Daniel Reid Nelsen

Jennifer Pribbeno

Group 47 - Project Step 6 Final

FortySevenPharma Drug Database

Website URL: http://flip3.engr.oregonstate.edu:32456/

Executive Summary:

- Purchases Table Renamed: Originally named 'Sales,' it was renamed to 'Purchases' for accuracy in data representation.
- CamelCase Implementation: Attribute names across the database were converted to camelCase for consistency following peer feedback.
- M:M Relationship Clarification: Enhanced descriptions of many-to-many (M:M) relationships, particularly between the Medications and Customers tables, were included to simplify user understanding on the home page. Each webpage now contains clear explanations of its corresponding database relationships.
- Foreign Key Adjustment: Made the wholesalersID in the Medications table NULLable to offer more flexibility in database relationships following peer feedback.
- Prescriptions entity was deemed unnecessary for our purposes and was deleted.
- Separate Automations and Refrigerator entities were created following a normalization check instead of keeping them as attributes in Inventories.
- Wholesalers entity was created following a normalization check (Wholesalers was originally part of Medications)
- Unneeded Attributes Excluded: Opted not to include an expiration date in Medications, as this is managed manually by pharmacy staff.
- Edge cases and unexpected user input: following a peer review, unexpected input by users was addressed by putting restrictions on inputs such as bottleCountSize (negative numbers were not permitted) and costToBuy (negative numbers prohibited and decimal points were rounded to the nearest 2 places). Formulation was also made to be a dropdown option rather than user input to decrease errors.
- UI: Upon peer reviews and our group's own experience with the site, we added a confirmation for DELETE functions, made hyperlinks in the table buttons to clarify they were interactive and not line items, and aligned the table headers with the order and wording of input fields for ADD and UPDATE throughout the site.

Conclusion:

The iterative process of incorporating peer feedback has been instrumental in evolving our database design and UI. These changes not only made the system more intuitive and efficient but also ensured better data management and user satisfaction. We remain committed to refining our project, with a focus on user experience and the integrity of our database design.

Project Outline:

FortySevenPharma processes roughly 60,500 prescriptions each year, utilizing an inventory of over 2,500 distinctive medications and generating $6 million in sales, which equates to an average of $100 per prescription. FortySevenPharma is attempting to refinance its banknote, but doesn't know the value of its current inventory cost. Additionally, all purchases and inventory management are recorded in a hand ledger. The proposed database-driven website integrated with barcode scanning technology will satisfy the requirement from the bank lender and also enhance inventory management by recording Purchases of Medications from both wholesalers and Customers. By scanning Medications as they're sold or restocked, the system will provide real-time updates on shelf contents, ensuring accurate tracking of sales and a precise grasp on current inventory levels. All sales that are recorded in the database will be from a legal prescription. This approach allows FortySevenPharma to ensure medicines are always available when needed, streamlining operations, and elevating customer service.

Database Outline:

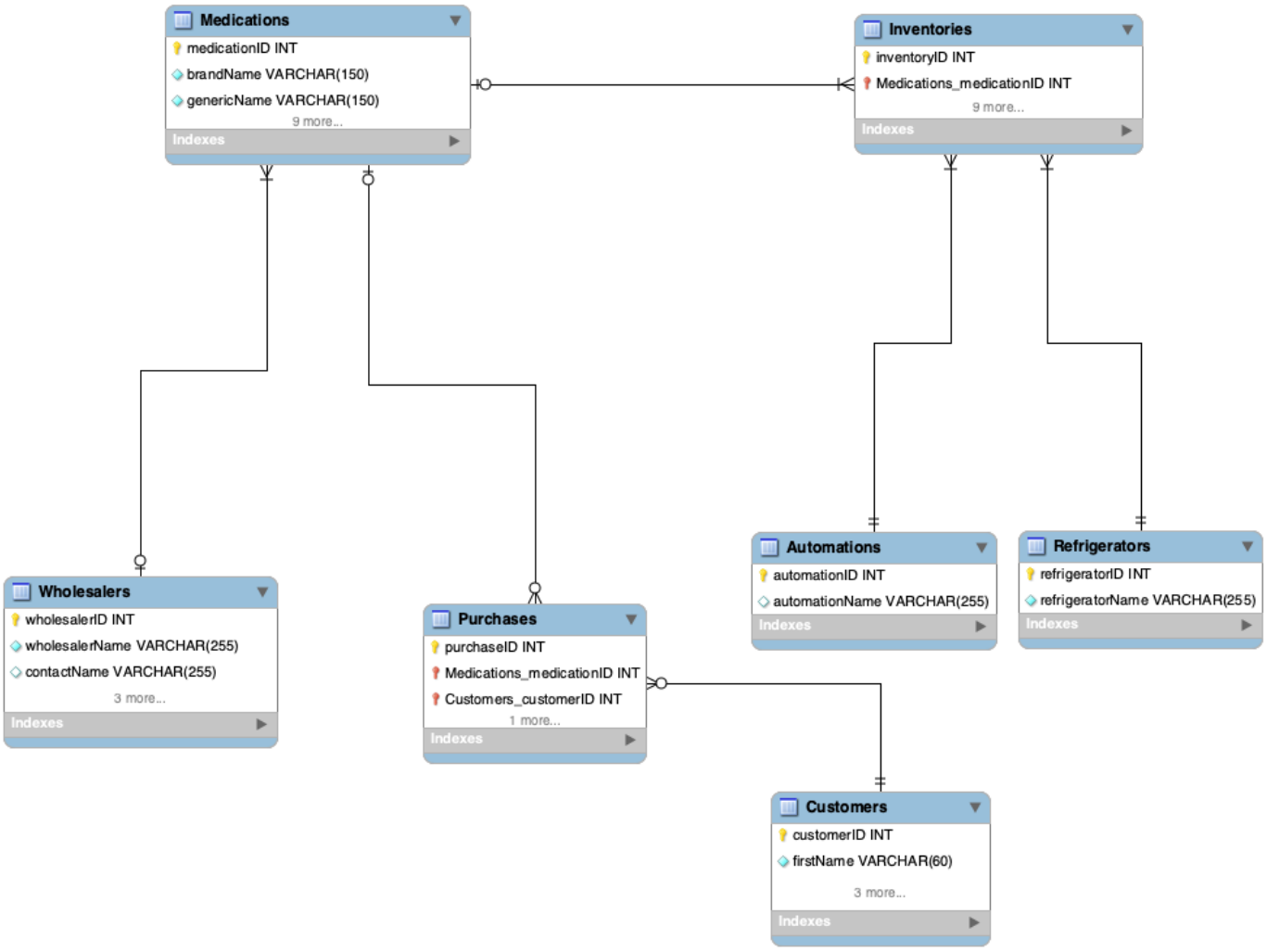
- **Medications**: records details of each drug in stock, containing comprehensive information about every medication available in the inventory. Each row corresponds to a unique medication, identifiable by the medicationID
    - medicationID: int, auto_increment, unique, not NULL, PK
    - brandName: varchar, not NULL - the official name given by the manufacturer
    - genericName: varchar, not NULL - the name of the medication without branding
    - strength: varchar, not NULL - the potency of the medication
    - formulation: varchar, not NULL - the composition of the drug, e.g., tablet, capsule
    - upcNumber: varchar, not NULL - Universal Product Code, a unique identifier
    - ndcNumber: varchar, unique, not NULL - National Drug Code, a unique identifier but can be non-unique for over-the-counter meds.
    - costToBuy: float - Cost price of the medication.
    - bottleCountSize: varchar, not NULL - Indicates how many tablets or capsules in a bottle.
    - wholesalerID: int, FK to Wholesalers - The wholesaler from whom the drug was purchased.
    - lastPurchased: date - Date of the last purchase from the wholesaler.
    - **Relationships:**

