Software Engineering Software Requirements Specification (SRS) Document

[PIN Activated Gas Pump]

https://github.com/ployphemus/PIN-Activated-Gas-Pump

[4/24/2023]

By:
William Vaughan
Grayson Williams
(Sections have annotations in parentheses to indicate who worked on what)

Table of Contents

1. Introduction (William and Greyson)

1.1. Purpose

The goal is to implement a simple gas pump control program for use in a fleet environment.

1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements for the PIN Gas Pump (PGP). Client-oriented requirements describe the system from the client's perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer's perspective. These requirements include a detailed description of functional, data, performance, and other important requirements.

1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager.
CSV	Comma Separated Values file, used for storing data
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
Spring Web	Will be used to access the remote API. This is one of the dependencies of our system.
IntelliJ	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function within the software where the estimated price of delivered fuel is calculated.
PIN	Personal Identification Number

1.4. Intended Audience

Document is intended for dev team and management; however, section 5.8 and 5.9 can be used by marketing to explain software functions to customers.

1.5. Project Scope

The goal of the software is to provide a robust, simple pump control system for use in off grid fleet environments. This includes construction, oil fields, military, farms, and other harsh environments. This requires a simple system that requires little to no maintenance and can run off of a small solar array or generator. An interface with high legibility and contrast is also required due to possible use in low visibility conditions. Additionally, where an internet connection is available, a fuel truck driver can get an estimate of the cost for delivered fuel.

1.6. Technology Challenges

Possibly the implementation of a Text or Graphics Based Interface (TUI or GUI). This may or may not be included based on time constraints. While not a concern in the first release, future version that are intended for use in integrated firmware applications would prioritize low power usage.

1.7. References

Alred, F., Brusaw, C., and Oliu, W. (2003). Handbook of Technical Writing (7th ed.). Boston: Bedford/St. Martin’s.

2. General Description (William and Greyson)

2.1. Product Perspective

PGP is a pump control system that is intended to be used in varied environments where a simple, robust system is required.

2.2. Product Features

The product features include the ability for fleet operators to create simple profiles to keep track of fuel use and the ability for administrators to manipulate those profiles. Normal users can access the pump and view lifetime fuel use through a simple PIN system. Fuel tanker truck drivers can also access the system to add fuel to the storage tank. Admins can add and remove users, check storage tank level, and enable or disable the pump. Possible additions include: low tank alerts for admin, capability to manage multiple pumps.

2.3. User Class and Characteristics

The software is intended to be used with minimal to no training. Basic users only have access to between two and four options after entering their PIN including pumping gas and checking lifetime gas pumped. Tanker drivers would only have access to add gas to the tank and check lifetime gas added. Admins will have access to five options including adding and removing users, turning the pump on and off, checking tank level, and pumping gas.

2.4. Operating Environment

The software is intended to be used in a simple, inexpensive system running a stripped down Windows or Linux OS. Ideally, a bootup script would automatically enable the program when the system is switched on. In the future, the program could be redeveloped to work in a low power microcontroller based system.

2.5. Constraints

Other than the deadline, possibly the implementation of a GUI.

2.6. Assumptions and Dependencies

The software will be dependent on Spring Web framework to create and execute the text based architecture that will be developed within IntelliJ. The application will also use the GasBuddy API to estimate the cost for when fuel is delivered.

3. Functional Requirements (William)

[Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.]

3.1. Primary

- FR0: The system will allow the user to pump a predetermined amount of gas.
- FR1: The system will allow the user to check how much gas they have pumped in total
- FR2: The system will allow a tanker truck diver to add gas to the storage tank

- FR3: The system will allow a tanker truck diver to estimate the cost of the added gas after adding
- FR4: The system will allow a tanker truck diver to check the total amount of gas that has been added in total
- FR5: The system will allow an admin to add or remove users from the system
- FR6: The system will allow an admin to enable or disable the pump
- FR7: The system will allow an admin to check the amount of gas in the storage tank

3.2. Secondary

- PIN access for authorized users only.
- API access to retrieve fuel price

4. Technical Requirements (William and Greyson)

4.1. Operating System and Compatibility

The application will be compatible with any operating system that is able to run a Java based program.

4.2. Interface Requirements

4.2.1. User Interfaces

Will be primarily text based but a GUI is a possibility.

4.2.2. Hardware Interfaces

A standard keyboard for the admin, although a nine digit keypad is all that a normal user or truck driver needs.

4.2.3. Communications Interfaces

A normal internet connection to access the API, but it is not strictly required as the program will run without one.

4.2.4. Software Interfaces

We will use Spring Boot for API access and possibly javaFX to help build the frontend, as well as a CSV file for the backend database functionality.

5. Non-Functional Requirements (William)

5.1. Performance Requirements

- NFR0(R): The local copy of the user CSV file will consume less than 5 MB of memory
- NFR1(R): The system (including the local copy of the user CSV file) will consume less than 50MB of memory

5.2. Security Requirements

- NFR4(R): The system will only be usable by authorized users.

5.3. Software Quality Attributes

5.3.1. Availability

24/7, as long as there is a power supply, it can function.

5.3.2. Correctness

Not doing any complex math or algorithms so not an issue.

5.3.3. Maintainability

None needed as software is intended to be used in an isolated environment. Any future upgrades would be introduced as a whole new stand-alone hardware unit.

