

RAPPORT FINAL

Guide Développeur

Par

AUBERT Vincent BELGUENDOZ Sekina

BOUCHERIMA Amina EZ-ZINE Najwa

Table des matières:

LE SUJET	3
TECHNOLOGIES EMPLOYEES	4
ARCHITECTURE TECHNIQUES	5
RÉALISATIONS TECHNIQUES	6
GESTION DE PROJET	6
Méthode Agile: SCRUM	7
Rôles et tâches effectuées	7
Gestion des risques	8
Outils	8
MÉTRIQUES LOGICIELLES	9
Temps ingénieur :	9
Langages :	10
CLOC : Analyse cloc en fin de projet	10
Analyse de GitLab	10
CONCLUSION	11

LE SUJET

Le système vestibulaire sous-tend le sens de l'équilibre. Il s'agit d'un sens bien particulier car on n'en prend conscience que lorsqu'il dysfonctionne, au cours d'un vertige par exemple.

L'oreille peut être découpée en 3 parties :

- L'oreille externe
- L'oreille moyenne
- L'oreille interne

C'est au niveau de l'oreille interne que se situent les capteurs du système vestibulaire, plus précisément au niveau du labyrinthe. Ce système étant fortement connecté au système moteur participe au maintien de la posture et à la coordination des mouvements réflexes des yeux et de la tête.

En réalité, l'équilibration est plus complexe et ne se résume pas qu'au système vestibulaire. Le cerveau fusionne les signaux en provenance de plusieurs sens pour nous permettre la perception de soi dans l'espace. Le labyrinthe est composé des canaux semi-circulaires, l'utricule et le saccule. Les canaux semi-circulaires servent à percevoir les accélérations angulaires. Ils sont disposés selon trois plans perpendiculaires de telle sorte que l'excitation de l'un provoque l'inhibition des autres.

Au cours d'un mouvement de tête, les canaux circulaires des oreilles droites et gauches sont soit stimulés soit inhibés. Le signal perçu par les oreilles est ensuite retransmis aux muscles des yeux. Cela permet de garder une cible en vue, de fixer quelque chose lors d'un mouvement.

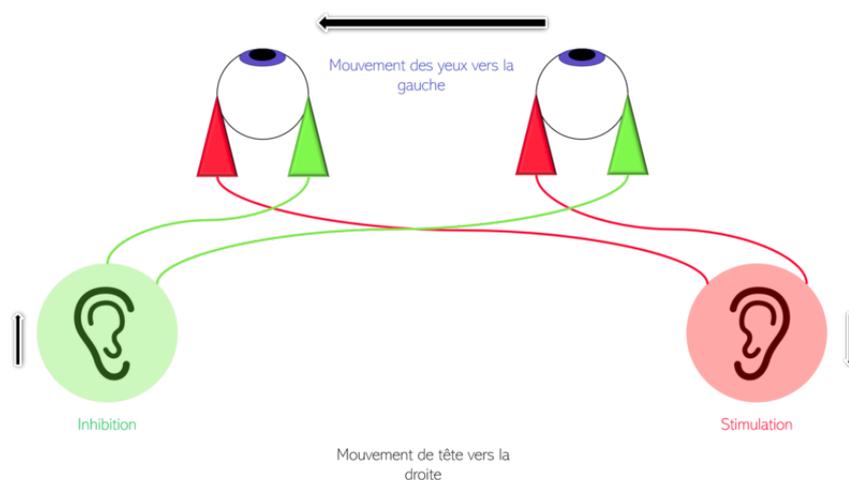


Schéma de fonctionnement du système vestibulaire.

On peut être amené à suivre une rééducation en cas de dysfonctionnement du système. Les exercices de cette rééducation vont consister à jouer sur ce réflexe vestibulo-visuel. Par exemple, un exercice pourrait consister à fixer une cible en bougeant

la tête à une vitesse suffisamment élevée et essayer de lire un mot (apparaissant dans la zone ciblée).

