RATION SHOP STOCK MANAGEMENT

A PROJECT REPORT

Submitted by

KEERTHANA S (220701124)

in partial fulfilment for the course

OAI1903 - INTRODUCTION TO ROBOTIC PROCESS AUTOMATION

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE RAJALAKSHMI NAGAR THANDALAM CHENNAI – 602 105

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE

CHENNAI - 602105

RAJALAKSHMI ENGINEERING COLLEGE

CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report **"Ration Shop Stock Management"** is the bonafide work of **"KEERTHANA S (220701124)"** who carried out the project work for the subject OAI1903 - Introduction to Robotic Process Automation under my supervision.

MRS.J. Jinu Sophia (Ph.D.)

SUPERVISOR,

Assistant Professor (SG),

Department of

Computer Science and Engineering, Rajalakshmi Engineering College, Rajalakshmi Nagar,

Thandalam,

Chennai – 602105.

Submitted to Project and Viva Voce Examination for the subject OAI1903 - Introduction to Robotic Process Automation held on_____.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Thiru. S.Meganathan, B.E., F.I.E.,** our Vice Chairman **Mr. M.Abhay Shankar, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, M.A., M.Phil., Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P.Kumar, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides, **Mrs. J. Jinu Sophia, M.E., (Ph.D.),** Assistant Professor (SG), Department of Computer Science and Engineering for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinators, **Dr. N.Durai Murugan, M.E., Ph.D.,** Associate Professor, and**Mr. B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering for their useful tips during our review to build our project.

Keerthana S (220701124)

ABSTRACT

The **Ration Shop Stock Management System** is an automated solution designed to streamline the management of stock, user registrations, and invoice generation for ration shops. Built using UiPath Studio, the system aims to reduce manual effort, improve operational efficiency, and ensure accurate stock management. The system integrates with Excel sheets for data processing and performs real-time validations to ensure that only available stock is distributed, providing a seamless experience for both shop owners and customers.

The system begins by extracting user information, including phone numbers and selected items, from an Excel sheet. When a user's phone number is not found, the system prompts for additional information such as the user's name and ration shop branch. The program ensures that the stock is available in the required quantities and handles stock-related errors efficiently, allowing customers to modify their orders if necessary.

Upon successful stock validation, the system generates an invoice in PDF format containing details about the user's transaction, such as the items purchased and the total cost. The generated invoice is then automatically emailed to the recipient using an SMTP-based email automation process. This feature eliminates the need for manual follow-up, improving both efficiency and customer satisfaction.

The design incorporates built-in error handling, logging, and performance monitoring to address issues such as missing data or failed email deliveries. The system's scalability is ensured, allowing it to handle a growing number of users and transactions without compromising performance. This automation minimizes human error, reduces the time spent on administrative tasks, and enhances the overall accuracy of ration distribution operations.

This Ration Shop Stock Management System is a versatile and scalable tool, ideal for government departments, retail outlets, or any organization managing ration or inventory-related tasks. It serves as a model for future automation projects, offering a reliable solution for modernizing outdated manual processes while enhancing data security, accuracy, and accessibility.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**5**
	LIST OF TABLES	**6**
	LIST OF FIGURES	**7**
	LIST OF ABBREVIATIONS	**9**
1.	**INTRODUCTION**	**10**
1.1	GENERAL	
1.2	OBJECTIVE	
1.3	EXISTING SYSTEM	
1.4	PROPOSED SYSTEM	
2.	**LITERATURE REVIEW**	**12**
2.1	GENERAL	
3.	**SYSTEM DESIGN**	**14**
3.1	GENERAL	
3.1.1	SYSTEM FLOW DIAGRAM	
3.1.2	ARCHITECTURE DIAGRAM	
3.1.3	SEQUENCE DIAGRAM	
4.	**PROJECT DESCRIPTION**	**18**
4.1	METHODOLOGIES	
4.1.1	MODULES	
5.	**CONCLUSIONS**	**20**
5.1	GENERAL	
	REFERENCES	**21**
	APPENDICES	**22**

LIST OF TABLES

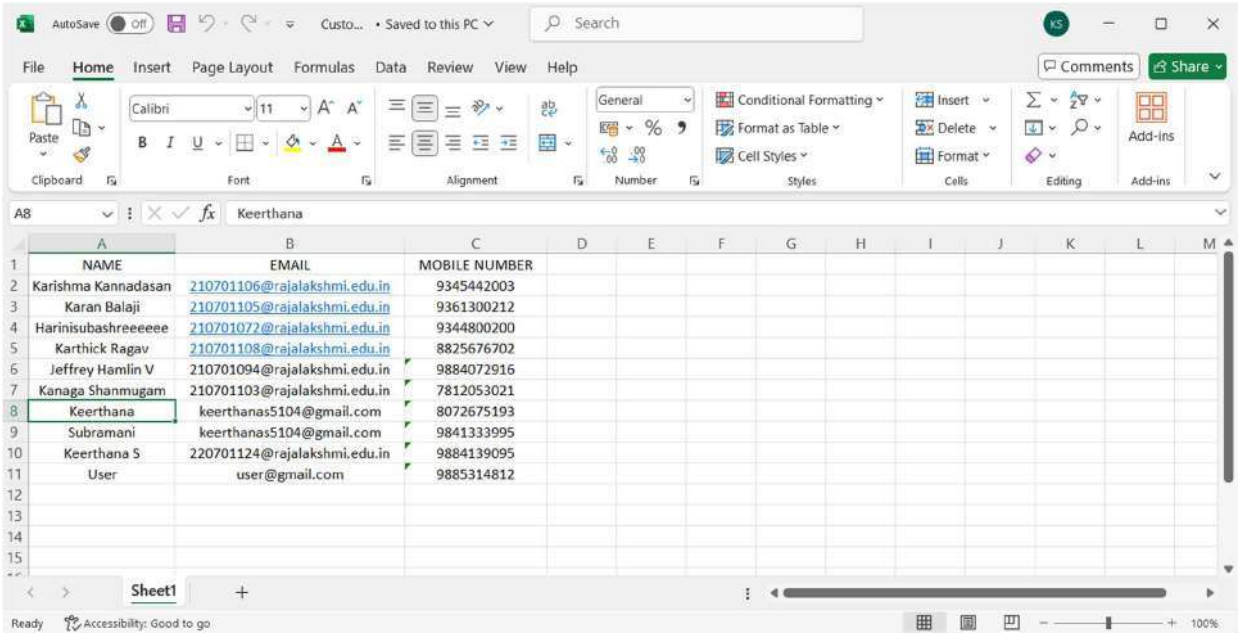
Field Name	Data Type	Description
Customers.Xlsx	xlsx	Contains the list of Customers with fields like Name, Email, and Phone number.
Invoice.Xlsx	xlsx	Contains the design template for Invoice, including placeholders for data.
Stock.Xlsx	xlsx	Contains the Stock details with fields like Items, Stocks left and Price.

Description:

The List of Tables provides a detailed overview of the key data files and their purposes used in the ration shop stock management project. Each table is crucial for ensuring the seamless functionality of the system. Below is the description for the included tables:

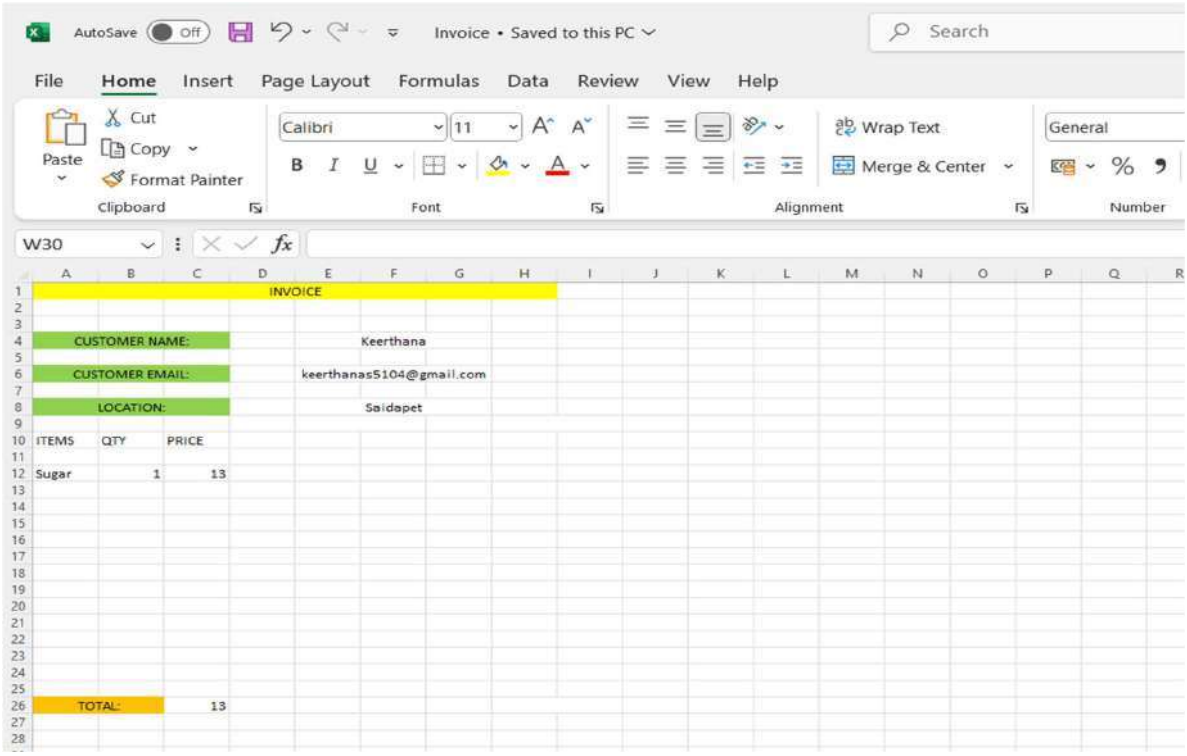
1. **Customers.Xlsx:**
   This Excel file contains the necessary information about the customers, including their names, email addresses, and the phone numbers. It acts as the primary input data source for the system.
2. **Invoice.Xlsx:**
   This Word document serves as the design template for the certificates. It includes placeholders where customers data (like Name, EmailID, Phone Number, Items, Quantity & Price) is dynamically inserted during the invoice generation process.
3. **Stock.Xlsx:**
   This Excel file contains the necessary information about the stocks, including item name, stocks left, and price. It is used to check the left out stocks and price of the items.