5.3.4. Reusability
Very reusable, code is meant to be easy to upgrade and add functionality to.

5.3.5. Portability
Very portable, anything running Java can run the software.

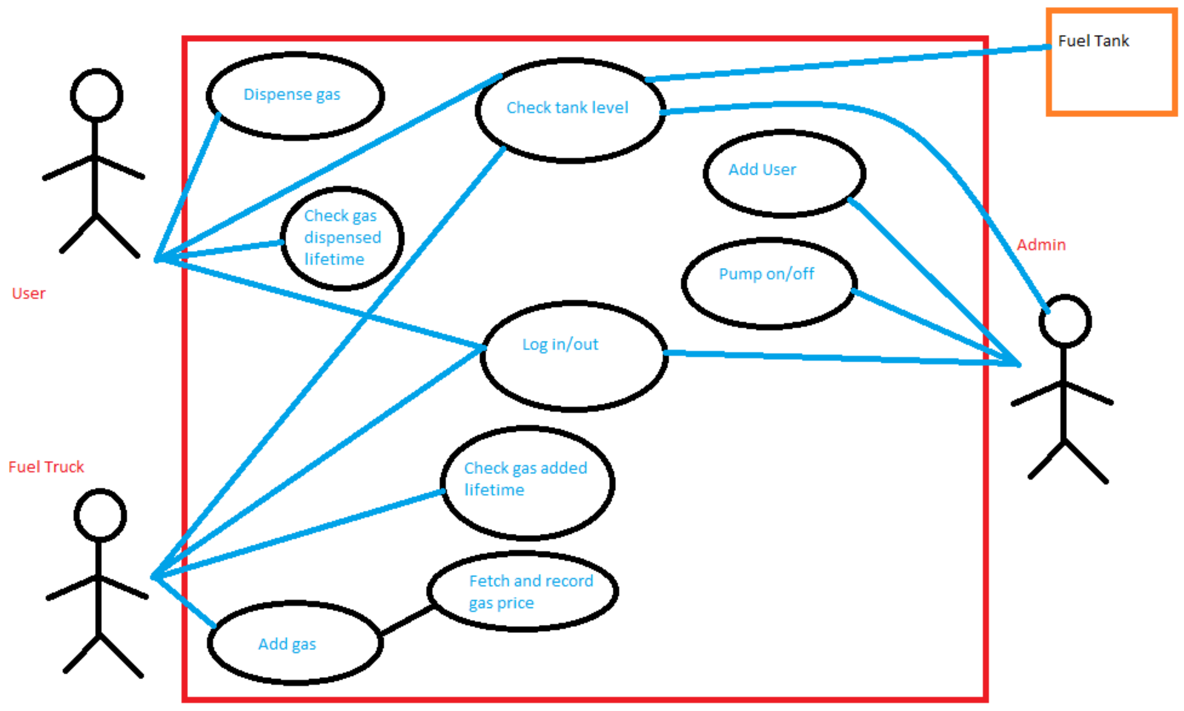
5.4. Process Requirements

5.4.1. Development Process Used
Agile

5.4.2. Time Constraints
Must be finished by April 18, 2023

5.4.3. Cost and Delivery Date
April 18, 2023

5.5. Use-Case Model Diagram (William)



5.6. Use-Case Model Descriptions (William)

5.6.1. Actor: Admin (William Vaughan)
- **Add User**: Add a user with a PIN, name, and login type
- **Pump Status**: Turn pump on and off
- **Check Tank Level**: Check tank level
-

5.6.2. Actor: User (Greyson Williams)
- **Pump Gas**: Dispense gas
- **Check Total Gas**: Check lifetime gas use
- **Check Tank Level**: Check tank level

5.6.3. Actor: Tanker Driver (???)
- **Add Gas**: Add gas to storage tank and estimate cost

5.7. Use-Case Model Scenarios (William)

5.7.1. Actor: Admin (William Vaughan)
- **Use-Case Name**: Login
  - **Initial Assumption**: User has entered PIN and is presented with the relevant menu.
  - **Normal**: User will enter a PIN and will be presented with a menu that matches their user role.
  - **What Can Go Wrong:** User's PIN is not accepted because csv is broken or not stored properly to array.
  - **Other Activities**: none
  - **System State on Completion**: User is logged in
- **Use-Case Name**: Add User
  - **Initial Assumption**: User has selected option and entered required information.
  - **Normal**: User enters a PIN with at least four digits, user then enters a name, and then chooses a user role.
  - **What Can Go Wrong**: Entered PIN has less than four digits, PIN is not a number, user role selection is not 1,2, or 3.
  - **Other Activities**: none
  - **System State on Completion**: Return to admin menu
- **Use-Case Name**: Pump Control
  - **Initial Assumption**: User has selected option and will turn pump on or off.
  - **Normal**: User is shown current pump status and can change to on/off.
  - **What Can Go Wrong**: User enters something other than "y" or "n".
  - **Other Activities**: none
  - **System State on Completion**: Return to admin menu
- **Use-Case Name**: Check tank level
  - **Initial Assumption**: User has logged into the system and is checking amount of gas in storage tank.
  - **Normal**: User selects option from menu and amount in gallons is displayed until enter is pressed.
  - **What Can Go Wrong**: nothing
  - **Other Activities**: None
  - **System State on Completion**: Return to user menu

5.7.2. Actor: User (Greyson and William)

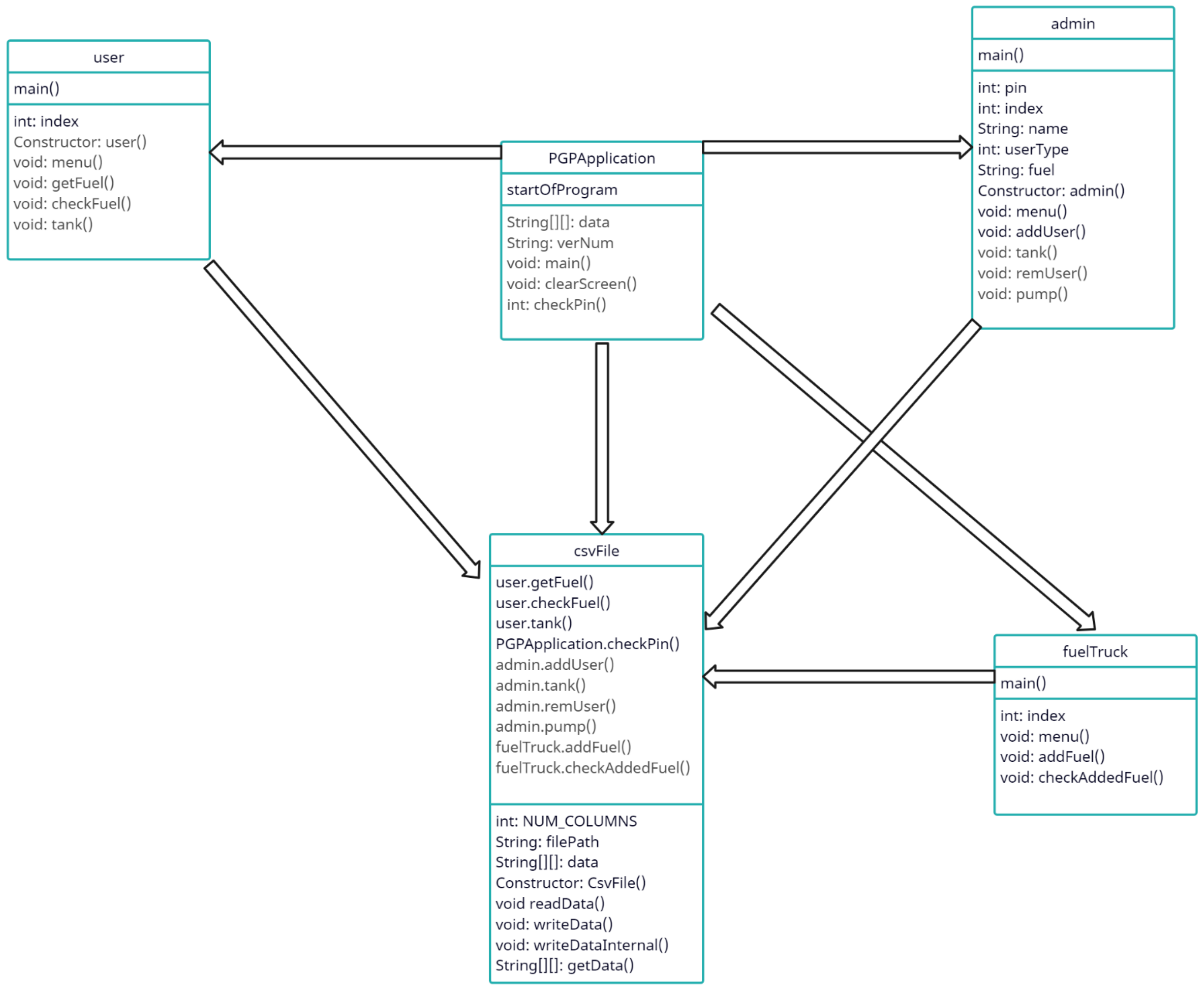
- **Use-Case Name**: Login
  - **Initial Assumption**: User has entered PIN and is presented with the relevant menu.
  - **Normal**: User will enter a PIN and will be presented with a menu that matches their user role.
  - **What Can Go Wrong:** User's PIN is not accepted because csv is broken or not stored properly to array.
  - **Other Activities**: none
  - **System State on Completion**: User is logged in
- **Use-Case Name**: Dispense Gas

