  |
| e-Członek RWSSu Moduł przeznaczony do obsługi bieżącej studenta w związku z bezpośrednim kontaktem studenta  z jednostką. Członek ma mieć możliwość możliwość rejestracji czasu w systemie zmianowym  z wykorzystaniem bieżącej lokalizacji, obsługę studenta na podstawie SSO uczelnianego, ma mieć dostęp do bezpiecznego komunikatora oraz ma mieć dostęp do przypisania sobie poszczególnych zadań w ramach działań RWSSu oraz organizowanych wydarzeń. e-Przewodniczący RWSSu Moduł administracyjny. Ma mieć możliwość nadawania poszczególnych uprawnień specjalnych poszczególnym członkom RWSSu. Ma mieć możliwość wysyłania e-maili do starostów, których lista ma być importowana z pliku csv lub xlsx. Lista starostów ma być stała i importowana tylko raz w miesiącu. Moduł ma mieć zadeklarowany konkretne wzory pism które są ustandaryzowane przez RWSS oraz Uczelnię jak i Sekcję Magazynu, Kanclerza oraz Zespół Kontrolingu. Ma mieć możliwość wprowadzania dyżurów członków w poszczególnych dniach oraz wprowadzenie do systemu wydarzeń i generowanie na tej podstawie poszcególnych pism do innych jednostek uczelni. | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| Rozwinięcie fukncionalności modułów Każdy moduł ma być łatwo skalowany oraz ma być zaprojektowany w sposób jednolity i klarowny. Ma mieć zaprojektowany układ logiczny oraz ma być wyprodukowany w najlepszych standardach programistycznych.  Z wykorzystaniem odpowiedniej bazy danych oraz odpowiednich frameworków moduły będą rozwijalne oraz cała ich budowa będzie zrozumiała dla każdego programisty. Funkcionalności krytyczne Lista funkcjonalności krytycznych:   1. Logowanie przez SSO uczelniane 2. Logowanie przez Węzeł Krajowy 3. Logowanie awaryjne 4. U2FA 5. Nadanie uprawnień i ich kontrola 6. E-Moduł Członka RWSSu 7. E-moduł Przewodniczącego RWSS   Funkcjonalności krytyczne mają mieć zdolność funkcjonalną na koniec pierwszej fazy wydania oprogramowania. Te funkcjonalności muszą mieć zapewnione SLA na poziomie co najmniej 95,99%. W razie awarii poszczególnych dwóch podstawowych logowań powinno się aktywować logowanie awaryjne. Do logowania awaryjnego powinien zawsze mieć dostęp przez specjalną serwis Przewodniczący RWSSu oraz każdy kto ma odpowiednio nadane uprawnienia. System w razie wątpliwości co do zasadności odpowiednich uprawnień do danego użytkownika ma w systemie administratora wyświetlić monit i pozwolić mu na decyzję, natomiast w przypadku braku akcji przez co najmniej 7 dni roboczych system sam ma odebrać nadmiarowe uprawnienia. Kontrola odpowiednich uprawnień powinna się odbywać raz na dwa tygodnie. | | | |
|  |  | |  |
| Celowość projektu  Rozwinięcie możliwości organizacyjne oraz możliwość bezpiecznej komunikacji poszczególnych jednostek organizacyjnych. | Bezpieczeństwo  Całośc ma być oparta na podstawie logowania do SSO uczelnianego oraz przez Węzeł Krajowy i umożliwiać podpisanie pisma zdalnie za pomocą metody U2F | | Monitoring wydajności  W module e-Przewodniczącego mają być widoczne poszczególne wskaźniki efektywności każdego z członków RWSSu |

|  |  |
| --- | --- |
|  |  |
| wymagania techniczne System powinien mieć klastrowy charakter oraz posiadać redundancję. Repilkacji bezpieczeństwa powinna podlegać co najmniej raz dziennie baza danych oraz folder z pismami wygenerowanymi oraz folder z pismami wysłanymi do poszczególnych jednostek organizacyjnych.  Aplikacja ma być wersją webowej aplikacji klastrowej, która będzie miała łatwe do rozszerzenia moduły przez przyszłe wydania. Przypisane zadania do konkretnych osób w zespole Przemysław Kasprzycki – Główny Specialista ds. Dev | |

Zadania:

1. Wybór odpowiednich języków oraz Frameworków do pracy z aplikacja – Backend
2. Wybór odpowiednich języków oraz Frameworków do pracy z aplikacja – Frontend
3. Stworzenie diagramu klas
4. Stworzenie API
5. Stworzenie diagramu stanów
6. Zapoznanie z metodami wymiany danych Frontend-Backend
7. Współtworzenie dokumentu projektowego

Adrian Nafalski – Specjalista ds. Dev

Zadania:

1. Opracowanie wymagań i zalet i wad bazy danych: Oracle
2. Opracowanie wymagań i zalet i wad bazy danych: MySql
3. Opracowanie wymagań i zalet i wad bazy danych: MariaDB
4. Stworzenie diagramu czynności
5. Stworzenie diagramu przypadków użycia
6. Współtworzenie dokumentu projektowego

Mateusz Smyk – Specjalista ds. DevOps

Zadania:

1. Opracowanie koncepcji bazy danych
2. Zapoznanie się z silnikiem bazy danych: MS SQL
3. Zapoznanie się z silnikiem bazy danych: Postgress
4. Stworzenie interfejsu GUI
5. Opracowanie dokumentacji dot. Serwerów operacyjnych i środowiska uruchomieniowego
6. Stworzenie loga
7. Współtworzenie dokumentu projektowego

