**Software Requirements**

**Specification**

**for**

**PetroPulse**

**Petrol Station**

**Management System**

**Prepared by**

***Anas Bin Rashid 22I-0907***

***Adan Malik 22I-1000***

***Hasnain Akhtar 22I-1241***

**April 20, 2025**

# **Product Backlog/User Stories**

| **ID** | **User Story** | **Acceptance Criteria** | **Estimation** | **Priority** |
| --- | --- | --- | --- | --- |
| 1 | **User Registration** - As a new user, I want to register on the system so that I can access PetroPulse services. | **Given**: A new user navigates to the registration page. **When**: They provide valid details (name, email, phone, password). **Then**: The system successfully creates an account and sends a confirmation email. | 3 | High |
| 2 | **User Login** - As a registered user, I want to log into the system so that I can access my dashboard. | **Given**: A user is on the login page. **When**: They enter valid credentials. **Then**: The system grants access to their dashboard. | 2 | High |
| 3 | **Password Reset** - As a user, I want to reset my password so that I can regain access to my account if I forget it. | **Given**: A user is on the login page. **When**: They click on "Forgot Password" and enter their email. **Then**: The system sends a password reset link to their email. | 3 | Medium |
| 4 | **Add Fuel Station** - As an owner, I want to add new fuel stations so that I can manage multiple stations in the system. | **Given**: An owner is logged in. **When**: They enter station details and submit the form. **Then**: The system registers the new station and updates the list. | 3 | High |
| 5 | **View Fuel Stations** - As an owner, I want to view a list of my fuel stations so that I can manage their operations efficiently. | **Given**: The owner is on the dashboard. **When**: They navigate to the "Manage Stations" section. **Then**: The system displays a list of owned stations. | 2 | High |
| 6 | **Update Fuel Station Details** - As an owner, I want to update my fuel station details so that I can keep the information accurate. | **Given**: The owner is on the "Manage Stations" page. **When**: They update station details and save changes. **Then**: The system reflects the updated details. | 3 | Medium |
| 7 | **Remove Fuel Station** - As an owner, I want to remove a fuel station so that I can stop managing a closed or sold station. | **Given**: The owner is on the "Manage Stations" page. **When**: They click delete and confirm the action. **Then**: The system removes the station from the database. | 3 | Medium |
| 8 | **Register Workers** - As a station owner, I want to register workers so that they can manage station activities. | **Given**: The owner is on the "Manage Workers" page. **When**: They enter the worker's details and submit. **Then**: The system registers the worker. | 3 | High |
| 9 | **Assign Workers to Stations** - As an owner, I want to assign workers to specific stations so that they can operate in their designated locations. | **Given**: An owner is on the worker management page. **When**: They select a worker and assign a station. **Then**: The system updates the worker's assigned station. | 3 | Medium |
| 10 | **Worker Login** - As a worker, I want to log into the system so that I can manage station operations. | **Given**: A worker is on the login page. **When**: They enter valid credentials. **Then**: The system grants access to the worker dashboard. | 2 | High |
| 11 | **Record Fuel Sales** - As a worker, I want to record fuel sales so that the system tracks all transactions. | **Given**: A worker is logged in. **When**: They enter fuel sale details. **Then**: The system stores the transaction. | 3 | High |
| 12 | **Generate Sales Reports** - As an owner, I want to generate sales reports so that I can analyze revenue and performance. | **Given**: The owner is on the reports page. **When**: They select a date range and generate a report. **Then**: The system displays the sales data. | 3 | High |
| 13 | **Track Fuel Inventory** - As a station owner, I want to track fuel levels so that I can ensure continuous supply. | **Given**: The owner is on the dashboard. **When**: They check the station inventory. **Then**: The system displays real-time fuel levels. | 4 | High |
| 14 | **Alert for Low Fuel Levels** - As a station owner, I want to receive alerts for low fuel levels so that I can order refills in time. | **Given**: A station's fuel level drops below a threshold. **When**: The system detects the low level. **Then**: It sends an alert to the owner. | 3 | High |
| 15 | **Order Fuel Supply** - As an owner, I want to order fuel refills so that my stations never run out of stock. | **Given**: The owner is on the fuel supply page. **When**: They place an order for fuel. **Then**: The system processes and confirms the order. | 4 | High |
| 16 | **View Loyalty Points** - As a customer, I want to view my loyalty points so that I can track my rewards. | **Given**: A customer is logged in. **When**: They navigate to the "Loyalty Points" page. **Then**: The system displays their current balance. | 2 | Medium |
| 17 | **Redeem Loyalty Points** - As a customer, I want to redeem my loyalty points so that I can get fuel discounts. | **Given**: A customer has sufficient points. **When**: They redeem them at checkout. **Then**: The system applies the discount. | 3 | Medium |
| 18 | **Customer Fuel Purchase** - As a customer, I want to purchase fuel from the station so that I can refuel my vehicle. | **Given**: A customer is at a station. **When**: They choose the fuel type, quantity, and payment method. **Then**: The system processes the payment and updates inventory. | 4 | High |
| 19 | **Manage Pricing for Fuel Types** - As a station owner, I want to update fuel prices so that I can adjust them based on market conditions. | **Given**: The owner is on the pricing page. **When**: They update the price of a fuel type. **Then**: The system reflects the new pricing in real time. | 3 | High |
| 20 | **Manage Maintenance Requests** - As a station owner, I want to log maintenance requests so that I can ensure station upkeep. | **Given**: A station has an issue. **When**: The owner submits a maintenance request. **Then**: The system records and tracks the request. | 3 | Medium |
| 21 | **View Transaction History -** As a customer, I want to view my transaction history, so that I can track my fuel purchases. | **Given:** A customer is logged in, **When:** They navigate to the transactions section, **Then:** The system displays all past fuel purchases. | 2 | Medium |
| 22 | **Payment via Cash -** As a customer, I want to pay for fuel using cash, so that I can complete my purchase without a card. | **Given:** A customer selects fuel, **When:** They choose cash as the payment method, **Then:** The system registers the payment and updates records. | 3 | High |
| 23 | **Payment via Card -** As a customer, I want to pay for fuel using my debit/credit card, so that I can make transactions easily. | **Given:** A customer selects fuel, **When:** They choose card as the payment method, **Then:** The system processes the payment and updates records. | 3 | High |
| 24 | **Manage Supplier Information -** As an owner, I want to manage my fuel suppliers, So that I can track orders and deliveries. | **Given:** The owner is on the supplier management page, **When:** They add, edit, or remove a supplier, **Then:** The system updates the supplier database. | 3 | Medium |
| 25 | **Generate Maintenance Reports -** As an owner, I want to track station maintenance, so that I can schedule repairs and ensure smooth operations. | **Given:** The owner is on the reports page, **When:** They request a maintenance report, **Then:** The system provides a list of past and upcoming maintenance tasks. | 3 | Medium |
| 26 | **Employee Attendance Tracking -** As a station owner, I want to track employee attendance, so that I can monitor working hours. | **Given:** A worker logs into the system, **When:** They mark attendance for the shift, **Then:** The system records their working hours. | 3 | Medium |
| 27 | **Secure Access -** As a user, I want to have secure access to system configurations, so that I can manage system settings safely. | **Given:** The user is on the login page, **When:** They enter valid admin credentials, **Then:** The system grants access to the admin panel. | 3 | High |
| 28 | **View Worker Profiles -** As an owner, I want to view worker profiles, so I can manage their information and roles. | **Given:** The owner is on the "Manage Workers" page. **When:** They select a worker profile. **Then:** The system displays the worker's details. | 2 | Medium |

