# VIS - Exercise 4

# Artifact 3
## Non-functional requirements

2021/2022

Martin Radvanský

# Repetition Exercise 3

**Artifact 2 – Use-Case diagrams**

List of functional requirements

- A Single image of Use-Case diagram of developed IS.
- Three selected Use-Case described in greater detail in Use-Case description (Scenarios) 3 pages of text.
- Three images of activity diagram for selected Use-Cases

| | |
|---|---|
| **Use Case ID:** | 3 |
| **Use Case Name:** | Deposit check |
| **Actors:** | Customer |
| **Description:** | Deposit cash without using ATM card by using E-Card system. |
| **Preconditions:** | 1- The Customer has an activated E-Bank username and password.<br>2- The agreement should be signed by the customer.<br>3- The check must be valid. |
| **Postconditions:** | 1- Customer account balance is increased by the amount of the deposit check. |
| **Normal Flow:** | 1- Open the application.<br>2- The application shows welcome screen.<br>3- Log in to the application.<br>4- Choose the account.<br>5- Choose the transaction then deposit check service.<br>6- Enter the amount of money of the check and submit it.<br>7- Receiving the barcode.<br>8- Scan the barcode.<br>9- Take pictures for the front and back of the check<br>10- Receive notification.<br>11- Log out of the application. |
| **Alternative Flows:** | 7a. if the customer didn't receive the barcode :<br>4- Customer will click on the get barcode bottom.<br>5- Bank sends a new barcode.<br>6- Use case resumes on step 8 of normal flow. |

# Discussion

## Use-Case diagram
## List of functional requirements

# Non-functional (technical) requirements

1. Conceptual domain model
2. Technological decisions
3. Estimation of entity sizes and quantities.
   - Estimation of the number of users working simultaneously with the system.
   - Types of user interactions with the system and estimation of their complexity.
   - First idea of the system layout.
   - Choice of the used platform.

**Justification of the choice of technology, storage…**

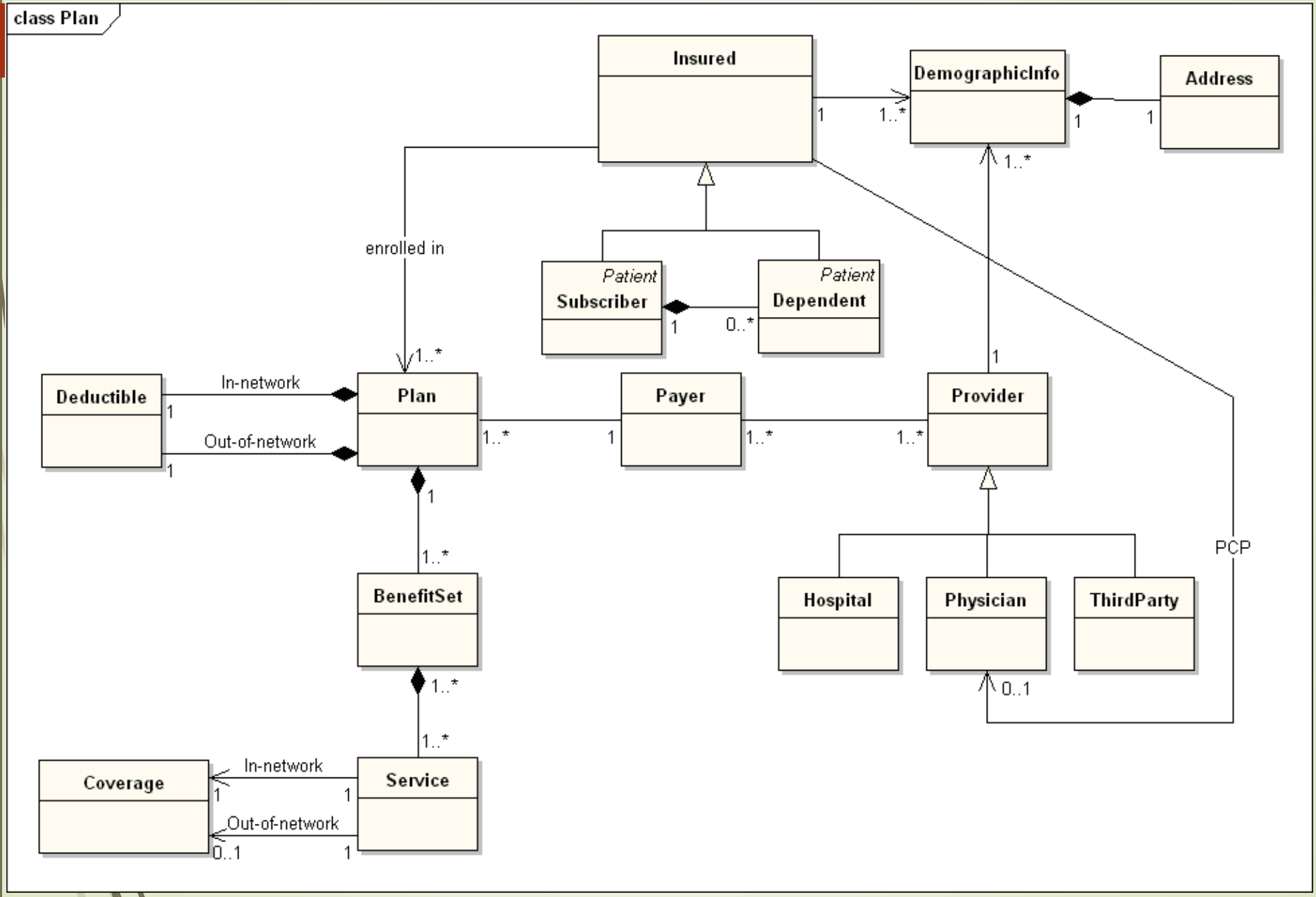
# Conceptual domain model

- **The domain model is based on the specification (Vision) and Use case diagram of the IS.**
- The classes in the domain model are highly simplified, do not contain methods and have only important attributes. The domain model describes the basic entities of the system and the relationships between them.
  **It is platform independent**.

**Model creation:**

- The creation of the domain model is based on the client's specifications. From it, we identify the key entities and the relationships between them. We plot these entities as classes in the model.

# Conceptual domain model



# Concept domain model

Informační systém *STAG* bude sloužit pro odporu správy kurzů včetně elektronického přihlašování a odhlašování.

Správa studijních kurzů umožní lektorům přidávání nových kurzů, mazání kurzů a úpravu stávajících parametrů, kterými jsou název, popis, prerekvizity, kapacita, nastavení rozvrhu a rozlišení, zda jde o prezenční nebo distanční kurz.

Studenti dostanou možnost využít systém k elektronickému zápisu do kurzů a odhlašování z kurzů. Zápis do kurzu bude umožněn jen v případě, že dosud není naplněna kapacita kurzu. Pokud dojde k vyčerpání kapacity kurzu, systém pošle upozornění na email lektora zodpovědného za kurz.

Jednou z doplňkových funkcí systému bude podpora pro vkládání pracovních výkazů o odpracovaných hodinách pro externí lektory, kteří na rozdíl od interních lektorů nepobírají měsíční mzdu a jsou placeni na základě odpracovaných hodin.

**noun -** candidate for class
**noun -** candidate for class attribute
**verb –** candidate for procedure/class method

# Estimation of entity size and number of users

- Estimation of the number of entities.
- Estimation of the number of concurrent users (from the Vision according to the scope of the system).
- Calculating the size of memory occupied by one entity.
- Estimation of storage requirements for the entire system.

# Programming tasks

- Create two classes with association links from UI to DB (.NET or JAVA)
- Use different Business logic patterns
- Use different Data patterns