

Travail de Bachelor 2020

Gestion de données et d'images multispectrales acquises sur la peau



Etudiant : Raphaël Buffet

Professeur : Henning Müller

Résumé

Les systèmes embarqués CosKin servent grâce à différentes LED de couleurs d'analyser la peau et de générer ainsi des images multispectrales. Pour le moment, toutes ces données sont stockées indépendamment sur chaque appareil. Les cabinets d'esthéticiens possèdent plusieurs de ces instruments chez eux. Chaque dispositif fonctionne indépendamment et rend de ce fait la synchronisation des données d'analyse difficiles.

Après observation de cette situation, nous avons défini l'hypothèse suivante : est-il possible de gérer grâce à une API web, des données médicales issues d'appareil d'analyse, pour les visualiser sur un portail tiers.

Dans un premier temps, nous devons synchroniser nos dispositifs d'analyse. Pour ce faire, nous transmettons les données de ces appareils vers un serveur utilisé en web API. Nous pouvons alors synchroniser ces éléments vers tous les autres appareils. Pour permettre la synchronisation sécurisée des données, nous avons défini un système d'authentification. Cela nous permet ainsi de garantir l'obtention exclusive des informations concernant chaque esthéticien.

Dès lors, nous avons développé une application web pour visualiser ces données. Comme pour les appareils, nous avons l'obligation de nous authentifier. Ce portail assure aussi un accès aisé pour le client à ces analyses. Avant disponibilité de cette application, les clients ne pouvaient consulter leurs analyses que chez leur esthéticien.

Tout au long du développement, nous avons dû nous assurer de la protection des données transférées. En effet, ces images étant liées directement à la santé des clients, ils sont considérées comme sensibles.

Mots-clés : API, Application web, Gestion de données, Gestion d'image

Avant-propos

Ce travail a été réalisé à la suite de ma dernière année de bachelor en informatique de gestion. Il a été proposé par Monsieur Henning Müller en collaboration avec CosKin SARL, une entreprise start-up de la HES-SO Valais. La rédaction de ce document a été réalisée de septembre 2020 à février 2021.

En parallèle à ce travail de bachelor, je travaille dans une start-up où je suis cofondateur. Ce sujet en collaboration avec CosKin m'a de ce fait attiré. L'environnement entrepreneurial et l'opportunité d'échange avec une autre jeune entreprise représentait une chance à prendre. L'orientation gestion des données de ce travail m'a aussi influencé sur le choix de ce sujet.

Les objectifs définis sur ce travail sont les suivants :

- Compréhension de l'exploitation des données médicales et cosmétiques
- Stockage de données médicales et cosmétiques
- Communication des appareils CosKin vers un serveur de base de données.
- Visualisation des données grâce à un portail web.

La difficulté majeure réside dans le fait que l'entreprise continue de développer ces appareils durant la rédaction et le développement de ce travail. De ce fait, il sera primordial et nécessaire que je m'adapte selon l'avancée et les changements de développement de l'entreprise. Ce travail dure une demi-année, les changements peuvent alors être fréquents. La communication régulière avec l'entreprise est donc fondamentale.

Pour faciliter la rédaction de ce travail, tous les acteurs décrits seront considérés comme masculin, mais ces désignations concernent aussi bien les hommes que les femmes. Par exemple, nous utiliserons l'appellation esthéticien pour désigner les esthéticiens et leur pendant féminin les esthéticiennes. En outre, les esthéticiens et les médecins possédant les mêmes besoins, nous avons pour faciliter la lecture de ce travail utilisé exclusivement esthéticien pour les désigner.

Remerciements

Je souhaite remercier toutes les personnes qui m'ont soutenu et aidé durant la réalisation de ce travail de bachelor.

Je remercie plus particulièrement les personnes suivantes :

- Monsieur le Docteur Henning Müller, mon professeur pour le suivi et le soutien tout au long de la rédaction et le développement de ce travail.
- Aurélien Siu, développeur chez CosKin SARL pour les développements des modules permettant aux appareils CosKin de communiquer avec ce projet.
- Jimmy Loup CEO chez CosKin SARL pour son soutien informatif ainsi que sa disponibilité tout au long du développement.

Je remercie finalement tous mes proches et amis qui ont pris le temps de relire mon travail avant son rendu. Ils m'ont permis d'éviter les problèmes de compréhension et les erreurs de syntaxe.

TABLE DES MATIÈRES

1. INTRODUCTION	1
1.1. CONTEXTE DU TRAVAIL	1
1.2. COSKIN	1
1.3. STRUCTURE DU TRAVAIL.....	1
2. ORGANISATION DU TRAVAIL	2
2.1. SCRUM.....	2
2.1.1. Utilisation	2
2.2. RÉUNION.....	2
3. ETAT DE L'ART	3
3.1. COSKIN	3
3.2. APPAREIL D'ANALYSE COSKIN	3
3.3. MÉTHODE DE SYNCHRONISATION	5
3.3.1. Cloud.....	5
3.3.2. Synchronisation avec un ordinateur	5
3.3.3. API avec base de données.....	5
3.4. SÉCURITÉ DES DONNÉES MÉDICALES	6
3.4.1. LPD.....	6
4. BESOINS.....	7
4.1. SYNCHRONISATION DES DONNÉES	7
4.2. PORTAIL DE VISUALISATION DES DONNÉES	7
5. MÉTHODES ET LEURS CHOIX	8
5.1. TECHNOLOGIE UTILISÉE SUR LE PORTAIL.....	8
5.1.1. Support de visualisation	8
5.1.2. Type de site web	9
5.1.3. Comparaison des technologies d'application web	10
5.1.4. React.js	11
5.1.5. Babel.js	13
5.2. TECHNOLOGIE UTILISÉE POUR LE SERVEUR	13
5.2.1. Node.js.....	13
5.2.2. Base de données.....	14
5.2.3. Protocole de transfert.....	14
5.3. PLATEFORME DE DÉPLOIEMENT	15

5.3.1.	<i>Infomaniak.....</i>	16
5.3.2.	<i>Heroku</i>	16
5.4.	PARTAGE DU CODE	17
5.4.1.	<i>GitHub.....</i>	17
6.	RÉSULTAT DU DÉVELOPPEMENT	17
6.1.	SERVEUR	17
6.1.1.	<i>Synchronisation des données techniques</i>	17
6.1.2.	<i>Synchronisation des données du client.....</i>	20
6.1.3.	<i>Envoi des images des clients.....</i>	21
6.1.4.	<i>Gestion des erreurs de données.....</i>	22
6.1.5.	<i>Documentation.....</i>	23
6.2.	PORTAIL WEB	23
6.2.1.	<i>Accueil.....</i>	23
6.2.2.	<i>Inscription esthéticien.....</i>	24
6.2.3.	<i>Connexion esthéticien.....</i>	25
6.2.4.	<i>Profil Esthéticien</i>	27
6.2.5.	<i>Liste des patients des esthéticiens.....</i>	28
6.2.6.	<i>Liste des appareils esthéticien</i>	31
6.2.7.	<i>Contact esthéticien.....</i>	31
6.2.8.	<i>Accueil client.....</i>	31
6.2.9.	<i>Inscription client</i>	32
6.2.10.	<i>Connexion client</i>	32
6.2.11.	<i>Profil client.....</i>	33
6.2.12.	<i>Analyse client.....</i>	34
6.2.13.	<i>Contact client.....</i>	34
6.2.14.	<i>Paramètre.....</i>	35
7.	RÉTROSPECTIVE SUR L'ORGANISATION DU DÉVELOPPEMENT	36
7.1.	SPRINT 1	36
7.2.	SPRINT 2	36
7.3.	SPRINT 3	37
7.4.	SPRINT 4	37
8.	CONCLUSION	38
8.1.	VALIDATION DE L'HYPOTHÈSE	38
8.2.	RÉTROSPECTIVE DU DÉVELOPPEMENT	38
8.3.	BILAN POUR L'ENTREPRISE PARTENAIRE.....	39

8.4.	BILAN PERSONNEL	39
9.	RECOMMANDATIONS ET PROCHAINES ÉTAPES	39
9.1.	APPLICATION MOBILE	39
9.2.	FONCTIONNALITÉ PORTAIL	40
9.3.	DESIGN DU PORTAIL	40
9.4.	SÉCURITÉ DES DONNÉES.....	40
10.	DÉCLARATION SUR L'HONNEUR	40
11.	RÉFÉRENCE	41
11.1.	RÉFÉRENCE TEXTUELLE	41
11.2.	RÉFÉRENCE DES TABLEAUX	42
11.3.	RÉFÉRENCE DES ILLUSTRATIONS.....	42
12.	ANNEXE	45
	ANNEXE I : PRODUCT BACKLOG	45
	ANNEXE II : SCHÉMA DE LA BASE DE DONNÉES.....	47
	ANNEXE III : DOCUMENTATION SWAGGER	48

Liste des tableaux

Tableau 1 Comparaison type de technologie	9
Tableau 2 Comparaison React VS Angular	11
Tableau 3 Comparaison HTTP VS FTP (techdifferences, 2016)	15

Liste des figures

Figure 1 logo CosKin	1
Figure 2 Appareil Coskin.....	3
Figure 3 Raspberry Pi de l'appareil	4
Figure 4 Dispositif de capture d'image	4
Figure 5 Schéma de connexion	8
Figure 6 Logo React	11
Figure 7 Logo Angular.....	11
Figure 8 Code de la page d'oubli de mot de passe	12
Figure 9 Logo Babel	13
Figure 10 Logo Node	14
Figure 11 Logo Maria DB.....	14
Figure 12 Logo Infomaniak	16
Figure 13 Logo Heroku.....	16
Figure 14 Logo GitHub	17
Figure 15 Exemple de requête pour synchronisation de la dernière version du système	18
Figure 16 Exemple de requête synchronisation des leds	18
Figure 17 Exemple de requête synchronisation des résistors	19
Figure 18 Exemple de requête synchronisation des valeurs de calibration	19
Figure 19 Exemple de requête de synchronisation des informations systèmes.....	20
Figure 20 Exemple de synchronisation des clients.....	21
Figure 21 Exemple d'envoi d'information d'une image.....	22
Figure 22 Gestion d'erreur des données liées	22
Figure 23 Erreur de duplicité des données	23
Figure 24 Ecran d'accueil CosKin	24
Figure 25 Ecran d'inscription Esthéticien	25
Figure 26 Ecran de connexion	25
Figure 27 AsyncStorage exemple de code	26
Figure 28 Login exemple de code	26
Figure 29 Ecran de profil esthéticien.....	27
Figure 30 Ecran de modification du profil esthéticien	28
Figure 31 Ecran de changement de mot de passe	28
Figure 32 Ecran Analyses des clients par date	29
Figure 33 Ecran analyse d'un client	29
Figure 34 Ecran du diagnostic détaillé	30
Figure 35 Ecran du diagnostic par onde	30
Figure 36 Ecran du listing d'appareil	31

Figure 37 Ecran de contact esthéticien	31
Figure 38 Ecran d'inscription client	32
Figure 39 Ecran de profil client	33
Figure 40 Ecran de modification du profil client	33
Figure 41 Ecran d'analyse par date.....	34
Figure 42 Ecran d'analyse par image	34
Figure 43 Ecran de contact client.....	35
Figure 44 Ecran de déconnexion	35
Figure 45 Taches sprint 1	36
Figure 46 Taches sprint 2	37
Figure 47 Taches sprint 3	37
Figure 48 Taches sprint 4	38
Figure 49 Product backlog (1)	45
Figure 50 Product backlog (2)	46
Figure 51 schéma de la Base de données	47
Figure 52 Documentation Swagger (1).....	48
Figure 53 Documentation Swagger (2).....	49

Liste des abréviations

- **API** : Application Programming Interface traduit en français Interface de programmation.
- **DOM** : Document Object Model traduit en français modèle d'objets de document
- **FTP** : File Transfer Protocol traduit en français protocole de transfert de fichier.
- **HTML** : Hypertext Markup Language traduit en français langage de balisage d'hypertexte.
- **HTTP** : Hypertext Transfer Protocol traduit en français protocole de transfert hypertexte.
- **JSON** : JavaScript Object Notation traduit en français objet de notation JavaScript.
- **LPD** : Loi fédérale sur la protection des données
- **MQTT** : Message Queuing Telemetry Transport traduit en français protocole de transport de messages en file d'attente par télémétrie.
- **PAAS** : Platform as a Service traduit en français Plateforme en tant que service.
- **SAPI** : Serveur Application Programming Interface traduit en français Interface de programmation d'applications Serveur
- **SARL** : Société à responsabilité limité
- **SQL** : Structured Query Language traduit en français langage de requête structurée.

Lexique

- **Agile** : Pratiques mettant en avant la collaboration entre des équipes pluridisciplinaires et leurs clients communs
- **Déployer** : Action de publier un travail informatique afin de le rendre disponible à ces utilisateurs.
- **Framework** : Ensemble de composant logiciel concevant une architecture de base pour développer.
- **Image multispectrale** : Image prise avec plusieurs spectres de lumière différente. Dans ce travail, il s'agit d'images prises avec des leds de couleurs distinctes.
- **Interface de programmation** : Ensemble de méthodes proposant des services à un logiciel tiers.
- **Librairie** : Collection de composants informatiques pouvant être intégré au développement.

- **Modèle de données** : Modèle qui décrit la manière dont sont structurées les données dans un système d'information ou une base de données.
- **Plateforme en tant que service** : Plateforme web fournissant une infrastructure physique pour le développement d'applications.
- **Portail** : Site conçu pour demeurer le point d'entrée sur Internet et proposant des services à des utilisateurs
- **Product backlog** : Document regroupant toutes les tâches à effectuer permettant la planification du développement.
- **Protocoles** : Ensemble de règles définissant le mode de communication entre deux ordinateurs.
- **Scrum** : Framework favorisant un cadre de développement de logiciels.
- **Système embarqué** : système électronique et informatique autonome, spécialisé dans une tâche précise.

