Project Assignment

IV Semester Student Project – Paradigms of Programming – II

Project on Android platform

‘The College App’ an Android application

Prepared by Shravan Karthik- 12IT77
Siddharth P Ramakrishnan -12IT79
Rishab Ketan Doshi -12IT59
R Gokul Shanth-12IT52
Rohit John Joseph-12IT61
Jashan Shetty -12IT26

March/April-2014

Department of Information Technology
National Institute of Technology, Karnataka

Table of Contents

Version History

Version	Date	Comments
1.0	March-7-2014	Problem Statement
1.1	March-20-2014	Additional features: Reminder application

Problem Statement

An Android application needs to developed that caters to the needs of students. The app should provide solutions to a student's common requirements,

Timetable-

Organize your daily college schedule with the help of the Timetable.
Create and customize according to your specific subjects for the semester.

Bunk Manager-

Keeps you in track with the number of classes you have missed and warns you if you're close to going below your attendance requirements for that course.

Expense Manager-

Manage your expenses directly on your smart phone. The manager will be used by students to check the amount of money spent daily, weekly and monthly with the help of graphs.
The application also provides a widget for the expense manager to make it easier for the user to interact with it.

Memos-

Quickly create, access and organize notes. This proves extremely useful when you need to write things down on the go.

Reminder-

It helps in reminding yourself with different important things which you don’t want to forget. This feature has convenient messages which pop up on your notification bar as and when instructed.

How an Android App Works?

Android applications are written in the Java programming language. The Android SDK tools compile your code—along with any data and resource files—into an APK: an Android package, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

Once installed on a device, each Android app lives in its own security sandbox:

- The Android operating system is a multi-user Linux system in which each app is a different user.
- By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.
- Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.
- By default, every app runs in its own Linux process. Android starts the process when any of the app's components need to be executed, then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

In this way, the Android system implements the principle of least privilege. That is, each app, by default, has access only to the components that it requires to do its work and no more. This creates a very secure environment in which an app cannot access parts of the system for which it is not given permission.

However, there are ways for an app to share data with other apps and for an app to access system services:

- It's possible to arrange for two apps to share the same Linux user ID, in which case they are able to access each other's files. To conserve system resources, apps with the same user ID can also arrange to run in the same Linux process and share the same VM (the apps must also be signed with the same certificate).
- An app can request permission to access device data such as the user's contacts, SMS messages, the mountable storage (SD card), camera, Bluetooth, and more. All app permissions must be granted by the user at install time.

Why Android?

Although iPhone with the iOS took a big share from RIM's Blackberry in 2010, Google, Apple and RIM are all vying for the top spot in the battle for smartphone supremacy.

In the latest market share data released by the Nielsen company, the Android OS has toppled the RIM BlackBerry and Apple iOS as it reached 29 percent market share compared to RIM BlackBerry and iOS which are tied at 27 percent.

But the data also says that one reason why RIM and Apple continue to do well is because as they are the manufacturers of their own operating system for their own devices. The updates and fixes are faster than those on the Android OS platform, as the latter is spread among different manufacturers. But the Android platform is on some of the most advanced devices, such as Motorola's Atrix 4G, which is as powerful as a small computer.

Here is a look at five reasons why Android-powered devices might fare better than others.

Hardware: Android smartphones are bigger and have better design. The Samsung Galaxy S2 and Motorola Atrix 4G are devices that are going define the future with their computer-equivalent specifications. They have better cameras, different display sizes, and more memory for mutitasking.

Accessibility: The Android powered smartphones are available from several different devices manufacturers like HTC, Motorola and Samsung. They are also available with a choice of carriers and different plans to suit different users' needs.

