

Jiri Michal

11530340

Subject: ITC508

Assignment #: 1

Lecturer: Simi Bajaj

Date: 3-Apr-15

Table of Contents

[Task 1 3](#_Toc415839537)

[Define the term ‘use case’ and explain the various types of actors in a Use Case. 3](#_Toc415839538)

[Explain the difference between the <<include>> and <<extends>> relationships in use case diagrams? Give examples 3](#_Toc415839539)

[Describe with examples ‘encapsulation’, ‘information hiding’, ‘polymorphism’ and ‘data abstraction’. 3](#_Toc415839540)

[What is the difference between USDP and the Waterfall lifecycles in the relationship of between activities and phases? 3](#_Toc415839541)

[Explain Class diagram. Outline the main steps in developing a class diagram for a Use Case (UC). 3](#_Toc415839542)

[Task 2 3](#_Toc415839543)

[Question 1 3](#_Toc415839544)

[Start line run 3](#_Toc415839545)

[Record employee joining the line 3](#_Toc415839546)

[Record employee leaving the line 3](#_Toc415839547)

[Stop line 3](#_Toc415839548)

[Question 2: Communication diagram 3](#_Toc415839549)

[Question 3: Class diagram 3](#_Toc415839550)

[Bibliography 4](#_Toc415839551)

# Task 1

## Define the term ‘use case’ and explain the various types of actors in a Use Case.

Use case is defined as a description of the system functionality (an activity that the system performs (Satzinger, Jackson, & Burd, 2010)) form users’ view (Bennett, McRobb, & Farmer, 2010).

There are two major types of actors:

* The role of humans who interact with a specific use case in the system.
* The role of other systems which interact with a specific use case in the system.
* The role of devices which interact with a specific use case in the system.

## Explain the difference between the <<include>> and <<extends>> relationships in use case diagrams? Give examples

The <<include>> relationship is used when some functionality is part of many use cases A, B, C ... so we separate this functionality to the separate use case S and then we include the use case S to use cases A, B, C... (Bennett, McRobb, & Farmer, 2010). For example if we have use cases “Pay anyone”, “Transfer”, and “Pay Bills” in each of them we have to check if is enough money on the account. So we form a new use case “Check balance” which will be included in previous use cases and simplifies their description.

The <<extends>> relationship is used when a use case X optionally adds some functionality to a use case Y. For example a use case “check availability of item” can extend a use case “add item to wish list”

## Describe with examples ‘encapsulation’, ‘information hiding’, ‘polymorphism’ and ‘data abstraction’.

The **encapsulation** means that each object contains its description (data) and its functionality (methods) in one unit (Satzinger, Jackson, & Burd, 2010). For example an object “wheel” is described by its centre position and its radius and it can spin (functionality).

The **information hiding** is a design principle in which an object’s data are not visible from outside of the object, and the access to its data can be provided by methods (setters and getters) (Satzinger, Jackson, & Burd, 2010). For example an object “dog” has attribute (data) name, and only way how to access dog’s name from outside is by methods getName(), and setName() .

The **polymorphism** is an ability to take shape in many forms that means in relationship with the object oriented modelling, that there is a possibility to call same message on different classes and the respond will be appropriate (Bennett, McRobb, & Farmer, 2010). For example if we have a array of shapes which contains triangles, squares, and circles. And we call a method getArea() on the array of shapes we get appropriate outcome, even each shape has different formula to calculate its area.

The **data abstraction** is a process of simplifying and pinpointing of important characteristics. The abstraction is widely use in modelling where we have a view of a physical system and with abstraction we reduce the irrelevant data and keep the significant data to model the system. What is and is not important depends on a purpose of the model. For example if we have a physical system of pipelines we will keep data about position and length but we omit a colour of the pipes.

## What is the difference between USDP and the Waterfall lifecycles in the relationship of between activities and phases?

## Explain Class diagram. Outline the main steps in developing a class diagram for a Use Case (UC).

# Task 2

## Question 1

### Start line run

### Record employee joining the line

### Record employee leaving the line

### Stop line

## Question 2: Communication diagram

## Question 3: Class diagram

# Bibliography

Bennett, S., McRobb, S., & Farmer, R. (2010). *Object-Oriented Systems Analysis and Design Using UML (4th edition).* Berkshire: McGraw-Hill Education.