

PROJET INTRODUCTION AUX SYSTÈMES D'INFORMATION

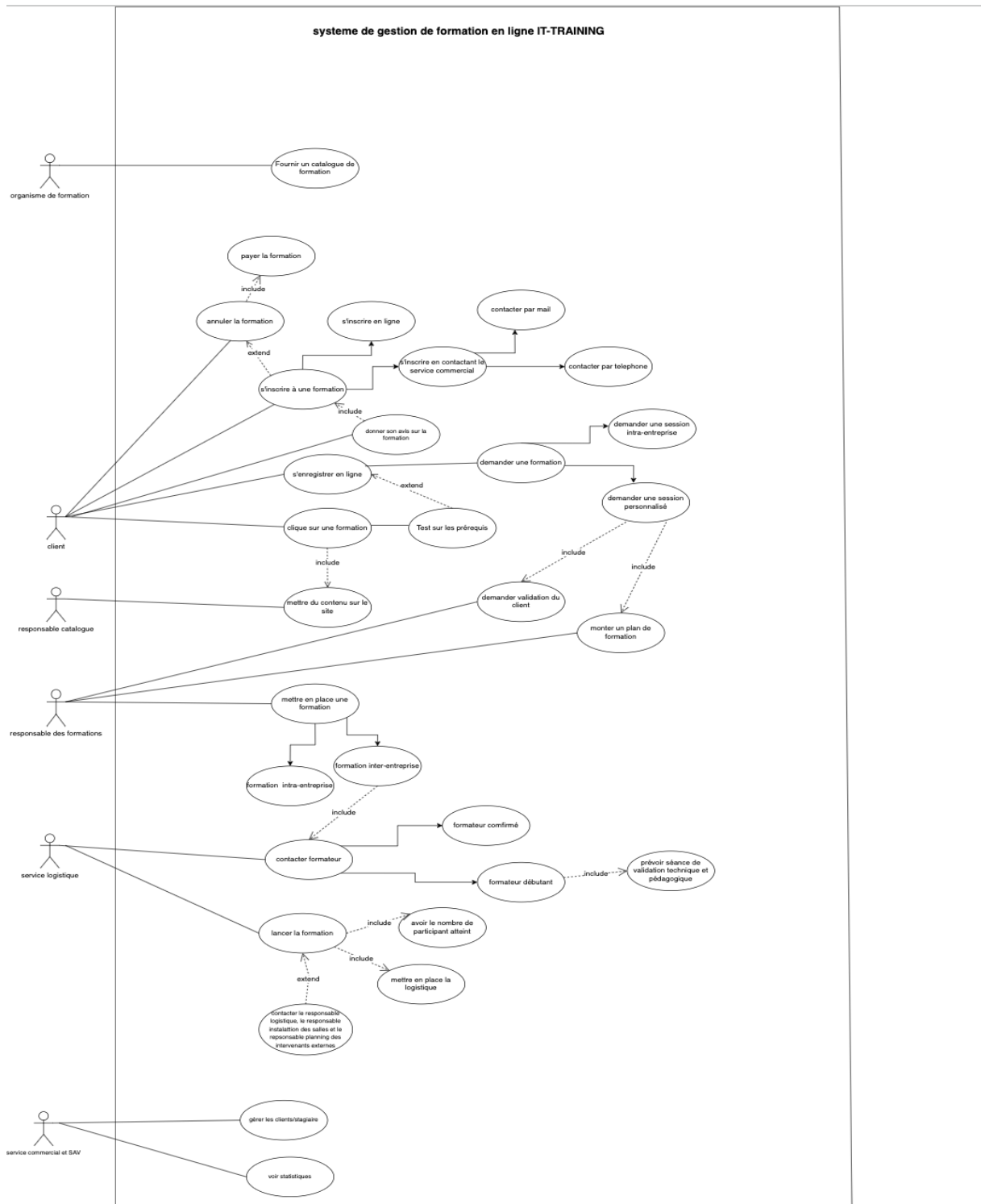
Amel Bouabdellah Boursali 40000259

Ayoub Khayef-allah 40000476

Ahmed El tewil 40008003

Ines Tohngodo 40011062

Diagramme US :



Scénario :

