Distribution Information Management

Individual report

Adrián Gómez Llorente

0904327G

Introduction

In this document is explained the individual import of the coursework make for Distribution Information Management, to do this I am going to follow next chapters:

- Overview of the coursework
- Individual working
- How the software could be extended
- Marking work group

Overview of the coursework

The coursework consisted in developing a system to give students the possibility to create questionnaires in certain courses with some questions and several possible answers to respond. The students could also answer questionnaires created by other students.

The system was composed by 4 principal tasks:

- A website as an interface for the students to work with the system
- An application to provide some statistics of the system
- An application to backup the database into an XML file
- An application to update current database from XML file

At first all the group was working in every task, because at first we needed to stabilise some standards to work with them through all the project.

That's why all the group was working together in the database diagram and how all the applications were going to be distributed and how they were going to connect each other. In that moment we decided things like the programming languages and databases administration system to be used.

In conclusion we decided to use MySQL as a database administration system because is very easy to install and it works so good with PHP. PHP for the website because we were used to work with it and is easy to use as well. And Java for the rest of the applications because we are used to work with it, is easy to develop and is also cross-platform.

Individual working

Once we had all the standards defined we divide the project into small tasks to give each of them to each member of the group to work harder on it. My task was developing the application to update current database from a given XML file.

First of all I had to work with Victor Catalan to determine how the structure of the XML was going to be, to concrete this we write a previous DTD to work together to avoid as many errors as possible. Finally we got next DTD:

```
<!ELEMENT students ( student *) >
<!ELEMENT student ( courses*, answers* ) >
<!ATTLIST student id #REQUIRED matriculationID #REQUIRED name #REQUIRED password
#REQUIRED surname #REQUIRED>
<!ELEMENT courses ( course+ ) >
<!ELEMENT course ( questionnaire* ) >
<!ATTLIST course id #REQUIRED name #REQUIRED>
<!ELEMENT questionnaire ( question+ ) >
<!ATTLIST questionnaire courseID #REQUIRED id #REQUIRED name #REQUIRED studentID
#REQUIRED>
<!ELEMENT question ( choice+ ) >
<!ATTLIST question content #REQUIRED id #REQUIRED questionnaireID #REQUIRED>
<!ELEMENT choice EMPTY >
<!ATTLIST choice content #REQUIRED correct #REQUIRED id #REQUIRED questionID #REQUIRED>
<!ELEMENT answers ( answer+ ) >
<!ATTLIST answer choiceID #REQUIRED id #REQUIRED questionID #REQUIRED questionnaireID
#REQUIRED studentID #REQUIRED>
```

First we also generate an XML file from the DTD because we know it will be going to give us an better overview of we had to work with.

I had worked before with XML using Java and I think that using Java is so easy to work with XML files because it gives all the operations needed to parse the file and treat it as a tree. So I didn't need previous time to know how it worked. I started developing directly the application with the XML file created and applying fixes to the DTD to give the functionality needed.

Once the application was capable of parsing and inserting remotely the XML file into a backup database with the same structure we test the application with the XML file generated by Victor's application, and as unexpected it works at first, no changes were needed.

Finally we made a few tests checking that the database was correctly created and we create the jar files to execute them into the lab PC.

How the software could be extended

Right now the software provides all the required functionality but it might also include many other functionalities to improve the student experience, for example:

- It would be great to use the application through a mobile device, to get this it would be needed to change the CSS file to provide an interface for mobile devices and load the CSS file depending of the information given by the browser.
- It might be interesting to have more functionalities than only create questionnaires, for example, making the application more social with profiles for students to give the opportunity to the students to request, for example, through private messages more information about a questionnaire.
- The application might also offer a way to turn it into a complete course support system with videos from teachers, PDF with slides, etc. but this would convert this small application into a big project so a lot of design would be needed to perform it. That's why it might be more usefull to reuse for example some subsystems of this application like the XML parsers to adapt them to other systems as Moodle, for example.

Marking work group

I think that every member of my group has worked so hard and finally we have gotten the application working on time, moreover we haven't had problems between us so my opinion is that the correct way to give the 30 marks would be divide them into each of my four people group as can be seen next:

- Victor Catalán: 10 points, because we got the XML applications working at first.
- Omar Agudo: 10 points, because his statistical application gives more information that I expected at first and moreover it also provides information into plain text format that might be parsed by other applications, etc.
- Jesús Rodriguez: 10 points, because the website covers all the functionality and it is also using AJAX for the questionnaires creation.