



CS 360 Project One GCZ79 11/12/2025

I. Articulate the **goals** of the project

The goal of the project is to develop an application that helps users monitor their fuel consumption, spending, and vehicle efficiency over time. The app “Fuel Tracker” aims to combine financial tracking, vehicle performance monitoring, and data visualization in a single, easy-to-use tool. It will support users, including professional drivers, budget-conscious car owners, and car enthusiasts, by helping them track expenses, set budgets, use graphs and bars to identify patterns and improve fuel efficiency, and set alerts to help them stay on track with their finances.

The major components of the app will be:

- Database: Stores date/time, fuel pump brand, mileage, cost per gallon, amount spent, and fuel volume to support MPG calculations and detailed statistics.
- Login Screen: Ensures secure access and personalized data management.
- Data Entry Screen: Allows users to record fuel refills quickly and accurately.
- Statistics Screen: Visualizes key data like total spending, MPG trends, and brand preferences.
- Alert Feature: Notifies users when nearing fuel budget limits or efficiency thresholds.

The key functionalities implemented in the app will be:

- Automatically records the date and time of each refill.
- Provides clear, intuitive data entry and visualization for ease of use.
- Tracks and analyzes spending and fuel efficiency over customizable time periods.
- Offers optional alerts based on usage patterns and user preferences.

II. Describe the **users** of the application and the assumptions being made regarding their needs and preferences

To better identify the needs my app will address, I developed three user personas, with their goals and pain points, who represent my idea of typical users.

Jason Miller



Age: 23
Height: 5' 9"
Weight: 155 lbs
Status: Single
Education: HSD
Job: Lyft Driver
Country: USA

Goals

- Track every refill accurately for tax reporting
- Reduce time spent on manual bookkeeping
- Gain clear insights of his actual profit after gas and car maintenance costs

Pain Points

- Losing paper receipts or forgetting to log them
- Complex fuel-tracking apps that require a degree in computer science to be used
- Understanding his net income after expenses

Quote

"If I can track every ride I give, I should be able to track every dollar I spend on gas just as easily."

Bio

Jason is a full-time rideshare driver who spends most of his day on the road. He prides himself on being efficient and organized, but keeping track of gas receipts and mileage manually has become a hassle.

Motivations

- Maximize take-home earnings
- Seeking simplicity: fewer steps, faster results
- Improve MPG performance

Behaviors

- Log fuel refills
- Uses multiple apps daily
- Plan his expenses ahead of time

Personality

Practical, independent, detail-oriented.
Prefers to manage his work and finances without relying on others.

Favorite Brands



Sarah Smith



Age: 28
Height: 5' 6"
Weight: 130 lbs
Status: Engaged
Education: BA
Job: Marketing Assistant
Country: USA

Quote "I budget everything else: why should gas be the one thing that always surprises me?"

Bio Sarah is a young professional living in the city, balancing a modest salary with student loans, rent, and daily commuting costs. She drives her compact car to work every day and occasionally for weekend trips.

Personality Organized, curious, goal-oriented.
Wants to understand the "why" behind her spending.

Goals

- Stick to a monthly fuel budget to save more for personal goals
- Simplify personal finance management with one integrated app

Pain Points

- Difficult to visualize where her gas money goes over time
- Tired of manually calculating monthly fuel totals
- Frustrated by generic budgeting apps

Motivations

- Wants to feel in control of her finances
- Enjoys the satisfaction of staying under budget
- Values convenience

Behaviors

- Checks her budgeting app at least once a day
- Logs purchases in real-time or on weekends

Favorite Brands



Vincent Toretto



Age: 43
Height: 6' 1"
Weight: 195 lbs
Status: Unavailable
Education: HSD
Job: Mechanic
Country: USA

Quote "I live my life a gallon at a time. Premium only."

Bio Vincent has been passionate about cars since his teenage years, when he rebuilt his first engine with his father. Today, he works as a mechanic and spends his weekends fine-tuning his two cars, a tuned Subaru WRX and a restored '90s Mazda Miata.

Personality Analytical, curious, detail-oriented.
Comfortable with data visualization tools.

Goals

- Optimize vehicle performance after upgrades
- Analyze fuel efficiency under different driving conditions
- Log performance data for online car community

Pain Points

- Data is often difficult to export or correlate with vehicle changes
- Generic apps don't allow for the level of personalization he needs

Motivations

- View long-term efficiency graphs before and after modifications
- Track brands, costs, and MPG with precision.