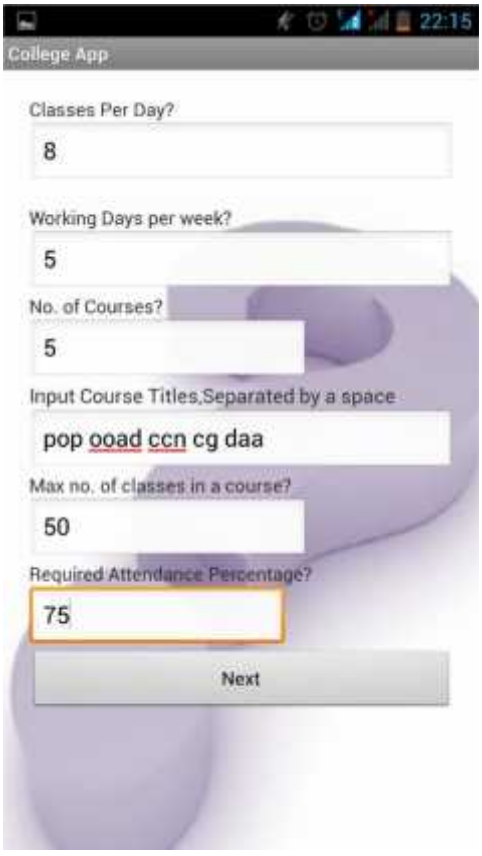
Pricing: The price factor is important. Most users tend to change their smartphone as it ages, and the availability of cheaper and more sophisticated Android phones might make it more viable than others, as Apple maintains high prices for the iPhone.

Apps: The applications store for Android smartphones is growing and more advanced and better applications are filling that marketplace. There is reasonable cross-compatibility across handsets.

Speed: How fast data can be accessed is an important factor. The Android smartphones can access the newer, faster 4G networks, while iPhones and Blackberrys cannot

The Study

Main Menu



The *splash page* on the left is the image that appears just as the android application is opened. It remains for 1.5s and is used to give time for the background applications to start running.

The main menu page is made to input details of the user, into the application. It appears only once, when the application is first opened. These details are necessary for the Time Table and Bunk manager pages which are discussed later on.

The user has to input the following details into the text boxes

- Classes per day (a maximum of 8)
- Working days per week (either 5 or 6)
- Total number of courses
- The names of the course titles
- Max number of classes in that course (max input of 60)
- Required Attendance percentage

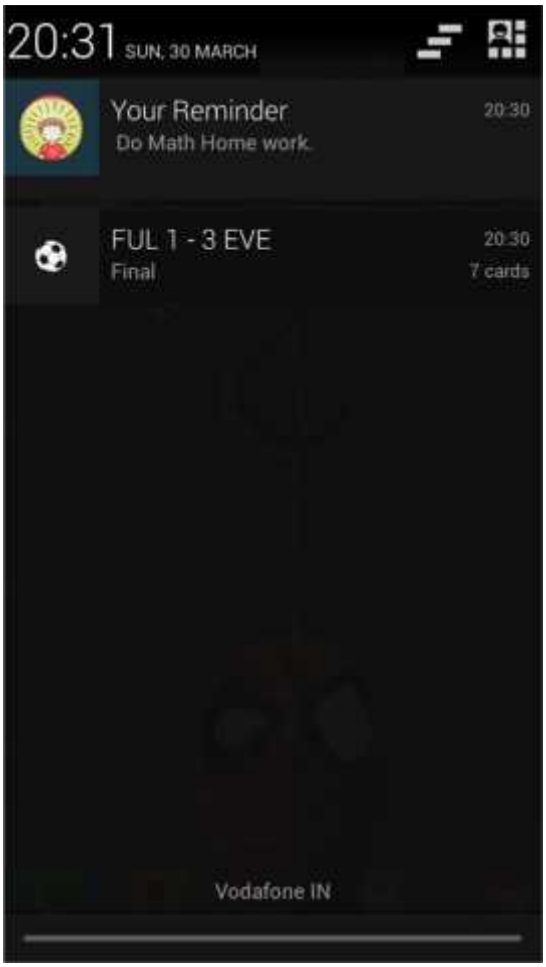
Menu screen

This is the initial page once the application is started and running. It serves as a menu for the user to choose a desired function. The menu screen of the *college app* consists of 5 buttons.
One button for each of the features present in the app:

- Expense Manager
- Timetable
- Memos
- Bunk Manager
- Reminder

A single tap of any of the above icons will open the respective feature and take you to the next screen.

