Trivia Parking Space

- Rahul Kalapala
- Maheswara Sai Ram Palakurthy
- Saketh Sai Mallepaddi
- Suhasini Polampelly

About Application

- Application Overview: Parking Space Management for Students and Faculty
- Different Service Levels: Green, Gold, and Orange Lines
- Key Features: Parking, Car Wash, Vehicle Inspection
- User Categories: Students, Faculty, and Guest Parking

SERVICES

- Green Line: Basic Parking Service
- Gold Line: Parking + Car Wash
- Orange Line: Parking + Vehicle Inspection

Registered Users:

- Students and Faculty
- Register with their details and Unique ID
- One-Time load money to their Wallet Balance

Guest Parking Users:

- Guest Parking Option
- Car Number Registration for Guest Parking
- Simplified Process for Visitors

Operating Instructions

- Run the program
- You'll be prompted with 2 options 1.Sign up 2.Proceed to Parking Lot
- If selected 1.Sign Up:
    - Enter Unique UserID,
    - Select User Type (1.Student 2.Faculty)
    - Enter the Wallet Balance ($) to be stored:
    - Select one of the ParkingLine below: 1. Green 2. Gold 3. Orange
    - Enter the number of vehicles to be registered:
    - Enter #X Vehicle Registered Number:
    - Enter #X Vehicle Name
    - You will be prompted with
    - User data saved successfully.
    - User with Id: XXX registered successfully!

Operating Instructions

- If selected 2 Proceed to Parking Lot:
  - Pick anyone of the options below: 1. Park Vehicle 2. Exit Vehicle from Parking Space
- Pick anyone of the options below: 1. Park Vehicle 2. Exit Vehicle from Parking Space
- If selected 1 Park Vehicle:

  Parking Lot [lotName=GREEN, totalSpaces=150]
  - Parking Lot [lotName=ORANGE, totalSpaces=40]
  - Parking Lot [lotName=GOLD, totalSpaces=100]
  - Select an option from below: 1. Park with UserId 2. Proceed without UserId
- If selected 1 Park with UserId:
  - Enter the Id - The unique id you registered with
  - Enter Vehicle Number - The vehicle number of the vehicle you are intending to park
  - Enter about to Park Line: 1. GOLD, 2. GREEN, 3. ORANGE - If you select any other line than you selected during registration than you will be charged a premium
  - Vehicle XXX successfully parked in parkingLine XXXXXXX