        - **Medications to Inventories (1:M)** using medicationID**:** One medication can have multiple inventory records. For instance, the same medication could be stocked in different quantities or at different times, leading to multiple rows in the Inventories table, but all linked to the same medication.
        - **Medications to Sales (1:M)** using medicationID**:** A single medication can be part of numerous sales transactions. This captures the real-world scenario where multiple customers might purchase the same medication over time.
- **Wholesalers:** keeps track of wholesaler information
    - wholesalerID: int, auto_increment, unique, not NULL, PK
    - wholesalerName: varchar not NULL - Name of primary contact
    - contactEmail: varchar - primary contact's email address
    - contactPhoneNumber: varchar - primary contact's phone number
    - address: varchar
    - **Relationships:**
        - Wholesalers to Medications (1:M) using wholesalerID: a single wholesaler can provide many medications to the pharmacy.
- **Inventories**: manages the stock of each Medication, which will allow tracking of the current stock, gathering information on ideal stock levels, and tracking of specific storage locations for Medication.
    - inventoryID: int, auto_increment, unique, not NULL, PK
    - medicationID: int, FK to Medications - Links the inventory to a specific medication
    - currentInventory: int - Current stock level of the medication.
    - minInventory: int - Minimum stock level before restocking is required.
    - maxInventory: int - Maximum stock level to be maintained.
    - locationShelf: varchar - Specific shelf where the medication is stored.
    - locationBox: varchar - Specific box or bin on shelf where the medication is stored.
    - locationAutomation: int, FK to Automations - If the drug is stored in automated systems, its location.
    - locationInRefrigerator: int, FK to Refrigerators - Location in the refrigerator if the drug requires cold storage.
    - **Relationship:**
        - **Inventories to Medications (M:1)** using medicationID**:** Each inventory record pertains to one specific medication. However, as medications can have multiple inventory records, it establishes an M:1 relationship

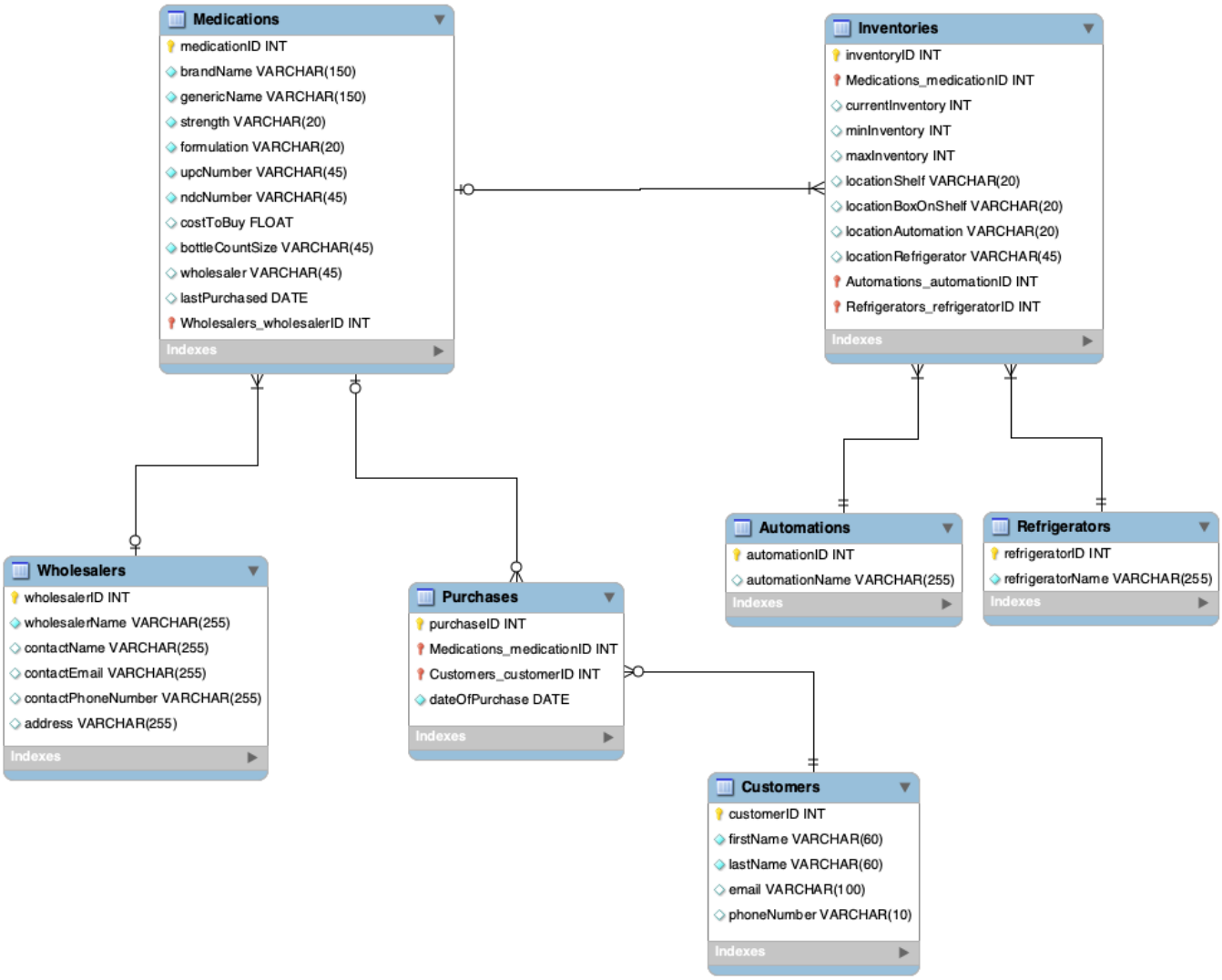
        - **Inventories to Automations (M:1)** using locationAutomation: Each inventory record that is stored in an automation system is associated with a specific automation robot. This relationship is optional as not all inventory items are stored in automation systems.
        - **Inventories to Refrigerators (M:1)** using locationInRefrigerator: Each inventory record that requires refrigeration is associated with a specific refrigerator. This relationship is also optional as not all inventory items require refrigeration.
- **Purchases**: records details of each sales transaction. Each row represents the sale of a medication to a customer on a particular date. Intersection table for Medications and Customers.
    - purchaseID: int, auto_increment, unique, not NULL, PK - Unique identifier for each purchase.
    - medicationID: int, FK to Medications - Links the purchase to a specific medication.
    - customerID: int, FK to Customers - Links purchase to specific customer
    - dateOfPurchase: date, not NULL - The date the purchase was made.
    - **Relationships:**
    - **A M:M relationship with Medications and Customers** is facilitated with a M:1 relationship with both Medications and Customers with foreign keys MedicationID and CustomerID inside the Purchases table.:
        - Purchases to Medications (M:1) and Purchases to Customers (M:1): A purchase corresponds to a specific medication and a specific customer. When combined, these two M:1 relationships effectively model the M:M relationship between Medications and Customers, as one medication can be sold to many customers and vice versa. The Purchases table acts as a bridge to capture this many-to-many interaction.
- **Customers**: records details of customers making purchases or holding prescriptions
    - customerID: int, auto_increment, unique, not NULL, PK
    - firstName: varchar, not NULL
    - lastName: varchar, not NULL
    - email: varchar, unique
    - phoneNumber: varchar
    - **Relationships:**
        - **Customers to Sales (1:M)** using customerID**:** A customer can make multiple purchases, each represented as a unique sale in the Sales table.

