# ID2207 - Modern Methods in Software Engineering

# Tutorial 1

September 2025

# Tutorial 1

## Problem Description:

In an IT company, software developers usually must manage their vacations with their direct manager. There might be possible schedule conflicts for important project deliveries and priorities among the colleagues. The company would like to automate the communication between the developers and their managers through an internal system that can be used to manage the communication workflow. The system should be able to retrieve schedules of all projects deadlines and order the vacation requests according to date of request and priorities. Figure 1 illustrates the usual communication process.
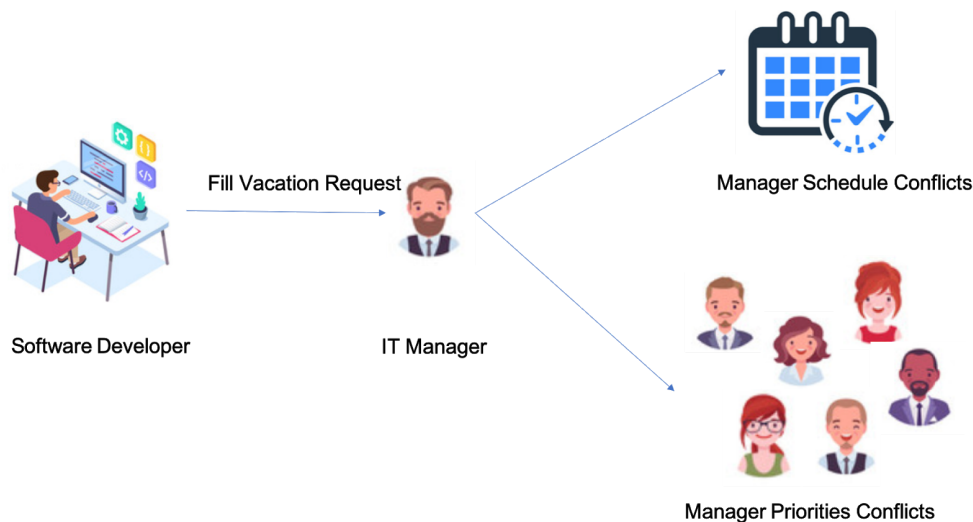


**Figure 1:** Manage vacation requests

Another functionality required for this system is to manage the salary raise requests. Developers can send a salary raise request to the manager who checks the performance reports of the developers and the priorities of giving raises among the colleagues. Then the manager sends this request to the Financial Manager who will be responsible for evaluating if the budget allows for a raise to this developer at the current time. Figure 2 illustrates the described functionality.
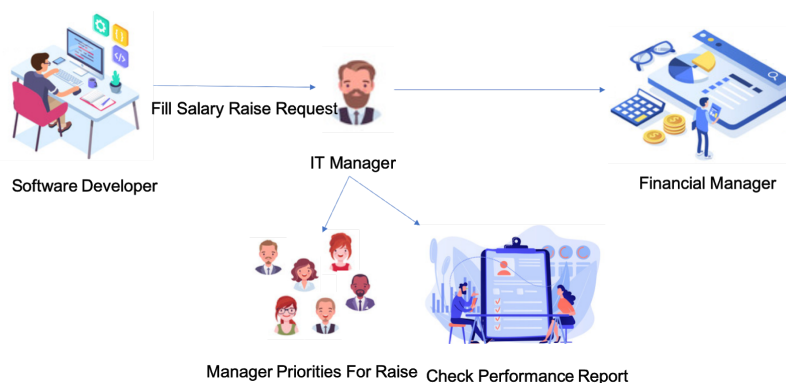


**Figure 2**: Manage Salary Raise requests

**Question 1:** Use Case Diagrams

*Use case diagrams provide a good way of getting an overall picture of what is happening in the existing system or is planned to happen in the new system. The use case diagram is very simple, and it is an effective means of communicating with users and other stakeholders about the system and what it is intended to do. Use case diagrams show use cases and actors and the associations among them. Use cases represent sequences of actions carried out by the system, and actors represent the people or other systems that interact with the system being modelled. Remember that use cases don't specify the detail of how the behavior is carried out. The details can be elaborated using other models (for example: sequence diagrams).*

Write a use case description and draw a use case diagram for the "Vacation Request" case, taking into consideration the following details:
- ❏ Both software developer and the manager should be registered in the new system and should be able to login and access the system. Ignore the registration use case for this example.
- ❏ The application can be approved or rejected. In some cases, it can be suspended for further negotiation.