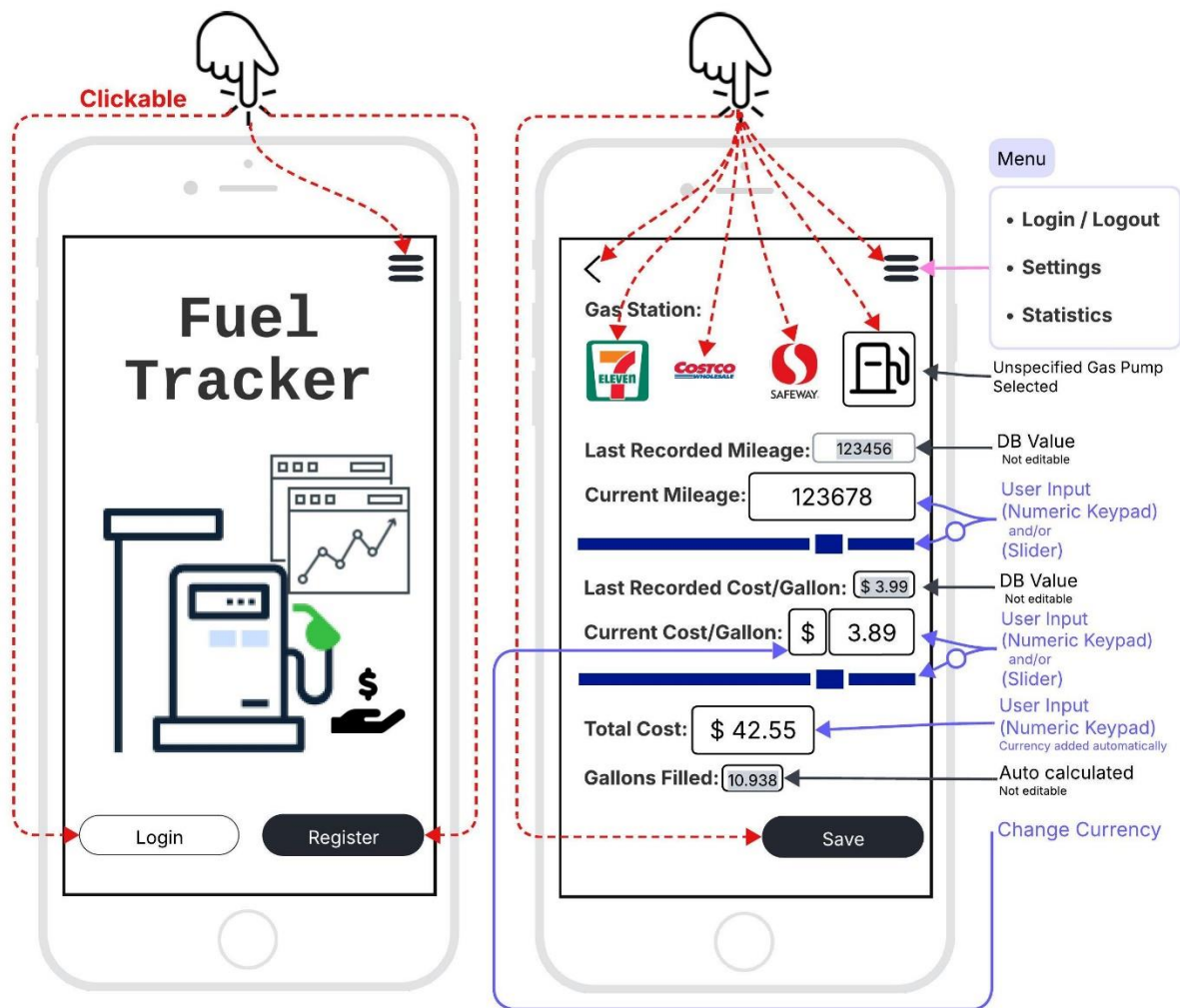
Behaviors

- Log fuel refills
- Compare MPG and cost trends monthly
- Watches automotive YouTube channels

Favorite Brands



III. Discuss the screens and features that will be necessary to produce a user-centered **UI design** for the app.



The **Landing/Login page** will have just a few elements: the name and logo of the app, the menu button on the top right corner, and, at the bottom of the screen, two buttons to register if new users, or to log in if already registered. When users register and use the app for the first time, they will be redirected to the **Settings page** to save their data and preferences in matter of:

- favorite gas station/brand #1, 2, and 3
- favorite currency
- current mileage of the vehicle
- preferences for alarms

Once authenticated and with the personal settings saved, users will be directed to insert the data related to a gas station visit in the **Fuel Refill page**. In this page, there will be the menu button in the top right corner, present in all the screens of the app, and a button to return to the previous page in the top left corner. The users will:

- pick with a click the gas station/brand they are using
- insert the current reading of their mileage {numeric keypad or slider}
- (optional) check the last recorded cost per gallon for this particular brand
- (optional) select a different currency
- insert the current cost per gallon {numeric keypad or slider}
- insert the total cost {numeric keypad}
- (optional) check the amount of gallons refilled
- click on Save
- confirm to Save the data or Go Back to edit

Finally, in the **Statistics page**, users will be able to visualize the data of their fueling sessions.

I will include many filters to manipulate the data and show useful insights. For example, I want to use the data to define patterns like weekly or monthly spending, which day of the week and time of the day users are more likely to refill their gas, or how the MPG changes over time.

