

# Soutenance de fin de projet

Groupe 06  
17/12/2019

Antoine Rivoire  
@antoine382

Jérémy Masson  
@Jacket20

Baptiste Bétend  
@baptiste.betend

Tom Solvery  
@TomSOLVERY

Maxence Brès  
@serB

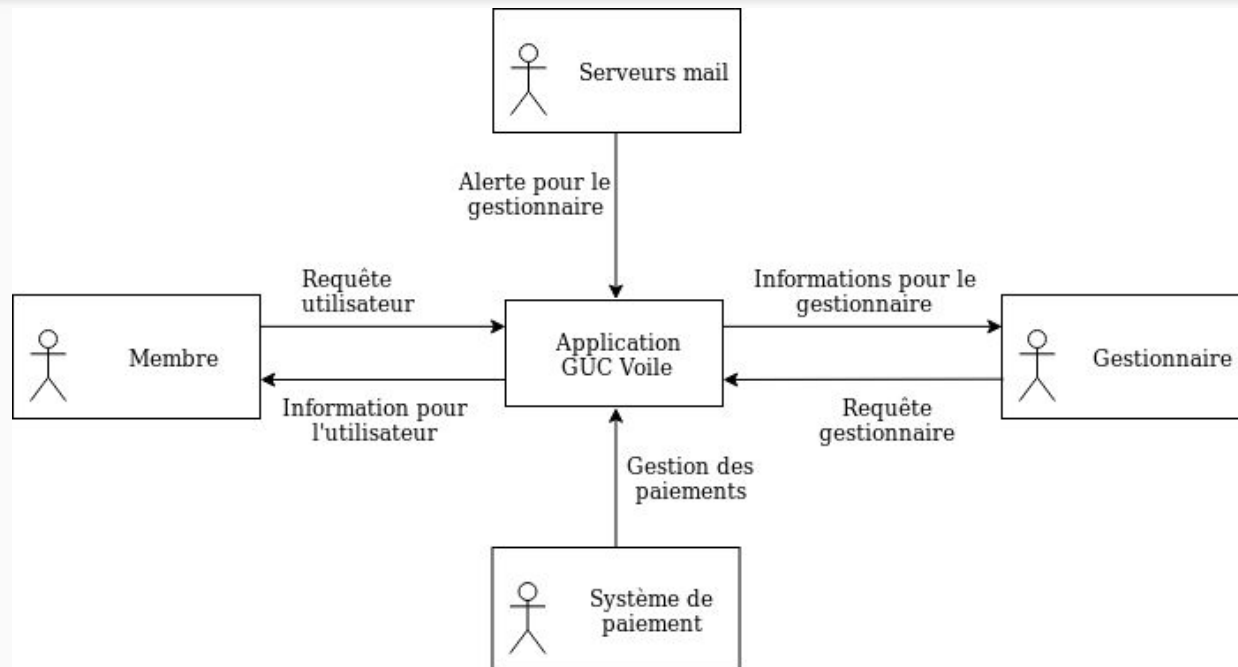
Nathan Dalaine  
@NathanDalaine

- INFO 5 -



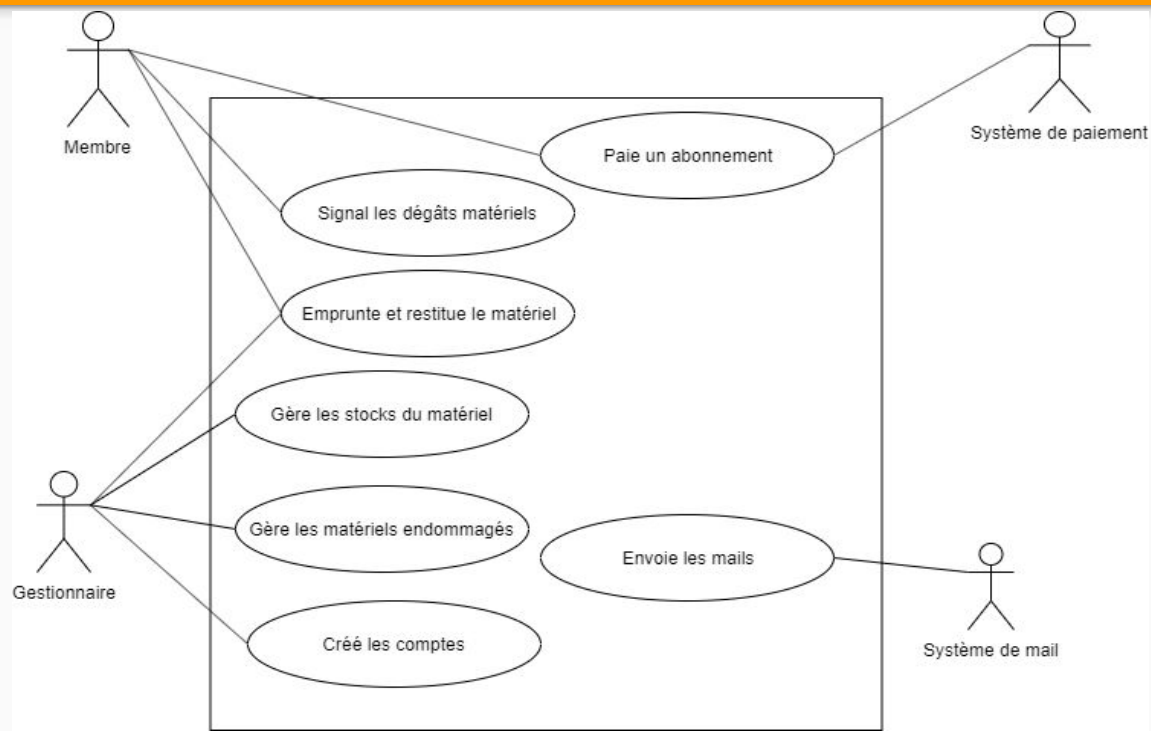
# Rappel du sujet

## Diagramme de contexte



# Rappel du sujet

## Cas d'usage UML



# Technologies et services utilisés



Angular (MVC + documentation + familiarité)



PostgreSQL (familiarité, gratuit)



Spring (familiarité, Jhipster)



Maven (documentation + familiarité)



JWTToken (simplicité)



VM Azure



Stripe (documenté et facile à implémenter)

# Fonctionnement du groupe

## Rôles de l'équipe

### **Chef de projet**

Maxence Brès - Système

### **Scrum Master**

Antoine Rivoire - Multimédia

### **Full-stack orienté Back End**

Jérémy Masson et Tom Solvery  
- Multimédia

### **Full-stack orienté Front End**

Baptiste Betend et Nathan Dalaine  
- Système



# Déroulement

Découpage du travail en sprint

Du 17/09 au 08/10 : Sprint 0 - Pré-étude

Du 08/10 au 05/11 : Sprint 1 - gestion d'une réservation

Du 12/11 au 03/12 : Sprint 2 - gestion des membres et tâches du gestionnaires

Du 03/12 au 17/12 : Sprint 3 - service de paiement et de mail

# Démarche de gestion de projet

## Planning sprint 1 : le coeur de l'application

- Authentification des membres
- Gestion de la sécurité
- Liste du matériel
- Effectuer une recherche afin de trouver du matériel spécifique
- Réservation du matériel
- Liste des réservations
- **Rendre une partie/la totalité du matériel emprunté**
- **Signalement des dommages (sans la partie email expédié automatiquement)**

# Démarche de gestion de projet

## Planning sprint 2 : gestion des utilisateurs et gestionnaire

- Authentification gestionnaires
- Création de comptes
- Accès aux statistiques
- Accès aux informations sur les membres
- Modification des informations personnelles
- Modification du niveau des membres par les gestionnaires
- Rendre une partie/la totalité du matériel emprunté
- Signalement des dommages (sans la partie email expédié automatiquement)



# Démarche de gestion de projet

## Planning sprint 3 : services externes

- Email envoyé automatiquement lorsqu'un matériel est rendu avec des dégâts
- Système de paiement
- Accès aux statistiques
- Correction de divers bugs
- Rendre une partie/la totalité du matériel emprunté
- Tri des voiles/planches par surface/volume
- Historique des réservations
- Pop-up de confirmation indique les planches/voiles non adapté à son niveau
- Modification des informations personnelles

