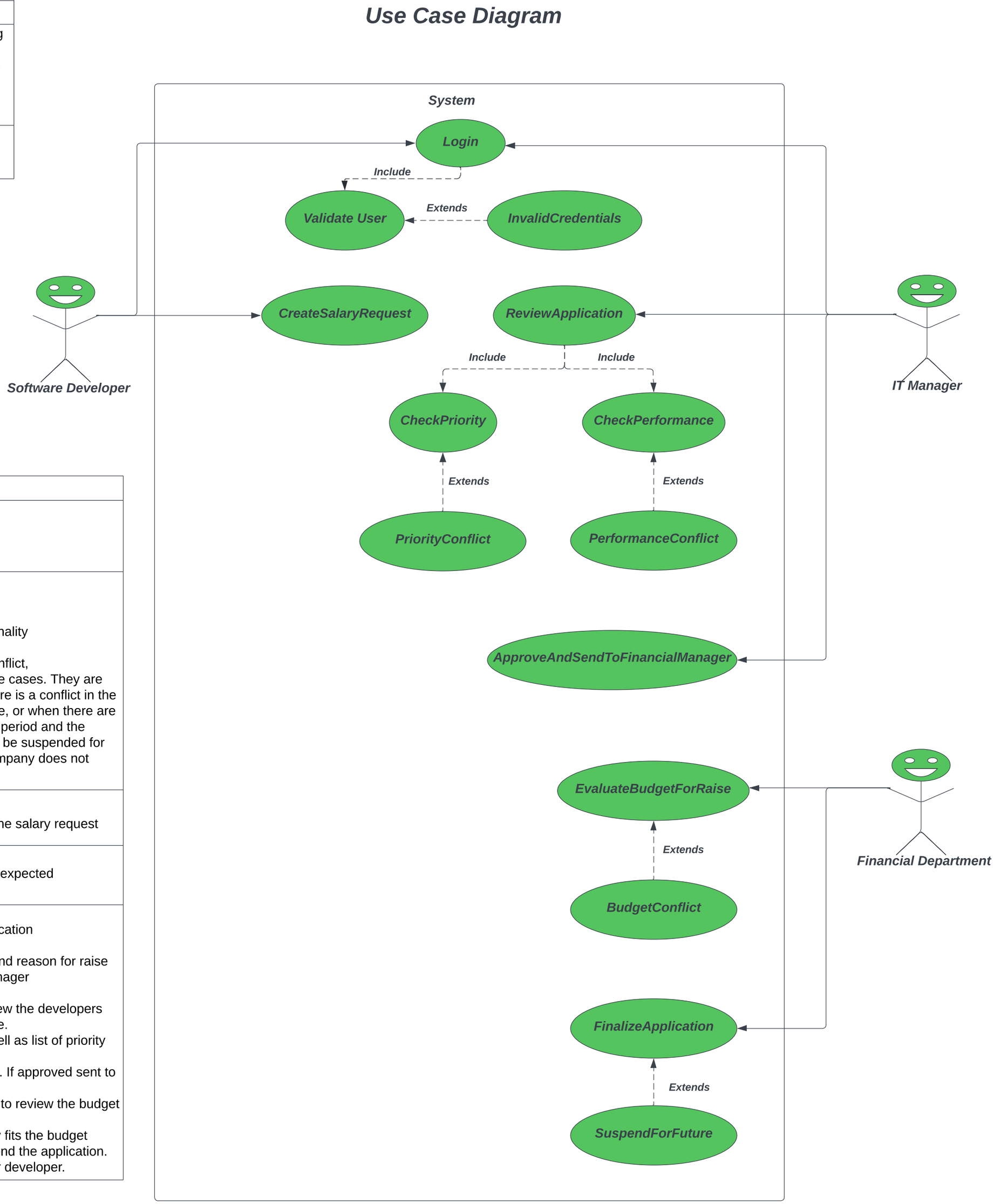
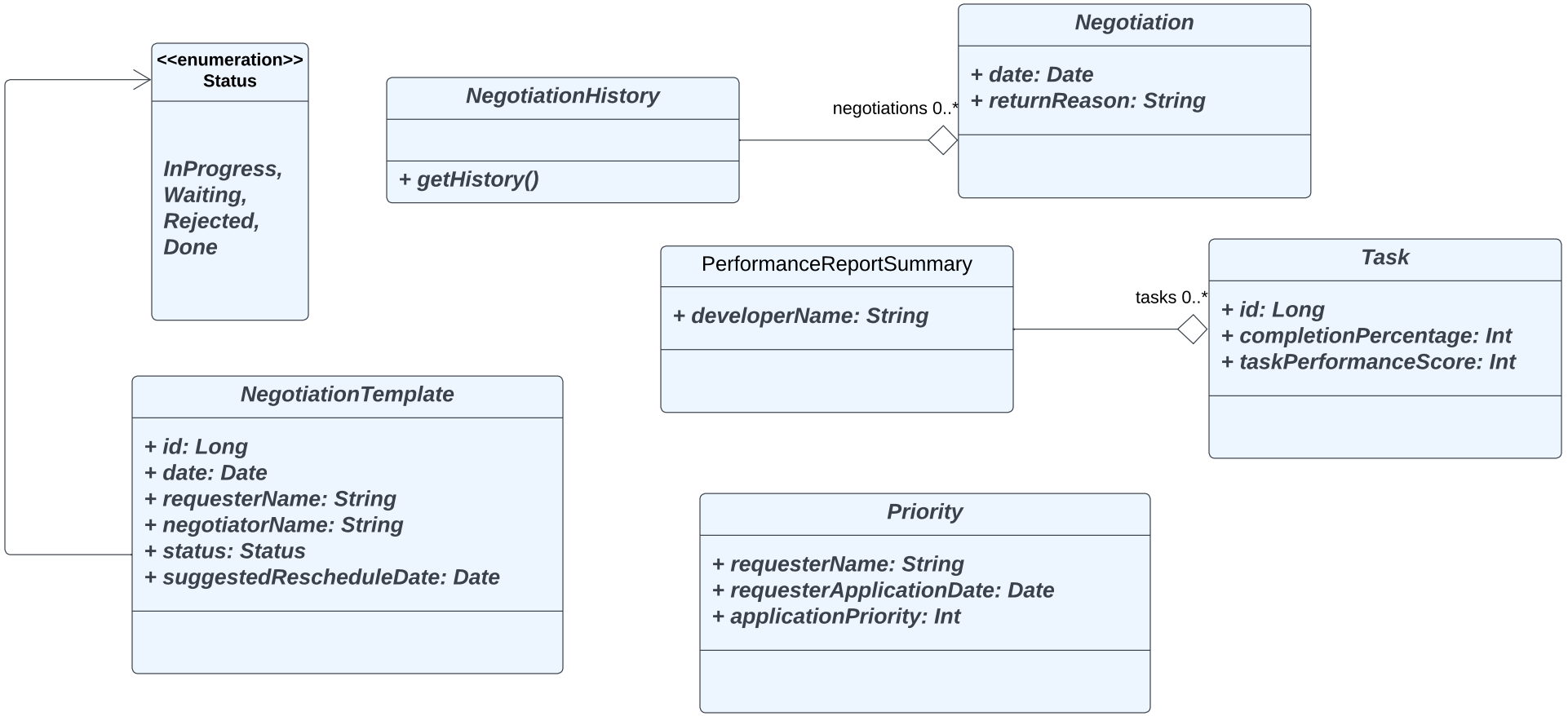
| Role | Responsibilities |
|---|---|
| Developer | a. Filing raise salary request |
| IT Manager | a. Reviewing, accepting or rejecting applications<br>b. Negotiating the application when possible<br>c. Sending application to financial manager |
| Financial Manager | a. Evaluating budget for requests<br>b. Approving or rejecting applications |

