**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 2**

**1. Sprint Overview**

* Objective : The goal of Sprint 2 was to implement core functionalities related to fuel station management, worker registration, sales reporting, inventory tracking, and customer loyalty points.
* Scope of Work :
  + Add, view, update, and remove fuel stations.
  + Register workers and manage their details.
  + Generate sales reports and track fuel inventory.
  + Implement low fuel alerts and order fuel supply.
  + Allow customers to view loyalty points.
* Duration : 2 weeks

**2. Sprint Backlog**

**Modules Worked On**

1. Fuel Station Management : Add, view, update, and remove fuel stations.
2. Worker Registration : Register workers and assign them to stations.
3. Sales Reporting : Generate sales reports based on date ranges.
4. Inventory Tracking : Track real-time fuel levels and send low fuel alerts.
5. Loyalty Points : Display loyalty points for customers.

**User Stories and Sub-User Stories**

**User Story 4 – Add Fuel Station**

Title : Add Fuel Station  
As an owner , I want to add new fuel stations so that I can manage multiple stations in the system.  
Context / Note : This story ensures that owners can register new stations by providing details like name, location, and capacity.  
This story is related to Sprint #2.

Task :

* Create a form for entering station details.
* Validate input fields before submission.
* Store station data in the database.
* Update the list of owned stations.

Subtask List :

* [Create station registration form UI]
* [Validate input fields (name, location, capacity)]
* [Store station data in MongoDB]
* [Update the station list dynamically]

Associated User Stories :

* User Story 5 : View Fuel Stations
* Rule : A station cannot be registered with duplicate details.
* ID : US-04

**User Story 5 – View Fuel Stations**

Title : View Fuel Stations  
As an owner , I want to view a list of my fuel stations so that I can manage their operations efficiently.  
Context / Note : This story ensures that owners can see all their stations in one centralized list.  
This story is related to Sprint #2.

Task :

* Fetch station data from the database.
* Display the list of owned stations on the dashboard.

Subtask List :

* [Fetch station data from MongoDB]
* [Display stations in a tabular format]
* [Refresh the list dynamically after updates]

Associated User Stories :

* User Story 6 : Update Fuel Station Details
* Rule : Only logged-in owners can view their stations.
* ID : US-05

**User Story 6 – Update Fuel Station Details**

Title : Update Fuel Station Details  
As an owner , I want to update my fuel station details so that I can keep the information accurate.  
Context / Note : This story allows owners to edit station details like contact information or capacity.  
This story is related to Sprint #2.

Task :

* Create an edit form for station details.
* Validate input fields before saving changes.
* Reflect updated details in the system.

Subtask List :

* [Create edit form UI]
* [Validate input fields]
* [Update station data in MongoDB]

Associated User Stories :

* User Story 7 : Remove Fuel Station
* Rule : Owners must confirm changes before saving.
* ID : US-06

**User Story 7 – Remove Fuel Station**

Title : Remove Fuel Station  
As an owner , I want to remove a fuel station so that I can stop managing a closed or sold station.  
Context / Note : This story allows owners to delete stations they no longer manage.  
This story is related to Sprint #2.

Task :

* Add a "Delete" button for each station.
* Prompt confirmation before deletion.
* Remove the station from the database.

Subtask List :

* [Add delete button to station list]
* [Prompt confirmation modal]
* [Remove station from MongoDB]

Associated User Stories :

* User Story 4 : Add Fuel Station
* Rule : Deleted stations cannot be recovered.
* ID : US-07

**User Story 8 – Register Workers**

Title : Register Workers  
As a station owner , I want to register workers so that they can manage station activities.  
Context / Note : This story ensures that owners can add workers to the system.  
This story is related to Sprint #2.

Task :

* Create a registration form for workers.
* Validate input fields (name, email, role).
* Store worker data in the database.

Subtask List :

* [Create worker registration form UI]
* [Validate input fields]
* [Store worker data in MongoDB]

Associated User Stories :

* User Story 9 : Assign Workers to Stations
* Rule : Each worker must have a unique email.
* ID : US-08

**User Story 12 – Generate Sales Reports**

Title : Generate Sales Reports  
As an owner , I want to generate sales reports so that I can analyze revenue and performance.  
Context / Note : This story enables owners to filter and view sales data by date range, station, or fuel type. The system will also allow exporting reports in standard formats like PDF or CSV.  
This story is related to Sprint #2.

Task :

* Create a form for selecting date ranges and filters (e.g., station, fuel type).
* Fetch sales data from the database based on selected filters.
* Display aggregated data (e.g., total revenue, quantity sold) in a table or chart.
* Allow exporting reports in PDF/CSV format.

Subtask List :

* [Create sales report generation UI]
* [Add dropdowns for filtering by station and fuel type]
* [Fetch and display aggregated sales data]
* [Implement export functionality for PDF/CSV]

Associated User Stories :

* User Story 11 : Record Fuel Sales
* Rule : Reports must include all transactions within the selected date range.
* ID : US-12

**User Story 13 – Track Fuel Inventory**

Title : Track Fuel Inventory  
As a station owner , I want to track fuel levels so that I can ensure continuous supply.  
Context / Note : This story ensures real-time visibility of fuel inventory levels at each station. It also includes automatic updates when fuel is sold or restocked.  
This story is related to Sprint #2.