Dominik Adamczuk – Specjalista ds. testów

Zadania:

1. Opracowanie koncepcji testów manualnych
2. Opracowanie koncepcji testów automatycznych
3. Stworzenie scenariuszy testowych dla modułu logowania
4. Stworzenie scenariuszy testowych dla modułu e-Przewodniczący
5. Stworzenie scenariuszy testowych dla API
6. Stworzenie scenariuszy testowych dla modułu e-Student
7. Stworzenie scenariuszy testowych dla modułu e-Dziekanat

Paweł Paluch – Product Manager

Zadania:

1. Stworzenie środowiska Jira
2. Dodanie członków zespołu
3. Wybór tematyki projektu
4. Utworzenie Jira Workflow
5. Zapoznanie z technologiami proponowanymi do użycia w projekcie
6. Konfiguracja pól zgłoszeń
7. Architektura serwera
8. Utworzenie repozytorium
9. Tworzenie dokumentu opisującego projekt (PDF)
10. Planowanie pracy zespołowej
11. Tworzenie planu przepływu informacji
12. Utworzenie spisu dokumentów
13. Uzupełnienie opisu tasków
14. Przygotowanie planu pracy na następne miesiące
15. Tworzenie opisu Epików i Mileston’ów. Dodanie etykiet

|  |
| --- |
|  |
| opis głównych technologii wykonania aplikacji webowej **FIGMA**  Jest to aplikacja do projektowania stron internetowych(np. interfejsu użytkownika). Jest darmowym oprogramowaniem graficznym do profesjonalnych zastosowań. Łączy w sobie najlepsze narzędzia i rozwiązania znane z komercyjnych pakietów graficznych. Umożliwia tworzenie i edycję zarówno rastrowej, jak i wektorowej grafiki. Ma przejrzysty i intuicyjny interfejs, więc nie ma problemu z użytkowaniem tej aplikacji.  Wybraliśmy Figmę ze względu na:  • Łatwość współpracy  • Brak konieczności instalowania specjalnego oprogramowania  • Możliwość używania aplikacji w dowolnym miejscu, na dowolnym urządzeniu  **POSTMAN**  Postman to narzędzie do testowania i debugowania interfejsów API (Application Programming Interface). Pozwala ono na wysyłanie zapytań HTTP (np. GET, POST, PUT, DELETE) do serwera i otrzymywanie odpowiedzi, co pozwala na testowanie działania API i debugowanie błędów. Postman oferuje szereg funkcji, takich jak automatyczne tworzenie i zarządzanie kolekcjami zapytań, możliwość ustawiania nagłówków, parametrów i danych w żądaniach, możliwość importu i eksportu zapytań. Jest to jedno z najbardziej popularnych narzędzi do testowania API i jest często używane przez programistów i testerów oprogramowania  Wybraliśmy Postmana ze względu na:  • Możliwość tworzenia i udostępniania dokumentacji interfejsów API  • Możliwość monitorowania zmian w API  • Bezpłatną wersję  • Dużą bazę wiedzy  **MICROSOFT SQL SERVER**  Microsoft SQL Server to relacyjny system zarządzania bazami danych opracowany przez firmę Microsoft. Oto kilka cech i zalet tego systemu:   1. Skalowalność: Microsoft SQL Server może obsługiwać duże ilości danych i zapewnia możliwość łatwego rozszerzania systemu w przypadku potrzeby zwiększenia pojemności. 2. Bezpieczeństwo: SQL Server oferuje szeroki zakres funkcji bezpieczeństwa, takich jak konta użytkowników, kontrole dostępu do danych, szyfrowanie danych i audytowanie działań. 3. Wysoka dostępność: SQL Server zapewnia wysoką dostępność danych dzięki funkcjom takim jak replikacja, grupowanie dostępności, migawki i awaryjne przywracanie. 4. Integracja z innymi narzędziami Microsoft: SQL Server jest zintegrowany z innymi narzędziami Microsoft, takimi jak Excel, SharePoint i Power BI, co ułatwia tworzenie raportów, analizowanie danych i ich wizualizację. 5. Wydajność: SQL Server oferuje wiele mechanizmów optymalizacji wydajności, takich jak indeksowanie, partycjonowanie tabel i procedury składowane, które pomagają przyspieszyć zapytania do bazy danych. 6. Łatwość użycia: SQL Server zapewnia łatwe tworzenie i zarządzanie bazami danych dzięki intuicyjnemu interfejsowi użytkownika, narzędziom do projektowania schematów i szybkiemu wdrażaniu aplikacji.   Te zalety sprawiają, że Microsoft SQL Server jest popularnym systemem zarządzania bazami danych wśród firm i organizacji na całym świecie.  **C#**  C# (wymawiane "C-sharp") to obiektowy język programowania opracowany przez Microsoft. Oto kilka cech i zalet tego języka:   1. Bezpieczeństwo: C# zapewnia bezpieczeństwo dzięki swojemu systemowi typów i mechanizmom zarządzania pamięcią, które ograniczają ryzyko błędów i awarii. 2. Łatwość nauki: C# jest łatwy do nauczenia się i zrozumienia, co czyni go dobrym wyborem dla początkujących programistów. 3. Przenośność: Kod napisany w C# może być uruchamiany na różnych platformach, takich jak systemy Windows, macOS i Linux. 4. Wysoka wydajność: C# oferuje wydajność zbliżoną do języka C++, dzięki czemu jest popularnym wyborem do tworzenia aplikacji wymagających szybkiej i efektywnej obsługi. 5. Wsparcie dla programowania obiektowego: C# jest językiem programowania obiektowego, co oznacza, że programiści mogą tworzyć i korzystać z obiektów, co ułatwia zarządzanie i organizowanie kodu. 6. Wsparcie dla asynchronicznego programowania: C# zapewnia obsługę programowania asynchronicznego, co umożliwia łatwe tworzenie aplikacji działających równolegle i reagujących na interakcje użytkownika w czasie rzeczywistym. 7. Wsparcie dla rozwijania aplikacji sieciowych: C# oferuje wiele narzędzi i bibliotek, które ułatwiają tworzenie aplikacji sieciowych i obsługę protokołów sieciowych.   Dzięki tym cechom C# jest popularnym językiem programowania wśród firm i organizacji na całym świecie, szczególnie w przypadku tworzenia aplikacji na platformę Windows.  C# jest bardzo dobrze udokumentowanym i intuicyjnym językiem. Jest ciągle on wspierany przez firmę Microsoft, więc jego dokumentacja jest również łatwo dostępna we wszystkich językach. Dodatkowo, zapewnia to również doskonałą współpracę z serwerami Windowsowymi. C# jest też  o wiele zwięźlejszym językiem niż np. Java, co czyni kod w nim napisany prostszy do przeczytania  i zrozumienia. Jest on również bardzo wydajny i szybki, co czyni go idealnym kandydatem na język backendowy aplikacji webowej. C# oferuje nie tylko pisanie backendu aplikacji, ale również frontend, wykorzystując do tego framework Blazor. Efektem tego jest to, że stworzenie całej aplikacji nie wymaga biegłości w kilku językach programowania. Technologia ta jest stale wspierana i rozwijana przez firmę Microsofti jest doskonale udokumentowana. Framework Blazor jest zarówno szybki jak i niezawodny, co czyni go idealnym kandydatem do stworzenia frontendu aplikacji webowej. | |
| ASP .NET  ASP.NET to framework do tworzenia aplikacji internetowych opracowany przez firmę Microsoft. Oto kilka cech i zalet tego frameworka:   1. Wsparcie dla języków programowania: ASP.NET oferuje wsparcie dla wielu języków programowania, takich jak C#, Visual Basic i F#, co pozwala programistom korzystać z preferowanego języka. 2. Wysoka wydajność: ASP.NET oferuje wysoką wydajność dzięki mechanizmom takim jak buforowanie wyjścia, rozwiązanie kodu źródłowego w czasie kompilacji i reużywalne komponenty. 3. Bezpieczeństwo: ASP.NET oferuje wiele funkcji bezpieczeństwa, takich jak autentykacja, autoryzacja, kontrola dostępu i szyfrowanie, które pomagają chronić aplikacje przed atakami hakerskimi. 4. Wsparcie dla programowania obiektowego: ASP.NET jest oparty na programowaniu obiektowym, co ułatwia organizację kodu i pozwala programistom tworzyć reużywalne komponenty. 5. Integrowanie z innymi technologiami Microsoft: ASP.NET jest zintegrowany z innymi technologiami Microsoft, takimi jak Microsoft SQL Server i Microsoft Azure, co ułatwia tworzenie aplikacji internetowych na platformie Microsoft. 6. Łatwość użycia: ASP.NET oferuje łatwy w użyciu interfejs użytkownika i wiele narzędzi deweloperskich, które ułatwiają tworzenie aplikacji internetowych. 7. Wsparcie dla modelu MVC: ASP.NET oferuje wsparcie dla modelu projektowego MVC (Model-View-Controller), który pomaga oddzielić logikę biznesową, widok i kontroler.   Dzięki tym cechom ASP.NET jest popularnym frameworkiem do tworzenia aplikacji internetowych, szczególnie w przypadku tworzenia aplikacji na platformę Microsoft. | |

