Volume 10, Issue 5, May – 2025 International Journal of Innovative Science and Research Technology
ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25may1660

Integrated Inventory and Finance Management Platform with Python and Tkinter

Gargi Roy[1]; Debodyuti Upadhaya[2]; Basudeb Dey[3]; Sayan Kar[4]; Rupam Mondal[5]; Susrita Khatun[6]; Gargi Ghosh[7]

[1,2,3,4,5,6,7]Department of Electrical Engineering, JIS College of Engineering, Kalyani, Nadia, 741235

Publication Date: 2025/06/04

Abstract: This is a system, which can do material and finance management using Python and Tkinter. The system is designed to facilitate different stocking operations, including user authentication transaction and statement generation, also basic operations to stock management functionalities. Functionality: This system supposes a range of functionalities. User authentication: secure login and sign up. Mechanisms for only authorized access. Transaction Management: Users can debit and credit money, transfer funds with users and can track transaction history. Stock Management: Users can add stock items and generate bills. Statement Generation: The application generates detailed bank statements in PDF format. Data Persistence: All used and transaction Data are stored persistently using JSON, by ensuring data integrity.

Keywords: *Stock Management, Tkinter GUI Application, Inventory Management, PDF Invoice Creator, Bill Generate Window, Python Base Billing System.*

How To Site: Gargi Roy; Debodyuti Upadhaya; Basudeb Dey; Sayan Kar; Rupam Mondal; Susrita Khatun; Gargi Ghosh (2025) Integrated Inventory and Finance Management Platform with Python and Tkinter. *International Journal of Innovative Science and Research Technology,* 10(5), 3269-3274. https://doi.org/10.38124/ijisrt/25may1660

I. INTRODUCTION

In this digital generation an effective finance and material management is essential for all small businesses. Traditional methods of handling these tasks are mostly paper based records or any other manual processes. Those are time consuming and prone to errors. The advent of digital solutions offers new fast track processes with enhanced accuracy, security and convenience.

This software aims to develop a comprehensive material and finance management system using Python and Tkinter and also to provide users with a robust platform to manage their financial activity and inventory in real time. This many functions like User authentication, master login statement generation, etc.

➢ *Objective*

The primary object is to create a user-friendly application for desktop that allows users to perform different stuking tasks such as balance tracking, crediting and debiting amounts, transfer amount and track transaction history. Additionally, this system includes features like generating bills and changing live data. It aims to provide a full package for small business and personal finance management.

➢ *Aim*

The main aim of this research paper is to create a stock management and any kind of effective financial transaction system related to that stock, and help in managing banking processes in a safe, organized and fast manner, it is dependent on the digital method for customer data bank account and stock related transactions and eliminates manual work, another objective of this management is to reduce the dependence on humans by using active tools in this banking and stock management system, prevent any errors and data loss, and use this technology to integrate banking and stock management systems without keeping the stock management of banking and businesses separate.

II. LITERATURE SURVEY

- Nowadays, inventory management or stock management is at least important in any business or industry, so Muhammad Ehsan Rana[1] has Discuss about inventory tracking using the Internet of Things to activate industrial and household processes, which is a very important system for collecting and managing information about manufacturers, wholesalers and warehouses, that is why if there is a stock management system for every business organization or industry, then it can keep the accounts of wholesalers, retailers and warehouses in an accurate and well-organized manner.

- To eliminate unnecessary manpower and increase the efficiency of automated stock measurement and order placement S. Jayant. [1], discussed a topic - in which an ultrasonic transducer is connected to the internet and

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25may1660

Raspberry Pi. After the order is confirmed, an email is sent to the supplier or company employee to maintain the authenticity of the product.

- MARTIN HELLWIG. - One of the objectives of this research paper is to discuss and analyze in detail the impact of globalization on the banking system and its impact on bank management. To achieve this goal, we started from the assumption that macroeconomic developments have a direct impact on bank management.

III. SYSTEM OVERVIEW

The material and finance management system is designed for desktop application using Python and Tkinter library (For graphical user interface) The system structure is designed to provide integration of material management and finance/ banking functionalities features, ensuring an efficient user-friendly interface. The Main components of the system include.

➢ *User Interface (Ui):*
Built using Tkinter, the UI component handles all user interactions with programming. It creates a graphical interface.

➢ *Business Logic:*
This component handles all the core functionalities of the system, including transaction, User authentication and data validation.

➢ *Data Management:*
This component is responsible for storing and retrieving data transactions, and storing information. JSON is used as a storage format.

➢ *Reporting:*
This component generates bank statements and other needed reports. It uses the FPDF library to create PDF documents.

IV. SYSTEM ARCHITECTURE

The system architecture follows a modular design, with various components handling specific functionalities. This modular approach enhances maintainability, scalability and the potential for future enhancement.