# **Team Member Roles**

***Product Owner*** *– Hasnain Akhtar*

**Responsibilities**:

* Define and prioritize user stories for the Petroleum Management System.
* Ensure that features such as fuel request handling, inventory tracking, and payment processing align with stakeholder needs.
* Communicate business requirements clearly to the development team.
* Accept or reject completed user stories based on their adherence to acceptance criteria.
* Continuously refine the backlog based on feedback from stakeholders and the development team.

***Scrum Master*** *– Anas Bin Rashid*

**Responsibilities***:*

* Facilitate Agile ceremonies such as Sprint Planning, Daily Standups, Sprint Review, and Retrospective.
* Identify and remove blockers affecting the development of features like shift management, employee check-in, and real-time alerts.
* Ensure smooth collaboration between Hasnain (Product Owner) and Adan (Developer).
* Track sprint progress and ensure adherence to Agile principles.
* Guide the team in improving workflow efficiency and maintaining high productivity.

***Scrum Team*** *– Adan Malik, Anas Bin Rashid, Hasnain Akhtar*

**Responsibilities**:

* Develop and implement features such as fuel request processing, analytics dashboard, and customer profile management.
* Ensure code quality, security, and efficiency while integrating various modules.
* Conduct unit testing and debugging to ensure smooth system functionality.
* Collaborate with Hasnain (Product Owner) to clarify feature requirements and address feedback.
* Participate in sprint planning and daily standups to track development progress.

# **Sprint Details**

Total number of Sprints are 3, and their details are as follows.