Use case	Traiter un enregistrement en ligne pour suivre une session de formation
Objectif dans le contexte	Un client clique sur le lien pour s'inscrire à une formation en ligne
Pré conditions	Le client a effectué le test de prérequis et son niveau est adéquat (facultatif) Le client s'est renseigné sur une formation
Post conditions (réussite)	L'inscription est enregistrée par le système informatique
Post conditions (échec)	L'inscription ne peut pas être enregistrée et est annulée

1. Le cas d'utilisation début lorsqu'un client clique sur le lien pour s'inscrire à une formation en ligne
2. Le client s'inscrit en ligne (choix du lieu et de la date)
3. Le client indique ses coordonnées et celles de son entreprise (s'il fait partie d'une)
4. Le client choisit le type de sa session (inter-entreprises, intra-entreprise ou personnalisée)
5. Le client règle le montant de sa prestation
6. Le nombre de participants est atteint
7. La salle est réservée
8. Les machines installées à distance (images virtuelles) avec les bonnes versions et logiciels requis pour la formation,
9. Le formateur est averti par mail et doit confirmer,
10. Les supports de cours/TP sont imprimés, CDs gravés
11. Les convocations sont envoyées aux stagiaires par courrier et par mail
12. Le planning est mis à jour
13. Les listes de présences sont imprimées
14. Les tickets repas sont imprimés
15. Les formulaires HTML d'évaluation en ligne de la session sont générés et prêts à être utilisés.

Enchaînements alternatifs

A1 : Le paiement est refusé

L'enchaînement A1 démarre au point 5 du scénario nominal

6. Le site indique au client que le paiement a été refusé

7. Le client choisit un autre mode de paiement (paypal ou autre carte bancaire)

Le scénario nominal reprend au point 6

A2 : Le formateur se désiste

L'enchaînement A2 démarre au point 15 du scénario nominal

16. Le système envoie une proposition à un autre formateur

Le scénario nominal reprend au point 9

Enchainements erreurs

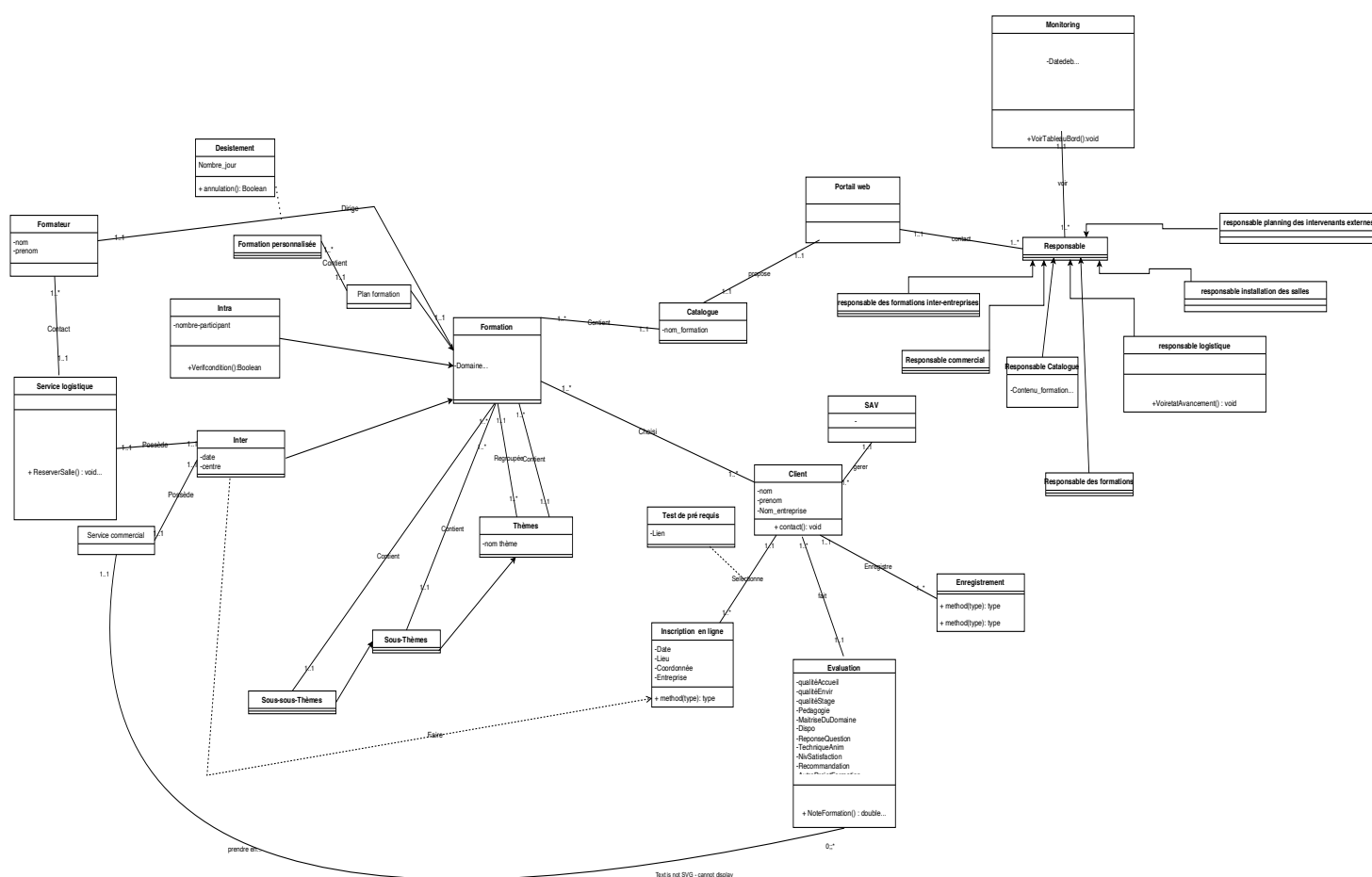
E1 : Le nombre de participants n'est pas suffisants 15 jours avant le début de la formation

