Ce document a pour objectif de vous présenter nos démarches de qualité logicielle.

Premièrement, après lecture et étude de l’expression des besoins, nous avons relevé des détails posant problème et en avons discuté avec le client. Tout d’abord, les contraintes sur la réf du matériel ne faisaient pas sens et étaient trop contraignantes pour leur utilité. Nous avons pensé que c’était le résultat du fait que l’expression des besoins avait été rédigée par/comme si c’était quelqu’un qui n’était pas familier avec l’informatique. Nous avons donc finalement permis à la réf d’être une suite de nombres quelconque. Nous avons aussi supprimé le champ obligatoire « version » d’un appareil, car ce champ n’a pas nécessairement d’utilité pour tous les matériels. Il est cependant possible de créer un champ version sur les matériels ou cela aurait du sens, grâce à l’ajout de champs supplémentaires optionnels. Ensuite, le matricule d’un utilisateur nous a semblé inutile car un utilisateur possède une adresse e-mail unique qui permet déjà d’identifier un utilisateur donc même déroulé : nous en avons discuté avec le client et avons décidé de supprimer l’attribut. Enfin, la version de chrome demandée est très vieille donc pour des questions de sécurité et d’ergonomie nous avons demandé au client de mettre ajouter Google Chrome sur tous ses postes pour travailler et tester sur la dernière version.

Ensuite, parlons des choix techniques que nous avons fait. Nous avons choisi React parmi tous les frameworks web car plusieurs membres du groupe avaient de l’expérience dessus et que d’autres voulaient simplement l’apprendre et/ou s’entraîner. En plus de connaître React, certains d’entre nous connaissions aussi la bibliothèque Material UI que nous avons donc utilisé pour le projet. Elle nous a permis d’obtenir un résultat visuel plaisant de façon relativement simple.

Pour le back-end, nous avons choisi Firebase et des composants d’authentification et de base de données. Cela nous a évité de créer une API REST entière, ce qui représente un gain de temps significatif à toutes les étapes du projet.

Problèmes rencontrés :

Le module d’authentification Firebase permet de gérer les utilisateurs et la connexion, en revanche il était compliqué d’accéder facilement aux données des utilisateurs (pour la gestion admin des utilisateurs).

Pour solutionner ce problème, tout utilisateur créé dans l’application via un formulaire est aussi stocké dans la base de données Firestore, en ne conservant que les informations utiles à l’affichage (pas le mot de passe par exemple). Cette dualité de stockage nous permet d’accéder aux données des utilisateurs comme n’importe quel autre document, de façon simplifiée.

Organisation de l’équipe :

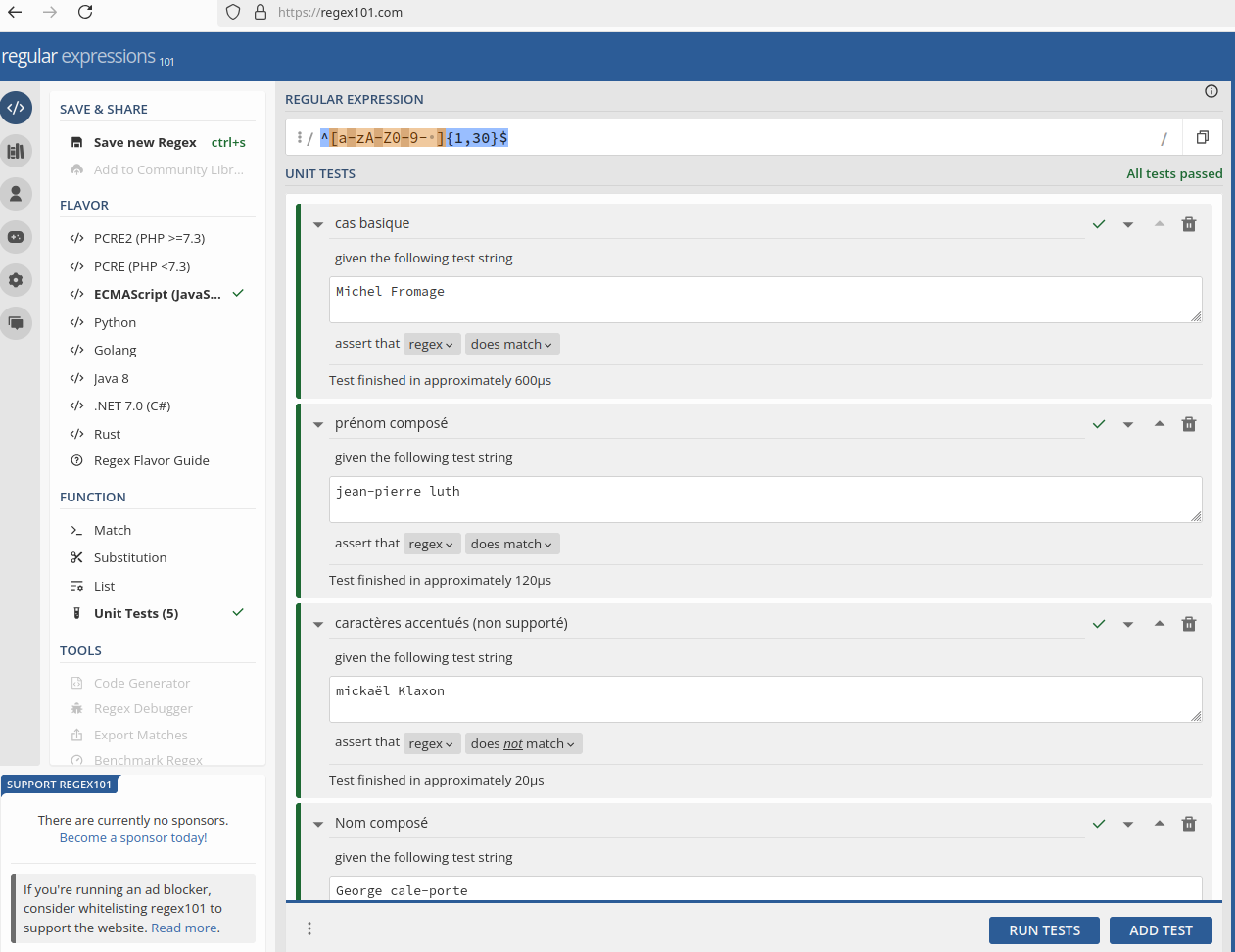
Lors de la phase de spécification, Evan et Alexandre R. ont commencé la rédaction du cahier des spécifications, Mehdi et Alexandre D. ont conçu les maquettes, et Raphaël a réfléchi à la structure de données.

