

BALASUBRAMANIAN V

(ROLL NO: 23121106)

CHRIST(DEEMED TO BE UNIVERSITY) PUNE, LAVASA CAMPUS

CIA 3-MINI PROJECT

Develop a management information system (MIS) to optimize customs clearance processes for import and export activities.

Course Instructor: Prof Prashant Joshi

April 29, 2024



CHRIST
(DEEMED TO BE UNIVERSITY)
PUNE LAVASA CAMPUS
The Hub of Analytics

PROJECT GOALS

Develop and deploy a modular Management Information System (MIS) with integrated data structures, classes, and file systems to optimize customs clearance processes for import and export activities. The system aims to expedite customs documentation, track shipment progress, and provide real-time visibility into clearance processes, thereby reducing delays and maintaining regulatory compliance.

1. The project aims to create a user-friendly interface for managing customs clearance records, automating tasks to reduce errors and processing time.
2. By implementing validation checks and documentation features, the system ensures adherence to regulatory standards, fostering trust with stakeholders and facilitating smooth international trade.

By its nature international export-import logistics services provider experiences difficulties relating to customs clearance that causes considerable delays, fines and rise in demand for the business services. Existing compliance systems are neither scalable nor organized, resulting in bottlenecks in order and rate processing in most clearance processes. The management information system (MIS) is the key to the described company problems namely clearing and control of import and export activities. The MIS will be developed and deployed specifically to address these problems. The MIS will lead to a modular design that finds its ground in the use of functions, data structures, and classes so as to enable the MIS to become truly flexible and scalable. Likewise, file storage systems will be included to simplify the storage and management of clearance data, to narrow the access route and the search for any data when needed. A MIS solution allows for monitoring the operations scheduling in real-time, which means that the MIS will make the way for greater efficiency in planning activities, eliminating delays and reducing the degree of resource usage. On the other hand, the task approach will give focus to regulatory compliance by developing features of the system for the purposes of checking the customs regulations and policies. This is believed to bring forth the improvement of operational efficiency, reduction of charges pertained to delays and penalties, and the enhancement in overall customer gratification through the smooth clearance.

Contents

1	CODE OVERVIEW	3
1.1	Introduction and Dependencies	3
1.2	Core Functionality and Implementation Details	3
2	EXPLANATION OF CODE	4
2.1	Decomposition and Reusability	4
2.2	Exception Handling	4
2.3	Integration and Comments	5
3	OUTPUT	6
3.1	Test Case 1	6
3.2	Test Case 2	7
3.3	Test Case 3	8
3.4	Outputs	8
4	CONCLUSION	10

Chapter 1

CODE OVERVIEW

1.1 Introduction and Dependencies

The code is created to handle a customs clearance management system. It starts by importing essential dependencies such as the tkinter library for building a graphical user interface (GUI), messagebox for displaying messages, and FPDF for producing PDF reports. Global variables are defined to store pointers to entry fields and text boxes, making it easier to manipulate data and interact with the GUI throughout the code.

1.2 Core Functionality and Implementation Details

The core functionality of the program is encapsulated within several function definitions. These functions include `read_data(file name)` and `write_data(data, file name)` for reading from and writing to a file respectively. The `read_data()` function parses data from a specified file, validating the format and storing it in a dictionary. Conversely, `write_data()` formats clearance records and writes them to a file.

The `add_clearance_record()` function is responsible for validating user input when adding a new clearance record. It checks for required fields and ensures that clearance IDs are unique. Additionally, input validation is performed to ensure that vessel names consist only of letters and spaces, while numeric fields contain only digits and dots. Once validated, the new record is added to the data dictionary and written to the file.

Displaying existing clearance records is facilitated by the `display_clearance_records()` function. This function clears the text box and formats each clearance record before inserting it into the text box for display. Similarly, the `print_clearance_data_as_pdf()` function generates a PDF report containing all clearance records, utilizing the FPDF library to create the document.

The `update_clearance_status()` function allows users to update the status of a clearance record. It validates the input clearance ID and new status before modifying the data dictionary accordingly and writing it back to the file. Additionally, the `clear_entries()` function clears all entry fields in the GUI, resetting them to empty strings.

The `main()` function is the program's entry point, initializing the GUI window with labels, entry fields, and buttons for user input. Functions are linked to buttons for adding, displaying, and editing records, with clearance records shown in a text box. The code follows best practices in exception handling and documentation, with a modular design.

