# Requirements Specification

**AttendNet: A Petrol Station Digitization Project**

**March 27, 2019**

**Prepared for Ali's Oil**

**By Fossil Files Digital Solutions**

Amandeep Johal · Amrit Jot Madahar · Charlie Friend · Cyrus Erfani

Daniel MacRae · Katelyn Van Gorp · Kristian Darlington · Oliver Lewis

# Table of Contents

# List of Figures

See Appendix B on pages B-2 to B-5 for details on DFD-2.

# List of Tables

# Revision History

| Name | Date | Reason for Changes | Version |
|---|---|---|---|
| Layout | February 26, 2019 | Document layout | 0.1 |
| Draft | February 28, 2019 | Initial drafting of RSD | 0.2 |
| Revised Draft | March 5, 2019 | Requirement refinement | 0.3 |
| Revised Draft | March 9, 2019 | Use Case & Storyboard addition | 0.4 |
| Revised Draft | March 12, 2019 | System diagram addition, final revisions to descriptions and requirements | 0.9 |
| Revised Draft | March 19, 2019 | Narrowing of scope, requirement revisions, diagram updating | 1.0 |
| Revised Draft | March 23, 2019 | Removal of ambiguity in requirements, DFD refactoring as per feedback, use case numbering, other client feedback | 1.1 |
| Final Copy | March 27, 2019 | Storyboard updates, addition of rationales and test scenarios, traceability matrix | 2.0 |

# 1. Introduction

## 1.1 Purpose

This requirements specification document (version 2.0) has been written for the Petrol Station Digitization project at Ali's Oil in Sahiwal, Pakistan. AttendNet, the software solution system for the Petrol Station Digitization project, addresses issues regarding data security, data accessibility, task automation and income/expense traceability to improve company privacy, administrative support, attendant productivity, and theft investigation within the currently implemented system, respectively. The effects of AttendNet on the problem domain meet the overall client objective of improving efficiency and minimizing losses.

## 1.2 Project Scope

The project scope encompasses systems used while logging transaction data, tracking inventory, managing customer debts, and exporting data relating to inventory, transactions and customer debt at the Ali's Oil petrol station and convenience store. AttendNet primarily acts as a secure and remotely accessible replacement for the current ledger-based system, where each ledger covers either transaction, inventory, or customer data for Ali's Oil.

## 1.3 Glossary

| Term | Description |
|---|---|
| Administrator | An employee of Ali's Oil and user class in AttendNet with the authority to configure the User Management subsystem |
| Ali's Oil | A petrol station, convenience store, and provider of basic automotive maintenance in Sahiwal, Pakistan |
| Attendant | A person employed at Ali's Oil |
| CSV | Comma Separated Value file format easily interpreted by analytical software, such as Microsoft Excel |
| Customer Debt Management subsystem | A component of AttendNet that stores customer information and enables Attendants to add or edit customer information. |
| Inventory | All products owned by Ali's Oil available for immediate sale |
| Inventory Management subsystem | A component of AttendNet that stores product information and allows Attendants to add or edit product information. |

| | |
|---|---|
| Ledger | A book containing transaction, inventory, or customer data for Ali’s Oil |
| Notification subsystem | A component of AttendNet that stores notifications and enables Attendants to view notifications. |
| Government of Pakistan Oil and Gas Regulatory Authority | A state-owned corporation regulating nationwide prices on petroleum products |
| Product | A specific type of item sold by Ali’s Oil |
| Transaction | Any exchange of goods or cash involving Ali’s Oil |
| Transaction Management subsystem | A component of AttendNet that stores transaction information and enables Attendants to add or edit transaction information. |
| Unit | A product quantity of 1 |
| User Management subsystem | A component of AttendNet that stores Attendant and Administrator accounts and enables Administrators to add or edit Attendant and Administrator accounts. |

## 1.4 References


[1] "Petrol Station Digitization Request for Proposal Version 1.1", *Ali's Oil*, 2019. [Online]. Available: https://img1.wsimg.com/blobby/go/49b7584f-fd30-4a54-9225-a1d0516c6a66/downloads/1d1sern19_447083.pdf. [Accessed: 16- Jan- 2019]

[2] Ze Shi Li, *Requirements Document Template*, 1st ed. 2019.

[3] "Oil & Gas Regulatory Authority", *Ogra.org.pk*, 2019. [Online]. Available: http://ogra.org.pk/. [Accessed: 02- Feb- 2019]

[4] Fossil Files Digital Solutions, 2019. *Documents*. [Online] Available: https://seng321group3.github.io/designer-site/assets/documents/Ali_Oil_Requirements_Elicitation_Notes.pdf [Accessed 2 Feb. 2019].

## 1.5 Overview

This requirements specification document consists of seven sections, each with a varying number of subsections that provide further detail regarding the topic discussed. Section 1 contains an outline of the AttendNet Petrol Station Digitization project and its scope, as well as a glossary of terms and references list. Section 2 elaborates on the problem domain with AttendNet's features, attendant information, system environment factors and constraints on development. AttendNet's features and each of these features' respective requirements are covered in Section 3, while Section 4 addresses user interface and software interface requirements. Performance requirements, security requirements and software quality attributes of AttendNet are outlined in Section 5. Section 6, System Specifications, details the subsystems of AttendNet through use cases, sequence diagrams, and user interface storyboards. System-wide diagrams like the context diagram (DFD-0), data-flow diagram 1 (DFD-1), entity relationship diagram (ERD), and use case model are displayed in Section 7.

# 2. Overall Description

## 2.1 Product Perspective

AttendNet is being developed to replace the existing finance and Inventory Management subsystem at Ali's Oil. Currently, Ali's Oil's finances and inventory are being tracked with an entirely paper-based system. Each transaction is manually recorded into a table in a notebook, inventory is counted manually, and business calculations are computed by hand. AttendNet will allow Ali's Oil to improve their efficiency by automatically tracking inventory, transactions and fuel prices.

As there are currently no electronic systems in place at Ali's Oil, AttendNet will not interface with any other internal systems. Any updates made to AttendNet, including adding new inventory or changing the price of a product, will be performed manually by attendants. The only external system that AttendNet will interact with is the Government of Pakistan Oil and Gas Regulatory Authority to receive updates on the price of fuel.

## 2.2 Product Features

The features of AttendNet include inventory management, transaction management, and customer debt management.

*Inventory Management*
Information about any product sold at Ali's Oil, including the quantity of each product currently in stock, is tracked by AttendNet. If the quantity of a product drops below a configured threshold, AttendNet notifies attendants and administrators during use.

*Transaction Management*
A record of transactions processed at Ali's Oil are kept by AttendNet.

*Customer Debt Management*
AttendNet tracks customer debt by keeping a record of each customer's contact information and amount owed.

## 2.3 User Classes and Characteristics

Users of AttendNet include the attendants and the administrators of Ali's Oil. Attendants and administrators have limited technical knowledge and minimal experience working with computerized systems.

*Attendant*
Each attendant will perform the following tasks with AttendNet (subject to administrator discretion):

- Login to AttendNet
- Logout of AttendNet
- Add, view, or edit product information
- Add, view, or edit transaction information
- Add, view, or edit customer debt information
- View notifications
- Export data for transactions, inventory, and customer debt

*Administrator*
Each administrator will perform the following tasks with AttendNet:

- Login to AttendNet
- Logout of AttendNet
- Create or edit an attendant account
- Create or edit an administrator account
- Adjust permissions for individual attendants
- Remotely access AttendNet

Exporting data for transactions, inventory, and customer debt is permitted for each attendant that has view permissions for the Transaction Management, Inventory Management, and Customer Debt Management subsystems. AttendNet is also remotely accessible to each administrator and attendant, and this permission cannot be revoked by an administrator.

## 2.4 Operating Environment

AttendNet runs on any computer with a Windows 7 OS or more modern Windows OS at Ali's Oil. An Administrator may remotely access AttendNet through the internet using any computer with a Windows 7 OS at minimum (Windows 8, Windows Vista, Windows 10 are sufficient).

## 2.5 Design and Implementation Constraints

*Budget:* The project budget is equivalent to $5,000 USD.

*Data retention:* Five years' worth of financial data is capable of being stored by AttendNet.

*Language:* English and Urdu characters are supported by AttendNet.

*Software capability:* AttendNet must run on a Windows 7, 8, Vista, or 10 operating system.

*Training time:* An attendant unfamiliar with AttendNet must learn to process transactions and monitor inventory with AttendNet within one hour of introduction to AttendNet.

## 2.6 Assumptions and Dependencies

*Price retrieval:* This project depends on the Government of Pakistan Oil and Gas Regulatory Authority providing fuel price changes for AttendNet to retrieve.

# 3. System Features

## 3.1 Inventory Management

### 3.1.1 Description

Inventory consists of every product for sale in the Ali’s Oil convenience store and at the Ali’s Oil fuel pumps. AttendNet aids in the management of inventory by maintaining a representation of inventory on a per-product basis within AttendNet itself. Each attendant may view, edit, and export AttendNet’s inventory records when necessary.
PRIORITY: High

### 3.1.2 Functional Requirements

**FR-01:** Each product within AttendNet is uniquely identifiable.

*Rationale:*
Each product at Ali’s Oil must be uniquely identifiable so that any two products can be differentiated in transactions. This is necessary for the accurate monitoring of Ali’s Oil inventory.

*Test Scenario [TS-01]:*
Add a product to AttendNet’s inventory. Verify that the added product has an identifying field with a unique value to that product within the scope of the AttendNet inventory.

**FR-02:** An attendant must be able to manually add products with the following product-specific information to AttendNet:

- Price per unit (e.g. $4 per litre, $0.99 per bag)
- Product name
- Product quantity
- Notes
- Low-quantity threshold

*Rationale:*
A product may be added to the inventory at Ali's Oil to satisfy customer demand or increase business. Each attendant manages the addition of products by noting important information pertaining to the added product.

*Test Scenario [TS-02]:*
Add a product to AttendNet's inventory with associated product information. Verify that the added product and associated product information stored within AttendNet is consistent with the information entered.

**FR-03:** An attendant must be able to manually edit the product-specific information of each product (see **FR-02**).

*Rationale:*
Product information necessary for the management of inventory at Ali's Oil may change with time. As such, each attendant may be required to update or otherwise change information pertaining to an existing product of Ali's Oil.

*Test Scenario [TS-03]:*
Edit one or multiple fields for an existing product within AttendNet's inventory. Save the entered product information to AttendNet. Verify that the updated fields for the product stored within AttendNet are consistent with the information entered.

**FR-04:** An attendant must be able to view all products and product-specific information within AttendNet (see **FR-02**).

*Rationale:*
It is necessary for each attendant at Ali's Oil to reference product information for the purpose of customer inquiry, business analysis, or transaction review.

*Test Scenario [TS-04]:*
Ensure multiple products exist within AttendNet's inventory. Verify for all products existing within AttendNet that each product and its associated product information is visible in AttendNet's inventory.

**FR-05:** Recorded quantity of a product is automatically updated when a transaction attributed to a change in inventory is added to AttendNet (see **FR-11**).

*Rationale:*
Each attendant at Ali's Oil has many tasks to complete throughout a workday. Removing the need to manually note any products associated with a transaction increases attendant productivity, eliminates possible human error, and improves data integrity for Ali's Oil inventory.

*Test Scenario [TS-05]:*
Add a transaction involving the sale of one or multiple products to AttendNet. Verify that the quantity of each product associated with that transaction is automatically updated in AttendNet's inventory.

**FR-06:** When the low-quantity limit of a product is reached (a product's quantity goes below a specified limit), a low-quantity notification is generated by AttendNet for each attendant.

*Rationale:*
When a product is of low-quantity at Ali's Oil, it may be necessary for an attendant to take immediate action to replenish the respective product quantity. As such, each attendant should be notified when a product is of low-quantity.

*Test Scenario [TS-06]:*
Update the product quantity for a product in AttendNet's inventory to a value below that of the associated low-quantity threshold for the product. Verify that a low-quantity notification is generated.

**FR-07:** When a fuel product's price is updated in AttendNet's inventory, a fuel-price-update notification is generated for viewing by any attendant in AttendNet (see **FR-08, FR-22**).

*Rationale:*
For Ali's Oil business integrity and customer satisfaction, it is necessary for a fuel product price on any signage at the petrol station to be accurate to the respective fuel product price supplied by a regulatory authority. It is necessary for each attendant at Ali's Oil to be

notified of a fuel product price change so immediate steps can be taken to reflect the fuel product price change throughout the petrol station.

*Test Scenario [TS-07]:*
Update the price of a fuel product in AttendNet's inventory. Verify that a notification indicating that the price of fuel has changed is generated.

**FR-08:** Prices for all fuel products in AttendNet are automatically updated (see **FR-22, NFR-04, NFR-05**).

*Rationale:*
Each attendant at Ali's Oil has many tasks to complete throughout a workday. Removing the need to manually check fuel product prices on the Government of Pakistan Oil and Gas Regulatory website at regular intervals increases attendant productivity and removes possible human error.

*Test Scenario [TS-08]:*
Verify that, for each fuel product, the fuel product's price is automatically updated in AttendNet when an update by the Government of Pakistan Oil and Gas Regulatory Authority is detected.

**FR-09:** A CSV file containing all products and associated product-specific information for each product in AttendNet must be exported from AttendNet (see **FR-02**).

*Rationale:*
An attendant may be required to analyze Ali's Oil inventory data to draw conclusions related to business performance. Each attendant must be able to analyze product information at Ali's Oil in detail to perform this task.

*Test Scenario [TS-09]:*
Ensure multiple products exist within AttendNet's inventory. Export all products and associated product information to a CSV file. Verify that a valid CSV file containing all products and respective product information stored within AttendNet has been generated.

## 3.2 Transaction Management

### 3.2.1 Description

Transaction Management at Ali’s Oil pertains to any exchange of goods or cash between Ali’s Oil and external entities, such as customers or banks. AttendNet aids the management of cash flow by maintaining records of individual transactions and the information associated with those transactions.
PRIORITY: Medium

### 3.2.2 Functional Requirements

**FR-10:** Each transaction within AttendNet is uniquely identifiable.

*Rationale:*
Each transaction at Ali’s Oil must be uniquely identifiable so that any two transactions can be differentiated. This is necessary for the accurate monitoring of Ali’s Oil transactions, inventory, and customer debts.

*Test Scenario [TS-10]:*
Add a transaction to AttendNet’s inventory. Verify that the added transaction has an identifying field with a value unique to that transaction within the scope of all AttendNet transactions.

**FR-11:** Each attendant must be able to manually add transactions with the following transaction-specific information to AttendNet:

- Amount paid
- Amount given in change
- Product(s) sold
- For each entered product, the quantity exchanged
- Customer name
- Notes

*Rationale:*
For reasons such as data analysis, theft prevention, or customer inquiry, it is necessary to document each transaction at Ali’s Oil.

*Test Scenario [TS-11]:*
Add a transaction to AttendNet with associated transaction information. Verify that the added transaction and associated transaction information stored within AttendNet is consistent with the entered information.

**FR-12:** Each attendant must be able to manually edit the transaction-specific information of each transaction (see **FR-11**).

*Rationale:*
In case of attendant error when entering transaction information during transaction documentation, each attendant must be able to edit transaction information after a transaction is documented.

*Test Scenario [TS-12]:*
Edit one or multiple fields for an existing transaction within AttendNet. Save the entered transaction information to AttendNet. Verify that the updated fields for the transaction are stored in AttendNet are consistent with the entered information.

**FR-13:** An attendant must be able to view all transactions and transaction-specific information within AttendNet (see **FR-11**).

*Rationale:*
For reasons such as customer inquiry, each attendant must be able to reference transaction records after initial documentation.

*Test Scenario [TS-13]:*
Ensure multiple transactions exist within AttendNet. Verify for all transactions existing within AttendNet that each transaction and its associated transaction information is visible within AttendNet.

**FR-14** A CSV file containing all transactions and associated information for each one of those transactions in AttendNet for a specified calendar month and year shall be exported from AttendNet.

*Rationale:*
An attendant may be required to analyze Ali's Oil transaction data to draw conclusions related to business performance. Each attendant must be able to analyze transaction information at Ali's Oil in detail to perform this task.

*Test Scenarios [TS-14]:*
Ensure multiple transactions exist within AttendNet. Export AttendNet transactions and associated transaction information for a specified calendar month and year to a CSV file. Verify that all transactions created within the specified calendar month and year are included in the CSV file.

## 3.3 Customer Debt Management

### 3.3.1 Description

If a customer requires a product that cannot be paid for immediately, the customer may be allowed to pay for that product at a later date. AttendNet aids in the management of customers who have accumulated debt with Ali's Oil.
PRIORITY: Medium

### 3.3.2 Functional Requirements

**FR-15:** Each customer within AttendNet is uniquely identifiable.

*Rationale:*
To monitor individual debts accrued by customers of Ali's Oil, customer information must be documented. Each customer with accrued debt must be uniquely identifiable so any two customers and the respective debts can be differentiated.

*Test Scenario [TS-15]:*
Add a customer to AttendNet. Verify that the added customer has an identifying field with a unique value to that customer within the scope of all AttendNet customers.

**FR-16:** Each attendant must be able to manually add customers to AttendNet with the following customer-specific information:

- customer name
- customer debt
- customer address
- customer phone number

*Rationale:*
A customer of Ali's Oil may be allowed to accrue debt. If this situation occurs, it is necessary to document any important contact information for the customer with debt such that the customer can be contacted in the future and the associated debt can be monitored.

*Test Scenario [TS-16]:*
Add a customer to AttendNet with associated customer information. Verify that the added customer and associated customer information stored within AttendNet is consistent with the entered information.

**FR-17:** Each attendant must be able to manually edit the customer-specific information of each customer (see **FR-16**).

*Rationale:*
It is possible for information associated with a customer at Ali's Oil to change throughout time or to have been entered incorrectly. It is necessary to update any customer information that has been changed or is incorrect to maintain accurate contact information for the management of customer debts.

*Test Scenario [TS-17]:*
Edit one or multiple fields for an existing customer within AttendNet. Save the entered customer information to AttendNet. Verify that the updated fields for the customer are stored in AttendNet are consistent with the entered information.

**FR-18:** An attendant must be able to view all customers and customer-specific information are available within AttendNet (see **FR-16**).

*Rationale:*
It is necessary that each attendant of Ali's Oil is able to reference customer specific information to satisfy customer inquiry, monitor customer debt, and ensure that correct information has been documented.

*Test Scenario [TS-18]:*
Ensure multiple customers exist within AttendNet. Verify for all customers existing within AttendNet that each customer and its associated customer information is visible within AttendNet.

**FR-19:** A CSV file containing all customers and associated information for each one of those customers in AttendNet shall be exported from AttendNet.

*Rationale:*
An attendant of Ali's Oil may be required to analyze customer debt or other information in detail.

*Test Scenario [TS-19]:*
Ensure multiple customers exist within AttendNet. Export all customer and associated customer information to a CSV file. Verify that a valid CSV file containing all customers and respective customer information stored within AttendNet has been generated.

**FR-20** For each customer, the associated customer debt is automatically updated by AttendNet if the customer is involved in a transaction involving the exchange of goods (see **FR-11**).

*Rationale:*
Each attendant or Ali's Oil has numerous tasks to complete throughout the workday. By removing the need to manually document changes in the debt for a customer when a transaction is documented, productivity is increased and opportunity for human error decreased.

*Test Scenario [TS-20]:*
Add a transaction to AttendNet involving a change in customer debt for an existing customer in AttendNet. Verify that the associated customer debt for the customer involved in the transaction has been updated in AttendNet to reflect the change in debt due to the added transaction.

# 4. External Interface Requirements

## 4.1 User Interfaces

### 4.1.1 Description

Attendants may not have prior experience with software as all store records are currently hard copies. Integrating AttendNet will be minimally difficult.

### 4.1.2 Functional Requirements

**FR-21:** Each fuel-price update notification and product low-quantity notification previously generated by AttendNet shall be viewed by an attendant (see **FR-06, FR-07**).

*Rationale:*
When the fuel price for a product is updated, an attendant may be required to manually change signage at the Ali's Oil fuel pumps. When a product is in low-quantity, an attendant may be required to replenish the quantity for that product. As such, it is important that each fuel-price-update notification and each low-quantity notification is viewable by an attendant so the appropriate actions may be taken.

*Test Scenario [TS-21]:*
View a fuel-price-update notification, then a product low-quantity notification. Verify that information associated with each viewed notification is displayed by AttendNet.

## 4.2 Software Interfaces

### 4.2.1 Description

To retrieve fuel prices regulated by a government body, AttendNet interfaces with an external website.

### 4.2.2 Functional Requirements

**FR-22:** Fuel product prices are retrieved from Pakistan Oil and Gas Authority.

*Rationale:*
To ensure Ali's Oil's business integrity and customer satisfaction, fuel product prices must reflect the respective nationally regulated fuel product prices provided by the Government of Pakistan Oil and Gas Regulatory Authority.

*Test Scenario [TS-22]:*
After the Government of Pakistan Oil and Gas Regulatory Authority updates the nationally regulated price of a fuel product, verify that the price of the respective fuel product in AttendNet's inventory is updated to match the price listed by the Government of Pakistan Oil and Gas Regulatory Authority.

## 4.3 Communication Interfaces

### 4.3.1 Description

AttendNet provides each administrator with the ability to access AttendNet when off Ali's Oil premises.

### 4.3.2 Functional Requirements

**FR-23:** AttendNet is remotely accessible through the internet.

*Rationale:*
Each attendant and administrator of Ali's Oil may be required to aid another attendant or administrator of Ali's Oil or perform work-related duties while off the premises of the petrol station.

*Test Scenario [TS-23]:*
Attempt to access AttendNet through the internet. Verify that AttendNet can be accessed.

# 5. Other Non-Functional Requirements

## 5.1 Performance Requirements

### 5.1.1 Description

AttendNet performs certain functional requirements within a specified time frame.

### 5.1.2 Non-Functional Requirements

**NFR-01:** Each attendant and administrator must be able to remotely access AttendNet within 10 seconds of attempted connection.

*Rationale:*
Tasks completed at Ali's Oil are often time-sensitive, thus an attendant or administrator should not be required to wait an extended amount of time to begin a task.

*Test Scenario [TS-24]:*
Remotely access AttendNet from a reliable nternet connection. Verify that AttendNet can be accessed in less than 10 seconds.

**NFR-02:** Product low-quantity notifications are generated within 10 seconds of the respective product quantity update (see **FR-06, FR-21**).

*Rationale:*
It may be necessary for an attendant to take immediate action to replenish product quantity when a product is of low quantity. As such, each attendant should be notified of a product being of low quantity without unreasonable delay.

*Test Scenario [TS-25]:*
Ensure a product exists within AttendNet's inventory. Update the product quantity for a product in AttendNet's inventory to a value below that of the associated low-quantity threshold for the product. Verify that a low-quantity notification is generated within 10 seconds of the change to the product quantity.

**NFR-03:** Data exported from AttendNet into a CSV file must be available within 30 seconds of initiating the export (see **FR-08**, **FR-13**, **FR-18**)

*Rationale:*
AttendNet data exports may be time-sensitive. To allow for quick reviewing of Ali's Oil processes and performance, an attendant should be able to analyze Ali's Oil data without unreasonable delay.

*Test Scenario [TS-26]:*
Export inventory data from AttendNet to a CSV file. Export transaction data from AttendNet to a CSV file. Export customer data from AttendNet to a CSV file. Verify that the CSV file corresponding to each export is made available to an attendant in less than 30 seconds.

**NFR-04:** Fuel prices are updated from Government of Pakistan Oil and Gas Regulatory Authority in under 10 seconds from the detection of an update (see **FR-08**, **FR-22**, **NFR-05**).

*Rationale:*
To ensure accurate and fair prices on fuel products for customers of Ali's Oil, it is important that fuel product prices are updated to reflect changes in the Government of Pakistan Oil and Gas Regulatory Authority fuel product price regulations. Once the change is discovered, the fuel product price at Ali's Oil should be updated without unreasonable delay.

*Test Scenario [TS-27]:*
Verify that, upon detection of a nationally regulated fuel product price update from the Government of Pakistan Oil and Gas Regulatory Authority, the respective fuel product price is updated within AttendNet in less than 10 seconds.

**NFR-05:** Fuel price updates from Government of Pakistan Oil and Gas Regulatory Authority are retrieved every 6 hours (see **FR-08**, **FR-22**).

*Rationale:*
The oil and gas prices set by the Government of Pakistan Oil and Gas Regulatory Authority are subject to change multiple times a day and it is important that the oil and gas sold at Ali's Oil accurately reflects these prices.

*Test Scenario [TS-28]:*
Verify that AttendNet retrieves fuel-price-updates from the Government of Pakistan Oil and Gas Regulatory Authority every 6 hours.

## 5.2 Security Requirements

### 5.2.1 Description and Priority

AttendNet maintains the confidentiality of sensitive information regarding business operations by allowing the administrator to invoke attendant accessibility limitations.

### 5.2.2 Non-Functional Requirements

**NFR-06:** Each administrator must be able to view and edit attendant profiles and permissions in AttendNet.

*Rationale:*
An administrator will need to be able to edit the information associated with each attendant. Additionally, the administrator needs to be able limit the tasks each attendant is permitted to perform to ensure no attendant can act maliciously.

*Test Scenario [TS-29]:*
Navigate to an attendant profile. Ensure that information pertaining to the attendant is displayed. Edit information about the attendant, including permissions. Save edited information to AttendNet. Verify that the attendant profile reflects these changes including the attendant's permissions.

**NFR-07:** At the discretion of an administrator, each attendant must be able to add, view, and edit products; add, view, and edit transactions; add, view, and edit customers; view notifications regarding low inventory and fuel price updates; and export inventory, transaction, and customer data (see **NFR-06**).

*Rationale:*
An administrator will decide which tasks each individual attendant can perform to assign responsibilities.

*Test Scenario [TS-30]:*
As an attendant, attempt to add, view, and edit products; add view, and edit transactions; add, view, and edit customers; view notifications regarding low inventory and fuel price updates; and export inventory, transaction, and customer data. Verify that only actions granted permissible by an administrator in the attendant profile were able to be performed. Verify that actions not granted permissible by the administrator in the attendant profile were not able to be performed.

## 5.3 Software Quality Attributes

### 5.3.1 Description

An attendant using AttendNet may not be experienced with software systems like AttendNet. As such, AttendNet must be user-friendly.

### 5.3.2 Non-Functional Requirements

**NFR-08:** AttendNet functionality is usable to those with little or no prior experience with software systems after one hour of training by an administrator or attendant experienced with using AttendNet.

*Rationale:*
The operations at Ali Oil's requires the coordination between several administrators and attendants, thus it is imperative that recently-hired attendants can begin working independently as soon as possible. There cannot be an unreasonably lengthy training period for each attendant as this would disrupt business operations and decrease efficiency.

*Test Scenario [TS-31]:*
Provide an attendant with little or no prior experience in using software systems one hour of training in the use of AttendNet. Verify that the attendant can perform basic tasks such as adding a transaction without assistance.

# 6. System Specifications

## 6.1 Inventory Management

Inventory is all products for sale in the Ali's Oil convenience store and at Ali's Oil fuel pumps. AttendNet assists in the management of inventory by maintaining a record of inventory on a per-product basis within AttendNet itself. Each attendant may view and edit AttendNet's inventory as required.

### 6.1.1 Use Case: Add Product [UC-01]

#### 6.1.1.1 Use Case Description: Add Product [UC-01]

This use case details the steps taken by an attendant to add a product and product information to the Inventory Management subsystem (see Table 1).

*Table 1: UC-01 Description - Add Product*

| **Actors** | Attendant |
|---|---|
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has the privileges to edit the Inventory Management subsystem as set by an Administrator (see UC-07).<br>• The entered product does not currently exist in the Inventory Management subsystem. |
| **Steps** | 1. The Attendant requests to enter the Inventory Management subsystem.<br>2. The Attendant is presented with<br>  • the option to add a product to the Inventory Management subsystem,<br>  • a list of products stored in the Inventory Management subsystem sorted by alphabetical order,<br>  • for each product, the option to view all information associated with that product, and,<br>  • the option to export information associated with each product in the Inventory Management subsystem.<br>3. The Attendant requests to add a product to the Inventory Management subsystem.<br>4. The Attendant is presented with the ability to enter information for a product and the option to save the |

| | |
|---|---|
| | product information in the Inventory Management subsystem.<br>5. The Attendant enters the following product information the respective fields:<br>• product name<br>• price per unit (e.g. $1.99/kg)<br>• quantity<br>• low-quantity notification threshold<br>• notes (optional)<br>6. The Attendant requests to save the entered product information.<br>7. The Attendant is presented with confirmation that the product and its information was saved successfully in the Inventory Management subsystem. |
| **Success Conditions** | • A product and its respective information have been added to the Inventory Management subsystem. |
| **Alternate Paths** | 6a.<br>I. The Attendant is notified that the entered quantity for a product cannot be a negative value and of the correct format for a product quantity.<br>II. The Attendant enters the quantity as a positive value.<br>III. Continue with Step 6.<br><br>6b.<br>I. The Attendant is notified that the entered product price is not a valid denomination of rupees and of the correct format for a product price.<br>II. The Attendant enters the price as a valid denomination of rupees.<br>III. Continue with Step 6. |

### 6.1.1.2 Sequence Diagram: Add Product [UC-01]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant adds a product to the Inventory Management subsystem (see Figure 1). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Inventory Management subsystem.

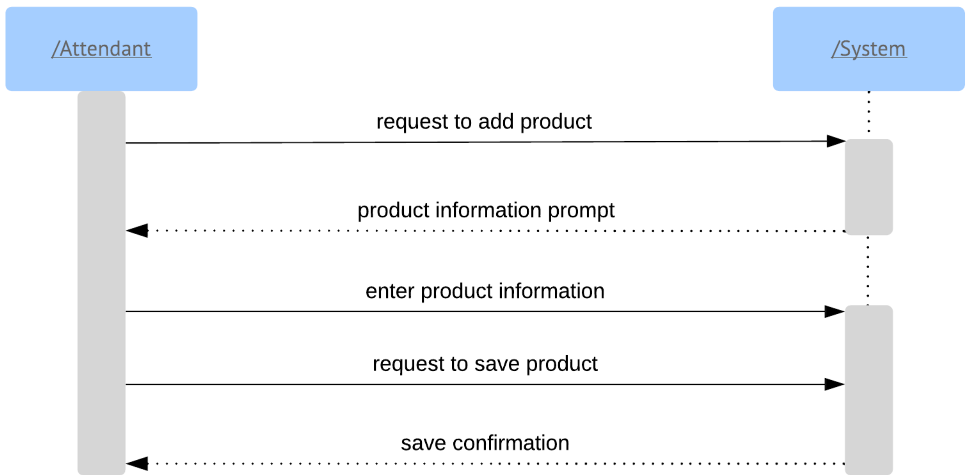
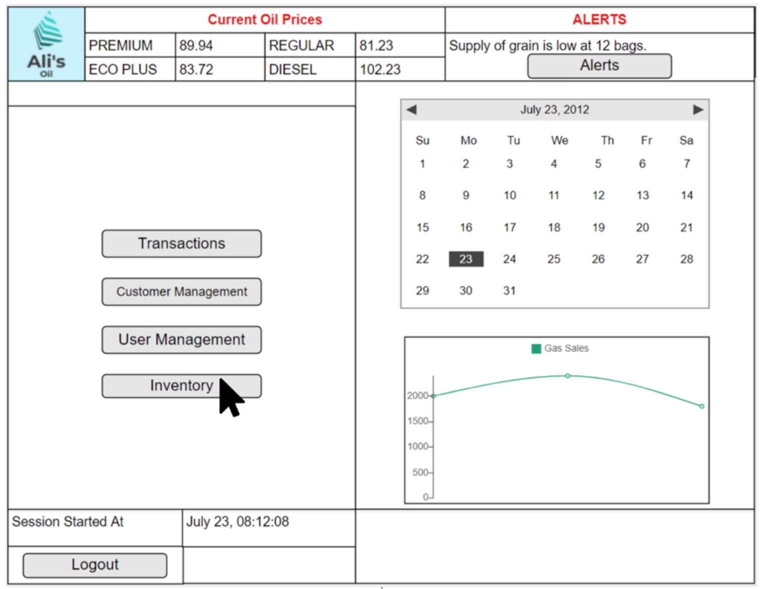



*Figure 1: Sequence diagram for adding a product to the Inventory Management subsystem*

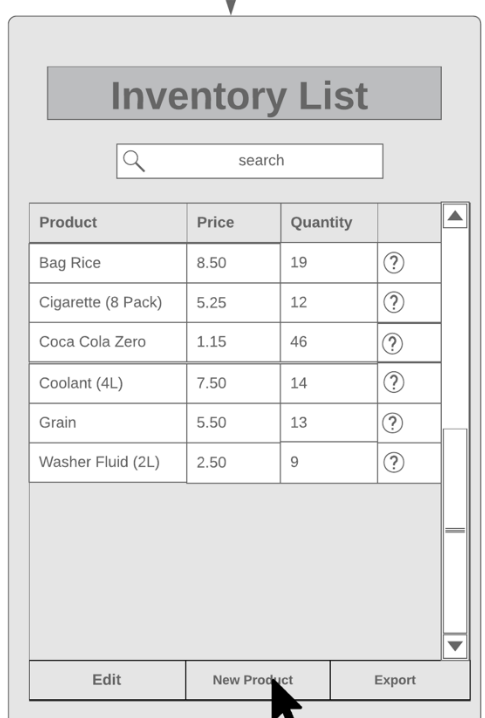
### 6.1.1.3 Storyboard: Add Product [UC-01]

The following storyboard details a scenario in which an attendant is adding a new product to the Inventory Management subsystem (see Figure 2). A shipment of products has been delivered. The shipment includes one-hundred 355 mL cans of Pepsi. An attendant wants to add the Pepsi to the inventory.

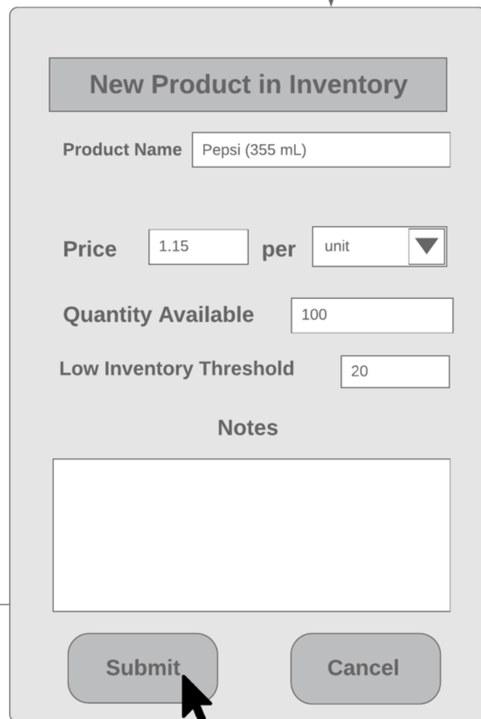
Step 1:
The attendant clicks the "Inventory" button on the home screen to access the inventory management screen.

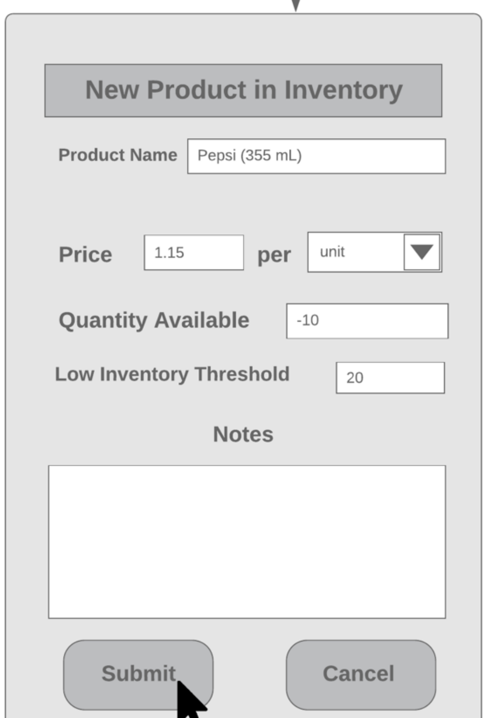



Step 2:
The attendant clicks the "New Product" button in the inventory list screen.

