**🧾 Software Requirements Specification (SRS)**

**for Batch Payroll Application**

**1. Introduction**

**1.1 Purpose**

The purpose of this SRS is to define the functional and non-functional requirements for the **Batch Payroll Application**. The system automates the payroll process by calculating, managing, and distributing employee payments accurately and efficiently.

**1.2 Scope**

The Batch Payroll Application manages employee payment processes including hourly, salaried, and commission-based employees. It records timecards, sales receipts, and deductions such as union dues, and generates payments through multiple methods (mail, pickup, or direct deposit). The system ensures employees are paid accurately and on time according to their payment schedule.

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS:** Software Requirements Specification
* **UI:** User Interface
* **DB:** Database
* **ID:** Identification Number
* **CRUD:** Create, Read, Update, Delete

**1.4 References**

* IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
* Company HR and Payroll Policy Manual

**1.5 Overview**

This document provides an overview of the system’s functionalities, performance requirements, user interactions, constraints, and other essential details required for development, testing, and maintenance.

**2. Overall Description**

**2.1 Product Perspective**

The Batch Payroll Application is a standalone system connected to an employee database. It interacts with external services such as banks for direct deposits and the postal service for paycheck mailing.

**2.2 Product Functions**

Major functions include:

* Managing employee information
* Recording timecards and sales receipts
* Calculating gross and net pay
* Applying union dues and deductions
* Generating pay slips and payment summaries
* Supporting multiple payment methods
* Maintaining transaction history

**2.3 User Characteristics**

* **Paymaster:** Initiates payroll runs and reviews payment reports.
* **Employees:** Submit timecards or sales receipts and receive payments.
* **System Administrator:** Manages user accounts and system configurations.

**2.4 Constraints**

* The system must run once every working day.
* Payments must follow organization-defined schedules.
* The application must comply with local labor and tax regulations.

**2.5 Assumptions and Dependencies**

* Accurate employee and sales data are available in the database.
* Reliable connectivity with banking and postal systems.
* Union deduction rates are regularly updated.

**3. Specific Requirements**

**3.1 Functional Requirements**

1. The system shall store employee details including name, ID, role, salary type, payment method, and union status.
2. The system shall allow hourly employees to submit timecards.
3. The system shall calculate overtime pay at 1.5x the hourly rate for hours exceeding 8 per day.
4. The system shall allow commission-based employees to submit sales receipts.
5. The system shall calculate commissions based on the employee’s commission rate.
6. The system shall apply union dues and additional service charges automatically.
7. The system shall generate pay slips for each employee.
8. The system shall issue payments through the selected payment method (mail, pickup, or direct deposit).
9. The system shall maintain records of all payments for audit and reporting.

**3.2 System Features**

**3.2.1 Payroll Processing**

* Automates payment calculation based on employee type.
* Generates payroll on the appropriate pay date.

**3.2.2 Employee Management**

* Adds, updates, or removes employee records.
* Tracks attendance and timecards.

**3.2.3 Union Management**

* Maintains union dues and service charge records.
* Deducts fees automatically from the employee’s paycheck.

**3.2.4 Payment Scheduling**

* Supports flexible pay schedules (weekly, bi-weekly, monthly).
* Notifies Paymaster of upcoming payroll runs.

**3.3 Interface Requirements**

**3.3.1 User Interface**

* Graphical dashboard for Paymaster and Admin.
* Simple forms for timecard and sales receipt submission.

**3.3.2 Database Interface**

* MySQL/SQL Server used for storing employee and payroll data.

**3.3.3 External System Interfaces**

* Integration with banking APIs for direct deposits.
* Interface with mailing services for physical paycheck delivery.

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* The system should process payroll for up to 1,000 employees in under 10 minutes.
* Database queries should execute within 2 seconds.

**4.2 Security Requirements**

* Authentication and authorization required for all users.
* Sensitive data (salary, bank account) must be encrypted.

**4.3 Reliability Requirements**

* System should maintain 99% uptime during payroll cycles.
* Auto-backup of payroll records daily.

**4.4 Availability**

* Accessible on all working days from 8 AM to 8 PM.

**4.5 Maintainability**

* Modular design to support updates in tax or pay rules.
* Codebase should follow standard naming and commenting practices.

**4.6 Scalability**

* System should support future expansion for more employees and additional departments.

**4.7 Usability**

* User-friendly interface requiring minimal training.
* Clear error and confirmation messages for all operations.