## **Sprint 3**

**1. Sprint Overview**

* Objective : The goal of Sprint 3 is to implement features related to customer fuel purchases, loyalty point redemption, fuel pricing management, maintenance requests, transaction history, payment methods (cash/card), supplier management, maintenance reports, and employee attendance tracking.
* Scope of Work :
  + Redeem Loyalty Points
  + Customer Fuel Purchase
  + Manage Pricing for Fuel Types
  + Manage Maintenance Requests
  + View Transaction History
  + Payment via Cash/Card
  + Manage Supplier Information
  + Generate Maintenance Reports
  + Employee Attendance Tracking
* Duration : 2 weeks

**2. Sprint Backlog**

**Modules Worked On**

1. Customer Features :
   * Redeem Loyalty Points
   * Customer Fuel Purchase
   * View Transaction History
   * Payment via Cash/Card
2. Owner Features :
   * Manage Pricing for Fuel Types
   * Manage Maintenance Requests
   * Manage Supplier Information
   * Generate Maintenance Reports
   * Employee Attendance Tracking

**User Stories and Sub-User Stories**

**User Story 17 – Redeem Loyalty Points**

* Title : Redeem Loyalty Points
* As a customer , I want to redeem my loyalty points so that I can get fuel discounts.
* Context / Note : This story ensures that customers can apply their loyalty points during checkout.
* Task :
  + Create a "Redeem Points" option at checkout.
  + Validate sufficient points before applying the discount.
  + Deduct redeemed points from the customer's account.
  + Update the transaction record with the applied discount.
* Subtask List :
  + [Add "Redeem Points" button to checkout UI]
  + [Validate sufficient points before redemption]
  + [Apply discount and update transaction record]
  + [Deduct points from the customer's account]
* Associated User Stories :
  + User Story 16 : View Loyalty Points
* Rule : Redemption cannot exceed available points.
* ID : US-17

**User Story 18 – Customer Fuel Purchase**

* Title : Customer Fuel Purchase
* As a customer , I want to purchase fuel from the station so that I can refuel my vehicle.
* Context / Note : This story allows customers to select fuel type, quantity, and payment method.
* Task :
  + Create a fuel purchase form.
  + Process payment via cash or card.
  + Deduct purchased fuel from inventory.
  + Generate a transaction receipt.
* Subtask List :
  + [Create fuel purchase form UI]
  + [Process payment and update records]
  + [Deduct fuel from inventory]
  + [Generate transaction receipt]
* Associated User Stories :
  + User Story 22 : Payment via Cash
  + User Story 23 : Payment via Card
* Rule : Inventory levels must update in real time.
* ID : US-18

**User Story 19 – Manage Pricing for Fuel Types**

* Title : Manage Pricing for Fuel Types
* As a station owner , I want to update fuel prices so that I can adjust them based on market conditions.
* Context / Note : This story allows owners to modify fuel prices dynamically.
* Task :
  + Create a pricing management page.
  + Validate new price inputs.
  + Reflect updated prices in real time.
* Subtask List :
  + [Create pricing management UI]
  + [Validate new price inputs]
  + [Update prices in the database]
* Associated User Stories :
  + User Story 18 : Customer Fuel Purchase
* Rule : Price changes must reflect immediately.
* ID : US-19

**User Story 20 – Manage Maintenance Requests**

* Title : Manage Maintenance Requests
* As a station owner , I want to log maintenance requests so that I can ensure station upkeep.
* Context / Note : This story allows owners to submit and track maintenance requests.
* Task :
  + Create a maintenance request form.
  + Log requests with timestamps.
  + Track request status (open/closed).
* Subtask List :
  + [Create maintenance request form UI]
  + [Log requests with timestamps]
  + [Track request status]
* Associated User Stories :
  + User Story 25 : Generate Maintenance Reports
* Rule : Requests must be logged accurately.
* ID : US-20

**User Story 21 – View Transaction History**

* Title : View Transaction History
* As a customer , I want to view my transaction history, so that I can track my fuel purchases.
* Context / Note : This story allows customers to review their past transactions.
* Task :
  + Fetch transaction data from the database.
  + Display transaction details (date, amount, station, payment type).
* Subtask List :
  + [Fetch transaction data from MongoDB]
  + [Display transaction details on the UI]
* Associated User Stories :
  + User Story 18 : Customer Fuel Purchase
* Rule : Data must be restricted to the logged-in customer.
* ID : US-21

**User Story 22 – Payment via Cash**

* Title : Payment via Cash
* As a customer , I want to pay for fuel using cash, so that I can complete my purchase without a card.
* Context / Note : This story handles cash payments and updates records accordingly.
* Task :
  + Add a "Cash" payment option.
  + Register cash payments and update inventory.
* Subtask List :
  + [Add "Cash" payment option to checkout UI]
  + [Register cash payments and update records]
* Associated User Stories :
  + User Story 18 : Customer Fuel Purchase
* Rule : Payments must be recorded accurately.
* ID : US-22

**User Story 23 – Payment via Card**

* Title : Payment via Card
* As a customer , I want to pay for fuel using my debit/credit card, so that I can make transactions easily.
* Context / Note : This story integrates card payment processing.
* Task :
  + Add a "Card" payment option.
  + Validate card details and process payment.
* Subtask List :
  + [Add "Card" payment option to checkout UI]
  + [Validate card details and process payment]
* Associated User Stories :
  + User Story 18 : Customer Fuel Purchase
* Rule : Payment failures must be handled gracefully.
* ID : US-23

**User Story 24: Manage Supplier Information**

**Title :**

Manage Supplier Information

**As a station owner ,**

I want to manage my fuel suppliers so that I can track orders and deliveries effectively.