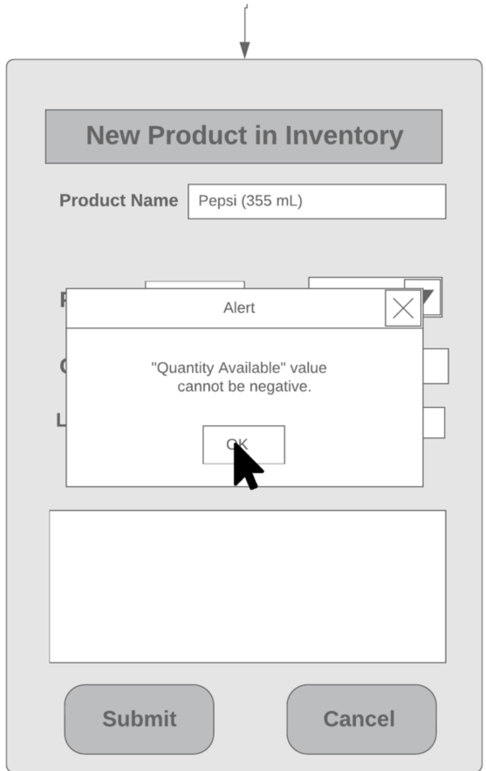


Step 3:
The attendant enters the product name, price, and quantity. When finished, the attendant clicks the "Submit" Button

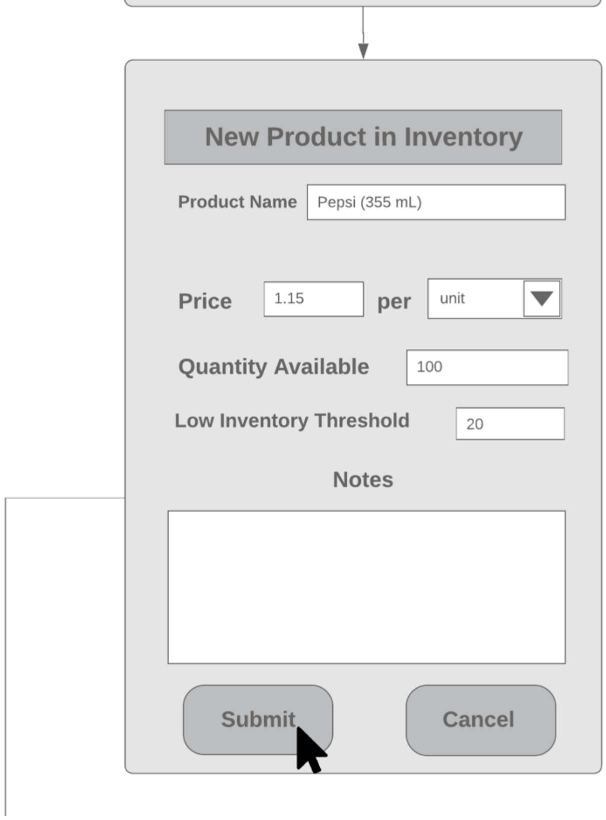


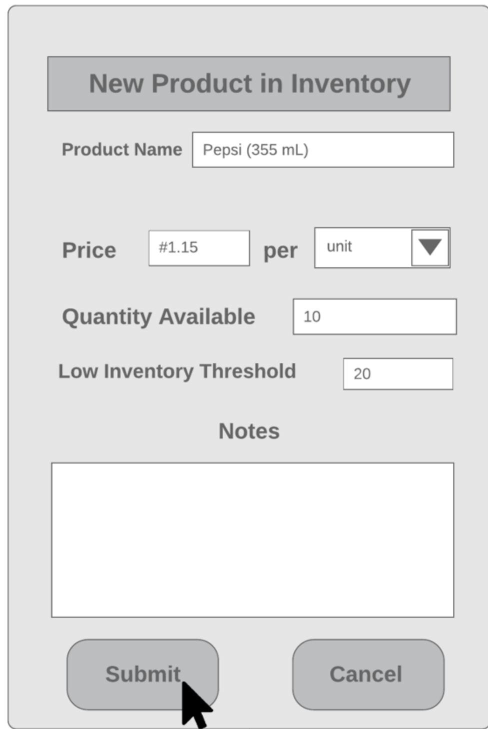

**Alternate Path**
Step 3a.i:
The attendant makes an error by entering a negative value into the "Quantity Available" field. The attendant clicks the "Submit" button to save the changes.




Step 3a.ii:
An alert is shown to the attendant with the error messages stating that the product quantity cannot be negative. The attendant clicks the "OK" button to return close the alert and fix the mistakes.



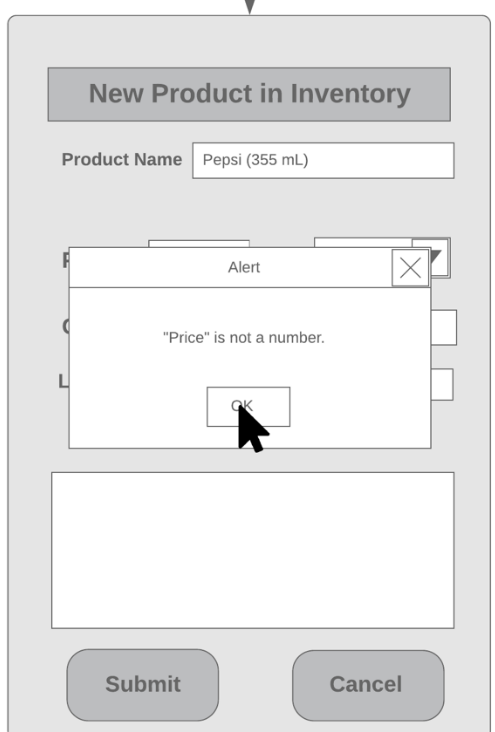

Step 3a.iii:
The attendant changes the quantity available to be a positive integer. The attendant clicks the "Submit" button to save the changes.




**Alternate Path**

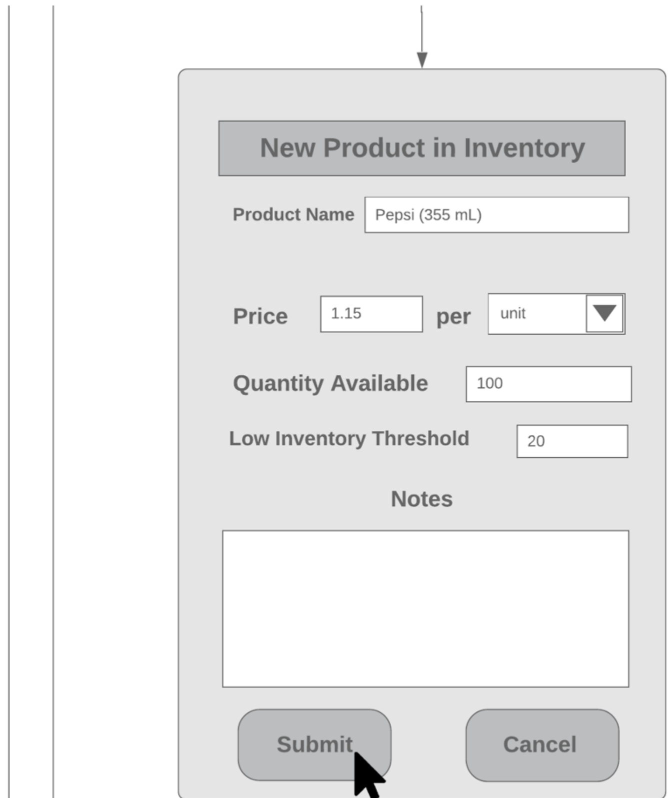
Step 3b.i:

The attendant makes an error by entering the '#' symbol into the price input field. The attendant clicks the "Submit" button to save the changes.

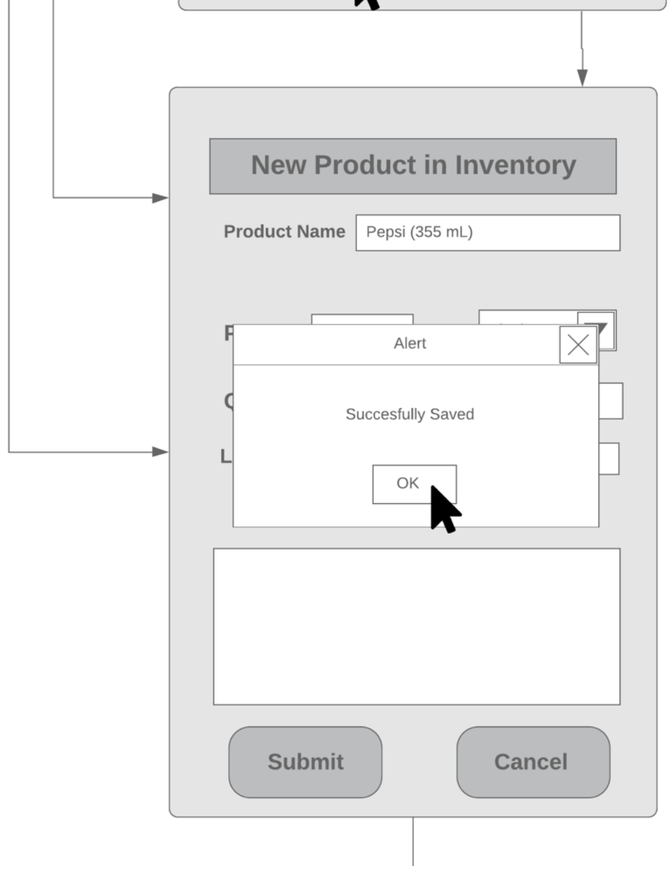



Step 3b.ii:

An alert is shown to the attendant with the error messages stating that the price is not a number. The attendant clicks the "OK" button to return close the alert and fix the mistake.




Step 3b.iii:
The attendant removes the '#' symbol from the price input field. The attendant clicks the "Submit" button to save the changes.




Step 4:
A confirmation messageis displayed to the attendant showing that the information has been saved. The user clicks "OK".

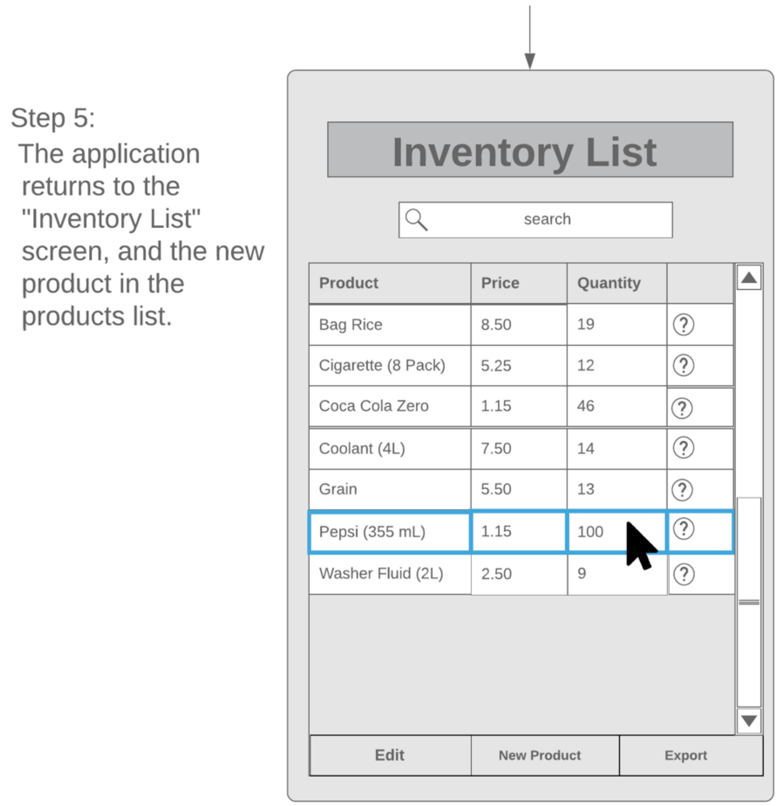



*Figure 2: Storyboard for adding a product to the Inventory Management subsystem*

### 6.1.2 Use Case: Edit Product [UC-02]

#### 6.1.2.1 Use Case Description: Edit Product [UC-02]

This use case details the steps taken by an attendant to edit the product information currently stored in the Inventory Management subsystem for a product (see Table 2).

*Table 2: UC-02 Description - Edit Product*

| | |
|---|---|
| **Actors** | Attendant |
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has editing privileges for the Inventory Management subsystem as set by an Administrator (see UC-07).<br>• The product information to be edited is associated with a product currently existing in the Inventory Management subsystem. |
| **Steps** | 1. The Attendant requests to enter the Inventory Management subsystem.<br>2. The Attendant is presented with |

| | |
|---|---|
| | • the option to add a product to the Inventory Management subsystem,<br>• a list of products stored in the Inventory Management subsystem sorted by alphabetical order,<br>• for each product, the option to view all information associated with that product, and,<br>• the option to export information associated with each product in the Inventory Management subsystem.<br>3. The Attendant requests to view all information associated with a specific product.<br>4. The Attendant is presented with information associated with the chosen product and the option to edit information for the chosen product.<br>5. The Attendant requests to edit product information for the chosen product.<br>6. The Attendant is presented with the ability to edit information associated with the chosen product.<br>7. For each of the following product information fields, the Attendant enters a value if the existing saved value for the respective field needs to be changed:<br>• product name<br>• price per unit (e.g. $1.99/kg)<br>• product quantity (e.g. Litres, kg, lb, unit)<br>• low-quantity notification threshold<br>• notes (optional)<br>8. The Attendant requests to save the entered product information.<br>9. The Attendant is presented with confirmation that the product information was entered successfully. |
| **Success Conditions** | • Any information associated with a product that underwent a change of value has been updated in the respective Inventory Management subsystem record. |
| **Alternate Paths** | 7a.<br>I. The Attendant enters a negative value for the product quantity.<br>II. The Attendant is notified that the entered quantity for a product cannot be a negative value and of the correct format for a product quantity.<br>III. The Attendant enters the quantity as a positive value. |

| | |
|---|---|
| | IV. Continue with Step 7.<br><br>7b.<br>I. The Attendant enters an invalid denomination of rupees for the product price.<br>II. The Attendant is notified that the entered product price is not a valid denomination of rupees and of the correct format for a product price.<br>III. The Attendant enters the price as a valid denomination of rupees.<br>IV. Continue with Step 7. |

### 6.1.2.2 Sequence Diagram: Edit Product [UC-02]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant edits the product information stored in the Inventory Management subsystem for a product (see Figure 3). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Inventory Management subsystem.

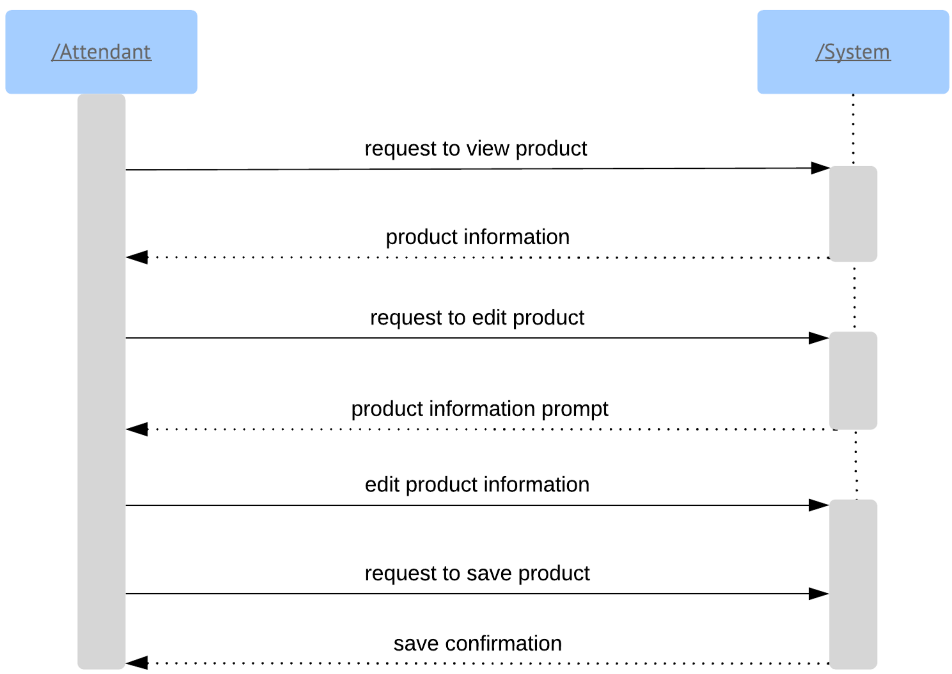
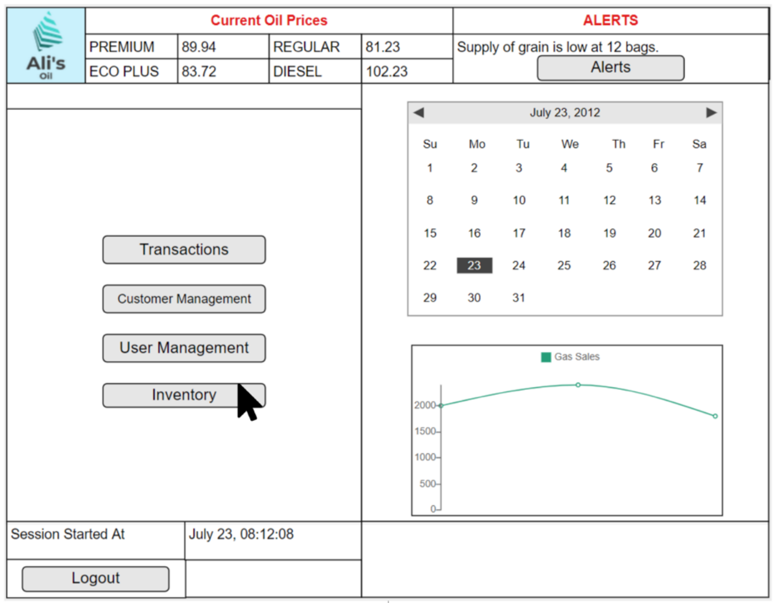



*Figure 3: Sequence diagram for editing a product in the Inventory Management subsystem*

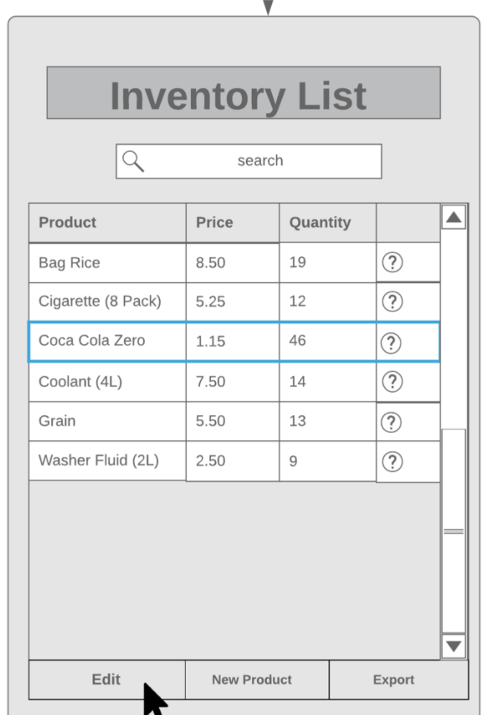
### 6.1.2.3 Storyboard: Edit Product [UC-02]

The following storyboard details a scenario in which an attendant is updating the information associated with Coca Cola Zero (see Figure 4). The price of Coca Cola Zero has increased from $2.10 to 2$2.60 and an attendant wants to increase the price of Coca Cola Zero in AttendNet.

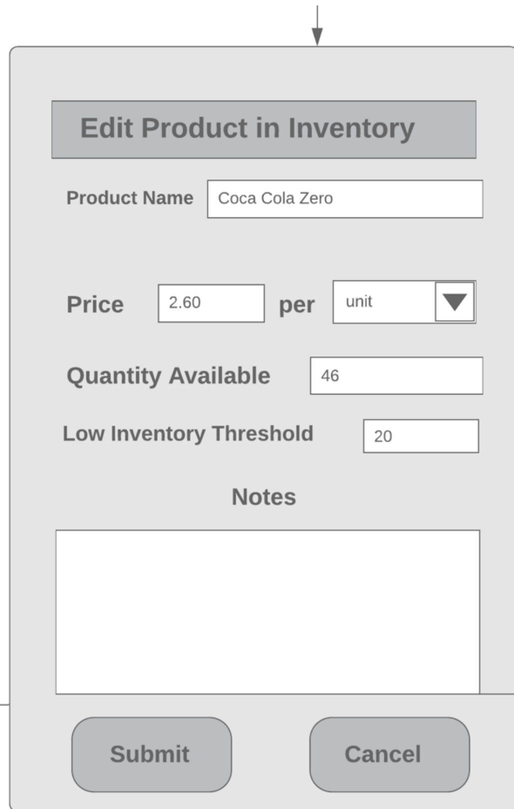
Step 1:
The attendant clicks the "Inventory" button on the home screen to access the inventory management screen.

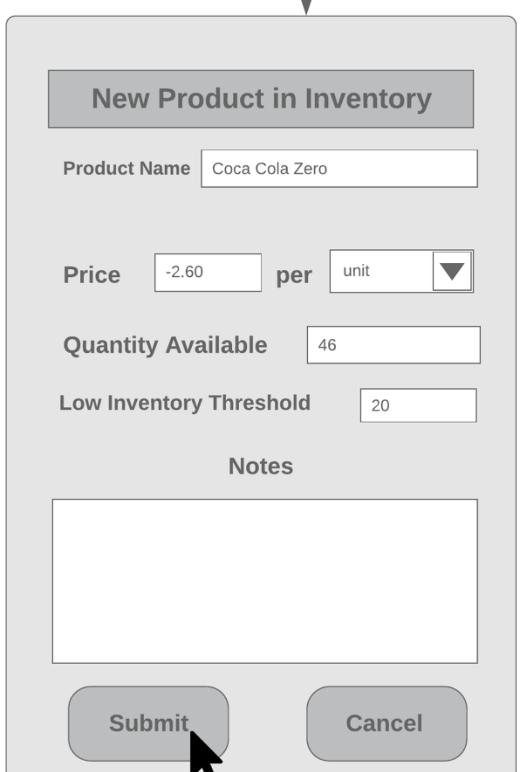



Step 2:
The attendant clicks the "Edit" button in the inventory list screen.



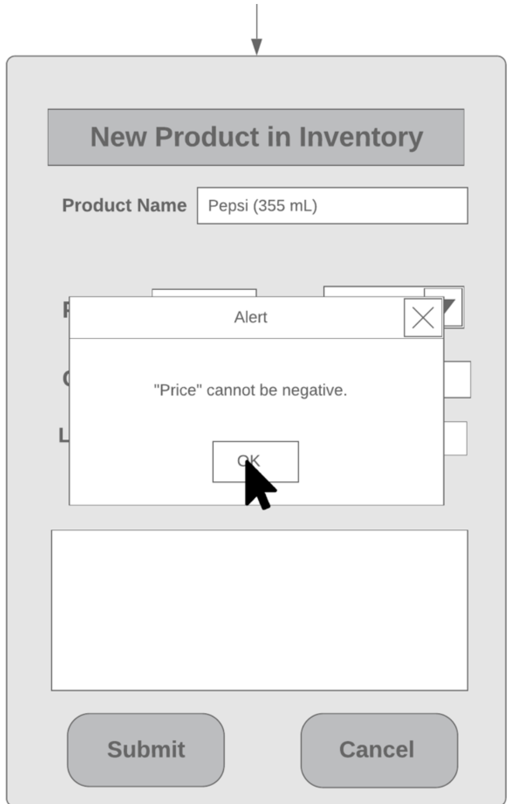
Step 3:
The attendant enters the updated price in the "Price" input field and then clicks the "Submit" button.



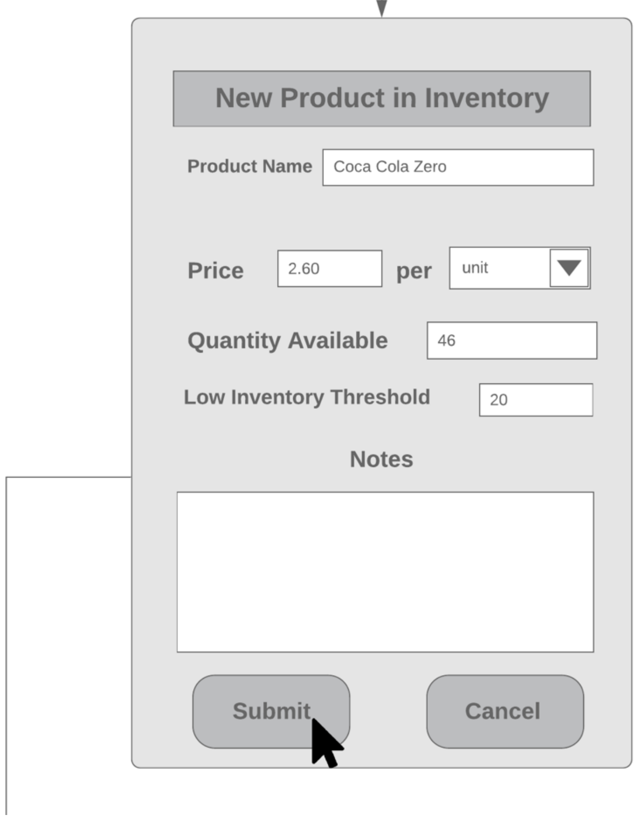




**Alternate Path**

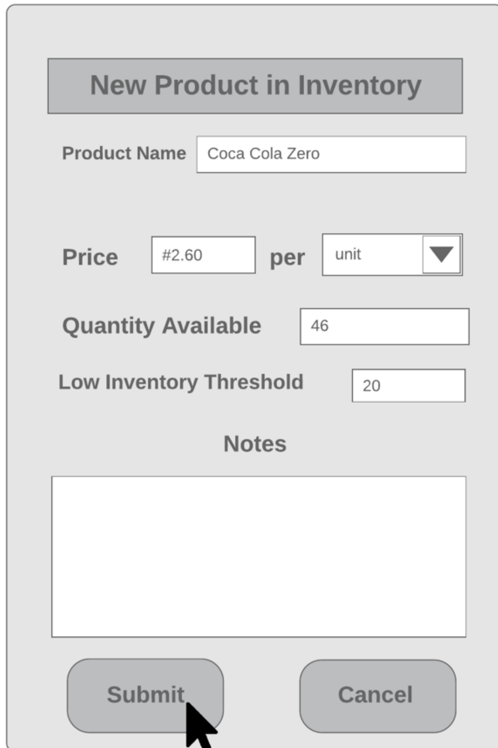
Step 3a.i:
The attendant makes an error by entering a negative value into the "Price" field. The attendant clicks the "Submit" button to save the changes.




Step 3a.ii:
An alert is shown to the attendant with the error messages stating that the price cannot be negative. The attendant clicks the "OK" button to return close the alert and fix the mistakes.



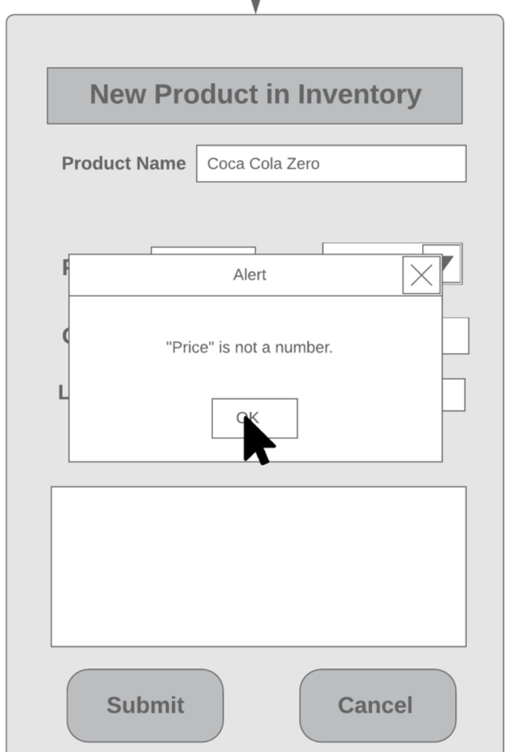

Step 3a.iii:
The attendant changes the price to be a positive number. The attendant clicks the "Submit" button to save the changes.




**Alternate Path**

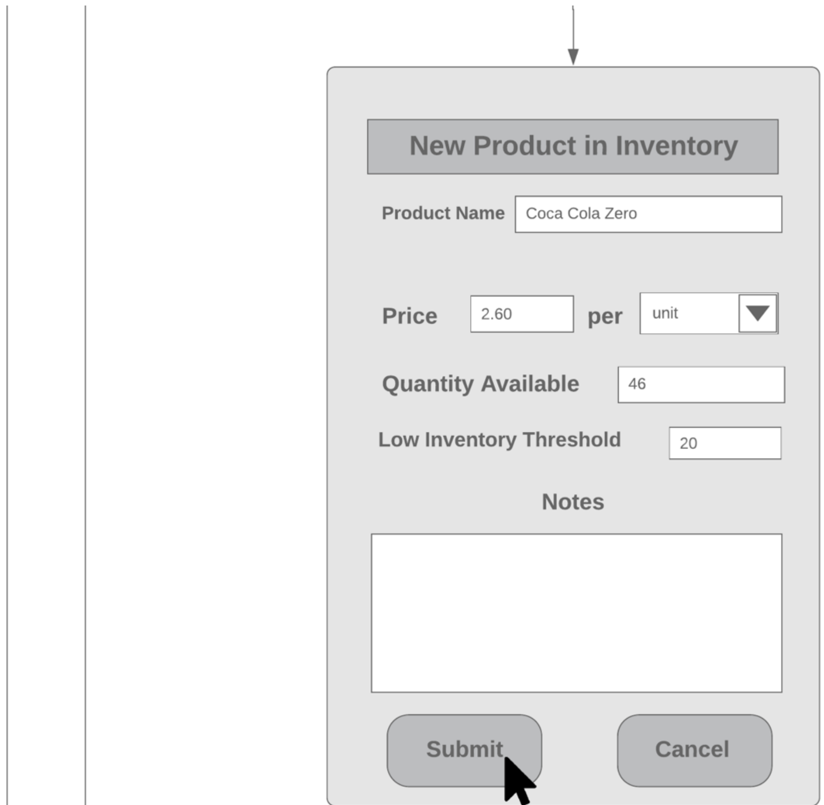
Step 3b.i:

The attendant makes an error by entering the '#' symbol into the price input field. The attendant clicks the "Submit" button to save the changes.

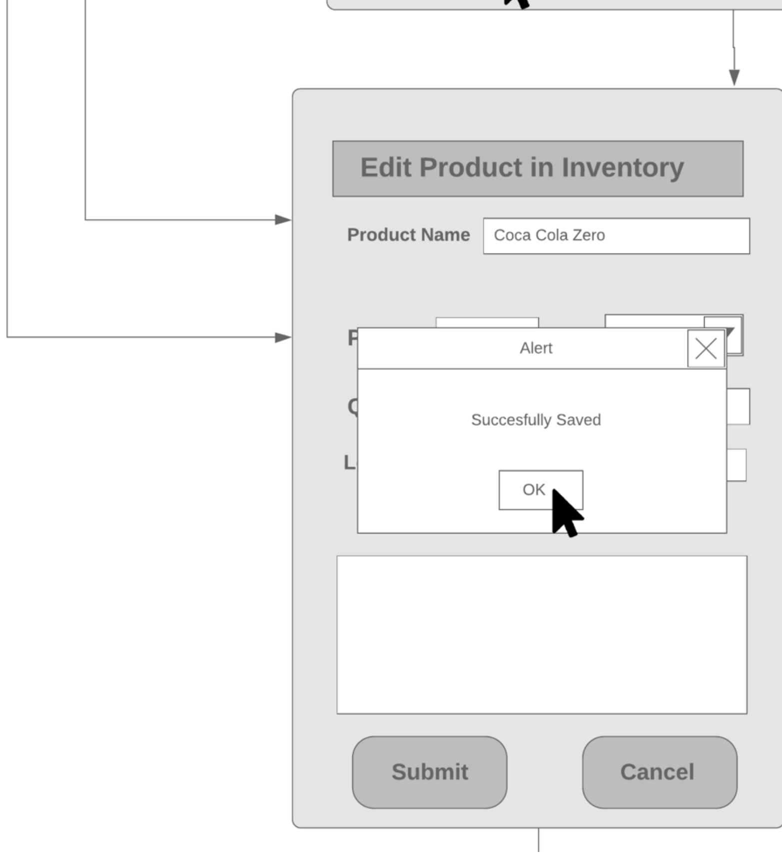



Step 3b.ii:

An alert is shown to the attendant with the error messages stating that the price is not a number. The attendant clicks the "OK" button to return close the alert and fix the mistakes.




Step 3b.iii:
The attendant removes the '#' symbol from the price input field. The attendant clicks the "Submit" button to save the changes.




Step 4:
A confirmation message is displayed to the attendant showing that the information has been saved. The user clicks "OK".

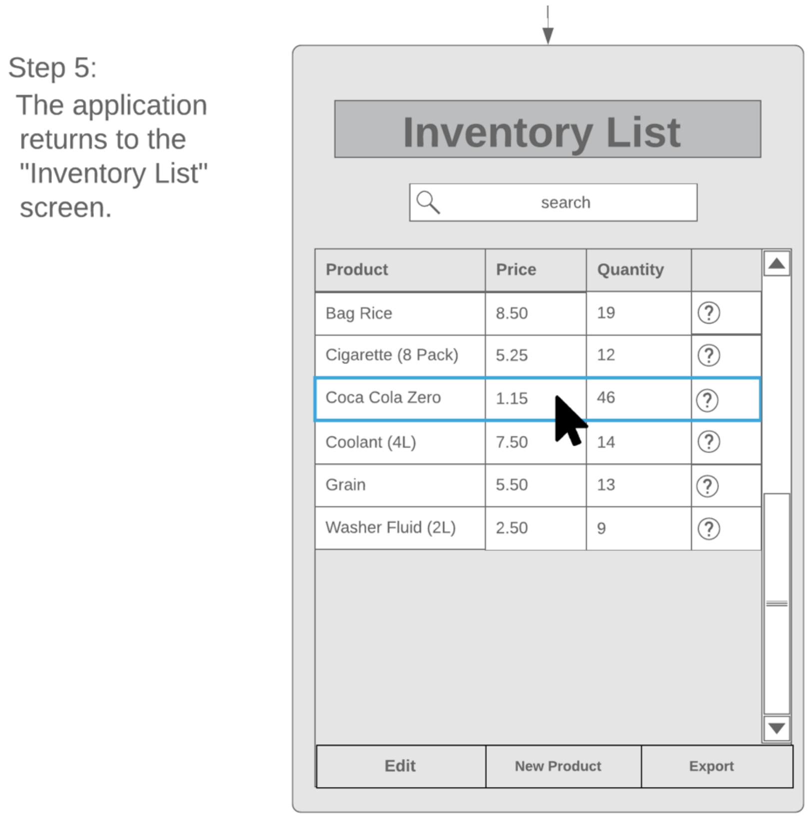



*Figure 4: Storyboard for editing a product in the Inventory Management subsystem*

### 6.1.3 Use Case: View Product [UC-03]

#### 6.1.3.1 Use Case Description: View Product [UC-03]

This use case details the steps taken by an attendant to view the product information currently stored in the Inventory Management subsystem for a product (see Table 3).

*Table 3: UC-03 Description - View Product*

| | |
|---|---|
| **Actors** | Attendant |
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has viewing privileges for the Inventory Management subsystem as set by an Administrator (see UC-07). |
| **Steps** | 1. The Attendant requests to enter the Inventory Management subsystem.<br>2. The Attendant is presented with<br>  • the option to add a product to the Inventory Management subsystem, |

| | |
|---|---|
| | • a list of products stored in the Inventory Management subsystem sorted by alphabetical order,<br>• for each product, the option to view all information associated with that product, and,<br>• the option to export information associated with each product in the Inventory Management subsystem.<br>3. The Attendant requests to view all information about a specific product.<br>4. The Attendant is presented with the following recorded values for the chosen product:<br>• product name<br>• price per unit (e.g. $1.99/kg)<br>• quantity<br>• low-quantity notification threshold<br>• notes |
| **Success Conditions** | • The Attendant views all product information associated with a specific product in the Inventory Management subsystem. |

### 6.1.3.2 Sequence Diagram: View Product [UC-03]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant views the product information stored in the Inventory Management subsystem for a product (see Figure 5). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Inventory Management subsystem.

