



RAPPORT TECHNIQUE – TABLETTE D'OBSERVATION PÉDAGOGIQUE

Rapport
Version 2.0 – 12/06/2018

Projet 09 – Tablette d'observation pédagogique
Auteurs: DIABY Soriba - LOPES Vincent - RAFFLIN Corentin
Formation d'ingénieur à l'IMT Atlantique : élèves 1^{ère} année
Année scolaire 2017-2018

Tuteurs: BEUGNARD Antoine - SEGARRA Maria Teresa

Destinataire: Equipe pédagogique du projet S2



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RÉSUMÉ

Avec l'émergence de nouvelles technologies, les outils actuels des enseignants peuvent être remis en question. Les nouvelles technologies peuvent augmenter l'efficacité des enseignants en supprimant des tâches laborieuses. L'application EasyNote propose aux enseignants de préparer des séances, d'observer des groupes d'élèves et de visualiser le débriefing de ces séances. Lors de ces observations, ils peuvent facilement prendre des notes sur les actions de chaque élève.

Notre objectif fut de continuer le développement de cette application, née il y a 5 ans, afin d'améliorer les fonctionnalités existantes et d'en créer de nouvelles répondant à de nouveaux besoins. A cela s'ajoute la création d'un site web permettant de préparer les séances depuis un navigateur web en utilisant une nouvelle base de données.

Dans un premier temps, nous avons évalué les nouveaux besoins des enseignants et pris connaissance de l'application actuelle. Par la suite, nous avons réalisé l'étape de conception logicielle et graphique de l'application ainsi que du site web. Cette étape de conception logicielle s'est suivie de plusieurs tests unitaires.

Rédacteur : RAFFLIN Corentin

Relecteur : LOPES Vincent

SOMMAIRE

Introduction	4
1. Analyse des besoins	5
1.1 Scénario	5
1.2 Cas d'utilisations	5
1.3 Les besoins fonctionnels	6
1.4 Les besoins non-fonctionnels	6
2. Description fonctionnelle	7
2.1 Description fonctionnelle du site web	7
2.2 Description fonctionnelle de l'application	8
3. Description technique	15
3.1 Le site web	15
3.2 La base de données	16
3.3 L'application android	18
3.4 Les ressources	21
4. Les améliorations possibles	22
4.1 Améliorations possibles du site Web	22
4.2 Améliorations possibles de l'application	22
Conclusion	23

INTRODUCTION

L'utilisation de tablettes par des enseignants de l'IMT Atlantique a pour but de leur permettre d'encadrer les élèves lors des nombreux projets se déroulant au cours de l'année. L'application EasyNote permet cette utilisation. La version précédente de l'application permettait aux enseignants de préparer une séance constituée d'un unique groupe, de noter les différents événements se produisant lors de celle-ci et d'en voir un bilan par la suite. Comme le précise le rapport du groupe de l'année précédente [1], le premier groupe de projet DEV a développé une application permettant d'observer les élèves tandis que le second groupe s'est plutôt concentré sur la préparation de l'observation et sur la persistance des données. Le troisième groupe de projet DEV a, quant à lui, codé ce qui permettait de faire une synthèse finale de la séance. Ensuite à la quatrième reprise du projet, un élève de troisième année a, lors de son projet technique d'option (S5), fusionné ces différents projets comme l'explique son rapport [2]. Le dernier groupe a ajouté une dimension spatiale et inclus des équipements dans l'observation.

Nos objectifs concernant l'application furent d'ajouter la possibilité d'observer lors d'une même séance non pas un mais plusieurs groupes, d'offrir la possibilité de créer depuis l'application toutes les composantes d'une séance [glossaire 1 à 8] - telles que des événements types [glossaire 4] - et d'améliorer les fonctionnalités existantes telles que le glisser-déposer. Concernant le site Web, l'objectif fut de pouvoir faire toute la préparation des séances sur celui-ci et de pouvoir visualiser le débriefing avec les statistiques sur le site Web.

Dans un premier temps, nous analyserons les nouveaux besoins évalués. Ensuite, nous ferons une description fonctionnelle du site Web et de l'application en comparant avec la version précédente. Puis, nous ferons une description technique du site Web et de l'application. Enfin, nous expliquerons quelles sont les améliorations possibles du site Web et de l'application.

Rédacteur : RAFFLIN Corentin

Relecteurs : DIABY Soriba, LOPES Vincent

1. ANALYSE DES BESOINS

1.1 SCÉNARIO

Pierre, enseignant à IMT Atlantique est chargé de la préparation d'une séance de projet Découverte. Pour ce faire, il utilise l'application EasyNote : il lit le manuel d'utilisation pour en comprendre le fonctionnement. Il commence par créer le projet "Projet Découverte" et les différents groupes de ce projet (groupe 1 à groupe 40) en utilisant le site Web de l'application. Pour préparer la séance, il crée un type de séance "course d'orientation" en sélectionnant des événements types préalablement créés : prend la parole, dirige le groupe, passe la parole, reste en retrait, est sur son téléphone, lit ses notes, coupe la parole, répond à une question, soutient son camarade, présente le groupe, est absent, est en retard. Il passe ensuite à la création de la séance en sélectionnant le projet "Projet Découverte" et le type de séance "course d'orientation". Pour finir, il choisit le nom de la séance "Séance n°1", les groupes, la date, l'heure de début et l'heure de fin de la séance.

Le lendemain, Marie, elle aussi enseignante à IMT Atlantique, est chargée de l'observation de deux groupes (8 et 9). Elle va ainsi se servir de l'application EasyNote sur sa tablette. Elle commence par lire le manuel pour comprendre l'utilisation de l'application. Avant que la séance ne débute, elle sélectionne la séance créée préalablement par son collègue Pierre et constate que ce dernier a oublié un groupe. Elle ajoute donc ce groupe à la séance.

La séance a commencé, un élève arrive en retard, Marie sélectionne donc l'élève sur l'application et clique sur l'événement "est en retard". Pendant la séance, elle constate qu'un élève (du groupe 9) dirige le groupe, elle sélectionne donc l'élève, puis l'événement "dirige le groupe". Un autre élève (du groupe 8) est plus en retrait, elle bascule donc sur l'observation du groupe 8, sélectionne cet élève, puis l'événement "reste en retrait". Pendant l'observation, elle regarde de temps en temps l'historique des événements pour voir tous les événements d'un groupe, leur instant, et l'élève associé, elle peut y voir par exemple "09:41:03 : Paul soutient son camarade". À la pause, elle met la séance en pause. À la fin de cette pause, elle reprend une observation en choisissant le groupe qu'elle va observer.

La séance est finie, Marie accède au débriefing de la séance. À partir du débriefing d'un groupe, elle a accès à l'historique des événements, aux statistiques liées au groupe et voit par exemple qui a été le membre le plus actif. Elle pourra, plus tard, visualiser le débriefing de chacun des groupes directement depuis le site Web.

1.2 CAS D'UTILISATIONS

Les cas d'utilisations sont résumés dans le diagramme des cas d'utilisations [annexe 1] mettant en scène le responsable [glossaire 11] et l'observateur [glossaire 12].

1.3 LES BESOINS FONCTIONNELS

Les besoins fonctionnels sont les suivants :

Gestion des groupes

- Créer ou supprimer un groupe d'élèves dans un projet
- Ajouter ou supprimer un élève d'un groupe

Gestion des séances

- Créer ou supprimer un type de séance (par exemple : "Réunion")
- Créer ou supprimer un événement type
- Créer ou supprimer une séance
- Ajouter ou supprimer un groupe d'une séance (en pause)
- Mettre une séance en pause
- Observer un groupe : la séance passe donc de "en pause" à "en cours"

Observation des groupes

- Changer d'observation
- Attribuer un événement à un élève

Gestion du débriefing

- Accéder au débriefing d'une observation terminée

Gestion des informations depuis le site Web

- Depuis le site Web, créer, supprimer ou modifier les données suivantes : projets, événements types, types de séance, élèves, groupes, séances
- Inscrire une liste d'élèves depuis le site Web
- Visualiser le débriefing et les statistiques depuis le site Web

Gestion du transfert de données entre la tablette et le site Web

- Actualiser les données entre la tablette et le site Web

1.4 LES BESOINS NON-FONCTIONNELS

- Facilité d'utilisation : l'application doit être simple et intuitive
- Maintenabilité : Le code source doit être facile à modifier
- Portabilité : L'application doit avoir un bon affichage quelle que soit la définition d'écran de la tablette utilisée

Rédacteur : DIABY Soriba

Relecteurs : RAFFLIN Corentin, LOPES Vincent

2. DESCRIPTION FONCTIONNELLE

2.1 DESCRIPTION FONCTIONNELLE DU SITE WEB

Le site Web est accessible depuis le lien suivant : <http://projets.telecom-bretagne.eu/easyobs/>

Le site Web est constitué d'un menu offrant, pour chaque partie, des interactions possibles avec une partie de la base de données.

