TSR School jest franczyzodawcą o zasięgu ogólnokrajowym, który swoje placówki ma zlokalizowane w średnio zaludnionych polskich aglomeracjach powyżej 50 tyś mieszkańców. Poza tym dostarcza narzędzia do nauczania zdalnego w języku angielskim.

Nasze przedsiębiorstwo jest akademią IT z prawie 10-letnim doświadczeniem proponującym nieszablonowane rozwiązania w zakładaniu innowacyjnego biznes w sektorze edukacyjnym. Korzystając z doświadczenia, „know-how”, materiałów edukacyjnych i platform do zarządzania znacznie zwiększa szanse na sukces w tej szybko rozwijającej się, ale konkurencyjnej branży.

Wypełniamy lukę w tradycyjnym systemie edukacji, który nie jest w stanie nadążyć za rosnącym zapotrzebowaniem na specjalistów IT. Dlatego konieczne są alternatywne i innowacyjne metody kształcenia osób posiadających umiejętności cyfrowe.

Headquarter firmy zlokalizowany jest w Olsztynie na ulicy Słonecznej 54, 10-561 i otwarty jest w godzinach 8-16 (pomoc telefoniczna dla franczyzobiorców oraz uczniów jest otwarta w godzinach pracy placówki).

Pragniemy dostarczać studentom jak i prowadzącym zajęcia bezkonkurencyjne rozwiązania w dziedzinie rozwoju IT. Mając to na uwadze i brak dotychczasowego pokrycia naszych rozwiązań dla tych najmłodszych studentów chcemy dostarczyć narzędzie desktopowe, które przybliży dzieciom jak i osobom dorosłym z luką wiedzy technologicznej, czym jest programowanie jak również problem solving. A prowadzącym zajęcia dostarczyć możliwość monitorowania osiągnięć podopiecznych, aby móc wyłonić jednostki przejawiające ponadprzeciętne umiejętności, i dopasować program nauczania do ich możliwości, czy zaoferować stypendium. Zadaniem celu rankingowego nie jest tylko premiowanie najbardziej uzdolnionych, ale również pomaganie uczniom, którzy nie dotrzymują tempa programu nauczania. Co pozwala osiągać znakomite wyniki związane z utrzymaniem klienta przy naszej ofercie.

Klienci/uczniowie po uruchomieniu sprzętu znajdującego się w placówce szkoły powinni posiadać zainstalowane oprogramowanie do nauki, oraz założone konto studenta przez prowadzącego zajęcia. To samo dotyczy się uczniów korzystających z naszego zdalnego programu nauczania – w takim przypadku student powinien samodzielnie lub z pomocą rodzica wyposażyć się w oprogramowanie, które zbudowane powinno być gotowe do pobrania w serwisie github projektu.

Po uruchomieniu programu uczeń wyposażony w swoje dane uwierzytelniana powinien móc zalogować się do platformy, w innym wypadku powinien skontaktować się z prowadzącym zajęcia bądź infolinią szkoły. Komunikacja ze szkołą odbywa się za pośrednictwem poczty elektronicznej pod adresem [contact@tsr.edu.pl](mailto:contact@tsr.edu.pl).

Po zalogowaniu użytkownikowi zaprezentowane zostają zadania do wykonania, których baza znajduje się na serwerze, a do prowadzącego zajęcia należy obowiązek przypisywania zadań dla ucznia. Pozwala to na zwiększenie efektywności nauczania, ponieważ uczeń nie korzysta z niedopasowanej puli zadań, a elastyczność zależy od naszego pracownika. Jednakże nauczyciele wyposażeni są w rekomendowane posortowane grupy zadań.

Zadaniem ucznia jest osiągnięcie określonego celu przy pewnych zadanych warunkach początkowych. Interfejs użytkownika podczas realizowania zadania podzielony jest na cztery główne części:

1. Pasek zadań z pomocą, informacjami o zadaniu oraz ustawieniami,
2. Wertykalną lista bloków dostępnych dla studenta
3. Konsola informującą o wyniku pracy programu
4. Główną część GUI zajmuje tablica na której następuje tworzenie przebiegu pracy programu

O sposobie działania programu decydują bloki, które realizują podstawowe operacje warunkowe (funkcje AND, OR, NOT…) oraz operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie, reszta z dzielenia), a wynik ich pracy jest dostarczany na konsolę.

Head of the Course realizuje swoje obowiązki za pośrednictwem dashboardu podglądu bazy, która znajduje się na serwerze.