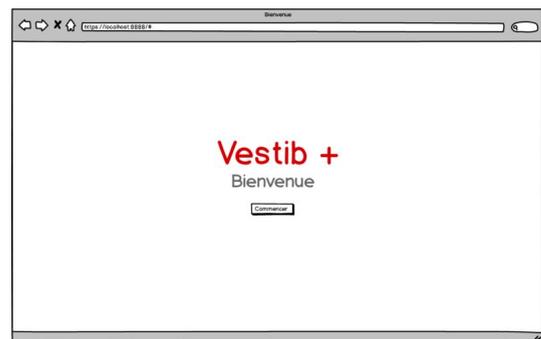
Le but de ce projet est de créer un dispositif et une application permettant de réaliser ces exercices depuis chez soi et d'offrir la possibilité au kinésithérapeute de suivre son patient à distance.

Pour la partie dispositif, une équipe de IESE4 est chargée de créer un casque avec un accéléromètre pour transmettre les données du patient à l'application créée par notre équipe.

Afin de gérer la réalisation des exercices pour les patients et l'affectation d'exercice pour les kinésithérapeute, nous avons choisi de faire deux applications différentes que l'on nommera: InterfaceKiné et InterfacePatient.



Page Bienvenue de l'InterfaceKiné

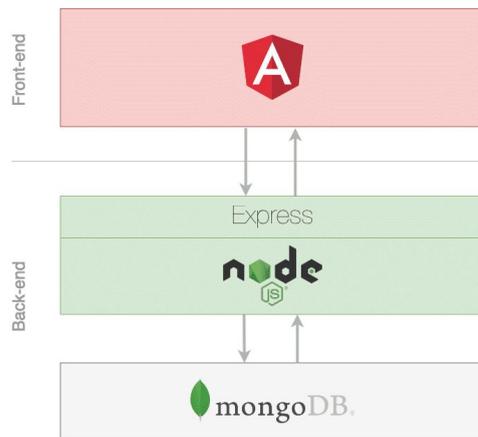


Page Bienvenue de l'InterfacePatient

TECHNOLOGIES EMPLOYEES

Pour réaliser nos applications, nous avons décidé de partir sur une stack MEAN utilisant :

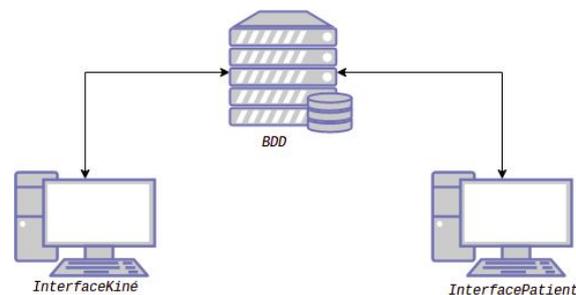
- **MongoDB** : Base de données No-SQL. La partie back-end de notre application l'utilise pour stocker les données sous forme de documents JSON. C'est donc une base de données orienté documents. Elle possède plusieurs avantages : on ne manipule pas la base de données directement mais des objets et les tables sont créées automatiquement.
- **Express.js** : Framework Backend de l'application web, tourne au dessus de Node.js. Très léger et simple d'utilisation, il permet par exemple de lancer un serveur en moins de 5 lignes.
- **Angular** : Framework Frontend de l'application web; exécute le code sur le navigateur de l'utilisateur. Nous l'avons choisi car plusieurs d'entre nous avait déjà eu l'occasion de l'utiliser précédemment pour des projets. Cela signifiait donc une période de prises en mains plus courte.
- **Nodejs** : Permet d'implémenter le backend en JavaScript.

SOURCE

<https://medium.com/codingthesmarthack-crash-course-part-2-implementing-the-back-end-642567e16bb9>

ARCHITECTURE TECHNIQUES

L'architecture de notre logiciel est la suivante :

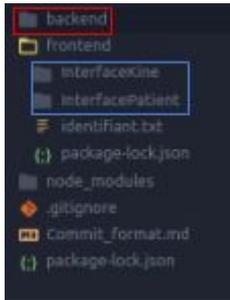


Nous avons d'un côté l'application patient, permettant au patient d'accéder à sa session du jour. Et de l'autre côté, il y a l'application kiné permettant la gestion des patients.

