**Automobile Spare Parts Shop Automation System (ASPAS)**

**Use Cases**

**Sales Management**

**ID:** **SM-001**

**Title:** Process Sale

**Description:** This use case allows sales staff to process customer purchases of automobile spare parts, calculate totals, process payment, and update inventory.

**Primary Actor:** Sales Staff

**Secondary Actor:** System

**Preconditions:** Customer has selected items to purchase.

**Postconditions:** Sale is completed, inventory is updated, receipt is generated.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Sales staff initiates new sale
2. System displays sale interface
3. Sales staff enters items being purchased
4. System calculates total price
5. Sales staff processes payment
6. System updates inventory

**Extensions or Alternate Flow:**

1. Item not found in system
   * Sales staff notifies customer
   * Sale is cancelled or continues with available items
2. Payment declined
   * System displays error message
   * Customer tries alternative payment method or cancels purchase

**Frequency of Use:** Multiple times daily

**Status:** Planned

**Owner:** Sales Module Lead

**Priority:** High

**ID:** **SM-002**

**Title:** Search Parts

**Description:** This use case enables searching for parts in the inventory database using various criteria.

**Primary Actor:** Sales Staff

**Secondary Actor:** System

**Preconditions:** User is authenticated in the system.

**Postconditions:** Relevant parts information is displayed.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Sales staff enters search criteria (part number, name, etc.)
2. System queries database
3. System displays matching results

**Extensions or Alternate Flow:**

1. No matching parts found
   * System displays "No results found" message
   * Suggests broadening search criteria

**Frequency of Use:** Multiple times daily

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** High

**ID:** **SM-003**

**Title:** Generate Daily Sales Report

**Description:** This use case creates a report of all sales transactions for the current day.

**Primary Actor:** Sales Staff

**Secondary Actor:** System

**Preconditions:** At least one sale has been processed during the day.

**Postconditions:** Daily sales report is available for viewing/printing.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Sales staff selects report generation option
2. System compiles all sales data for the specified day
3. System generates formatted report

**Extensions or Alternate Flow:**

1. No sales for the day
   * System generates empty report with zero totals
2. System error during report generation
   * System displays error message and logs the issue
   * User can retry or contact support

**Frequency of Use:** Daily

**Status:** Planned

**Owner:** Reporting Module Lead

**Priority:** Medium

**ID:** **SM-004**

**Title:** Generate Monthly Sales Graph

**Description:** This use case creates visual representation of sales data over a month.

**Primary Actor:** Sales Staff

**Secondary Actor:** System

**Preconditions:** Sales data for the month is available.

**Postconditions:** Monthly sales graph is available for viewing/analysis.

**Dependency:** Extends Generate Daily Sales Report (SM-003)

**Generalization:** None

**Main Success Scenario:**

1. Sales staff selects monthly graph option
2. System aggregates sales data by day/week
3. System generates visual graph representation

**Extensions or Alternate Flow:**

1. Insufficient data for meaningful graph
   * System displays warning about limited data
   * Generates graph with available data

**Frequency of Use:** Monthly

**Status:** Planned

**Owner:** Reporting Module Lead

**Priority:** Medium

**Vendor Management**

**ID:** **VM-001**

**Title:** Receive Orders from Vendors

**Description:** This use case processes incoming shipments from parts vendors.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** Vendor has sent parts to the shop.

**Postconditions:** New inventory is registered in the system.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner initiates receiving process
2. System displays order details
3. Shop owner confirms items received
4. System updates inventory

**Extensions or Alternate Flow:**

1. Received items do not match order
   * Shop owner notes discrepancy
   * System records partial receipt and flags discrepancy for follow-up
2. Damaged items received
   * Shop owner marks items as damaged
   * System creates return request automatically

**Frequency of Use:** Weekly

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** High

**ID:** **VM-002**

**Title:** Add Vendor

**Description:** This use case adds a new vendor to the system.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** New vendor relationship has been established.

**Postconditions:** New vendor is added to the system database.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner selects add vendor option
2. Shop owner enters vendor details
3. System validates and saves information

**Extensions or Alternate Flow:**

1. Validation fails
   * System displays error message
   * Shop owner corrects information and resubmits
2. Duplicate vendor detected
   * System alerts shop owner about potential duplicate
   * Shop owner confirms new entry or cancels operation

**Frequency of Use:** Occasionally

**Status:** Planned

**Owner:** Vendor Module Lead

**Priority:** Medium

**ID:** **VM-003**

**Title:** Place Orders with Vendors

**Description:** This use case creates and submits orders to vendors for new parts.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** Shop owner has determined parts to order.

**Postconditions:** Order is created and sent to vendor.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner creates new order
2. System displays order form
3. Shop owner selects parts and quantities
4. System calculates order total
5. Shop owner confirms order

**Extensions or Alternate Flow:**

1. Required part information missing
   * System alerts shop owner about incomplete information
   * Shop owner adds missing details or cancels specific items
2. Vendor currently unavailable in system
   * Shop owner must add vendor first
   * System redirects to Add Vendor use case

**Frequency of Use:** Weekly

**Status:** Planned

**Owner:** Vendor Module Lead

**Priority:** High

**ID:** **VM-004**

**Title:** Update Vendor Information

**Description:** This use case modifies existing vendor details in the system.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** Vendor exists in the system database.