Task :

* Display current fuel levels for each station on the dashboard.
* Deduct fuel from inventory when a sale is recorded.
* Update inventory when fuel is restocked.
* Log inventory changes for future reference.

Subtask List :

* [Display real-time fuel levels on the dashboard]
* [Automatically deduct fuel during sales]
* [Update inventory after restocking]
* [Log inventory changes in the database]

Associated User Stories :

* User Story 14 : Alert for Low Fuel Levels
* Rule : Inventory levels must update in real time.
* ID : US-13

**User Story 14 – Alert for Low Fuel Levels**

Title : 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.  
Context / Note : This story ensures that the system notifies owners when fuel levels drop below a predefined threshold. Owners can configure alert thresholds for each station.  
This story is related to Sprint #2.

Task :

* Monitor fuel levels against predefined thresholds.
* Trigger alerts when levels fall below the threshold.
* Notify the respective station owner via email or SMS.
* Allow owners to configure alert thresholds.

Subtask List :

* [Monitor fuel levels against thresholds]
* [Trigger alerts when thresholds are breached]
* [Send notifications to station owners]
* [Allow configuration of alert thresholds]

Associated User Stories :

* User Story 15 : Order Fuel Supply
* Rule : Alerts must be sent in real time to prevent stockouts.
* ID : US-14

**User Story 15 – Order Fuel Supply**

Title : Order Fuel Supply  
As an owner , I want to order fuel refills so that my stations never run out of stock.  
Context / Note : This story allows owners to place refill orders for specific fuel types and quantities. The system will log the order and notify suppliers.  
This story is related to Sprint #2.

Task :

* Create a form for placing fuel refill orders.
* Validate fuel type and quantity before submission.
* Log the order in the database and notify the supplier.
* Display the status of each order (e.g., pending, confirmed, delivered).

Subtask List :

* [Create fuel order placement UI]
* [Validate fuel type and quantity]
* [Log orders in the database]
* [Notify suppliers via email or API integration]

Associated User Stories :

* User Story 14 : Alert for Low Fuel Levels
* Rule : Orders must be validated before submission.
* ID : US-15

**User Story 16 – View Loyalty Points**

Title : View Loyalty Points  
As a customer , I want to view my loyalty points so that I can track my rewards.  
Context / Note : This story allows customers to view their loyalty point balance after logging in. The system fetches and displays the correct point total from the database.  
This story is related to Sprint #2.

Task :

* Create a "Loyalty Points" page accessible only to logged-in customers.
* Fetch and display the customer's loyalty point balance.
* Restrict loyalty point visibility to the logged-in customer.

Subtask List :

* [Create Loyalty Points page UI]
* [Fetch loyalty points from the database]
* [Display loyalty points on the page]
* [Restrict access to logged-in customers]

Associated User Stories :

* User Story 17 : Redeem Loyalty Points
* Rule : Loyalty points must be securely fetched and displayed.
* ID : US-16

**3. Software Testing**

**1. Station Details Entry Page: Test Design**

**Fields:**

* **Station Name**
* **Station Address**
* **Contact Number**
* **Fuel Capacity (Liters)**
* **License Number**

**Field-Wise Analysis and Test Design (Station Details Entry)**

| **Field** | **Input Criteria** | **Valid Equivalence Class** | **Invalid Equivalence Class** | **Boundary Values** |
| --- | --- | --- | --- | --- |
| **Station Name** | **Alphabetic/Alphanumeric, 3–50 characters** | **Petro Max Station** | **Too short/long, symbols-only** | **2 (invalid), 3 (valid), 50 (valid), 51 (invalid)** |
| **Station Address** | **10–100 characters, alphanumeric + special** | **123 Main Rd, NY** | **Too short, only symbols** | **9 (invalid), 10 (valid), 100 (valid), 101 (invalid)** |
| **Contact Number** | **10-digit numeric** | **9876543210** | **<10, >10 digits, non-numeric** | **9 (invalid), 10 (valid), 11 (invalid)** |
| **Fuel Capacity** | **Integer between 1000 and 100000** | **50000** | **Negative, 0, non-numeric, >100000** | **999 (invalid), 1000 (valid), 100000 (valid), 100001 (invalid)** |
| **License Number** | **Alphanumeric, 6–12 characters** | **LIC12345** | **<6, >12, special chars** | **5 (invalid), 6 (valid), 12 (valid), 13 (invalid)** |