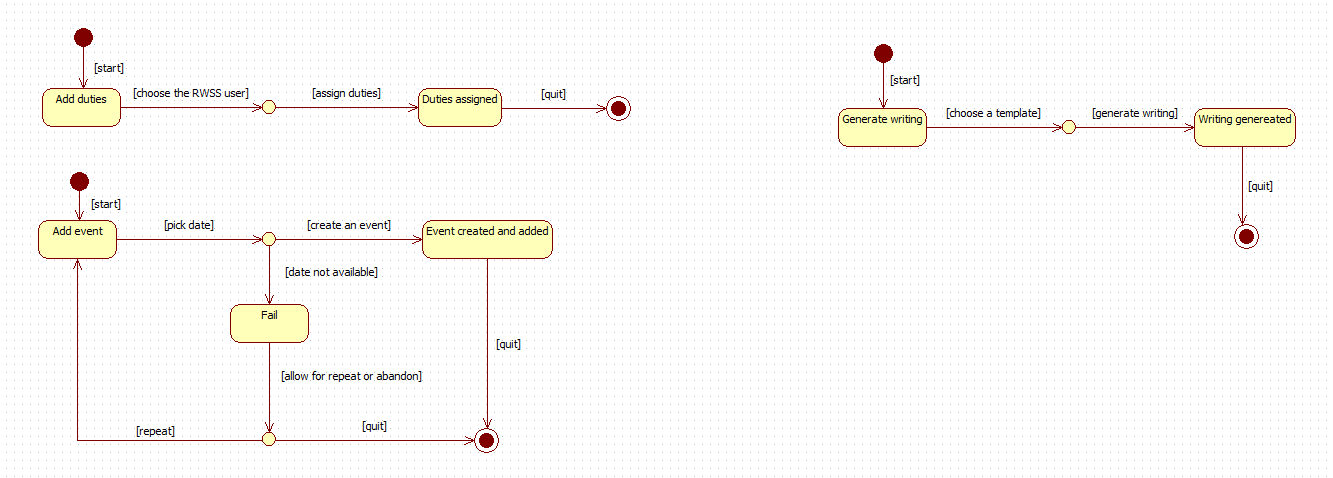
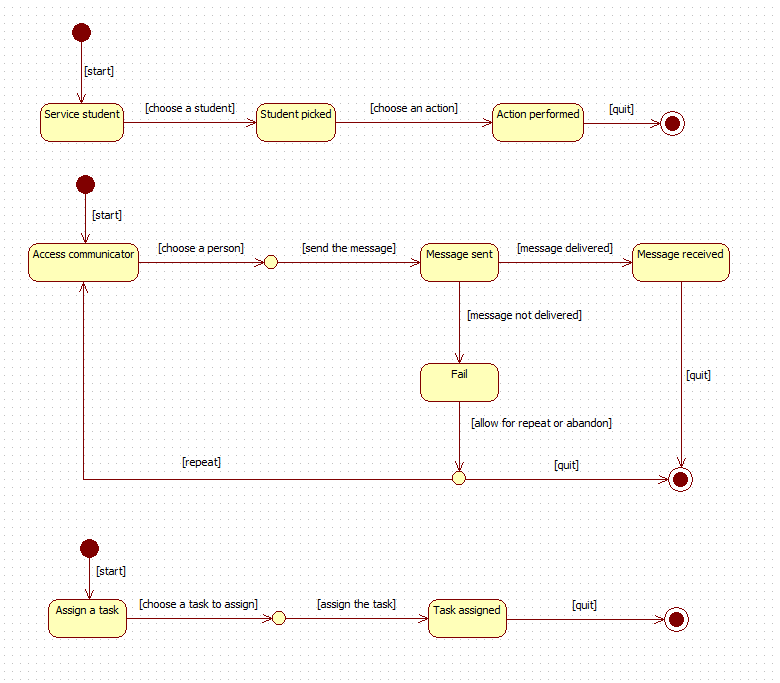
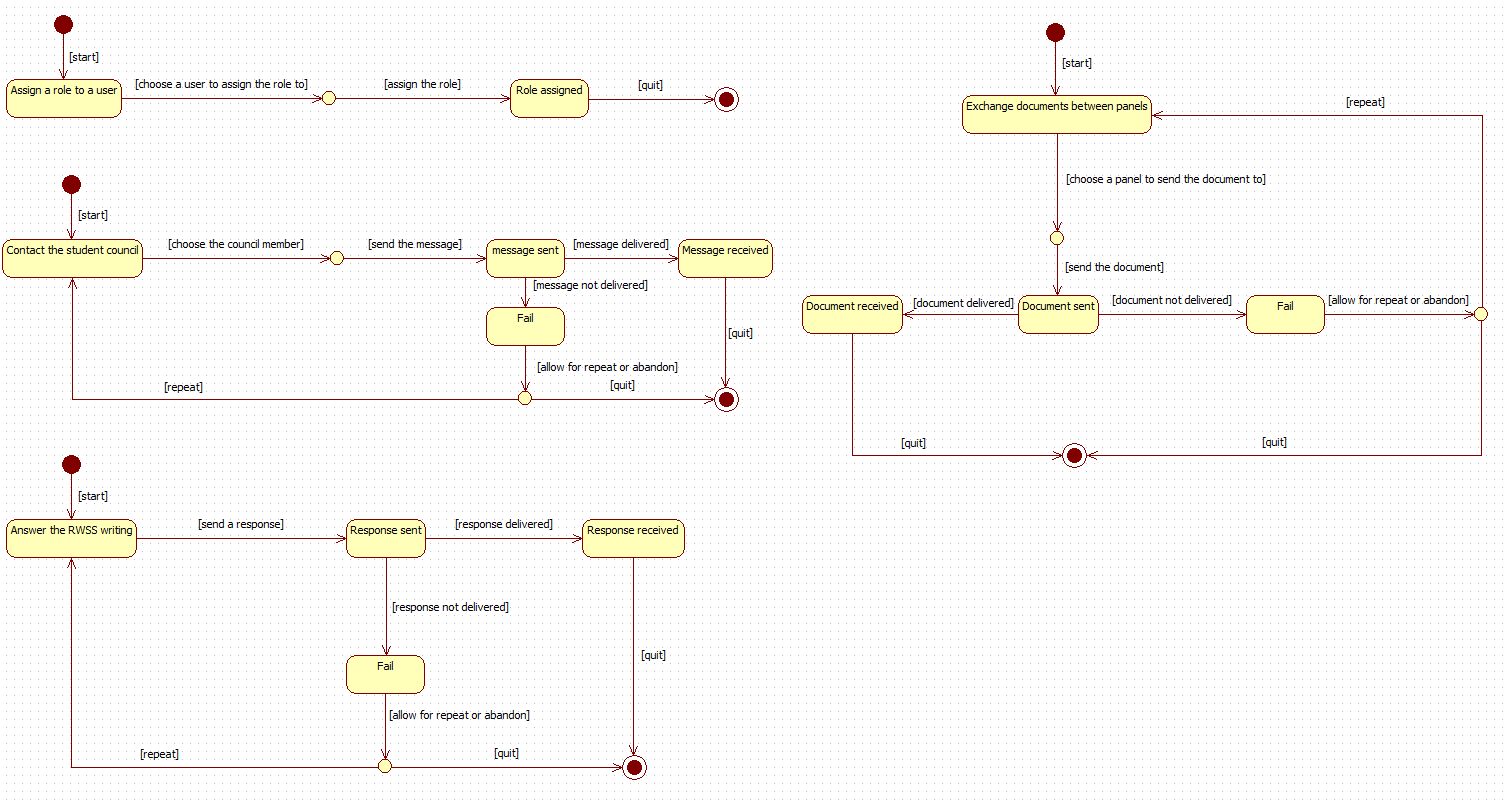
