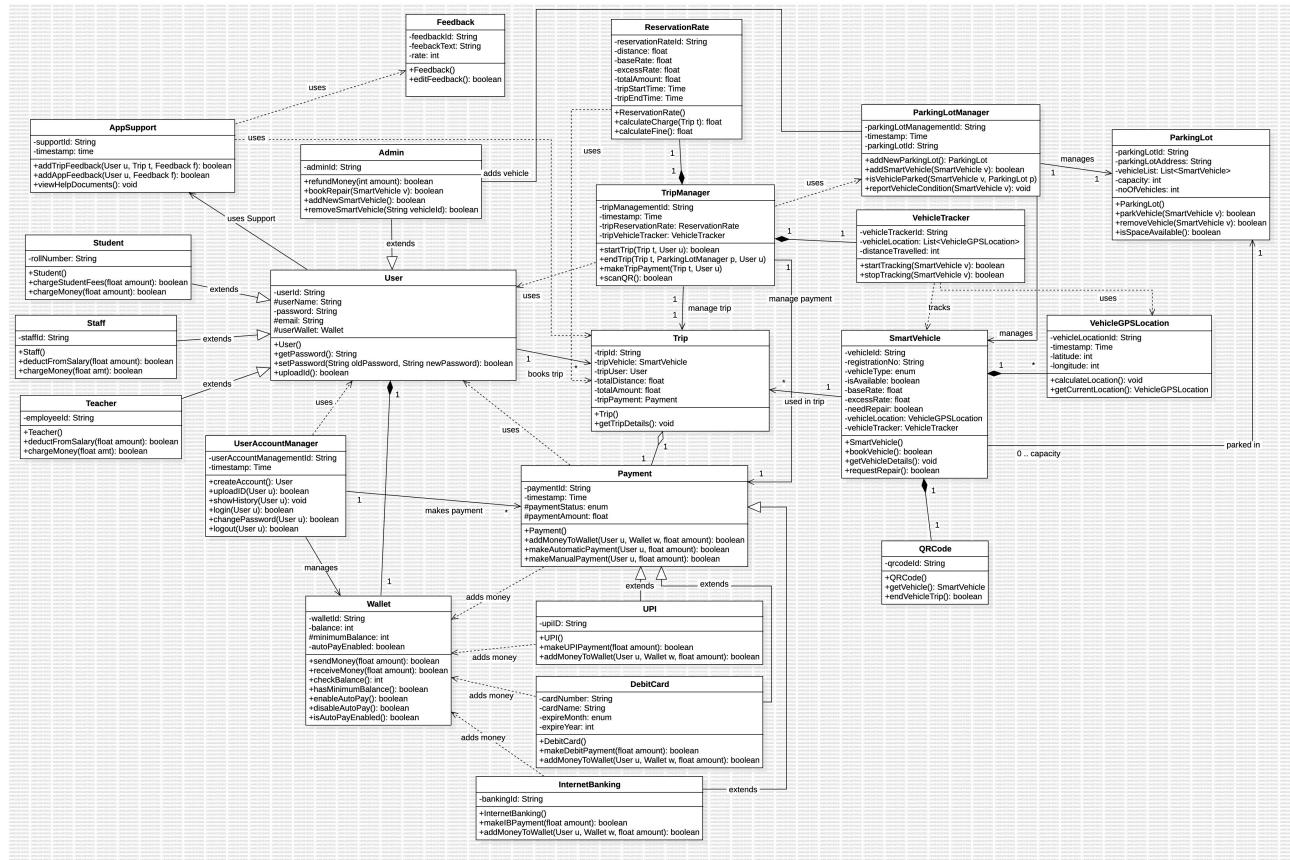