Ensuite, Evan et Alexandre R. se sont occupés de la rédaction de cahiers de tests et ont effectué ces derniers lorsque c’était possible, et Mehdi, Alexandre D. et Raphaël se sont occupés du développement.

Durant tout le projet, même si chacun était assigné à sa tâche, nous collaborions souvent tous ensemble sur certains points et prenions les décisions ensemble, que ça soit des décisions concernant les visuels, la base de données, ou les fonctionnalités. De plus, pour l’organisation en dehors des séances, nous communiquions via un serveur discord, et nous tenions au courant des modifications sur le projet bio un hook qui envoyait des messages à chaque push sur le repository git. Cela nécessitait une rédaction des messages de commit exhaustive et précise.

Tests unitaires :

L’utilisation de firebase pour ce projet en tant que backend nous as donc permis de ne pas avoir de tests à effectuer sur cette partie du système. Nous avons effectué cependant quelques tests unitaires, essentiellement sur les regeX qui nous ont servi à contrôler le format des données sur le frontend. Pour ce faire, nous avons utilisé le site regex101.com, qui permet d’effectuer un ensemble de tests unitaires à définir, pour une regeX donnée.



Cahier de recette :

Tous les tests fonctionnels manuel ont été fait sur Google Chrome 131 sur la version en ligne disponible à l’adresse locamat.ninja .

Lors de la première écriture du cahier de test nous n’arrivions pas à assez détailler les différents tests fonctionnels possible. Nous avions une vingtaine de test compliqué qui, finalement, n’était pas forcément des tests fonctionnels. Dans un second temps, nous avons pu comprendre comment détailler les différents tests et leurs objectifs. Cependant, nous avions écrit des tests redondants qui étaient les tests d’affichages. En effet, nous voulions trop faire les tests d’affichages de chaque objet sur la page qui causait une grande redondance de tests qui peuvent être fait en une fois.

Pour la partie tests automatisés nous avons utilisé l’outil Selenium. Cet outil est facile à prendre en main et nous a permis de découvrir comment automatiser les tests fonctionnels. Nous avons trouvé ça ludique de pouvoir voir s’afficher sur notre écran et voir l’évolution de notre code de test visuellement.

La mise en place du pipeline CI sur github a pu se faire simplement pour la partie build du projet. Nous avons utilisé le template disponible qu’on a pu adapter pour qu’il se lance tous les jours dans la nuit du lundi au vendredi. Cependant, nous avons essayé de mettre en place les tests fait sur Selenium mais nous avons rencontré une erreur dont nous ne connaissons pas l’origine.

Améliorations possibles :

-Nous avons envisagé le fait de vérifier au moment de créer un compte, de vérifier que l’adresse mail n’existe pas déjà (le compte est déjà créé avec cet identifiant), mais étant donné le contexte du projet, nous avons conclu que ce n’était pas primordial, dans le sens où un utilisateur lambda ne peut pas créer de compte, et ne fait la demande à l’administrateur qu’en cas de nécessité, donc s’il n’en possède pas déjà un.

-Nous avons aussi envisagé de pouvoir rendre en avance un matériel par rapport à la fin de réservation, mais ce n’était pas prioritaire, cela pourrait être un axe d’amélioration.