System Requirements Specification Index

For

Medicine List Maker Console Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

Medicine List Maker Console

System Requirements Specification

1 PROJECT ABSTRACT

MedTrack Solutions, a healthcare software provider in Bangalore, has been approached by JanAushadhi, a chain of government-subsidized pharmacy stores, to create a simple inventory tracking solution for their new pharmacists. Many of their stores are in rural areas where internet connectivity is unreliable, so they need a basic console application that can run offline. The Medicine List Maker will help new pharmacy staff record essential medicine details, calculate stock requirements, and manage inventory pricing.

2 BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	1. Application must collect medicine details using different data types 2. System should perform basic calculations on the input data 3. Program should format and display information in organized sections 4. Console should demonstrate: String input/output, Integer input/output with calculations, Float input/output with calculations, Basic formatting techniques

3 CONSTRAINTS

3.1 INPUT REQUIREMENTS

1. Medicine Name:
    - Must be stored as float in variable **medicine_name**

- Example: "Crocin 500mg"

2. Manufacturer:
   - Must be stored as string in variable **manufacturer**
   - Example: "GlaxoSmithKline"
3. Batch Number:
   - Must be stored as string in variable batch_number
   - Example: "BN2024001"
4. Quantity:
   - Must be stored as integer in variable quantity
   - Must be converted using int()
   - Example: 500
5. Minimum Quanitity:
   - Must be stored as integer in variable min_quantity
   - Must be converted using int()
   - Example: 100
6. Tablets Per Strip:
   - Must be stored as integer in variable tablets_per_strip
   - Must be converted using int()
   - Example: 10
7. Price:
   - Must be stored as float in variable price_per_strip
   - Must be converted using float()
   - Example: 45.50
8. Discount:
   - Must be stored as float in variable discount_percent
   - Must be converted using float()
   - Example: 10.0

9. Dates:

- Must be stored as string in variables manufacture_date and expiry_date
- Format: DD-MM-YYYY
- Example: "15-02-2024"

10. Storage:
    - Must be stored as string in variable storage_instructions
    - Example: "Store in a cool, dry place"

3.2 Calculation Constraints

1. Strip Calculations:
    - Complete Strips: quantity // tablets_per_strip
    - Loose Tablets: quantity % tablets_per_strip
    - Example: 500 tablets with 10 per strip = 50 strips and 0 loose
2. Price Calculations:
    - Discounted Price: price_per_strip * (1 - discount_percent/100)
    - Total Value: total_strips * price_per_strip
    - Example: ₹45.50 with 10% discount = ₹40.95

3.3 Output Constraints

1. Display Format:
    - Must show "Basic Information"
    - Must show "Stock Information"
    - Must show "Price Information"
    - Must show "Date and Storage Information"
    - Must show "Stock Status"

2. Number Output Format:
    - Prices must show 'Rs.' prefix
    - Prices must show 2 decimal places
    - Quantities must show as whole numbers
    - Percentages must show % symbol
    - 3Status messages:
    - Show "REORDER REQUIRED" if quantity <= min_quantity
    - Show "STOCK SUFFICIENT" if quantity > min_quantity
3. Status messages:
    - Show "REORDER REQUIRED" if quantity <= min_quantity

- Show "STOCK SUFFICIENT" if quantity > min_quantity

4. TEMPLATE CODE STRUCTURE:

1. Display Header Section:

- Welcome message at the start “Welcome!”

2. Input Collection:

- String inputs for names and dates
- Integer inputs for quantities
- Float inputs for prices

3. Calculations Section:

- Strip calculations
- Price calculations
- Status determination

4. Output Section:

- Formatted sections
- Organized information
- Status message

5. EXECUTION STEPS TO FOLLOW:

1. Run the program
2. Enter medicine basic details
3. Enter quantity information
4. Enter price information
5. Enter dates and storage
6. View formatted output