LIST OF FIGURES



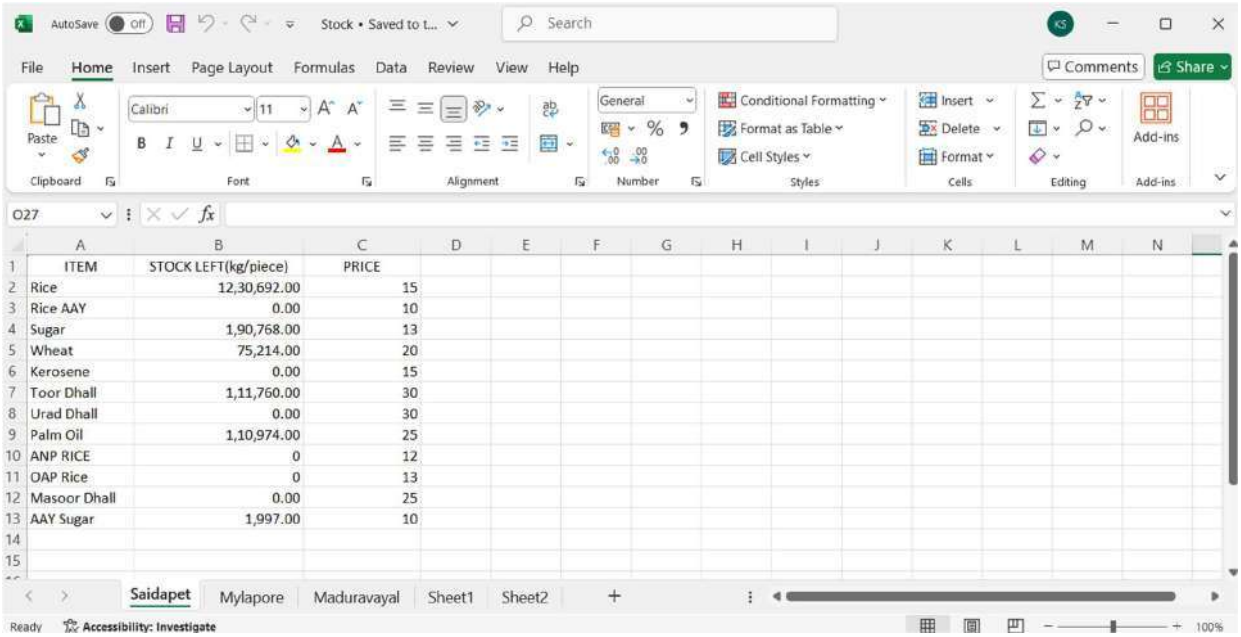
	NAME	EMAIL	MOBILE NUMBER
2	Karishma Kannadasan	210701106@rajalakshmi.edu.in	9345442003
3	Karan Balaji	210701105@rajalakshmi.edu.in	9361300212
4	Harinisubashreeeeee	210701072@rajalakshmi.edu.in	9344800200
5	Karthick Ragav	210701108@rajalakshmi.edu.in	8825676702
6	Jeffrey Hamlin V	210701094@rajalakshmi.edu.in	9884072916
7	Kanaga Shanmugam	210701103@rajalakshmi.edu.in	7812053021
8	Keerthana	keerthanas5104@gmail.com	8072675193
9	Subramani	keerthanas5104@gmail.com	9841333995
10	Keerthana S	220701124@rajalakshmi.edu.in	9884139095
11	User	user@gmail.com	9885314812

Fig 1.1 Customer Details from the excel



1	INVOICE		
4	CUSTOMER NAME:	Keerthana	
6	CUSTOMER EMAIL:	keerthanas5104@gmail.com	
8	LOCATION:	Saidapet	
10	ITEMS	QTY	PRICE
12	Sugar	1	13
26	TOTAL:		13

Fig 1.2 Invoice Template



	A	B	C
1	ITEM	STOCK LEFT(kg/piece)	PRICE
2	Rice	12,30,692.00	15
3	Rice AAY	0.00	10
4	Sugar	1,90,768.00	13
5	Wheat	75,214.00	20
6	Kerosene	0.00	15
7	Toor Dhall	1,11,760.00	30
8	Urad Dhall	0.00	30
9	Palm Oil	1,10,974.00	25
10	ANP RICE	0	12
11	OAP Rice	0	13
12	Masoor Dhall	0.00	25
13	AAY Sugar	1,997.00	10

