Behavior Modelling

Unified Modelling Language (UML), Use Case Diagrams and Descriptions

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://about.softuni.bg

You Have Questions?

Software University

sli.do

#IT-BA

facebook.com

/groups/ITBusinessAnalysisAugust2023/

Table of Contents

Introduction to Unified Modelling Language (UML)

Car Assembly Line Example

Software University

STAMPING 1

MOULD WIPING

BODYSHOP 2

ROBOT PROTECTION

SOLVENT WIPING

3 CATHAPHORESIS

BACKING PADS

TACKRAGS

4 MASTIC APPLICATION

ROBOT PROTECTION

DRY WIPING

PAINT BOOTH 5

INSPECTION AREA 6

ASSEMBLY 7

ROBOT MAINTENANCE

ROBOT PROTECTION

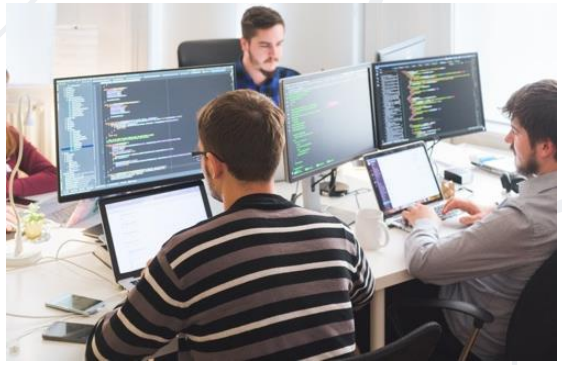
LAMP INSPECTION

POLISHING

PRE-TREATMENT WIPING

POLISHING

Modelling?



Modellers: They describe the world as they see it – either the world as is, whether it's a system, a domain, an application or a world they imagine to come

Designers: They explore possible solutions, to compare, to trade off different aspects, or to communicate approaches to garner criticism

Implementers: They construct solutions using models as part of (or as the entire) implementation approach

What is Unified Modelling Language?

Unified Modelling Language (UML) a rich language to **model software solutions, application structures, system behavior** and **business processes**

- 2 Categories
- 14 Official types

UML Categories

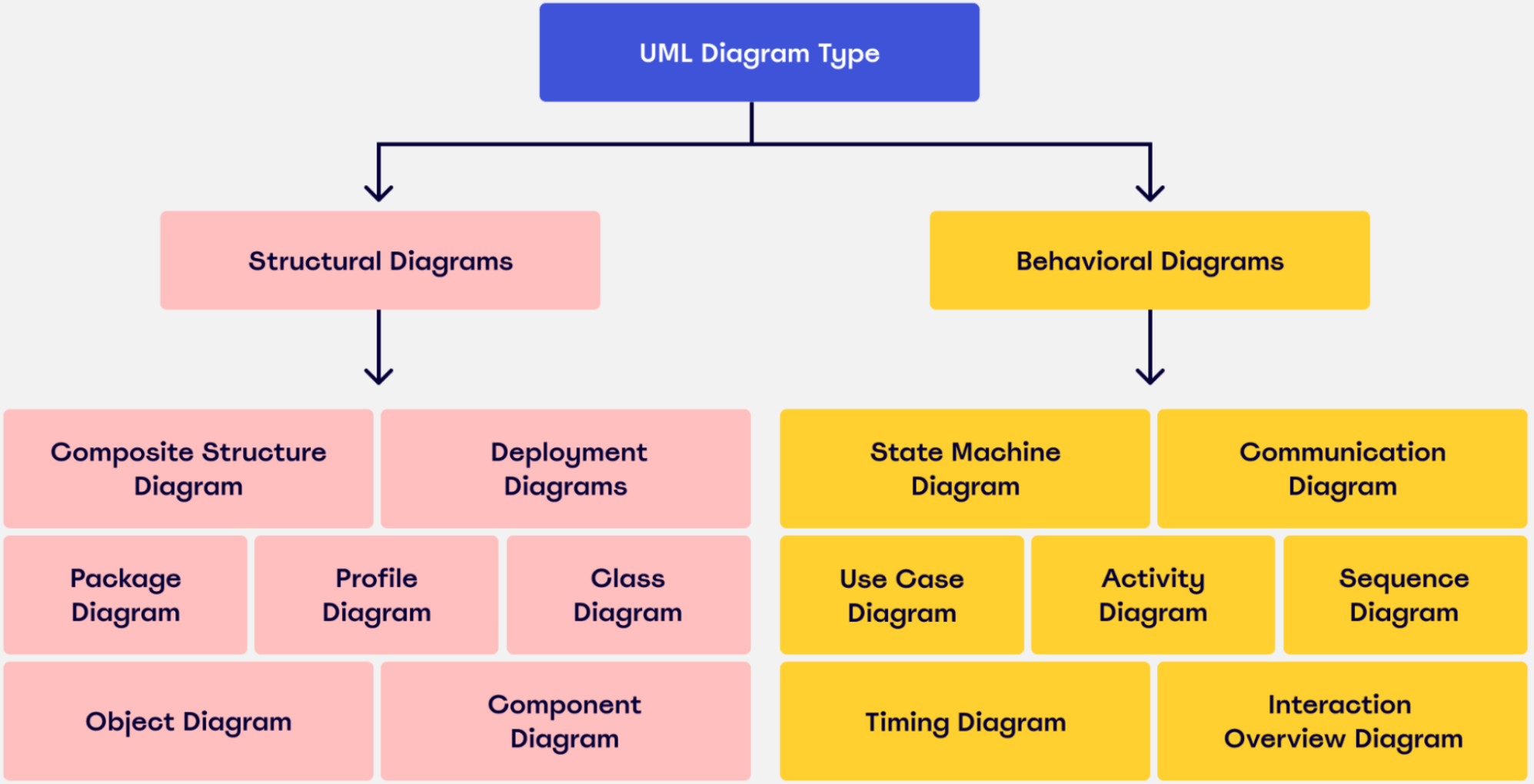
Software University

Structural Diagrams

Behavioral Diagrams

UML Diagram Types

Software University



BPMN vs UML

	BPMN	UML
Domain	Primarily used for modeling business processes and workflow	Widely used for software systems modeling and design
Purpose	Describes business processes, activities, and their flow	Captures software system structure, behavior, and interactions
Focus	Emphasizes the sequence of activities and process flow	Focuses on software system components, interactions, and logic
Main Users	Business analysts, process owners, and stakeholders	Software developers, system architects, and designers

Why do We Need UML?