Chapter 2

EXPLANATION OF CODE

2.1 Decomposition and Reusability

The code indeed follows the principle of decomposition and reusability by defining separate functions for different tasks such as reading data from a file, writing data to a file, adding a clearance record, displaying records, editing records, printing records to PDF, updating clearance status, and clearing entry fields. This approach reduces redundancy and makes the code modular and easier to maintain.

2.2 Exception Handling

Exception handling is implemented in several places. For instance, when reading data from a file, the code handles the *FileNotFoundException* exception gracefully by showing an error message to the user. Additionally, when adding a new clearance record, the code checks if all fields are filled and displays an error message if any field is empty. It also checks if the clearance ID already exists and notifies the user accordingly.

I also have added validation to ensure that specific fields, such as exporter name, consignee name, product name, port of loading, port of discharge, vessel name, and status, contain only letters and spaces. This validation is important because these fields typically represent textual information such as names or locations, where special characters are not expected.

By restricting input to only letters and spaces, it ensures that the data entered into these fields remains consistent and follows expected conventions. This can prevent potential issues such as data corruption, unintended behavior, or formatting errors in downstream processes that rely on this data.

Additionally, for numeric fields like quantity, rate, and net weight, I added validation to ensure they contain only digits and dots. This helps maintain data integrity and ensures that numeric values are entered correctly, reducing the risk of errors in calculations or data processing. These are just a couple of examples where exceptions are handled efficiently, ensuring a smoother user experience.

I have added few test cases in Chapter 3

2.3 Integration and Comments

All major functions are integrated into the code and called appropriately based on user actions. For example, when the user clicks the "Add New Clearance Record" button, the `add clearance record()` function is called to handle the addition process. Similarly, when the user clicks the "Print Clearance Data as PDF" button, the `print clearance data as pdf()` function is invoked. Comments are present at the beginning of each major function, providing a brief explanation of their purpose and functionality. This helps in understanding the code's logic and behavior.

Chapter 3

OUTPUT

3.1 Test Case 1

Here, if all fields are not entered, the message 'All fields are required' is displayed to the user.

The screenshot displays the 'Customs Clearance Management System' window. It features a list of input fields on the left: Clearance ID, Exporter, Consignee, Invoice No, Product, Quantity, Rate, Net Weight, Port of Loading, Port of Discharge, Container No, Vessel, and Status. The 'Clearance ID' field contains the value '1'. Below the fields are four buttons: 'Add New Clearance Record', 'Display All Clearance Records', 'Edit Clearance Record', and 'Print Clearance Data as PDF'. An error dialog box is open in the foreground, titled 'Error', with a red 'X' icon and the message 'All fields are required.' and an 'OK' button.

3.2 Test Case 2

If the user inputs the Exporter name as any character or number, it won't be accepted, and the message 'Exporter name should contain only letters and spaces' will be displayed.

The screenshot displays the 'Customs Clearance Management System' window. It features a form with the following fields and their current values:

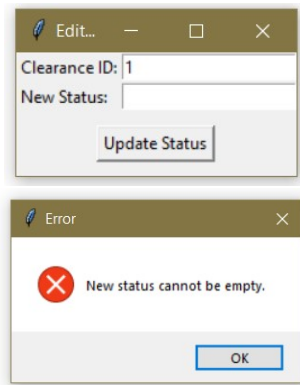
Field	Value
Clearance ID:	aa
Exporter:	12
Consignee:	10
Invoice No:	aa
Product:	1919
Quantity:	@#
Rate:	KB
Net Weight:	*8&8
Port of Loading:	&*
Port of Discharge:	*8*8&
Container No:	%\$%\$
Vessel:	^%^
Status:	^%^

Below the form are four buttons: 'Add New Clearance Record', 'Display All Clearance Records', 'Edit Clearance Record', and 'Print Clearance Data as PDF'.

An error dialog box is open in the foreground, titled 'Error'. It contains a red 'X' icon and the message: 'Exporter name should contain only letters and spaces.' The 'OK' button is at the bottom right of the dialog.