Fig 1.3 Stock Details from the excel

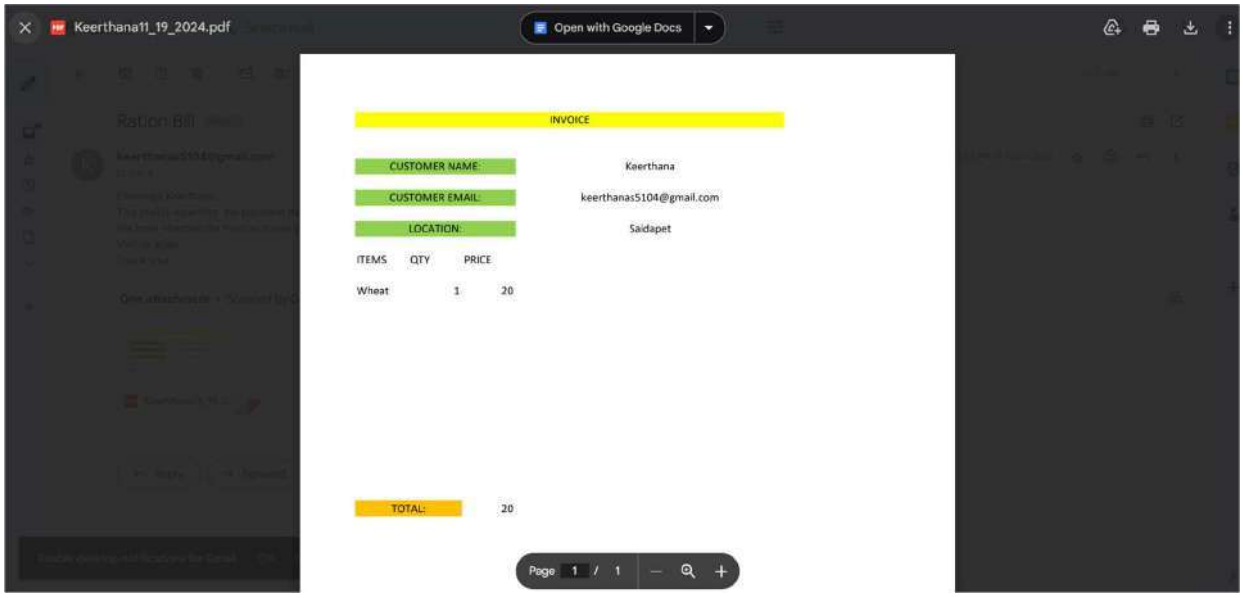


Fig 1.4 Generated Details in PDF Format

Fig 1.5 Generated PDF in the Folder

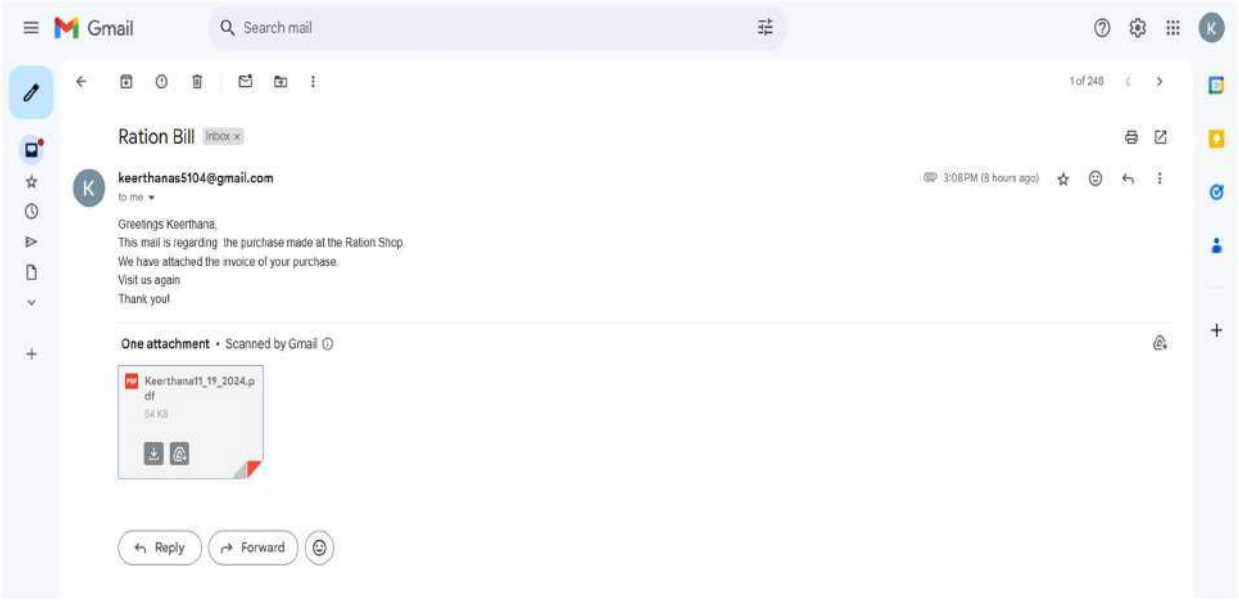
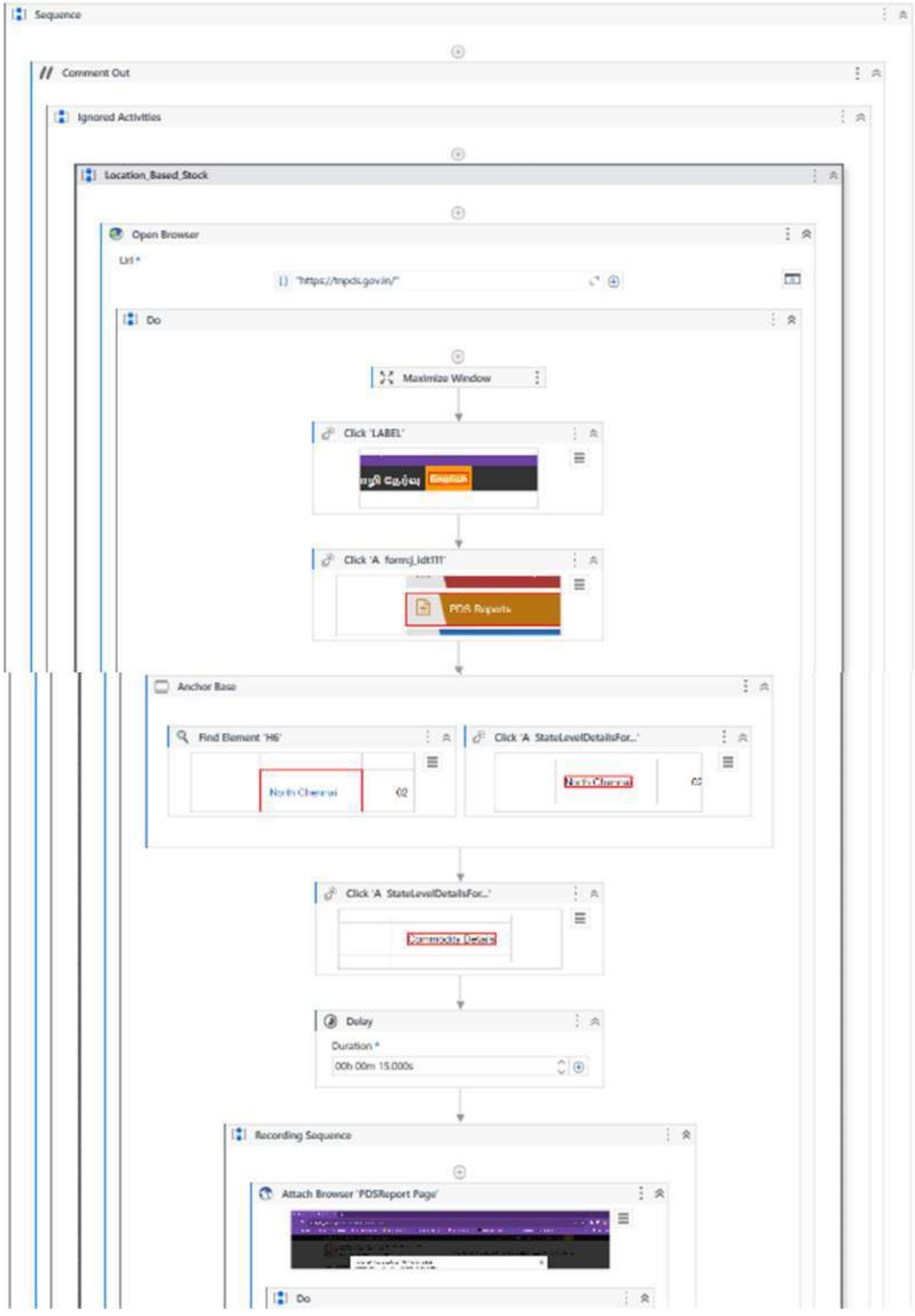
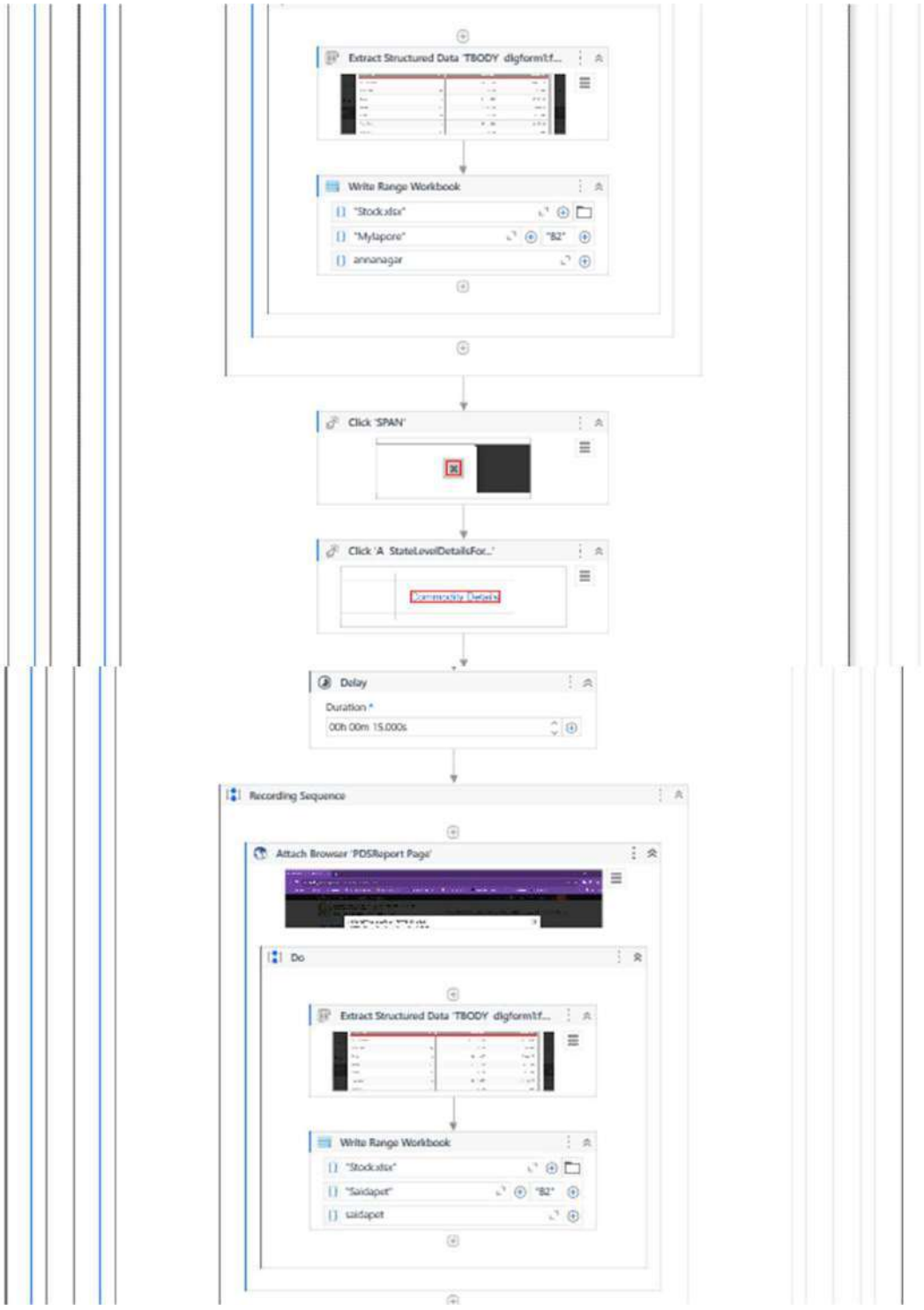
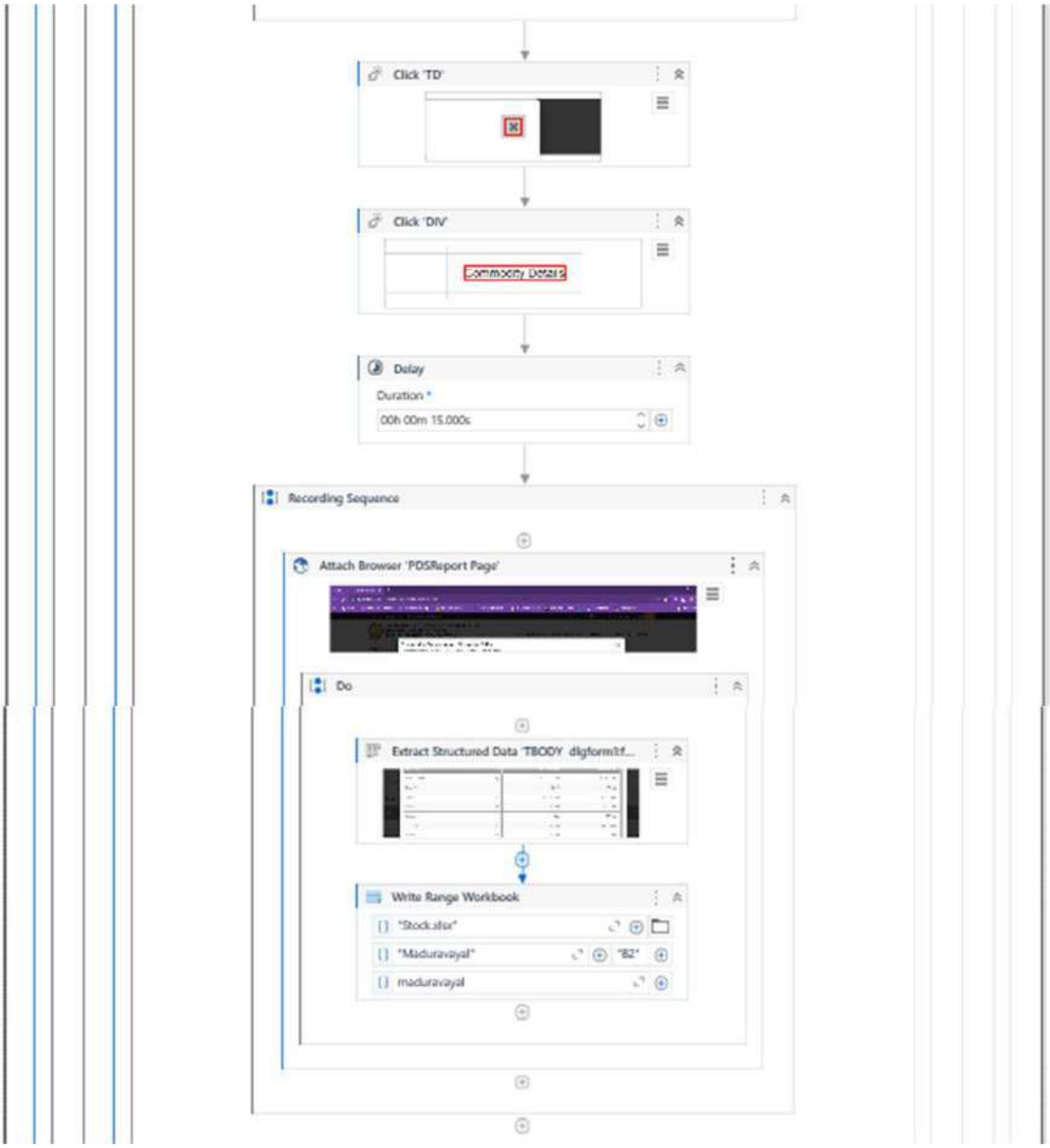
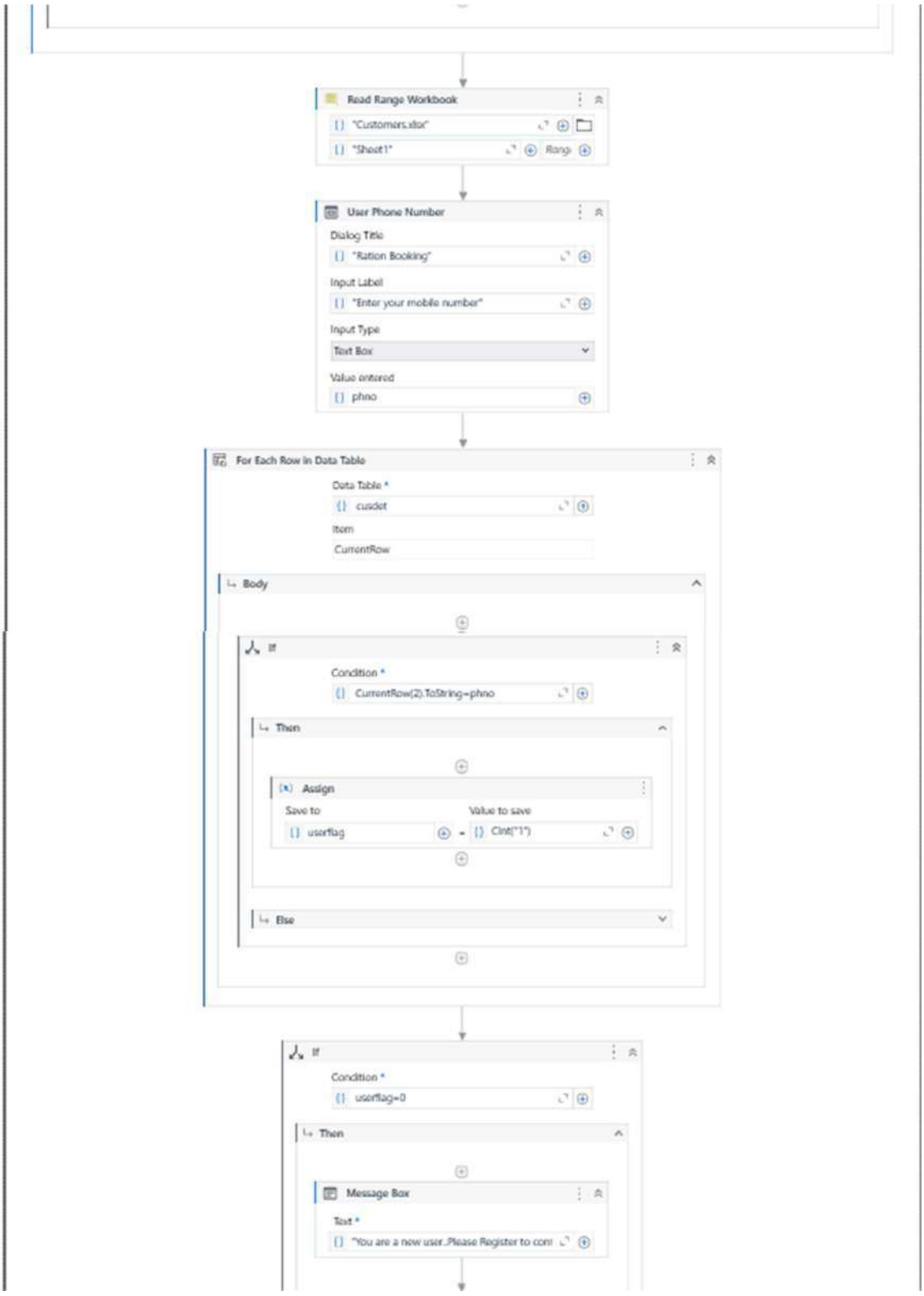