Pour faire le lien entre ces deux applications, nous avons décidé de déployer une base de données en ligne. Ainsi, à chaque fois que le kiné ajoutera une session à un patient ou autre, l'information sera stockée dans la base de donnée. Et lorsque le patient se connectera, il récupérera les informations de la BDD, et la mettra à jour une fois les exercices effectués.

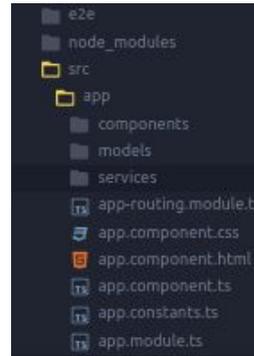
RÉALISATIONS TECHNIQUES

Arborescence du projet



Arborescence du projet:

- en rouge les fichier du backend: API, BDD, etc.
- en bleu, les deux interfaces



Arborescence des interfaces:

- *components* contient toutes les pages de l'interface (le html, le ts, et le css).
- *models* contient les types des objets stockés des tables de la BDD.
- *services* contient toutes les fonctions appelées par plusieurs page (appel à l'API, store, ...)

Communication hardware/software

Gestion exercices : Store

Une autre notion importante de notre code est les stores. Pour s'assurer que l'utilisateur n'envoie pas n'importe quelles données nous stockons temporairement les exercices dans un observable qui va notifier les autres éléments si tout changements a été effectués sur les exercices. Cela permet non-seulement de d'être à jour mais aussi d'envoyer nettement moins de requêtes à la BDD.

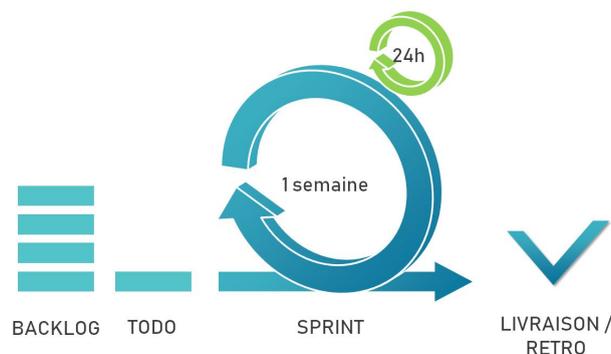
GESTION DE PROJET

Méthode Agile: SCRUM

Pour l'organisation et la gestion de projet, nous sommes partis sur une méthode SCRUM:

- Tous les matins, un **daily meeting** était prévu pour savoir ce qui avait été fait, les problèmes rencontrés, et ce qu'il est prévu de faire dans la journée.
- En fin de semaine, le vendredi soir, la rétrospective de sprint était organisée pour savoir ce qui s'est bien passé au sein de l'équipe et ce qui pourrait être amélioré. Nous faisons également un rétrospective sur l'avancement général du projet et définissons des objectifs globaux pour la semaine suivante.

La durée de nos sprints était d'une semaine, nous avons eu en tout 5 sprints pour finir ce projet.



Organisation SCRUM

Rôles et tâches effectuées

Au niveau du rôle de chaque membre, nous avons mis en place l'organisation suivante :