**Context / Note :**

This story allows station owners to add, edit, or remove supplier details. It ensures that supplier information is accurate and up-to-date for managing fuel orders and deliveries.

**Task :**

* Create a form for entering supplier details (e.g., name, contact information, delivery terms).
* Validate input fields before saving changes.
* Store supplier data in the database.
* Prevent deletion of suppliers linked to active orders.

**Subtask List :**

* [Create supplier management form UI]
* [Validate input fields (name, contact, delivery terms)]
* [Add functionality to create, edit, and delete suppliers]
* [Prevent deletion of suppliers with active orders]
* [Log all supplier modifications for auditing purposes]

**Associated User Stories :**

* User Story 15 : Order Fuel Supply

**Rules :**

* Suppliers cannot be deleted if they are linked to active orders.
* All modifications to supplier data must be logged for auditing.

**ID : US-24**

**User Story 25: Generate Maintenance Reports**

**Title :**

Generate Maintenance Reports

**As a station owner ,**

I want to track station maintenance so that I can schedule repairs and ensure smooth operations.

**Context / Note :**

This story enables owners to generate reports for past and upcoming maintenance tasks. The reports help in planning and ensuring timely repairs to avoid operational disruptions.

**Task :**

* Create a form for selecting filters (e.g., station, time period).
* Fetch maintenance data from the database based on selected filters.
* Display aggregated data (e.g., task description, date, status) in a table or chart.
* Allow exporting reports in standard formats like PDF or CSV.

**Subtask List :**

* [Create maintenance report generation UI]
* [Add filters for station and time period]
* [Fetch and display aggregated maintenance data]
* [Implement export functionality for PDF/CSV]

**Associated User Stories :**

* User Story 20 : Manage Maintenance Requests

**Rules :**

* Reports must include all maintenance tasks within the selected filters.
* Exported reports should be formatted for readability.

**ID : US-25**

**User Story 26: Employee Attendance Tracking**

**Title :**

Employee Attendance Tracking

**As a station owner ,**

I want to track employee attendance so that I can monitor working hours and ensure accountability.

**Context / Note :**

This story allows workers to mark their attendance at the start of their shifts. The system records their login time and calculates working hours for payroll and performance tracking.

**Task :**

* Add an "Attendance" feature to the worker dashboard.
* Record login timestamps when workers mark attendance.
* Calculate total working hours for each shift.
* Provide attendance history to the owner for review.

**Subtask List :**

* [Add "Mark Attendance" button to the worker dashboard]
* [Record login timestamps in the database]
* [Calculate and display total working hours]
* [Provide attendance history to the owner]

**Associated User Stories :**

* User Story 8 : Register Workers
* User Story 9 : Assign Workers to Stations

**Rules :**

* Attendance marking should only be allowed during valid user sessions.
* Owners should have access to detailed attendance logs for each worker.

**ID : US-26**

**3. Software Testing**

**1. Supplier Details Page – Test Design**

**Fields:**

* **Supplier Name**
* **Company Name**
* **Contact Number**
* **Email**
* **Fuel Type (Dropdown: Petrol, Diesel, etc.)**
* **Supply Capacity (Liters)**

**Field-Wise Analysis and Test Design (Supplier Details Page)**

| **Field** | **Input Criteria** | **Valid Equivalence Class** | **Invalid Equivalence Class** | **Boundary Values** |
| --- | --- | --- | --- | --- |
| **Supplier Name** | **Alphabetic, 3–50 characters** | **John Smith** | **<3, >50, digits/special chars** | **2 (invalid), 3 (valid), 50 (valid), 51 (invalid)** |
| **Company Name** | **Alphanumeric, 3–100 characters** | **FuelCorp Pvt Ltd** | **<3, >100, symbols only** | **2 (invalid), 3 (valid), 100 (valid), 101 (invalid)** |
| **Contact Number** | **Exactly 10 digits, numeric only** | **9876543210** | **<10, >10 digits, non-numeric** | **9 (invalid), 10 (valid), 11 (invalid)** |
| **Email** | **Valid email format** | **supplier@mail.com** | **Missing @, domain errors** | **Structure-based** |
| **Fuel Type** | **Predefined values only** | **Petrol, Diesel** | **Any non-listed string like Gasoline, blank** | **Not applicable** |
| **Supply Capacity** | **Integer between 1000 and 500000** | **15000** | **<1000, >500000, negative** | **999 (invalid), 1000 (valid), 500000 (valid), 500001 (invalid)** |

