

# I&C Billing Tool - Comprehensive Application Analysis

## Executive Summary

**I&C Billing Tool** is a Python-based desktop application designed for I&C Shipping and Logistics company to automate the generation of customer invoices from Excel data exported from China. The application processes shipping manifest data, calculates charges based on cubic meter (CBM) measurements, and generates professional PDF invoices along with WhatsApp messaging templates for customer notifications.

## 1. Framework & Technology Stack

### Primary Technologies

- **GUI Framework:** **Tkinter** (Python’s built-in GUI library)
- **Application Type:** Desktop application (standalone executable)
- **Programming Language:** Python 3.10+ (uses modern type hints like `tuple[str, str]` )

### Key Dependencies

Library	Purpose
pandas	Excel file parsing and data manipulation
jinja2	HTML template rendering for invoice generation
weasyprint	PDF generation from HTML templates
xlsxwriter	Excel export functionality for summary reports
Standard library modules: <code>tkinter</code> , <code>csv</code> , <code>datetime</code> , <code>pathlib</code> , <code>dataclasses</code> , <code>uuid</code> , <code>re</code> , <code>math</code>	

### Architecture Pattern

- **Desktop GUI Application** with file input/output operations
- **Data Pipeline:** Excel → Pandas DataFrame → Bill Objects → PDF/CSV/Excel outputs
- **Template-based rendering:** Jinja2 templates for dynamic HTML/PDF generation

## 2. Key Features & Functionality

---

### Core Features



#### Excel Data Processing

- Reads shipping manifest Excel files exported from China
- Automatically removes container header lines (lines containing “N005=” or “N006=”)
- Implements intelligent data parsing with fill-down for continuation lines
- Groups shipments by customer (using phone number as primary identifier)



#### Intelligent Billing Calculations

- **Rate-based charges:** USD per CBM (Cubic Meter)
- **Minimum charge rule:** Items below 0.05 CBM are charged a fixed \$10.00
- **Additional costs:** Optional “Other Cost” field for extra charges
- **Automatic totaling:** Subtotal + Other Costs = Total



#### Multi-Format Output Generation

1. **PDF Invoices** - Professional, branded invoices for each customer
2. **Summary Excel** - Consolidated billing summary with all calculations
3. **WhatsApp CSV** - Ready-to-send messages with billing details for each customer



#### Customer Grouping Logic

- Groups multiple shipping marks per customer
- Primary identifier: Phone number
- Fallback identifiers: Customer name → Customer ID
- Combines all CBM measurements for a single customer into one invoice