1. Introduction

1.1. Contexte du travail

Ce travail clôture mon cursus d'informaticien de gestion suivi à la HES-SO Valais de Sierre. Il sert de mise en pratique des connaissances acquises lors de ma formation. Ce travail a démarré le 14 septembre 2020 et se termine le 12 février 2021. L'objectif de ce dernier est la gestion, la visualisation et le stockage d'image multispectrale ainsi que de leurs données liées.

1.2. CosKin

CosKin One SARL est une start-up créée en 2017 par Jimmy Loup ingénieur en électronique spécialisé en analyse d'image et en Machine Learning. Cette entreprise est soutenue par la fondation The Ark basée à Sion. Elle est passée en SARL en janvier 2020.

Elle exerce dans le domaine de l'analyse de la peau grâce à des systèmes embarqués permettant la prise de photo multispectrale (multiples photos avec des spectres de couleurs distinctes). Ce procédé permet d'analyser la peau en profondeur, grâce à la mise en évidence par les couleurs d'anomalie ou de problème esthétique de la peau.

Leurs principaux clients sont les esthéticiens, mais elle travaille aussi avec des médecins pour les maladies de la peau. Leurs appareils sont vendus chez ces personnes qui les utilisent par la suite sur leur patient pour leur conseiller des soins adéquats.



Figure 1 Logo CosKin

1.3. Structure du travail

Ce travail comporte trois parties distinctes : une partie de recherche, une partie d'analyse et une dernière partie de développement informatique. De ce fait, il démarrera par un aperçu des technologies pour le moment utilisées au sein de l'entreprise ainsi que des méthodes existantes pour résoudre cette problématique. La deuxième partie s'intéressera plus en détails aux choix faits pour ce projet suite aux besoins retenus par cette analyse. Il se clôturera par l'explication du développement effectué ainsi que des hypothèses d'améliorations futures.

2. Organisation du Travail

2.1. Scrum

Afin de m'aider sur le développement et la rédaction de ce travail, nous utiliserons certains outils issus de Scrum, une méthodologie Agile. Cette méthodologie à l'inverse d'une méthodologie traditionnelle en cascade fixe le client au centre du projet et non le travail.

2.1.1. Utilisation

L'utilisation de cette méthodologie dans mon travail s'est effectuée comme suit : le travail ainsi que le développement sont séparés en petites itérations durant chacune trois semaines. Ces itérations dans le langage de Scrum sont appelées Sprint. Cela fournit des objectifs à atteindre régulièrement. De ce fait, elle permet de valider constamment le développement avec l'entreprise mandante.

En relation avec cette méthodologie, nous avons rédigé avant le démarrage de ce travail un product backlog résumant toutes les tâches à effectuer. Celles-ci sont ensuite planifiées selon leur importance. Après réalisation, elles doivent être validées par le client dans ce cas CosKin. En cas de refus, la tâche est à refaire à l'itération suivante. Ce product backlog a permis dès le départ de définir les objectifs prioritaires de ce travail. Elle a aussi permis de nous mettre d'accord sur des fonctionnalités optionnelles pouvant en sortir.

Afin de séparer les développements de ce projet, nous avons défini initialement 5 Sprints. Ils dureront approximativement 3 semaines. Nous les avons remplis de manière plus ou moins équitable en termes de durée.

2.2. Réunion

Pour faciliter le suivi de ce projet, une réunion hebdomadaire est organisée avec CosKin pour la partie technique de ce travail. Cette réunion permet de dresser un compte-rendu régulier de l'avancée technique de ce projet. Cependant, le développement sur les systèmes embarqués CosKin sort de mon domaine de compétence. Ces réunions ont alors permis d'avancer conjointement sur le développement de ce projet. Durant ces dernières, nous définissions chaque semaine les besoins de développement des deux côtés pour pouvoir avancer au mieux.

Une seconde réunion est organisée chaque 1 à 2 semaines avec mon professeur responsable. Elle sert au suivi organisationnel du travail ainsi que pour lui dresser un compte-rendu des avancées. Lors de celle-ci, il m'a donné des pistes de résolution sur mes problèmes.

3. Etat de l'art

3.1. CosKin

Aujourd'hui, l'entreprise CosKin gère toutes ces données utilisateurs en local sur leurs appareils. En effet, chacun de leur système embarqué possède une base de données recensant les informations des différents utilisateurs. Cette structure, déjà existante en local, représente une base pour le développement d'une nouvelle base de données externe à l'appareil. Depuis peu, leurs appareils ont la possibilité de se connecter à Internet. Ils ont également développé un cloud regroupant toutes les photos prises. Ce Cloud ne possède pas de structure. Retracer une image d'un client particulier devient alors compliqué. La seule donnée disponible pour retourner à cette image est la date et l'heure de la prise photographique. Le transfert entre ces deux systèmes fonctionne avec le protocole FTP.

L'entreprise a pour objectif de se diversifier avec de nouveaux systèmes embarqués plus spécifiques à certains domaines. Actuellement, elle se focalise sur un appareil analysant le vieillissement de la peau vis-à-vis de l'exposition au soleil.

3.2. Appareil d'analyse CosKin

L'appareil d'analyse CosKin se divise en 3 parties regroupées dans une armature en plastique. Ces trois composants sont un Raspberry Pi, un dispositif de capture d'image et un écran tactile.



Figure 2 Appareil Coskin

La partie centrale de cet appareil est un micro-ordinateur appelée Raspberry Pi qui gère toutes les fonctions. Il est relié à une batterie rechargeable pour son utilisation. Il possède une carte SD qui stocke le logiciel de traitement ainsi que les données.

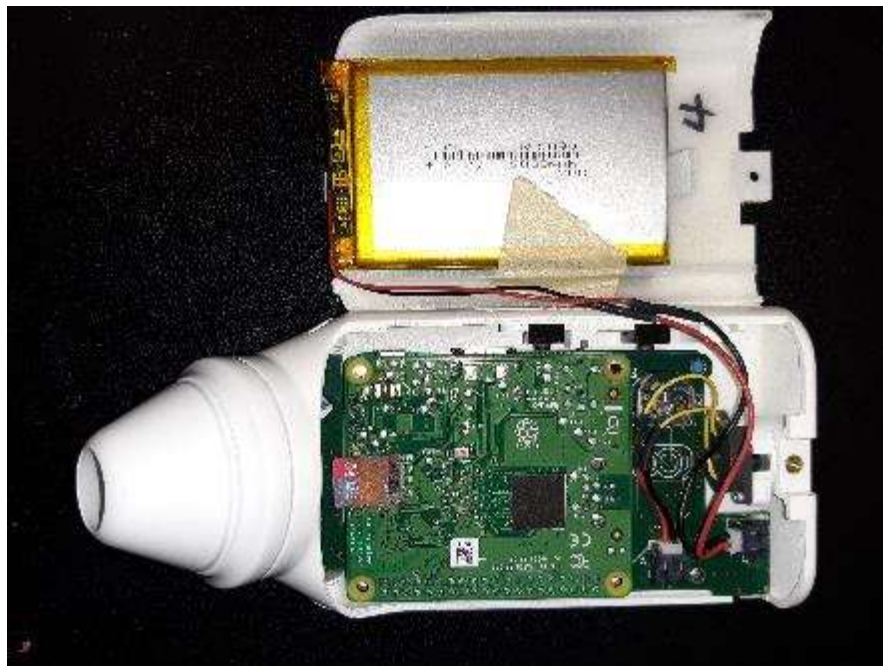


Figure 3 Raspberry Pi de l'appareil

En second lieu, il possède un dispositif pour effectuer des prises d'images de la peau. Celui-ci est composé de 6 LED de couleurs distinctes répliquées 3 fois autour d'une caméra.

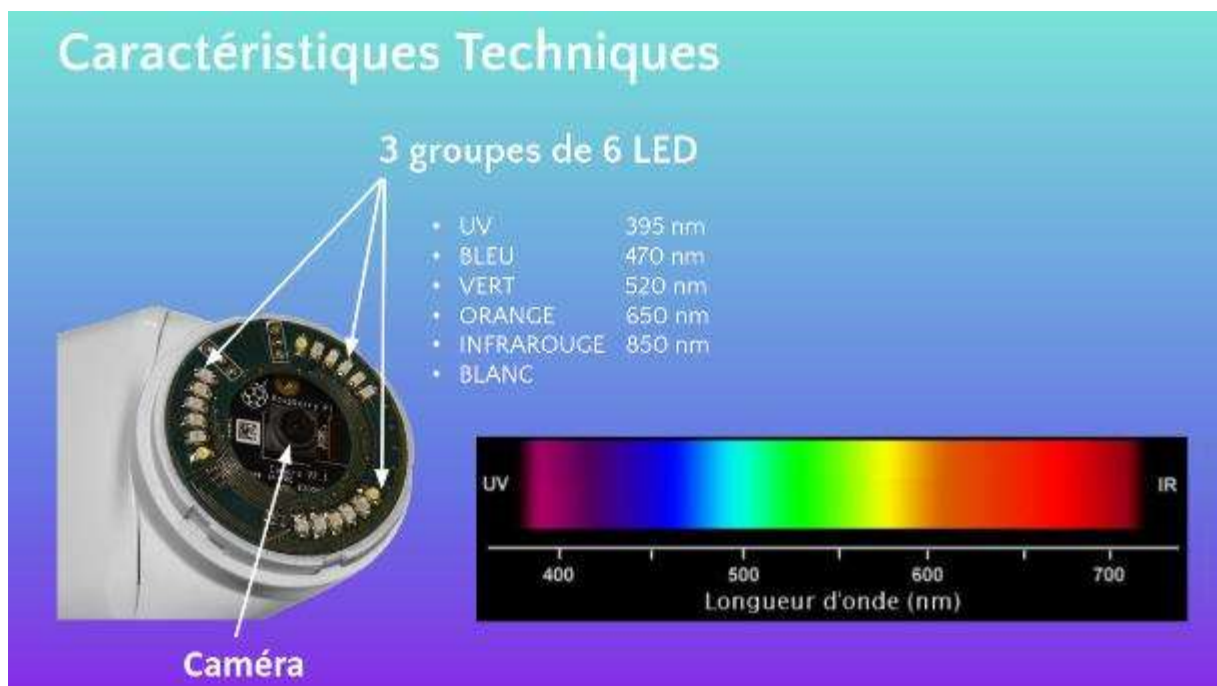


Figure 4 Dispositif de capture d'image

Pour finir, il dispose d'un écran tactile pour favoriser l'interaction avec le système. Cet écran reste cependant trop petit pour permettre l'écriture de rapport.

3.3. Méthode de synchronisation

Aujourd'hui, il existe une multitude de méthodes diverses pour synchroniser ces données. Nous allons lister ci-dessous différents moyens envisageables pour les gérer. Cette analyse permettra de nous situer par rapport au choix de l'hypothèse sur une gestion par Web API.

3.3.1. Cloud

La première méthode envisageable pour synchroniser ces données est de passer par un environnement de Cloud. Nous pouvons citer Dropbox et OneDrive comme exemple actuel. Ces deux plateformes admettent le partage de fichier via un compte sur plusieurs appareils différents. Ils permettent notamment de partager une image entre un smartphone et un ordinateur. Dans ce cas, cette technologie pourrait être envisageable pour le partage d'images. Malheureusement, pour lier les données d'analyse aux images, cela devient vite compliqué à utiliser. Cette méthode reste tout de même envisageable avec une bonne arborescence. Nous entendons par là une structure claire de classement. Par exemple posséder un dossier par client, puis à l'intérieur un dossier par date, pour finalement recueillir l'image avec le nom de l'onde utilisée. Cette image devra alors disposer d'un fichier explicatif contenant toutes les informations d'analyse.

3.3.2. Synchronisation avec un ordinateur

Une seconde méthode existante est de synchroniser les données via un ordinateur. Ce principe stocke les informations récoltées dans le dossier que nous définissons. Par la suite, nous pouvons transférer ces données en connectant un autre appareil à l'ordinateur. Cette méthode fonctionne parfaitement dans une entreprise avec peu d'appareils. La raison à cela est l'obligation de les brancher après chaque analyse. Il permet même avec un logiciel adéquat d'afficher sur l'ordinateur toutes les données de celle-ci. La discussion du diagnostic de la peau du patient peut alors se faire. Il reste cependant inaccessible aux clients hors du cabinet médical.

3.3.3. API avec base de données

La méthode choisie de notre hypothèse est de gérer les données grâce à une SAPI (Serveur Application Programming Interface). Ce type de serveur permet via des requêtes HTTP de récupérer ou de transmettre des données. Les transferts se font majoritairement avec un fichier au format JSON (JavaScript Object Notation). Ce type de document autorise l'envoi de structure de données avec des informations rattachées. Après obtention de celles-ci, le serveur effectue la récupération ou la modification sur la base de données. En règle générale, nous effectuons ces demandes avec des requêtes SQL (Structured Query Language) entre le serveur et la base de données.

Un avantage de ce type de gestion est l'accès par d'autres entreprises à ces fonctionnalités. Cela permettra dans le cas où l'entreprise possède son propre portail de lui donner directement accès à

ces données. Nous restons cependant responsables des données gérées via cette API ainsi que sa sécurité d'accès.

3.4. Sécurité des données médicales

Afin de répondre au mieux à cette problématique, nous avons dû nous renseigner sur les lois qui régissent les données transitées. Etant donné que CosKin est basée en Suisse, nous devons respecter les directives de la LPD (loi fédérale sur la protection des données).