Reminder page

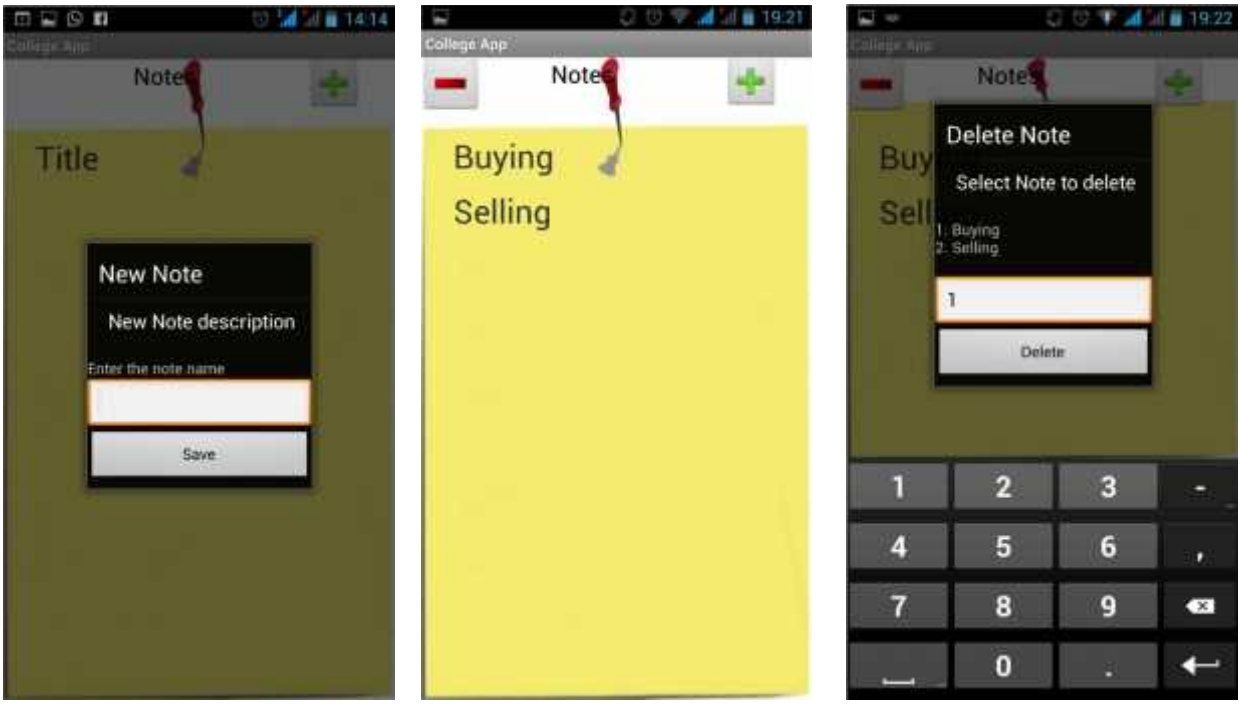


The Reminder application firstly starts off with a text box, which is used to give a title to your reminder. The user has to type in the message in this top space.
Next the date and time of the event must be registered. The buttons when pressed opens a drop down panel which you can scroll through to find the correct date and time.
The date is in the DD/MM/YY format and the time in the HH/MM format as shown above. Finally the bottom *CREATE REMINDER* button will finish the process.

The *VIEW REMINDERS* button stores all the previous reminders along with their date stamp for future reference.

The Reminder Message will pop up at the top of the screen in the notification bar at the specified time on the specified date. It also includes the 'Big View' Notification display. A notification in normal view appears in an area that's up to 64 dp tall, whereas the big view can expand beyond with a single swipe of your finger.

