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

Stage - 3(Z Specification)

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

Member 1: Maida Kosser – Class Bus

1. Project code in Z Notation:

Schema's:

```
Driver
getName : ℕ ↦ CHAR
```

```
Student
getIdNumber : STUDENT_ID
```

```
Stop
getName : STOP_NAME
```

```
AddStop
ΔBus

stop? : Stop

stop? ∉ route
route' = route ∪ {stop?}
driver' = driver
students' = students
licensePlateNumber' = licensePlateNumber
speed' = speed
```

```
ChangeStop
ΔBus

s?, newStop? : Stop

s? ∈ route ∧ newStop? ∉ route
∧ route' = (route \ {s?}) ∪ {newStop?}
∧ students' = students
∧ driver' = driver
∧ licensePlateNumber' = licensePlateNumber
```

```
RemoveStop
ΔBus

stopToRemove? : Stop

stopToRemove? ∈ route
∧ route' = route \ {stopToRemove?}
∧ students' = students
∧ driver' = driver
∧ licensePlateNumber' = licensePlateNumber
∧ speed' = speed
```

```
Bus
route : ℙ Stop
driver : Driver
students : ℙ Student
licensePlateNumber : LICENSE_PLATE
speed : ℕ

speed ≥ 0
```

```
InitBus
Bus'

route' = ∅
students' = ∅
licensePlateNumber' =  NULL_PLATE
speed' = 0
```

```
ChangeDriver
ΔBus

newDriver? : Driver

driver' = newDriver?
route' = route
students' = students
licensePlateNumber' = licensePlateNumber
speed' = speed
```

```
AddStudent
ΔBus

student? : Student

student? ∉ students
students′ = students ∪ {student?}
route′ = route
driver′ = driver
licensePlateNumber′ = licensePlateNumber
speed′ = speed
```

RemoveStudent

ΔBus

$idNumber? : STUDENT\_ID$

$\exists\, s : students \bullet s.getIdNumber = idNumber? \wedge$

$students' = students \setminus \{s\}$

$route' = route$

$driver' = driver$

$licensePlateNumber' = licensePlateNumber$

$speed' = speed$

Accelerate

ΔBus

$speedIncrease? : \mathbb{N}$

$speed' = speed + speedIncrease?$

$route' = route$

$driver' = driver$

$students' = students$

$licensePlateNumber' = licensePlateNumber$

Brake

ΔBus

$speed \geq 5 \Rightarrow speed' = speed - 5$

$speed < 5 \Rightarrow speed' = 0$

$route' = route$

$driver' = driver$

$students' = students$

$licensePlateNumber' = licensePlateNumber$

RobustAddStudent

$\Delta\ Bus$

$studentToAdd? : Student$

$result! : REPORT$

$result! : REPORT$

$\wedge\ students' = students \cup \{studentToAdd?\}$

$\wedge\ result! = ok)$

$\vee\ (studentToAdd? \in students$

$\wedge\ result! = already\_known)$

RobustRemoveStudent

$\Delta\ Bus$

$studentToRemove? : Student$

$result! : REPORT$

$(studentToRemove? \in students$

$\wedge\ students' = students \setminus \{studentToRemove?\}$

$\wedge\ result! = ok)$

$\vee\ (studentToRemove? \notin students$

$\wedge\ result! = not\_known)$

AddStopError

$\Xi\ Bus$

$stop? : Stop$

$stop? \in route$

RemoveStopError

$\Xi\ Bus$

$stopToRemove? : Stop$

$stopToRemove? \notin route$

AddStudentError

$\Xi\ Bus$

$student? : Student$

$student? \in students$

RemoveStudentError

$\Xi\ Bus$

$idNumber? : STUDENT\_ID$

$\neg\ (\exists\, s : students \bullet s.getIdNumber = idNumber?)$

ChangeDriverError

$\Xi\ Bus$

$newDriver? : DRIVERNAME$

$newDriver? = NULL\_DRIVER$

ChangeStopError

$\Xi\ Bus$

$s?, newStop? : Stop$

$result! : REPORT$

$(s? \notin route \wedge result! = not\_known)$

$\vee\ (newStop? \in route \wedge result! = already\_known)$

Success

$result! : Report$

$result! = ok$

Failure

$result! : Report$

$result! \neq ok$

$Report ::= ok \mid already\_known \mid$

not_known

Robust Operations:

- RobustAddStudent ≙ (AddStudent ∧ Success) ∨ AddStudentError
- RobustRemoveStudent ≙ (RemoveStudent ∧ Success) ∨ RemoveStudentError
- RobustAddStop ≙ (AddStop ∧ Success) ∨ (AddStopError ∧ Failure)
- RobustRemoveStop ≙ (RemoveStop ∧ Success) ∨ (RemoveStopError ∧ Failure)
- RobustChangeStop ≙ (ChangeStop ∧ Success) ∨ (ChangeStopError ∧ Failure)
- RobustChangeDriver ≙ (ChangeDriver ∧ Success) ∨ (ChangeDriverError ∧ Failure)

2. Conclusion:

The **Bus Class** in **Z Notation** helped me understand how to model a real-world system using formal specification techniques. I learned how to define the system's state using the Bus schema, which included key components like **route**, **driver**, **students**, **license plate number**, and **speed**. I explored how the system is **initialized** using InitBus, where the **new state** is defined with **prime notation (')**, and there's no reference to a previous state. I also understood the role of **Δ Bus** in showing that the state is **changing** in operations (like add/remove stop or student), and **Ξ Bus** to represent operations where the state stays the same**.**

I also learned how to manage error conditions using **error schemas**, which ensure that invalid actions do not affect the system. The concept of **robust schemas** was introduced, allowing me to handle both **success** and **failure** cases using a result! value that returns messages like **ok**, **already_known**, or **not_known**. This task improved my understanding of state transitions**,** input validation**,** and designing systems that are safe**,** predictable, and error-resilient. Overall, it gave me a strong foundation in using **Z notation** for creating well-structured and reliable system models.

FMSE Project Status Report:

Date/Day: 06-08-2025

Project Name: University Smart Bus & Route Management System

Project Lead: Maida Kosser

Report Date: 06-08-2025

Section: A

Group No: 11

Task	Member Roll #	Assigned Task %	Completed %
Convert Project code in Z Notation & Conclusion	221400091	28%	28%
	221400095	24%	%
	221400089	24%	%
	221400087	24%	%
Total		100%	%

Project stage Completion Percentage: %

Comments:

____

____

____

____

____

Project Lead: ____________ Instructor: ______________