

The slide features a white background with a dark blue vertical bar on the right side. In the top left corner, there is a logo for 'Disrupt Campus' with the text 'Université Grenoble Alpes' below it. The 'Disrupt Campus' logo consists of a stylized green mountain, a red hand icon, and a grey speech bubble. In the top right corner, there is a logo for 'POLYTECH GRENOBLE' with a blue circular icon. The central part of the slide is dominated by a large orange horizontal bar containing a dark blue speech bubble icon and the text 'EDCampus' in a bold, dark blue font. To the left of this bar are three dark blue horizontal bars, and to the right is a grid of small orange squares. Below the orange bar, the names of the team members are listed: Giulia BONASPETTI MARTINS, Marceau DECAMPS, Antoine RIVOIRE, and Maxence VINCENT. In the bottom right corner of the dark blue bar, the date '27/03/2020' is displayed.

**Disrupt Campus**  
Université Grenoble Alpes

**POLYTECH**  
GRENOBLE

**EDCampus**

Giulia BONASPETTI MARTINS  
Marceau DECAMPS  
Antoine RIVOIRE  
Maxence VINCENT

27/03/2020

Bonjour à tous. Cette présentation à pour but de vous montrer le travail que nous avons effectué lors de notre projet de 5ème année.  
Nous étions un groupe de 4 et nous avons travaillé sur la plateforme ED-Campus, un projet déjà opérationnel proposé par Disrupt Campus.

(Pour plus de détails, veuillez vous référer à notre [rapport final](#).)

## Rappel du sujet

Plateforme de **gestion**, de **travail** et d'**échanges**  
pour les projets interdisciplinaires.

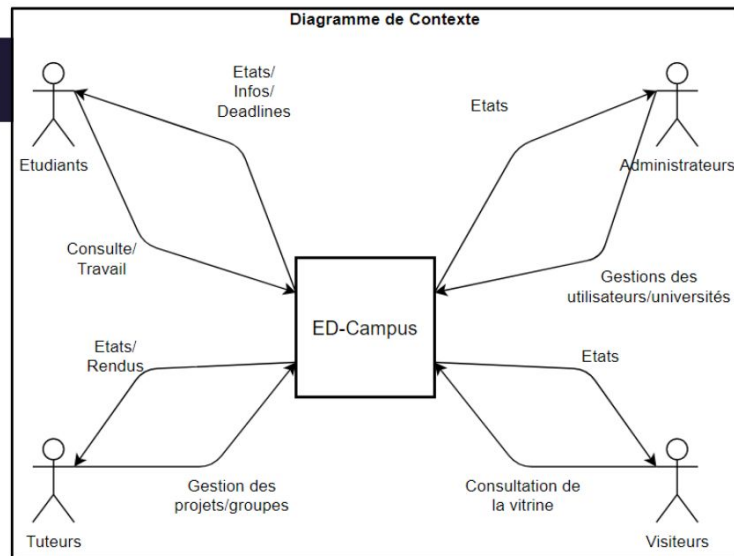
Cibles : Étudiants, Professeurs, Entreprises.

2

La plateforme ED-Campus est une solution permettant la gestion de projets pluridisciplinaires complète, pouvant regrouper des entreprises et/ou des étudiants de différentes universités.

Cette plateforme est donc accès principalement sur les projets universitaires et intègre une multitude d'outils afin de gérer des projets et de travailler en groupe. Entre autres, la plateforme propose un outil de gestion des tâches (Kanban, liste, gantt), une vue calendrier regroupant les différentes deadlines, un serveur permettant le dépôt de fichier, un chat, un module de gestion de finance et bien d'autre.

## Rappel du sujet



En plus de cette partie principale qui est la gestion et la réalisation de projets, ED-Campus intègre aussi un aspect un peu plus "commercial" qui permet aux étudiants/universités de promouvoir leurs travaux auprès des différentes entreprises au travers d'une vitrine de projets. Cet aspect est donc un point essentiel puisqu'il permet aux entreprises de se rendre compte très facilement du travail qui à été réalisé, et donc de savoir qui contacter si des postes sont ouverts.

