

Web-Based Music School Manager

Project Report

Author: Aaron Redmond

Student ID: C00002396

Supervisor: Paul Barry

Contents

[1 Introduction 3](#_Toc289947944)

[2 Initial system, as planned 4](#_Toc289947945)

[2.1 Login 5](#_Toc289947946)

[2.2 Administration 6](#_Toc289947947)

[2.3 Teacher 8](#_Toc289947948)

[2.4 Student 10](#_Toc289947949)

[3 Actual system, as implemented 12](#_Toc289947950)

[3.1 Setup 13](#_Toc289947951)

[3.2 Login 14](#_Toc289947952)

[3.3 Administrator 15](#_Toc289947953)

[3.4 Teacher 16](#_Toc289947954)

[3.5 Student 17](#_Toc289947955)

[4 Conformance to Specification 18](#_Toc289947956)

[4.1 Setup 18](#_Toc289947957)

[4.2 Login 18](#_Toc289947958)

[4.3 Administrator 18](#_Toc289947959)

[4.4 Teacher 18](#_Toc289947960)

[4.5 Student 18](#_Toc289947961)

[5 Major challenges 19](#_Toc289947962)

[5.1 Genetic Algorithms 19](#_Toc289947963)

[5.2 The Curve 19](#_Toc289947964)

[5.2.1 Zend Framework/PHP 20](#_Toc289947965)

[5.2.2 MVC 20](#_Toc289947966)

[5.2.3 Doctrine 21](#_Toc289947967)

[6 Personal Appraisal 22](#_Toc289947968)

[7 If I was to start again 23](#_Toc289947969)

[8 Conclusion 24](#_Toc289947970)

[9 Acknowledgements 25](#_Toc289947971)

# Introduction

My project, the web based music school manager is a web based timetabling application which allows the administrator of a small music school, to easily manage the day to day activities of their school. Activities such as timetabling can be very tedious and time consuming; this web application hopes to simplify this task.

In the following sections, I will discuss the project under the following headings:

* The initial system, as planned and outlined in the design document.
* The actual system, as implemented.
* Conformance to specification, this is where I compare the actual idea to the initial idea.
* Major challenges, where I will discuss the difficulties I encountered along the way.
* Personal appraisal, this is where I give my opinion on the project.
* If I was to start again, knowing what I know now, what would I do differently?

# Initial system, as planned

The potential users of this system were identified as small to medium music school, which are generally run by a small group of people, usually it’s one of these people that are responsible for the complete administration side of the business. As per the design manual, the aim of this project was to build an easy to use piece of software, which would help reduce the time spent on the administration and timetabling tasks within one of these music schools, by simplifying some of the everyday tasks such as data entry and timetable creation.

The initial idea for this project was to design a system comprising of four main sections: login, administration, teacher and student. These sections would be joined together using the homepage, it is here that the user would log into the section of the system associated with their user type.

We will now look at each of these sections.

## Login



Figure : Login options as detailed in the functional specification

This section should deal with all the details and functionality around user registration and log in. A user logs into the appropriate section via this page, and can access both the register new account and request username\password pages from here.

As you can see from the diagram above, from the initial login page a user should be able to:

* Register a new account
* Request Login detail
	+ If the user has forgotten their username or password, they can request their detail via email from this page.
* Login as one of the following:
	+ Administrator
	+ Student
	+ Teacher



Figure : Initial Login Page (Prototype)

## Administration



Figure : Administration options as detailed in functional specification

This section is involved with implementing the administration side of the application. The administration section is by far the largest section of the project and should contain the most functionality.

Within the administration section, an admin user would be able to access the following functionality:

* Manage
* Timetabling
* User Panel

The manage section allows the user to manage all the school requirements such as students, teachers, rooms, etc. As you can see from the diagram above, it consists of many sections, each of which can be broken down again into edit/add pages.

The Timetabling section allows the user to edit the timetabling information, and generate the timetables.