**Weak Equivalence Class Testing (Station Details Entry)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Station Name** | **STN-ECT-WEAK-01** | **MaxFuel** | **Accepted** | **Valid** |
| **Station Name** | **STN-ECT-WEAK-02** | **!!** | **Rejected** | **Too short, only symbols** |
| **Station Address** | **STN-ADR-WEAK-01** | **12 Oak Avenue** | **Accepted** | **Valid** |
| **Station Address** | **STN-ADR-WEAK-02** | **Short** | **Rejected** | **<10 characters** |
| **Contact Number** | **STN-CON-WEAK-01** | **9876543210** | **Accepted** | **Valid contact** |
| **Contact Number** | **STN-CON-WEAK-02** | **12345ABCD** | **Rejected** | **Non-numeric** |
| **Fuel Capacity** | **STN-CAP-WEAK-01** | **10000** | **Accepted** | **Within range** |
| **Fuel Capacity** | **STN-CAP-WEAK-02** | **-500** | **Rejected** | **Negative number** |
| **License Number** | **STN-LIC-WEAK-01** | **AB1234** | **Accepted** | **Valid** |
| **License Number** | **STN-LIC-WEAK-02** | **A@1** | **Rejected** | **Too short, invalid characters** |

**Strong Equivalence Class Testing (Station Details Entry)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Station Name** | **STN-STRONG-01** | **Petro Point** | **Accepted** | **Valid name** |
| **Station Name** | **STN-STRONG-02** | **P** | **Rejected** | **Too short** |
| **Station Name** | **STN-STRONG-03** | **A repeated 51 times** | **Rejected** | **Too long** |
| **Station Name** | **STN-STRONG-04** | **!@#$%^** | **Rejected** | **Invalid characters** |
| **Station Address** | **STN-STRONG-05** | **15 Main St, Sector 22, NY** | **Accepted** | **Valid** |
| **Station Address** | **STN-STRONG-06** | **TooShort** | **Rejected** | **Too short** |
| **Station Address** | **STN-STRONG-07** | **A repeated 101 times** | **Rejected** | **Too long** |
| **Contact Number** | **STN-STRONG-08** | **123456789** | **Rejected** | **9 digits** |
| **Contact Number** | **STN-STRONG-09** | **12345678901** | **Rejected** | **11 digits** |
| **Contact Number** | **STN-STRONG-10** | **0123456789** | **Accepted** | **Valid 10-digit number** |
| **Fuel Capacity** | **STN-STRONG-11** | **999** | **Rejected** | **Just below min** |
| **Fuel Capacity** | **STN-STRONG-12** | **1000** | **Accepted** | **Min valid** |
| **Fuel Capacity** | **STN-STRONG-13** | **100001** | **Rejected** | **Above max** |
| **License Number** | **STN-STRONG-14** | **LIC001** | **Accepted** | **Valid** |
| **License Number** | **STN-STRONG-15** | **LIC** | **Rejected** | **Too short** |
| **License Number** | **STN-STRONG-16** | **LICNUMBER1234X** | **Rejected** | **Too long** |

**Boundary Value Analysis (Station Details Entry)**

| **Field** | **Test Case ID** | **Input Length/Value** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- | --- |
| **Station Name** | **BVA-STN-NAME-01** | **2 (invalid)** | **AB** | **Rejected** | **Below min** |
| **Station Name** | **BVA-STN-NAME-02** | **3 (valid)** | **ABC** | **Accepted** | **Lower bound** |
| **Station Name** | **BVA-STN-NAME-03** | **50 (valid)** | **A x 50** | **Accepted** | **Upper bound** |
| **Station Name** | **BVA-STN-NAME-04** | **51 (invalid)** | **A x 51** | **Rejected** | **Above max** |
| **Fuel Capacity** | **BVA-STN-CAP-01** | **999** | **999** | **Rejected** | **Just below valid range** |
| **Fuel Capacity** | **BVA-STN-CAP-02** | **1000** | **1000** | **Accepted** | **Lower bound** |
| **Fuel Capacity** | **BVA-STN-CAP-03** | **100000** | **100000** | **Accepted** | **Upper bound** |
| **Fuel Capacity** | **BVA-STN-CAP-04** | **100001** | **100001** | **Rejected** | **Above valid range** |

**Summary (Station Details Entry)**

* **Weak EC Tests: 10**
* **Strong EC Tests: 16**
* **BVA Tests: 8**
* **Total Test Cases: 34**

**2. Edit Profile Page: Test Design**

**Fields:**

* **Name**
* **Username**
* **Email**
* **Phone Number**
* **Bio (Optional, 0–160 characters)**

**Field-Wise Analysis and Test Design (Edit Profile)**

| **Field** | **Input Criteria** | **Valid Equivalence Class** | **Invalid Equivalence Class** | **Boundary Values** |
| --- | --- | --- | --- | --- |
| **Name** | **Alphabetic only, 1–50 characters** | **John Smith** | **Digits/symbols, empty** | **0 (invalid), 1 (valid), 50 (valid), 51 (invalid)** |
| **Username** | **Alphanumeric, 4–15 characters** | **john123, userX** | **<4, >15, special characters** | **3 (invalid), 4 (valid), 15 (valid), 16 (invalid)** |
| **Email** | **Valid email format** | **john@example.com** | **Missing @, domain errors** | **Structure-based** |
| **Phone Number** | **Numeric string of exactly 10 digits** | **9876543210** | **<10 or >10 digits, letters** | **9 (invalid), 10 (valid), 11 (invalid)** |
| **Bio** | **Optional, 0–160 characters** | **Empty, or up to 160 chars** | **>160 characters** | **0 (valid), 160 (valid), 161 (invalid)** |