L'enchaînement E1 démarre au point 15 du scénario nominal

16. Le système lance la procédure d'annulation : prévenir le formateur, les stagiaires, annuler la réservation des salles, le service planning, etc.

Le cas d'utilisation se termine en échec.

Diagramme de classe :



Génération de code :

Package CDC

```
Public class Formation {  
  
    Private String domaine ;  
  
    Public String getdomaine () {  
  
        return domaine ;  
  
    }  
  
    Public void setdomaine (String domaine) {  
  
        this.domaine=domaine ;  
  
    }  
  
}  
  
Public class Intra extends formation {  
  
    Private int nb_participant ;  
  
    Public int getnb_participant(){  
  
        return nb_participant ;  
  
    }  
  
    Public void setnb_participant( int nb_participant){  
  
        this.nb_participant=nb_participant ;  
  
    }  
  
    Public list <Intra> nb_participant=new ArrayList<Personne>() ;  
  
    Public void verifcondition ( formation F ) {  
  
    }  
  
    Public class Inter extends formation {  
  
        Private String date ;  
  
        Private String centre ;  
  
        Public String getdate(){  
  
            Return date ;  
  
        }  
  
    }  
  
}
```

```

    Public void setdate( String date){

        this.date=date ;

    }

    Public String getcentre (){

        Return centre ;

    }

    Public void setcentre ( String centre) {

        this.centre=centre ;

    }

    Public class Formateur extends formation {

        Private String nom ;

        Private String prenom ;

        Public String getnom() {

            Return nom ;

        }

        Public void setnom (String nom) {

            this.nom=nom ;

        }

        Public String getprenom (){

            return prenom ;

        }

        Public void setprenom ( String prenom) {

            this.prenom=prenom ;

        }

    }

```

Test JUnit :

```

Public class Testintra extends TestCase {

    Intra intraA= null ;

```

```

IntraA = new Intra ( 15 ) ;

Protected void setup () throws Exception {}

Protected void teardown () throws Exception {}

Public void test getnb_participant() {

    //fail (« Not yet implemented ») ;

//Getnb_participant doit retourner le même nombre de participant que celui utilisé dans le constructeur

    Int nb_participant = 15 ;

    assertEquals (int ; IntraA.getnb_participant()) ;

}

Public void test setnb_participant() {

    Fail (« not yet implemented ») ;

}

Public void testverifcondition() {

    // fail (« not yet implemented ») ;

    assertEquals(3,result.size() ) ;

// ici on vérifie si les 3 conditions sont valide.

```