## Cahier des charges

### Maintenance du site

- Corrections de bugs

### Implémentation de nouvelles fonctionnalités

- Vitrine
- Chat
- Téléchargement groupé

4

Cette plateforme est un projet Open-source de Disrupt Campus. Les principaux contributeurs étant des étudiants, notre travail s'est accès sur la maintenance du site avec principalement des corrections de bugs et des améliorations techniques. Notre rôle était donc d'intégrer le projet comme si nous étions en entreprise et de réaliser les différentes tâches qui nous étaient données pour chaque sprint.

Les bugs que nous devons modifier étaient plus ou moins critiques et se situaient aussi bien au niveau du front qu'au niveau du back. Il fallait donc que nous soyons compétent sur les deux facettes pour pouvoir faire face aux bugs.

Au niveau de l'implémentation, nous sommes intervenu principalement à ces trois endroits :

1. La vitrine est un espace ouvert au public regroupant le travail des différentes université et le détail des projets qui sont rendus publiques. C'est une fonctionnalité clé pour la visibilité et la communication des universités avec les entreprises.
2. Le chat et similaire au chat d'un réseau social, il permet de communiquer avec une seule personne ou bien tout un groupe en utilisant un bdd firebase. C'est une fonctionnalité très importante pour la communication au sein d'une équipe et très attendu car il facilite grandement le travail en en évitant de passer par un logiciel tiers tel que Messenger ou Slack.
3. Enfin, le téléchargement groupé concernait les livrables. Cette fonctionnalité permet donc de télécharger tous les rendus dans un zip et améliore énormément l'utilisation auprès des tuteurs.

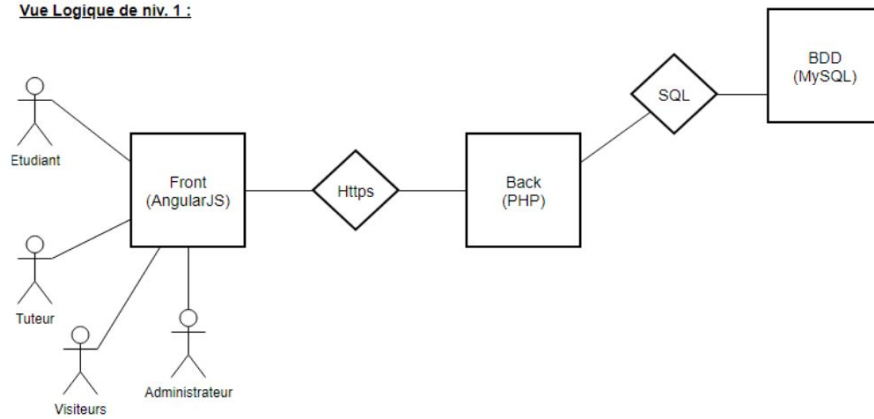
## Les technologies utilisées



Le frontend de cette plateforme est codée avec AngularJS. Ce framework commence à disparaître dans les nouveaux projets en étant remplacé par de l'Angular ou du React notamment, mais cela ne nous a pas empêché de réaliser nos tâches avec brio. Le backend est en PHP et la base de données est codée en MySQL : Ce sont des technologies que nous connaissons déjà mais travailler dessus, notamment pour le php, nous a permis de beaucoup progresser.

# Architecture technique

Vue Logique de niv. 1 :



6

L'architecture technique de notre application est la plus basique des architecture pour les projets web : je vous présente l'architecture 3-tiers.

On retrouve ici les différents langages/framework qui correspondent à chaque partie pour mieux vous situer.

Tous les utilisateurs utilisent la même interface en front, bien que celle-ci soit adaptée à l'utilisateur en question.

## Réalisations techniques

- Beaucoup de correctifs de bugs plus ou moins légers
- Amélioration et correction de la vitrine
- Correction et mise en marche du chat
- Nouvelle fonctionnalité, Téléchargement d'un zip des livrables

7

Nous avons participé à un projet collaboratif, qui était déjà en cours. Les tâches réalisées ont été définies à chaque sprint par Anthony Geourjon (ingénieur à temps plein qui travaille dans le projet et notre \*Scrum Master\*). Elles étaient surtout choisies par priorité. Priorité définies par les demandes utilisateurs et des clients potentielles. Ainsi, nous nous occupons d'abord des tâches critiques. De nombreuses tâches ont été réalisées, et elles étaient globalement très indépendantes. Cela a permis de facilement se répartir le travail. Parmi les tâches accomplies, celles qui sont les plus significatives :

