# Reference Document for CRUD Operations Assignment

## 1. Introduction

This document provides a comprehensive guide to complete the CRUD operations assignment using NativeScript, Angular, Node.js, and CockroachDB. The assignment aims to demonstrate proficiency in these technologies by implementing functional CRUD operations for entities including "Customer" and "Multiple Addresses" in both mobile and web environments.

## 2. Step-by-Step Instructions

### Mobile CRUD Operations

#### Create a New Customer

1. **Validate Input Fields**:

- Ensure all required fields (first name, last name, phone number, address details, city, state, pin code) are filled.

- Perform validation checks (e.g., phone number format, pin code length).

2. **Create Customer Record**:

- Trigger a function to save validated data to the database.

- Display a success message upon successful creation.

#### Read Existing Customer Details

1. **Navigate to Customer Profile**:

- Obtain the customer ID from the database.

- Display customer details in a clear, organized layout on the profile screen.

#### Update Customer Information

1. **Modify Customer Fields**:

- Fetch the customer data and allow modification of required fields.

- Save the changes and display a confirmation message.

#### Delete Customer Record

1. **Search and Select Customer**:

- Navigate to the customer list screen and locate the customer by ID or other parameters.

- Confirm the deletion action before permanently removing the data.

#### View Available Multiple Addresses

1. **Search Functionality**:

- Implement a search to locate customers with multiple addresses.

- List these addresses along with corresponding customer IDs.

#### Save Updated Multiple Address

1. **Modify and Save Addresses**:

- Allow modification of address details while preserving the customer ID association.

- Update the address through a server API call.

#### Mark Customer as Having Only One Address

1. **Set One Address Flag**:

- Select profiles where only one address exists and set an appropriate flag.

- Invalidate the setting for customers with multiple addresses.

#### Search by City, State, or Pincode

1. **Implement Search Options**:

- Provide filters to search customer records by city, state, or zip code.

- Display the filtered results alongside customer IDs.

#### Clear Filters

1. **Reset Filters**:

- Ensure all applied filters during searches or modifications can be reset for unrestricted viewing.

#### Page Navigation

1. **Implement Paging System**:

- Provide functionalities to navigate between pages containing customer data.

- Include sorting options for efficient data retrieval.

### Web CRUD Operations

#### Create New Customer

1. **Design Input Form**:

- Include mandatory fields with client-side validation.

- Handle server-side rules to prevent duplicates and ensure data validity.

- Redirect to the customer dashboard on successful validation.

#### Read Existing Customer Data

1. **Retrieve and Display Data**:

- Fetch customer-related info using their unique identifier.

- Organize and display information in respective sections/tabs.

#### Update Customer Information

1. **Edit and Save Changes**:

- Facilitate editing of personal, shipping, and billing information.

- Ensure immediate updates and provide instant feedback.

#### Delete Customer Record

1. **Select and Confirm Deletion**:

- Identify the customer record and validate linked transactions before deletion.

- Provide confirmation messages during the process.

#### View Available Multiple Addresses

1. **Display Addresses with Filters**:

- Show all multiple addresses linked to a customer.

- Provide filtering options based on location attributes.

#### Save Updated Multiple Address

1. **Modify and Update Addresses**:

- Dynamically update address status through server API calls.

#### Mark Customer as Having Only One Address

1. **Set One Address Flag**:

- Check for single registered addresses, set a flag, and display on the profile view.

#### Search by City, State, or PIN

1. **Full-Text Searches**:

- Enable searches on customer attributes and display matching records.

#### Clear Filters

1. **Reset All Filters**:

- Reset search queries and filter settings for complete data viewing.

#### Page Navigation

1. **Infinite Scrolling and Sorting**:

- Support infinite scrolling and sorting options for efficient data presentation.

#### Responsive Design

1. **Optimize for Various Devices**:

- Use CSS media queries to ensure compatibility across different screen sizes.

### Error Handling & Logging System

1. **Establish Error Detection**:

- Implement mechanisms for error detection and logging during CRUD operations.

### Test Case Scenarios

1. **Write and Execute Tests**:

- Ensure correctness by writing and executing test cases for each function.

- Identify and rectify bugs for optimal performance.

## 3. Best Practices

- **Code Clarity**: Maintain clean and readable code with appropriate comments.

- **Consistent Validation**: Perform thorough client and server-side validations.

- **User Feedback**: Provide clear and immediate user feedback on actions.

- **Error Handling**: Implement comprehensive error handling and logging mechanisms.

- **Testing**: Write extensive test cases to cover all possible scenarios.

## 4. Submission Guidelines

- **GitHub Repository**: Create a repository for the code, upload the complete project, and grant access to shilpa.chaladi@qwipo.com, kranthi.lebaka@qwipo.com, and siva.mamidi@qwipo.com.

- **Demo Video**: Record a video demonstrating the functionalities implemented and upload it as part of the submission.

- **Interview**: Upon code review, an interview call will be scheduled.

## 5. Frequently Asked Questions (FAQ)

1. **What if I encounter issues with technology setup?**

- Refer to the official documentation for NativeScript, Angular, Node.js, and CockroachDB for setup instructions.

2. **How do I handle duplicate entries in the database?**

- Implement server-side checks to prevent duplicates and inform the user appropriately.

3. **What should be included in the demo video?**

- The video should showcase all CRUD functionalities and navigational mechanisms implemented.

4. **How do I ensure my application is responsive?**

- Test your application on various devices and use CSS media queries for adjustments.

5. **What is the best way to handle errors?**

- Implement error handling at every stage of CRUD operations and log errors for troubleshooting.

By following this document, candidates should be able to successfully complete the CRUD operations assignment and showcase their proficiency in the specified tech stack.