**<span style="color:darkred">Solution</span>**:
Identified roles in this business are:

| Role | Responsibilities |
|------|------------------|
| Developer | a. Filling vacation requests with required details |
| *Fill ? ……..* | *Fill ?………..* |

Identified use cases in this business:
  a. Login
  b. ValidateUser
  c. CreateVacationRequest
  d. *Fill ?…..*
  e. *Fill ?…..*
  f. *Fill ?…..*
  g. *Fill ?…...*

Some exceptional use cases:
  a. InvalidCredentials
  b. ScheduleConflict

c. *Fill ?…...*
d. *Fill ?…...*

**Use case description:**

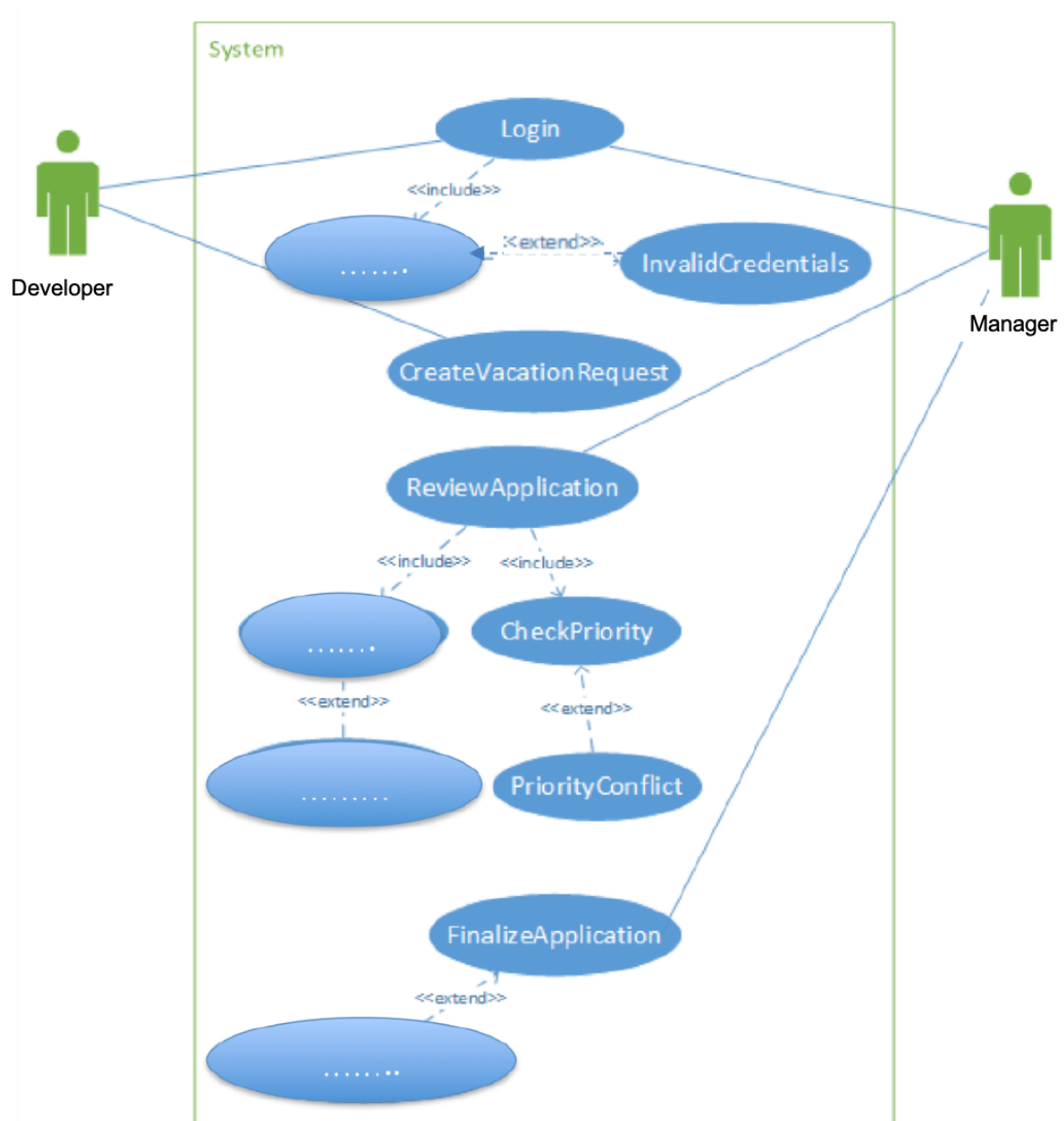| |
|---|
| **Name:** ManageVacationRequests |
| **Participating actor(s):**<br>*Fill ?……*<br>*Fill ?…...* |
| **Entry Conditions:**<br>1. Developer logins to the system.<br>2. System validates the developer credentials.<br>3. Developer accesses the create vacation request functionality |
| **Exit Condition:**<br>Developer and manager have agreed whether to approve, cancel or negotiate the vacation request. |
| **Quality Conditions:**<br>The system should be available and functioning without unexpected interruptions. |
| **Event Flow:**<br>*(Hint) How does the use case start ?*<br>    1. Developer chooses to create a new vacation request application.<br>    2. System displays the requested form.<br>*(Hint) What does the developer have to fill in the application ?*<br>    3. …….<br>    4. …….<br>*(Hint) How does the manager get informed about the application ?*<br>    5. System displays the new request to the manager.<br>    6. Manager reviews the application and chooses to review the schedule.<br>    7. System displays the schedule to the manager.<br>    8. Manager chooses to review the vacation priorities for the other developers.<br>*(Hint) What is the action that should be taken by the manager ?*<br>    9. ……<br>    10. …...<br>*How does the developer know the final decision ?*<br>    11. System displays the result of the request to the initiator developer. |

