# Mini Project Travling system

Företaget Wigellkoncertnen skall sätta upp en nytt web project för bokning av resor ”Wigell Travels”. Systemen består av både en Front och backend.

Ni kommer att få fortsätta att arbeta med ett befintligt backens kod och förbättra om det behövs. Nytt frontens interface behöver byggas och integreras.

Uppgiften:

Om möjlighet att använda befintligt backend kod och sluta föra uppgiften,modifiera koden efter behov och integrerbara det med nytt web GUI.

Systemet ska kunna hantera kunder, resor. Bokningar. Resor ska ha pris per veck vecka. Namn på hotell och destination (stad och land). Kunder ska ha unika ID-nr, Användarnamn, och kunduppgifter. Sam namn och adress. Bokningar ska inkudera avresedatum, destination, namn på hotell och totalpris. Pris ska presenteras i SEK och Zloty.

Rest API

Backend Rest Api är skriven i Java Spring Boot. Ett antal endpoint är inluderade och det är dessa som skal användas. Data lagras under utvecklingsarbetet i en H2 databas.

Kunder

Följande endpoint skall inkluderas. (info finns redan i befinligt kod)

• Lista resmål GET /api/v1/trips

• Boka resa POST /api/v1/trips

• Uppgradera resa PUT /api/v1/trips/{id}

• Se tidigare och aktiva bokningar GET /api/v1/trips/{id}

Admin

Administratörer ska kunna göra ett antal aktiviteter med följande endpoints:

• Lista kunder GET /api/v1/customers

• Lägga till kund POST /api/v1/customers

• Ta bort kund DELETE /api/v1/customers/{id}

• Uppdatera kund PUT /api/v1/customers/{id}

• Lägga till resmål POST /api/v1/destinations

• Ta bort resmål DELETE /api/v1/destinations/{id}

• Uppdatera resmål PUT /api/v1/destinations/{id}

• Lista resmål GET /api/v1/trips

Viktigt att alla endpoints följer specifikationen eftersom de ska fungera med användargränssnittet.

KundWeb

WebGui ska tillhandahålla möjlighet för kunder att göra sina affärer med företaget.

Specifikationer komponenter

**Inloggning:**

Kund ska kunna logga in med sitt användarnamne och lössenord. (basic auth)

**Lista produkter/tjänster:**

Visa en lista med tillgängliga tjänster eller produkter. Vad du ska lista beror på vilken api du har byggt eller finns.

Kunden ska kunna sortera tjänster på namn och typ. På en rad ska bas-informationen om tjänsten finnas samt en knapp där kunden kan välja och komma vidare beställningen.

**Beställningen:**

En beställning görs efter att tjänsten valts och utgör av ett formulär (beställning fomrulär) där kunden fyller i informationen som krävs för att komma vidare i beställningen.

Dvs att kunden behöver vara inloggad för att kunna utföra beställningen och om kunden inte finns i systemen så skall kunden registers genom registrering formulären innan beställningen utförs.

**Lista beställningen:**

Kunden skall kunna lista sin order historik samt status på befintligt ordar. Från den lustan sjka kunden kunna gå vidare och se detjaer på ordern.

**Redigera en beställning:**

Aktuell order/tjänst skall kunna redigeras/uppdateras.

**Administrationswebb**

Adminwebben ska ge anställda på företaget möjlighet att administrera kunder och produkter/tjänster.

**Lista kunder:**

Visa en lista med kunder som visar kundens uppgifter och en summa av antal hyrestillfällen. Listan ska kunna sorteras baserat på alla egenskaper och filtreras på antal hyrtillfällen.

**Lista produkter/tjänster:**

Visa en lista med tillgängliga produkter/tjänster. Användaren ska kunna sortera tjänsterna/produkterna på namn och typ. På en rad ska bas-information om tjänsten/produkten finnas samt en knapp där användaren kan välja tjänsten/produkten och komma vidare till uppdatering av tjänsten/produkten. Raden ska också innehålla knappar för att ta bort tjänsten/produkten.

**Lägg till tjänst/produkt:**

När användaren lägger till en ny tjänst/produkt ska han/hon kunna fylla i egenskaper för tjänsten/produkten i ett formulär.

**Ta bort tjänst/produkt:**

När användare vill ta bort en tjänst/produkt ska den få en kontrollfråga för att bekräfta att den ska tas bort. Om tjänsten/produkten finns med i en aktuell uthyrning ska den kunden få en ny tjänst/produkt av samma typ.

**Uppdatera tjänst/produkt:**

Användare kan uppdatera tjänsten/produkten i en specialiserad vy som kan vara samma som lägg till tjänst/produkt.

**Säkerhet**

Implementeringen ska stödja CORS så att SPA klienten kan komma åt REST API:et. Ett minimikrav är att implementera basic security för inloggning av kunder och administratörer.

**Arbetssätt och teknik**

Ta fram en skiss (digital) som underlag för implementeringen innan ni startar arbetet. Skissen ska ligga som en pdf i projektet när det levereras.

Ta fram en style-guide med den CSS som du tänker använda och var konsekvent i användningar. Styleguiden ska bifogas i projektet.

Det är tillåtet att använda style-bibliotek som Bootstrap i projektet.

Både kundwebben och adminwebben ska utvecklas som en **SPA** med HTML5, CSS och JavaScript. React, Angular eller liknande ramverk får, efter godkännande av koncernen, användas som ramverk för att utveckla SPA-applikationen.

JQuery tillsammans med nativ JavaScript får användas.

**Lombok får inte användas**

Se till att ha grunddata som laddas i H2 databasen när API:et startas så du har något att jobba med.

Även om uppgiften är individuell och var och en ska lämna in en egenutvecklad version så uppmuntras ni diskutera under arbetet för att förstärka inlärningen.