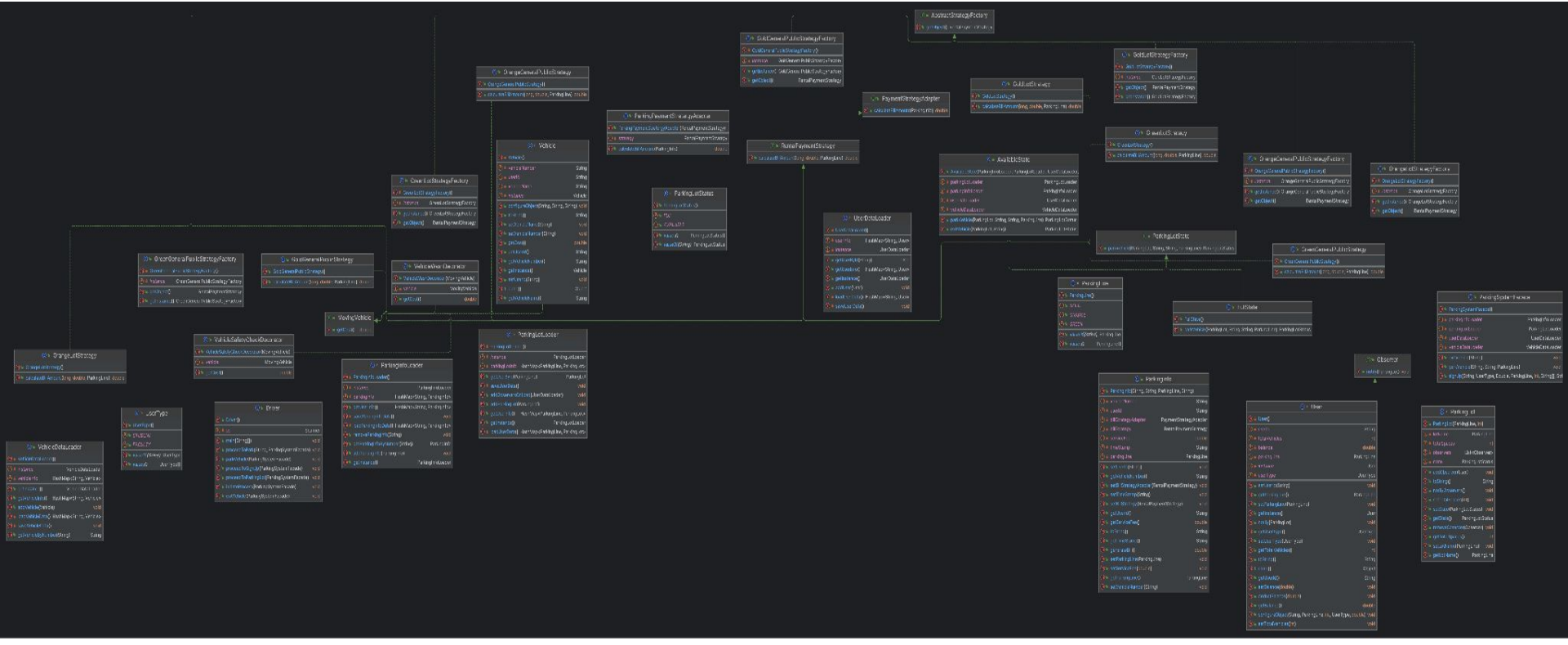
Operating Instructions

- If selected 2 Proceed without UserId
  - Enter Vehicle Number
  - Enter about to Park Line: 1. GOLD, 2. GREEN, 3. ORANGE
  - Vehicle XXX successfully parked in parkingLine XXXXXXX
- If selected 2 Exit Vehicle from Parking Space
  - Enter the Registered Vehicle Number
  - Bill Amount deducted: $XXXXXX.XXX
  - Vehicle XXX successfully exited!

Design Patterns Used

- **Singleton** - Used to ensure that only one instance of the class is present throughout the application lifecycle.
- **Factory** - Used it to provide consistent way of creating objects for different billing strategies.
- **Strategy** - Used it define different billing strategies which encapsulate the logic for calculating the bill.
- **Adapter** - Used to adapt different payment strategies into a common interface
- **Prototype** - Used to clone the user and vehicle base model
- **Decorator** - Used to enhance the services provided by adding additional features like safety check and vehicle washing.
- **State** - Used to represent various states of parking lot, like available and full
- **Observer** - Used to establish relationship between the parking lot and user classes.
- **Facade** - Used to abstract the complexity of interacting with various loaders, observers, and other components. It centralizes the operations related to user sign-up, parking, and exiting, making it easier to manage the underlying complexity.

UML Diagram



Future Enhancements

- Unique userId can be generated from the platform.
- Remaining wallet balance for each user can be displayed after the vehicle exited.
- Option to recharge the user's wallet.
- We can also expand the application to charge the customers based on the hours they parked.

Contributions

- Rahul Kalapala - Identified the problem statement and came up with the strategy to solve the problem statement. Also contributed to developing the code by helping out in pointing out various design patterns.
- Maheswara Sai Ram Palakurthy - Contributed to the basic structure of the code and implemented design patterns and methods to solve the problem statement.
- Saketh Sai Mallepaddi - Contributed to the user flow and to implement design patterns. Also helped in creating the presentation.
- Suhasini Polampelly - Contributed to identifying the possible scenarios and helped in implementing few design patterns and testing the code. Also helped in creating the presentation.

THANK YOU