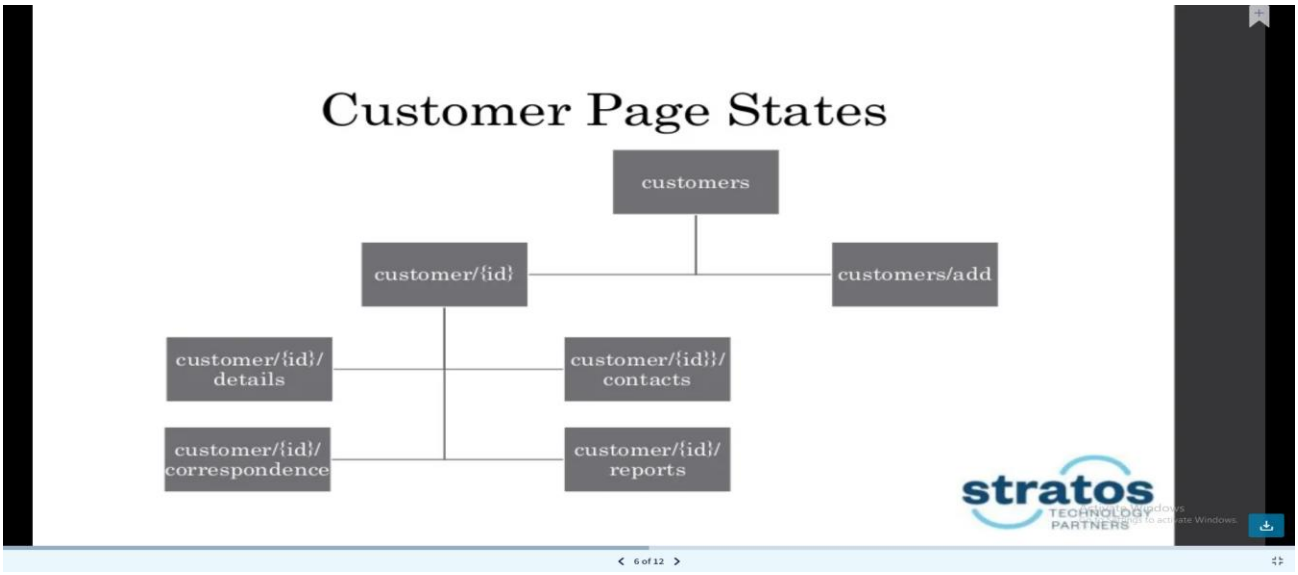
Case Study on Angular

Case Study: Customer Page States



Objective:

To streamline the navigation and user interaction on the customer management platform by organizing customer-related pages into a structured hierarchy.

Hierarchy Overview:

1. **Main Customer States:**

- **customers**: The main page that lists all customers.
- **customers/add**: A dedicated page for adding a new customer to the system.

2. **Customer-Specific Pages:** Each customer has a unique ID, represented as {id}, and their information is accessible via specific states:
    - **customer/{id}**: The main dashboard or overview page for a specific customer.
    - **Child States of customer/{id}**:
        - **customer/{id}/details**: Provides detailed information about the customer.
        - **customer/{id}/contacts**: Lists the customer's associated contacts.
        - **customer/{id}/correspondence**: Displays correspondence or communication history.
        - **customer/{id}/reports**: Shows reports related to the customer.

Use Case Scenarios:

1. **Customer Management:**
    - A user navigates to the customers page to view all registered customers.
    - If a new customer needs to be added, the user transitions to customers/add.
2. **Accessing Customer Information:**
    - Clicking on a specific customer navigates the user to customer/{id}.
    - From there, users can dive deeper into:
        - Viewing customer-specific **details**.
        - Managing customer **contacts**.
        - Reviewing customer **correspondence** for historical communication logs.
        - Analyzing **reports** for insights.

Benefits of the Page Structure:

1. **User-Friendly Navigation:** The hierarchy ensures clear, intuitive paths for users to follow based on their specific needs (e.g., adding customers or viewing reports).
2. **Scalability:** Adding new customer-specific functionalities (like billing or orders) is straightforward, as new states can seamlessly branch from customer/{id}.

3. **Separation of Concerns:** The structure isolates different concerns, like details, contacts, and reports, reducing complexity on individual pages and improving performance.

Challenges and Considerations:

1. **Data Loading:** Ensure efficient data fetching for child states (details, contacts, etc.) to avoid performance lags.
2. **Access Control:** Implement role-based access to ensure users only view or edit customer data they're authorized for.
3. **Consistency:** Maintain consistent UI/UX patterns across states for a smooth user experience.
