|  |  |
| --- | --- |
|  | **Description Projet final 5IRC** |

**Objectifs :**

Réaliser un projet sur un sujet au choix. Le sujet devra être suffisamment conséquent (100 heures \* 6 étudiants).

**Organisation :** Equipe de 6 étudiants.

**Planning :**

* 24 séances de 4h dédiées à la conception et au développement du projet en suivant une démarche SCRUM
* 1 dernière séance de 4h dédiée à la soutenance et à la finalisation des livrables

|  |  |  |  |
| --- | --- | --- | --- |
| **Séances** | **Date** | **Contenu** | **Enseignant** |
| 1 (4h) | 13/12 AM | Cf. fichier « Contenu Séances 1 à 5 » | Anthony Chomienne  Vincent Couturier |
| 2 (4h) | 13/12 PM | Vincent Couturier Anthony Chomienne |
| 3 (4h) | 14/12 AM |  |
| 4 (4h) | 14/12 PM | Javier Espinosa |
| 5 (4h) | 15/12 AM |  |
| 6 (4h) | 15/12 PM | Développement Sprint 1 |  |
| 7 (4h) | 08/01 AM |  |  |
| 8 (4h) | 08/01 PM |  | Javier Espinosa |
| 9 (4h) | 09/01 AM |  |  |
| 10 (4h) | 09/01 PM |  |  |
| 11 (4h) | 10/01 AM | Revue sprint n°1 | Anthony Chomienne  Vincent Couturier |
| 12 (4h) | 10/01 PM | Développement Sprint 2 | Vincent Couturier |
| 13 (4h) | 11/01 AM |  |  |
| 14 (4h) | 11/01 PM |  | John Samuel |
| 15 (4h) | 12/01 AM |  |  |
| 16 (4h) | 12/01 PM |  |  |
| 17 (4h) | 22/01 AM | Revue sprint n°2 | Anthony Chomienne  Vincent Couturier |
| 18 (4h) | 22/01 PM | Développement Sprint 3 | Vincent Couturier |
| 19 (4h) | 23/01 AM |  |  |
| 20 (4h) | 23/01 PM |  | John Samuel  Anthony Chomienne |
| 21 (4h) | 24/01 AM |  |  |
| 22 (4h) | 24/01 PM |  |  |
| 23 (4h) | 25/01 AM | Revue sprint n°3 | Anthony Chomienne  Vincent Couturier |
| 24 (4h) | 25/01 PM | Préparation de la soutenance | Vincent Couturier |
| 25 (4h) | 26/01 AM | Soutenance du projet  Finalisation des livrables | Anthony Chomienne  Vincent Couturier  Jacques Saraydaryan |
| **TOTAL** | **100h** | | |

**\* A chaque fin de sprint remplir les documents SCRUM Sprint N.doc (N>=1)**

**Technologies :**

Vous devrez développer les applications suivantes :

* 1 application back-end
* 1 application front-end
* 1 application mobile

OU

* 2 applications back-end (dont 1 back spécifique de traitement de l’information par ex.)
* 1 application mobile ou front-end

EN :

* Choisissant des technologies différentes pour chacune de ces applications, parmi : Java, .NET, Android, iOS (Swift), Python, R, JS (Angular, React, Vue, etc.), multiplateforme (Flutter, Xamarin/MAUI, React Native, Ionic, etc.), etc.
* Utilisant au moins 1 service extérieur pertinent (Facebook, Twitter, Google/Bing/OpenStreet Maps, Amazon Alexa, etc.).
* Appréhendant des paradigmes de programmation variés : Services Web, Micro-services, etc.
* Intégrant de l’analyse de données, traitement de l’information (prédiction, clustering, classification) et/ou du text mining. Les choix effectués ici devront être cohérents avec le sujet et être dûment justifiés.

