Équipe F

CANAVA Thomas - GARDAIRE Loïc - JUNAC Jérémy - MORTARA Johann

BlablaMove F

Projet AL SI5 V5

Scope

- Recherche d'une solution
  - Appel d'un service minimum et éventuellement un service externe
- Pricing
  - Appel d'un service externe
- Possibilité de signaler un problème / une déviation
- Notification de problème / reprise du parcours
- Gestion d'erreur en cas de non-disponibilité
- Contact de l'assurance
- Gestion des retards
- Notre service est appelé par le reste de l'application et ne gère que ce qui est décrit dans ce document.

Utilisateurs

- L'étudiant Norbert voulant déplacer ses cartons
- Le conducteur Gérard du véhicule
- Deux autres conducteurs, Jacqueline et Camille, arrivant sur la même route et prêts à prendre le carton
- Une étudiante Georgette prête à accueillir un carton

Scenarii Cockburn

Scénario : Le conducteur a un accident

1. La voiture tombe en panne.
2. Le colis n'a pas de dégâts.
3. Le conducteur signale le problème.
4. Le propriétaire est notifié.
5. L'application recherche une solution.
6. L'application trouve un conducteur proche pouvant reprendre le trajet pour le colis.
7. Lorsque Jacqueline récupère la marchandise, elle le signale.
8. Le propriétaire est notifié du départ du carton.

- Variantes

- 2-3.a Dommage sur le colis
  - 2. Le colis est endommagé.
  - 3. Le conducteur signale les dégâts et l'application envoie les données à l'assurance.
- 6-9.a Il n'y a pas de conducteur
  - 6. L'application trouve un étudiant proche pouvant garder le colis jusqu'à ce qu'un conducteur puisse reprendre le colis.
  - 7. Georgette notifie la réception du colis.
  - 8. Le système redirige Camille vers l'appartement de Georgette.
  - 9. L'application notifie le propriétaire lors du départ du carton.

Scénario : Le conducteur change de direction

1. Le conducteur Gérard souhaite changer de direction
2. L'application recherche une solution.
3. L'application trouve un conducteur proche pouvant reprendre le trajet pour le colis.
4. Lorsque Jacqueline récupère la marchandise, elle le signale.
5. Le propriétaire est notifié du départ du carton.

- *Variantes*
  - 3-6a. Il n'y a pas de conducteur
    - 3. L'application trouve un étudiant proche pouvant garder le colis jusqu'à ce qu'un conducteur puisse reprendre le colis.
    - 4. Georgette notifie la réception du colis.
    - 5. Le système redirige Jacqueline vers l'appartement de Georgette.
    - 6. Jacqueline notifie le propriétaire du départ du carton.

Scénario : Un conducteur a du retard

1. La personne qui a actuellement le colis souhaite s'arrêter sur le chemin.
2. Le système notifie le propriétaire du colis et éventuellement le transporteur suivant.
3. Le retard ne dérange pas le propriétaire.
4. Le conducteur notifie le propriétaire lors du départ de l'escale.

- *Variantes*
  - *2-5a. Le retard dérange le propriétaire ou le transporteur suivant*
    - 2. Le système recalcule un trajet depuis le point d'arrêt actuel.
    - 3. Le système notifie Camille du nouveau colis à récupérer.
    - 4. Lorsque Jacqueline récupère la marchandise, elle le signale.
    - 5. Le propriétaire est notifié du départ du colis.

Équipe F

CANAVA Thomas - GARDAIRE Loïc - JUNAC Jérémy - MORTARA Johann

Roadmap

Stratégie pour livrer un POC

Pour réaliser notre POC, ce qui nous semble le plus important à faire est de réaliser le scénario "*Le conducteur a un accident*", sans les variantes. Ceci nous permet de passer à travers toutes les couches de notre projet et donc d'en avoir une version minimale.

Les composants utilisés pour ce scénario sont les composants qui sont nécessaires à la réalisation de tous nos scénarii. Il semble donc intéressant de les intégrer à notre POC.

Pour réaliser cette preuve de concept, nous avons comme objectif de coder les composants nécessaires et de leur appliquer des scénarii d'acceptation permettant de démontrer les fonctionnalités.

De plus, nous souhaitons mettre en place l'intégration continue dès le début pour s'assurer qu'il n'y a pas de régression à chaque modification du code.