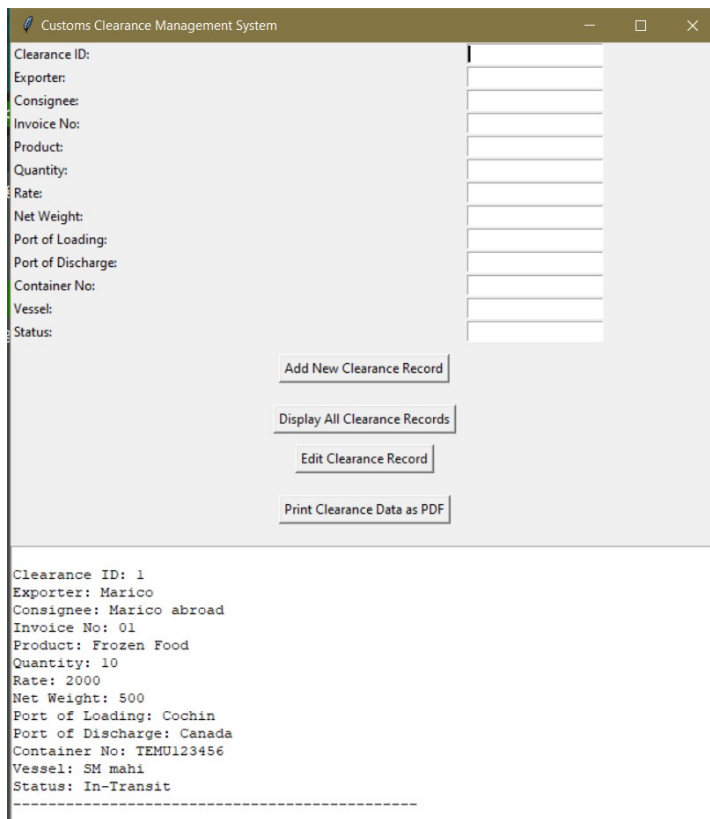
3.3 Test Case 3

If the status field is empty, a message will be displayed indicating that the new status cannot be empty.



3.4 Outputs

Basically, this project deals with reading and updating data. In this case, I have implemented the functionality to update the status of shipments. For example, if a container is being exported from India to Jebel Ali, the usual duration for the vessel to reach Jebel Ali is 14 days. Therefore, the user can update the status of that shipment from 'In-Transit' to 'Cleared'.



Customs Clearance Management System

Clearance ID:

Exporter:

Consignee:

Invoice No:

Product:

Quantity:

Rate:

Net Weight:

Port of Loading:

Port of Discharge:

Container No:

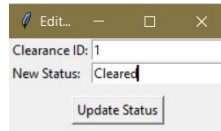
Vessel:

Status:

Clearance ID: 1
Exporter: Marico
Consignee: Marico abroad
Invoice No: 01
Product: Frozen Food
Quantity: 10
Rate: 2000
Net Weight: 500
Port of Loading: Cochin
Port of Discharge: Canada
Container No: TEMU123456
Vessel: SM mahi
Status: In-Transit

In the Shipment of Clearance ID:1 the status shown in-transit

When User updates the status to Cleared, It gets updated and clicking the display record button we can see the status being updated to Cleared.

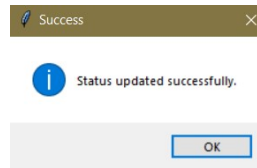


Edit... — □ ×

Clearance ID: 1

New Status: Cleared

Update Status



```
Clearance ID: 1
Exporter: Marico
Consignee: Marico abroad
Invoice No: 01
Product: Frozen Food
Quantity: 10
Rate: 2000
Net Weight: 500
Port of Loading: Cochin
Port of Discharge: Canada
Container No: TEMU123456
Vessel: SM mahi
Status: Cleared
-----
```

Chapter 4

CONCLUSION

In conclusion, the customs clearance management system presented here exemplifies effective software design principles, including modularization, exception handling, and code documentation. By decomposing the program into coherent and reusable units, unnecessary repetition is eliminated, making the codebase more maintainable and adaptable to future changes.

Exception handling is implemented throughout the system, ensuring that errors are gracefully managed, enhancing the user experience by providing informative error messages. Additionally, the integration of all functions within the codebase, coupled with appropriate comments for major functions, enhances code readability and comprehension, facilitating easier maintenance and collaboration among developers.

Overall, this project demonstrates a robust approach to customs clearance management, emphasizing efficiency, reliability, and user-friendliness. As businesses continue to navigate complex logistics challenges, such software solutions play a crucial role in streamlining operations and ensuring compliance with regulatory requirements.