**Skipti <3**

1. **PRESENTATION DU SUJET**
2. **Principe d’utilisation - Yannis**

**L’université : l’apogée de l’éduction et des liens sociaux.**

A l’IUT Paris DESCARTES, dès la première année les étudiants du département informatique sont soumis à une charge de travail conséquente, mais pas irréalisable. Le secret provient de l’organisation et de la solidarité.

Pour aider leurs camarades de première année à supporter le rythme du DUT, le BDE informatique propose un système de parrainage entre 1A et 2A. Ce système partant d’une volonté solidaire d’intégrer les nouveaux à cependant ses limites. Il est très long et très compliqué d’étudier le profil de chaque étudiant et par conséquent le BDE associe parrain et filleul de manière aléatoire. L’initiative perd donc une partie de son sens…

A l’heure du digital, le fichier Parainage.xls du BDE vit ses derniers instants, notre équipe de PJS4 a trouvé la solution : skipti.fr une application WEB et mobile qui permet aux étudiants de se matcher et de se découvrir à partir de traits de caractères commun et d’une messagerie instantanée.

Si trouver le bon parrain permet de partager des vecteurs d’activités communs, il permet surtout d’instancier le bien-être de l’étudiant quand celui-ci vient à l’IUT. Vous l’aurez compris, Skipti est bien plus qu’une simple application de rencontre puisqu’elle s’inscrit dans l’innovation éducative afin d’atténuer le nombre d’abandons dans la filière.

1. **Les différentes fonctionnalités - Yannis**

Avant de pouvoir trouver le parrain de ses rêves, l’étudiant doit utiliser les différentes fonctionnalités de l’application. Le BDE pourra suivre l’avancement de l’activité grâce à une plateforme administrateur.

**Fonctionnalité pour l’étudiant**

* L’inscription : A partir de son mail @etu.parisdescartes.fr, l’étudiant peut s’inscrire sur notre site à grâce un mail de confirmation qu’il pourra consulter dans sa boite mail.
* La caractérisation : L’étudiant doit cliquer sur 3 adjectifs pour définir ses traits de caractère
* La modification du profil : il est possible d’ajouter une description de 280 caractères et une photo de profil personnalisée.
* Le Swipe : l’étudiant parcours tous les profils connectés il a la possibilité de :
  + Liker si le profil du parrain/marraine ou du filleul.le lui plait
  + Drop si le profil ne lui correspond pas
* Le Match : si un étudiant de première année et de deuxième année s’entre-likent, l’application leur délivre un match
* Le Messenger : Si match il y a, les étudiants pourront échanger entre eux
* Le Choix : A partir du Messenger, si l’étudiant clique sur « The Right One » il exhausse le souhait d’un parrainage en question
* Le Parrainage : Si deux étudiants ont exhaussé le souhait d’être parrainés, l’application retient le parrainage et :
  + Le/la 1A se retrouve invisible du Swipe.
  + Le/la 2A peut avoir jusqu’à 4 filleuls, il/elle sera rendu invisible du Swipe lorsque qu’il aura utilisé tous ses parrainages.

**Fonctionnalité pour le BDE :**

* Export CSV, obtenir un fichier .csv contenant :
  + Liste de tous les étudiants
  + Liste des Parrainage
  + Liste des étudiants non-inscrits
* Matching :
  + Matcher les étudiants non-inscrits
  + Matcher les étudiants non parrainés
* Envoi d’e-mails :
  + Relancer par mail les étudiants non-inscrits
  + Relancer par mail aux étudiants non-parrainés
  + Envoyer un mail récapitulatif aux étudiants.

La plateforme admin, est un outil indispensable du BDE car ce sont eux qui auront le pouvoir final d’attribuer les parrainages.

1. **Business Plan :**

Grace à une bonne conception applicative, l’application est facilement maintenable ce qui nous permet d’y raccorder plusieurs bases de données. Skipti n’aura donc aucun mal à se voir utiliser au-delà des frontières du DUT informatique et pourra être rapidement utiliser sur tous les IUT.

116 800, c’est le nombre d’étudiants en DUT chaque année

1. **Normes**

Pour qu’une application comme skipti.fr puisse être commercialisée en toute légalité en 2018, elle doit être aux normes du règlement général pour la protection des données (RGPD) récemment instauré par l’Union Européenne le 25 mai dernier.

Les principaux objectifs du RGPD sont d'accroître à la fois la protection des personnes concernées par un traitement de leurs données à caractère personnel et la responsabilisation des acteurs de ce traitement. Ces principes pourront être appliqués grâce à l'augmentation du pouvoir des autorités de régulation.

