Volume : 2 | Issue : 11 | November 2013 • ISSN No 2277 - 8179

Research Paper

Engineering

KEYWORDS :

Automatic management of student and faculty information through mobile phone

Dr.Shantakumar.B.Patil	Professor &HOD, CSE, Nagarjuna College of Engineering & Technology
Dr.premjyoti.G.Patil	Professor,CSE, Nagarjuna College of Engineering & Technology
Sudheendra.K	Students of CSE, Nagarjuna College of Engineering & Technology
Prathamesh V	Students of CSE, Nagarjuna College of Engineering & Technology

ABSTRACT

This is a system which helps both the lecturers and parents of students in an educational institution. In the present system all the activities are done manually. For example the marks, fees structure, fee paid, attendance etc are entered into the database manually. When there are any recent activities that also updated in a time consuming manner and also lack of various features. Parents get to know about the details of their ward only when college authorities inform them. Even the graphic user interface (GUI) is missing. The communication is in a single direction. When parents communicate with the college authorities, they cannot get complete details at a time, as the details will be in different departments. In this paper we proposed the system which overcome all the disadvantages. That is, Graphical user interface (a friendly interface) is provided to the user. A parent whenever needed can login and access the student record that is present in SQLite at the server end. He/she can enter the USN of their ward, with a password and check various fields in the record such as marks obtained, fees paid, final results, attendance, counselor's name etc. Thus they can keep track of their ward and all the necessary updates reach them, without waiting for the college authorities. This application also helps the lecturers/professors query their details such as salary, subjects handled, pass percentage, number of CL, EL etc. Along with this we have provided an audio alarm feature. This can be set as a reminder to all faculty and students to update them regarding their daily schedules.

Added feature in this application is the group messaging module. Using this service, all the users like lecturers of all departments, placement officer etc can be addressed in a single SMS and any circulars can be sent across instantly.

People who do not have Android devices may query all the student details via SMS service using the GSM manager.

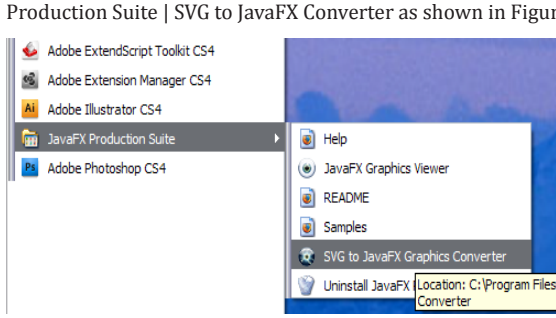
Graphic Design and JavaFX

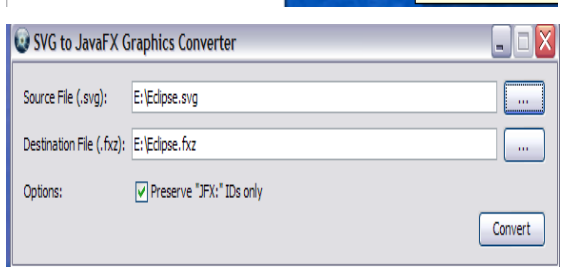
The GUI of mobile is designed by making use of JavaFX. JavaFX is partially a declarative language. Using a declarative language, a developer describes what needs to be done, then lets the system get it done. JavaFX Script blends declarative programming concepts with object orientation. This provides a highly productive, yet flexible and robust, foundation for applications. However, with this flexibility comes responsibility from the developer. JavaFX Script is a forgiving language and being declarative, it assumes inherent rules that may obscure a programming fault. The most obvious of these is that null objects are handled by the runtime engine and seldom cause a Java Null Pointer exception. As a result, the program will continue when a null is encountered within an expression, and will produce a valid result. However, the result may not have been what you expected. Therefore, the developer needs to be extra vigilant when writing code and more thorough when testing it. At first, this may seem alarming; however, this is offset by the ease of use and greater productivity of JavaFX and by the fact that JavaFX tries to mitigate the user from experiencing a crash. One of the benefits of JavaFX being a declarative language is that much of the "plumbing" to make objects interact is already provided within the language. This allows the developer to be able to concentrate more on what needs to display, and less on how to do it.

In a JavaFX environment, the goal for the graphic designer is to use his or her creativity to forge graphical assets and then export them for JavaFX in the form of JavaFX Objects. First, the graphic designer helps to design the visual presentation and then often generates his or her designs using various symbols, drawings, texts, images, colors, and special effects. After the graphic designer generates the graphical assets in the form of JavaFX Objects, it is up to the application developer to use those graphical objects in a Rich Internet Application. Typically, graphic designers use specialized design tools to develop graphical assets. Of these, Adobe Illustrator CS3 and Adobe Photoshop CS3 are the most popular. Another common graphical format is Scalable Vector Graphics, or SVG, and most graphic design programs provide SVG export capabilities. This chapter discusses the process that the graphic designer will need to follow to export his artwork to a form that can be used in JavaFX. Specifically, we describe the procedure to export graphical assets from Adobe Illustrator CS3 and Adobe Photoshop CS3. In addition, we discuss the SVG to JavaFX Convertor utility to convert Scalable Vector Graphic files to JavaFX.

Scalable Vector Graphics

Scalable Vector Graphics (SVG) is an open and free standard language for describing two-dimensional graphics using t he Extensible Markup Language (XML) developed under the World Wide Web Consortium (W3C) process (http:// www.w3.org/Graphics/SVG/). SVG provides for vector graphic shapes using paths of either lines or curves, images, and text. JavaFX Production Suite includes an SVG conversion utility that converts SVG graphic files to JavaFX Archive File format. To demonstrate, we saved the solar eclipse images we created previously in the Adobe Illustrator section to an SVG file, Eclipse.svg.To run the conversion utility in Windows, select Start | All Programs | JavaFX Production Suite | SVG to JavaFX Converter as shown in Figure.