**Weak Equivalence Class Testing (Edit Profile)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Name** | **EP-NAME-WEAK-01** | **Alice** | **Accepted** | **Valid name** |
| **Name** | **EP-NAME-WEAK-02** | **Alice123** | **Rejected** | **Contains digits** |
| **Username** | **EP-USER-WEAK-01** | **john\_doe** | **Rejected** | **Contains underscore** |
| **Username** | **EP-USER-WEAK-02** | **user99** | **Accepted** | **Valid** |
| **Email** | **EP-EMAIL-WEAK-01** | **new@mail.com** | **Accepted** | **Valid email** |
| **Email** | **EP-EMAIL-WEAK-02** | **nomail.com** | **Rejected** | **Invalid format** |
| **Phone Number** | **EP-PHONE-WEAK-01** | **9998887776** | **Accepted** | **Valid 10 digits** |
| **Phone Number** | **EP-PHONE-WEAK-02** | **1234abcd** | **Rejected** | **Alphanumeric** |
| **Bio** | **EP-BIO-WEAK-01** | **Empty string** | **Accepted** | **Optional** |
| **Bio** | **EP-BIO-WEAK-02** | **A x 200** | **Rejected** | **Exceeds 160** |

**Strong Equivalence Class Testing (Edit Profile)**

| **Field** | **Test Case ID** | **Test Input** | **Expected Result** | **Remarks** |
| --- | --- | --- | --- | --- |
| **Name** | **EP-NAME-STRONG-01** | **Emma** | **Accepted** | **Valid** |
| **Name** | **EP-NAME-STRONG-02** | **123** | **Rejected** | **Only digits** |
| **Name** | **EP-NAME-STRONG-03** | **Empty** | **Rejected** | **Required field** |
| **Name** | **EP-NAME-STRONG-04** | **A x 51** | **Rejected** | **Too long** |
| **Username** | **EP-USER-STRONG-01** | **AlphaUser** | **Accepted** | **Valid** |
| **Username** | **EP-USER-STRONG-02** | **a** | **Rejected** | **Too short** |
| **Username** | **EP-USER-STRONG-03** | **averyverylongusername** | **Rejected** | **Too long** |
| **Username** | **EP-USER-STRONG-04** | **user!name** | **Rejected** | **Special characters** |
| **Bio** | **EP-BIO-STRONG-01** | **Empty** | **Accepted** | **Optional** |
| **Bio** | **EP-BIO-STRONG-02** | **A x 160** | **Accepted** | **Max valid** |
| **Bio** | **EP-BIO-STRONG-03** | **A x 161** | **Rejected** | **Above max** |

**Boundary Value Analysis (Edit Profile)**

**Name (1–50)**

| **Test Case ID** | **Input Length** | **Test Input** | **Expected Result** |
| --- | --- | --- | --- |
| **BVA-EP-NAME-01** | **0** | **``** | **Rejected** |
| **BVA-EP-NAME-02** | **1** | **A** | **Accepted** |
| **BVA-EP-NAME-03** | **50** | **A x 50** | **Accepted** |
| **BVA-EP-NAME-04** | **51** | **A x 51** | **Rejected** |

**Username (4–15)**

| **Test Case ID** | **Input Length** | **Test Input** | **Expected Result** |
| --- | --- | --- | --- |
| **BVA-EP-USR-01** | **3** | **usr** | **Rejected** |
| **BVA-EP-USR-02** | **4** | **user** | **Accepted** |
| **BVA-EP-USR-03** | **15** | **user12345678901** | **Accepted** |
| **BVA-EP-USR-04** | **16** | **user123456789012** | **Rejected** |

**Bio (0–160)**

| **Test Case ID** | **Input Length** | **Test Input** | **Expected Result** |
| --- | --- | --- | --- |
| **BVA-EP-BIO-01** | **0** | **``** | **Accepted** |
| **BVA-EP-BIO-02** | **160** | **A x 160** | **Accepted** |
| **BVA-EP-BIO-03** | **161** | **A x 161** | **Rejected** |

**Summary (Edit Profile)**

* **Weak EC Tests: 10**
* **Strong EC Tests: 11**
* **BVA Tests: 11**
* **Total Test Cases: 32**

**4. Meetings Conducted**

**Sprint Planning Meeting**

Overview :

a. Date Time Stamp : 13 March 2025  
b. Attendees :

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

Topics Discussed and Conclusions Reached :

