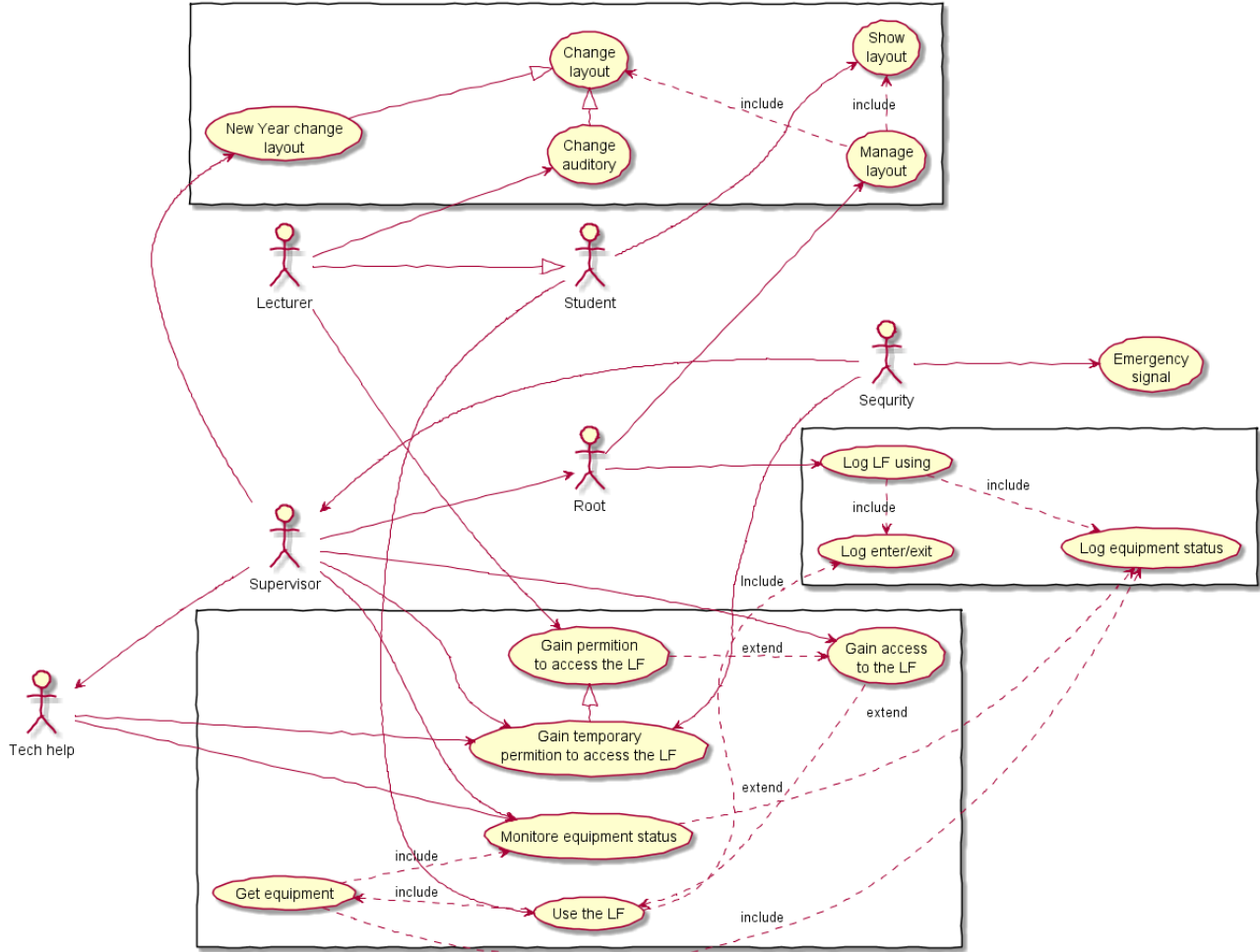


Юзкейс діаграма системи



3.3.1

@startuml

actor User

User -> System : ShowGUI Async

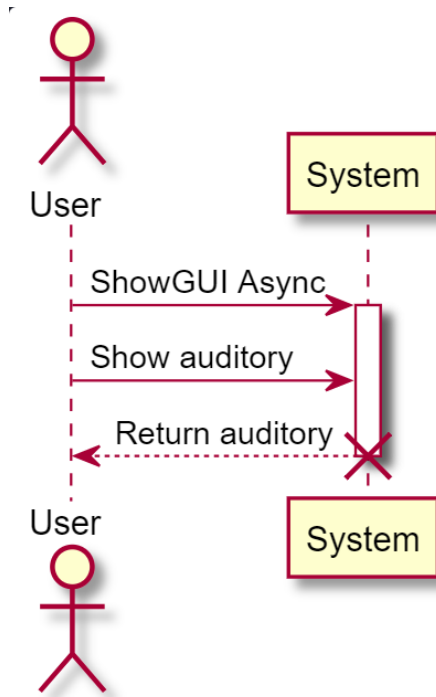
activate System

User -> System : Show auditory

System --> User : Return auditory

destroy System

@enduml



3.3.2

@startuml

actor User

User -> Supervisor : Change layout request

activate Supervisor

Supervisor -> User : Give form

User -> User : Fill form

User --> Supervisor : Return form

alt Request\nacceptable

Supervisor -> Root : Send changelog

activate Root

ref over Root : System::KeepChange();\nSystem::RefreshGUI();

