

# Project Summary

**RideWise** is a simplified, console-based **Ride-Sharing (Uber/Ola style) System**, where users can:

- Book rides
- Find drivers nearby
- Select different **ride allocation strategies** (nearest driver, least active driver, etc.)
- Apply different **fare calculation strategies**
- Track trip life-cycle

The goal is NOT to build a full app, but to **demonstrate high-quality LLD and design thinking**, showing correct application of:

Composition over inheritance

Strategy Pattern using interfaces

SOLID Principles

Low coupling / High cohesion

Law of Demeter

## Example Relationships

Relationship	Type
Rider → Ride	Association
Driver → Ride	Association
Ride → FareReceipt	Composition
RideService → Strategies	Composition

## H. File Structure

src/

  └── com/

    └── airtribe/

```
└── ridewise/
    ├── Main.java
    └── model/
        ├── Rider.java
        ├── Driver.java
        ├── Ride.java
        ├── FareReceipt.java
        └── RideStatus.java
    └── strategy/
        ├── RideMatchingStrategy.java
        ├── NearestDriverStrategy.java
        ├── LeastActiveDriverStrategy.java
        ├── FareStrategy.java
        ├── DefaultFareStrategy.java
        └── PeakHourFareStrategy.java
    └── service/
        ├── RiderService.java
        ├── DriverService.java
        └── RideService.java
    └── exception/
        └── NoDriverAvailableException.java
    └── util/
        └── IdGenerator.java
docs/
    ├── Requirements.md
    └── Class_Model.md
```

└── SOLID\_Reflection.md

└── Object\_Relationships.md