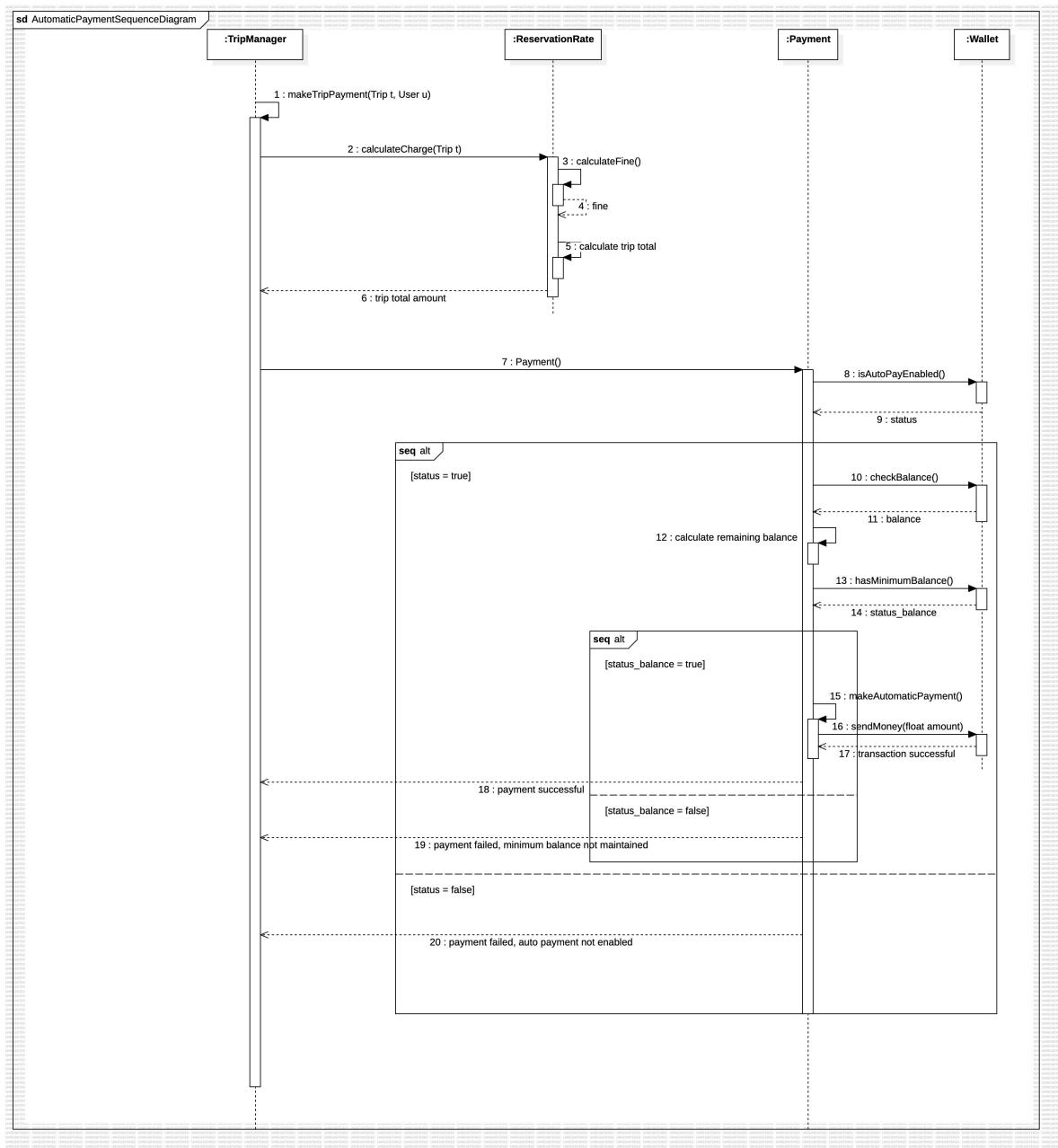