* Organizacja – TSR/TSR School zwany również franczyzodawca zajmujący się organizacją usług w dziedzinie edukacji w branży IT
* Otoczenie organizacji – firmy rekrutujące specjalistów IT, inne szkoły programowania
* Intersariusze organizacji - dzieci i nastolatkowie, human reasources team
* Dziedzina problemowa – nowoczesne technologie
* Intersariusze dziedziny problemowej - firmy technologiczne
* Pracownicy biznesowi – headquarter firmy, firmy rekrutujące specjalistów IT

Zakres organizacji, dziedziny problemowej, aktorzy

Firma TSR School

Organizacja = cała firma

Struktura firmy = prowadzący zajęcia, placówka dydaktyczna, główna siedziba firmy

Dziedzina problemowa : prowadzenie zajęć, nauka programowania i problem solving

Aktorzy biznesowi: student, nowy student

Pracownicy biznesowi: prowadzący zajęcia

PU:

- ~~Utwórz studenta~~ -> Administrowanie systemem

- ~~Przydziel zadanie~~ -> Administrowanie systemem

- ~~Autoryzuj użytkownika~~ -> Autoryzowanie

- ~~Wybierz zadanie~~ -> Nauka w programie

- ~~Buduj program~~ -> Nauka w programie

- ~~Kompiluj program~~ -> Nauka w programie

- ~~Oceń zadanie~~ -> Administrowanie systemem

- ~~Dostosuj program~~ -> Nauka w programie

Aktorzy biznesowi: student, ~~nowy student~~

Pracownicy biznesowi: prowadzący zajęcia

PU:

- Administrowanie systemem

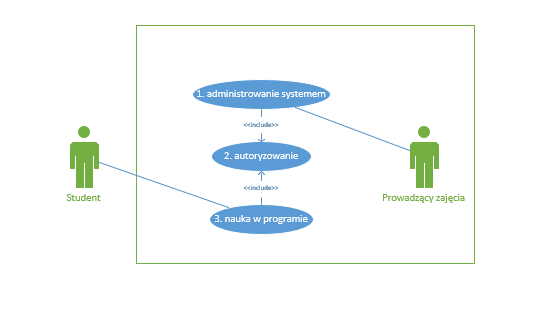
- Autoryzowanie

- Nauka w programie

Aktorzy biznesowi: student

Pracownicy biznesowi: prowadzący zajęcia

Biznesowy Diagram Przypadków Użycia



Scenariusze PU

1. Administrowanie systemem

Administrowanie systemem lokalnie realizuje prowadzący zajęcia i do jego zadań należy tworzenie kont studentów, analiza pracy jego studentów oraz przydzielanie zadań.

1. Autoryzowanie

Autoryzowanie polega na uwierzytelnianiu studentów i prowadzących zajęcia do dostępu do danych.

1. Nauka w programie

Student nabywa nowych umiejętności w programie, a jego obsługa sprowadza się do: dostosowaniu preferencji obsługi, wybieraniu przydzielanych zadań oraz tworzeniu skryptu za pośrednictwem bloków i sprawdzaniu i analizy wyników.

Przykład:

3.1) Student Franek Bachleda wybiera zadanie przypisane mu przez prowadzącego Grzegorza Kminka w którym ma wypisać liczby nieparzyste od 1 do 9. Franek tworzy zmienną x = 1, wybiera blok pętli while po czym dodaje do niego warunek kończący x<10. Wybiera blok porównania, sprawdzając czy x%2==1. Tworzy blok mówiący że jeśli tak to wypisuje liczbę na standardowe wyjście. Jeśli nie przechodzi dalej. Dodaje blok zwiększający wartość x o 1. Franek wciska przycisk start w celu przekonania się czy program działa poprawnie. Ponieważ wynik jest poprawny, dopisywany jest do rankingu i zwracany jest mu komunikat o poprawnym działaniu programu.

Specyfikacja:

3.2)

