



L'Université POLITEHNICA de Bucarest
Faculté d'Ingénierie en Langues Etrangères



Rapport de Documentation

STUDNOTE

Dispositifs et applications pour les plateformes mobiles

Auteurs: GRECULESCU Ana-Maria
HOTĂSCU Oana-Andreea
RADU Andreea-Nicoleta
STOIAN Lavinia-Georgeta
URSE Mirela-Cristina

Professeur coordinateur: Dragoș STOICA

- Janvier 2015 -

SOMMAIRE

- I. Informations Générales
- II. Structure de l'application
- III. Fonctionnalités

Informations générales

StudNote est une application Android qui permet aux étudiants de gérer les informations concernant les cours et l'emploi de temps. Nous sommes étudiantes dans la quatrième année à l'Université « Politehnica » de Bucarest. Chaque jour on a des problèmes à compléter nos devoirs d'une manière efficace et faire notre temps compter. Alors, on a fait cette application au but de mieux organiser notre agenda, pour avoir une vision plus claire sur nos devoirs quotidiens.

Android est un système d'exploitation mobile pour Smartphones, tablettes tactiles, PDA et terminaux mobiles. C'est un système open source utilisant le noyau Linux. Il a été lancé par une startup du même nom rachetée par Google en 2005.

Android comporte une machine virtuelle nommée Dalvik, qui permet d'exécuter des programmes prévus pour la plate-forme Java. C'est une machine virtuelle conçue dès le départ pour les appareils mobiles et leurs ressources réduites - peu de puissance de calcul et peu de mémoire. En effet les appareils mobiles contemporains de 2011 ont la puissance de calcul d'un ordinateur personnel vieux de dix ans. La majorité, voire la totalité des applications est exécutée par la machine virtuelle Dalvik.

Le bytecode de Dalvik est différent de celui de la machine virtuelle Java de Oracle (JVM), et le processus de construction d'une application est différent : le code source de l'application, en langage Java est tout d'abord compilé avec un compilateur standard qui produit du bytecode pour JVM (bytecode standard de la plateforme Java) puis ce dernier est traduit en bytecode pour Dalvik par un programme inclus dans Android, du bytecode qui pourra alors être exécuté.

L'ensemble de la bibliothèque standard de Android ressemble à J2SE (Java Standard Edition) de la plateforme Java. La principale différence est que les

bibliothèques d'interface graphique AWT et Swing sont remplacées par des bibliothèques d'Android.

Le développement d'applications pour Android s'effectue avec un ordinateur personnel sous Mac OS, Windows ou Linux en utilisant le JDK de la plate-forme Java et des outils pour Android. Des outils qui permettent de manipuler le téléphone ou la tablette, de la simuler par une machine virtuelle, de créer des fichiers APK (les fichiers de paquet d'Android), de déboguer les applications et d'y ajouter une signature numérique. Ces outils sont mis à disposition sous la forme d'un plugin pour l'environnement de développement.

La bibliothèque d'Android permet la création d'interfaces graphiques selon un procédé similaire aux frameworks de quatrième génération que sont XUL, JavaFX ou Silverlight: l'interface graphique peut être construite par déclaration et peut être utilisée avec plusieurs skins - chartes graphiques. La programmation consiste à déclarer la composition de l'interface dans des fichiers XML ; la description peut comporter des ressources (des textes et des pictogrammes). Ces déclarations sont ensuite transformées en objets tels que des fenêtres et des boutons, qui peuvent être manipulés par de la programmation Java. Les écrans ou les fenêtres (activités dans le jargon d'Android), sont remplis de plusieurs vues ; chaque vue étant une pièce d'interface graphique (bouton, liste, case à cocher...). Android 3.0, destiné aux tablettes, introduit la notion de fragments: des panneaux contenant plusieurs éléments visuels. Une tablette ayant - contrairement à un téléphone - généralement suffisamment de place à l'écran pour plusieurs panneaux.

La structure de l'application

L'application comprend une interface principale qui permet à l'utilisateur de sélectionner une des fonctionnalités disponibles. Cette interface contient un espace qui affiche à l'écran les événements du jour courant, un assistant (par exemple l'animal préféré) et des boutons correspondant aux éléments du menu principal de l'application: emplois de temps, calendrier, cours, notes et configurations. En cliquant ces boutons, l'utilisateur peut accéder à des interfaces graphiques nouvelles.

