Formal Methods In Software Engineering (SE-306-A)

Stage – 2 : (OCL Research)

Project Title:

University Smart Bus & Route Management System

Submitted by:

Maida Kosser (221400091)

Zoya Khalid (221400095)

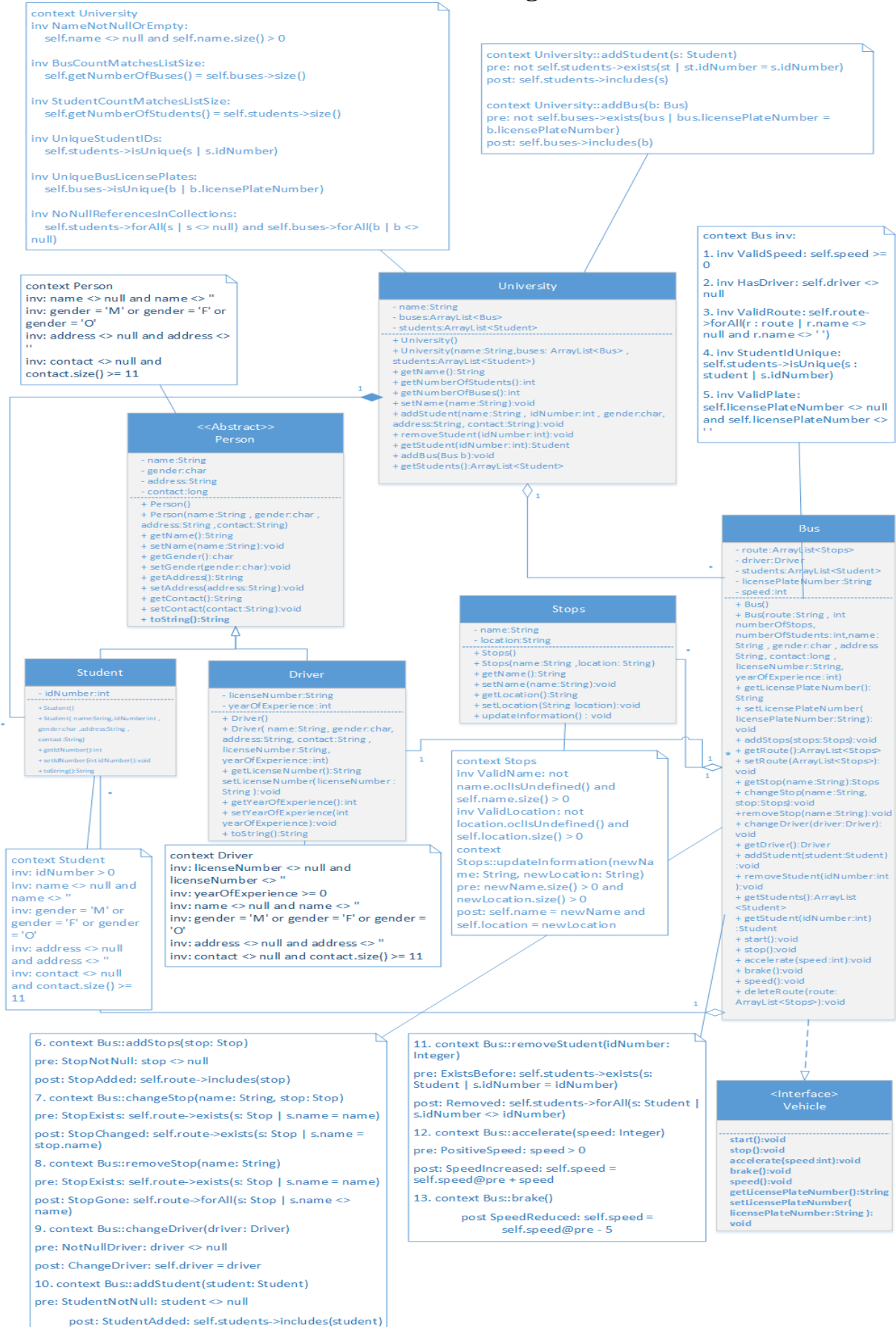**Muhammad Asad Ullah (221400089)**

Muhammad Noman (221400087)

Submitted to:

Mam Sajiya Tariq

BS Software Engineering (VI)