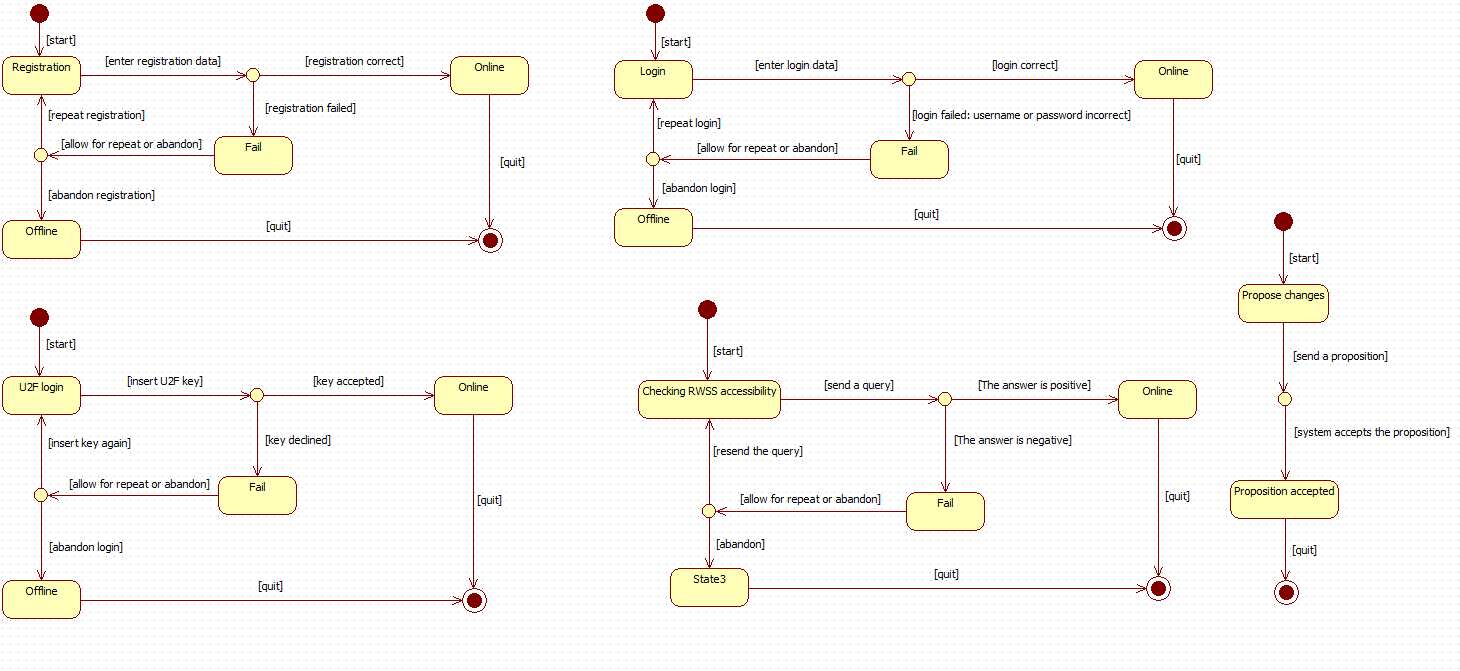
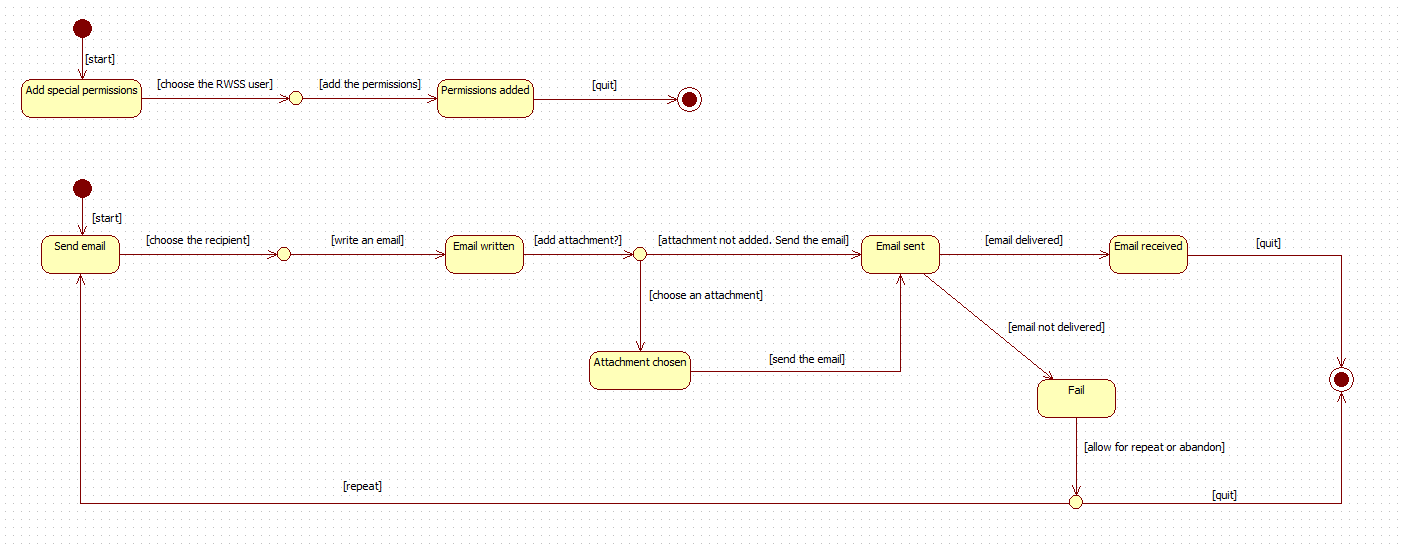
Blazor