Pour répondre à ces normes, il a donc fallu :

* **Encrypté les données à caractères personnel de notre base de données :** 
  + Photos de profil
  + Surnom
  + Adresse e-mail
  + Mot de passe
* **Sécurisé l’authentification :** 
  + Certification HTTPS pour assurer une connexion encryptée sur le web
  + Possibilité de changé de mot de passe
  + Authentification via Token : Le mot de passe ne circule jamais en claire dans la BDD, il est encrypté via le protocole de hachage Blowfish (Haché plusieurs fois)
* **Le droit à l’oublie** : Possibilité de supprimer son compte à tout moment
* **Avertissement d’utilisation de cookies**.

Si nous sommes si sensibles à ces normes, c’est grâce à l’état d’esprit d’un établissement comme l’IUT Paris Descartes. Au-delà du développement informatique, le « Learning by Doing » nous a été inculqué dès BAC +2. De ce fait nous développons des applications maintenables et réfléchies en fonction des normes en vigueurs et à venir.

1. **ORGANISATION & CONCEPTION DU PROJET**
2. **Réflexions sur le sujet :**

En deux ans de DUT Informatique, s’il y a bien une erreur qu’on a appris à ne pas faire, c’est foncer tête baisser dans le code. Avant de pouvoir produire une application aussi complexe que celle demandée pour le PJS4, il faut trouver un projet dont nous sommes sûr de pouvoir réaliser.

Le concept a spontanément été trouvé, mais cependant il a fallu définir notre cahier des charges. La première problématique a été de se demander « Dans un projet qui aime faire quoi ? ».

Avec un groupe mix et des perspectives d’avenir différentes nous avons constaté qu’a 6 nous pouvions réaliser notre application avec une vision à 360 degrés du web. C’est-à-dire associé design thinking, développement technique, gestion de projet et ingénierie d’affaire pour mener à bien ce développement.

Le concept une fois bien développé nous a permis d’être utilisé comme cahier des charges (CDC). En partant de la problématique « Comment améliorer le parrainage d’étudiant entre 1A et 2A ? ». De ce fil rouge ont émané les fonctionnalités décrites précédemment (I.2) et que nous avons développées en toute agilité.

1. **Développement agile :**
   1. **Méthode de travail**

Dès le premier module de programmation en première année nous avons été initié à la méthode de SCRUM. Cette approche agile du développement informatique permet d’optimiser le cadrage du projet. Cette méthode de gestion permet de gérer la conduite du projet à l’aide d’échéances hebdomadaire ayant pour but de vérifier si la fonctionnalité en développement correspond bien au besoin définis dans le CDC.

Ces échéances fréquentes appelées « sprint » ont permis à notre chef de projet de planifier nos tâches et de donner un aspect graphique à notre projet. Pour des raisons réalistes, nous avons décidé de ne pas utiliser un diagramme de Gantt comme scheduler. Notre expérience en développement n’est pas encore assez mûre pour savoir combien de temps le développement d’une ressource peut durer. Nous avons donc choisi de créer notre propre fichier. Ce fichier que vous pourrait trouvé en Annexe nous a permis de :

* Lister toutes les tâches
* Identifier les tâches critiques
  + Et donc choisir des ordres de priorités
* Avoir une dimension graphique et rassurante de l’avancement du projet

// Comme précédemment énoncé notre équipe est pleines de profils différents. La répartition des tâches a donc été simplifié puisque chacun a pu se spécifier des tâches qui lui tiennent à cœur et qui lui serviront dans son futur.

**Design thinking :**

**Front :**

Du login au Swipe en passant par l’interface administrateur, Julie et Thibaud ont travaillés en binôme pour réaliser la plupart des vues. De plus Marina a ajouté une touche de professionnalisme à l’application en ajoutant l’insertion et la retouche d’image dans la modification de profil. Raphael a amené son expérience de développeur web full-stack pour débloquer des situations et pour amener une touche de propreté dans le code. ☺