- **Automations**: manages and tracks medications stored in automated dispensing system
    - automationID: int, auto_increment, not NULL, PK
    - automationName: varchar, not NULL
    - **Relationships:**
        - **Automations to Inventories (1:M)** using automationID: A specific location within one of the automation systems can store multiple different medications (represented as inventory entries). However, each inventory entry is stored at one specific location within the automation system or none if not stored there. This is modeled as a 1:M relationship.
- **Refrigerators:** manages and tracks medications that require refrigeration
    - refrigeratorID: int, auto_increment, not NULL, PK
    - refrigeratorName: varchar not NULL
    - **Relationships:**
        - **Refrigerators to Inventories (1:M)** using refrigeratorID: A specific location one of many possible refrigerators can store multiple different medications (represented as inventory entries). However, each inventory entry is stored at one specific location in one of the refrigerators or none if not required to be stored cold. This is modeled as a 1:M relationship.

ER Diagram:


Medications
medicationID INT
brandName VARCHAR(150)
genericName VARCHAR(150)
9 more...
Indexes
Inventories
inventoryID INT
Medications_medicationID INT
9 more...
Indexes
Wholesalers
wholesalerID INT
wholesalerName VARCHAR(255)
contactName VARCHAR(255)
3 more...
Indexes
Purchases
purchaseID INT
Medications_medicationID INT
Customers_customerID INT
1 more...
Indexes
Automations
automationID INT
automationName VARCHAR(255)
Indexes
Refrigerators
refrigeratorID INT
refrigeratorName VARCHAR(255)
Indexes
Customers
customerID INT
firstName VARCHAR(60)
3 more...
Indexes

Schema:


Medications
medicationID INT
brandName VARCHAR(150)
genericName VARCHAR(150)
strength VARCHAR(20)
formulation VARCHAR(20)
upcNumber VARCHAR(45)
ndcNumber VARCHAR(45)
costToBuy FLOAT
bottleCountSize VARCHAR(45)
wholesaler VARCHAR(45)
lastPurchased DATE
Wholesalers_wholesalerID INT
Indexes
Inventories
inventoryID INT
Medications_medicationID INT
currentInventory INT
minInventory INT
maxInventory INT
locationShelf VARCHAR(20)
locationBoxOnShelf VARCHAR(20)
locationAutomation VARCHAR(20)
locationRefrigerator VARCHAR(45)
Automations_automationID INT
Refrigerators_refrigeratorID INT
Indexes
Wholesalers
wholesalerID INT
wholesalerName VARCHAR(255)
contactName VARCHAR(255)
contactEmail VARCHAR(255)
contactPhoneNumber VARCHAR(255)
address VARCHAR(255)
Indexes
Purchases
purchaseID INT
Medications_medicationID INT
Customers_customerID INT
dateOfPurchase DATE
Indexes
Automations
automationID INT
automationName VARCHAR(255)
Indexes
Refrigerators
refrigeratorID INT
refrigeratorName VARCHAR(255)
Indexes
Customers
customerID INT
firstName VARCHAR(60)
lastName VARCHAR(60)
email VARCHAR(100)
phoneNumber VARCHAR(10)
Indexes

Sample Data:

Medications Entity:

medicationID	brandName	genericName	strength	formulation	upcNumber	ndcNumber	costToBuy	bottleCountSize	wholesalerID	lastPurchased
1	Advil	Ibuprofen	200mg	Tablet	123456	12345678913	10	100	1	2023-10-01
2	Tylenol	Acetaminophen	500mg	Tablet	123457	12343465211	15.23	100	2	2023-09-15
3	ProAir	Albuterol	90mcg	Inhaler	745869230	98765132113	23.5	1	3	2023-10-10
4	Aleve	Naproxen	220mg	Tablet	32251425	65132165461	8.99	100	4	2023-09-10

Purchases Entity:

purchaseID	medicationID	customerID	dateOfPurchase
1	1	1	2023-10-05
2	2	2	2023-10-06
3	3	3	2023-10-07
4	4	4	2023-10-06

Customers Entity:

customerID	firstName	lastName	email	phoneNumber
1	Walter	White	heisenberg@gmail.com	123-456-7890
2	Jesse	Pinkman	jesse@hotmail.com	623-765-4321
3	Saul	Goodman	bettercallsaul@outlook.com	321-654-9870
4	Benjamin	Linus	thothers@yahoo.com	602-765-4321

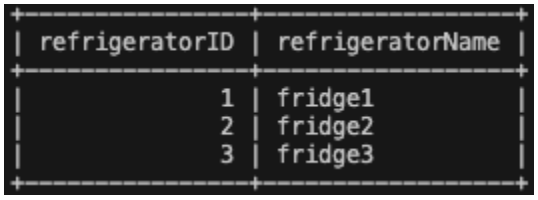
Wholesalers Entity:

wholesalerID	wholesalerName	contactName	contactEmail	contactPhoneNumber	address
1	Dharma Pharma	Desmond Hume	dhume@dharma.com	555-0123	108 Main St
2	InGen Medical Innovations	John Hammond	jhammond@ingen.com	555-0223	Jurassic Park
3	Krusty Krab Health Solutions	Eugene Krabs	mkrabs@krustykrab.com	555-0323	Bikini Bottom
4	Stark Health Tech	Tony Stark	tstark@starkindustries.com	555-0423	10880 Malibu Point

Inventories Entity:

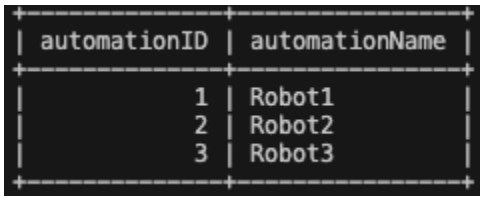
inventoryID	medicationID	currentInventory	minInventory	maxInventory	locationShelf	locationBox	locationAutomation	locationInRefrigerator
1	1	50	10	100	A1	B2	1	NULL
2	2	30	5	50	A2	B3	1	NULL
3	3	20	2	25	A3	B4	NULL	1
4	4	60	10	120	A4	B5	1	NULL