# Démarche de gestion de projet

## Sprint Supplémentaire

- Amélioration des statistiques
  - Sélectionner une période d'observation
  - Les plus utilisés, abîmés, etc... ou l'inverse
  - Etude sur les membres
- Réservation à partir de l'historique
- Ajout de créneau de réservation

# Qualité / Tests

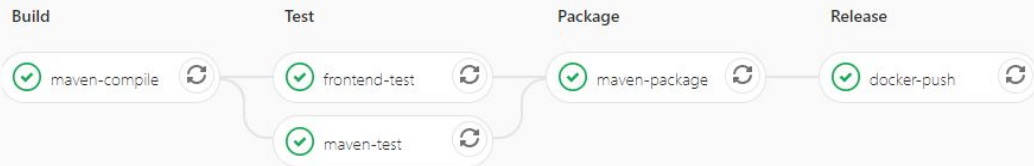
Intégration continue + code review :

A chaque merge request, on passe les tests pour vérifier que la branche à merge ne contient pas d'erreurs.

Le merge est alors fait seulement si les tests sont passés.

Les tests backend permettent d'assurer que les données qui communiquent entre le front et la bdd sont les bons.

Les tests frontend permettent de tester si les fonctions typescript et les services retournent les bonnes données.



# Qualité / Tests

## Front End

```
[INFO] Test Suites: 54 passed, 54 total
[INFO] Tests:      155 passed, 155 total
[INFO] Snapshots:   0 total
[INFO] Time:        65.106s
[INFO] Ran all test suites.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:26 min
[INFO] Finished at: 2019-12-16T13:29:05Z
[INFO] -----
```

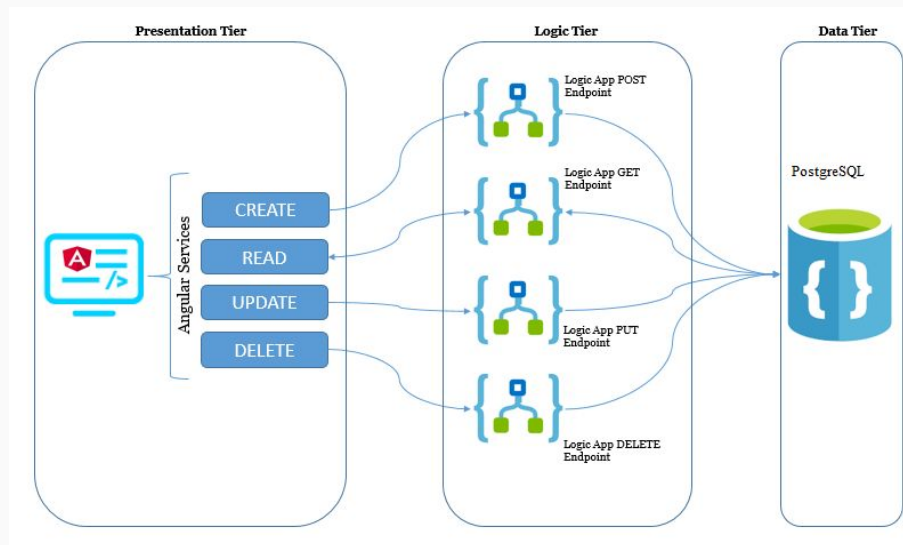
## Back End

```
452 [INFO] Results:
453 [INFO]
454 [INFO] Tests run: 180, Failures: 0, Errors: 0, Skipped: 0
455 [INFO]
456 [INFO]
457 [INFO] --- jacoco-maven-plugin:0.8.4:report-integration (post-integration-tests)
458 [INFO] Loading execution data file /builds/Polytech-INF05-2019-2020/g6/2019-2020
Test/integrationTest.exec
459 [INFO] Analyzed bundle 'Ecomgucvoile' with 116 classes
460 [INFO]
461 [INFO] --- maven-failsafe-plugin:3.0.0-M3:verify (verify) @ ecomgucvoile ---
462 [INFO] -----
463 [INFO] BUILD SUCCESS
464 [INFO] -----
465 [INFO] Total time: 01:22 min
466 [INFO] Finished at: 2019-12-16T13:20:39Z
467 [INFO] -----
```