3.4.1. LPD

Comme indiqué précédemment, nous avons dû nous renseigner sur les dispositifs en vigueur concernant le stockage et l'utilisation des données que nous collectons sur ce projet. Nous nous sommes basés sur la version LPD du 1^{er} mars 2019 pour définir les contraintes vis-à-vis de ce projet. Il s'agit de la version en vigueur au moment de l'écriture de ce travail. Nous allons à présent examiner en détails les articles fondamentaux de cette loi vis-à-vis du développement de ce projet.

Selon l'art. 1 de la loi fédérale sur la protection des données du 19 juin 1992 (LPD ; RS 235.1), « La présente loi vise à protéger la personnalité et les droits fondamentaux des personnes qui font l'objet d'un traitement de données. » Cela signifie que nous sommes soumis à cette loi. Nous récupérons les données de différents utilisateurs dans ce travail et nous devons en conséquence respecter cette loi.

Selon l'art. 3c LPD « On entend par données sensibles, les données personnelles sur la santé, la sphère intime ou l'appartenance à une ethnie ». Cet article de loi nous situe sur les données que nous gérons. Elles sont considérées comme sensibles, car nous touchons au domaine de la santé avec les analyses de peau effectuées.

Selon l'art. 7.1 LPD « Les données personnelles doivent être protégées contre tout traitement non autorisé par des mesures organisationnelles et techniques appropriées ». Cela signifie que nous devons protéger les données gérées.

Ces articles de lois servent à nous indiquer par avance les contraintes que nous devons respecter lors du développement. Elle nous aidera aussi dans la prise de décision tout au long du processus de création de ce travail.

4. Besoins

4.1. Synchronisation des données

Pour favoriser la résolution de notre hypothèse, nous devons dans un premier temps synchroniser les données des divers appareils d'analyse. Avant le début de ce travail, chaque appareil CosKin pouvait réaliser des photos et les stocker en local. Ce qui signifie que chaque appareil peut réaliser des photos de la peau, mais il ne peut toutefois que les afficher. L'inconvénient, est que chaque esthéticien doit utiliser le même appareil sur son client afin d'éviter de perdre le suivi de son patient. Ces appareils ayant des paramètres différents selon l'analyse recherchée, cela complique le suivi plus approfondi du client. La synchronisation des données entre les différents appareils est donc devenue nécessaire. Grâce à cette synchronisation, chaque système sera capable de récupérer les images des différents appareils CosKin ainsi que toutes les données relatives.

Ces systèmes ne peuvent se connecter que récemment à Internet. Grâce à cette amélioration, nous pouvons dès à présent connecter ces appareils à un serveur distant, lui-même rattaché à une base de données. Ce développement fait alors directement suite à l'amélioration des appareils CosKin.

4.2. Portail de visualisation des données

Afin de valider la seconde partie de l'hypothèse, nous avons décidé de réaliser un portail permettant de visualiser les analyses de la peau. Ce besoin a aussi été exprimé par l'entreprise partenaire à ce travail.

L'objectif de ce portail est de permettre aux clients d'accéder à leurs données d'analyse sans pour autant devoir prendre rendez-vous chez leur esthéticien. Ce portail sera aussi utile pour ajouter des commentaires et recommandations directement aux patients. Par la suite, les esthéticiens pourront les rediriger vers les produits conseillés pour le maintien ou la réparation de la peau. Ce portail possède aussi son importance dû au fait que l'écran des appareils CosKin ne mesure que 6 cm sur 4,5 cm. Cette taille modeste accepte sans problème la prise de photo sur le système embarqué, mais rend difficile la rédaction de commentaire et la visualisation des données.

Ce portail nécessite d'être accessible par tout le monde et sera développé dans un premier temps en web pour toucher la majorité des clients. Pour faciliter la compréhension de toutes les relations parlées jusqu'à présent, la figure 5 présente ci-dessous résume visuellement ces connexions. Seul l'appareil CosKin était existant avant la mise en place de ce schéma. Tout le reste est à développer durant ce travail.

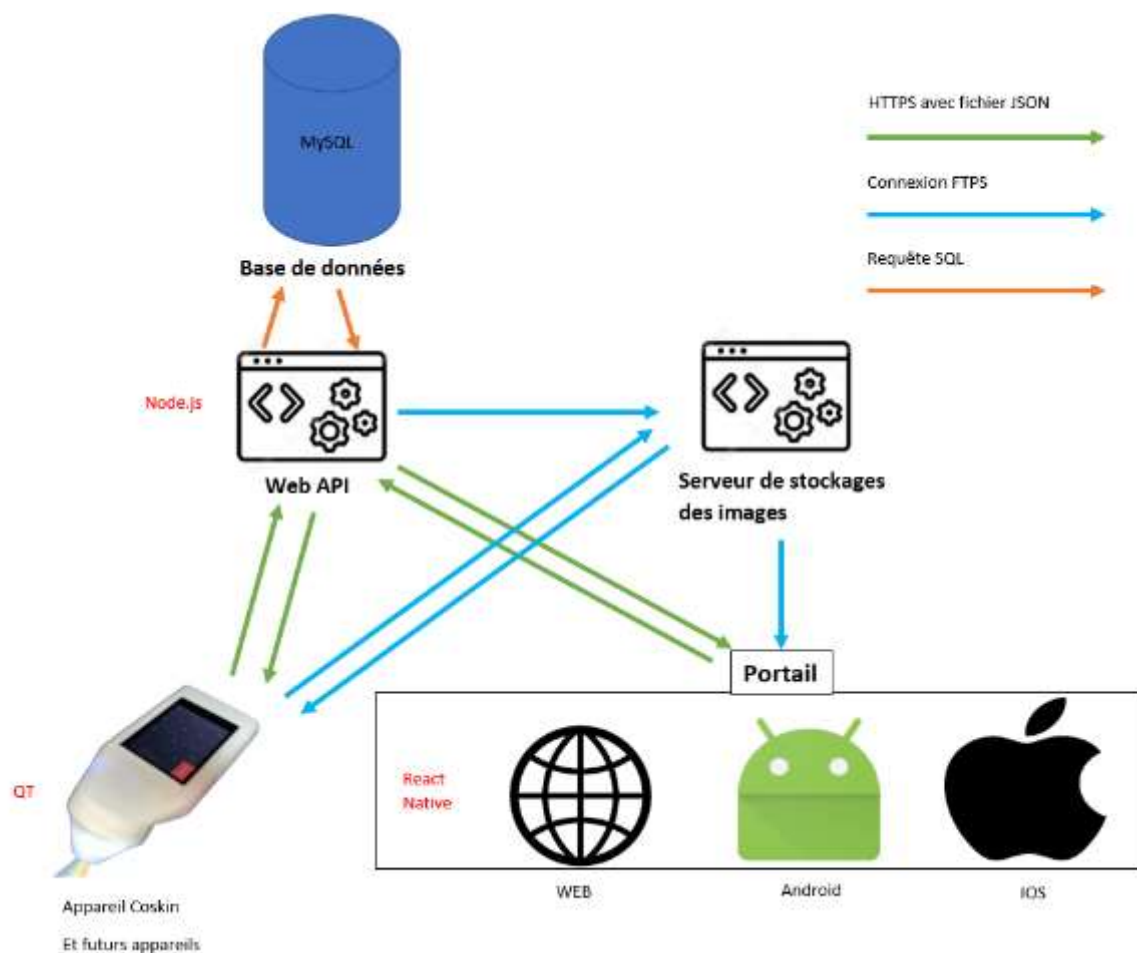


Figure 5 Schéma de connexion

5. Méthodes et leurs choix

5.1. Technologie utilisée sur le portail

5.1.1. Support de visualisation

Afin de pouvoir visualiser les données stockées sur le serveur, nous devons développer un portail. Les deux supports les plus utilisés sont l'application mobile et le site web. Après discussion avec CosKin, nous sommes partis dans un premier temps sur le développement d'un portail web. Par ailleurs, nous n'excluons pas un futur déploiement en application mobile Android et Apple. Ce choix a été fait, car le site web est disponible sur ordinateur et téléphone via un navigateur. En revanche, une application mobile est exclusive d'accès au téléphone portable. De ce fait, la solution la plus optimale est de lire ce portail sur un navigateur web pour toucher la majeure partie des futurs utilisateurs.

5.1.2. Type de site web

Pour ce développement, deux types de conceptions sont disponibles. Nous pouvons partir sur une technologie conçue exclusivement pour le web qui donnera un site web. Nous pouvons aussi nous diriger vers une technologie hybride appelée application web progressive.

Ces deux technologies fonctionnent avec des composants initialement conçus pour le développement web. En revanche, la structure de programmation utilisée pour une application web permet une migration sur une application mobile. Malgré ce point fort d'utilisation multiple, l'application web reste limitée dans les fonctionnalités de personnalisation. En effet, les composants utilisés pour ce type de développement doivent être compatibles pour Android, iOS et le déploiement web. Ceci limite grandement les degrés de perfectionnement du design.

Pour le côté utilisation de ces deux types de structure, nous avons des divergences. Le site web traditionnel fait principalement office de vitrine. Nous entendons par là que nous visualisons toutes les informations mises à disposition sans trop d'interaction. En revanche une application web demande plus d'actions à l'utilisateur. Pour citer un exemple, une application web oblige la connexion pour accéder aux pages du site. Inversement, sur un site web l'utilisateur peut s'enregistrer. Cependant, il dispose d'accès à des pages du site ne nécessitant pas de privilège lié à un compte.

Du point de vue développeur, la maintenance d'une application mobile reste plus complexe. En effet, par rapport au site traditionnel l'application web doit être recompilée avant son déploiement. Cela signifie que nous devons reconstruire depuis le code source toutes les pages avant de les envoyer à notre hébergeur.

Critères	Site web	Application web
Personnalisation	Personnalisation avancée	Personnalisation basique
Déploiement	Seulement web	Web et mobile
Utilisation	Contenu statique, toutes les données représentent des pages accessibles à tous les visiteurs.	Conçue pour interagir avec un utilisateur qui doit se connecter pour accéder aux informations.
Compilation	Le site web est immédiatement accessible en cas de mise à jour.	L'application web doit être recompilée depuis son lieu de stockage avant chaque déploiement

Tableau 1 Comparaison type de technologie

Après discussion et l'envie de développer une application mobile ultérieurement, nous avons opté pour une technologie d'application web. Ce type de technologie nous permettra de récupérer la majorité du développement pour un déploiement futur en application mobile.

5.1.3. Comparaison des technologies d'application web

Pour ce type de développement, les langages de programmation utilisés restent traditionnels au web. Nous développons sur du HTML et du JavaScript. Pour définir le HTML, il représente le langage de base utilisé sur les pages web. Le JavaScript quant à lui est un langage de programmation léger orienté objet. (Contributeurs MDN, 2020) En revanche en comparaison à un site web standard les styles ne sont pas gérés par le langage CSS mais directement par le JavaScript.

Parmi le JavaScript, deux framework sont principalement conçus pour les applications web. Nous avons d'un côté React et de l'autre Angular. Ces deux technologies sont open source, c'est-à-dire que nous disposons gratuitement de leur code source. Plusieurs extensions appelées librairie permettent d'ajouter des fonctionnalités plus avancées. Elles sont développées majoritairement par la communauté.

Afin de procéder à un choix nous allons détailler les différences majeures entre ces deux framework. Premièrement, React à une liaison de données unidirectionnelle. Cela signifie que la structure de la page ainsi que le modèle de données sont envoyés de manière unilatérale vers la page web. (Maria & Kostya, 2020) À l'inverse, Angular présente une gestion bidirectionnelle. Comme pour React, nous envoyons la structure ainsi que le modèle de données à la page. En plus de ce processus, lors de modifications sur la page, nous renvoyons les changements effectués par l'utilisateur au modèle de données. Il s'occupera alors de l'enregistrer. En résumé, pour React, nous devons développer des méthodes de changement de données. Pour Angular, la page va spontanément actualiser les données de l'application.

Nous allons à présent parler des différences vis-à-vis de la DOM (Document Object Model). La DOM représente la structure de votre page web. Il regroupe tous les composants ainsi que leur hiérarchie. La différence entre ces deux technologies se trouve sur le fonctionnement de ce modèle. Pour le React, nous parlons de DOM virtuelle, car nous pouvons transformer la structure des composants sans devoir la recréer entièrement. En revanche, pour Angular, nous utilisons un DOM physique. Résultat, à chaque modification nous démantelons et recréons la structure.

Les deux derniers points pertinents pour la prise de décision entre ces deux technologies sont les mises à jour de version et la taille de la communauté. L'avantage du React comparé à Angular résulte dans sa rétrocompatibilité. Cela signifie que lorsqu'une nouvelle version de ce framework est disponible, les extensions rattachées peuvent pour la majorité être gardées. En ce qui le concerne, Angular demande de mettre à jour toutes ces librairies pour continuer de fonctionner.

Vis-à-vis de la communauté, React en possède une plus vaste. De ce fait, cela nous permet plus aisément de trouver une solution aux problèmes rencontrés sur des sites tels que Stackoverflow ou GitHub. Ces sites sont fréquemment utilisés par les développeurs pour partager leurs travaux ainsi que leurs erreurs. Nous pouvons alors partager les problèmes rencontrés. Dans la majorité des cas, la communauté de ces plateformes est réactive.

Critère	React	Angular
Liaison des données	Unidirectionnelle	Bidirectionnelle
Dom	Virtuelle	Régulière
Mise à jour de version	Rétro compatible	Besoin de mettre à jour
Communauté sur GitHub	146 000 développeurs	59 200 développeurs

Tableau 2 Comparaison React VS Angular

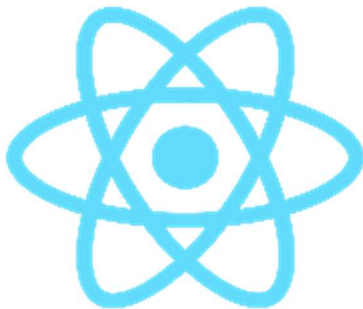


Figure 6 Logo React



Figure 7 Logo Angular

5.1.4. React.js

Suite à cette analyse, nous sommes partis pour la partie portail web sur du JavaScript accompagné du framework React.js. Nous pourrons plus tard ajouter la librairie React native pour favoriser un déploiement en application mobile.