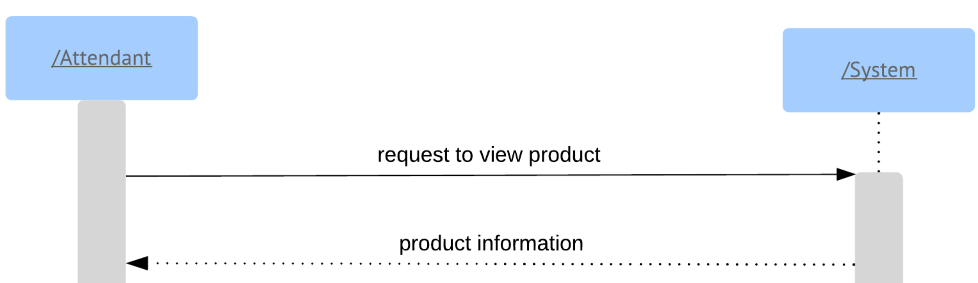
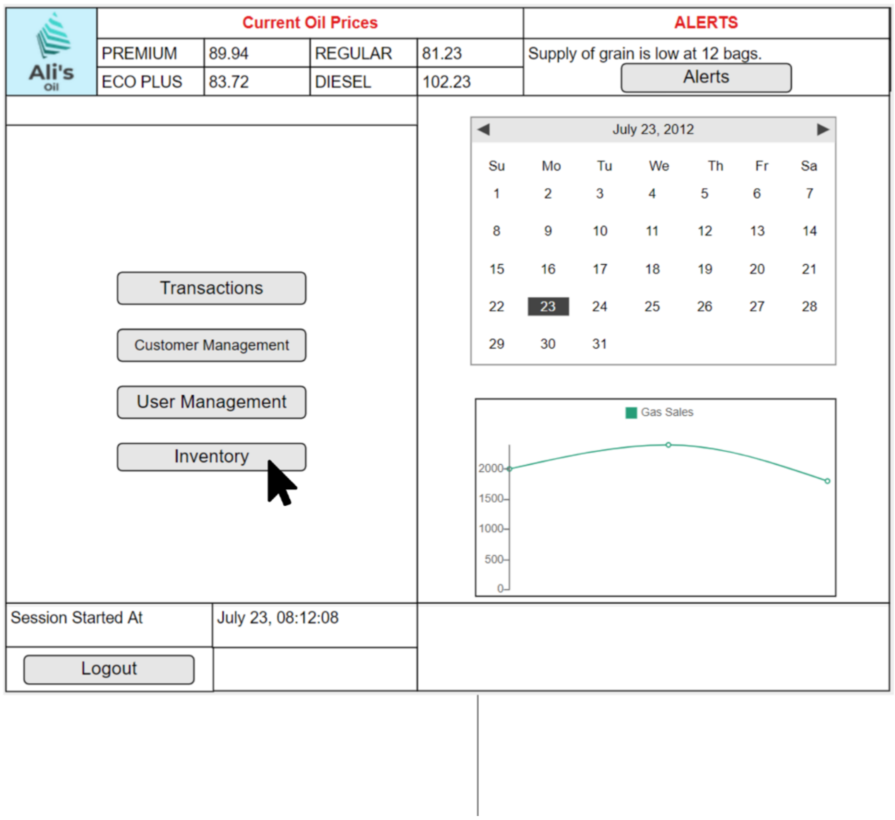



*Figure 5: Sequence diagram for viewing a product in the Inventory Management subsystem*

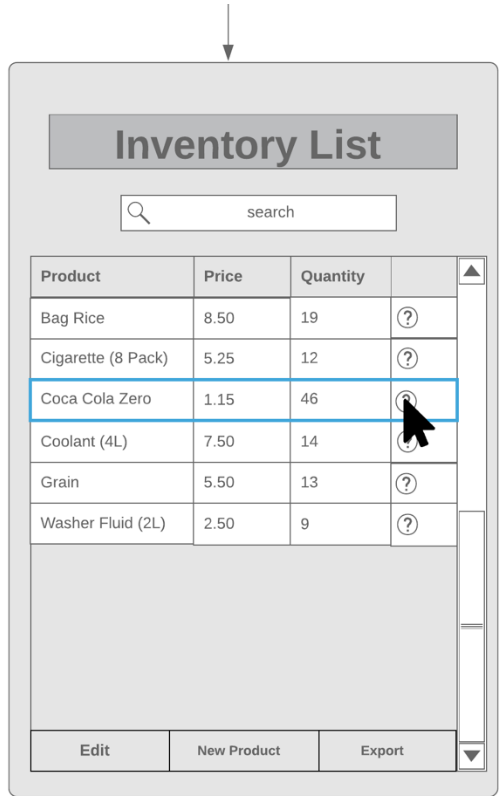
### 6.1.3.3 Storyboard: View Product [UC-03]

The following storyboard details a scenario in which an attendant is viewing the information associated with Coca Cola Zero (see Figure 6). Summer is approaching and demand for Coca Cola Zero is going to increase. An attendant wants to check the low inventory threshold for Coca Cola Zero to ensure it is high enough. The attendant wants to see the threshold value but not edit any information.

Step 1:
The attendant clicks the "Inventory" button on the home screen to access the inventory management screen.



Step 2:
The attendant finds the product in the inventory list screen and clicks the "?" button in the product row.




Step 3:
The attendant checks the low inventory threshold. It appears to be the right value. The attendant clicks "OK" to exit and return to the inventory list screen shown in Step 1.

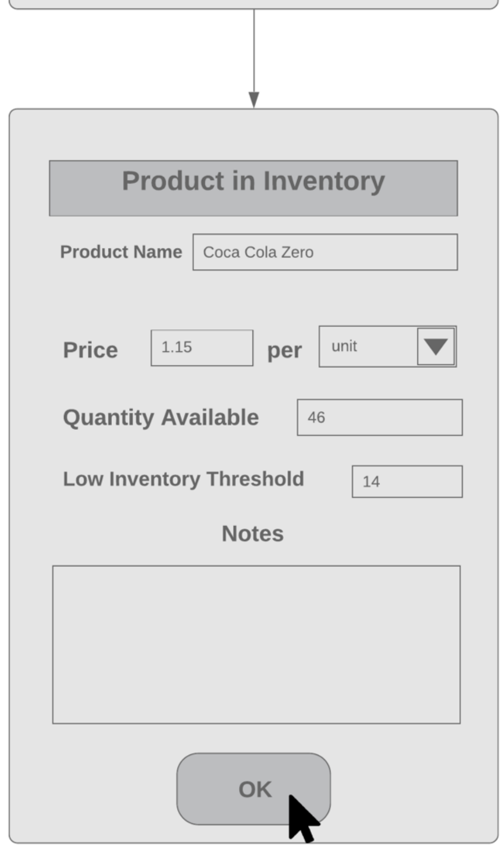



*Figure 6: Storyboard for viewing a product in the Inventory Management subsystem*

## 6.1.4 Use Case: Export Inventory [UC-04]

### 6.1.4.1 Use Case Description: Export Inventory [UC-04]

Allows an Attendant to export information on all inventory in the Inventory Management subsystem to a CSV file (see Table 4).

*Table 4: UC-04 Description - Export Inventory*

<table>
<tr><td>Actors</td><td>Attendant</td></tr>
<tr><td>Preconditions</td><td><ul><li>An Attendant is logged into AttendNet (see UC-05).</li><li>The Attendant has Inventory Management subsystem viewing privileges as set by an Administrator (see UC-07).</li></ul></td></tr>
<tr><td>Steps</td><td><ol><li>The Attendant requests to access the Inventory Management subsystem.</li><li>The Attendant is presented with<ul><li>the option to add a product to the Inventory Management subsystem,</li><li>a list of products stored in the Inventory Management subsystem sorted by alphabetical order,</li><li>for each product, the option to view all information associated with that product, and,</li><li>the option to export information associated with each product in the Inventory Management subsystem.</li></ul></li><li>The Attendant requests to export information associated with each product in the Inventory Management subsystem.</li></ol></td></tr>
<tr><td>Success Conditions</td><td><ul><li>A CSV file containing information for every transaction in the Transaction Management subsystem has been placed into the Downloads folder on the local machine.</li></ul></td></tr>
</table>

### 6.1.4.2 Sequence Diagram: Export Inventory [UC-04]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant exports the inventory information stored in the Inventory Management subsystem (see Figure 7). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Inventory Management subsystem.

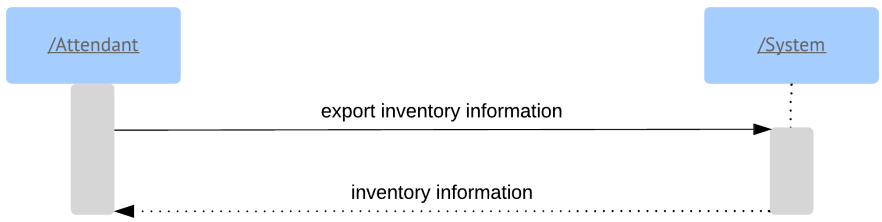
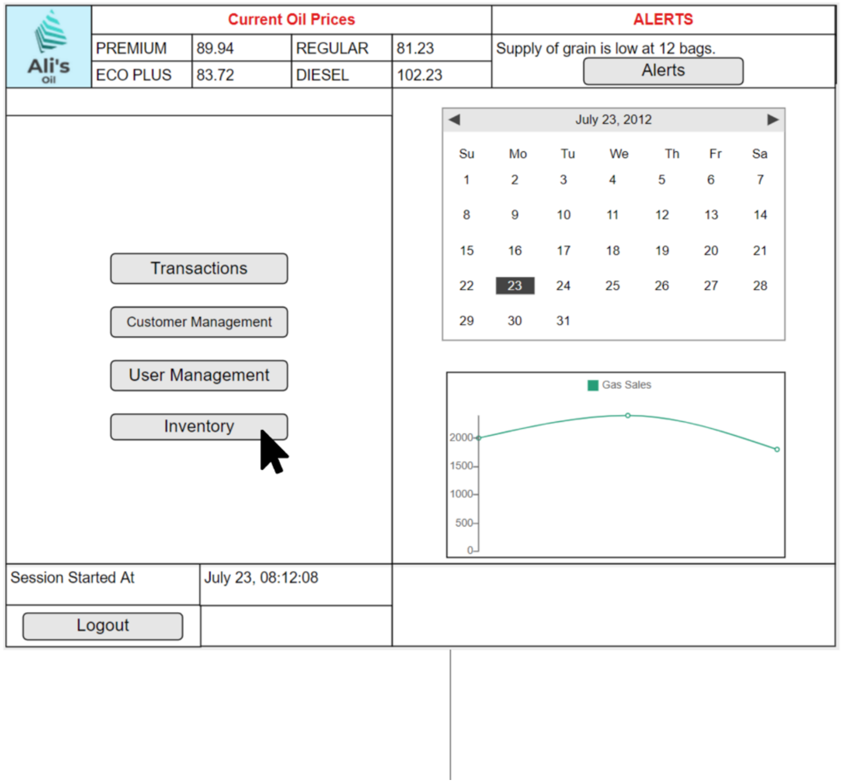



*Figure 7: Sequence Diagram for exporting AttendNet inventory information to a CSV file.*

## 6.1.4.3 Storyboard: Export Inventory [UC-04]

This storyboard details a scenario where an attendant exports the inventory information in the Inventory Management subsystem of AttendNet (see Figure 8).

Step 1:
The attendant clicks the "Inventory" button on the home screen to access the inventory management screen.



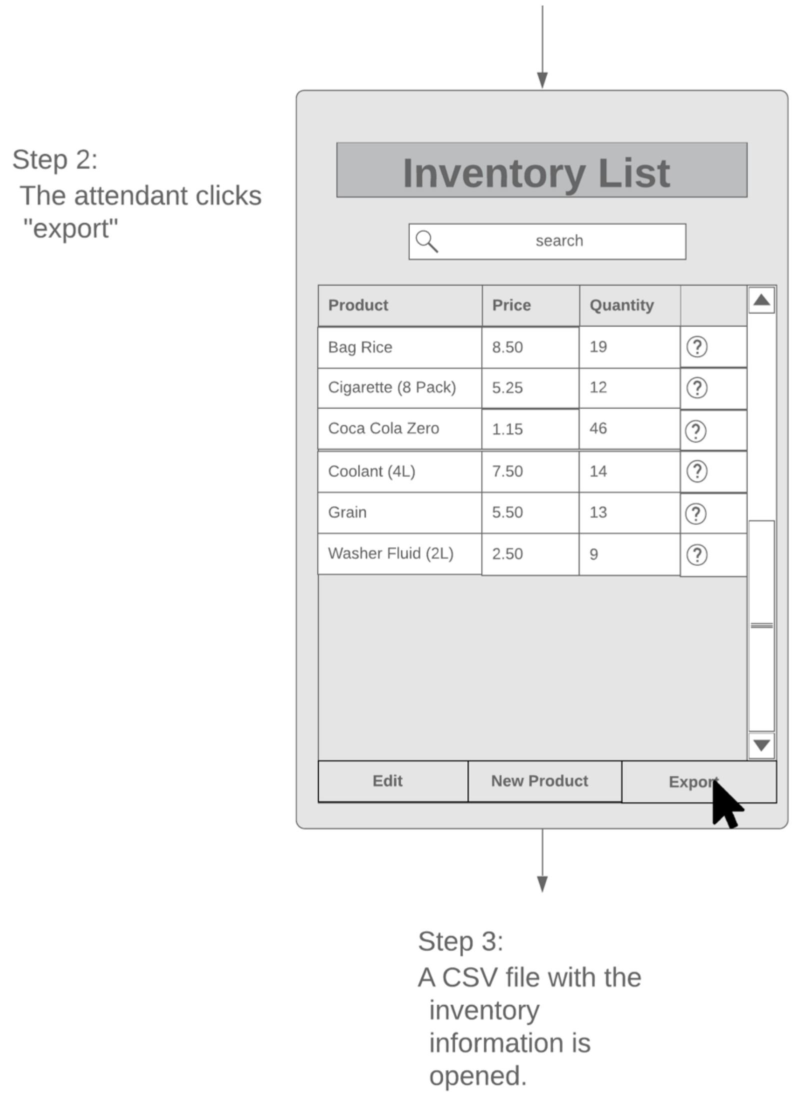



*Figure 8: Storyboard for exporting AttendNet inventory information to a CSV file.*

# 6.2 User Authentication

Sensitive data essential to operations at Ali's Oil, such as customer debt information, must be secure within AttendNet. To ensure confidentiality, integrity and availability, AttendNet safeguards sensitive data from potential intruders and eliminates security vulnerabilities by requiring user authentication.

## 6.2.1 Use Case: Login [UC-05]

### 6.2.1.1 Use Case Description: Login [UC-05]

This use case describes the steps an administrator or attendant would take to log into AttendNet. An administrator or attendant must be logged into AttendNet lefore being able to access AttendNet (see Figure 5).

*Table 5: UC-05 Description - Login*

| | |
|---|---|
| **Actors** | Administrator or Attendant |
| **Preconditions** | • An Administrator or Attendant exists in the User Management subsystem (see UC-07).<br>• The Administrator or Attendant is not logged into AttendNet. |
| **Steps** | 1. The Administrator or Attendant requests to log into AttendNet.<br>2. The Administrator or Attendant is presented with the ability to enter a username and personal identification number (PIN).<br>3. The Administrator or Attendant enters the following information:<br>  • administrator or attendant username<br>  • personal identification number (PIN)<br>4. The Administrator or Attendant is presented with confirmation that a session with privileges associated with the information entered in Step 3 has been instantiated. |
| **Success Conditions** | • The Administrator or Attendant has logged into AttendNet. |
| **Alternate Paths** | 3a.<br>I. The Administrator or Attendant enters a username or personal identification number (PIN) unassociated with any one Attendant or |

| | |
|---|---|
| | Administrator account in the User Management subsystem.<br>II. The Attendant is notified that the entered information was invalid for logging into AttendNet.<br>III. Continue with Step 3. |

### 6.2.1.2 Sequence Diagram: Login [UC-05]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant logs into AttendNet (see Figure 9). The preconditions for this sequence diagram are that the attendant has an account in the User Management subsystem and is not logged into AttendNet.

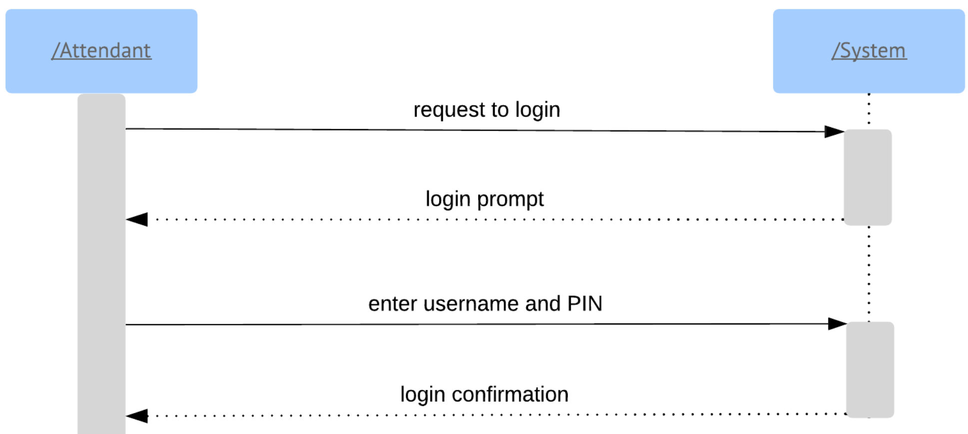



*Figure 9: Sequence diagram for logging into AttendNet*

## 6.2.1.3 Storyboard: Login [UC-05]

The following storyboard details a scenario in which an attendant is logging into AttendNet (see Figure 10). A customer wants to make a purchase. An attendant must access AttendNet in order to complete the transaction.

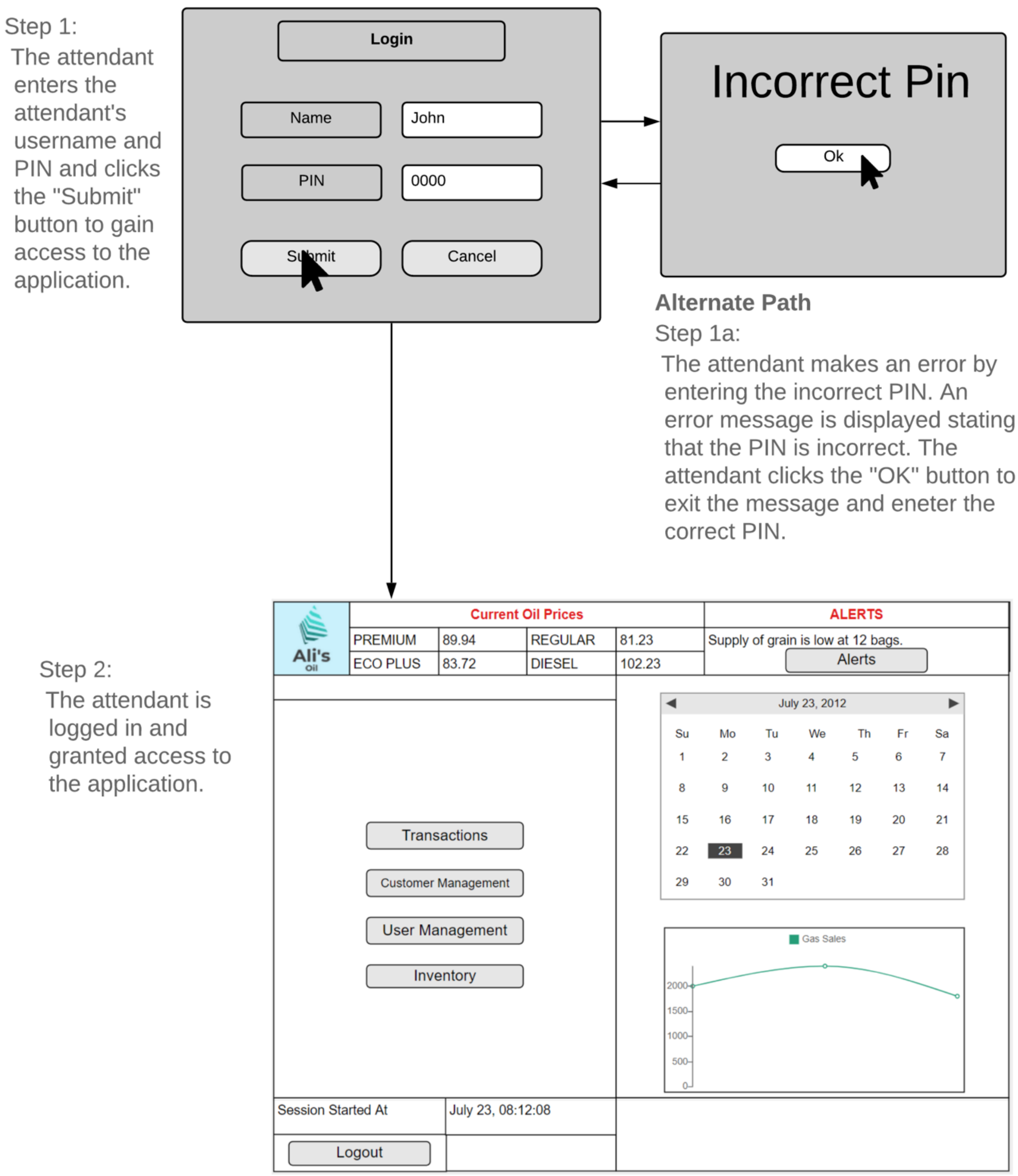



*Figure 10: Storyboard for logging into AttendNet*

## 6.2.2 Use Case: Logout [UC-06]

### 6.2.2.1 Use Case Description: Logout [UC-06]

The use case below describes the steps an administrator or attendant will take to logout of AttendNet (see Table 6). The administrator or attendant will not be able to access AttendNet until the administrator or attendent is logged into AttendNet again (see UC-05).

*Table 6: UC-06 Description - Logout*

| | |
|---|---|
| **Actors** | Administrator or Attendant |
| **Preconditions** | • An Administrator or Attendant is logged into AttendNet (see UC-05). |
| **Steps** | • The Administrator or Attendant requests to log out of AttendNet.<br>• The Administrator or Attendant is presented with a confirmation that the session has been terminated. |
| **Success Conditions** | • The Administrator or Attendant has logged out of AttendNet. |

### 6.2.2.2 Sequence Diagram: Logout [UC-06]

This sequence diagram illustrates the interactions that occur between an administrator or attendant and AttendNet when an administrator or attendant logs out of AttendNet (see Figure 11). As a precondition for this sequence diagram, the administrator or attendant must be logged into AttendNet.

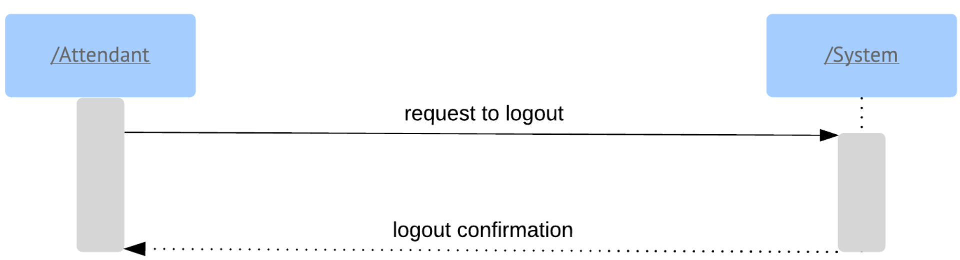



*Figure 11: Sequence diagram for logging out of AttendNet*

### 6.2.2.3 Storyboard: Logout [UC-06]

The following storyboard details a scenario in which an attendant is viewing the information associated with Coca Cola Zero (see Figure 12). The attendant wishes to take a break. The attendant wants to logout of AttendNet so no unauthorized users can gain access to it.

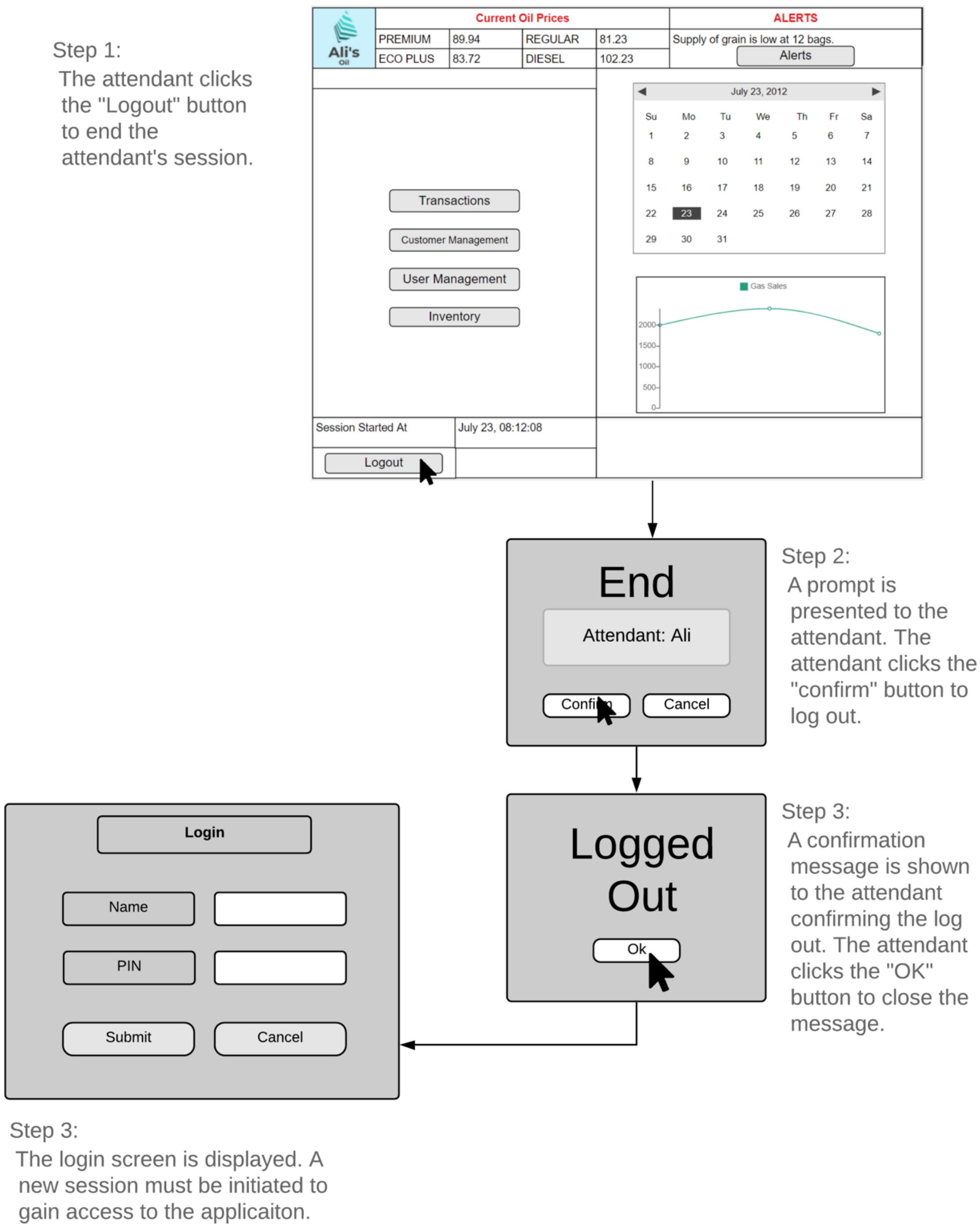



*Figure 12: Storyboard for logging out of AttendNet*

## 6.3 User Management

A discretionary access control authentication subsystem containing two user classes, administrators and attendants, must be provided by AttendNet. To accommodate the fluctuating number of attendants and administrators, a User Management subsystem must be provided by AttendNet.

### 6.3.1 Use Case: Add Administrator or Attendant [UC-07]

#### 6.3.1.1 Use Case Description: Add Administrator or Attendant [UC-07]

This use case describes the steps an administrator would take to add an administrator or an attendant account to the User Management subsystem (see Table 7).

*Table 7: UC-07 Description - Add Administrator or Attendant*

| **Actors** | Administrator |
|---|---|
| **Preconditions** | • An Administrator is logged into AttendNet (see UC-05). |
| **Steps** | 1. The Administrator requests to enter the User Management subsystem.<br>2. The Administrator is presented with all Administrator and Attendant accounts in the User Management subsystem and the option to add an Administrator or an Attendant to the User Management subsystem.<br>3. The Administrator requests to add an Administrator or Attendant to the User Management subsystem.<br>4. The Administrator is presented with the ability to enter information for an Administrator or Attendant and save that information in the User Management subsystem.<br>5. The Administrator enters account specific information for the Administrator or Attendant account.<br>6. The Administrator requests to save the entered Administrator or Attendant and information associated with the entered Administrator or Attendant in the User Management subsystem.<br>7. The Administrator is presented with confirmation that the entered Administrator or Attendant account and information associated with the entered Administrator or Attendant |

| | |
|---|---|
| | account has been saved in the User Management subsystem. |
| **Success Conditions** | • An administrator or attendant account with account specific information exists within AttendNet. |

### 6.3.1.2 Sequence Diagram: Add Administrator or Attendant [UC-07]

This sequence diagram illustrates the interactions that occur between an administrator and AttendNet when an administrator adds an attendant or an administrator account to the User Management subsystem (see Figure 13). The preconditions for this sequence diagram are that the administrator has logged into AttendNet and has entered the User Management subsystem.

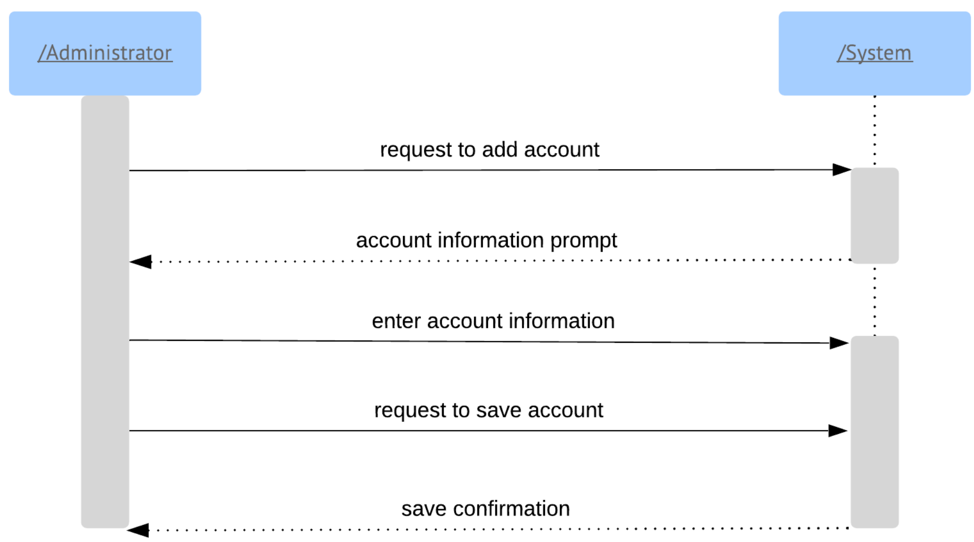
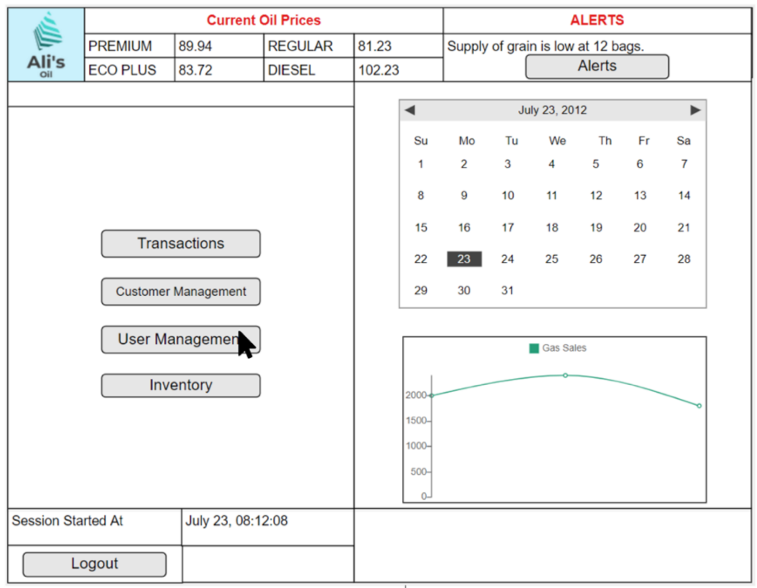



*Figure 13: Sequence diagram for adding an attendant or administrator to the User Management subsystem*

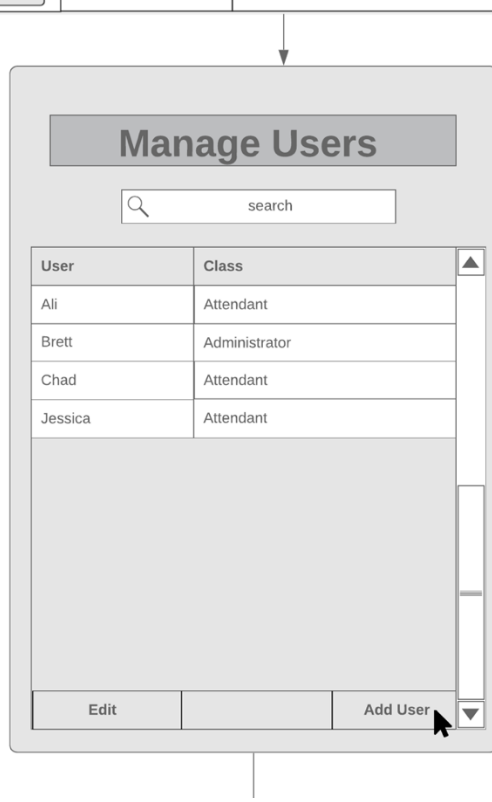
### 6.3.1.3 Storyboard: Add Administrator or Attendant [UC-07]

The following storyboard details a scenario in which an administrator is adding an attendant to the User Management subsystem (see Figure 14). A new attendant named John has been hired at Ali's Oil. An administrator must create an attendant account for John to use AttendNet. The administrator wishes to give John permissions to both edit and view the Transaction Management and Inventory Management subsystems.

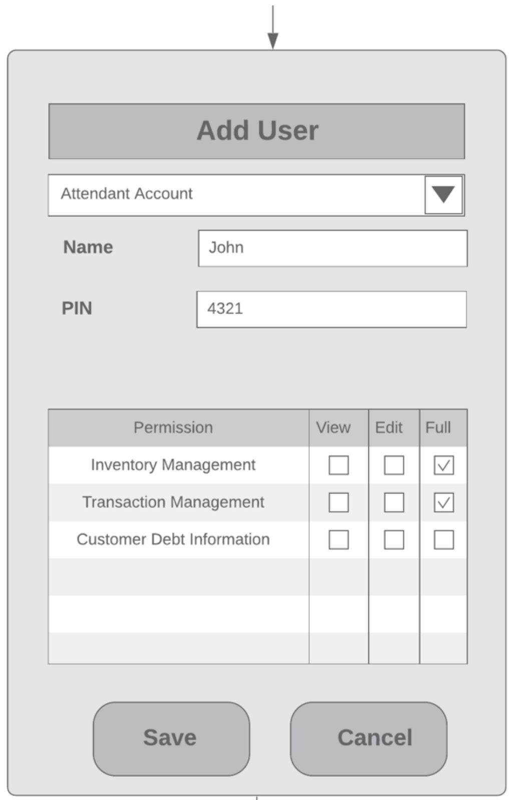
Step 1:
The administrator clicks the "User Management" button to enter the user management screen.