Root --> Supervisor : Report success

destroy Root

Supervisor --> User : Report success

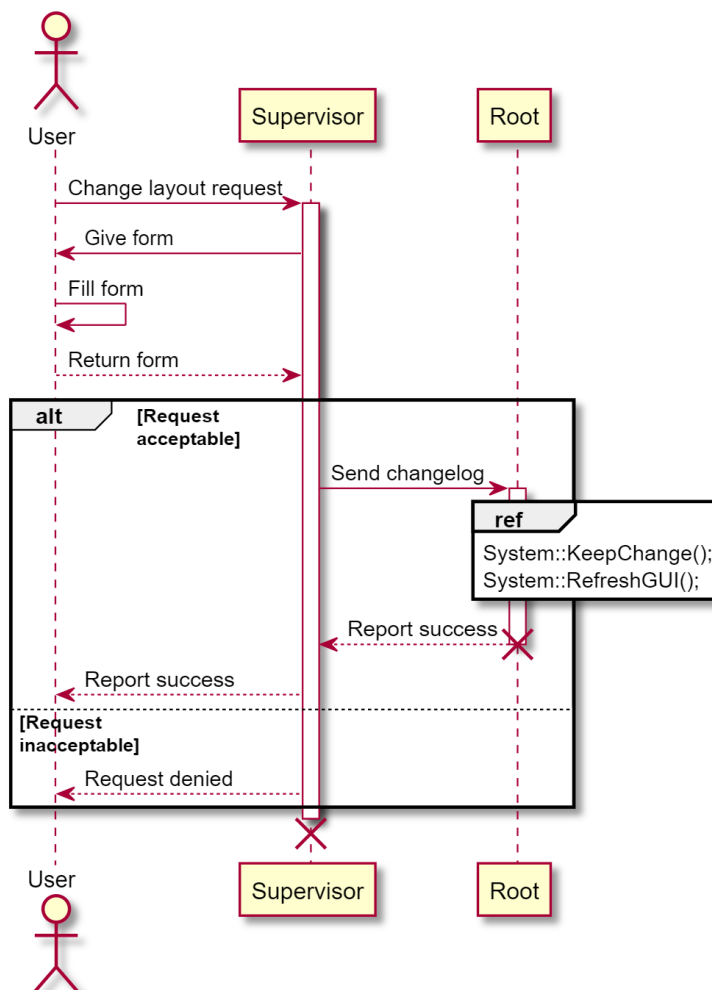
else Request\ninacceptable

Supervisor --> User : Request denied

end

destroy Supervisor

@enduml



3.3.3

@startuml

actor Lecturer

Lecturer -> Supervisor : Give access request

activate Supervisor

Supervisor -> Lecturer : Give form

Lecturer -> Lecturer : Fill form

Lecturer --> Supervisor : Return form

ref over Supervisor : Create lecturer's UUID

Supervisor -> Root : Give access request

activate Root

ref over Root : System::AddLecturer();

