Krzysztof Szaniawski, Patryk Wolski, gr 2

Zadanie 1: - koncepcja planowanej w projekcie aplikacji

1. Technologie:
   * ASP.NET Web API
   * Angular
   * MongoDB
   * JSON
   * IIS
2. Wymagania
   * funkcjonalne
     + dodawanie, edytowanie, usuwanie i wyszukiwanie
       1. użytkowników
       2. klientów
       3. pokoi
       4. rezerwacji
     + logowanie do systemu
   * niefunkcjonalne:
     + interfejs użytkownika powinien być responsywny
     + interfejs użytkownika powinien być w języku polskim
     + system powinien być niezawodny i wydajny
     + hasła powinny być przechowywane w bazie danych formie hashu
3. aktorzy i przypadki użycia
   * administrator
     + logowanie
     + zarządzanie:
       1. użytkownikami
       2. klientami
       3. pokojami
       4. rezerwacjami
     + wyszukiwanie danych
   * użytkownik
     + logowanie
     + zarządzanie:
       1. klientami
       2. pokojami
       3. rezerwacjami
     + wyszukiwanie danych

Zadanie 2: - Planowana architektura aplikacji

1. Przewidywane komponenty: usługi, bazy danych, interfejs użytkownika, rejestr usług, itd.

|  |  |  |
| --- | --- | --- |
| Komponent | Baza danych (MongoDB) |  |
| Identity service | IdentityDb | Usługa odpowiedzialna za zarządzanie danymi użytkowników i logowania |
| Customers service | CustomersDb | Usługa odpowiedzialna za zarządzanie danymi klientów |
| Rooms service | RoomsDb | Usługa odpowiedzialna za zarządzanie danymi pokoi |
| Reservations service | ReservationsDb | Usługa odpowiedzialna za zarządzanie danymi rezerwacji |
| API | Brak | Internetowy interfejs api odpowiedzialny za komunikację interfejsu użytkownika z serwisami |
| Client App (interfejs użytkownika) | Brak | Interfejs użytkownika |

1. Przewidywany protokół komunikacyjny w projekcie: Schemat komunikacji, jakie usługi będą się komunikować ze sobą, jakie żądania / odpowiedzi będą przesyłane do i z usług, ścieżki url do wywołania poszczególnych zasobów / funkcjonalności usług. Można zrobić w tabelce, opisać w punktach lub opisać zdaniami.

Identity service

|  |  |  |  |
| --- | --- | --- | --- |
| Ścieżka | Metoda | parametry | odpowiedź |
| api/users | Get | - | List User |
| api/users/{id} | Get | id użytkownika | User |
| api/users | Post | User | User |

Customer service

|  |  |  |  |
| --- | --- | --- | --- |
| Ścieżka | Metoda | parametry | odpowiedź |
| api/customers | Get | - | List Customer |
| api/customers/{id} | Get | id klienta | Customer |
| api/customers | Post | Customer | Customer |
| api/customers | Put | id klienta, Customer | Customer |
| api/customers | Delete | id klienta | - |

Rooms service

|  |  |  |  |
| --- | --- | --- | --- |
| Ścieżka | Metoda | parametry | odpowiedź |
| api/rooms | Get | - | List Room |
| api/rooms /{id} | Get | id pokoju | Room |
| api/rooms | Post | Room | Room |
| api/rooms | Put | id pokoju, Room | Room |
| api/rooms | Delete | id pokoju | - |

Reservations service

|  |  |  |  |
| --- | --- | --- | --- |
| Ścieżka | Metoda | parametry | odpowiedź |
| api/reservation | Get | - | List Reservation |
| api/reservation /{id} | Get | id rezerwacji | Reservation |
| api/reservation | Post | Reservation | Reservation |
| api/reservation | Put | id rezerwacji, Reservation | Reservation |
| api/reservation | Delete | id rezerwacji | - |

1. Przewidywana struktura baz danych, jakie dane będą przechowywane w jakiej bazie / tabelach itd.

Identity service, baza IdentityDb, kolekcja:

1. user

* id
* email
* password
* role

Customers service, baza CustomersDb, kolekcja:

2. customers

* id
* firstname
* lastname
* birthdate
* phone
* address

Rooms service, baza RoomsDb, kolekcja:

3. rooms

* id
* number
* peopleNumber
* price
* description

Reservations service, baza ReservationsDb, kolekcja:

4. reservations

* id
* customerId
* startDate
* endDate
* price
* rooms

1. Elementy interfejsu użytkownika – co będzie zawierać, co umożliwiać dla poszczególnych typów użytkowników.

* Logowanie
* (Admin) Zarządzanie użytkownikami
* Zarządzanie klientami
* (Admin) Zarządzanie pokojami
* Zarządzanie rezerwacjami
* Wyszukiwanie klientów, pokojów, rezerwacji

1. Sposób logowania i zarządzania rolami / użytkownikami – jak będzie rozwiązany i powiązany z innymi usługami.

Aplikacja kliencka i interfejs api łączy się z serwisem identyfikacji w celu autoryzacji przy użyciu danych logowania. Interfejs api i aplikacja kliencka przechowują sesję użytkownika oraz jego rolę. Dzięki temu w interfejsie api można wywoływać metody i odwoływać się do innych serwisów dostępnych dla zalogowanego użytkownika o danej roli.

Zadanie 3: Podział zadań i prac:

1. Przedstaw podział prac planowany w projekcie, jakie osoby będą odpowiedzialne za wykonanie jakich części projektu.

Krzysztof Szaniawski

* Rooms service
* Reservations service
* API
* Aplikacja kliencka

Patryk Wolski

* Identity service
* Customers service
* API
* Aplikacja kliencka

API i aplikacje kliencką wspólnie, każdy będzie odpowiedzialny za komponenty związane z stworzonymi przez siebie serwisami.