Software University

- Gather system requirements.
- Get an outside view of system.
- Identify internal and external factors influencing the system.
- Show interaction among the requirements.

Introduction to Behavioural Modelling

What is Behavioural Modelling?

Behavioural modelling describes the **required behaviour** of an **existing or planned system** necessary to **interact successfully** with the environment

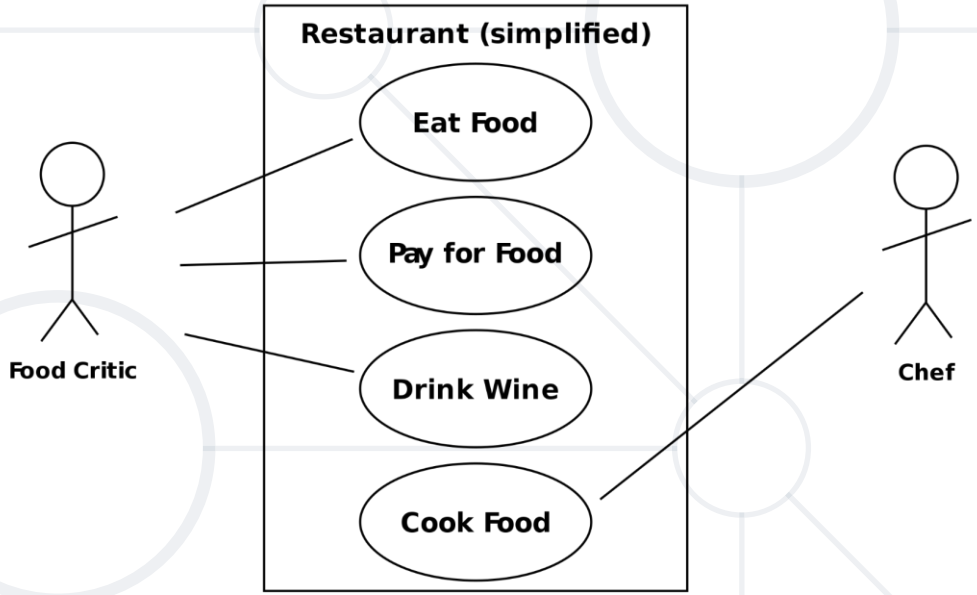
- Shows how a system responds to requests or evolves over time
- Visualizes how a system interacts with itself, users, and other systems

Introduction to Use Cases

Use Case Diagram

A **Use Case** is a methodology used in system analysis to **identify, clarify and organize system requirements**.

The **Use Case** is made up of a set of possible sequences of **interactions between systems and users in a particular environment** and **related to a particular goal**.

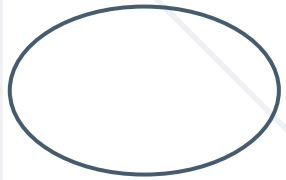


Components of Use Case Diagram (1)

- **System**
  - A rectangle is used to **depict the system boundary** indicating the scope of the system. Anything within the box represents functionality that is in scope and anything outside the box is not

- **Actor**
  - Stick figures are used to represent an actor. An actor is a **person**, **organization**, or **external system** that plays a role in one or more interactions with your system

- **Use Case**
  - A use case is drawn as an oval. It is the **behavior/process** that the **system offers to the actors** to help meet the actors' goals



Components of Use Case Diagram (2)

- **Association**
  - Associations **between actors and use cases** are indicated in use case diagrams by solid lines

- **Dependency**
  - Dependencies are dotted arrows that represent the **relationship between actors and use case**
    - include = common behaviour
    - extend = optional behaviour

Why is Use Case Modelling important?