**Weak Equivalence Class Testing (Supplier Details Page)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Supplier Name** | **SPD-NAME-WEAK-01** | **Alfa** | **Accepted** | **Valid** |
| **Supplier Name** | **SPD-NAME-WEAK-02** | **J@** | **Rejected** | **Too short & invalid characters** |
| **Company Name** | **SPD-COMP-WEAK-01** | **PowerFuel Ltd** | **Accepted** | **Valid** |
| **Company Name** | **SPD-COMP-WEAK-02** | **!!** | **Rejected** | **Invalid input** |
| **Contact Number** | **SPD-CON-WEAK-01** | **9988776655** | **Accepted** | **Valid 10-digit number** |
| **Contact Number** | **SPD-CON-WEAK-02** | **1234abcd** | **Rejected** | **Alphanumeric** |
| **Email** | **SPD-EMAIL-WEAK-01** | **valid@supply.com** | **Accepted** | **Valid format** |
| **Email** | **SPD-EMAIL-WEAK-02** | **invalid.com** | **Rejected** | **No @** |
| **Fuel Type** | **SPD-FUEL-WEAK-01** | **Diesel** | **Accepted** | **Dropdown value** |
| **Fuel Type** | **SPD-FUEL-WEAK-02** | **LPG** | **Rejected** | **Not in allowed options** |
| **Supply Capacity** | **SPD-CAP-WEAK-01** | **250000** | **Accepted** | **Valid** |
| **Supply Capacity** | **SPD-CAP-WEAK-02** | **-500** | **Rejected** | **Negative** |

**Strong Equivalence Class Testing (Supplier Details Page)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Supplier Name** | **SPD-STRONG-01** | **FuelMaster** | **Accepted** | **Valid** |
| **Supplier Name** | **SPD-STRONG-02** | **AB** | **Rejected** | **Too short** |
| **Supplier Name** | **SPD-STRONG-03** | **A x 51** | **Rejected** | **Too long** |
| **Company Name** | **SPD-STRONG-04** | **FuelWorks Ltd** | **Accepted** | **Valid** |
| **Company Name** | **SPD-STRONG-05** | **FW** | **Rejected** | **<3 chars** |
| **Company Name** | **SPD-STRONG-06** | **A x 101** | **Rejected** | **>100 chars** |
| **Contact Number** | **SPD-STRONG-07** | **9999999999** | **Accepted** | **Valid** |
| **Contact Number** | **SPD-STRONG-08** | **12345678** | **Rejected** | **Too short** |
| **Email** | **SPD-STRONG-09** | **a@b.c** | **Accepted** | **Valid minimal email** |
| **Email** | **SPD-STRONG-10** | **@domain.com** | **Rejected** | **Missing local part** |
| **Fuel Type** | **SPD-STRONG-11** | **Petrol** | **Accepted** | **Valid selection** |
| **Fuel Type** | **SPD-STRONG-12** | **PremiumFuel** | **Rejected** | **Not in options** |
| **Supply Capacity** | **SPD-STRONG-13** | **999** | **Rejected** | **Below range** |
| **Supply Capacity** | **SPD-STRONG-14** | **1000** | **Accepted** | **Min valid** |
| **Supply Capacity** | **SPD-STRONG-15** | **500001** | **Rejected** | **Above max** |

**Boundary Value Analysis (Supplier Details Page)**

| **Field** | **Test Case ID** | **Input Length/Value** | **Test Input** | **Expected Result** |
| --- | --- | --- | --- | --- |
| **Supplier Name** | **BVA-SPD-NAME-01** | **2** | **AB** | **Rejected** |
| **Supplier Name** | **BVA-SPD-NAME-02** | **3** | **ABC** | **Accepted** |
| **Supplier Name** | **BVA-SPD-NAME-03** | **50** | **A x 50** | **Accepted** |
| **Supplier Name** | **BVA-SPD-NAME-04** | **51** | **A x 51** | **Rejected** |
| **Supply Capacity** | **BVA-SPD-CAP-01** | **999** | **999** | **Rejected** |
| **Supply Capacity** | **BVA-SPD-CAP-02** | **1000** | **1000** | **Accepted** |
| **Supply Capacity** | **BVA-SPD-CAP-03** | **500000** | **500000** | **Accepted** |
| **Supply Capacity** | **BVA-SPD-CAP-04** | **500001** | **500001** | **Rejected** |

**Summary (Supplier Details Page)**

* **Weak EC Tests: 12**
* **Strong EC Tests: 15**
* **BVA Tests: 8**
* **Total Test Cases: 35**