- Correctif de bug : C'était des bugs plus ou moins critiques. Ils pouvaient toucher à la fois au *back* ou au *front*, voire les deux. La durée de la tâche dépendait également de la difficulté. Globalement tout le monde a eu ce genre de tâche. Mais c'est surtout Giulia qui s'est occupé des bugs liés au *front*.
- Amélioration et correction de la vitrine : C'était une tâche assez importante car la vitrine n'était pas encore achevée et quelques points ne fonctionnaient pas. L'amélioration de cette partie était particulièrement demandée par un client. C'est notamment Marceau qui s'est occupé de cette partie en apportant des correctifs *front* et *back*.
- Correction et mise en marche du *chat* : Lorsque nous sommes arrivés dans le projet, cette tâche n'était pas fonctionnelle et utilisable, de gros bugs empêchaient de l'utiliser. C'était une tâche qui était très demandée par de nombreux utilisateurs car le chat permet d'utiliser pleinement EDCampus, et donc d'utiliser un seul et unique outil centralisateur. C'est Maxence qui s'est occupé de cette tâche et a permis l'utilisabilité de celui-ci.
- Téléchargement d'un zip des livrables : Il s'agit d'une nouvelle fonctionnalité. C'est notamment la seule fonctionnalité qui a été réalisée entièrement, de l'élaboration du cahier des charges au test. Le but de cette fonctionnalité est assez simple, il permet de télécharger pour un enseignant tous les livrables rendus par des élèves dans un même

projet en un seul clic, et donc d'avoir un gain de temps. C'était une demande par les enseignants utilisateurs de EDCampus. C'était une tâche assez complexe car elle demandait des notions dans beaucoup de domaines : Le *back* et le *front*, sur la gestion de base de donnée, La création d'archive zip et la gestion de fichier.

- Vérification et de formatage automatique de code (CI) : Tâche réalisée à la fin du projet par Giulia. Cette tâche a été demandée pour avoir un code de meilleure qualité pour les prochaines contributions et une diminution de la dette technique de la plate-forme. C'était l'ajout d'un script lancer automatiquement à chaque commit pour la vérification et la correction du formatage du code.

Nous vous montrerons l'ensemble des tâches les plus importantes à la fin de notre présentation avec une démonstration.



## Gestion de projet

- Sprint de 1 semaine, défini par Anthony
- Sprint bien estimé
- Bonne communication avec le Scrum Master
- Tâches très indépendantes :
  - Beaucoup d'autonomie
  - Peu de communication entre membres

8

Plusieurs points vont être abordé concernant la gestion de groupe

- Planning : Nous avons fonctionné par sprint de une semaine avec la méthode *Scrum*. Notre *Scrum Master* était Anthony, nous avons une réunion avec lui à chaque début de semaine. Le but de celle-ci était de discuter des tâches du sprint précédent avec un tour de table, les difficultés rencontrées ainsi que les tâches réalisées. Chaque *sprint* était évalué et la quantité de travail était bien estimée. Cependant, deux tâches ont été sous estimées et ont pris bien plus de temps que prévu. Cette erreur comprenait la tâche d'Antoine sur les téléchargement de livrable en zip ainsi que celle de Maxence sur le chat (expliquées dans les réalisations techniques), mais globalement, le planning était bien respecté. Selon les sprints, nous avons souvent quelques tâches en plus, prévues initialement pour le sprint suivant, au cas ou nous étions plus efficace que prévu. Pour finir, le *backlog* et la gestion d'*issue* étaient réalisés sur le board de GitLab, ce qui permettait d'avoir une gestion des *issues* en temps réel.
- Méthode de travail : Afin de faciliter la gestion de projet, l'architecture 3-tiers est divisée en 2 *repo* Git, avec le *back* et la base de données d'un côté, et la partie *front* de l'autre côté. Ainsi, il est plus facile de dissocier/diviser les *issues*, et cela facilite grandement l'organisation, notamment pour le *Scrum Master*, mais aussi pour les développeurs qui ont besoin d'ouvrir qu'une partie de l'application pour faire une tâche.