Les différentes interactions possibles avec la base de données sont :

- créer ou supprimer un projet,
- créer, supprimer ou modifier un groupe d'élèves d'un projet,
- rechercher, créer ou supprimer un type d'événement,
- créer, supprimer, ou modifier un type de séance,
- créer, supprimer ou modifier une séance,
- visualiser le débriefing d'une séance,
- consulter la liste des élèves et rechercher un élève.

Figure 1 : Interface du site Web permettant de gérer les groupes d'un projet

Par exemple, la partie "Groupes" [figure 1] permet de sélectionner un projet, puis pour ce projet, de visualiser la liste des groupes existants ainsi que les élèves le composant, de supprimer un groupe, d'ajouter un groupe, de supprimer tous les groupes existants, d'ajouter un élève à un groupe et enfin de supprimer un élève d'un groupe.

Le groupe numéro 2 a bien été ajouté

Figure 2 : Exemple d'alerte Bootstrap lors de l'ajout du groupe 2 à un projet

Pour chaque action ayant un effet sur la base de données, une alerte Bootstrap [figure 2] [glossaire 13] apparaît sur la page suivante indiquant si l'action s'est bien déroulée ou non avec des tests de vérification au niveau des requêtes PHP, des requêtes SQL et vérification si nécessaire que les requêtes SQL ont bien supprimé ou ajouté des lignes.

Rédacteur : LOPES Vincent

Relecteur : RAFFLIN Corentin

2.2 DESCRIPTION FONCTIONNELLE DE L'APPLICATION

LES TROIS PARTIES DE L'APPLICATION



Figure 3 : Menu de l'application de la version précédente (à gauche) et de la nouvelle version (à droite)

Le menu de l'application précédente [figure 3] permettait d'accéder à quatre parties dont l'ordre n'avait pas de sens et ne représentait pas le cheminement d'un utilisateur. Il a été modifié afin de mieux mettre en avant la démarche d'un utilisateur lors de l'utilisation de la tablette : d'abord il prépare la séance, puis y accède et enfin visualise son débriefing. Le menu permet ainsi d'accéder à trois autres activités [glossaire 9] : la première permet de manipuler les objets de la base de données commune entre le site web et la tablette, afin de préparer une séance depuis la tablette, la deuxième permet d'accéder à une observation via une séance et la dernière permet d'accéder au débriefing.



L'activité de gestion de la base de données présente sur la version précédente [figure 4] permettait d'insérer un fichier .txt contenant du code SQL afin d'insérer rapidement des informations et elle permettait également d'insérer une liste d'étudiants. Ces fonctionnalités n'ont plus d'intérêt avec l'introduction du site Web qui a pour but d'insérer rapidement ces données. De plus, la connexion avec la base de données depuis la tablette ne nécessite pas une activité à part [voir section 3.2].

Figure 4 : Activité de gestion de la base de données de la version précédente

PRÉPARATION DE LA SÉANCE



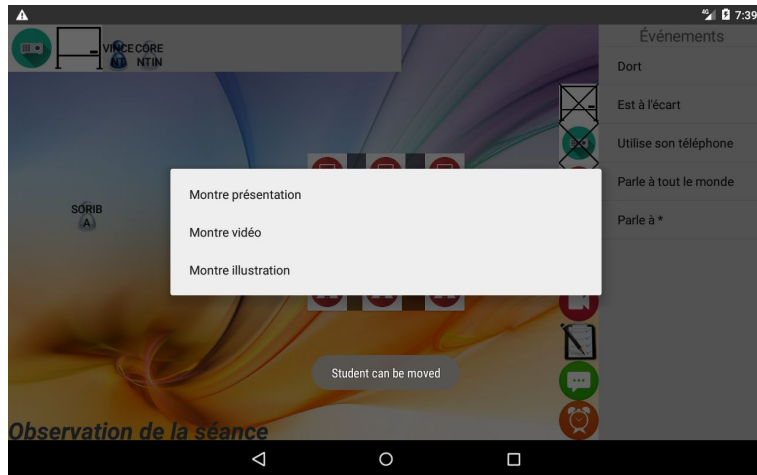
Figure 5 : Préparation de la séance de la version précédente (à gauche) et de la nouvelle version (à droite)

Toujours dans un souci de facilité d'utilisation, les boutons définis dans le layout [glossaire 10] de l'activité de préparation de la séance [figure 5] proposent désormais un ordre logique d'accès aux autres activités. Précédemment l'application ne permettait que de créer, modifier ou supprimer des projets, des élèves (cette activité était accessible depuis le menu principal), des groupes et des séances. Dès cette année, l'application est beaucoup plus personnalisable puisqu'un enseignant peut créer toutes les composantes d'une observation.



Figure 6 : Gestion de la liste des élèves de la version précédente et gestion de la liste des événements ajoutée cette année

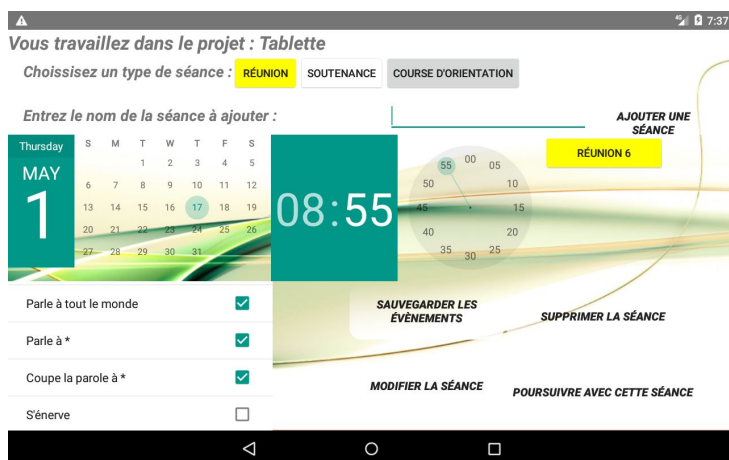
Nous avons en effet ajouté la possibilité de gérer les événements types, appelés simplement événements pour un utilisateur [figure 6]. Ce design, ainsi que le design des layouts de préparation, s'est grandement basé sur celui de l'activité de gestion de la liste des élèves des années précédentes, en adaptant celui-ci pour convenir aux particularités de chaque composante. Ainsi, pour l'activité de gestion des événements types [figure 6], nous avons offert la possibilité d'ajouter des événements types aux équipements [glossaire 6], qui eux, restent prédéfinis.



En effet, précédemment les équipements n'avaient que des événements types prédéfinis. Par exemple, les ordinateurs ne présentaient que les événements types visibles [figure 7].

Figure 7 : Événements types définis pour les ordinateurs sur la version précédente

De même, l'application permet de gérer les types de séance en y incluant les événements de notre choix. Cette interface possède la même allure que celle en figure 6 à droite.



Une grande particularité de notre application est d'être moins contraignante que les versions précédentes : auparavant pour accéder à la création d'une séance [figure 8] il était nécessaire de passer par une activité projet, de revenir au sous-menu, de passer par l'activité de gestion des groupes, de revenir au sous-menu et enfin de pouvoir accéder à l'activité de gestion des séances. Cela parce que les développeurs ont utilisé une classe Transition contraignante [section 3.3].