**2. Reports Generation Page – Test Design**

**Fields/Filters:**

* **Date Range (Start Date, End Date)**
* **Report Type (Dropdown: Sales, Supply, Expense, etc.)**
* **Station Filter (Dropdown or search by station name)**
* **Export Format (Radio or dropdown: PDF, Excel, CSV)**

**Field-Wise Analysis and Test Design (Reports Page)**

| **Field** | **Input Criteria** | **Valid Equivalence Class** | **Invalid Equivalence Class** | **Boundary Values** |
| --- | --- | --- | --- | --- |
| **Start Date** | **Must be a valid past date** | **2023-01-01** | **Future date, invalid format** | **N/A** |
| **End Date** | **Valid date ≥ start date** | **2023-01-31 (after start)** | **Before start date, invalid format** | **N/A** |
| **Report Type** | **Predefined values only** | **Sales, Supply** | **Earnings, blank** | **N/A** |
| **Station Filter** | **Alphanumeric, 3–50 characters** | **Main Station** | **<3 or >50, symbols only** | **2 (invalid), 3 (valid), 50 (valid), 51 (invalid)** |
| **Export Format** | **Must be PDF/Excel/CSV only** | **PDF, Excel** | **TXT, DOCX, blank** | **N/A** |

**Weak Equivalence Class Testing (Reports Page)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Start Date** | **REP-DATE-WEAK-01** | **2024-01-01** | **Accepted** | **Valid past date** |
| **Start Date** | **REP-DATE-WEAK-02** | **2026-01-01** | **Rejected** | **Future date** |
| **End Date** | **REP-DATE-WEAK-03** | **2024-02-01** | **Accepted** | **Valid end date after start** |
| **End Date** | **REP-DATE-WEAK-04** | **2023-12-01** | **Rejected** | **Before start date** |
| **Report Type** | **REP-TYPE-WEAK-01** | **Sales** | **Accepted** | **Predefined option** |
| **Report Type** | **REP-TYPE-WEAK-02** | **Profit** | **Rejected** | **Not in options** |
| **Station Filter** | **REP-STAT-WEAK-01** | **Fuel Point A** | **Accepted** | **Valid station name** |
| **Station Filter** | **REP-STAT-WEAK-02** | **X** | **Rejected** | **Too short** |
| **Export Format** | **REP-EXP-WEAK-01** | **PDF** | **Accepted** | **Valid format** |
| **Export Format** | **REP-EXP-WEAK-02** | **TXT** | **Rejected** | **Invalid format** |

**Strong Equivalence Class Testing (Reports Page)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Start Date** | **REP-STRONG-01** | **2023-10-10** | **Accepted** | **Valid date** |
| **Start Date** | **REP-STRONG-02** | **2027-01-01** | **Rejected** | **Future date** |
| **End Date** | **REP-STRONG-03** | **Same as start date** | **Accepted** | **Boundary case** |
| **End Date** | **REP-STRONG-04** | **Before start date** | **Rejected** | **Invalid range** |
| **Report Type** | **REP-STRONG-05** | **Supply** | **Accepted** | **Valid** |
| **Report Type** | **REP-STRONG-06** | **UnknownType** | **Rejected** | **Invalid** |
| **Station Filter** | **REP-STRONG-07** | **MainFuelDepot** | **Accepted** | **Valid station** |
| **Station Filter** | **REP-STRONG-08** | **!!** | **Rejected** | **Symbols only** |
| **Export Format** | **REP-STRONG-09** | **Excel** | **Accepted** | **Valid format** |
| **Export Format** | **REP-STRONG-10** | **DOC** | **Rejected** | **Not allowed** |

**Boundary Value Analysis (Reports Page)**

| **Field** | **Test Case ID** | **Input Length/Value** | **Test Input** | **Expected Result** |
| --- | --- | --- | --- | --- |
| **Station Filter** | **BVA-REP-STAT-01** | **2** | **AB** | **Rejected** |
| **Station Filter** | **BVA-REP-STAT-02** | **3** | **ABC** | **Accepted** |
| **Station Filter** | **BVA-REP-STAT-03** | **50** | **A x 50** | **Accepted** |
| **Station Filter** | **BVA-REP-STAT-04** | **51** | **A x 51** | **Rejected** |

**Summary (Reports Page)**

* **Weak EC Tests: 10**
* **Strong EC Tests: 10**
* **BVA Tests: 4**
* **Total Test Cases: 24**

**4. Meetings Conducted**

**Sprint Planning Meeting**

Overview :

a. Date Time Stamp : 28 March 2025  
b. Attendees :

| **Present** | **Absent** |
| --- | --- |
| Anas Bin Rashid | - |
| Adan Malik | - |
| Hasnain Akhtar | - |

Topics Discussed and Conclusions Reached :

* Defined sprint goals: Implement customer fuel purchase, loyalty point redemption, payment methods, supplier management, maintenance reports, and employee attendance tracking.
* Prioritized user stories: US-17, US-18, US-22, US-23, US-24, US-25, US-26.
* Assigned tasks to team members.