-student wybiera zadanie

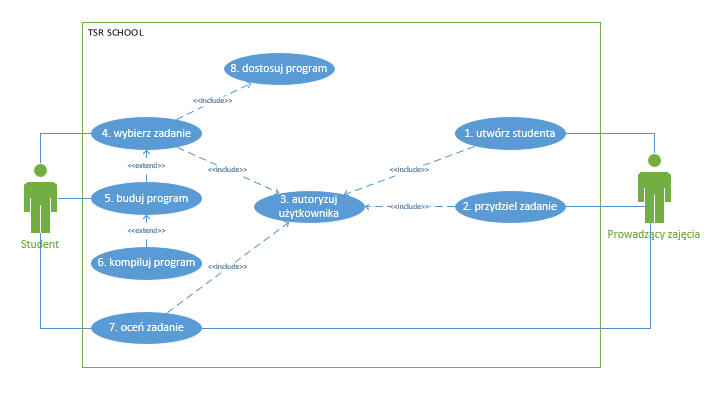
-student wybiera bloki

-student łączy bloki

-student uzupełnia informacje o blokach

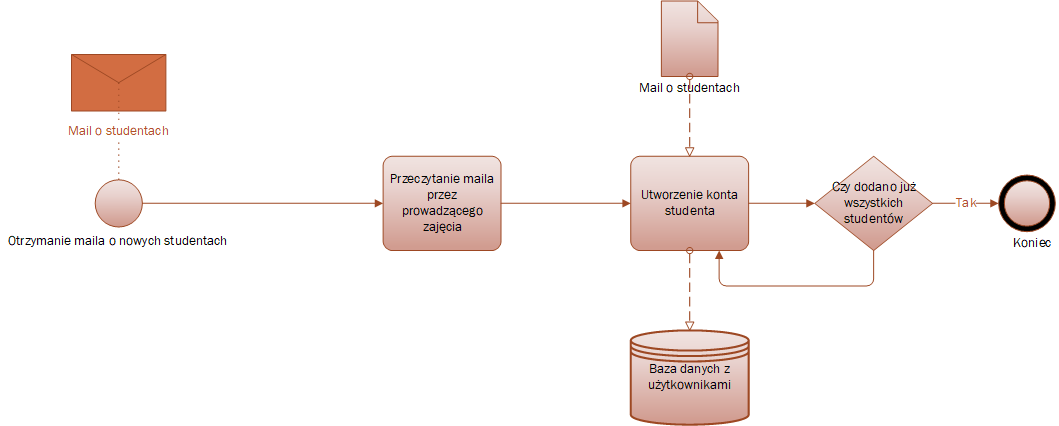
-student wciska przycisk start

Systemowy diagram przypadków użycia



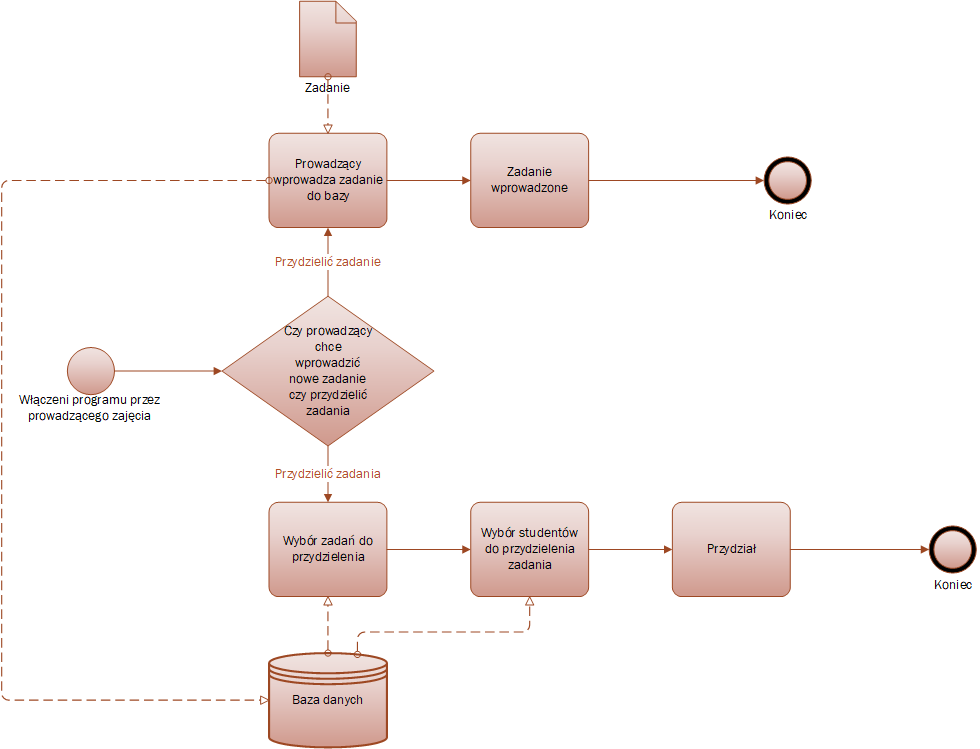
1. Utwórz studenta

Prowadzący zajęcia dostaje emaila z danymi studenta (lub listy studentów) który został przypisany do jego grupy. Na ich podstawie tworzy nowe konto dla studenta w platformie aplikacji.