The data does not have to be overwhelming, but it must be easily accessible and intuitive to understand. This section will be crucial for the hypothetical success of the app on the market, so I will experiment with different solutions until I find the optimal one.

The design choices for the Fuel Tracker app follow the Android Design and Quality Guidelines, with particular attention to clarity and consistency in the design, and an intuitive user interaction. The **Landing/Login page** will contain minimal elements to reduce visual clutter and focus only on registration and login. Positioning the action buttons under the app name and logo follows Android's "Interaction ergonomics" recommendation to keep primary actions within comfortable reach (Google, n.d.). The persistent menu button in the top-right corner provides consistent navigation across all pages, supporting consistency and predictability as advised by the guidelines.

Directing first-time users to the **Settings page** after registration follows Android's onboarding best practices, which recommend allowing users to personalize their settings and experience as soon as it makes sense.

The **Fuel Refill page** prioritizes clarity and efficiency. Interactive elements like sliders and numeric keypads support quick data entry with minimal typing, which is very important while being busy refilling our gas tank. Seeing the last recorded mileage helps with needing to remember mainly just the last 3 digits of the current reading, while seeing the last price of fuel at the current gas pump helps to recognize pricing trends.

Finally, the **Statistics page** will clearly show users the registered data, and help them with graphs to understand at a glance trends such as spending habits and fuel efficiency. Following the directions of Material Design on visual hierarchy, data will be organized using colors and layouts to draw attention to key insights, while avoiding information overload as outlined in the "Essential interactions" (Google, n.d.).

IV. Discuss how the functional app requirements will be represented in the **code design** and connected to the UI.

The Fuel Tracker app will follow a Model-View-Controller (MVC) approach to separate data storage, the user interface, and application logic. Android Studio is well-suited for MVC development because it facilitates organizing components, supporting clear separation and modularization throughout the project.

1. Model: It represents the app's data, including fuel records, mileage, fuel brand preferences, and user settings. This data will be stored in a local SQLite with Room database, and accessed via data classes in the code.
2. View: The UI screens (Landing/Login, Settings, Fuel Refill, and Statistics) display data from the model, and allow users to input new values.
3. Controller: It handles the communication between the UI and the model. For instance, when the user enters a new refill:
 - Values from the DB are shown (last mileage and last cost/gallon)
 - Values from the DB affect the ranges of the sliders
 - The Fuel Refill page collects new input values
 - The controller validates the data and updates the DB
 - Observers on the Statistics page react to DB changes to update graphs

FuelRefillFragment → FuelRefillViewModel.onSaveClicked() → RefillRepository.insertRefill()

RefillRepository → Room Database DAO → SQLite insert

Database update → LiveData<List<RefillRecord>> notifies all observers

StatisticsFragment observing the LiveData automatically updates charts

The major UI components in the **Landing/Login page** are the app logo and name, the menu button in the top-right corner (present in every page), and the Login/Register buttons.

In the **Settings page**, the interface allows users to save up to 3 favorite gas stations/brands, to pick the favorite currency, to input an initial mileage, select alarm preferences, and a Save button.

In the **Fuel Refill page**, the various elements are listed logically. Following the top → down direction of the page, the users will first select the brand they are using, which will affect the positioning of the slider for selecting the cost of the fuel per gallon by picking the last saved value as the default/middle of the scale. This slider will allow users to make an accurate selection, since fuel is generally priced at \$ X.Y99, and it goes up or down gradually in the short term.

Users visualize the previous mileage before entering the current reading. Having the old reading on display will allow them to get used to memorizing just the last 3 digits of the current mileage.

Users have the option to enter the current mileage with a keypad or through a slider. The slider for the mileage might be less precise/useful than the previous one for the gas cost, but, starting from the last mileage recorded as the leftmost value, and finishing to the right with a value equal to the last mileage recorded + 500 miles, it will allow for selecting a number close to the actual one. Users might need to correct the last one or two digits, but the optional use of the slider should make the entire operation faster than entering all 5 or 6 digits (or whatever the reading is).

After entering the current cost of the fuel in {currency} / gallon, users will input with the numeric keypad the total amount spent, and the interface will calculate how many gallons of fuel were refilled ($\{\text{total cost}\} / \{\text{cost/gallon}\}$). As a last step, they will click the Save button, and the interface will ask for confirmation to Save the current data, or Go Back and edit it.

In the **Statistics page**, there will be graphs and charts to visualize the data. Filters will be crucial to show what the users want to visualize, and I predict to allow users to visualize values like:

- (Total / Average) (Spent / MPG) divided by Month/Year and displaying trends
- Day and Time of the week when users are more likely to refill gas
- Favorite Brand: (Total / Average) Spent, Number of Visits, (Highest / Lowest / Average) Cost/Gallon price in the last Month/Year
- Same statistics for Brand #2, Brand #3, and “Other/Unlisted”, if any
- Comparisons among brands
- Any other relevant statistical data

References

Google. (n.d.). *Layout basics: Mobile - Layout & content*.

<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-basics>