* Defined sprint goals: Implement fuel station management and worker registration features.
* Prioritized user stories: US-4, US-5, US-6, US-7, US-8, etc.
* 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 | 38 | | 38 | | 0 | Sprint starts; no tasks done. | |
| Day 2 | 34 | | 36 | | 2 | Minor delays in setting up environments. | |
| Day 3 | 30 | | 33 | | 3 | UI design for fuel station management. | |
| Day 4 | 26 | | 30 | | 3 | Authentication and worker registration UI. | |
| Day 5 | 22 | | 27 | | 3 | Dashboard UI progress. | |
| Day 6 | 18 | | 24 | | 3 | Minor bug fixes during testing. | |
| Day 7 | 14 | | 21 | | 3 | Inventory tracking UI completed. | |
| Day 8 | 10 | | 18 | | 3 | Testing ongoing for sales reports. | |
| Day 9 | 6 | | 12 | | 3 | Leftover tasks identified (e.g. | |
| Day 10 | 0 | | 6 | | 5 | Sprint ends; some tasks left incomplete. | |
|  | |  | |  | | |

A graph with green and orange lines

AI-generated content may be incorrect.

**Sprint Review Meeting**

Overview :

a. Date Time Stamp : 15 March 2025  
b. Attendees :

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

Topics Discussed and Conclusions Reached :

* Reviewed completed user stories: US-4, US-5, US-6, US-7, US-8, etc.
* Demonstrated working features: Add/View/Update/Remove Fuel Stations, Worker Registration, Sales Reports.

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 : 21 March 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. | Communication gaps during task handovers. | Use Trello for better task tracking. |
| Well-documented codebase. | Limited focus on UI/UX. | Involve a designer for future sprints. |

**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 About  
Bug Report: Incorrect Fuel Inventory Deduction During Sales

Expected Behavior  
When a worker records a fuel sale, the system should deduct the exact quantity of fuel sold from the station's inventory in real time.

Actual Behavior  
The system deducts an incorrect quantity of fuel (e.g., double the amount sold) from the inventory, leading to inaccurate stock levels.

Steps to Reproduce

1. Log in as a worker and navigate to the "Record Fuel Sale" page.
2. Enter the fuel type, quantity (e.g., 10 liters), and payment details.
3. Submit the form to record the sale.
4. Check the fuel inventory level on the dashboard.

Context / Environment  
This issue occurs in the staging environment when testing the "Record Fuel Sales" feature. It is related to User Story #11: Record Fuel Sales .

Possible Solutions

* Review the backend logic for inventory deduction to ensure it processes only the entered quantity.
* Add validation to prevent duplicate API calls during submission.

Definition of Done (DoD)

* All unit tests are written to verify correct inventory deduction.
* The written code is documented.
* The bug is resolved and tested in the staging environment.

Date Time Stamp  
2025-03-15 10:30 AM

Reported By:  
Adan Malik

**Bug Report 2**

Name About  
Bug Report: Low Fuel Alerts Not Triggered Below Threshold

Expected Behavior  
When the fuel level at a station drops below the predefined threshold, the system should send a low fuel alert to the station owner immediately.

Actual Behavior  
The system fails to trigger low fuel alerts even when the fuel level is below the threshold, leaving owners unaware of potential shortages.

Steps to Reproduce

1. Log in as a station owner and navigate to the "Fuel Inventory" page.
2. Simulate a scenario where the fuel level drops below the configured threshold (e.g., 100 liters).
3. Observe whether the system sends a notification or alert.

Context / Environment  
This issue was identified during testing in the development environment. It is related to User Story #14: Alert for Low Fuel Levels .

Possible Solutions

* Debug the logic for monitoring fuel levels and ensure thresholds are correctly evaluated.
* Verify that the notification service is properly integrated and functioning.

Definition of Done (DoD)

* All unit tests are written to validate low fuel alert functionality.
* The written code is documented.
* The bug is resolved and verified in both development and staging environments.

Date Time Stamp  
2025-03-16 02:45 PM

Reported By:  
Hasnain Akhtar

**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 an owner (assuming credentials are predefined)*

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

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

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

*// Step 3: Navigate to the "Add Fuel Station" page*

*WebUI.click(findTestObject('Object Repository/link\_Manage\_Stations'))*

*WebUI.click(findTestObject('Object Repository/button\_Add\_Fuel\_Station'))*

*// Step 4: Fill out the form with valid station details*

*WebUI.setText(findTestObject('Object Repository/input\_Station\_Name'), 'Downtown Fuel Station')*

*WebUI.setText(findTestObject('Object Repository/input\_Station\_Location'), '123 Main St, City')*

*WebUI.setText(findTestObject('Object Repository/input\_Station\_Capacity'), '5000')*