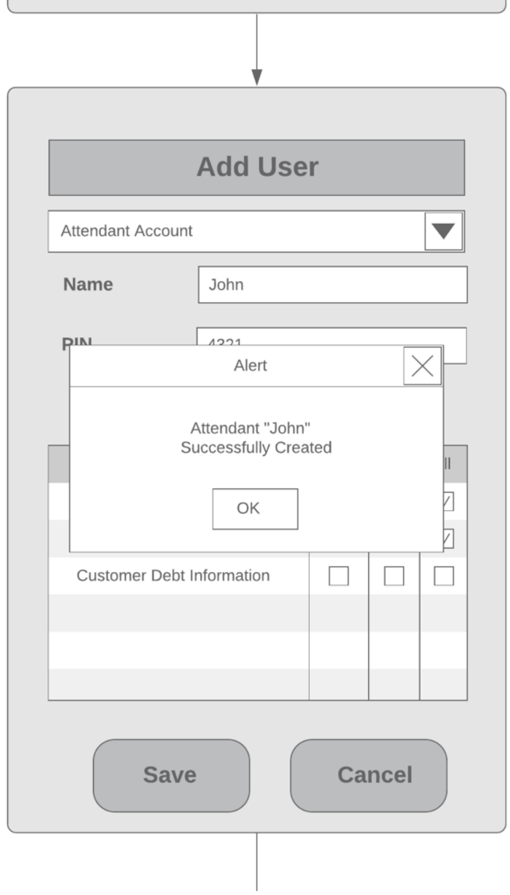

Step 2:
The manager clicks the "Add User" button in the user management screen.



Step 3:
The administrator enters the new username and PIN in the appropriate input fields. Then the administrator checks the "Full" box for inventory and transaction managment to give John the permissions required to manage transactions and inventory. Finally, the administrator clicks the "Save" button to create the new attendant.




Step 4:
The administrator is shown a confirmation message stating that the attendant was successfully created. The administrator clicks OK to the user management screen.



Step 5:
The new attendant "John" now appears in the user management screen.

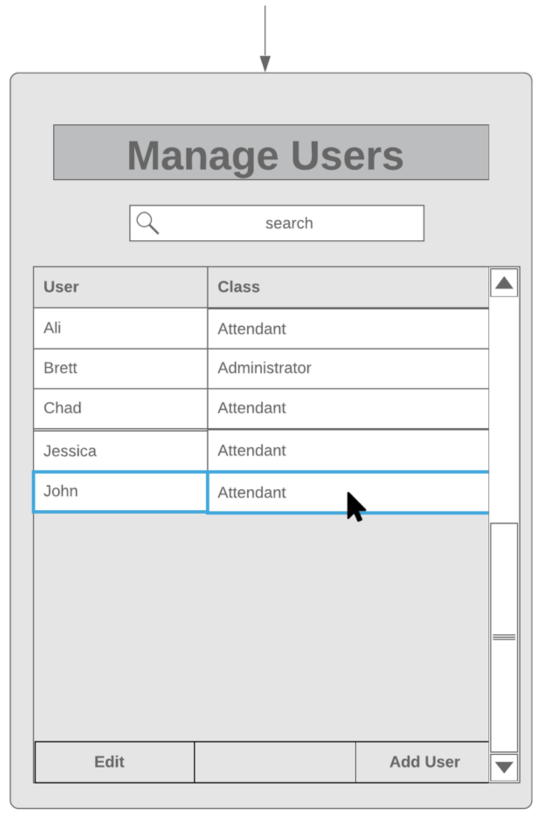



*Figure 14: Storyboard for adding an attendant or administrator to the User Management subsystem*

## 6.3.2 Use Case: Edit Administrator or Attendant [UC-08]

### 6.3.2.1 Use Case Description: Edit Administrator or Attendant [UC-08]

The following use case details the steps required for an administrator to modify an existing administrator or an existing attendant account in the User Management subsystem (see Table 8).

*Table 8: UC-08 Description- Edit Administrator or Attendant*

| | |
|---|---|
| **Actors** | Administrator |
| **Preconditions** | • An Administrator is logged into AttendNet (see UC-05).<br>• The Administrator or Attendant account to be edited exists in the User Management subsystem. |
| **Steps** | 1. The Administrator requests to enter the User Management subsystem.<br>2. The Administrator is presented with all Administrator and Attendant accounts in the |

| | |
|---|---|
| | User Management subsystem and the option to view all information associated with an Administrator or Attendant account in the User Management subsystem.<br>3. The Administrator requests to view all information associated with an Administrator or an Attendant account.<br>4. The Administrator is presented with all information associated with the Administrator or Attendant account selected in Step 3 and the option to edit the Administrator or Attendant account.<br>5. The Administrator requests to edit the Administrator or Attendant account.<br>6. The Administrator is presented with the ability to edit the Administrator or Attendant account.<br>7. The Administrator edits any of the following information associated with the Administrator or Attendant account:<br>• administrator or attendant name<br>• personal identification number (PIN)<br>*IF* editing Attendant information, the Administrator edits the access to any one of the subsystems to "none", "view only", or "view and edit" access. The subsystems are as follows:<br>• Inventory Management<br>• Transaction Management<br>• Customer Debt Management<br>8. The Administrator requests to save the edited Administrator or Attendant account.<br>9. The Administrator is presented with confirmation that the edited Administrator or Attendant account has been updated in the User Management subsystem. |
| **Success Conditions** | • The Administrator or Attendant account is updated and saved in the User Management subsystem to reflect the changes made by the Administrator. |

### 6.3.2.2 Sequence Diagram: Edit Administrator or Attendant [UC-08]

This sequence diagram illustrates the interactions that occur between an administrator and AttendNet when an administrator edits an existing attendant or administrator account in the User Management subsystem (see Figure 15). The preconditions for this sequence diagram are that the administrator has logged into AttendNet and has entered the User Management subsystem.

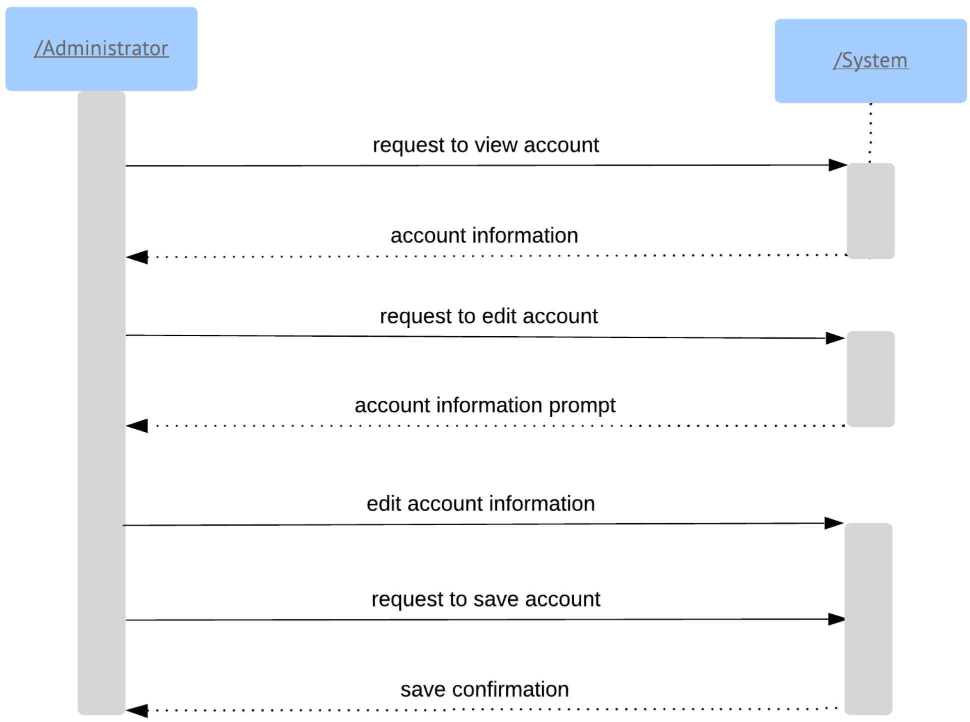
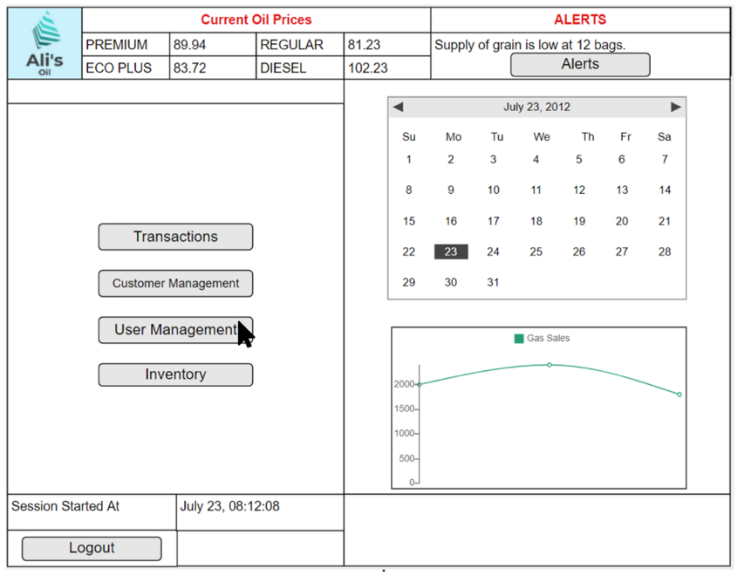



*Figure 15: Sequence diagram for editing an administrator or attendant account*

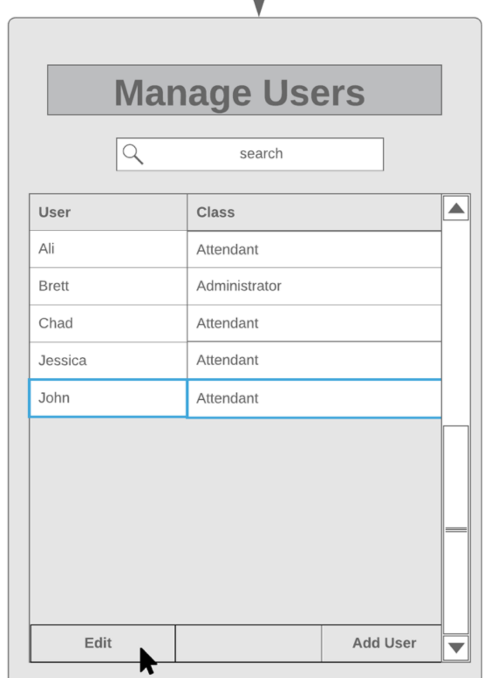
### 6.3.2.3 Storyboard: Edit Administrator or Attendant [UC-08]

The following storyboard details a scenario in which an administrator is updating the information associated with an attendant account (see Figure 16). An administrator has a lot of work and wants to delegate the work of exporting transaction data and inventory data to an attendant. The administrator decides that an employee named John should be the one to export data. The manager changes John's permissions to enable exporting of transactions and inventory.

Step 1:
The administrator clicks the "User Management" button to enter the user management screen.




Step 2:
The administrator selects John from the user list and clicks the "Edit" button to edit his permissions.



Step 3:
The administrator checks the "Full" checkbox for "Inventory Management" and "Transaction Management". The attendant then clicks the "Save" button to save the changes.

Step 4:
The administrator is shown a confirmation message stating that the user was successfully updated. The administrator clicks OK to exit the user management screen and return to the user list.

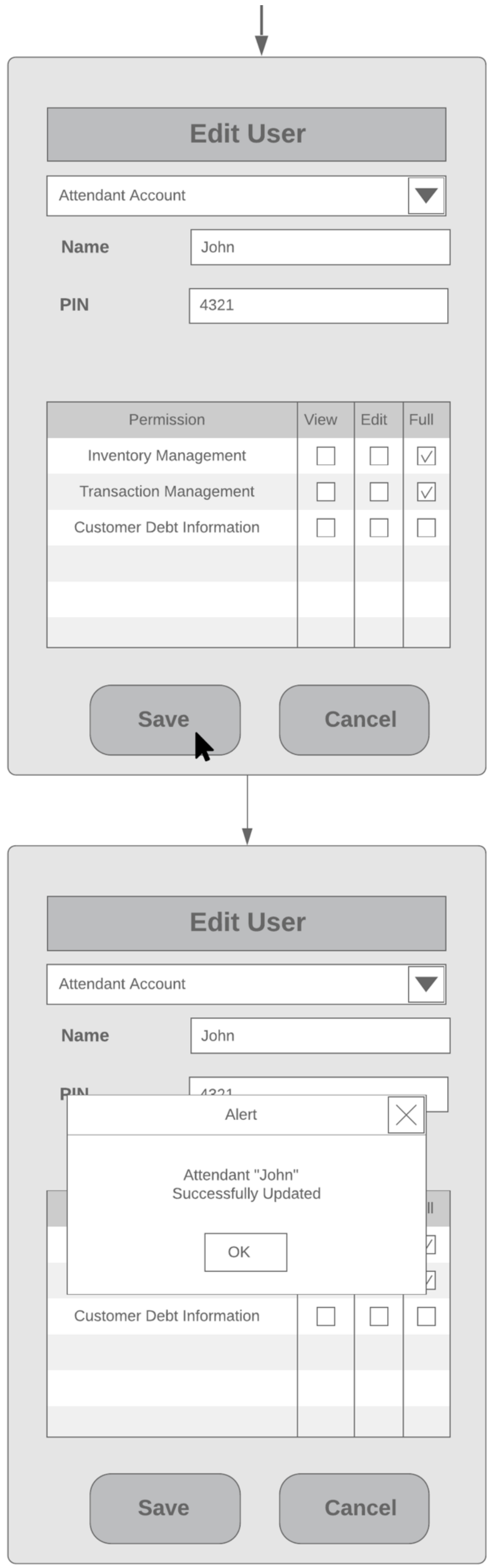



*Figure 16: Storyboard for editing an administrator or attendant account*

## 6.4 Notifications

Ali's Oil productivity relies upon attendant efficiency. To ensure attendant efficiency, a notification subsystem triggered on events such as low inventory for a product or a fuel price update from Government of Pakistan Oil and Gas Regulatory Authority must be provided by AttendNet.

### 6.4.1 Use Case: View Notification [UC-09]

#### 6.4.1.1 Use Case Description: View Notification [UC-09]

The Notification subsystem notifies attendants that a product in the Inventory Management subsystem is below the low-quantity threshold or, that the current price of fuel in the Inventory Management subsystem has been changed (see Table 9).

*Table 9: UC-09 Description - View Notification*

| **Actors** | Attendant |
|---|---|
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05).<br>• The recorded quantity for a product in the Inventory Management subsystem is a value below the same product's specified low-quantity threshold *OR* the recorded price for a fuel product in the Inventory Management subsystem has been changed.<br>• AttendNet has generated a low-quantity notification for a product designated as being of low-quantity *OR* AttendNet has generated a fuel-price-update notification for a fuel product with a price that has been changed. |
| **Steps** | 1. The Attendant requests to enter the Notifications subsystem.<br>2. The Attendant is presented with<br>    • a limited amount of notifications ordered by time of creation,<br>    • the option to request more notifications be displayed, and,<br>    • for each notification, the option to view all information associated with that notification.<br>3. The Attendant requests to view all information associated with a specific notification. |

| | |
|---|---|
| | 4. The Attendant is presented with the time of creation for the chosen notification, as well as the message that notification contains. |
| **Success Conditions** | • The Attendant has viewed the information associated with a fuel-price-update notification or a low-quantity notification. |

### 6.4.1.2 Sequence Diagram: View Notification [UC-09]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant views a notification in the Notifications subsystem (see Figure 17). The preconditions for this sequence diagram include the attendant having logged into AttendNet, as well as a product in the Inventory Management subsystem having reached its low-quantity threshold or the recorded price for a fuel product in the Inventory Management subsystem having been changed.

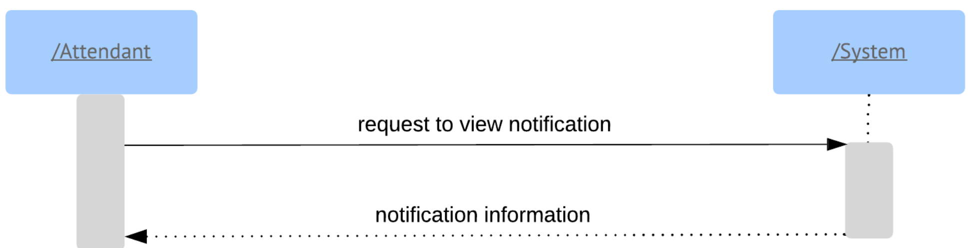
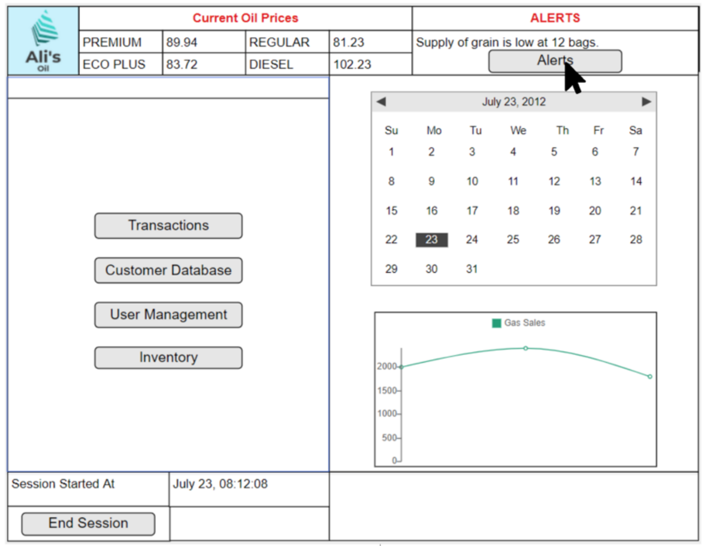



*Figure 17: Sequence diagram for viewing a notification in the Notifications subsystem*

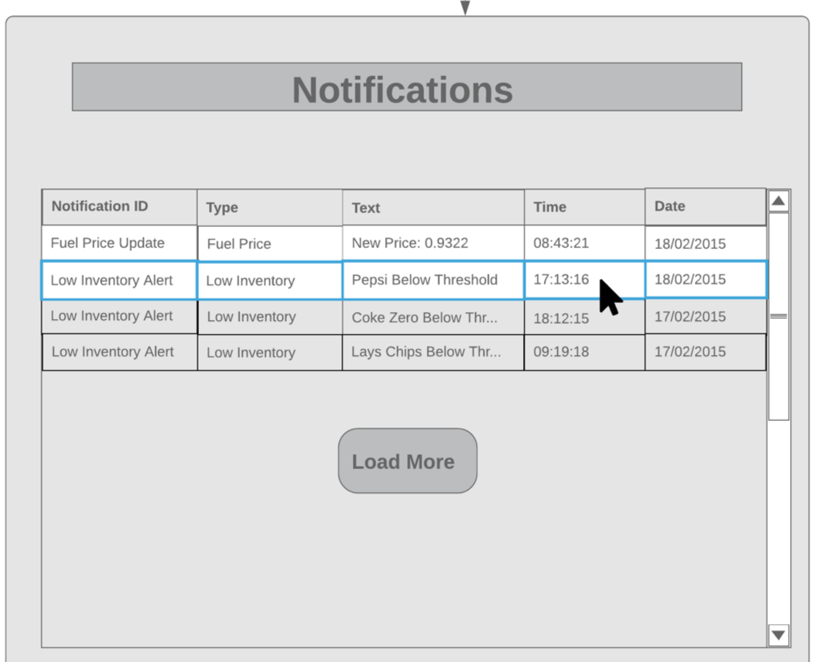
## 6.4.1.3 Storyboard: View Notification [UC-09]

The following storyboard details a scenario in which an attendant is viewing a notification (see Figure 18). An attendant the wants to ensure that no products are in short supply. The attendant goes into AttendNet notifications to check if AttendNet has given a warning notification for any products that has fallen below the low inventory threshold.

Step 1:
The attendant clicks the "Alerts" button to access the notifications screen.




Step 2:
In the notification screen, the attendant sees that there are two notifications. One notification is low inventory alert for Pepsi, meaning that the Pepsi in the inventory has fallen below the threshold value. The attendant double-clicks the notification to open it.



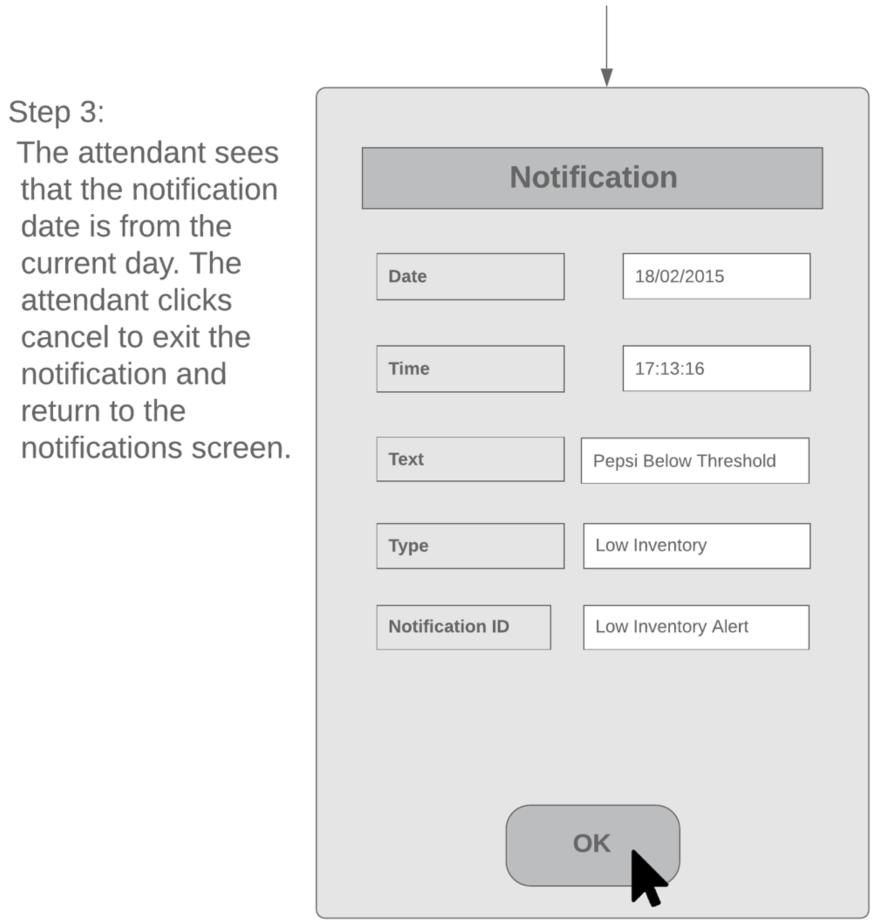



*Figure 18: Storyboard for viewing a notification in the Notifications subsystem*

## 6.5 Transaction Management

Ali's Oil records all transactions that take place. To ensure all transactions are recorded correctly and consistently, transaction management must be provided by AttendNet.

## 6.5.1 Use Case: Add Transaction [UC-10]

### 6.5.1.1 Use Case Description: Add Transaction [UC-10]

To facilitate record-keeping of transactions in AttendNet, the user must enter the transaction information into the Transaction Management subsystem (see Figure 10).

*Table 10: UC-10 Description - Add Transaction*

| **Actors** | Attendant |
|---|---|
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05: Login).<br>• The Attendant has privileges to edit the Transaction Management subsystem as specified by an Administrator (see UC-07: Add Administrator or Attendant). |
| **Steps** | 1. The Attendant requests to enter the Transaction Management subsystem.<br>2. The Attendant is presented with<br>  • the option to add a transaction to the Transaction Management subsystem,<br>  • a list of transactions stored in the Transaction Management subsystem sorted in order by creation date,<br>  • for each transaction, the option to view all information associated with that transaction, and,<br>  • the option to export information associated with each transaction in the Transaction Management subsystem.<br>3. The Attendant requests to add a transaction to the Transaction Management subsystem.<br>4. The Attendant is presented with the ability to enter information for a transaction and the option to save the transaction in the Transaction Management subsystem.<br>5. The Attendant enters the following transaction information:<br>  • amount paid<br>  • amount given in change<br>  • product(s) sold (optional)<br>  • for each entered product, the quantity exchanged |

| | |
|---|---|
| | • customer name (optional)<br>• notes (optional)<br>6. The Attendant requests to save the information in the Transaction management subsystem.<br>7. The Attendant is presented with confirmation that the transaction was saved in the Transaction Management subsystem. |
| **Success Conditions** | • A record of the entered transaction exists within the Transaction Management subsystem.<br>• The quantity recorded in the Inventory Management subsystem for each product specified in the entered transaction has been changed to reflect the respective quantity exchanged for each product. |
| **Alternate Paths** | 5a.<br>I. The Attendant enters invalid denominations of rupees for amount paid *AND*/*OR* amount given in change.<br>II. The Attendant is notified that the values entered for amount paid *AND*/*OR* amount given in change is not a valid denomination of rupees and of the correct format for an amount paid and amount given in change.<br>III. The Attendant enters the amount paid *AND*/*OR* the amount given in change as a valid denomination of rupees.<br>IV. Continue with Step 6.<br><br>5b.<br>I. The Attendant enters values for the amount paid and amount given in change and the sum of the value entered for amount paid, the value entered for amount given in change, and the calculated monetary value of the total product exchanged does not equal zero, but a customer name has not been entered.<br>II. The Attendant is notified that a customer debt must be updated, but a customer name has not been entered.<br>III. The Attendant enters a customer name.<br>IV. Continue with Step 6. |

| | |
|---|---|
| | 5c.<br>I. The Attendant enters a customer name that is associated with multiple different customer records in the Customer Debt Management subsystem.<br>II. The Attendant is presented with the address associated with each duplicate customer name in the Customer Debt Management subsystem.<br>III. The Attendant selects the correct customer information from the customer information displayed.<br>IV. Continue with Step 5. |

### 6.5.1.2 Sequence Diagram: Add Transaction [UC-10]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant adds a transaction to the Transaction Management subsystem (see Figure 19). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Transaction Management subsystem.

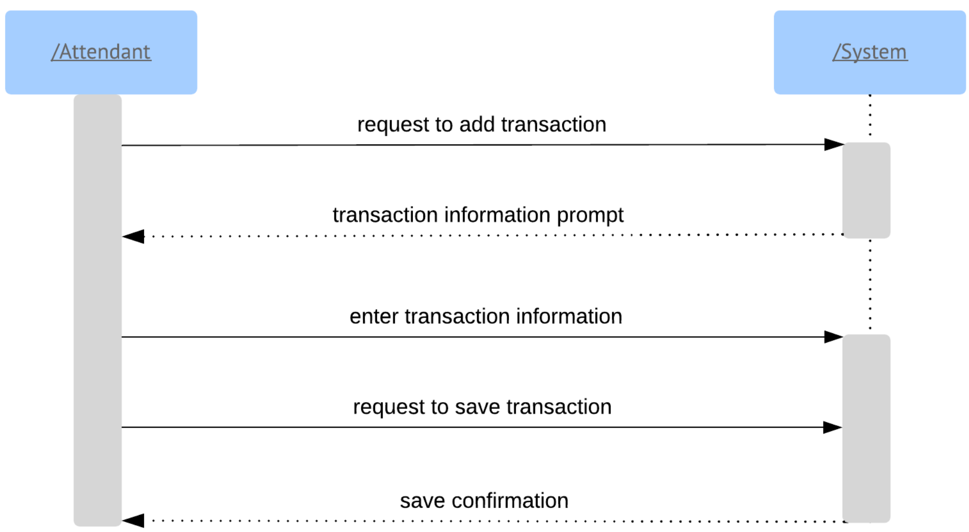
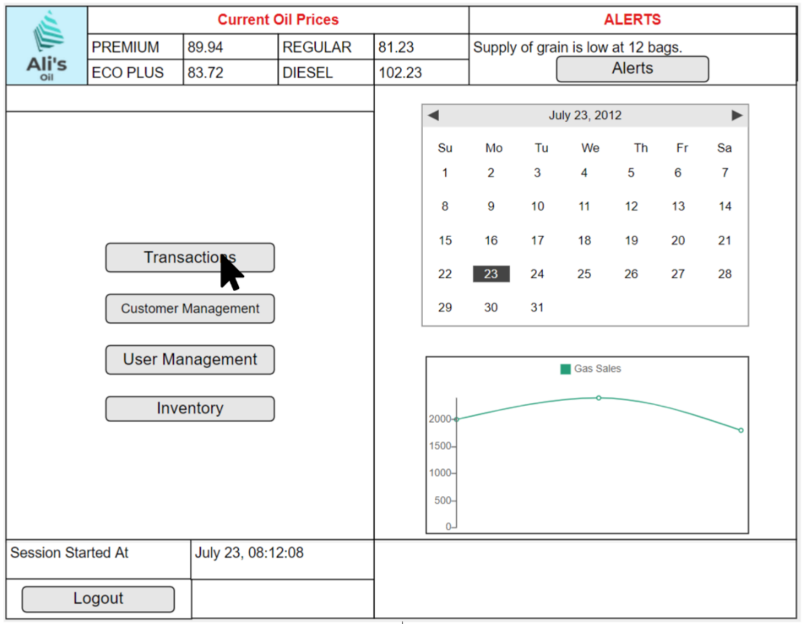



*Figure 19: Sequence diagram for adding a transaction to the Transaction Management subsystem*

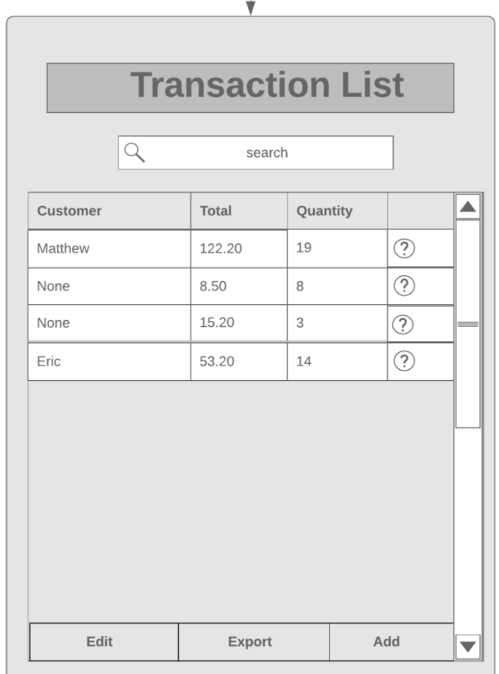
## 6.5.1.3 Storyboard: Add Transaction [UC-10]

The storyboard below details a scenario in which a customer purchases goods from Ali's Oil (see Figure 20). An attendant processes the transaction by entering transaction details into the Transaction Management subsystem user interface and submitting it for storage. The process details the navigation of views, and the fields to be entered by the attendant.

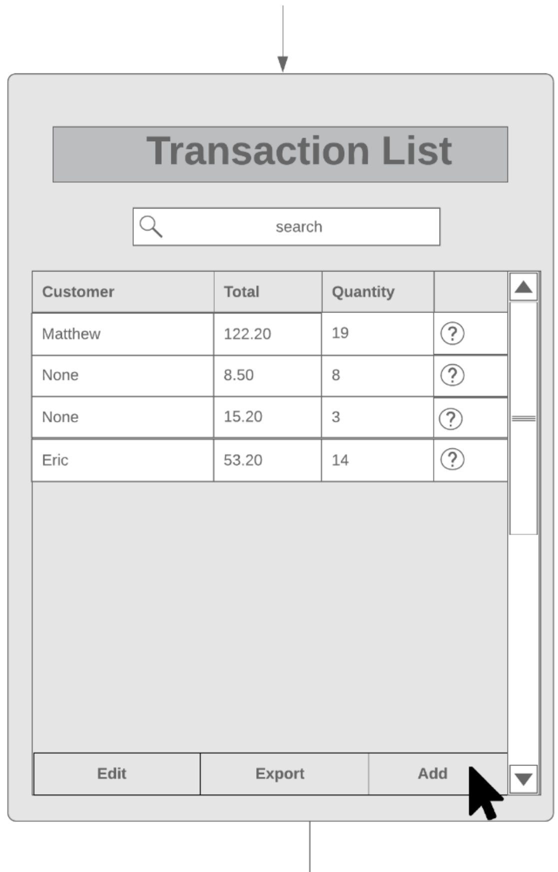
Step 1:
The attendant opens the transaction management screen by clicking the "Transactions" button.



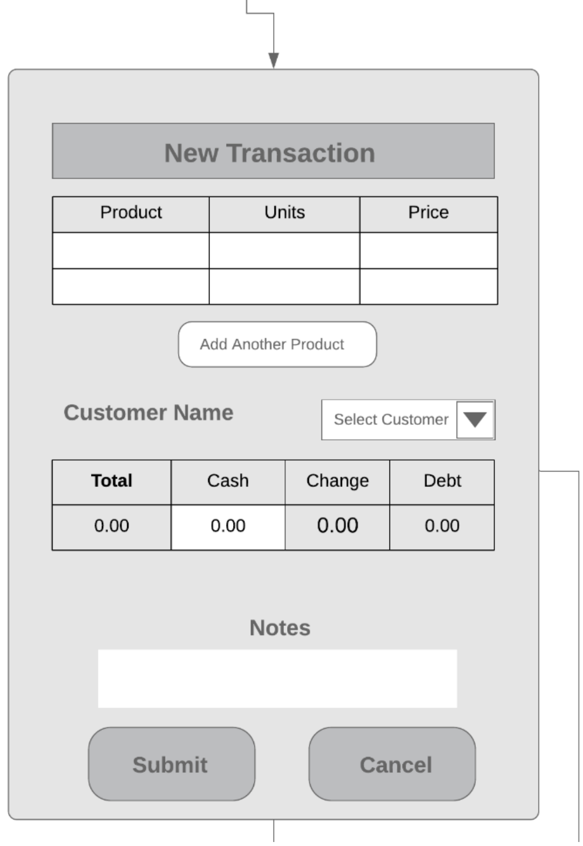

Step 2:
The attendant can view a number of transactions and request more with the scroll bar.



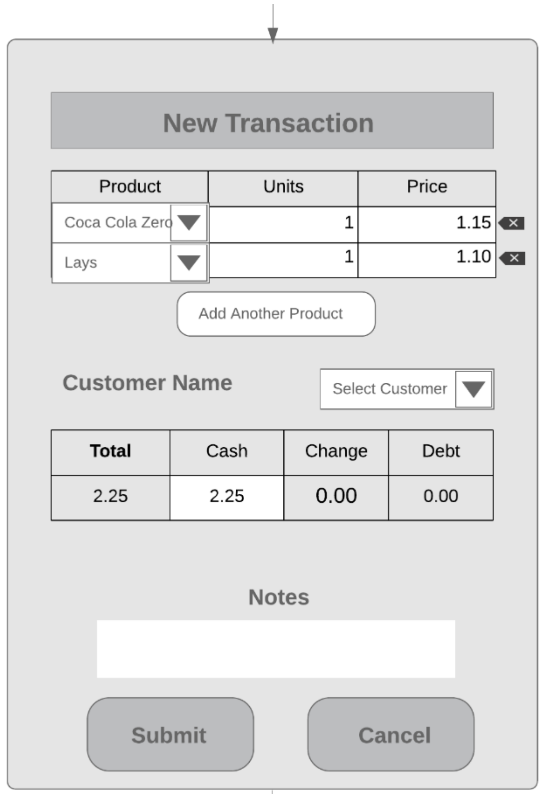
Step 3:
The attendant clicks the "Add" button to add a new transaction.



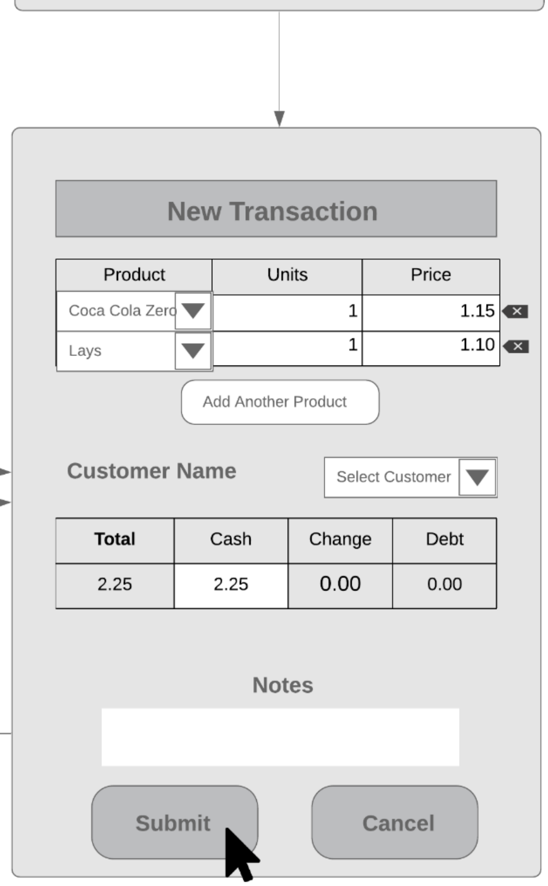

Step 4:
The attendant views an empty transaction record.