**Backend :**

Le back-office de l’application

**Les**

Rédaction

Gestion

1. **Conception**

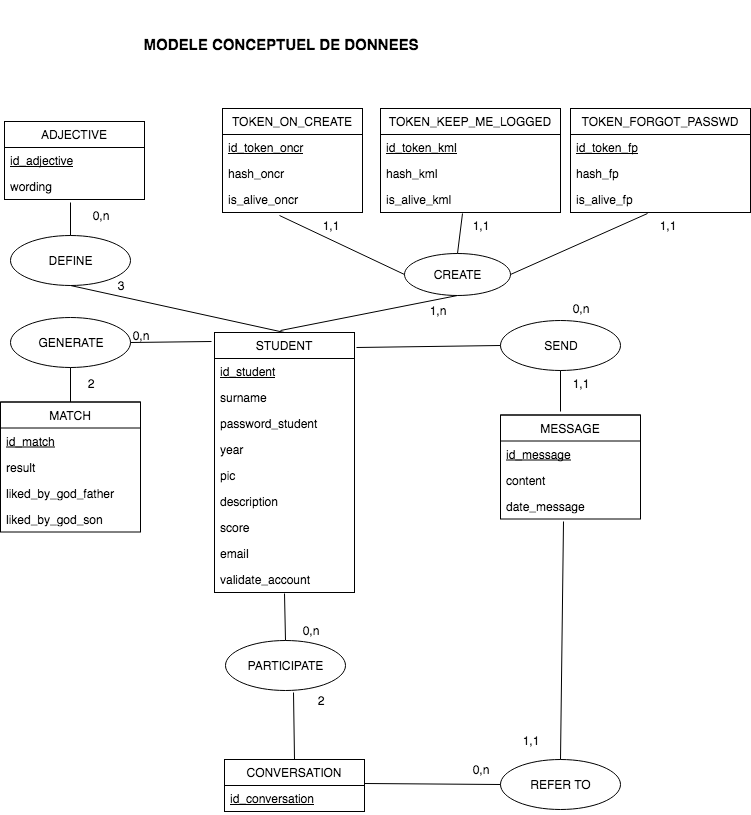
Raphael et Yannis se sont partagé les taches de conception. Raphaël ayant déjà des bases en programmation WEB a tenu à nous présenter le principe MVC.

Yannis, qui continuera sa formation informatique en L3 MiAGE, a pris en main la conception de la base de données et a défini les diagrammes de séquence afin que l’équipe puisse développer efficacement toutes les fonctionnalités de l’application.

* 1. **Modèle conceptuel de données (MCD)**

Le pilier d’une application client/serveur c’est sa base de données. Lorsque l’on envisage de créer une base de données, il faut réfléchir à son architecture. « Quelles données mes services ont besoin pour établir leur fonction ? » est la première question à se poser pour définir un dictionnaire de données. A partir de ce dictionnaire il convient ensuite de modéliser sa base avec le MCD.

Pour skipti.fr, le model repose sur des données utilisateurs qu’il a fallu mettre aux normes en vigueur. (Voir 8. Mise aux normes).

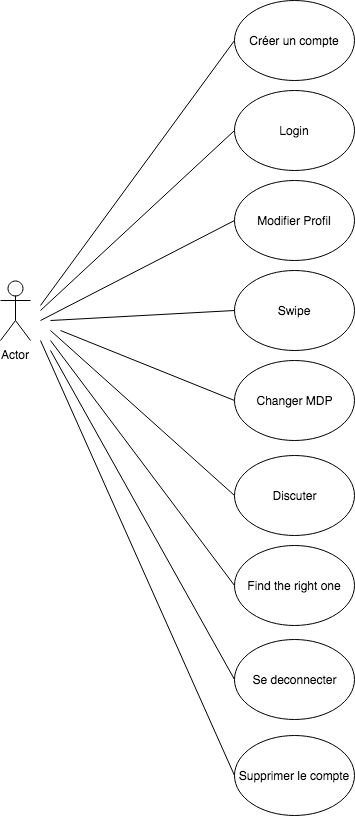


*Table student de la bdd. Voir MCD complet => ANNEXES : MCD*

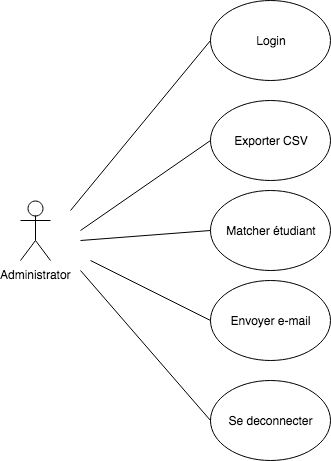
* 1. **Diagramme des cas d’utilisation (DCU)**

Dans cette application, vous êtes soit un étudiant soit un administrateur. Le DCU permet de schématiser de manière simple les fonctionnalités visibles par l’étudiant. Dans le monde professionnel il est souvent utile pour permettre à des non-technicien de savoir si les développeurs ont bien compris le besoin.

SKIPTI. FR pour un étudiant lambda

**DCU Etudiant Lambd**

Skipti.fr pour un membre du BDE



* 1. **Architecture Client/Server**

En tant qu’étudiant de deuxième année nous souhaitons vraiment aider ceux qui nous succèderont dès la rentrée prochaine, nous avons donc décider de mettre en place notre propre serveur WEB.