Comme mentionné plus haut, React est une librairie de JavaScript. Elle a été initialement créée par Facebook en 2013. Elle est à présent maintenue par cette entreprise ainsi que par la communauté. Ce framework fonctionne sur le principe de composant indépendant avec des états. Ces composants forment par la suite un tout tel un puzzle pour former une page web ou mobile. Pour faciliter cette compréhension nous allons examiner le composant « mot de passe oublié » qui représente la page de récupération d'un mot de passe perdu.

```

1  import React from 'react';
2  import { Text, View, ScrollView } from "react-native";
3  import { componentstyle, textstyle } from "../assets/Style";
4  import { ButtonRetour, ButtonBig } from '../Component/Button';
5  import Header from '../Component/Header';
6  import { InputForm } from '../Component/Input'
7
8  class MotDePasseOublie extends React.Component {
9    state = {
10      email:"",
11      utilisateur:""
12    }
13    render() {
14      return (
15        <View style={componentstyle.container}>
16          <Header/>
17          <ScrollView>
18            <View style={componentstyle.textcontainer}>
19              <View style={componentstyle.form}>
20                <View style={componentstyle.formelement}>
21                  <Text style={textstyle.text}>Nom d'utilisateur :</Text>
22                </View>
23                <InputForm
24                  onChangeText={(utilisateur) => this.setState({ utilisateur })}
25                  value={this.state.utilisateur}
26                  secure={false}
27                />
28              </View>
29              <View style={componentstyle.form}>
30                <View style={componentstyle.formelement}>
31                  <Text style={textstyle.text}>E-mail :</Text>
32                </View>
33                <InputForm
34                  onChangeText={(email) => this.setState({ email })}
35                  value={this.state.email}
36                  secure={false}
37                />
38              </View>
39            </View>
40            <ButtonRetour onPress={() => this.props.navigation.goBack()}>Retour</ButtonRetour>
41            <ButtonBig onPress={() => this.props.navigation.navigate('Accueil')}>Demande de mot de passe</ButtonBig>
42          </ScrollView>
43        </View>
44      )
45    }
46  }
47  export default MotDePasseOublie

```

Figure 8 Code de la page d'oubli de mot de passe

En premier lieu, sur cette page nous importons tous les composants utiles pour son développement (ligne 1 à 6). Nous définissons ensuite les états pouvant changer sur cette page (ligne 9 à 12). Dans cet exemple, nous avons l'adresse e-mail et le nom d'utilisateur. Cet état changera au remplissage des champs mis à disposition sur la page. Nous disposons ensuite de la méthode « render » qui représente le rendu graphique de la page (ligne 13 à 46). Dans celle-ci, nous appelons les composants importés en début de fichier. Les changements d'état mentionnés précédemment sont représentés par des « setstate » rattachée au composant des zones de texte (ligne 24 et 34). En définitive, nous exportons la page (ligne 47). Cela permet de l'appeler n'importe où dans notre développement et de la transformer en un composant. Il sera par la suite intégré dans le composant général de la connexion au portail.

5.1.5. Babel.js

En plus du React, nous utilisons Babel.js. Il s'agit d'un compilateur JavaScript. Celui-ci est en quelque sorte la pierre angulaire de ce développement. Il permet de traduire le React sur les différentes plateformes. De ce fait, il va traduire le code du projet dans les langages natifs ciblés. Nous pouvons alors faire fonctionner ce projet sur les navigateurs web (Chrome, Explorer, Edge, Firefox, ...) sans problème. Avec le même code, nous pouvons également déployer une application Android/Google ou IOS/Apple. Pour ce deuxième cas de figure, nous devons cependant redéfinir le rendu graphique de ce projet et ajouter la librairie React native.



Figure 9 Logo Babel

5.2. Technologie utilisée pour le serveur

Pour développer cette partie serveur, plusieurs technologies correspondent aux besoins. Nous pouvons partir sur du python facile à implémenter. Nous avons la possibilité de développer sur du PHP procurant une facilité à lier avec des pages web. Parmi cette multitude de technologie nous avons sélectionné le node.js. Ce choix a été fait pour la continuité et l'unité du développement. Cela requiert les mêmes connaissances que pour la partie portail en termes de langage de programmation.

5.2.1. Node.js

Comme expliqué précédemment Node.js est une librairie JavaScript. Elle a été créée en 2010 par Ryan Lienhart Dahl pour faciliter le transfert de données via un serveur. Fonctionnant sur JavaScript, son langage de transmission par défaut est le HTTP. En revanche, différents modules existent pour favoriser l'utilisation d'autres protocoles. La technologie node.js est principalement utilisée pour créer des API web. Cette librairie correspond alors parfaitement vu que l'on souhaite développer une API reliée à une base de données.



Figure 10 Logo Node

5.2.2. Base de données

Pour fonctionner convenablement, ce serveur stockera les données reçues sur une base de données. Pour ce faire, nous avons utilisé le système de gestion Maria DB. Ce système nous permet la gestion complète de la base de données. Pour la rendre disponible en ligne, nous la stockons actuellement sur le service de cloud Amazon appelé RDS. Amazon s'engage selon leur charte à la protection de ces données selon les points décrits sur leur site web. Vous pouvez la consulter à cette adresse : https://docs.aws.amazon.com/fr_fr/AmazonRDS/latest/UserGuide/UsingWithRDS.html



5.2.3. Protocole de transfert

Nous sommes initialement partis sur l'idée de transférer les données des appareils CosKin vers le serveur via le protocole MQTT pour sa rapidité. Malheureusement, la taille des données transmises devenant trop conséquente, nous avons plutôt opté pour un mix entre du FTP et du HTTP. Pour ce faire, nous transférons toutes les données des différents clients depuis les appareils CosKin vers le serveur. Ces données sont stockées sur des fichiers Json générés par l'appareil ou le portail. Par la suite, elles seront envoyées via le protocole HTTPS vers le serveur. Les images doivent conserver une certaine qualité pour obtenir une analyse correcte. Nous avons en conséquence décidé de les transférer via une connexion FTPS vers un serveur de stockage. Cette connexion FTPS autorise le transfert de fichier plus lourd qu'avec du HTTPS. Cependant, ce serveur exigera la création au préalable du dossier de destination.

Le HTTP comme dit précédemment sert d'accès aux ressources disponibles sur le web. Par défaut, tous les navigateurs web utilisent ce protocole pour les adresses web données. Le FTP, lui, sert de passerelle pour le transfert de fichier entre un client et un serveur. Ces deux protocoles établissent une connexion entre le client et le serveur pour l'échange de données. Le FTP crée une deuxième

connexion pour la gestion des transmissions. Du point de vue du transfert le HTTP est limité dans la taille des fichiers mais rien ne vous empêche de créer de multiples connexions au même moment. Vous devez cependant séparer toutes les données en plusieurs fichiers et réussir à faire comprendre au serveur de les fusionner. Le FTP permet quant à lui une taille plus imposante pour les transferts et sera alors privilégié pour le transfert d'image.

BASIS FOR COMPARISON	HTTP	FTP
Basic	HTTP is used to access websites.	FTP transfers file from one host to another.
Connection	HTTP establishes data connection only.	FTP establishes two connection one for data and one for the control connection.
TCP ports	HTTP uses TCP's port number 80.	FTP uses TCP's port number 20 and 21.
URL	If you are using HTTP, http will appear in URL.	If you are using FTP, ftp will appear in URL.
Efficient	HTTP is efficient in transferring smaller files like web pages.	FTP is efficient in transferring larger files.
Authentication	HTTP does not require authentication.	FTP requires a password.
Data	The content transferred to a device using HTTP is not saved to the memory of that device.	The file transferred to the host device using FTP is saved in the memory of that host device.

Tableau 3 Comparaison HTTP VS FTP (techdifferences, 2016)

5.3. Plateforme de déploiement

Pour le déploiement des divers services de ce projet nous avons utilisé deux plateformes différentes. Nous utilisons principalement la plateforme Infomaniak qui met à disposition des étudiants de la HES-SO un hébergement web. Pour compléter les besoins de ce projet, nous utilisons aussi la plateforme Heroku.

5.3.1. Infomaniak

Pour le développement de la partie portail ainsi que pour le stockage des images nous utilisons la plateforme Infomaniak. Infomaniak est un hébergeur web suisse. Cet hébergeur fournit gratuitement aux étudiants un hébergement web durant leurs études. Pour plus d'informations sur les détails techniques de l'offre : <https://news.infomaniak.com/hebergement-gratuit-pour-les-etudiants/>. Grâce à cette offre, nous avons pu héberger la majorité de ce travail.

Comme mentionné plus haut, nous avons déployé le portail sur cet hébergeur avec un nom de domaine mis à disposition par l'entreprise CosKin. Nous avons également investi l'espace de stockage mis à disposition pour synchroniser via une connexion FTPS les images de ce projet. Infomaniak ne fournissant pas gratuitement aux étudiants un hébergement de serveur, nous avons dû trouver une alternative pour publier celui-ci.



Figure 12 Logo Infomaniak

5.3.2. Heroku

Pour le développement de la partie serveur, nous déployons cette Web API sur Heroku. Heroku est une PAAS (plateforme en tant que service). Cela signifie que Heroku possède physiquement l'infrastructure qui gère ce serveur. A la façon dont un hébergeur web propose d'héberger un site web sur ses propres serveurs, cette solution permet de déployer une application sur le Cloud pour favoriser son utilisation. Ce serveur est alors exécuté dans des « dynos », à savoir des ordinateurs virtuels dont la puissance peut être ajustée en fonction des besoins de fonctionnement. (Bastien, 2019)



Figure 13 Logo Heroku

5.4. Partage du code

5.4.1. GitHub

Afin de partager au mieux les avancées techniques sur le projet, nous avons utilisé GitHub un système de partage de code. Grâce à ce service, nous pouvons gérer les versions de développement de ce projet, ainsi que ses autorisations d'accès. Nous l'avons utilisé principalement pour la partie serveur afin de faire tester le serveur en local chez CosKin avec leurs appareils. Les éditeurs de code utilisés nous permettent de publier gratuitement avec un compte le développement effectué sur cette plateforme. Par la suite, nous pouvons récupérer tout ce travail sur n'importe quel ordinateur. Pour cela, nous devons disposer des droits d'accès à ce projet et nous pouvons tout simplement télécharger le code source du projet ainsi que ses dépendances.



Figure 14 Logo GitHub

6. Résultat du développement

6.1. Serveur

Le serveur comme dit plus haut dans ce document a été développé en JavaScript avec le framework node.js. Il a été développé afin de l'utiliser comme une web API. Après discussion avec l'entreprise partenaire, nous avons décidé d'implémenter diverses méthodes de transfert de données décrites ci-dessous. Chaque requête envoyée depuis l'appareil sera asynchrone. Cela signifie que chaque requête peut être envoyée et traitée indépendamment sur le serveur. Cela permet aux appareils CosKin d'être utilisés durant les processus de synchronisation.

6.1.1. Synchronisation des données techniques

Dans un premier temps, les appareils CosKin et le serveur synchronisent leurs données. Leurs premières données à synchroniser sont les données techniques de l'appareil. Cette synchronisation est établie pour permettre de reconnaître les divers composants du hardware qui effectuent la photo de la peau. Cela améliore le suivi de la peau car certains changements sont dus à la modification de la peau et d'autres aux caractéristiques techniques de l'appareil. Un autre point qui rend cette synchronisation nécessaire est que chaque professionnel dispose de plusieurs appareils avec des caractéristiques distinctes. En effet, chaque appareil possède une base de données indépendante.

Cette synchronisation fonctionne comme suit : l'appareil transmet sa version du système au serveur. Le serveur va alors contrôler dans la base de données distante si la version envoyée correspond à la dernière version enregistrée pour cet appareil. Si tel est le cas, le serveur transmettra une réponse négative qui signifie qu'aucune synchronisation est nécessaire.

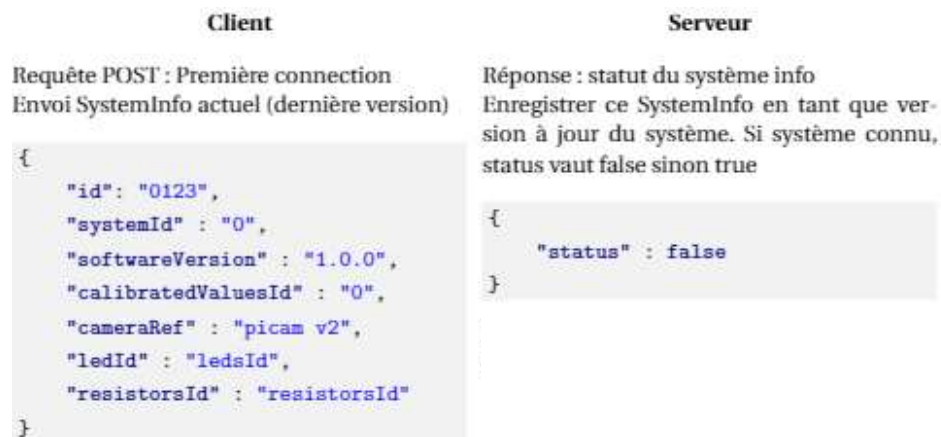


Figure 15 Exemple de requête pour synchronisation de la dernière version du système

Dans le cas contraire, le serveur transmettra une réponse positive qui indique qu'un processus de synchronisation doit avoir lieu. Après confirmation par l'utilisateur, l'appareil va alors transmettre toutes les informations du système. Il commencera par la synchronisation des informations relatives aux leds.