Refrigerators Entity:



refrigeratorID	refrigeratorName
1	fridge1
2	fridge2
3	fridge3

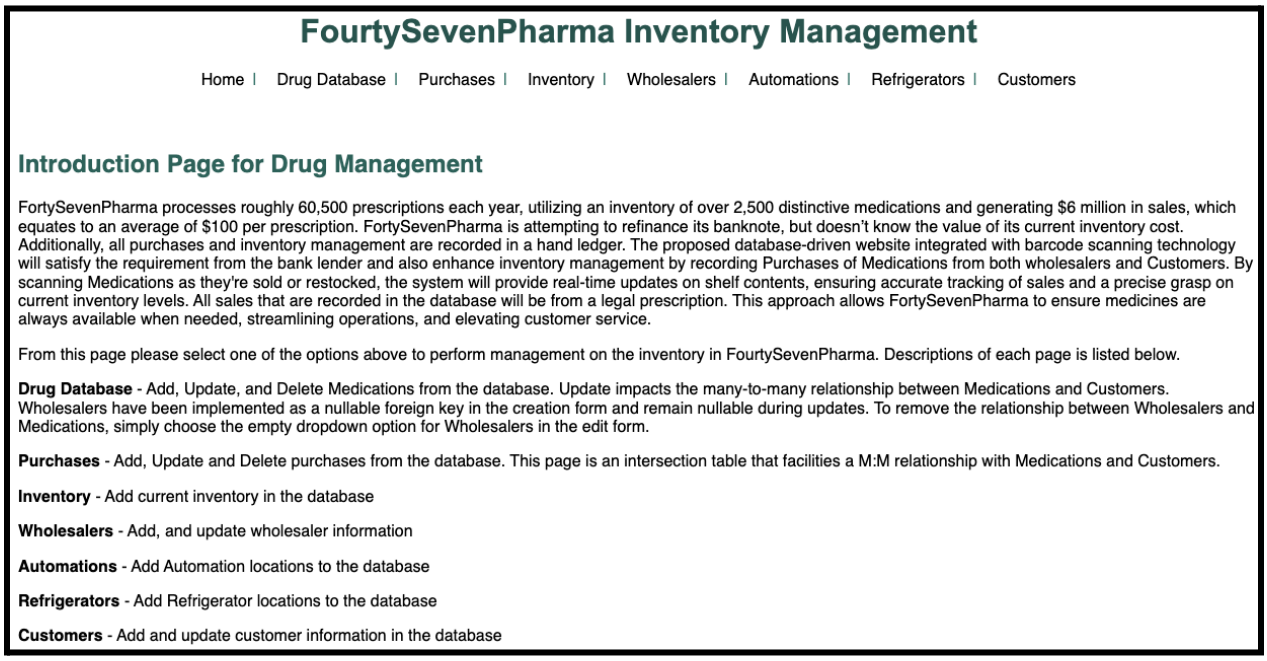
Automations Entity:



automationID	automationName
1	Robot1
2	Robot2
3	Robot3

UI Screen Shots with Informative Titles

1. Home: Contains project overview and page descriptions



FourtySevenPharma Inventory Management

Home | Drug Database | Purchases | Inventory | Wholesalers | Automations | Refrigerators | Customers

Introduction Page for Drug Management

FortySevenPharma processes roughly 60,500 prescriptions each year, utilizing an inventory of over 2,500 distinctive medications and generating $6 million in sales, which equates to an average of $100 per prescription. FortySevenPharma is attempting to refinance its banknote, but doesn't know the value of its current inventory cost. Additionally, all purchases and inventory management are recorded in a hand ledger. The proposed database-driven website integrated with barcode scanning technology will satisfy the requirement from the bank lender and also enhance inventory management by recording Purchases of Medications from both wholesalers and Customers. By scanning Medications as they're sold or restocked, the system will provide real-time updates on shelf contents, ensuring accurate tracking of sales and a precise grasp on current inventory levels. All sales that are recorded in the database will be from a legal prescription. This approach allows FortySevenPharma to ensure medicines are always available when needed, streamlining operations, and elevating customer service.

From this page please select one of the options above to perform management on the inventory in FourtySevenPharma. Descriptions of each page is listed below.

Drug Database - Add, Update, and Delete Medications from the database. Update impacts the many-to-many relationship between Medications and Customers. Wholesalers have been implemented as a nullable foreign key in the creation form and remain nullable during updates. To remove the relationship between Wholesalers and Medications, simply choose the empty dropdown option for Wholesalers in the edit form.

Purchases - Add, Update and Delete purchases from the database. This page is an intersection table that facilities a M:M relationship with Medications and Customers.

Inventory - Add current inventory in the database

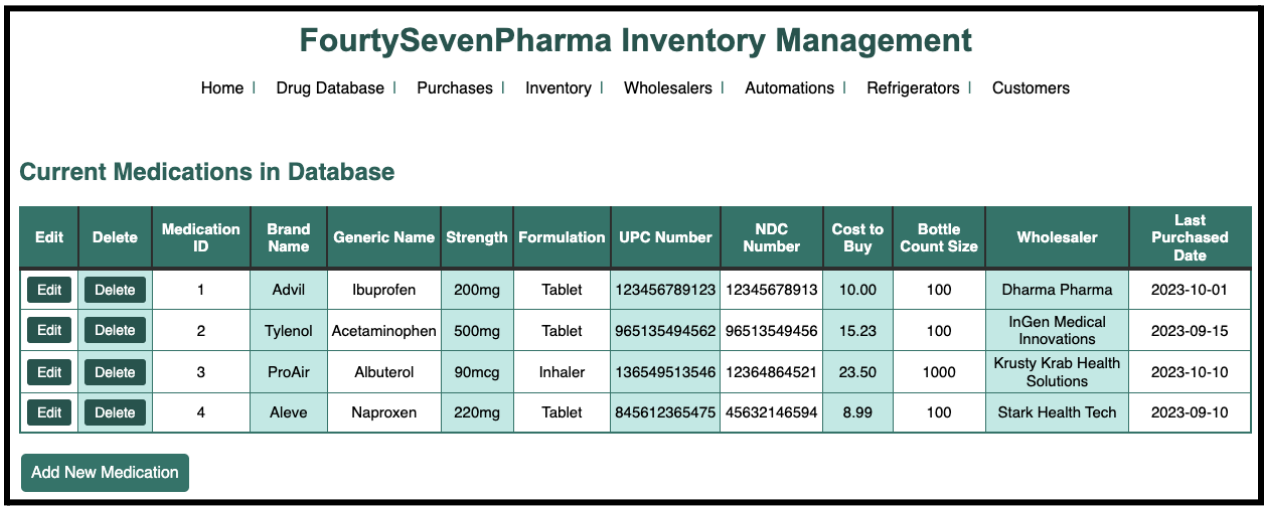
Wholesalers - Add, and update wholesaler information

Automations - Add Automation locations to the database

Refrigerators - Add Refrigerator locations to the database

Customers - Add and update customer information in the database

2. READ/BROWSE/DISPLAY/DELETE Medications on Drug Database page:



FourtySevenPharma Inventory Management

Home | Drug Database | Purchases | Inventory | Wholesalers | Automations | Refrigerators | Customers

