Idea Submission

● Problem Statement : - What's the big idea? What are you going to solve?

Bounce Pooling: - We are going to solve the issue of demand and supply between commuters travelling on similar routes. As we all know there is a limited supply of bounce scooters, so everyone is not able to pick up the bounce as per his needs. we have observed that many bounce riders are going alone so they can accommodate one more rider. So why not give them that option.

● Description of the desired application: -

For this hackathon idea we are putting ourselves in the shoes of bounce, and **Bounce Pool** will be an extra feature over the existing app**.** It will consist of two parts intended for different user audience i.e. person picking up the bounce and person who is wanting to be pooled. Rider while booking the ride will be prompted for pooling. If he opts for the same, then he will be shown the nearby poolers. The person who wants himself to be pooled, can request for the same using the bounce app. For this, he needs to enter his source and destination. The person who is being pooled can opt for pickup on his source location or the nearby bounce spots. This advantage of the latter will be a reduction on the cost of the travel + time for the same as well. The pickup option will be shown only on the basis of possibility if any based on the algorithm, the details of which will be mentioned in the solutioning part. Also, it is possible that the person who is wanting to be pooled may not wish to be available for all the time so we will be having a request expiration time per user being pooled which the pooler will enter on raising a request. If his request is not looked after by any person in a given amount of time, then the same will be cancelled after the end of the time.

● Core user scenarios: -

- Demand of commuters is more and there are less vehicles
- Commuter doesn’t have a driving license
- Dividing cost of travel
- Commuter does not know driving.

● Team details: -

Himanshu Gupta (Mobile and Backend)
Jay Joshi (Mobile and Backend)
Rishi Sharma (Mobile and Backend)

● Tech Stack: -

Backend: Java, Spring-boot, Postgres, Kafka
Mobile: Java, Flutter (Android)

● Solutioning: -

Backend Implementation: - On the Backend side, we will have 3 Microservices backed by spring boot.

1. **Rider Microservice**: - The Rider Microservice will identify the nearby bounce scooters via analysing the location of the scooters idle and will show them to the users in the fixed circle of say 1Km. The rider upon reaching the scooter after starting the ride will be given option if he wants to pool his vehicle. If he agrees, then Rider microservice will find out the nearest poolers from their locations shared and list them and the

rider can then select one of those poolers who he wants to share the ride with. The rider microservice will not allow cross genders to match as well.

2. **Pooling Person Microservice**: - The pooling microservice will allow pooling person to raise the request for being pooled. This request will then be sent as a Kafka event to the rider microservice which will consume and process it and this request will be added to the eligible riders list. The rider on accepting the request, will publish an event to the pooling microservice which on consuming it will find out if any bounce spot fits on optimizing the time and the path. If the path is getting optimized, the pooling person will be presented with 2 options saying, get a ride from the nearest bounce and save more or continue with the same location. The former will be suggested only if helps to decrease the time of the travel. Upon confirming the same, the rider will be notified of the intermediate location he needs to go to pick up the pooling person.
3. **Payments Microservice**: - The payment will work on the payment part that is it will calculate the pay for both riders while in case of pooling condition. The polling person will be shown pay according to the distance from his pickup to destination + distance rider has taken to pick him up + margin. The pay for the rider will be calculated considering the factor that rider may take some extra turns may be for his work or something else. So the total distance travelled by the rider – the distance taken by him to reach rider if that is out of its way + margin.

In case of bounce spots, the distance taken by the rider to pick up the pooling person will be divided in half among them.

Mobile Implementation :-

Mobile App will have effectively two use cases : Rider and Pooling Person

Rider : He can book the bounce and can enable the pool option if he wants to allow pooling of bounce. After starting the ride, he can see the nearby Pool requests based on the location. He'll be able to see the Pick-up and drop location and based on that he can make a choice to accept the pool request and also based on the feasibility of rider's own route. After pool ride starts, then pooler can end the ride once he reaches the destination. Once pool ride ends, rider can again get a new pool or follow his own path.

Pooling Person: He can initiate a Pool request and fill-in the details like pickup location and destination and the expiry time for the pool request. Expiry time for the pool request denotes the time for which time he'll be available at that location to be picked up. After the rider confirms the pool request, he can see the progress of his pool request and location of bounce vehicle. After reaching the destination, he can end the ride. So that rider is again free to follow his own path.