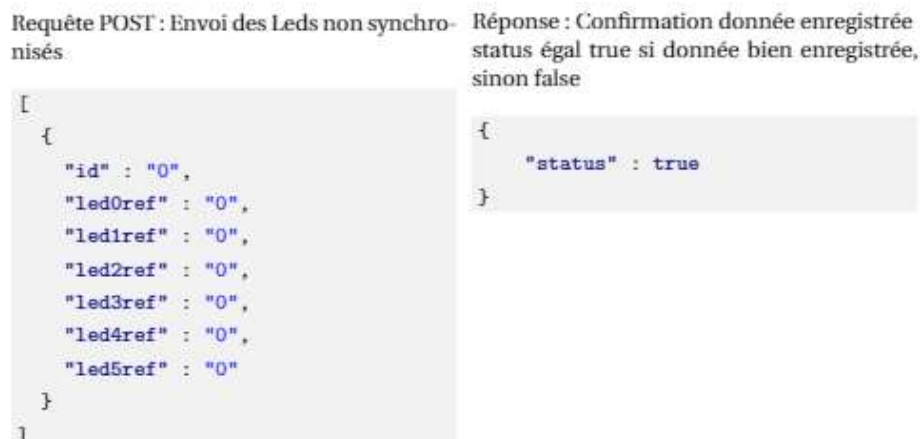


Figure 16 Exemple de requête synchronisation des leds

Par la suite, il s'occupera des informations relatives aux résistances.

Requête POST : Envoi des Resistor non synchronisés

```
[
  {
    "id" : "0",
    "r0ref" : "0",
    "r1ref" : "0",
    "r2ref" : "0",
    "r3ref" : "0",
    "r4ref" : "0",
    "r5ref" : "0"
  }
]
```

Réponse : Confirmation donnée enregistrée
status égal true si donnée bien enregistrée,
sinon false

```
{
  "status" : true
}
```

Figure 17 Exemple de requête synchronisation des résistors

Puis, conclura avec la synchronisation des informations de calibration.

Requête POST : Envoi des CalibratedValues non synchronisés

```
[
  {
    "id" : "0",
    "shutterspeed0" : "0",
    "awb0" : "0",
    "stddev0" : "0",
    "shutterspeed1" : "0",
    "awb1" : "0",
    "stddev1" : "0",
    "shutterspeed2" : "0",
    "awb2" : "0",
    "stddev2" : "0",
    "shutterspeed3" : "0",
    "awb3" : "0",
    "stddev3" : "0",
    "shutterspeed4" : "0",
    "awb4" : "0",
    "stddev4" : "0"
  }
]
```

Réponse : Confirmation donnée enregistrée
status égal true si donnée bien enregistrée,
sinon false

```
{
  "status" : true
}
```

Figure 18 Exemple de requête synchronisation des valeurs de calibration

Toutes ces synchronisations sont gérées simultanément. Dès que toutes les réponses sont positives, nous pouvons synchroniser les données globales du système. Lors de la réception de cette dernière confirmation, le système rendra possible l'envoi d'images.

Requête POST : Envoi des SystemInfo non synchronisés

```
[ {  
  "id": "0123",  
  "systemId" : "0",  
  "softwareVersion" : "1.0.0",  
  "calibratedValuesId" : "0",  
  "cameraRef" : "picam v2",  
  "ledsId" : "ledsId",  
  "resistorsId" : "resistorsId"  
}
```

Réponse : Confirmation donnée enregistrée
status égal true si donnée bien enregistrée,
sinon false

```
{  
  "status" : true  
}
```

Figure 19 Exemple de requête de synchronisation des informations systèmes

6.1.2. Synchronisation des données du client

En parallèle à la synchronisation du système, nous devons aussi synchroniser les données des clients. Cette étape est nécessaire car ces appareils peuvent être utilisés par plusieurs esthéticiens au sein d'un même cabinet. Cette étape commence par l'envoi des identifiants de l'utilisateur sur le serveur. L'appareil enverra par la suite le reste de ses données utilisateurs afin de les synchroniser avec le serveur. A l'intérieur de ces données se trouvera une liste de clients. Le serveur comparera cette liste et renverra les informations des clients manquants sur l'appareil.

Requête POST : Envoi des User avec liste de Client

```
{
  "id" : "user id",
  "username": "Ram",
  "email": "ram@gmail.com",
  "birthdate": "yyyy-MM-dd",
  "phoneNumber": "5489489649",
  "clients" : [
    {
      "id" : "0",
      "userId" : "",
      "firstname" : "",
      "lastname" : "",
      "birthdate" : "yyyy-MM-dd"
    },
    {
      "id" : "1",
      "userId" : "",
      "firstname" : "",
      "lastname" : "",
      "birthdate" : "yyyy-MM-dd"
    }
  ]
}
```

Réponse : Confirmation donnée enregistrée
status égal true si donnée bien enregistrée,
sinon false

```
{
  "status" : true,
  "newClients": [
    {
      "id" : "2",
      "userId" : "",
      "firstname" : "",
      "lastname" : "",
      "birthdate" : "yyyy-MM-dd"
    }
  ]
}
```

Figure 20 Exemple de synchronisation des clients

6.1.3. Envoi des images des clients

La dernière étape est l'envoi des images du client. Cette requête est démarrée manuellement sur l'appareil. Une liste de mesures est envoyée pour chaque client. Après l'obtention de ces informations, le serveur enverra le chemin d'accès afin de pouvoir publier les images sur le serveur. La nomenclature de stockage décidée avec l'entreprise est la suivante : référence client/ timestamp /nom de l'image par exemple : 123/ 5489489649 /UV.jpeg.

Requête POST : Envoi par client, la liste de ses mesures non synchronisées avec le serveur

```
[
  {
    "id": "123",
    "folderPath": "/home/pi/CosKin/c",
    "nbImages": 6,
    "timestamp": "5489489649",
    "clientId": "0123",
    "xCoordinate": 0,
    "yCoordinate": 0,
    "captureMode": 0,
    "note": "",
    "systemInfosId": "",
    "humidity": "20%"
  }
]
```

Réponse : Confirmation donnée enregistrée
status égal true si donnée bien enregistrée,
sinon false

```
{
  "status": true,
  "ftpAddress": "ex.ftp.infomaniak.com/0123",
  "user": "exemple_coskin",
  "password": "password"
}
```

Figure 21 Exemple d'envoi d'information d'une image

Après l'envoi de toutes ces photos, l'appareil pourra demander une synchronisation des photos au serveur. Le serveur expédie alors l'emplacement de toutes celles non présentes relatives au client.

6.1.4. Gestion des erreurs de données

Jusqu'à présent, nous avons traité les requêtes dans le cas de figure où tout fonctionne. Nous allons à présent nous intéresser à la gestion des erreurs de ces requêtes. L'erreur la plus courante demeure la non-présence des informations liées. Par exemple, nous voulons inclure une mesure à cette image, mais la mesure que nous voulons insérer n'existe pas. Nous renvoyons de ce fait un statut « false » à l'appareil CosKin avec un message d'erreur. Dans le cas où la donnée est introuvable, nous stoppons la possibilité d'inclure l'information de l'image.

```
db.query("Select * from measurement where id= ?", [images[i].measurementId])
.then((result)=>{
  console.log("measure")
  console.log(result)
  if(result[0]==undefined){
    console.log(["measurementid doesn't exist"])
    error.push(images[i].id)
    response = { status: false, error: "id measurement doesn't exist", errorid: error }
  }
})
```

Figure 22 Gestion d'erreur des données liées

Une deuxième erreur est la duplication des données. Pour éviter que certaines informations soient envoyées à double sur le serveur, nous avons défini un contrôle des données stockées dans la base de

données. De ce fait, avant l'insertion des données envoyées au serveur, nous contrôlons qu'elle ne soit pas auparavant enregistrée. A cet effet, avant chaque ajout de donnée, nous recherchons par l'identifiant unique la présence ou non de cette information. Si cette donnée est déjà présente nous rajoutons une erreur à la réponse envoyée à l'appareil CosKin. De plus, nous renvoyons les identifiants uniques qui génèrent cette erreur.

```
{
  "status": false,
  "error": "id already exist",
  "errorid": [
    "1"
  ]
}
```

Figure 23 Erreur de duplicité des données

6.1.5. Documentation

Afin de pouvoir garder un suivi sur les méthodes créées sur le serveur, nous avons généré une documentation Swagger. Cette documentation permettra de connaître les futures avancées du projet ainsi que les méthodes disponibles sur le serveur. En outre, cette documentation est générée automatiquement. Ce qui signifie qu'elle s'actualise automatiquement lors d'ajouts ou de suppressions de méthodes sur le serveur. Pour pouvoir accéder à cette liste, il faudra se connecter avec un identifiant administrateur. Par la suite, nous aurons accès à la liste de toutes les méthodes disponibles sur cette API. Un exemple de documentation sera ajouté en annexe de ce travail.

6.2. Portail web

Afin de pouvoir visualiser ces données sans recourir aux systèmes CosKin, nous avons développé en parallèle du serveur un portail Web. Ce portail permet après l'identification de bénéficiaire de l'accès aux différentes données de son profil. Les esthéticiens bénéficient aussi des accès aux données personnelles de tous leurs clients. Bien que le portail soit déployé sur un site, il se rapproche plus d'une application qu'à un réel site web. Nous avons principalement mis l'accent sur les fonctionnalités avant l'affichage. Par conséquent, un travail sur le design devra être fait ultérieurement afin de le rendre plus attractif.

6.2.1. Accueil

En arrivant sur le portail, l'utilisateur accède à l'écran d'accueil. Il aura alors le choix de se diriger vers une page de connexion s'il possède un compte. Dans le cas contraire, il peut naviguer vers un formulaire d'inscription. Nous allons dans un premier temps décrire plus précisément les

fonctionnalités liées aux esthéticiens et ultérieurement, nous passerons en revue les fonctionnalités pour les clients.



Figure 24 Ecran d'accueil CosKin

6.2.2. Inscription esthéticien

Concernant l'inscription sur cette plateforme, l'esthéticien devra préalablement sélectionner son type de compte. Dans cet exemple, il doit sélectionner « Esthéticien ». Puis, il devra renseigner ces informations de nom d'utilisateur et de mot de passe ainsi que son e-mail. Sur cette page, il pourra alors choisir de valider ces informations et s'inscrire. Par ailleurs, s'il s'aperçoit qu'il possède déjà un compte, il aura la possibilité de revenir à la page précédente.

La base de données utilisateur est synchronisée entre les appareils CosKin et le portail. De ce fait, l'utilisateur n'aura pas besoin de posséder un compte distinct sur le portail et sur l'appareil. Cette fonctionnalité d'inscription sera généralement non utilisée dû au fait que les esthéticiens peuvent créer leur compte directement via l'appareil CosKin. Cependant, elle reste indispensable dans le cas où l'esthéticien réalise son compte avant la réception de ses appareils CosKin. Il pourra alors renseigner ces informations personnelles plus tôt dans le processus.

Type de compte : Esthéticien

Nom d'utilisateur :

Mot de passe :

E-mail :

Retour

S'inscrire

Figure 25 Ecran d'inscription Esthéticien

6.2.3. Connexion esthéticien

Dans le cas où l'utilisateur possède déjà un compte, il se retrouvera sur l'écran de connexion du portail. Sur cette page, l'utilisateur renseignera son nom d'utilisateur, son mot de passe et son type de compte. Une redirection vers une page de réinitialisation du mot de passe est disponible pour les utilisateurs l'ayant oublié. Après remplissage de ces champs, l'utilisateur transmettra ces informations aux serveurs. Par la suite, l'utilisateur accédera à son profil ainsi qu'à la navigation du portail. Comme pour l'inscription, l'utilisateur peut retourner à l'écran précédent grâce à un bouton retour disponible sur la page.

Pour information, le bouton retour est nécessaire car nous sommes sur une gestion dynamique des pages. De ce fait, il n'y a qu'une adresse web unique pour tout le portail. L'avantage est que l'utilisateur est guidé tout au long de sa navigation. Il ne pourra pas accéder à une page spécifique en modifiant manuellement l'adresse dans son navigateur. Ce dernier point permet de garantir que l'utilisateur n'accède à une page dont il ne dispose pas les accès.

Type : Esthéticien (-ne)

Nom d'utilisateur :

Mot de passe :

Mot de passe oublié ?

Retour

Se Connecter

Figure 26 Ecran de connexion

En parallèle à la connexion avec le compte, nous recueillons toutes les informations de l'utilisateur via des requêtes adressées au serveur. Ces informations seront alors enregistrées dans le composant « AsyncStorage ». Cela permet avec le cache du navigateur de récupérer toutes les données de l'utilisateur impliquant l'état de connexion. Résultat, l'accès à son profil s'effectue sans se reconnecter à chaque visite.

