# TERM PROJECT

**Programming Java SE**  
**Class: 2023F\_CSD\_3464\_4**  
**Project Name: JustGo**

SHARATH MOHAN  : C0894212

CHANDRALEKHA POLEPALLE  : C0887776

REVATHI VANNAMPALLI SREENIVASA REDDY  : C0895334

HARSHAVARDHAN BABU GONDIPALLI  : C0893432

**Description:**

The "Just Go" ride-sharing app is a user-centric platform with a focus on simplicity and efficiency. It aims to streamline the ride-finding process for users and drivers through an intuitive way. Key objectives include optimizing route planning to reduce travel time, ensuring affordability for riders, and maintaining a fair compensation structure for drivers.

Safety is a top priority, with real-time tracking and emergency assistance features to create a secure environment.

By combining these elements, "Just Go" aspires to redefine the commuting experience, offering a seamless and cost-effective solution while fostering safety, sustainability, and fairness within its community.

**Why it's Interesting:**

The "Just Go" ride-sharing app stands out in the market due to its holistic approach to addressing various aspects of transportation. By combining convenience, efficiency, safety, and sustainability, the app aims to provide a comprehensive solution for modern commuting challenges. The emphasis on innovation ensures that the platform remains relevant and innovative, offering users and drivers a dynamic and evolving experience. Ultimately, Just Go seeks to redefine the ride-sharing landscape by not only meeting basic transportation needs but also by contributing to a more connected, efficient, and sustainable urban environment.

**Problem Statement**

JustGo is a ride-sharing application that allows users to act as participants in the application either as a rider requesting a ride or a driver who offers trips to riders in need. The application was built keeping scalability and dynamic programming design as the key priority.

With the help of OOPS and DRY principles, we designed a solution to distinguish entities in the system design and their associated properties and behaviors into separate modules.

The entities are:

1. Riders
2. Drivers
3. Trips

**System Design & Implementation**

1. **OOPS (Object-Oriented Programming):**

OOPS is a programming paradigm based on the concept of objects. It involves the use of classes and objects, encapsulation, inheritance, and polymorphism.

1. Encapsulation: Bundling data and methods that operate on the data into a single unit (class).
2. Inheritance: A mechanism to create a new class using properties and behaviors of an existing class.
3. Polymorphism: The ability of a class to take on multiple forms, achieved through method overloading and overriding.
4. **Singleton Design Pattern:**

Ensures that a class has only one instance and provides a global point to access it. Typically involves a private constructor, a static method to get the instance, and lazy instantiation. Commonly used for logging, driver objects, caching, thread pools, or database connections.