- **Sekina** était **chef de projet, scrum master** et **lead développeur**: Elle était en charge de mener les réunions, de mettre à jour et d'ajouter les documents sur la page Air, de s'assurer la bonne avancée du groupe, co-gérer les branches sur repository Git, etc. Elle était également la développeuse principale sur l'implémentation de l'Interface Patient. Elle est aussi intervenue pour l'implémentation de la notion de *Store* dans l'implémentation des exercices.
- **Amina** était **lead développeur**: Elle était d'abord chargée de mettre en place l'environnement de programmation: installer la mean stack, vérifier son fonctionnement, et mettre en place la base de donnée et les premières fonctions de l'API. Elle s'est ensuite vu responsable de toute l'interface kiné.
- **Vincent** était **responsable du git** et **développeur**: Il était en charge de protéger et de gérer les branches. Il était également responsable de l'implémentation des exercices.
- **Najwa** était **développeuse en backend** et **responsable de communication**: Elle était donc chargée d'implémenter les fonctions de l'API nécessaires pour les interfaces, mais aussi de déployer la base de donnée, de la sécuriser, etc. Elle s'est ensuite vue en charge de la communication entre le dispositif des IESE4 et de l'application. Enfin, elle était notre lien entre tous les partis de ce projet: Professeurs, Cabinet Vestib+ et les IESE4.

Gestion des risques

L'application étant médicale, nous gérons des données sensibles et privées. De plus, nous stockons dans notre base de nombreuses données personnelles comme le numéro de téléphone, la date de naissance, la maladie du patient, etc. Il nous est donc nécessaire de sécuriser notre base, pour éviter ou du moins minimiser les risques.

Nous avons donc faire en sorte de forcer un kiné à envoyer son username avant d'accéder à un patient, pour s'assurer que seul le kiné gérant ce dossier peut y accéder. Nous avons aussi fait en sorte de ne jamais afficher le champs servant à authentifier un patient pour garder un minimum de sécurité.

Cependant, nous nous rendons bien compte que si l'identifiant est dans un fichier sur une clef usb, la sécurité se voit réduite. Mais nous nous sommes assurés que le client

était conscient des risques (un patient changeant son fichier pour accéder un dossier d'un autre, etc).

Outils

GitLab : Nous avons choisi GitLab comme outil de versionnage du code et avons convenu d'un mode d'utilisation très strict.

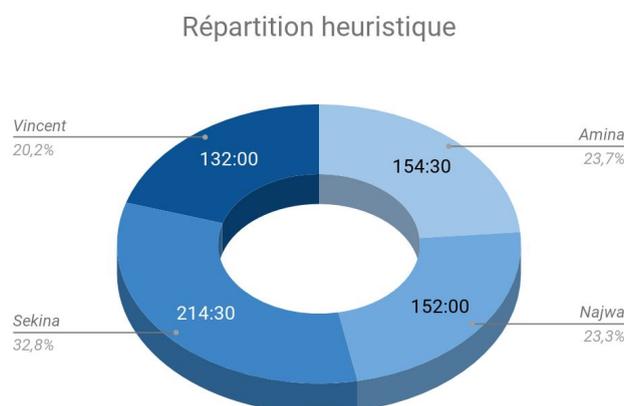
- La branche master est sécurisée. Seule 2 membres, le responsable du git et le chef de projet en cas d'incapacité du responsable du git, peuvent directement y accéder.
- Chaque feature doit être développée sur une branche portant le nom de cette feature.
- Tout merge de 2 branches doit passer par une branche intermédiaire.
- Chaque commit doit respecter sur format :
 - ADD: <message>, Si il y a eu un ajout de fichier, dossier ou code.
 - DEL: <message>, Si il y a eu une suppression de fichier, dossier ou code.
 - FIX: <message>, Si une erreur a été corrigée.
 - REF: <message>, Si il y a une un petite optimisation du code.
- Les messages doivent être écrit en anglais.

Trello : Nous l'avons utilisé principalement pour organiser nos différentes tâches mais on peut aussi y trouver un résumé de nos réunions et notre journal de bord commun. En plus, Nous y avons ajouté des cartes "mémos" pour lancer l'application, pour les installations de modules, etc.

Drive : Nous avons utilisé Google Drive pour le partage les différents document et fichier annexe au code (les diapositives de présentations, les maquettes, etc).

