PrestoCardSystem ~ users: ArrayList<User> = new ArrayList<>() ~ admin: AdminUser = new AdminUser("Boss", "1234")

+ PrestoCardSystem()

~ doEvent(data: ArrayList<String>()): void

User

userName: StringemailAddress: StringmyCards: ArrayList<card>

- password: String

- month: int

- totalMonth: int = 1

- averageMonthlyFare: double = 0.0

+ User()

~ buyCard(): void

~ addCard(card: Card): void

 \sim lostCard(id: int): void

~ getMyCards(): ArrayList<Card>

 $\sim changeName(newName;\,String):\,void$

~ getUserName(): String

 \sim getEmailAddress(): String

 $\sim update Average Monthly Fare (month: String, fare: double): void\\$

~ getAverageMonthlyFare(): double

~ getTotalMonth(): int

- convertMonth(month: String): int

AdminUser

- userName: String = "Admin"

- password: String

~ totalFare: Double

~ totalStations: int

+ AdminUser()
~ report(): String
~ clearData(): void

Card

- isSuspended: boolean = false

- id: int - user: User

balance: double

- idIncrementer: int = 1000

- myTrip: ArrayList<Trip> = new ArraryList<>()

~ Card()

~ reverseSuspended(): void

~ getId(): int

~ setUser(user: User): void ~ increaseBalance(i: int): void

- deductFare(vehicle: String): void

~ recordTrip(vehicle: String, enterOrExit: String, time: Date, station: Station): void

 \sim getBalance(): void

~ recentTripString(): String

StationManager

- stationSet: ArrayList<Station> = new ArrayList<>()

- stationIDSet: ArrayList<String> = new ArrayList<>()

- stopSet: ArrayList<Station> = new ArrayList<>()

- stopIDSet: ArrayList<Station> = new ArrayList<>()

Station

- name: String

- distance: Integer

- neighbours: ArrayList<Station> = new ArrayList<>()

vehicle: String

~ Station(name: String)

~ setDistance(distance: Integer): void

- + StationManager()
 ~ addStations(stations)
- ~ addStations(station: Station); void
- ~ getStationSet(): ArrayList<Station>
- ~ getStationIDSet(): ArrayList<String>
- ~ minDistance(source: Station, destination: Station): int
- ~ addStops(station: Station); void
- ~ getStopSet(): ArrayList<Station>
- ~ getStopIDSet(): ArrayList<String>
- ~ newStation(StationID: String, vehicle: String): Station

- ~ getDistance(): integer
- ~ getNeighbours(): ArrayList<Station>
- ~ getName(): String
- ~ addNeighbours(neighbour: Station): void
- ~ getVehicle(): String

Trip

- entrance: Station
- exit: Station
- enterTime: Date
- exitTime: Date
- transportation: String
- isContinous: boolean = false
- currentFare: double
- continousTime: Long
- ~ Trip(entrance Station, enterTime: Date, vehicle: String)
- ~ getEntrance(): Station
- ~ getExit(): Station
- ~ setExit(exit: Station, exitTime: Date): void
- ~ setContinous(): void
- ~ setDiscontinous(): void
- ~ setCurrentFare(fare; double): void
- ~ setContinuousTime(time: Long): void
- ~ getCurrentFare(): double
- ~ getContinuousTime(): Long
- ~ getIsContinuous(): boolean
- ~ tripTime(): Long
- ~ getEnterTime(): Date
- ~ getExitTime(): Date
- ~ getTransportation(): String