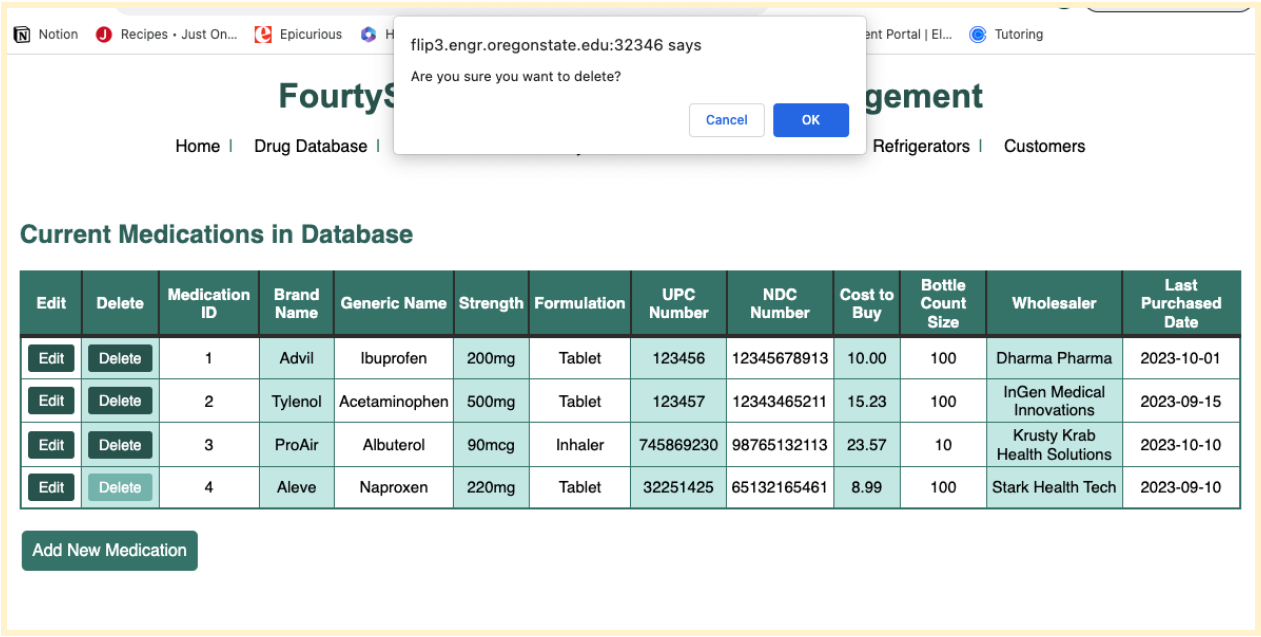
Current Medications in Database

Edit	Delete	Medication ID	Brand Name	Generic Name	Strength	Formulation	UPC Number	NDC Number	Cost to Buy	Bottle Count Size	Wholesaler	Last Purchased Date
Edit	Delete	1	Advil	Ibuprofen	200mg	Tablet	123456789123	12345678913	10.00	100	Dharma Pharma	2023-10-01
Edit	Delete	2	Tylenol	Acetaminophen	500mg	Tablet	965135494562	96513549456	15.23	100	InGen Medical Innovations	2023-09-15
Edit	Delete	3	ProAir	Albuterol	90mcg	Inhaler	136549513546	12364864521	23.50	1000	Krusty Krab Health Solutions	2023-10-10
Edit	Delete	4	Aleve	Naproxen	220mg	Tablet	845612365475	45632146594	8.99	100	Stark Health Tech	2023-09-10

Add New Medication

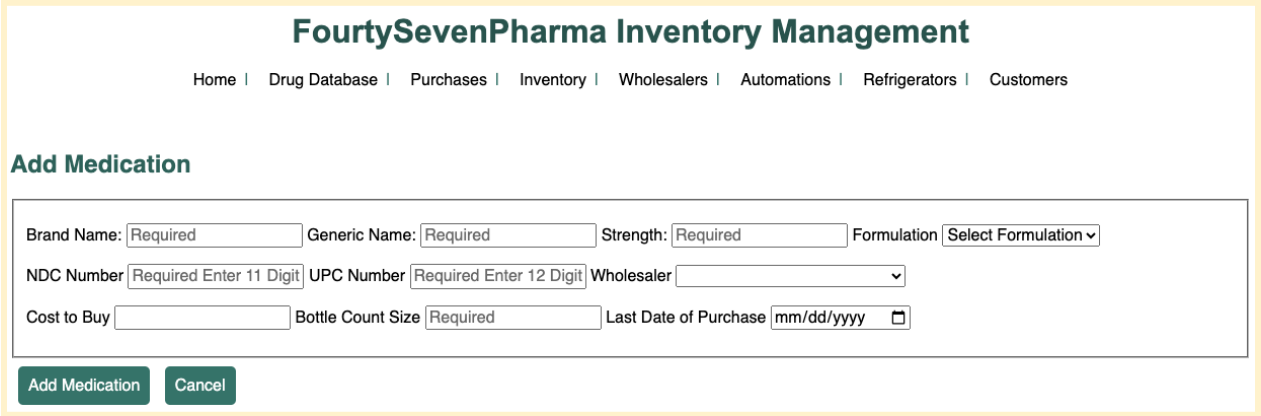
**This deletes an item from the Medications table, which is in a M:N relationship with Customers. Purchases intersection table facilitates the M:N relationship and deleting a medication from this table will also delete all purchases containing the deleted medicationID.

3. DELETE Confirmation on Medications Page:



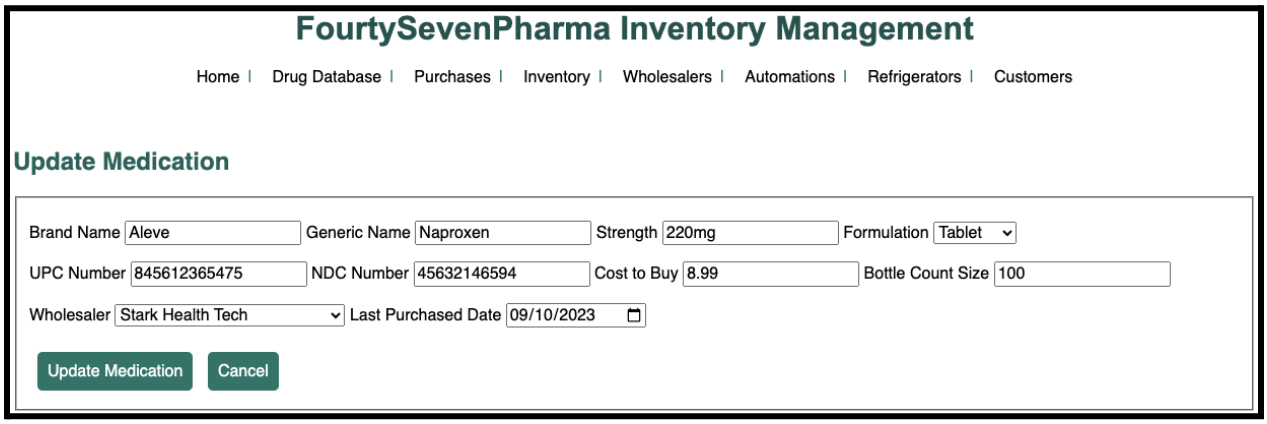
4. CREATE/INSERT/ADD NEW Medication on Drug Database page:

**Wholesaler is a NULLable foreign key. It can be left blank or it can be selected from the dropdown.

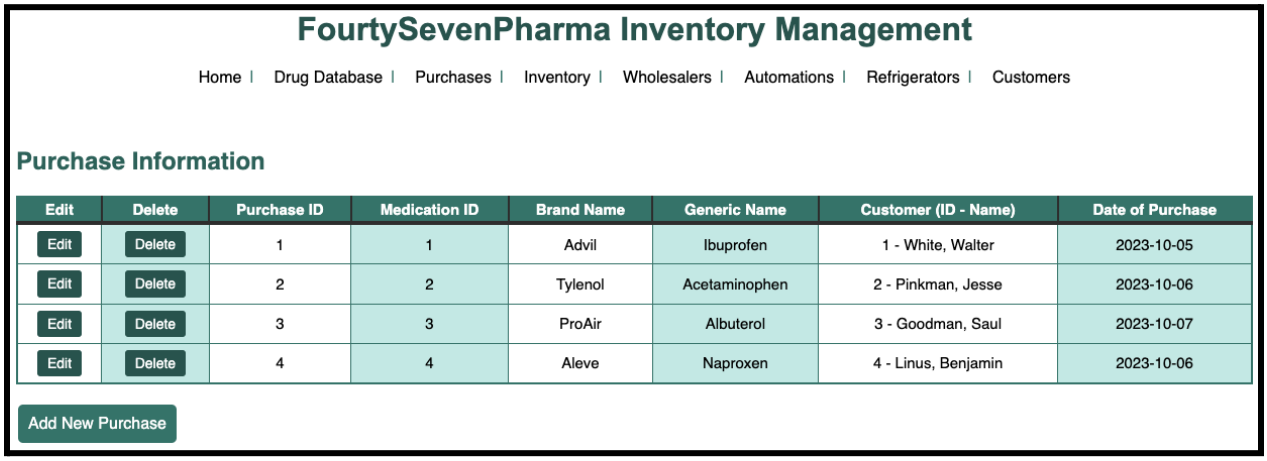


5. EDIT/UPDATE Medication and NULLable FK on Drug Database page:

**This updates an item from the Medications table, which is in a M:N relationship with Customers. Purchases intersection table facilitates the M:N relationship. Wholesaler is a NULLable foreign key and an existing relationship can be updated to NULL by selecting the empty value in the dropdown box.

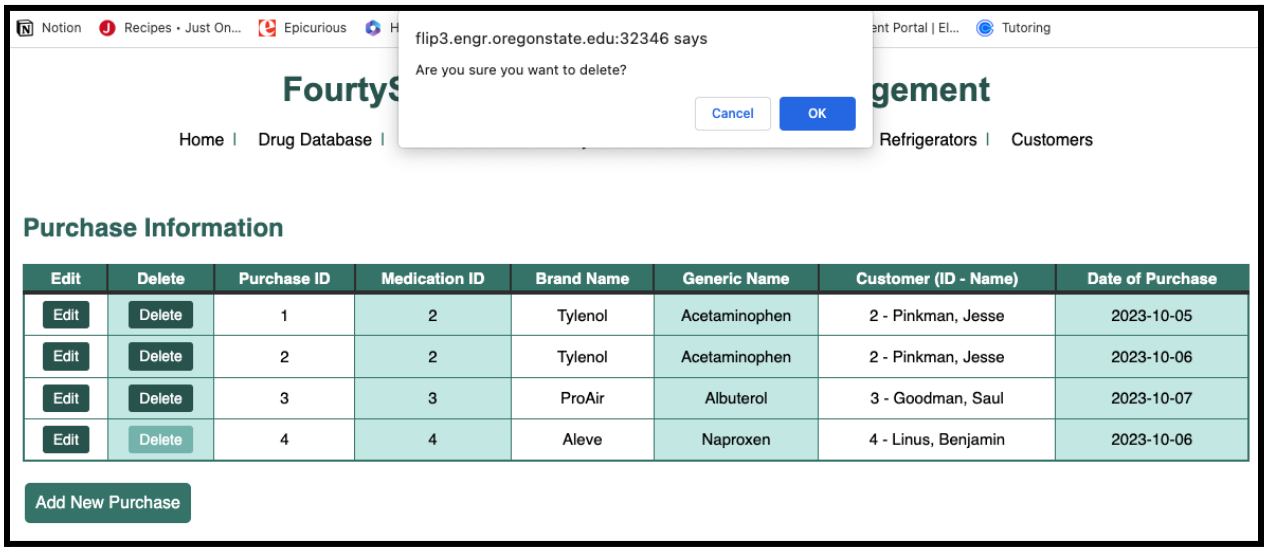


6. READ/BROWSE/DISPLAY/DELETE on Purchases page:



7. DELETE Confirmation on Purchases Page:

**Purchases facilitates M:N relationship between Medications and Customers, but deleting a purchase here will not delete the associated customers or medications from their respective tables.

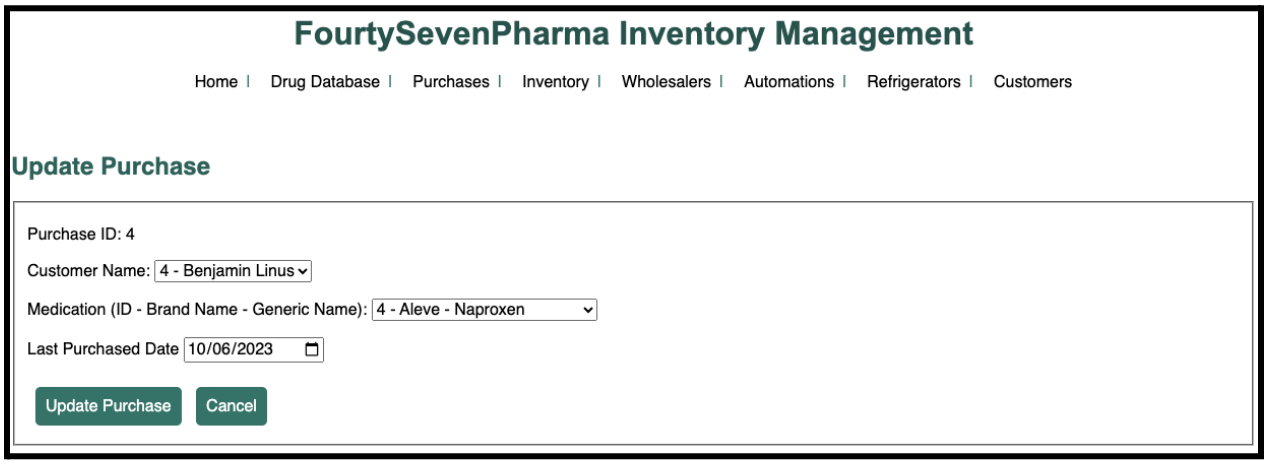


8. CREATE/INSERT/ADD NEW Purchase on Purchases Page:

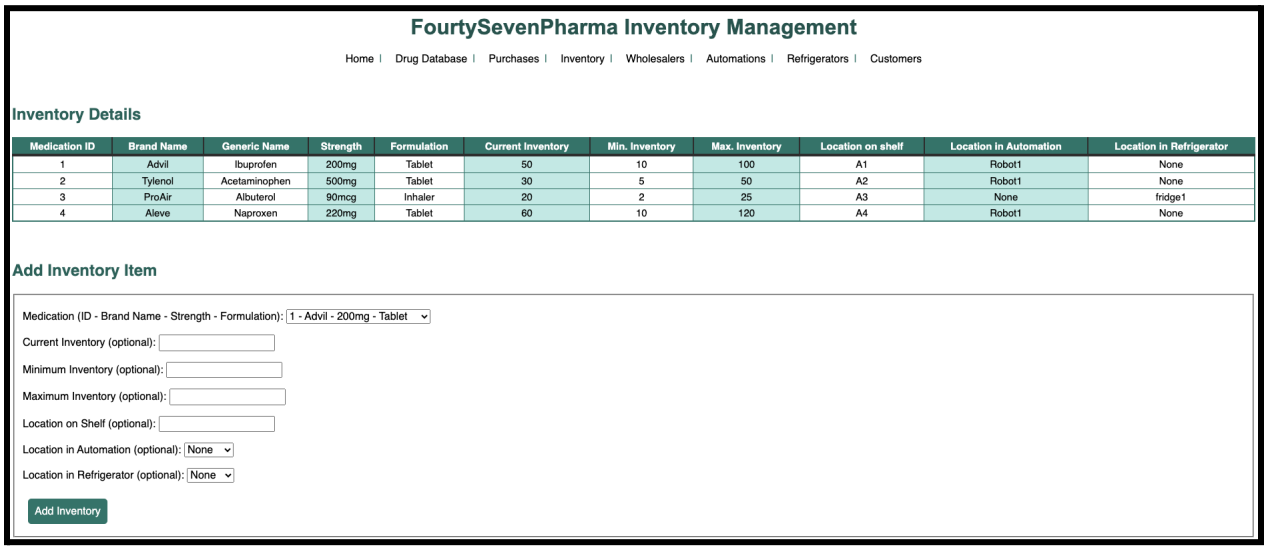
**Purchases facilitates a M:N relationship between Medications and Customers but does not add new medications or customers to those respective tables when a purchase is added. Medications and Customers are selected via dynamic dropdown boxes.