## Use Case Diagram: Fill in the blanks



**System**

Developer

Manager

Login

<<include>>

.......  <<extend>>  InvalidCredentials

CreateVacationRequest

ReviewApplication

<<include>>  <<include>>

......  CheckPriority

<<extend>>  <<extend>>

.........  PriorityConflict

FinalizeApplication
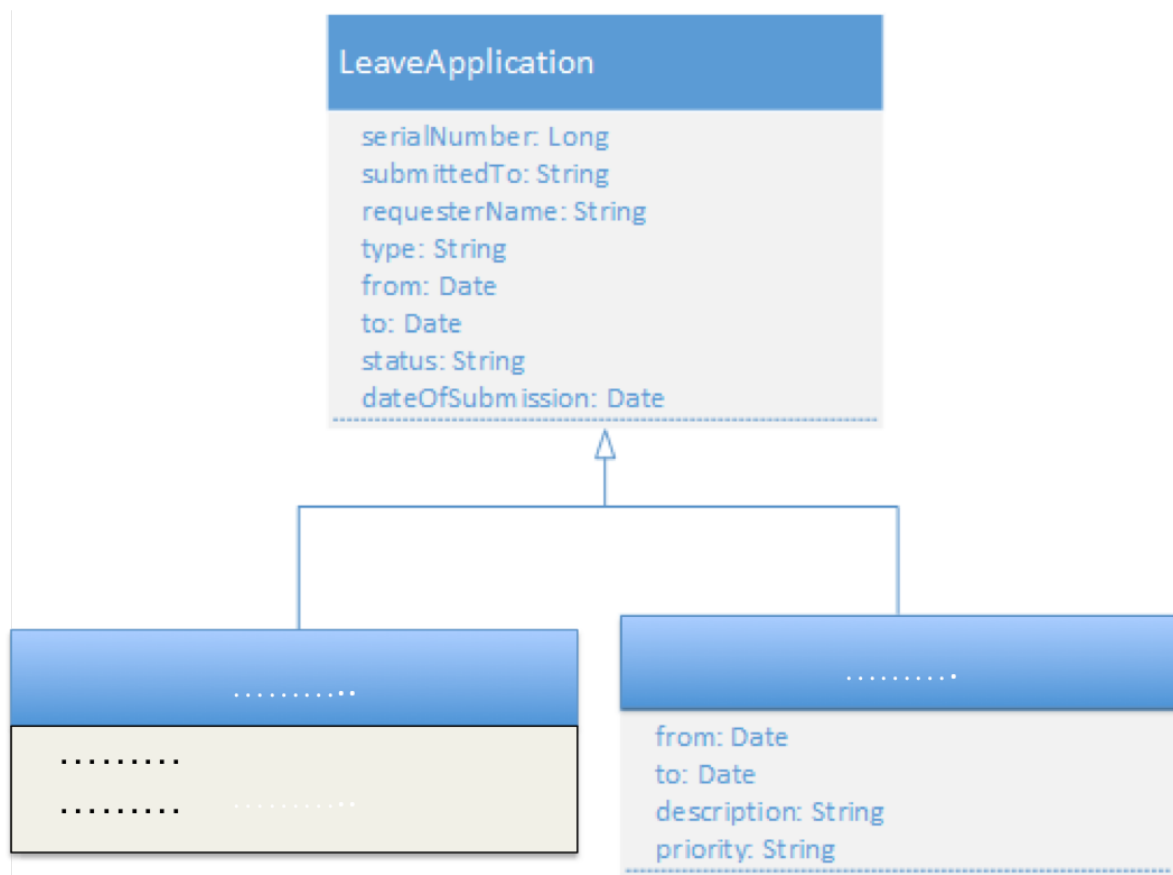
<<extend>>

.......

**Question Two:** Class Diagrams

*Class diagrams show the classes that make up the system with their attributes and operations. The collaborations and interactions among these classes are shown as relationships between these classes. Non-Technical participants in the software development process are less likely to understand the more detailed concepts that are shown in class diagrams.*

The developed system has a request which might be a vacation or travel request. Each request has a serial number, name of requester, name of the person to whom it is submitted, request type (vacation or travel request), and date of submission. Vacation request has other specific fields such as: vacation reason, priority, duration of vacation (From - To). Travel request has destination, travel reason, (From -To), and information whether it is paid by the inviting organization or not.
Each request has a status (initiated, approved, rejected, under negotiation).

Draw a class diagram that presents the applications in the system. Fill in the blanks

**LeaveApplication**

serialNumber: Long
submittedTo: String
requesterName: String
type: String
from: Date
to: Date
status: String
dateOfSubmission: Date

△

Left class:
..........
..........
..........
..........

Right class:
..........
from: Date
to: Date
description: String
priority: String

**Question Three:** Sequence Diagrams

*While class diagrams model the static structure of the system, the sequence diagrams, as part of the interaction diagrams, model the dynamic aspects of the system. asks are performed by objects interacting with each other by passing messages. Sequence diagrams are used to show these interactions and emphasize the order of the messages over time.*

Complete the sequence diagram for the following scenario of the "**Assign doctor to a surgery**" use case:

| scenario name | AssignDoctorToASurgery |
|---|---|
| **Participating actor instances** | Alfred: Doctor<br>Simon: HeadOfUnit |
| **Flow of events** | 1. Simon logins to the system and chooses to create a new surgery assignment request.<br>2. Simon selects "Alfred" to be assigned to the surgery, and he fills the other required details in the assignment form and chooses to send the application.<br>3. System displays the new task in Alfred tasks list.<br>4. Alfred checks if he can be available for the surgery and approves the application if it is okay. Otherwise, Alfred rejects the assignment with description of the reasons.<br>5. System updates the status of the form to be approved or rejected and sends the update to Simon.<br>6. Simon opens the application and checks Alfred feedback.<br>7. Simon closes the application. |
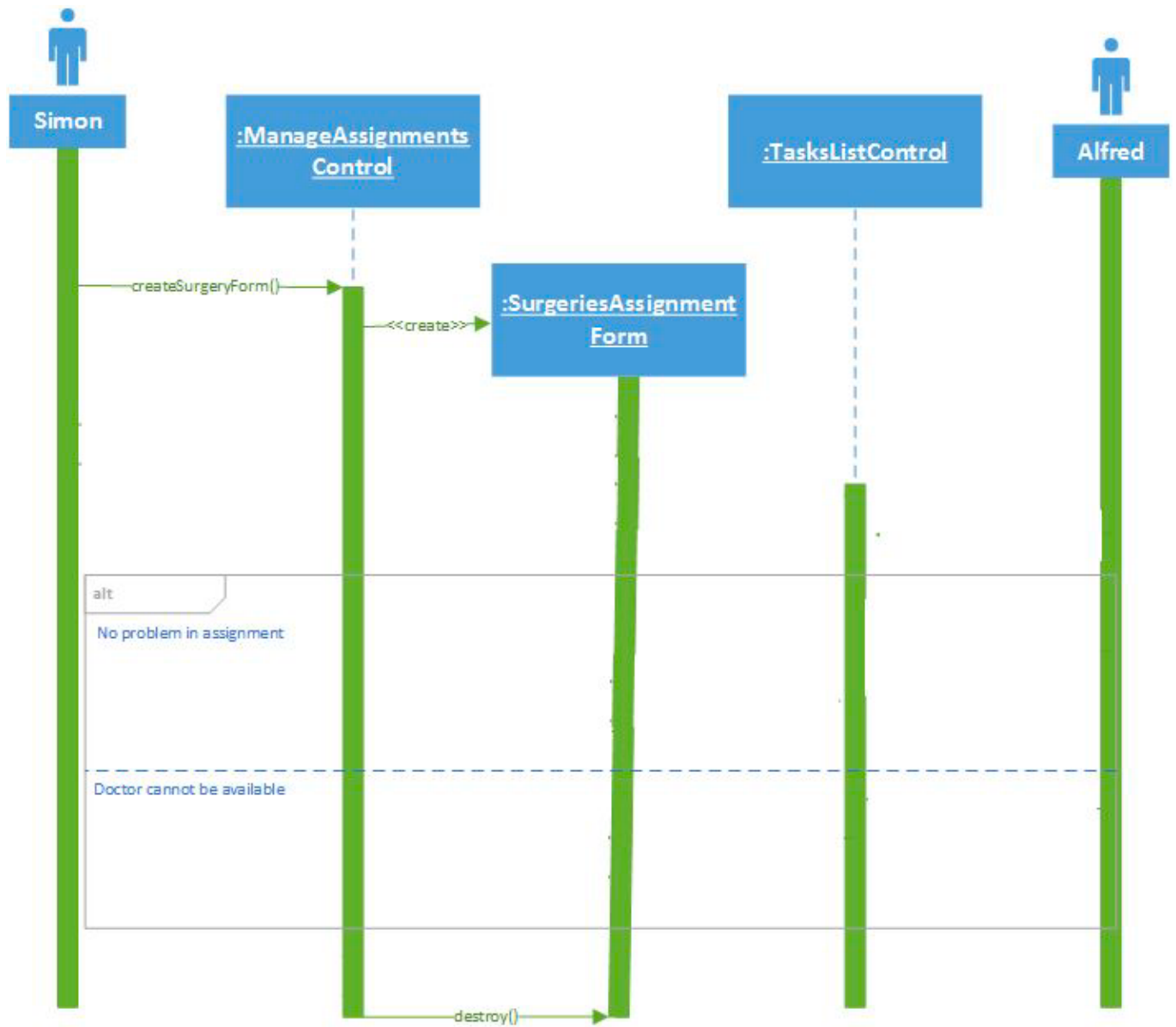
**Question Four:** Activity Diagrams

*Activity diagrams are a means of describing workflows. They can used as an analysis tool to describe business flows in varying levels of detail. They can be also used to describe complex flows within or between use cases. At the design phase, they can be used to describe in detail the flow within an operation. So, they can be used before the identification of use cases in the determination of high-level business requirements, or as a means of describing use cases or complex behaviour within an object.*

Our hospital has a main branch which has more medical materials in its local store. Sometimes, when there is a need to request some resources, a specific employee uses the system to request items from the main branch. If the items are available, the employee sends an order to the delivery department to deliver the package to its branch. Otherwise, they request the items online, and change the status of the request to be in the waiting list (Pending). In this part, draw an activity diagram that visualizes the described procedures. Use parallel flows where possible.