➢ *User Interface:*
The user interface is designed to be intuitive and user friendly. Key elements:

- Login and sign-up forms: Allow to login and sign up.
- Dashboard: User financial status, recent transaction and stock inventory.
- Transaction Page: credit, debit, transferring funds.
- Stock Management Page: Adding, updating and viewing stocks.
- Reports Page: Generate and view bank statements.

➢ *Data Management:*
Data Management is very crucial for integrity and consistency of data and transaction records. JSON files are used to store data persistently. Key aspects:

- Users Data: Information such as user name, password, account balance.
- Transaction Records: Logs all financial transactions (credits, debits, transfers)
- Material Information: keeps track of material including details like item name, quantity, price and descriptions.

➢ *Business Logic*:
The business Logic model handles the core operations of the system and ensures all transactions are maintaining rules and constraints. Key Function:

- Transaction processing: Validates and processes credit, debit and transfer transactions, updating user balances accordingly.
- User Authentication: Manage login and sign up processes.
- Validation: make sure that all inputs are correctly formatted and valid before processing transactions or updating materials.

➢ *Reporting And Statement Generation:*
This component works based on the FPDF library to generate all kinds of statements and other reports. This report provides detailed records of transactions, including dates, types, amounts, balance and remarks. PDF format is used because it is easily printable.

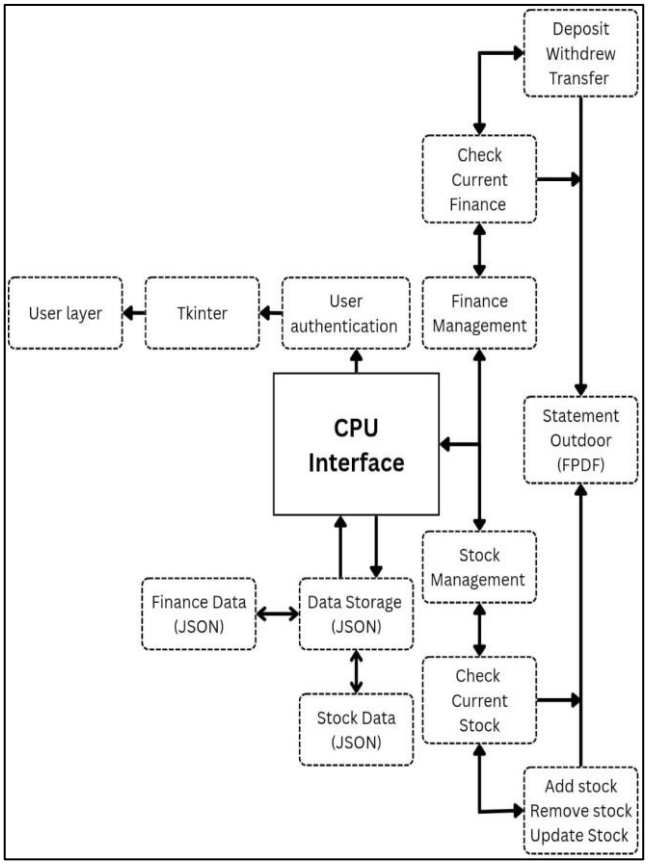

Fig 1 System Architecture

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25may1660

V. COMPONENTS

➢ *User Authentication Module*

This module provides secure mechanisms for user registration and log in, with password protection. It employs industry-standard practices of live password hashing and validation to prevent unauthorized access.

➢ *Function*:

User Registration: Creating a new user account with a unique username and secure password. User Login: This process verifies users to grant access to the system.

➢ *Financial Transaction Module*

This module is designed for various financial transactions, including deposits, withdrawals and transaction history monitoring. This financial transaction module enables users to perform and manage financial activity.

➢ *Functions*:

- Deposit Money: Deposit money into accounts.
- Withdraw Money: Withdraw money from account.
- Transfer Fund: Money/Fund can be transferred to other user accounts.
- View Transaction History: Transaction history can be tracked.

➢ *Material Management Module*

This crucial component handles inventory related operations. This method makes sure efficient management of inventory including adding new material, generating bills and tracking current stocks.

➢ *Functions:*

- Adding Stock: New inventory update.
- Track Current Stock: Keep records of current stocks.
- Generating Bills: Generates bills for sales and current stock.

➢ *Statement Generations*

Statement generation module designs to create detailed financial statements for users. It gives access to users to track comprehensive records of transactions. It focuses on generating PDF statements using the FPDF library. PDF is the easiest way to print and dead data. Also, this module helps to export inventory list.

➢ *Function*:

- PDF Statement Creation: Generate PDF statements.
- Formatting and Details: Store and generate statements in a well formatted way with all essential details.