9. EDIT/UPDATE Purchase on Purchases Page:

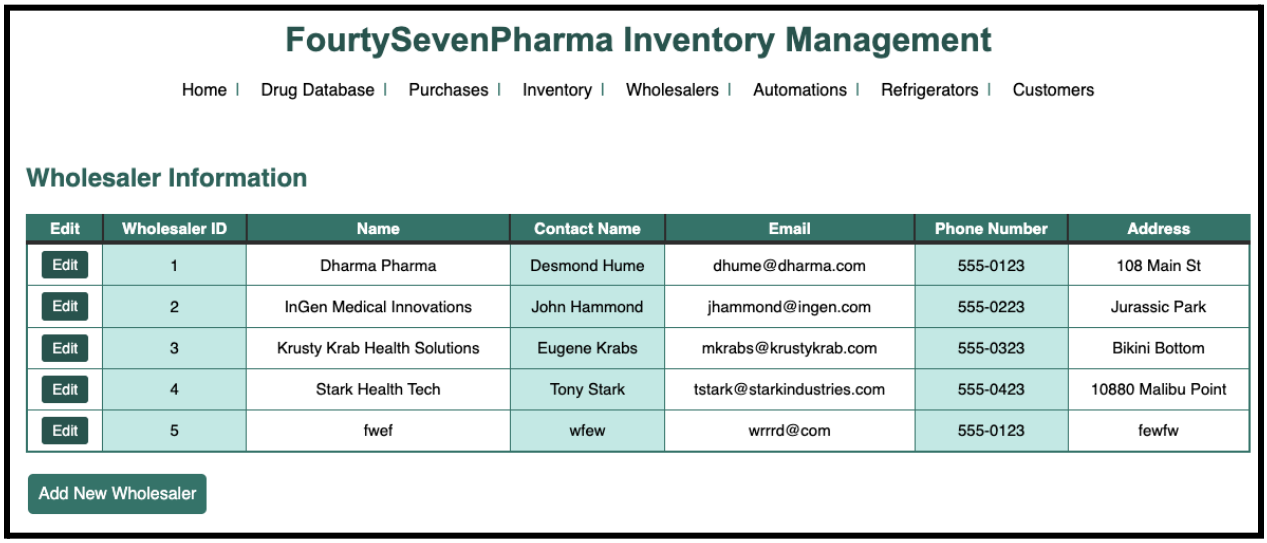
**This updates an item from the Purchases intersection table using dropdowns for selecting from the Medication and Customer data.