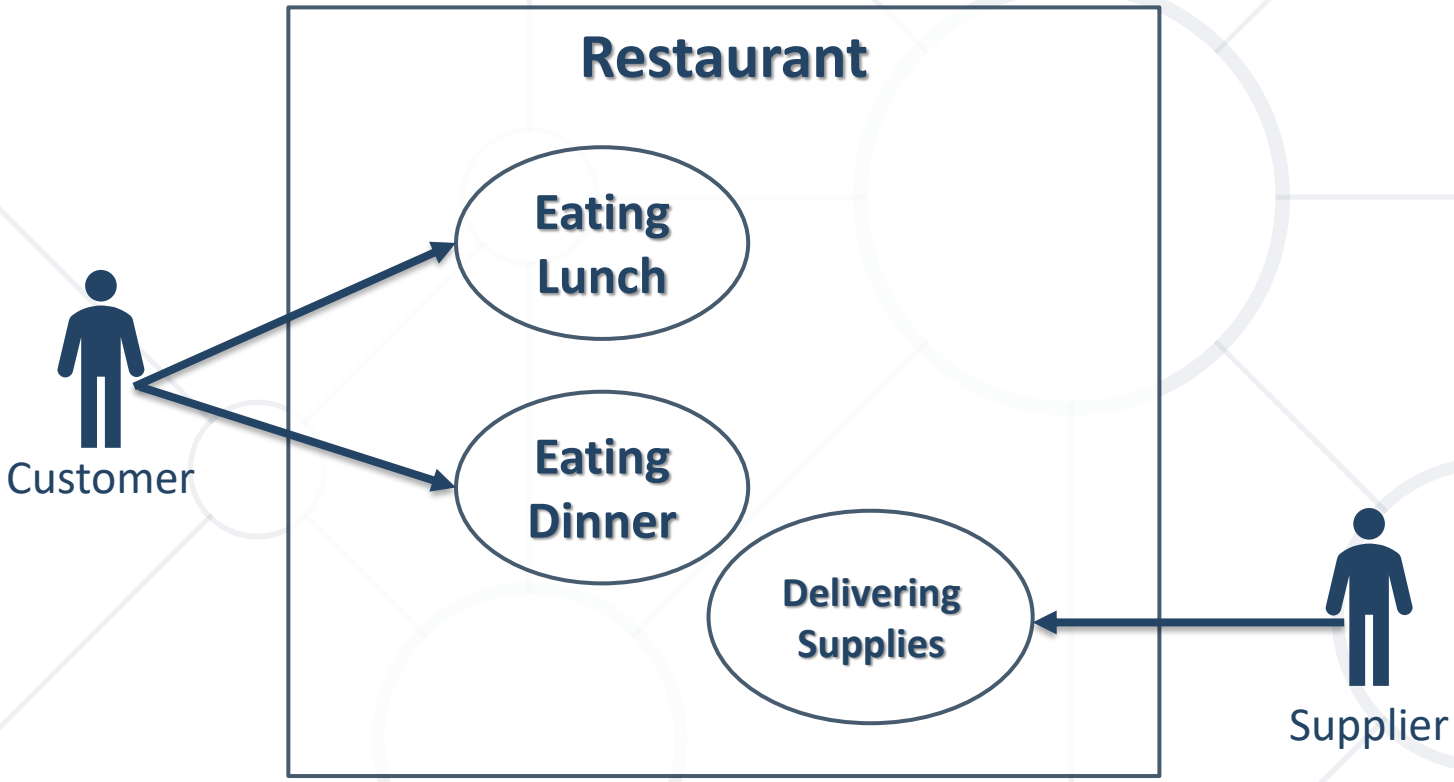
- **Functional requirements** need to be captured
  - Use case modeling enables a **standardized and systematic approach**
- Defines the **system boundaries**
- Encourages dialogue between **client and system designers**
- Enables verification of **coverage, accuracy, and sufficiency of detail**

Use Case

What activities can occur at a restaurant with a bar?

Think about it in 5 mins.

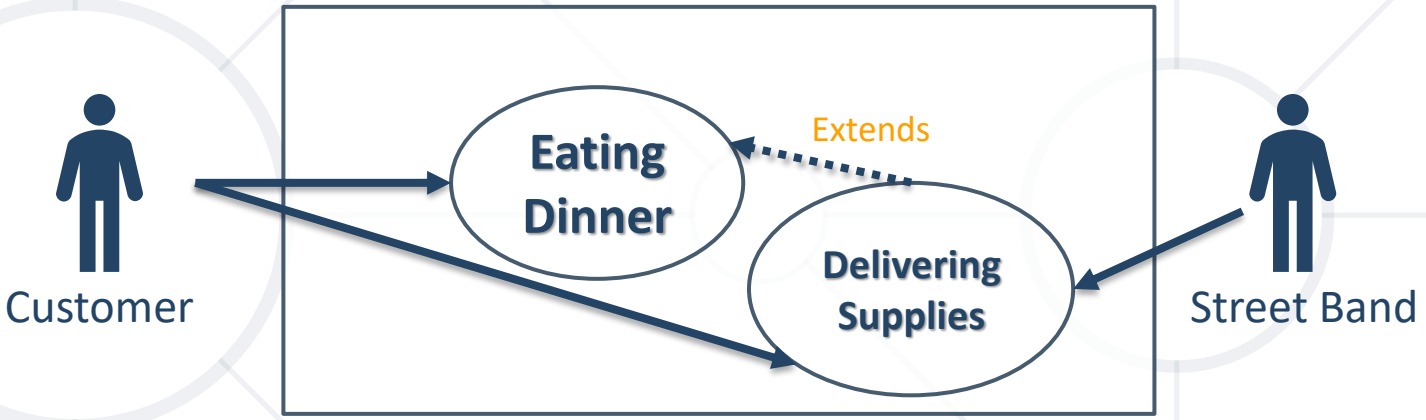
Use Case Diagram Example 1



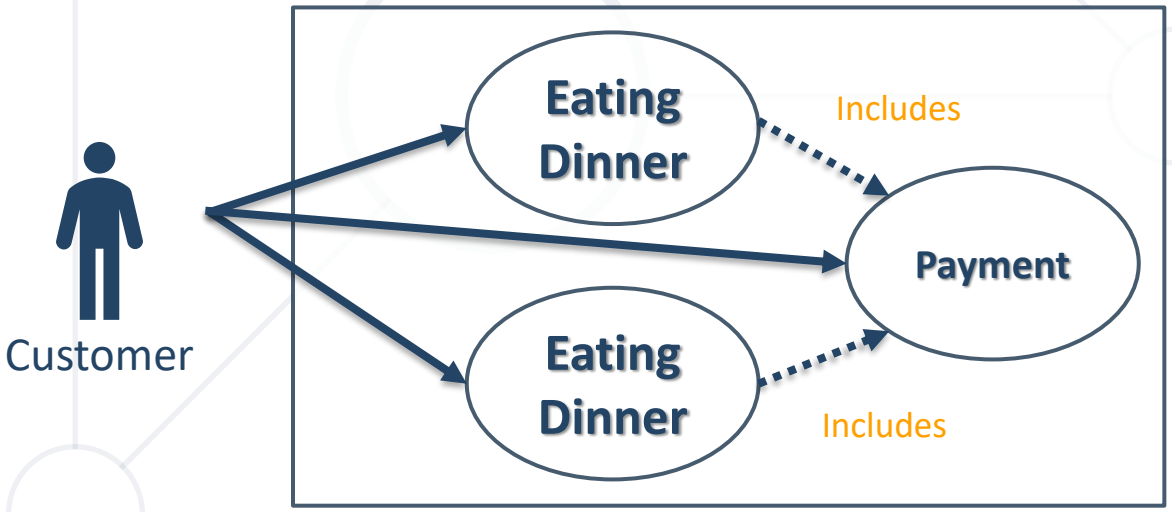
Note that we don't have such actors as Chef or Waiter. They are not external roles but are in fact part of the business we are modeling (the Restaurant), thus they are not actors.

