

|                                                                  |
|------------------------------------------------------------------|
| <b>SPE-CLO5:</b> My Booking Services                             |
| <b>Date:</b> 01/06/2020, <b>Date de fin:</b> 31/07/2020          |
| <b>Groupe:</b> rouman_s, legal_q, bincza_m, massar_t et simoes_t |

---

## Etape 0 - Organisation

Dans le cadre du déroulement de l'UV SPE-CLO5, nous aurons à développer une solution de gestion de réservations pour l'hôtellerie.

Nous aurons pour cela à développer une architecture en microservices.

**Définition microservice:** Les architectures en microservices sont une approche pour développer une seule application en tant qu'une suite de services portant une seule responsabilité, chacun s'exécutant dans son propre processus et communiquant avec des mécanismes légers, souvent des API REST à l'aide du protocole HTTP. Ces services sont indépendamment déployables et entièrement automatisées.

Nous allons à présent, détailler l'organisation qui sera mise en place lors du déroulement du projet.

### Méthode de gestion de projet:

Les membres de l'équipe ont l'habitude de travailler méthode agile, plus particulièrement en scrum, nous utiliserons donc cette méthode afin de réaliser ce projet.

Lors de ce projet, nous utiliserons des outils de versionning.

Le versionning se fera à l'aide d'un serveur Git qui aide à partager des informations aux membres d'une équipe, et nous permettra de réaliser des branches suivis de Pull Request.

Exemples de serveurs Git actuellement disponibles:

- Gitlab
- Bitbucket
- Github
- Gitosis

Github est un serveur Git permettant de gérer :

- Le cycle de vie des projets git
- Les participants des différents projets
- La communication entre ces participants

Github aide à construire de la documentation, et permet de réaliser des build, test et déploiement automatiques.

Github sera donc utilisé pour le versionning des codes sources de notre projet.

#### Méthode de production:

On va travailler en DevOps

**DevOps:** Méthode de production pour les équipes techniques visant à mettre en commun les compétences d'opérateurs, et de développeurs lors du déroulement d'un projet.

Lors de ce projet, nous serons amenés à tester les différents microservices, pour cela nous réaliserons des tests unitaires et des tests fonctionnels. Nous chercherons à exécuter les tests de manière automatisés.

Par la suite, nous aurons pour objectif de déployer les différents microservices, également, de manière automatisés.

Au cours du projet nous aurons à réaliser des schémas d'architecture applicatif.

Application utilisé pour réaliser les schémas d'architecture:

- **Lucidchart**

Application utilisé pour répertorier l'ensemble des fonctionnalités(les issues) à développer:

- **Github**

Suivi régulier par le chef de groupe pour voir l'avancé des tâches.

#### Langages:

Les microservices seront réalisés en **NodeJS** et en **Python**, afin de tirer profit de cet aspect de l'architecture microservice.

L'API REST sera, elle, en **NodeJS**.

**Répartition des tâches:**

| <b>Tâche</b>                                        | <b>Owner</b>                        |
|-----------------------------------------------------|-------------------------------------|
| Création de compte / Connexion                      | simoet, bincza_m                    |
| Gestion des permissions                             | massar_t                            |
| Listing du catalogue de chambre                     | legal_q, rouman_s                   |
| Gestion des prix                                    | rouman_s, bincza_m                  |
| Envoi d'email                                       | simoet, rouman_s                    |
| Gestion des disponibilités d'une chambre            | simoet, bincza_m                    |
| Affichage du catalogue des chambres                 | rouman_s, bincza_m                  |
| Développement des tests unitaires et fonctionnels   | simoet, legal_q                     |
| Mise en place de la CI/CD                           | massar_t                            |
| Création des images Docker                          | massar_t                            |
| Mise en place des pipelines avec test automatisés   | massar_t                            |
| Développement des microservices                     | simoet, bincza_m, legal_q, rouman_s |
| Automatisation de la generation de la documentation | massar_t                            |
| Gestion de l'infrastructure                         | massar_t                            |

La répartition des tâches peut être amené à changer au fil du projet.