1. Title:

Design activity document of open ended Assessment

2. Team Members:

**Name USN Roll NO**

**Abhishek PM 01FE18BCS010 10**

**Aman Khan 01FE18BCS028 28**

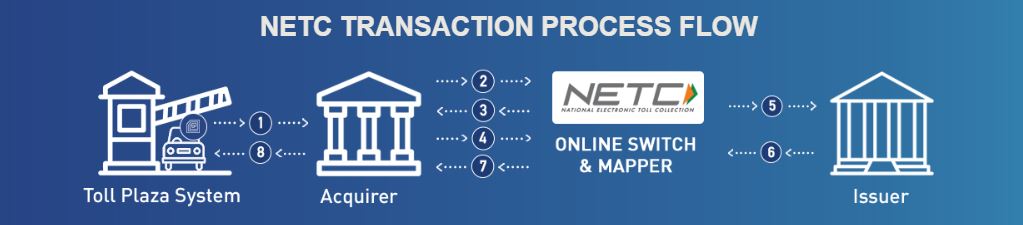
**Aditya Mishra 01FE18BCS018 18**

3. Problem Definition:

To simulate the working of National Payments Corporation of India's (NPCI) National Electronic Toll Collection (NETC) program designed to meet electronic tolling requirements of Indian Market. NETC enables a device known as FASTag as a payment mode on any toll plaza. FASTag is attached on the windscreen of a vehicle and employs Radio frequency Identification technology (RFID) for making toll payments directly from prepaid or savings account linked to it.

4. Objects Identified: ([source](https://www.npci.org.in/netc))

Following diagram shows the flow of operation on any toll plaza



1. Toll plaza captures tag details and sends it to acquirer bank

2. Acquirer sends details to NETC Mapper to validate tag

3. NETC responds with details if registered or with not registered

4. Acquirer calculates appropriate toll fare and initiates debit   
 request to NETC System.

5. NETC switches request to respective issuer bank for debiting  
 the account of the customer

6. Issuer shall debit and send the response to the NETC system.

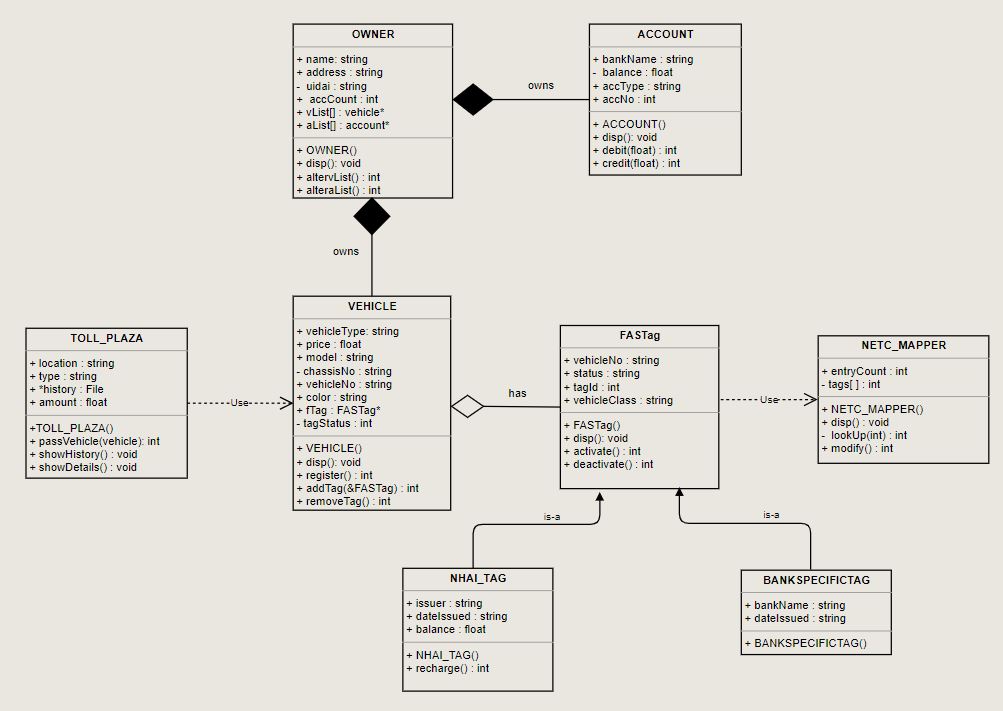
7. NETC notifies acquirer host.