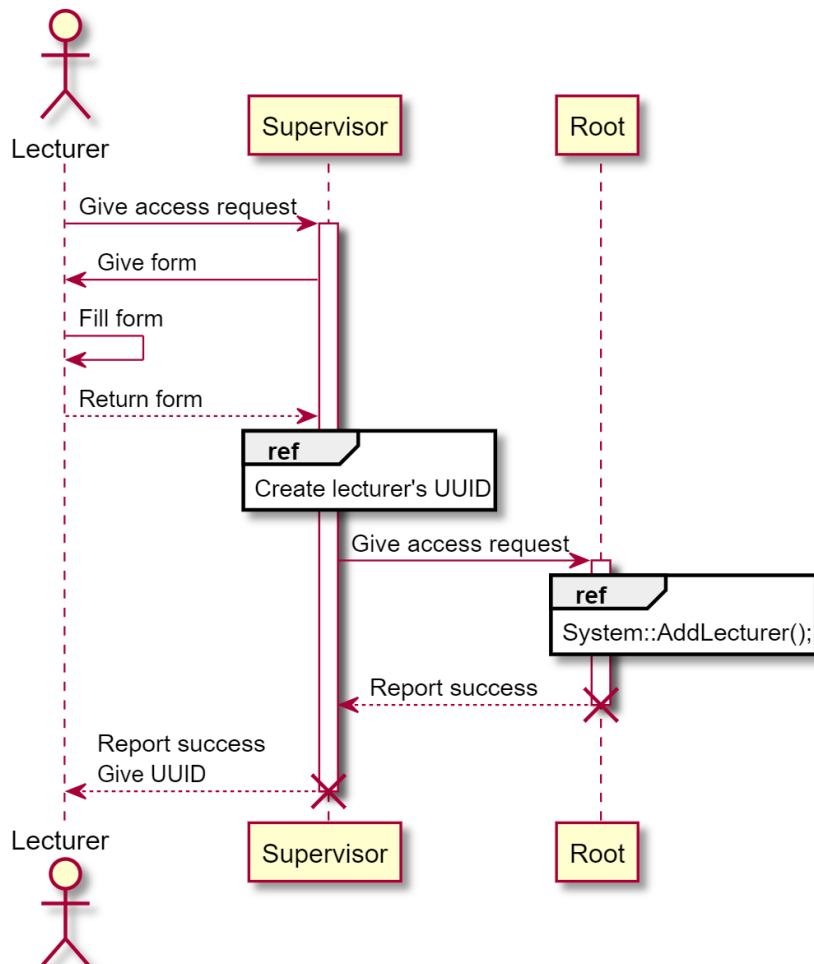
Root --> Supervisor : Report success

destroy Root

Supervisor --> Lecturer : Report success\nGive UUID

destroy Supervisor

@enduml



3.3.4

@startuml

actor Lecturer

Lecturer -> Supervisor : Change auditory request

activate Supervisor

alt Change acceptable

Supervisor -> Root : Change layout request

activate Root

ref over Root : System::DeleteLaws();\nSystem::GiveLaws();\nSystem::RefreshGUI();

