

Temat Projektu: Obsługa systemu parkingowego

Celem projektu jest przećwiczenie integracja systemów informatycznych za pomocą mechanizmu usług sieciowych. Projektowany system działała zgodnie z modelem B2B oraz B2C.

Zakres:

W ramach projektu przygotowują Państwo system wspierający obsługę pracowników odpowiedzialnych za obsługę strefy płatnego parkowania. Strefa obejmuje kilka/kilkanaście ulic i kilka parkometrów. Obsługuje ją kilku pracowników, każdy obsługując wybrany fragment strefy.

System składa się modułu identyfikacji zajętości miejsca parkingowego, parkometru (do zakupu biletu postojowego) oraz aplikacji głównej służącej do identyfikacji aktualnego stanu i detekcji zjawisk niepożądanych.

Ogólny scenariusz funkcjonowania jest następujący:

1. Przyjeżdża samochód – zajmuje miejsce parkingowe – system powiadamia aplikację główną o zajętości konkretnego miejsca.
2. Kierowca wysiada i idzie kupić bilet do parkometru. Po zakupie biletu informacje o bilecie są przesyłane do systemu.
3. Aktualizuje sobie stan wiedzy na temat strefy parkingowej. W sytuacji gdy po zajętości miejsca nie nastąpi zakup biletu system wysła powiadomienie na urządzenie mobilne pracownika obsługi strefy z informacją że być może samochód z miejsca X nie ma opłaconego biletu.

Innym takim zdarzeniem jest sytuacja gdy na podstawie informacji biletów miejsca powinny być już zwolnione a ilość zajętości miejsc przekracza stan wynikający z aktualnych zakupionych biletów (ważność niektórych biletów uległa wygaśnięciu).

Do realizacji:

1. Implementacja mocka symulującego zajętość miejsca parkingowego (wystarczy interfejs tekstowy). Symulator ma za pomocą usług SOAP-owych komunikować się z systemem głównym przekazując niezbędne informacje.
2. Implementacja mock parkometru - podobnie jak wyżej (wystarczy interfejs tekstowy pozwalający na definiowanie parametrów biletu). Komunikacja za pomocą usługi typu REST wysyłającej dane w formacie JASON lub XML na adres modułu głównego.
3. Moduły aplikacji głównej - w jej skład wchodzi:
 - a. przygotowanie Dashboard wizualizującego informacje o stanie zajętość miejsc parkingowych i aktualnej ilości zakupionych biletów (ważnych oraz tych, które wygasły do 10 min).

- b. Moduły detekcji określonych zdarzeń związanych z obsługą strefy parkingowej: nowy pojazd, brak zakupu biletu, zwolnienie miejsca, wygaśnięcie ważności biletu bez zwolnienia miejsca parkingowego itp.
Sugerowany tryb pracy: asynchroniczny – system reaguje gdy jest powiadamiany lub wykrywa jakieś zdarzenie na podstawie stanu danych.
- c. System alertów dla kontrolerów strefy. Wykorzystując JMS wysłać dane do modułu wystawiającego powiadomienia dla urządzeń mobilnych – Powiadomienia są wystawiane za pomocą API REST zasilanego danymi za pomocą JMS.
- d. Zewnętrzne API dla zewnętrznych systemów monitorowania zajętości miejsca na ulicach - pozwalające na udostępnienie podstawowych informacji innym systemom zewnętrznym (implementacji SOAP oraz REST).

W systemie każde wolne i zajęte miejsce jest precyzyjnie identyfikowane wykorzystując id detektora. Dlatego w sytuacji gdy zostaje zajęte można z pewnym prawdopodobieństwem określić jakiego typu bilety zostały zakupione przez właściciela pojazdu. Dlatego gdy minie szacowany czas pobytu i nie zostały dokupione nowe bilety a samochód nie zwolnił miejsca należy wysłać alert do pracownika odpowiadającego za dane rejon (na urządzenie mobilne) oraz zasygnalizować ten fakt w dashboard.

Dostęp do aplikacji mogą posiadać jedynie zalogowani użytkownicy. W tym celu serwis musi być zabezpieczony za pomocą mechanizmów JAAS. Powinna być (możliwa jest) możliwość zmiany języka obsługi aplikacji.

W systemie można wyróżnić kilka ról: admin, pracownik strefy.

1. Dostęp do serwisu ma być ograniczony tylko dla zalogowanych użytkowników. Należy wykorzystać zabezpieczenie na poziomie web.
2. Pracownik strefy ma dostęp do danych dotyczący tylko jego strefy, admin ma dostęp do wszystkiego i widzi całość danych.
3. Wszyscy użytkownicy mają korzystać z tych samych stron JSF, rozróżnienie ról ma odbywać się w metodach EJB.
4. Użytkownik powinien mieć możliwość zmiany swojego hasła, natomiast administrator powinien być w stanie zmieniać hasła wszystkich użytkowników. Hasła nie mogą być trzymane jako plain text.
5. Nie ma możliwości realizacji więcej niż jedna sesja zalogowanego użytkownika. Jeśli nastąpiła by próba zalogowanie w trakcie pracy - zwracany jest albo kontekst sesyjny albo informacji że dany użytkownik jest już zalogowany.