Use Case Diagram Example 2

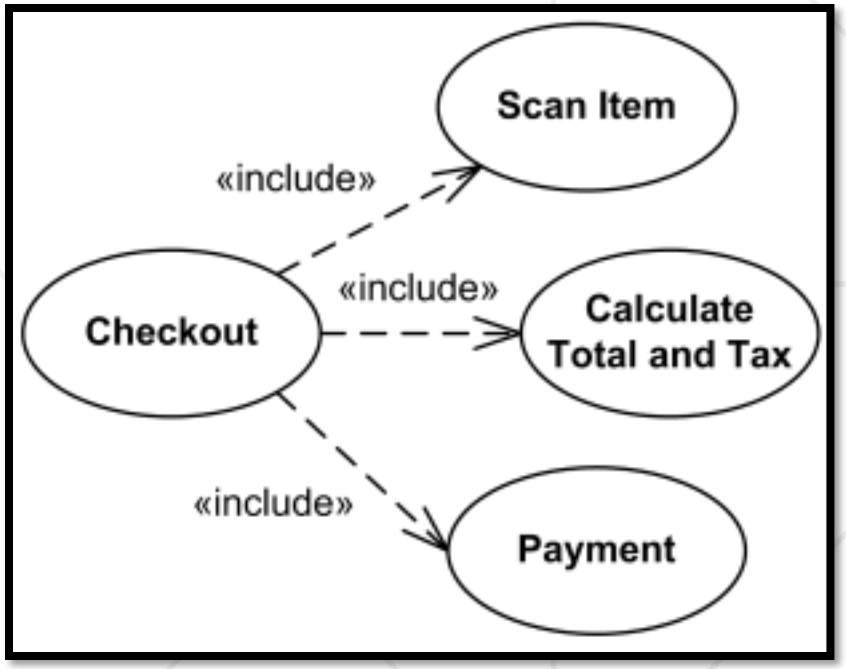
Extends: Used to present **alternative** flows. Indicates additional releases of a use case.

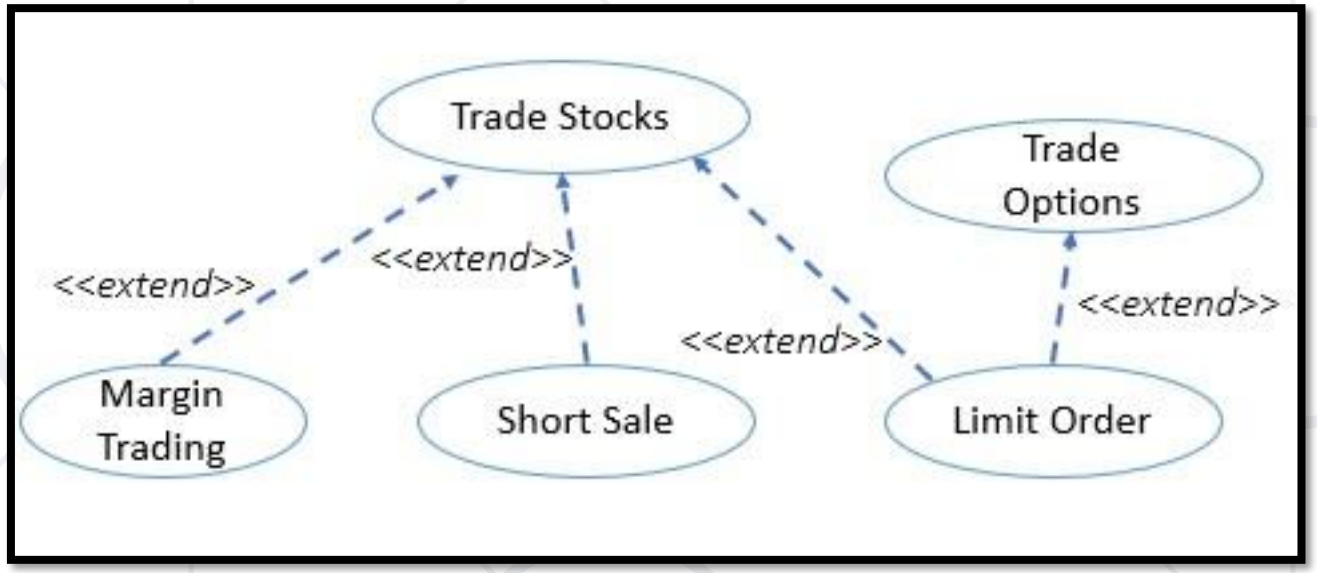


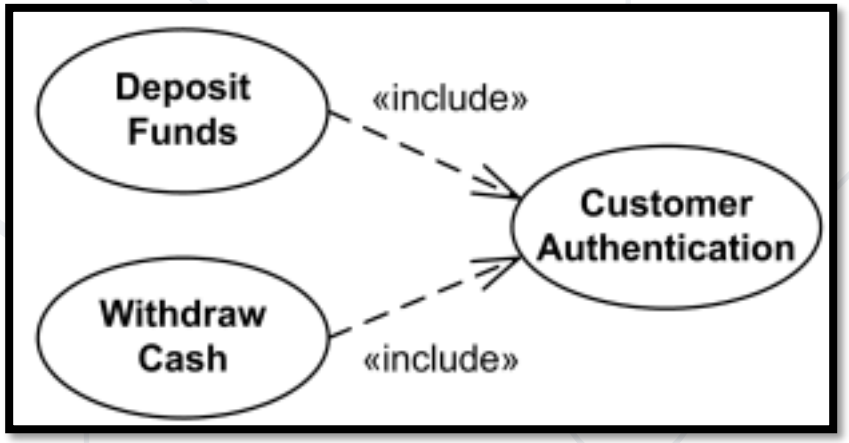
Includes: Representation of **common behavior.** Indicates use cases that are a necessary part of the interaction.

