

INF 3176 : Environnement de développement avancé

Pratique

Dr Azanzi Jiomekong

Objectif: L'objectif principal de ce projet est de permettre aux étudiants, d'une part, de capitaliser les techniques de développement acquises au cours des deux premières années de leurs études. D'autre part, utiliser les techniques avancées de développement qui seront présentées dans cette conférence dans un projet réel. Ainsi, pour éviter les doublons, les étudiants utiliseront les techniques présentées dans ce cours dans le projet du cours INF3196.

Méthodologie: La méthodologie qui sera utilisée est la méthodologie Scrum. Pour permettre aux étudiants de maîtriser la méthodologie Scrum, les étudiants travailleront en groupe de 5 et appliqueront les pratiques de groupe lors du travail en équipe. Le conférencier sera le product propriétaire et les étudiants désignent chaque semaine le Scrum Master. Le conférencier évaluera l'avancement des travaux et donnera toutes les informations nécessaires qui aideront le groupe à améliorer ses compétences en développement d'applications mobiles.

Attention : L'ensemble des exercices est sur plus de 20 pour le contrôle continu et plus de 30 pour la pratique. Toutefois, les notes des étudiants seront maintenues sur 20 pour le contrôle continu et 30 pour la pratique.

Contrôle continu /20

Exercice 1 : ingénierie des exigences (5pts)

Application de la pratique Scrum qui sera identifiée lors du cours

Exercice 2 : ingénierie des exigences (4pts)

- 1- Discutez avec le superviseur de votre projet et identifiez les exigences fonctionnelles. Les exigences fonctionnelles décrivent ce que le système doit faire. **2 points**
- 2- Identifier les exigences non fonctionnelles en justifiant chacune d'entre elles. **1pt**
- 3- Donnez le Product-Backlog impliquant les éléments du backlog que vous devez traiter afin de construire votre produit **1pt**

Exercice 3 : Acteurs et cas d'usage (3,5pts)

Dans cet exercice, le langage UML est utilisé pour déterminer les limites du système. 1- Compte tenu de l'exercice, quels sont les acteurs ? **0,5pt**

2- Donnez les cas d'usage classés par acteurs (pour une bonne présentation, vous devriez présenter dans le tableau). **0,5pt** 3-

Donner la description des cas d'usage en utilisant le formalisme texte **0,5pt** 4- Donner la description des cas d'utilisation à l'aide de diagrammes de séquence **0,5pt**

5- Donnez les diagrammes de machine d'état de certains aspects importants de votre système **0,5pt**

6- Construire un tableau contenant les principales classes (suivies de leur description) nécessaires à cet exercice **1pt**

Exercice 4 : environnement de développement (5,5pts)

1- Compte tenu de vos compétences dans différents langages de programmation, déterminez celui que vous souhaitez utiliser lors de cet exercice **0,5pt**

2- Choisissez (en justifiant) et paramétrez votre système d'exploitation **0,5pt**

3- Identifier et mettre en place l'environnement de développement intégré, les frameworks et les API **1pt**

Compte tenu de la complexité de votre projet, vous avez décidé d'utiliser

DevOps. 4- Proposer l'architecture DevOps pour le développement de ce jeu **0,5pt**

5- Mettre en place les outils DevOps choisis compte tenu de l'architecture proposée à la question 4 **2 points**

6- Construisez un tableau (voir le tableau ci-dessous) contenant la liste des technologies que vous souhaitez utiliser dans votre projet, leur licence leur équivalence Open Source, leur équivalence propriétaire et la justification de votre choix **1pt**

Technologie	Licence	Equivalence open source (2)	Équivalence exclusive (2)	Brève justification du choix

Exercice 4 : Conception de l'interface utilisateur (2 pts) 1-

Concevoir l'interface utilisateur principale 2- Concevoir l'interface

interface utilisateur **Exercice 5 : Conception de logiciel (4pts) 1pt**

1- Proposer un diagramme de classes **0,5pt**

2- Proposer des diagrammes d'objets (justifier chaque diagramme d'objets que vous proposez) 3- **0,5pt**

Donné à la classe et aux diagrammes d'objets, proposer un diagramme de package **0,5pt**

4- En considérant que nous utilisons l'architecture monolithique, proposer cette architecture **1pt** 5- Identifier

et décrire les principaux algorithmes qui seront utilisés lors du développement de ce projet **1,5pt**

Pratique /30

Exercice 6 : développement (14pts)

1- A partir des exercices précédents, développez votre produit - développement + algorithme 2- **(5 + 5 = 10pts)**

Construisez le test fonctionnel **1pt**

3- Testez votre programme **1pt**

4- Proposez en justifiant la licence logicielle que vous souhaitez appliquer à votre produit. Une analyse pourra être utilisée pour justifier votre réponse) **1pt** (**Attention** :une composition logicielle

5- Évaluez votre système à l'aide du modèle d'acceptation technologique **1pt**

Exercice 8 : Architecture des microservices (11pts)

Étant donné que l'application peut croître chaque jour et devenir très volumineuse, pour éviter les limitations de l'architecture monolithique, vous avez décidé d'adopter l'architecture des microservices. Cette architecture nous aide à structurer l'application comme un ensemble de services hautement maintenables, testables et faiblement couplés. Il s'agit de permettre une livraison rapide, fréquente et fiable de l'application.

1- Donnez cinq raisons de déplacer l'application vers l'architecture microservices (s'il n'y a pas de raison, expliquez) **1pt**

2- Construire un tableau contenant les différents services et leur description et la justification du choix **1pt**

Microservice	Description	Justification

3- Utiliser la question 2 pour proposer l'architecture microservices de l'application **1pt** 4- Utilisez la

question 3 pour proposer une architecture de déploiement **1pt**

5- Proposer le nouvel environnement de développement en considérant les nouveaux choix 6-

2 points

Proposer la nouvelle version du projet, basée sur une architecture microservices **Exercice**

5 points

9 : Ingénierie pilotée par les modèles (10 pts)

Finalement, vous avez décidé de donner la possibilité aux joueurs de concevoir et générer automatiquement cette application. 1-

Proposer le modèle correspondant (DSL)

2- Identifier les cas d'usage de la modélisation et de l'environnement de génération 3-

Proposer l'UI pour la modélisation de votre application

4- Proposer le backend utilisé pour la génération du nouveau produit 5-

Testez votre application