Pour faciliter la compréhension, nous allons vous montrer directement dans le code ce qui se passe. Avant tout, nous avons défini la méthode « componentDidMount » qui se lance quand l'application web est lancée. De ce fait, nous pouvons accéder aux données du cache via l'AsyncStorage où nous récupérons tout ce qui concerne l'utilisateur. Ils seront alors enregistrés dans les propriétés de l'application. Après validation de ces étapes, nous transformons l'état de l'application pour définir que nous n'attendons plus de chargement de données.

```
componentDidMount() {
  AsyncStorage.getItem('user')
    .then(user =>{
      if(user != null){
        this.props.reduxUpdateUser(JSON.parse(user));
      }
      this.setState({
        loading:false
      });
    });
}
```

Figure 27 AsyncStorage exemple de code

La méthode « rendercurrentstate » correspond au retour visuel de l'application. Dans celle-ci, nous contrôlons si l'application est en chargement. Quand le chargement est terminé, nous lisons les propriétés de l'application où sont stockées les informations de l'utilisateur. En second lieu, nous renvoyons les fonctionnalités selon le type de clients. Dans l'objectif de maintenir une sécurité, nous définissons une expiration de ces données. Ce qui signifie qu'après un certain temps, l'utilisateur devra se reconnecter obligatoirement. Dans le cas où l'utilisateur est introuvable dans le cache, nous renvoyons les pages de connexion/inscription.

```
rendercurrentstate() {
  if(this.state.loading){
    return (<LoadingScreen/>)
  }
  else if (this.props.user.isLoggedIn || this.state.login) {
    return (
      this.state.client ?
      <View style={{flex:1}}><AppContainerClient/></View> :
      <View style={{flex:1}}><AppContainerUser/></View>
    )
  }
  console.log("login")
  return (
    <LoginScreen />
  )
}
```

Figure 28 Login exemple de code

6.2.4. Profil Esthéticien

Une fois la connexion réussie sur le portail, nous accédons immédiatement au profil. Sur cette page s'affichent toutes les informations courantes. Du côté esthéticien, nous retrouvons le nombre de patients que nous avons pour le moment, mais également tous les appareils que nous possédons. Au besoin, nous pouvons modifier les données ou le mot de passe.

Votre profil

Nom d'utilisateur:	Raphael Buffet
Adresse e-mail :	raphael.buffet@stdents.hevs.ch
Numéro de téléphone :	0789354775
Nombre de patient :	2
Nombre d'appareil :	3

Modifier mes données

Changer mon mot de passe

Figure 29 Ecran de profil esthéticien

Nom d'utilisateur:	<input type="text" value="Raphael Buffet"/>
Adresse e-mail :	<input type="text" value="exemple@bla.com"/>
Numéro de téléphone :	<input type="text" value="0789354775"/>
Nombre de patient :	2
Nombre d'appareil :	3
<input type="button" value="Enregistrer"/>	

Figure 30 Ecran de modification du profil esthéticien

Nom d'utilisateur:	Raphael Buffet
Adresse e-mail :	raphael.buffet@stdents.hevs.ch
Numéro de téléphone :	0789354775
Nombre de patient :	2
Nombre d'appareil :	3
Ancien mot de passe	<input type="password"/>
Nouveau mot de passe	<input type="password"/>
Nouveau mot de passe	<input type="password"/>

Figure 31 Ecran de changement de mot de passe

6.2.5. Liste des patients des esthéticiens

Le second onglet concerne la gestion des patients. Sur cette page, nous pouvons voir la liste de tous les clients ainsi que toutes les analyses qui s'y rattachent. Par défaut, elles sont triées par date des plus récentes au plus anciennes. Nous retrouvons la date ainsi que les informations de l'appareil

ayant pris la mesure. Il est fondamental de disposer de ces informations pour identifier les potentielles différences d'images dues à un changement de version. Nous pouvons alors sélectionner le rapport qui nous intéresse afin d'afficher les détails et modifier le diagnostic pour un patient.

Analyse			
Date <input type="text"/>			
<u>Raphael Michaud</u>			
10.02.2021	2	4.0.0	Rapport
11.01.2021	3	2.0.0	Rapport
12.12.2020	1	1.0.0	Rapport
<u>Raphy Buffet</u>			
09.11.2020	2	4.0.0	Rapport

Figure 32 Ecran Analyses des clients par date

Les informations globales de ce rapport se trouvent sur cette nouvelle page d'analyse. A cette date, nous voyons toutes les images prises rattachées à l'onde de la photo. Une zone de texte est disponible pour rédiger le diagnostic global de ce rapport. Plus tard, ce rapport pourra être téléchargé par le patient concerné via le portail. Pour finir, nous trouvons une note indicative sur l'état de la peau du patient ainsi que son taux d'humidité. Cette valeur est calculée grâce à un capteur situé à l'arrière de l'appareil CosKin.


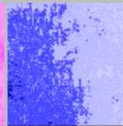
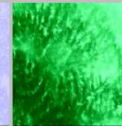
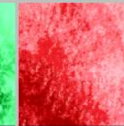
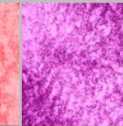
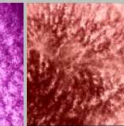
<u>Raphy Buffet</u> <u>Analyse du 09.11.2020</u>					
<u>Images</u>					
					
UV	Bleu	Vert	Orange	IR	Blanc
<u>Diagnostic générale</u>					
<div></div>					
Humidité de la peau : 5% Score générale : 9/10					

Figure 33 Ecran analyse d'un client

En cliquant sur l'une des images de l'analyse, nous pouvons accéder à ces détails. Sur cette nouvelle page, nous retrouvons l'image plus grande avec les différents critères analysés. Tous ces critères attribueront une note. La moyenne de toutes ces notes attribuera un score pour cette image. Le score affiché sur le rapport global représente de ce fait la moyenne des notes de toutes les images du rapport. Comme pour le diagnostic global, nous retrouvons l'humidité de la peau. Ici, l'esthéticien peut rédiger une analyse plus précise sur cette image particulière s'il détecte par exemple une anomalie.

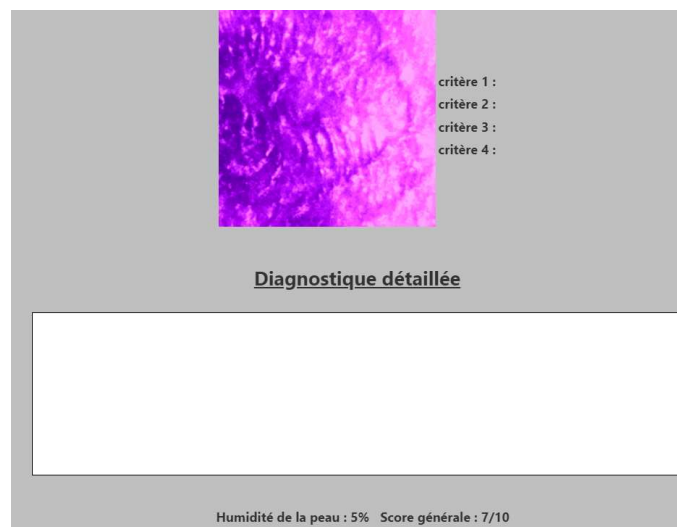


Figure 34 Ecran du diagnostic détaillé

Dans le cas où nous avons trié par ondes d'image, nous arriverons sur l'évolution de la peau par date de la longueur d'onde. L'esthéticien peut alors rédiger un commentaire d'amélioration ou de détérioration de la peau. De plus, comme pour les analyses précédentes, il peut cliquer sur l'image pour aller plus dans les détails d'une date en particulier.

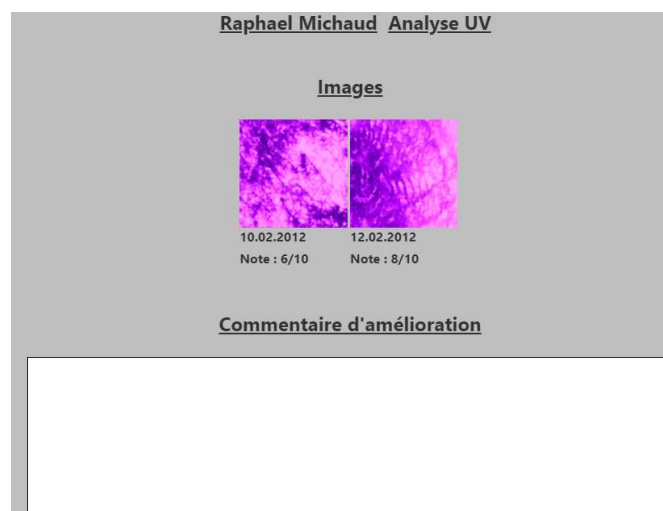


Figure 35 Ecran du diagnostic par onde

6.2.6. Liste des appareils esthéticien

Sous le troisième onglet de ce portail, nous trouvons un listing de tous les appareils utilisés par l'esthéticien. Il s'agit d'un bref récapitulatif de la référence de l'appareil avec sa version et la date de sa dernière synchronisation.

Appareils		
Référence de l'appareil	Version du Système	Dernière synchronisation
1	4.0.0	6.02.2021
2	1.0.0	2.03.2021
3	4.0.0	6.02.2021

Figure 36 Ecran du listing d'appareil

6.2.7. Contact esthéticien

Le 4e onglet contient les informations de contact en cas de problème. Il contient les informations directes de contact de CosKin.

Contact

En cas de problème ou de questions vous pouvez nous contacter

par e-mail : info@coskin.ch

par téléphone : **079 467 70 72**

Figure 37 Ecran de contact esthéticien

6.2.8. Accueil client

Nous allons désormais examiner plus en détail les différentes fonctionnalités disponibles par les clients sur le portail. Le client partage le même écran d'accueil que l'esthéticien, par conséquent, il a le choix entre se connecter sur la plateforme ou s'inscrire.

6.2.9. Inscription client

Pour l'inscription sur la plateforme, le client devra avant tout changer le type de compte sur « client ». Par la suite, il devra renseigner ces informations de nom d'utilisateur et mot de passe ainsi que son e-mail. A la différence de l'esthéticien, le client devra ajouter sa date de naissance. Cette information servira à définir son âge pour les futures analyses. En outre, Il devra aussi désigner son médecin référent parmi les médecins de la base de données.

Généralement, les esthéticiens engendrent automatiquement un compte client lorsqu'il publie une analyse avec l'appareil CosKin. Cette fonctionnalité sera cependant utile dans le cas où le client conçoit son compte avant son premier rendez-vous chez son esthéticien.

The image shows a web form for client registration. At the top, there is a dropdown menu labeled 'Type de compte :' with 'Client' selected. Below this are four input fields: 'Nom d'utilisateur :', 'Mot de passe :', 'E mail :', and 'Date de naissance :'. The date field is a date picker showing '23/09/1997'. Below the date field is another dropdown menu labeled 'Esthéticien référent :' with 'Inconnu' selected. At the bottom of the form are two large buttons: 'Retour' and 'S'inscrire'.

Figure 38 Ecran d'inscription client

6.2.10. Connexion client

La connexion du client respecte les mêmes règles que l'esthéticien. L'unique différence réside dans le changement de type de compte. Nous recueillons ses données de connexion dans le cache afin d'éviter de se reconnecter à chaque fois. Nous avons défini comme pour l'esthéticien, une durée de validité sur la connexion. Lorsque cette date est passée, le client devra nécessairement se reconnecter. En cas de question sur le principe de fonctionnement, vous pouvez vous référer à la partie connexion esthéticien.

6.2.11. Profil client

Après la connexion réussie sur le portail, nous accédons immédiatement au profil. Sur cette page, s'affiche toutes les informations courantes. Du côté client, nous retrouvons le nom du médecin traitant, le nombre d'analyses effectuées ainsi que la date de la dernière analyse. Comme pour l'esthéticien au besoin, nous pouvons modifier les données ou le mot de passe.



Votre profil

Nom :	Michaud
Prénom :	Raphael
Médecin traitant :	Raphael Buffet
Nombre d'anaylse :	3
Dernière anaylse :	11.01.2021

Modifier mes données

Changer mon mot de passe

Figure 39 Ecran de profil client



Nom :	<input type="text" value="Michaud"/>
Prénom :	<input type="text" value="Raphael"/>
Médecin traitant :	Raphael Buffet
Nombre d'anaylse :	3
Dernière anaylse :	11.01.2021

Enregistrer

Figure 40 Ecran de modification du profil client

6.2.12. Analyse client

Le deuxième onglet disponible pour le client est l'analyse de sa peau. Nous retrouvons par défaut toutes les analyses effectuées des plus récentes aux plus anciennes. Nous récupérons ici par date l'appareil utilisé, sa version ainsi que le médecin ayant fait l'analyse. L'information de la personne ayant fait l'analyse est importante, car certains esthéticiens sont plus spécialisés sur certaines anomalies de la peau. Le client peut alors télécharger le rapport de l'analyse. Le type de document envoyé au client étant encore indéfini, nous téléchargeons actuellement un PDF.

Date ▼				
Date	Appareil	Version	Médecin	Rapport
12.12.2020	A547HF	V12.1	Michaud Muriel	télécharger
25.08.2020	UV123A	UV V11.1	Gifo Boniface	télécharger
12.02.2020	A547HF	V12.1	Gifo Boniface	télécharger

Figure 41 Ecran d'analyse par date

Au besoin, le client peut changer de type de rapport regroupé par l'onde de l'image. Pour se faire, il sélectionne le tri par image au lieu du tri par date. Il aura à sa disposition la liste des rapports par onde d'image. Par la suite, il peut les télécharger. Ces Rapports contiennent l'évolution dans le temps de sa peau par rapport à l'onde d'image choisie.

Image ▼					
UV	Bleu	Vert	Orange	IR	Blanc
télécharger	télécharger	télécharger	télécharger	télécharger	télécharger

Figure 42 Ecran d'analyse par image

6.2.13. Contact client

Sur cette page, nous regroupons les informations de contact pour le client. Nous recueillons les informations pour contacter l'esthéticien actuel du client. En outre, nous nous procurons aussi toutes les informations de contact direct de CosKin.

Votre esthéticien(-ne)

En cas de conseil ou de questions vous pouvez contactez votre médecin

nom : Raphael Buffet

par e-mail : exemple@bla.com

par téléphone : 0789354775

Coskin

En cas de problème ou de questions vous pouvez nous contacter

par e-mail : info@coskin.ch

par téléphone : 079 467 70 72

Figure 43 Ecran de contact client

6.2.14. Paramètre

Le dernier onglet disponible pour les deux types d'utilisateurs sert pour le moment uniquement à se déconnecter du portail afin de changer d'utilisateur. Pour le futur, il pourra aussi être utilisé pour transformer les propriétés du portail tel que le thème ou encore la langue.