Avant de rendre accessible notre travail au World Wide Web, il y a quelques étapes :

Premièrement nous avons acheté un serveur j’ai oublié le fucking nom chez OVH A NOS PUTAIN DE FRAIS, J’AI DU INVITER JULIE AU MCDO <3 . Une fois le serveur acquis nous nous y connectons via le protocole SSH. Nous paramétrons son accès selon les « best practices » que l’on peut trouver sur le web. Nous inst Je suis cuis, on fera sa demain

* 1. **Architecture Model View Controller (MVC)**

Nous nous somme auto-formés sur un modèle de conception très prisé sur le marché du travail : l’architecture MVC. Elle permet un développement séparé de l’application. Exemple : Les données entrée par l’utilisateur ne seront jamais inscrites directement dans le serveur

PHOTO MVC

Développer selon l’architecture MVC c’est développer de manière séparée entre :

* Le Model : Tous les fichiers .php qui vont permettre d’interagir avec la base de données. Toutes les requêtes SQL sont initialisées dans ce dossier.
* La View : Ce dossier contient tous les fichiers que l’utilisateur peut voir afficher dans son navigateur en utilisant Skipti
* Le Controller : Enfin le Controller gère les entrées en provenance de l’utilisateur et les interactions à destination de ce dernier utiliser dans des fichiers .php. Métaphoriquement il s’agit d’un tunnel entre la vue et le modèle
  1. **Diagramme de séquences**

En conception, la dernière étape avant de commencer à coder est l’élaboration du diagramme de séquence. Nous nous sommes rendus qu’utiliser ces diagrammes au niveau de la structure MVC permet d’avoir une vision de la structure de code à adopter.

Nous avons commencé le développement de certain service avant de créer ces diagrammes et nous nous sommes rendus compte qu’avoir ce diagramme permet de factoriser du code. Exemple du Swipe. L’architecture pourrait être optimisé, car des services similaires sont appelés plusieurs fois.

Ces diagrammes renseignent donc sur la propretés des liens et donc sur la maintenabilité du code.

A finir demain

1. **Ambiance de développement**
   1. **Communication**

C’est une vérité étudiée en cours de communication, le travail en équipe gagne en productivité dès lors que nous sommes en groupe. Pour optimiser le team working, le plus important est de savoir communiquer. Chez Skipti.fr on a décidé de privilégier les échanges directs et la qualité de notre environnement de travail.

Au-delà des séances dédiées au PJS4, nous avons décidé de rester solidaire au-delà des murs de l’IUT en s’organisant des marathons code à plusieurs au quatre coins de l’Ile de France. Avec l’arrivée des beaux jours, ces moments ont amené le plus de productivité puisqu’ils ont renforcé la cohésion du groupe.

Lorsque que nous devions travailler chez nous, nous avons utilisé un groupe Messenger. Cette application que notre génération utilise à sa perfection est très efficace pour la prise d’information et s’interroger à plusieurs.

* 1. **Partage de ressource**

**GitHub :** GitHub est un hébergeur de dépôt git gratuit. Un git est un logiciel de control de version. Une application qui génère de nouvelles version de projet sans jamais les écrasés A finir demain

* 1. **Environnement de développement**
  2. **Les gars vous avez utilisé quoi ??????**

1. **Web design**

Dans l’optique de devenir designer, Julie a réalisé les maquettes des vues en établissant au préalable une charte graphique. (Voir 5.1). Marina, passionnée par le dessin a créé les graphismes de l’application.

* 1. **Charte graphique**

**JUL y a que toi qui me le mieux en parler ☺**

* 1. **Design thinking**

Le Design Thinking est une des problématiques actuelles du développement WEB. Comment simplifier l’utilisation d’une application ? Certain parlent également d’User eXperience (UX).

En vérité, il s’agit d’une problématique ergonome que notre équipe a souhaitée d’étudier. En regardant les vues de l’application (Annexes : Vues), vous pouvez constater qu’il n’y a que très peu de bouton, et très peu de texte. L’effet voulu sur l’utilisateur est qu’il comprenne le plus rapidement possible comment fonctionne la fonctionnalité qu’il est en train d’utiliser.

Continuer le blabla ok ?

Exemple : Sur le Swipe, pour interagir il ny a que deux boutons

1. **Technologies utilisées**

**HTML**

**CSS**

**JavaScript**

**PHP**

**Python**

**…**

1. **Réalisation des fonctionnalités les plus importantes**

**Raconter le fonctionnement de chaque fonctionnalitée**

**Pour chaque => Ergo, dev**

1. **Développement Android**

**ALICE A L’AIDE**

1. **Mise aux normes**

**BLABLA SUR OPEN SSL ET HTTPS**

**Conclusion**

LE BORDEL TOTAL AVEC L’APPRENTISSAGE ET TOUT LES PROJETS, MAIS LE KIFF TOTAL AVEC LA FAMILLE ET LA PETITE COMPETITION AVEC MA**P THE TUB**

**ANNEXES**

**To Do list**

**Résumés in english**

* **Alice**
* **Julie**
* **Marina**
* **Raphael**
* **Thibaud**
* **Yannis**

**Vues**