➢ *Data Persistence*

It is a very important aspect of the application. It controls and manages all used data, transaction, and other crucial information. It stores all this data and information in a system using JSON data handling format.

➢ *Functions*:

- JSON Data Handling: Uses JSON files to manage and store data.
- User Information Design: Well organised UI infrastructure can interact with the data easily.

➢ *Flow Chart*

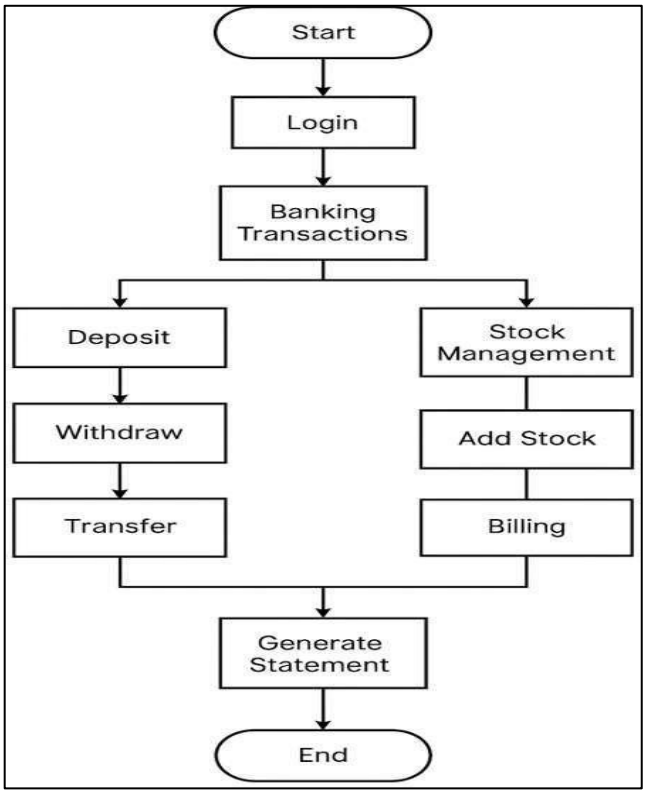

Fig 2 Flow chart

VI. WORKING PROCEDURE

➢ *Application Startup*

- The Python script is executed.
- All essential data files (bank_data.json, inventory_data.json, invoice_number.json) are loaded
- Tkinter main window is initialized.
- Global variables (products, users, transactions, and balances) are stored for runtime access.

➢ *User Authentication*

- *Sign Up*

✓ User clicks "Sign Up".
✓ Inputs:
✓ Username
✓ Password
✓ System checks if username exists.
✓ If not:
✓ Adds username and password to bank_data.json
✓ Initializes balance to 0.
✓ Empty transaction log are set up.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25may1660

- *Login*

✓ User enters credentials.
✓ If matched, current_user is set.
✓ User is taken to the Dashboard/Main Menu.

➢ *Dashboard/Main Menu*
User is shown options:

- Banking Module
- Stock Management Module
- Logout/Exit

➢ *Banking Transactions Module*

- *Deposit*

✓ User selects "Deposit"
✓ Enters amount and optional remark
✓ Validation: amount > 0
✓ Balance is updated
✓ Transaction is logged with:
✓ Date/time
✓ Type: Credit
✓ Amount
✓ New balance
✓ Remark
✓ Data is saved to bank_data.json

- *Withdraw*

✓ User selects "Withdraw"
✓ Enters amount
✓ Validation:
✓ Amount > 0
✓ Balance >= amount
✓ If valid:
✓ Balance is deducted
✓ Transaction is logged with type: Debit

- *Transfer*

✓ User selects "Transfer"
✓ Enters:
✓ Recipient username
✓ Amount
✓ Validations:
✓ Recipient exists
✓ Sufficient balance
✓ If valid:
✓ Sender balance reduced
✓ Recipient balance increased
✓ Both transactions logged separately
✓ Saved in bank_data.json and transactions

➢ *View Transactions / Generate Bank Statement*

- User selects "Generate Statement"
- System fetches transactions
- Uses FPDF to create a PDF (Lists: date, type, amount, remarks, balance)
- File saved locally

➢ *Stock Management Module*

- *Add Stock*

✓ Opens "Add Stock" window
✓ Inputs:
✓ Product Name
✓ Quantity (Kg/Ltr)
✓ Price (per unit)
✓ Validates:
✓ Name is unique or not
✓ Quantity/Price are numbers
✓ If valid:
✓ Product is added to products list
✓ Data saved in inventory_data.json

- *Check Stock*

✓ Opens full-screen stock window
✓ Shows:
✓ SL No
✓ Product Name
✓ Quantity
✓ Price
✓ Right-click options:
✓ Edit quantity/price
✓ Delete product
✓ Search feature filters products
✓ "Export to PDF" creates a stock summary PDF

