

ANDRIOD APPLICATION: FOR MONITORING AVAILABILITY OF FACULTY

¹K. Rangaswamy, ²T.Sravani, ³Nagalakshmi, ⁴V.Kiran Kumar

¹Assistant Professor, ^{2,3,4}Final B.Tech Students
Department of Computer Science & Engineering,
VBIT-Proddatur

ABSTRACT: As all mobiles are using android operating system now. At present, Head of Department is manually monitoring the staff. The main problem with that is wastage of time for checking the presence of staff in their respective classes. So we proposed a system for implementing the idea on staff monitoring. Here HOD can view the staff names who are attending the classes. This can be done only if faculty is using this application before entering to classes. In the view of HOD, the faculties who are in class are highlighted by some indication. The HOD only have some special permissions to modify the faculty data.HOD will provide the logins to all the staff under him. The data entered by faculty will be stored in database and is retrieved by HOD when he checks. If any faculties are going to put leave, the notification can be sent in this app without interacting with HOD.

INTRODUCTION

Now a day, everyone is using android devices. Not only in mobiles, are some other devices also using this technology. Android is going to rule the world in future. By the usage of apps the work will be done in smart way and in less time. Even by using this, we can access from anywhere. They are fulfilling the almost all the user requirements today.

As android have some special features. It is intent based operating system. The back button is a killer feature and is way more than just a physical button. The android operating system is essentially stack based. It is open and its applications are free. Android gives us a world-class platform for creating apps.Android is continuo sly pushing the boundaries of hardware and software forward to bring new capabilities to Users and developers. It automatically adapts our user interface to look its best on each device. We can create virtual devices that emulate any hardware configuration by this android.

EXISTING SYSTEM

- ▶ To check whether faculty are going to classes or not, HOD should physically go and have the confirmation of their presence at class.
- ▶ Presence of faculty should be noted in records manually.
- ▶ There are many ways for monitoring the staff like writing down all their timetables and checking, by making a call to interact.
- ▶ Again their records should enter into a system for future purpose.
- ▶ All this are delay processes.

PROPOSED SYSTEM

- ▶ We are proposing the idea for implementing the staff monitoring.
- ▶ Here HOD can view the names of faculty who are in classes.
- ▶ For that faculty should use the app before going to class.
- ▶ The names of faculty who are in class are highlighted.
- ▶ HOD can also view updates i.e., notifications of faculty if any classes are adjusted.
- ▶ HOD can only register the faculty.
- ▶ After logging into faculty page they must insert some details and that are stored in database.
- ▶ ID is given to every faculty and by that HOD can retrieve their details as that ID act as unique key.
- ▶ Database contains two tables' i.e., staff & tracking.

MODULES

According to our implementation, this application is mainly used by 2 users and app divided into 2 modules regarding to their respective functionality. They are

- HOD
- Staff

HOD

HOD will have to register first to login to our app. After login, HOD will get the page containing two buttons named check, update. If HOD clicks check button, it shows all faculty names. It automatically highlights the faculty who are in classrooms. This automatic indication is

done only if faculty uses this app before entering into classroom. HOD will only can provide logins to all the staff under him. In this way, HOD can check out faculty who are in classrooms. HOD can also view updates i.e., any notifications in app sent by faculty.

STAFF

All the staff should login with their respective login credentials which were provided by HOD. After login to the faculty page, there will be form in which faculty should enter their attending class name, hour number, section name and also contain two buttons named start & stop. All these details are stored in database sequentially. Faculty should click start button before going to class and their names should be highlighted. After completion of class they should click the stop button. After clicking stop button their names should view in the normal mode as without any highlights.

TECHNOLOGIES USED

WHAT IS ANDROID?

Android is also an operating system developed by Google. Basically it was started by some other company which was taken by Google. Google improved the operating system and made it an open source platform. It was widely adapted over the world. As it is open source it is so popular amongst the smart phones. Android OS can also be used on tablet PCs. One of the most widely used mobile OS these days is ANDROID. Android does a Software Bunch comprise not only operating system but also middleware and key applications. Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android.

History of Android

Android, Inc. was founded in Palo Alto, California in October 2003 by Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc.), Nick Sears (once VP at T-Mobile), and Chris White (headed design and interface development at WebTV) to develop, in Rubin's words "smarter mobile devices that are more aware of its owner's location and preferences".

Android, This word means a lot in present High-Tech World. Today Smartphone are known for its operating system which is Android. Android is a software stack for mobile devices that include on O.S and middleware and key Applications. The goal of Android open source project is to create a successful real world product that improves the mobile experiences for end-users. And one that can run any application written by third-parity developers using Android using Android SDK.