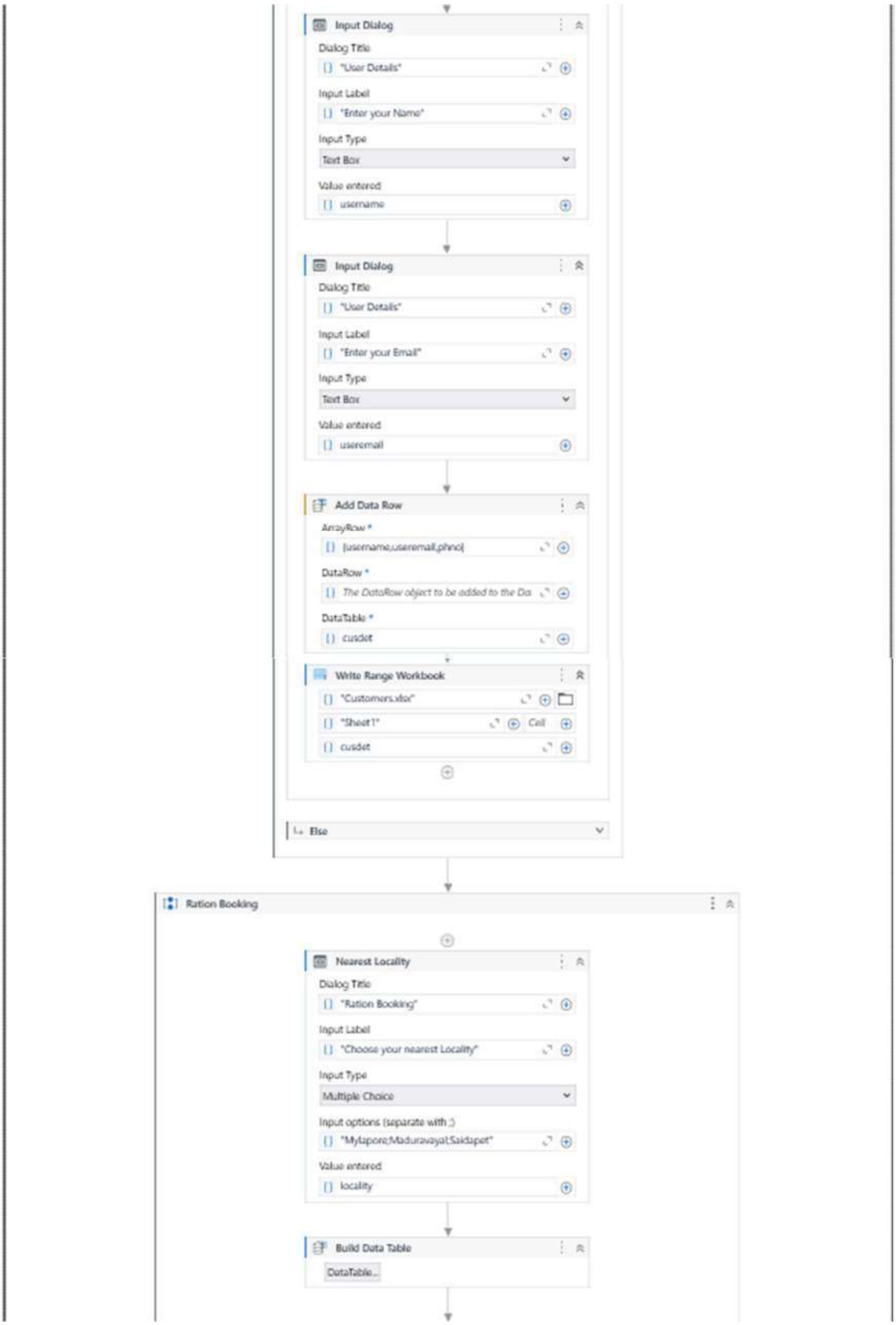
Fig 1.6 Sample Email Sent to Customer

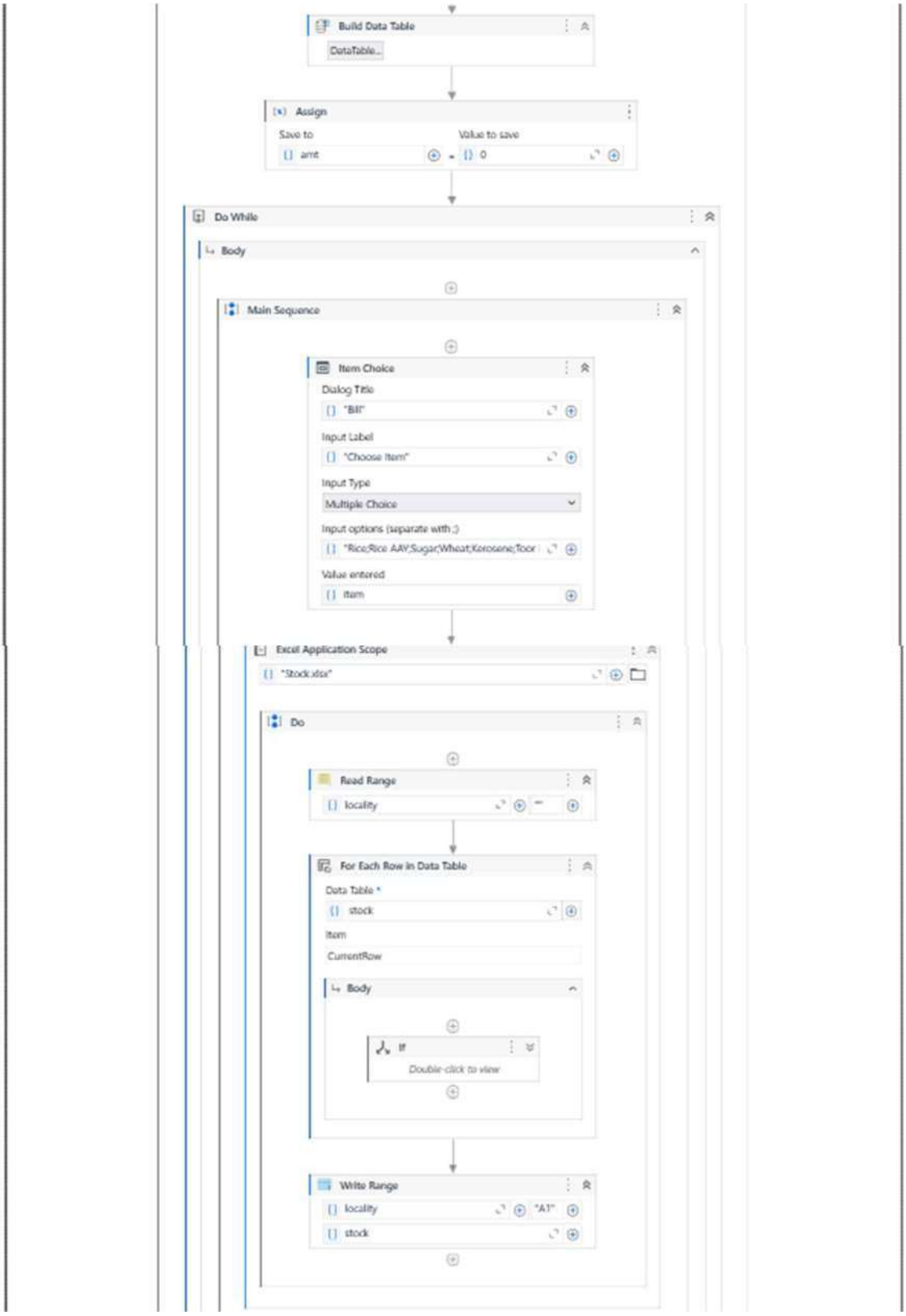

Sequence
Comment Out
Ignored Activities
Location_Based_Stock
Open Browser
Url *
"https://tnpds.gov.in/"
Do
Maximize Window
Click 'LABEL'
English
Click 'A formj_lidt111'
PDS Reports
Anchor Base
Find Element 'H6'
North Chennai
02
Click 'A StateLevelDetailsFor...'
North Chennai
Click 'A StateLevelDetailsFor...'
Commodity Details
Delay
Duration *
00h 00m 15.000s
Recording Sequence
Attach Browser 'PDSReport Page'
Do