Step 5:
The attendant selects the products that the customer wishes to purchase. The total cost is shown in the "Total" field, which 2.25 in this case. The customer pays in full with cash.

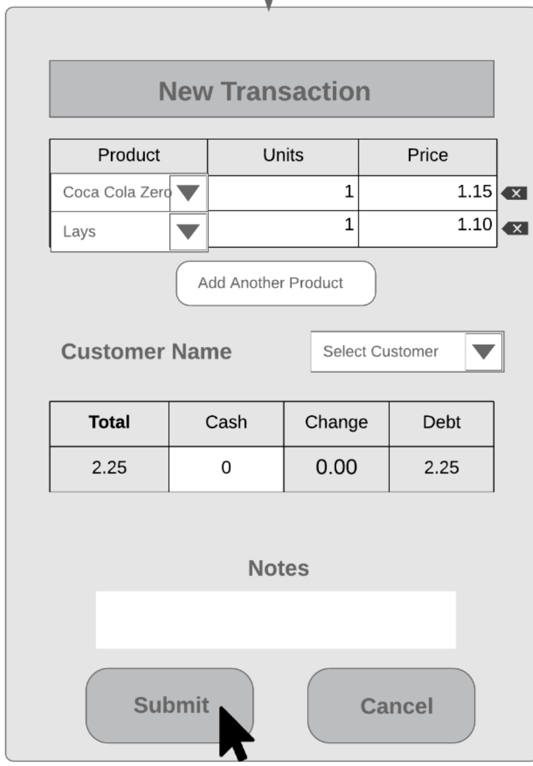



Step 6:
The attendant clicks the "Submit" button to complete the transaction.




Step 5b.iii

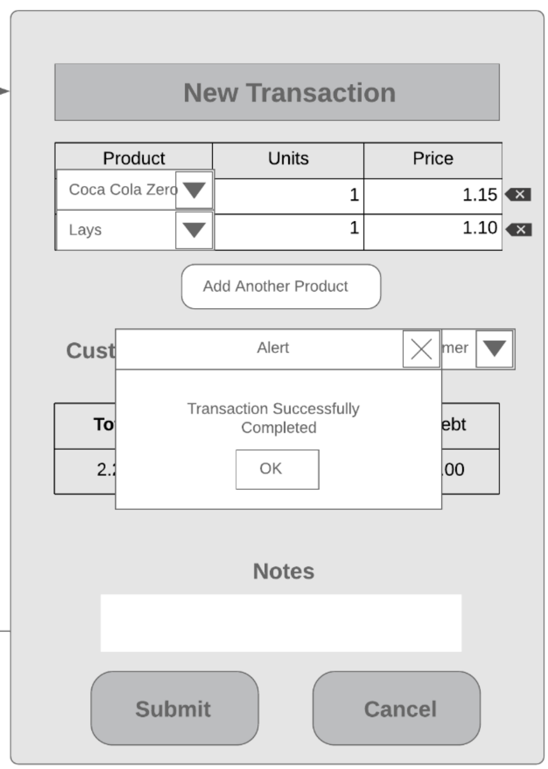
Step 5a.iii




**Alternate Path**

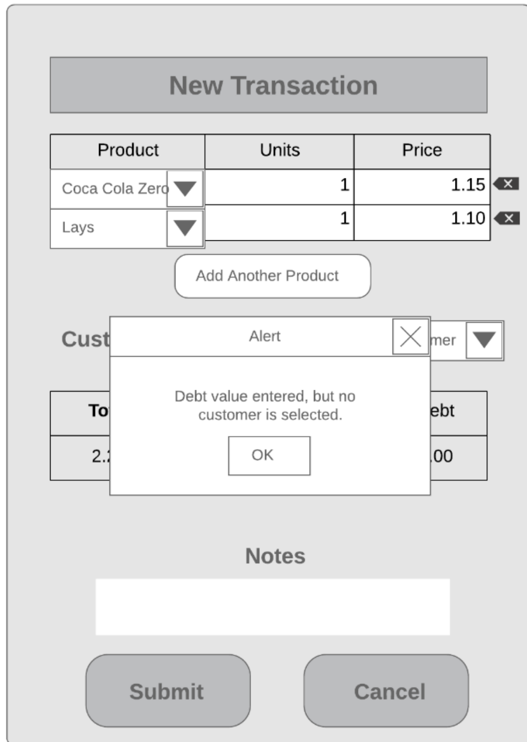
Step 5a.i:

The attendant makes an error, where there is a customer debt value, but no customer selected.




Step 7:

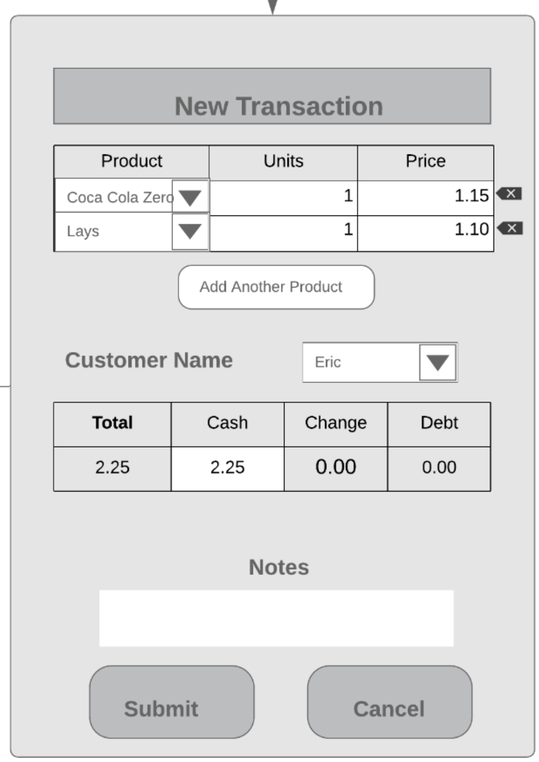
An alert appears telling the attendant that the transaction was successfully completed. The attendant clicks "OK" to return to the transaction list screen.




**Alternate Path**

Step 5a.ii:

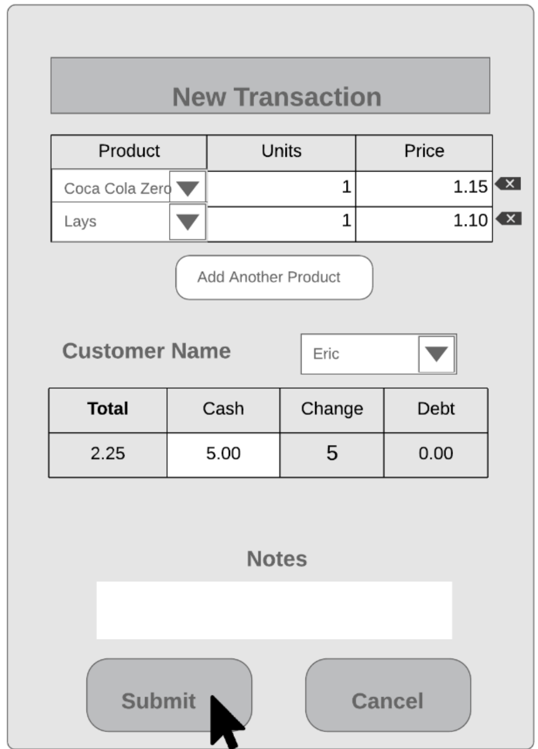
The attendant is presented with an alert that the transaction has not been completed, and the attendant needs to enter a customer name.



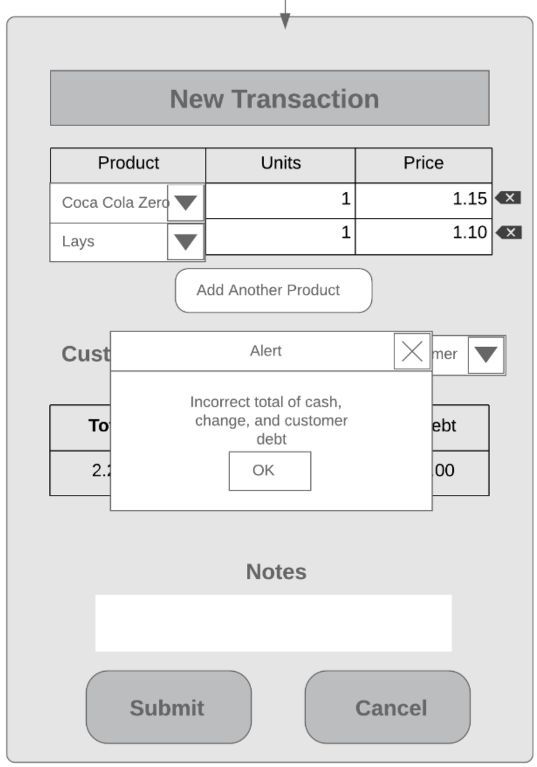

**Alternate Path**

Step 5a.iii:

The attendant completes the "Customer Name" entry. Proceed with step 6.




**Alternate Path**
Step 5b.i:
The attendant enterred an incorrect combination of cash, change and debt, and tries to submit.




**Alternate Path**
Step 5b.ii:
The attendant is presented with an error message.

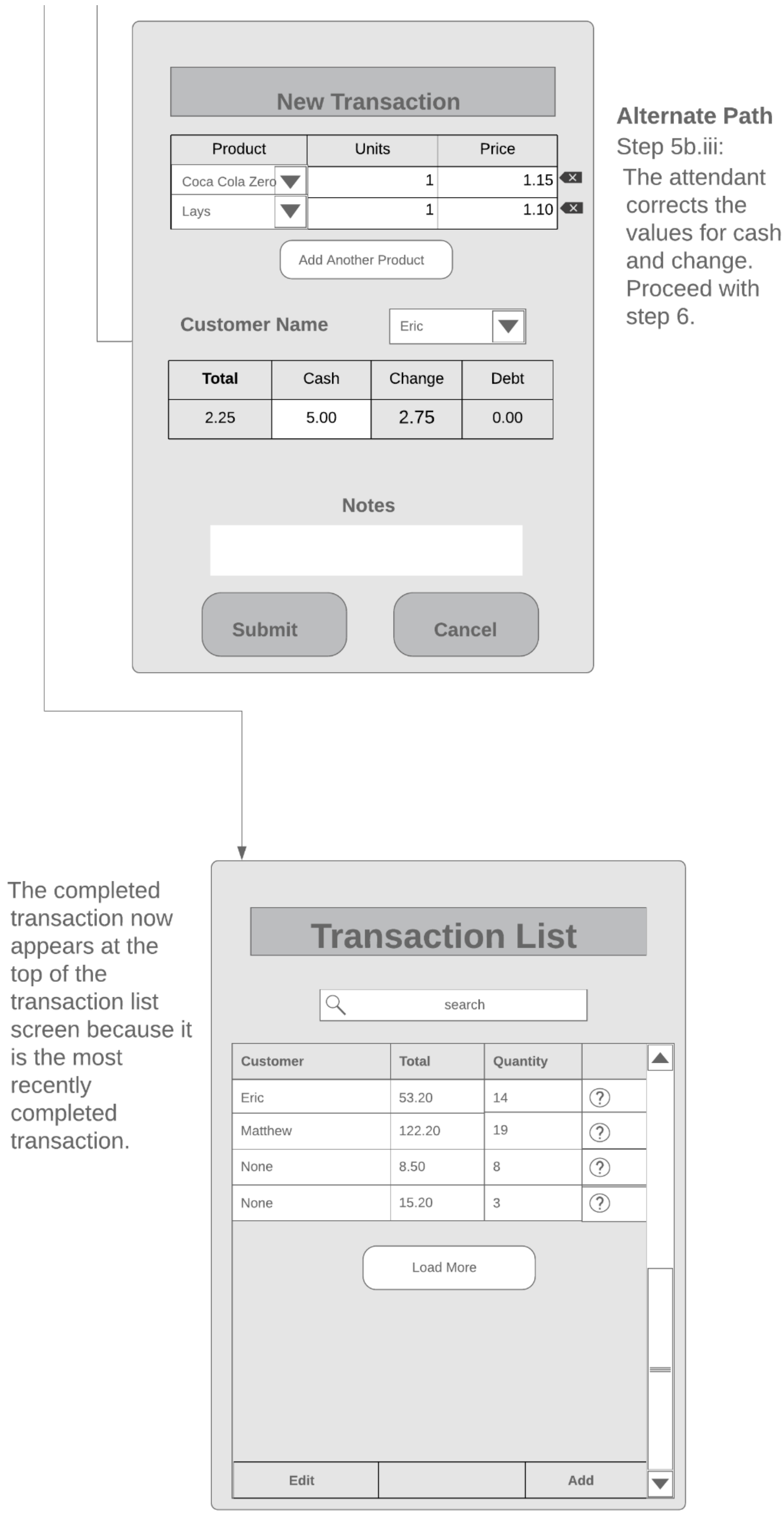



*Figure 20: Storyboard for adding a transaction to the Transaction Management subsystem*

## 6.5.2 Use Case: Edit Transaction [UC-11]

### 6.5.2.1 Use Case Description: Edit Transaction [UC-11]

Due to human error, such as incorrectly entering transaction information, a transaction record stored in the Transaction Management subsystem may need editing. In the situation that an attendant has added a transaction with incorrect transaction information, an attendant may edit the information associated with the incorrectly completed transaction (see Table 11).

*Table 11: UC-11 Description - Edit Transaction*

| **Actors** | Attendant |
|---|---|
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has editing privileges for the Transaction Management subsystem as set by an Administrator (see UC-07). |
| **Steps** | 1. The Attendant requests to enter the Transaction Management subsystem.<br>2. The Attendant is presented with<br>• the option to add a transaction to the Transaction Management subsystem,<br>• a list of transactions stored in the Transaction Management subsystem sorted in order by creation date,<br>• for each transaction, the option to view all information associated with that transaction, and,<br>• the option to export information associated with each transaction in the Transaction Management subsystem.<br>3. The Attendant requests to view all information associated with a transaction.<br>4. The Attendant is presented with all information associated with the transaction selected in Step 3 and the option to edit that transaction.<br>5. The Attendant requests to edit the transaction.<br>6. For each of the following transaction information fields, the Attendant enters a value if the existing saved value for the respective field needs to be changed:<br>• amount paid<br>• amount given in change |

| | |
|---|---|
| | • product(s) sold (optional)<br>• for each entered product, the quantity exchanged<br>• customer name (optional)<br>7. The Attendant requests to save the transaction in the Transaction Management subsystem.<br>8. The Attendant is presented with confirmation that the transaction and its associated information was saved in the Transaction Management subsystem successfully. |
| **Success Conditions** | • The transaction information for the transaction has been saved in the Transaction Management subsystem.<br>• If the change(s) made to the transaction information included product information, the respective product information associated with the transaction has been updated in the Inventory Management subsystem.<br>• If a customer is associated with the transaction and the change(s) made to the transaction information resulted in a different contribution to that customer debt, the respective customer debt has been updated in the Customer Debt Management subsystem. |
| **Alternate Paths** | 6a.<br>I. The Attendant enters invalid denominations of rupees for amount paid *AND*/*OR* amount given in change.<br>II. The Attendant is notified that the values entered for amount paid *AND*/*OR* amount given in change is not a valid denomination of rupees and of the correct format for an amount paid and amount given in change.<br>III. The Attendant enters the amount paid *AND*/*OR* the amount given in change as a valid denomination of rupees.<br>IV. Continue with Step 7.<br><br>6b.<br>I. The Attendant enters values for the amount paid and amount given in change and the sum of the |

| | |
|---|---|
| | value entered for amount paid, the value entered for amount given in change, and the calculated monetary value of the total product exchanged does not equal zero, but a customer name has not been entered.<br>II. The Attendant is notified that a customer debt must be updated, but a customer name has not been entered.<br>III. The Attendant enters a customer name.<br>IV. Continue with Step 7.<br><br>8c.<br>I. The Attendant enters a customer name that is associated with multiple different customer records in the Customer Debt Management subsystem.<br>II. The Attendant is presented with the business name and address associated with each duplicate customer name in the Customer Debt Management subsystem.<br>III. The Attendant selects the correct customer information from the customer information displayed.<br>IV. Continue with Step 5. |

### 6.5.2.2 Sequence Diagram: Edit Transaction [UC-11]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant edits a transaction in the Transaction Management subsystem (see Figure 21). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Transaction Management subsystem.

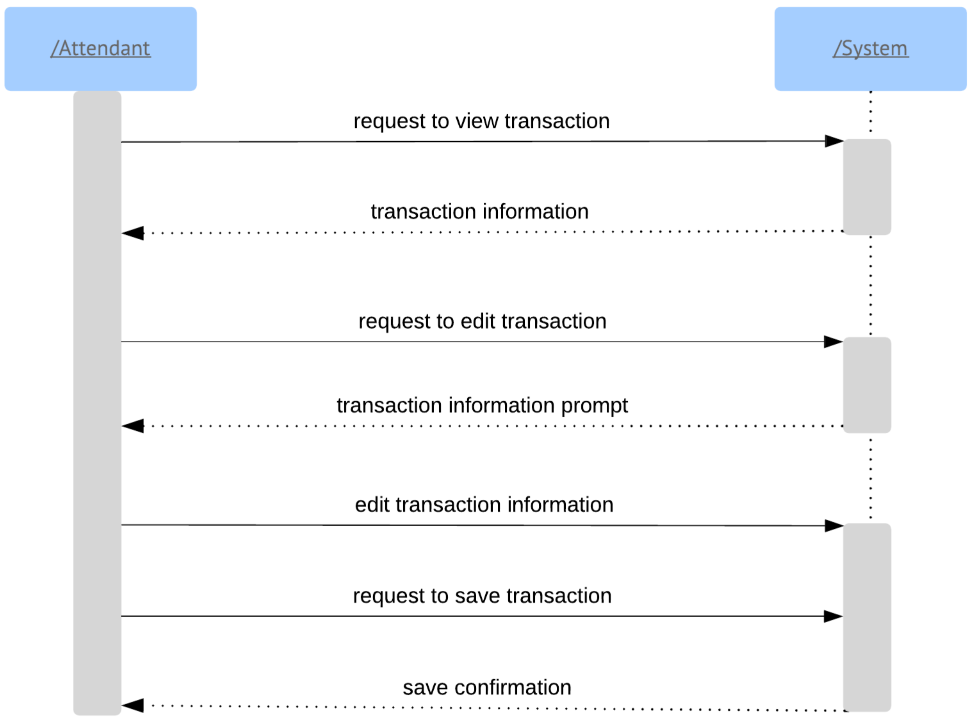
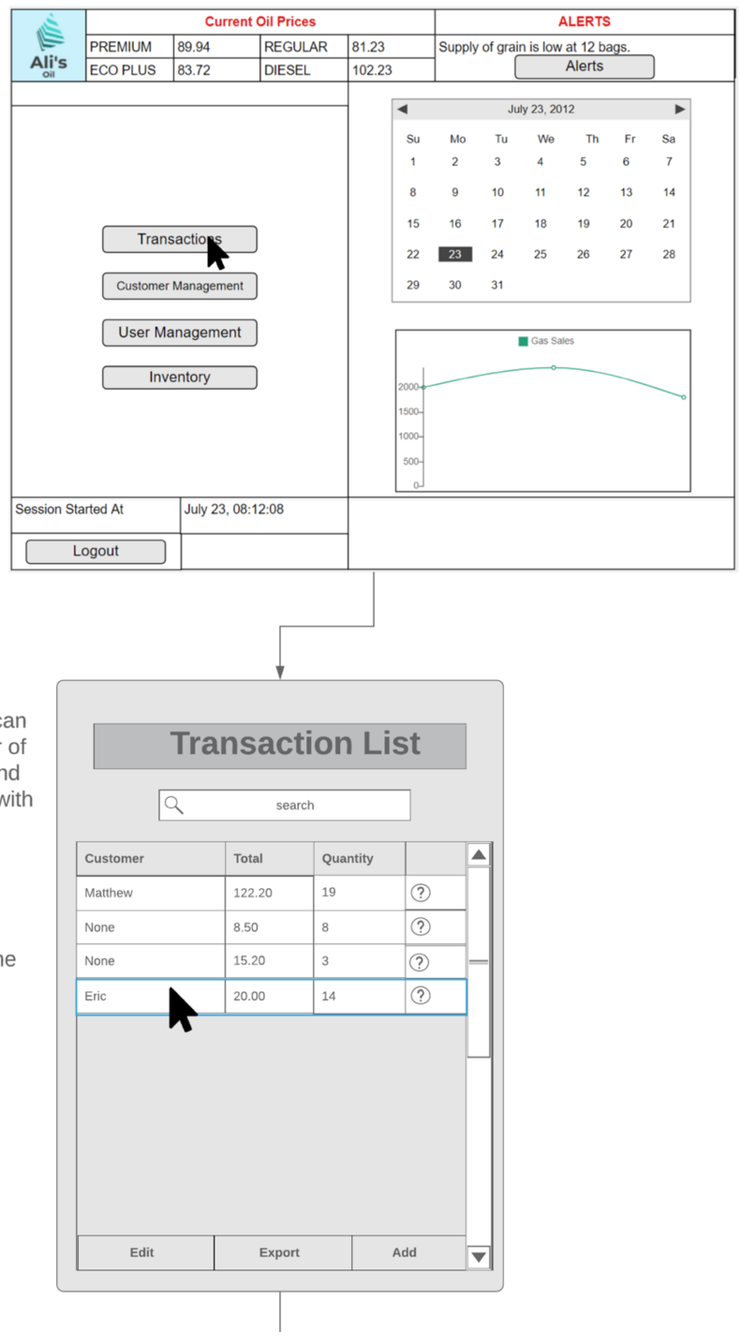



*Figure 21: Sequence diagram for editing a transaction in the Transaction Management subsystem*

### 6.5.2.3 Storyboard: Edit Transaction [UC-11]

The following storyboard details a scenario in which an attendant is updating the information associated with a transaction (see Figure 22). A customer named "Eric" has made a purchase within the last 5 minutes. The attendant who completed the transaction realizes that the amount paid by the customer has been entered into AttendNet incorrectly. The attendant wrongly entered that the customer paid 20.00 for the purchase, when the customer paid 21.00. The attendant wants to fix this error. The attendant must apply the fix by editing the transaction.

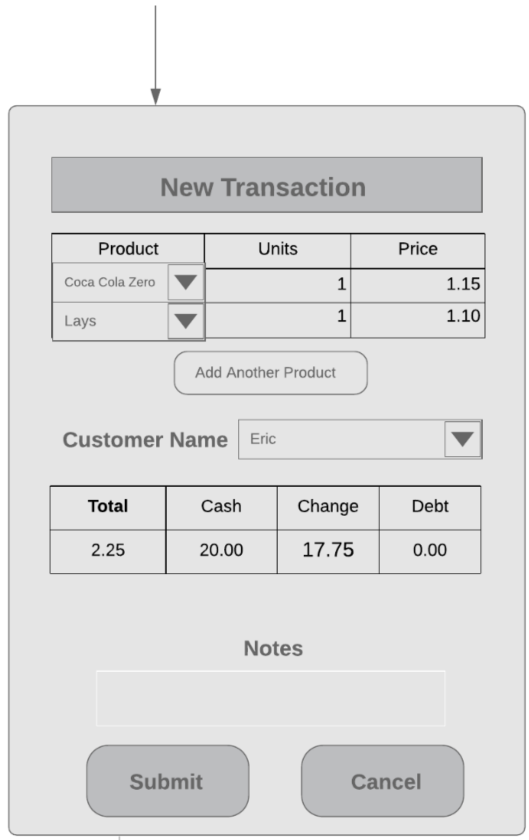
Step 1:
The attendant opens the transaction management screen by clicking the "Transactions" button.

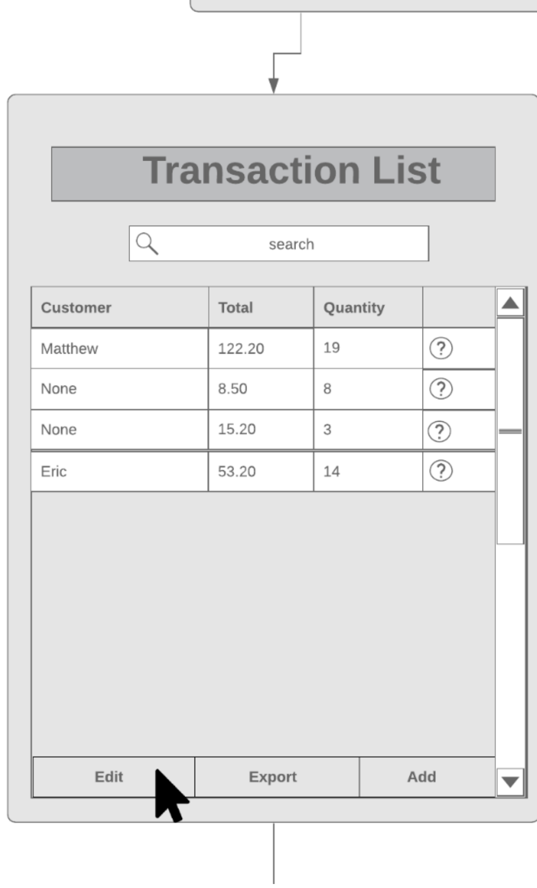



Step 2:
The attendant can view a number of transactions and request more with the scroll bar.

Step 3:
The attendant clicks on a transaction. The transaction opens.

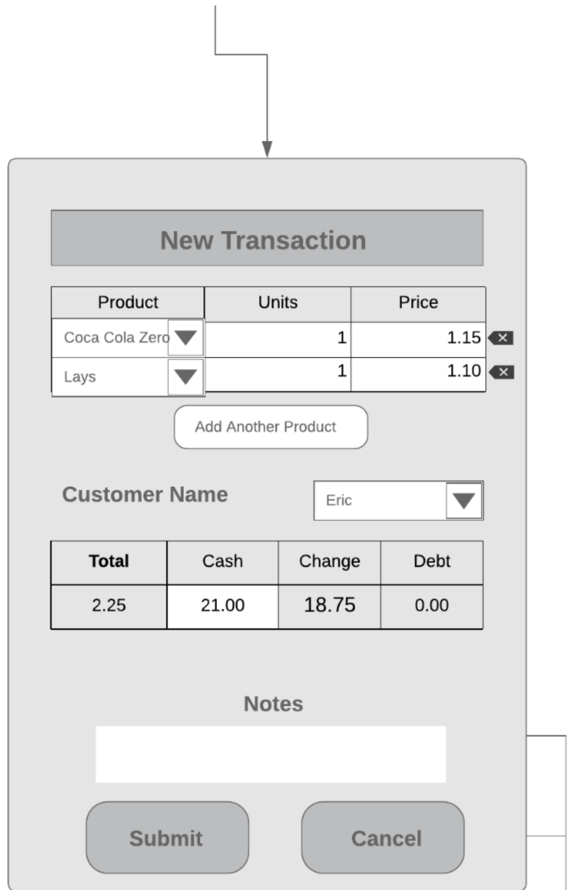
Step 4:
The attendant can view the transaction record, but the fields are gray to indicate they are not open to edits.



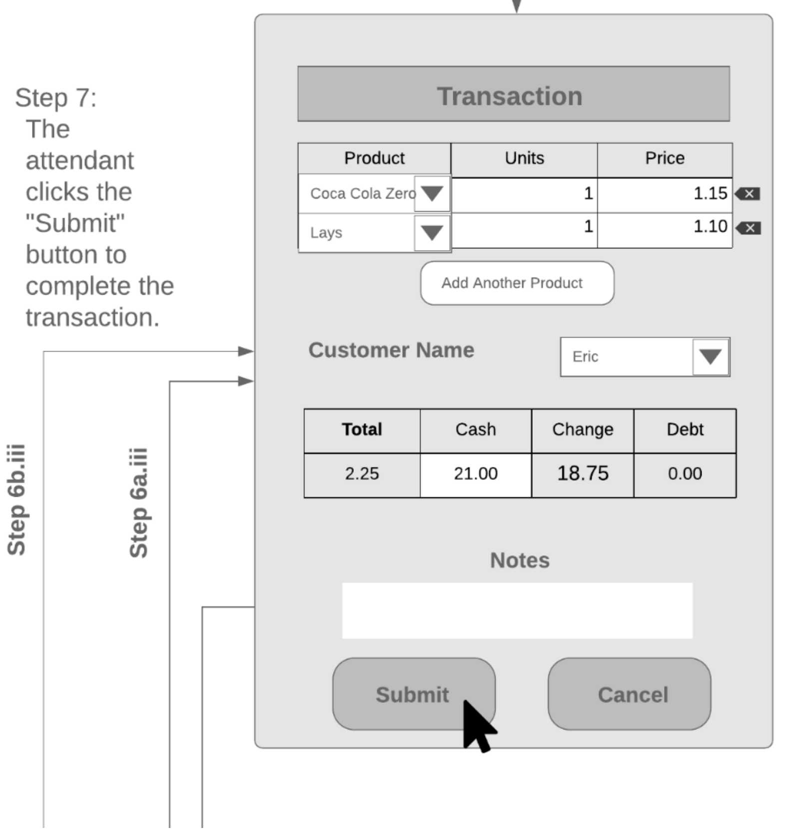



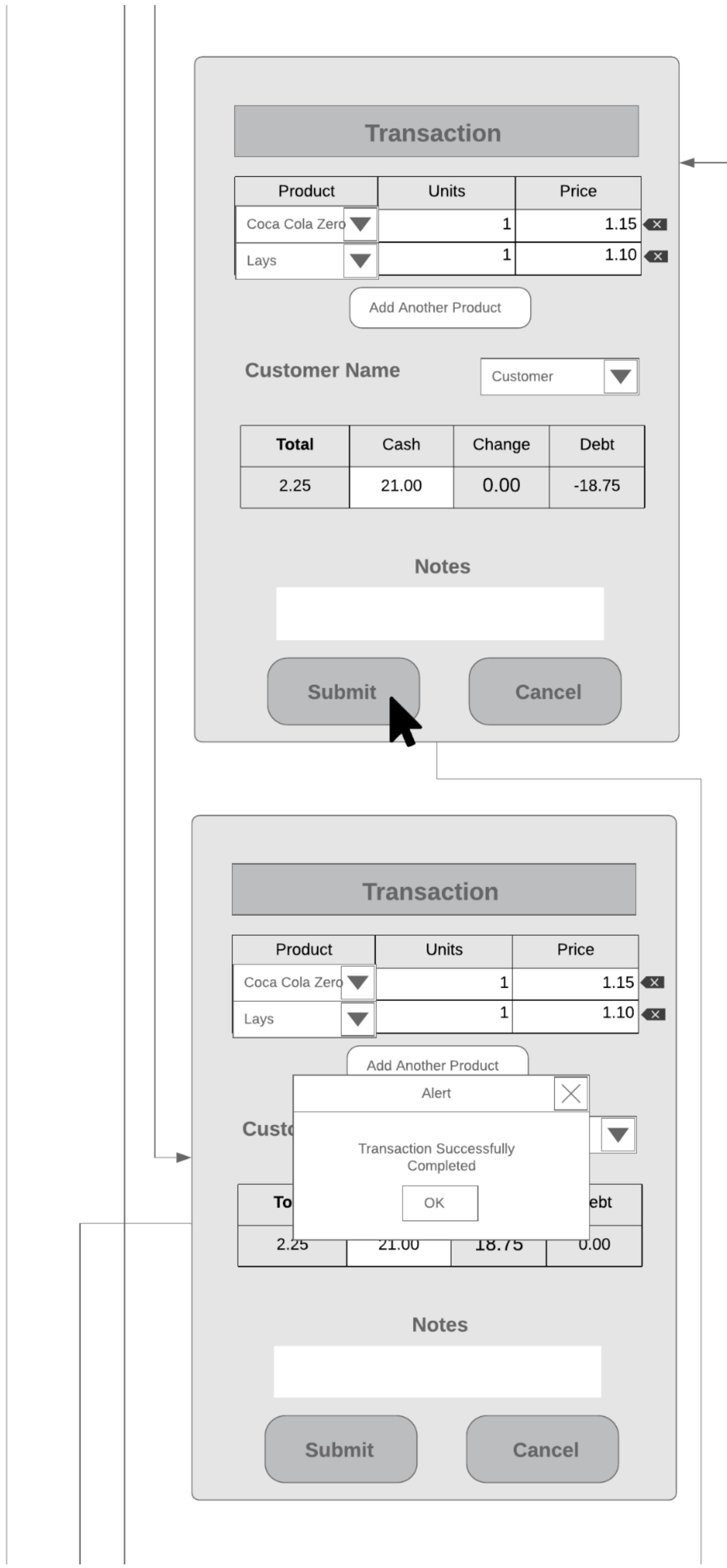

Step 5:
The attendant clicks "edit", in the transaction list to make the fields open to edits.

Step 6:
The attendant returns to the transaction record view, and can edit the fields.




Step 7:
The attendant clicks the "Submit" button to complete the transaction.





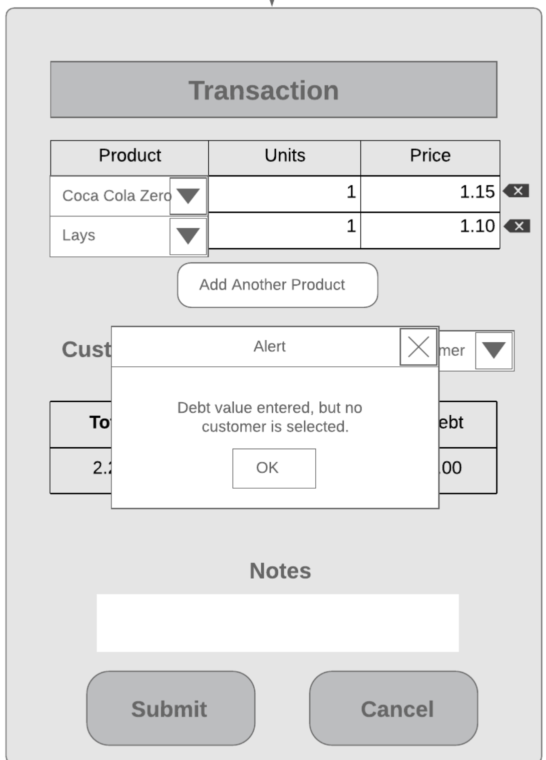

**Alternate Path**

Step 6a.i:

The attendant makes an error, where there is a customer debt value, but no customer selected.

Step 8:

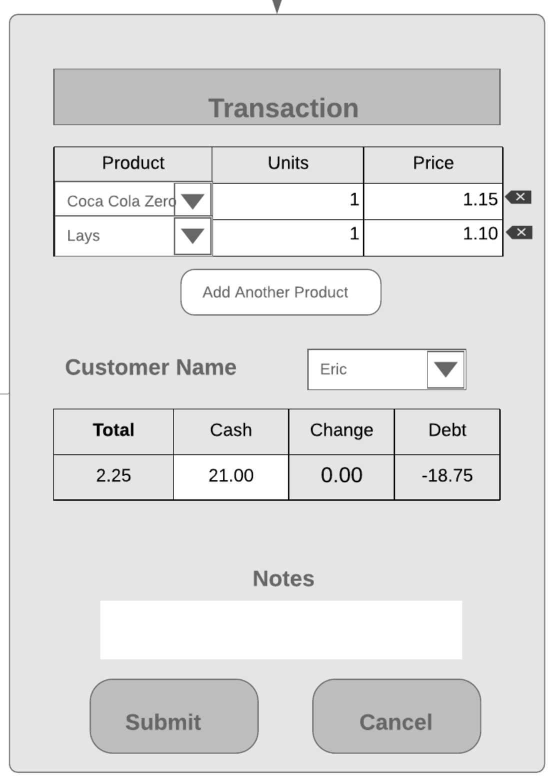
An alert appears telling the attendant that the transaction was successfully completed. The attendant clicks "OK" to return to the transaction list screen.




