

Order Dispute Notification System

Overview

This feature automatically sends dispute notifications to appropriate personnel when an order is created with engineer or partner requirements set to "No". This helps ensure proper oversight and dispute handling for orders that may require additional review.

Notification Logic

When Engineer Required = "No"

- **Who gets notified:** All employees with role "BDO" (Business Development Officer)
- **Purpose:** BDO can review if the order actually needs engineer services and submit a dispute if necessary
- **Message Type:** Engineer Dispute Notification

When Partner/Contractor Required = "No"

- **Who gets notified:** All employees with role "CRO" (Customer Relations Officer)
- **Purpose:** CRO can review if the order actually needs partner/contractor services and submit a dispute if necessary
- **Message Type:** Partner Dispute Notification

When Both Are Set to "No"

- **Result:** Both BDO and CRO receive their respective dispute notifications
- **Purpose:** Comprehensive review by both teams

Technical Implementation

Files Modified

1. **order.js** - Added dispute notification logic to `sendOrderNotifications()` function
2. **test-order-dispute.js** - Test cases for the new functionality

New Functions Added

`sendDisputeNotifications(orderData, potentialSiteInfo, submitterEmployee)`

- Main function that checks engineer/partner requirements and sends appropriate notifications
- Looks up BDO and CRO employees using `findEmployeesByRole()`
- Sends WhatsApp notifications to relevant personnel

`createDisputeNotificationMessage(orderData, potentialSiteInfo, submitterEmployee, disputeType, reason)`

- Creates formatted dispute notification messages
- Includes order details, project information, and action required
- Different messages for engineer vs partner disputes


Integration Points


- Automatically triggered when `handleOrderFormSubmit()` processes new orders
- Uses existing WhatsApp notification system (`sendWhatsAppMessage()`)
- Leverages existing employee lookup functions (`findEmployeesByRole()`)

Message Format

Engineer Dispute Notification (to BDO)

 DISPUTE NOTIFICATION REQUIRED

 Engineer Requirement Alert


 Order Details:

Order ID: ORD-001

Order Type: Cement Order

Potential Site: P.S-001


Site Name: Construction Site Name

 Project Information:


Address: Project Address

Building: Ground Floor to 3rd Floor

Quantity: 100 bags cement

 Dispute Reason: No Engineer Required

The customer has indicated that engineer services are NOT required for this order.

 Requirements Status:

Engineer Required: No

Partner/Contractor Required: Yes

 Order Submitted by: Employee Name

 Contact: email@anwar.com

 Order Time: Aug 2, 2025, 6:30:00 PM

 ACTION REQUIRED:

Please review this order and submit a dispute if engineer services are actually needed for this type of project.

Use the Dispute Creation form to escalate this issue if necessary.

Partner Dispute Notification (to CRO)

Similar format but with 🤝 emoji and "Partner/Contractor" terminology.

Testing

Test File: `test-order-dispute.js`

Contains comprehensive test cases:

1. Engineer Required = "No" (BDO notification)
2. Partner Required = "No" (CRO notification)
3. Both = "No" (Both BDO and CRO notifications)
4. Both = "Yes" (No dispute notifications)
5. Message formatting tests

Running Tests

```
// In Apps Script Editor  
runOrderDisputeTests();
```

Configuration Requirements

Employee Data Setup

- Employees must be properly configured in the EMPLOYEES sheet
- BDO employees must have role = "BDO"
- CRO employees must have role = "CRO"
- WhatsApp numbers must be populated for notifications to work

Form Configuration

The ORDER_CREATION form must include:

- "Is Engineer Required?" - Multiple choice: ['Yes', 'No']
- "Is Partner/Contractor Required?" - Multiple choice: ['Yes', 'No']

Error Handling

- Graceful handling when no BDO/CRO employees are found
- Logging of all notification attempts and failures
- Continues processing even if notifications fail
- Individual error handling for each notification recipient

Future Enhancements

1. Territory-based dispute routing (send to BDO/CRO in specific territories)
2. Escalation timers (auto-escalate if no response within X hours)
3. Integration with actual dispute creation workflow

4. Analytics on dispute notification frequency by territory/order type