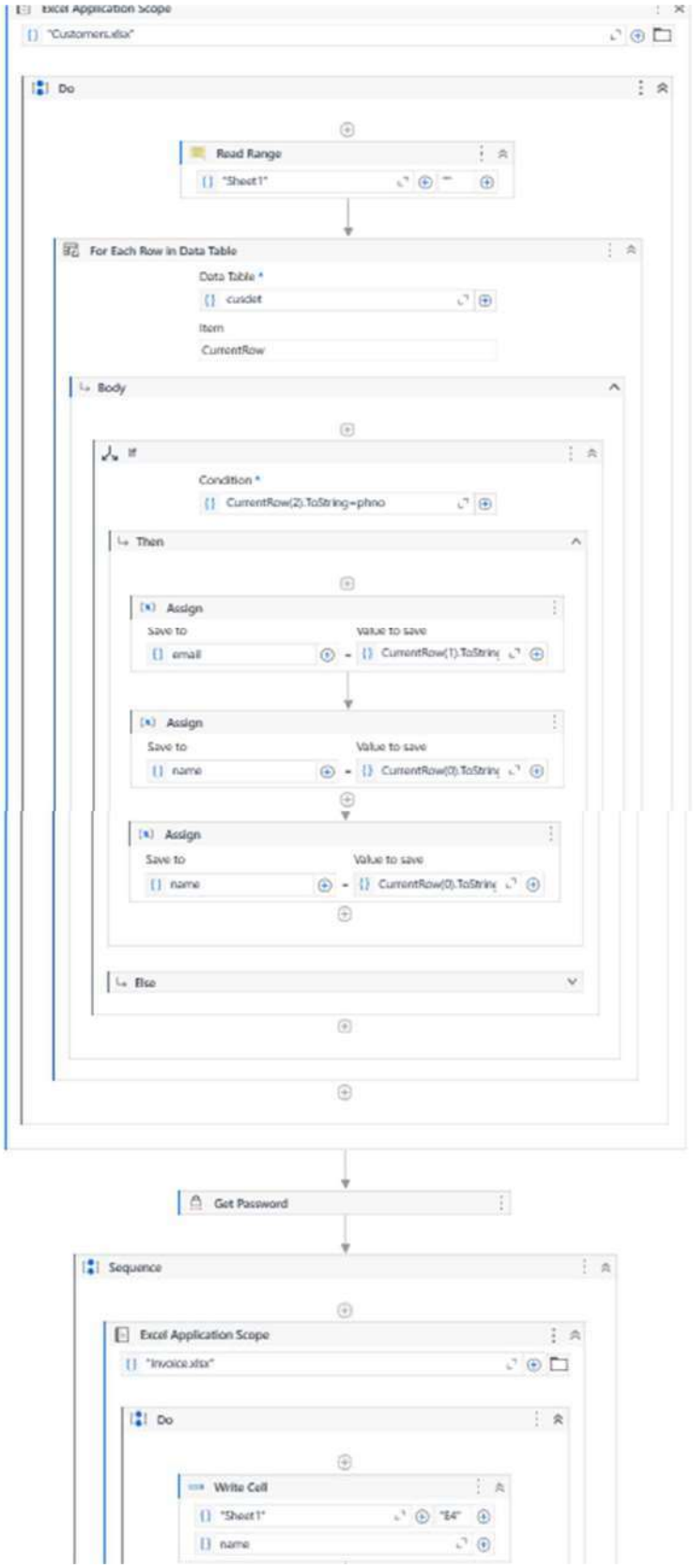

Extract Structured Data 'TBODY dlgform\1.f...
Write Range Workbook
"Stock.xlsx"
"Mylapore"
"B2"
annanagar
Click 'SPAN'
Click 'A StateLevelDetailsFor...'
Delay
Duration *
00h 00m 15.000s
Recording Sequence
Attach Browser 'PDSReport Page'
Do
Extract Structured Data 'TBODY dlgform\1.f...
Write Range Workbook
"Stock.xlsx"
"Saidapet"
"B2"
saidapet

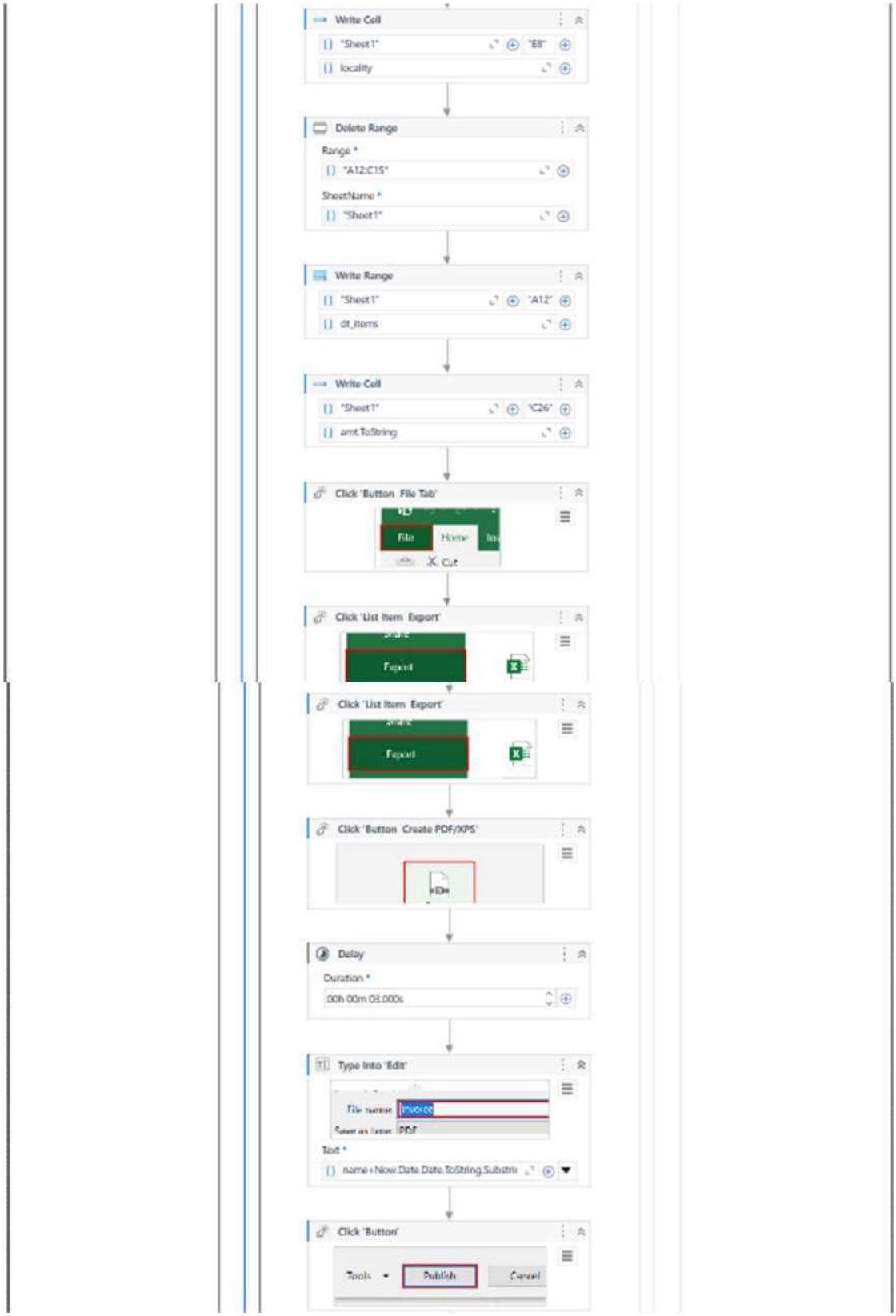

Click 'TD'
Click 'DIV'
Commodity Details
Delay
Duration *
00h 00m 15.000s
Recording Sequence
Attach Browser 'PDSReport Page'
Do
Extract Structured Data 'TBODY dlgform1f...
Write Range Workbook
"Stock.xlsx"
"Maduravayal"
"B2"
maduravayal


Read Range Workbook
"Customers.xlsx"
"Sheet1"
User Phone Number
Dialog Title
"Ration Booking"
Input Label
"Enter your mobile number"
Input Type
Text Box
Value entered
phno
For Each Row in Data Table
Data Table *
cusdet
Item
CurrentRow
Body
If
Condition *
CurrentRow(2).ToString=phno
Then
Assign
Save to
userflag
Value to save
CInt("1")
Else
If
Condition *
userflag=0
Then
Message Box
Text *
"You are a new user ,Please Register to con


Input Dialog
Dialog Title
"User Details"
Input Label
"Enter your Name"
Input Type
Text Box
Value entered
username
Input Dialog
Dialog Title
"User Details"
Input Label
"Enter your Email"
Input Type
Text Box
Value entered
useremail
Add Data Row
ArrayRow *
{username,useremail,phno}
DataRow *
The DataRow object to be added to the Da
DataTable *
cusdet
Write Range Workbook
"Customers.xlsx"
"Sheet1"
Cell
cusdet
Else
Ration Booking
Nearest Locality
Dialog Title
"Ration Booking"
Input Label
"Choose your nearest Locality"
Input Type
Multiple Choice
Input options (separate with ;)
"Mylapore;Maduravayal;Saidapet"
Value entered
locality
Build Data Table
DataTable...


