

Webkönyvtárak

Tantárgy kód: THE_00583_L_5_G

Webes adatkezelő környezetek

Gyógyszeralapanyag rendelése

Készítette: Antal Béla

Neptunkód: GTE1Y4

Dátum: 2025.05.18

Tartalomjegyzék

1. A feladat leírása	3
1.1 Az adatbázis ER modell tervezése.....	4
1.2 Az adatbázis konvertálása XDM modellre.....	5
1.3 Az XDM modell alapján XML dokumentum készítése.....	7
1.4 Az XML dokumentum alapján XMLSchema készítése	10
2. feladat	13

1. A feladat leírása

A feladat egy gyógyszerhatóanyag gyártó cég rendeléseinek nyomon követése és azok adatainak rendszerezett formában történő feldolgozása, valamint megjelenítése. A kiértékelést követően látni kell a rendelt készítmény nevét, a rendelt készítmény azonosítóját, a minőségellenőrző dolgozó nevét a minőségbiztosító dolgozó nevét, a vevő nevét és a szállítást lebonyolító logisztikai cég adatait.

A feladat megvalósítása során elsőként egy ER (Entity-Relationship) modellt kell készíteni, amely legalább öt különböző entitást tartalmaz. Ezek az entitások reprezentálják az adatbázis főbb szereplőit.

Miután az ER modell elkészült, a következő lépésként XDM modell elkészítése szükséges, amely az entitások és kapcsolataik XML struktúrában történő leképezését szolgálja. Ez a modell már közelebb áll a technikai megvalósításhoz, és lehetővé teszi a későbbi XML dokumentum egyszerűbb létrehozását.

Ezt követően az XDM modell alapján létre kell hozni egy XML dokumentumot, amely tartalmazza a gyógyszergyártó cég működéséhez kapcsolódó adatokat – például egy adott rendeléshez tartozó gyártó részleg nevét, készítmény nevét, dolgozók adatait, vevői és szállítási információkat. Az XML dokumentumot szabványos módon kell megfogalmazni.

Az XML dokumentumhoz XMLSchema (XSD) fájl is készítendő, amely az XML adatszerkezetét validálja és biztosítja, hogy az adatok a megfelelő formátumban és logikai szerkezetben szerepeljenek.

Az XSD használata elengedhetetlen az adatok konzisztenciájának biztosításához.

A végső lépésként egy Java nyelven írt DOM (Document Object Model) alapú programot kell létrehozni, amely képes az XML fájl beolvasására, az adatok értelmezésére és azok strukturált, áttekinthető formában történő kiírására. A programnak a konzolban megjelenített adatokat ki kell tudnia másolnia fájlba. A feladatot a jegyzőkönyv 2. feladata részletezi.

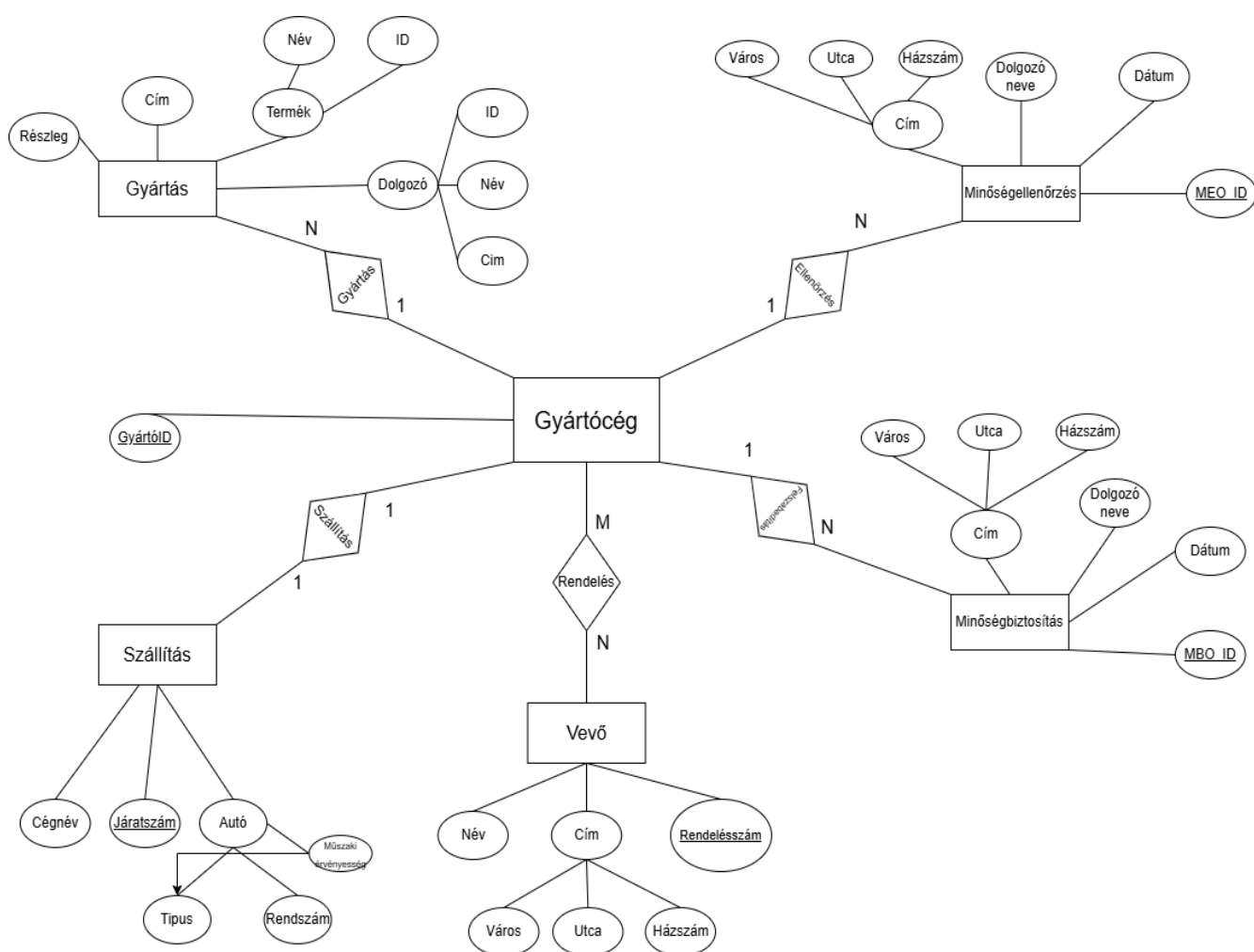
1.1 Az adatbázis ER modell tervezése

Alább látható az elkészített ER modell képe.

Tervezéskor figyelembe vettem a gyártócég és a hozzá kapcsolódó osztályok (egyedek) tevékenységét. A gyártás, minőségellenőrzés, minőségbiztosítás 1:N kapcsolatban van mivel 1 gyártócégen belül több gyártó részleg is létezhet.

A gyártócég által gyártott termékeket több vevő is rendelheti ezért őket N:M több – több kapcsolatban vannak.

A gyártócég és a szállítás azt feltételezve, hogy 1 terméket csak egy szállítóval forgalmaznak így őket 1:1 kapcsolatba vannak.



1.2 Az adatbázis konvertálása XDM modellre

Alább látható az elkészített XDM modell képe.

A modell alapján minden egyed a gyártócégalatt helyezkedik el. A gyártócégalatti kódokkal rendelkezik, ami *GyártóID* elnevezést kapta.

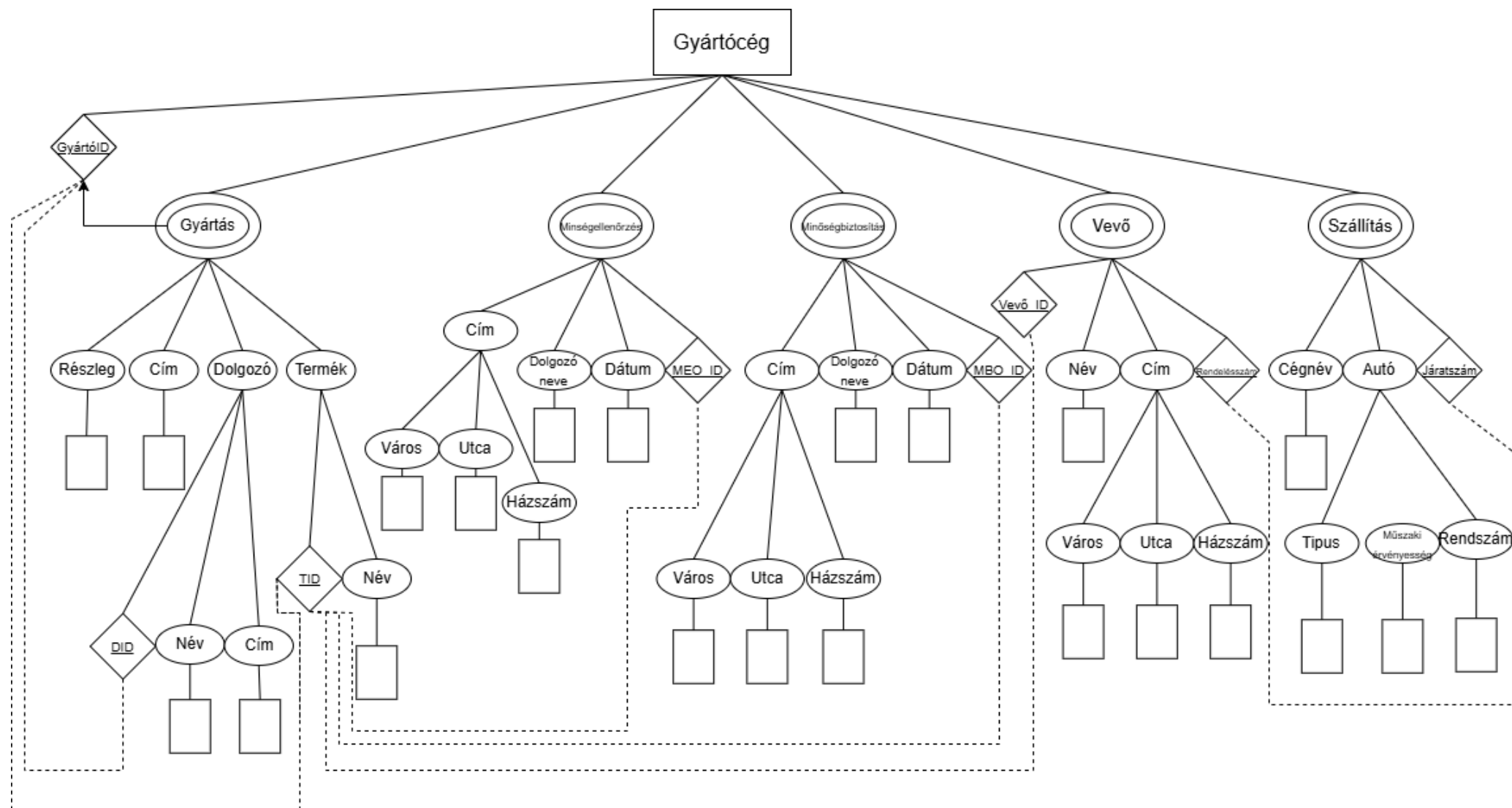
A **gyártáshoz** tartozik a részleg, cím, dolgozó (akinek van egyedi azonosítója: *DID*) és a gyártott termék, aminek van neve és egyedi azonosítója: *TID*.

A **minőségellenőrzéshez** tartozik egy cím (azon belül város, utca, házszám), dolgozó neve, dátum majd egy egyedi azonosító (*MEO_ID*) ami a termék azonosítóval össze van kötve (vagyis az adott gyártott terméket az a dolgozó ellenőrizte).

A **minőségbiztosításhoz** tartozik egy cím (azon belül város, utca, házszám), dolgozó neve, dátum majd egy egyedi azonosító (*MBO_ID*) ami a termék azonosítóval össze van kötve (vagyis az adott gyártott terméket az a dolgozó ellenőrizte).

A **vevőhöz** tartozik egyedi azonosító: *vevő_ID*, név, cím (azon belül: város, utca, házszám) és rendelés azonosító: *rendelészám*. A vevő azonosító össze van kötve a termék azonosítóval azzal a feltételezéssel, hogy egy adott vevő egy típusú terméket akar megvásárolni.

A **szállításhoz** tartozik cégnév, autó (azon belül: típus, műszaki érvényessége, rendszám) és egy egyedi azonosító: *járatszám* ami össze van kötve a vevővel.



1.3 Az XDM modell alapján XML dokumentum készítése

Az XDM modell alapján elkészített XML dokumentum forráskódja alább látható. Az XDM modellben meghatározott egyedi azonosító attributumként rögzítésre kerültek az XML dokumentumban.

```
<?xml version="1.0" encoding="utf-8"?>

<GTE1Y4_WebXMLSemTask xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="XMLSchemaGTE1Y4.xsd">
    <gyartoceg gyartoID="gy1">
        <!-- gyártó részlegek adatai beleértve a dolgozók adatait és a termékek
adatait -->
        <gyartas>
            <reszleg>Alapaanyag gyártó részleg</reszleg>
            <cim>Budapest, Simon utca 2.</cim>
            <dolgozo DID="d1">
                <nev>Kiss István</nev>
                <cim>Budapest, Kiss utca 2.</cim>
            </dolgozo>
            <dolgozo DID="d2">
                <nev>Kiss György</nev>
                <cim>Budapest, György utca 2.</cim>
            </dolgozo>
            <termek TID="t1" meo_id="meo1" mbo_id="mbo1" vevoID="vevo1">
                <nev>Koleszterin szint csökkentő</nev>
            </termek>
            <termek TID="t2" meo_id="meo2" mbo_id="mbo2" vevoID="vevo2">
                <nev>Fájdalom csillapító</nev>
            </termek>
        </gyartas>
        <gyartas>
            <reszleg>Injekció gyártó részleg</reszleg>
            <cim>Budapest, Injekció utca 2.</cim>
            <dolgozo DID="d3">
                <nev>Piros István</nev>
                <cim>Budapest, Piros utca 2.</cim>
            </dolgozo>
            <dolgozo DID="d4">
                <nev>Kék György</nev>
                <cim>Budapest, Kék utca 2.</cim>
            </dolgozo>
            <termek TID="t3" meo_id="meo3" mbo_id="mbo1" vevoID="vevo3">
                <nev>Covid-19 oltás</nev>
            </termek>
            <termek TID="t4" meo_id="meo1" mbo_id="mbo2" vevoID="vevo1">
                <nev>Fájdalom csillapító</nev>
            </termek>
        </gyartas>
        <gyartas>
            <reszleg>Infúzió gyártó részleg</reszleg>
            <cim>Budapest, Infúzió utca 2.</cim>
            <dolgozo DID="d5">
```

```
<nev>Gibsz István</nev>
<cim>Budapest, Gibsz utca 2.</cim>
</dolgozo>
<dolgozo DID="d6">
  <nev>Gibsz Jakab</nev>
  <cim>Budapest, Jakab utca 2.</cim>
</dolgozo>
<termek TID="t5" meo_id="meo2" mbo_id="mbo1" vevoID="vevo2">
  <nev>Sóoldat</nev>
</termek>
<termek TID="t6" meo_id="meo3" mbo_id="mbo2" vevoID="vevo3">
  <nev>Cukoroldat</nev>
</termek>
</gyartas>
<!-- minőségellenőrzés adatai-->
<minosegellenorzes meo_id="meo1">
  <cim>
    <varos>Gödöllő</varos>
    <utca>Gödöllő utca</utca>
    <hazszam>10</hazszam>
  </cim>
  <dolgozo_neve>Kiss Jánosné</dolgozo_neve>
  <datum>2025.05.11</datum>
</minosegellenorzes>
<minosegellenorzes meo_id="meo2">
  <cim>
    <varos>Budapest</varos>
    <utca>Budapest utca</utca>
    <hazszam>10</hazszam>
  </cim>
  <dolgozo_neve>Virág Róbertné</dolgozo_neve>
  <datum>2025.05.10</datum>
</minosegellenorzes>
<minosegellenorzes meo_id="meo3">
  <cim>
    <varos>Vásárosnamény</varos>
    <utca>Tétény utca</utca>
    <hazszam>10</hazszam>
  </cim>
  <dolgozo_neve>Gibsz Jakabné</dolgozo_neve>
  <datum>2025.05.11</datum>
</minosegellenorzes>
<!-- minőségbiztosítás adatai-->
<minosegbiztositas mbo_id="mbo1">
  <cim>
    <varos>Vásárosnamény</varos>
    <utca>Kökény utca</utca>
    <hazszam>20</hazszam>
  </cim>
  <dolgozo_neve>Veres Andrásné</dolgozo_neve>
  <datum>2025.05.15</datum>
```



```
</minosegbiztositas>
<minosegbiztositas mbo_id="mbo2">
  <cim>
    <varos>Budapest</varos>
    <utca>Hadak utca</utca>
    <hazszam>2</hazszam>
  </cim>
  <dolgozo_neve>Kapirsák József</dolgozo_neve>
  <datum>2025.04.20</datum>
</minosegbiztositas>
<!-- vevők adatai-->
<vevo vevoID="vevo1" rendelesszam="rendeles1" jaratszam="szallitas1">
  <nev>Astra Zeneca</nev>
  <cim>
    <varos>Budapest</varos>
    <utca>Alíz utca</utca>
    <hazszam>4</hazszam>
  </cim>
</vevo>
<vevo vevoID="vevo2" rendelesszam="rendeles2" jaratszam="szallitas2">
  <nev>Pfizer</nev>
  <cim>
    <varos>Budapest</varos>
    <utca>Alkotás utca</utca>
    <hazszam>53</hazszam>
  </cim>
</vevo>
<vevo vevoID="vevo3" rendelesszam="rendeles3" jaratszam="szallitas3">
  <nev>Richter Gedeon</nev>
  <cim>
    <varos>Budapest</varos>
    <utca>Gyömrői utca</utca>
    <hazszam>19</hazszam>
  </cim>
</vevo>
<!-- szállítás adatai-->
<szallitas jaratszam = "szallitas1">
  <cegnev>GLS Futárszolgálat</cegnev>
  <auto>
    <tipus>Ford</tipus>
    <muszaki_ervenyesseg>2025.10.10</muszaki_ervenyesseg>
    <rendszam>XX-YY-312</rendszam>
  </auto>
</szallitas>
<szallitas jaratszam = "szallitas2">
  <cegnev>Kumi Futárszolgálat</cegnev>
  <auto>
    <tipus>Peugeot</tipus>
    <muszaki_ervenyesseg>2026.06.10</muszaki_ervenyesseg>
    <rendszam>YY-ZZ-512</rendszam>
  </auto>
```

```

</szallitas>
<szallitas jaratszam = "szallitas3">
  <cegnev>DPD Futárszolgálat</cegnev>
  <auto>
    <tipus>Opel</tipus>
    <muszaki_ervenyesseg>2027.06.10</muszaki_ervenyesseg>
    <rendszer>ZZ-WW-912</rendszer>
  </auto>
</szallitas>
</gyartoceg>
</GTE1Y4_WebXMLSemTask>

```

1.4 Az XML dokumentum alapján XMLSchema készítése

Az gyartoceg elem a fő entitás, amely több gyartas elemet tartalmaz, ahol a részlegek, dolgozók és termékek szerepelnek. A termékekhez minőségellenőrzési és minőségbiztosítási attribútumok kapcsolódnak. Emellett a séma tartalmazza a vevo és szallitas elemeket is, amelyek az üzleti folyamat további aspektusait reprezentálják.

Az XSD séma biztosítja, hogy az XML dokumentum megfeleljen a meghatározott adatszerkezetnek, így könnyen feldolgozható és integrálható más rendszerekkel. Ez a megvalósítás segíti az adatok pontos tárolását és hatékony visszakeresését.

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GTE1Y4_WebXMLSemTask">
    <xs:complexType>
      <xs:sequence>
        <!-- A fő struktúra, amely az egész gyártó céget leírja-->
        <xs:element name="gyartoceg" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="gyartas" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <!-- Gyártási részlegk amely a dolgozókat és a termékeket is tartalmazza-->
                    <xs:element name="reszleg" type="xs:string"/>
                    <xs:element name="cim" type="xs:string"/>
                    <!-- dolgozók adatai-->
                    <xs:element name="dolgozo" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="nev" type="xs:string"/>
                          <xs:element name="cim" type="xs:string"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:attribute name="DID" type="xs:string" use="required"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```
<!-- termékek adatai-->
<xs:element name="termek" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nev" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="TID" type="xs:string" use="required"/>
    <xs:attribute name="meo_id" type="xs:string" use="required"/>
    <xs:attribute name="mbo_id" type="xs:string" use="required"/>
    <xs:attribute name="vevoID" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- minőségellenőrzés adatai-->
<xs:element name="minosegellenorzes" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="cim">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="varos" type="xs:string"/>
            <xs:element name="utca" type="xs:string"/>
            <xs:element name="hazszam" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="dolgozo_neve" type="xs:string"/>
      <xs:element name="datum" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="meo_id" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<!-- minőségbiztosítás adatai-->
<xs:element name="minosegbiztositas" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="cim">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="varos" type="xs:string"/>
            <xs:element name="utca" type="xs:string"/>
            <xs:element name="hazszam" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="dolgozo_neve" type="xs:string"/>
      <xs:element name="datum" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="mbo_id" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
```

```

        </xs:complexType>
    </xs:element>
    <!-- vevő adatai-->
    <xs:element name="vevo" maxOccurs="unbounded">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="nev" type="xs:string"/>
                <xs:element name="cim">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="varos" type="xs:string"/>
                            <xs:element name="utca" type="xs:string"/>
                            <xs:element name="hazszam" type="xs:string"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="vevoID" type="xs:string" use="required"/>
            <xs:attribute name="rendelesszam" type="xs:string" use="required"/>
            <xs:attribute name="jaratszam" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <!-- szállítás adatai-->
    <xs:element name="szallitas" maxOccurs="unbounded">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="cegnev" type="xs:string"/>
                <xs:element name="auto">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="tipus" type="xs:string"/>
                            <xs:element name="muszaki_ervenyesség" type="xs:string"/>
                            <xs:element name="rendszer" type="xs:string"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="jaratszam" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:sequence>
<xs:attribute name="gyartoID" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Az XML dokumentum validálásának eredmény Oxygen Editor segítségével:

D:\...\2025\GTE1Y4WebXML_LEV\WebXMLSemTaskGTE1Y4\XMLGTE1Y4.xml

Validation successful

2. feladat

```
package hu.dom.parseGTE1Y4;

import java.io.File;
import java.io.PrintWriter;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DomReadGTE1Y4 {

    public static void main(String[] args) {

        try {
            //fájl beolvasás
            File xmlFile = new File("D:\\THE -Programtervezo
info\\6_felev\\Webprogramozás\\2025\\GTE1Y4WebXML_LEV\\WebXMLSemTaskGTE1Y4\\XMLGTE1Y4.xml");
            //XML dokumentum feldolgozása
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            //Az XML dokumentum DOM struktúrába rendezése
            Document doc = builder.parse(xmlFile);
            //A dokumentum normalizálása
            doc.getDocumentElement().normalize();

            //Részlegek kiírása konzolra -> Külön függvény
            System.out.println("=====GYÁRTÁS=====");
            gyartas(doc);
            //Minőségellenőrzés adatainak kiírása konzolra -> Külön függvény
            System.out.println("=====MINŐSÉGELLENŐRZÉS=====");
            minosegellenorzes(doc);
            //Minőségbiztosítás adatainak kiírása konzolra -> Külön függvény
            System.out.println("=====MINŐSÉGBIZTOSÍTÁS=====");
            minosegbiztositas(doc);
            //Vevő adatainak kiírása konzolra -> Külön függvény
            System.out.println("=====VEVŐ=====");
            vevo(doc);
            //Szállítási adatainak kiírása konzolra -> Külön függvény
            System.out.println("=====SZÁLLÍTÁS=====");
            szallitas(doc);
            //Rendelések kilistázása konzolra + a listázott adatok a DomReadGTE1Y4 gyökérkönyvtárba
            "rendelesek_output.txt" fájlba kiírása -> külön függvény
            System.out.println("=====RENDELÉSEK
LISTÁZÁSA=====");
            try (PrintWriter writer = new PrintWriter("rendelesek_output.txt")) {
                listazas(doc, writer);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}

public static void gyartas (Document doc) {

    //Részlegek neve és címe
    //XML dokumentum lekérdezi az összes olyan elemet aminek a tagneve "gyartas" és
    listába rendezzi
    NodeList reszlegList = doc.getElementsByTagName("gyartas");
    //végigmegyünk a lista elemein
    for(int i = 0; i < reszlegList.getLength(); i++) {
        //Lekéri a for ciklus alapján az aktuális lista elemet
        Node gyartasNode = reszlegList.item(i);

        //ellenőrzi, hogy a kapott elem az valóban egy XML elem
        if(gyartasNode.getNodeType() == Node.ELEMENT_NODE) {

            //Az elem átalakítása Element típusú
            Element reszlegElem = (Element) gyartasNode;

            //Az aktuális részleg elem nevének lekérése
            String reszleg =
            reszlegElem.getElementsByTagName("reszleg").item(0).getTextContent();
            //Az aktuális részleg elem címének a lekérése
            String cim =
            reszlegElem.getElementsByTagName("cim").item(0).getTextContent();

            //A lekért név és cím adatok kiíratása konzolra
            System.out.println("*****");
            System.out.println("Részleg neve: " + reszleg.toUpperCase());
            System.out.println("Részleg címe: " + cim);
            System.out.println();

            //Dolgozók kiírítása a részlegeken belül => ebben a formában "block-ént
            jeleníti meg az adatokat
            NodeList dolgozoList = reszlegElem.getElementsByTagName("dolgozo");
            for(int j = 0; j < dolgozoList.getLength(); j++) {
                Node dolgozoNode = dolgozoList.item(j);

                if(dolgozoNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element dolgozoElem = (Element) dolgozoNode;

                    String dolgozo_nev =
                    dolgozoElem.getElementsByTagName("nev").item(0).getTextContent();
                    String dolgozo_cim =
                    dolgozoElem.getElementsByTagName("cim").item(0).getTextContent();
                    String dolgozo_ID = dolgozoElem.getAttribute("DID");

                    System.out.println("Dolgozó ID: " + dolgozo_ID);
                    System.out.println("Dolgozó neve: " + dolgozo_nev);
                    System.out.println("Dolgozó címe: " + dolgozo_cim);
                    System.out.println();

                }
            }

            //Termékek kiírítása amiket a részlegek gyártanak
            NodeList termékList = reszlegElem.getElementsByTagName("termek");
            for(int k = 0; k < termékList.getLength(); k++) {
                Node termékNode = termékList.item(k);

                if(termekNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element termékElem = (Element) termékNode;

                    String termék_nev =
                    termékElem.getElementsByTagName("nev").item(0).getTextContent();
                    String termék_ID = termékElem.getAttribute("TID");

                    System.out.println("Termék ID: " + termék_ID);
                    System.out.println("Termék neve: " + termék_nev);
                }
            }
        }
    }
}
```

```
        System.out.println();
    }
}

}

}

}

//Minőségellenőrök kilistázása
public static void minosegellenorzes(Document doc) {
    NodeList minosegellList = doc.getElementsByTagName("minosegellenorzes");
    for(int i = 0; i < minosegellList.getLength(); i++) {
        Node minosegellNode = minosegellList.item(i);
        if(minosegellNode.getNodeType() == Node.ELEMENT_NODE) {
            Element minosegellElem = (Element) minosegellNode;

            String minosegellID = minosegellElem.getAttribute("meo_id");
            String nev =
minosegellElem.getElementsByTagName("dolgozo_neve").item(0).getTextContent();
            String cim =
minosegellElem.getElementsByTagName("varos").item(0).getTextContent() + ", " +
minosegellElem.getElementsByTagName("utca").item(0).getTextContent() +
" " +
minosegellElem.getElementsByTagName("hazszam").item(0).getTextContent() + ".";

            System.out.println("Minőségellenőr ID: " + minosegellID);
            System.out.println("Minőségellenőr neve: " + nev);
            System.out.println("Minőségellenőr címe: " + cim);
            System.out.println();
        }
    }
}

//Minőségbiztosítók kilistázása
public static void minosegbiztositas(Document doc) {
    NodeList minosegbizList = doc.getElementsByTagName("minosegbiztositas");
    for(int i = 0; i < minosegbizList.getLength(); i++) {
        Node minosegbizNode = minosegbizList.item(i);
        if(minosegbizNode.getNodeType() == Node.ELEMENT_NODE) {
            Element minosegbizElem = (Element) minosegbizNode;

            String minosegbizID = minosegbizElem.getAttribute("mbo_id");
            String nev =
minosegbizElem.getElementsByTagName("dolgozo_neve").item(0).getTextContent();
            String cim =
minosegbizElem.getElementsByTagName("varos").item(0).getTextContent() + ", " +
minosegbizElem.getElementsByTagName("utca").item(0).getTextContent() +
" " +
minosegbizElem.getElementsByTagName("hazszam").item(0).getTextContent() + ".";

            System.out.println("Minőségbiztosító ID: " + minosegbizID);
            System.out.println("Minőségbiztosító neve: " + nev);
            System.out.println("Minőségbiztosító címe: " + cim);
            System.out.println();
        }
    }
}

//Vevők kilistázása
public static void vevo(Document doc) {
    NodeList vevoList = doc.getElementsByTagName("vevo");
    for(int i = 0; i < vevoList.getLength(); i++) {
        Node vevoNode = vevoList.item(i);
        if(vevoNode.getNodeType() == Node.ELEMENT_NODE) {
            Element vevoElem = (Element) vevoNode;

            String vevoID = vevoElem.getAttribute("vevoID");
            String nev =
vevoElem.getElementsByTagName("nev").item(0).getTextContent();

```

```

        String cim =
vevoElem.getElementsByTagName("varos").item(0).getTextContent() + ", " +
vevoElem.getElementsByTagName("utca").item(0).getTextContent() +
        " " +
vevoElem.getElementsByTagName("hazszam").item(0).getTextContent() + ".";
        String rendelesID = vevoElem.getAttribute("rendelesszam");
        String jaratID = vevoElem.getAttribute("jaratszam");

        System.out.println("Vevő ID: " + vevoID);
        System.out.println("Vevő neve: " + nev);
        System.out.println("Vevő címe: " + cim);
        System.out.println();
    }
}

//Szállítók kilistázása
public static void szallitas(Document doc) {
    NodeList szallitoList = doc.getElementsByTagName("szallitas");
    for(int i = 0; i < szallitoList.getLength(); i++) {
        Node szallitasNode = szallitoList.item(i);
        if(szallitasNode.getNodeType() == Node.ELEMENT_NODE) {
            Element szallitasElem = (Element) szallitasNode;

            String szallitasID = szallitasElem.getAttribute("jaratszam");
            String nev =
szallitasElem.getElementsByTagName("cegnev").item(0).getTextContent();
            String tipus =
szallitasElem.getElementsByTagName("tipus").item(0).getTextContent();
            String muszaki =
szallitasElem.getElementsByTagName("muszaki_ervenyesség").item(0).getTextContent();
            String rendszam =
szallitasElem.getElementsByTagName("rendszam").item(0).getTextContent();

            System.out.println("Szállítás kódja: " + szallitasID);
            System.out.println("Szállító cég neve: " + nev);
            System.out.println("  Autó adatai: ");
            System.out.println("      Tipus: " + tipus);
            System.out.println("      Műszaki érvényessége: " + muszaki);
            System.out.println("      Rendszám: " + rendszam);
            System.out.println();
        }
    }
}

//Rendelések listázása
public static void listazas(Document doc, PrintWriter writer) {
    //Gyártás elemek lekérése az XML dokumentumból
    NodeList gyartasok = doc.getElementsByTagName("gyartas");

    for (int i = 0; i < gyartasok.getLength(); i++) {
        //Egy adott gyártási elem beolvasása
        Element gyartas = (Element) gyartasok.item(i);
        //A beolvasott gyártási elem kiolvasása és kiírása
        String reszleg = gyartas.getElementsByTagName("reszleg").item(0).getTextContent();
        String cim = gyartas.getElementsByTagName("cim").item(0).getTextContent();

        Log("Részleg: " + reszleg.toUpperCase(), writer);
        Log("Cím: " + cim, writer);
        Log("\nDolgozók:", writer);

        //A gyártó részleg beolvasása alapján a dolgozók listázása
        NodeList dolgozok = gyartas.getElementsByTagName("dolgozo");
        for (int j = 0; j < dolgozok.getLength(); j++) {
            Element dolgozo = (Element) dolgozok.item(j);
            String nev = dolgozo.getElementsByTagName("nev").item(0).getTextContent();
            String dcim = dolgozo.getElementsByTagName("cim").item(0).getTextContent();
            Log("- " + nev + " (" + dcim + ")", writer);
        }
    }
}

```



```
Log("\nTermékek:", writer);
//A gyártó részleg beolvasása alapján a termékek listázása
NodeList termekek = gyartas.getElementsByTagName("termek");
for (int k = 0; k < termekek.getLength(); k++) {
    Element termek = (Element) termekek.item(k);
    String termekNev = termek.getElementsByTagName("nev").item(0).getTextContent();
    String tid = termek.getAttribute("TID");
    String meo_id = termek.getAttribute("meo_id");
    String mbo_id = termek.getAttribute("mbo_id");
    String vevoID = termek.getAttribute("vevoID");

    Log(" - Termék: " + termekNev, writer);
    Log("   Termék ID: " + tid, writer);

    // Minőségellenőrzés
    Element meo = findElementById(doc, "minosegellenorzes", "meo_id", meo_id);
    if (meo != null) {
        Element cimElem = (Element) meo.getElementsByTagName("cim").item(0);
        String dolgozoNev =
meo.getElementsByTagName("dolgozo_neve").item(0).getTextContent();
        String datum = meo.getElementsByTagName("datum").item(0).getTextContent();
        String varos =
cimElem.getElementsByTagName("varos").item(0).getTextContent();
        String utca = cimElem.getElementsByTagName("utca").item(0).getTextContent();
        String hazzsam =
cimElem.getElementsByTagName("hazzsam").item(0).getTextContent();
        Log("   MEO: " + dolgozoNev + ", " + datum + ", " + varos + ", " + utca + "
" + hazzsam, writer);
    }

    // Minőségbiztosítás
    Element mbo = findElementById(doc, "minosegbiztositas", "mbo_id", mbo_id);
    if (mbo != null) {
        Element cimElem = (Element) mbo.getElementsByTagName("cim").item(0);
        String dolgozoNev =
mbo.getElementsByTagName("dolgozo_neve").item(0).getTextContent();
        String datum = mbo.getElementsByTagName("datum").item(0).getTextContent();
        String varos =
cimElem.getElementsByTagName("varos").item(0).getTextContent();
        String utca = cimElem.getElementsByTagName("utca").item(0).getTextContent();
        String hazzsam =
cimElem.getElementsByTagName("hazzsam").item(0).getTextContent();
        Log("   MBO: " + dolgozoNev + ", " + datum + ", " + varos + ", " + utca + "
" + hazzsam, writer);
    }

    // Vevő + szállítás
    Element vevo = findElementById(doc, "vevo", "vevoID", vevoID);
    if (vevo != null) {
        String nev = vevo.getElementsByTagName("nev").item(0).getTextContent();
        Element cimElem = (Element) vevo.getElementsByTagName("cim").item(0);
        String varos =
cimElem.getElementsByTagName("varos").item(0).getTextContent();
        String utca = cimElem.getElementsByTagName("utca").item(0).getTextContent();
        String hazzsam =
cimElem.getElementsByTagName("hazzsam").item(0).getTextContent();
        String jaratszam = vevo.getAttribute("jaratszam");
        Log("   Vevő: " + nev + " - " + varos + ", " + utca + " " + hazzsam,
writer);

        Element szallitas = findElementById(doc, "szallitas", "jaratszam",
jaratszam);

        if (szallitas != null) {
            String cegnev =
szallitas.getElementsByTagName("cegnev").item(0).getTextContent();
            Element auto = (Element) szallitas.getElementsByTagName("auto").item(0);
            String tipus =
auto.getElementsByTagName("tipus").item(0).getTextContent();

```

```
        String muszaki =
auto.getElementsByTagName("muszaki_ervenyesseg").item(0).getTextContent();
        String rendszam =
auto.getElementsByTagName("rendszam").item(0).getTextContent();
        Log("    Szállítás: " + cegnev + ", " + tipus + ", Rendszám: " + rendszam
+ ", műszaki: " + muszaki, writer);
        Log("", writer);
    }
}
}
}

//A megadott XML dokumentumban keres egy elemet az adott címkével és attributum értékkel
private static Element findElementById(Document doc, String tagName, String attrName, String
id) {
    //Lekérdezi az XML dokumentumból a "tagName"-ben meghatározott adatokat
    NodeList list = doc.getElementsByTagName(tagName);
    //végigmegy a listán
    for (int i = 0; i < list.getLength(); i++) {
        Element elem = (Element) list.item(i);
        //Ha az elem attributuma megegyezik a keresett ID-val akkor visszatér vele
        if (elem.getAttribute(attrName).equals(id)) {
            return elem;
        }
    }
    //egyébként üres értékkel tér vissza
    return null;
}

public static void log(String text, PrintWriter writer) {
    //szöveg kiírása konzolra
    System.out.println(text);
    //Ha a writer nem 0 akkor kiírja fájlba
    if (writer != null) {
        writer.println(text);
    }
}
}
```

Lefutás eredménye:

=====GYÁRTÁS=====

Részleg neve: ALAPAANYAG GYÁRTÓ RÉSZLEG

Részleg címe: Budapest, Simon utca 2.

Dolgozó ID: d1

Dolgozó neve: Kiss István

Dolgozó címe: Budapest, Kiss utca 2.

Dolgozó ID: d2

Dolgozó neve: Kiss György

Dolgozó címe: Budapest, György utca 2.

Termék ID: t1

Termék neve: Koleszterin szint csökkentő

Termék ID: t2

Termék neve: Fájdalom csillapító

Részleg neve: INJEKCIÓ GYÁRTÓ RÉSZLEG

Részleg címe: Budapest, Injekció utca 2.

Dolgozó ID: d3

Dolgozó neve: Piros István

Dolgozó címe: Budapest, Piros utca 2.

Dolgozó ID: d4

Dolgozó neve: Kék György

Dolgozó címe: Budapest, Kék utca 2.

Termék ID: t3

Termék neve: Covid-19 oltás

Termék ID: t4

Termék neve: Fájdalom csillapító

Részleg neve: INFÚZIÓ GYÁRTÓ RÉSZLEG

Részleg címe: Budapest, Infúzió utca 2.

Dolgozó ID: d5

Dolgozó neve: Gibsz István

Dolgozó címe: Budapest, Gibsz utca 2.

Dolgozó ID: d6

Dolgozó neve: Gibsz Jakab

Dolgozó címe: Budapest, Jakab utca 2.

Termék ID: t5

Termék neve: Sóoldat

Termék ID: t6

Termék neve: Cukoroldat

=====MINŐSÉGELLENŐRZÉS=====

Minőségellenőrző ID: meo1

Minőségellenőrző neve: Kiss Jánosné

Minőségellenőrző címe: Gödöllő, Gödöllő utca 10.

Minőségellenőrző ID: meo2
Minőségellenőrző neve: Virág Róbertné
Minőségellenőrző címe: Budapest, Budapest utca 10.

Minőségellenőrző ID: meo3
Minőségellenőrző neve: Gibbs Jakabné
Minőségellenőrző címe: Vásárosnamény, Tétény utca 10.

=====MINŐSÉGBIZTOSÍTÁS=====

Minőségbiztosító ID: mbo1
Minőségbiztosító neve: Veres Andrásné
Minőségbiztosító címe: Vásárosnamény, Kökény utca 20.

Minőségbiztosító ID: mbo2
Minőségbiztosító neve: Kapirsák József
Minőségbiztosító címe: Budapest, Hadak utca 2.

=====VEVŐ=====

Vevő ID: vevo1
Vevő neve: Astra Zeneca
Vevő címe: Budapest, Alíz utca 4.

Vevő ID: vevo2
Vevő neve: Pfizer
Vevő címe: Budapest, Alkotás utca 53.

Vevő ID: vevo3
Vevő neve: Richter Gedeon
Vevő címe: Budapest, Gyömrői utca 19.

=====SZÁLLÍTÁS=====

Szállítás kódja: szallitas1
Szállító cég neve: GLS Futárszolgálat

Autó adatai:

Tipus: Ford
Műszaki érvényessége: 2025.10.10
Rendszám: XX-YY-312

Szállítás kódja: szallitas2
Szállító cég neve: Kumi Futárszolgálat

Autó adatai:

Tipus: Peugeot
Műszaki érvényessége: 2026.06.10
Rendszám: YY-ZZ-512

Szállítás kódja: szallitas3
Szállító cég neve: DPD Futárszolgálat

Autó adatai:

Tipus: Opel
Műszaki érvényessége: 2027.06.10
Rendszám: ZZ-WW-912

=====RENDELÉSEK LISTÁZÁSA=====

Részleg: ALAPAANYAG GYÁRTÓ RÉSZLEG
Cím: Budapest, Simon utca 2.

Dolgozók:

- Kiss István (Budapest, Kiss utca 2.)
- Kiss György (Budapest, György utca 2.)

Termékek:

- Termék: Koleszterin szint csökkentő

Termék ID: t1

ME0: Kiss Jánosné, 2025.05.11, Gödöllő, Gödöllő utca 10

MB0: Veres Andrásné, 2025.05.15, Vásárosnamény, Kökény utca 20

Vevő: Astra Zeneca – Budapest, Alíz utca 4

Szállítás: GLS Futárszolgálat, Ford, Rendszám: XX-YY-312, műszaki: 2025.10.10

- Termék: Fájdalom csillapító

Termék ID: t2

ME0: Virág Róbertné, 2025.05.10, Budapest, Budapest utca 10

MB0: Kapirsák József, 2025.04.20, Budapest, Hadak utca 2

Vevő: Pfizer – Budapest, Alkotás utca 53

Szállítás: Kumi Futárszolgálat, Peugeot, Rendszám: YY-ZZ-512, műszaki: 2026.06.10

Részleg: INJEKCIÓ GYÁRTÓ RÉSZLEG

Cím: Budapest, Injekció utca 2.

Dolgozók:

- Piros István (Budapest, Piros utca 2.)

- Kék György (Budapest, Kék utca 2.)

Termékek:

- Termék: Covid-19 oltás

Termék ID: t3

ME0: Gibsz Jakabné, 2025.05.11, Vásárosnamény, Tétény utca 10

MB0: Veres Andrásné, 2025.05.15, Vásárosnamény, Kökény utca 20

Vevő: Richter Gedeon – Budapest, Gyömrői utca 19

Szállítás: DPD Futárszolgálat, Opel, Rendszám: ZZ-WW-912, műszaki: 2027.06.10

- Termék: Fájdalom csillapító

Termék ID: t4

ME0: Kiss Jánosné, 2025.05.11, Gödöllő, Gödöllő utca 10

MB0: Kapirsák József, 2025.04.20, Budapest, Hadak utca 2

Vevő: Astra Zeneca – Budapest, Alíz utca 4

Szállítás: GLS Futárszolgálat, Ford, Rendszám: XX-YY-312, műszaki: 2025.10.10

Részleg: INFÚZIÓ GYÁRTÓ RÉSZLEG

Cím: Budapest, Infúzió utca 2.

Dolgozók:

- Gibsz István (Budapest, Gibsz utca 2.)

- Gibsz Jakab (Budapest, Jakab utca 2.)

Termékek:

- Termék: Sóoldat

Termék ID: t5

ME0: Virág Róbertné, 2025.05.10, Budapest, Budapest utca 10

MB0: Veres Andrásné, 2025.05.15, Vásárosnamény, Kökény utca 20

Vevő: Pfizer – Budapest, Alkotás utca 53

Szállítás: Kumi Futárszolgálat, Peugeot, Rendszám: YY-ZZ-512, műszaki: 2026.06.10

- Termék: Cukoroldat

Termék ID: t6

ME0: Gibsz Jakabné, 2025.05.11, Vásárosnamény, Tétény utca 10

MB0: Kapirsák József, 2025.04.20, Budapest, Hadak utca 2

Vevő: Richter Gedeon – Budapest, Gyömrői utca 19

Szállítás: DPD Futárszolgálat, Opel, Rendszám: ZZ-WW-912, műszaki: 2027.06.10



rendelesek_output.txt