Figure 8 : Création d'une séance de la version précédente

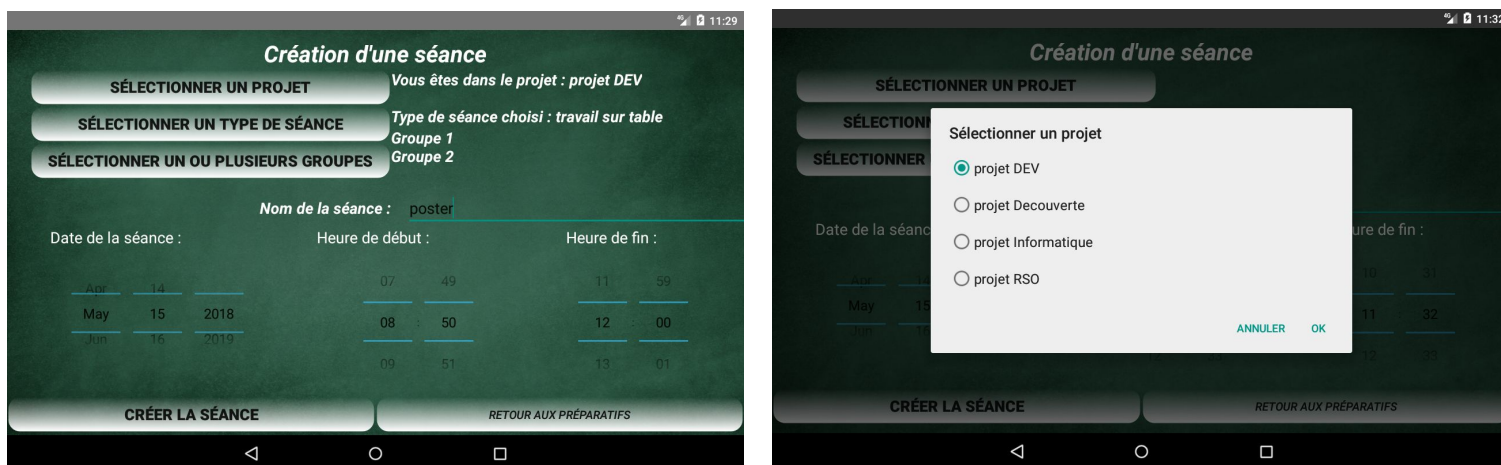


Figure 9 : Création d'une séance de la nouvelle version (à gauche) et utilisation d'un AlertDialog (à droite)

Dorénavant l'application permet d'accéder à chacune des activités de gestion indépendamment des autres, cette option se fait grâce au AlertDialog [section 3.3].

Ainsi, nous pouvons lors de la création d'une séance choisir directement les informations nécessaires (projet, type de séance et groupes) depuis une seule activité [figure 9].

La création de séance prend dorénavant un autre critère qui est l'heure de fin. L'utilité de ces informations horaires est maintenant exploitée comme expliqué dans la suite, alors qu'elles n'étaient pas utilisées dans la version précédente.

Une autre particularité de la création d'une séance est de ne plus choisir les événements types sur cette activité. En effet, les événements types sont dorénavant inclus dans le type de séance. Cela s'est traduit par une modification de la base de données [section 3.2]. Nous avons en effet fait en sorte que les informations qui n'apportaient rien ne soient plus présentes : les types de séance sont donc associés aux événements types, et la notion de type de projet qui n'apportait aucune information a été supprimée.

LA PHASE D'OBSERVATION



Une fois la séance préparée, l'utilisateur peut se rendre dans l'activité de gestion des séances [figure 10]. A partir de celle-ci, il peut rechercher puis modifier ou supprimer une séance. Il peut aussi sélectionner la séance et un des groupes participants afin d'accéder à l'observation proprement dite. Par la suite ces deux fonctionnalités pourraient être séparées en deux activités distinctes.

Figure 10 : Gestion des séances et choix d'une observation

Trois cas peuvent survenir lors du lancement d'une observation, si l'utilisateur souhaite accéder à l'observation alors que la séance :

- est en cours, c'est-à-dire dans l'intervalle défini par la date et heure de début et celle de fin, alors l'utilisateur accède à l'observation.
- est terminée, c'est-à-dire après la date et heure de fin, alors l'utilisateur en est informé et a la possibilité d'accéder au débriefing de cette séance.
- n'a pas encore commencé, c'est-à-dire avant la date et heure de début, alors l'utilisateur en est informé et a la possibilité d'accéder tout de même à la séance.

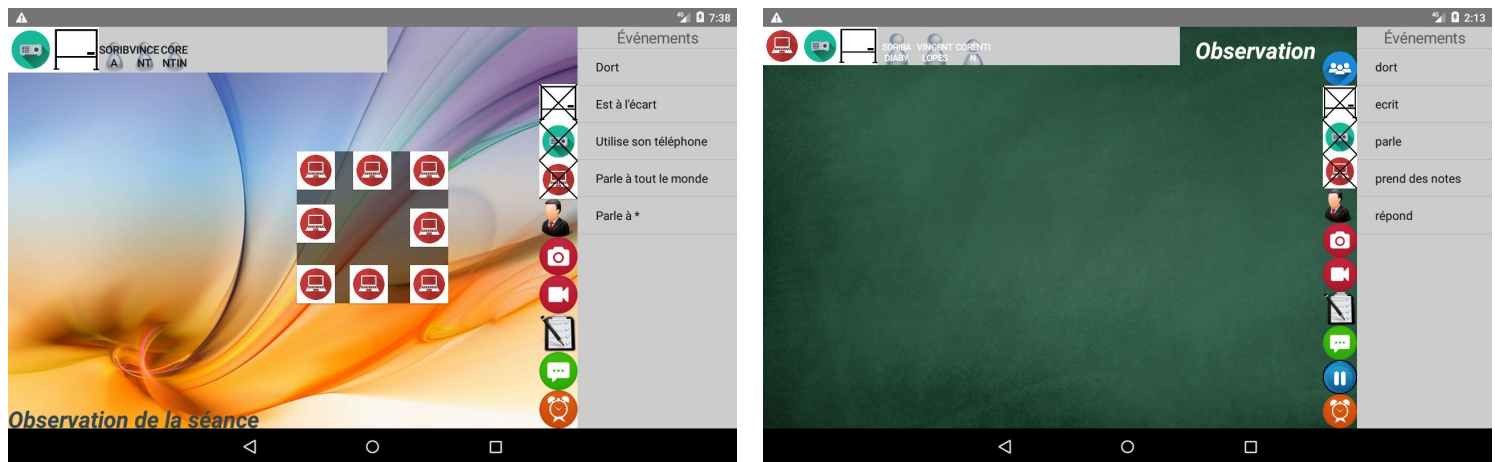


Figure 11 : Observation d'une séance de la version précédente (à gauche) et de la nouvelle version (à droite)

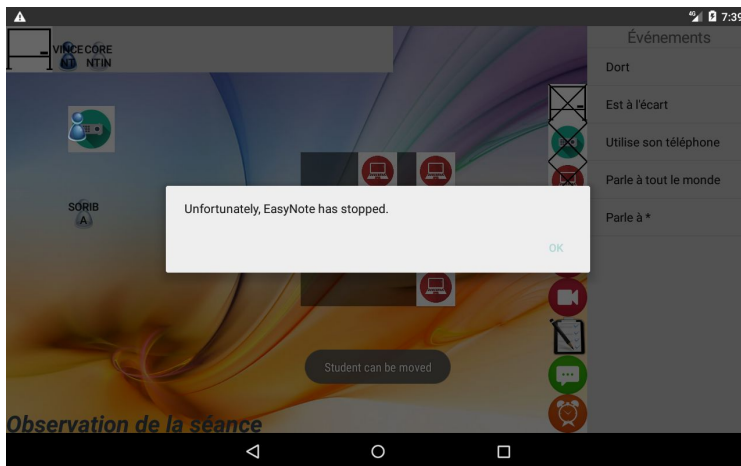
Concernant l'observation, nous retrouvons l'interface créée par les groupes précédents à quelques différences près [figure 11].

De plusieurs boutons d'ordinateurs placés et fixés au centre, nous passons à un seul bouton ordinateur comme pour le projecteur et le tableau. Ce choix a été fait car nous trouvions que ce bloc centré n'était pas utile et que nous pourrions faire mieux par la suite [section 4.2].

Les équipements sont maintenant prédéfinis, au nombre de 4 – le tableau, le projecteur, l'ordinateur et le tuteur – et un glisser-déposer d'un bouton élève sur un de ces boutons, excepté le bouton du professeur, permettra d'afficher une liste des événements types associés. Le projecteur et le tableau ont donc désormais plus qu'une simple fonction graphique. Le bouton du professeur, lui, s'utilise sans action préalable sur un élève, et permet d'ajouter aussi un événement. Ce bouton est donc mis sur la liste des boutons à droite où nous retrouvons les autres boutons permettant de : prendre une photo, afficher l'historique, ajouter un commentaire et supprimer un bouton d'un équipement.