Build Data Table
DataTable...
Assign
Save to
amt
Value to save
0
Do While
Body
Main Sequence
Item Choice
Dialog Title
"Bill"
Input Label
"Choose Item"
Input Type
Multiple Choice
Input options (separate with ;)
"Rice;Rice AAY;Sugar;Wheat;Kerosene;Toor
Value entered
item
Excel Application Scope
"Stock.xlsx"
Do
Read Range
locality
For Each Row in Data Table
Data Table *
stock
Item
CurrentRow
Body
If
Double-click to view
Write Range
locality
"A1"
stock


Excel Application Scope
"Customers.xlsx"
Do
Read Range
"Sheet1"
For Each Row in Data Table
Data Table *
cusdet
Item
CurrentRow
Body
If
Condition *
CurrentRow(2).ToString=phno
Then
Assign
Save to
email
Value to save
CurrentRow(1).ToStrin
Assign
Save to
name
Value to save
CurrentRow(0).ToStrin
Assign
Save to
name
Value to save
CurrentRow(0).ToStrin
Else
Get Password
Sequence
Excel Application Scope
"Invoice.xlsx"
Do
Write Cell
"Sheet1"
"B4"
name


Write Cell
"Sheet1"
"E8"
locality
Delete Range
Range *
"A12:C15"
SheetName *
"Sheet1"
Write Range
"Sheet1"
"A12"
dt_items
Write Cell
"Sheet1"
"C26"
amt.ToString
Click 'Button File Tab'
File
Home
Cut
Click 'List item Export'
Export
Click 'List item Export'
Export
Click 'Button Create PDF/XPS'
Delay
Duration *
00h 00m 03.000s
Type into 'Edit'
File name:
Invoice
Save as type:
PDF
Text *
name + Now.Date.Date.ToString.Substri
Click 'Button'
Tools
Publish
Cancel

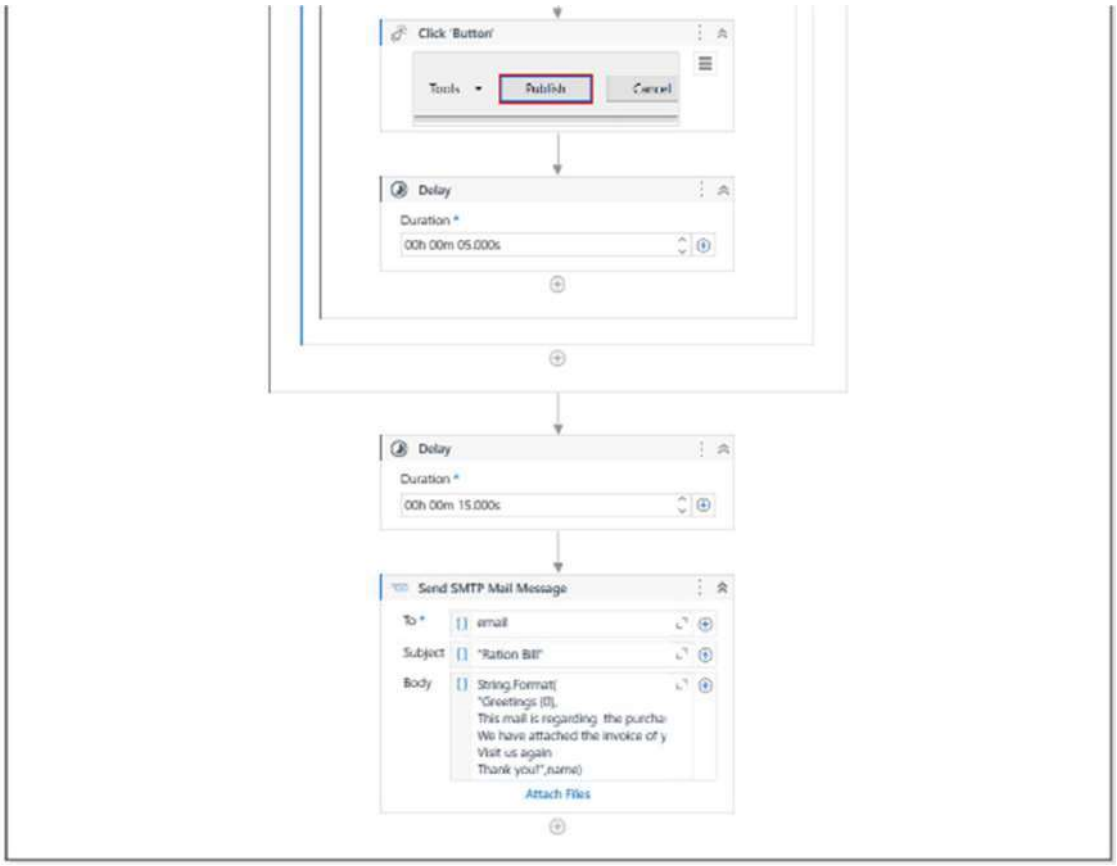


Fig 1.7 UiPath Workflow Overview

LIST OF ABBREVIATIONS

ABBREVIATION	ACCRONYM
RPA	Robotic Process Automation
UiPath	Ui Path Studio
Excel	Microsoft Excel
SMTP	Simple Mail Transfer Protocol
PDF	Portable Document Format
API	Application Programming Interface
GUI	Graphical User Interface

INTRODUCTION

Ration shops play a crucial role in distributing essential commodities to the public, especially in developing nations. Efficient stock management in such shops is necessary to avoid shortages and ensure fairness. The traditional methods of manual record-keeping often lead to inefficiencies, human errors, and delays. This project aims to automate the stock management process using UiPath Studio, making it more reliable and user-friendly. It includes user registration, stock verification, and transaction processing, culminating in the generation and email delivery of invoices. By automating these tasks, the system reduces human effort, improves accuracy, and enhances customer experience. This approach ensures a streamlined workflow for both shopkeepers and customers.

1.1 GENERAL

The project automates ration shop stock management using UiPath Studio, addressing inefficiencies and errors in traditional manual systems. It streamlines processes like user registration, stock verification, and transaction recording while ensuring real-time updates and accuracy. Automation reduces human intervention, enhances reliability, and improves service efficiency. Features include Excel integration, PDF invoice generation, and email delivery to users. This system ensures a transparent, user-friendly experience for both shopkeepers and customers. It modernizes ration shop workflows, making them faster and more effective.

1.2 OBJECTIVE

The primary objective of this project is to automate the stock management process in ration shops using UiPath Studio. It aims to streamline workflows, reduce errors, and enhance the overall efficiency of operations.

- Automate user registration and verification by checking phone numbers against an existing database.
- Enable real-time stock verification and notify users of insufficient quantities.
- Calculate the total price of selected items and display it to the user for transparency.
- Generate detailed PDF invoices for transactions and send them to users via email.

This system ensures a more reliable and user-friendly experience for both shopkeepers and customers, modernizing traditional ration shop workflows.

1.3 EXISTING SYSTEM

The existing system for ration shop stock management relies heavily on manual processes, leading to inefficiencies and a high risk of human errors. User details are often recorded on paper or in spreadsheets, making it time-consuming to verify and update information. Stock tracking is also done manually, resulting in delays and inaccuracies, particularly when checking availability for multiple items. Invoices are typically handwritten or generated using basic tools, lacking automation and real-time updates. Additionally, there is no mechanism for promptly notifying users about stock shortages or providing transaction details. These limitations reduce operational efficiency and user satisfaction, highlighting the need for an automated solution to streamline processes and ensure accuracy.

1.4 PROPOSED SYSTEM

This project incorporates several advanced features to automate and optimize the stock management process in ration shops. These features ensure accuracy, efficiency, and a seamless experience for both shopkeepers and users.

- **User Registration and Verification**
  The system verifies phone numbers against an existing database to identify new and registered users, ensuring accurate record management.
- **Real-Time Stock Verification**
  Automatically checks stock availability for selected items and notifies users of insufficient quantities, preventing overbooking.
- **Automated Invoice Generation**
  Generates detailed PDF invoices with user details, selected items, and total price, ensuring accurate and professional documentation.
- **Email Notification**
  Sends the generated PDF invoices to users via email, providing a convenient and transparent transaction experience.
- **Dynamic Transaction Management**
  Allows users to select multiple items, calculates the total price, and updates transaction records efficiently in an Excel sheet.

LITERATURE REVIEW

The literature on **Ration Shop Stock Management** system highlights the challenges of manual processes, such as inaccurate stock tracking, time-consuming record-keeping, and human errors. Previous studies have shown that automation can significantly improve efficiency by eliminating these issues, allowing for real-time updates, accurate inventory management, and streamlined operations. Robotic Process Automation (RPA) tools like UiPath have been recognized for their ability to automate repetitive tasks, enhancing productivity in various sectors, including public distribution systems. Research also indicates that automation can improve transparency and customer satisfaction by providing quicker services and reducing errors. This project leverages such advancements to create a more efficient and reliable system for ration shop management.

2.1 GENERAL

Ration shops are a critical part of the public distribution system, providing essential food items at subsidized rates to eligible households. Managing these shops effectively requires accurate stock tracking, timely updates, and efficient transaction processing. However, traditional methods often rely on manual efforts, which can lead to human errors, delays, and inefficiencies in operations. These challenges underscore the need for automation in improving ration shop management.

The proposed system leverages UiPath Studio to automate the entire process, from user registration to stock verification and invoice generation. It ensures accurate tracking of user details and stock availability, reducing the risk of errors. By automating these tasks, the system can provide real-time updates on stock levels, notify users of shortages, and generate invoices automatically, saving valuable time for shopkeepers and improving service quality.

Through the use of Robotic Process Automation (RPA), this system enhances the overall efficiency and transparency of ration shop operations. It eliminates manual intervention, ensuring faster and more reliable processing. Additionally, by integrating email notifications and PDF invoice generation, the system offers a convenient experience for users while streamlining administrative tasks for ration shop staff. This approach modernizes the workflow and ensures that both shopkeepers and customers benefit from a more efficient, error-free system.

Key Areas of Research Covered:

1. **Robotic Process Automation (RPA) in Stock Management**
   Research shows that RPA tools like UiPath can automate repetitive tasks in stock management, reducing manual efforts and minimizing errors. Automation ensures more accurate stock tracking, faster updates, and improved efficiency in inventory management.