- *Bill Generation*

✓ Opens "Generate Bill" window
✓ User:
✓ Adds items from stock to bill
✓ Enters sold quantity
✓ System checks stock availability
✓ Price calculated = quantity × price per unit
✓ Items are added to bill list
✓ PDF invoice is generated:
✓ Invoice Number
✓ Product Details
✓ Total Amount
✓ File saved to: C:\Users\user\Downloads\INVOICE\
✓ Inventory is auto-updated
✓ inventory_data.json is saved

- *Exit or Logout*

✓ User can exit app from menu
✓ All modified data saved to JSON files:
✓ bank_data.json
✓ inventory_data.json
✓ invoice_number.json
✓ Tkinter windows close and session ends

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25may1660

➢ *Advantages*

- Its main advantage is that it can be used to manage stock and calculate banking transactions in a very simple manner with an accessible interface.
- Through this, all types of transactions and stock management accounting are recorded and all transactions that take place throughout the day or throughout the month are properly maintained, the data is stored and used when needed later.
- It increases the reliability of this system by performing all tasks perfectly and accurately, without relying on workers.
- Using Python and Tkinter, the application is capable of running on multiple operating systems with very minimal changes.

➢ *Disadvantages*

- Because it is a computer application, it cannot be used by a large number of customers at the same time and cannot handle large-scale banking transactions.
- It is a software best application that requires regular updates and maintenance for security and secure accounting.
- Setting up a Python environment on a customer's or user's system is very necessary, which can be a bit of a hassle for someone who doesn't know Python or a non-technical customer.

VII. CHALLENGES AND SOLUTIONS

➢ *Data Integrity and Persistence*

- *Challenge:*

Handling concurrent data access and preventing data loss during read/write operation.

➢ *Solution:*

- JSON Data Handle is Implementing for data storage,it ensures easy read/write operations.
- File Lock Mechanism is used to prevent concurrent data access issues.
- Regular Backups of data files are implemented to prevent data loss.

➢ *User Authentication Security*

- *Challenge:*

Implementing secure storage and verification methods for passwords.

➢ *Solution:*

- Hashed passwords are used instead of plain text to enhance security.
- Secure Login Mechanism is implemented for incorrect credentials.

➢ *Input Validation and Error Handling*

- *Challenge:*

Handling various input errors with proper feedback to users.

➢ *Solution:*

- Implemented validation for all user inputs to ensure valid data.
- Using try-except blocks to provide meaningful feedback to users.

➢ *User Interface Responsiveness*

- *Challenge:*

Managing dynamic updates for UI user interactions.

➢ *Solution:*

- Updating the UI dynamically based on user interactions.
- Consistent Design is maintained for better user experience.

➢ *PDF Generation for Statements*

- *Challenge:*

Formatting data correctly and handling large datasets within PDF constraints.

➢ *Solution:*

- FPDF Library is used for creating PDF statements.
- Handling large datasets by managing column widths and line breaks.

VIII. FUTURE ENHANCEMENT

- Right now, this system is being planned through Python and Tender, but in the future, if IoT and AI are already added, then the system can be improved a lot more.

- If we can develop this application for our Android handsets in the future, it will be even more convenient for users.

- Right now, it only keeps track of Indian Rupees, but later if we add currencies from other countries, it will become an easily accessible and user-friendly application for international traders.

- If biometric upgrades can be made later and connected to cloud computing, then its scalability & accessibility will be better.

IX. CONCLUSION

This project is successfully developed a **Integrated Meterial and Finance Management System using Python and Tkinter**. It is designed to securely manage basic meterial stock listing and accounting function. Key function it has are,

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25may1660

fund deposit, withdrawal, fund transfer, stock listing, statement generate etc.

All data and transections are stored as JSON file format, ensuring data persistence without any complex database. The FPDF library generates all the statements and perform continuous data sharing with JSON data format. This system is mainly design to work flawless with Windows desktop, aslo suitable for low performance desktop with high-speed. The main advantage of the system is its user-friendlyness.

REFERENCES

[1]. Rana, Muhammad Ehsan, Kamalanathan Shanmugam, and Chan Yu Hang. "Recommendations for Implementing an IoT based Inventory Tracking and Monitoring System." *International Journal of Psychosocial Rehabilitation* 24, no. 5 (2020): 3908-3917.

[2]. Hellwig, Martin. "Systemic aspects of risk management in banking and finance." *Revue Suisse D Economie Politique Et De Statistique* 131 (1995): 723-738.

[3]. Jayanth, S., M. B. Poorvi, and M. P. Sunil. "Inventory management system using IOT." In *Proceedings of the First International Conference on Computational Intelligence and Informatics: ICCII 2016*, pp. 201-210. Springer Singapore, 2017.