8. Acquirer host notifies respective toll plaza system.

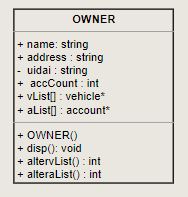
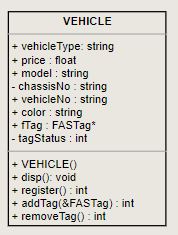
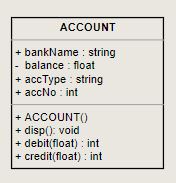
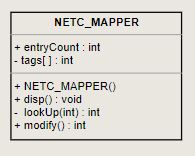
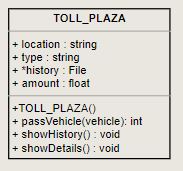
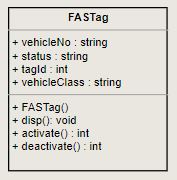
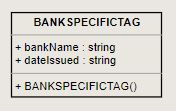
List of objects identified:

* Toll Plaza
* Owner
* NETC Mapper
* Accounts
* FASTag
  + NHAI Tag
  + Bank Specific Tag
* Vehicle

5. Class Diagram Sketch: (UML)



6. Description of classes

1. OWNER  
     
     
     
   methods altervlist () and alteralist () will be used to add or reduce the no: of account and vehicle owned by the owner as shown in the UML diagram as well.  
   We may be using parameterized constructor along with default constructor during implementation.
2. Vehicle  
     
     
     
   method register will be used to register the vehicle during which unique vehicle number will be allotted to the vehicle, addTag(&FASTag) will be used to attach Tag to the vehicle and address of Tag will be stored in ftag attribute which is a pointer of FASTag type, also tagStatus will be changed.  
   addTag(&FASTag) and removeTag() both return int values indicating success or failure of operation.
3. Account  
     
      
     
   Default constructor Account() assigns a unique account no: right away to every object that is created, account type maybe savings or current which is of no further interest in our problem, debit() and credit() alters the balances attribute of the class.
4. NETC Mapper  
     
     
     
   This object will be referenced by FAStag’s every time they are activated or deactivated as tags are only valid if they are present inside NETC Mapper, in some cases we might have several instances of Mappers in which case tags will be looked up in some or all of them. Modify () will both add and delete tag entries inside mapper.
5. Toll Plaza  
     
      
     
   Toll plaza allows vehicle to pass in passVehicle(vehicle) method after validating tag and deducting fare money. It also saves the record in a file which keeps the history of all transaction of the toll plaza. showDetails () shows the details of toll plaza .
6. FASTag  
     
     
     
   FASTag if valid allows vehicle to pass through toll plaza. A vehicle can only have one tag at a time. FASTag is given according vehicle class which is stored in vehicleClass attribute which also decides the fare prices. FASTag are of two types which is extended from this class as shown in UML. ([source](https://www.businesstoday.in/top-story/which-banks-sell-fastags-and-how-to-recharge-all-you-need-to-know/story/390801.html))
7. NHAI Tag  
     
     
     
   This are the tags issued by e-commerce websites or from walk in sales counter, this tags are bank neutral and have to be recharged in order to be used , issuer has the name from where tag is issued. Balance attribute serve as the source for the toll fare, the method recharge () returns an int indicating success or failure of the operation.
8. Bank Specific Tags  
     
     
     
   This are the tags issued by one of the designated banks (22 as of now in India), these tags don’t need to recharged as they are linked to a bank account and available balance on the account act as the source of the money that goes into toll fare at the plaza.

7. Main function & Flow of object creation and function calls:

//Class definition’s  
#define max

NETC\_Mapper map1;  
int registrationCount; //updated every time vehicle is registered  
string vehicleNumber; //Used to give unique number to vehicles  
int tagids; //for fastag unique ids

Int main ()   
{  
 OWNER owners [max];  
 FILE \*fp;  
  
 //read details from file into owners  
  
 //selection\_menu ()  
   
 //switch

}

Object creation:  
  
Every time an object of owner class is created, Pointer array for vehicles and accounts is created which stores the addresses of vehicles and accounts linked with an owner after calls to function to add/delete vehicles or accounts.  
  
Vehicle has ftag attribute which is a pointer of FASTag type which stores the address of tag linked(if any) or Null , tags are added through addTag(&FASTag) method.