uOttawa

SEG2505 - Introduction au génie logiciel - automne 2023

Projet Android : Service Novigrad (20%)

Cette application implémentera des fonctionnalités de base pour les services offerts par une province imaginaire appelée Novigrad à ses résidents. Pensez aux services offerts par Service Novigrad comme étant similaires à ceux offerts par Service Ontario ou Services Québec qui permettent aux citoyens d'obtenir un permis de conduire, une pièce d'identité avec photo ou une carte Santé.

INSTRUCTIONS

1. Le projet sera réalisé dans la même équipe que vous avez rejoint pour vos laboratoires.
2. Vous allez soumettre chaque livrable sous forme de version (release) sur GitHub.
3. L'équipe ne doit présenter qu'une seule version de l'application. Par exemple, un étudiant ayant un écran avec la fonctionnalité de recherche et un autre ayant un autre écran différent (fonctionnant sur un téléphone différent) avec la fonctionnalité du fournisseur de services, **NE SERA PAS accepté**. L'équipe doit produire une seule application avec toutes les fonctionnalités requises.

Le but de ce projet est d'élargir le travail théorique, permettant aux étudiants d'acquérir une expérience pratique de la mise en œuvre des concepts appris en classe. Ce projet est également conçu pour permettre aux étudiants d'apprendre à travailler avec leurs collègues et à développer des applications mobiles.

Le résultat principal du projet est le développement d'une application Service Novigrad pour les appareils Android. Les étudiants doivent implémenter toutes les composantes du projet à partir de leur spécification de design, modèles UML, ressources graphiques et code source. Les étudiants sont encouragés à utiliser l'ensemble d'outils disponibles dans Android Studio, mais doivent éviter de copier des blocs entiers de code à partir de l'Internet pour implémenter les fonctionnalités. Si un groupe souhaite utiliser un outil / une API non standard, il doit demander l'autorisation avant de le faire.

L'application doit supporter trois types d'utilisateurs : administrateur, employé d'une succursale de Service Novigrad, et client. L'administrateur gère tous les services possibles qui peuvent être proposés aux clients par les différentes succursales de Service Novigrad. L'employé de la succursale crée un profil pour la succursale, sélectionne les services offerts par cette succursale et spécifie les heures de travail. Les clients doivent pouvoir rechercher une succursale de Service Novigrad par adresse / type de service fourni / heures de travail.

Les fonctionnalités qui devraient être disponibles pour chaque type d'utilisateur sont indiquées ci-dessous. Notez que ce sont les fonctionnalités minimales requises et que vous êtes libres d'ajouter d'autres fonctionnalités qui, selon vous, pourraient enrichir votre application.

L'administrateur peut :

1. Se connecter à un compte administrateur - le développeur doit pré-créer un tel compte (par exemple, nom d'utilisateur : *admin*, mot de passe: *123admin456*)
2. Créer, modifier, ou supprimer des services (**au moins 3** comme décrit ci-dessous, mais plus il y en a, mieux c'est) qui pourraient être offerts par les succursales Service Novigrad, et spécifier les informations et documents requis qui doivent être fournis par un client qui demande un service :
    a. Permis de conduire :
        i. Formulaire : Incluant le prénom, nom, date de naissance, adresse, et type de permis du client (G1, G2, G)
        ii. Documents : Preuve de domicile (une image d'un relevé bancaire ou d'une facture d'électricité indiquant l'adresse)
    b. Carte santé :
        i. Formulaire : Incluant le prénom, nom, date de naissance, et adresse du client
        ii. Documents : Preuve de domicile (une image d'un relevé bancaire ou d'une facture d'électricité indiquant l'adresse) - Preuve de statut (une image d'une carte de résidence permanente au Canada ou d'un passeport Canadien)
    c. Pièce d'identité avec photo :
        i. Formulaire : Incluant le prénom, nom, date de naissance, et adresse du client
        ii. Documents : Preuve de domicile (une image d'un relevé bancaire ou d'une facture d'électricité indiquant l'adresse) - Une photo du client
3. Supprimer les comptes des succursales et des clients.

L'employé de la succursale Service Novigrad peut :

1. Créer un compte pour la succursale et se connecter à ce compte
2. Sélectionner les services fournis
3. Entrer les heures de travail de la succursale
4. Visualiser les demandes de service soumises et approuver ou rejeter chaque demande

Le client peut :

1. Créer un compte et se connecter à ce compte
2. Rechercher une succursale de Service Novigrad par adresse / type de service fourni / heures de travail

- Vous devez également afficher la note (rating) de chaque service trouvé dans le processus de recherche

3. Sélectionner un service qu'ils souhaitent acheter
4. Remplir les informations et les documents requis, puis soumettre la demande de service.
5. Évaluer leur expérience avec la succursale de Service Novigrad

Remarque : Ce cours ne se concentre pas sur la conception d'interfaces usagers ; par conséquent, nous ne nous concentrons pas sur les aspects d'utilisabilité. Cependant, les étudiants sont encouragés à « Embellir » leurs projets, s'ils sont à l'aise avec la conception de l'interface usager. Tenez compte des consignes de design Android lors de la conception de votre application. Ce sujet sera couvert dans un tutoriel et des informations détaillées sont disponibles sur : https://developer.android.com/design/index.html

---

LIVRABLES

Le projet est divisé en 4 livrables incrémentales. Les étudiants doivent soumettre chaque livrable avant la date limite affichée en ligne à l'aide de Brightspace.

Livrable	Date d'échéance
1 - Référentiel (Repository) Github et comptes des utilisateurs (3%)	29 octobre
2 - Fonctionnalité de l'administrateur (3%)	12 novembre
3 - Fonctionnalité de l'employé de la succursale de Service Novigrad (3%)	26 novembre
4 - Fonctionnalité du client et de l'application (9%)	3 décembre
5 - Démonstration (2%)	Dernière semaine de cours

Le projet doit être réalisé tout au long du semestre et les étudiants sont fortement encouragés à maintenir un journal des activités de leur projet, car l'attribution des tâches et le déroulement du projet font partie du rapport que vous allez soumettre à la fin du projet. Nous suggérons aux étudiants de prendre note de l'attribution des tâches, de la complexité des tâches attribuées, et les dates de complétion.

Votre application doit être écrite en Java et construite à l'aide d'Android Studio (vous pouvez utiliser un autre IDE, mais les TAs peuvent vous aider avec Android Studio seulement). Vous devez compiler votre projet avec la version la plus récente possible du SDK autorisée par les méthodes d'API que vous utilisez. À la fin du semestre, vous devez implémenter et soumettre une application qui fonctionne basée sur les spécifications données. Firebase ou SQLite peuvent être utilisés pour stocker et récupérer les données d'application.

HONNÊTETÉ ACADÉMIQUE

Tout le travail que vous faites dans ce cours doit être le vôtre, sauf si la collaboration est explicitement autorisée. Voir ou copier le travail d'un autre individu (même s'il est sauvé dans un répertoire public) ou soulever du matériel d'un livre, magazine, site Web ou autre source - même en partie - et le présenter comme le vôtre constitue une malhonnêteté académique. Le même s'applique si vous montrez ou donnez votre travail, même en partie, à un autre étudiant.

LIVRABLE 1

Vous devez rejoindre notre GitHub Classroom, créer votre équipe et accepter ce projet. Veuillez voir le document « Étapes pour créer votre équipe sur GitHub » pour des instructions détaillées.

Dans ce livrable, vous devez implémenter la composante de gestion des comptes utilisateurs. Autrement dit, l'application doit permettre aux utilisateurs de créer des comptes utilisateur.

Pour simplifier le développement, il y aura un seul compte administrateur pré-créé (par exemple nom d'utilisateur : *admin*, mot de passe : *123admin456*), mais il devrait être possible de créer autant de comptesd'employés et de clients de la succursale de Service Novigrad que souhaité.

Une fois que l'utilisateur s'est connecté, il devrait voir un deuxième écran avec le message suivant : «Bienvenue « *prénom* »! Vous êtes connecté en tant que *« rôle »*. Aucune fonctionnalité supplémentaire ne doit être implémentée dans cette étape.

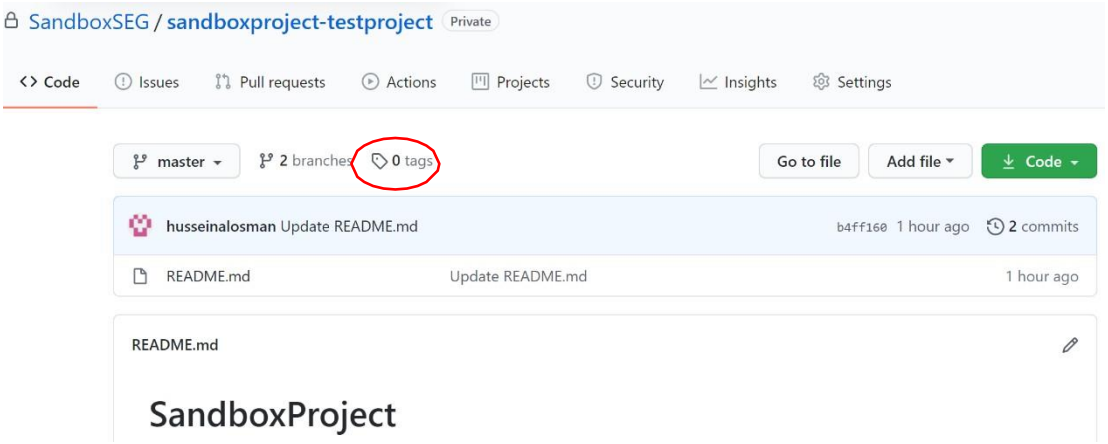
Vous pouvez utiliser Firebase ou SQLite pour supporter la base de données. Si vous n'utilisez pas de base de données dans cette étape, vous pouvez simplement stocker les informations en mémoire (elles seraient perdues lorsque vous fermez l'application) ou dans un fichier.

Dans votre référentiel (repository) GitHub, incluez un document PDF contenant un diagramme de classes UML de votre modèle de domaine. Cela n'inclut que les classes UML liées au livrable 1. De plus, dans votre référentiel (repository), fournissez un fichier README contenant les informations d'identification nécessaires pour vous connecter en tant qu'administrateur (si un compte d'administrateur prédéfini a été créé).

Comment soumettre :

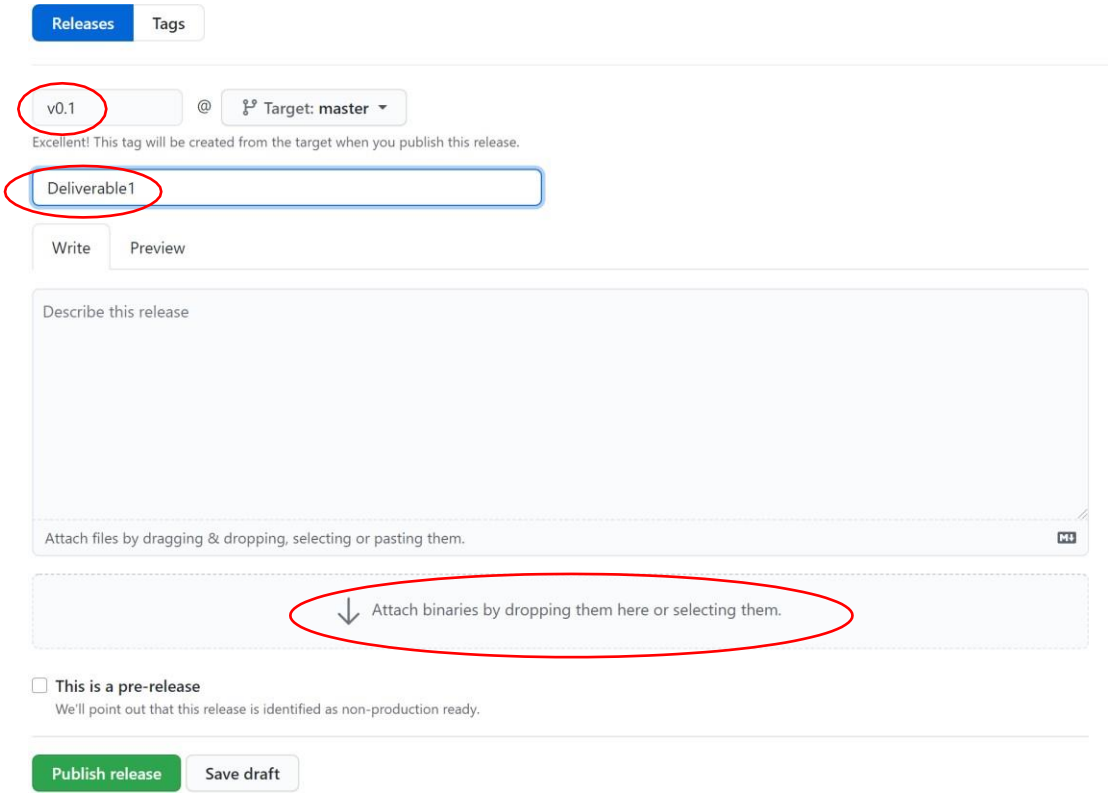
Pour soumettre votre code, vous allez créer une version à partir de votre référentiel (repository) comme suit :

1. Dans votre référentiel (repository), cliquez sur « tags » (voir figure ci-dessous)



2. Cliquez sur le bouton « Create new release» et remplissez le formulaire comme suit (voir la figure ci-dessous)
    a. Pour la « version tag » (tag version), entrez v0.1
    b. Pour « Titre de la version » (release title), entrez Livrable1

c. Dans la section du formulaire où vous pouvez joindre des binaires, téléchargez l'APK de votre application.
d. L'APK se trouve dans <yourAndroidProject>/app/build/outputs/apk/app-debug.apk. Nommez l'APK après le nom de votre équipe "group1_debug_apk".



Barème pour Livrable 1:

Fonctionnalité ou Tâche	**% Poids (sur 100)**
Github: L'équipe créée dans Github Classroom contient tous les membres du groupe	10
Github: Chaque membre du groupe a effectué au moins UN « commit » sur le référentiel (repository).	20
Diagramme de classes UML de votre modèle de domaine (-2 pour chaque classe manquante) (-2 pour une généralisation incorrecte) (-0,5 pour chaque multiplicité incorrecte) (-0,5 pour chaque attribut manquant)	30
APK soumis	5
Capacité de créer un compte pour employer de la succursale de Service Novigrad	10
Capacité de créer un compte client	10

Capacité de voir « L'écran de bienvenue» après une authentification réussie. Capacité de voir le rôle de l'utilisateur Capacité de voir le nom ou le nom d'utilisateur associé au compte	5
Tous les champs sont validés dans tous les écrans (par exemple, vous ne pouvez pas utiliser une adresse courriel ou un nom, etc. invalides) (-1 pour chaque champ dans lequel les données de l'utilisateur ne sont pas validés)	10
OPTIONNEL - Le groupe utilise une base de données (par exemple Firebase ou SQLITE, ou une autre technologie similaire)	+5 (bonis)

LIVRABLE 2

Dans ce livrable, vous devez implémenter les fonctionnalités liées à l'administration. Autrement dit, l'application doit permettre à l'administrateur d'ajouter, de modifier et de supprimer des services qui sont offerts par les succursales de Service Novigrad à l'aide de l'application. Pour chaque service à créer, une liste des informations client et des documents requis doit être spécifiée.

En outre, le compte administrateur doit pouvoir supprimer les comptes des succursales de Service Novigrad (le compte de la succursale est le même que le compte d'employé) et les comptes clients.

Vous devez inclure au moins 5 cas de tests unitaires (tests locaux simples). Il n'est pas nécessaire d'inclure l'instrumentation ou les tests Espresso (UI).

Dans votre référentiel (repository) Github, incluez un document PDF contenant un diagramme de classes UML de votre modèle de domaine. Cela n'inclut que les classes UML liées aux livrables 1 et 2. De plus, dans votre référentiel (repository), fournissez un fichier README contenant les informations d'identification nécessaires pour vous connecter en tant qu'administrateur (si un compte d'administrateur prédéfini a été créé).

Comment soumettre :

Pour soumettre votre code, créez une version v0.2 dans votre référentiel (repository). Assurez-vous que le fichier APK est ajouté à votre version (release).

Remarques :

- Les catégories et sous-catégories de services sont facultatives. Cela peut cependant vous aider à organiser vos écrans (imaginez que vous affichez 50 services dans un seul écran).
- Assurez-vous que votre APK est correctement généré.
    - Allez dans Build -> Build Bundle (s) / APK (s)> Build APK (s)
    - Android Studio enregistre les APK que vous créez dans projectname/modulename/build/outputs/apk/
    - Utilisez-le _Debug pour vos soumissions.
    - Pour plus d'informations: https://developer.android.com/studio/run/

Barème du livrable 2:

Fonctionnalité ou Tâche	**% Poids (sur 100)**
Diagramme de classes UML mis à jour de votre modèle de domaine (-2 pour chaque classe manquante) (-2 pour une généralisation incorrecte) (-0,5 pour chaque multiplicité incorrecte) (-0,5 pour chaque attribut manquant)	10
APK soumis et TOUTES les fonctionnalités sont fonctionnelles	5
5 Cas de test unitaires (tests locaux simples). Pas besoin d'inclure l'instrumentation ou les tests Espresso (UI)	30
Au moins 3 services peuvent être proposés par les succursales de Service Novigrad utilisant l'application implémentée. Un service a un nom et une liste des informations et documents requis	15
Capacité de supprimer des services qui ne sont plus proposés	15
Capacité de modifier des services	15
Tous les champs sont validés. Par exemple, vous ne devriez pas pouvoir créer un service sans nom. Cela doit être mis en œuvre avec des messages d'erreur valides. (-1 pour chaque champ dans lequel les données de l'utilisateur ne sont pas validées)	10
OPTIONNEL - Intégration avec CircleCI pour voir les builds automatisés et les tests automatisés.	+5 (bonis)

LIVRABLE 3

Dans ce livrable, vous devez implémenter la fonctionnalité relative aux employés de la succursale de Service Novigrad. Autrement dit, l'application doit permettre aux employés de la succursale de compléter le profil de la succursale et de l'associer à un ensemble de services (qui ont été créés par l'administrateur). L'employé de la succursale doit également être en mesure de définir les heures de travail de sa succursale.

Vous devez inclure au moins **2 cas de test unitaires supplémentaires** (tests locaux simples). Il n'est pas nécessaire d'inclure l'instrumentation ou les tests Espresso (UI).

Dans votre référentiel (repository) Github, incluez un document PDF contenant un diagramme de classes UML de votre modèle de domaine. Cela n'inclut que les classes UML liées aux livrables 1, 2 et 3. De plus, dans votre référentiel (repository), fournissez un fichier README contenant les informations d'identification nécessaires pour vous connecter en tant qu'administrateur (si un compte d'administrateur prédéfini a été créé).

Assurez-vous que votre APK est correctement généré.

- Allez dans Build -> Build Bundle (s) / APK (s)> Build APK (s)
- Android Studio enregistre les APK que vous créez dans projectname/modulename/build/outputs/apk/
- Utilisez-le _Debug pour vos soumissions.
- Pour plus d'informations : https://developer.android.com/studio/run/

Comment soumettre :

Pour soumettre votre code, créez une version v0.3 dans votre référentiel (repository). Assurez-vous que le fichier APK est ajouté à votre version (release).

Barème du livrable 3:

Fonctionnalité ou Tâche	% poids(sur 100)
Diagramme de classes UML mis à jour de votre modèle de domaine (-2 pour chaque classe manquante) (-2 pour une généralisation incorrecte) (-0,5 pour chaque multiplicité incorrecte) (-0,5 pour chaque attribut manquant)	10
APK soumis et TOUTES les fonctionnalités sont fonctionnelles (voir les notes ci-dessous)	5
2 Cas de test unitaires supplémentaires (tests locaux simples). Pas besoin d'inclure l'instrumentation ou les tests Espresso (UI)	10
Capacité de créer un compte pour la succursale et de s'y connecter	15
Capacité d'ajouter des services au compte de la succursale (parmi la liste des services ajoutés par l'administrateur, l'employé de la succursale peut en sélectionner un ou plusieurs services.)	10
Capacité de supprimer des services du compte de la succursale lorsqu'ils ne sont plus offerts.	10
Capacité de spécifier les heures de travail de la succursale. Vous êtes libre d'implémenter cette fonctionnalité comme vous le souhaitez. Autrement dit, avec un calendrier ou en sélectionnant des plages horaires prédéfinies, par semaine, par jour, etc. La convivialité est la clé de cette fonctionnalité !	20
Capacité de visualiser et d'approuver ou rejeter les demandes de service soumises à la succursale	10
Tous les champs sont validés. Par exemple, vous ne devriez pas pouvoir entrer un numéro de téléphone ou une adresse invalide pour la succursale. Cela doit être mis en œuvre avec des messages d'erreur valides. (-1 pour chaque champ dans lequel les données de l'utilisateur ne sont pas validées)	10
OPTIONNEL - Capacité de modifier les heures de travail	+5(bonis)

LIVRABLE 4

Dans ce livrable, vous devez implémenter la fonctionnalité relative au client. Autrement dit, l'application doit permettre aux clients de rechercher une succursale de Service Novigrad par adresse / heures de travail / type de services fournis. L'application doit afficher la liste des succursales de Service Novigrad disponibles, en fonction de la recherche effectuée par l'utilisateur, et permettre à l'utilisateur de soumettre une demande.

Vous devez inclure au moins **5 cas de test unitaires supplémentaires** (tests locaux simples). Il n'est pas nécessaire d'inclure l'instrumentation ou les tests Espresso (UI).

Dans votre référentiel (repository) Github, incluez un document PDF qui comprend les éléments suivants (**ceci est votre rapport final**):

- Une page titre
- Une brève introduction
- Les diagrammes de classes UML de votre modèle de domaine. Cela inclura toutes les classes que vous avez développées.
- Une section qui traite des leçons apprises
- Un tableau indiquant les rôles et les contributions des membres de l'équipe pour chaque livrable. Vous devez ajouter des explications dans les cas où vous constatez que les contributions n'étaient pas justes.
- Captures d'écran de votre application.

Dans votre référentiel (repository), fournissez un fichier README contenant les informations d'identification nécessaires pour vous connecter en tant qu'administrateur (si un compte d'administrateur prédéfini a été créé).

Assurez-vous que votre APK est correctement généré.

- Allez dans Build -> Build Bundle (s) / APK (s)> Build APK (s)
- Android Studio enregistre les APK que vous créez dans projectname/modulename/build/outputs/apk/
- Utilisez le _Debug pour vos soumissions.
- Pour plus d'informations: https://developer.android.com/studio/run/

Comment soumettre :

Pour soumettre votre code, créez une version v0.4 dans votre référentiel (repository). Assurez-vous que le fichier APK est ajouté à votre version (release).

Barème du livrable 4:

Fonctionnalité ou Tâche	% Poids(sur 100)
Diagramme de classes UML mis à jour de votre modèle de domaine (-2 pour chaque classe manquante) (-2 pour une généralisation incorrecte) (-0,5 pour chaque multiplicité incorrecte) (-0,5 pour chaque attribut manquant)	10
APK soumis et TOUTES les fonctionnalités sont fonctionnelles Assurez-vous de tester votre APK. Un APK non valide recevra 0.	5
Rapport final comprenant : - Une page titre (2,5 points) - Brève introduction (2,5 points) - Diagramme de classe UML mis à jour - Tableau avec les rôles dans l'équipe et les contributions des membres de l'équipe pour chaque livrable. (15 points) - Toutes les captures d'écran de votre application. (10 points) - Leçons apprises (5 points)	35
5 Cas de test unitaires supplémentaires (tests locaux simples). Pas besoin d'inclure l'instrumentation ou les tests Espresso (UI). Les cas de test doivent être pertinents par rapport aux caractéristiques du livrable 4	10
Capacité de rechercher une succursale de Service Novigrad par: - adresse - heures de travail - type de services fournis	10
Capacité de remplir les informations et documents requis et soumettre la demande de service	15
Capacité d'évaluer une succursale avec une note entre 1 et 5 (ou un autre système de notation)	5
Tous les champs sont validés. Par exemple, vous ne devriez pas pouvoir sélectionner une date dans le passé lors de la prise de rendez-vous. Cela doit être mis en œuvre avec des messages d'erreur valides. (-1 pour chaque champ dans lequel les données de l'utilisateur ne sont pas validées)	10
OPTIONNEL - Le client reçoit une notification lorsque sa demande de service est approuvée / rejetée	+10 (bonis)