- **Initial Assumption**: The user has requested gas.
    - **Normal**: User will enter an amount in gallons to be dispensed and amount will be logged to csv file. If pump is disabled, a warning will be displayed.
    - **What Can Go Wrong**: Gas amount is not in proper format (whole number or decimal)
    - **Other Activities**: none
    - **System State on Completion**: Log-out back to PIN menu
- **Use-Case Name**: Check tank level
    - **Initial Assumption**: User has logged into the system and is checking amount of gas in storage tank.
    - **Normal**: User selects option from menu and amount in gallons is displayed until enter is pressed.
    - **What Can Go Wrong**: nothing
    - **Other Activities**: None
    - **System State on Completion**: Return to user menu
- **Use-Case Name: Check total gas dispensed**
    - **Initial Assumption**: User has logged into the system and is checking amount of gas in dispensed over user's lifetime.
    - **Normal**: User selects option from menu and amount in gallons is displayed until enter is pressed.
    - **What Can Go Wrong**: nothing
    - **Other Activities**: none
    - **System State on Completion**: Return to user menu

- **Actor: Tanker Driver (William)**
- **Use-Case Name**: Login
    - **Initial Assumption**: User has entered PIN and is presented with the relevant menu.
    - **Normal**: User will enter a PIN and will be presented with a menu that matches their user role.
    - **What Can Go Wrong:** User's PIN is not accepted because csv is broken or not stored properly to array.
    - **Other Activities**: none
    - **System State on Completion**: User is logged in
- **Use-Case Name: Add Fuel**
    - **Initial Assumption**: User has chosen option and entered an amount of fuel to add to storage tank.
    - **Normal**: User will enter amount in gallons and will be presented with an estimated cost for delivered fuel.
    - **What Can Go Wrong**: User enters something that is not a number, user enters an amount greater than what can be stored.
    - **Other Activities**: none
    - **System State on Completion**: logout back to PIN menu

6. Design Documents

6.1. Software Architecture (Greyson)



6.2. High-Level Database Schema (William)

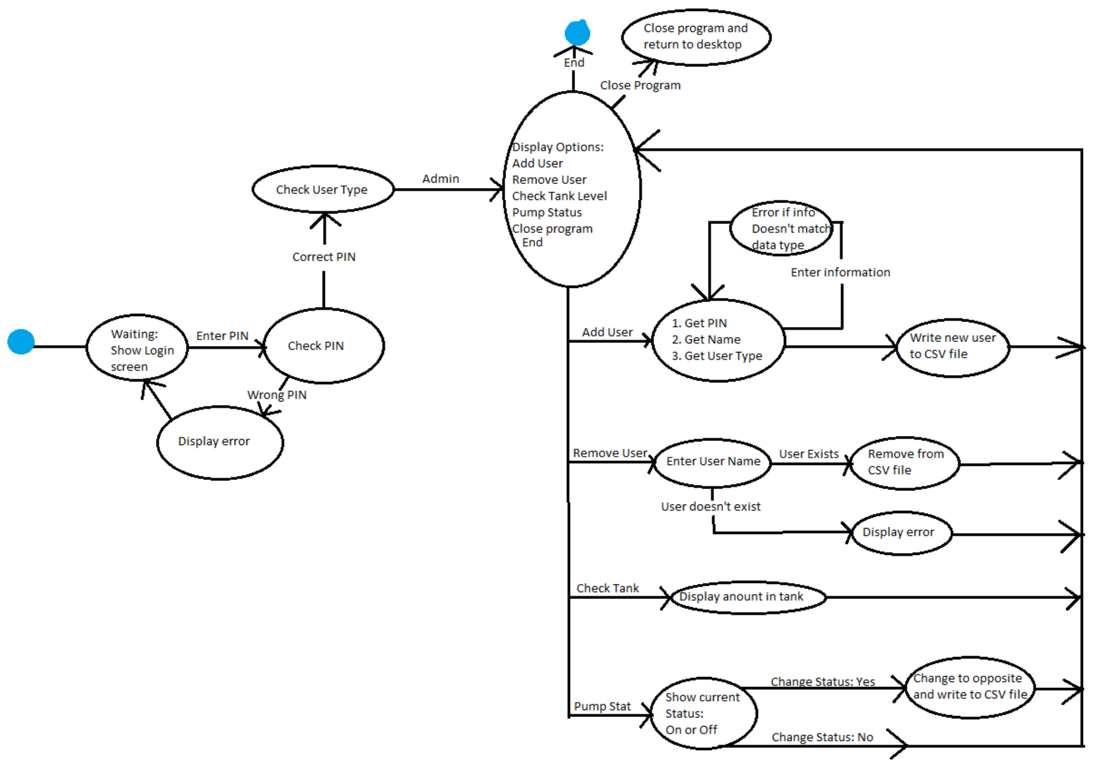
The data is stored in a CSV file for ease of editing. The first two entries are for the storage tank and pump status, all following entries are for users. The file itself is set up with four columns with a newline (\n) at the end to define a new row. An example follows:

PIN, Name, Usertype, Gas amount (either added or used)\n

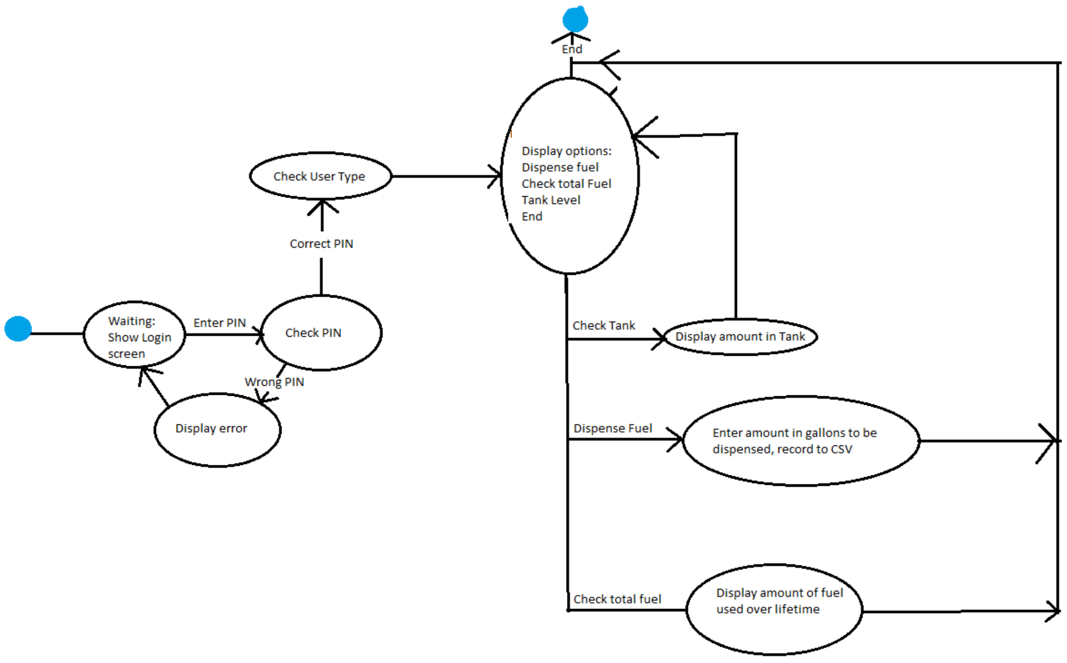
The CSV file is pulled into the program as a 2d array where it is edited and written straight back to the file.

6.3. Software Design (Greyson)

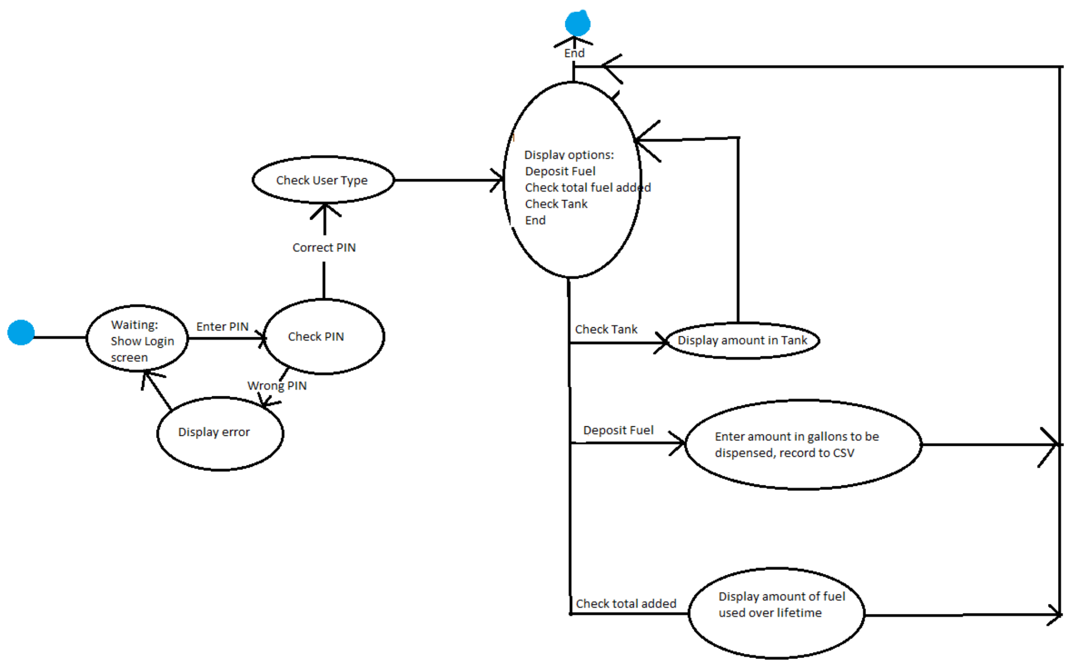
6.3.1. State Machine Diagram: Admin (William)



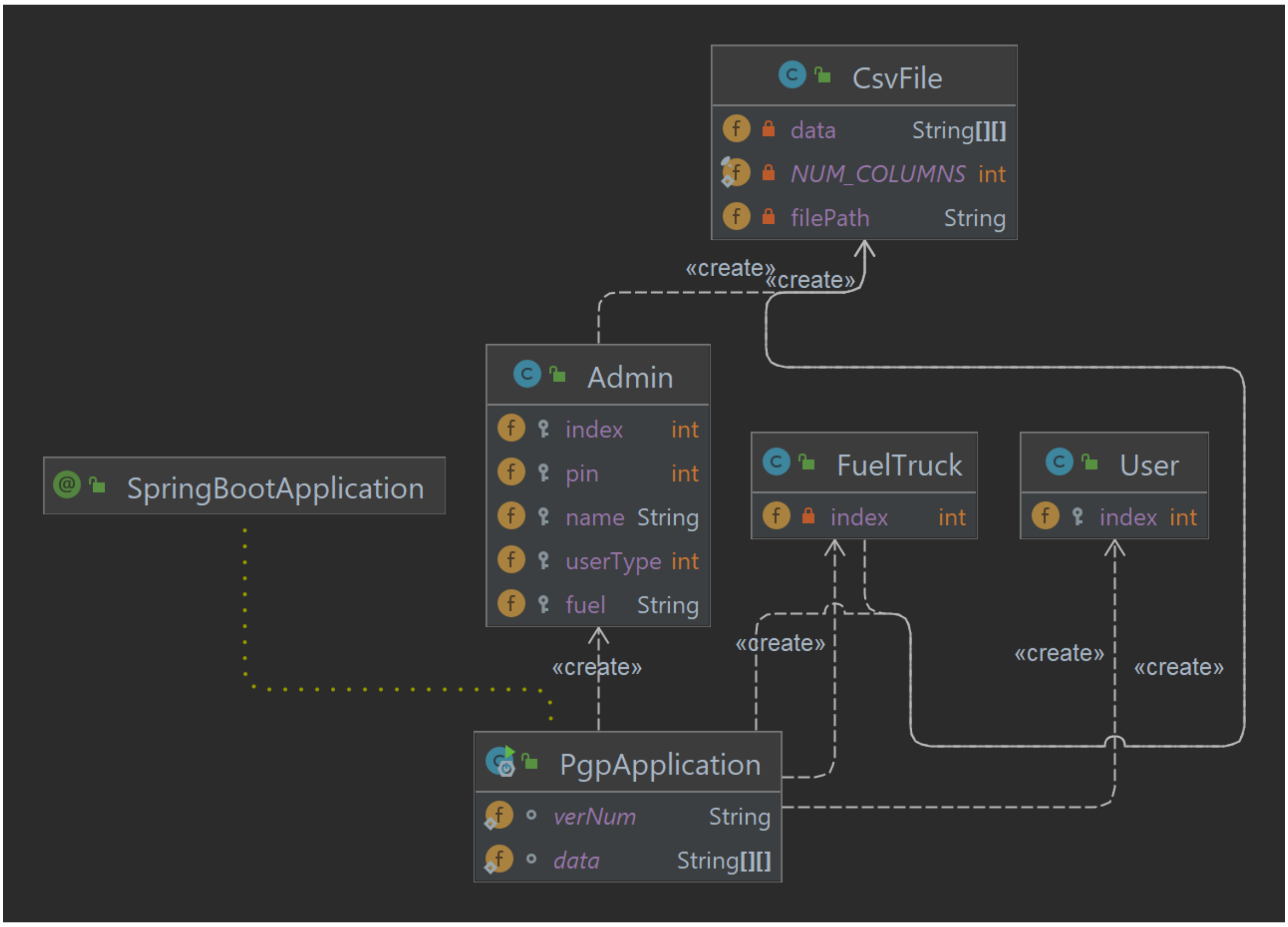
6.3.2. State Machine Diagram: User (Greyson and William)