**Alternate Path**

Step 6a.ii:

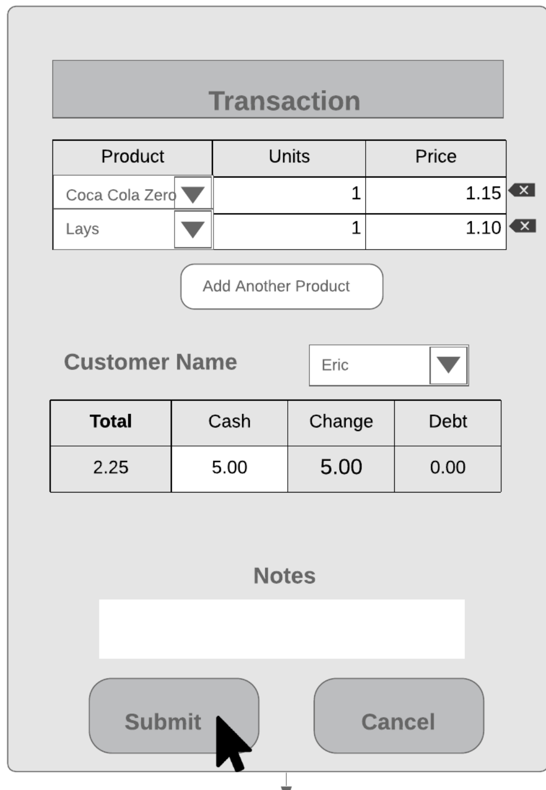
The attendant is presented with an alert that the transaction has not been completed, and the attendant needs to enter a customer name.



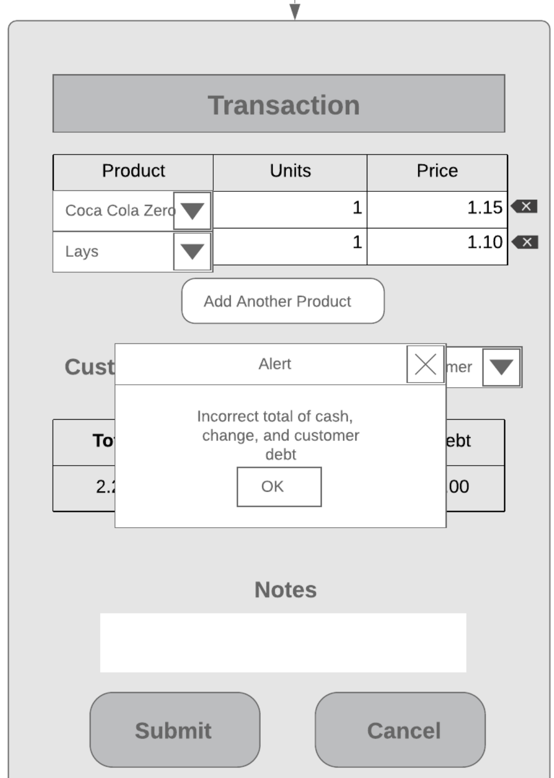

**Alternate Path**

Step 6a.iii:

The attendant completes the "Customer Name" entry. Proceed with step 7.




**Alternate Path**
Step 6b.i:
The attendant enterred an incorrect combination of cash, change and debt, and tries to submit.




**Alternate Path**
Step 5b.ii:
The attendant is presented with an error message.

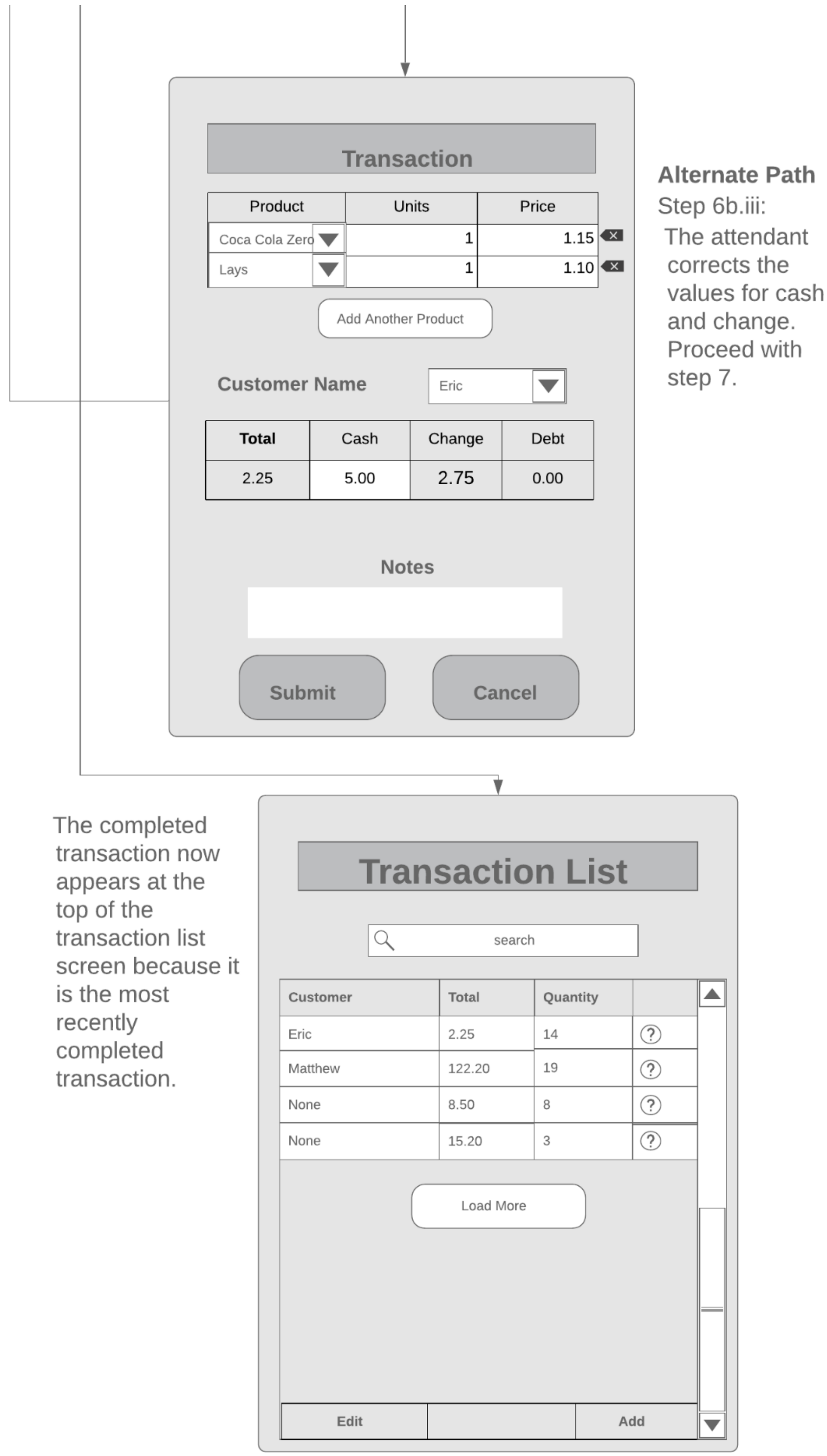



*Figure 22: Storyboard for editing a transaction in the Transaction Management subsystem*

## 6.5.3 Use Case: Export Transactions [UC-12]

### 6.5.3.1 Use Case Description: Export Transactions [UC-12]

Allows an Attendant to export transactions in the Transaction Management subsystem to a CSV file. The transactions that are exported will be from a specified year and month (see Table 12).

*Table 12: UC-12 Description - Export Transactions*

| **Actors** | Attendant |
|---|---|
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has Transaction Management subsystem viewing privileges as set by an Administrator (see UC-07). |
| **Steps** | 1. The Attendant requests to access the Transaction Management subsystem.<br>2. The Attendant is presented with<br>  • the option to add a transaction to the Transaction Management subsystem,<br>  • a list of transactions stored in the Transaction Management subsystem sorted in order by creation date,<br>  • for each transaction, the option to view all information associated with that transaction, and,<br>  • the option to export information associated with each transaction in the Transaction Management subsystem.<br>3. The Attendant requests to export information associated with each transaction in the Transaction Management subsystem.<br>4. The Attendant specifies the year and month of the exported transactions.<br>5. The Attendant requests to submit their request to export the specified transaction information. |
| **Success Conditions** | • A CSV file containing information for the transaction in the Transaction Management subsystem for the specified year and month has been placed into the Downloads folder on the local machine. |

### 6.5.3.2 Sequence Diagram: Export Transaction Information [UC-12]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant exports the information in the Transaction Management subsystem (see Figure 23). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Transaction Management subsystem.

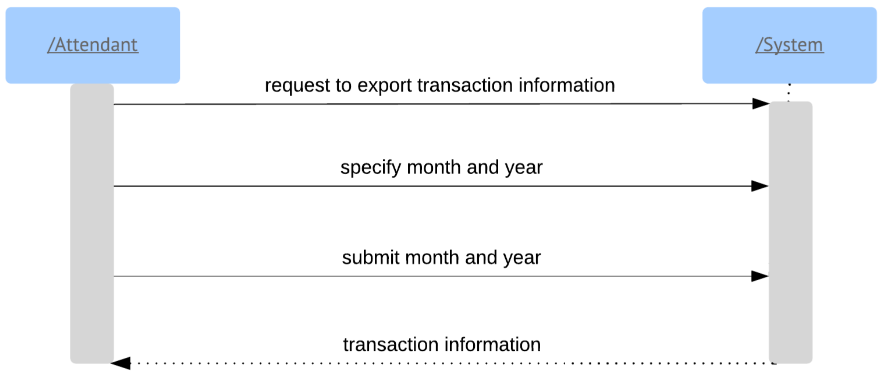
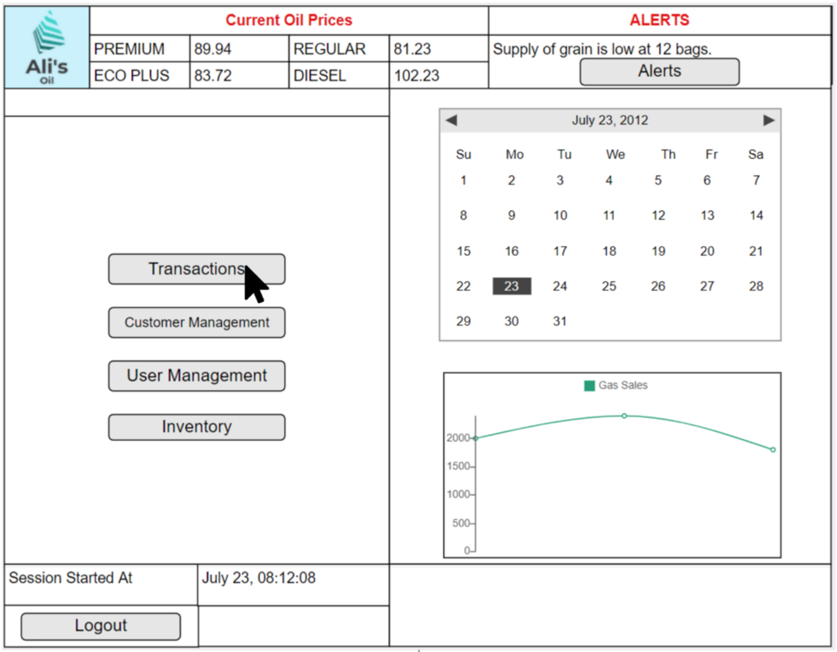


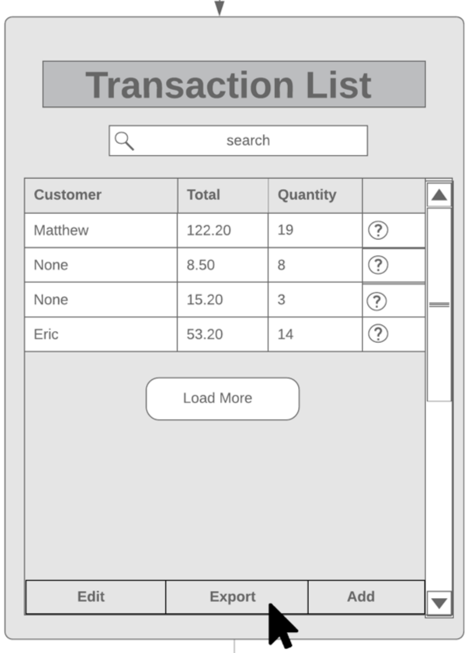

*Figure 23: Sequence Diagram for exporting AttendNet transactions to a CSV file.*

## 6.5.3.3 Storyboard: Export Transaction Information [UC-12]

This storyboard describes a scenario in which an attendant exports transaction information for each transaction in the Transaction Management subsystem (see Figure 24).




Step 1:
The attendant selects "Transactions" and the Transaction List opens.




Step 2:
The attendant clicks the "export" button.

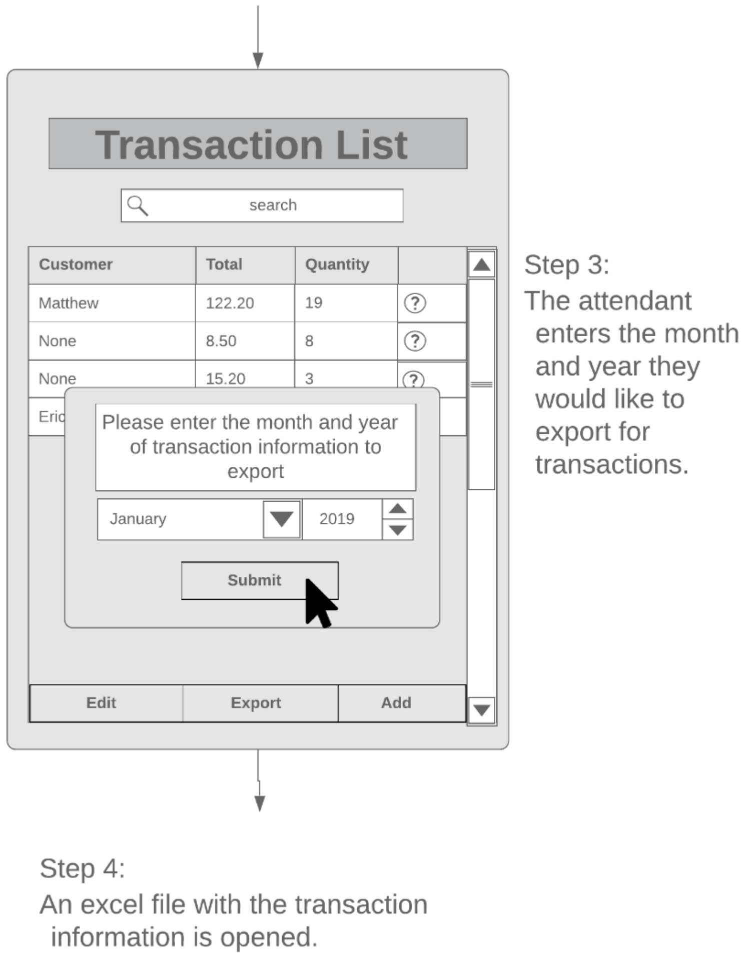



*Figure 24: Storyboard for exporting AttendNet transactions to a CSV file.*

## 6.6 Customer Debt Management

Ali's Oil allows for customers to pay for products at later dates if they cannot be paid for immediately. The management of customer debt must be allowed by AttendNet.

## 6.6.1 Use Case: Add Customer [UC-13]

### 6.6.1.1 Use Case Description: Add Customer [UC-13]

This use case details the interactions between an attendant and AttendNet when adding a customer to the Inventory Management subsystem (see Table 13).

*Table 13: UC-13 Description - Add Customer*

| | |
|---|---|
| **Actors** | Attendant |
| **Preconditions** | • An entered customer does not currently exist in the Customer Debt Management subsystem.<br>• An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has editing privileges for the Customer Debt Management subsystem as set by an Administrator (see UC-07). |
| **Steps** | 1. The Attendant requests to enter the Customer Debt Management subsystem.<br>2. The Attendant is presented with<br>• the option to add a customer to the Customer Debt Management subsystem,<br>• a list of customers stored in the Customer Debt Management subsystem sorted in alphabetical order by customer name,<br>• for each customer, the option to view all information associated with that customer, and,<br>• the option to export information associated with each customer in the Customer Debt Management subsystem.<br>3. The Attendant requests to add a customer to the Customer Debt Management subsystem.<br>4. The Attendant is presented with the ability to enter information for a customer and save that information in the Customer Debt Management subsystem.<br>5. The Attendant enters the following information:<br>• customer name<br>• customer phone number<br>• customer address<br>• notes (optional) |

| | |
|---|---|
| | 6. The Attendant requests to save the customer information in the Customer Debt Management subsystem.<br>7. The Attendant is presented with confirmation that the customer and associated customer information has been saved in the Customer Debt Management subsystem. |
| **Success Conditions** | • A customer and associated customer information for that customer has been saved in the Customer Debt Management subsystem. |

### 6.6.1.2 Sequence Diagram: Add Customer [UC-13]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant adds a transaction to the Customer Debt Management subsystem (see Figure 25). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Customer Debt Management subsystem.

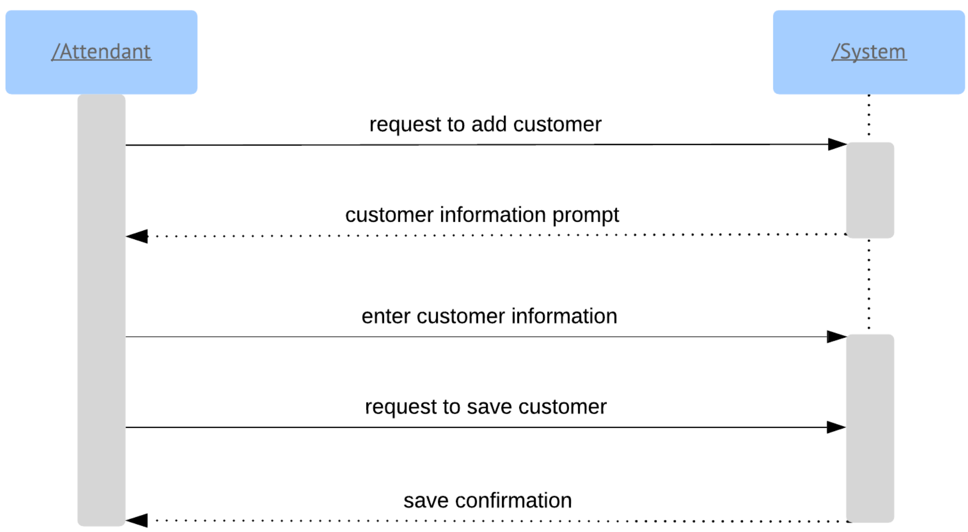
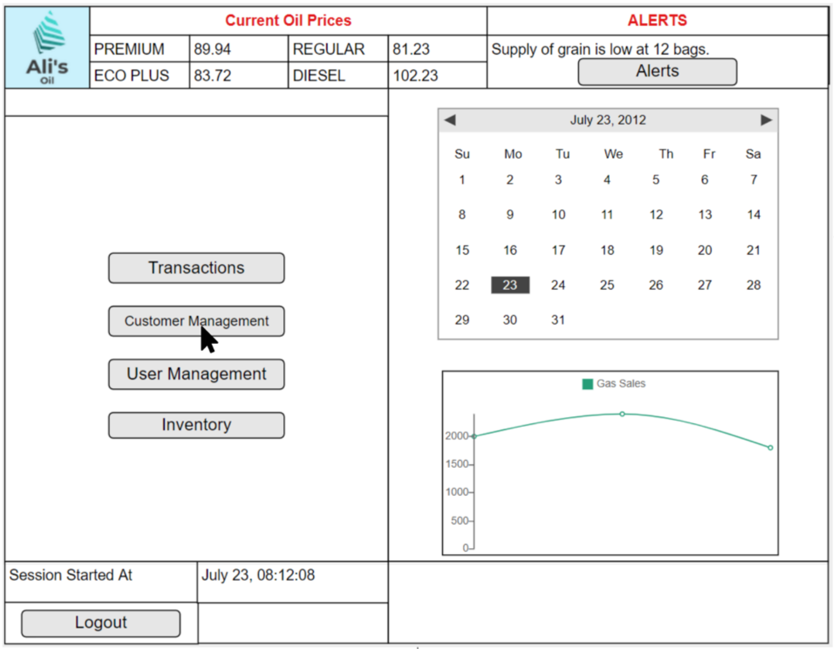


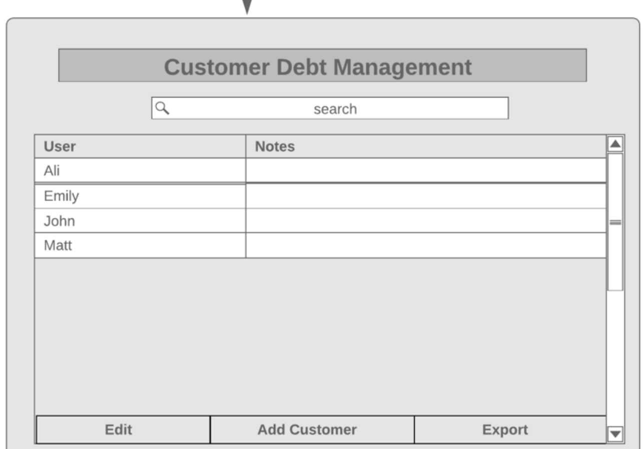

*Figure 25: Sequence diagram for adding a customer to the Customer Debt Management subsystem*

### 6.6.1.3 Storyboard: Add Customer [UC-13]

The following storyboard details a scenario in which an attendant is adding a new customer to the Customer Debt Management subsystem (see Figure 26). The attendant wishes to add a new customer, named Jason, to the Customer Debt Management subsystem. This scenario follows the process of adding a new customer.

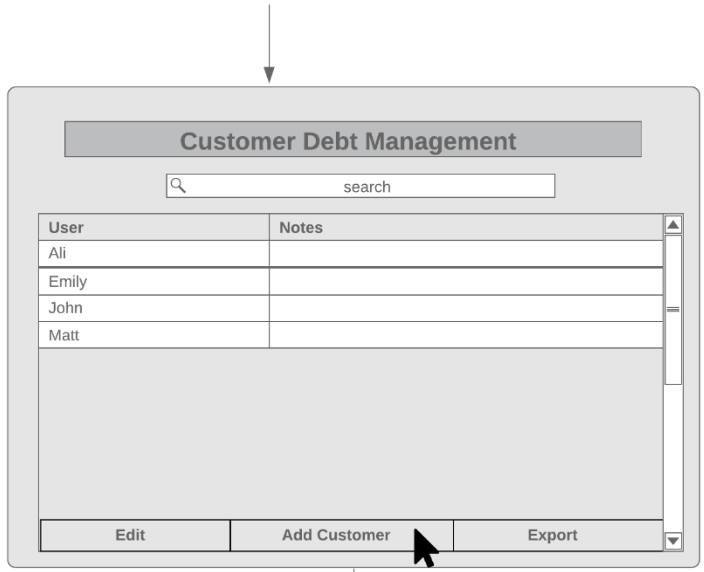



Step 1:
The attendant clicks "Customer Management" to enter the customer management screen.

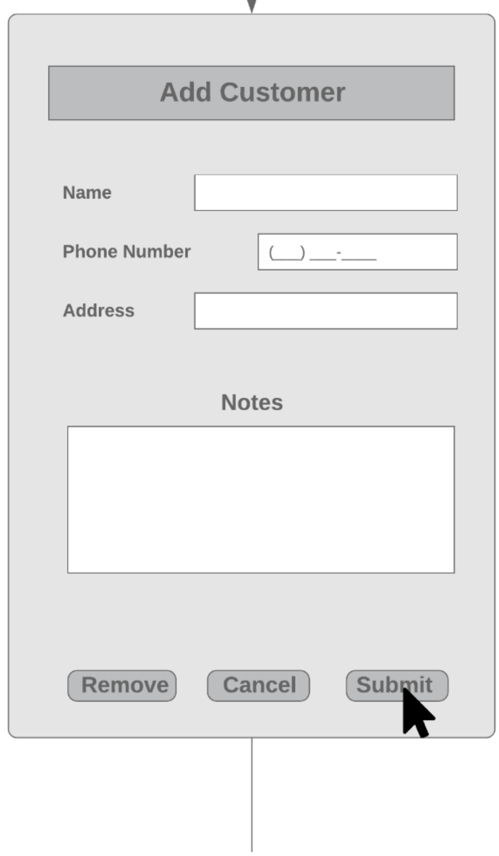



Step 2:
The attendant enters the customer management system and can view the customers, and request to view more with the slider.

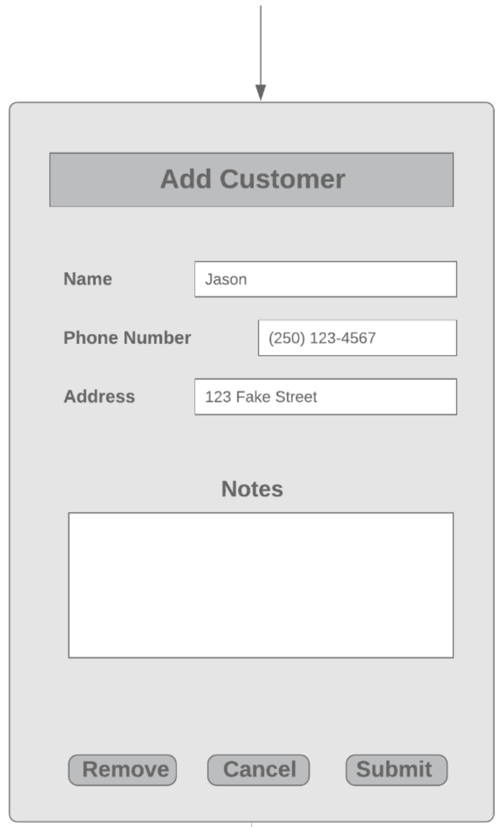
Step 3:
The attendant clicks the "Add Customer" button in the customer debt management screen.



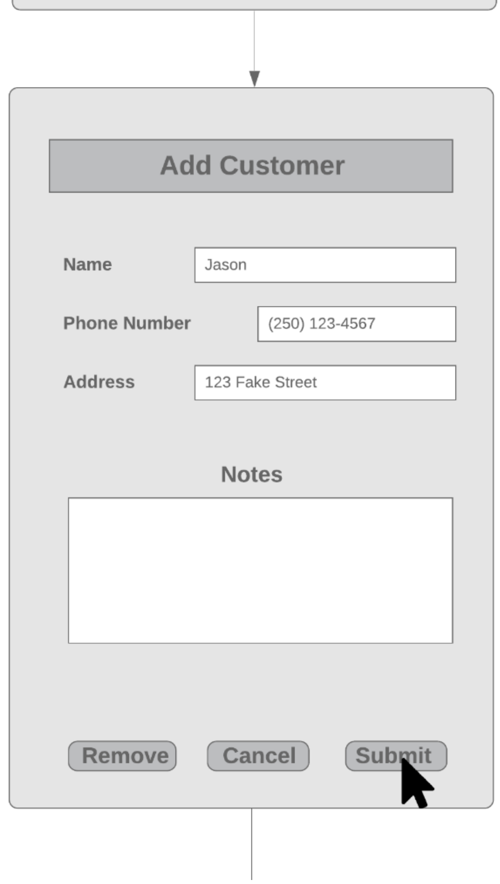

Step 4:
The add customer form opens and the fields are open to edits.



Step 5:
The attendant enters Jason's information into the appropriate input fields.




Step 6:
The attendant clicks the "Submit" button to save the changes.



Step 7:
An alert is shown to the attendant indicating the the new customer named "Jason" was successfully created. The attendant clicks the "OK" button to exit and return to the management screen.

The attendant is returned to the "Customer Debt Management" view and the changes are reflected in the list.

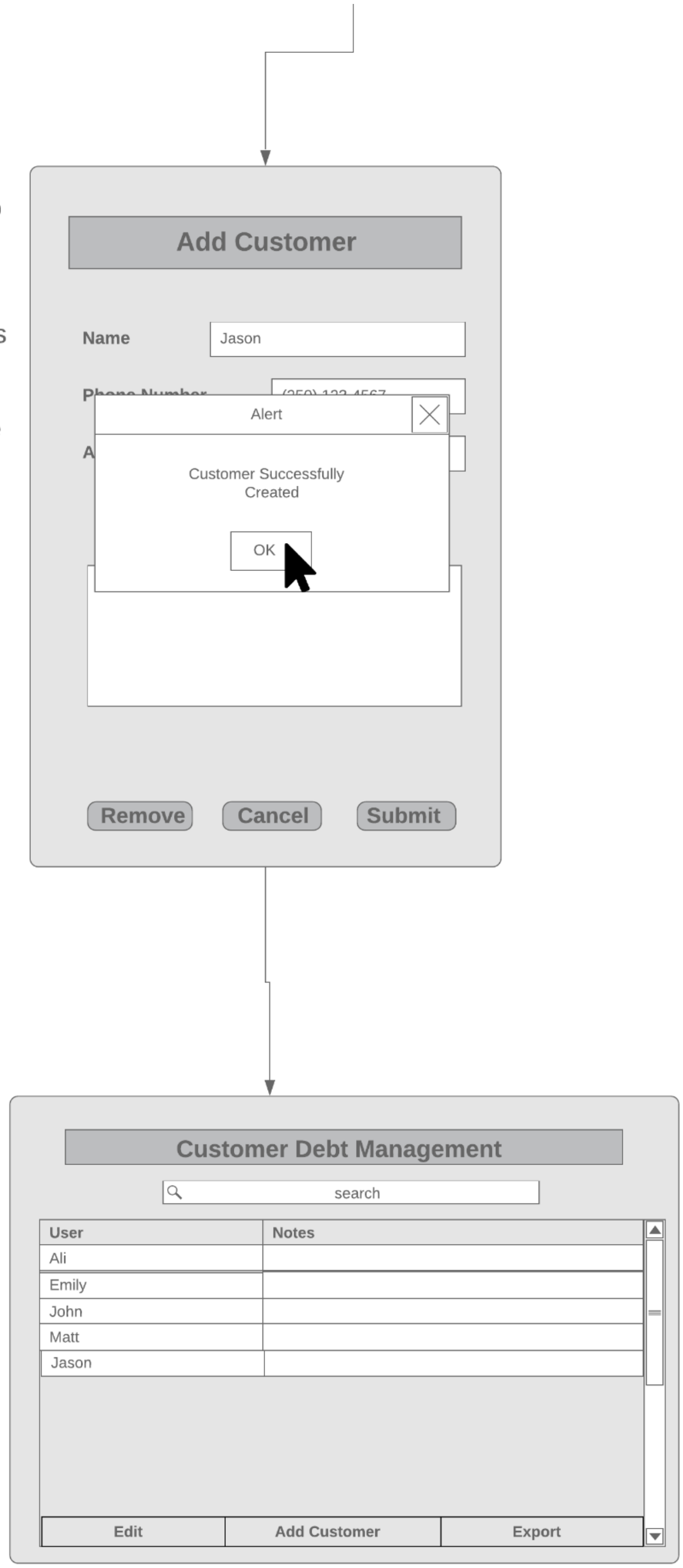



*Figure 26: Storyboard for adding a customer to the Customer Debt Management subsystem*

### 6.6.2 Use Case: View Customer [UC-14]

#### 6.6.2.1 Use Case Description: View Customer [UC-14]

Allows an attendant to view the debt and contact information associated with an existing customer (see Table 14).

*Table 14: UC-14 Description - View Customer*

| | |
|---|---|
| **Actors** | Attendant |
| **Preconditions** | • A customer exists in the Customer Debt Management subsystem (see UC-14).<br>• An Attendant is logged into AttendNet (see UC-05).<br>• The Attendant has viewing privileges for the Customer Debt Management subsystem as set by an Administrator (see UC-07). |
| **Steps** | 1. The Attendant requests to enter the Customer Debt Management subsystem.<br>2. The Attendant is presented with<br>• the option to add a customer to the Customer Debt Management subsystem,<br>• a list of customers stored in the Customer Debt Management subsystem sorted in alphabetical order by customer name,<br>• for each customer, the option to view all information associated with that customer, and,<br>• the option to export information associated with each customer in the Customer Debt Management subsystem.<br>3. The Attendant requests to view information associated with a specific customer.<br>4. The Attendant is presented with information associated with the customer they had previously selected in step 3. |
| **Success Conditions** | • The Attendant is provided with information regarding a specific customer. |

### 6.6.2.2 Sequence Diagram: View Customer [UC-14]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant views a customer's profile in the Customer Debt Management subsystem (see Figure 27). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Customer Debt Management subsystem.

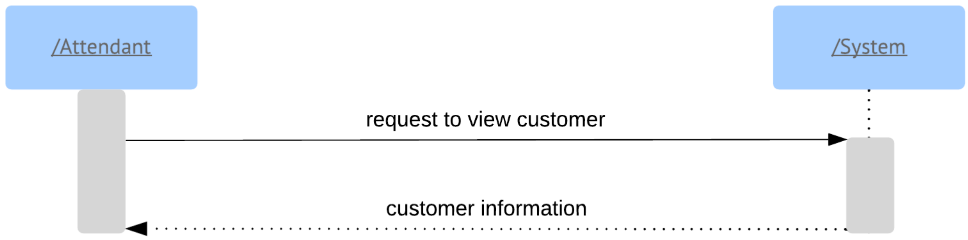
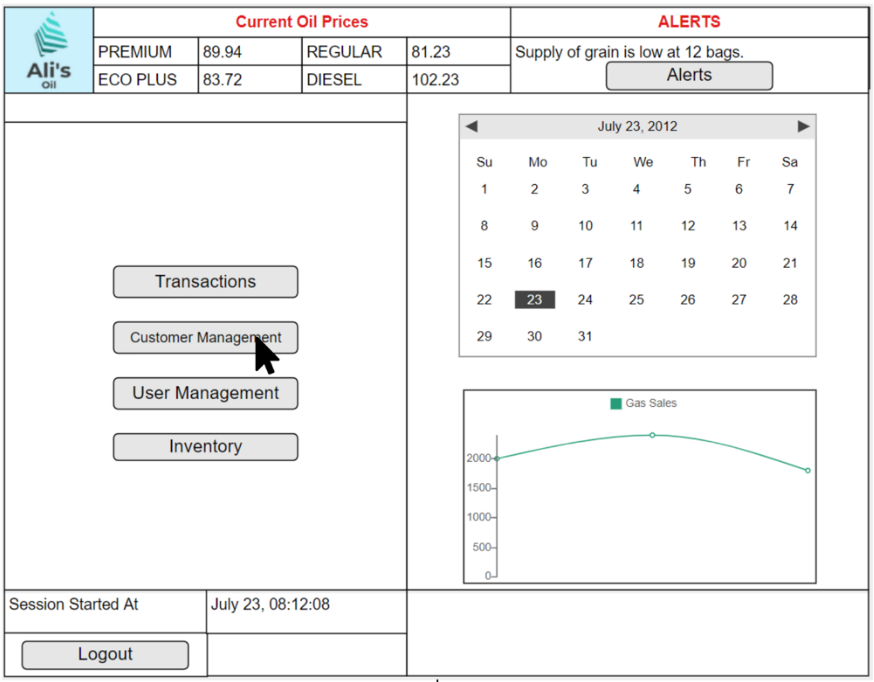


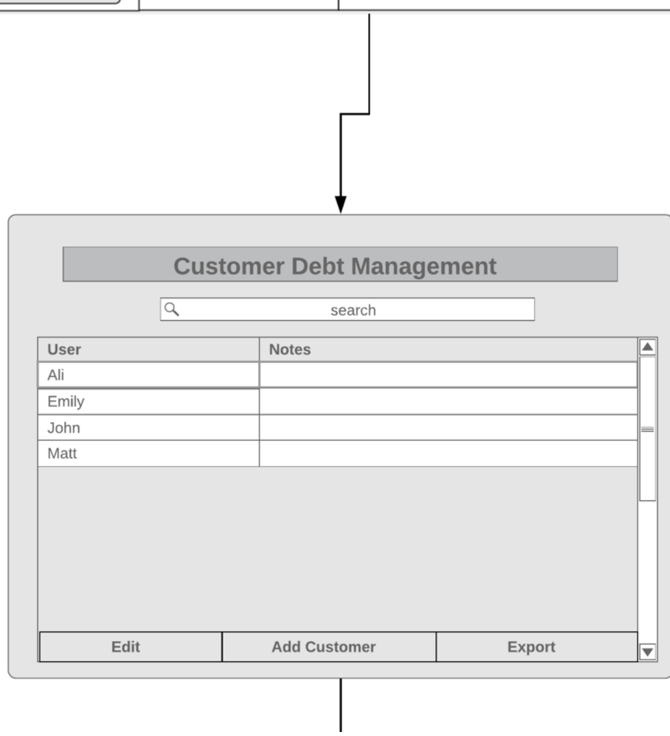

*Figure 27: Sequence diagram for viewing a customer's profile in the Customer Debt Management subsystem*

### 6.6.2.3 Storyboard: View Customer [UC-14]

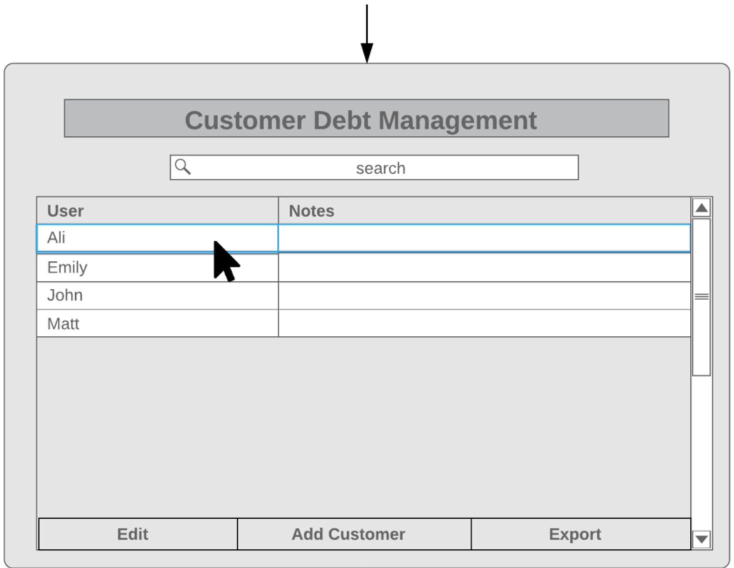
The following storyboard details a scenario in which an attendant is viewing the information associated with a customer in the Customer Debt Management subsystem (see Figure 28). An attendant wishes to check the contact information of a customer to ensure that their new address is listed in the Customer Debt Management subsystem.



