This document outlines the API endpoints required for the Ticket Management System. The API will allow the frontend application to interact with ticket data, including retrieving, creating, updating, and deleting tickets.

**Base URL** : https://api.yourdomain.com/v1/

**Endpoints**

1. **Get All Tickets**

* Endpoint: /tickets
* Method: GET
* Description: Retrieves a list of all tickets.
* Response:
  + 200 OK

[

{

"ticketNumber": "123",

"tableNumber": 1,

"waiterName": "John Doe",

"orderDateTime": "2024-01-01T12:00:00Z",

"items": [

{

"itemName": "RITA JUM HH",

"quantity": 2,

"specialItems": []

}

],

"isCompleted": false,

"isTimeExceeded": false

}

]

1. **Get Ticket by Number**

* Endpoint: /tickets/{ticketNumber}
* Method: GET
* Description: Retrieves a ticket by its number.
* Parameters:
  + ticketNumber: The unique identifier for the ticket.
* Response:
  + 200 OK

{

"ticketNumber": "123",

"tableNumber": 1,

"waiterName": "John Doe",

"orderDateTime": "2024-01-01T12:00:00Z",

"items": [

{

"itemName": "RITA JUM HH",

"quantity": 2,

"specialItems": []

}

],

"isCompleted": false,

"isTimeExceeded": false

}

* + 404 Not Found

{

"message": "Ticket not found."

}

1. **Create a Ticket**

* Endpoint: /tickets
* Method: POST
* Description: Creates a new ticket.
* Request Body:

{

"ticketNumber": "124",

"tableNumber": 2,

"waiterName": "Jane Doe",

"orderDateTime": "2024-01-01T12:30:00Z",

"items": [

{

"itemName": "CCQ BLANCO lg",

"quantity": 1,

"specialItems": []

}

]

}

* Response:
  + 201 Created

{

"message": "Ticket created successfully."

}

1. **Update a Ticket**

* Endpoint: /tickets/{ticketNumber}
* Method: PUT
* Description: Updates an existing ticket.
* Parameters:
  + ticketNumber: The unique identifier for the ticket.
* Request Body:

{

"tableNumber": 3,

"waiterName": "John Smith",

"items": [

{

"itemName": "TEA",

"quantity": 2,

"specialItems": []

}

]

}

* Response:
  + 200 OK

{

"message": "Ticket updated successfully."

}

**5. Delete a Ticket**

* Endpoint: /tickets/{ticketNumber}
* Method: DELETE
* Description: Deletes a ticket by its number.
* Parameters:
  + ticketNumber: The unique identifier for the ticket.
* Response:
  + 204

No Content

* + 404 Not Found

{

"message": "Ticket not found."

}

**6. Complete a Ticket**

* Endpoint: /tickets/{ticketNumber}/complete
* Method: PATCH
* Description: Marks a specified ticket as completed.
* Parameters:
  + ticketNumber: The unique identifier for the ticket.
* Request Body: *(Optional, can be empty)*

{}

* Response:
  + 200 OK

{

"message": "Ticket marked as completed."

}

* + 404 Not Found

{

"message": "Ticket not found."

}

**Data Models**

**Ticket**

public class Ticket

{

public string TicketNumber { get; set; }

public int TableNumber { get; set; }

public List<Items> Items { get; set; }

public string WaiterName { get; set; }

public DateTime OrderDateTime { get; set; }

public int ItemNumber { get; set; }

public bool IsCompleted { get; set; }

public bool IsTimeExceeded { get; set; }

}

**Items**

public class Items

{

public string ItemName { get; set; }

public int Quantity { get; set; }

public List<SpecialItems> SpecialItems { get; set; }

public bool IsCompleted { get; set; }

}

**SpecialItems**

public class SpecialItems

{

public string SpecialItemName { get; set; }

}

\*Items and SpecialItems can be empty arrays if there are no items to list.