Memo Page



Clicking on the memo icon will take you this page above. This memo application is made to look like any ordinary piece of paper pinned down by a thumbtack. It's an easy way to jot down any important points which you'll need for further use.

The user can type in a title for the note on the top and continue with the details of the memo in the pop up for the New Note.

Once saved, the user can create more and more memos by simply clicking on the green plus sign on the top of the screen. The minus sign on the left is used to remove a note that has already been created. It appears only after the first note is created and a pop-up gives you options to select the note you want to delete along with a numpad.

Timetable Page

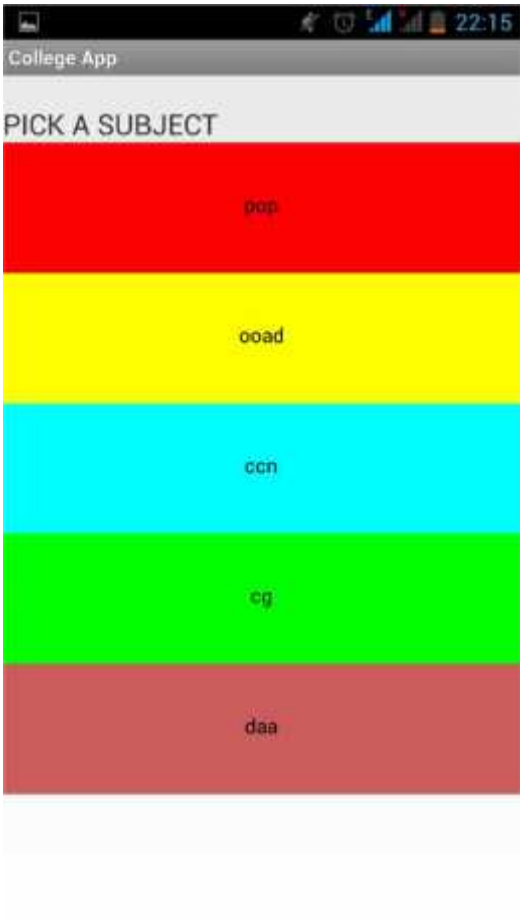
In the timetable feature the user is meant to input the details once and then use the timetable for further use.

The timetable page is made after initially asking the user whether they have a 5-day or 6-day college week. The number of classes in a day is also inputted along with the time intervals. After which the layout for the timetable is formed.
Then one by one the classes are to be added. This is a one time process.

After filling in all of the details, the user hits the confirm button to store the timetable. The timetable is saved and can be checked whenever necessary. It will be color coded according to the subjects to simplify searching for the user.

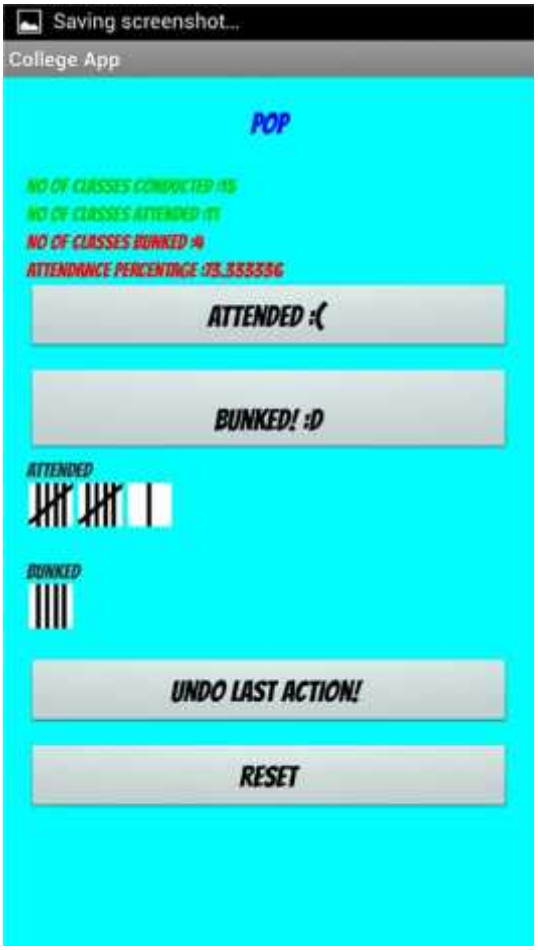
The edit button on the bottom of the screen allows the user to change the timetable if any adjustments are needed. The save button saves the changes made.

