Trade Document Processing Platform - Technical Requirements Document

Version: 1.0
Purpose: Development guide for AI prototyping tool

---

1. Platform Overview

A web-based platform for processing trade compliance documents using Claude AI. The platform classifies PDF documents, extracts specific fields based on document type, and stores the data in a structured format with manual review capabilities for low-confidence extractions.

Document Types Supported:

- Shipping Bill
- Logistics Document
- Invoice
- FIRA/FIRC (Payment Advice)
- Bank Statement (future)

---

2. Core Functionality

2.1 Customer Management

Customer Model

```json
{
  "id": "unique_identifier",
  "name": "customer_name",
  "bank": "bank_name",
  "created_at": "timestamp"
}
```

Features Required:

- **Add Customer**: Form with name and bank fields
- **View Customers**: Grid/card display of all customers
- **Customer Selection**: Click to access customer profile

2.2 Customer Profile Structure

Each customer profile has two main sections:

2.2.1 Documents Section

- **Upload Documents**:
  - Multiple PDF upload capability
  - Show upload progress
  - PDF validation (reject non-PDF files)
- **Document Flags**:
  - Display documents classified as "Not Specified"
  - Show fields with low confidence scores
  - Interface for manual classification correction
  - Interface for manual field value correction

2.2.2 Shipment Data Section

Four tabs for different document types:

- **Shipment Table** (from Shipping Bills)
- **Invoice Table** (from Invoices)
- **Logistics Table** (from Logistics Documents)
- **Transactions Table** (from FIRA/FIRC)

Each table displays extracted data with confidence indicators.

---

3. Document Processing Pipeline

3.1 Step 1: Document Classification

Process:

1. User uploads PDFs
2. Each PDF is sent to Claude Haiku API with classification prompt
3. Returns one of: "Logistics Document", "Invoice", "Shipping Bill", "FIRA/FIRC", "Bank Statement", "Not Specified"
4. Documents marked "Not Specified" are flagged for manual review

API Integration:

- Model: Claude Haiku
- Use classification prompt from prompt sheet

- Handle API errors gracefully
- Implement retry logic

3.2 Step 2: Field Extraction

Process:

1. Based on classification, select appropriate extraction prompt
2. Send document to Claude Sonnet 4 API
3. Receive JSON response with fields and confidence scores
4. Store extracted data in appropriate table

Confidence Levels:

- **High (90-100%)**: Process automatically
- **Medium (70-89%)**: Process but mark for optional review
- **Low (<70%)**: Flag for mandatory manual review

---

4. Data Schemas

4.1 Logistics Document Schema

```json
{
  "primary_transport_id": {"value": "string", "confidence": "High/Medium/Low"},
  "shipping_bill_number": {"value": "string", "confidence": "High/Medium/Low"},
  "invoice_number": {"value": "string", "confidence": "High/Medium/Low"},
  "document_date": {"value": "string", "confidence": "High/Medium/Low"},
  "transport_type_detected": "Ocean/Air/Postal/Multi-modal"
}
```

4.2 Invoice Schema

```json
{
  "invoice_number": {"value": "string", "confidence": "High/Medium/Low"},
  "invoice_date": {"value": "string", "confidence": "High/Medium/Low"}
}
```

4.3 Shipping Bill Schema

```json
{
  "sb_number": {"value": "string", "confidence": "High/Medium/Low"},
  "sb_date": {"value": "string", "confidence": "High/Medium/Low"},
  "cb_name": {"value": "string", "confidence": "High/Medium/Low"},
  "port_of_loading": {"value": "string", "confidence": "High/Medium/Low"},
  "hawb_number": {"value": "string", "confidence": "High/Medium/Low"},
  "iec_number": {"value": "string", "confidence": "High/Medium/Low"},
  "port_of_final_destination": {"value": "string", "confidence": "High/Medium/Low"},
  "account_number": {"value": "string", "confidence": "High/Medium/Low"},
  "invoice_term": {"value": "string", "confidence": "High/Medium/Low"},
  "fob_value": [
    {"currency": "string", "value": "string", "confidence": "High/Medium/Low"}
  ],
  "exporter_name_address": {"value": "string", "confidence": "High/Medium/Low"},
  "consignee_name_address": {"value": "string", "confidence": "High/Medium/Low"},
  "invoices": [
    {
      "invoice_number": {"value": "string", "confidence": "High/Medium/Low"},
      "invoice_date": {"value": "string", "confidence": "High/Medium/Low"},
      "invoice_value": {"value": "string", "confidence": "High/Medium/Low"}
    }
  ],
  "ad_code": {"value": "string", "confidence": "High/Medium/Low"},
  "buyer_name_address": {"value": "string", "confidence": "High/Medium/Low"},
  "freight": {"value": "string", "confidence": "High/Medium/Low"},
  "insurance": {"value": "string", "confidence": "High/Medium/Low"},
  "discount": {"value": "string", "confidence": "High/Medium/Low"},
  "commission": {"value": "string", "confidence": "High/Medium/Low"}
}
```