The User Panel is a catch all which incorporates the following: system setup which allows the user to setup various aspects of the system not covered in the manage section; such as instruments, personalisation and news; Account Information Section simply allows the user to change their username and password; and Log out.

A Help section is also available which will display the help for the administration section.



Figure : Administrator home (Prototype)

## Teacher



Figure : Teacher options as detailed in functional specification

The teacher section should allow a registered teacher to log in, check their timetable and change their personal account settings such as username and password. Also when a teacher logs into the system they should be greeted by the latest news articles.

If a user logs in as a teacher, they should have the following options available to them.

* Timetable
	+ From here a teacher can view their individual timetable.
* Tools
	+ Lesson Attendance
		- From here a teacher can input the attendance records for their lessons.
* My Account
	+ Account Information
		- Displays the teacher’s information, as entered by the administrator.
	+ Log Out
		- If the user selects this option; it will log them out of the system and return them to the initial home page.
* Help
	+ Displays system help, possibly page specific.



Figure : Teacher home (Prototype)

## Student



Figure : Student options as detailed in functional specification

The student section is similar to the teacher section, in that a registered student can log in, check their timetables or change their personal account settings such as username and password. Also when a student logs into the system they should be greeted by the latest news articles.

When a student logs in, they should have the following options available to them.

* Timetable
	+ From here, the student can view their individual timetable.
* My Account
	+ Account Information
		- Displays the student’s information, as entered by the administrator.
	+ Log Out
		- If the user selects this option; it will log them out of the system and return them to the initial home page.
* Help
	+ Displays system help, possibly page specific.

For more in depth information on these sections please refer to the following documents:

* Design Document
* Functionality Specification

These sections are also fully prototyped and available via the projects website.

#  Actual system, as implemented

The final system which was produced comprises of five sections:

* Setup
* Login
* Administrator
* Teacher
* Student

With the Administrator, Teacher and Student sections being joined via the Login page.



Figure : Basic system navigation map

 I will now look at each of these sections individually.

## Setup



Figure : Steps involved in the setup process

This is a two step installation process. The first step is to setup the database, once this has been setup correctly you can move onto step 2, which is to setup the basic school information such as school name, address, etc.

Once these steps have been successfully carried out, you should be directed to the login page, where you can login and start the administration process.



Figure : Step 1 - Database setup



Figure : Step 2 - School Setup

## Login



Figure : Login Screen

From the login screen a registered user can log into the section of the site associated with their user type. If a previously registered user has forgotten their username or password they can request that their login detail be sent to them, this can be done by clicking the “Forgot your Username/Password” link and entering your email address, the system will them generate a new password which it will mail to them along with their existing username.

## Administrator



Figure : Administration Home

The administration section is comprised of the following sections:

* Manage
* Timetabling
* User Panel

The Manage section allows the administrator to manage all aspects of the music school, from teachers to students, subjects to lessons. This section is really all about setting up the data needed for the timetabling section.

The timetabling section allows the administrator to generate timetables and edit the timetable information.

The User Panel section allows the administrator to edit the instruments used by the school, personalise the website by changing the banner or edit the news articles which will appear on the login page, teacher home and student home, it also allows the administrator to change their account settings such as username and password.

## Teacher



Figure : Teacher Home

When a teacher logs into the system they will be greeted with the latest news articles. The teacher can then view their timetable by clicking on the timetable link. They can also change account setting such as username and password from the My Account section.

## Student



Figure : Student Home

When a student logs into the system they will be also greeted with the latest news articles. The student can then view their timetable by clicking on the timetable link, unlike a teacher; a student may have more than one timetable depending on how many groups they are in. They can also change account setting such as username and password from the My Account section.

All sections also have a Logout function which logs the current user out of the system and returns them to the login page, a help section which displays the appropriate help page for the logged in user type and a home function which returns the users to their main logged in starting page.

# Conformance to Specification

I will approach this section by outlining each section described in the above section 3, and comparing them to the sections described in section 2.