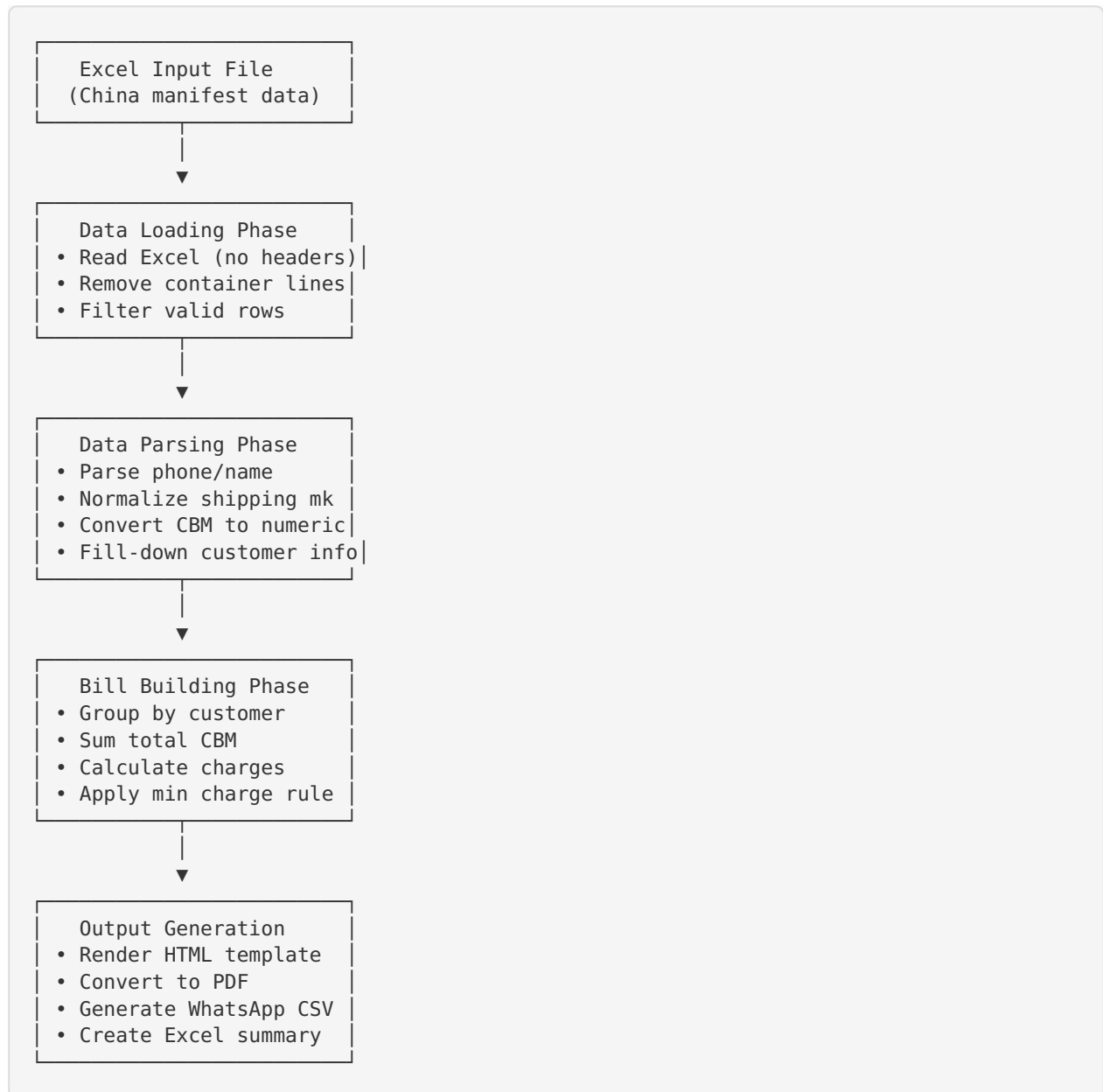


#### Batch Processing

- Processes multiple customers in a single operation
  - Creates timestamped output folders for organization
  - Generates unique invoice numbers (format: `1C[YEAR][8-digit-UUID]` )
-

## 3. Invoice Generation Workflow

### Step-by-Step Process



### PDF Generation Technical Details

1. **Template Rendering:** Jinja2 processes `template_invoice.html` with bill data
2. **HTML to PDF Conversion:** WeasyPrint converts rendered HTML to A4 PDF
3. **Asset Embedding:** Logo image ( `logo.png` ) must be in same directory as app
4. **Styling:** CSS-in-HTML with professional formatting and proper page margins

## 4. Invoice Template Fields & Structure

### Document Header Section

- **Company Logo:** I&C Shipping and Logistics branded logo

- **Chinese Address:** 广东省佛山市南海区里水镇北沙村石龙潭工业区9号加纳仓
- **Document Title:** I&C LOGISTICS GOODS BILL

Invoice Metadata

Field	Description	Example
Invoice Number	Unique identifier	1C2025XXXXXXXX
Invoice Date	Generation date	12TH JAN, 2026

Customer Information (“Bill To”)

- Customer Name
- Location (default: “ACCRA GHANA”)
- Phone Number

Service Type

- Fixed value: “SEA CARGO”