Root --> Supervisor : Report success

destroy Root

Supervisor --> Lecturer : Report success

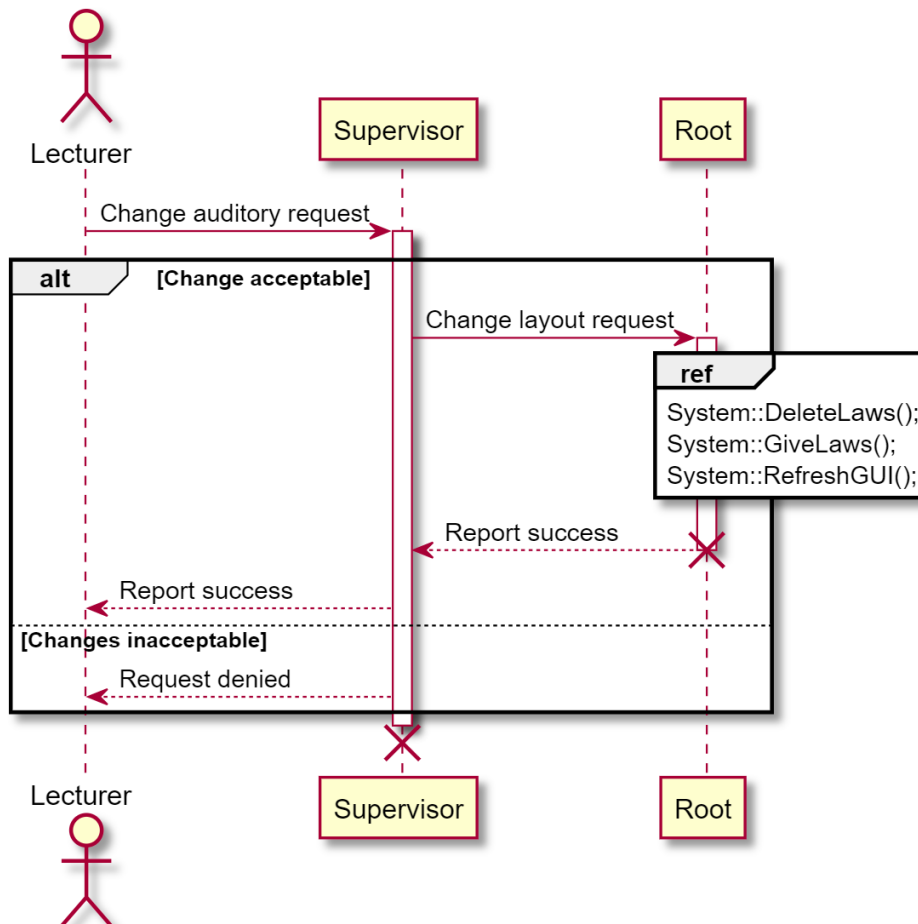
else Changes unacceptable

Supervisor --> Lecturer : Request denied

end

destroy Supervisor

@enduml



3.3.5

@startuml

actor Supervisor

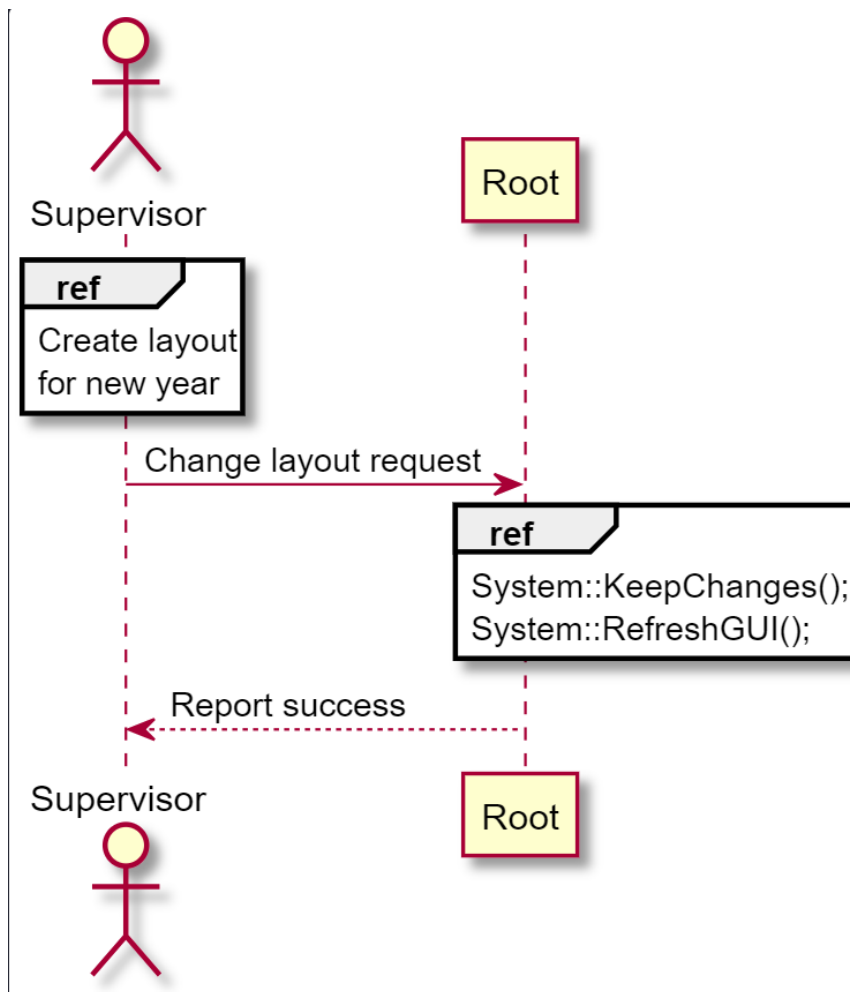
ref over Supervisor : Create layout\nfor new year

Supervisor -> Root : Change layout request

ref over Root : System::KeepChanges();\nSystem::RefreshGUI();

