**Partie 1 cycle de vie logiciel**

**Questions répondre par vrai ou faux. (4.5 Pts)**

1. Le modèle Agile permet de s'adapter rapidement aux changements des besoins clients.
2. La phase de tests intervient uniquement à la fin du cycle de vie du logiciel.
3. Le modèle en spirale combine des éléments du modèle en cascade et du développement itératif.
4. La maintenance corrective consiste à ajouter de nouvelles fonctionnalités au logiciel.
5. Le modèle en V est une extension du modèle en cascade qui met l'accent sur les tests.
6. La phase d'analyse des besoins est cruciale dans le cycle de vie du logiciel.
7. Des exigences changeantes ne permettent pas de définir tous les besoins du système.
8. Pour le modèle en cascade les exigences sont stables, par contre pour le modèle agile les exigences sont changeantes ou instables
9. Le modèle en V est choisi pour des exigences changeantes (instable).

**Exercice** (4pts)

 Vous êtes chargé de développer une application de gestion de bibliothèque. Les besoins du client ne sont pas entièrement définis et pourraient évoluer au fil du temps.

1. Quel modèle de cycle de vie recommanderiez-vous ? Pourquoi ? (1.5pts)
2. Décrivez les étapes que vous suivriez pour mener à bien ce projet. (1.5pts)
3. Comment géreriez-vous les changements de besoins pendant le développement ? (1pt)

**Partie 2 modélisation UML (11.5 Pts)**

Le département informatique souhaite automatiser sa gestion des absences des étudiants lors des séances de **Travaux Dirigés (TD)** et **Travaux Pratiques (TP)**. Actuellement, la gestion des présences est effectuée manuellement, ce qui entraîne des erreurs, des retards dans le traitement des données et des difficultés à identifier les étudiants ayant dépassé 3 absences. Le nouveau système doit être **numérique, automatisé et accessible** à la fois aux étudiants, aux enseignants et à l'administration.

**Objectifs du système**

1. **Suivi des absences** : (Enregistrer les présences et absences des étudiants pour chaque séance de TD/TP, Permettre aux enseignants de marquer les absences directement dans le système après chaque séance).
2. **Justification des absences** :(Permettre aux étudiants de justifier leurs absences en soumettant un justificatif (ex : certificat médical, motif personnel) via une interface en ligne, Les justificatifs doivent pouvoir être téléchargé sous forme de fichiers (PDF)).
3. **Validation des justificatifs** : (Permettre aux enseignants de consulter les justificatifs soumis par les étudiants, Les enseignants doivent pouvoir **valider** ou **refuser** les justificatifs).
4. **Alertes (**Identifier automatiquement les étudiants ayant accumulé **plus de 3 absences non justifiées**, Envoyer des **notifications** aux étudiants (avertissements) et aux enseignants (alertes) lorsque le seuil est dépassé).

Questions

* 1. Soient les classes (absence, justification, séance), complétez les classes manquantes de ce système et donner les attributs de chaque classe. (5 pts)
	2. Dressez le diagramme de classes pour représenter le système de gestion des absences. (2.5pts)
	3. Proposez un diagramme de séquence pour illustrer le scénario suivant

Un étudiant justifie son absence auprès du système de gestion des absences, l'enseignant peut valider ou refuser le justificatif.(2.5 pts)

* 1. Je désire mieux comprendre la classe absence quel diagramme utiliser. (0.5pt)
	2. Dressez ce diagramme. (1 pts)

Solution

**Partie 1**

**Questions (4.5 Pts)**

1. **Vrai**
2. **Faux**
3. **Vrai**
4. **Faux**
5. **Vrai**
6. **Vrai**
7. **Vrai**
8. **Vrai**
9. **Vrai**

**Exercice**

1. **Le modèle Agile** vue qu’il est Adapté aux besoins changeants et aux itérations fréquentes.
2. **Les étapes** : Analyse des besoins, conception itérative, développement, tests, déploiement, maintenance.
3. **Gestion des changements** : Utilisation de réunions régulières (scrums) pour ajuster les priorités et les fonctionnalités.

**Partie modélisation total (11.5pts)**

1. Total pour les classes 5pts.

Les classes (séance, absence, justification) pour les attributs 1 pt par classe 3pts

Et 0.5 pt pur les autres classes (4\*0.5) 2 pts

1. Personne (nom, prénom)
2. Etudiant (id,  groupe,  email). Un étudiant peut avoir plusieurs absences (Absence).
3. Enseignant (id,  email). Un enseignant valide les justifications (Justification).
4. Administration **(**idAdmin ). l’administrateur gère les étudiants et leur envois les alerte.
5. Séance **(**id, jour, type (TD ou TP),  heure début, heure fin , salle). Une séance peut avoir plusieurs absences (Absence).
6. Absence **(**id, date, justifiée (booléen indiquant si l'absence est justifiée)). Une absence est liée à un étudiant (Étudiant), une séance (Séance), et peut avoir une justification (Justification).
7. Justification  **(**id, motif, pièce jointe (string, chemin du fichier), statut (string, "en attente", "validée", "refusée"). Une justification est liée à une absence (Absence) et est validée par un enseignant (Enseignant).
8. Le diagramme de classe 2.5pts



1. Diagramme de séquence 2.5

 

1. Diagramme d’état 0.5pt
2. Diagramme d’état 1pts