Pour répondre à l'objectif principal de pouvoir observer plusieurs groupes au sein d'une séance, nous avons ajouté un nouveau bouton, situé au-dessus des boutons cités préalablement, qui permet de choisir un autre groupe (appartenant à la même séance) afin de l'observer. À l'heure actuelle, ce changement de groupe se fait par l'envoi d'un Parcelable [section 3.3] qui permet de relancer l'activité avec les nouvelles informations, mais cela ne permet pas de garder la position des boutons déplacés.

Un bouton a aussi été implémenté afin de mettre la séance en pause et de revenir au menu principal. Ceci dans le but de pouvoir modifier une séance en pause pour, par exemple, ajouter un groupe ou créer de nouveaux événements types.



La méthode de glisser-déposer a été améliorée afin que l'application ne s'arrête plus si le professeur dépose un bouton élève en dehors de la zone fixée (par exemple sur la liste des événements ou un autre bouton). Comme l'a décrit le groupe précédent, ce souci pouvait « frustrer l'utilisateur si jamais il essay[ait] de déposer un élève en dehors de la zone possible ».

Figure 12 : Plantage de l'application précédente lors d'un glisser-déposer dans une mauvaise zone

L'utilisateur peut toujours déplacer les boutons élèves ainsi que les boutons équipements à sa guise afin de simuler au mieux la disposition réelle.

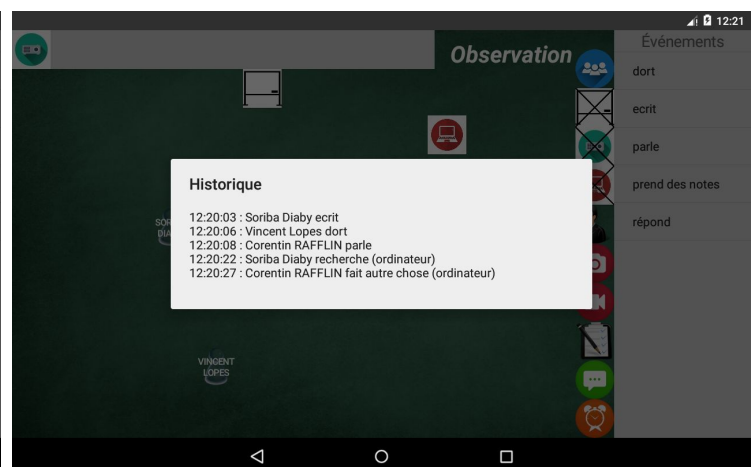
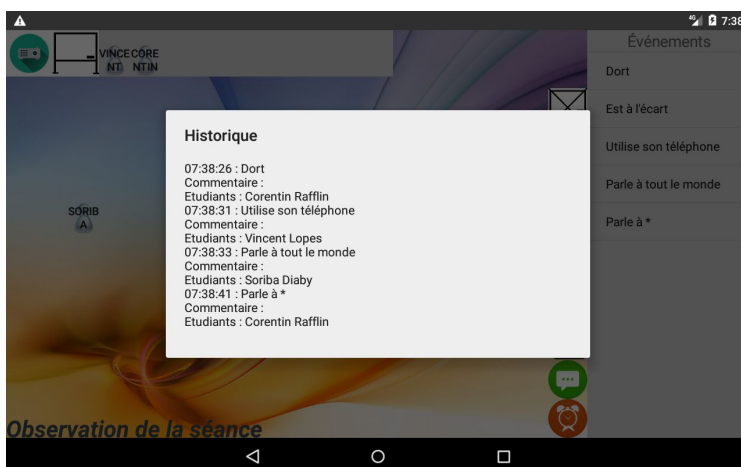
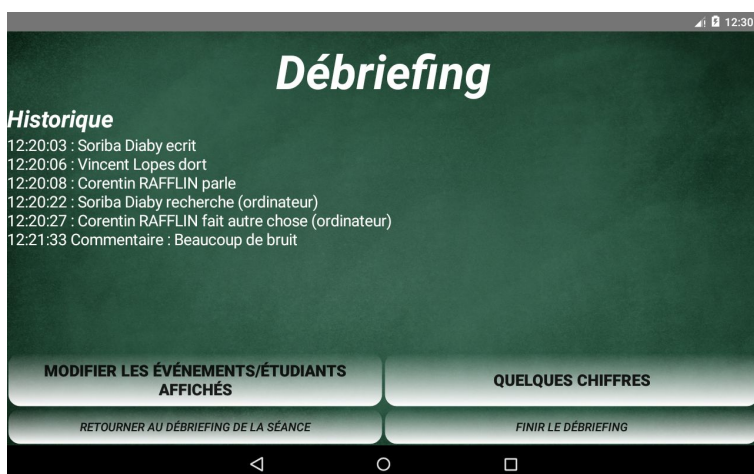


Figure 13 : Historique de la version précédente (à gauche) et de la nouvelle version (à droite)

L'historique [figure 13] affiché est propre à une observation et est conservé malgré le basculement entre les groupes. En effet, il est construit à partir de la base de données et donc les données sont conservées malgré les basculements. On y visualise les événements qui ont eu lieu à chaque instant et nous avons fait en sorte d'améliorer la lecture de cet historique.

PHASE DE DÉBRIEFING

Arrivé au débriefing soit par l'activité de gestion des débriefings, accessible depuis le menu principal, soit à partir d'une observation en la terminant, l'utilisateur retrouve un débriefing de la séance qui lui permet de naviguer entre les observations. Cette activité pourrait disposer d'autres fonctionnalités [section 4.2].



Au niveau du débriefing d'une observation [figure 14], l'utilisateur retrouve l'historique. Il a la possibilité d'afficher dans l'historique seulement les événements liés à des étudiants et des événements types spécifiés. Il peut aussi accéder aux statistiques de cette observation.

Figure 14 : Débriefing d'une observation de la nouvelle application

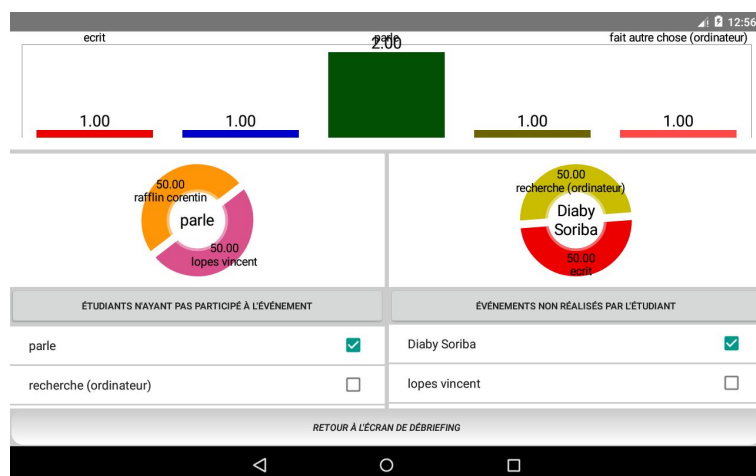
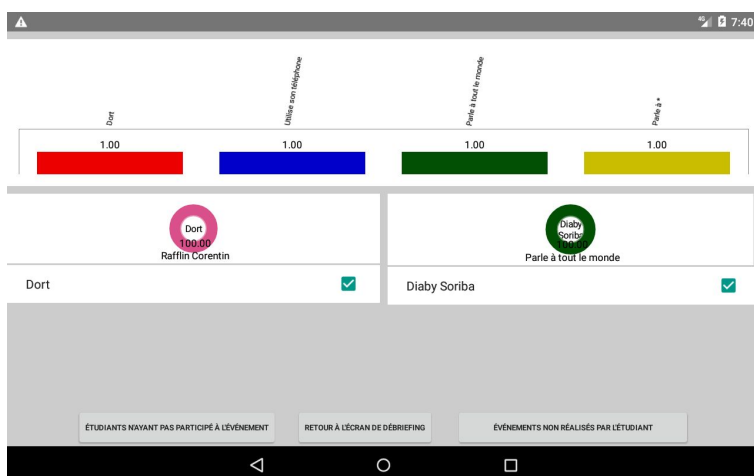


Figure 15 : Statistiques d'une observation de la version précédente (à gauche) et de la nouvelle version (à droite)

Sur la partie statistique [figure 15], on retrouve des éléments définis par les groupes précédents que nous avons simplement rendus compatibles avec notre application et modifiés pour en améliorer le design en rendant cette partie plus lisible. D'une part, on retrouve en haut de l'écran les événements avec le nombre d'utilisations, des soucis de design sont encore à régler avec notamment le nom de certains événements types qui n'apparaissent pas. D'autre part, on retrouve des diagrammes circulaires pour un événement type donné et un autre pour un étudiant donné, il est aussi possible de trouver les étudiants n'ayant pas effectué un événement type ainsi que de savoir les événements types non effectués par un étudiant.

Rédacteur : RAFFLIN Corentin

Relecteurs : DIABY Soriba, LOPES Vincent

3. DESCRIPTION TECHNIQUE

3.1 LE SITE WEB

Le site est codé en HTML5 et utilise des feuilles de style du framework web Bootstrap. Les classes CSS de Bootstrap contribuent ainsi au design du site Web. La partie dynamique est assurée par du PHP : les accès à la base de données se font via les commandes de type `mysqli_*` et la majorité des données transitant entre les pages se font via POST, pour éviter qu'une requête de suppression d'un élève soit effectuée deux fois à tort.

Bien que Bootstrap permette de créer un responsive design qui s'adapte à la taille de l'écran de l'utilisateur, qu'il soit sur smartphone, tablette ou ordinateur, la plupart des pages sont plus adaptées à une visualisation sur un écran d'ordinateur.

Concernant le design du site Web, chaque page du site a un design adapté en fonction des informations qui doivent s'afficher dessus. Par exemple, la partie débriefing est constituée de plusieurs onglets (cf .nav-tabs de Bootstrap), un par groupe. Dans chaque onglet s'affiche un tableau HTML avec la liste des événements associés à ce groupe. Ce tableau permet de trier les événements selon plusieurs critères.

Accueil	Séance 1					
Événements	Groupe 1	Groupe 2				
Groupes	Heure	Prénom	Nom	Événement	Équipement	Commentaire
Types de séances	16:00	Guy	TARES	Est sur son téléphone		
Projets						
Elèves						
Séances						

Figure 16 : Interface de visualisation d'un débriefing sur le site Web

Le tri des tableaux se fait en l'état côté serveur par le SGBD. Il pourra à l'avenir être implémenté côté client en JavaScript.

La partie utilisateur du site Web a été protégée contre les failles XSS et les injections SQL du mieux que nous pouvions.

Rédacteur : LOPES Vincent

Relecteur : RAFFLIN Corentin

3.2 LA BASE DE DONNÉES

MODÈLE CONCEPTUEL

La base de données, dont le modèle conceptuel est visible en figure 16, contient toutes les informations qui sont conservées et accessibles depuis le serveur de notre site Web.

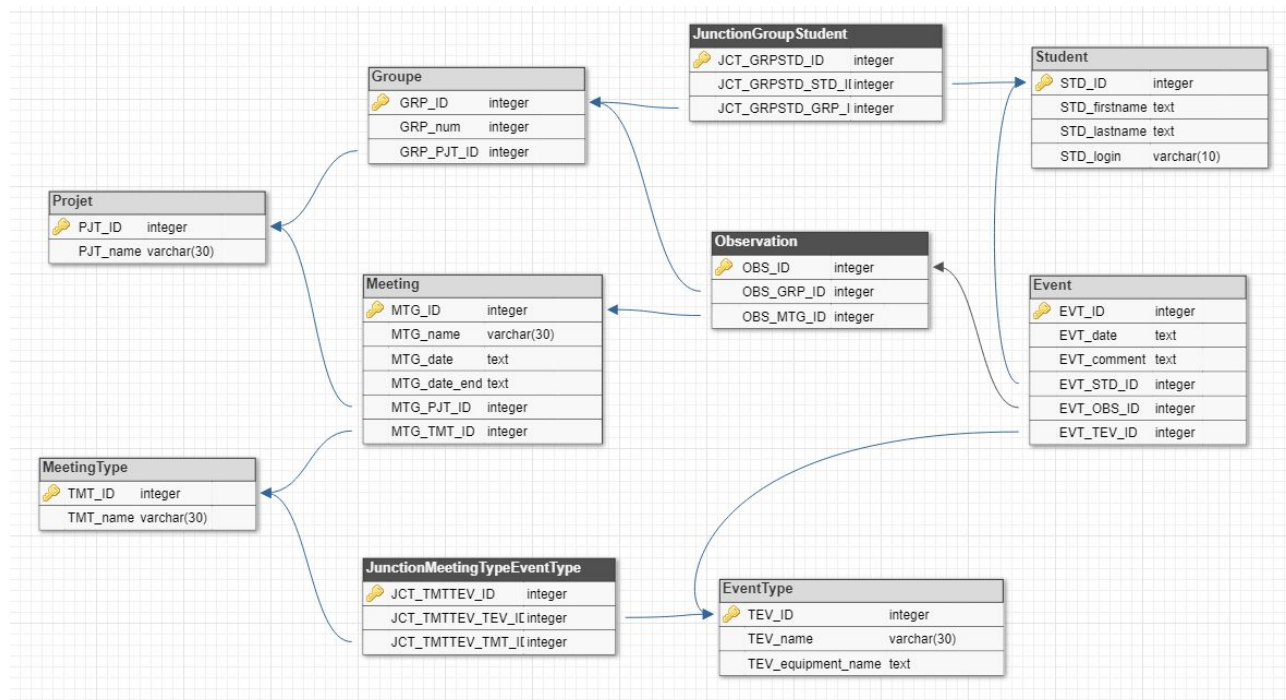


Figure 17 : Modèle conceptuel de la base de données de la nouvelle application

Nous avons supprimé les éléments qui n'apportaient pas d'informations et dont les tables n'étaient pas utiles. Ainsi, nous pouvions trouver dans les versions précédentes les tables *ProjectType*, *Equipment* et *EventEquipment*. Les rôles de ces deux dernières tables n'étaient pas déterminants, c'est pourquoi nous avons décidé d'inclure la propriété *EVT_equipmentname* dans la table *Event* et de supprimer ces deux tables. Nous avons d'autre part créé d'autres jonctions pour permettre d'attribuer des rôles aux entités n'en ayant pas. Ainsi, un type de séance est dorénavant associé à un ou plusieurs événements types avec la table *JunctionMeetingTypeEventType* et la table *JunctionMeetingEventType* a donc aussi été supprimée.

Une nouvelle table essentielle est la table *Observation* qui a un rôle de jonction entre les tables *Meeting* et *Group*.

Les différentes tables de la base de données présentent beaucoup de dépendances entre elles. Nous avons donc ajouté l'expression "ON DELETE CASCADE" de SQL lors des créations des contraintes d'intégrité. Cette expression permet, lors de la suppression d'un élément, de supprimer toutes les dépendances de cet élément. Par exemple, lors de la suppression d'un projet seront supprimés tous les groupes associés à ce projet.

ETABLISSEMENT DE LA CONNEXION DEPUIS LA TABLETTE

Au départ, nous avons pensé que nous ferions en sorte que l'application dispose d'une activité pour gérer la base de données à la manière des années précédentes. Cette activité aurait possédé un bouton pour récupérer l'intégralité de la base de données du serveur et l'aurait copiée sur la tablette. Ainsi les opérations sur la base de données auraient encore été réalisées en SQLite sur l'application. Un autre bouton aurait permis de renvoyer les informations de la tablette vers le serveur du site Web en écrasant la base précédente. Cette méthode aurait permis de ne pas nécessiter une connexion internet permanente. Toutefois elle avait un autre défaut lors de l'export, puisque plusieurs enseignants auraient une base de données différente sur leur tablette et lors de l'export, leur base de données écraserait les modifications faites par les autres professeurs qui auraient exporté préalablement.

Nous avons donc décidé qu'il y aurait une seule base de données commune entre toutes les tablettes et le site Web afin d'éviter les problèmes de synchronisation. Cette solution pose encore des problèmes puisqu'il faut une connexion internet permanente et qu'un professeur pourrait modifier par erreur les données d'un autre professeur en train d'observer un groupe (en supprimant un groupe d'un autre professeur par exemple). Mais cela a aussi l'avantage que toutes les modifications peuvent être faites en temps réel.

Par cette méthode, nous nous détachons du diagramme des cas d'utilisation [annexe 1] et de l'analyse des besoins qui font apparaître une utilisation pour actualiser les données sur la tablette, ceci n'étant alors pas nécessaire car automatisé.

Nous nous sommes alors basés sur un tutoriel [3] pour établir la connexion entre l'application et le site Web [figure 18]. Nous avons utilisé la classe JSONParser telle que présentée dans le tutoriel afin d'établir une connexion via des requêtes HTTP.

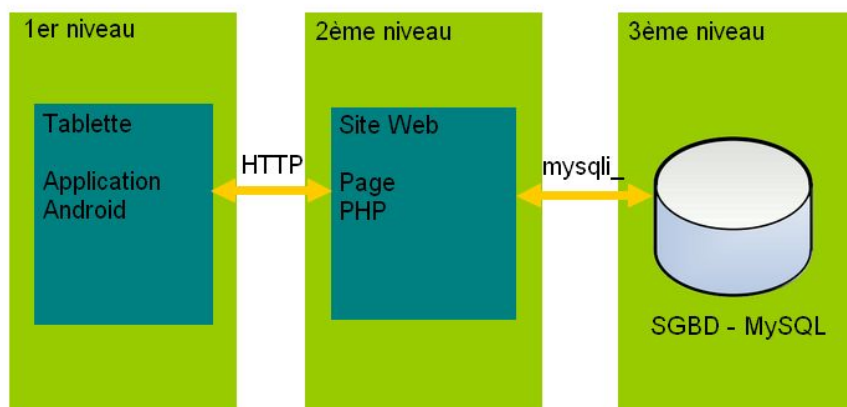


Figure 18 : Schéma modélisant la connexion entre les différents éléments

Depuis la tablette, nous envoyons des informations à une page PHP via une méthode définie dans la classe JSONParser qui permet de faire des requêtes HTTP à une URL précisée. La page PHP contient les requêtes MySQL à exécuter et renvoie si nécessaire des informations de types JSON à l'application. Cette fonctionnalité est opérée sur la tablette dans un autre fil d'exécution [section 3.3].

Rédacteurs : RAFFLIN Corentin, LOPES Vincent

Relecteur : DIABY Soriba

3.3 L'APPLICATION ANDROID

PACKAGE ACTIVITIES

Ce package contient toutes les activités de l'application dont la description est donnée dans le tableau en annexe 2.

PACKAGE OBJECTS

Le package *Objects* contient les classes suivantes : *Equipment*, *Event*, *EventType*, *Group*, *Meeting*, *MeetingType*, *Observation*, *Project* et *Student*. Toutes ces classes, sauf la classe *Equipment*, ont une correspondance dans la base de données SQL. La classe *Equipment* n'a pas de correspondance dans la base de données car nous avons décidé que les catégories d'équipements étaient fixes : ordinateur, tableau, projecteur et tuteur ; ceci car nous pensons qu'il n'y a pas beaucoup d'autres équipements possibles qui pourraient être ajoutés et qu'en ajouter pourrait saturer l'écran inutilement. Nous l'avons tout de même gardée si il devait y avoir d'autres équipements par la suite.

Précédemment, ce package contenait d'autres objets que nous n'avons pas réutilisés dans notre conception : l'objet *Debriefing* mis en relation avec l'objet *Timeline* avait pour but de construire dans l'activité de débriefing un fil chronologique des événements. Malheureusement cette idée, bien qu'intéressante, n'a pas été terminée par le groupe précédent et la reprise fut trop compliquée puisque tous les attributs y étaient déclarés static, c'est-à-dire que ces attributs étaient communs à toutes les instances de cette classe *Debriefing*. En effet, préalablement, il n'y avait qu'un seul groupe observé et les développeurs utilisaient la classe *Transition* qui gardait les informations d'une séance (au sens observation pour les années précédentes). Nous avons donc décidé de garder l'historique et que pouvoir sélectionner les événements propres à un étudiant était tout aussi intéressant et pratique pour le débriefing, qui de toute manière sera à terme lu sur le site Web.

Le diagramme de classe [figure 19] montre le lien entre les différents objets, leurs attributs et les méthodes clés (les constructeurs, setters et getters ne sont pas indiqués). On aperçoit dans ce diagramme de nombreux identifiants id qui permettent par exemple de retrouver facilement dans la base de données des informations recherchées.

Ce diagramme de classe est proche de celui de 2016 [2] avec la classe *Equipment* et *Observation* en plus. On y trouve cependant moins de classes que dans celui de 2017 où une classe *Environment* qui avait pour but de définir le nombre d'équipements (tableau, ordinateur et projecteur) préalablement à la séance, mais cette classe n'avait pas été exploitée dans le code. Il y avait aussi une classe par équipement - classe *Board*, *Computer* et *Projector* - sans pour autant apporter d'informations utiles. A la place de ces classes, on trouve dans notre diagramme de classe des attributs *equipment* et *equipmentName* de types String respectivement dans les classes *Event* et *EventType*.

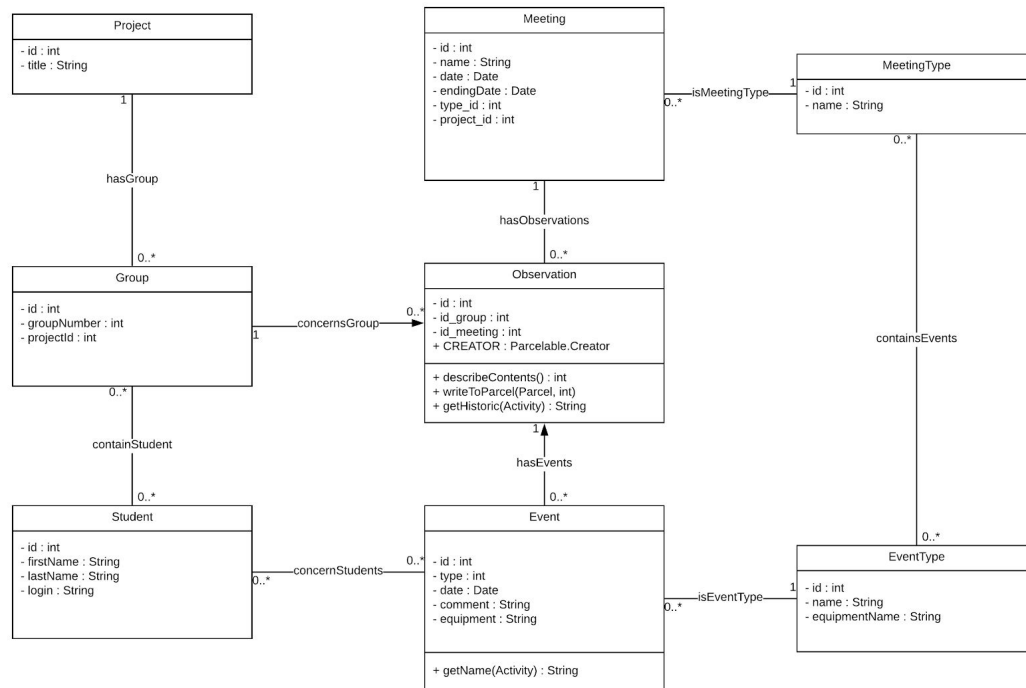


Figure 19 : Diagramme de classe de l'application actuelle

PACKAGE TOOLS

Ce package contient les classes qui ne sont ni des activités ni des objets. On y trouve la classe *Database* qui permet de faire des opérations avec la base de données. Auparavant, cette classe permettait de faire les requêtes sur la base de données SQLite présente sur la tablette. Dorénavant, ce sont les pages PHP qui contiennent les requêtes SQL, la classe *Database* ne fait que la communication à l'aide des objets de type JSON. Pour cela, le package tools contient la classe *JSONParser* qui permet de faire des requêtes HTTP avec les méthodes GET ou POST pour obtenir un objet JSON depuis une URL.

Ce package contenait une autre classe, la classe *FilePicker*, qui permettait de traiter les fichiers .txt ou .xml contenant respectivement des requêtes SQL et des listes d'élèves. Cette classe était donc utilisée dans l'activité de gestion de la base de données de la version précédente de l'application, mais comme nous l'avons expliqué, avec l'introduction du site Web, cette activité n'a plus d'intérêt et donc la classe *FilePicker* non plus.

LES ÉLÉMENTS CLÉS

PARCELABLE

Les années précédentes, la transmission des données entre les activités n'était pas optimale puisqu'elle se faisait à l'aide d'une classe *Transition* qui stockait les données sélectionnées par l'utilisateur pour être réutilisées dans d'autres activités.

Cette année, nous avons supprimé la classe *Transition* et à la place, nous avons implémenté l'interface *Parcelable* [4] dans la classe *Observation* du package *Objects*. Implémenter cette classe permet de transmettre toute instance de la classe *Observation* d'une activité vers une autre en l'incluant dans un *Intent*, l'agent qui permet le mécanisme d'échange entre activités. La classe *Observation* est la seule classe nécessitant ce paramètre puisque nous pouvons récupérer toutes les informations d'une observation (les groupes, les événements types...) à partir d'une instance de celle-ci.