GIFT University, Gujranwala

Task – 1: OCL Diagram

context University
inv NameNotNullOrEmpty:
  self.name <> null and self.name.size() > 0

inv BusCountMatchesListSize:
  self.getNumberOfBuses() = self.buses->size()

inv StudentCountMatchesListSize:
  self.getNumberOfStudents() = self.students->size()

inv UniqueStudentIDs:
  self.students->isUnique(s | s.idNumber)

inv UniqueBusLicensePlates:
  self.buses->isUnique(b | b.licensePlateNumber)

inv NoNullReferencesInCollections:
  self.students->forAll(s | s <> null) and self.buses->forAll(b | b <> null)

context University::addStudent(s: Student)
pre: not self.students->exists(st | st.idNumber = s.idNumber)
post: self.students->includes(s)

context University::addBus(b: Bus)
pre: not self.buses->exists(bus | bus.licensePlateNumber = b.licensePlateNumber)
post: self.buses->includes(b)

context Person
inv: name <> null and name <> ''
inv: gender = 'M' or gender = 'F' or gender = 'O'
inv: address <> null and address <> ''
inv: contact <> null and contact.size() >= 11

context Bus inv:

1. inv ValidSpeed: self.speed >= 0

2. inv HasDriver: self.driver <> null

3. inv ValidRoute: self.route->forAll(r : route | r.name <> null and r.name <> ' ')

4. inv StudentIdUnique: self.students->isUnique(s : student | s.idNumber)

5. inv ValidPlate: self.licensePlateNumber <> null and self.licensePlateNumber <> ' '

University

- name:String
- buses:ArrayList<Bus>
- students:ArrayList<Student>

+ University()
+ University(name:String,buses: ArrayList<Bus> , students:ArrayList<Student>)
+ getName():String
+ getNumberOfStudents():int
+ getNumberOfBuses():int
+ setName(name:String):void
+ addStudent(name:String , idNumber:int , gender:char, address:String, contact:String):void
+ removeStudent(idNumber:int):void
+ getStudent(idNumber:int):Student
+ addBus(Bus b):void
+ getStudents():ArrayList<Student>

<<Abstract>> Person

- name:String
- gender:char
- address:String
- contact:long

+ Person()
+ Person(name:String , gender:char , address:String ,contact:String)
+ getName():String
+ setName(name:String):void
+ getGender():char
+ setGender(gender:char):void
+ getAddress():String
+ setAddress(address:String):void
+ getContact():String
+ setContact(contact:String):void
+ toString():String

Bus

- route:ArrayList<Stops>
- driver:Driver
- students:ArrayList<Student>
- licensePlateNumber:String
- speed:int

+ Bus()
+ Bus(route:String , int numberOfStops, numberOfStudents:int,name:String , gender:char , address String, contact:long , licenseNumber:String, yearOfExperience:int)
+ getLicensePlateNumber(): String
+ setLicensePlateNumber(licensePlateNumber:String): void
+ addStops(stops:Stops):void
+ getRoute():ArrayList<Stops>
+ setRoute(ArrayList<Stops>):void
+ getStop(name:String):Stops
+ changeStop(name:String, stop:Stops):void
+ removeStop(name:String):void
+ changeDriver(driver:Driver):void
+ getDriver():Driver
+ addStudent(student:Student):void
+ removeStudent(idNumber:int):void
+ getStudents():ArrayList<Student>
+ getStudent(idNumber:int):Student
+ start():void
+ stop():void
+ accelerate(speed:int):void
+ brake():void
+ speed():void
+ deleteRoute(route:ArrayList<Stops>):void

Student

- idNumber:int

+ Student()
+ Student(name:String,idNumber:int , gender:char ,address:String , contact :String)
+ getIdNumber():int
+ setIdNumber(int idNumber):void
+ toString():String

Driver

- licenseNumber:String
- yearOfExperience :int

+ Driver()
+ Driver(name:String, gender:char, address:String, contact:String , licenseNumber:String, yearOfExperience:int)
+ getLicenseNumber():String
setLicenseNumber(licenseNumber:String):void
+ getYearOfExperience():int
+ setYearOfExperience(int yearOfExperience):void
+ toString():String