Name: Madhusree Bera
Roll: 2022202007
SE Assignment 1

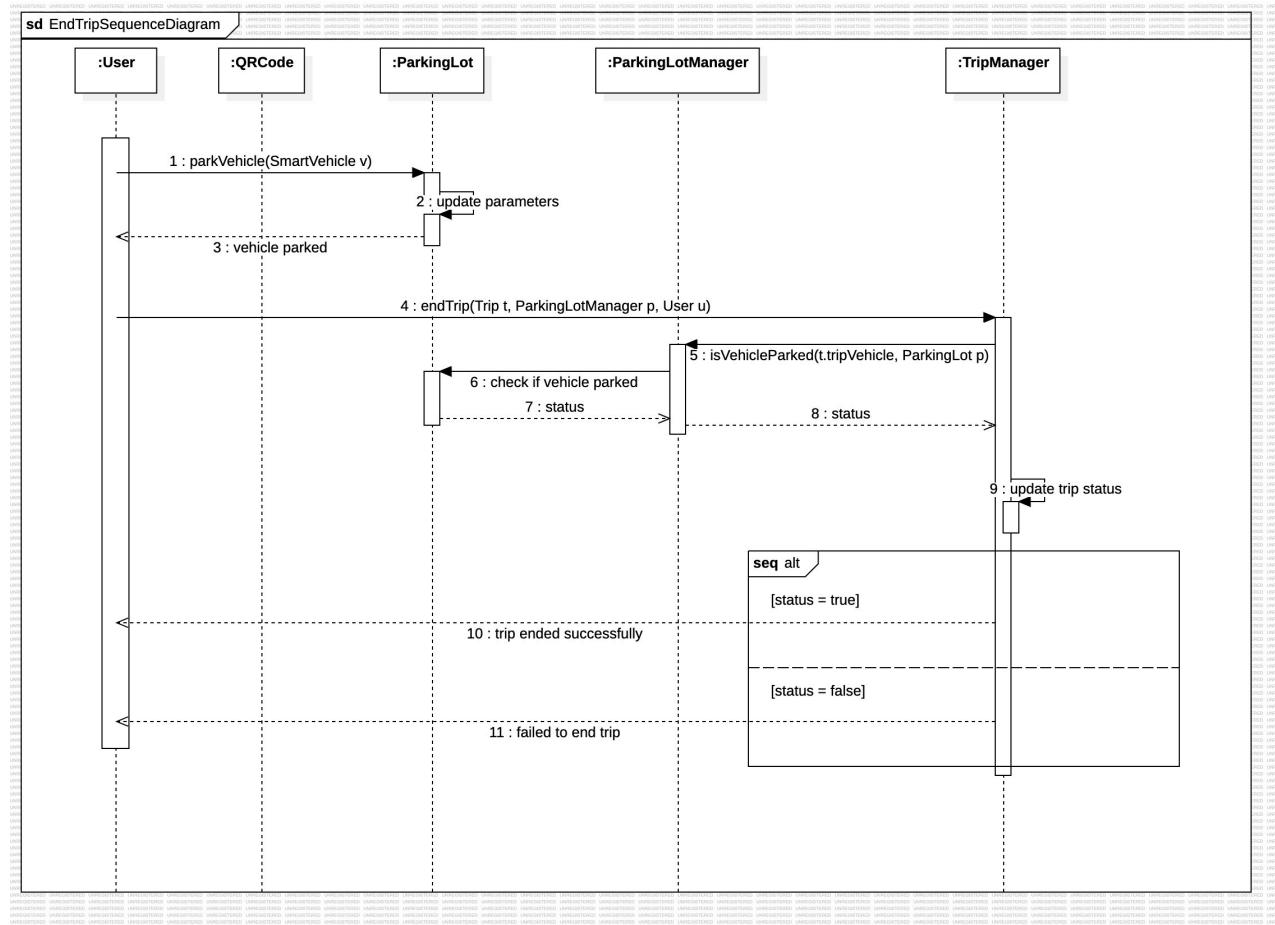
1. a i) Class Diagram



1. a ii) Sequence Diagram for Automatic Payment



Sequence Diagram for Ending Trip



1 b. Explanation

Users

1. **User Class:** This class is the base user class with basic functionalities like `getPassword`, `setPassword`, `uploadID`. One user can own one wallet.
2. **Student Class:** Subclass of User, used to create student objects having roll number. A student can be charged money (`chargeMoney` method) by adding to fees using `chargeStudentFees` method.
3. **Staff Class:** Subclass of User, has staffID. A staff can be charged money (`chargeMoney` method) by deducting from salary using `deductFromSalary` method.
4. **Teacher Class:** Subclass of User, has employeeID. A teacher can be charged money (`chargeMoney` method) by deducting from salary using `deductFromSalary` method.
5. **Admin Class:** Subclass of User, has administrative powers to add and remove new vehicles, book a repair for a vehicle, refund money to other users.

User Account Management

6. **UserAccountManager class:** used to manage user account. It makes use of User class, Wallet class and manages the Payment class by making their objects and calling the payment functions.

Smart Vehicle and related classes

7. **SmartVehicle Class:** Used to create objects of smart vehicle. The `vehicleType` is enum that can take values: bike, bicycle or moped. Since the functionalities related to various type of vehicle is same, so further subclasses are not created in this class diagram.
8. **VehicleGPSLocation Class:** A smart vehicle can have GPS location object for getting the current latitude and longitude of the vehicle at a given time.
9. **QRCode Class:** Every vehicle must have only one unique QRCode.

Bike reservation rate

10. ReservationRate Class: create objects to calculate fine and total amount based on the startTime, endTime, baseRate for initial ‘x’ km distance travelled and excessRate for every 100 meters. If a bike is not returned to the docking station within 8 hours and the bike is not renewed, a fine of 50 rupees is deducted every day.

Payment Management

11. Wallet Class: Class for all the wallet related functionalities. It has property and methods to enable and disable automatic payment.

12. Payment Class: Super class to handle the payment related methods. UserAccountManager can make payment and TripManager can initiate payment using this class.