Le bouton *Calendrier* permet l'affichage d'une interface graphique qui contient un calendrier. Pour chaque jour du mois sont affichés les événements introduit pour chaque cours, par exemple un examen. Les événements du calendrier sont synchronisés automatiquement avec les autres fonctionnalités de l'application. Le bouton *Emploi de temps* affiche une liste avec les jours de la semaine et pour chaque jour l'utilisateur peut visualiser les cours, les horaires et les salles.

Le bouton *Cours* affiche une liste avec tous les cours introduits par l'utilisateur dans l'application. Si l'utilisateur clique sur un des cours, il peut regarder les détails de ce cours dans une nouvelle interface graphique. Les cours sont définis par plusieurs champs optionnels, comme: nom du cours, professeurs coordonateurs, contact des professeurs, salle, critères de notation, bibliographie, ou d'autres détails supplémentaires. L'utilisateur peut éditer chacun de ces champs ou peut ajouter un nouveau champ défini par lui-même.

Le bouton *Notes* fait apparaître à l'écran un tableau éditible avec les notes correspondant à chaque cours. L'application calcule automatiquement les moyennes finales en fonction de notes et coefficients introduits par l'utilisateur.

Le bouton *Configurations* affiche des fonctionnalités supplémentaires ou des possibles configurations de l'application telles que créer des notifications, partager en ligne les détails des cours, aide, des détails sur le fonctionnement de l'application et contact.

Fonctionnalités

L'application contient une activité principale qui permet l'affichage des événements du jour courant. Ces événements vont être enregistrés dans un objet de type TextView ou dans une ListView avec plusieurs items.

Ensuite, l'activité principale contient un menu avec plusieurs boutons : emplois de temps, calendrier, cours et configurations. Si l'utilisateur clique un de ces boutons, une nouvelle interface graphique va apparaître à l'écran. On va définir le layout pour chaque interface dans le fichier .xml correspondant et pour chaque layout on va créer une classe .java qui étend la classe Fragment.

L'activité principale va étendre la classe FragmentActivity pour avoir accès aux autres interfaces et pour ajouter dynamisme à l'application. Pour chaque bouton du menu on va définir les méthodes setOnClickListener et onClick qui vont permettre le changement de l'interface quand le bouton est cliqué. Le bouton Calendrier permet l'affichage d'une interface graphique qui contient un calendrier.

Pour définir le calendrier on va utiliser le Calendar Provider fourni par Android. Les événements du calendrier seront synchronisés automatiquement avec les autres fonctionnalités de l'application et il sera possible pour l'utilisateur de fixer des alarmes et l'application va lui envoyer des notifications quand un événement du calendrier a lieu. Le bouton Emploi de temps affiche une liste avec les jours de la semaine et pour chaque jour l'utilisateur peut visualiser les cours, les horaires et les salles. On va utiliser un objet de type ExpandableListView qui affiche sur le premier niveau de items les jours de la semaine et sur le deuxième niveau des items contenant des informations concernant les cours : matière, horaire, salle.

L'emploi de temps peut être édité par l'utilisateur (avec DatePicker et TimePicker). Le bouton Cours affiche une interface contenant un ListView avec tous les cours introduit par l'utilisateur dans l'application. Il est possible d'ajouter un nouvel item ou d'enlever un item de la liste. Si l'utilisateur clique un des cours (un item de la liste), il peut regarder les détails de ce cours dans une nouvelle interface graphique. Cette interface contient un TextView avec les informations sur les cours et un bouton pour éditer les cours. Si l'utilisateur veut éditer les informations sur les cours, d'abord il doit compléter les éléments de type EditText

correspondant aux champs obligatoires (nom du cours, professeurs coordonateurs, salle, heure).

Pour ajouter des champs supplémentaires, il peut cliquer le bouton Ajouter qui va faire apparaître une ListView avec des champs optionnels. L'item choisit par l'utilisateur va être ajouté aux informations du cours comme un EditText. Le bouton Configurations affiche une ListView(ou des boutons) avec des fonctionnalités supplémentaires ou des possibles configurations de l'application telles que créer des notifications, transmettre les détails des cours par Gmail ou par messages texte, aide, détails sur le fonctionnement de l'application et contact.

Pour chacune de ces fonctionnalités une nouvelle interface graphique va s'afficher à l'écran. Par exemple les interfaces pour aide, détails sur le fonctionnement de l'application et contact vont contenir des TextView pour afficher ces information. Pour transmettre les détails des cours par Gmail, l'implémentation est plus complexe parce qu'il faut implémenter l'authentification avec un compte Google.