Blazor to framework do tworzenia aplikacji internetowych opracowany przez firmę Microsoft. Oto kilka cech i zalet tego frameworka:

1. Programowanie w języku C#: Blazor pozwala programistom na tworzenie aplikacji internetowych w języku C#, co ułatwia rozwijanie aplikacji, szczególnie dla programistów, którzy już pracowali z tym językiem.
2. Komponentowy model programowania: Blazor jest oparty na komponentowym modelu programowania, który ułatwia organizację kodu i pozwala na tworzenie reużywalnych komponentów.
3. Bezstanowość: Blazor pozwala na tworzenie bezstanowych aplikacji internetowych, co pozwala na łatwe skalowanie aplikacji i obsługę wielu użytkowników.
4. Wsparcie dla WebAssembly: Blazor wykorzystuje technologię WebAssembly, co pozwala na uruchamianie aplikacji w przeglądarce internetowej z prędkością zbliżoną do aplikacji napisanych w językach takich jak C++ czy Rust.
5. Łatwość użycia: Blazor oferuje łatwy w użyciu interfejs użytkownika i wiele narzędzi deweloperskich, które ułatwiają tworzenie aplikacji internetowych.
6. Wsparcie dla modelu MVVM: Blazor oferuje wsparcie dla modelu projektowego MVVM (Model-View-ViewModel), który pozwala na oddzielenie logiki biznesowej, widoku i modelu.
7. Integracja z platformą .NET: Blazor jest zintegrowany z platformą .NET, co pozwala na łatwe korzystanie z bibliotek .NET i innych narzędzi dostępnych w tej platformie.

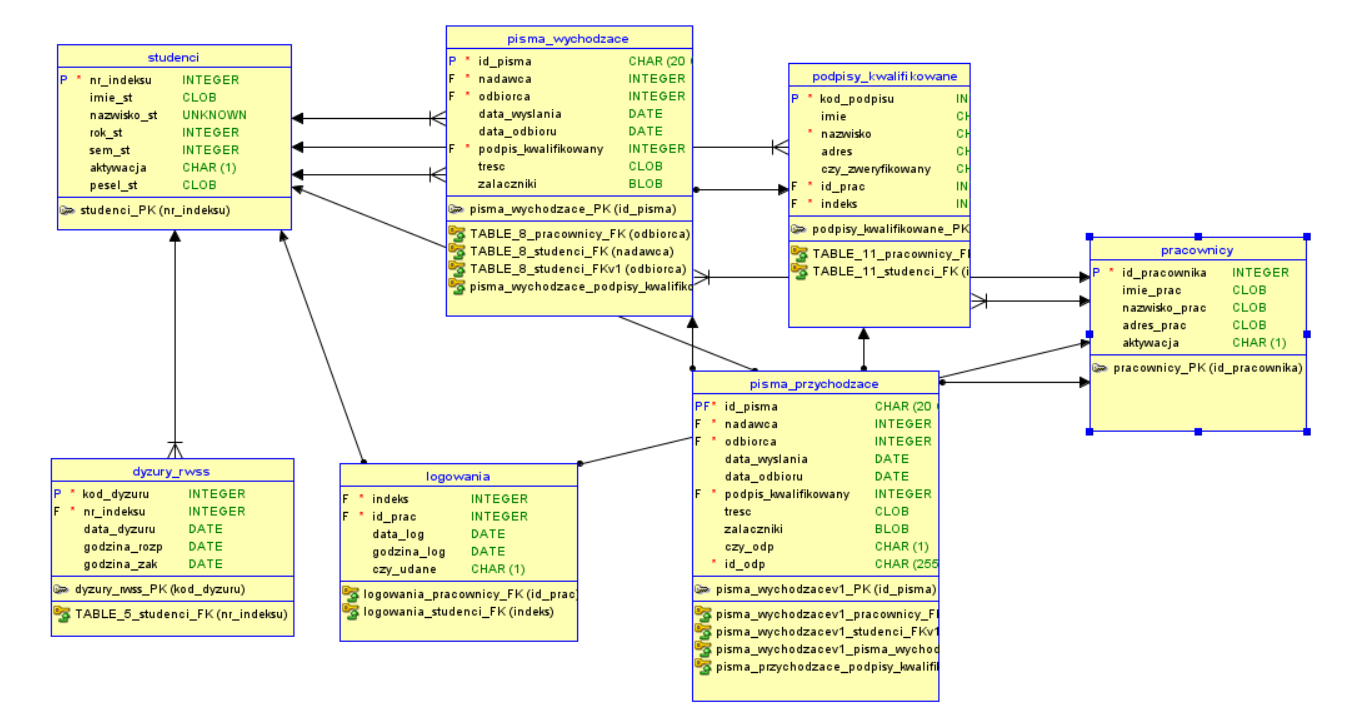
Dzięki tym cechom Blazor jest popularnym frameworkiem do tworzenia aplikacji internetowych, szczególnie w przypadku tworzenia aplikacji internetowych na platformie .NET i w przeglądarce internetowej.