2. **Automation of User Registration and Verification**
   Automating user registration speeds up the process, ensuring accurate data entry and reducing errors. Studies highlight how RPA can verify user details in real time, enhancing the registration experience and reducing the workload for staff.

3. **Stock Validation and Real-Time Data Extraction**
   Automating stock validation allows real-time checks against live inventory data, improving accuracy in stock management. Research focuses on integrating systems to automatically extract data from sources like Excel, ensuring up-to-date stock information.

4. **Automated Invoice Generation and Email Delivery**
   Automating invoice generation eliminates manual data entry and ensures consistent, accurate billing. Studies emphasize how automating email delivery of invoices improves communication by ensuring timely, error-free transactions for customers.

5. **Error Handling and Quality Assurance in Automated Systems**
   Effective error handling is crucial in automated systems to ensure data accuracy and prevent issues like incorrect stock levels or invoice mistakes. Research focuses on building robust error management protocols to maintain reliability in automation workflows.

6. **Scalability and Efficiency of Automation Systems**
   Scalable automation systems are designed to handle increasing transaction volumes without performance degradation. Research highlights techniques to optimize workflows for faster processing, ensuring the system can support both small and large-scale operations efficiently.

SYSTEM DESIGN

The System Design for the Ration Shop Stock Management System outlines the architecture and components necessary to automate the process of user registration, stock validation, transaction recording, and invoice generation. The system uses UiPath Studio to streamline tasks such as verifying user details, checking stock availability, calculating prices, generating invoices, and sending emails. The design focuses on ensuring accuracy, speed, and reliability, with built-in error handling to address potential issues. By automating the entire workflow, the system reduces manual effort, minimizes errors, and ensures smooth and timely transactions.

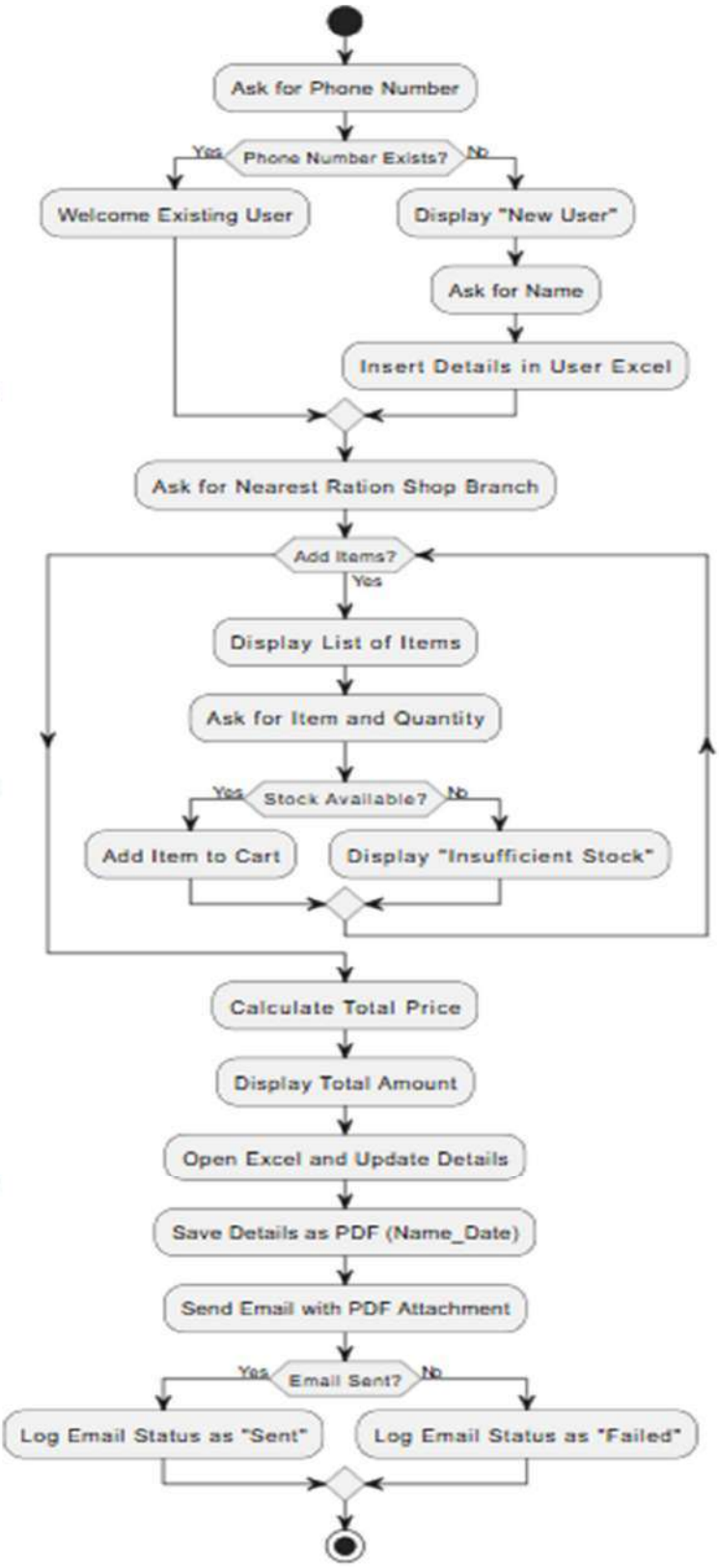
3.1 GENERAL

The System Design specifies the structure and operation of the **Ration Shop Stock Management** System, detailing its components and their interactions. It covers data flow from user input (phone number, item selection) to output (stock validation, invoice generation, and email distribution). Key considerations include ensuring seamless integration between modules, optimizing performance, and incorporating error handling for data accuracy. The design also addresses scalability to manage increasing user volumes and transactions. It prioritizes automation to enhance efficiency, reduce manual tasks, and provide a user-friendly experience for both shopkeepers and customers.

Key Objectives of the System Design:

1. **Automation:** Streamline the entire process of user registration, stock validation, and invoice generation to minimize manual intervention.
2. **Accuracy:** Ensure error-free data entry, stock verification, and price calculation for accurate transactions.
3. **Efficiency:** Optimize workflows for fast stock checks, quick invoice generation, and prompt email notifications.
4. **Scalability:** Design the system to handle increasing numbers of users, stock items, and transactions as the system grows.
5. **Integration:** Ensure smooth interaction between components like user details, stock data, invoice templates, and email distribution systems.
6. **Security:** Safeguard sensitive user and transaction data with appropriate security measures, including encryption and access controls.

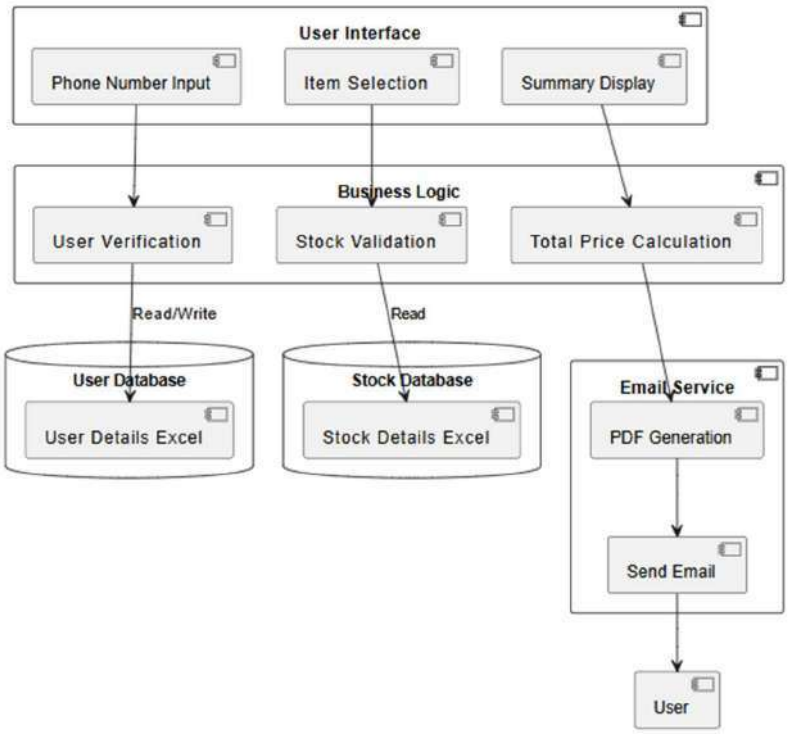
3.1.1 SYSTEM FLOW DIAGRAM



Key Stages in the System Flow Diagram:

- **User Login/Registration**: Ask for the phone number, check if the user exists, and register new users if needed.
- **Branch Selection**: Ask the user to choose the nearest ration shop branch.
- **Item Selection**: Display available items and allow the user to select items and quantities.
- **Stock Check**: Verify the requested quantity against available stock and notify if insufficient.
- **Order Finalization**: Calculate the total price and update the order details in Excel.
- **Save and Send**: Save order details as a PDF and email it to the user.

3.1.2 ARCHITECTURE DIAGRAM



The Architecture Diagram illustrates the structural design of the **Ration Shop Stock Management System**. It shows the major components, their interactions, and the data flow between them. The architecture typically includes:

Main Components:

1. **User Input Module**: Captures user details like phone numbers, names, and branch preferences through the user interface.
2. **User Verification System**: Checks the phone number against the User Excel database to determine if the user is new or existing and handles the registration of new users.
3. **Branch and Item Selection Module**: Allows users to select the nearest ration shop and choose items and quantities from a predefined list.
4. **Stock Validation System**: Verifies the availability of selected items in the Stock Excel database and alerts users if the requested quantity is insufficient.
5. **Order Summary and Calculation Module**: Calculates the total cost of all selected items and displays the summary to the user.
6. **Order Management and Storage**: Updates the User Excel with order details, including the user's name, phone number, selected items, quantities, and total price.
7. **PDF Generation System**: Automatically generates a PDF receipt containing the order details, naming the file based on the user's name and date.
8. **Email Notification System**: Sends the generated PDF as an email attachment to the user, with a status logged to track whether the email was sent successfully or failed.
9. **Error Handling and Logging Module**: Captures and logs errors like invalid inputs, stock unavailability, or email failures for debugging and monitoring.
10. **Security Features**: Ensures user data is protected with access control and encryption for secure storage and communication.

3.1.3 SEQUENCE DIAGRAM

A **Sequence Diagram** visually represents the interactions between system components in a sequential manner. Below is an outline for the sequence of operations in the **Ration Shop Stock Management** project:

Sequence Diagram for the Ration Shop Stock Management:

1. User Verification

- **User Action**: The user enters their phone number.
- **System Process**:
    - The system checks the phone number in the **User Excel**.
    - If the phone number exists, the system welcomes the user as an existing user.
    - If the phone number is new, the system prompts the user to enter their name and registers the user by saving the details in the **User Excel**.