**Attention, vous ne pourrez (normalement) pas vous baser sur des données que vous générerez pour réaliser la phase d’apprentissage (données potentiellement biaisées, ou impossibilité due à la durée du projet) et devrez utiliser des données externes (open data, API externes, etc.), par exemple, disponibles sur Kaggle, Google Dataset Search, CMU Libraries, etc.**

* Pouvant mettre en œuvre un pipeline d’intégration continue et de déploiement continu (Gitlab, Azure DevOps, etc.).
* Mettant en œuvre des bonnes pratiques de programmation (par ex., celles du Software Craftsmanship <https://fr.wikipedia.org/wiki/Software_craftsmanship>) :
  + Des tests unitaires (OBIGATOIREMENT !) et éventuellement des tests d’intégration (mock, etc.) et/ou des tests end-to-end (Selenium, Sencha, etc.)
  + Documentation du code.
  + Des patterns (architectural patterns, design patterns)
  + ORM,
  + Etc.

**Environnements :**

Vous pourrez utiliser les environnements de programmation et SGBD à votre convenance, ainsi que des services cloud (Azure, GCP, AWS, Scalingo, Alwaysdata, Atlas Cloud Database, Heroku, etc.)

Attention à bien vérifier que vous utilisez la même version de l’environnement de développement (par exemple, Visual Studio 2022 Update 4) et les mêmes versions de librairies.

Vous pourrez mettre en œuvre les frameworks DM et ML vus en cours Big Data ou d’autres frameworks.

**Gestion de projet :**

Le projet sera réalisé en suivant une démarche SCRUM :

* Un membre de l’équipe sera PO.
* Un membre de l’équipe sera Scrum Master
* Vous définirez le backlog produit
* 3 sprints
* Le projet devra être géré sous Trello (ou Azure DevOps, etc.).

Vous pourrez ajouter des histoires techniques (en plus des User Stories).

Si vous avez oublié tous les concepts de SCRUM, cf. le très bon site : [https://scrumlife.tv](https://scrumlife.tv/)

Comme pour tout projet SCRUM, vous vous attacherez à développer les fonctionnalités à forte valeur ajoutée en premier.

**Livrables :**

* Cahier des charges (maquettes d’écran, diagramme des UC, traitement des données)
* Dossier technique : architectures des applications, diagrammes de classes des applications, design patterns appliqués, services extérieurs utilisés, modèle physique de la base de données, architecture d’analyse des données, etc.
* Dossier de recette :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Histoire utilisateur** | **Technologies utilisées** | **%age de réalisation** | **Raisons %age >0 et <100%** | **Par qui ?** |
| 1. En tant que… | WS REST .Net | 0% |  |  |
| 1. En tant que… | Android | 100% |  | 20% Etudiant 1  30% Etudiant 2  50% Etudiant 3 |
| 1. En tant que… | … | 30% | Problème pour récupérer la liste des contacts | 50% Etudiant 1  50% Etudiant 4 |

* Dossier d’installation
* Une vidéo (et non des copies d’écran) montrant les fonctionnalités de chaque application (déposée sur Youtube)
* Le code source des applications
* Slides de la soutenance

**Contenu (non exhaustif) de la soutenance : 20 minutes par groupe + 10 à 15 minutes de questions :**

* Introduction/contexte du projet
* Besoins réalisés et testés : testés (coche verte), réalisés mais non testés (coche orange). A ajouter sur le diagramme des UC.
* Architecture de l'application : MVC/MVVM, WS, etc. Pourquoi cette architecture ? Technologies ?
* Objectif du traitement de l’information / analyse de données, architecture, principaux algorithmes/modèles/fonctions statistiques utilisés et leur justification.
* Service externe : objectifs et fonctionnalités.
* Histoires techniques
* Exemples de réalisation : MPD, morceaux de code, diagramme de classes etc. Bonnes pratiques de programmation appliquées (Tests : couverture de test, patterns, etc.).
* DevOps :
* Pipeline CI/CD (éventuellement)
* Résumé́ des sprints (nombre d'histoires prévues/réalisées par sprint, priorité́ des histoires réalisées, vélocité). Ce qui a fonctionné́ / ce qui n'a pas fonctionné́ dans votre GP SCRUM. Prise de recul / SCRUM. Si vous aviez à mener ce même projet prochainement, que garderiez / modifieriez-vous ?

NB : Vélocité = nombre de points réalisés.

* Démo. 5 minutes maxi (une vidéo serait préférable afin de respecter le temps imparti).
* Bilan groupe : ce que nous avons apprécié́ / pas apprécié́ dans le projet final.