6.3.3. State Machine Diagram: Actor Name (William)



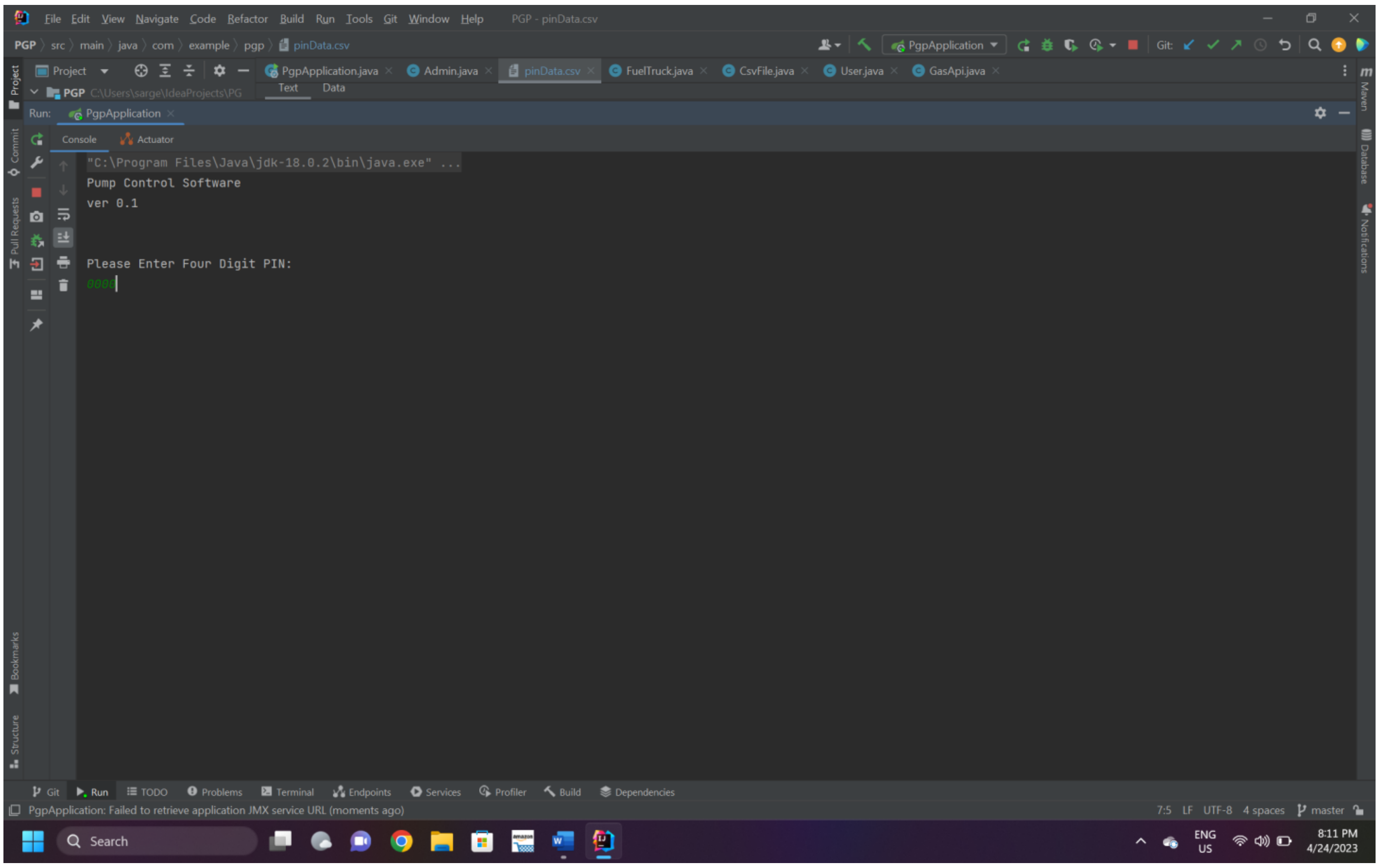
6.4. UML Class Diagram (William)