*// Step 5: Submit the form*

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

*// Step 6: Verify that the station was successfully added*

*WebUI.verifyTextPresent('Fuel station added successfully.', false)*

*// Step 7: Verify that the station appears in the list of managed stations*

*WebUI.click(findTestObject('Object Repository/link\_View\_Stations'))*

*WebUI.verifyTextPresent('Downtown Fuel Station', false)*

*// Step 8: Close the browser*

*WebUI.closeBrowser()*

**8. Project Planner and Gantt Chart**

**Project Planner**

* Task 1: Set Up Project Infrastructure (Day 1–2)
  + Initialize the backend project using Node.js and Express.js.
  + Install necessary dependencies (e.g., bcrypt, JWT, MongoDB driver).
  + Set up MongoDB Atlas for cloud-based storage.
  + Configure environment variables for security (e.g., API keys, database credentials).
* Task 2: Implement Fuel Station Management Features (Day 3–5)
  + Add functionality to add new fuel stations (User Story 4).
  + Allow owners to view their fuel stations (User Story 5).
  + Enable owners to update fuel station details (User Story 6).
  + Provide the ability to remove fuel stations (User Story 7).
* Task 3: Worker Management Features (Day 6–8)
  + Implement worker registration (User Story 8).
  + Add functionality to assign workers to specific stations (User Story 9).
  + Ensure worker login is functional (User Story 10).
* Task 4: Inventory and Sales Management Features (Day 9–12)
  + Develop the feature to track fuel inventory in real time (User Story 13).
  + Implement low fuel alerts for station owners (User Story 14).
  + Add functionality to order fuel supply when levels are low (User Story 15).
  + Allow workers to record fuel sales (User Story 11).
* Task 5: Reporting and Analytics Features (Day 13–15)
  + Build the feature to generate sales reports (User Story 12).
  + Integrate reporting APIs with the frontend for data visualization.
  + Ensure reports can be exported in standard formats (PDF/CSV).
* Task 6: Loyalty Points and Pricing Features (Day 16–18)
  + Implement the ability for customers to view loyalty points (User Story 16).
  + Allow owners to manage pricing for fuel types (User Story 19).
* Task 7: Testing and Integration (Day 19–20)
  + Conduct thorough testing of all implemented features.
  + Fix any bugs or issues identified during testing.
  + Ensure smooth integration between frontend and backend components.

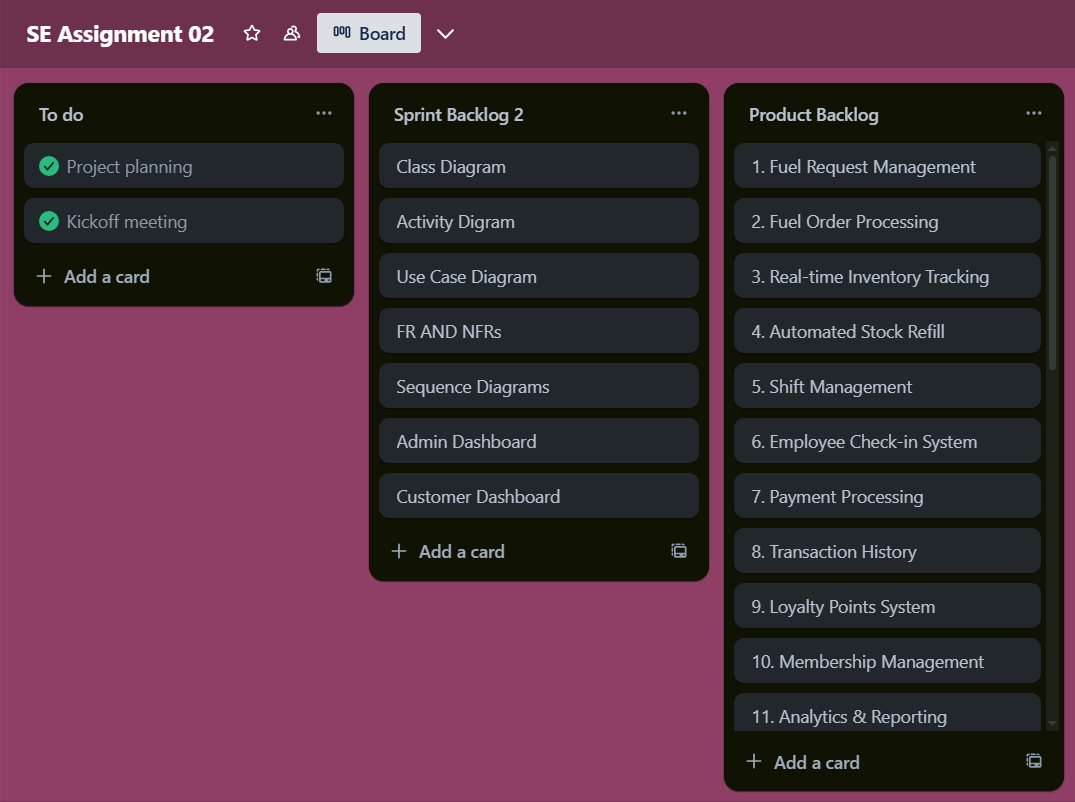
**Gantt Chart**

A graph of a graph

AI-generated content may be incorrect.