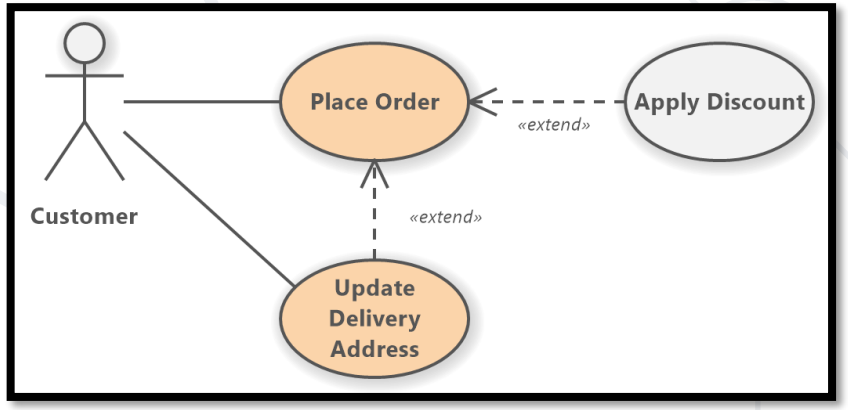


Use Case Example 3









Description of a Use Case

Use Case: "Eating dinner"

Goal: Satisfy hunger and have a nice time

Sequence:

A. Guest enters the restaurant
B. Guest is shown the table
C. Sits at the table and studies the menu
D. Places order
E. Waiter informs the kitchen
F. Cook prepares food
G. Food is served
H. Guest eats food
I. Use the "Payment" use case
J. Guest leaves

Alternatives:

- At A, if the restaurant is full, guest waits at the bar
- At D, if the dish is not available, the guest chooses another one
- At H, the sequence can be extended by the "Flowers and Champagne Proposal" use case

Description of Use Case Template

Software University

Use Case id and name	UC 001: Flight Check In			
Version	**Version**	**Issue**	**Author**	**Changes**
	0.01	26/01/2005	MW	Initial draft
Scope & Level	Primary use case for customer check in			
Goal in context	A customer who has previously booked a flight wishes to check in before his or her flight			
Preconditions	A customer has already registered and successful purchased a ticket for the flight			
Successful outcome	Customer has checked in			
Failure outcomes	**Failure**	**Outcome**	**Condition leading to outcome**	
	Customer has not purchased a ticket	1.If seats are available on the requested flight the customer is given the choice of purchasing a ticket 2.If no seats are available customer is offered a seat on a later flight	No ticket was previously purchased	
Primary actor	Customer or Check In Desk Attendant			
Secondary actors	None			
Main scenario	1.System prompts for customer name and flight number 2.Customer supplies name and flight number 3.System requests customer verification confirmation 4.Customer supplies passport details 5.System successfully verifies customer and flight details, confirming that the customer has checked in 6.Use Case ends successfully			
Alternatives	2a. Customer name is not found to be registered for the given flight number 2a1. If seats are available, customer may choose to purchase a seat. 2a2. If seats are not available on the originally selected flight, customer may choose to purchase a ticket on a later flight.			
Variations	If using a self-service kiosk a customer may scan his or her own passport, otherwise a Check In Desk Attendant will check his or her passport on their behalf. This should cause no difference in terms of system function, but the different users may have different usability requirements.			

How to implement Use Case?

- Choose an appropriate name
  - The title of the Use Case sets the context
- Illustrate complete behavior
- Provide inverse use cases
- Limit use cases to one behavior
- Represent actors' points of view
  - Easier to capture the use cases if you are "in the shoes" of the actor

Remember

User Story

Defines Who, What and Why of a product feature

≠

Use Case

Focuses on the functionalities of a feature or process

Practice

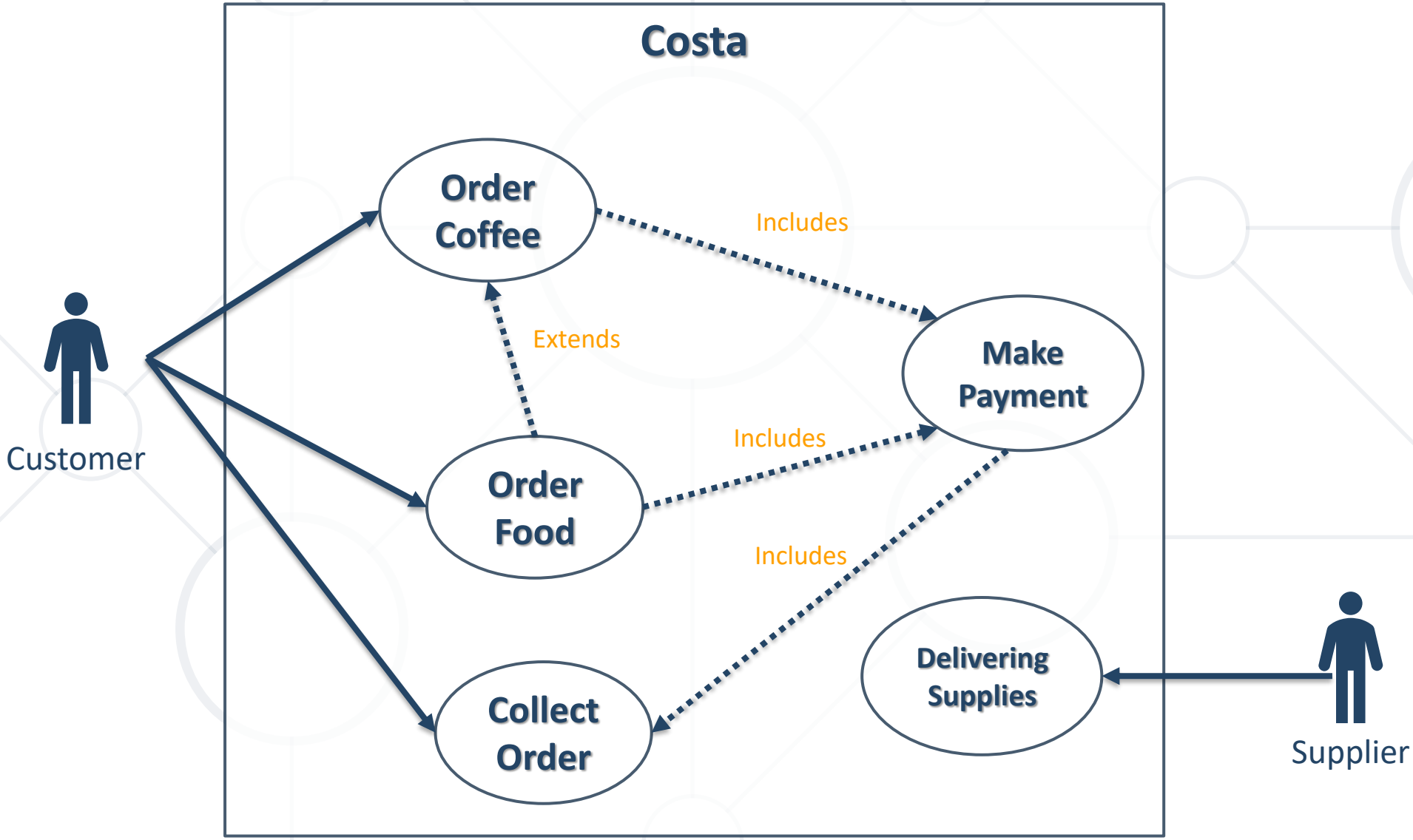
Live Exercise in Class (Lab)