7. Scenario

7.1. Brief Written Scenario with Screenshots (William)

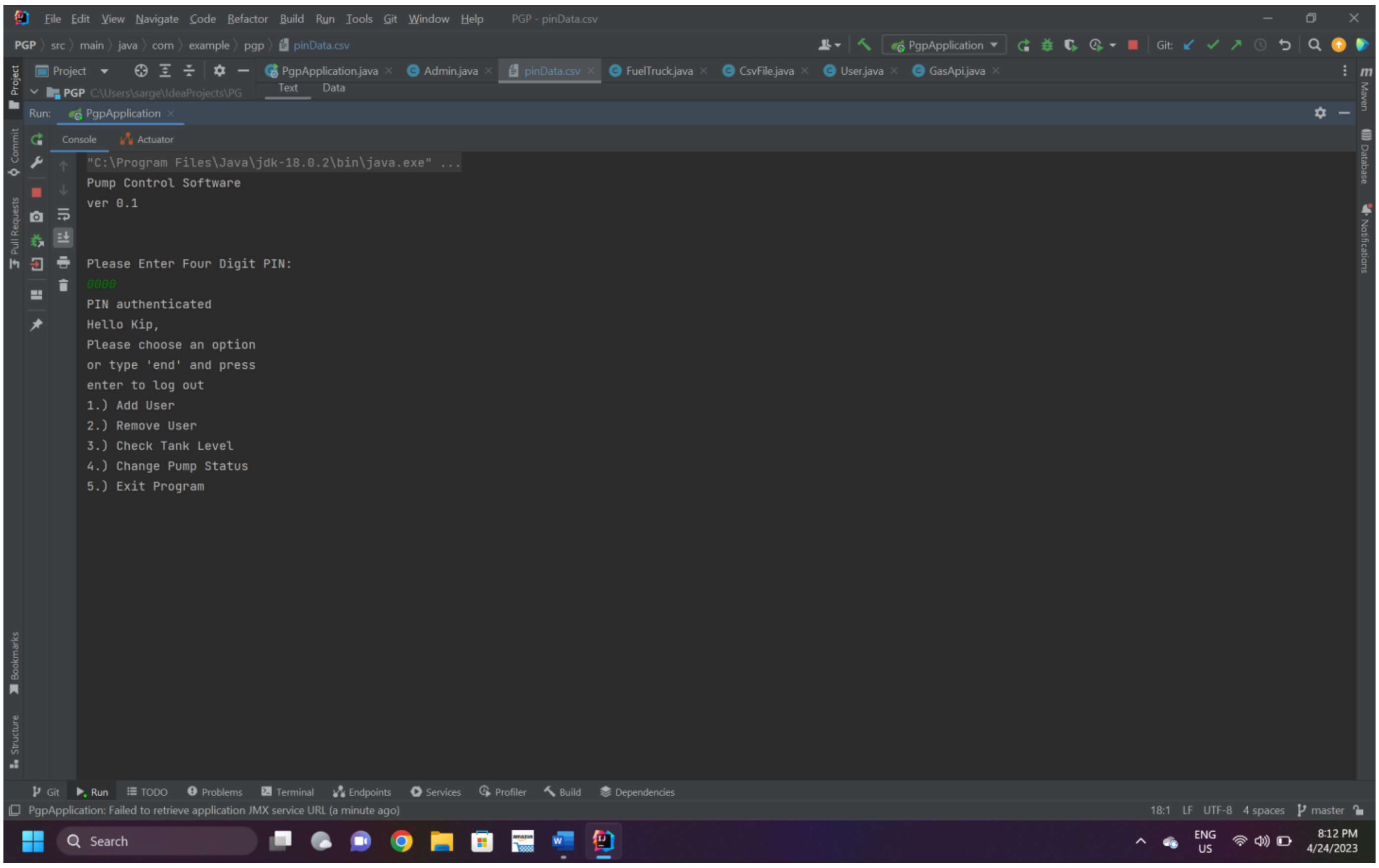
Admin is presented with the main login page



```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" ...
Pump Control Software
ver 0.1


Please Enter Four Digit PIN:
0000
```

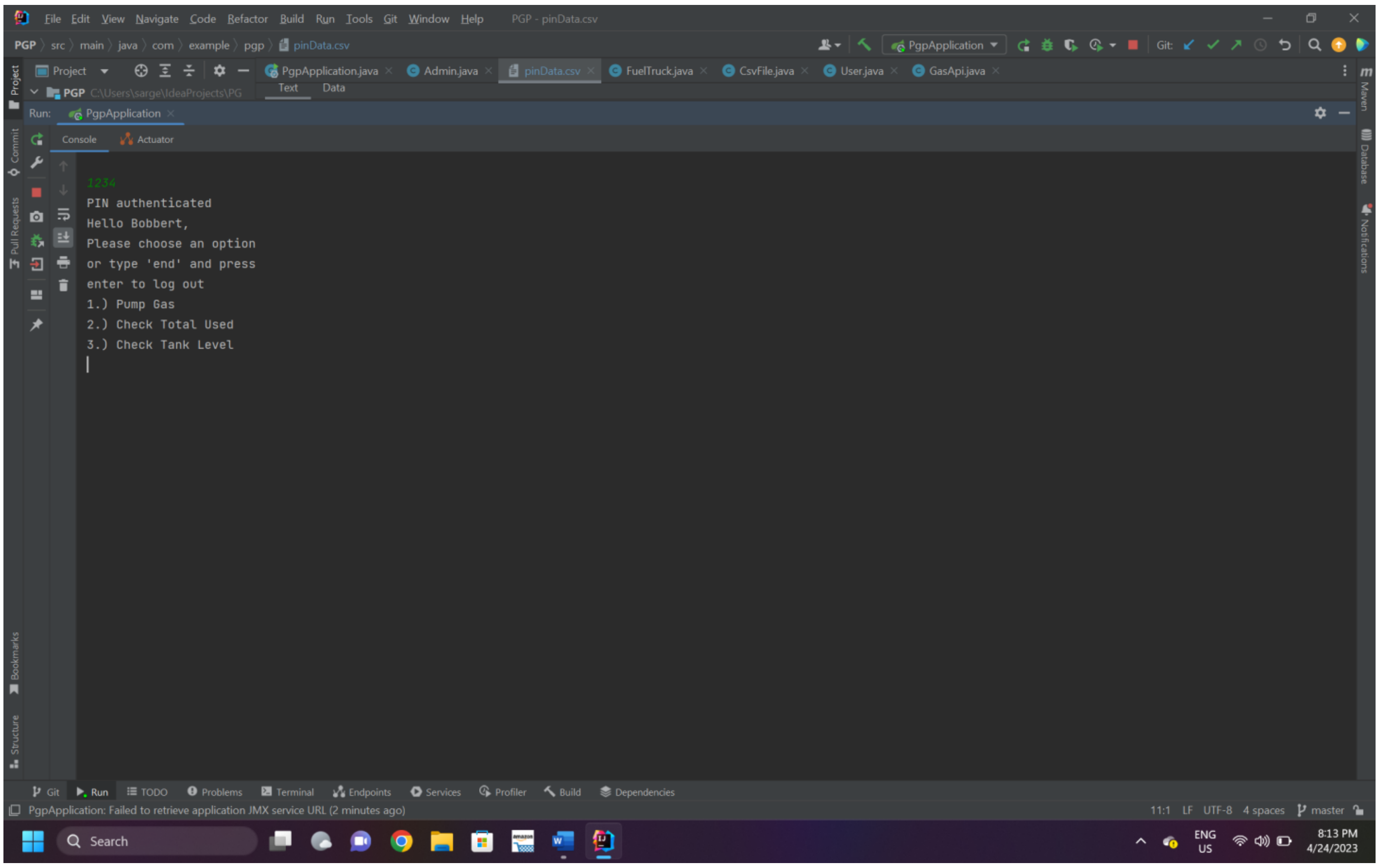
and logs in using PIN 0000, is presented with the admin menu,



```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" ...
Pump Control Software
ver 0.1


Please Enter Four Digit PIN:
0000
PIN authenticated
Hello Kip,
Please choose an option
or type 'end' and press
enter to log out
1.) Add User
2.) Remove User
3.) Check Tank Level
4.) Change Pump Status
5.) Exit Program
```