À chaque *sprint*, chaque personne choisissait une tâche et se l'assignait sur le board via GitLab. Il apportait sa modification en s'assurant du bon fonctionnement et en vérifiant qu'il n'y avait pas de régression sur les autres fonctionnalités. Ensuite il faisait une demande de merge request pour intégrer son code, et c'était Anthony qui vérifiait et validait, ou non, la merge request.

Il est assez difficile de résumer nos méthodes de travail car cela était très différent d'une tâche à l'autre :

- Certaines tâches en *front* faisait par exemple plusieurs aller-retour entre développement et validation afin d'être le plus intuitif possible et avoir un bon retour-utilisateur.
- Pour la recherche de bug, nous utilisons beaucoup le debugger côté back de PhpStorm afin de localiser le problème.
- Certaines tâches ont nécessité de la programmation en binôme (pair programming) avec Anthony car nous étions bloqué, et qu'un point de vue extérieur de sa part avec l'apport de son expérience était très bénéfique et pouvait rapidement débloquer la situation.

D'autres habitudes plus générales ont été prises notamment de créer une nouvelle issue à chaque fois qu'un bug ou problème était trouvé et qu'il n'avait pas de lien avec la tâche courante.

- Rôles des membres : Globalement, chaque personne a eu un rôle de développeur *full stack* (*back* et *front*). Mais au final, chacun d'entre nous a eu un rôle assez différent.
  - Giulia a corrigé des bugs et a apporté des améliorations à divers endroits du site. Elle a aussi réalisé une tâche de DevOps.
  - Maxence a réalisé toutes les issues liées au chat.
  - Marceau a majoritairement corrigé des bugs, mais a aussi ajouté des améliorations, notamment liées à la vitrine.
  - Antoine a ajouté la fonctionnalité de téléchargement des livrables sous le format zip.
- Communication : Un aspect très important dans la gestion de groupe reste la communication. Mais c'est quelque chose qui a été un peu mis de côté vu le contexte de notre projet. En effet, chaque tâche était trop indépendante, il y avait donc peu d'entraide et de travail de groupe. Nous avons tout de même choisi de mettre une réunion journalière à chaque début de journée afin de savoir l'avancement des autres et donner un autre point de vue en cas de problème.

Cependant nous avons eu une bonne communication avec notre tuteur Anthony et nous le remercions énormément pour cela. Il était bien disponible et à notre écoute. Cela a été une plus-value dans notre projet.

## Les outils utilisés



9

Les outils que nous utilisons peuvent être divisés en outils de développement et en outils de communication et de gestion de groupe.

Outils de développement :

- \* Visual Studio Code - L'éditeur de code utilisé principalement pour le développement du frontend.
- \* PhpStorm - L'éditeur de code utilisé principalement pour le développement du backend.
- \* phpMyAdmin - L'application Web utilisée pour la gestion de base de données MySQL.
- \* pre-commit - Le framework utilisé pour ajouter un contrôle de la qualité du code. Il fait la gestion des scripts que Git exécute avant un commit, alias pre-commit hooks. Dans le framework pre-commit, les hooks utilisés étaient prettier (formateur de code des fichiers JS, CSS, JSON, Markdown et YAML), htmlint (vérificateur de code des fichiers HTML), pre-commit-hooks (formateur de attributs généraux).

Outils de communication et de gestion de groupe :

- \* Slack - Le logiciel utilisé pour la communication entre les membres du groupe ainsi qu'avec le responsable du projet.
- \* GitLab Gricad - La plate-forme utilisée pour le travail collaboratif et la gestion de tâches.

## Les métriques logicielles

- 1200 users 300 projets
- PC > Mobile
- 40 étudiants + un ingénieur à temps plein
- Coût du groupe ↔ 15.572,94€
- + 2 jours de dette technique
- 35 *issues*

10

EDCampus compte actuellement environ 1.200 utilisateurs et plus de 300 projets. Son utilisation est en majorité via les ordinateurs, mais il y a aussi des utilisations via les appareils mobiles. Depuis 14 mois, 40 étudiants et un ingénieur à temps plein ont contribué à la plateforme.