Coskin

Déconnexion

Figure 44 Ecran de déconnexion

7. Rétrospective sur l'organisation du développement

Comme annoncé en début de document, nous avons adopté la méthodologie Scrum. Pour rappel, cette méthodologie divise le travail en Sprint représentant les échéances de développement. Nous avons initialement défini 5 Sprints mais la dernière période de développement a été annulée. Les fonctionnalités définies dans cette dernière n'influencent pas le résultat de l'hypothèse de ce travail.

7.1. Sprint 1

L'objectif de cette première échéance était d'avoir une Api connectée à une base de données. Durant cette itération, nous avons développé la structure de la base de données ainsi que notre SAPI. Nous pouvons à la suite de ces développements récupérer les données via le serveur.

ID	Task Name	Resp.	Initial Estimate
6	publier des données sur la Base de données	raphael	5
11	posséder un serveur API	raphael	2
19	posséder une base de donnée	raphael	2
20	publier le serveur API	raphael	5
21	envoyer des requetes à ma WebAPI	raphael	10
30	récupérer les données de la Base de données	raphael	5

Figure 45 Taches sprint 1

7.2. Sprint 2

La deuxième période définie avait comme but la liaison des appareils d'analyse avec le serveur. Nous avons défini avec l'entreprise partenaire les informations d'analyse à stocker. Nous avons par la suite développé toutes les méthodes de publication et de récupération des diverses informations des analyses.

ID	Task Name	Resp.	Initial Estimate
10	recupérer les images sur mon appareil Coskin	raphael	5
14	envoyer les info user au serveur	raphael	8
22	recupérer les info user sur mon appareil Coskin	raphael	5
24	recupérer les informations système sur mon app	raphael	5
29	envoyer les images au serveur	raphael	8
31	envoyer les information système au serveur	raphael	8

Figure 46 Taches sprint 2

7.3. Sprint 3

L'itération suivante consistait à la création du portail web. Durant cette période, nous avons développé les pages pour réceptionner les analyses et les images. Nous avons pris en compte tous les cas de figure de navigation sur le portail.

ID	Task Name	Resp.	Initial Estimate
7	me loguer sur le portail web	raphael	5
9	avoir un listing de mes clients	raphael	5
12	m'inscrire sur le portail web	raphael	5
13	accéder à mon profil	raphael	5
17	accéder à un portail web	raphael	2
18	me déconnecter du portail	raphael	2
23	contacter mon esthéticien	raphael	2
26	avoir un listing de mes appareils	raphael	2
27	accéder à mes analyses	raphael	20

Figure 47 Taches sprint 3

7.4. Sprint 4

Durant cet ultime sprint, nous avons lié nos pages précédemment développées avec la web API. Cela consistait uniquement à redéfinir certaines méthodes de notre serveur pour favoriser l'utilisation de celui-ci avec notre portail.

ID	Task Name	Resp.	Initial Estimate
2	Récupéré les données utilisateurs sur le serveur	raphael	5
8	Récupéré les données de nos clients sur le serveur	raphael	5
15	Récupéré les données des différentes analyses sur le serveur	raphael	5
16	Récupéré les données de contact de mon esthéticien sur le serveur	raphael	5
25	Récupéré les données de mes appareils sur le serveur	raphael	5
28	Récupéré les données des analyses de mes clients sur le serveur	raphael	5

Figure 48 Taches sprint 4

8. Conclusion

8.1. Validation de l'hypothèse

Tout au long de ce travail, nous avons traité de la gestion de données via une web API. L'hypothèse formulée au début de ce travail a été validée. Nous recueillons toutes nos informations d'analyses médicales et les visualisons sur notre portail. La gestion par web API a pu être prouvée.

8.2. Rétrospective du développement

Avant le début de ce travail, nous avions des systèmes embarqués qui fonctionnaient indépendamment les uns des autres. Nous disposons désormais d'appareils d'analyse interconnectés ainsi qu'un portail web pour visualiser ces données. Ces appareils ne présentant pas tous la même utilité, il est ardu de lier les différentes analyses d'un même client entre elles. À présent, les analyses peuvent être regroupées et forment un état global de la personne.

Le serveur API développé durant ce travail assure parfaitement le service de synchronisation des appareils. Il crée dorénavant une interconnexion entre les appareils CosKin. Avant ce développement, chaque appareil devenait dépendant de son esthéticien et de son patient. Il rend désormais possible l'utilisation d'appareils différents sur un même client. Dès lors, il permet aisément de consulter toutes ces analyses sur une application tierce.

Le portail mis en place tout au long de ce travail rend les diagnostics plus simples d'accès. Les clients devaient auparavant prendre rendez-vous chez leur esthéticien et n'ont accès à leurs informations qu'à ce moment précis. Par la mise en place de ce portail, les clients peuvent dorénavant accéder à leurs analyses depuis chez eux.

8.3. Bilan pour l'entreprise partenaire

Ce travail a permis à l'entreprise CosKin de disposer d'un serveur permettant de synchroniser les données entre les systèmes d'analyse. Cela implique aussi la possibilité aux utilisateurs finaux de consulter leurs propres données. L'interface web mise à disposition est actuellement fonctionnelle et possède une connexion au serveur. Les fonctions basiques sont opérationnelles. Par la suite, l'entreprise CosKin a la possibilité d'utiliser le code mis à disposition. Elle pourra alors ajouter les fonctions complémentaires telles que l'analyse des images de la peau.

8.4. Bilan personnel

D'un point de vue personnel, ce travail m'a permis d'approfondir mes connaissances du JavaScript. En effet, durant ma formation scolaire nous n'avons que peu touché à ce langage. Nous avons auparavant utilisé le framework React sur un projet. J'ai aussi pu perfectionner mes compétences en matière de développement en API. Par ailleurs, j'ai approfondi mes connaissances sur la gestion des données sensibles. Outre ces connaissances techniques apprises, j'ai pu me rendre compte des diverses propriétés de la lumière qui m'étaient auparavant inconnues.

Ce travail m'a surtout permis de me confronter à un projet réellement utilisé. Il permet de sortir des projets effectués jusqu'à présent lors de mes études. Parallèlement à ce travail, j'ai commencé à travailler pour une start-up où je suis cofondateur. J'ai pu alors acquérir tout au long de ce travail des connaissances sur le monde de l'entrepreneuriat.

9. Recommandations et prochaines étapes

Nous allons à présent vous partager certaines idées pour la poursuite et l'amélioration de ce travail. La plupart de ces propositions ne sont pas primordiales pour le fonctionnement de base du projet. Cependant, il serait avantageux de les implémenter dans les développements futurs.

9.1. Application Mobile

La partie portail a été développée en React.js dans le but de faciliter la transition vers une application mobile future. En effet, en ajoutant le module de React native, nous pouvons réutiliser la majorité du code du portail web. Toutes les fonctionnalités de récupération et de synchronisation avec le serveur ne changeront pas. Les seules parties à devoir reprogrammer demeurent les parties purement visuelles. Le développement d'une application mobile est une recommandation, mais ne représente pas une obligation pour le fonctionnement de ce projet. Il permettra uniquement une facilité supplémentaire aux clients.

9.2. Fonctionnalité Portail

Sur le portail, certaines fonctionnalités n'ont pas pu être terminées. La première fonctionnalité qui doit être terminée est l'envoi de mot de passe en cas d'oubli. La page est existante pour récupérer l'e-mail et le nom d'utilisateur. Par contre, aucun système n'a été développé pour l'envoi d'un e-mail de réinitialisation du mot de passe. Une autre fonctionnalité est de télécharger le rapport à la suite des analyses CosKin. La page de téléchargement a été créée, mais du côté de CosKin rien n'a été défini sur le format du rapport.

9.3. Design du portail

La partie portail a été conçue principalement sur les fonctionnalités et les connexions avec le serveur. De ce fait, les parties plus graphiques sont basiques et doivent être améliorées pour rendre le portail plus attractif pour les utilisateurs.

9.4. Sécurité des données

Dans le but d'améliorer la sécurité des données, nous recommandons d'effectuer les requêtes de l'API sur des vues au lieu de les établir sur des tables. Cela signifie que nous devons réaliser une vue par requête dans la base de données. Une vue en base de données est tout simplement un regroupement de données venant de plusieurs tables. Malheureusement, cela implique de modifier toutes les requêtes du serveur.

10. Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après : Monsieur Jimmy Loup et Monsieur le Dr. Henning Müller.

Lieu et date

Signature

(Buffet Raphaël, étudiant HES-SO)

11. Référence

11.1. Référence Textuelle

Contibuteurs MDN. (2020, juin 10). *JavaScript*. Récupéré sur Mozilla MDn Web Docs:
<https://developer.mozilla.org/fr/docs/Web/JavaScript>

cuelogic. (2020). *Angular vs. React: A Complete Comparison Guide*. Récupéré sur cuelogic.com:
<https://www.cuelogic.com/blog/what-are-the-differences-between-angular-and-react>

Différence entre site web et application web. (2019, mars 22). Récupéré sur waytolearnx.com:
<https://waytolearnx.com/2019/03/difference-entre-site-web-et-application-web.html>

Guru99. (2018). *JSON vs XML: What's the Difference?* Récupéré sur guru99:
<https://www.guru99.com/json-vs-xml-difference.html>

Jain, R. (2015). *Messaging Protocols for Internet of Things: MQTT*. Récupéré sur cse.wustl.edu:
https://www.cse.wustl.edu/~jain/cse570-15/ftp/m_14mqt.pdf

L, B. (2019, juillet 18). *Heroku : tout savoir sur le pionnier des PaaS Cloud*. Récupéré sur lebigdata.fr:
<https://www.lebigdata.fr/heroku-definition>

Maria, D., & Kostya, G. (2020, mars 31). *Angular vs. React: What to Choose for Your Web App?*
Récupéré sur cleveroad: <https://www.cleveroad.com/blog/angular-vs-react>

MariaDB foundation. (2019). *Introduction to Maria DB*. Récupéré sur mariadb:
<https://mariadb.com/kb/en/mariadb-server-10-5-6/>

Multiple contributeurs de Stack Overflow. (2020, septembre 10). *eBook pour apprendre heroku (PDF)*.
Récupéré sur riptutorial.com: <https://riptutorial.com/Download/heroku-fr.pdf>

techdifferences. (2016, Novembre 4). *Difference Between HTTP and FTP*. Récupéré sur
techdifferences: <https://techdifferences.com/difference-between-http-and-ftp.html>

Website vs Web App: What's the Difference? (2019, Juillet). Récupéré sur <https://medium.com/>:
<https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4>

11.2. Référence des tableaux

Tableau 1 Comparaison type de technologie réalisé par l'auteur selon les sources suivantes :

- a) Différence entre site web et application web (Différence entre site web et application web, 2019)
- b) Site web vs application web (Website vs Web App: What's the Difference?, 2019)

Tableau 2 Comparaison React VS Angular réalisé par l'auteur selon les sources suivantes :

- a) Angular ou React que choisir pour votre application web (Maria & Kostya, 2020)
- b) Angular et React comparaison complète (cuelogic, 2020)

Tableau 3 comparaison HTTP VS FTP (techdifferences, 2016)

11.3. Référence des illustrations

Figure 1 aperçue sur : <https://CosKin.ch/>

Figure 2 réalisée par l'auteur sur l'appareil mis à disposition par CosKin

Figure 3 réalisée par l'auteur sur l'appareil mis à disposition par CosKin

Figure 4 réalisée par CosKin

Figure 5 réalisée par l'auteur

Figure 6 aperçue sur : <https://www.ubidreams.fr/wp-content/uploads/2020/06/logo-react-js.png>

Figure 7 aperçue sur : <https://fr.wikipedia.org/wiki/Angular>

Figure 8 réalisée par l'auteur sur Visual studio code

Figure 9 aperçue sur : <https://babeljs.io/docs/en/>

Figure 10 aperçue sur : https://miro.medium.com/max/4540/1*q9ww_u32hhpMaA-Q_s1ujw.png

Figure 11 aperçue sur : <https://michauko.org/blog/migration-mysql-mariadb-1717/>

Figure 12 aperçue sur : <https://www.hebergeurs-suisse.ch/news/new-infomaniak-prix-domaine-ch-suisse.php>

Figure 13 aperçue sur : <https://dev.to/lucysuddenly/easy-step-by-step-guide-to-deploying-on-heroku-3m9p>

Figure 14 aperçue sur : <https://medium.com/better-programming/a-deep-dive-into-github-actions-51e234da0c50>