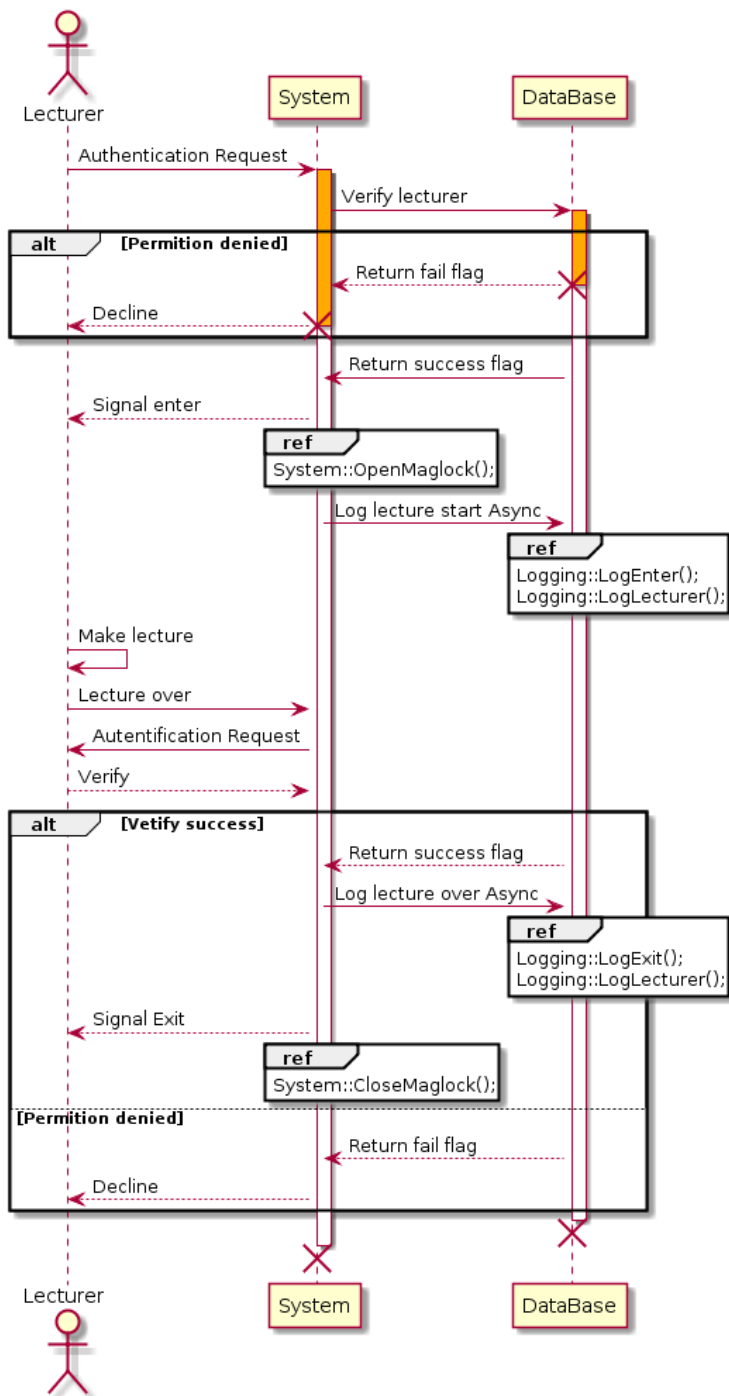
Root --> Supervisor : Report success

@enduml



3.3.6

```
actor Lecturer
Lecturer -> System : Authentication Request
activate System
activate System #ffaa00
System -> DataBase : Verify lecturer
alt Permission denied
activate DataBase
activate DataBase #ffaa00
DataBase --> System : Return fail flag
destroy DataBase
System --> Lecturer : Decline
destroy System
end
DataBase -> System : Return success flag
System --> Lecturer : Signal enter
ref over System : System::OpenMaglock();
System -> DataBase : Log lecture start Async
ref over DataBase : Logging::LogEnter();\nLogging::LogLecturer();
Lecturer -> Lecturer : Make lecture
Lecturer -> System : Lecture over
System -> Lecturer : Autentification Request
Lecturer --> System : Verify
alt Vetify success
DataBase --> System : Return success flag
System -> DataBase : Log lecture over Async
ref over DataBase : Logging::LogExit();\nLogging::LogLecturer();
System --> Lecturer : Signal Exit
ref over System : System::CloseMaglock();
else Permission denied
DataBase --> System : Return fail flag
System --> Lecturer : Decline
end
destroy DataBase
destroy System
```



3.3.7

@startuml

actor "Supervising member"

participant Supervisor

participant Administrator

participant System

"Supervising member" -> Supervisor: request

activate Supervisor

Supervisor -> "Supervising member": Authentication request

"Supervising member" --> Supervisor: verify

alt successful case

Supervisor -> Administrator: card request

activate Administrator

Administrator -> "Supervising member": Authentication request

"Supervising member" --> Administrator: verify

alt successful case

Administrator -> "Supervising member": give card

else verify failed

Administrator -> "Supervising member": verifying failed

deactivate Administrator

end

else verify failed

Supervisor -> "Supervising member": verifying failed

deactivate Supervisor

end

"Supervising member" -> System: door opening request

activate System

System -> "Supervising member": Authentication request

"Supervising member" --> System: verify

alt successful case

ref over System : System::logEnter();

System -> "Supervising member": give access

ref over System :logEquipment;

else verifying failed

System -> "Supervising member": verifying failed

end

"Supervising member" -> System: door closing request

System -> "Supervising member": Authentication request

"Supervising member" --> System: verify

alt successful case

ref over System : System::logExit();