Versions of Android

Version	Code name	Release date	API level	Distribution (February 5, 2013)
4.2	Jelly Bean	November 13, 2012	17	1.4%
4.1.x	Jelly Bean	July 9, 2012	16	12.2%
4.0.x	Ice Cream Sandwich	December 16, 2011	15	29.0%
3.2	Honeycomb	July 15, 2011	13	1.0%
3.1	Honeycomb	May 10, 2011	12	0.3%
2.3.3–2.3.7	Gingerbread	February 9, 2011	10	45.4%
2.3–2.3.2	Gingerbread	December 6, 2010	9	0.2%
2.2	Froyo	May 20, 2010	8	8.1%
2.0–2.1	Éclair	October 26, 2009	7	2.2%
1.6	Donut	September 15, 2009	4	0.2%

ANDROID ARCHITECTURE

Android Architecture

The following diagram shows the major components of the Android operating system. Each section is described in more detail below.

Applications

Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language.

Application Framework

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

- A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser.
- Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data.
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.
- A Notification Manager that enables all applications to display custom alerts in the status bar.

Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

System C library: BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.

Media Libraries: Based on Packet Video's Open CORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.

Surface Manager: Manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.

LibWebCore: A modern web browser engine which powers both the Android browser and an embeddable web view.

SGL: the underlying 2D graphics engine.

3D libraries: an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software pasteurizer.

Free Type - bitmap and vector font rendering.

SQLite - a powerful and lightweight relational database engine available to all applications.

Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Underlying all applications is a set of services and systems, including:

A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser.

Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data. A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.

A Notification Manager that enables all applications to display custom alerts in the status bar. An Activity Manager that manages the lifecycle of applications and provides a common navigation back stack.

Using Databases:

Android provides full support for SQLite databases. Any databases you create will be accessible by name to any class in the application, but not outside the application. The recommended method to create a new SQLite database is to create a subclass of SQLiteOpenHelper and override the onCreate () method, in which you can execute a SQLite command to create tables in the database.

Database debugging:

The Android SDK includes a sqlite3 database tool that allows you to browse table contents, run SQL commands, and perform other useful functions SQLitedatabases. See Examininsqlite3 databases from a remote shell to learn how to run this tool.

Using a Network Connection:

You can use the network (when it's available) to store and retrieve data on your own web-based services. To do network operations, use classes in the following packages:

- java.net.*
- android.net.*

SERVLETS

- Here focus on both sides of a client-server relationship.
- The client requests that some action be performed and the server performs the action and responds to the client.
- This request-response model of communication is the foundation for the highest-level view of networking in Java—servlets.
- A servlet extends the functionality of a server. The **javax.servlet** package and the **javax.servlet.http** package provide the classes and interfaces to define servlets.

Servlet technology today is primarily designed for use with the HTTP protocol of the Web, but servlets are being developed for other technologies. Servlets are effective for developing Web-based solutions that help provide secure access to a Web site, that interact with databases on behalf of a client, ... that dynamically generate custom HTML documents to be displayed by browsers and that maintain unique session information for each client.

Methods of Interface Servlet

- void init(ServletConfig config).
- This method is automatically called once during a servlet's execution cycle to initialize the servlet. The ServletConfig argument is supplied automatically by the server that executes the servlet.
- ServletConfig getServletConfig().
- This method returns a reference to an object that implements interface **ServletConfig**. This object provides access to the servlet's configuration information such as initialization parameters and the servlet's ServletContext, which provides the servlet with access to its environment (i.e., the server in which the servlet is executing).

- void service(ServletRequest request, ServletResponse response).
- This is the first method called on every servlet to respond to a client request.
- String getServletInfo().
- This method is defined by a servlet programmer to return a **String** containing servlet information such as the servlet's author and version.
- void destroy().
- This "cleanup" method is called when a servlet is terminated by the server on which it is executing.

HttpServlet Class

- Web-based servlets typically extend the class HttpServlet.
- This class HttpServlet overrides the method **service** to distinguish between the typical requests received from a client Web browser.
- The two most common HTTP request types (also known as request methods) are *get* and *post*.
- A **get** request *gets* (or retrieves) information from the server.
- Common uses of get requests are to retrieve an HTML document or an image.
- A post request *posts* (or sends) data to the server.
- Common uses of post requests are to send the to Web-server, information from an HTML form in which the client enters data, to send to the server, information so that it can search on the Internet, or query a database for the client, to send authentication information to the server, etc.
- The class HttpServlet defines the methods *doGet* and *doPost*, which respond to get and post request.

HttpServletRequest Interface:

Important methods of interface HttpServletRequest Method Descriptions:

- Every call to doGet or doPost for an HttpServlet receives an object that implements interface HttpServletRequest.
- The Web server that executes the servlet creates an HttpServletRequest object and passes this to the servlet's service method (which, in turn, passes it to doGet or doPost).
- This object contains the request from the client. A variety of methods are provided to enable the servlet to process the client's request. Some of these methods are from interface **ServletRequest**—the interface that

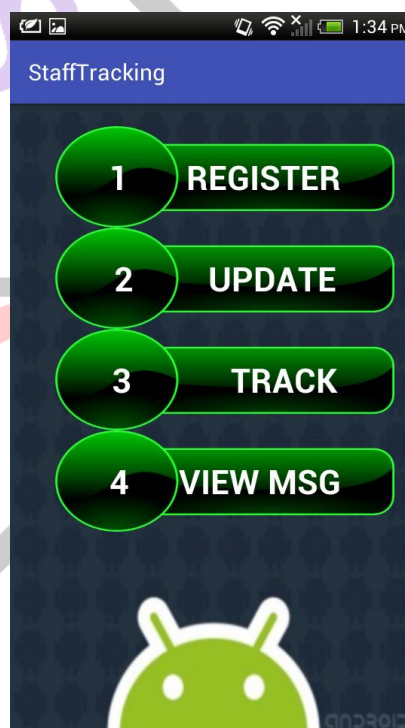
HttpServletRequest extends.uests from a client,respectively.

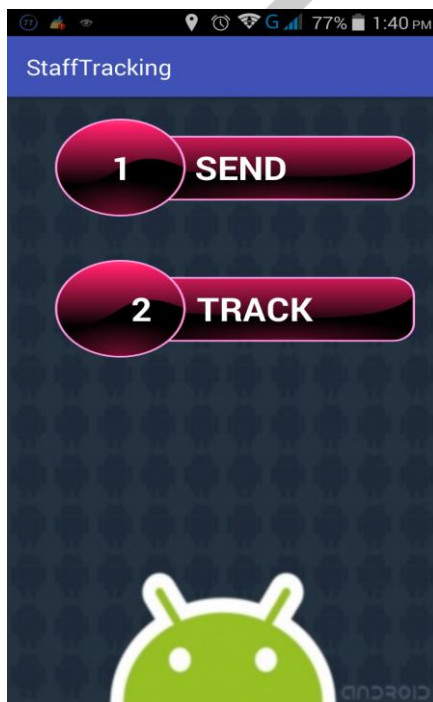
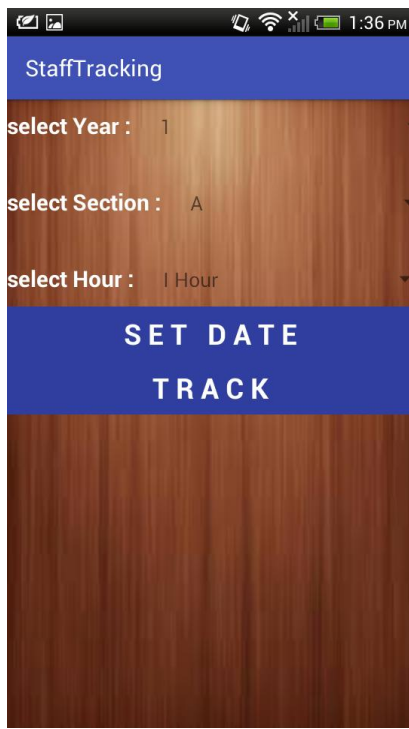
HttpServletRequest Interface:

Important methods of interface HttpServletResponse Method Descriptions:

- Every call to doGet or doPost for an HttpServlet receives an object that implements interface HttpServletResponse.
- The Web server that executes the servlet creates an HttpServletResponse object and passes this to the servlet's service method (which, in turn, passes it to doGet or doPost).
- This object contains the response to the client. A variety of methods are provided to enable the servlet to formulate the response to the client. Some of these methods are from interface **ServletResponse**—the interface that HttpServletResponse extends.

Results:





[3]. P. SZAKACS-SIMON, S. A. Moraru, L. Perniu, "Pulse OXI meter-Based monitoring system for people at risk", CINTI IEEE Conference, Budapest, Hungary.

[4]. J. Steele, N. To, "The Android Developers Cookbook: Building Applications with de Android SDK", Addison-Wesley Professional, First edition.

[5]. E. Ozdalga, A. Ozdalga, N. Ahuja, "The Smartphone in Medicine: A Review of Current and Potential Use Among Physicians and Students", J Med Internet Res.

[6]. E. Alepis, C. Lambrinidis, "M-health: supporting diagnosis and electronic health records", Springerplus

Authors Bibliography

T.Sravani, Pursuing Final Year B.Tech in the stream of Computer Science and Engineering in VBIT College, proddatur, KADAPA (distic).

Nagalakshmi, Pursuing Final Year B.Tech in the stream of Computer Science and Engineering in VBIT College, proddatur, KADAPA (distic)

V.Kiran kumar, Pursuing Final Year B.Tech in the stream of Computer Science and Engineering in VBIT College, proddatur, KADAPA (distic).

K.Rangaswamy is currently an Assistant Professor of Computer Science and Engineering at Bharathi Educational Society Proddatur, Andrapradesh. His most focus on networks. His published the papers in various journals. He received M.Tech degree in Computer Science at Bharath University, Chennai in 2011, his B.Tech degree in Computer Science and Engineering from JNTUA Anapataramu in 2009.

References:

[1]. MIT APP INVENTOR 2 BY David Wolber, Hal Abelson, Ellen Spertus & Liz Looney.

[2]. R. MEYER, "Professional Android 2 Application Development", Wrox, First Edition.