Task 1

Make a Use Case Diagram for Costa:

- 2 Actors (Customer and Supplier)
- 5 Use Cases (Order Coffee/Make Payment/Order Food/Collect Order/Deliver Supplies)
- 3 Includes
- 1 Extend

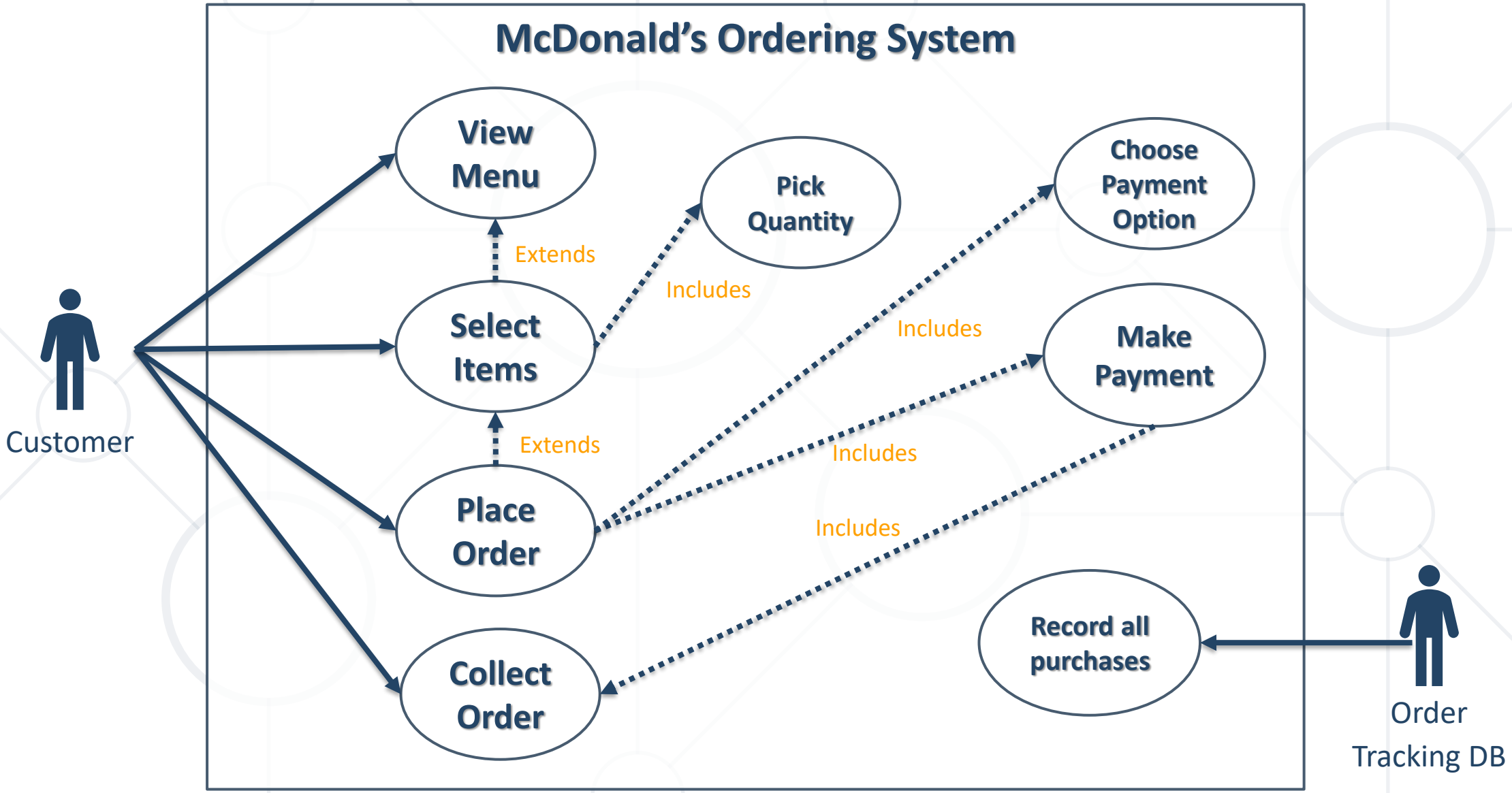
Solution of Task 1



Task 2