1. Przydziel Zadania

Prowadzący zajęcia tworzy nowe zadanie dla studentów w aplikacji programowania blokowego z przykładowymi niekompletnymi rozwiązaniami, po czym umieszcza je w bazie danych i przypisuje wybrane z bazy danych zadania wybranym studentom.



1. Autoryzuj użytkownika

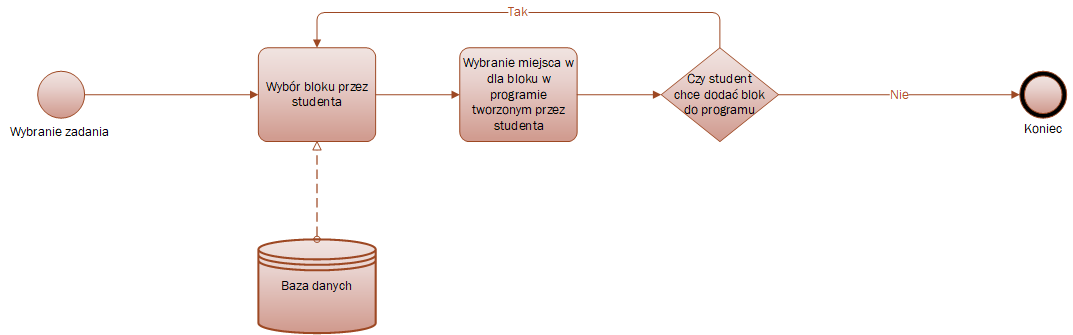
Student/Prowadzący zajęcia wpisuje login I hasło w aplikacji. Jeżeli wprowadził prawidłowe dane przechodzi do przeglądu zadań. Jeśli wpisał złe informacje proszony jest o ponowne wpisanie hasła i loginu. Dashboard studenta i prowadzącego zajęcia różni się (student ma możliwość realizacji zadań). Wprowadzone dane są później wykorzystywane w celu weryfikacji dostępu do danych.

1. Wybierz Zadanie

Student po zalogowaniu widzi ekran z możliwymi do wyboru zadaniami przypisanymi mu przez prowadzącego zajęcia. Wybiera konkretne zadanie i przechodzi do budowania programu.

1. Buduj Program

Student za pomocą bloków(zestaw instrukcji bądź instrukcja do wykonania przez procesor) które są układane przez niego w wybranej kolejności tworzy algorytm po czym może wcisnąć przycisk Start.



1. Kompiluj program

Po wciśnięciu przycisku start program blokowy jest parse’owany do pythona i jest interpretowany przez interpreter. Jeżeli program jest napisany poprawnie jego wynik jest wyświetlany studentowi na konsoli, jeżeli nie zwracany jest błąd.

1. Ocena studenta

Prowadzona jest automatyczna lista studentów o najlepszych wynikach (ze względu na szybkość kompilacji, wykorzystaną pamięć) do której mają dostęp studenci i prowadzący zajęcia.

1. Dostosuj program

Użytkownik i prowadzący ma możliwość podstawowej możliwości dostosowania programu do własnych preferencji za pośrednictwem paska zadań (m.in. rozdzielczość).

Dokumenty występujące w ramach dziedziny problemowej:

**Graphical user interface, text, application

Description automatically generatedformularz\_rejestracji** = rodzaj\_kursu + { dane\_uczestnika } + { dane\_opiekuna } + skad\_wiesz + uwagi

**rodzaj\_kursu** = [on-site | remote]

**dane\_uczestnika** = email + rok\_urodzenia + imie\_i\_nazwisko

**dane\_opiekuna** = email + nr\_telefonu + imie\_nazwisko + kod\_pocztowy

Graphical user interface, text, application

Description automatically generated

**dyplom** = imie\_nazwisko\_uczestnika + prowadzacy\_zajecia + data

Text

Description automatically generated

**zgoda\_przetwarzania\_danych\_osobowych** = data + podpis



**faktura\_za\_serwer** = data\_wystawienia + data\_wystawienia + nr\_faktury + nazwa\_sprzedawcy + adres\_sprzedawcy + nip\_sprzedawcy + vat\_ue + nabywca + adres\_nabywcy + płatnosc + zamowienie + pelna\_cena + uwagi

**platnosc** = [ karta | przelew | przelewy24 ]

**zamówienie** = { usluga }

**usluga** = nazwa\_serwera + ilosc + cena\_netto + vat + warto\_brutto