4.4 FIRA/FIRC Schema

```json
{
  "provider": {"value": "string", "confidence": "High/Medium/Low"},
  "utr_number": {"value": "string", "confidence": "High/Medium/Low"},
  "date": {"value": "string", "confidence": "High/Medium/Low"},
  "total_settlement_amount_inr": {"value": "string", "confidence": "High/Medium/Low"},
  "account_number": {"value": "string", "confidence": "High/Medium/Low"},
  "remitter": {"value": "string", "confidence": "High/Medium/Low"},
  "receiver": {"value": "string", "confidence": "High/Medium/Low"},
  "purpose_code": {"value": "string", "confidence": "High/Medium/Low"},
  "transaction_breakup": [
    {
      "reference_no": {"value": "string", "confidence": "High/Medium/Low"},
      "buyer_name": {"value": "string", "confidence": "High/Medium/Low"},
      "buyer_address": {"value": "string", "confidence": "High/Medium/Low"},
      "buyer_country": {"value": "string", "confidence": "High/Medium/Low"},
      "date": {"value": "string", "confidence": "High/Medium/Low"},
      "amount_inr": {"value": "string", "confidence": "High/Medium/Low"},
      "amount_foreign_currency": {"value": "string", "confidence": "High/Medium/Low"},
      "currency": {"value": "string", "confidence": "High/Medium/Low"}
    }
  ]
}
```

---

5. User Interface Requirements

5.1 Main Dashboard

- Display all customers in a grid
- "Add Customer" button (opens modal/form)
- Click on customer card to enter profile

5.2 Customer Profile Page

- Header with customer name and bank
- Tab navigation: "Documents" | "Shipment Data"

5.3 Documents Tab

- **Upload Section**:
  - Drag-and-drop area or file picker
  - Upload progress indicators
  - Success/error messages

- **Document Flags Section**:
  - Table showing flagged documents/fields
  - Columns: Document Name, Issue Type, Field Name, Current Value, Action
  - Edit capability for manual correction
  - Save button to update values

5.4 Shipment Data Tab

- Sub-tabs for each document type
- Each sub-tab contains:
  - Table with all extracted fields as columns
  - Confidence indicator (icon/color) for each cell
  - Sort and filter capabilities
  - Export to CSV functionality

---

6. Processing Flow

```
1. User uploads PDFs to customer profile
   ↓
2. System queues documents for processing
   ↓
3. Classification API call (Claude Haiku)
   ↓
4. Route based on classification result:
   - If "Not Specified" → Flag for manual review
   - Else → Continue to extraction
   ↓
5. Extraction API call (Claude Sonnet 4)
   ↓
6. Parse JSON response and store in database
   ↓
7. Check confidence scores:
   - Low confidence → Flag for review
   - Medium/High → Display in tables
   ↓
8. Update UI with results
```

---

7. API Integration Requirements

7.1 Claude API Configuration

- Store API keys securely

- Implement request queuing to handle rate limits
- Add retry logic for failed requests
- Log all API interactions for debugging

7.2 Prompt Management

- Store prompts in configuration (not hardcoded)
- Use exact prompts from provided prompt sheet
- Include document content in `[DOCUMENT_CONTENT]` placeholder

7.3 Error Handling

- Network errors: Retry with exponential backoff
- API errors: Display user-friendly message
- Parsing errors: Flag document for manual review
- Validation errors: Log and flag for review

---

8. Data Storage Requirements

8.1 Database Tables

- **customers**: Store customer information
- **documents**: Store document metadata and S3 references
- **logistics_data**: Store extracted logistics document data
- **invoice_data**: Store extracted invoice data
- **shipping_bill_data**: Store extracted shipping bill data
- **fira_firc_data**: Store extracted FIRA/FIRC data
- **manual_corrections**: Audit trail of manual changes

8.2 File Storage

- Store original PDFs in cloud storage (S3 or equivalent)
- Organize by customer ID and document type
- Maintain reference in database

---

9. Key Implementation Notes

1. **Confidence Score Display**: Use color coding (green=high, yellow=medium, red=low)
2. **Manual Review Interface**: Inline editing with save confirmation
3. **Bulk Operations**: Process multiple documents in parallel

4. **Status Tracking**: Show processing status for each document
5. **Data Validation**: Validate extracted data format before storage
6. **Array Handling**: Properly handle array fields (multiple invoices, transactions)
7. **Field Mapping**: Ensure exact field name matching from prompts to schema