2. Branch Selection

- **User Action**: The user selects the nearest ration shop branch from a list.
- **System Process**: The system records the selected branch for the user's transaction.

3. Item Selection

- **User Action**: The system displays a list of available items. The user selects an item and specifies the quantity.
- **System Process**:
    - The system validates the requested quantity by checking the **Stock Excel**.
    - If sufficient stock is available, the system adds the item to the user's cart.
    - If stock is insufficient, the system notifies the user and prompts them to select another item.

4. Repeated Item Selection

- **User Action**: The user can continue adding more items by repeating the selection process.
- **System Process**:
    - The system continues validating stock and adding items to the cart until the user finishes item selection.

5. Total Price Calculation

- **System Process**:
    - The system calculates the total price of all selected items in the cart.
    - The total price is displayed to the user.

6. Order Record and PDF Generation

- **System Process**:
    - The system updates the **User Excel** with the user's name, phone number, selected items, quantities, and total price.
    - The system generates a PDF receipt containing the order details and saves it with a filename based on the user's name and date.

7. Email Notification

- **System Process**:
    - The system sends an email to the user with the PDF receipt attached.
    - It logs whether the email was successfully sent or failed.

8. Completion

- **System Process**: The process ends after confirming that the email has been sent, and all user details are logged.

4. PROJECT DESCRIPTION

The **Ration Shop Stock Management System** is an automated solution developed using UiPath Studio to optimize the process of user registration, stock validation, invoice generation, and email distribution. The system integrates with an Excel sheet containing user and stock data, validating user phone numbers and checking the availability of requested stock. When a user selects items, the system checks the inventory in real time, calculates the price, and generates an invoice. The system automatically sends the invoice as a PDF attachment via email to the user. The design focuses on reducing manual intervention, ensuring accuracy, and enhancing efficiency in ration shop operations.

4.1 METHODOLOGIES

The methodology for the Ration Shop Stock Management System follows a structured approach to ensure efficient automation. The key phases are as follows:

- **Requirement Analysis:**
  The first step involves understanding the system's specific needs, including user registration, stock validation, price calculation, and email distribution. Stakeholders provide input on required features, such as handling stock requests, validating user information, and generating invoices.
- **System Design:**
  In this phase, the system architecture is developed, detailing the components and interactions. Key components include user data input, stock validation, invoice generation, and email distribution. Security considerations, such as protecting user data and ensuring compliance, are integrated into the design.
- **Implementation:**
  Using UiPath Studio, workflows are created for each module (user registration, stock validation, invoice generation, email distribution). Data extraction from Excel, stock validation, price calculation, and automated email handling are implemented. Error handling mechanisms are incorporated to ensure smooth operation.
- **Testing:**
  The system undergoes comprehensive testing to ensure all modules work as intended. Testing focuses on validating user data, checking stock availability, generating accurate

invoices, and ensuring successful email delivery. Error scenarios such as missing stock or failed email deliveries are also tested.

- **Deployment:**
  After successful testing, the system is deployed in the target environment, such as a ration shop or distribution center. User training ensures effective system adoption and usage.
- **Maintenance and Improvement:**
  Post-deployment, the system is monitored for performance, and any issues are addressed. Regular updates and improvements are made based on user feedback, enhancing system functionality and addressing emerging needs.

4.1.1 MODULES

The system is broken down into several key modules, each responsible for specific tasks in the ration shop stock management process:

- **Input Data Module:**
  Reads and validates user details (phone numbers, names) from an Excel sheet to ensure correct registration.
- **Stock Validation Module:**
  Compares user requests with available stock data and ensures accurate validation of stock quantities.
- **Price Calculation Module:**
  Calculates the total price based on the items selected by the user, considering the available stock and quantity.
- **Invoice Generation Module:**
  Generates an invoice in PDF format, detailing the selected items, quantities, prices, and total amount.
- **Email Distribution Module:**
  Sends the generated invoice as a PDF attachment to the user's email, ensuring timely and accurate delivery.
- **Error Handling and Logging Module:**
  Monitors for errors, such as insufficient stock or failed email delivery, and logs these issues for troubleshooting.

- **Monitoring and Reporting Module:**
  Tracks system performance and generates reports on transactions, stock levels, and user activity.
- **Security and Compliance Module:**
  Ensures that sensitive user and transaction data is protected with encryption and complies with data privacy regulations.

5. CONCLUSIONS

The **Ration Shop Stock Management System** offers an efficient, automated solution for managing stock, processing user orders, generating invoices, and distributing them via email. By leveraging UiPath Studio, the system significantly reduces manual intervention, enhancing accuracy and speed while minimizing errors. The modular design ensures scalability, allowing the system to adapt to varying levels of demand and user activity. Key benefits include improved efficiency, reduced manual workload, and secure handling of user data. Overall, the system provides a reliable, automated solution for ration shop operations, streamlining the entire process and enhancing productivity.

5.1 GENERAL

The System Design specifies the structure and operation of the Ration Shop Stock Management System, detailing its components and their interactions. It covers data flow from user input (phone number, item selection) to output (stock validation, invoice generation, and email distribution). Key considerations include ensuring seamless integration between modules, optimizing performance, and incorporating error handling for data accuracy. The design also addresses scalability to manage increasing user volumes and transactions. It prioritizes automation to enhance efficiency, reduce manual tasks, and provide a user-friendly experience for both shopkeepers and customers.

Key Objectives of the System Design:

1. **Automation:** Streamline the entire process of user registration, stock validation, and invoice generation to minimize manual intervention.
2. **Accuracy:** Ensure error-free data entry, stock verification, and price calculation for accurate transactions.
3. **Efficiency:** Optimize workflows for fast stock checks, quick invoice generation, and prompt email notifications.
4. **Scalability:** Design the system to handle increasing numbers of users, stock items, and transactions as the system grows.
5. **Integration:** Ensure smooth interaction between components like user details, stock data, invoice templates, and email distribution systems.
6. **Security:** Safeguard sensitive user and transaction data with appropriate security measures, including encryption and access controls.

REFERENCES

- **UiPath Documentation**: UiPath provides comprehensive resources on how to build automation workflows, including integrating data from various sources and automating email distribution. Available at: https://docs.uipath.com/.
- **Excel File Handling in UiPath**: A guide to working with Excel files in UiPath to extract, process, and validate data for use in automated workflows. Available at: https://docs.uipath.com/activities/docs/excel-application-scope.
- **SMTP Email Automation in UiPath**: UiPath documentation on automating email sending through SMTP for sending certificates to recipients. Available at: https://docs.uipath.com/activities/docs/send-mail.
- **Robotic Process Automation (RPA) Overview**: Introduction to RPA and its benefits in streamlining business processes, including certificate generation and distribution. Available at: https://www.uipath.com/rpa/robotic-process-automation.
- **Data Privacy and GDPR Compliance**: Guidance on data protection and compliance with regulations such as GDPR when handling sensitive personal data, available at: https://gdpr.eu/.
- **Security in Automation**: An overview of securing automated systems, including encryption and access control. Available at: https://www.uipath.com/security.

APPENDICES

The **Appendices** section includes supplementary material that supports the main content of the report. It provides detailed information that might be too lengthy or technical to include in the main chapters but is still relevant to the project.

Possible Contents of the Appendices:

Appendix A: UiPath Workflow and Code Snippets

This appendix includes key steps in the Ration Shop Stock Management system, showcasing the core workflows in UiPath.

1. **Reading Data from Excel**
    - The system reads user and stock data (e.g., phone numbers, names, stock items, and quantities) from Excel sheets using the **Read Range** activity. This creates a DataTable with all necessary details, which is used for validation and subsequent actions.
2. **User Registration and Validation**
    - When a user's phone number is entered, the system checks the Excel sheet for an existing entry. If no match is found, it triggers the user registration process by asking for the name and nearest ration shop branch. The user information is then added to the Excel sheet for future reference.
3. **Stock Validation**
    - The system retrieves the stock details from an attached Excel file. It checks the availability of the requested quantity. If the stock is insufficient, the system informs the user and allows them to modify their order.
4. **Invoice Generation and PDF Creation**
    - After stock validation and item selection, the system calculates the total price and generates an invoice. The invoice is automatically populated with user details, selected items, and the total price. It is saved as a PDF with a filename that includes the user's name and the date of transaction.
5. **Email Distribution**
    - The system utilizes the **Send SMTP Mail** activity to send the generated PDF invoice as an attachment to the user's email. The email body is personalized, containing details of the transaction and any necessary information about their order.
6. **Error Handling and Logging**
    - If any issues arise during data processing, such as missing information or failed email deliveries, the system logs the error in a text file for troubleshooting. The error handling ensures that the system can recover gracefully and continue operations without manual intervention.

Appendix B: System Testing and Validation

This section describes the testing procedures and validation used to ensure the Ration Shop Stock Management System operates correctly.

1. **Test Case 1: Data Accuracy**
    - **Objective**: Ensure the system accurately reads user and stock details from the Excel files.
    - **Test**: Input sample user data and stock information into the system's Excel sheets.
    - **Expected Result**: The system should extract and display correct user and stock data without errors.
2. **Test Case 2: User Registration**
    - **Objective**: Verify that new users are registered correctly when their phone number is not found in the system.
    - **Test**: Enter a new phone number that is not listed in the Excel file.
    - **Expected Result**: The system should prompt for the user's name, ration shop branch, and item selection, and the data should be correctly added to the Excel file.
3. **Test Case 3: Stock Validation**
    - **Objective**: Ensure the system can check the stock levels and notify the user in case of insufficient quantities.
    - **Test**: Request a quantity that exceeds the available stock in the Excel file.
    - **Expected Result**: The system should display a message indicating insufficient stock and offer the option to modify the order.
4. **Test Case 4: Invoice Generation and Email Distribution**
    - **Objective**: Verify the correct generation of an invoice and its successful email distribution.
    - **Test**: After confirming a valid order, generate the invoice and send it to a test email.
    - **Expected Result**: The invoice should be saved as a PDF with the correct details, and the email should be delivered to the recipient without errors.
5. **Test Case 5: Error Handling**
    - **Objective**: Test the system's ability to handle missing or incorrect data during any step.
    - **Test**: Provide incomplete user data or enter an invalid email format.
    - **Expected Result**: The system should capture the error, log it for future review, and proceed with the other tasks without crashing.