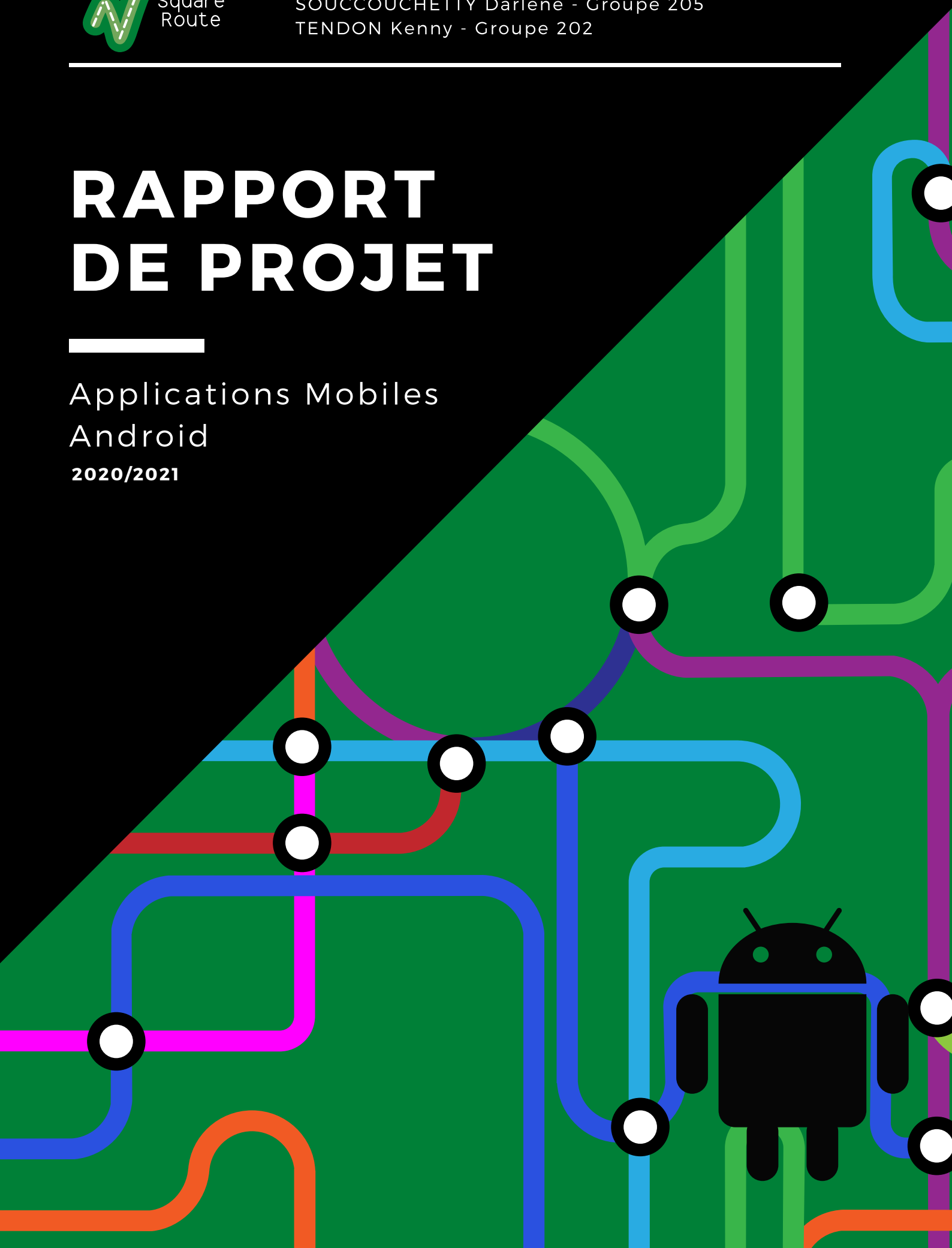


Square
Route

GAMO Christian - Groupe 206
SOUCCOUCHETTY Darlène - Groupe 205
TENDON Kenny - Groupe 202

RAPPORT DE PROJET

Applications Mobiles
Android
2020/2021



PRINCIPE DE L'APPLICATION

Notre application SquareRoute a pour but de répertorier les différentes bibliothèques universitaires et universités de région parisienne.

Nous souhaitons toutefois la restreindre aux transports en commun parisiens de la RATP puisque notre application est avant tout dédiée aux étudiants de la région. Toutefois, il est envisageable que cette application évolue vers une plus grande gamme de transports, nous avons d'ailleurs fait en sorte que cela soit possible.

Grâce à l'API Google Maps et l'API RATP, les étudiants pourront avoir à portée de main et en temps réel : les horaires d'arrivée des RER et Métros du Réseau RATP Parisien, et pourront se rendre depuis leur domicile/leur position (fonctionnalité de géolocalisation) à un établissement/bibliothèque universitaire que nous recensons.

Ils auront également la possibilité de consulter cette liste d'établissements et de bibliothèques aisément grâce à des markers. Nous avons choisi Firebase pour héberger notre base de données, car nous avons également utilisé Firebase authenticator qui est beaucoup plus intuitif et adapté à une application de navigation.

Notre but est de voir plus loin que Google map, car nous avons souhaité notre adapter nos applications aux étudiants, puisque SquareRoute, est faite pour limiter les distractions, inviter les étudiants à renouer avec la culture, en trouvant aisément des lieux pour étudier à savoir des bibliothèques et des établissements scolaires.

ARCHITECTURE LOGICIELLE

Application Android

Layout XML



Drawable



Bibliothèques Code JAVA



API
Android

API'S



Firestore realtime
database +
authenticator



LOGICIELS UTILISÉS



**Adobe
XD**



**Android
Studio**



GitHub



Canva

Adobe XD : Conception et exportation de drawables

Android Studio : Pour coder le projet (back et front).

GitHub : Partage du code grâce à la création d'un repository commun où il est possible de "commit" et "push" du code afin de les "merge" à la master branch du projet.

Canva : Conception du logo/charte graphique et de l'UI en général.

CAHIER DES CHARGES

Au niveau du cahier des charges, la répartition des tâches s'est faite de manière équitable, et chacun d'entre nous a contribué à juste valeur à l'accomplissement du projet.

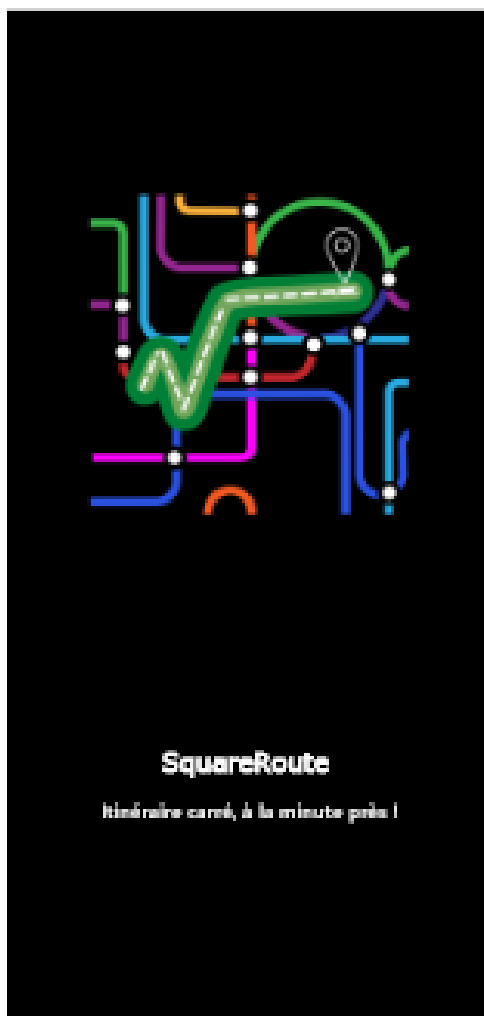
Christian s'est occupé de mettre en place tout ce qui est en rapport avec l'activité `SquareInRealTime` (Requêtes HTTP, Utilisation de la Lib `OKHTTP3`, du lien entre une session Utilisateur avec la base de donnée (sauvegarde de son identifiant, mot de passe et prenom), ainsi que de la fonctionnalité de géolocalisation disponible sur les deux map.

Darlène s'est quant à elle occupée des layouts sur Adobe XD, de la mise en place des map activity de la barre de recherche des maps activity, que du recensement des établissements et bibliothèques en JSON array sur la realtime database, ainsi que de l'installation de GitHub pour mieux gérer le projet à plusieurs.