DIAGRAMY STANÓW W POSZCZEGÓLNYCH MODUŁACH:



SCHEMAT BAZY DANYCH:

* W WERSJI DDL PRZEZNACZONEJ DLA MS SQL SERVER:
  + -- Generated by Oracle SQL Developer Data Modeler 22.2.0.165.1149
  + -- at: 2023-02-14 18:02:11 CET
  + -- site: SQL Server 2012
  + -- type: SQL Server 2012
  + CREATE TABLE documents
  + (
  + kod\_dokumentu CHAR (15) NOT NULL ,
  + TRESC TEXT
  + )
  + GO
  + ALTER TABLE documents ADD CONSTRAINT documents\_PK PRIMARY KEY CLUSTERED (kod\_dokumentu)
  + WITH (
  + ALLOW\_PAGE\_LOCKS = ON ,
  + ALLOW\_ROW\_LOCKS = ON )
  + GO
  + CREATE TABLE documents\_to\_dwmfi
  + (
  + kod\_dokumentu CHAR (15) NOT NULL ,
  + data\_wygenerowania DATE ,
  + data\_wyslania DATE ,
  + godzina\_wyslania TIME ,
  + czy\_odebrano BIT ,
  + decyzja INTEGER ,
  + pesel\_gen CHAR (11) NOT NULL ,
  + pesel\_wys CHAR (11) NOT NULL
  + )
  + GO
  + ALTER TABLE documents\_to\_dwmfi ADD CONSTRAINT dokumenty\_wyslane\_PK PRIMARY KEY CLUSTERED (kod\_dokumentu)
  + WITH (
  + ALLOW\_PAGE\_LOCKS = ON ,
  + ALLOW\_ROW\_LOCKS = ON )
  + GO
  + CREATE TABLE praviliages
  + (
  + pesel CHAR (11) NOT NULL ,
  + student BIT ,
  + przewodniczący BIT ,
  + dziekanat BIT NOT NULL ,
  + czlonek\_rwss BIT ,
  + serwis BIT ,
  + czy\_aktywne BIT ,
  + waznosc INTEGER
  + )
  + GO
  + ALTER TABLE praviliages ADD CONSTRAINT praviliages\_PK PRIMARY KEY CLUSTERED (pesel)
  + WITH (
  + ALLOW\_PAGE\_LOCKS = ON ,
  + ALLOW\_ROW\_LOCKS = ON )
  + GO
  + CREATE TABLE shifts
  + (
  + id INTEGER NOT NULL ,
  + pesel CHAR (11) NOT NULL ,
  + data\_dyzuru DATE ,
  + godzina\_rozpoczecia TIME ,
  + godzina\_zakonczenia TIME
  + )
  + GO
  + ALTER TABLE shifts ADD CONSTRAINT shifts\_PK PRIMARY KEY CLUSTERED (id)
  + WITH (
  + ALLOW\_PAGE\_LOCKS = ON ,
  + ALLOW\_ROW\_LOCKS = ON )
  + GO
  + CREATE TABLE "user"
  + (
  + id INTEGER ,
  + imie CHAR (255) ,
  + nazwisko CHAR (255) ,
  + pesel CHAR (11) NOT NULL ,
  + "index" INTEGER
  + )
  + GO
  + ALTER TABLE "user" ADD CONSTRAINT user\_PK PRIMARY KEY CLUSTERED (pesel)
  + WITH (
  + ALLOW\_PAGE\_LOCKS = ON ,
  + ALLOW\_ROW\_LOCKS = ON )
  + GO
  + ALTER TABLE documents\_to\_dwmfi
  + ADD CONSTRAINT document\_to\_dwmfi\_user\_FK FOREIGN KEY
  + (
  + pesel\_gen
  + )
  + REFERENCES ""user""
  + (
  + pesel
  + )
  + ON DELETE NO ACTION
  + ON UPDATE NO ACTION
  + GO
  + ALTER TABLE documents\_to\_dwmfi
  + ADD CONSTRAINT document\_to\_dwmfi\_user\_FKv2 FOREIGN KEY
  + (
  + pesel\_wys
  + )
  + REFERENCES ""user""
  + (
  + pesel
  + )
  + ON DELETE NO ACTION
  + ON UPDATE NO ACTION
  + GO
  + ALTER TABLE praviliages
  + ADD CONSTRAINT praviliages\_user\_FK FOREIGN KEY
  + (
  + pesel
  + )
  + REFERENCES ""user""
  + (
  + pesel
  + )
  + ON DELETE NO ACTION
  + ON UPDATE NO ACTION
  + GO
  + ALTER TABLE shifts
  + ADD CONSTRAINT TABLE\_4\_user\_FK FOREIGN KEY
  + (
  + pesel
  + )
  + REFERENCES ""user""
  + (
  + pesel
  + )
  + ON DELETE NO ACTION
  + ON UPDATE NO ACTION
  + GO
  + ALTER TABLE documents
  + ADD CONSTRAINT TABLE\_9\_documents\_to\_dwmfi\_FK FOREIGN KEY
  + (
  + kod\_dokumentu
  + )
  + REFERENCES documents\_to\_dwmfi
  + (
  + kod\_dokumentu
  + )
  + ON DELETE NO ACTION
  + ON UPDATE NO ACTION
  + GO
  + -- Oracle SQL Developer Data Modeler Summary Report:
  + --
  + -- CREATE TABLE 5
  + -- CREATE INDEX 0
  + -- ALTER TABLE 10
  + -- CREATE VIEW 0
  + -- ALTER VIEW 0
  + -- CREATE PACKAGE 0
  + -- CREATE PACKAGE BODY 0
  + -- CREATE PROCEDURE 0
  + -- CREATE FUNCTION 0
  + -- CREATE TRIGGER 0
  + -- ALTER TRIGGER 0
  + -- CREATE DATABASE 0
  + -- CREATE DEFAULT 0
  + -- CREATE INDEX ON VIEW 0
  + -- CREATE ROLLBACK SEGMENT 0
  + -- CREATE ROLE 0
  + -- CREATE RULE 0
  + -- CREATE SCHEMA 0
  + -- CREATE SEQUENCE 0
  + -- CREATE PARTITION FUNCTION 0
  + -- CREATE PARTITION SCHEME 0
  + --
  + -- DROP DATABASE 0
  + --
  + -- ERRORS 0
  + -- WARNINGS 0
* W WERSJI GRAFICZNEJ DO PRZEDSTAWIENIA RELACJI I INTERFEJSÓW