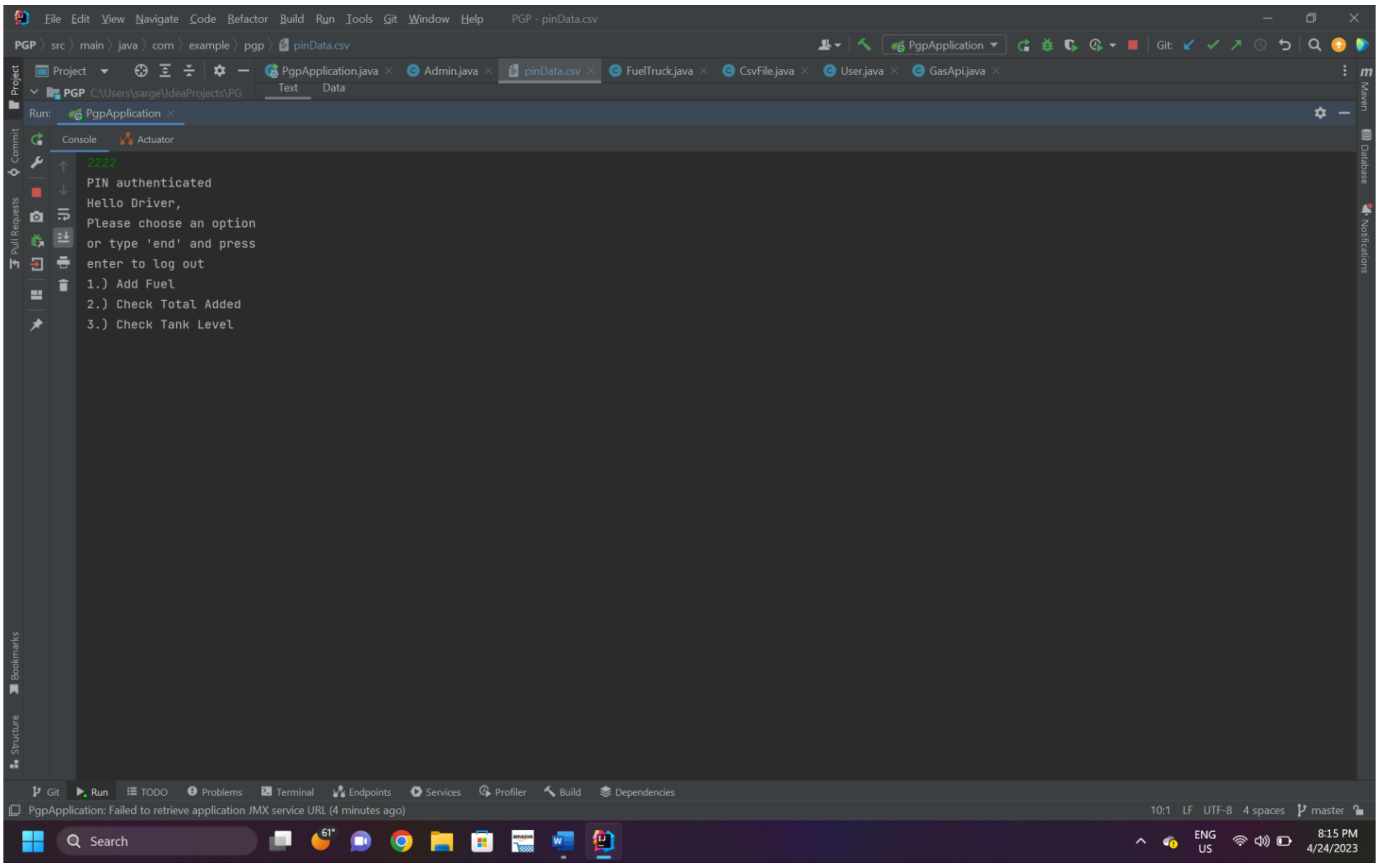
and creates a new user with the PIN 1234, name “Bobbert”, and user type 2 (normal user). Admin then logs out by entering “end” and pressing enter. Bobbert then logs in with his PIN and is presented with the normal user menu.



```
1234
PIN authenticated
Hello Bobbert,
Please choose an option
or type 'end' and press
enter to log out
1.) Pump Gas
2.) Check Total Used
3.) Check Tank Level
```

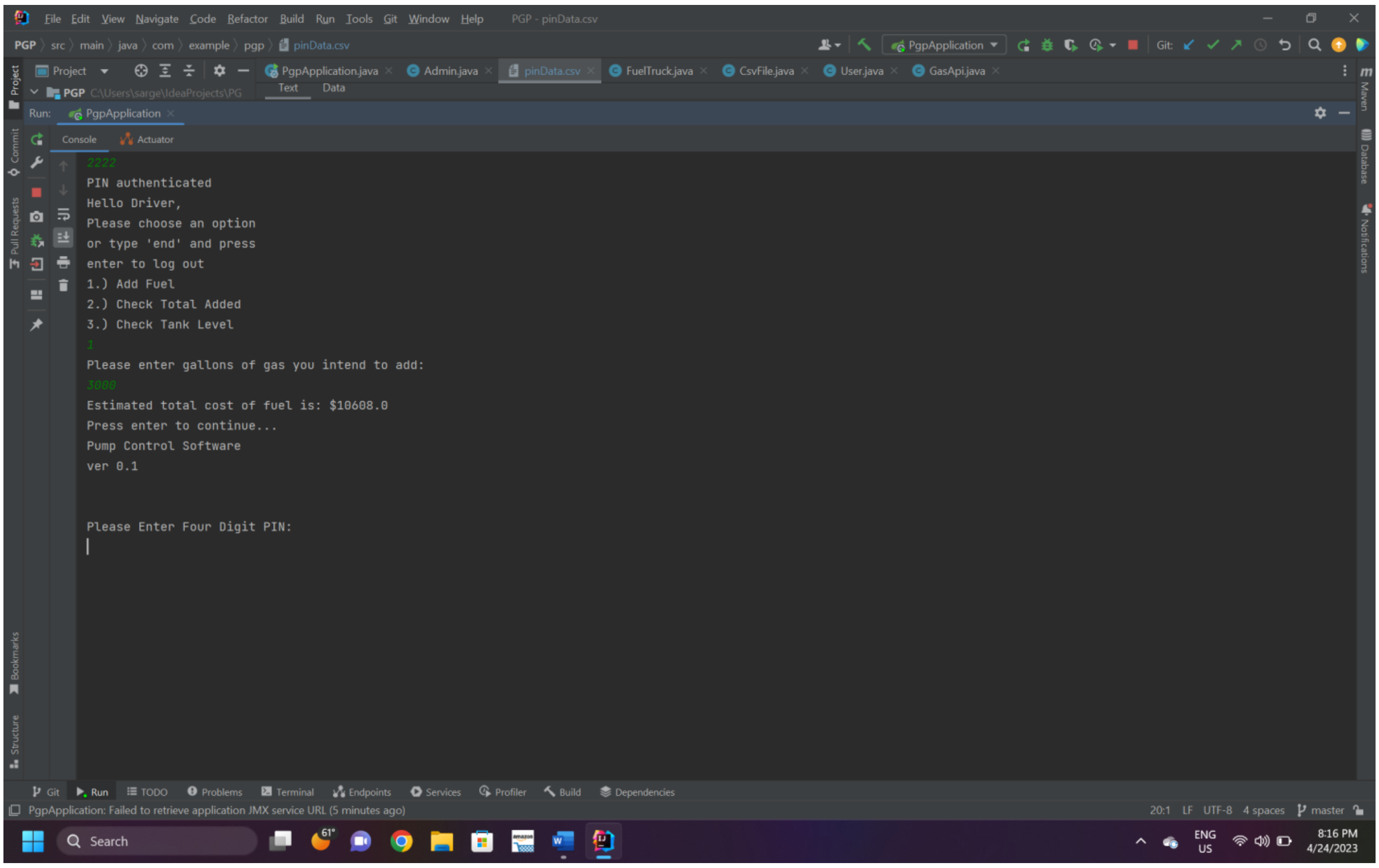
PgpApplication: Failed to retrieve application JMX service URL (2 minutes ago)