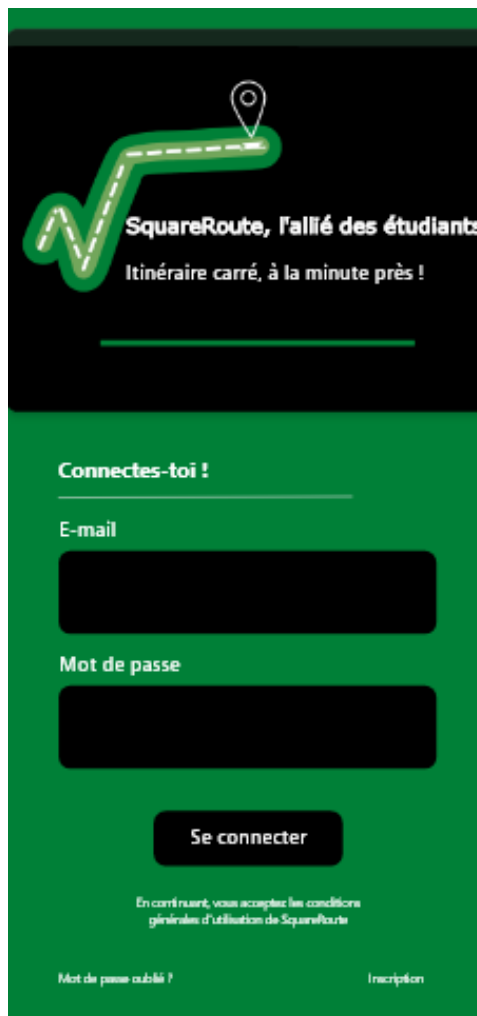
Enfin Kenny s'est chargé de la mise en place des service Google Authenticator et de la realtime Database et des fonctionnalités de base de notre application (authentification, dashboard...)

Cette organisation nous a permis de pouvoir respecter les délais imposés pour ce projet, tout en ne lésinant pas sur les détails tels que la compatibilité des layouts pour différents types d'appareils mobiles fonctionnant sous Android en mode portrait.

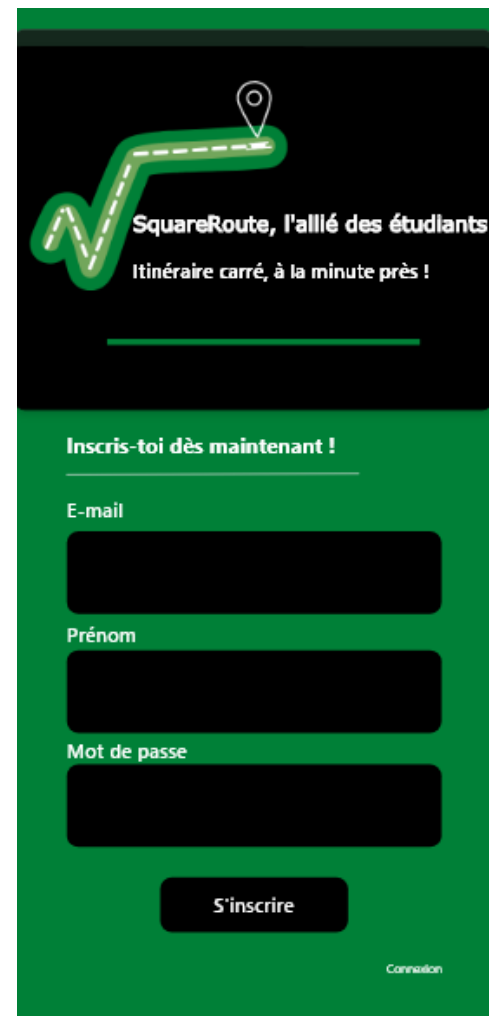
IHM ET MAQUETTES EXEMPLES



**Ecran de démarrage
(Splash screen)**



Connexion



Inscription



IHM CHARTE GRAPHIQUE (COLORS.XML)



CHARTE GRAPHIQUE

FONTS

Segoe UI

Android default Sans Serif

COLORS



#72A65B



#008037



#025023



#292828

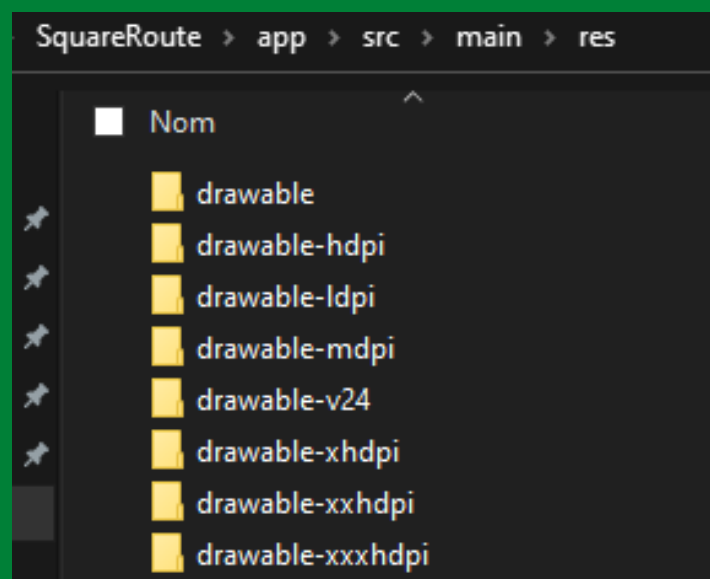


#222121

IHM

EXPLICATIONS

Pour établir un design de view moderne et rapide, Adobe XD nous a grandement facilité la tâche car il permet la création de drawables de toutes qualités (low to xxxhigh quality) qu'il est possible d'importer dans l'arborescence du projet comme ci-dessous :

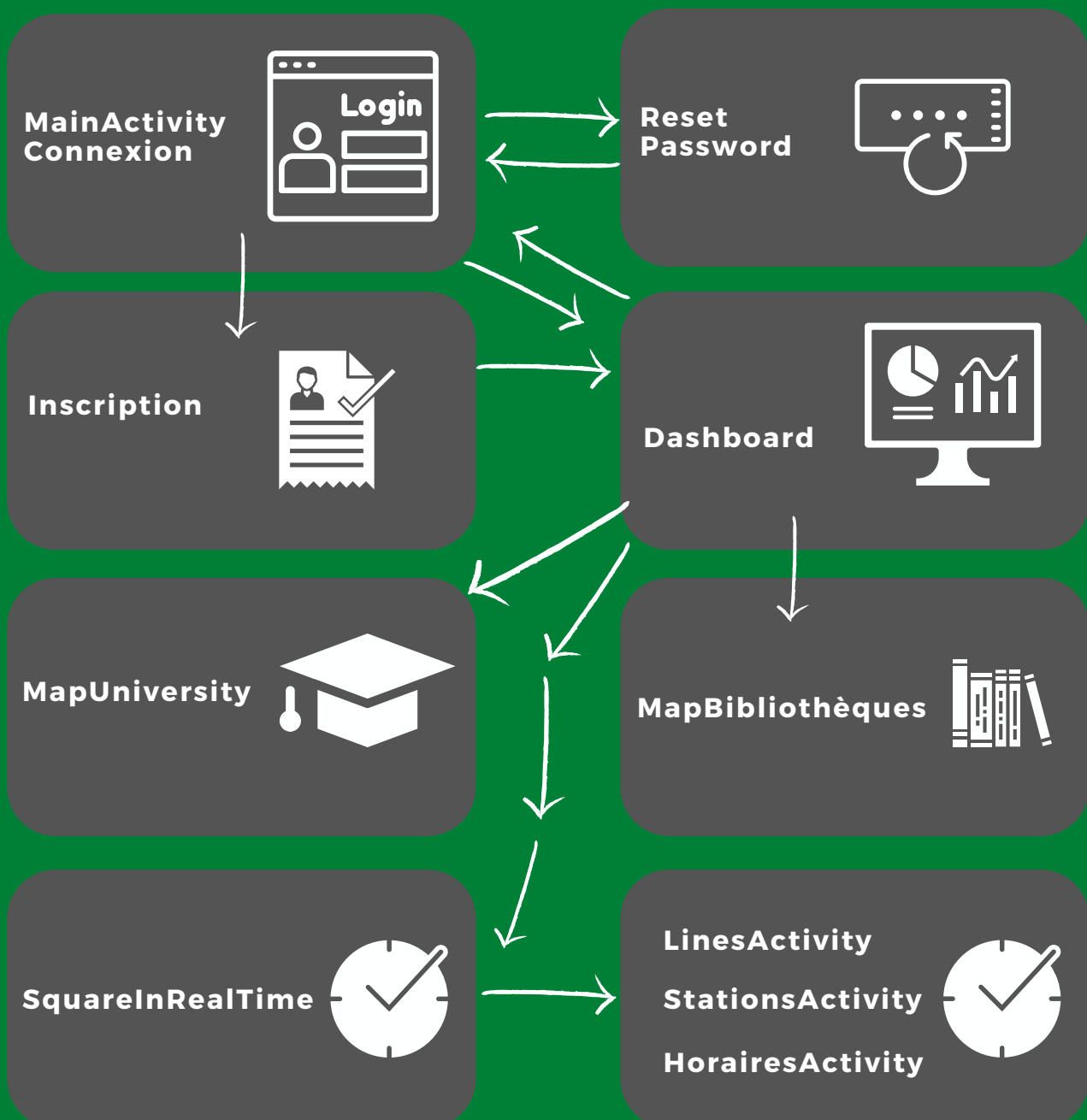


une fois importés, les drawables ont pu être positionnés en tant "qu'image view" et le design s'est fait assez rapidement grâce aux constraint layout et l'aperçu de compatibilité entre différents types d'appareils qui nous a permis de vérifier si l'affichage était correct pour tout type d'appareils. nous avons également fait en sorte pour ne pas créer de problèmes d'affichage de ne pas autoriser l'orientation paysage grâce à de simples lignes de code dans le fichier XML :

```
android:configChanges="orientation"  
android:screenOrientation="portrait"
```

Les contraintes permettant d'établir un lien entre les différents éléments du layout et leurs parents permettent, en combinaison des normes graphiques que nous avons établies d'avoir un résultat moderne et quasi-identique, voire plus complet que nos maquettes.

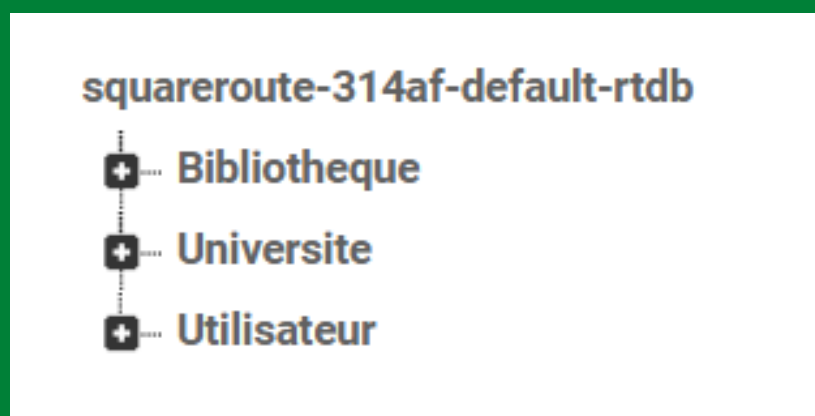
ARCHITECTURE RELATIONNELLE DES VIEW



BASE DE DONNÉES

Pour notre projet, nous avons choisi d'utiliser Firebase (realtime database) qui avec la fonctionnalité Google authenticator, permet une gestion simplifiée des sessions utilisateur.

Pour mettre en lien chacune des entités de notre realtime database, que l'on compte au nombre de 3, avec notre code Java, nous avons créé 3 classes Java User, Bibliothèque et Université qui vont entrer en dialogue avec Firebase (Pattern DAO).



Voici donc comment se structure notre base de donnée, nous n'avons pas choisi SQLite, car après avoir pesé le pour et le contre entre Firebase et SQLite, étant donné que notre application utilise déjà des services Google comme l'API Google Maps, et que l'utilisation du Google authenticator et de la realtime database sont très intuitifs grâce à la documentation et aux nombreux tutoriels très bien expliqués, par souci de cohérence nous nous sommes tournés vers Firebase.

CLIENT/SERVEUR

Au sein de SquareRoute, nous utilisons également des services réseaux, en particulier à travers la lib OKHTTP3 qui nous permet d'avoir accès aux données serveur de la RATP.

Notre classe APIServices se charge de toutes les requêtes HTTP à l'API RATP, et permet de récupérer des données codées au format JSON.

