

Nusuk System

Student name: Mauth Alnemari

Date: 2025/11/11

1. Introduction

Nusuk is the official digital platform launched by the Kingdom of Saudi Arabia to enhance the experience of pilgrims, Umrah performers, and visitors. The platform aims to enable users to perform their religious rituals easily and safely through an integrated set of e-services, including visa issuance, booking appointments for the Holy Mosques, managing authorized campaigns, and arranging accommodation and transportation within the holy sites.

Nusuk is part of the Kingdom's Vision 2030 digital transformation initiative, combining technology with religious services to provide a more efficient, transparent, and seamless experience for all users, both locally and internationally.

2.Model 1: PilgrimModel

```
@NotNull(message="Pilgrim ID is required")
@Min(value=1000, message="Pilgrim ID must be at least 4 digits")
private int pilgramId;

@NotEmpty(message="First name is required")
@Size(min =2,max=40,message="First name must be between 2 and 40
characters")
private String firstName;

@NotEmpty(message="Last name is required")
@Size(min= 2,max=40,message="Last name must be between 2 and 40
characters")
private String lastName;

@NotEmpty(message="Nationality is required")
private String nationality;

@NotEmpty(message="Passport number is required")
@Pattern(regexp= "[A-Z0-9]{8,10}",message="Passport number must be 8
to 10 uppercase letters or digits")
private String passportNumber;

@NotEmpty(message ="Visa type is required")
@Pattern(regexp = "Umrah|Hajj",message="Visa type must be one of: Umrah , Hajj")
private String visaType;

@NotEmpty(message="Email is required")
@email(message ="Email must be a valid format")
private String email;

@NotEmpty(message ="Phone number is required")
@Pattern(regexp = "05\d{8}",message ="Phone number must start with 05
and contain exactly 10 digits")
private String phoneNumber;

@NotNull(message ="Age is required")
@Min(value =18,message ="Age must be 18 or older")
private int age;
```

3.Model 2: CampaignModel

```
@NotNull(message ="Campaign ID is required")
@Min(value =100,message ="Campaign ID must be at least 3 digits")
private int campaignId;

@NotEmpty(message ="Campaign name is required")
@Size(min= 3,max =50,message="Campaign name must be between 3 and
50 characters")
private String name;

@NotEmpty(message="License number is required")
@Pattern(regexp="^\\d{8}$",message="License number must contain exactly
8 digits")
private String licenseNumber;

@NotEmpty(message="Country is required")
private String country;

@NotEmpty(message="Contact number is required")
@Pattern(regexp ="^(05\\d{8}||+9665\\d{8})$",message ="Contact number
must start with 05 or +9665 and contain 10 digits")
private String contactNumber;

@NotNull(message ="Capacity is required")
@Min(value= 1,message="Capacity must be at least 1")
@Max(value =500,message ="Capacity cannot exceed 500 pilgrims")
private int capacity;

@NotNull(message="Price per person is required")
@Positive(message="Price per person must be positive")
private double pricePerPerson;

@NotNull(message="Available from date is required")
@PastOrPresent(message="Available from date cannot be in the future")
private LocalDateTime availableFrom;

@NotNull(message ="Available to date is required")
@Future(message= "Available to date must be in the future")
private LocalDateTime availableTo;
```

```
@NotNull(message ="Rating is required")
@Min(value= 1,message ="Rating must be between 1 and 5")
@Max(value =5,message ="Rating must be between 1 and 5")
private int rating;
```

4. Model 3: BookingModel

```
@NotNull(message="Booking ID is required")
@Min(value =100, message ="Booking ID must be at least 3 digits")
private int bookingId;

@NotNull(message="Pilgrim information is required")
private PilgrimModel pilgrim;

@NotNull(message="Campaign information is required")
private CampaignModel campaign;

@NotEmpty(message ="Visit type is required")
@Pattern(regexp= "Umrah|Hajj",message ="Visit type must be one of:
Umrah,Hajj")
private String visitType;

@NotNull(message= "Visit date is required")
@FutureOrPresent(message= "Visit date cannot be in the past")
private LocalDateTime visitDate;

@NotNull(message= "Visit time is required")
private LocalDateTime visitTime;

@NotEmpty(message ="Location is required")
@Pattern(regexp ="Haram|Mina|Arafah|Muzdalifah", message="Location
must be one of: Haram, Mina, Arafah, Muzdalifah")
private String location;

@NotEmpty(message = "Status is required")
@Pattern(regexp= "Booked|Completed|Canceled",message="Status must be
one of: Booked, Completed, Canceled")
private String status;
```