## Setup

This is something that I never initially considered when designing the application. I think this is because that being completely new to web design I never considered how a website/web application actually got deployed.

## Login

This section now has less functionality than initially outlined; originally the idea was to allow a user (teacher or student) to register on the site. This idea was abandoned because it is of no use allowing just anybody to send a registration request which would then have to be manually processed by the administrator before the user could actually login. Also if we look at the example of how IT Carlow‘s system works, you will notice that you do not create your own account, you are assigned an account.

This is the only major difference between the initial idea and the actual idea, otherwise the system works as specified allow a user to log in or request their username/password.

##  Administrator

There is virtually no difference between the initial idea as described in section 2 and the actual idea described in section 3. The administration section is divided into several sections as described above and in the design documentation and each section works as initially envisaged.

## Teacher

The only difference between the initial system described in section 2 and the actual system as described in section 3 is the exclusion of the Tools->Lesson Attendance section.

Other than this exclusion there is virtually no difference between the initial system as described in section 2 and the actual system described in section 3. The teacher section works as initially envisaged.

## Student

There is virtually no difference between the initial system as described in section 2 and the actual system described in section 3. The student section works as initially envisaged.

# Major challenges

In this section I will discuss the challenges I encountered while working on this project. The main challenges I will be discussing are:

* Genetic Algorithms
* The Curve

## Genetic Algorithms

The first big challenge was when deciding how I was going to implement the timetabling process. Initially I was looking at on open source Genetic Algorithm (GA) program written in visual C++. This program took its input from a text file which defined the teachers, groups, classes and subjects. It that used a genetic algorithm to produce a timetable by taking the input file and using a GA to produce several timetable generations before ending up with an optimal timetable. The main problem with this application was the fact that it was a only windows executable and did not accept any console input and for me to utilise it within my system I would have to either a) produce a console version or b) wrap the C++ classes into a PHP extension.

The main problem with solution was that my C++ skills are very weak and the GA algorithm is quite complex and actually recoding the application to produce a console application would be quite time consuming with no guarantees that the finished would perform as expected.

The problems with solution b were many fold, the main one being that for every C++ class you would have to write an equivalent PHP class, again I run into the same problem I was having with solution a, I would need an in depth knowledge of the application, but this time there was even less of a guarantee that it would work.

The good point to come out of this challenge was that I discovered how good the FET application was.

If you wish to read more on this subject, please see section 6 – Appendix A of the Design Manual.

## The Curve

What I mean when I say the curve is the learning curve on the project for me. This learning curve started off very steep and it really only started to level off in the last few weeks of the project.

For me the curve comprises of:

* Zend Framework
* PHP/Javascript
* Doctrine
* MVC

### Zend Framework/PHP

The Zend framework can be very daunting as first glance. It is based on the eclipse framework which I have had no experience with and it can take quite a while to learn your way around the development environment. It is a PHP based framework, a language which I have had no exposure to. I found that I actually wrote more code in the last three weeks, as opposed to the several weeks before; I think this is because obviously starting off not knowing the language the learning curve was rather high and as I started to code and learn the language with respect to the Zend framework, which has many of its own classes and methods to master, the curve became gentler and the productivity higher.



Figure : Zend Framework

### MVC

The Model View Controller architecture coupled with the Zend framework can be very confusing at first, as you try unsuccessfully to figure out where all the pieces fit. To me the MVC pattern as implemented within the Zend framework is a bit like a jigsaw, you start off with a lot of individual pieces that you do not know where they go, but after a while you begin to see the big picture and where each piece fits. Once you get use to the Zend MC, you really begin to appreciate it and how efficient it is, it becomes easy to know where to look for certain code if you have a bug and it also helps keep you code organised.

