**Title of the project:** **UBER REPLICATION MODEL**

**Team members:** Raashi Agarwal (9917103218), Vrinda Goyal (9917103238), Tripti Shukla (9917103253)

**INTRODUCTION**

“Uber Replication Model” aims at providing a systematic management of resources for the cab drivers and the customers using the cab service. The main objective of the project is to handle the management, movement and finances of cabs and their customers using various data structures in an effective manner. It will not only increase the efficiency of a typical cab management system but will also provide a raw implementation of how a backend server actually works.

We will create a tree of a particular city with child nodes as the areas and localities in it.

* Each locality of this tree will be linked with multiple cab drivers/cabs present in that locality using a class object.
* There will be two interfaces of the cab management server: -

1. User /customer Interface: It will give the customers the ability to enter their pickup and destination location.

* Apply coupons available with them.
* Rate the drivers based on the service provided.
* Change or update the personal details.

1. Cab Driver Interface: It will calculate the total price of each ride taken by a particular customer that the driver will receive.

* Calculate the total earning per day of each driver.
* Allow the drivers to update their current location so as to synchronize them with the locality tree.
* Cabs available in a locality will be ordered according to the rating given by each customer.
* Data Structures used:
* Tree – for making the classification of localities.
* Heap – for connecting the rides
* Graph – for creating a network of cabs.
* Queue – customers will be inserted in the queue on clicking the ride now option.
* Priority Queue – if a customer clicks on ride later option, then they will be inserted into the priority queue with the time at which they want to take the cab.
* File Handling – managing the details of cabs and customers’ personal information
* Linked List – used for storing the various destinations wherein the rate between two nodes will be fixed.
* Stack, Graph and Standard Template Libraries may also be used in various functionalities.
* Customer class, cab driver class will be created and implemented using UML.