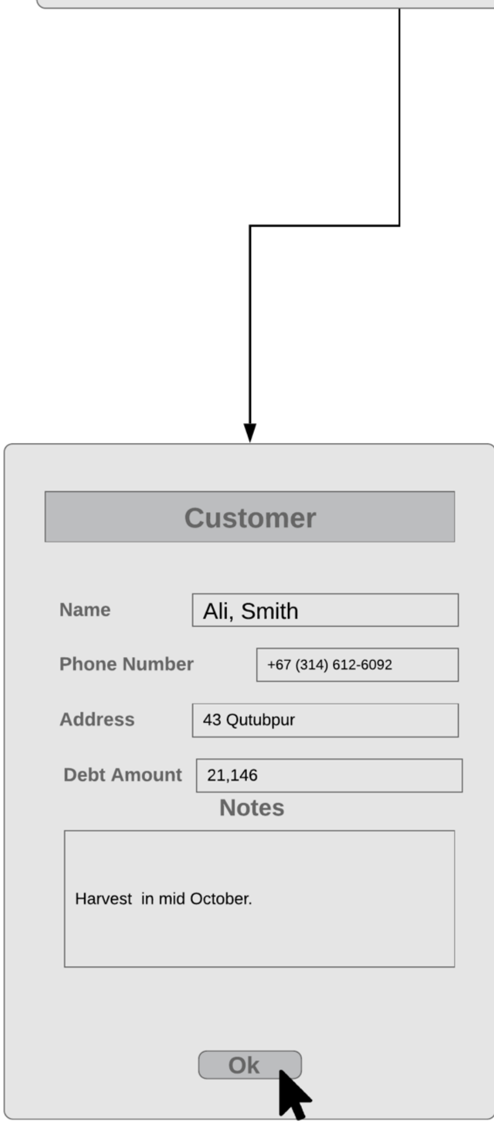

Step 1:
The attendant clicks on "Customer Management" to open the customer management screen.




Step 2:
The attendant can view the list of customers and request to view more by using the scroll bar.




Step 3:
The attendant selects a customer they wish to view the information of. The customer is highlighted blue, and the customer information view is opened.




Step 4:
The attendant can view the customer information, and click "Ok" once they are done.

*Figure 28: Storyboard for viewing a customer's profile in the Customer Debt Management subsystem*

## 6.6.3 Use Case: Edit Customer [UC-15]

### 6.6.3.1 Use Case Description: Edit Customer [UC-15]

Allows an attendant to edit information associated with an existing customer in the debt management subsystem. This does not include modifying the debt associated with a customer (see Table 15).

*Table 15: UC-15 Description - Edit Customer*

<table>
<tr><td>Actors</td><td>Attendant</td></tr>
<tr><td>Preconditions</td><td><ul><li>A customer exists in the Customer Debt Management subsystem (see UC-14).</li><li>An Attendant is logged into AttendNet (see UC-05).</li><li>The Attendant has Customer Debt Management subsystem editing privileges as set by an Administrator (see UC-07).</li></ul></td></tr>
<tr><td>Steps</td><td><ol><li>The Attendant requests to access the Customer Debt Management subsystem.</li><li>The Attendant is presented with<ul><li>the option to add a customer to the Customer Debt Management subsystem,</li><li>a list of customers stored in the Customer Debt Management subsystem sorted in alphabetical order by customer name,</li><li>for each customer, the option to view all information associated with that customer, and,</li><li>the option to export information associated with each customer in the Customer Debt Management subsystem.</li></ul></li><li>The Attendant requests to view all information associated with a customer.</li><li>The Attendant is presented with all information associated with the customer selected in Step 2 and the option to edit information associated with that customer.</li><li>The Attendant requests to edit information associated with the customer.</li><li>The Attendant is presented with the ability to edit any of the following information associated with the customer:<ul><li>customer name</li></ul></li></ol></td></tr>
</table>

| | |
|---|---|
| | • customer phone number<br>• customer address<br>• notes<br>7. The Attendant requests to save the customer information in the Customer Debt Management subsystem.<br>8. The Attendant is presented with confirmation that the customer and associated customer information has been saved in the Customer Debt Management subsystem. |
| **Success Conditions** | • The customer and all information associated with that customer has been saved in the Customer Debt Management subsystem. |

### 6.6.3.2 Sequence Diagram: Edit Customer [UC-15]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant edits a customer's profile in the Customer Debt Management subsystem (see Figure 29). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Customer Debt Management subsystem.

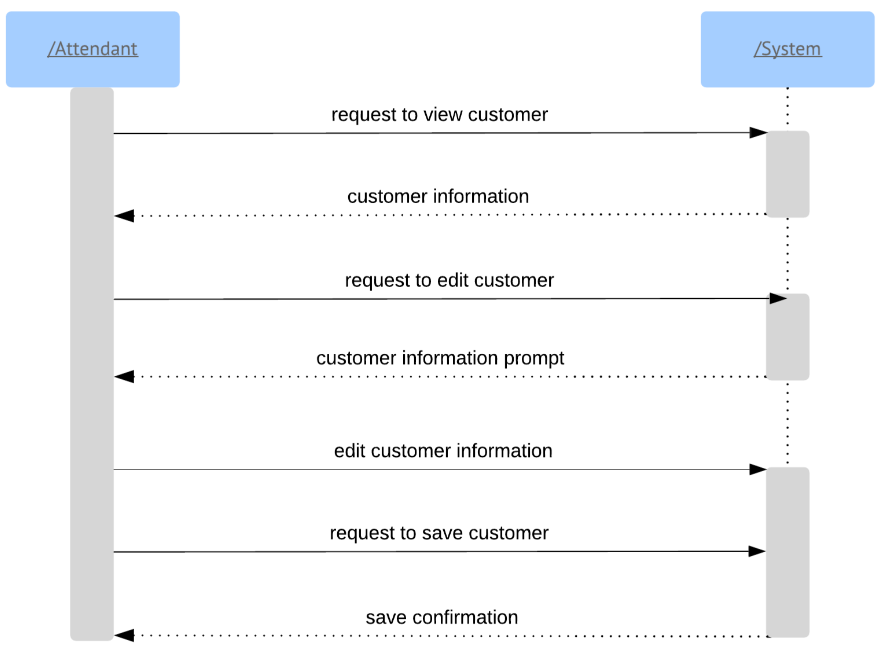
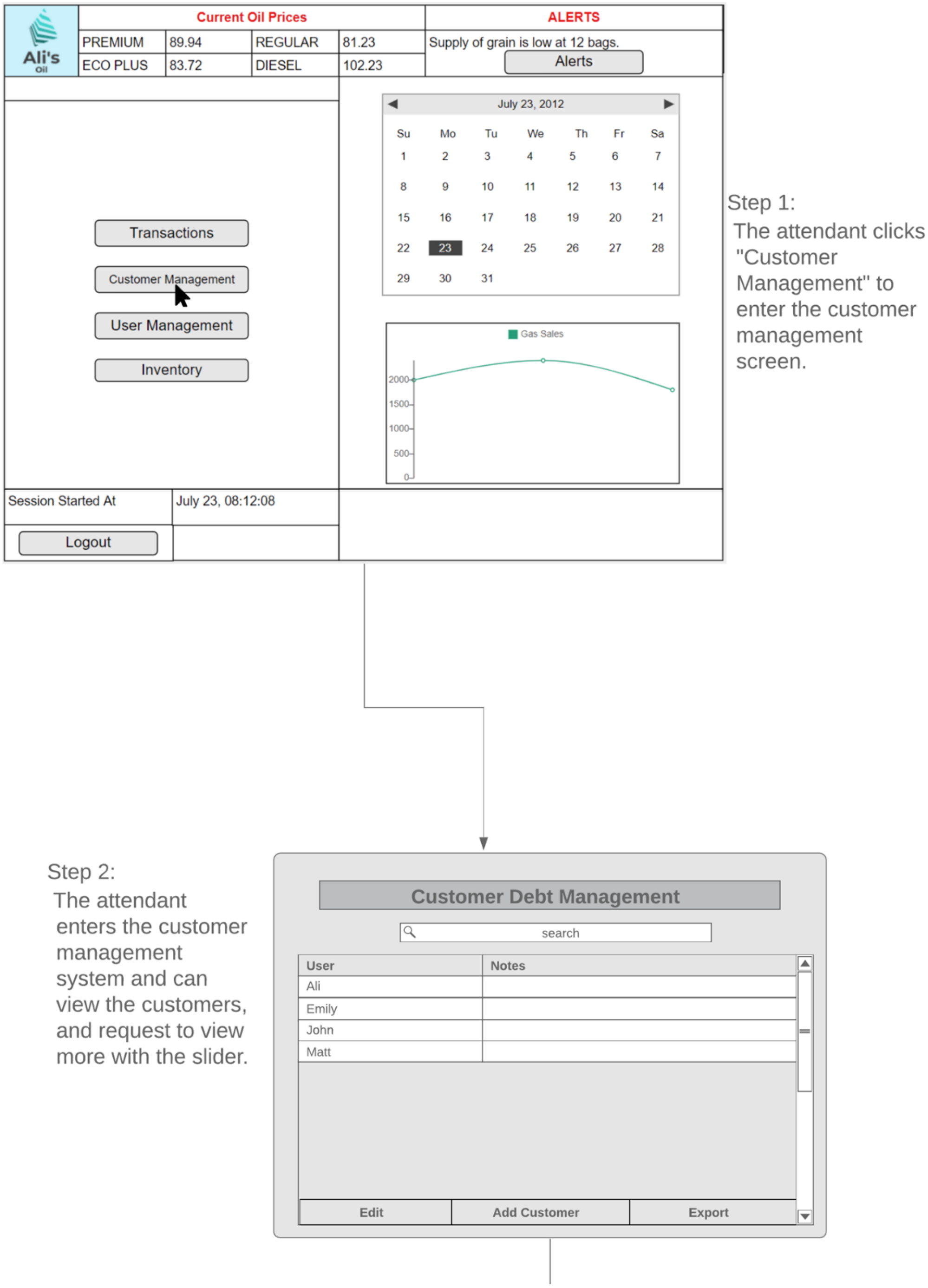


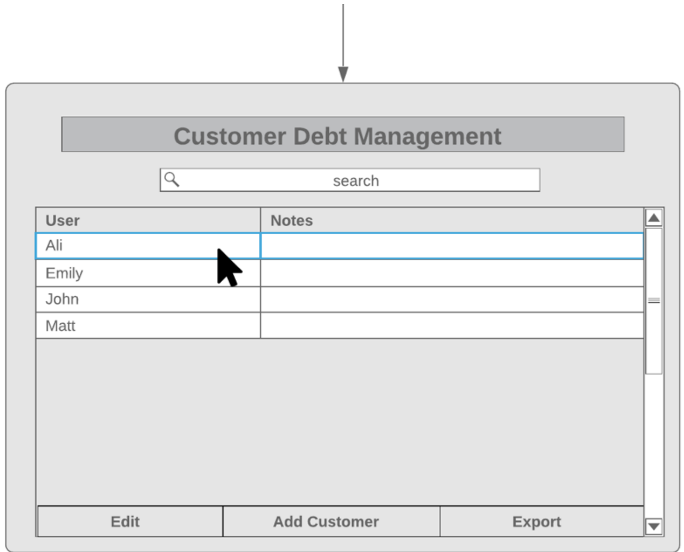

*Figure 29: Sequence diagram for editing a customer's profile in the Customer Debt Management subsystem*

### 6.6.3.3 Storyboard: Edit Customer [UC-15]

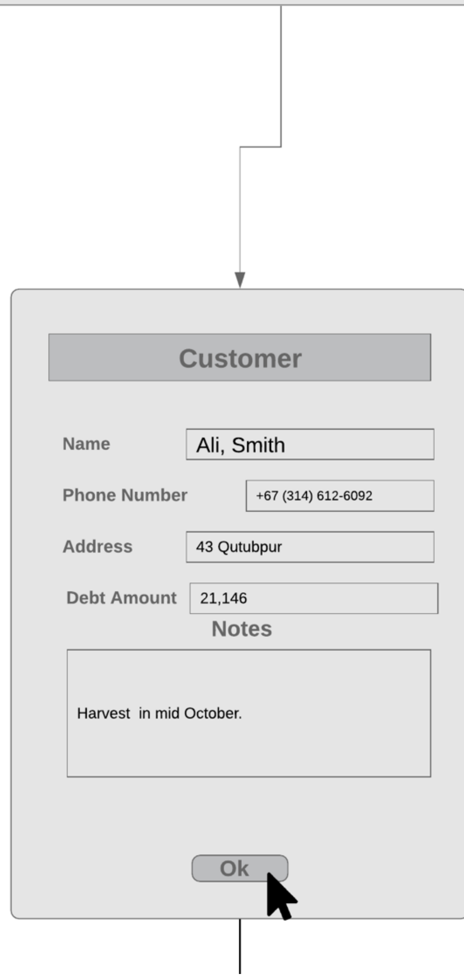
The following storyboard details a scenario in which an attendant is updating the information associated with a customer in the Customer Debt Management (see Figure 30). A customer has obtained a new cell phone, and the customer's contact information needs to be updated to reflect changes.



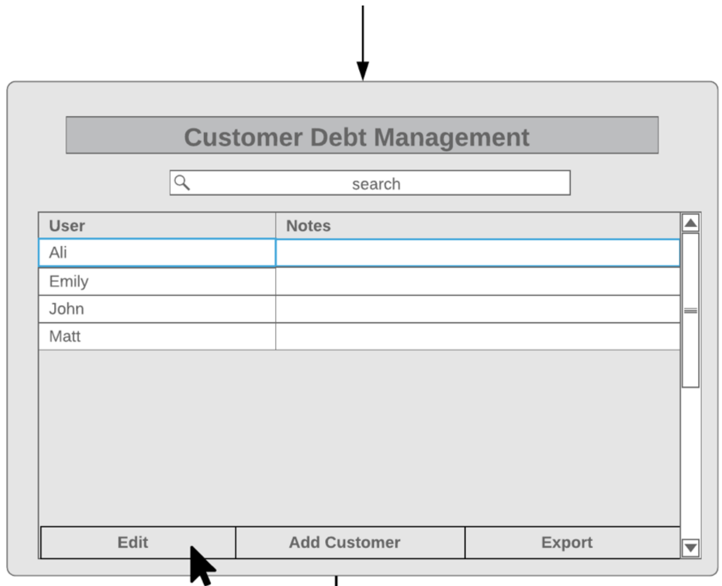
Step 3:
The attendant clicks the on a customer to view their information.



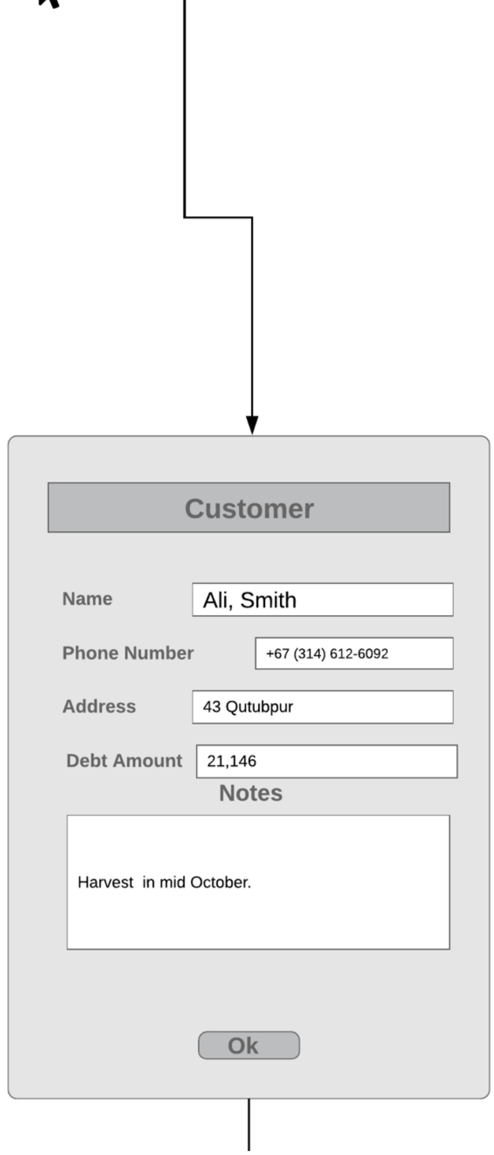

Step 4:
The customer's information is opened.



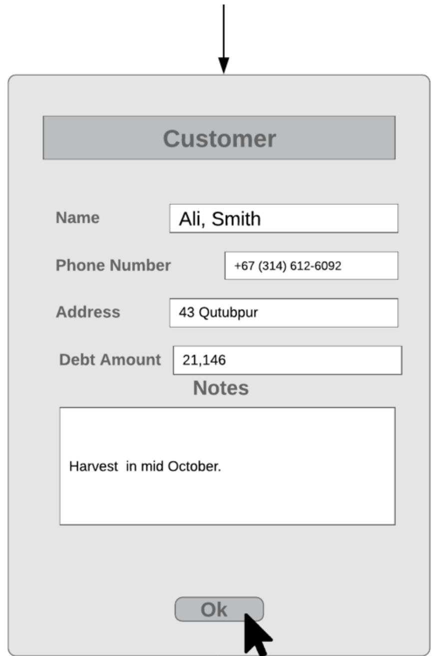
Step 5:
The attendant requests to edit the customer's information by clicking "edit"




Step 6:
The customer information fields are now open to edits.



Step 7:
The attendant clicks the "Ok" button to save the changes.




Step 8:
An alert is shown to the attendant indicating the customer information was saved.

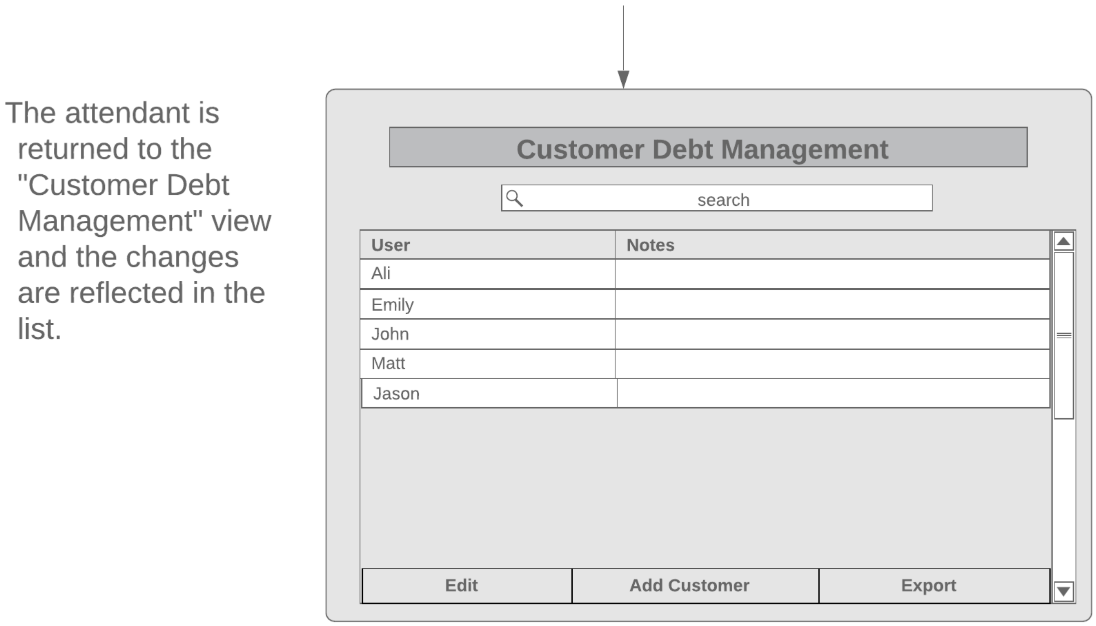



*Figure 30: Storyboard for editing a customer's profile in the Customer Debt Management subsystem*

## 6.6.4 Use Case: Export Customers [UC-16]

### 6.6.4.1 Use Case: Export Customers [UC-16]

Allows an Attendant to export information associated with each customer in the Customer Debt Management subsystem to a CSV file (see Table 16).

*Table 16: UC-16 Description - Export Customers*

| **Actors** | Attendant |
|---|---|
| **Preconditions** | • An Attendant is logged into AttendNet (see UC-05: Login).<br>• The Attendant has Customer Debt Management subsystem viewing privileges as set by an Administrator (see Use Case 3.1: Add Administrator or Attendant). |
| **Steps** | 1. The Attendant requests to access the Customer Debt Management subsystem.<br>2. The Attendant is presented with<br>  • the option to add a customer to the Customer Debt Management subsystem,<br>  • a list of customers stored in the Customer Debt Management subsystem sorted in alphabetical order by customer name, |

| | |
|---|---|
| | • for each customer, the option to view all information associated with that customer, and,<br>• the option to export information associated with each customer in the Customer Debt Management subsystem.<br>3. The Attendant requests to export information associated with each customer in the Customer Debt Management subsystem. |
| **Success Conditions** | • A CSV file containing information for each customer in the Customer Debt Management subsystem has been placed into the Downloads folder on the local machine. |

### 6.6.4.2 Sequence Diagram: Export Customers [UC-16]

This sequence diagram illustrates the interactions that occur between an attendant and AttendNet when an attendant exports the data in the Customer Debt Management subsystem (see Figure 31). The preconditions for this sequence diagram are that the attendant has logged into AttendNet and has entered the Customer Debt Management subsystem.

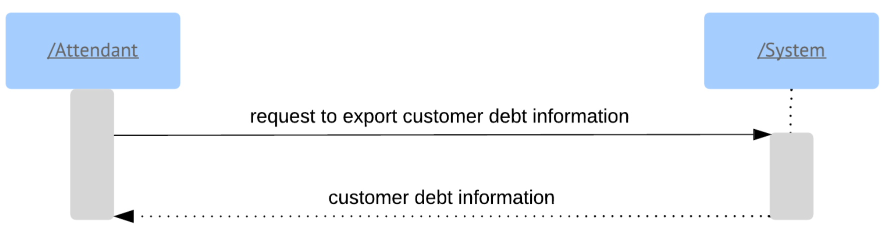



*Figure 31:Sequence Diagram for exporting AttendNet customers to a CSV file.*

### 6.6.4.3 Storyboard: Export Customers [UC-16]

The following storyboard describes a scenario in which an attendant wants to export customer information associated with each customer in the Customer Debt Management subsystem (see Figure 32).

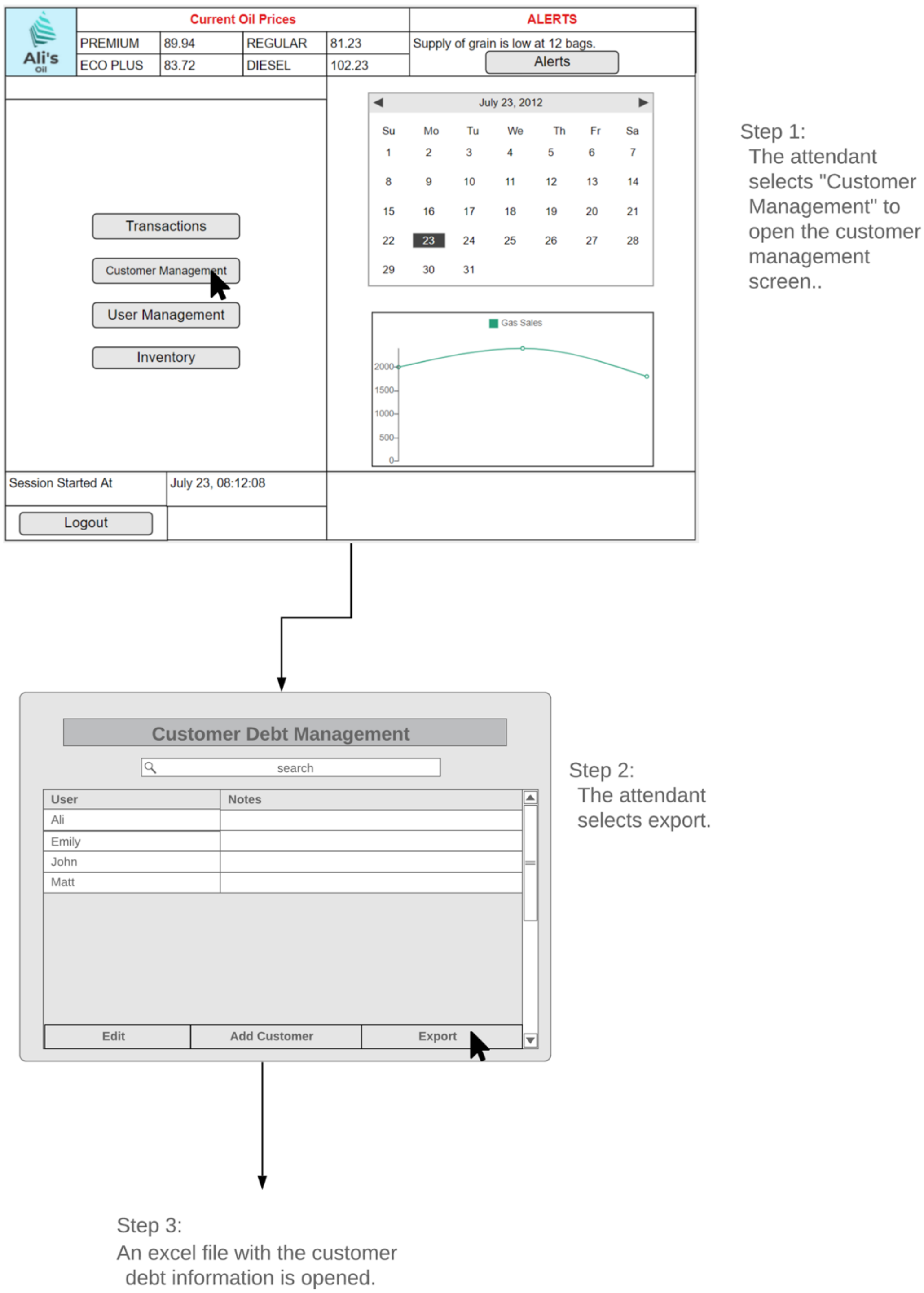



*Figure 32:Storyboard for exporting AttendNet customer to a CSV file.*

# 7. System Diagrams

## 7.1 Data-Flow Diagrams

### 7.1.1 Context Diagram (DFD-0)

DFD-0 provides an overview of the movement and processing of data between external entities and AttendNet. Each edge represents a piece of data written to or read from AttendNet.

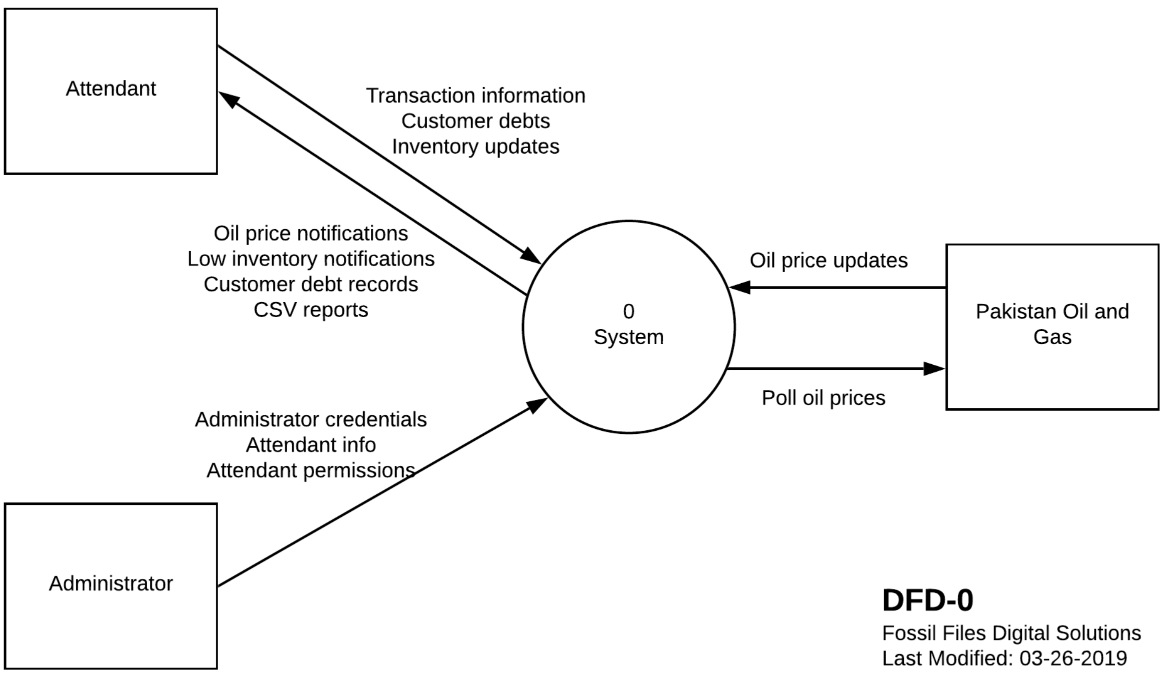



*Figure 33: Data-Flow Diagram Level 0, system context diagram*

## 7.1.2 Data-Flow Diagram Level 1 (DFD-1)

DFD-1 elaborates upon the context diagram detailed in Section 7.1.1. Each process in AttendNet, external entity outside of AttendNet, and datastore is displayed, as well as the flow of data between these elements. Numbers in the processes represent the chronological order of data flow.

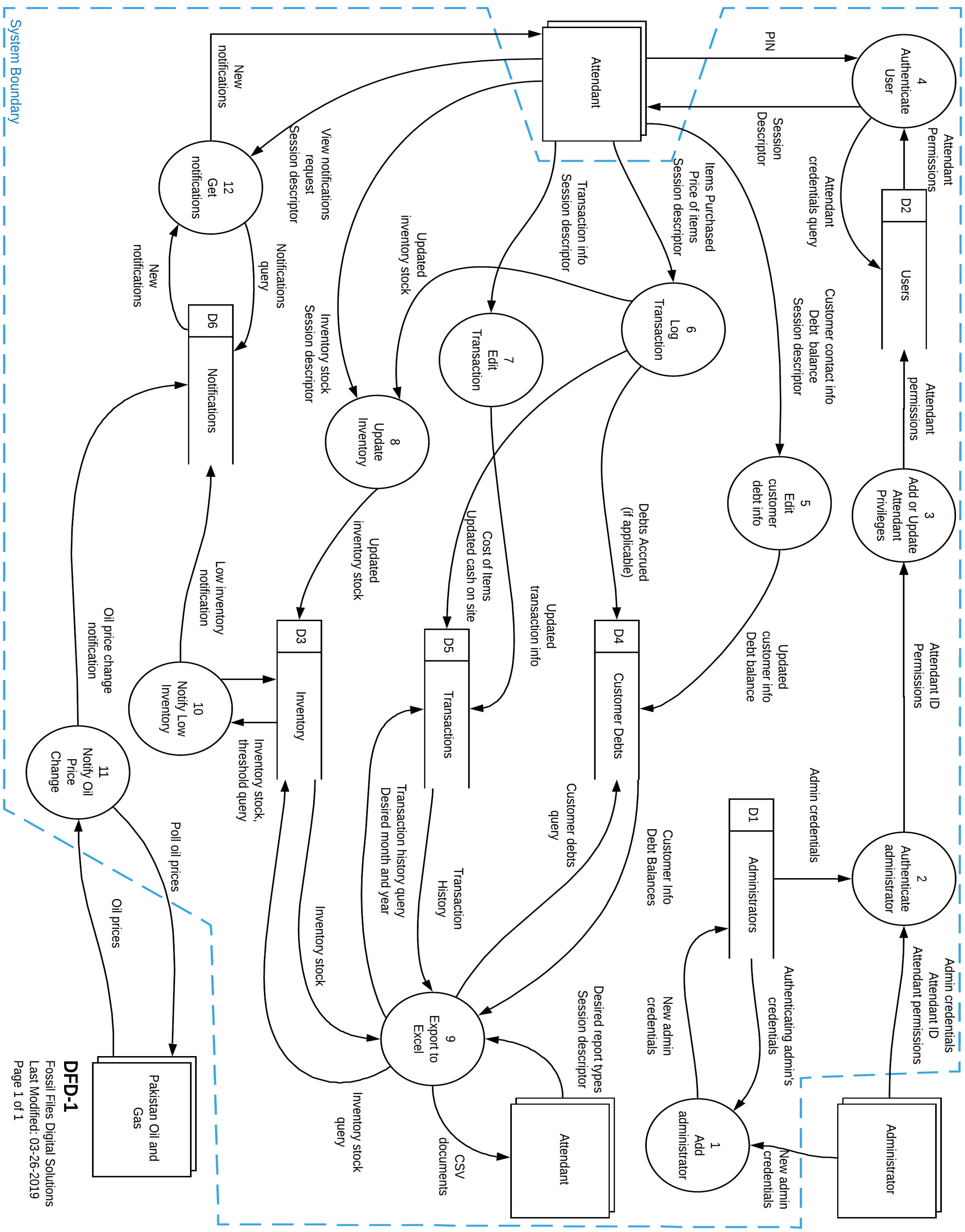



*Figure 34: Data-Flow Diagram level 1*

### 7.1.3 Data-Flow Diagram Level 2 (DFD-2)

DFD-2 provides the highest-fidelity information regarding the data flow between processes, entities, and datastores within the AttendNet domain. Each process represents a sub-task of a process illustrated in DFD-1. The Data-Flow Diagram Level 2 is broken up into four separate pages. See Appendix B for diagram.