MÉTRIQUES LOGICIELLES

Temps ingénieur :



En ce basant sur le simulateur de salaire de StackOverflow (<https://stackoverflow.com/jobs/salary>), nous avons les salaires bruts suivants:

- 38 000 € pour un développeur backend => Charge salariale = 55 100
- 36 000 € pour un développeur frontend => Charge salariale = 52 200
- 37 000 € pour un développeur fullstack => Charge salariale = 53 650
- 50 000 € pour un chef de projet débutant (estimer après avoir visité plusieurs sources)=> Charge salariale = 72 500

On a travaillé 35 jours au total pour ce projet, on obtient donc un budget financier de : $35 \times (199 + 143 + 147 + 151) = \mathbf{22400}$

Langages :

CLOC : Analyse cloc en fin de projet

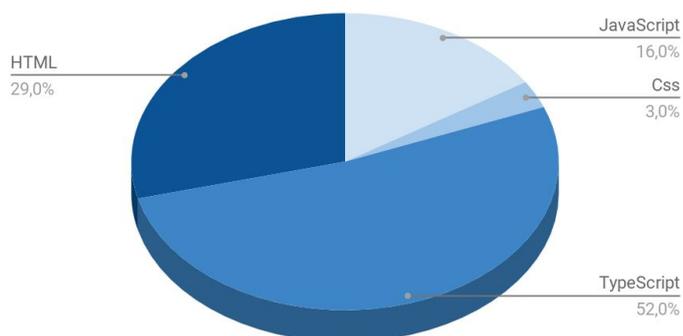
```

214 text files.
197 unique files.
48 files ignored.

github.com/AlDanial/cloc v1.74 T=0.96s (194.0 files/s, 38349.0 lines/s)
-----
Language           files    blank    comment    code
-----
JSON                19         0         0        29482
Typescript          105        632        209        3092
HTML                32         207         4        1568
JavaScript           11         101        220        1045
CSS                  15          33         12         245
Sass                  2           5           0          85
Markdown             3           28           0          34
-----
SUM:                 187       1006        445       35506
-----
    
```

Analyse de GitLab

Langages utilisés



Les langages que nous avons utilisés pour notre application sont le TypeScript, le HTML et le CSS, avec le framework Angular, et enfin le JavaScript, qui lui a été utilisé pour le backend.

CONCLUSION

Ce projet nous a été très enrichissant à plusieurs niveaux. Premièrement, le domaine médical nous principalement inconnu, de ce fait nous en avons appris beaucoup sur le fonctionnement de l'oreille interne et sa rééducation en visitant le Cabinet Vestib+, notre client. De plus, avoir un véritable client et pouvoir communiquer avec lui directement nous a été très instructif. En effet, nous nous sommes retrouvés plusieurs fois dans le cabinet pour établir et discuter du cahier de charge au début du projet.

Deuxièmement, le projet s'est effectué en collaboration avec un groupe de IESE4. Ces derniers étaient en charge du dispositif envoyant les données du mouvement de tête du patient. Nous nous sommes rapidement mis d'accord sur le type de données à envoyer, cependant il nous a fallu de longues discussion et des compromis pour savoir comment elles seront envoyées (WiFi ? Bluetooth ?). Bien sûre ces discussions nous ont appris beaucoup sur le domaine des IESE, n'étant pas habitués à travailler aussi proche du hardware.

Enfin, le dernier plus de ce projet était l'utilisation du framework Angular. Deux de nos membres n'avait jamais eu l'occasion de programmer avec cette technologie lors de précédents projets. C'était donc un moyen pour eux de connaître une technologie très populaire au sein des entreprises. Bien sûr, ils leur a fallu du temps pour la prise en main mais les deux autres membres étant déjà familières avec l'outil, ont pu leur venir en aide et leur donner des conseils. Cela a permis à un bon équilibre dans le groupe mais surtout maintenant chacun de nous est aise avec Angular.

En conclusion, nous sommes parvenu à remplir notre contrat auprès du Cabinet. Même si, nous n'avons pas pu répondre à leurs demandes supplémentaires, nous avons pu respecter le cahier de charge établis lors de notre première réunion. Nous sommes donc satisfaits de nos efforts et nous espérons de l'an projet un autre groupe d'INFO pourra récupérer ce projet pour y ajouter les fonctions supplémentaires.