CodingWeek 2021

TELECOM Nancy - 13/12 au 17/12

TelecomNancy ProfRDV

> TelecomNancy ProfRDV est une application qui permet de prendre des rendez-vous avec les enseignants de l'école. Elle permet également aux enseignants de gérer leur disponibilité.

Sujet

L'objectif de ce projet est de réaliser une application permettant de gérer des rendez-vous entre un ou des élèves et un enseignant. Il existe de nombreuses plateformes de prise de rendez-vous en ligne (à destination des médecins, des garagistes, des coiffeurs, etc.). Si vous n'êtes pas familier avec ce genre de plateforme nous vous invitons à étudier le fonctionnement de quelques applications existantes.
Il est supposé que plusieurs enseignants et élèves utilisent la même instance de votre application (il faudra donc distinguer les utilisateurs d'une manière ou d'une autre).

Les principales fonctionnalités attendues sont :

- la création d'une demande de rendez-vous à un créneau donné avec un enseignant donné. On supposera que tous les rendez-vous ont une durée fixe (20 minutes).
- l'ajout de différentes informations à un rendez-vous en plus de son intitulé (une description, un lieu, etc.)
- la validation d'une demande de rendez-vous.
- l'annulation ou la modification d'un rendez-vous.
- la création/suppression/modification d'enseignants.
- la visualisation des demandes de rendez-vous par un enseignant / par un élève.
- la visualisation des rendez-vous par un enseignant / par un élève.
- la gestion de la persistance des données.
- la définition par un enseignant de ses disponibilités/indisponibilités habituelles.
- l'ajout par l'enseignant de disponibilités/indisponibilités exceptionnelles (par exemple, un créneau d'un jour particulier où il n'est finalement pas disponible).

Il est possible d'envisager un certain nombre d'extensions aux fonctionnalités basiques présentées ci-dessus.

- ajouter des catégories sur les rendez-vous.
- pouvoir visualiser les rendez-vous sous forme d'un planning.
- pouvoir visualiser un planning "anonymisé" d'un enseignant (en masquant les informations détaillées).
- pouvoir réaliser des rendez-vous impliquant plusieurs enseignants.
- exporter les rendez-vous au format ICAL pour pouvoir les importer dans son calendrier personnel (par exemple, celui de Google Calendar).
- gérer des rendez-vous dont la durée est variable (en fonction du type de rendez-vous ou d'une durée que l'utilisateur précise au moment de la demande).
- ajouter un système de notification (par email ou par messagerie instantanée).
- réaliser une version client/serveur où plusieurs instances de votre application JavaFX partagent/échangent des données à travers un serveur.

Organisation

Constitution des groupes

Les groupes sont constitués de 4 élèves qui ne sont pas issus obligatoirement du même groupe de TD. Il vous est donc demandé pour le vendredi 3 décembre 2021 au plus tard de fournir les informations concernant votre groupe en remplissant le formulaire dédié à l'adresse https://forms.gle/iRWFeefdZLceJZFC9.

Il vous sera demandé de fournir :

- le nom de votre groupe,
- sa composition (pour chaque membre du groupe : son nom, prénom, son adresse email (terminant par `@telecomnancy.eu`).

Ces informations nous permettront de créer un projet sur le serveur GitLab de l'école dédié à votre groupe.

Présence et assiduité

La semaine est entièrement dédiée au module pour vous permettre de vous concentrer sur ce projet.
Cette année, cette semaine bloquée, aucune présence à l'école ne sera obligatoire. Vous êtes donc libres de gérer votre planning vous-même en coordination avec les autres membres de votre groupe. Toutefois, nous vous invitons fortement à vous regrouper par groupe pour pouvoir travailler de manière efficace et conviviale.

Encadrement

Chaque groupe se verra assigner un enseignant référent. Celle-ci ou celui-ci veillera à la bonne avancée de votre projet au cours de la semaine. Nous vous invitons à prendre contact avec lui pour définir de quelle manière il souhaitera faire le point avec votre groupe.
Cette année, le nombre de groupes étant élevé, il est important de documenter votre projet et de déposer régulièrement ces documents dans votre dépôt git pour que votre enseignant référant puisse les consulter et vous faire des commentaires si nécessaire.
Au cours des journées, les enseignants seront disponibles pour répondre à vos questions. Ceux-ci seront joignables à travers la plateforme Microsoft Teams et à certains moment en présentiel à l'école. Vous trouverez un canal dédié à votre groupe et différents canaux partagés dans l'équipe "CodingWeek 2021".

Travail et collaboration

Suite à l'enregistrement de votre groupe, un dépôt privé Git vous sera créé où vous déposerez le code source et les différentes informations nécessaires à la compilation et à l'exécution de votre application (Java).
Il est nécessaire de bien organiser le contenu de ce dépôt et de "committer" régulièrement afin que l'ensemble des membres du groupe aient accès aux données les plus récentes. Il sera tenu compte des contributions de chacun des membres du groupe. Si un des membres n'a pas su "committer" ses contributions, sa participation sera considérée comme nulle entrainant l'invalidation du module.

Vous devrez :

- documenter la planification et l'avancement du projet ;
- vous assurer des livraisons (releases) quotidiennes et de la bonne utilisation de Git ;
- mettre en œuvre des tests et des procédures assurant la qualité du logiciel ;
- documenter votre architecture et votre conception pour les rendre compréhensibles ;
- documenter l'installation et l'exécution de votre application.

Ces différents points seront pris en compte pour un tiers de la note.

Travail préparatoire

Vous disposez de quelques jours pour vous mettre à niveau sur les technologies que vous devrez utiliser :

- renforcement de vos connaissances sur l'utilisation de JavaFX et le fonctionnement interne de Git
- mise en place de l'environnement de travail (outils de développement, gestion de versions et de bases de données, etc.) ;
- étude des technologies liées qui pourraient être utilisées.

Quand vous utilisez Git faites attention le cas échéant à ne pas publier vos clés d'accès aux APIs que vous utilisez [a]. Pour cela, il convient de placer ces clés dans un fichier de configuration et de filtrer ce fichier en utilisant le fichier `.gitignore`. Vous devez vous arranger pour que chacun utilise ses propres clés d'accès aux API. Si possible, prévoyez des clés d'accès pour vos démonstrations, sinon documentez la procédure pour obtenir et fournir ces clés.

a. Ars Technica. *PSA : Don't upload your important passwords to* GitHub. Janvier 2013

Évaluations et rendus

Le développement se fait de **façon itérative et incrémentale**. Vous devez produire un **logiciel opérationnel chaque jour**. Nous vous recommandons d'utiliser un outil tel que Gradle pour automatiser la compilation, l'exécution des tests et la construction de votre application. Vous y ajoutez des fonctionnalités en fonction d'une feuille de route (roadmap) que vous aurez définie à l'avance et que vous ferez évoluer en fonction de votre avancée. Il n'est pas question d'avoir plusieurs morceaux de programmes fonctionnant indépendamment les uns des autres. Le programme doit être intégré chaque jour et vous devez à tout moment pouvoir faire une démonstration de la dernière livraison. Le document `README` du dépôt doit indiquer comment exécuter ce programme.

Rendus intermédiaires

Chaque groupe devra effectuer un rendu journalier sous la forme de commits Git (cela ne signifie pas qu'il faut se limiter à un commit par jour dans le dépôt, au contraire). Ce rendu devra être étiqueté (au moyen de la commande `git tag` dans la branche master) selon la convention de nommage suivante : `RELEASE_DAY_X` où X indique le jour de la semaine (lundi : 1, mardi : 2, etc.). Il s'agira d'une livraison, donc d'un système opérationnel que l'on peut déployer (sous forme d'une archive `.jar` par exemple). Par défaut, la commande `git push` ne partage pas les tags, il faut explicitement partager ces tags (en utilisant l'option `--tags`).

La contribution de chacun des participants d'un groupe pourra être mesurée à son activité sur GitLab, i.e. le nombre de commit qu'il aura pu faire et leur taille. Il ne sera pas admis que certains d'entre vous n'aient pas commité de code à leur nom. Il est donc important de bien configurer votre client Git afin que vos commits vous soient attribués [a].

a. Vérifiez les variables `user.name` et `user.email` au moyen de la commande `git-config --list`.

Rendu final

Le rendu final sera étiqueté (toujours la branche master) selon la convention de nommage suivante : `RELEASE_FINAL`.

Il devra comporter a minima :

- les documents d'analyse et de conception que vous aurez réalisés ;
- les documents de gestion de projet ;
- le code source de l'application client lourd (Java), les instructions textuelles (texte brut ou fichier PDF) indiquant comment compiler, configurer et exécuter cette application (précisant les dépendances externes et comment avoir accès à ces dépendances) ;
- une archive `.jar` de l'application que l'on doit pouvoir exécuter facilement.

Les derniers commit auront lieu avant 16h le vendredi 17 décembre 2021.

Il ne sera pas possible de faire de démonstration le dernier jour. Vous nous rendrez une démo de 10 mn de votre application sous la forme d'un screencast/vidéo présentant ses fonctionnalités. Vous pouvez utiliser un outil de capture d'écran tel que OBS Studio https://obsproject.com/ ou tout autre outil. Le plus simple est de déposer votre vidéo sur une plateforme en ligne telle que YouTube ou autres et de publier ensuite le lien dans votre fichier `README` de votre dépôt Git (ce n'est pas un problème si le commit publiant ce lien est réalisé après l'heure de rendu limite de votre projet).

Cette vidéo devra être rendue au plus tard samedi 18 décembre 2021 midi.

Critères d'évaluation

- résultats concrets présentés à la fin de la semaine (vidéo) ;
- respect des méthodes (par exemple, diagrammes de conception, maquette d'interface graphique, respect du MVC, etc.) ;
- organisation du travail (par exemple, documents de gestion de projet) ;
- mise en œuvre de tests démontrant le fonctionnement du logiciel ;
- qualité du développement (principe de conception mis en œuvre, architecture adaptée).

Communication avec les encadrants

Afin de poser vos questions et de discuter durant la semaine, nous avons crée sur la plateforme Microsoft Teams une canal Général. N'hésitez pas à y poser des questions sur le sujet ou sur des points techniques. Nous avons également crée un canal par groupe de projet si vous voulez vous y réunir.

https://teams.microsoft.com/l/team/19%3aXTzaE0jFsJIJ5khSo72ez7rJt9YvCKCWcTUiLtQonxE1%40thread.tacv2/conversations?groupId=8e720f42-4513-49ab-b222-161502937669&tenantId=158716cf-46b9-48ca-8c49-c7bb67e575f3

Afin de centraliser les différentes réponses, nous avons mis également en place un document partagé. N'hésitez pas à consulter/éditer ce document régulièrement au cours de la semaine.

https://docs.google.com/document/d/1chFSlcE0ScfGcLo-ypxyluLmi2Lw1z0d8IlfKxMrrXI/edit

Quelques pointeurs

Vous trouverez dans cette section divers pointeurs sur des librairies que vous pouvez utiliser. Vous pouvez bien entendu utiliser d'autres librairies que celles présentées.

HttpClient. Depuis Java 11, une API HTTPClient est intégrée à l'API Java pour réaliser des requêtes HTTP et permettre à vos programmes de converser avec des serveurs Web. Un tutoriel d'introduction de cette API est disponible en ligne (https://openjdk.java.net/groups/net/httpclient/intro.html). Vous pouvez également utiliser la librairie Apache HTTPComponents (https://hc.apache.org/).

Base de données relationnelles embarquée. Si vous le souhaitez, vous pouvez utiliser une base de données SQL embarquée tels que Apache Derby (https://db.apache.org/derby/), H2 (https://www.h2database.com/), HSQLDB (http://hsqldb.org/) ou SQLite (https://github.com/xerial/sqlite-jdbc) pour gérer la persistance des données de votre application.

Elles sont facilement utilisables (https://dzone.com/articles/3-java-embedded-databases) et supportent toutes la spécification JDBC (Java Database Connectivity). Une introduction à cette technologie est disponible à l'adresse : https://www.jmdoudoux.fr/java/dej/chap-jdbc.htm.

Gradle / JavaFX. Le projet disponible à l'adresse (https://gitlab.telecomnancy.univ-lorraine.fr/Gerald.Oster/boilerplate-gradle-jdk15) peut vous servir de point de départ pour configurer un projet Gradle supportant Java/JavaFX. Ce projet est adapté à la version 17 mais fonctionne également avec les versions antérieures.