Stops

- name:String
- location:String

+ Stops()
+ Stops(name:String ,location: String)
+ getName():String
+ setName(name:String):void
+ getLocation():String
+ setLocation(String location):void
+ updateInformation() : void

context Stops
inv ValidName: not name.oclIsUndefined() and self.name.size() > 0
inv ValidLocation: not location.oclIsUndefined() and self.location.size() > 0
context Stops::updateInformation(newName: String, newLocation: String)
pre: newName.size() > 0 and newLocation.size() > 0
post: self.name = newName and self.location = newLocation

context Student
inv: idNumber > 0
inv: name <> null and name <> ''
inv: gender = 'M' or gender = 'F' or gender = 'O'
inv: address <> null and address <> ''
inv: contact <> null and contact.size() >= 11

context Driver
inv: licenseNumber <> null and licenseNumber <> ''
inv: yearOfExperience >= 0
inv: name <> null and name <> ''
inv: gender = 'M' or gender = 'F' or gender = 'O'
inv: address <> null and address <> ''
inv: contact <> null and contact.size() >= 11

6. context Bus::addStops(stop: Stop)

pre: StopNotNull: stop <> null

post: StopAdded: self.route->includes(stop)

7. context Bus::changeStop(name: String, stop: Stop)

pre: StopExists: self.route->exists(s: Stop | s.name = name)

post: StopChanged: self.route->exists(s: Stop | s.name = stop.name)

8. context Bus::removeStop(name: String)

pre: StopExists: self.route->exists(s: Stop | s.name = name)

post: StopGone: self.route->forAll(s: Stop | s.name <> name)

9. context Bus::changeDriver(driver: Driver)

pre: NotNullDriver: driver <> null

post: ChangeDriver: self.driver = driver

10. context Bus::addStudent(student: Student)

pre: StudentNotNull: student <> null

post: StudentAdded: self.students->includes(student)

11. context Bus::removeStudent(idNumber: Integer)

pre: ExistsBefore: self.students->exists(s: Student | s.idNumber = idNumber)

post: Removed: self.students->forAll(s: Student | s.idNumber <> idNumber)

12. context Bus::accelerate(speed: Integer)

pre: PositiveSpeed: speed > 0

post: SpeedIncreased: self.speed = self.speed@pre + speed

13. context Bus::brake()

post SpeedReduced: self.speed = self.speed@pre - 5

<Interface> Vehicle

start():void
stop():void
accelerate(speed:int):void
brake():void
speed():void
getLicensePlateNumber():String
setLicensePlateNumber(licensePlateNumber:String):void

Member 1: Maida Kosser

Task – 2: OCL Code:

context Bus inv:

inv ValidSpeed: self.speed >= 0

inv HasDriver: self.driver <> null

inv ValidRoute: self.route->forAll(r: Stop | r.name <> null and r.name <> ' ')

inv StudentIdUnique: self.students->isUnique(s: Student | s.idNumber)

inv ValidPlate: self.licensePlateNumber <> null and self.licensePlateNumber <> ' '

context Bus::addStops(stop: Stop)

pre: StopNotNull: stop <> null

post: StopAdded: self.route->includes(stop)

context Bus::changeStop(name: String, stop: Stop)

pre: StopExists: self.route->exists(s: Stop | s.name = name)

post: StopChanged: self.route->exists(s: Stop | s.name = stop.name)

context Bus::removeStop(name: String)

pre: StopExists: self.route->exists(s: Stop | s.name = name)

post: StopGone: self.route->forAll(s: Stop | s.name <> name)

context Bus::changeDriver(driver: Driver)

pre: NotNullDriver: driver <> null

post: ChangeDriver: self.driver = driver

context Bus::addStudent(student: Student)

pre: StudentNotNull: student <> null

post: StudentAdded: self.students->includes(student)

context Bus::removeStudent(idNumber: Integer)

pre: ExistsBefore: self.students->exists(s: Student | s.idNumber = idNumber)

post: Removed: self.students->forAll(s: Student | s.idNumber <> idNumber)

context Bus::accelerate(speed: Integer)

pre: PositiveSpeed: speed > 0

post: SpeedIncreased: self.speed = self.speed@pre + speed

context Bus::brake()

post SpeedReduced: self.speed = self.speed@pre - 5

FMSE Project Status Report:

Date/Day: 18-07-2025

Project Name: University Smart Bus & Route Management System

Project Lead: Maida Kosser

Report Date: 18-07-2025

Section: A

 Group No: 11

Task	Member Roll #	Assigned Task %	Completed %
Class Diagram to OCL Diagram, Project code in OCL, Research Book	221400091	40%	40%
	221400095	20%	%
	221400089	20%	%
	221400087	20%	%
Total		100%	%

Project stage Completion Percentage: %

Comments:

_____

_____

_____

_____

_____

Project Lead: _____   **Instructor: _____**