Billing Details Table

Field	Purpose
ITEM DESCRIPTION	Type of goods being shipped
TOTAL CBM	Rate per cubic meter (USD)
CBM [value]	Total cubic meters for this customer
PAYMENT DETAILS	Calculation formula (e.g., “240*0.42”)
Subtotal	CBM × Rate (or \$10 if < 0.05 CBM)
OTHER COST	Additional charges
Total Cost	Final amount due

Legal Terms & Conditions

1. Payment instructions (checks payable to I&C Shipping and Logistics)
2. Pickup deadline (1 week to avoid 1% daily storage fee)
3. Abandonment clause (1 month unpaid goods may be sold)
4. Shipping mark requirement
5. Packaging requirements for breakable goods
6. Liability disclaimer for improperly packaged items

Contact Information

- Phone: 0552161900/0549009957

## Important Notes

- **Minimum charge rule:** CBM below 0.05 = fixed \$10.00
- **Invoice Breakdown:** Lists all shipping marks for this customer

---

## 5. Dependencies & Requirements

### Python Package Requirements

```
pandas>=1.5.0
jinja2>=3.0.0
weasyprint>=60.0
xlsxwriter>=3.0.0
openpyxl>=3.0.0 # For reading .xlsx files
```

### System Requirements

- **Python Version:** 3.10 or higher (uses modern type syntax)
- **Operating System:** Cross-platform (Windows, macOS, Linux)
- **Display:** GUI requires graphical display environment
- **WeasyPrint Dependencies:**
- **Windows:** GTK3 Runtime required
- **macOS:** Cairo, Pango (via Homebrew)
- **Linux:** libpango, libcairo packages

### External Files Required

File	Location	Purpose
template_invoice.html	Same directory as app.py	Invoice template
logo.png	Same directory as app.py	Company logo for PDFs

### Installation Command

```
pip install pandas jinja2 weasyprint xlsxwriter openpyxl
```

---

## 6. Code Structure & Organization

### Architecture Overview

```

app.py (Single-file application)
├── Helpers Section
│   ├── money_usd()          - Format currency
│   ├── safe_filename()      - Sanitize file names
│   ├── parse_phone_name()   - Extract phone/name from text
│   ├── normalize_shipping_mark() - Clean shipping marks
│   └── is_container_header_line() - Detect header rows
├── Data Models
│   └── Bill (dataclass)      - Represents customer invoice
│       ├── Properties: shipping_mark, customer_name, phone, etc.
│       └── Computed: subtotal_usd, min_charge_usd, total_usd
├── Excel Processing
│   ├── load_and_prepare_rows() - Parse Excel into DataFrame
│   └── build_bills()          - Group data into Bill objects
├── Export Functions
│   ├── render_pdf_for_bill() - Generate PDF invoice
│   ├── make_whatsapp_message() - Create WhatsApp text
│   ├── export_summary_xlsx() - Create Excel summary
│   └── export_whatsapp_csv() - CSV with messages
└── GUI Application
    └── App (Tkinter)
        ├── __init__()      - Setup window
        ├── _build()         - Create UI elements
        ├── select_excel()   - File picker
        ├── select_out_dir() - Directory picker
        └── generate()        - Main processing
  
```

### Design Patterns Used

#### 1. Dataclass Pattern

```

@dataclass
class Bill:
    shipping_mark: str
    customer_id: Optional[str]
    customer_name: str
    # ... fields ...

    @property
    def total_usd(self) -> float:
        # Computed property
  
```

- Clean data encapsulation
- Automatic `__init__`, `__repr__`, `__eq__`
- Type hints for safety

#### 2. Single Responsibility Functions

- Each function has one clear purpose
- Easy to test and maintain

- Reusable components

### 3. Template Pattern

- Jinja2 separates presentation from logic
- Easy to modify invoice design without code changes

