Požadavky a
Use Case Diagramy

SW požadavky

- **Požadavek** lze definovat stručně jako „specifikaci toho, co by mělo být implementováno”.

- **Funkční požadavky** (functional req.) určují, jaké chování nebo jaké služby bude systém nabízet

- **Nefunkční požadavky** (non-functional req.) specifikují vlastnosti nebo omezující podmínky daného systému (kvalita, technologie, dokumentace, řízení projektu, ..)

- Požadavek vyjadřuje CO, nikoliv JAK.

Nefunkční požadavky

- Výkon
- Kapacita
- Dostupnost
- Shoda se standardy
- Zabezpečení
- Dokumentace
- Organizace projektu
- Technologie
- …

Nefunkční požadavky

Nefunkční požadavky je třeba zaznamenat a neustále je promítat do návrhu řešení. Pro tyto účely lze použít katalogový záznam

- Identifikační číslo
- Datum poslední změny
- Kdo ho požaduje, kdo ho přijímá
- Priorita požadavků
- Stručný název požadavku
- Detailní popis požadavku
- Návaznost na další požadavky
- Komplexnost/složitost požadavku

„Podobně“ lze zacházet i s funkčními požadavky.

Zápis požadavků

Notation	Description
Natural language	Numbered sentences in natural language, each sentence expressing one requirement. *E.g. Project assignment.*
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement. *E.g. Textual specification of UML use cases. (the table view)*
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. *E.g. Main flow in the UML UC textual specification.*
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system. *E.g. UML use case and activity diagrams.*
Mathematical specifications	Notations based on mathematical concepts; *E.g. finite-state machines or sets.* Although they can reduce the ambiguity in a requirements document, most customers don't understand them and are reluctant to accept it as a system contract

Příklad zápisu požadavků

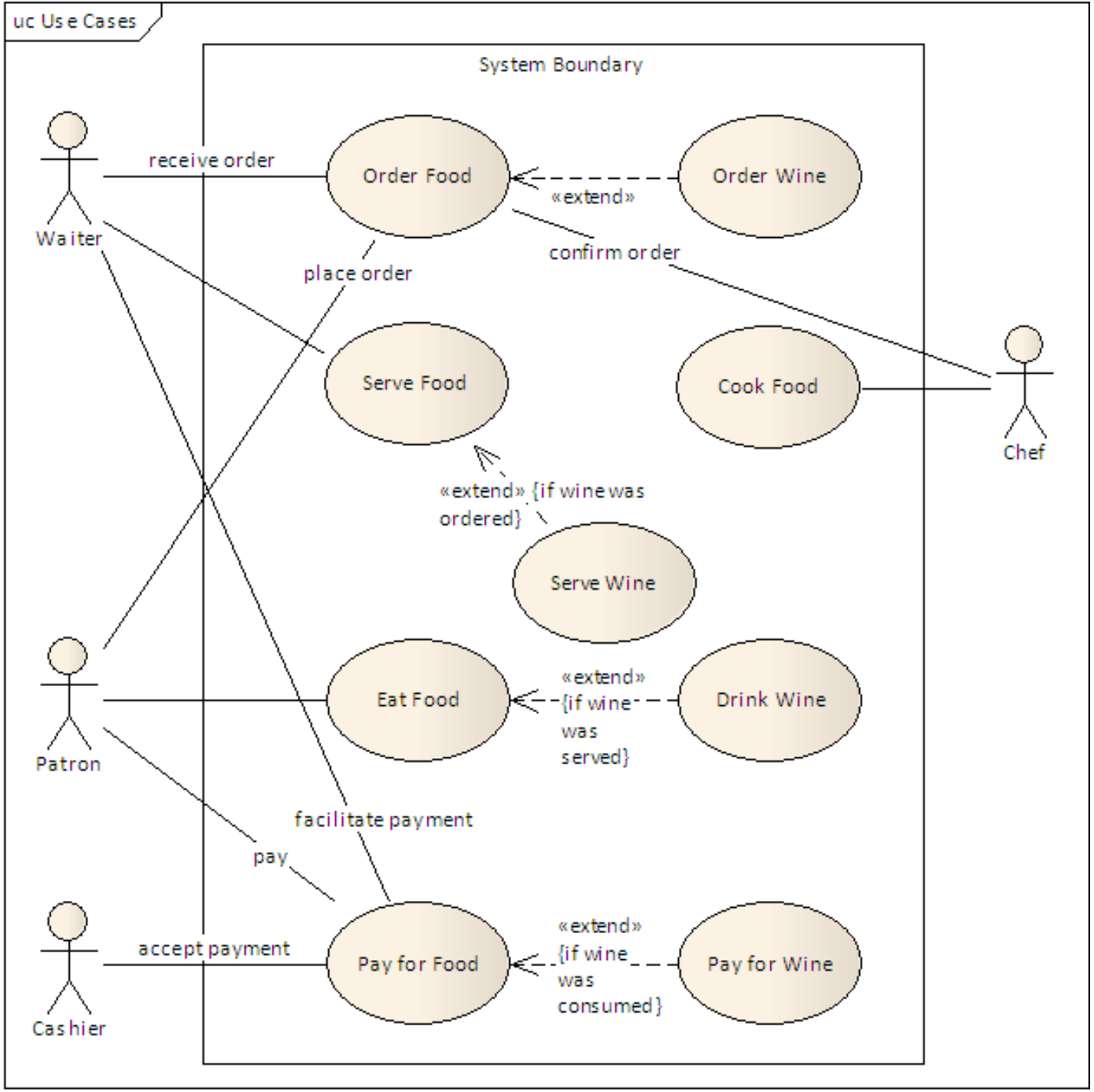
ID	Název požadavku	Akceptační kritérium
1	Evidence a dokumentace entit památkového fondu	Aplikace umožňuje založit, změnit, smazat, udržovat historii, vyhledávat, třídit, exportovat a tisknout záznamy k jednotlivým entitám PF s prostorovým určením v rámci celého IISPP včetně jejich atributů definovaných NPÚ, standardů GIS, RÚIAN a mezinárodních doporučení/standardů pro uživatele a zájmové skupiny podle územní působnosti a kompetencí. (kap. 3.2.1, 3.6.2)
2	Evidence právních stavů	Aplikace umožňuje založit, změnit, zneplatnit, udržovat historii, vyhledávat, třídit, exportovat a provádět vazby právních stavů na jednotlivé entity (skupiny entit) PF. (kap. 3.6.2.3)
3	Evidence, dokumentace a monitoring KP	Aplikace umožňuje založit, změnit, smazat, udržovat historii, vyhledávat, třídit, exportovat a tisknout karty stavu KP podle jejich typu (viz doplněk této přílohy 2.1 Tabulky stavu KP) a automaticky provádět vazbu na

Funkční požadavky – případy užití

Diagram případů užití (Use Case Diagram) zachycuje vnější pohled na modelovaný systém a tím pomáhá odhalit hranice systému.

Jde o posloupnost souvisejících transakcí mezi účastníkem (zpravidla uživatelem v určité roli, ale také jiným systémem) a systémem během vzájemného dialogu. Hlavním účelem je zachycení aktérů, kteří se systémem komunikují a vztahů mezi službami a těmi, kterým jsou poskytovány, a to vizuální i textovou podobou, která je srozumitelná vývojářům systému i zákazníkům (tj. těm, kteří jej mají používat).

Use Case Diagram



Komponenty a vztahy

Případ užití (značený oválně) – posloupnost akcí ve vztahu s aktéry, může obsahovat vazby

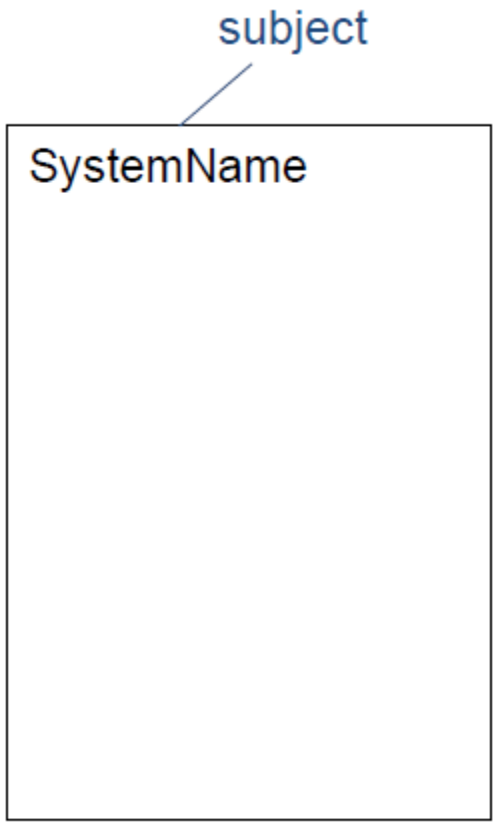
- Include – případ užití může obsahovat jiný (Zahrnující – např.: Editovat text → Psát text, Vložit obrázek, atd.)
- Extend – případ užití může rozšiřovat jiný (Rozšiřující – např.: Otevřít dokument → Import z jiného formátu, atd.)
- Generalizace (dědičnost) – případ užití může být speciálním případem jiného

Aktér/účastník (značený figurou) – popis externích objektů vstupujících do vztahu s procesy, může obsahovat vazby

Generalizace (dědičnost) – aktér může být speciálním případem jiného

Komponenty a vztahy

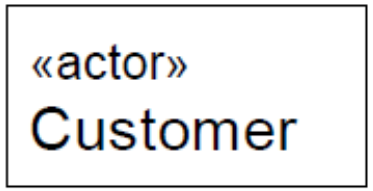
- We create a Use Case model containing:
  - **Subject** – the edge of the system
    - also known as the system boundary
  - **Actors** – who or what uses the system
  - **Use Cases** – things actors do with the system; functions the system should offer to its users
  - **Relationships** – between actors and use cases

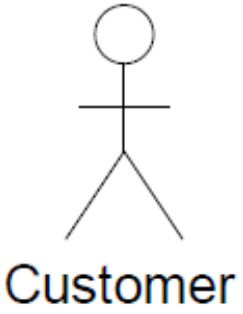


- Can there be a direct relationship between actors?

Komponenty a vztahy

- An actor is anything that interacts **directly** with the system
  - Actors identify who or what uses the system and so indicate where the system boundary lies



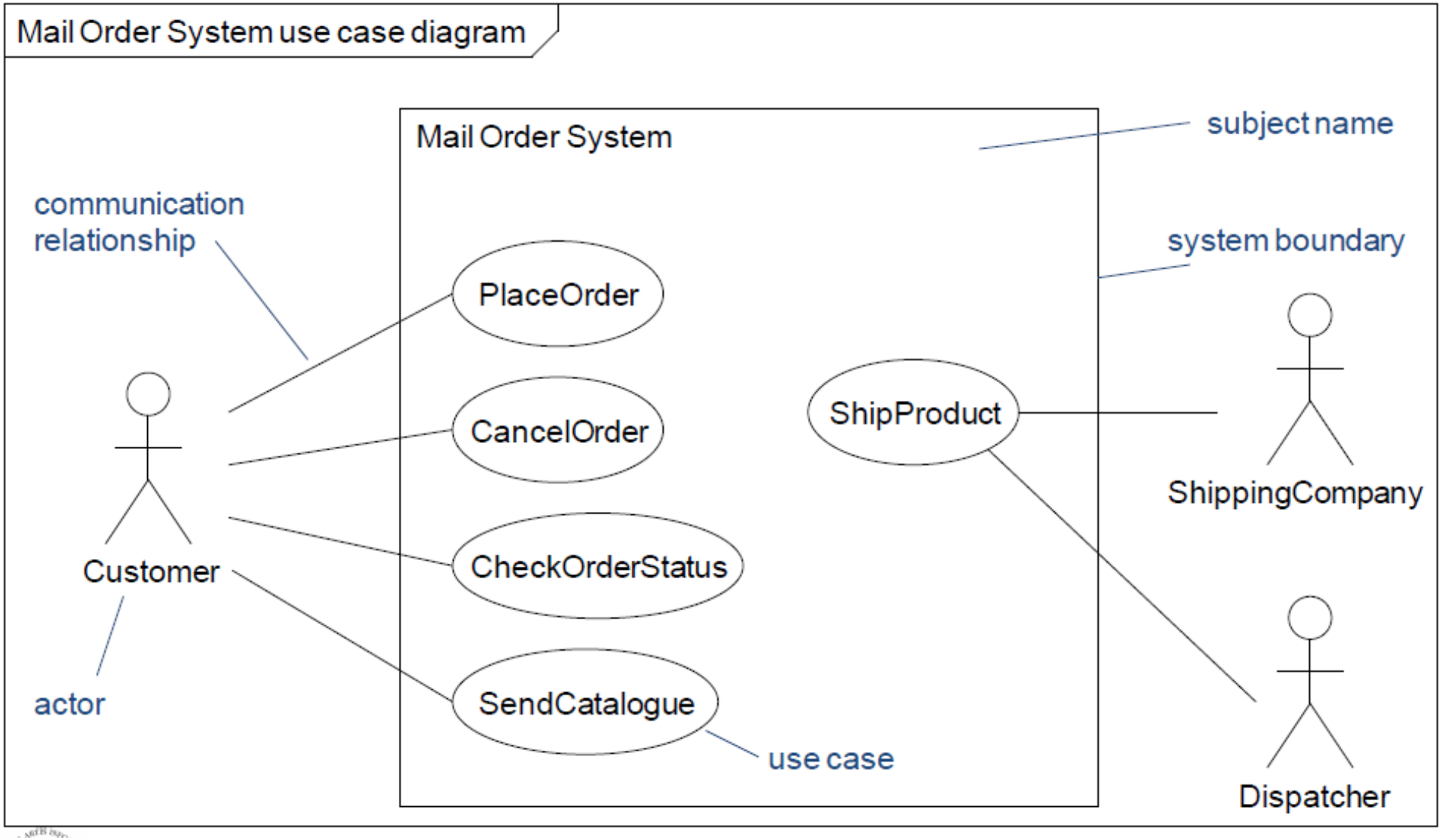


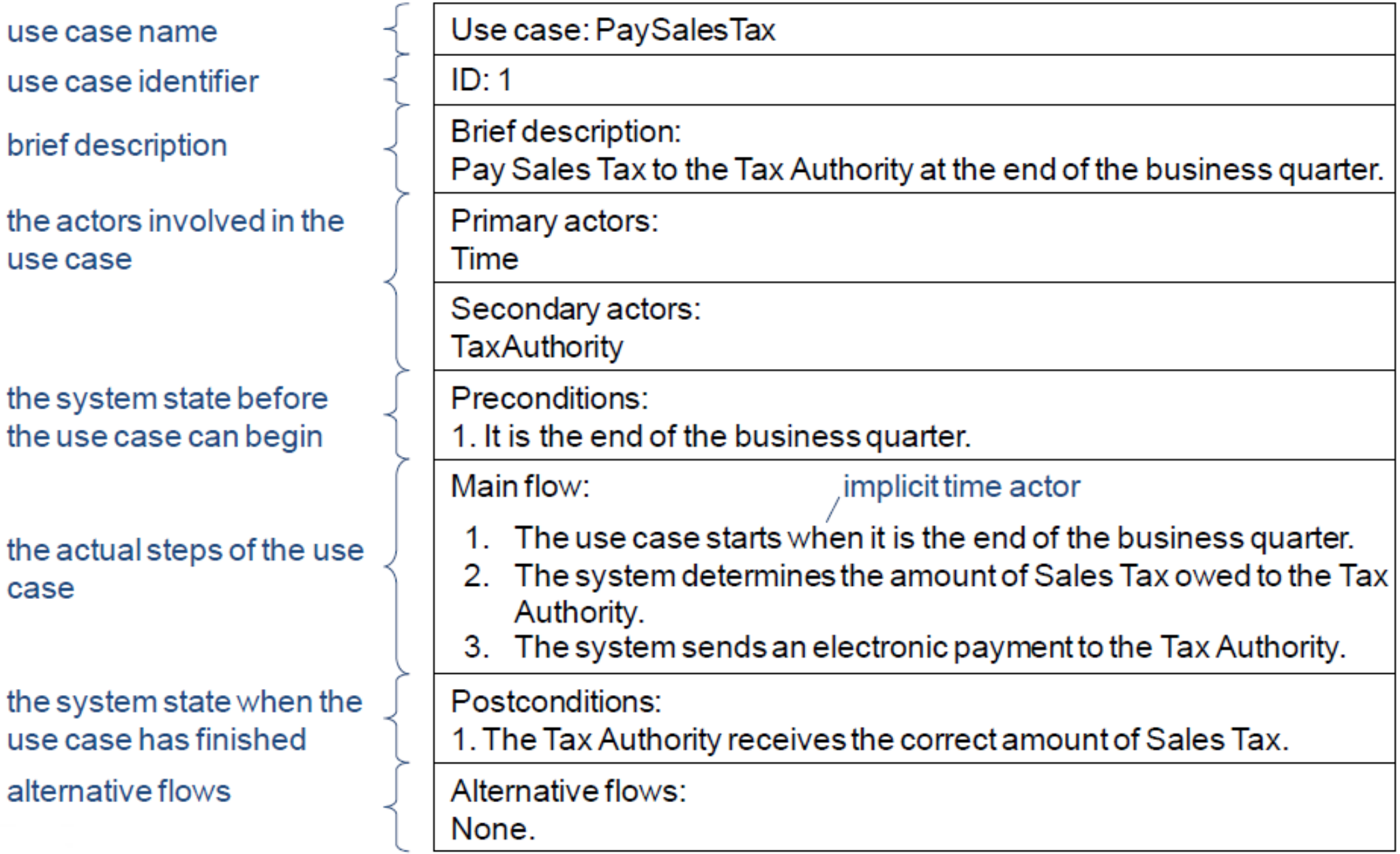
- Actors are **external** to the system
- An Actor specifies a **role** that some external entity adopts when interacting with the system
  - Can one actor represent two physical persons?
  - Can one physical person match to two actors?
  - Can there be two actors with the same name in the model?

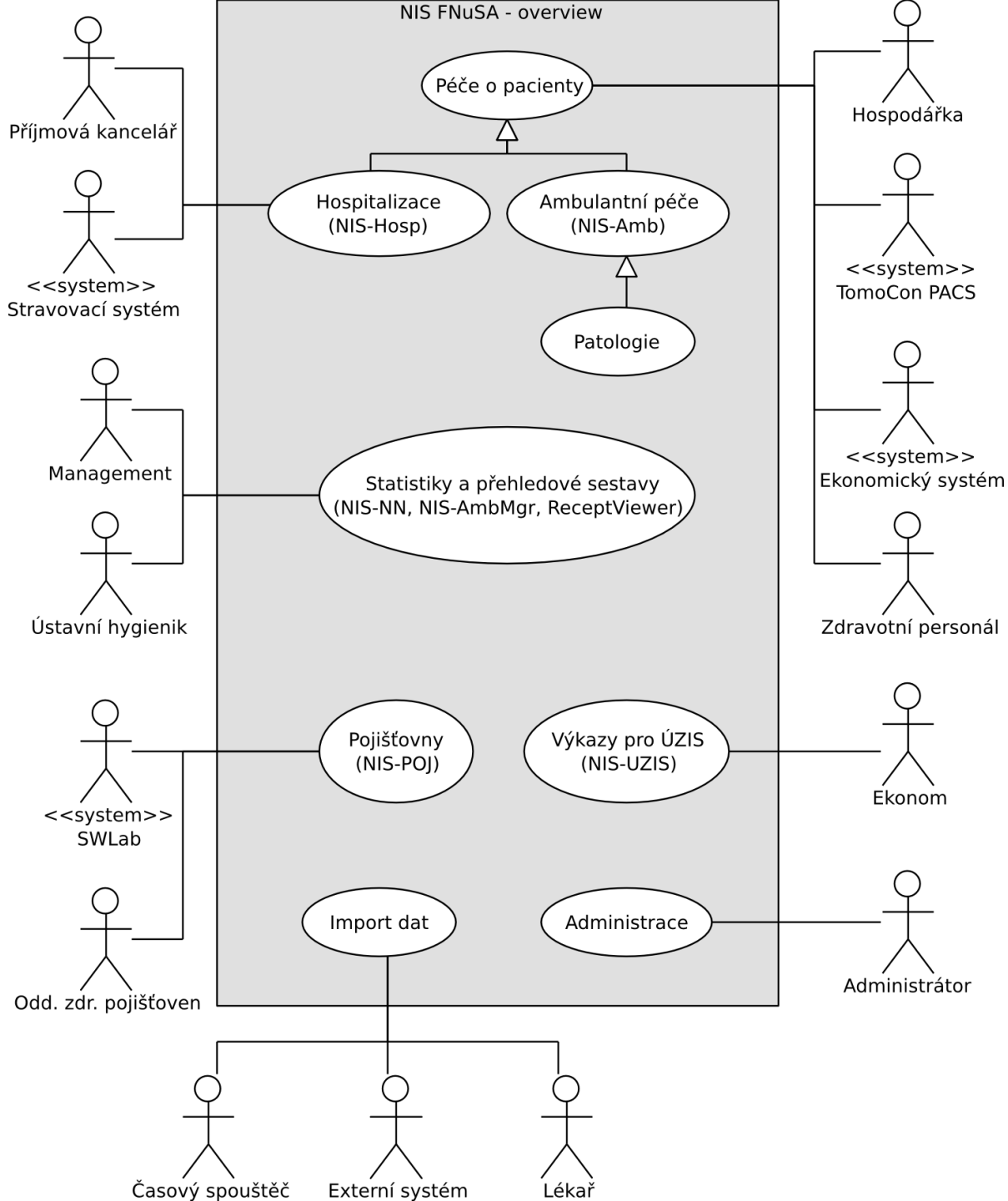
Komponenty a vztahy

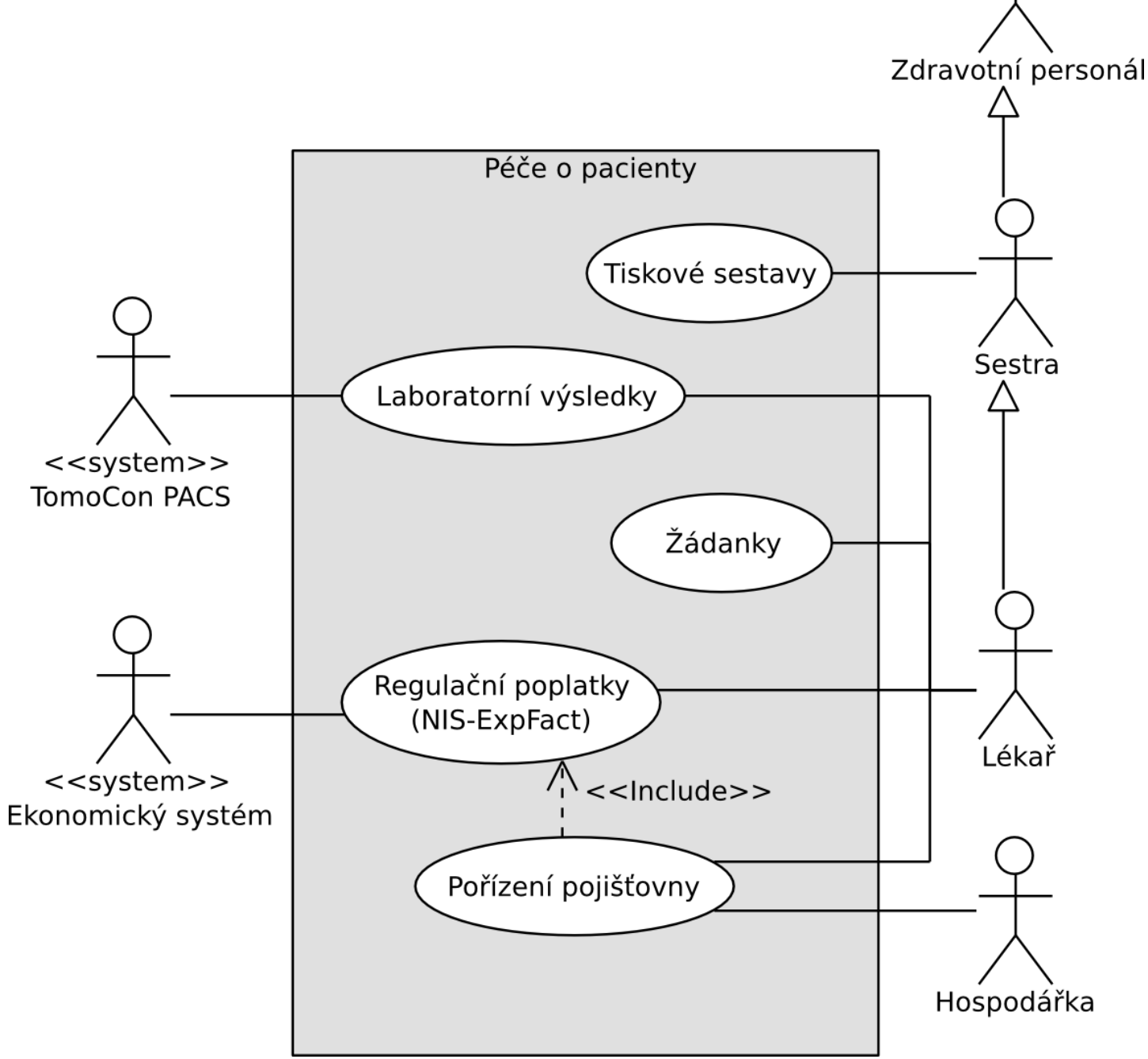
- A use case is something an actor needs the system to do. It is a “case of use” of the system by a specific actor.
- Use cases are always started by an actor
  - The **primary actor** triggers the use case
  - Zero or more **secondary actors** interact with the use case in some way
  - Does the UC diagram tell me which actor is primary/secondary?
- Use cases are always written from the **point of view of the actors**.

Komponenty a vztahy




use case name
use case identifier
brief description
the actors involved in the use case
the system state before the use case can begin
the actual steps of the use case
the system state when the use case has finished
alternative flows
Use case: PaySalesTax
ID: 1
Brief description:
Pay Sales Tax to the Tax Authority at the end of the business quarter.
Primary actors:
Time
Secondary actors:
TaxAuthority
Preconditions:
1. It is the end of the business quarter.
Main flow:
implicit time actor
1. The use case starts when it is the end of the business quarter.
2. The system determines the amount of Sales Tax owed to the Tax Authority.
3. The system sends an electronic payment to the Tax Authority.
Postconditions:
1. The Tax Authority receives the correct amount of Sales Tax.
Alternative flows:
None.


NIS FNuSA - overview
Péče o pacienty
Hospitalizace
(NIS-Hosp)
Ambulantní péče
(NIS-Amb)
Patologie
Statistiky a přehledové sestavy
(NIS-NN, NIS-AmbMgr, ReceptViewer)
Pojišťovny
(NIS-POJ)
Výkazy pro ÚZIS
(NIS-UZIS)
Import dat
Administrace
Příjmová kancelář
<<system>>
Stravovací systém
Management
Ústavní hygienik
<<system>>
SWLab
Odd. zdr. pojišťoven
Časový spouštěč
Externí systém
Lékař
Hospodářka
<<system>>
TomoCon PACS
<<system>>
Ekonomický systém
Zdravotní personál
Ekonom
Administrátor


Zdravotní personál
Péče o pacienty
Tiskové sestavy
Sestra
Laboratorní výsledky
<<system>>
TomoCon PACS
Žádanky
Regulační poplatky
(NIS-ExpFact)
<<system>>
Ekonomický systém
Lékař
<<Include>>
Pořízení pojišťovny
Hospodářka