Ces données sont ensuite dynamiquement parsées au niveau de nos Lines, Stations et Horaires Activity ce qui évite la redondance de code, car nous n'avons pas codé des classes pour chaque mode de transport, mais plutôt des classes qui sont compatibles avec chaque mode de transport.

L'API interne AsyncTask d'Android permet d'autre part l'encapsulation de thread, c'est pourquoi notre classe APIServices étend cette classe.

The logo for OkHttp, featuring the text "OkHttp" in a white, sans-serif font on a teal rectangular background.

DIFFICULTÉS RENCONTRÉES

Au cours de la réalisation du projet, nous n'avons pas énormément rencontré de difficultés car certains tutoriels vidéo disponibles sur YouTube ont rendu les choses plus simples à appréhender et nous ont donné de nombreuses idées pour commencer à coder, contrairement à la documentation qui n'est dans certains cas pas très claire (Service de géolocalisation)

Néanmoins, le fait que nous n'étions que de simples débutants au départ était sans doute le plus gros problème et nous avons plus eu des difficultés à commencer ce projet qu'à le finaliser, en fin de compte.

Le fait d'avoir en quelque sorte été plongés dans l'inconnu sans aucune connaissances initiales nous a selon nous plus été préjudiciable que bénéfique puisque malgré nos connaissances en programmation Java, il n'est pas simple de partir de 0.

Au regard de cela, nous pensons qu'il aurait été plus judicieux de travailler les premières séances en petits groupes et d'apprendre à mettre en place de simples services de registration, qui à première vue paraissaient compliqués à mettre en place mais qui avec le recul sont relativement simple à appréhender, et qui d'autre part est utile pour de nombreux projets en général.

Nous avons également eu quelques problèmes d'environnement, car Android Studio est un IDE extrêmement gourmand en RAM ce qui a souvent été problématique au lancement de l'application.

Finalement, la difficulté n'était pas tellement relative au temps, mais plutôt à la restitution de connaissances dans le cadre de ce projet que l'on devait acquérir grâce à de l'autoformation et en partant de "rien".

POINTS POSITIFS

Au final, de ce projet ressortent beaucoup de points positifs car il a grandement été enrichissant pour chacun d'entre nous et a même ouvert des perspectives pour certains.

Ce projet nous a permis également de gagner en autonomie, et à renforcer notre esprit, que cela soit à l'échelle de notre équipe qu'avec d'autres équipes avec qui nous avons su expliquer le fonctionnement des layouts et de firebase par exemple.

Grâce à SquareRoute, nous avons pu imaginer un projet ambitieux, utile et innovant pour rendre service aux étudiants, et c'est sans aucun doute cette satisfaction d'apercevoir le rendu d'une application que nous avons réalisé nous mêmes de A à Z, apparaître sur nos téléphones qui nous a conduit à la "réussite" de ce projet.

Nous avons d'autre part énormément appris sur le fonctionnement des applications en général, sur l'environnement Android et les différents aspects à considérer à la conception d'une application.

C'est dans ce genre de projet que nous nous rendons compte de l'ampleur des choses que nous pouvons apprendre en autodidacte et en petit comité et c'est d'ailleurs dans ce type de contexte que le dicton "on est jamais mieux servi que par soit même" prend tout son sens, en particulier grâce à Internet qui est une mine d'or en termes de ressources pour les développeurs.

En fin de compte, nous trouvons que cette initiative de proposer des projets libres, faisant appel à notre créativité a été une expérience très enrichissante et formatrice et a presque constitué un avant-goût de la vie en entreprise.

AMÉLIORATIONS POSSIBLES

Concernant les améliorations à envisager, notre code étant tout à fait adapté et ouvert à l'évolution grâce à l'application de design patterns, il est tout à fait envisageable d'ajouter d'autres fonctionnalités telles que la mise en place d'un lien entre les horaires obtenues via les requêtes HTTP à l'API RATP pour déterminer un temps de trajet juste.

Néanmoins, nous n'avons pas exploré cette perspective pour le moment mais comptons bien reprendre cette application même après ce projet pour y ajouter encore plus de fonctionnalités, la rendre encore plus complète et répondre aux besoins des étudiants.

Ainsi cette application pourrait évoluer en un véritable compagnon de route pour les étudiants ce qui serait un accomplissement supplémentaire pour nous.