System -> "Supervising member": give access

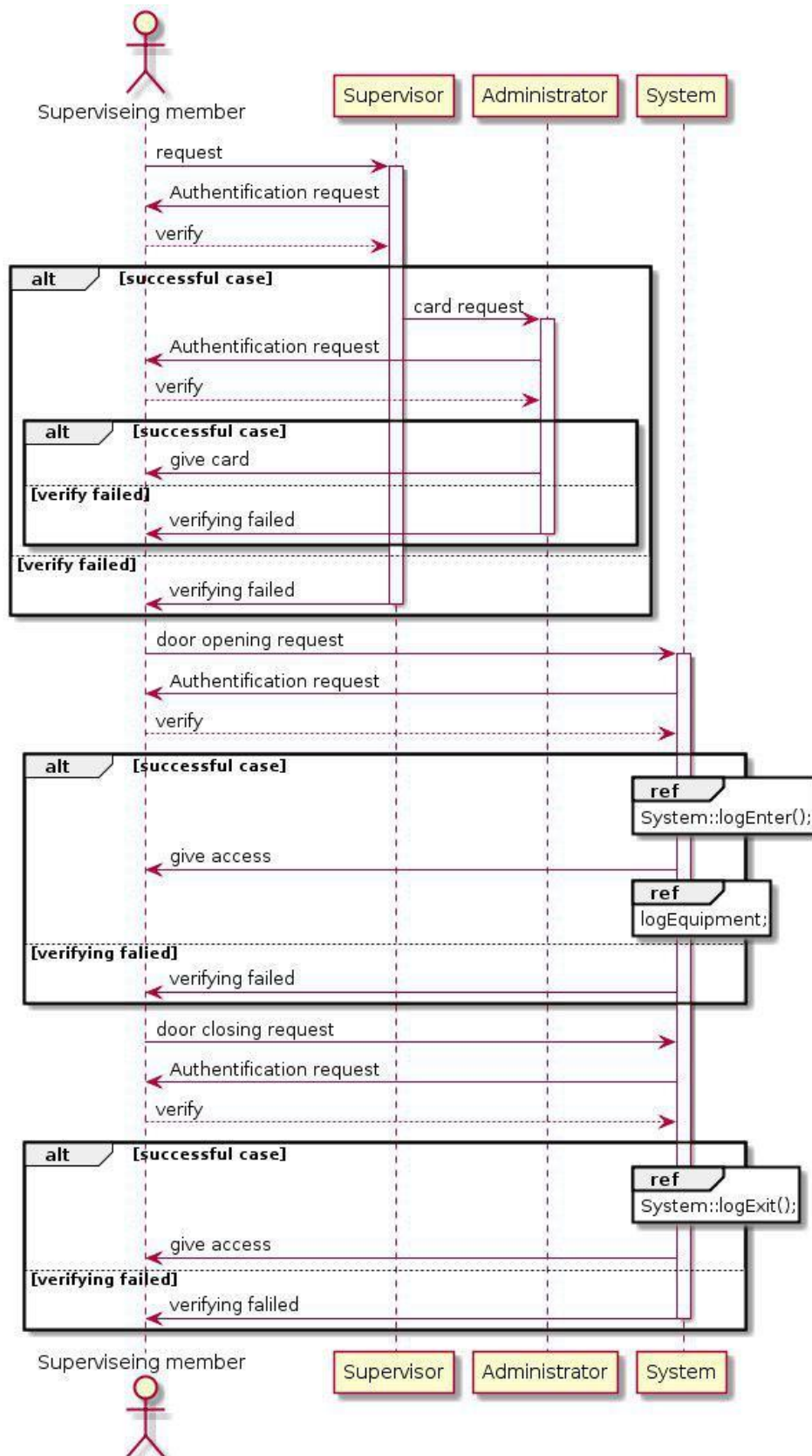
else verifying failed

System -> "Supervising member": verifying failed

deactivate System

end

@enduml



3.3.8

@startuml

actor Supervisor

Supervisor -> Supervisor : define auditorium/time

Supervisor -> Logging_System : request place/time

activate Logging_System

alt Place/time available

Logging_System --> Supervisor : requestCreated

Logging_System -> Tech_Help : place/time to receive tools

activate Tech_Help

Logging_System -> Security : place/time to receive tools

activate Security

ref over Tech_Help : Tech_Help::receive_tools()

Security -> Tech_Help : give access

Tech_Help --> Logging_System : tools received

destroy Tech_Help

Security --> Logging_System : access given

destroy Security

ref over Logging_System: System::log_tools_access()