Software University

Solution of Task 2



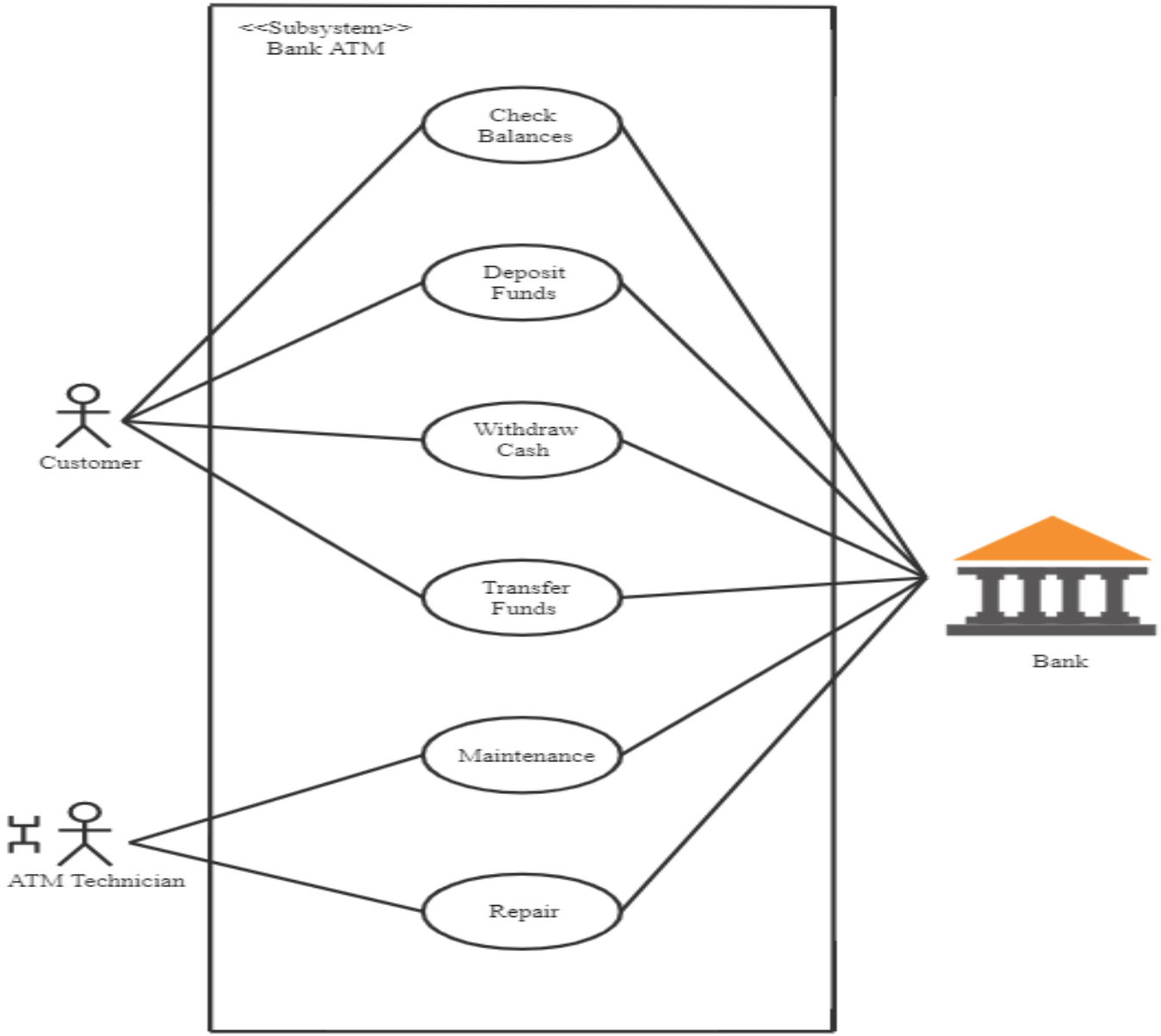
Task 3

Make Use Case Diagram for ATM Machine:

- 3 Actors (Customer/Bank/ATM Technician)
- 6 Use Cases (Check Balances/Deposit Funds/Withdraw Cash/Transfer Funds/Maintenance/Repair)

Write Description for Check Balances

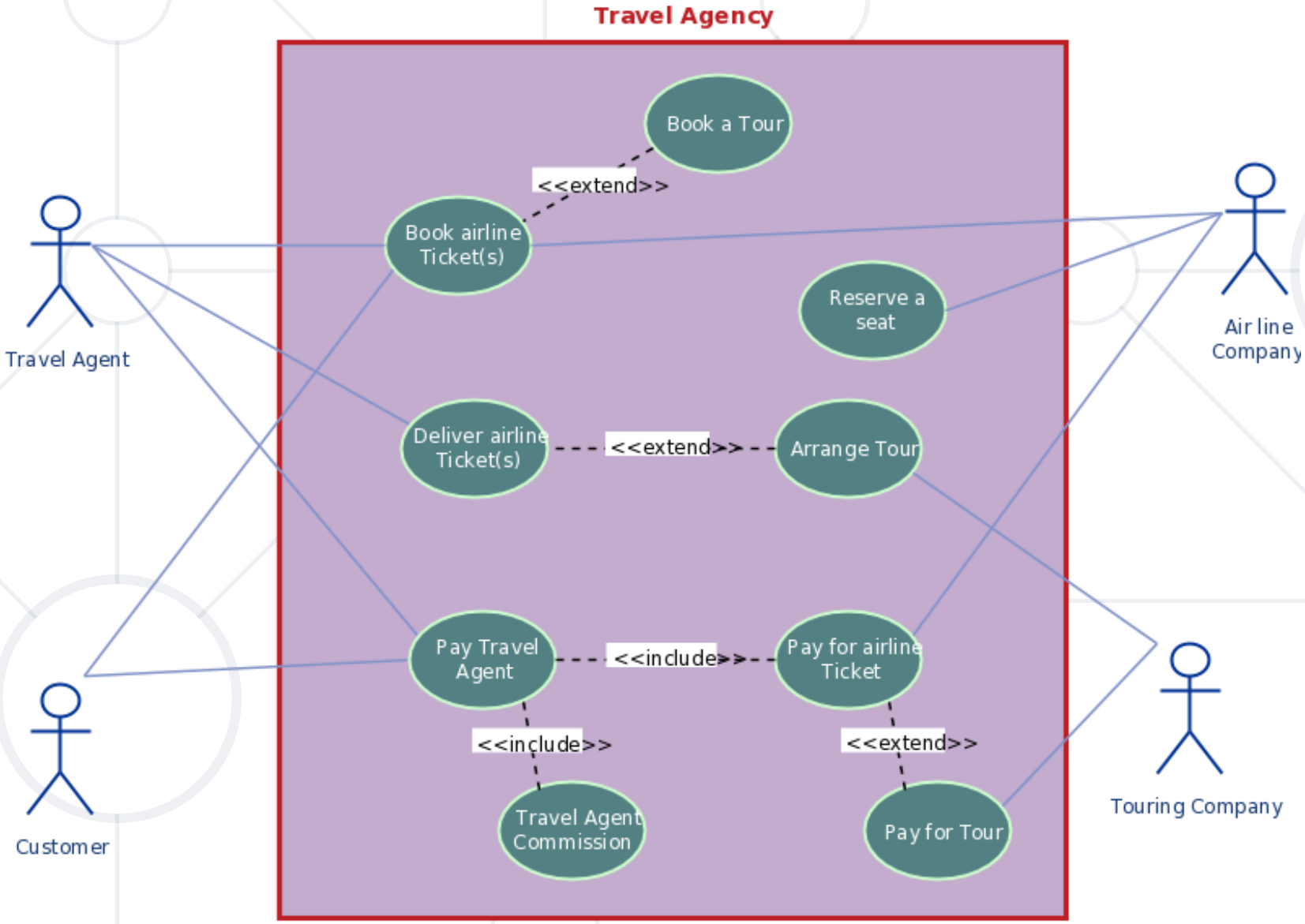
Solution of Task 3 – Diagram



Task 4

Make a Use Case Diagram for Travel Agency

Solution of Task 4



Homework

- Identify key Actors in the Case Study that are part of key Use Cases
- Start writing down and grouping Use Cases that will be included in the BRD
- Try linking the identified Use Cases with previously identified requirements
  - **One Use Case Diagram and Description can cover more than 1 Requirement**

Summary

- **Unified Modelling Language (UML)** a rich language to model software/system structure and behaviour
- There are two UML types – **Structural** and **Behavioural**
- **Use Case Modelling** is the fundamental UML method

SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

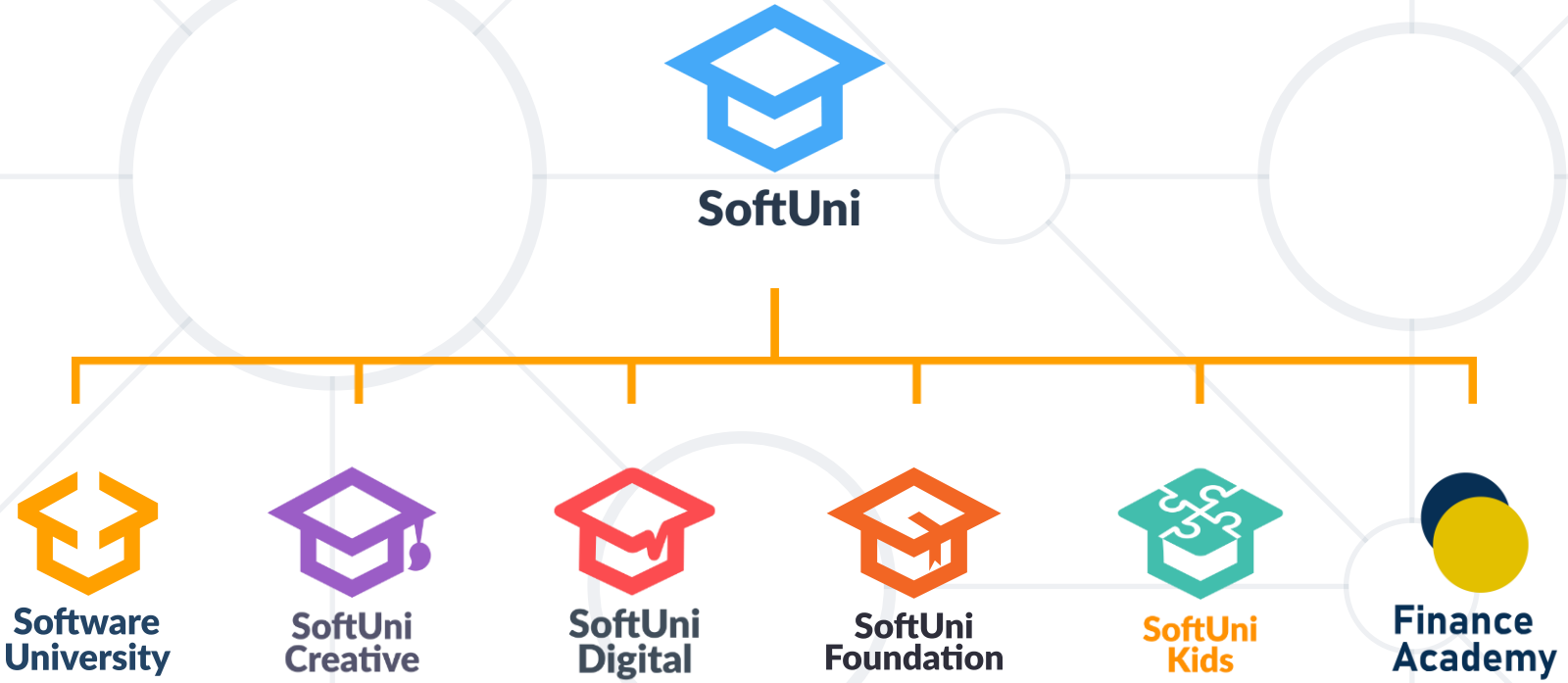
Postbank
Решения за твоето утре

BOSCH

DXC TECHNOLOGY

SmartIT

Questions?



License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg