



**UNIVERSITÉ
DE LORRAINE**

Documentation technique

Groupe 2 : Surveys

Membres :

CONRAD Alexandre

DAVIES Liam

CHEVALET Clément

BIR Vincent

Table des matières

Membres :	1
Samedi 05/09/2020	4
Dimanche 06/09/2020	4
Lundi 07/09/2020	4
Samedi 12/09/2020	4
Samedi 31/10/2020	5
Mercredi 04/11/2020	5
Jeudi 05/11/2020	5
Dimanche 08/11/2020	5
Lundi 09/11/2020	5
Jeudi 12/11/2020	5
Vendredi 13/11/2020	6
Dimanche 17/11/2020	6
Dimanche 29/11/2020	6
Jeudi 03/12/2020	6
Mardi 08/12/2020	6
Lundi 14/12/2020	6
Mardi 15/12/2020 et Dimanche 20/12/2020	6
Lundi 21/12/2020	7
Mardi 22/12/2020	7
Lundi 08/01/2021	7
Lundi 11/01/2021	7
Mardi 12/01/2021	7
Mercredi 13/01/2021	7
Jeudi 14/01/2021	8
Vendredi 15/01/2021	8

Samedi 05/09/2020

Nous avons pris la décision d'utiliser SwaggerEditor pour notre projet.

Swagger est une technologie jamais utilisée par les membres du groupe, qui permet de générer une documentation très claire et fonctionnelle mais surtout une structure pour le code.

MongoDB permet de faire une base de données au format JSON, ce que nous souhaitons tester ensemble. Cela simplifiera énormément les requêtes, et permettra de faire le moins de traitement possible avant d'insérer le résultat en base de données.

Dimanche 06/09/2020

Après une grosse analyse du projet, nous en avons retiré les spécifications suivantes du cahier des charges.

Le projet sera découpé en plusieurs routes pour l'API, ce qui permet d'avoir une route par grande fonctionnalité.

/sondages/ => pour le sondage en lui même

/choices/ => pour ajouter / supprimer des choix / dates au sondage

/votes/ => voter / modifier le vote a un sondage

/analytics/ => obtenir les meilleures dates

/comments/ commenter/ modifier/ supprimer un sondage

Un Schéma UML a été réalisé afin de mettre en place notre structure pour le code, structure qui respectera le modèle par couche étudié en cours.

Lundi 07/09/2020

Mise en place de swaggerEditor avec notre découpage de route.

Samedi 12/09/2020

SwaggerEditor nous a généré un très gros nombre de classes non souhaitées, qui entraînent une grosse redondance de code, du code smell ainsi que des vulnérabilités. Nous avons donc nettoyé le code entièrement.

Samedi 31/10/2020

Pour donner suite aux cours sur Git, nous avons décidé d'appliquer ces bonnes pratiques à notre projet :

- Un git initial, création du projet.
- Ajout des différentes dépendances essentielles au projet.
- Ajout d'un gitignore spécifique aux développeurs du projet.
- Ajout de mockito, qui est l'une des technologies vues en cours. Aucun membre du groupe n'a utilisé cette technologie. Nous allons donc l'utiliser pour nos tests unitaires.

Mercredi 04/11/2020

Lombok est une autre technologie vue en cours que nous n'avions jamais vu auparavant, nous avons donc décidé de l'ajouter au projet. Nous pouvons dorénavant réduire considérablement le code. Nous n'avons plus besoin de faire de « Getter / setter / constructeur / Hashcode / toString / etc ».

Nous avons ensuite débuté les premiers tests de mockito et le mapping de la classe Survey pour se familiariser avec les deux outils. Le mapping qui a été réalisé à l'aide d'Hibernate. Cette technologie a aussi été choisie par rapport au cours suivis.

Jeudi 05/11/2020

Création de la base de données H2 pour approfondir le cours dessus.

Nous avons ensuite créé un importSQL pour les données, et un pipeline d'intégration pour pouvoir build le projet.

Dimanche 08/11/2020

La base de données est entièrement installée, ce qui nous a permis de configurer le mapping et les différentes informations pour la rendre fonctionnelle, avec une ouverture de session au début du projet pour en avoir un singleton.

Lundi 09/11/2020

Le pipeline d'intégration a été configurée pour pouvoir accéder à notre projet depuis SonarQube. De ce fait, nous avons dorénavant un début de notations de notre code par SonarQube.

Jeudi 12/11/2020

Nous avons personnalisé le ReadMe pour respecter au mieux la convention des importants projets GIT. Cela nous a permis d'avoir toutes la documentations, informations sur la base de données, pré

requis et informations sur l'installations. Le ReadMe, dispose de toutes les informations importantes au fonctionnement du projet.

Vendredi 13/11/2020

Nous avons configuré et implémenté Hibernate pour le mapping. C'est une fois de plus l'une des technologies vues en cours que nous avons choisi d'utiliser pour approfondir nos connaissances, car personne de notre groupe ne l'avait jamais utilisé auparavant.

Dimanche 17/11/2020

Nous avons choisi JUnit et Mockito pour créer un premier service « Survey » disposant de tests unitaires.

Dimanche 29/11/2020

Création de plus de 70% des services de « survey », accompagnés de leurs tests unitaires.

Jeudi 03/12/2020

Finalisation des services de survey ainsi que de leurs tests. Le service dispose de toutes les différentes requêtes SQL pour la rubrique des survey.

Mardi 08/12/2020

ImportSQL avait quelques problèmes que nous avons revus et nous avons corrigé certaines erreurs de mise en forme. L'importSQL doit être sous la forme d'une ligne, une mise en forme n'est pas acceptée par Hibernate.

Lundi 14/12/2020

Correctif sur le dialecte de discussion de la base de données entre H2 et Hibernate, et du « on cascade » pour supprimer les tables découlant d'une table.

Mardi 15/12/2020 et Dimanche 20/12/2020

Nous avons créé entièrement la partie sur les commentaires, accompagnés des tests respectifs.

Lundi 21/12/2020

Nous avons revu le code dans le but d'améliorer drastiquement la note de sonar et de réduire le code smell.

Mardi 22/12/2020

La partie Analytics a été développée et mise en place.

Lundi 08/01/2021

Nous avons généré la documentation Postman en plus de Swagger afin d'explorer plusieurs outils pour la documentation.

Lundi 11/01/2021

Nous avons entrepris une importante revue de code :

- Organisation d'une structure pour les tests unitaires.
- Utilisation de Mockito pour réaliser les tests les plus compliqués.
- Le coverage restait à 0% avec 33 tests implémentés. Dans nos recherches, nous sommes tombés sur Jacoco qui permet de générer un rapport lisible par SonarQube.
- Ceci fait, la note Sonar a grimpé de 0% à 35.3%.

Mardi 12/01/2021

Le changement de la version de Java a entraîné une correction de notre pipeline, et nous avons dû rajouter une exclusion par sonarQube par rapport à son model à l'aide de Lombok.conf.

Nous avons aussi retiré de sonarQube à l'aide de l'exclusion certaine classe qui n'avait pas besoin d'être testé. Par exemple certains fichiers de configuration.

Une augmentation du coverage de 35% à 62% a été réalisé afin d'atteindre l'objectif de la quality gate de sonarQube.

De plus, une architecture des exceptions a été mise en place, avec certaines première exceptions pour les services de Survey.

Mercredi 13/01/2021

Toutes les exceptions ont été réalisées dans le projet. De plus une importante revue de code a été réalisée afin de passer de 64 codes smells à 42 suivis du passage à 84% de coverage ce qui nous a permis de passer en quality gate passed !

Jeudi 14/01/2021

Améliorations du code de 45 codes smell à 7. De plus certains correctifs sur les bades ou le visuel du ReadMe ont été mise en place.

Vendredi 15/01/2021

Nous avons eu un problème de sonarQube important. Notre projet été fusionné avec un autre groupe qui ne respectait pas certains quality gate. Nous avons donc décidé de faire une importante revue de code dans les configurations afin de ne plus être en conflit avec un autre groupe.

Nous avons donc un nouveau dépôt sonarQubes qui ne rentre plus en conflit. De plus nous avons modifié tous nos chemins des badges vers le nouveau sonarQube.

Toutes les documentations ont été réalisées et importées dans le ReadMe.