

JEGYZŐKÖNYV

Webes adatkezelő környezetek

Féléves feladat

Pszichiátria Nyilvántartó Rendszer

Készítette: Berki Viktor

Neptunkód: ZY5P7F

Dátum: 2025. december 1.

Miskolc, 2025

Tartalomjegyzék

1. XML alapú adatkezelés tervezése és megvalósítása	2
1.1. Az adatbázis ER modell tervezése	2
1.2. Az adatbázis konvertálása XDM modellre	3
1.3. Az XDM modell alapján XML dokumentum készítése	3
1.4. Az XML dokumentum alapján XMLSchema készítése	4
2. DOM program készítése	4
2.1. Adatolvasás	4
2.2. Adat-lekérdezés	5
2.3. Adatmódosítás	6

Bevezetés

A féléves feladat során egy pszichiátriai osztály nyilvántartó rendszerét valósítottam meg. A rendszer célja az osztályok, orvosok, páciensek, a hozzájuk tartozó kórlapok és a felírt gyógyszerek adatainak strukturált tárolása és kezelése XML alapokon. [cite: 10]

1 XML alapú adatkezelés tervezése és megvalósítása

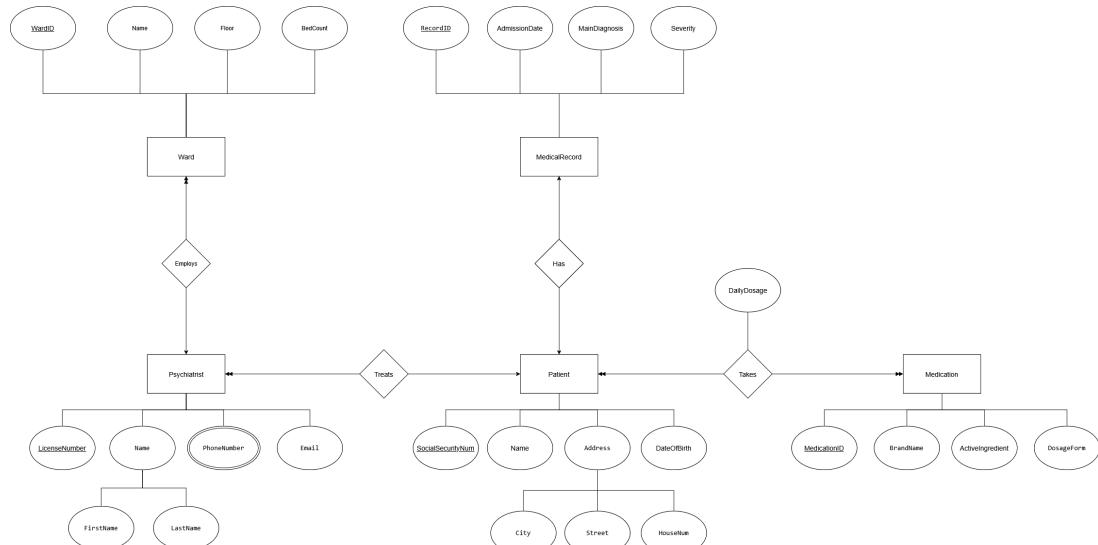
1.1 Az adatbázis ER modell tervezése

A tervezés során öt egyedet definiáltam: **Ward** (Osztály), **Psychiatrist** (Pszichiáter), **Patient** (Páciens), **MedicalRecord** (Kórlap) és **Medication** (Gyógyszer).

A kapcsolatok a következők:

- **1:N kapcsolat:** Egy osztályon több orvos dolgozik, egy orvos több beteget kezel.
- **1:1 kapcsolat:** Egy betegnek pontosan egy kórlapja van.
- **M:N kapcsolat:** A betegek és gyógyszerek között (a beteg szedi a gyógyszert), melyet a *Takes* kapcsolat valósít meg *DailyDosage* tulajdonsággal.

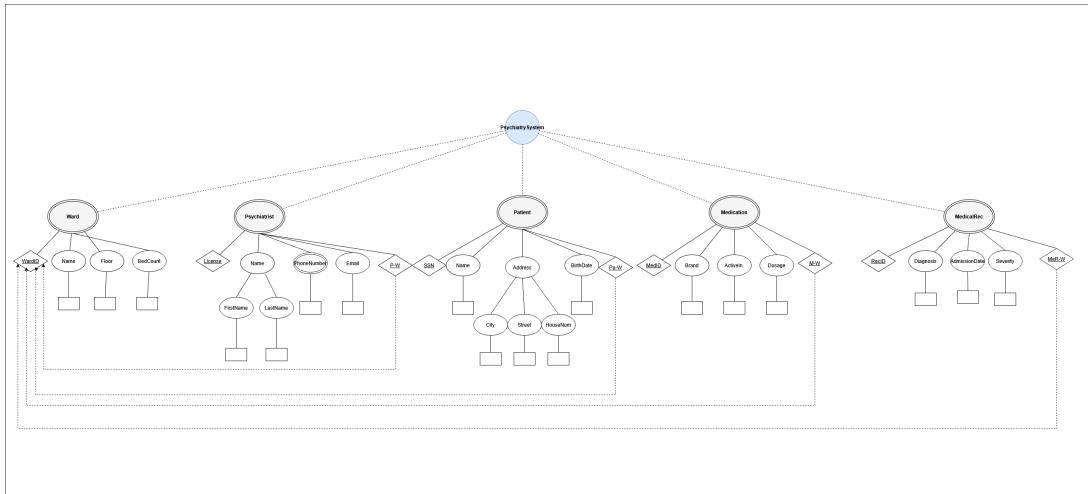
Az egyedek rendelkeznek kulcsokkal (PK), összetett (pl. Lakcím, Név) és többértékű (pl. Telefonszám) tulajdonságokkal is. [cite: 15]



1. ábra. A rendszer ER modellje

1.2 Az adatbázis konvertálása XDM modellre

Az ER modellt hierarchikus XDM modellé alakítottam át. A "vonalak ne keresztezzék egymást" szabály betartása érdekében a **Medications** listát külön ágon, törzsadatként kezelem. A hierarchia jobb oldala a **Ward → Psychiatrist → Patient** láncolatot követi. A **MedicalRecord** a páciensbe ágyazódik be. Az M:N kapcsolatot a **Prescription** elem oldja meg, amely IDREF attribútummal mutat a bal oldali gyógyszer listára. [cite: 17, 19]



2. ábra. A rendszer XDM modellje

1.3 Az XDM modell alapján XML dokumentum készítése

Az XML dokumentum a ‘PsychiatrySystem’-gyökérelem alatt tárolja az adatokat. minden többszörösen előforduló elemből (pl. Ward, Psychiatrist, Medication) legalább két példány készült. A dokumentum tartalmazza a szükséges attribútumokat (id, ref) és a kommenteket. [cite: 21, 22]

Fájlnév: ZY5P7F_XML.xml

1. Listing. Részlet a lapos szerkezetű XML fájlból

```
<PsychiatrySystem>
    <Wards>
        <Ward id="W01">
            <Name>Acute Care Unit</Name>
        </Ward>
    </Wards>
    <Psychiatrists>
        <Psychiatrist license="LIC-12345" wardRef="W01">
            <Name>...</Name>
        </Psychiatrist>
    </Psychiatrists>
    <Patients>
        <Patient ssn="111-22-3333" doctorRef="LIC-12345">
```

```

<Name>John Doe</Name>
</Patient>
</Patients>
</PsychiatrySystem>

```

1.4 Az XML dokumentum alapján XMLSchema készítése

Az XSD sémában saját egyszerű és összetett típusokat definiáltam (pl. SSNTType, AddressType). A kapcsolati integritást xs:key és xs:keyref elemekkel biztosítottam, így a receptek csak létező gyógyszer ID-ra hivatkozhatnak. [cite: 24, 25, 26]

Fájlnév: ZY5P7F_XMLSchema.xsd

2. Listing. Részlet az XSD sémából (Kulcsok és Unique)

```

<xs:keyref name="DoctorToWard" refer="WardKey">
    <xs:selector xpath="Psychiatrists/Psychiatrist"/>
    <xs:field xpath="@wardRef"/>
</xs:keyref>

<xs:unique name="UniqueRecordPerPatient">
    <xs:selector xpath="MedicalRecords/MedicalRecord"/>
    <xs:field xpath="@patientRef"/>
</xs:unique>

```

2 DOM program készítése

A feladat második részében egy Java alkalmazást készítettem, amely DOM (Document Object Model) segítségével dolgozza fel az XML állományt.

Projekt adatok: [cite: 30, 31, 32]

- **Project name:** ZY5P7FDOMParse
- **Package:** zy5p7f.domparse.hu
- **Class names:** ZY5P7FDomRead, ZY5P7FDomQuery, ZY5P7FDomModify
- **XML name:** ZY5P7F_XML.xml

2.1 Adatolvasás

A ZY5P7FDomRead osztály beolvassa a teljes XML dokumentumot a memóriába, majd egy rekurzív metódus segítségével végigjárja a DOM fát. A program blokkos formában írja ki a konzolra az elemeket, attribútumokat és a szöveges tartalmakat. [cite: 33, 37]

3. Listing. Részlet az Adatolvasás kódból

```
// Rekurzív metódus a fa bejárására
private static void printNode(Node node, String indent) {
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        System.out.print(indent + "Elem: " + node.getNodeName());

        // Attribútumok kiírása
        if (node.hasAttributes()) {
            NamedNodeMap nodeMap = node.getAttributes();
            for (int i = 0; i < nodeMap.getLength(); i++) {
                Node tempNode = nodeMap.item(i);
                System.out.print(" | " + tempNode.getNodeName() + "=" +
                    tempNode.getNodeValue());
            }
        }
        // ... (további logika)
    }
}
```

2.2 Adat-lekérdezés

A ZY5P7FDomQuery osztályban **4 különböző lekérdezést** valósítottam meg a feladatkiírásnak megfelelően. A lekérdezések során a lapos szerkezet miatt az elemeket az azonosítóik (ID) és hivatkozásaik (Ref) alapján kellett összekapcsolni.

A megalósított lekérdezések:

1. Osztályok neveinek listázása.
2. Adott osztályon ("Acute Care Unit") dolgozó orvosok lekérdezése (FK kapcsolat).
3. Adott beteg ("John Doe") diagnózisának lekérdezése (1:1 kapcsolat).
4. Gyógyszerek számának aggregálása.

4. Listing. Részlet: Orvosok keresése osztály alapján

```
// Megkeressük azokat az orvosokat, akiknek a wardRef attribútuma
// egyezik
NodeList docList = doc.getElementsByTagName("Psychiatrist");
for (int i = 0; i < docList.getLength(); i++) {
    Element doctor = (Element) docList.item(i);
    if (doctor.getAttribute("wardRef").equals(wardId)) {
        // ... Orvos nevénak kiírása ...
    }
}
```

2.3 Adatmódosítás

A ZY5P7FDomModify osztályban **4 különböző módosítást** végeztem el az XML adatain. A program a módosítások után a frissített DOM fát elmenti egy ZY5P7F_XML_Modified.xml nevű fájlba.

A végrehajtott módosítások:

1. Gyógyszer átnevezése ("Xanax" → "Xanax SR").
2. Ágyszám növelése egy osztályon.
3. Beteg lakcímének (Város) javítása.
4. Orvos email címének frissítése.

5. Listing. Részlet: Több módosítás végrehajtása

```
// 1. Gyógyszer átnevezése
if ("MED001".equals(med.getAttribute("id"))) {
    brand.setTextContent("Xanax SR");
}
// ...
// 2. Ágyszám növelése
if ("W01".equals(ward.getAttribute("id"))) {
    beds.setTextContent("35");
}
// Mentés fájlba...
```