**9. Trello Screenshots**

**Start of Sprint**

****

**Middle of Sprint**

**A screenshot of a computer

AI-generated content may be incorrect.**

**End of Sprint**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Scrum Master Activities and Leanings**

**A screenshot of a computer program

AI-generated content may be incorrect.**

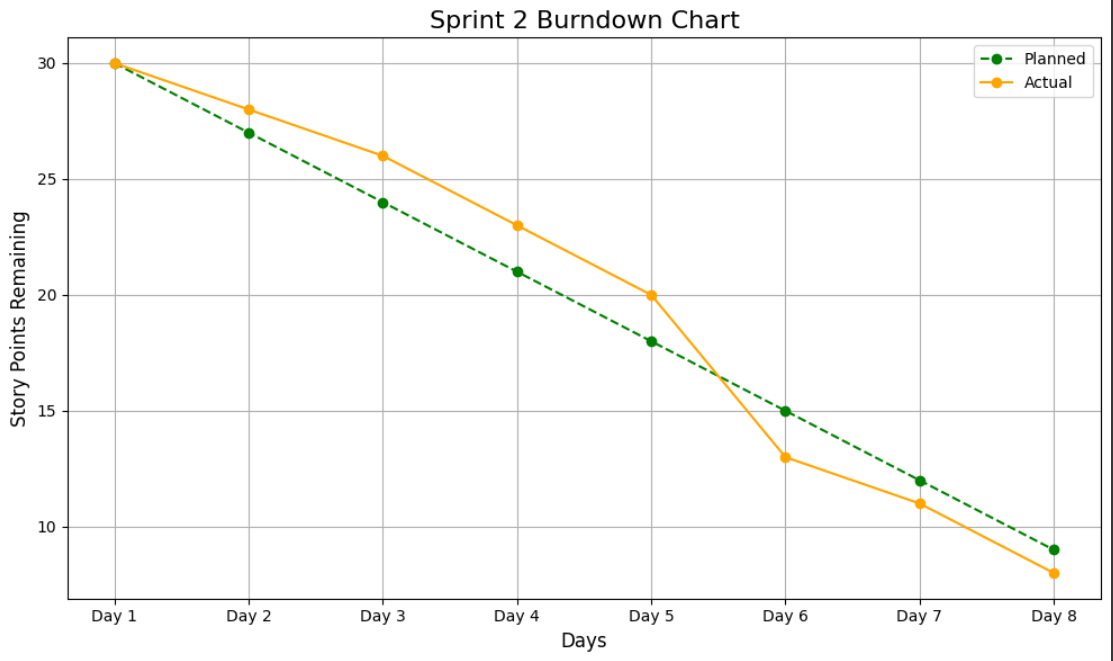
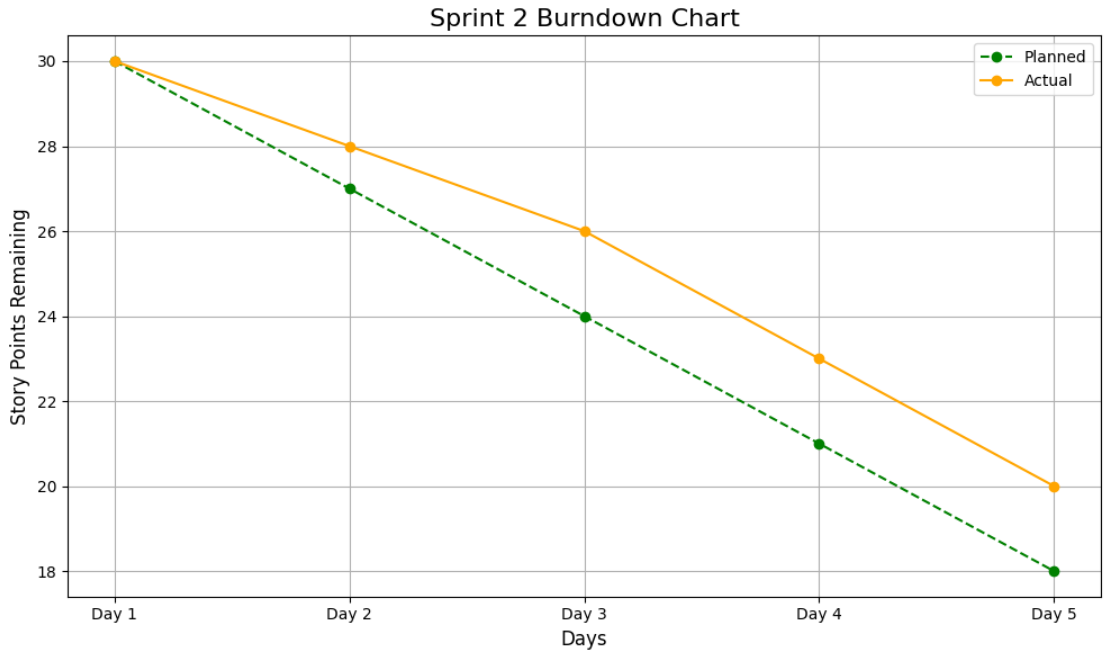