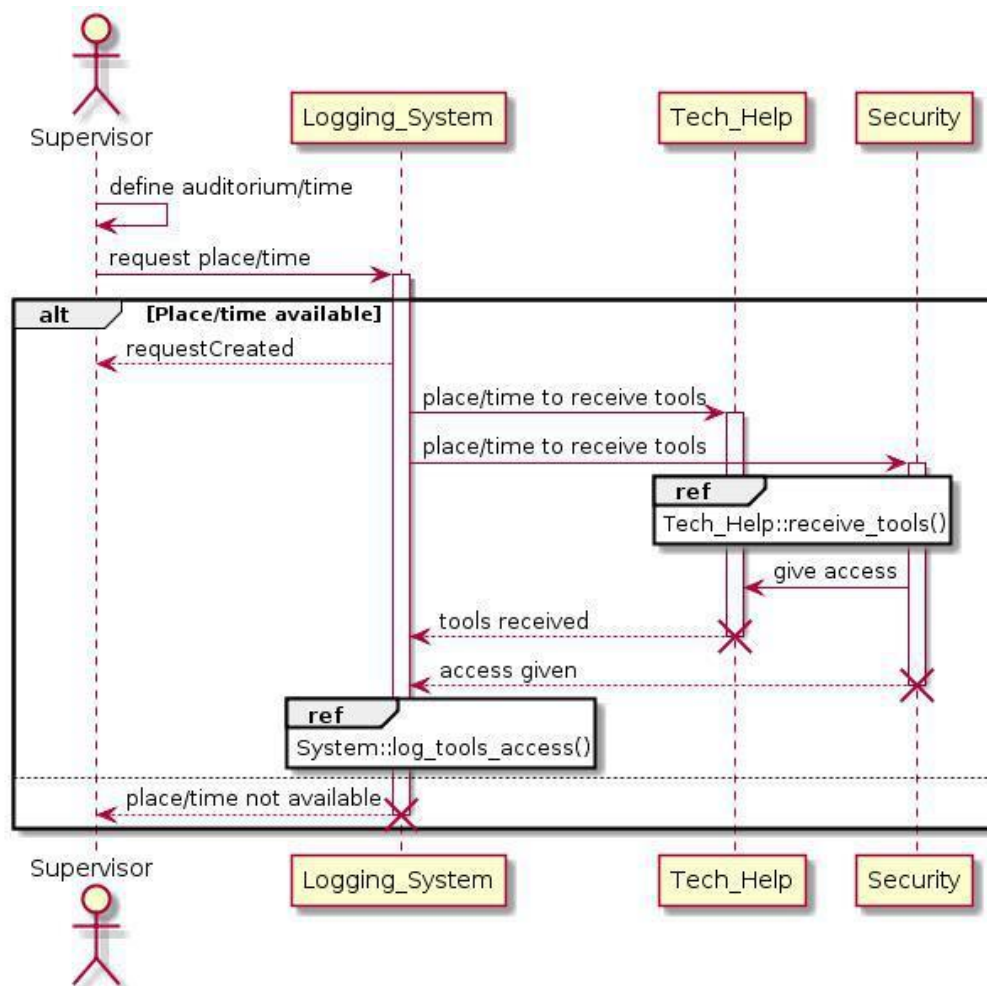
else

Logging_System --> Supervisor : place/time not available

destroy Logging_System

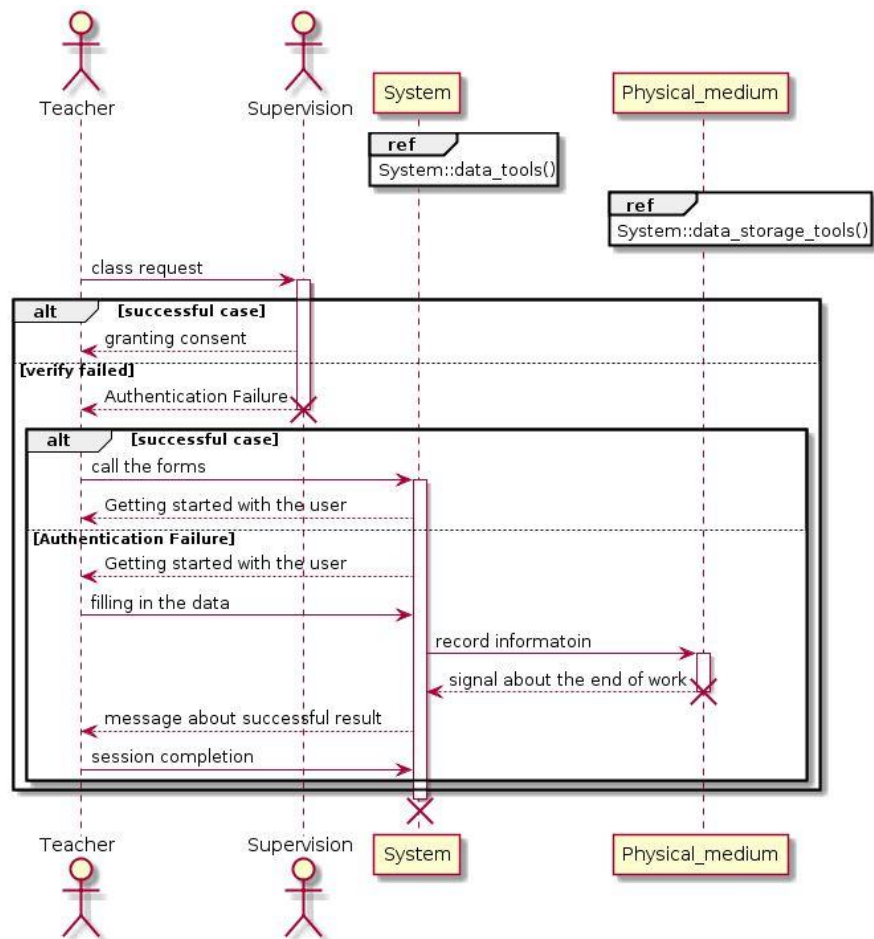
end

@enduml



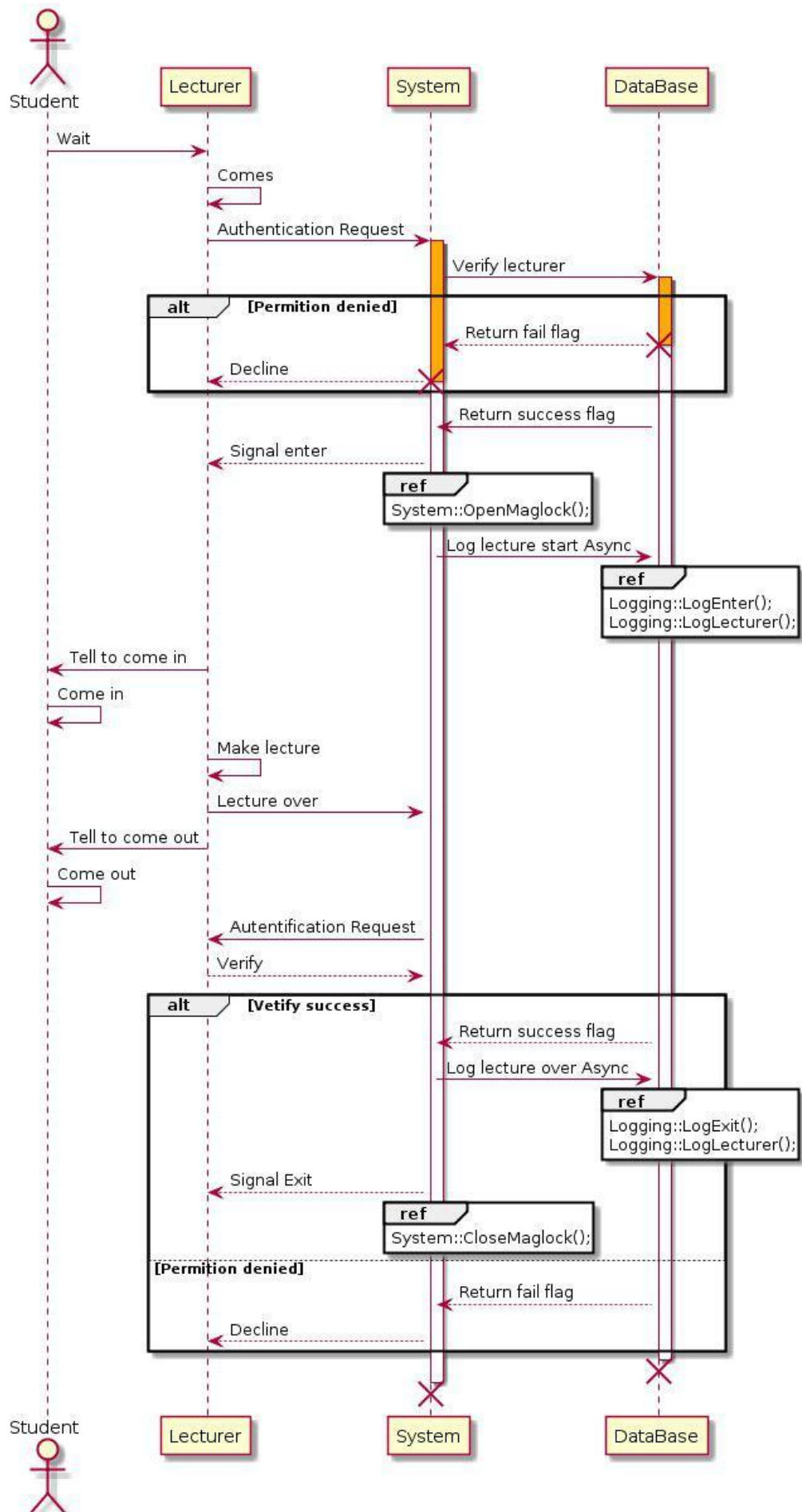
3.3.9

```
actor Teacher
actor Supervision
participant System
participant Physical_medium
ref over System
  System::data_tools()
end ref
ref over Physical_medium
  System::data_storage_tools()
end ref
Teacher -> Supervision: class request
activate Supervision
alt successful case
Supervision --> Teacher: granting consent
else verify failed
  Supervision --> Teacher: Authentication Failure
  destroy Supervision
alt successful case
  Teacher -> System: call the forms
  activate System
  System --> Teacher: Getting started with the user
  else Authentication Failure
  System --> Teacher: Getting started with the user
  Teacher -> System: filling in the data
  System -> Physical_medium: record information
  activate Physical_medium
  Physical_medium --> System: signal about the end of work
  destroy Physical_medium
  System --> Teacher: message about successful result
  Teacher -> System: session completion
  end
end
destroy System
```



3.3.10

```
actor Student
Student -> Lecturer : Wait
Lecturer -> Lecturer : Comes
Lecturer -> System : Authentication Request
activate System
activate System #ffaa00
System --> DataBase : Verify lecturer
alt Permission denied
activate DataBase
activate DataBase #ffaa00
DataBase --> System : Return fail flag
destroy DataBase
System --> Lecturer : Decline
destroy System
end
DataBase -> System : Return success flag
System --> Lecturer : Signal enter
ref over System : System::OpenMaglock();
System -> DataBase : Log lecture start Async
ref over DataBase : Logging::LogEnter();\nLogging::LogLecturer();
Lecturer -> Student : Tell to come in
Student -> Student : Come in
Lecturer -> Lecturer : Make lecture
Lecturer -> System : Lecture over
Lecturer -> Student : Tell to come out
Student -> Student : Come out
System -> Lecturer : Autentification Request
Lecturer --> System : Verify
alt Vetify success
DataBase --> System : Return success flag
System -> DataBase : Log lecture over Async
ref over DataBase : Logging::LogExit();\nLogging::LogLecturer();
System --> Lecturer : Signal Exit
ref over System : System::CloseMaglock();
else Permission denied
DataBase --> System : Return fail flag
System --> Lecturer : Decline
end
destroy DataBase
destroy System
```



3.3.11

@startuml

actor Tech_Help

Tech_Help -> Logging_System : receive tools/get access to auditorium

activate Logging_System

alt permission received

Logging_System -> Security : ask to give tools

activate Security

Security --> Logging_System : tools available

Logging_System --> Tech_Help : access given

Security -> Tech_Help : give tools

ref over Tech_Help : Tech_Help::repair_university()

Tech_Help -> Logging_System : alarm about tools return

Logging_System -> Security : alarm about tools return

Security --> Logging_System : will arrive to return tools

Logging_System --> Tech_Help : guards will return the tools

Security -> Tech_Help : return tools to storage

else

Security --> Logging_System : tools unavailable

destroy Security

Logging_System --> Tech_Help : access denied

destroy Logging_System

end

Tech_Help -> Logging_System : check out tools condition

activate Logging_System

alt permission received

Logging_System -> Security : ask to give tools

activate Security

Security --> Logging_System : tools available

Logging_System --> Tech_Help : access given

Security -> Tech_Help : give tools

ref over Tech_Help : Tech_Help::check_out_condition()

Tech_Help -> Logging_System : alarm about tools condition

ref over Logging_System : Logging_System::write_into_LogFile()

alt tools condition\nis good

Logging_System -> Supervisor : send a LogFile

activate Supervisor

Supervisor --> Logging_System : have read a LogFile

destroy Supervisor

Logging_System -> Security : alarm about tools return

Security --> Logging_System : will arrive to return tools

Logging_System --> Tech_Help : guards will return the tools

Security -> Tech_Help : return tools to storage

else

ref over Logging_System : Logging_System::tool_needs_repair()

activate Supervisor

Logging_System -> Supervisor : send information about a tool

Supervisor --> Logging_System : have received the information

ref over Supervisor : Supervisor::make_decision_about_tool()

Supervisor -> Logging_System : send a decision about a tool

destroy Supervisor

Logging_System -> Security : alarm about tools return

Security --> Logging_System : will arrive to return tools

Logging_System --> Tech_Help : guards will return the tools

Security -> Tech_Help : return tools to repair/utilize

end

else

Security --> Logging_System : tools unavailable


```

destroy Security
Logging_System --> Tech_Help : access denied
destroy Logging_System
end
@enduml

```