Après le POC

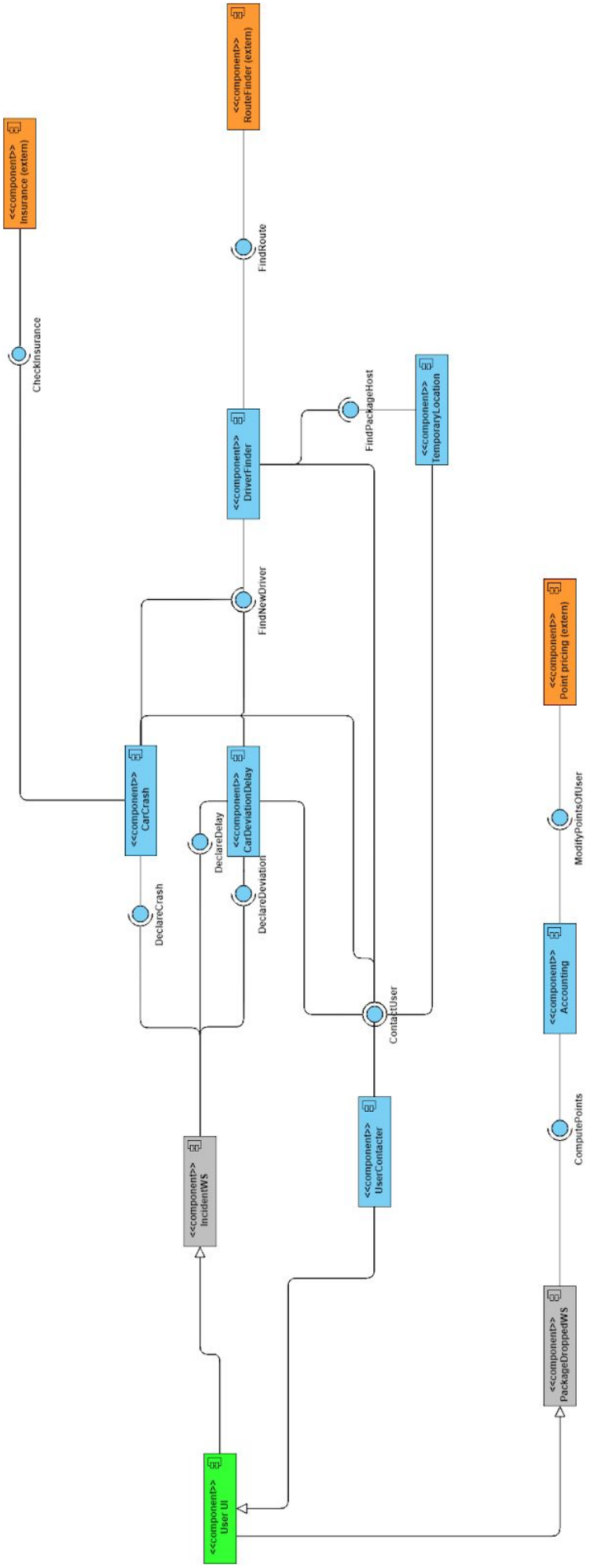
- Nous continuerons en implémentant les variantes du scénario principal.
- Pour la suite, nous comptons développer en priorité les retards, car ils sont plus complexes fonctionnellement que les déviations. En effet, il existe toute une partie *"Est-ce que le retard dérange le propriétaire ?"* qui n'est pas présente dans la déviation.
- La dernière fonctionnalité sera l'ajout des déviations comme cas de problème.

Rôles

- Thomas : Integrator, user experience expert
- Loïc : Team leader, architecture consultant
- Jérémy : Tech leader, test expert
- Johann : Scrum master, Product owner

Équipe F

CANAVA Thomas - GARDAIRE Loïc - JUNAC Jérémy - MORTARA Johann



Équipe F

CANAVA Thomas - GARDAIRE Loïc - JUNAC Jérémy - MORTARA Johann

Justification

Dans notre variante, nous devons gérer les problèmes qui peuvent survenir sur la route. Nous avons donc pris la décision de séparer le cas d'un crash et les cas de retard ou de déviation. En effet, un retard ou une déviation sur le trajet vont utiliser les mêmes composants pour résoudre le problème. À l'inverse, un crash oblige un contact avec l'assurance en cas de dommages sur le colis. L'assurance étant hors de notre scope, il est considéré comme un composant externe ici.

Nous avons un composant nommé *DriverFinder* permettant de chercher un conducteur pouvant reprendre le colis. Ce composant fait appel à un composant externe (*RouteFinder*) nous permettant de trouver la meilleure route pour le colis en partant du lieu où il a été abandonné. Encore une fois, cette fonctionnalité sort de notre scope et est donc utilisée comme un composant externe. Si aucun conducteur n'est trouvé, le composant *DriverFinder* contacte *TemporaryLocation* ayant pour rôle de trouver un hôte proche pouvant accueillir le colis en attendant qu'un conducteur puisse le récupérer.

Dans le cas d'un crash, comme dit précédemment, nous contactons l'assurance via le composant externe, en plus du scénario expliqué ci-dessus.

Nous avons également un composant *UserContacter* permettant de mettre en lien tous les acteurs du système. À chaque fois qu'un composant a besoin de notifier un utilisateur, il envoie les informations nécessaires à ce composant qui se charge de notifier le destinataire. Ce composant est nécessaire quand le colis a trouvé un nouveau pilote pour le déménager par exemple. Le propriétaire est alors notifié de l'évolution.

Enfin, nous avons un dernier composant pour la gestion des points des utilisateurs appelé *Accounting*. Il est accessible via un *Web Service* depuis l'interface utilisateur et fait le lien avec un composant externe spécialisé dans ce domaine.

Pour faire la liaison avec ces composants, nous avons fait le choix d'avoir 2 *Web Services* : un dont le seul rôle est la gestion des points et un autre permettant de connecter les composants résolvant les différents problèmes de nos utilisateurs. Le choix a donc été fait de séparer ces responsabilités en 2 *Web Services* car ce n'est pas le même métier.