Functional requirement	Class Name	Method Name
R1: Register vehicle	Main Class	registerVehicle():void
	Controller Class	registerVehicle(…):boolean
	Motorcycle Class	Motorcycle(…)
	Controller Class	registerVehicle(…):boolean
	GasolineCar Class	GasolineCar(…)
	Controller Class	registerVehicle(…):boolean
	ElectricCar Class	ElectricCar(…)
	Controller Class	registerVehicle(…):boolean
	HybridCar Class	HybridCar()
	Controller Class	registerDocuments(…):Document[]
	Controller Class	isOld(Vehicle vehicle):boolean
	Parking Class	add(Vehicle vehicle):void
	Parking Class	findSpaceInColumn(int column):int
	SOAT Class	SOAT(…)
	TechnoMechanical Class	TechnoMechanical(…)
	PropertyCard Class	PropertyCard(…)
R2: Calculate selling price	Main Class	calculateSellingPrice():void
	Controller Class	calculateSellingPrice(String id):double
	Vehicle Class	calculateSellingPrice():double
R3: Generate report	Main Class	generateReport():void
	Controller Class	generateReport(int option):String
	Controller Class	generateReportByType():String

	Controllller Class	generateReportByFuel():String
	Controller Class	generateReportByUsage():String
	Motorcycle, GasolineCar, ElectricCar, HybridCar Class	toString():String
R4: Show document status	Main Class	showDocumentStatus():void
	Controller Class	showDocumentStatus(String id):String
	Controller Class	findVehicle(String id) : Vehicle
	Vehicle Class	getDocument(int index) : Document
	LocalDate Class	LocalDate.now().getYear() : int
R5: Show parking map	Main Class	showParkingMap():void
	Controller Class	showParkingMap():String
	Parking Class	showParkingMap():String
R6: Generate parking report	Main Class	generateParkingReport() :void
	Controller Class	oldCarsInRange(int start, int end):String
	Parking Class	oldCarsInRange(int start, int end):String
	Controller Class	oldestVehicle():String
	Controller Class	newestVehicle():String
	Vehicle Class	toString():String
	Controller Class	parkingOccupancyPercentage():String
	Parking Class	parkingOccupancyPercentage():String