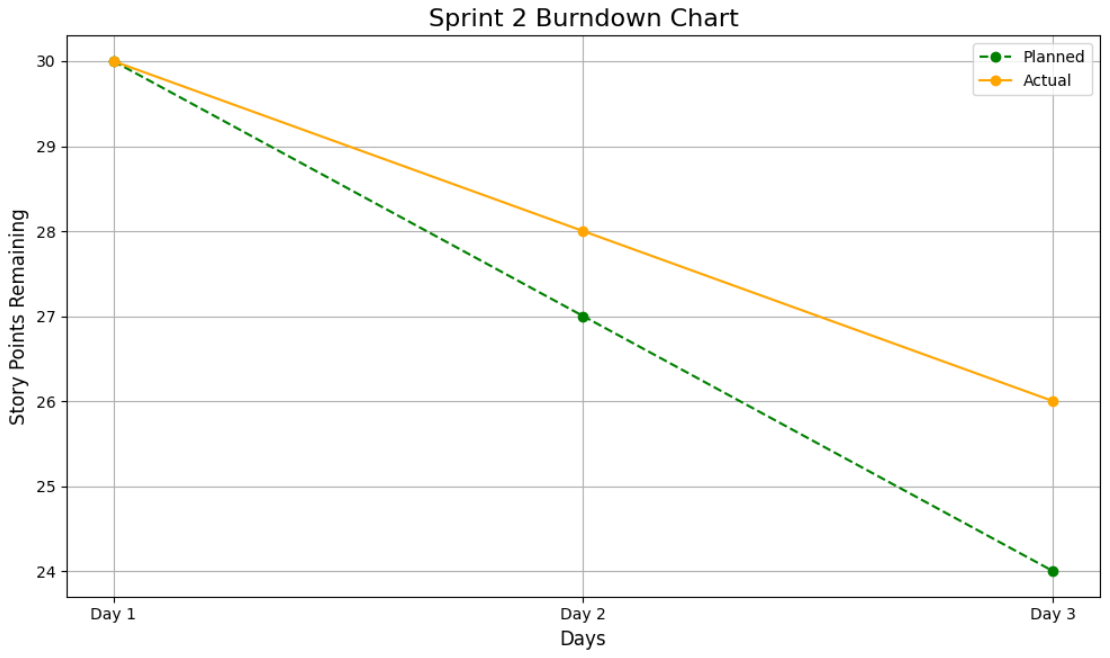
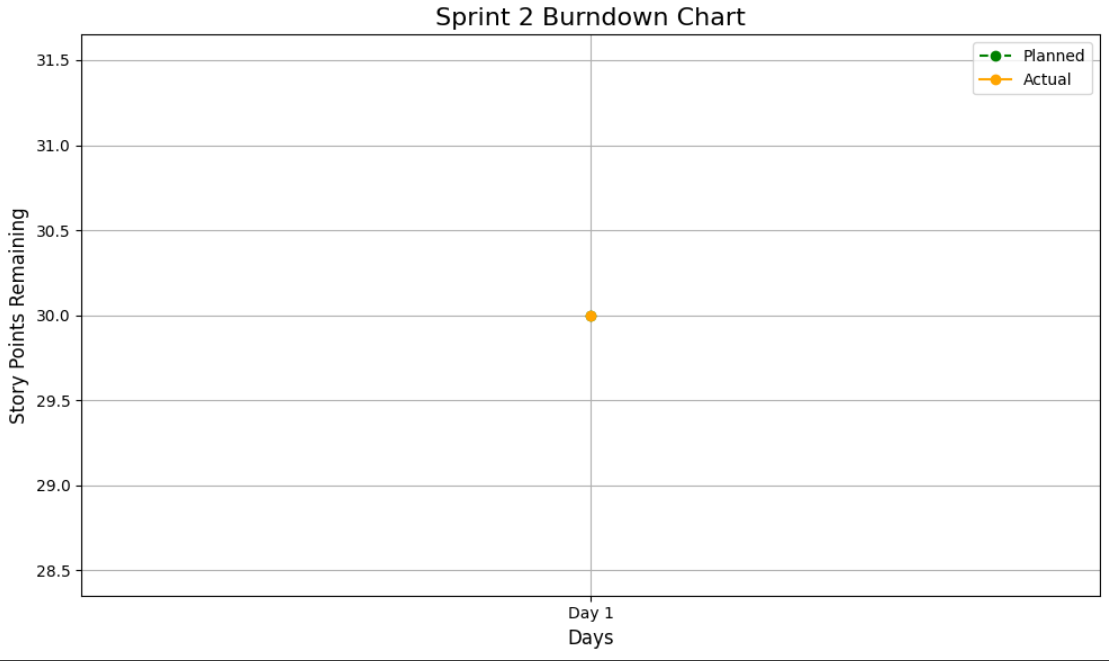
Topics Discussed and Conclusions Reached :

* Reviewed completed user stories: US-4, US-5, US-6, US-7, US-8, etc.
* Demonstrated working features: Add/View/Update/Remove Fuel Stations, Worker Registration, Sales Reports.

Conclusions :

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

**10. Burndown Charts**

****