Bobbert then enters “2” and presses enter to access the fuel dispense option. He then enters the amount he needs and presses enter. Fuel is dispensed and removed from the tank. Bobbert is automatically returned to the login menu. Bobbert then informs the admin that the pump is not working properly. Admin then logs in and disables the pump through the admin menu until the pump can be fixed. After being serviced, admin then reenables the pump. After being informed by a user that the tank is running low, admin then contacts a fuel service for a delivery. When the driver shows up, they login using their PIN and are presented with the fuel delivery menu.



```
2222
PIN authenticated
Hello Driver,
Please choose an option
or type 'end' and press
enter to log out
1.) Add Fuel
2.) Check Total Added
3.) Check Tank Level
```

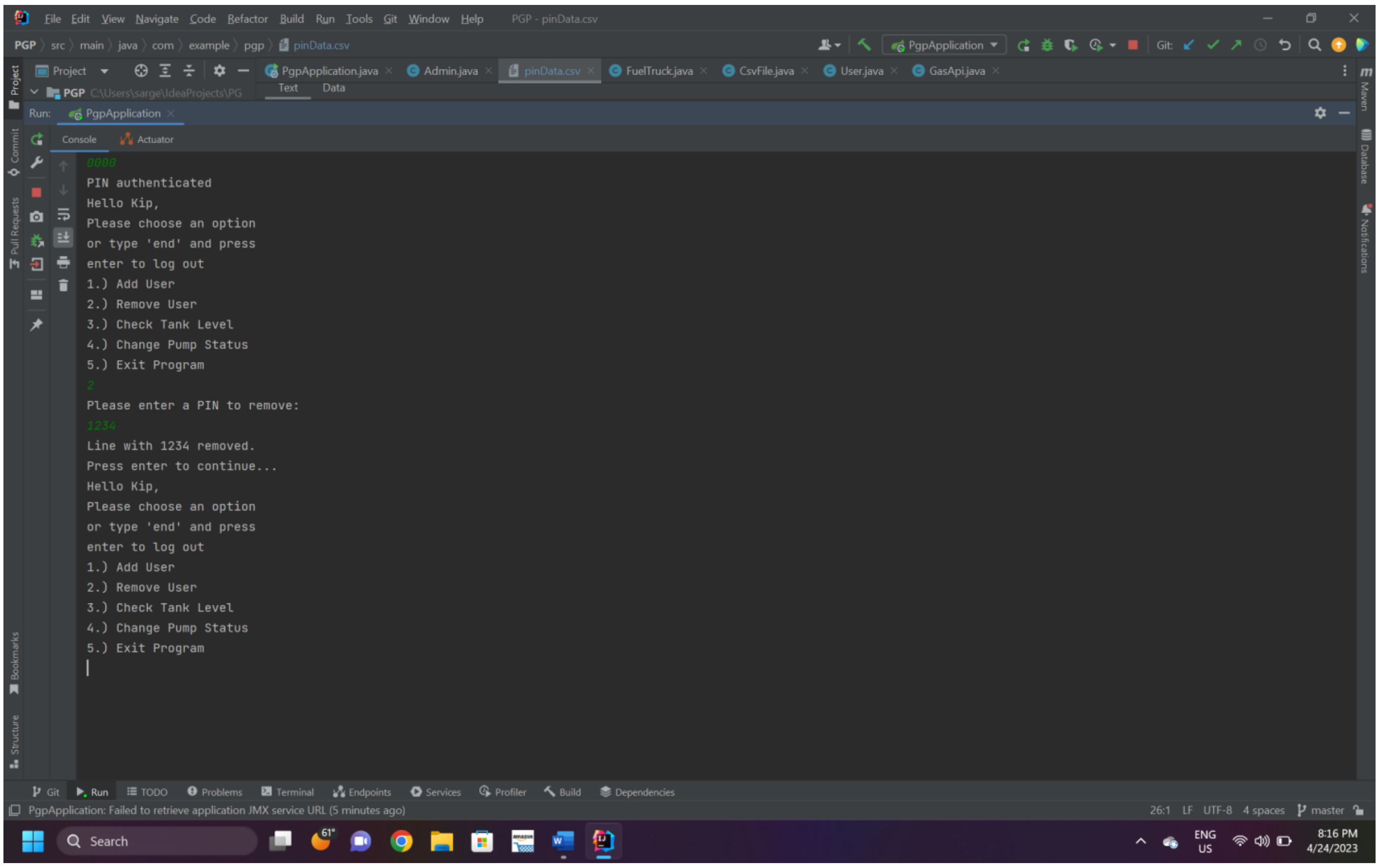
After checking the amount left in the tank and the total amount that they have delivered over time, the driver then deposits an amount of gas in the tank and is presented with an estimated cost as determined by a GasBuddy API.



```
2222
PIN authenticated
Hello Driver,
Please choose an option
or type 'end' and press
enter to log out
1.) Add Fuel
2.) Check Total Added
3.) Check Tank Level
1
Please enter gallons of gas you intend to add:
3000
Estimated total cost of fuel is: $10608.0
Press enter to continue...
Pump Control Software
ver 0.1


Please Enter Four Digit PIN:
```

Admin then logs in to the system to remove Bobbert from the database.



```
8080
PIN authenticated
Hello Kip,
Please choose an option
or type 'end' and press
enter to log out
1.) Add User
2.) Remove User
3.) Check Tank Level
4.) Change Pump Status
5.) Exit Program
2
Please enter a PIN to remove:
1234
Line with 1234 removed.
Press enter to continue...
Hello Kip,
Please choose an option
or type 'end' and press
enter to log out
1.) Add User
2.) Remove User
3.) Check Tank Level
4.) Change Pump Status
5.) Exit Program
```

PgpApplication: Failed to retrieve application JMX service URL (5 minutes ago)