**Study the diagram and discuss it with your colleagues**
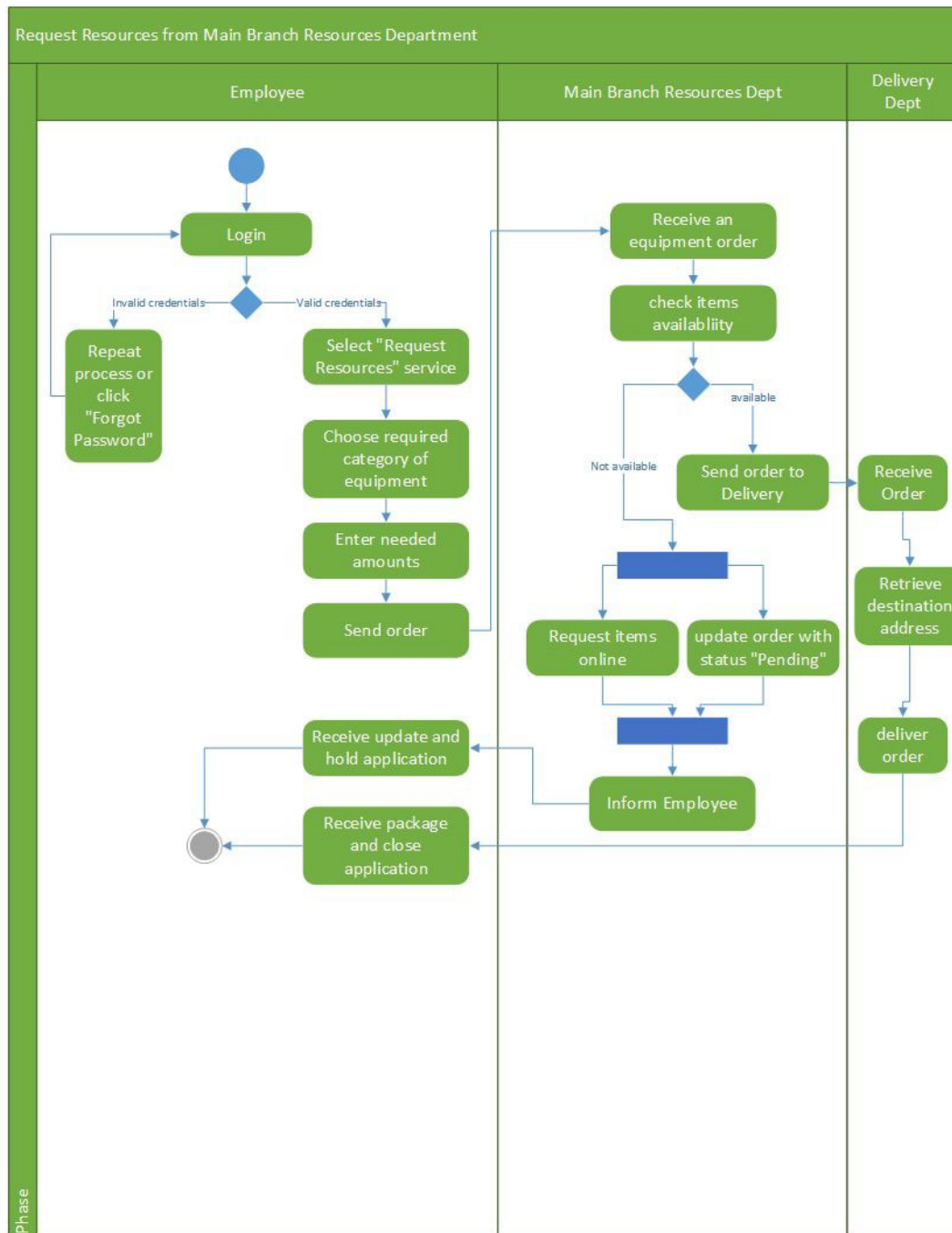
**The business of this tutorial is not based on real case, so use your imagination in the solutions to develop your skills in UML !**