Scenariusze testowe

1. Scenariusz testowy: Moduł logowanie

1.

Given: Prawidłowe dane logowania

When: Metoda logowania przez SSO

Then: Zalogowanie do systemu

2.

Given: Błędne dane logowania

When: Metoda logowania przez SSO

Then: Niezalogowanie do systemu i komunikat o błędzie

3.

Given: Prawidłowe dane logowania

When: Metoda logowania przez Węzeł Krajowy

Then: Zalogowanie do systemu

4.

Given: Błędne dane logowania

When: Metoda logowania przez Węzeł Krajowy

Then: Niezalogowanie do systemu i komunikat o błędzie

5.

Given: Prawidłowe dane logowania

When: Metoda logowania przez logowanie awaryjne

Then: Zalogowanie do systemu

6.

Given: Błędne dane logowania

When: Metoda logowania przez logowanie awaryjne

Then: Niezalogowanie do systemu i komunikat o błędzie

1. Scenariusz testowy: moduł e-Przewodniczący

1.

Given:Wybór użytkownika

When:Metodanadania uprawnień

Then:Zmiana uprawnień użytkownika

2.  
Given:Wybór użytkownika

When:Metodawysłania wiadomości

Then:Wysłanie wiadomości do użytkownika

3.

Given:Wybór prawidłowego pliku

When:Metoda importowania bazy starostów

Then:Baza starostów zostaje zaktualizowana

4.

Given:Wybór błędnego pliku

When:Metoda importowania bazy starostów

Then:Baza starostów niezostaje zaktualizowanai wyświetla komunikat o błędzie

5.

Given:Wybór użytkownika

When:Metoda ustawienia dyżuru

Then:Dodawanie dyżuru dla użytkownika

6.

Given:Dodanie wydarzenia

When:Metoda dodania wydarzenia

Then:Dodawanie wydarzeniado systemu

7.

Given:Generowanie pism

When:Metoda generowania pism

Then:Wygenerowanie poszczególnych pism do innych jednostek uczelni

8.

Given:Deklaracja wzorów pism

When:Metoda deklaracji wzorów pism

Then:Zadeklarowanie konkretnych wzorów pism

1. Scenariusz testowy: e-Student

1.

Given: Przycisk nowego działania

When: Metoda nowego działania

Then: Student może utworzyć nowe działanie

2.

Given: Przycisk komunikacji

When: Metoda komunikacji

Then: Student może komunikować się z organem Samorządu Studentów działającym na danym wydziale

Plus możliwość logowania za pomocą SSO oraz Węzła Krajowego, które są zawarte w HD-43 scenariusze logowania.

1. Scenariusz testowy: e-Dziekanat

1.

Given: Użytkownik zalogowany

When: Metoda tworzenia sprawy

Then: Utworzona przez użytkownika sprawa ma unikalne znaki

2.

Given: Wybór użytkownika

When: Metoda przypisania ról

Then: Użytkownik ma przypisaną rolę

3.

Given: Przycisk wysłania dokumentacji

When: Metoda transakcji dokumentów

Then: Dokumenty są bezpiecznie wymieniane między jednostkami organizacyjnymi