Figure 15 réalisée par l'auteur sur www.overleaf.com/

Figure 16 réalisée par l'auteur sur www.overleaf.com/

Figure 17 réalisée par l'auteur sur www.overleaf.com/

Figure 18 réalisée par l'auteur sur www.overleaf.com/

Figure 19 réalisée par l'auteur sur www.overleaf.com/

Figure 20 réalisée par l'auteur sur www.overleaf.com/

Figure 21 réalisée par l'auteur sur www.overleaf.com/

Figure 22 réalisée par l'auteur sur Visual Studio code

Figure 23 réalisée par l'auteur sur Visual Studio code

Figure 24 réalisée par l'auteur sur le portail web

Figure 25 réalisée par l'auteur sur le portail web

Figure 26 réalisée par l'auteur sur le portail web

Figure 27 réalisée par l'auteur sur Visual Studio code

Figure 28 réalisée par l'auteur sur Visual Studio code

Figure 29 réalisée par l'auteur sur Visual Studio code

Figure 30 réalisée par l'auteur sur le portail web

Figure 31 réalisée par l'auteur sur le portail web

Figure 32 réalisée par l'auteur sur le portail web

Figure 33 réalisée par l'auteur sur le portail web

Figure 34 réalisée par l'auteur sur le portail web

Figure 35 réalisée par l'auteur sur le portail web

Figure 36 réalisée par l'auteur sur le portail web

Figure 37 réalisée par l'auteur sur le portail web

Figure 38 réalisée par l'auteur sur le portail web

Figure 39 réalisée par l'auteur sur le portail web

Figure 40 réalisée par l'auteur sur le portail web

Figure 41 réalisée par l'auteur sur le portail web

Figure 42 réalisée par l'auteur sur le portail web

Figure 43 réalisée par l'auteur sur le portail web

Figure 44 réalisée par l'auteur sur le portail web

Figure 45 réalisée par l'auteur sur le Product backlog

Figure 46 réalisée par l'auteur sur le Product backlog

Figure 47 réalisée par l'auteur sur le Product backlog

Figure 48 réalisée par l'auteur sur le Product backlog

Figure 49 réalisée par l'auteur sur le Product backlog

Figure 50 réalisée par l'auteur sur le Product backlog

Figure 51 réalisée par l'auteur sur <https://app.dbdesigner.net/>

Figure 52 réalisée par l'auteur sur le serveur

Figure 53 réalisée par l'auteur sur le serveur

12. Annexe

Annexe I : Product backlog

US Nr.	Theme	utilisateur Stories			Acceptance Criteria	Priority	Status	Story Points	Sprint	US accepted (done done)	MoSCoW
		en tant que	Je veux	Afin de							
1	Portail	User	Supprimer un client	Mettre fin au suivi d'un client et libérer de la place sur le server	Utilisateur plus présent sur la Base de données	310	<div></div>	1	5		Could
2	Portail/Serveur	User	Récupéré les données utilisateurs sur le serveur	Avoir les données actuelles de mon compte	Données présente identique à ceux présentes sur le serveur	230	<div></div>	5	4	done	Must
3	Portail	Super utilisateurs(admin)	Supprimer des comptes	Libérer de la place sur le server	Utilisateur plus présent sur la Base de données	280	<div></div>	2	5		Could
4	Portail	Super utilisateurs(admin)	Ajouter des nouvelles analyse	Satisfaire les demandes de la clientèle	Analyse présente sur le portail	290	<div></div>	5	5		Could
5	Portail	Super utilisateurs(admin)	Changer le médecin/esthéticien d'un client	Donner toutes les infos du client à autre médecin/esthéticien	Donnée modifier dans la base de données	300	<div></div>	5	5		Could
6	Base de données	Développeur	Publier des données sur la Base de données	Gérer ces données sur ma Web API	Donnée enregistrer correctement dans la Base de données	60	<div></div>	5	1	done	Must
7	Portail	User	Me loguer sur le portail web	Accéder à mes données	Accès au portail	140	<div></div>	5	3	done	Must
8	Portail/Serveur	User	Récupéré les données de nos clients sur le serveur	Avoir les données actuelles de mon client	Données présente identique à ceux présentes sur le serveur	220	<div></div>	5	4	done	Must
9	Portail	User	Avoir un listing de mes clients	Commenter les analyses de la peau de mes clients	Liste de mes clients et accès à chaque analyse	180	<div></div>	5	3	done	Must
10	Serveur	User	Récupérer les images sur mon appareil Coskin	Mettre à jour mes images systèmes sur mon appareil	Donnée enregistrer correctement dans la Base de données	90	<div></div>	5	2	done	Must
11	Serveur	Développeur	Posséder un serveur API	Appeler mes méthodes de récupération et envoi de données	Web Api créer	20	<div></div>	2	1	done	Must
12	Portail	User	M'inscrire sur le portail web	Créer un compte pour synchroniser mes données	Compte créer sur le serveur	150	<div></div>	5	3	done	Must
13	Portail	User	Accéder à mon profil	Modifier mes informations personnelles	Information personnel affichée et synchronisée avec le serveur en cas de modification	160	<div></div>	5	3	done	Must
14	Serveur	User	Envoyer les infos user au serveur	Mettre à jour mes informations systèmes sur le serveur	Donnée récupérer sur l'appareil Coskin	110	<div></div>	8	2	done	Must
15	Portail/Serveur	Clients	Récupéré les données des différentes analyses sur le serveur	Avoir les données actuelles de mes analyses	Données présente identique à ceux présentes sur le serveur	240	<div></div>	5	4	done	Must

Figure 49 Product backlog (1)

US Nr.	Theme	utilisateur Stories			Acceptance Criteria	Priority	Status	Story Points	Sprint	US accepted (done done)	MoSCoW
		en tant que	Je veux	Afin de							
16	Portail/Serveur	Clients	Récupéré les données de contact de mon esthéticien sur le serveur	Avoir les données actuelles de mon médecin	Données présente identique à ceux présentes sur le serveur	260	●	5	4	done	Should
17	Portail	User	Accéder à un portail web	Visualiser mes données hors des appareils Coskin	Portail web disponible	130	●	2	3	done	Must
18	Portail	User	Me déconnecter du portail	Changer de compte sur le portail	Renvoi de la page d'accueil	200	●	2	3	done	Must
19	Base de données	Développeur	Posséder une base de données	Stocker les données de mes différentes plateforme	Base de données créer	10	●	2	1	done	Must
20	Serveur	Développeur	Publier le serveur API	Avoir accès n'importe ou au service de cette API	Accès distant à la Web API	30	●	5	1	done	Must
21	Serveur	Développeur	Envoyer des requêtes à ma Web API	Me connecter à mon API	Réponse reçus de la web API	40	●	10	1	done	Must
22	Serveur	User	Récupérer les infos user sur mon appareil Coskin	Mettre à jour mes utilisateurs sur mon appareil	Donnée enregistrer correctement dans la Base de données	80	●	5	2	done	Must
23	Portail	Client	Contacter mon esthéticien	Prendre rendez-vous et discuter de mes analyses	Page de contact	210	●	2	3	done	Should
24	Serveur	User	Récupérer les informations système sur mon appareil Coskin	Mettre à jour mes informations systèmes sur mon appareil	Donnée enregistrer correctement dans la Base de données	70	●	5	2	done	Must
25	Portail/Serveur	User	Récupéré les données de mes appareils sur le serveur	Avoir les données actuelles de mes appareils	Données présente identique à ceux présentes sur le serveur	270	●	5	4	done	Should
26	Portail	User	Avoir un listing de mes appareils	Contrôler les versions de mes appareils Coskin	Listes des appareils	190	●	2	3	done	Should
27	Portail	Client	Accéder à mes analyses	Voir mes problèmes cutanés	Liste des analyses effectué	170	●	20	3	done	Must
28	Portail/Serveur	User	Récupéré les données des analyses de mes clients sur le serveur	Avoir les données actuelles des analyses de mes clients	Données présente identique à ceux présentes sur le serveur	250	●	5	4	done	Must
29	Serveur	User	Envoyer les images au serveur	Mettre à jour mes informations systèmes sur le serveur	Donnée récupérer sur l'appareil Coskin	120	●	8	2	done	Must
30	Base de données	Développeur	Récupérer les données de la Base de données	Gérer ces données sur ma Web API	Donnée récupérer sur le server	50	●	5	1	done	Must
31	Serveur	User	Envoyer les information système au serveur	Mettre à jour mes informations systèmes sur le serveur	Donnée récupérer sur l'appareil Coskin	100	●	8	2	done	Must

Figure 50 Product backlog (2)

Annexe II : Schéma de la Base de données

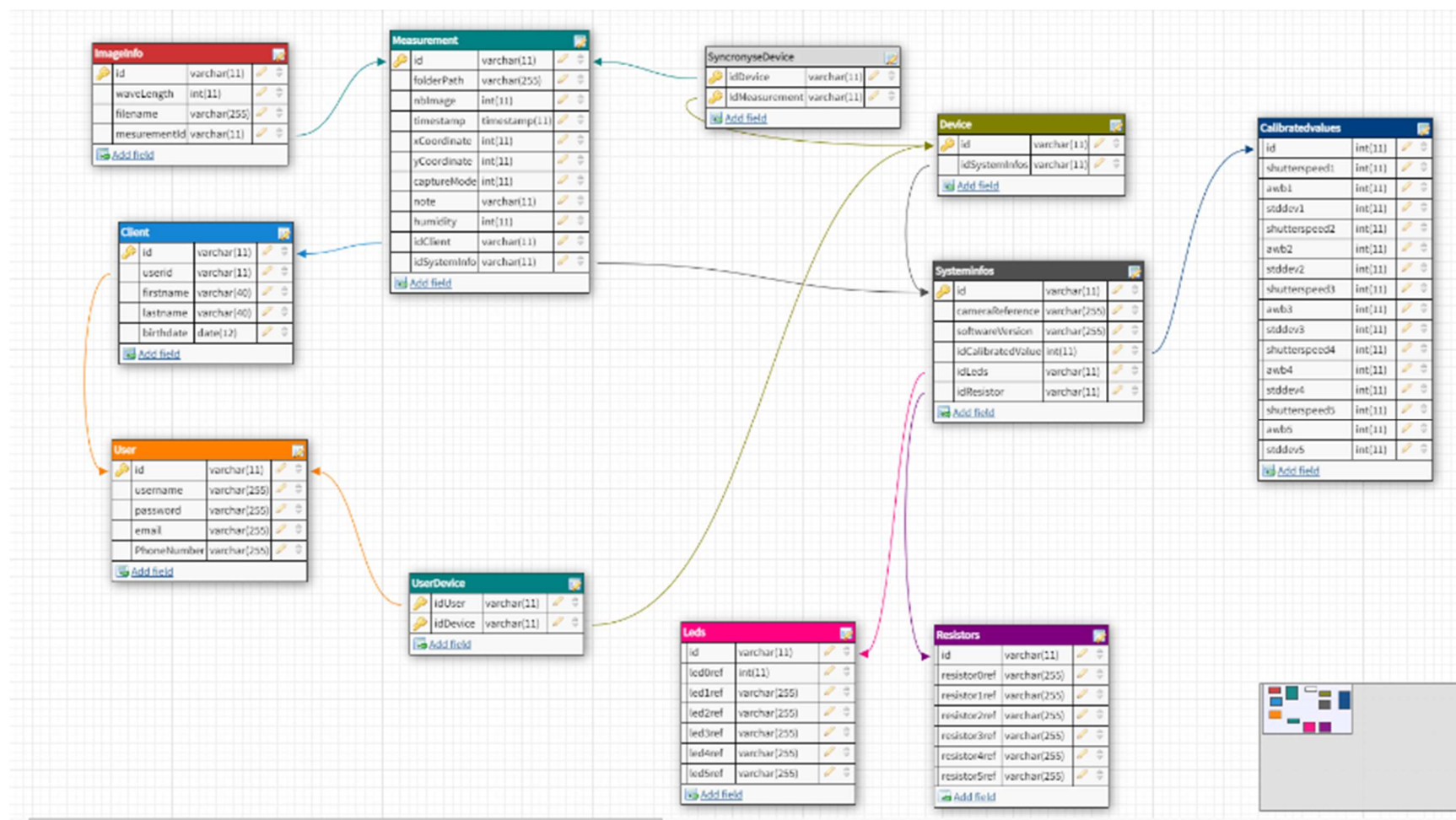


Figure 51 schéma de la Base de données

Annexe III : Documentation Swagger

Appareils	
GET	/api/v1/device/{id} :apiv1device[id]
Client	
GET	/api/v1/client/count/{id} :apiv1clientcount[id]
GET	/api/v1/client/user/{id} :apiv1clientuser[id]
POST	/api/v1/client/login :apiv1clientlogin
POST	/api/v1/client/{id} :apiv1client[id]
GET	/api/v1/client/{id} :apiv1client[id]
POST	/api/v1/client/password/{id} :apiv1clientpassword[id]
GET	/api/v1/client :apiv1client
POST	/api/v1/client :apiv1client
Esthéticien	
POST	/api/v1/user/withclient :apiv1userwithclient
POST	/api/v1/user/login :apiv1userlogin
POST	/api/v1/user/{id} :apiv1user[id]
GET	/api/v1/user/{id} :apiv1user[id]
POST	/api/v1/user/password/{id} :apiv1userpassword[id]
GET	/api/v1/user :apiv1user
POST	/api/v1/user :apiv1user
Information d'Images	
GET	/api/v1/imageinfos/{id} :apiv1imageinfos[id]
GET	/api/v1/imageinfos :apiv1imageinfos
POST	/api/v1/imageinfos :apiv1imageinfos
GET	/api/v1/imageinfos/id/{id} :apiv1imageinfos[id]

Figure 52 Documentation Swagger (1)

Informations Systèmes	
POST	/api/v1/systeminfos/lastSystem :api/v1/systeminfos/lastSystem
POST	/api/v1/systeminfos/synchronize :api/v1/systeminfos/synchronize
POST	/api/v1/systeminfos/updateversion :api/v1/systeminfos/updateversion
GET	/api/v1/systeminfos :api/v1/systeminfos
GET	/api/v1/systeminfos/{id} :api/v1/systeminfos/{id}
LED	
POST	/api/v1/leds :api/v1/leds
GET	/api/v1/leds :api/v1/leds
GET	/api/v1/leds/{id} :api/v1/leds/{id}
Mesures	
POST	/api/v1/measurement :api/v1/measurement
GET	/api/v1/measurement :api/v1/measurement
GET	/api/v1/measurement/{id} :api/v1/measurement/{id}
Resistances	
POST	/api/v1/resistors :api/v1/resistors
GET	/api/v1/resistors :api/v1/resistors
GET	/api/v1/resistors/{id} :api/v1/resistors/{id}
Valeurs de calibrations	
POST	/api/v1/calibratedValues :api/v1/calibratedValues
GET	/api/v1/calibratedValues :api/v1/calibratedValues
GET	/api/v1/calibratedValues/{id} :api/v1/calibratedValues/{id}

Figure 53 Documentation Swagger (2)