ASYNCTASK : TRAITEMENT DES FILS D'EXÉCUTION

Une activité qui réagit en un temps trop long est tuée par l'*ActivityManager*. Les interfaces homme-machine sont dans un fil d'exécution qui leur est propre et sont chargées de l'interaction avec l'utilisateur [5]. Pour effectuer un traitement, il faut lancer un autre fil d'exécution, dit d'arrière-plan, car aucun traitement ne doit être effectué dans le fil d'exécution d'IHM, aussi appelé thread UI (UI pour User Interface, « interface utilisateur » en français). Pour utiliser les fils d'exécutions, deux objets sont disponibles nativement sur Android, les Handlers et les AsyncTasks. C'est ce dernier objet que nous avons choisi d'utiliser.

AsyncTask est une classe permettant un usage propre et facile des fils d'exécutions. Cette classe permet de faire des opérations en arrière-plan sans avoir à manipuler les fils d'exécutions.

Cette classe a été nécessaire pour les connexions via HTTP pour ne pas avoir l'exception *NetworkOnMainThreadException*, l'exception levée quand une application tente de faire une opération via internet sur son fil d'exécution principal.

ALERTDIALOG

Afin de faire en sorte que les activités soient plus indépendantes, nous avons utilisé, pour les sélections des objets, la classe *AlertDialog* qui permet d'afficher, dans l'activité, une fenêtre contenant une liste de chaînes de caractères et des boutons. Le résultat des *AlertDialog* est visible sur la figure 10, à droite.

Ceci permet de faire toutes les sélections sur une seule activité sans saturer l'écran de celle-ci. A chaque fois que l'on fait une sélection, nous avons fait en sorte que le choix soit noté à côté afin que l'utilisateur puisse vérifier le résultat de son action.

Rédacteur : RAFFLIN Corentin

Relecteur : LOPES Vincent

3.4 LES RESSOURCES

LES IMAGES

Les images sont contenues dans le répertoire *drawable*. L'idéal pour gérer les différentes densités d'écran est de placer dans ce dossier plusieurs copies d'une même image mais adaptées à chaque densité. Ceci se fait en ajoutant un quantificateur qui précise pour quel matériel un fichier est destiné. Pour notre application, certaines images respectent déjà ce critère, cependant nous n'avons pas encore pris le soin de respecter ceci pour toutes les images, ce qui peut être contraignant en cas de changement de tablette.

LES LAYOUTS

Chaque activité possède son layout [annexe 3], un fichier XML qui définit l'apparence d'un écran, ceci permet de séparer son apparence de ses actions.

La plupart des activités, sauf celle d'observation, sont réalisées avec des *LinearLayout* qui permettent de placer les éléments sur une ligne mais ne permettent pas une adaptation idéale aux changements d'écrans, à la différence d'un *RelativeLayout* qui définit les positions des éléments les uns par rapport aux autres.

LES VALEURS

Ce dossier contient les informations qui pourront être réutilisées dans le code afin d'éviter des problèmes de redondance.

Le fichier *string.xml* contient les chaînes de caractères permettant de changer l'affichage de texte efficacement et permettant éventuellement d'adapter facilement l'application à un changement de langue.

Le fichier *styles.xml* contient les styles de l'application, tel que le style des boutons d'un menu ou celui des boutons permettant de faire des requêtes.

Le fichier *colors.xml* contient les couleurs pouvant être utilisées dans l'application. Cela permet de changer facilement les couleurs dominantes de l'application.

Le fichier *dimens.xml* contient les dimensions pouvant être utilisées dans l'application.

Rédacteur : RAFFLIN Corentin

Relecteur : LOPES Vincent

4. LES AMÉLIORATIONS POSSIBLES

4.1 AMÉLIORATIONS POSSIBLES DU SITE WEB

Les améliorations envisageables pour une reprise par un futur groupe sont :

- Observation des statistiques d'une observation avec graphiques
- Débriefing et statistiques d'une séance (donc d'un ensemble d'observations)
- Connexion au LDAP de l'école pour avoir les listes des élèves
- Sécuriser le site Web en ajoutant une interface de connexion
- Implémenter les tris de tableaux côté client en JavaScript

Rédacteur : LOPES Vincent

Relecteur : RAFFLIN Corentin

4.2 AMÉLIORATIONS POSSIBLES DE L'APPLICATION

Les améliorations envisageables pour une reprise par un futur groupe sont :

- ajouts des options de modification d'une séance en pause : ajouts ou suppressions de groupes et changement du type de séance.
- personnalisation des affichages. Ainsi pour le débriefing nous pourrions lire : "débriefing du groupe X" (avec X un numéro) dans le titre au lieu de "débriefing".
- les positions des boutons élèves et équipements seront mémorisées au sein d'une séance en cours malgré le changement d'observation. Pour cela, l'utilisation des fragments [glossaire 14] est une solution.
- améliorations des layouts pour qu'ils soient adaptés à différentes tablettes et adapter les images pour chaque densité d'écran.
- ajout d'une configuration prédéfinie des séances : nombre d'ordinateurs, de tableaux... Cette information pourrait être contenue dans le type de séance.
- ajout de la notion de tuteur à un groupe d'un projet pour éviter que plusieurs professeurs se retrouvent à observer le même groupe
- réalisation du débriefing d'une séance (et non d'une observation) avec les informations relatives à tous les groupes : groupes ayant le plus d'action etc...
- ajout de la possibilité de basculer entre le traitement des données sur la tablette avec SQLite ou sur le serveur commun au site Web avec MySQL. Ceci permettrait de pouvoir utiliser la tablette sans connexion internet si une copie de la base de données a été faite auparavant.

Au niveau du code, dans un souci de maintenabilité, nous avons pris la peine de commenter les méthodes clés et les points difficiles du code afin d'assurer une reprise facile du projet. Ainsi, les améliorations à apporter devraient pouvoir se faire rapidement.

Nous avons notamment gardé les classes définies par les groupes précédents, que nous n'utilisons pas dans l'application actuelle : la classe FilePicker, la classe Database qui traitait la base de données en local avec SQLite...

Rédacteur : RAFFLIN Corentin

Relecteur : LOPES Vincent

CONCLUSION

L'application réalisée est plus personnalisable qu'elle ne l'était les années passées. Par exemple, toutes les informations peuvent être ajoutées sur la tablette et les équipements sont associables à des événements. Le principal objectif du projet, l'observation de plusieurs groupes dans une séance, a lui aussi été atteint. Mais la grande nouveauté du projet reste le partage d'une base de données entre le site Web et la tablette. Le site Web permet de préparer entièrement les séances, ainsi que de visualiser les débriefings de celles-ci.

L'application et le site Web livrés permettent ainsi aux enseignants de préparer les séances et d'observer plusieurs groupes d'élèves de manière simple, efficace et personnalisée. Pour autant, le projet pourrait être repris pour améliorer l'application et le site Web afin de rendre l'expérience de l'utilisateur encore plus agréable et complète.

Rédacteur : RAFFLIN Corentin

Relecteur : LOPES Vincent

RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Matthieu Allingrin, Axel Durand, Cheickina Fofana, Projet 02 - Tablette d'observation pédagogique - Rapport technique, <https://redmine.telecom-bretagne.eu/projects/easyobs/files>, 07/06/2017

[2] Marie-Cécile Tregouet, Projet 303 - Tablette d'observation pédagogique - Rapport bibliographique, <https://redmine.telecom-bretagne.eu/projects/easyobs/files>, 19/11/2015

[3] Abhishek, Connecting Android App to Remote Database using PHP and MySQL <http://www.androiddeft.com/2017/12/24/connecting-android-app-remote-database-using-php-mysql/>, dernièrement consulté le 15/05/2018

[4] Parcelable, <https://developer.android.com/reference/android/os/Parcelable>, dernièrement consulté le 02/05/2018

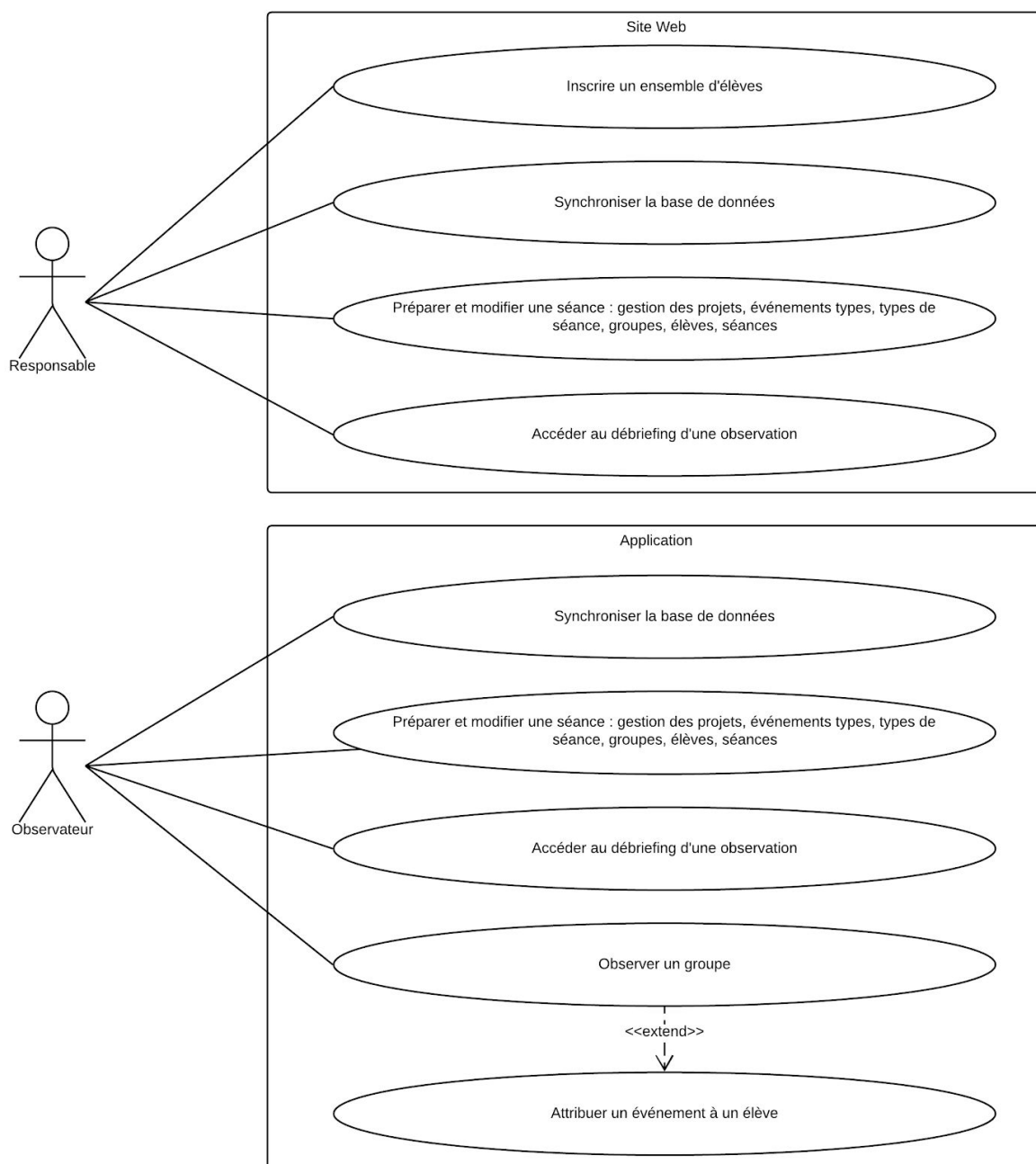
[5] Mathias Seguy, Tutoriel Android : tout comprendre sur les Threads, les Handlers, les AsyncTasks et les fuites mémoires <https://mathias-seguy.developpez.com/tutoriels/android/comprendre-thread-handler-async-task-fuites-memoires/>

GLOSSAIRE

N°	Terme	Description
1	Projet	Ensemble de séances portant sur le même thème, généralement avec les mêmes élèves. Par exemple : projet DEC, projet DEV, projet DD
2	Groupe d'élèves	Élèves travaillant ensemble et sur lesquels se concentrera l'observation de l'observateur.
3	Type de séance	Ensemble d'événements types possibles d'une séance associée à un projet. Par exemple : réunion, soutenance
4	Événement type	Nom d'un événement pouvant être réalisé
5	Événement	Activité ou action d'un élève
6	Équipement	Outil utilisable, par un élève ou non, dans une observation. Ils sont définis au nombre de quatre : ordinateur, tableau, projecteur et tuteur.
7	Séance	Période de temps pendant laquelle les groupes d'élèves sont observés. Une séance a un type, fait partie d'un projet, comprend un ou plusieurs groupes, a une date de début et de fin.
8	Observation	Période de temps pendant laquelle un seul groupe d'élèves est observé.
9	Activité	Écran spécifique de l'application
10	Layout	Objet permettant d'organiser visuellement les objets contenus (images, boutons...) et ce de différentes manières selon son type
11	Responsable	Personne qui utilisera le site Web ou la tablette pour gérer l'ajout ou le retrait de données telles que des élèves, des groupes ou des séances d'un projet... Les fonctions déclenchées par le responsable peuvent donc l'être depuis le site Web de l'application ou depuis la tablette.

12	Observateur	Personne, généralement un enseignant, qui utilisera la tablette pour observer les élèves. L'observateur peut jouer le rôle du responsable. Les fonctions déclenchées par l'observateur ne peuvent l'être que depuis la tablette.
13	Bootstrap	Bootstrap est une collection d'outils utile à la création du design de sites et d'applications web.
14	Fragment	Sous-activité ou fragment d'écran (par opposition à une activité, qui utilise nécessairement l'intégralité de l'écran).

ANNEXES



Annexe 1 : Diagramme des cas d'utilisations

Activité	Description
HomeActivity	Liée à l'écran d'accueil permettant d'accéder à PreparationActivity, ChooseObservationActivity et DebriefingManagementActivity
PreparationActivity	Liée à un sous-menu permettant d'accéder à toutes les activités de préparation d'une séance
ProjectActivity	Liée à la gestion des projets
EventsActivity	Liée à la gestion des événements types
MeetingTypeActivity	Liée à la gestion des types de séance
StudentsListActivity	Liée à la gestion des élèves
GroupActivity	Liée à la gestion des groupes
CreationMeetingActivity	Liée à la gestion des séances
ChooseObservationActivity	Liée à la gestion des séances et permettant l'accès à une observation
ObservationActivity	Liée à une observation
DebriefingMeetingActivity	Liée à un sous-menu permettant d'accéder au débriefing d'une observation
DebriefingObservation	Liée au débriefing d'une observation
StatisticsActivity	Liée aux statistiques d'une observation

Annexe 2 : Tableau des activités avec leur description

Activité	Layout
HomeActivity	home_activity
PreparationActivity	preparation_activity
ProjectActivity	project_activity
EventsActivity	event_type_activity
MeetingTypeActivity	meeting_type_activity
StudentsListActivity	students_list_activity
GroupActivity	group_activity
CreationMeetingActivity	creation_meeting_activity
ChooseObservationActivity	choose_observation_activity
ObservationActivity	observation_activity
DebriefingMeetingActivity	debriefing_meeting_activity
DebriefingObservation	debriefing_observation
StatisticsActivity	statistics_activity

Annexe 3 : Tableau de correspondance des activités et des layouts

PLANNING ET ANALYSE DES ÉCARTS



Annexe 4 : Planning initial comparé avec le planning réel

Le planning proposé au départ était idéaliste. Nous savions dès le départ que nous ne finirions pas aussi tôt que planifié car n'ayant jamais réalisé d'application Android ou de site Web, la durée des tâches était difficile à prévoir. D'une part, nous pensions que les tâches pouvaient être réalisées en parallèle. Or, certaines découlaient d'autres tâches. Par exemple, la partie du codage des débriefings ne pouvaient être faite qu'après avoir codé la partie observation. D'autre part, la durée des tâches n'a pas été proprement établie. Les premières tâches ont pris plus de temps que prévu, le temps d'adopter certains réflexes de programmation. En particulier les premières fonctionnalités du site Web ont pris du temps à être implémentées, aucun d'entre nous n'ayant jamais codé en HTML ou en PHP auparavant. Par la suite, une fois ces réflexes acquis, les tâches prirent moins de temps à réaliser que prévu.

En ce qui concerne la répartition des tâches entre nous, nous pensions faire chacun un peu de tout, mais il s'est avéré que chacun d'entre nous s'est spécialisé dans un domaine.

Rédacteur : LOPES Vincent

Relecteur : RAFFLIN Corentin