### 4. Pipeline Processing

- Data flows through clear stages
  - Each stage transforms data predictably
- 

## 7. Notable Strengths

---



### Technical Excellence

#### 1. Robust Data Parsing

- Handles messy Excel formats from real-world sources
- Intelligent fill-down for multi-line entries
- Phone/name parsing with fallback logic
- Filters out invalid rows automatically

#### 2. Type Safety

- Modern Python type hints throughout
- Prevents common bugs at development time
- Self-documenting code

#### 3. Error Handling

- Validates user inputs (file existence, numeric values)
- User-friendly error messages via message boxes
- Graceful handling of missing/NaN values

#### 4. Professional Output Quality

- High-quality PDF generation (A4, proper margins)
- Branded company logo integration
- Clean, readable invoice design
- Legal terms and conditions included

#### 5. User Experience

- Simple, intuitive GUI
- Real-time progress logging
- Automatic output folder organization with timestamps
- Default values pre-filled (rate: 240, other cost: 0)
- One-click batch processing

#### 6. Business Logic Accuracy

- Minimum charge rule correctly implemented
- Clear calculation transparency (shows formula)
- Handles edge cases (CBM < 0.05)
- Multiple shipping marks per customer properly combined

## 7. Multi-Channel Communication

- PDF for formal documentation
- WhatsApp messages for quick customer notification
- Excel summary for business analytics
- All outputs generated simultaneously

## 8. Areas for Potential Improvement

### Suggested Enhancements

#### 1. Configuration Management

**Current State:** Hardcoded values in code

```
location_default: str = "ACCRA GHANA"
invoice_no = "1C" + dt.datetime.now().strftime("%Y") + ...
```

**Improvement:**

- Create `config.json` or settings dialog for:
- Default location
- Invoice number prefix/format
- Company contact details
- Rate presets
- Benefits: No code editing needed for business changes

#### 2. Invoice Numbering

**Current State:** Random UUID-based (not sequential)

```
invoice_no = "1C" + dt.datetime.now().strftime("%Y") + str(uuid.uuid4().int)[0:8]
```

**Improvement:**

- Sequential numbering with persistent counter (stored in file/database)
- Format: IC-2025-0001 , IC-2025-0002 , etc.
- Benefits: Professional, trackable, audit-friendly

#### 3. Error Recovery

**Current State:** Stops on first error during batch generation

**Improvement:**

- Try-catch around individual bill generation
- Log failed bills, continue processing others
- Summary report showing success/failure count
- Benefits: Doesn't lose entire batch for one bad record

#### 4. Preview Functionality

**Current State:** No preview before generation

**Improvement:**

- "Preview" button to show first invoice in browser



- Review parsed data before PDF generation
- Benefits: Catch issues early, save time/paper

## 5. Input Validation Enhancement

**Current State:** Basic validation on generate

### Improvement:

- Real-time validation as user types
- Visual indicators (red/green borders)
- Excel preview showing detected rows/customers
- Benefits: Prevent user errors proactively

## 6. Excel Template Flexibility

**Current State:** Hardcoded column positions (A=0, B=1, etc.)

```
work = raw[[0, 1, 2, 3, 4, 5]].copy()
```

### Improvement:

- Column mapping interface
- Support for different Excel formats
- Auto-detect columns by header names
- Benefits: Works with varied supplier formats

## 7. Historical Tracking

**Current State:** No record of past generations

### Improvement:

- SQLite database to track:
- All generated invoices
- Customer history
- Payment status tracking
- Benefits: Business intelligence, dispute resolution

## 8. Packaging & Distribution

**Current State:** Requires Python installation

### Improvement:

- Use **PyInstaller** or **cx\_Freeze** to create standalone executable
- No Python installation needed for end users
- Benefits: Easy deployment, professional appearance

## 9. Export Customization

**Current State:** Fixed output formats

### Improvement:

- Checkboxes for optional outputs (PDF/WhatsApp/Excel)
- PDF filename template customization
- Email integration for sending invoices
- Benefits: Flexibility for different workflows

