

OOP Analysis and Design Exercise

High level system design for cycling app.

UseCases

- User can track rides (Off-road or City)
- User can challenge friends for a match of speed or distance
- User can see last week ranking by how long they've pedaled (Distance).
- User can compare performance with past week.

Classes

User

```
class User {  
    String name;  
    List<User> friends;  
    // Other properties  
    ..  
  
    Ride createRide(Ride.RideType rideType) { .. }  
  
    Challenge createChallengeType(List<Users> friends,  
                                   Challenge.ChallengeType challengeType) { .. }  
  
    Boolean compareRides(List<Ride> latestRides,  
                          List<Ride> previousRides,  
                          Boolean isCompareByDistance) { .. }  
}
```

Ride

```
class Ride {
    enum RideType { OFFROAD, ROAD }
    RideType rideType;
    List<Coordinates> waypoints;
    Calendar startDate;
    Calendar endDate;

    /* Constructor */
    Ride(RideType rideType, Date date) {
        ..
    }

    void startRide(Calendar date) { .. }

    void endRide(Calendar date) { .. }

    Double totalDistance() { .. }

    long totalTime() { /* returns total time in milliseconds */ }

    Double averageSpeed() { .. }
}
```

Challenge

```
class Challenge {
    enum ChallengeType { SPEED, DISTANCE }

    ChallengeType challengeType;
    List<Users> participants;

    /* Constructor */
    Challenge(ChallengeType challengeType) { .. }

    void addParticipants(User.. participants) { .. }
    void removeParticipants(List<User> participants) { .. }
}
```

LeaderBoard

```
/* Singleton Class */
class LeaderBoard {

    /* Private constructor */
    private LeaderBoard() { .. }

    static LeaderBoard getInstance() { .. }

    User getLeader() { .. }
}
```