Conclusions :

* Sprint backlog finalized.
* Daily scrum schedule set for 9:00 AM daily.

Overall Duration : 1 hour

**Daily Scrum Meetings**

The meeting focused on different topics each day and they are reflected in the burndown chart below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Day | Story Points Remaining (Planned) | | Story Points Remaining (Actual) | | Tasks Completed | Notes | |
| Day 1 | 40 | | 40 | | 0 | Sprint starts; no tasks done. | |
| Day 2 | 36 | | 38 | | 2 | Minor delays in setting up environments. | |
| Day 3 | 32 | | 35 | | 3 | UI design for fuel station management. | |
| Day 4 | 28 | | 32 | | 3 | Backend logic for payment methods. | |
| Day 5 | 24 | | 29 | | 3 | Loyalty points UI completed. | |
| Day 6 | 20 | | 26 | | 3 | Bug fixes during testing. | |
| Day 7 | 16 | | 23 | | 3 | Supplier management feature added. | |
| Day 8 | 12 | | 20 | | 3 | Testing ongoing for maintenance. | |
| Day 9 | 8 | | 15 | | 3 | Leftover tasks identified. | |
| Day 10 | 0 | | 8 | | 5 | Sprint ends; some tasks incomplete. | |
|  | |  | |  | | |

A graph with blue and orange dots

AI-generated content may be incorrect.

**Sprint Review Meeting**

Overview :

a. Date Time Stamp : 8 April 2025  
b. Attendees :

| **Present** | **Absent** |
| --- | --- |
| Anas Bin Rashid | - |
| Adan Malik | - |
| Hasnain Akhtar | - |

Topics Discussed and Conclusions Reached :

* Reviewed completed user stories: US-17, US-18, US-22, US-23, US-24, US-25, US-26.
* Demonstrated working features: Loyalty point redemption, customer fuel purchase, payment methods, supplier management, maintenance reports, and employee attendance tracking.
* Identified minor bugs during testing.

Conclusions :

* All high-priority user stories completed successfully.
* Minor bugs identified during testing.

Overall Duration : 1 hour

**Sprint Retrospective Meeting**

Retrospective Meeting : Iteration/Sprint #2  
Overview :

c. Date Time Stamp : 10 April 2025

d. Attendees :

| **Present** | **Absent** |
| --- | --- |
| Anas Bin Rashid | - |
| Adan Malik | - |
| Hasnain Akhtar | - |

Retrospective Content :

| **Successes** | **To Improve** | **Actions and Suggestions** |
| --- | --- | --- |
| Completed all high-priority user stories. | Minor delays in integrating password reset functionality. | Allocate more time for testing in future sprints. |
| Strong collaboration among team members. | Limited focus on edge-case testing. | Involve QA team earlier for comprehensive testing. |
| Well-documented codebase. | Communication gaps during task handovers. | Use Trello for better task tracking and communication. |

**5. Definition of Done (DoD)**

* All unit tests are written.
* The written code is documented.
* Code reviewed and approved by peers.
* Features tested in staging environment.
* No critical bugs reported.

**6. Bug Report**

**Bug Report 1**

Name : Incorrect Loyalty Point Deduction During Redemption

Expected Behavior : The system should deduct the exact number of loyalty points used for a discount.

Actual Behavior : The system deducts an incorrect quantity of loyalty points, leading to discrepancies in the balance.

Steps to Reproduce :

Log in as a customer and navigate to the "Redeem Loyalty Points" page.

Redeem points for a discount at checkout.

Check the loyalty point balance after the transaction.

Context / Environment : This issue occurs in the staging environment when testing the "Redeem Loyalty Points" feature (User Story #17).

Possible Solutions :

Review backend logic for loyalty point deduction.

Add validation to prevent duplicate API calls during submission.

Definition of Done (DoD) :

All unit tests are written to verify correct loyalty point deduction.

The written code is documented.

The bug is resolved and tested in the staging environment.

Date Time Stamp : 2025-04-08 11:00 AM

Reported By : Adan Malik

**Bug Report 2**

Name : Employee Attendance Not Logged Correctly

Expected Behavior : The system should log the timestamp and calculate working hours accurately when a worker marks attendance.

Actual Behavior : The system logs incorrect timestamps or fails to calculate working hours correctly.

Steps to Reproduce :

Log in as a worker and mark attendance.

Check the attendance logs for the recorded timestamp and calculated working hours.

Context / Environment : This issue was identified during testing in the development environment. It is related to User Story #26.

Possible Solutions :

Debug the logic for logging timestamps and calculating working hours.

Ensure synchronization between frontend and backend timestamps.

Definition of Done (DoD) :

All unit tests are written to validate attendance logging functionality.