### Doctrine

The Doctrine ORM (Object Relation Mapper) is a set of PHP libraries which provides persistence services and related functionality, which when used in conjunction with the Zend architecture provides powerful database functionality. One of the greatest things about Doctirne is the fact that it allows you to easily change the type of database you are using, you can change from MySQL to PostgreSQL very easily. While Doctrine was certainly on the lower end of the curve for me it certainly threw a few issues my way, the main one being while setting up table relations, the lessons table being a good example of this, as it basically pulls 80% of its information from other tables.



# Personal Appraisal

I am very happy with the final version of the application I have produced, considering that at the start the project I knew next to nothing about developing web applications, PHP, Doctrine or the MVC, as this is the first ever web application I have worked on from inception through to completion. The application performs as expected and for the most part conforms to both the design and functional specifications, with the exception of the setup section and the changes made to the login.

The one thing that I am a little disappointed in is how it looks; it does not look as professional as most of the modern day web sites/application out there today. Even though this is a negative for me I think it stresses the fact that the production of a web application or any application for that matter is no longer a solitarily pursuit, as while I am proud to call myself a software engineer/developer, I am certainly no graphic designer.

Something I am very happy with is the wireframe prototypes I developed as part of the design. These helped me immensely and taught me the value of good prototyping. With these prototypes, before I ever started developing I could see the layout and planning of what I had to achieve, and when it came time to develop, these prototypes were an invaluable guide.

With regards to timetabling and the whole timetabling process, I discovered that this problem is one which is still being researched to this day and generally seems to be a thesis level subject, with Genetic Algorithms being one of the more popular methods.

Looking back on the design document, my mindset as far as the design was concerned was centred on a desktop application as this is the area where I have the most experience. What I mean by this is if you look at the prototypes, they are designed to act like a similar desktop application might. When it came time to implement my design I realised that there is a big difference between designing a web application and designing a desktop application. A good example of this would be the save timetable feature, which is described in the design document as:

When the user clicks the save button, the following processes should take place:

* The system presents a Save As dialog box.
* The user selects the type and the system saves the timetable in that format.

 However as I found out something like this is not as easily done in a web application, as it would be in a similar desktop environment, especially on windows which has native support for this type of feature.

# If I was to start again

If I was to start again, what would I do differently? I would definitely make better use of the Zend MVC pattern. While I did use the MVC, I could have used it to logically separate my code better, especially the manage section which is rather large. The View part of my Manage section contains 49 objects with 26 associated forms; I could definitely have separated these more.

I would also like to have tried my hand at creating a cloud based application, as I really think that this is the future, especially for this type of application.

Were I to start coding again right now, but keeping the knowledge I have learned, I would like to be able to implement the following:

* Implement an error queue, so that if an email fails to send it is added to that queue and retried again at a set interval and only removed from the queue after a set number of attempts.
* Implement pagination, as once there are a lot of students and teachers, the list would get quite long.
* Learn and incorporate some CSS design.
* Add some more teacher functionality, such as grading students.
* Improve the constraints systems, as the system currently only handles teacher time constraints, group time constraints and room constraints.
* Improve the save functionality for the timetabling feature.

# Conclusion

In conclusion, I think that this has been a great experience for me. Coming from a work environment where the main focus of my work was localisation, reengineering and desktop applications, it was exciting to work with and learn so many new technologies. As far as the project is concerned I am very happy with the final product, as I think I have achieved what I set out to do.

I think I can happily take what I have learned from this project and take it forward with me into a new work environment, where I can enhance and build upon the skill I have learned while working on this project.

Excluding the technologies I worked with on the project, compiling the research report opened my eyes to all the brilliant technologies out there, from cloud technologies such as Google App Engine and Windows Azure to web application frameworks such as Django and Ruby on Rails.

Regardless of the outcome, I can say without a doubt that I enjoyed this project experience.

# Acknowledgements

I would like to thanks the following people:

* Mr. Paul Barry, my project supervisor for keeping things real and making web development interesting.
* Illya for his Zend tutorials.
* Liviu Lalescu for FET.
* Mary for the many many cups of tea.