**Postconditions:** Vendor information is updated in the database.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner searches for vendor
2. System displays vendor details
3. Shop owner modifies information
4. System saves updated details

**Extensions or Alternate Flow:**

1. Vendor not found
   * System displays error message
   * Shop owner can add new vendor instead
2. Invalid information entered
   * System alerts shop owner about invalid fields
   * Shop owner corrects information and resubmits

**Frequency of Use:** Occasionally

**Status:** Planned

**Owner:** Vendor Module Lead

**Priority:** Low

**Inventory Management**

**ID:** **IM-001**

**Title:** Track Inventory Levels

**Description:** This use case monitors current stock levels of all parts.

**Primary Actor:** System

**Secondary Actor:** None

**Preconditions:** Parts exist in inventory database.

**Postconditions:** Current inventory status is maintained.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. System checks inventory counts
2. System compares against threshold values
3. System flags items below threshold

**Extensions or Alternate Flow:**

1. Database connection error
   * System logs error
   * System attempts reconnection at set intervals
   * System sends notification to administrator if problem persists

**Frequency of Use:** Continuously

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** Critical

**ID:** **IM-002**

**Title:** View Parts by Location

**Description:** This use case displays parts organized by their storage location.

**Primary Actor:** Sales Staff/Shop Owner

**Secondary Actor:** System

**Preconditions:** Parts have assigned locations in the system.

**Postconditions:** Parts are displayed by their storage location.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. User selects location view option
2. System queries database by location
3. System displays parts organized by location

**Extensions or Alternate Flow:**

1. No locations defined
   * System prompts user to set up locations first
2. Selected location contains no parts
   * System displays empty location message

**Frequency of Use:** Multiple times daily

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** Medium

**ID:** **IM-003**

**Title:** Update Part Information

**Description:** This use case modifies details for existing parts in inventory.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** Part exists in the system database.

**Postconditions:** Part information is updated in the database.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner searches for part
2. System displays part details
3. Shop owner modifies information
4. System saves updated details

**Extensions or Alternate Flow:**

1. Part not found
   * System displays error message
   * Shop owner can add new part instead
2. Invalid information entered
   * System alerts shop owner about invalid fields
   * Shop owner corrects information and resubmits

**Frequency of Use:** Occasionally

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** Medium

**ID:** **IM-004**

**Title:** Set Threshold Values

**Description:** This use case establishes minimum stock levels that trigger reordering.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** Parts exist in inventory database.

**Postconditions:** Inventory thresholds are updated.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner selects threshold management
2. System displays current threshold settings
3. Shop owner modifies thresholds as needed
4. System saves new values

**Extensions or Alternate Flow:**

1. Invalid threshold values entered
   * System alerts shop owner about invalid thresholds
   * Shop owner corrects values and resubmits

**Frequency of Use:** Monthly

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** Medium

**ID:** **IM-005**

**Title:** Add New Part

**Description:** This use case creates entries for new parts in the inventory system.

**Primary Actor:** Shop Owner

**Secondary Actor:** System

**Preconditions:** New part needs to be added to inventory.

**Postconditions:** New part is added to the system database.

**Dependency:** None

**Generalization:** None

**Main Success Scenario:**

1. Shop owner selects add part option
2. System displays part entry form
3. Shop owner enters part details
4. System validates and saves information

**Extensions or Alternate Flow:**

1. Validation fails
   * System displays error message
   * Shop owner corrects information and resubmits
2. Duplicate part detected
   * System alerts shop owner about potential duplicate
   * Shop owner confirms new entry or cancels operation

**Frequency of Use:** Weekly

**Status:** Planned

**Owner:** Inventory Module Lead

**Priority:** High

**System**

**ID:** **SYS-001**

**Title:** Calculate Average Weekly Sales

**Description:** This use case automatically calculates average sales for each part on a weekly basis.

**Primary Actor:** System

**Secondary Actor:** None

**Preconditions:** Sales data exists for parts.

**Postconditions:** Weekly sales averages are available for reporting.

**Dependency:** Includes Track Inventory Levels (IM-001)

**Generalization:** None

**Main Success Scenario:**

1. System aggregates sales data by part
2. System calculates weekly averages
3. System updates sales metrics

**Extensions or Alternate Flow:**

1. Insufficient data for calculation
   * System marks metrics as preliminary until sufficient data exists
2. Calculation error
   * System logs error
   * System uses last valid calculation
   * System notifies administrator

**Frequency of Use:** Weekly

**Status:** Planned

**Owner:** System Module Lead

**Priority:** Medium

**ID:** **SYS-002**

**Title:** Generate Order Recommendations

**Description:** This use case automatically suggests parts to reorder based on inventory levels and sales data.

**Primary Actor:** System

**Secondary Actor:** None

**Preconditions:** Inventory tracking is active, sales data exists.

**Postconditions:** Order recommendations are available for review.

**Dependency:** Extends Track Inventory Levels (IM-001) Includes Calculate Average Weekly Sales (SYS-001)

**Generalization:** None

**Main Success Scenario:**

1. System identifies parts below threshold
2. System analyzes sales velocity
3. System generates recommended order quantities
4. System presents recommendations to shop owner

**Extensions or Alternate Flow:**

1. No parts below threshold
   * System generates empty recommendation report
2. Insufficient sales data for certain parts
   * System uses default recommendation based on thresholds only
   * System flags these recommendations as limited

**Frequency of Use:** Weekly

**Status:** Planned

**Owner:** System Module Lead

**Priority:** High