Nous pouvons calculer, par exemple, le coût que notre groupe de 4 développeurs aurait eu pour le projet, en tenant compte du fait que nous travaillons officiellement 21 jours. Nous ne compterons pas les locaux car l'université nous offre la place et le calcul deviendrait trop compliqué. Le salaire moyen d'un développeur est de 2.200€ net/mois, ce qui devient 3.900€/mois en comptant les charges patronales, soit 185€/jour/personne. Nous avons ajouter au calcul l'essence que Maxence a dépenser pour rejoindre notre tuteur dans ses bureaux, ainsi que l'amortissement du matériel d'Antoine qui à moins d'un an. Tout cela nous amène à un total de 15.572,94€ sur notre période de projet.

Dès le début de ce projet, 2 jours de dette technique ont été ajoutés. Cette dette est principalement due à deux facteurs :

\* Nous travaillons sur un grand projet qui n'a pas de tests d'intégration, de tests unitaires, ou d'autres formes de contrôle en dehors de la révision du code et des tests manuels effectués par notre Anthony ou par nous-mêmes.

\* Comme nous avons travaillé avec un vrai client et en utilisant des méthodes agiles, il était parfois plus important pour le client que le service soit fait rapidement et non que le code soit fait avec rigueur.

Notre groupe a participé à 35 issues au total, dont 27 ont déjà été fusionnées dans le code principal, 5 sont en cours de révision et 3 sont encore en phase de réalisation.

## Conclusion, une rétrospective

### Positifs :

- Projet intéressant :
  - Complet (approfondissement et apprentissage de compétences)
  - Concret (travail "utile")
- Encadrant investi et disponible :
  - Découpage des tâches optimales
  - Suivi de qualité
  - Grande aide
- Liberté d'horaires
- Travail de qualité

### Négatifs :

- Langage inconnu en début de projet
- Peu de travail commun
- Manque de documentation
- Code pas très propre

11

Pour conclure le projet, nous avons organisés une rétrospective au sein de notre équipe. Cette rétrospective avait pour but de définir ce qui a bien marché et ce qui a moins bien marché pendant ce projet.

### Points positifs :

Les points positifs sont très nombreux.

- Le projet était en lui même très intéressant. C'est un projet complet, qui nous a fait découvrir de nouveaux langages et outils (php, javascript, PhpStorm, phpMyAdmin, ...) et nous a permis de mettre en application nos connaissances. Le projet est également concret, notre travail a été utile. Il y a de vrais utilisateurs et donc une demande en temps réel. De plus, le projet étant en open-source, cela peut nous servir de portfolio.

- Notre encadrant et scrum master (Anthony GEURJON) était très investi et disponible. Les tâches était bien défini ce qui rendait les rôles clairs et précis. On ne perdait donc pas de temps pour l'organisation. Le suivi qu'il nous assurait était de qualité et nous avons même travaillé en pair programming avec lui dans les cas les plus extrêmes.

- Nous avons une grande liberté d'horaires : peu importe les horaires de travail, tant que le travail est fait dans les temps.

- Pour terminer, nous jugeons que le travail que nous avons accompli a été de qualité.

### Points négatifs :

Ils sont bien moins nombreux que les points positifs et en sont leur conséquences directes.

- Les langages utilisés dans ce projet nous était inconnu au début du projet, ce qui nous a fait perdre du temps à la prise en main.

- Nos tâches étaient si bien découpées que nous avions peu de travail en commun à proprement parler. Nous avons remédié à ce souci en organisant un daily-meeting entre nous chaque jour.

- Comme le projet est collaboratif, il y a un manque de documentation sur l'existant et le code n'est pas très propre à certains endroits.



(cf. Screencast démonstration)

# Annexes

## L'organisation d'équipe

- Plateforme open-source
  - Travail en contribution avec d'autres développeurs
- Tous les développeurs sont fullstack
- Méthode agile
  - SM : Anthony GEOURJON



## Le backlog

- Sélection des issues du sprint
  - Anthony Geourjon
- Plusieurs issues par sprint avec différents niveaux de difficulté
  - Le développeur choisi sa prochaine issue

15

Anthony sélectionne un nombre X d'issues qu'il souhaite voir traiter dans le cadre du sprint actuel, on partage les issues entre nous en tenant compte des facilités et des difficultés de chaque développeur et des issues.

En général, on choisit juste un issue et quand il est terminé, on choisit notre prochain issue, et ainsi de suite, jusqu'à ce que tous les issues soient terminés.