SVG to JavaFX Converter

Volume : 2 | Issue : 11 | November 2013 • ISSN No 2277 - 8179

This opens the SVG Converter tool. This tool has an option to only export JFX: layer IDs (the default) or all IDs. Figure shows the SVG to JavaFX Graphics Converter window SVG to JavaFX Graphics Converter Window There are many modules in this application. Some of the modules are sensor manager module, threshold monitor module, contact manager module, SMS sending module, timer module, reply verification module and call manager module. The block diagram below shows the different modules and the relationship between them. First the accelerometer will be running in the background and when the value crosses the threshold the fall is detected. This activity comes under the sensor manager module and the threshold monitor module. In the timer module timer is started within which the user has to confirm whether the fall detected is a true alarm or a false one.

SERVER

Block diagram showing different modules (server)

The architecture of MoSQL decouples some of the components typically bundled together in onolithic databases. In particular, concurrency control and logging management are separate from data storage and access methods. In this sense our approach is similar to the model proposed expands upon the work. In brief,MoSQL is composed of three main components:

MySQL servers handle client requests by parsing SQL statements and executing them against the storage engine. MySQL allows third-party storage engines through its pluggable storage engine API. This allows us to plug in our calls to our distributed storage nodes.

Storage nodes handle MySQL requests and keep track of the transactional state. Each storage node keeps a subset of the dataset and indexes in-memory only. Storage nodes act as a distributed store in that read requests are either served from local memory or from the memory of a remote storage node. This is effective since retrieving rows from a remote storage node is usually faster than retrieving rows from local disk.

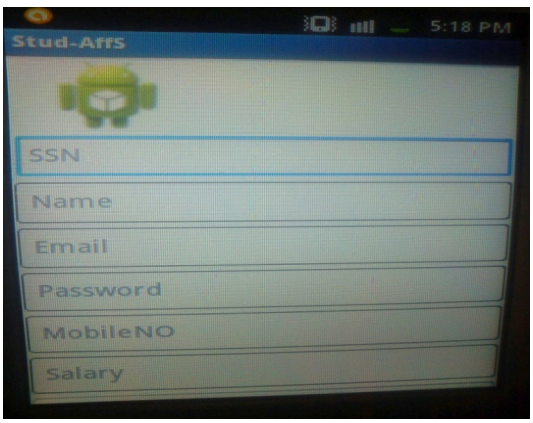
The **Certifier** is a replicated state machine that logs transactions on disk, ensures serializable transaction executions, and synchronizes system events (i.e., recovery, storage node additions and removals).

Messages: We assume that each transaction is self-contained, meaning that it performs its computation without any direct communication with other transactions. Transactions do communicate indirectly, of course, by storing and retrieving data in the database. However, this is the only way they can affect each other's execution.

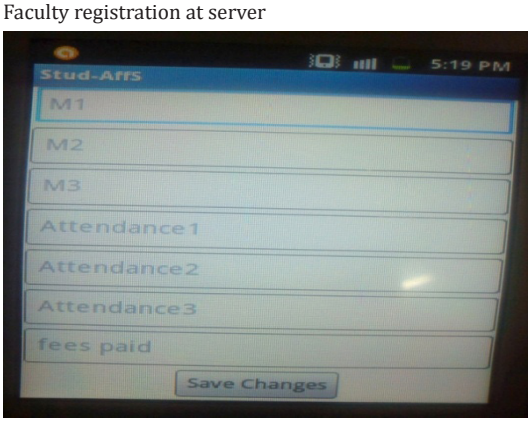
To ensure transaction atomicity, the DBS must control all of the ways that transactions interact. This means that the DBS must mediate each transaction's operations that can affect other transactions. In our model, the only such operations are accesses to shared data. Since a transaction accesses shared data by issuing database operations to the DBS, the DBS can control all such actions, as required.

In many systems, transactions are allowed to communicate by sending messages. We allow such message communication in our model, provided that those messages are stored in the database. A transaction sends or receives a message by writing or reading the data item that holds the message. This restriction on message communication only applies to messages between transactions. Two or more processes that are executing on behalf of the same transaction can freely exchange messages, and those messages need not be stored in the database. In general, a transaction is free to control its internal execution using any available mechanism. Only interactions between different transactions need to be controlled by the DBS.

Results:



Faculty registration at server



Student update at server

In the contact manager module the social contacts which are to be contacted in case of the fall detection is managed. In the SMS sending module the SMS is sent to the contact which contains the GPS coordinates and also the key.

When the reply is received by the broadcast receiver it is verified whether the reply message contains the key or not. This is done in the SMS verification module or reply verification module. In the last module that is the call manager module a call is made to the social contact who has replied and the speaker is turned on and the communication between the contact and the user takes place.

REFERENCE

[1] B. Fling, Mobile design and development. O'Reilly, 2009. | [2] Alexander Tomic1, Daniele Sciascia1, Fernando Pedone1 MoSQL: An Elastic Storage Engine for MySQL | [3] Jim Clarke Jim Connors Eric Bruno Java FX™ Developing Rich Internet Applications | [4]http://developer.android.com/training/index.html | [5] Concurrency Control and Recovery in Database Systems Philip A. Bernstein, Vassos Hadzilacos, Nathan Goodman | [6] Bringing Big Systems to Small Schools: Distributed Systems for Undergraduates Jeannie R. Albrecht Williams College Williamstown,