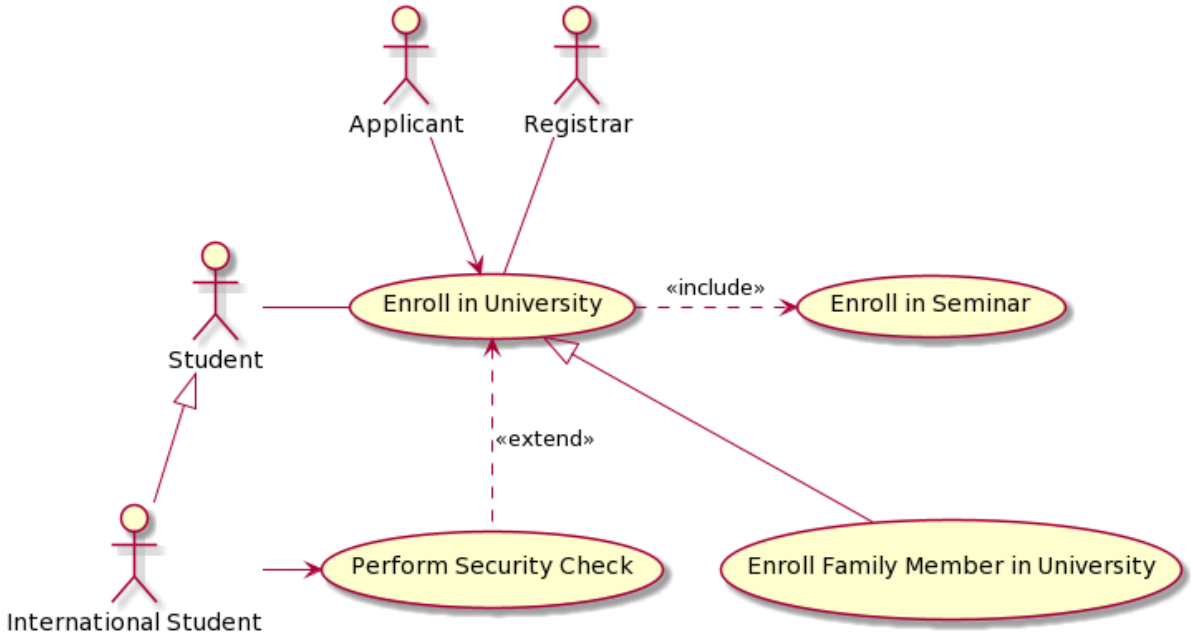
Andrico Mauludi Junianto

081811633052

Pembangunan Perangkat Lunak

1. **Usecase Diagram**

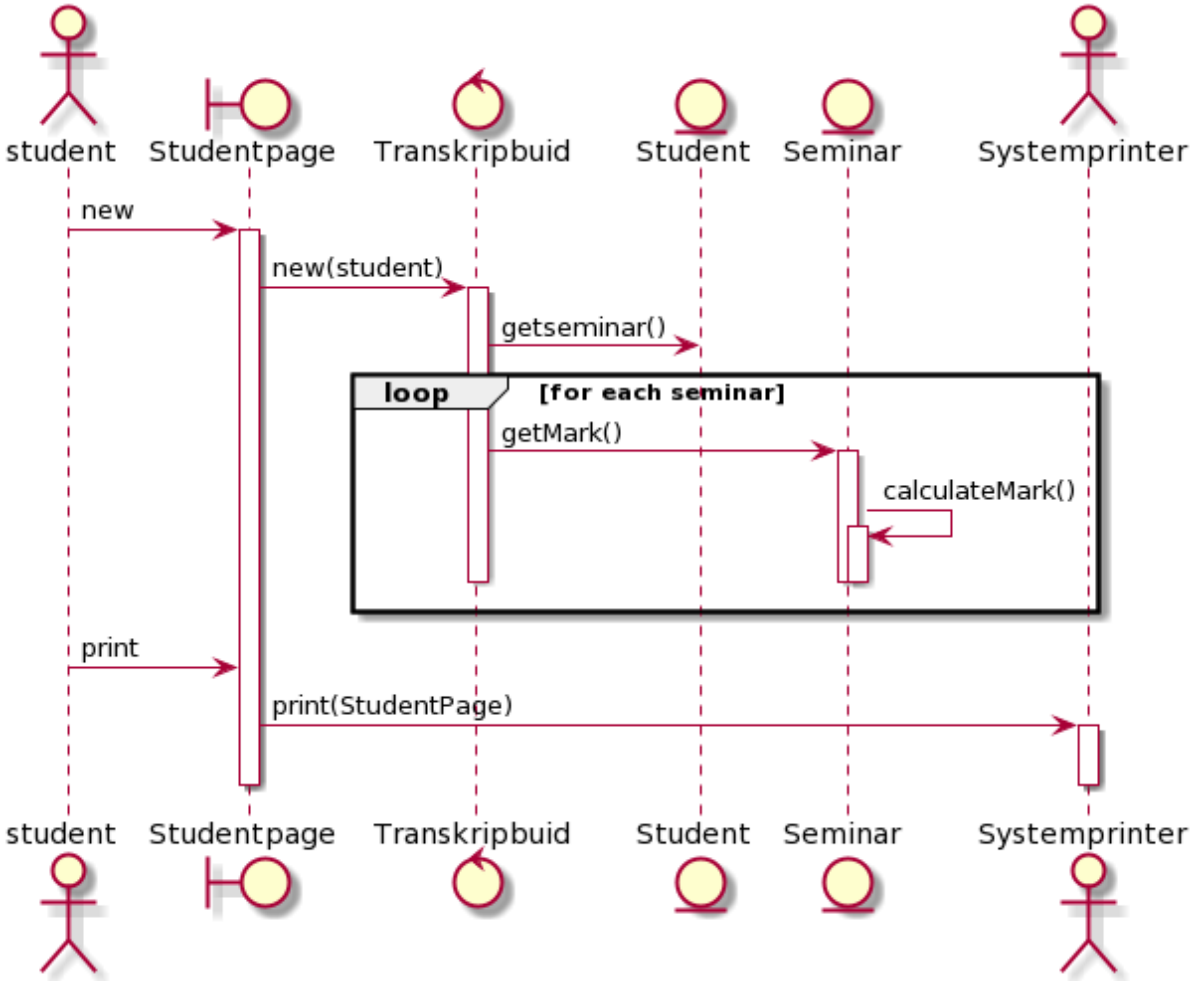


usecase.svg

PlantUML Script

```
@startuml

actor Applicant as ap

actor Registrar as rg

actor Student as st

actor "International Student" as in

Usecase "Enroll in University" as UC1

Usecase "Enroll in Seminar" as UC2

Usecase "Perform Security Check" as UC3

Usecase "Enroll Family Member in University" as UC4


UC1 <-up- ap

UC1 -up- rg

st -right- UC1

in -right-> UC3

UC1 .right.> UC2 : <<include>>

UC3 .up.> UC1 : <<extend>>

UC4 -up-|> UC1

in -up-|> st

@enduml
```

2. Sequence Diagram



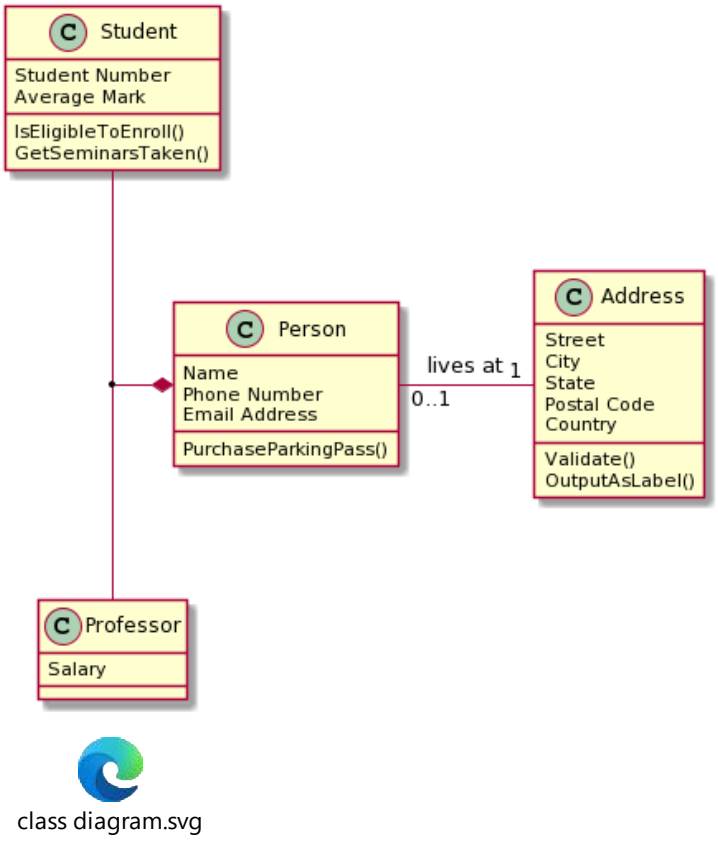
sequence diagram.svg

PlantUML Script :

```
@startuml
actor       student       as Foo1
boundary    Studentpage    as Foo2
control     Transkripbuid     as Foo3
entity      Student       as Foo4
entity      Seminar       as Foo5
actor       Systemprinter       as Foo6

Foo1 -> Foo2 : new
activate Foo2
```

```
Foo2 -> Foo3 : new(student)
activate Foo3
Foo3 -> Foo4 : getseminar()
loop for each seminar
   Foo3 -> Foo5: getMark()
   activate Foo5
      Foo5 -> Foo5 : calculateMark()
      activate Foo5
      deactivate Foo5
   deactivate Foo5
   deactivate Foo3
end
Foo1 -> Foo2 : print

Foo2 -> Foo6 : print(StudentPage)
activate Foo6
deactivate Foo6

deactivate Foo2
@enduml
```

3. Class Diagram



class diagram.svg

PlantUML Script:

```
@startuml
class Person{
  Name
  Phone Number
  Email Address
  PurchaseParkingPass()
}

class Address{
  Street
  City
  State
  Postal Code
```

```
  Country
  Validate()
  OutputAsLabel()
}

class Professor{
  Salary
}

class Student{
  Student Number
  Average Mark
  IsEligibleToEnroll()
  GetSeminarsTaken()
}

Person "0..1" -right- "1" Address : lives at

(Student, Professor) -up-* Person
@enduml
```