## 7.2 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram provided on the following page represents the database structure for AttendNet. Each entity displayed in the diagram corresponds with an existing physical body or logical idea within the problem domain of AttendNet. Listed in each entity is the data associated with the entity, along with an indication of it being a primary key, foreign key, or neither. The Data Dictionary, or DD, is provided inside of the diagram in the form of data types aside respective field names within the entities (e.g. "{alphanumeric}$^{12}$"). This method was used to be clear and concise (see Figure 35).

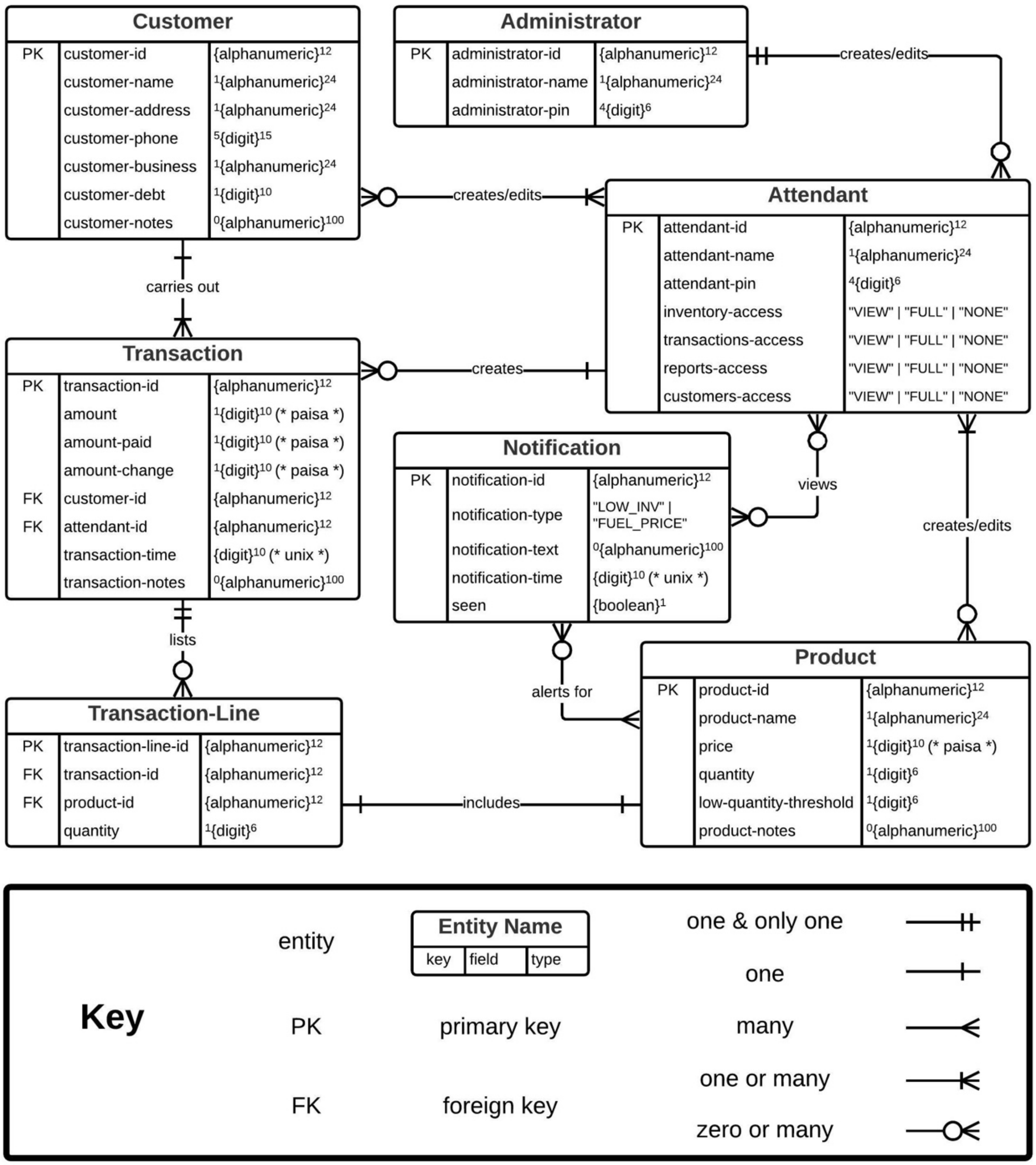



*Figure 35: Entity Relationship Diagram*

## 7.3 Use Case Model

The system-level use case model displayed below details the key interactions that AttendNet's two primary user groups, attendants and administrators, have with AttendNet. Each use case shown pertains to a use case in Section 6 of this document, where they are discussed in further detail. Each use case, excluding the "Login" use case, has an "includes" relationship with the "Login" use case. For simplicity, Fossil File Digital Solutions excluded these "includes" relationships.

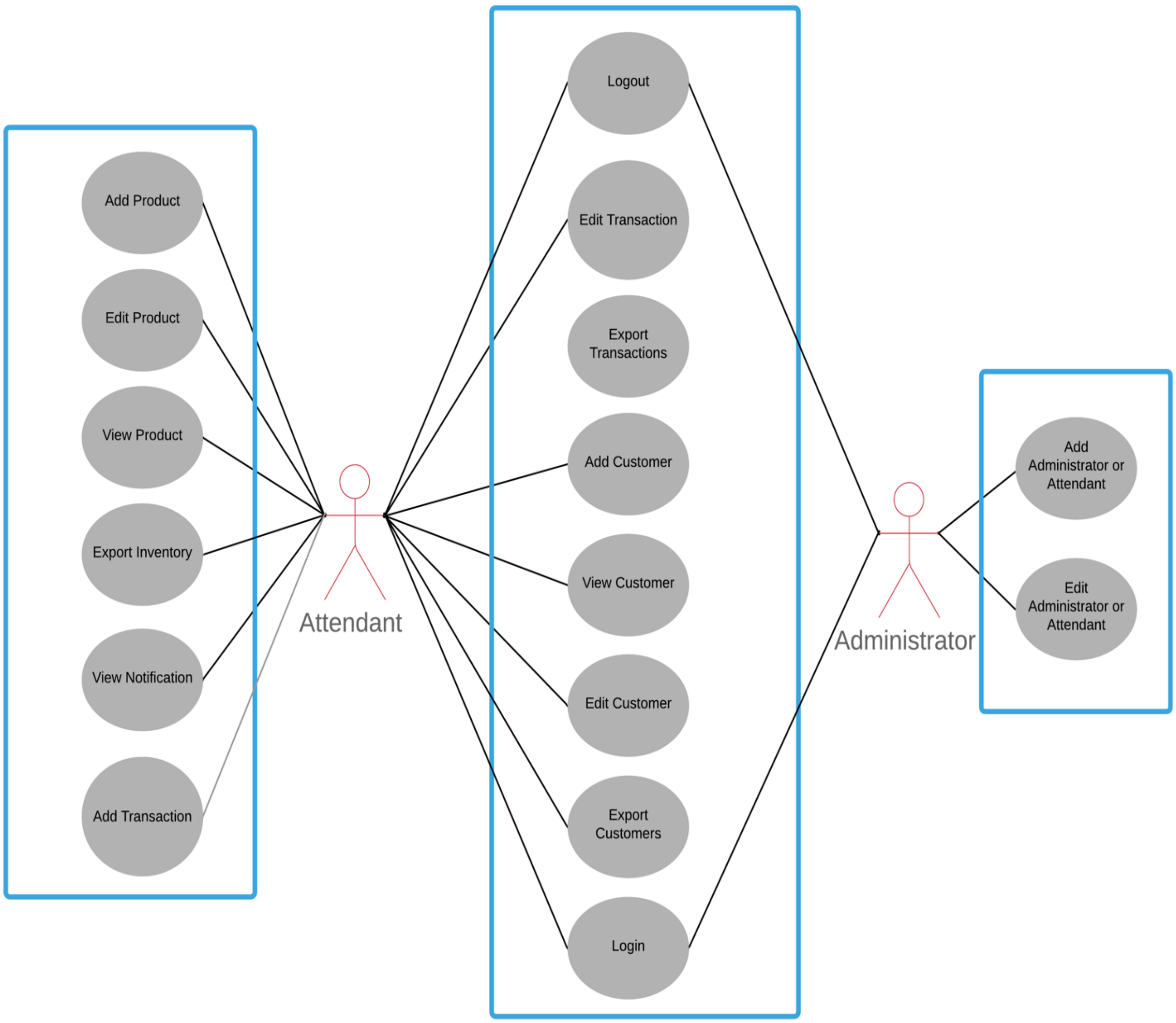



*Figure 36: Use case model for AttendNet*

# 7.4 Requirements Traceability Matrix

The Requirements Traceability Matrix outlines the functional requirements and test scenarios for the system features and external interfaces of AttendNet. In addition, the Requirements Traceability Matrix outlines the non-functional requirements and test scenarios for the non-functional attributes of AttendNet (see Table 17).

*Table 17: Requirements Traceability Matrix for AttendNet.*

| **Requirements Traceability Matrix** | | | | | | |
|---|---|---|---|---|---|---|
| **Project Name: AttendNet** | | | | | | |
| **System Features** | | | **Functional Requirements** | | **Test Scenarios** | |
| ID | Description | Priority | ID | Description | ID | Description |
| 3.1 | Inventory Management | High | FR-01 | Unique product identifier | TS-01 | Verify products have a unique identifier |
| | | | FR-02 | Add product | TS-02 | Verify an attendant can add a product |
| | | | FR-03 | Edit product | TS-03 | Verify an attendant can edit a product |
| | | | FR-04 | View product | TS-04 | Verify an attendant can view a product |
| | | | FR-05 | Product quantity updated from transaction | TS-05 | Verify product quantity is updated from a transaction |
| | | | FR-06 | Low-quantity notification | TS-06 | Verify a low-quantity notification is generated |
| | | | FR-07 | Fuel-price update notification | TS-07 | Verify a fuel-price update notification is generated |
| | | | FR-08 | Fuel prices update automatically | TS-08 | Verify that fuel prices are automatically updated |
| | | | FR-09 | Export product information to CSV | TS-09 | Verify that an attendant can export product information to a CSV file |
| 3.2 | Transaction Management | Medium | FR-10 | Uniquely identifiable transaction | TS-10 | Verify that each transaction is uniquely identifiable |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | FR-11 | Add transaction | TS-11 | Verify that an attendant can add a transaction |
| | | | FR-12 | Edit transaction | TS-12 | Verify that an attendant can edit a transaction |
| | | | FR-13 | View transaction | TS-13 | Verify that an attendant can view a transaction |
| | | | FR-14 | Export transactions | TS-14 | Verify that an attendant can export all transactions to a CSV file |
| 3.3 | Customer Debt Management | Medium | FR-15 | Uniquely identifiable customer | TS-15 | Verify that each customer is uniquely identifiable |
| | | | FR-16 | Add customer | TS-16 | Verify that an attendant can add a customer to the Customer Debt Management system |
| | | | FR-17 | Edit customer information | TS-17 | Verify that an attendant can edit a customer's information |
| | | | FR-18 | View customer information | TS-18 | Verify that an attendant can view a customer's information |
| | | | FR-19 | Export customers | TS-19 | Verify that an attendant can export customer debt information to a CSV file |
| | | | FR-20 | Customer debt value updated from transaction | TS-20 | Verify that debt associated with a customer is updated from a transaction |

| **External Interfaces** | | **Functional Requirements** | | **Test Scenarios** | |
|---|---|---|---|---|---|
| ID | Description | ID | Description | ID | Description |
| 4.1 | User Interfaces | FR-21 | Fuel-price and low-quantity notifications | TS-21 | Verify that an attendant can receive fuel-price updates and low-quantity notifications |
| 4.2 | Software Interfaces | FR-22 | Retrieve fuel-price updates | TS-22 | Verify that AttendNet retrieves fuel prices |
| 4.3 | Communication Interfaces | FR-23 | Remote access | TS-23 | Verify that AttendNet can be accessed remotely |

| Non-Functional Attributes | | Non-Functional Requirements | | Test Scenarios | |
|---|---|---|---|---|---|
| ID | Description | ID | Description | ID | Description |
| 5.1 | Performance | NFR-01 | Remote access within 10 second | TS-24 | Verify that an attendant or administrator can connect to AttendNet within 10 seconds |
| | | NFR-02 | Low-quantity notification generated within 10 seconds | TS-25 | Verify that a low-quantity notification is generated within 10 seconds |
| | | NFR-03 | Exporting within 30 seconds | TS-26 | Verify that CVS files are created within 30 seconds |
| | | NFR-04 | Updating fuel product prices within 10 seconds of detection | TS-27 | Verify that fuel product prices are updated within 10 seconds of detecting a fuel price update |
| | | NFR-05 | Retrieve fuel prices every 6 hours | TS-28 | Verify that fuel prices are retrieved every 6 hours |
| 5.2 | Security | NFR-06 | Administrator viewing and editing of attendant profiles | TS-29 | Verify an administrator can view and edit an attendant profile |
| | | NFR-07 | Attendant add, view, and edit permissions | TS-30 | Verify an attendant can perform only the tasks specified by an administrator in the attendant's profile |
| 5.3 | Software Quality | NFR-08 | Usability to those with little technical experience | TS-31 | Verify a user will little technical experience is able to process a transaction with minimal training |

# Appendix A: Changelogs

Due to the cyclical process of elicitation and analysis in requirements engineering, numerous changes have been made to this document throughout its lifetime. The following sections in this appendix detail changes made to the document after major milestones in the project. For completeness, the previous versions of the document that are mentioned in the headings of this appendix are available for cross-reference upon request.

## Changelog from RD 1.0, RD 1.1 Client Feedback and RD 1.0 External Review

- **All Sections**
  - Named solution system AttendNet – added references to AttendNet
- **1.1 Project Purpose**
  - Project purpose description was changed (previously discussed project scope)
- **1.2 Project Scope**
  - Overall client objective was changed from "increasing profit and decreasing expenses" to "improving efficiency and decreasing expenses"
  - Project scope description was changed (previously discussed project purpose)
- **1.3 Glossary**
  - Added definition of "Ledger"
  - Changed Pakistan State Oil to Pakistan Oil and Gas Regulatory Authority
- **1.5 Overview**
  - Added description of Section 6, System Specifications, and Section 7, System Diagrams
- **2.3 User Classes and Characteristics**
  - Changed Cash Flow Management heading to Transaction Management
- **2.3 User Classes and Characteristics**
  - Changed "Employee" and "Manager" user classes to "Attendant" and "Administrator"
  - Moved inventory management and notification features from administrator to attendants
  - Added "Adjust permissions for individual employees" as administrator feature
  - Removed "and other expenses" from administrator financial report features
  - Changed "the employee cannot view reports on the system that contain long-term financial information" to "the employee cannot view reports on the system that contain financial information"
- **2.4 Operating Environment**
  - Clarified what "common laptops" and "remote access" means

- **2.5 Design and Implementation Constraints**
  - Updated Budget constraint to "$5,000 USD"
  - Updated Language constraint to "English and Urdu characters are supported by AttendNet."
- **3.1 Inventory Management**
  - Defined "product-specific information" in FR-03
  - Updated priority
- **3.2 Metric Reporting**
  - Removed Metric Reporting feature
- **3.3 Cash Flow Management (Transaction Management)**
  - Changed section name from Cash Flow Management to Transaction Management
  - Updated description and priority
  - Removed requirements related to total cash on site
- **3.4 Customer Debt Management**
  - Defined "customer-specific information"
  - Updated priority
  - Removed FR-18
- **4.1 User Interfaces**
  - Specified who receives notifications in FR-26
  - Deleted what was previously FR-24
- **4.2 Software Interfaces**
  - Specified that oil prices are retrieved at approximately 6 hour intervals
- **4.3 Communication Interfaces**
  - FR-28 updated from "computer network" to "Internet"
- **5.1 Performance Requirements**
  - NFR-01, NFR-02, and NFR-03 updated from "10 minutes" to "10 seconds"
  - Removed NFR-02
  - NFR-06 added
  - Changed Pakistan State Oil to Pakistan Oil and Gas Regulatory Authority in NFR-05 and NFR-06
- **5.2 Security Requirements**
  - Expanded to 2 requirements (NFR-07 and NFR-09)
  - Removed NFR-08
- **5.3 Software Quality Attributes**
  - Removed NFR-07: "The system is portable."

# Changelog from RSD 1.0 and RSD 1.1 Client Feedback

- **Client feedback not implemented, with reasoning**
  - Changes should be made to attendant and administrator roles in AttendNet based on manager and employee roles at Ali's Oil
    - Roles and permissions of an attendant and an administrator in AttendNet are not a one-to-one relationship with the manager and employees of the petrol station
  - Customer names do not need to be recorded for each transaction added to AttendNet
    - It is specified elsewhere that a customer name is only necessary in a transaction if the debt for a customer is affected by the transaction
  - Encrypt internet traffic when AttendNet is being remotely accessed
    - Deemed to be scope creep; issue was previously discussed with clients and deemed unnecessary
  - Clarify "limited number" in select use cases
    - Deemed unnecessary as the exact number of transactions, customers, or products displayed to an attendant accessing AttendNet is not important; "limited number" exists to acknowledge that it is not feasible to display *all* transactions, customers, or products in AttendNet at one time
  - Show CSV file in use case storyboards
    - A CSV file itself is outside of the scope of AttendNet; it is sufficient to state that a CSV file has been generated and is available to the attendant that requested the CSV file
  - Capitalize all instances of "attendant" and "administrator"
    - Deemed to be grammatically incorrect due to attendant and administrator being titles referring to unspecified people
  - Attendant and administrator PINs should not have a default value
    - A PIN for an attendant or administrator of AttendNet does not have a default value – it is stated elsewhere that a PIN is a required field when adding an attendant or administrator to AttendNet
  - Data dictionary is not included with the ERD
    - Data dictionary is included in the ERD
  - "seen" should not be a field in the Notification entity of the ERD
    - As requested by the clients, a method of checking whether a notification has been viewed by an attendant was provided through the existence of the "seen" boolean field in the notification entity – as such, it is a necessary value unless the clients wish to eliminate this feature of notifications
- **All Sections**
  - Removed every instance of "all" used ambiguously
  - Replaced "authenticating a session" with "logging into AttendNet"

- Replaced “invalidating a session” with “logging out of AttendNet”
- Replaced “system” with “subsystem” in the following terms:
- “Inventory Management system”
    - “User Authentication system”
    - “User Management system”
    - “Notifications system”
    - “Transaction Management system”
    - “Customer Debt Management system”

- **3.1 Inventory Management**
    - Removed FR-06
    - Moved FR-21 and FR-22 up to this section from Section 4.1
- **3.4 Customer Debt Management**
    - Removed ‘customer business’ field from list of customer-specific information in FR-16
- **4.1 User Interfaces**
    - Removed FR-20
    - Replaced FR-21 and FR-22 with a more appropriate requirement
- **6.0 System Specifications**
    - Client request to show the Windows file explorer (6.6.4.3) was not implemented because it is outside the system boundary.
    - Added alternate paths to storyboards 6.1.1.3 and 6.1.2.3 to match alternate paths in use cases
    - Clarified success conditions for UC-07 and specified the saved information
    - Client request to add steps to the start of each sequence diagram was not implemented as the requested steps are covered in the pre-conditions listed in the description for each sequence diagram.
- **7.1 Data-Flow Diagrams**
    - Added “poll oil prices” path to Pakistan Oil and Gas actor
    - Added “create attendant” path to DFD-1
    - Increased size of DFD-1 in document
- **7.3 Use Case Model**
    - Replaced “Customer Debt Information” with “Customer”
    - Added relationship between administrator and “Invalidate Session” use case

## Changelog from RSD 1.0 External Review

- **TA Feedback not implemented, with reasoning**
    - Budget of $5000 may be too low
        - After further discussion with clients, the budget cannot be increased any more from the original value of $500 due to the economic differences between Pakistan and Canada

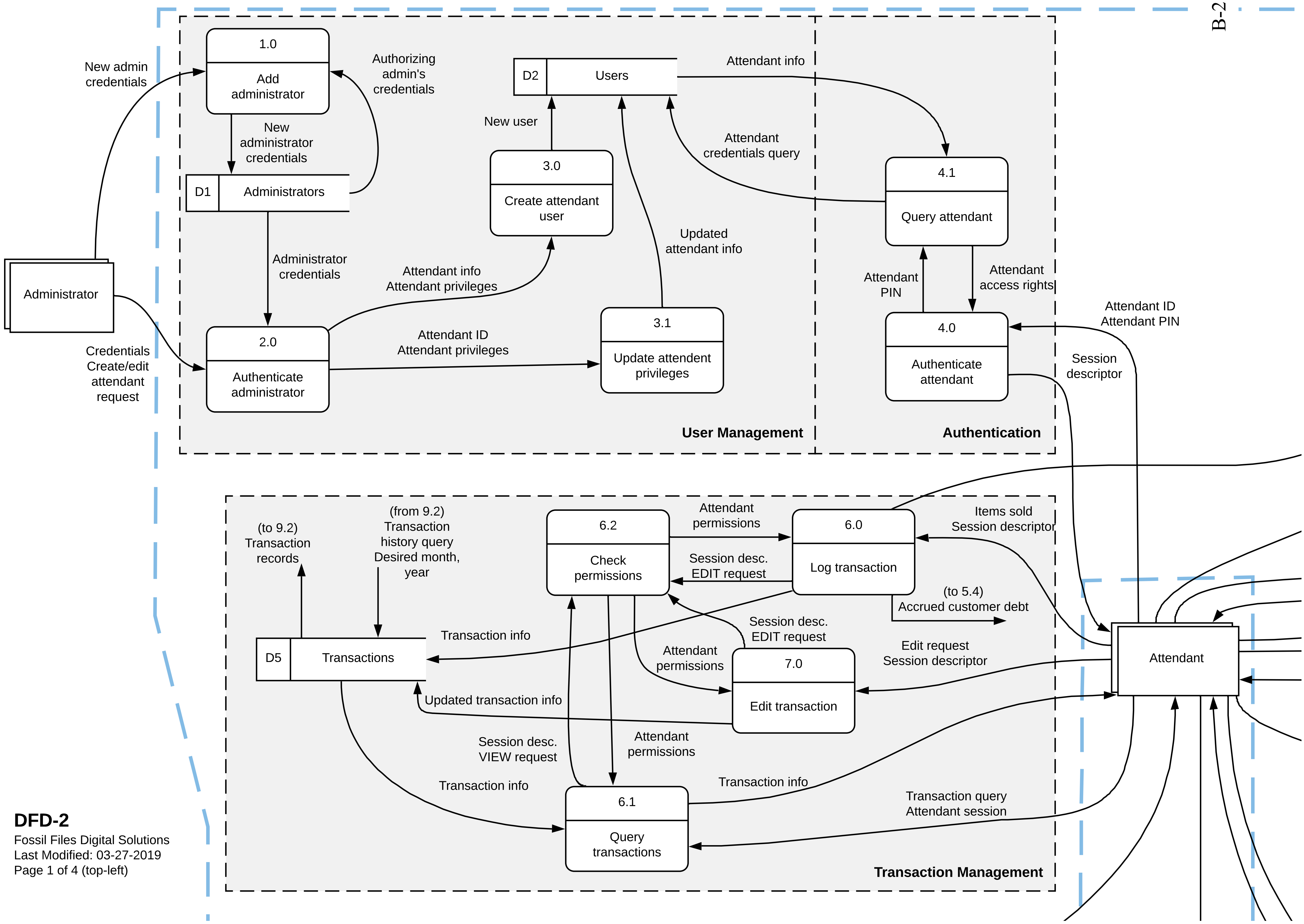
- Use case references to

- **All Sections**
  - Replaced “system” with “subsystem” in the following terms:
    - “Inventory Management system”
    - “User Authentication system”
    - “User Management system”
    - “Notifications system”
    - “Transaction Management system”
    - “Customer Debt Management system”
    - Replaced every instance of “the system” with “AttendNet”
- **1.3 Glossary**
  - Added the following terms to the glossary:
    - “Inventory Management system”
    - “User Authentication system”
    - “User Management system”
    - “Notifications system”
    - “Transaction Management system”
    - “Customer Debt Management system”
- **6.0 System Specifications**
  - Added numbering for use cases (e.g. UC-01)
  - Edited use case references in use cases to use numbering
  - The steps of each sequence diagram were modified to better match the steps in each associated use case
  - Changed Step 5 in UC-07 to be more generic
- **7.1 Data-Flow Diagrams**
  - Added numbering for system in DFD-0
  - Corrected numbering in DFD-1 and DFD-2 (0 should be reserved for context diagram)
  - Added query requests for each interaction with a datastore
  - Included notifications datastore
  - Added data flow label between administrator actor and process 1.0

# Appendix B: DFD-2

DFD-2 is separated onto the next 4 pages; it should be re-assembled to get the full diagram as displayed below based on page number.

| | |
|---|---|
| **B-2** | **B-3** |
| **B-4** | **B-5** |

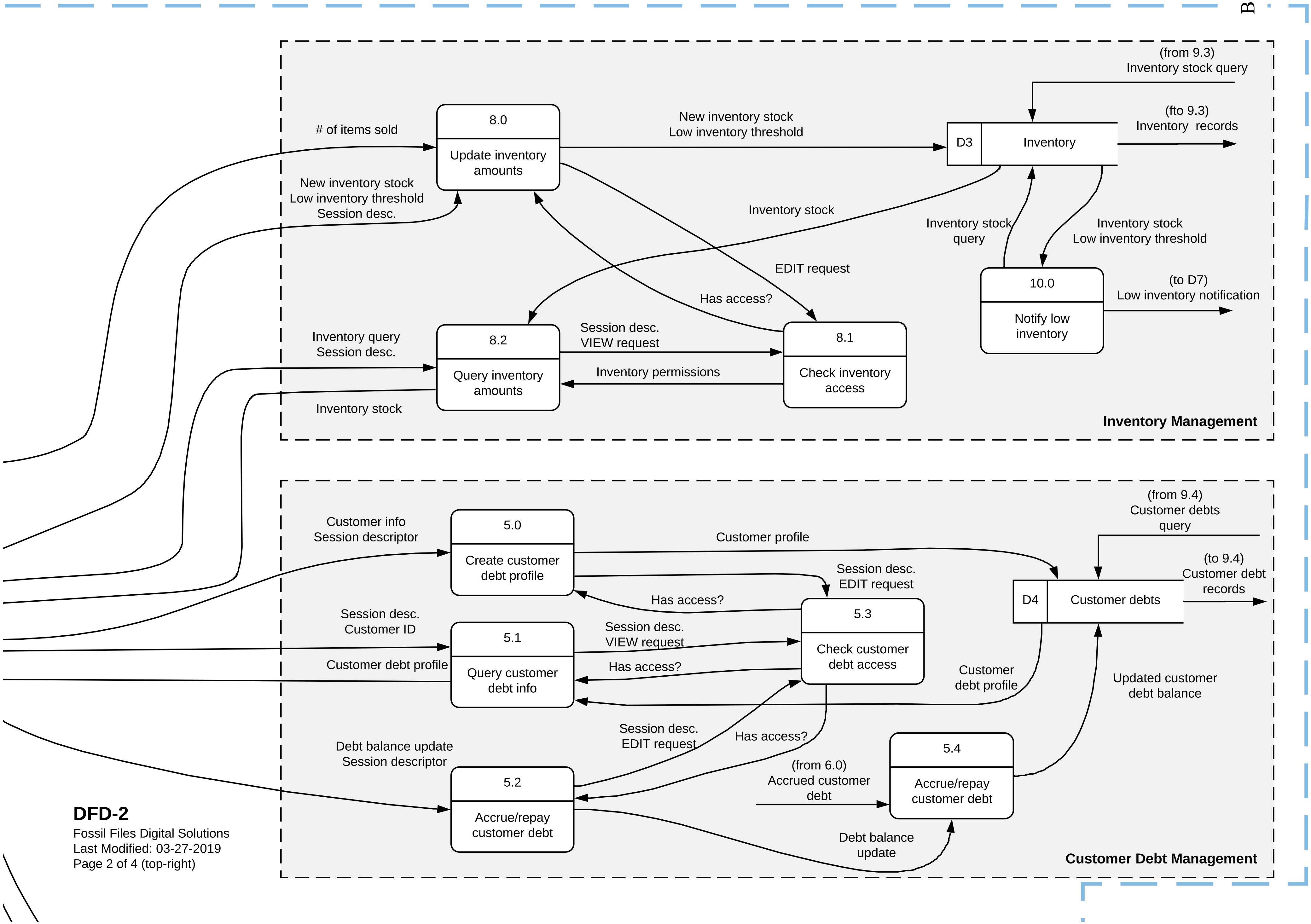



**DFD-2**

Fossil Files Digital Solutions

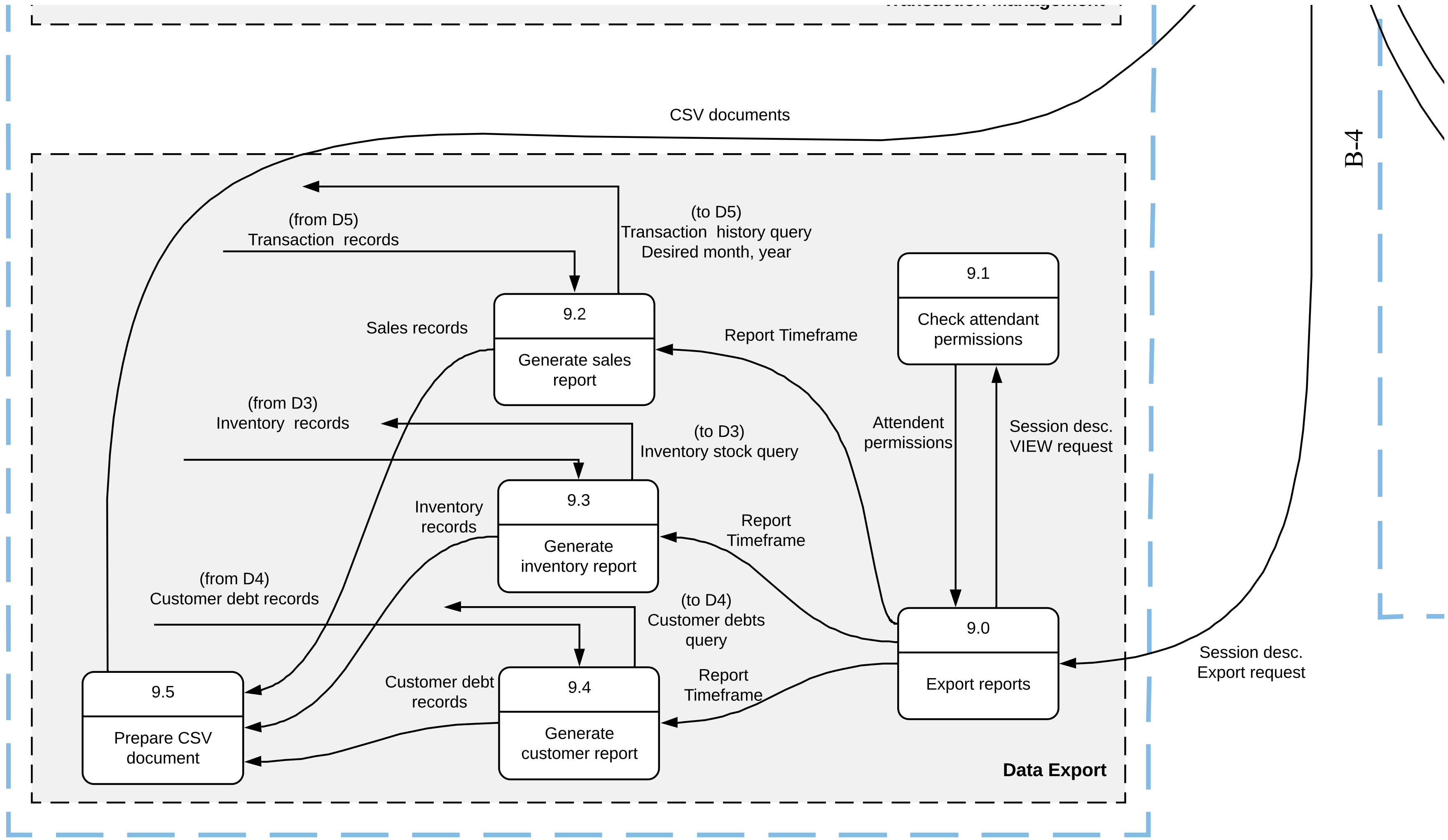
Last Modified: 03-27-2019

Page 1 of 4 (top-left)




**DFD-2**

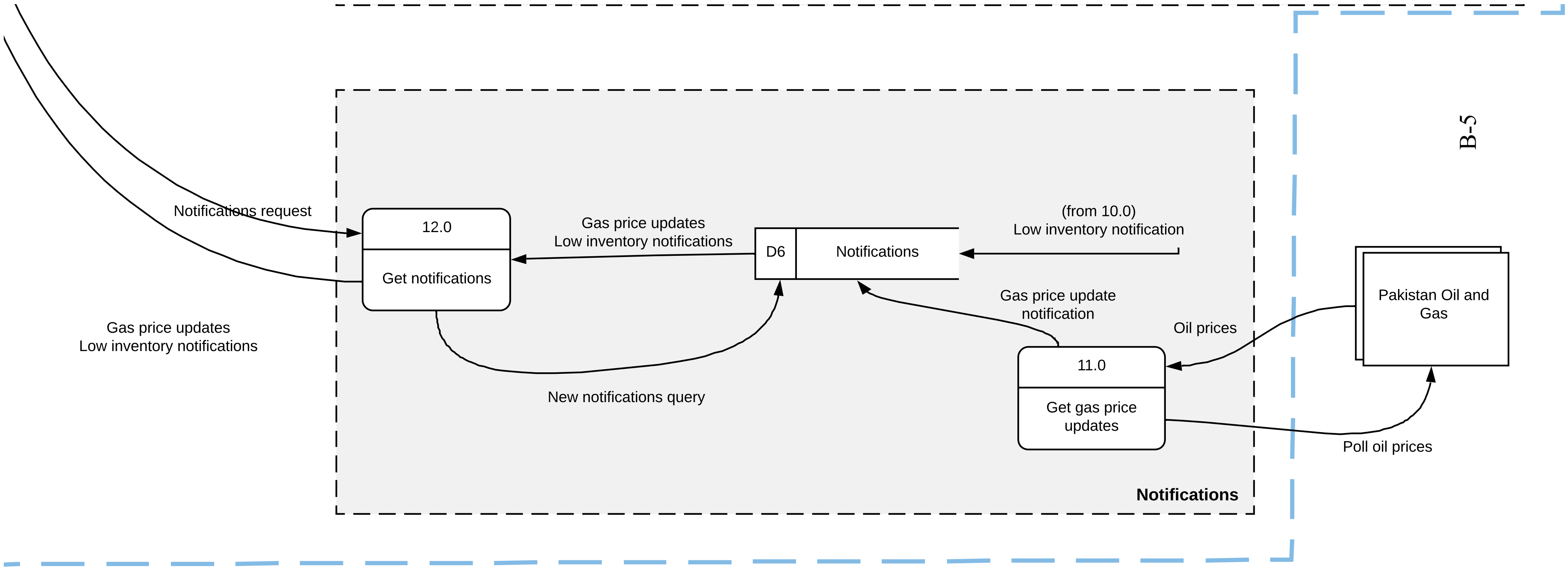
Fossil Files Digital Solutions
Last Modified: 03-27-2019
Page 2 of 4 (top-right)



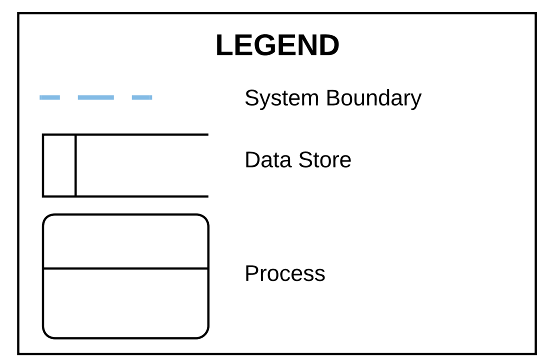



**DFD-2**

Fossil Files Digital Solutions
Last Modified: 03-27-2019
Page 3 of 4 (bottom-left)






**DFD-2**
Fossil Files Digital Solutions
Last Modified: 03-27-2019
Page 4 of 4 (bottom-right)