# Résultats logiciel

## Architecture

### Application monolithique



# Résultats logiciel

## Use case réalisés

- Authentification des membres
- Gestion de la sécurité
- Liste du matériel
- Effectuer une recherche afin de trouver du matériel spécifique
- Réservation du matériel
- Liste des réservations
- Rendre une partie/la totalité du matériel emprunté
- Historique de réservation
- Authentification des gestionnaires
- Création de comptes
- Accès aux statistiques
- Vérification du niveau de l'utilisateur lors de la réservation
- Accès aux informations sur les membres
- Modification du niveau des membres par les gestionnaires
- Système de paiement
- Signalement des dommages

# Plus value

- Plusieurs moyens de sélection possible
- Gestion du matériel endommagé
- Respect du cahier des charges
- Aspect fonctionnel
- Robuste

# Résultats gestion de projet

## Répartition individuel des tâches

- Nombre de tâches par personnes pour 84 tâches réalisées:
  - Tom : 24/84 -> 9 fonctionnalités, 11 modifications, 4 bugfix
  - Maxence : 3/84 -> 1 modification, 2 DevOps
  - Baptiste : 11/84 -> 2 fonctionnalités, 6 modifications, 3 bugfix
  - Jérémie : 37/84 -> 9 fonctionnalités, 15 modification, 13 bugfix
  - Antoine : 9/84 -> 5 fonctionnalités, 1 modification, 1 bugfix, 2 docs
  - Nathan : 0/84



# Résultats gestion de projet

Répartition global sur chacune des étapes

- *Utilisation de Git kraken pour la gestion des tâches à effectuer (tickets)*
- *Retour sur les tâches effectuées avec le scrum master (journal de bord)*
- *Git Kraken Glo suit les sprints*

# Résultats gestion de projet

## Coût du projet global

Absence/Retard :  
19 heures d'absence.

Soit environ 3h10 par personne.  
Soit un coût de 22€/heures, on perd  
environ 70€ par personne donc 420€.

Temps de travail par semaine pour  
une personne : 12h

### Results

#### Software Development (Elaboration and Construction)

Effort = 98.8 Person-months  
Schedule = 16.7 Months  
Cost = \$339619

Total Equivalent Size = 24445 SLOC

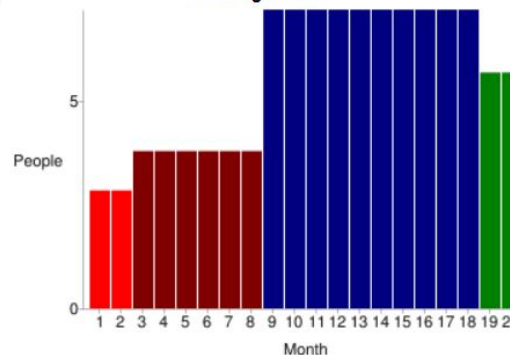
#### Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	5.9	2.1	2.8	\$20377
Elaboration	23.7	6.3	3.8	\$81509
Construction	75.1	10.4	7.2	\$258111
Transition	11.9	2.1	5.7	\$40754

#### Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.8	2.8	7.5	1.7
Environment/CM	0.6	1.9	3.8	0.6
Requirements	2.3	4.3	6.0	0.5
Design	1.1	8.5	12.0	0.5
Implementation	0.5	3.1	25.5	2.3
Assessment	0.5	2.4	18.0	2.8
Deployment	0.2	0.7	2.3	3.6

#### Staffing Profile



# Résultats gestion de projet

## Rétrospective

- Déploiement en production plus régulier et plus tôt pour anticiper les erreurs
- Chaque tâche doit être associée à un test
- Estimation du sprint est à améliorer
- Consacrer plus de temps à l'IHM
- Meilleure automatisation du déploiement

# Résultats gestion de projet

## Rétrospective

- Respect du cahier des charges au niveau des fonctionnalités
- Monté en compétence en Angular, en DevOps, en architecture, Spring Boot, projet sous méthode agile (condition professionnelle)
- Exemple concret

# Résultats logiciel

Démonstration