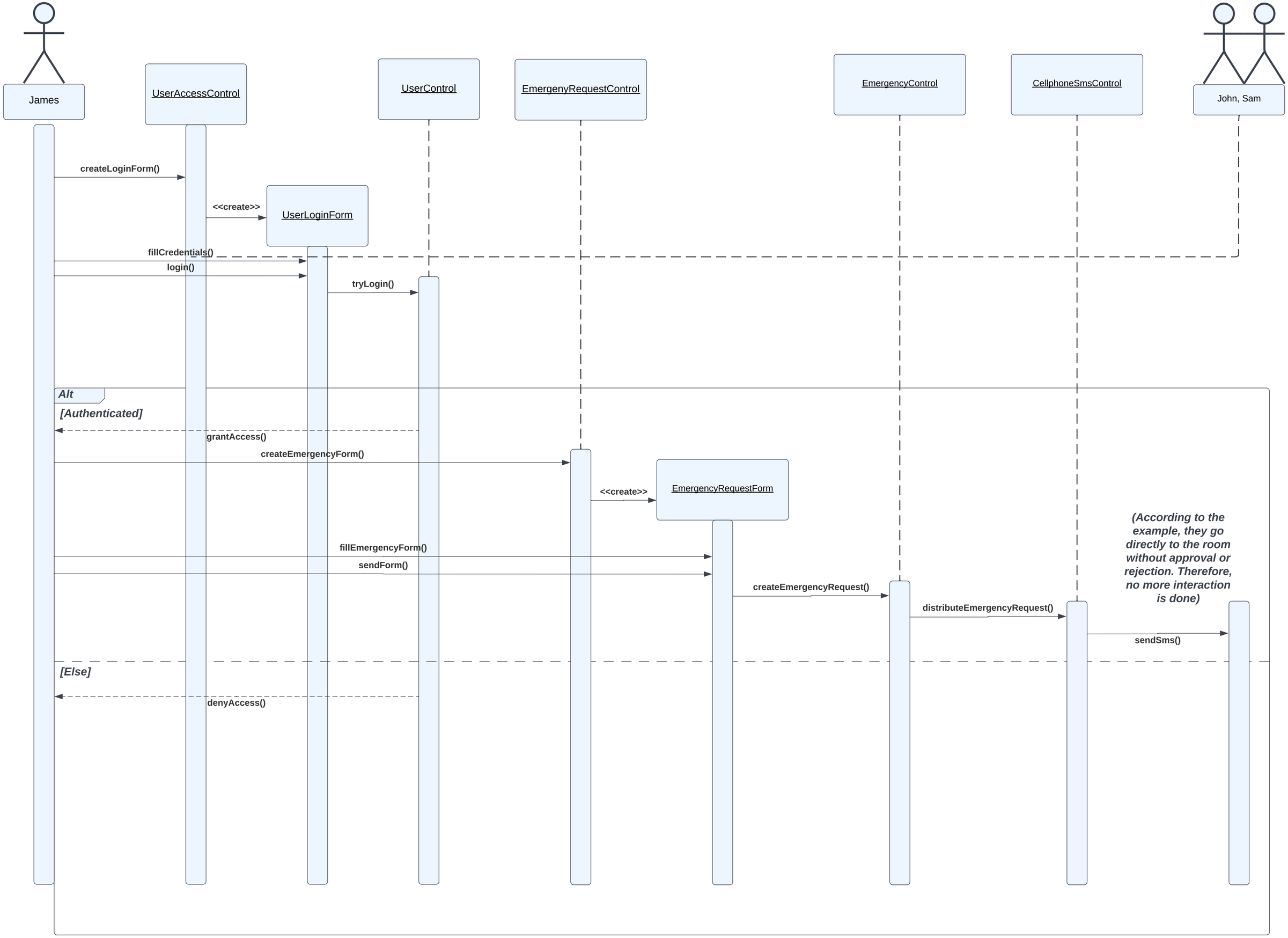
***Use cases:***
a. Login <DEV>
b. ValidateUser <System>
c. CreateSalaryRequest <DEV>
d. ReviewApplication <IT>
e. CheckPerformanceReports <IT>
f. CheckPriority <IT>
g. AcceptAndSendToFinancialManager <IT>
h. EvaluateBudgetForRaise <FINANCE>
i. FinalizeApplication <FINANCE>