Bunk Manager page



The bunk manager opens up once the “Lets Party” icon is tapped on. It opens up to an initial screen including all the subjects of the students. The student simply has to click on the subject which is currently taking place.

Once the subject is selected, the next page that opens is the one that calculates your attendance percentage. If the user is attending the class, he taps on the *Attended* button. This draws a single tally mark under the attended column. The same thing happens if the user is missing a class.

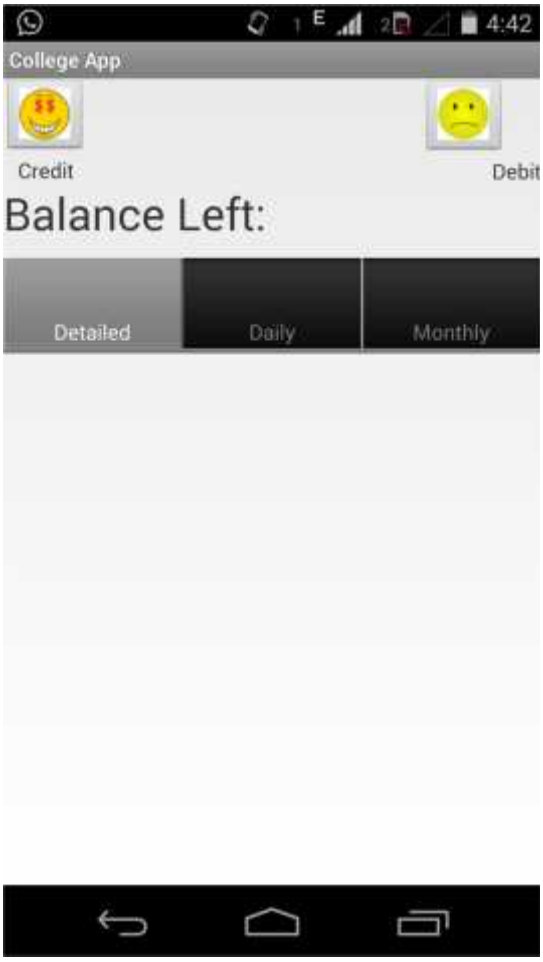


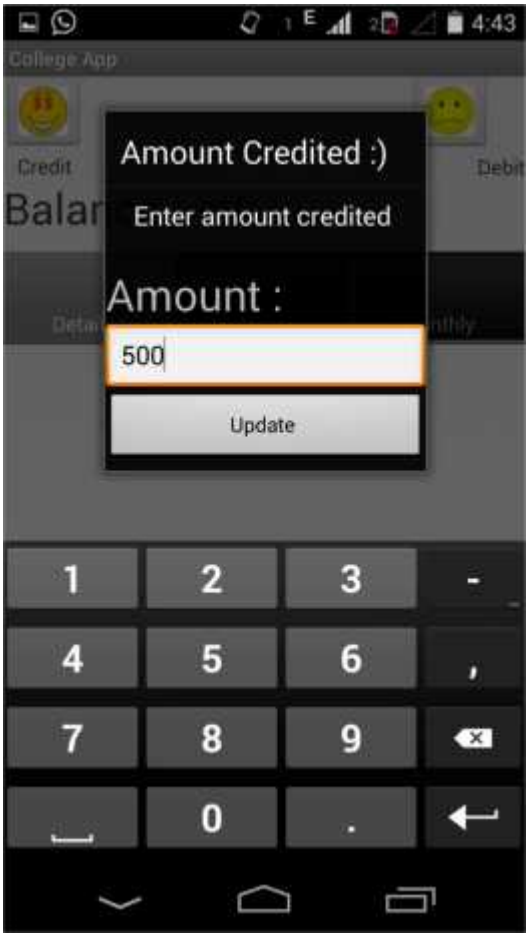
In the bunk manager, the data is presented on the top of the screen. Also the attendance percentage is automatically calculated as and when the classes are attended/bunked.

This screen also includes an undo and reset button just in case you make a mistake or decide to attend class in the last moment.

The last screenshot above shows the warning pop up that opens up when the student's attendance percentage goes lower than the minimum requirement.

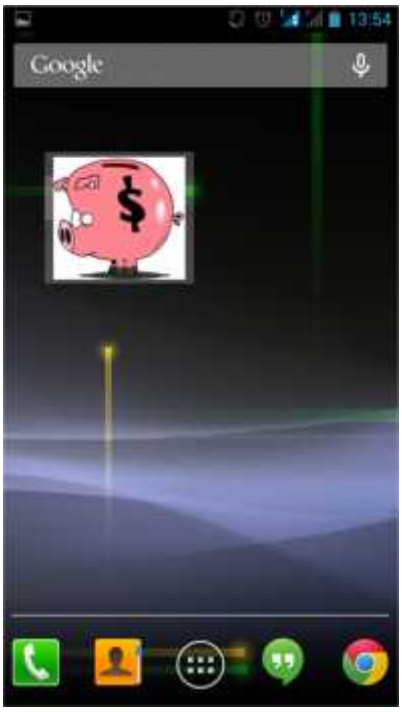
Expense manager page





The expense manager page contains two buttons on the top. One *dollar eyed face* for crediting money on the top left corner and another for debiting money on the top right.

First it asks for the initial amount of money in your account which you have to punch in. After that the user can update their expenses by filling in the pop-ups with the given number pads along with the reason for you spending your cash.
The application will keep track of your expenses and make sure that you don't overspend.



Another feature of this expense manager is the timely graphs showing the different ways you have spent your money.
There exist three columns:

- A detailed expense for that specific day
- Expenses with graphs per day
- A monthly expense

The graph has *the reason for expense* on the x-axis and the *amount* used on the y-axis.

Another advantage in the expense manager is the **widget.** In Android, the word widget is a generic term for a bit of self-contained code that displays a program, or a piece of a program, that is also (usually) a shortcut to a larger application. A really quick and convenient way to get to this application straight from the home screen.

The widget is an image of piggy-bank and once clicked opens up the next page where you can update the expenses. The amount and description is to be entered and will be directly debited from your account. This is an extremely useful utility when the user is on the go.

Conclusion

The College App is developed to be an innovative application and its main purpose is to get familiar with Android SDK and its development in general. The android OS has many advantages, listed earlier, and thus is why we picked this over others.

We primarily chose Android platform as it is open-source and well documented, and the most commonly used mobile operating system among college students.

The advantage of declaring your User Interface in XML is that it enables you to better separate the presentation of your application from the code that controls its behavior. UI descriptions are external to your application code, which means that you can modify or adapt it without having to modify your source code and recompile. For example, you can create XML layouts for different screen orientations, different device screen sizes, and different languages. Additionally, declaring the layout in XML makes it easier to visualize the structure of your UI, so it's easier to debug problems. This feature of Android is used extensively in the development of our application.

Moving forward, we hope to push the user specific data on to a server. We also plan to push to notifications, tasks such as updating attendance records, wherein the user is notified after the class using the system time and date. Another functionality we would like to include is to sync expenses to a bank account thereby automatically updating available balance when an amount is credited or debited. The application should be up on the Android playstore soon once further testing and improvements have been made.

We would like to thank Mr. Ram Shasty, for having provided us with an opportunity to work on a completely new platform, thereby broadening our horizons.

References

http://developer.android.com/training/index.html

http://www.tutorialspoint.com/android/

http://thenewboston.org/

Android in Action Edition 3, Frank Ableson and Robi Sen