## 10. Multi-Language Support

**Current State:** English/Chinese mixed

**Improvement:**

- Language selection for invoice template
- Separate templates for different markets
- Benefits: International expansion ready

## 11. Logging System

**Current State:** GUI text widget logging only

**Improvement:**

- File-based logging with rotation
- Log levels (DEBUG, INFO, WARNING, ERROR)
- Separate log file per run
- Benefits: Troubleshooting, audit trail

## 12. Unit Tests

**Current State:** No automated tests

**Improvement:**

- Test suite for:
- Data parsing functions
- Bill calculation logic
- Edge cases (empty fields, special characters)
- Benefits: Confidence in code changes, catch regressions

## 9. Data Flow Example

### Sample Input (Excel)

A	B	C	D	E	F
CUST001	12345	0202425612	BLESSING	...	0.42 SHOES
					0.18 BAGS
CUST002	67890	61466818614	...	0.03	ELECTRONICS

### Processed Output Structure

```
Output Folder: IC_OUTPUT_20260112_143022/
├── PDFs/
│   ├── CUSTOMER - BLESSING KUMASI - 0202425612.pdf
│   └── CUSTOMER - UNKNOWN - 61466818614.pdf
├── WhatsApp_Messages.csv
└── Summary.xlsx
```

## WhatsApp Message Example

I&C CARGO – GOODS BILL  
Name: BLESSING KUMASI  
Phone: 0202425612  
Shipping Mark: 12345  
Total CBM: 0.60  
Rate: \$240.00/CBM →  $240 * 0.60 = \$144.00$   
Other Cost: \$0.00  
Total: \$144.00  
Note: CBM below 0.05 is charged fixed \$10.

## 10. Business Logic Summary

### Pricing Rules

- 1. **Standard Charge:**  $\text{Total CBM} \times \text{Rate per CBM}$
- 2. **Minimum Charge:** If  $\text{CBM} < 0.05 \rightarrow \text{Fixed } \$10.00$
- 3. **Additional Costs:** Added to subtotal
- 4. **Final Total:**  $\max(\text{Subtotal}, \$10) + \text{Other Costs}$

### Customer Identification Hierarchy

- 1. **Primary:** Phone number (best identifier)
- 2. **Secondary:** Customer name
- 3. **Tertiary:** Customer ID from column A
- 4. **Fallback:** "UNKNOWN"

### Shipping Mark Handling

- Multiple marks per customer are combined
- Displayed as comma-separated list in summary
- Shown as line-separated in invoice breakdown section

## 11. Logo & Branding Details

### Logo Design






- **Company Name:** I&C SHIPPING AND LOGISTICS
- **Design Elements:**
  - Abstract swoosh symbol suggesting movement and speed
  - Three curved shapes in vibrant colors (orange-yellow gradient, teal/cyan, magenta/pink)
  - Classic serif font for "I&C" initials in teal
  - Professional sans-serif for "SHIPPING AND LOGISTICS" tagline
- **Color Scheme:**
  - Dark navy blue background
  - Teal/cyan (primary brand color)
  - Orange-yellow gradient (energy, movement)
  - Magenta accent

- **Brand Positioning:** Modern, dynamic, yet reliable and professional
- 

## Conclusion

---

The **I&C Billing Tool** is a well-engineered, production-ready application that successfully automates a critical business process. It demonstrates:

-  **Solid Python fundamentals** with modern best practices
-  **Practical problem-solving** for real-world data messiness
-  **User-centric design** with simple, effective GUI
-  **Professional output quality** suitable for customer-facing documents
-  **Business logic accuracy** with edge case handling

The suggested improvements focus on **scalability**, **maintainability**, and **enhanced user experience**, but the current implementation is functional and reliable for its intended purpose.

---

**Analysis Date:** January 12, 2026

**Application Version:** As provided in uploaded files

**Analyzed By:** DeepAgent (Abacus.AI)