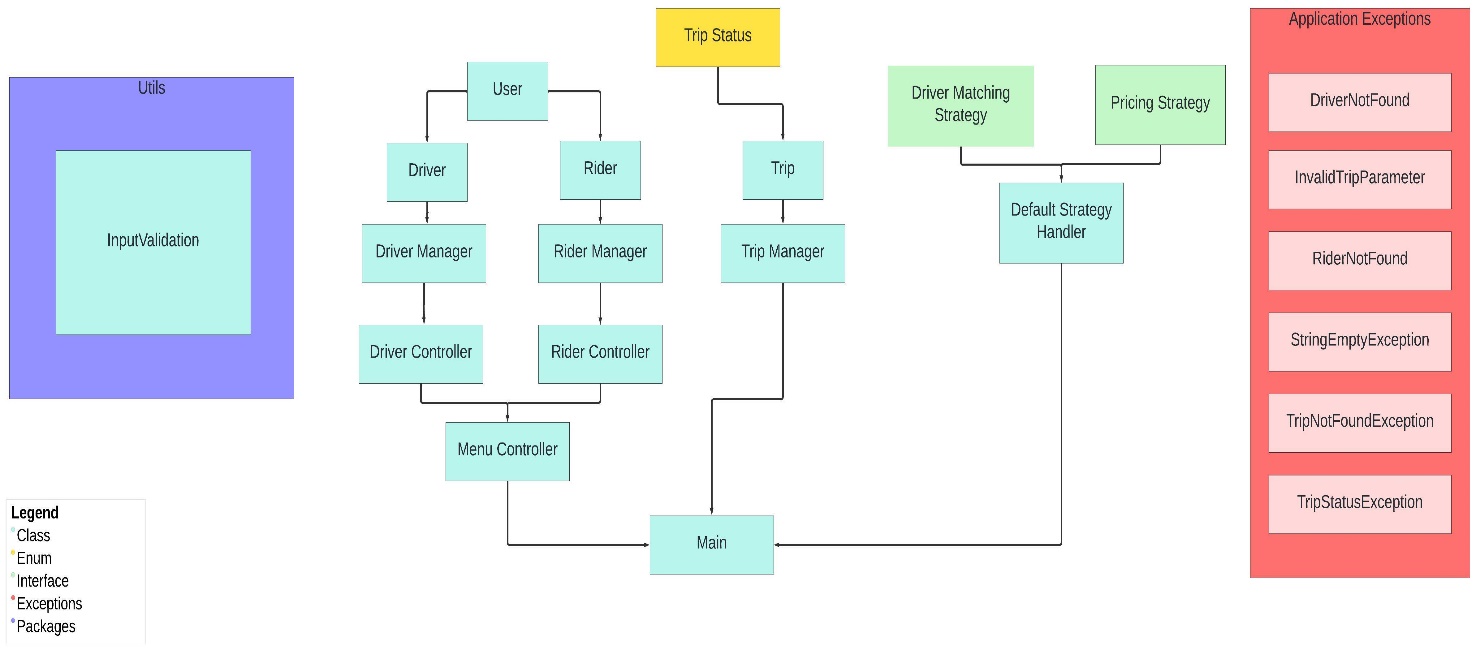
1. **Interfaces:**

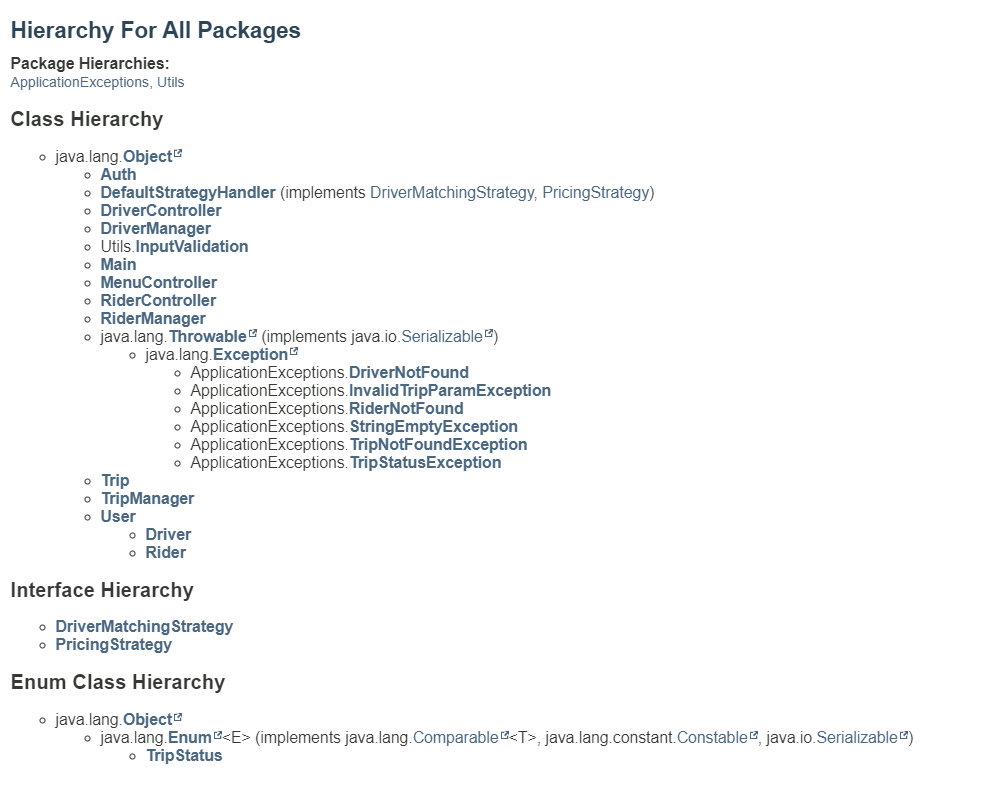
A contract specifying a set of methods that a class must implement. Provides a way to achieve abstraction in Java. Multiple inheritance is supported through interfaces.

1. **Custom Exception Handling:**

* Creating user-defined exceptions to handle specific scenarios in a program.
* Extends the Exception class or its subclasses.
* Helps in making code more readable and handling application-specific errors gracefully.

JustGo was designed based on the principles stated above, it leverages Classes, Enums, Interfaces, Exceptions, and packages for building the application. The system design diagram is as follows:





**Key Features**

* + Register Driver
  + Register Rider
  + Driver & Rider Authentication
  + Rider Features
    - Create Trip
    - Update Trip
    - Cancel Trip
    - Get Trip Details
    - Rate driver and generate receipt
  + Driver Features
    - View Current trip
    - Start Trip
    - Complete Trip
    - Cancel Trip
    - Switch Availability
* Results & System Evaluation
  + Trello: https://trello.com/b/VvheMp0t/justgo-java-project
  + Github: https://github.com/SharathRajMohan/JustGo
  + Output Documentation: https://github.com/SharathRajMohan/JustGo/blob/main/docs/Output.pdf

**Conclusion:**

Throughout the development of the "Just Go" ride-sharing app, our team gained invaluable insights into the practical application of Object-Oriented Programming (OOPS) principles, including encapsulation, inheritance, and polymorphism. Leveraging the Singleton Design Pattern and implementing custom exception handling further solidified our understanding of robust system design. The use of Java interfaces and careful consideration of scalability enhanced the application's adaptability.

Technologies such as Trello and GitHub facilitated transparent project management and collaborative development. "Just Go" exemplifies the successful integration of technology and user-centric design, emphasizing safety, efficiency, and affordability. The app's usability is underscored by its offering features such as driver and rider registration, authentication processes, and real-time tracking. By prioritizing innovation and continuous improvement, "Just Go" not only provides a dynamic and reliable transportation solution but also sets a standard for the future of ride-sharing applications.