**5. Other Supporting Information**

**5.1 Appendices**

* Sample pay slip format
* Sample employee record schema
* Test case examples

**5.2 Index**

* Alphabetical index of key terms and abbreviations

**🧾 Software Requirements Specification (SRS)**

**for Recycling Machine Application**

**1. Introduction**

**1.1 Purpose**

The purpose of this SRS is to define the functional and non-functional requirements for the **Recycling Machine Application**. The system automates the process of accepting returnable bottles and cans, issuing receipts to customers, and allowing operators to manage and monitor the machine’s operations.

**1.2 Scope**

The Recycling Machine Application enables customers to deposit recyclable bottles and cans, automatically identifies item types, and provides refunds through printed receipts. Operators can view daily reports, update deposit values, and handle technical issues. The system also monitors machine status and triggers alarms when a malfunction occurs.

**1.3 Definitions, Acronyms, and Abbreviations**

* **RMA:** Recycling Machine Application
* **Operator:** Authorized person managing and maintaining the machine
* **Customer:** User depositing recyclable items
* **Deposit Value:** Refund amount per item type

**1.4 References**

* IEEE Std 830-1998: Software Requirements Specification Standard
* Local recycling and refund regulations

**1.5 Overview**

This document provides the overall system description, key functions, and performance requirements of the Recycling Machine Application.

**2. Overall Description**

**2.1 Product Perspective**

The system is an embedded software component running inside a recycling machine. It interacts with item recognition sensors, receipt printers, and operator interfaces.

**2.2 Product Functions**

* Identify bottles and cans deposited by customers.
* Maintain a running total of items and refunds.
* Generate and print customer receipts.
* Produce daily operational reports for operators.
* Allow operators to update deposit values.
* Detect technical issues and trigger alarms.

**2.3 User Characteristics**

* **Customer:** Regular users with no technical knowledge.
* **Operator:** Technically trained personnel managing machine operation and maintenance.

**2.4 Constraints**

* Machine must operate continuously during service hours.
* System must function in an embedded environment with limited resources.
* Receipt printer and sensors must be available for operation.

**2.5 Assumptions and Dependencies**

* Sensors accurately identify item types (bottles/cans).
* Machine has a functioning printer and sufficient paper.
* Operator regularly checks system logs and alarms.

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Customer Interactions**

1. The system shall allow customers to insert bottles and cans.
2. The system shall identify each item type (bottle or can).
3. The system shall record the quantity and type of returned items.
4. The system shall calculate the refund value based on deposit rates.
5. The system shall generate a receipt containing:
   * List of items deposited.
   * Deposit value for each item type.
   * Total refund amount.

**3.1.2 Operator Interactions**

1. The system shall allow the operator to view daily statistics (number of items, total refunds).
2. The system shall generate an end-of-day report summarizing total bottles and cans collected.
3. The system shall allow the operator to modify deposit values.
4. The system shall detect malfunctions (e.g., jammed items, empty paper roll).
5. The system shall trigger an alarm when a malfunction occurs.
6. The system shall allow the operator to acknowledge and clear the alarm once resolved.

**3.2 System Features**

**3.2.1 Item Recognition and Recording**

* Detects bottles and cans through sensors.
* Stores transaction data for each customer session.

**3.2.2 Receipt Printing**

* Generates receipts with item list and total refund.
* Sends print commands to the embedded receipt printer.

**3.2.3 Daily Reporting**

* Aggregates daily data and generates operator reports.

**3.2.4 System Monitoring and Alerts**

* Continuously monitors machine status.
* Triggers alerts in case of technical issues.

**3.3 Interface Requirements**

**3.3.1 User Interface**

* Simple touchscreen or button interface for customers.
* Admin interface for operators with options to view and modify data.

**3.3.2 Hardware Interface**

* Sensors for detecting item types.
* Embedded printer for receipts.
* Display for customer messages.

**3.3.3 Database Interface**

* Local database storing transactions and reports.

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* Item recognition should occur within 2 seconds.
* Receipts must print within 5 seconds after customer request.
* System should process up to 500 items per day.

**4.2 Security Requirements**

* Only authorized operators can access configuration settings.
* Operator actions must be logged for auditing.

**4.3 Reliability Requirements**

* System must recover automatically after minor power interruptions.
* Transaction data should not be lost during failures.

**4.4 Availability**

* System must be operational at least 95% of the time during service hours.

**4.5 Maintainability**

* System should support easy updates to deposit values and reporting parameters.

**4.6 Usability**

* Customer interface must be intuitive with minimal steps.
* Operator dashboard should clearly display key information.

**5. Other Supporting Information**

**5.1 Appendices**

* Sample Customer Receipt Format
* Example Daily Operator Report

**5.2 Index**

* Definitions of Customer, Operator, Deposit Value