13. UPI, DebitCard, InternetBanking classes: Subclasses of Payment class, all of these have some different properties, hence inheritance is used. Money can be added to wallet as well as manual payment for trips can be done.

Support, Feedback and ratings

14. Feedback Class: Class used to create feedback and rating given by user for the app or any trip

15. AppSupport class: handles adding feedback for trip and app, also provides Support help documents to User.

Trip Management

16. Trip Class: A trip is used to create objects which is associated with User, Smart Vehicles and aggregation of Payment object as Payment must be a part of Trip.

17. TripManager Class: The trip manager class does the main functionality of starting trip when QR is scanned and Trip object is created using the vehicle details and User. It also handles ending the trip, confirms from Parking Lot if vehicle is parked, calculates fare using ReservationRate and VehicleTracker object for the trip, finally initiates payment and make user pay (auto or manual)

18. VehicleTracker class: Uses VehicleGPSLocation to track the vehicle and calculate total distance travelled.

Parking Lot Management

19. ParkingLot class: Class to create parking lot objects that has list of smart vehicles. It parks smart vehicles and can have maximum till the capacity of parking lot

20. ParkingLotManager class: used to park, add or remove vehicles, check if vehicle is parked properly. It also reports vehicle condition if repair is needed so that admin can book a repair.

2. Assumptions

1. All classes having private attributes have getter and setter methods that allows only the validated classes to access the variables using getters and setters.
2. All users are authenticated users having driving license as they are at IIIT
3. The vehicle location is traced using GPS to calculate the latitude and longitude
4. For booking repair, manual inspection needs to be done periodically by assigned staffs.
5. The ID uploaded by users will be correct and does not need further verification
6. After parking a vehicle, the QR code is scanned to ensure that the vehicle is placed at the parking lot and location is tracked.
7. Separate functions not created for repairing each kind of vehicle as the need of each repair can only be understood after manual inspection
8. Time is a data type assumed to exist and stores epoch time.