10. READ/BROWSE/DISPLAY/CREATE/INSERT/ADD NEW Inventory item on Inventory page:

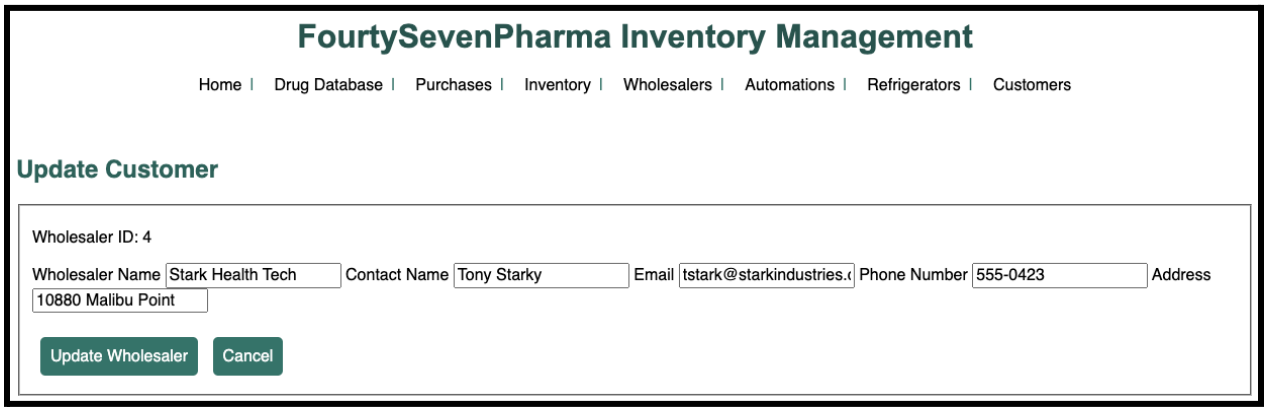


11. READ/BROWSE/DISPLAY Wholesaler on Wholesalers page:

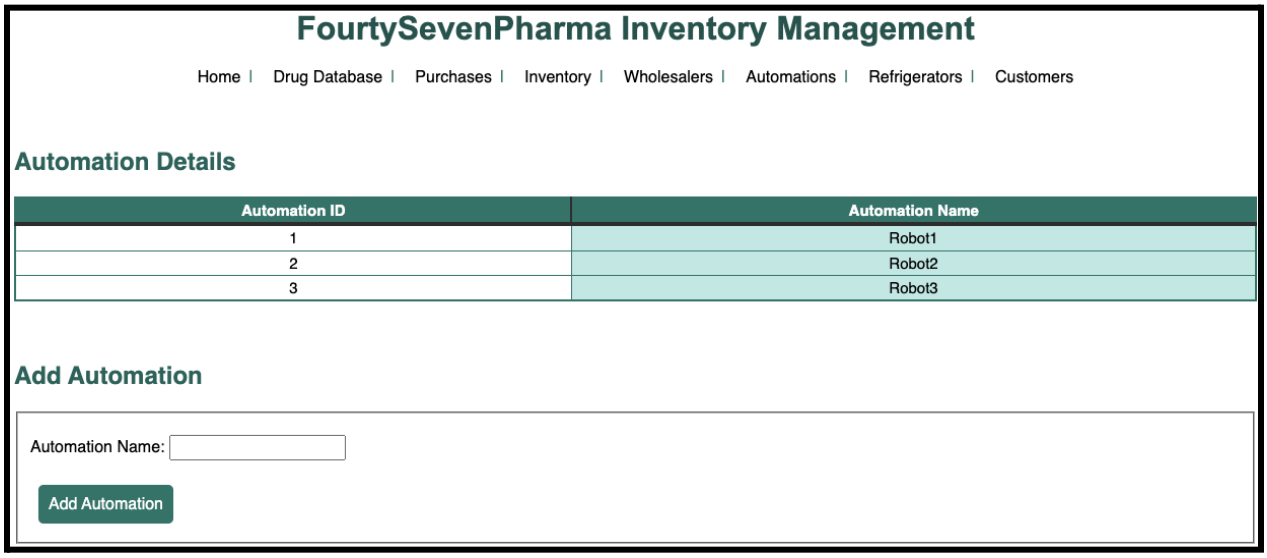


12. CREATE/INSERT/ADD NEW Wholesaler on Wholesalers Page:

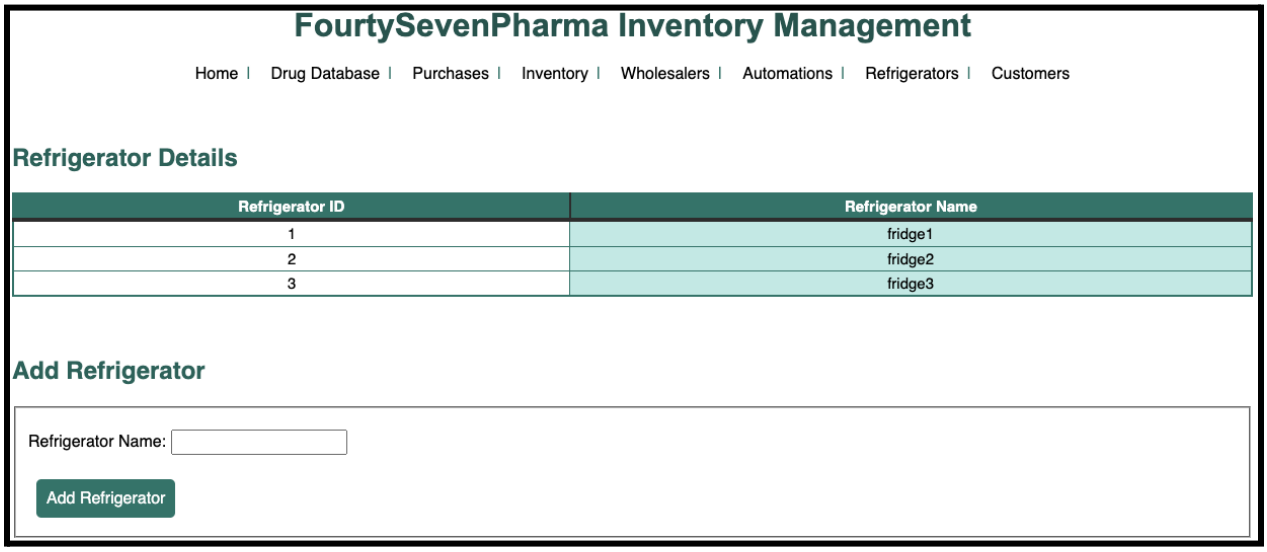
13. EDIT/UPDATE Wholesaler on Wholesalers Page:



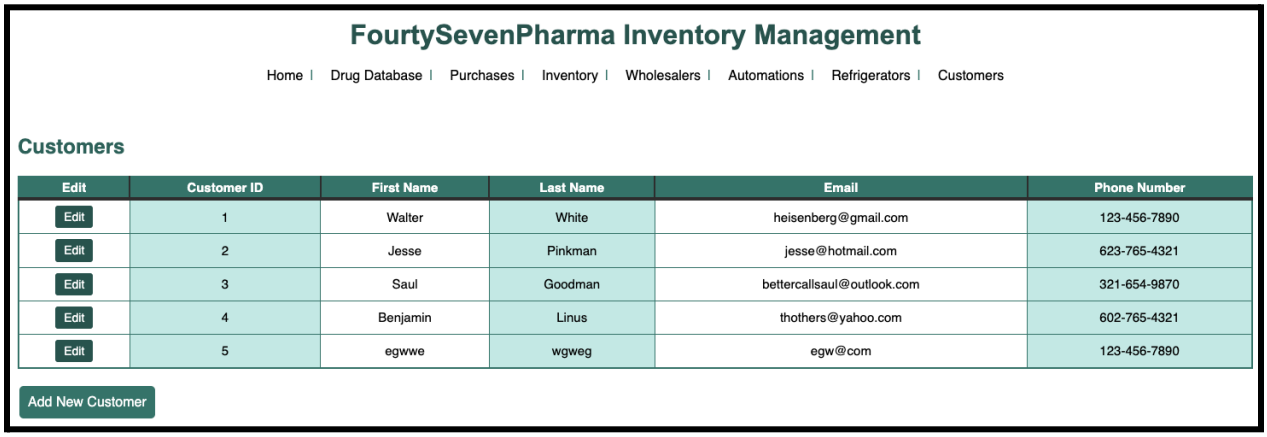
14. READ/BROWSE/DISPLAY/ADD/CREATE Automation on Automations Page:



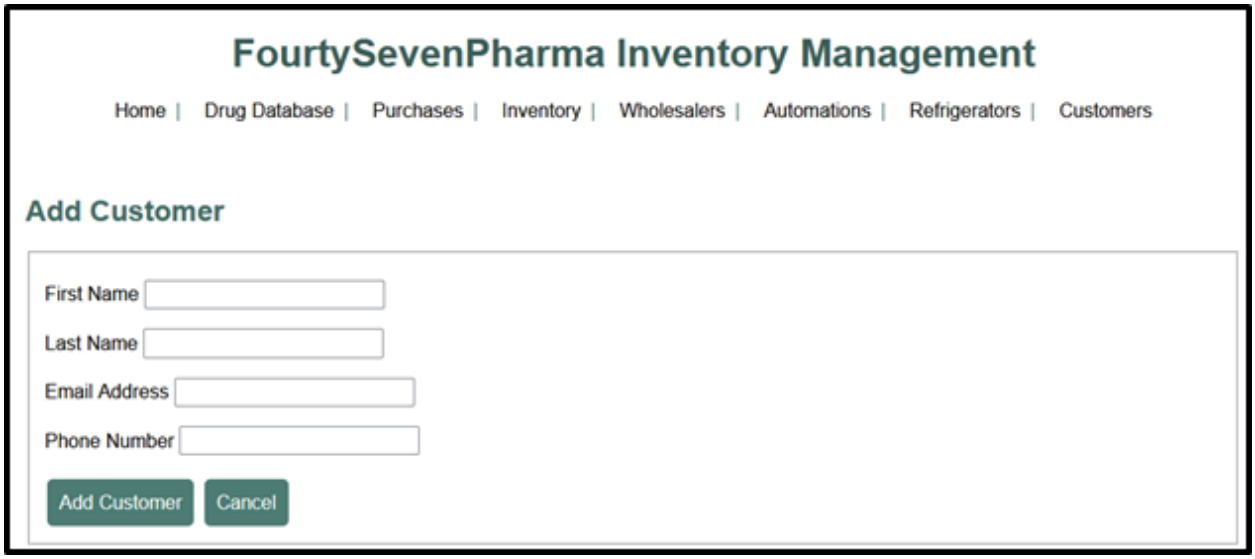
15. READ/BROWSE/DISPLAY/ADD/CREATE Refrigerator on Refrigerators Page:



16. READ/BROWSE/DISPLAY Customers on Customers Page:



17. CREATE/INSERT/ADD NEW Customer on Customers Page:



18. EDIT/UPDATE Customer on Customers Page: