

Partie 2

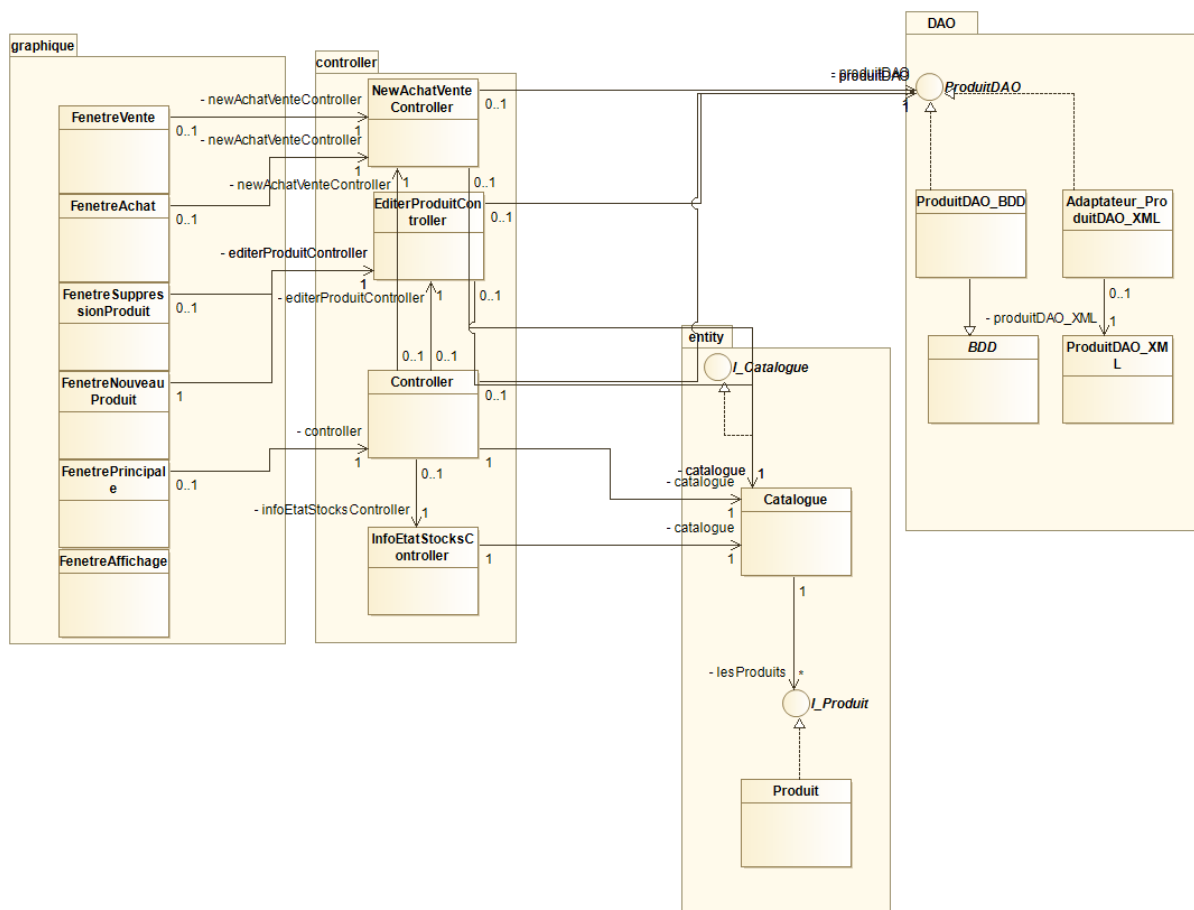
SQL :

```
CREATE TABLE Produits(
    numProduit NUMBER, nomProduit VARCHAR(25), quantiteStock NUMBER, prixProduit
INTEGER,
    CONSTRAINT pk_numProduit PRIMARY KEY (numProduit),
    CONSTRAINT ck_nomProduit UNIQUE (nomProduit),
    CONSTRAINT ck_prix CHECK (prixProduit > 0),
    CONSTRAINT ck_quantite CHECK (quantiteStock >= 0)
);

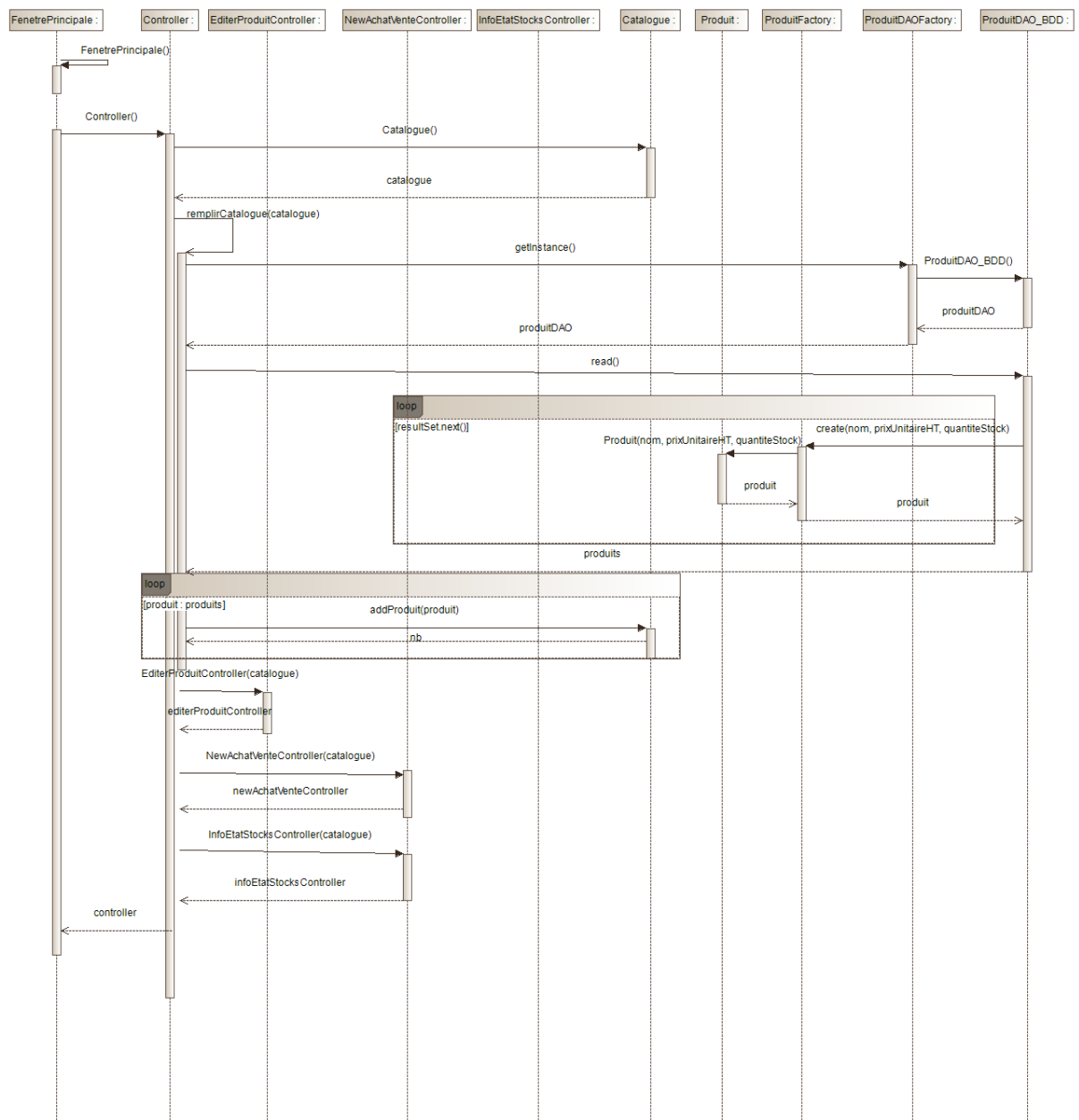
CREATE SEQUENCE sq_produit START WITH 1 INCREMENT BY 1;

CREATE PROCEDURE procedure_add_produit(nomProduit Produits.nomProduit%TYPE,
quantiteStock Produits.quantiteStock%TYPE, prixProduit Produits.prixProduit%TYPE
) AS
BEGIN
    INSERT INTO Produits(numProduit , nomProduit , quantiteStock , prixProduit)
    VALUES (sq_produit.NEXTVAL, nomProduit, quantiteStock, prixProduit);
END;
```

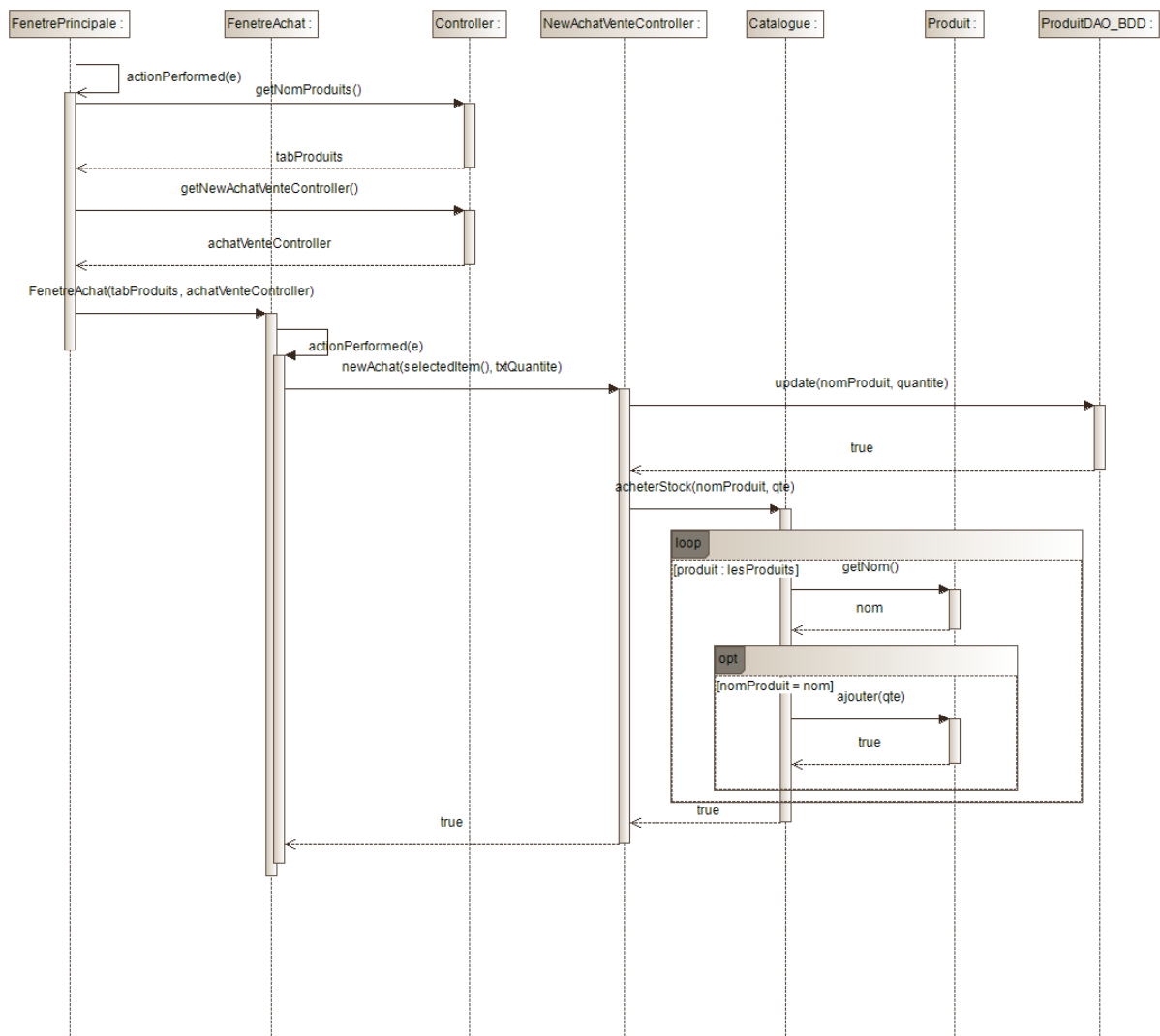
Diagramme paquetage :



Initialisation :



Achat :



Partie 3

SQL :

```

DROP TABLE Produits;
DROP TABLE Catalogue;

CREATE TABLE Catalogue(
    numCatalogue NUMBER, nomCatalogue VARCHAR(25),
    CONSTRAINT pk_numCatalogue PRIMARY KEY (numCatalogue),
    CONSTRAINT ck_nomCatalogue UNIQUE (nomCatalogue)
);

CREATE TABLE Produits(
    numProduit NUMBER, nomProduit VARCHAR(25), quantiteStock NUMBER, prixProduit
INTEGER, numCatalogue NUMBER,
    CONSTRAINT pk_numProduit PRIMARY KEY (numProduit),
    CONSTRAINT fk_numCatalogue_Catalogue FOREIGN KEY (numCatalogue) REFERENCES
Catalogue(numCatalogue) ON DELETE CASCADE,
    CONSTRAINT ck_nomProduit UNIQUE (nomProduit),
    CONSTRAINT ck_prix CHECK (prixProduit > 0),
    CONSTRAINT ck_quantite CHECK (quantiteStock >= 0)
);

CREATE OR REPLACE SEQUENCE sq produit START WITH 1 INCREMENT BY 1;
  
```

```

CREATE OR REPLACE PROCEDURE procedure_add_produit
(p_nomProduit Produits.nomProduit%TYPE, p_quantiteStock
Produits.quantiteStock%TYPE, p_prixProduit Produits.prixProduit%TYPE,
p_nomCatalogue Catalogue.nomCatalogue%TYPE )AS
    v_nomProduit NUMBER;
    v_numCatalogue Catalogue.numCatalogue%TYPE;
BEGIN
    SELECT numCatalogue INTO v_numCatalogue
    FROM Catalogue
    WHERE nomCatalogue=p_nomCatalogue;
    SELECT count(*) INTO v_nomProduit
    FROM Produits
    WHERE nomProduit=p_nomProduit AND numCatalogue=v_numCatalogue;
    IF v_nomProduit = 1 THEN
        RAISE_APPLICATION_ERROR (-20001, 'Ce produit est déjà dans le catalogue!');
    END IF;
    INSERT INTO Produits(numProduit ,nomProduit ,quantiteStock ,prixProduit,
numCatalogue)
    VALUES (sq_produit.NEXTVAL, p_nomProduit, p_quantiteStock, p_prixProduit,
v_numCatalogue);
END;

CREATE OR REPLACE PROCEDURE procedure_update_produit
(p_nomProduit Produits.nomProduit%TYPE, p_quantiteStock
Produits.quantiteStock%TYPE, p_nomCatalogue Catalogue.nomCatalogue%TYPE )AS
    v_numCatalogue Catalogue.numCatalogue%TYPE;
BEGIN
    SELECT numCatalogue INTO v_numCatalogue
    FROM Catalogue
    WHERE nomCatalogue=p_nomCatalogue;
    UPDATE Produits
    SET quantiteStock=quantiteStock+p_quantiteStock
    WHERE nomProduit=p_nomProduit AND numCatalogue=v_numCatalogue;
END;

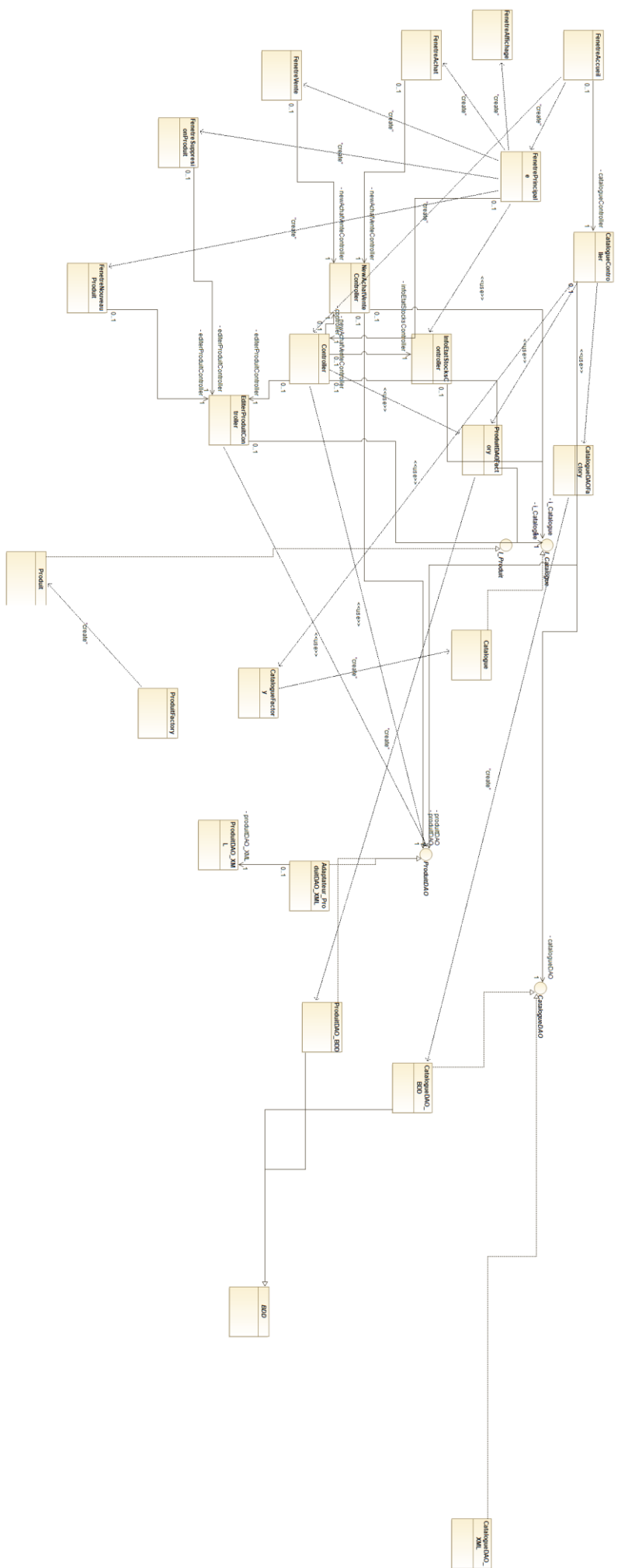
CREATE OR REPLACE PROCEDURE procedure_delete_produit
(p_nomProduit Produits.nomProduit%TYPE, p_nomCatalogue Catalogue.nomCatalogue%TYPE
)AS
    v_numCatalogue Catalogue.numCatalogue%TYPE;
BEGIN
    SELECT numCatalogue INTO v_numCatalogue
    FROM Catalogue
    WHERE nomCatalogue=p_nomCatalogue;
    DELETE FROM Produits
    WHERE nomProduit=p_nomProduit AND numCatalogue=v_numCatalogue;
END;

CREATE OR REPLACE SEQUENCE sq_catalogue START WITH 1 INCREMENT BY 1;

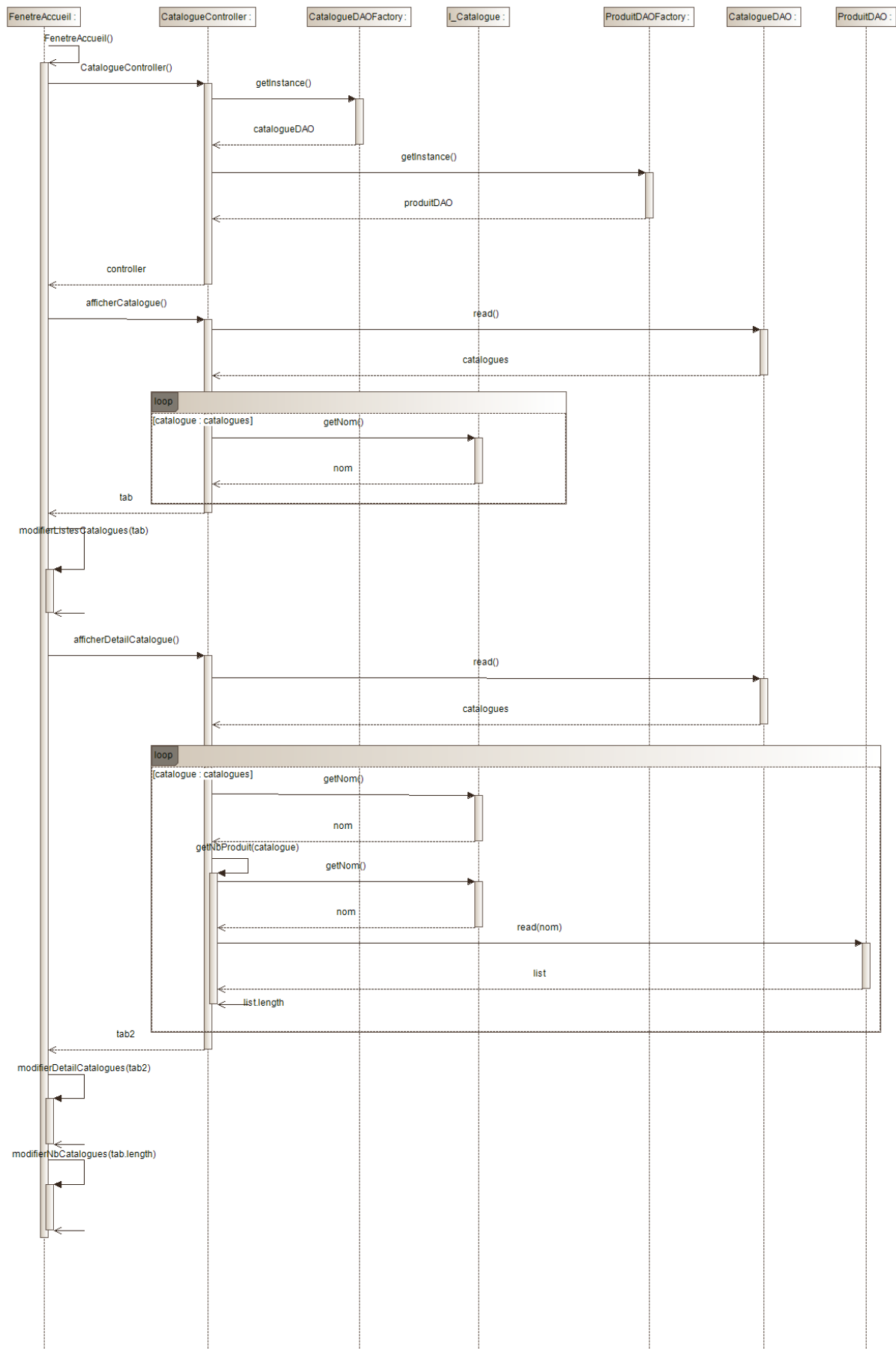
CREATE OR REPLACE PROCEDURE procedure_create_catalogue
(p_nomCatalogue Catalogue.nomCatalogue%TYPE)AS
BEGIN
    INSERT INTO Catalogue(numCatalogue, nomCatalogue)
    VALUES (sq_catalogue.NEXTVAL, p_nomCatalogue);
END;

```

Diagramme de classe :



Initialisation :



Choix catalogue :