***Exceptional use cases:***
1. InvalidCredentials
2. PerfomanceConflict
3. PriorityConflict
4. BudgetConflict
5. SuspendForFuture

| |
|---|
| **Name:** ManageSalaryRaiseRequests |
| **Participating actor(s):**<br>Developer (Initiator)<br>IT Manager<br>Financial Manager |
| **Entry Conditions:**<br>1. Developer logins to the system.<br>2. System validates the developer credentials.<br>3. Developer accesses the create vacation request functionality<br><br>This use case extends InvalidCredentials, PerformanceConflict, PrirorityConflict, BudgetConflict and SuspendForFuture use cases. They are initiated when the login credentials are invalid, or when there is a conflict in the performance of the developer for the requested salary raise, or when there are other developers who requested raised salary in the same period and the developer has no priority, or when the application needs to be suspended for the future when the developer deserves a raise but the company does not have enough money. |
| **Exit Condition:**<br>Financial Manager decides whether to approva or cancel the salary request |
| **Quality Conditions:**<br>The system should be available and functioning without unexpected interruptions |
| **Event Flow:**<br>1. Developer chooses to create a new salary request application<br>2. System displays the requested form<br>3. Developer fills the application with the required salary, and reason for raise<br>4. Developer chooses to send the application to the IT manager<br>5. System displays the new request to the IT manager<br>6. IT manager reviews the application and chooses to review the developers performance and other developers priority of getting a raise.<br>7. System displays the performance of the developer as well as list of priority among developers<br>8. IT Manager chooses to approve or reject the application. If approved sent to the Financial Manager<br>9. Financial manager reviews the application and chooses to review the budget for the possibility of a raise.<br>10. System displays the budget and if the requested salary fits the budget<br>11. Financial Manager chooses to approve, reject or suspend the application.<br>12. System displays the result of the request to the initiator developer. |




<<enumeration>>
Status
InProgress,
Waiting,
Rejected,
Done
NegotiationHistory
+ getHistory()
negotiations 0..*
Negotiation
+ date: Date
+ returnReason: String
PerformanceReportSummary
+ developerName: String
tasks 0..*
Task
+ id: Long
+ completionPercentage: Int
+ taskPerformanceScore: Int
NegotiationTemplate
+ id: Long
+ date: Date
+ requesterName: String
+ negotiatorName: String
+ status: Status
+ suggestedRescheduleDate: Date
Priority
+ requesterName: String
+ requesterApplicationDate: Date
+ applicationPriority: Int


James
UserAccessControl
UserControl
EmergenyRequestControl
EmergencyControl
CellphoneSmsControl
John, Sam
createLoginForm()
<<create>>
UserLoginForm
fillCredentials()
login()
tryLogin()
Alt
[Authenticated]
grantAccess()
createEmergencyForm()
<<create>>
EmergencyRequestForm
fillEmergencyForm()
sendForm()
createEmergencyRequest()
distributeEmergencyRequest()
(According to the example, they go directly to the room without approval or rejection. Therefore, no more interaction is done)
sendSms()
[Else]
denyAccess()

Online ordering of materials if not available locally

Employee

Online System

Delivery Department

Payment gateway

Valid Credentials

Select "Online Order" Service

Login

Choose required category of equipment

Invalid Credentials

Repeat process or click "Forgot Password"

Enter needed amounts

Send order

Receive info and get other product (?)

Receive payment request

Send payment

Receive an equipment order

Check items availability

Special case a.

Not available

Available

Inform employee

Send order to delivery

Send order payment info

Online system begär betalning?

Receive order

Receive order payment info

Send payment request

Receive payment

Inform payment completed

Receive payment completed

Retrieve address

Deliver order

Successful delivery

Failed delivery

Try again

Receive package

Verify package

Correct contents

Wrong contents

Close application

Return package