The written code is documented.

The bug is resolved and verified in both development and staging environments.

Date Time Stamp : 2025-04-09 03:30 PM

Reported By : Anas Bin Rashid

**7. Output**

**Tested Code Implementation**

***Katalon Studio Code***

*import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject*

*import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI*

*// Step 1: Open the browser and navigate to the application URL*

*WebUI.openBrowser('')*

*WebUI.navigateToUrl('http://localhost:3000/login')*

*// Step 2: Log in as a customer*

*WebUI.setText(findTestObject('Object Repository/input\_Email'), 'customer@example.com')*

*WebUI.setText(findTestObject('Object Repository/input\_Password'), 'password123')*

*WebUI.click(findTestObject('Object Repository/button\_Login'))*

*// Step 3: Navigate to the "Redeem Loyalty Points" page*

*WebUI.click(findTestObject('Object Repository/link\_Redeem\_Loyalty\_Points'))*

*// Step 4: Redeem loyalty points*

*WebUI.setText(findTestObject('Object Repository/input\_Loyalty\_Points'), '100')*

*WebUI.click(findTestObject('Object Repository/button\_Redeem'))*

*// Step 5: Verify that the loyalty points were deducted*

*WebUI.verifyTextPresent('Loyalty points redeemed successfully.', false)*

*// Step 6: Close the browser*

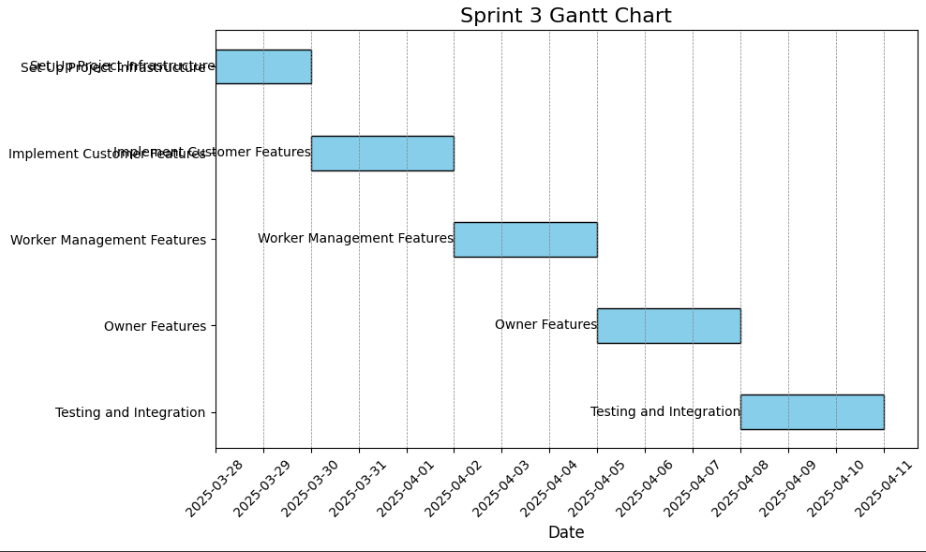
*WebUI.closeBrowser()*

**8. Project Planner and Gantt Chart**

**Project Planner**

* Task 1 : Set Up Project Infrastructure (Day 1–2)
  + Initialize backend project using Node.js and Express.js.
  + Install necessary dependencies (e.g., bcrypt, JWT, MongoDB driver).
  + Configure environment variables for security.
* Task 2 : Implement Customer Features (Day 3–5)
  + Redeem loyalty points (US-17).
  + Customer fuel purchase (US-18).
  + Payment via cash/card (US-22, US-23).
* Task 3 : Worker Management Features (Day 6–8)
  + Employee attendance tracking (US-26).
* Task 4 : Owner Features (Day 9–12)
  + Manage supplier information (US-24).
  + Generate maintenance reports (US-25).
* Task 5 : Testing and Integration (Day 13–15)
  + Conduct thorough testing of all implemented features.
  + Fix any bugs or issues identified during testing.

**Gantt Chart**



**9. Trello Screenshots**

**Start of Sprint**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Middle of Sprint**

**Screens screenshot of a computer screen

AI-generated content may be incorrect.**

**End of Sprint**

**Screens screenshot of a computer screen

AI-generated content may be incorrect.**

**Scrum Master Activities and Leanings**

**A screenshot of a computer program

AI-generated content may be incorrect.**

Topics Discussed and Conclusions Reached :

* Reviewed completed user stories: US-17, US-18, US-22, US-23, US-24, US-25, US-26.
* Demonstrated working features: Loyalty point redemption, customer fuel purchase, payment methods, supplier management, maintenance reports, and employee attendance tracking.
* Identified minor bugs during testing.

Conclusions :

* All high-priority user stories completed successfully.
* Minor bugs identified during testing.

**10. Burndown Charts**

